Springer Series in Statistics

Peter Filzmoser · Karel Hron
Matthias Templ

Applied Compositional Data Analysis

With Worked Examples in R

Springer

Springer Series in Statistics

More information about this series at http://www.springer.com/series/692

Peter Filzmoser • Karel Hron • Matthias Templ

Applied Compositional Data Analysis

With Worked Examples in R

Springer

Peter Filzmoser
Institute of Statistics and Mathematical Methods in Economics
TU Wien
Vienna, Austria

Karel Hron
Department of Mathematical Analysis and Applications of Mathematics
Palacký University Olomouc
Olomouc, Czech Republic

Matthias Templ
Institute of Data Analysis and Process Design
ZHAW Zurich University of Applied Sciences
Winterthur, Switzerland

ISSN 0172-7397 ISSN 2197-568X (electronic)
Springer Series in Statistics
ISBN 978-3-319-96420-1 ISBN 978-3-319-96422-5 (eBook)
https://doi.org/10.1007/978-3-319-96422-5

Library of Congress Control Number: 2018952636

Mathematics Subject Classification (2010): 62H25, 62H30, 62H20, 62J05, 15A03, 62P12, 62P25

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To our families

Preface

Compositional data are nowadays widely accepted as multivariate observations carrying relative information: those following the principle of scale invariance, typically being represented in proportions and percentages, but also in other units like mg/kg and mg/l that reflect their relative nature. In other words, for compositional data the relevant information is contained in the (log-)ratios between the components (parts). In 2006, 20 years after the seminal book of John Aitchison, *The statistical analysis of compositional data*, has been published, we met compositional data and the logratio methodology for the first time—to be honest, not as something highly appealing, but originally for the reason to get a research paper finally accepted for publication, after a tedious reviewing process. We were not fully convinced that this approach would be so important for practical applications, because at that time the methodology was presented more from a theoretical perspective, and the applications were partially even based on invented data. On the other hand, it was clear that the logratio methodology formed a consistent approach to deal with this type of data, and further interesting directions were proposed: the paper on orthonormal coordinates for compositional data [Egozcue, J.J., Pawlowsky-Glahn, V., Mateu-Figueras, G., Barceló-Vidal, C. *Isometric logratio transformation for compositional data analysis* in *Mathematical Geology*] was published just 3 years before, and also the principle of working in coordinates was just born.

When working more and more in this area, we felt at some point that there could be a need for a practical guide to compositional data analysis—not just for people from applications, but also for our own curiosity, to understand which value added the logratio methodology could yield when processing compositional data. How do the results differ when simply taking a log-transformation, compared to working in an appropriate geometry? And are the results (more) reasonable and justified? In the last ten or more years, we did quite an effort in this direction, by touching systematically almost all popular multivariate statistical methods and those fields that are of primary importance for practical data analysis (robust statistics, outlier detection, and dealing with missing and zero values).

This book provides a summary of our efforts. We wrote it in a great freedom from what should be followed or mentioned from historical or any other reasons.

The focus is on a proper orthonormal coordinate representation of compositional data that indeed provides a useful way for a reasonable processing of multivariate observations. The central point are so-called *pivot coordinates* that aim to extract all relative information about one of the parts in a composition. These coordinates have proven their advances in a number of applications and provoked many discussions. We present the pivot coordinates in a form that shows their flexibility in various data processing contexts and their strength for the interpretation of the results. Nevertheless, we admit that also other representations, like more general orthonormal coordinates, balances, but also centered logratio coefficients, or pairwise logratios, are useful in concrete contexts.

The book can be taken as a concise, self-contained manual on how to apply the logratio methodology for compositional data analysis in everyday practice, using the statistical software environment R and the package `robCompositions`. We tried to illustrate the theoretical parts with several examples from applications with general understandability, like those from official statistics, economics, geology, or chemometrics. As a minimum prerequisite for accessing the book, just a basic course on probability and statistics is required, although additional experience with multivariate statistics and statistical computing might be advisable. On the other hand, the book can also be considered as a source of inspiration for those who are familiar enough with standard knowledge on compositional data analysis, as presented in the book by V. Pawlowsky-Glahn, J.J. Egozcue, and R. Tolosana-Delgado, *Modeling and analysis of compositional data*. According to these aims, after providing the geometrical reasoning for a relevant (not exclusively statistical) processing of compositional data, many popular statistical methods, like principal component analysis, cluster analysis, classification and regression analysis, are adapted for dealing with data carrying relative information. Moreover, exploratory and preprocessing issues are discussed: visualization, outlier detection, and dealing with missing values and particularly with zeros that form a touchstone of the logratio analysis. Last but not least, also emerging fields like analyzing high-dimensional compositional data and compositional tables, with great potential for future developments, are discussed. This clearly illustrates that not a closed methodological framework but rather just a state of the art of an intensively developing research field is presented.

Finally, the structure of the book can also be used for a one-semester course on applied compositional data analysis. The interactive form of the book enables students to practice theoretical knowledge directly with data sets coming from different fields of their possible future expertise. Our sincere wish is to contribute to the education of a new generation of people for which statistical analysis of compositional data is a matter of creative thinking.

Vienna, Austria — Peter Filzmoser
Olomouc, Czech Republic — Karel Hron
Winterthur, Switzerland — Matthias Templ
August 25, 2018

Acknowledgments

We are very grateful to our colleagues from the Vienna University of Technology and from the Palacký University Olomouc and to many collaborators who helped us to get familiar with compositional data analysis. In particular, we like to thank Dr. Clemens Reimann from the Geological Survey of Norway, for bringing us in touch with real data applications from geochemistry, which made it necessary at some point to get acquainted with compositional data. We are grateful to Prof. Vera Pawlowsky-Glahn from the University of Girona and to Prof. Juan José Egozcue from the Polytechnic University of Catalonia for numerous fruitful discussions—they are really "parents" of compositional data analysis. Their hints and ideas greatly helped to write the book in this present form. And, finally, our greatest gratitude is to our families: without their long-term support, any research activities would by far not be possible.

Contents

Acronyms

AIC	Akaike information criterion
BIC	Bayesian information criterion
BS	Backward stepwise
CRAN	Comprehensive R Archive Network
FS	Forward stepwise
GUI	Point and click graphical user interface
LDA	Linear discriminant analysis
LS	Least squares
MCD	Minimum covariance estimator
PCA	Principal component analysis
QDA	Quadratic discriminant analysis
RSS	Residual sum of squares
SBP	Sequential binary partitioning
SVD	Singular value decomposition
TLS	Total least squares

Chapter 1
Compositional Data as a Methodological Concept

Abstract Compositional data were defined traditionally as constrained data, like proportions or percentages, with a fixed constant sum constraint (1 or 100, respectively). Nevertheless, from a practical perspective it is much more intuitive to consider them as observations carrying relative information, where proportions stand just for one possible representation. Equivalently, all relevant information in compositional data is contained in ratios between components (parts). According to this broader definition, the decision whether the data at hand are compositional or not depends primarily on the purpose of the analysis, i.e. if the relative structure of the compositional parts is of interest or not. As a consequence, the use of standard statistical methods for the analysis of compositional data that obey specific geometrical properties leads inevitably to biased results. A reasonable way out is to set up an algebraic-geometrical structure that follows the principles of compositional data analysis (scale invariance, permutation invariance, and subcompositional coherence). Nowadays, this is called the Aitchison geometry and it enables to express compositional data in interpretable real coordinates, where standard statistical procedures can directly be applied. These coordinates are formed by logratios of pairs of compositional parts and their aggregations: the logratio methodology was born.

1.1 What Are Compositional Data?

People who already have a rough idea about compositional data will probably not have any doubts to answer the above question. Sure, compositional data consist of multivariate observations with positive values that sum up to a constant! Well, examples are proportional data or percentages, for which the values of an observation sum up to 1 or 100. However, does this still hold if one variable of these multivariate data is not available or has not been measured? Or what if rounding errors are present in one variable such that the sum does not exactly meet the prescribed constraint? Or what happens if the sum is not constant at all, but very different for different observations in the data set, although the units clearly indicate their "compositional" character?

P. Filzmoser et al., *Applied Compositional Data Analysis*, Springer Series in Statistics, https://doi.org/10.1007/978-3-319-96422-5_1

The aim of this book is not to introduce compositional data as such observations, characterized by any *constraint* on the sum of components that either naturally occurs or is set more or less artificially. Rather, throughout the book, compositional data are treated as multivariate observations where *relative* rather than *absolute* information is relevant for the analysis. A decision of the analyst whether one deals with compositional data or not might be based on the units in which the samples are measured, but also on the purpose of the analysis and on the goals of the study.

Absolute information: refers to data where the difference (in the sense of "minus") makes sense, i.e. data which can be analyzed using usual operations in real Euclidean space. In other words, using the operations we learned in school. By any rescaling of the original raw data from their given units, such as counts, monetary units, weight, height, to any other units, like to percentages, their informative value would be affected, or even lost.

Relative information: refers to a representation of quantitatively described contributions on a whole. Information about the total amount itself is irrelevant. The data units are typically proportions or percentages. Nevertheless, from the essence of the problem also concentrations of chemical elements in parts per million (ppm), mg/kg or mg/l, as well as household expenditures to commodities like foodstuff, housing, transport and communications in EUR are candidates for observations carrying relative information. Note that in cases of proportional and percentage representations or units like ppm and mg/kg a constant sum constraint is implicitly imposed, though it must not necessarily be fulfilled in practice. This happens frequently for units like mg/kg, if not all chemical elements are measured–but also then the components clearly express relative contributions on a whole. In case of mg/l, a prescribed sum constraint is even not present, yet the relative structure of the components is clearly indicated. Considering relative information in the household expenditures case indicates that not wealth of households (given by concrete amounts of EUR spent on commodities), but rather the distribution of the total income into the given categories is of primary interest.

If relative information is being analyzed, it is irrelevant whether raw data, proportions, or percentage data are used as input for the analysis: the ratios between the components remain unaltered. Therefore, also the sum of these multivariate observations, which can even vary within the data set, is irrelevant.

As indicated above, a better question in this context thus might be: "Which type of information are you interested in?" Thinking about geochemical data, it might be interesting to look also at the absolute values of element concentrations in order to identify locations that exceed a certain threshold or action level. On the other hand, chemical processes might be better characterized by analyzing the relative information of the composition.

In that sense, data sets may be compositional or not at the same time–depending on the underlying question to be answered. However, for many data sets, the measurement units already indicate the relative nature of the data. Units like mg/kg,

ppm, mg/l, etc., refer to a "whole" which is used as a reference, and the values are reported with respect to this reference.

Compositional data thus quantitatively express relative contributions of variables under consideration of a certain whole, which carry relative information between the components (Egozcue 2009). Or, a more recent definition from Pawlowsky-Glahn et al. (2015) states that *compositional data are vectors with strictly positive components that carry relative information.* Equivalently, the relevant information is contained in ratios between the components. To be honest, in the previous definitions "*exclusively/only* relative information" was stated originally. Nevertheless, practical experiences show that this must not necessarily be the case: as it was presented above, even with compositional data one might be interested in absolute values of components, but they naturally contain also relative information.

This leads to the question in which cases it is preferable to analyze the relative information conveyed by compositional data. The following example gives a more closer look using an artificial data set, representing selected monthly household expenditures in EUR (Table 1.1). These data contain only some specific expenditure groups, and other possible expenditures are not reported. Variables like *health* or *clothing* which can form significant monthly expenditures as well are not available. So, expressing these data in terms of the total expenditures is not possible. One can, however, express the data relative to the sum of the four reported expenditures. This gives the percentage data, which in this case are identical for the three observations. Both the original values and their percentage representations stand for contributions of single items to the overall expenses of these four components. However, the data reported in EUR show a clear difference in the amount of expenditures of the three observations, while the percentage data are the same. Relative information could now be represented by the percentage data, resulting in four numbers. On the other hand, relative information may refer to the ratios between the components. For example, one gets for the ratio *housing*/*transport* of the three observations the value $1710/570 = 540/180 = 900/300 = 3$, i.e., all three households spend three times as much for *foodstuff* than for *transport*. Computing these ratios from the percentage data gives exactly the same value, and reporting the data for the observations in different currencies would also not alter the ratios. Overall, there are $\binom{4}{2} = 6$ ratios, up to their reciprocals, which form this representation of relative information. One can see that ratios contain much more detailed information than just percentages

Table 1.1 Artificial data set: household expenditures for three observations, expressed in EUR and in percentages (which are the same for all three observations)

Type	Observation	*Housing*	*Foodstuff*	*Transport*	*Communications*	Sum
Absolute	1	1710	950	570	570	3800
information	2	540	300	180	180	1200
in EUR	3	900	500	300	300	2000
Information	1	45	25	15	15	100
expressed	2	45	25	15	15	100
in %	3	45	25	15	15	100

to the total, and they remain the same if the data are rescaled. Ratios will thus form the representation of relative information that is considered in compositional data analysis.

Note that from another perspective, the absolute information might be preferable if, for instance, the goal of the analysis is an investigation of the wealth level in households, resulting in higher expenditures on the mentioned items. Finally, one might be interested also in a combination of the previous two settings, i.e., in the relative structure of the household expenditures by considering the total (sum of the variables) simultaneously. Therefore, it fully depends on the analyst, what kind of information should be extracted from the measurements. In line with that, the sample space of observations, as well as its structure, need to be specified, so that the different analyses are compatible.

Considering all pairwise ratios between the available variables as the basic input information for a new methodology might still not be the final best option. The reason is the asymmetric behavior of the ratios. Any ratio can take a value from the interval $(0, +\infty)$, where 1 means a perfect balance between both compositional variables, like for *transport* and *communications* in Table 1.1. Thus, the whole interval $(1, +\infty)$ corresponds to a variable that "dominates" another one. For the interval $(0, 1)$ the variable in the denominator is dominating, like the case of *transport*/*housing* $= 1/3$. In order to symmetrize the interpretation of ratios, the first choice is to use logarithms for the following reasons. The range of *logratios* (=logarithm of ratios) is the real line from $-\infty$ to $+\infty$, where the balance is represented by 0. For both possibilities, when one variable dominates the other one, a half line, $(-\infty, 0)$ and $(0, +\infty)$, is reserved. Logratios and their reciprocals differ just up to the sign. Logratios are also easier to handle from a mathematical point of view, because the logratio of two variables can be expressed as difference between their logarithms. When all logratios between components are known, any representation of the original compositional variables can be derived and vice versa. Such representations are further discussed in Chap. 3.

The use of logratios to characterize compositional data has several direct and indirect implications. The foremost important one is that zero values in components lead to problems since a logratio with a zero in the denominator is infinity and a zero in the numerator will result in minus infinity (by considering the extended real line). This is the reason that compositional data were defined as *positive* vectors and any zero components are subject to a special treatment as discussed further in Chap. 13. Although excluding zeros from the definition of compositions seems to be quite a serious handicap and indeed results in some complications when dealing with real world compositional data, it is compensated by a number of advantages that logratios provide to a statistical processing of compositions.

Applying standard statistical methods to compositions can lead to several problems, resulting from ignoring their underlying sample space. Since compositional data are strictly positive, a resulting negative value is not a valid solution. Even stronger, if the data are expressed as proportions, the results need to keep the range $(0, 1)$. In particular, confidence intervals computed with the proportional data could easily lead to intervals exceeding the range $(0, 1)$. But even if this is not the case, the results could be biased and lead to wrong conclusions. One should also not forget

that the results of any reasonable statistical processing should be invariant to an arbitrary rescaling of the original observations. Some initial problems are introduced in the next section.

The beginning of a systematic interest in compositional data analysis dates back to the end of the nineteenth century, namely to a famous paper by Pearson (1897). In this paper on *spurious correlations* he pointed out problems with correlation analysis of relative data, i.e. when ratios form the source of relevant information. Almost during the whole twentieth century, the developments in compositional data analysis were devoted either to building specific statistical models to analyze *proportional data*, i.e., a particular representation of compositions with constant sum equal to 1, or to cope with restrictions resulting from their direct statistical processing, particularly in the field of geosciences (Chayes 1960). Eventually, in the early 1980s, the Scottish mathematician John Aitchison introduced the *logratio methodology* to the statistical analysis of compositional data (Aitchison 1986). The aim was to define a family of logratio transformations, formed by pairwise logratios or their proper aggregation into new variables, to move compositional data from their original sample space to an unrestricted real space, where standard statistical methods can be applied for their further analysis. Hereafter, also a specific wording was introduced, like *parts* instead of *variables* or *components*, which will also be used in this book. It is worth to note that J. Aitchison identified compositional data in the above sense with proportional data, where the aim was to keep the prescribed sum constraint. During the following years, by a number of discussions in journals like Mathematical Geology, J. Aitchison and the research group formed around him realized that this methodology is capable to extend the definition of compositional data so that the constant sum constraint does not play any role for the analysis itself and can be stored purely for the purpose of interpretation. These thoughts were closely related to the introduction of the vector space structure of compositional data (Billheimer et al. 2001; Pawlowsky-Glahn and Egozcue 2001), named as the Aitchison geometry. This algebraic-geometrical structure of compositions made it possible to consider logratio transformations as coordinates with respect to a basis, or a generating system, that ease further theoretical developments and enhance interpretation of the results. This approach, followed in recent books on compositional data (Buccianti et al. 2011; van den Boogaart and Tolosana-Delgado 2013; Pawlowsky-Glahn et al. 2015), is applied also here. Its consequences are thoroughly discussed in the following chapters.

1.2 Introductory Problems

1.2.1 PhD Students Example

Table 1.2 shows a table of absolute numbers of PhD students in several countries of Europe, Japan, and the US. The data are available from Eurostat, http://ec.europa.eu/eurostat/. The student numbers are reported for different study groups.

Table 1.2 PhD students in Europe, Japan, and the US based on the standard classification system, split by different kinds of studies

	Total	Male	Female	Technical	Soc-eco-law	Human	Health	Agriculture
BE	7500	59.0	41.0	3462	1469	997	1041	532
BG	5200	49.7	50.3	2064	1102	1170	666	198
CZ	22,600	62.1	37.9	10,668	3748	3518	3633	1035
DK	4800	54.2	45.8	1886	614	696	1210	394
EE	2000	46.5	53.5	847	424	420	196	112
IE	5100	52.1	47.9	2633	787	1124	450	107
GR	22,500	55.6	44.4	12,590	3941	5090	495	383
ES	77,100	49.0	51.0	19,751	20,704	18,885	16026	1733
FR	69,800	53.9	46.1	27,152	21,429	18,846	2303	70
IT	38,300	48.3	51.7	16,403	7621	5803	6035	2437
LV	1800	39.6	60.4	542	603	434	182	40
LT	2900	43.4	56.6	1183	916	400	293	107
HU	8000	53.0	47.0	2576	1648	1992	1304	480
AT	16,800	54.3	45.7	4978	6374	4103	790	555
PL	32,700	50.7	49.3	10,202	7881	9974	3008	1635
PT	20,500	44.0	56.0	6027	6191	4879	3034	369
RO	21,700	51.7	48.3	6864	3801	3323	6017	1694
SI	1100	53.5	46.5	526	174	189	168	43
SK	10,700	57.1	42.9	4220	2121	1971	2024	364
FI	22,100	48.4	51.6	8875	4990	5365	2406	464
SE	21,400	51.3	48.7	8872	2651	2694	6756	428
UK	94,200	55.4	44.6	38,266	19,747	20,408	14,456	1323
CR	1300	53.3	46.7	601	94	286	235	84
TK	32,600	60.6	39.4	10,888	7922	7335	3814	2641
NO	5000	53.6	46.4	2055	870	635	1220	220
CH	17,200	59.7	40.3	6849	4537	2691	2640	483
JP	75,000	70.3	29.7	25,255	10,102	10,408	24,796	4439
US	388,700	48.2	51.8	117,658	104,456	94,748	68,731	3106

Source: Eurostat, http://ec.europa.eu/eurostat/, © European Union, 1995–2018

A scatterplot of two variables of Table 1.2—the absolute number of PhD students in natural and technical sciences and the absolute number of PhD students in health and life sciences—is displayed in Fig. 1.1a. A classical correlation measure would report high positive correlation, especially because of the large absolute values for USA. This "outlier" will very likely also dominate other statistical methods and lead to biased results. For this reason, a first attempt could be the use of a logarithmic scale for both variables, as shown in Fig. 1.1b. The joint data distribution now seems to be close to a bivariate normal distribution, and still a positive relationship between the variables is visible.

One could also convert this information to percentages, i.e. divide the values regarding the five studies in Table 1.2 by their sum and multiply by 100. The result

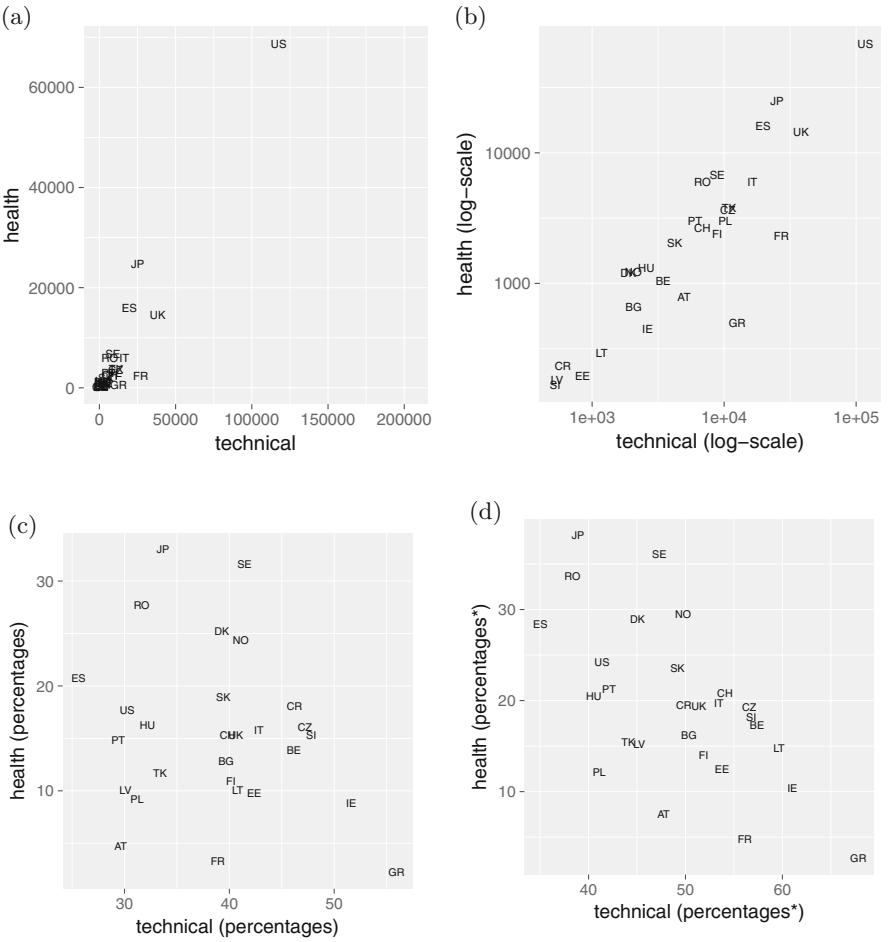


Fig. 1.1 PhD students in Europe, Japan, and the US; natural and technical sciences versus health and life sciences. (**a**) Absolute values. (**b**) Log-transformed data. (**c**) Percentages considering all variables. (**d**) Percentages* without socio-economic and law sciences

for the two considered variables is shown in Fig. 1.1c, and the strong relationship observed before is no longer visible here. The negative bias of correlation with percentage data becomes visible when one variable, for example socio-economic or law studies, is not considered. Assume this variable is not measured or it is not of interest for the analyst. Then percentages would be calculated based on the remaining four variables. Figure 1.1d shows the result and now the percentages of PhD students in natural and technical sciences seem to have a negative correlation with the percentages of PhD students in health and life sciences. For now it is not clear how to deal with such different results (the correlation becomes "spurious" (Pearson 1897)). We come back to this issue and introduce new concepts based on

logratios in Chaps. 4 and 8. It will become clear that logratios are a key to analyze data where the relative information is important. For example, logratios provide the same results independent if they are calculated from absolute values or from percentages.

1.2.2 Beer Data Example

As a further example, a data set with 86 different beers is considered, where the concentration for 15 chemical compounds is available. The study was conducted and presented in Varmuza et al. (2002) and the composition of chemical compounds of beers was analyzed using non-compositional methods. The beers originate from two groups: fresh beers and "old" beers. It can be assumed that the chemical composition of the two groups is different and distinguishable.

The data set is investigated in the following by principal component analysis (PCA). This important multivariate statistical method will be treated in detail in Chap. 7 from a compositional data analysis point of view. Here, PCA is applied to different kinds of preprocessed data, and the results are presented in biplots for the first and second principal component. Figure 1.2a shows the biplot for the raw data. Since the concentrations for the compound "Furfural" are very dominating, the data have been scaled to unit variance. The arrows represent the chemical compounds, and the symbols "o" and "f" stand for "old" and "fresh," respectively. The two groups show a clear overlap, one observation "f" is outlying, and the variables are arranged in a half-plane.

A further attempt is to log-transform the data. As common with concentration data, they are skewed to the right, and a log-transformation leads to better symmetry. The resulting biplot is shown in Fig. 1.2b. The outlier disappeared, and the groups are quite well separated. Only the variables are still arranged in a half-plane, which should somehow worry an experienced data analyst.

A final step is to create "closure," which means that the concentrations are divided by the sum of the values of each observation. Some people would say that only now we have compositional data since they sum up to one—see previous remarks in this chapter. The proportional data are then used for PCA, and Fig. 1.2c presents the resulting biplot. This biplot is clearly dominated by two variables, which in fact have high proportions. Therefore, Fig. 1.2d shows the biplots for the scaled proportional data by subtracting the respective mean and dividing the result by the standard deviation for each variable. The picture is not so "nice" as Fig. 1.2b, but there are certainly similarities. The arrangement of the variables in a half-plane is still present.

One can conclude from these biplots that it matters whether data are analyzed as raw data, as transformed data, or as data expressed in proportions. Moreover, the results change if PCA is based on the scaled input data or not. In any case, the configuration of the variables in the biplot seems to be spoiled due to the arrangement in a half-plane.

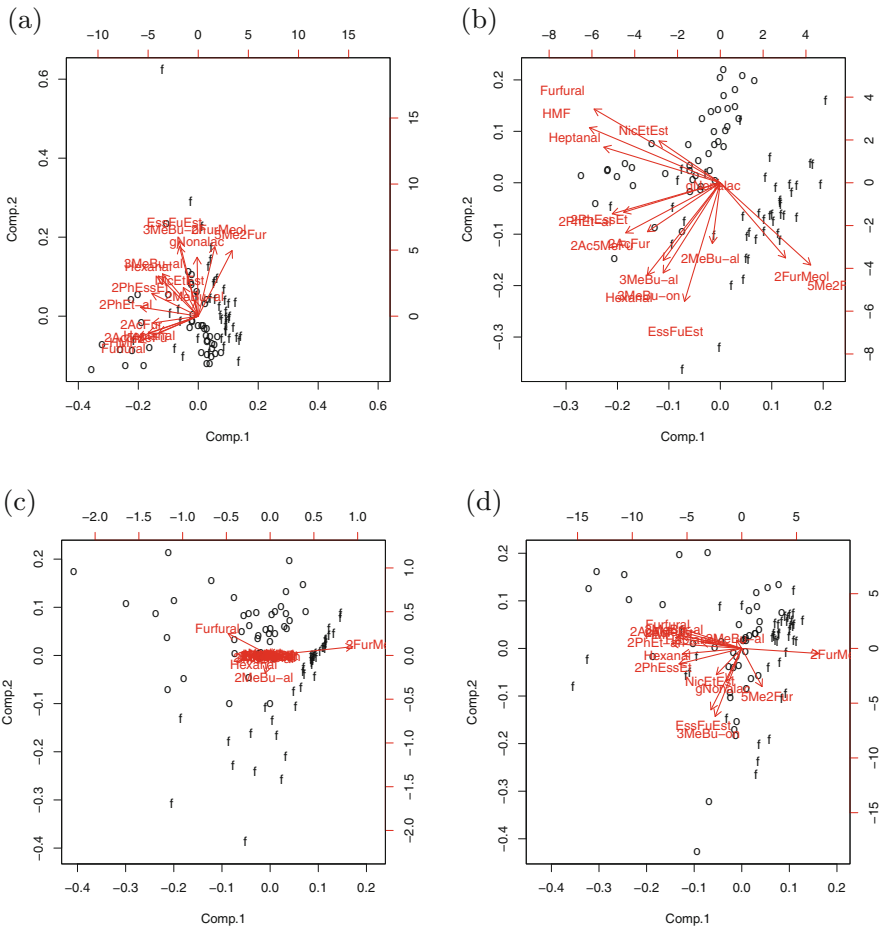


Fig. 1.2 Principal component analysis of concentrations of chemical compounds in beer. Two groups are visible, the fresh ("f") and the old ("o") beers. (**a**) Scores and loadings (biplot) resulting from principal component analysis on scaled raw absolute concentrations. (**b**) Scores and loadings (biplot) resulting from principal component analysis on scaled log-transformed concentrations. (**c**) Scores and loadings (biplot) resulting from principal component analysis on concentrations expressed as proportional data. (**d**) Scores and loadings (biplot) resulting from principal component analysis on concentrations of beers expressed as scaled proportional data

All these shortcomings will be addressed in Chap. 7 where PCA is treated from the perspective of compositional data analysis. A change of scale (units) of the variables will not affect the outcome. Also, there is no need to think about different transformations or representations of the data, because the information to be exploited for PCA will be logarithms of ratios between the compositional parts. Of course, there will be no guarantee that the old and fresh beers form separate clusters in the scores space, since this is not a feature of the methodology.

1.2.3 Geochemical Data Example

The GEMAS project was a large-scale geochemical mapping project carried out in most European countries. Concentrations of chemical elements in agricultural soils, as well as several other parameters have been measured (Reimann et al. 2012). As an example, Fig. 1.3 shows a 3D-scatterplot of the absolute concentrations of Aluminium (Al), Sodium (Na), and Silica (Si). The data are reported in mg/kg, and one can see that both Si and Al have high concentrations in most soil samples with almost 500,000 mg/kg. In other words, the remaining chemical elements, like Na, are constrained by the natural boundary of one million mg/kg, and this is visible in the plot. This artifact of the constraint is also called "closure effect," and it would certainly have implications on the statistical analysis if it were applied on the raw concentration data. A compositional data analysis methodology would use the information contained in the ratios of the chemical elements.

An even more extreme example, also borrowed from the GEMAS project, is shown in Fig. 1.4a, where the percentages of sand, silt, and clay in the soils are visualized in a 3D-scatterplot. Up to rounding errors, they sum up to 100%, and thus all points are on the plane going through the values 100% in each coordinate. Thus, if the percentages for two variables are provided, the percentage for the third variable is automatically determined. The data are thus said to be constrained or

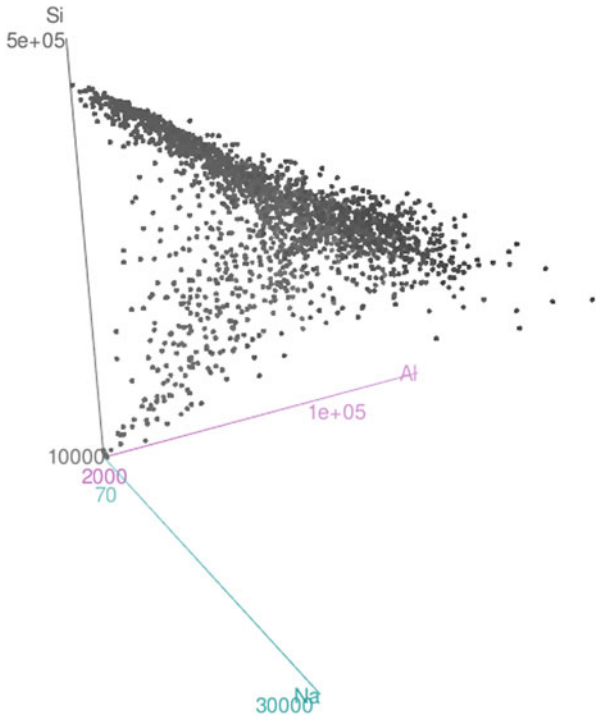


Fig. 1.3 Absolute concentrations (in mg/kg) for Aluminium, Sodium, and Silica for the agricultural soil sample survey in Europe (GEMAS data)

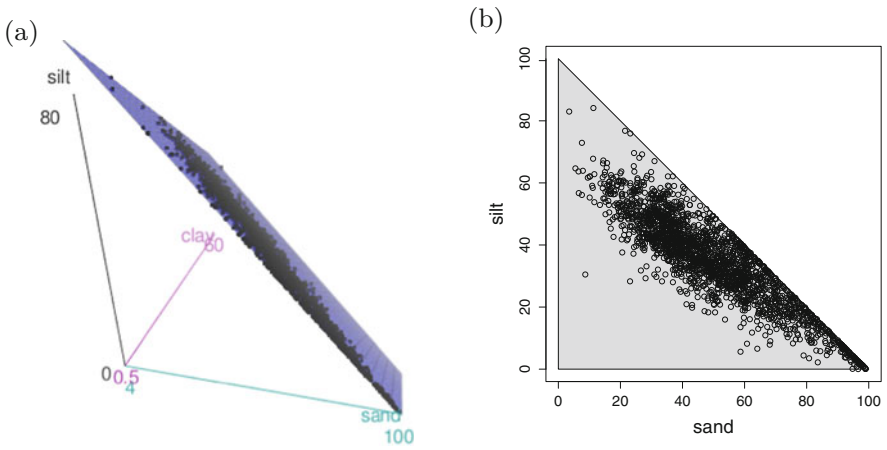


Fig. 1.4 Sand, silt, and clay from the agricultural soil sample survey of Europe (GEMAS data). These three variables sum up to 100%. (**a**) Due to the constraint, all data points are arranged on a plane. (**b**) The correlation between sand and silt is forced to a negative one

closed to 100%. If the interest is now only in the relation between two of the variables, e.g. between sand and silt, one needs to be aware of this constrained space. Figure 1.4b shows a scatterplot of both variables, and due to the constraints, all data points need to be inside the dark triangle. It is clear that computing the correlation between the two variables would be inappropriate, since it will be automatically forced to a negative one. The point is that even for unconstrained compositional data, correlations can be spoiled, and thus there is a need for another more appropriate approach.

1.3 Principles of Compositional Data Analysis

The intrinsic properties of compositional data as introduced in Sect. 1.1 can be formally summarized into three principles that should be followed by any reasonable method for their (not exclusively statistical) analysis. Just to remind, a compositional vector, or simply a composition, $\mathbf{x} = (x_1, \ldots, x_D)'$ with D parts (arranged into a column vector) is by definition a positive real vector with D components, describing quantitatively the parts of some whole, which carry relative information between the parts. According to Egozcue (2009), compositional data analysis should respect the following principles:

Scale invariance: The information in a composition does not depend on the particular units in which the composition is expressed. Proportional positive vectors represent the same composition. Any sensible characteristic of a composition should be invariant under a change of scale. This principle thus corresponds to

the fact that a multiplication of a compositional vector by an arbitrary positive number does not alter the ratios between compositional parts.

Permutation invariance: Permutation of parts of a composition does not alter the information conveyed by the compositional vector, similarly as in standard multivariate statistics.

Subcompositional coherence: Information conveyed by a composition of D parts should not be in contradiction with that coming from a subcomposition (i.e., a subvector of the original compositional vector) containing d parts, $d < D$. This principle can be formulated more precisely as

- *Subcompositional dominance*: If $\Delta_p(\mathbf{x}, \mathbf{y})$ is any distance between compositions of p parts, then

$$\Delta_D(\mathbf{x}, \mathbf{y}) \geq \Delta_d(\mathbf{x}_d, \mathbf{y}_d),$$

  where $\mathbf{x}, \mathbf{y}$ are compositions with D parts and $\mathbf{x}_d, \mathbf{y}_d$ are subcompositions of the previous ones with d parts, $d < D$.
- *Ratio preserving*: Any relevant characteristic expressed as a function of the parts of a composition is exclusively a function of the ratios of its parts. In a subcomposition, these characteristics depend only on the ratios of the selected parts and not on the discarded parts of the parent composition. Scale invariance applies to the subcomposition.

While the principles of permutation invariance and subcompositional dominance should be fulfilled by *any* reasonable statistical analysis, aware of the corresponding geometrical consequences (Eaton 1983), scale invariance is a specific principle resulting directly from the definition of compositional data. In particular, scale invariance means that the relevant information, conveyed by ratios, remains the same by an arbitrary rescaling of the input observations. An important consequence is that one does not "generate" compositional data by expressing them in proportions, percentages or any similar well-established representation. It is the purpose of the analysis (absolute versus relative) that induces whether already the original observations are compositional or not. It is also important to understand properly what subcompositional coherence says, supported by examples from the previous section: it can be highly misleading to apply standard statistical methods to compositional data directly, because an arbitrary rescaling of the input can change the results completely. This is closely linked to the fact that the Euclidean geometry, on which most standard statistical methods rely (Eaton 1983), is not appropriate for compositional data.

Subcompositional dominance induces that the distance computed between two compositions cannot be less than the distance between the corresponding subcompositions. For standard real observations, this is illustrated in Fig. 1.5, where a planar graph with two observations, A and B, is displayed. By projecting them to the x-axis and to the y-axis, respectively, it can be observed that their distance (between a_1, b_1 and a_2, b_2, respectively) is less than the distance between the original data. Therefore, it is logical to expect that a similar property should be fulfilled also

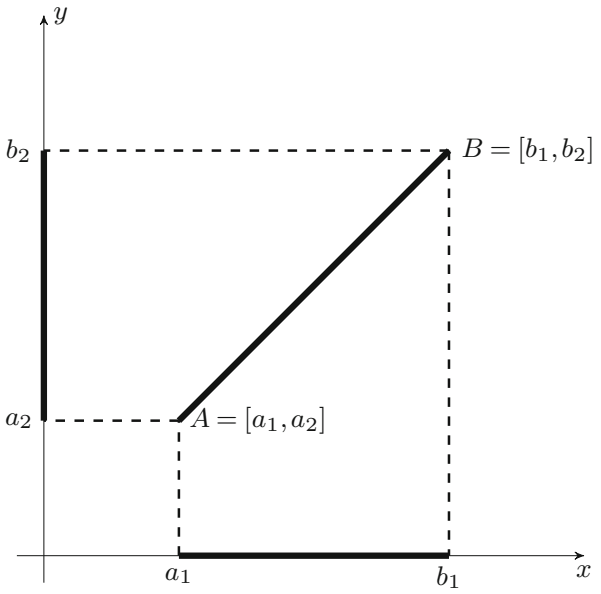


Fig. 1.5 Euclidean distance for two observations A and B, and their projections on the x- and y-axes

for compositions. Consider two compositions $\mathbf{x} = (0.55, 0.40, 0.05)'$ and $\mathbf{y} = (0.10, 0.80, 0.10)'$, expressed in proportional representation. Their Euclidean distance is $d(\mathbf{x}, \mathbf{y}) = \sqrt{(0.55 - 0.10)^2 + (0.40 - 0.80)^2 + (0.05 - 0.10)^2} = 0.604$. When computing the Euclidean distance between the vectors consisting of the first two components, $\sqrt{(0.55 - 0.10)^2 + (0.40 - 0.80)^2} = 0.602$, everything seems to work well. But the point is that such a property should be fulfilled for *any* representation of these subcompositions. If the subcompositions are expressed as proportions, i.e. $0.55/(0.55 + 0.40)$, etc., resulting in $\mathbf{x}_s = (0.579, 0.421)'$ and $\mathbf{y}_s = (0.111, 0.889)'$, their Euclidean distance is 0.661, what clearly contradicts the assumption of subcompositional dominance.

Another natural principle of compositional data that should be addressed, though it is not directly included in the above listing, concerns *relative scale* of compositional data. Its basic idea is that for expressing the dissimilarity between two compositions, the ratio between the values of a component should be considered instead of taking the difference between them. Consider a simple example where the number of votes in a village for a political party in a particular year is 200, while the corresponding number in the previous election was 300. It is natural to conclude that this party has lost one third of the votes, rather than talking about a loss of 100 votes. The reason is comparability: If in another village the number of votes for this party in the considered year is 2900, while it was 3000 in the previous elections, the loss is only $3.\overline{3}\%$, while the absolute is again the same with 100 votes. Consequently, the

relative scale effect applies mostly for components with lower values, e.g., for trace elements in geochemistry. Of course, any compositional part does not stay alone, it is always linked through logratios also to other parts in the actual composition. This fact should be considered when any reasonable distance, respecting the relative scale, is developed.

Although these principles just help to formalize the concept of compositional data together with some obvious geometrical requirements, they caused a number of misunderstandings and controversy outside as well as inside the "compositional community." The principal misunderstanding results from the attempts to apply them consistently to proportional data, identified with the above broader definition of compositional data, as recently done, e.g., in Cortés (2009) and Scealy and Welsch (2014). Without accepting scale invariance as the generic principle that drives also the remaining two principles, particularly the subcompositional coherence principle might become quite misleading. Accordingly, Scealy and Welsch (2014) have even "proved" that the logratio methodology itself is not subcompositionally coherent. But the problem is a different one. The misunderstanding comes from the fact that proportional data assume a fixed whole, to which single proportions relate, and absolute values of proportions are considered informative. On the other hand, from the perspective of the logratio methodology, proportions stand just for a concrete representation of the compositional vector. It is nowhere stated that such a concept of "absolute proportions" cannot be useful in particular situations, especially when a clearly stated whole is provided. But one should be aware that both scale invariance and relative scale of compositions, together with further geometrical implications (see subcompositional coherence), are obviously linked closely to the broader definition of compositional data.

1.4 Steps to a Concise Methodology

The fact that scale invariance forms the generic principle of compositional data analysis should be reflected by any reasonable geometrical representation of compositional data. Without any doubts, (log)ratios between parts will play an important role there, since they contain the essential information of compositional data. Such a geometry needs to be set up by an appropriate algebraic-geometrical structure, represented by the properties of the Euclidean vector space (Eaton 1983). Among other possibilities, it is exclusively the Aitchison geometry as introduced in Pawlowsky-Glahn and Egozcue (2001) that follows all the above requirements. Although it would also be possible to analyze compositions directly in this geometry, it would require an inadequate effort with an uncertain output. The reason is that most standard statistical methods are designed for the Euclidean geometry in real space. Therefore, it is preferable to construct a family of "transformations" from the original sample space of compositional data to the real space, where standard multivariate methods can be applied for their statistical processing. It turns out that this goal can be achieved by the construction of interpretable *logratio coordinates*

with respect to a basis, or a generating system in the Aitchison geometry. As the coordinates will be formed by pairwise logratios or their aggregation, particular interest will be devoted to interpretation: To which extent can the coordinates be identified with the original compositional parts, and which implications for the implementation and interpretation of statistical methods are to be expected? By construction of the logratio coordinates, it will also turn out that for object oriented methods (like cluster or discriminant analysis), any reasonable coordinate representation can be chosen without altering the outputs. Although is it just a matter of terminology, in the sequel it will be systematically referred to *coordinates* instead of *transformations*. This corresponds to the "staying-in-the-simplex approach," as it is followed also in other recent books in this field (van den Boogaart and Tolosana-Delgado 2013; Pawlowsky-Glahn et al. 2015).

The book is organized as follows. In the next chapter, the statistical software environment R (R Development Core Team 2017) is introduced in the context of compositional data analysis. This software environment will accompany the rest of the book and provide routines in order to apply most of the presented methods. Chapter 3 is devoted to the Aitchison geometry of compositional data together with various logratio coordinate representations, which are crucial for the statistical processing with common multivariate statistical methods. Specific features of compositions and the interpretation of logratio coordinates imply certain peculiarities when visualizing compositional data—this is treated in Chap. 4. Chapter 5 concludes the general part of the book by providing further methodological contributions for analyzing compositional data, in particular in the direction of parametric statistical inference and robust methods. Starting with Chap. 6, where exploratory analysis is introduced, the core methodological part of the book follows, containing many popular multivariate statistical methods (cluster analysis, principal component analysis, correlation analysis, discriminant analysis, and regression), adapted to deal with compositional data in logratio coordinates. Chapter 11 extends the basic data setting to high-dimensional compositions, for which special methods like partial least squares regression are required. Chapter 12 develops another specific data structure that allows to link two factors through a compositional table. Finally, Chap. 13 deals with practical issues present in many real world data sets, namely with missing and zero values, and proposes several methods how these effects can be successfully overcome in order to continue with further statistical processing using the logratio methodology.

References

J. Aitchison, *The Statistical Analysis of Compositional Data* (Chapman & Hall, London, 1986). Reprinted in 2003 with additional material by The Blackburn Press

D. Billheimer, P. Guttorp, W.F. Fagan, Statistical interpretation of species composition. J. Am. Stat. Assoc. **96**(456), 1205–1214 (2001)

A. Buccianti, G. Mateu-Figueras, V. Pawlowsky-Glahn (eds.), *Compositional Data Analysis: Theory and Applications* (Wiley, Chichester, 2011)

F. Chayes, On correlation between variables of constant sum. J. Geophys. Res. **65**(12), 4185–4193 (1960)

J.A. Cortés, On the Harker variation diagrams; a comment on "The statistical analysis of compositional data. Where are we and where should we be heading?" by Aitchison and Egozcue (2005). Math. Geosci. **41**(7), 817–828 (2009)

M.L. Eaton, *Multivariate Statistics. A Vector Space Approach* (Wiley, New York, 1983)

J.J. Egozcue, Reply to "On the Harker variation diagrams; . . . " by J.A. Cortés. Math. Geosci. **41**(7), 829–834 (2009)

V. Pawlowsky-Glahn, J.J. Egozcue, Geometric approach to statistical analysis on the simplex. Stoch. Env. Res. Risk A. **15**(5), 384–398 (2001)

V. Pawlowsky-Glahn, J.J. Egozcue, R. Tolosana-Delgado, *Modeling and Analysis of Compositional Data* (Wiley, Chichester, 2015)

K. Pearson, Mathematical contributions to the theory of evolution. On a form of spurious correlation which may arise when indices are used in the measurement of organs. Proc. R. Soc. Lond. **LX**, 489–502 (1897)

R Development Core Team, *R: A Language and Environment for Statistical Computing* (R Foundation for Statistical Computing, Vienna, 2017). http://www.R-project.org/, ISBN 3-900051-07-0

C. Reimann, P. Filzmoser, K. Fabian, K. Hron, M. Birke, A. Demetriades, E. Dinelli, A. Ladenberger, The GEMAS Project Team, The concept of compositional data analysis in practice-Total major element concentrations in agricultural and grazing land soils of Europe. Sci. Total Environ. **426**, 196–210 (2012)

J.L. Scealy, A.H. Welsch, Colours and coctails: compositional data analysis 2013 Lancaster Lecture. Aust. N. Z. J. Stat. **56**(2), 145–169 (2014)

K.G. van den Boogaart, R. Tolosana-Delgado, *Analyzing Compositional Data with R* (Springer, Heidelberg, 2013)

K. Varmuza, I. Steiner, H. Glinsner, H. Klein, Chemometric evaluation of concentration profiles from compounds relevant in beer ageing. Eur. Food Res. Technol. **215**(3), 235–239 (2002)

Chapter 2
Analyzing Compositional Data Using R

Abstract The theoretical concepts explained in the book are illustrated by examples which make use of the statistical software environment R. In this chapter, a brief introduction to some functionalities of R is given. This introduction does not replace a general introduction to R, but it provides the background that is necessary to understand the examples and the R code in the book.

The methods explained in this book are exclusively available in the R package **robCompositions**. The package includes methods for the analysis of compositional data including robust methods, algorithms for the imputation of missing values, methods to replace rounded zeros, outlier detection for compositional data, (robust) principal component analysis for compositional data, (robust) discriminant analysis for compositional data (Fisher rule), robust regression with compositional predictors, (robust) Anderson-Darling normality tests for compositional data, as well as functions to express compositional data in coordinates.

2.1 Brief Overview on Packages Related to Compositional Data Analysis

The implementation of methods in software is essential to apply compositional data analysis methods in practice. A variety of software tools have been written for compositional data analysis, starting with Basic routines from John Aitchison, grouped under the name **CODA**. Later, these programs have been reimplemented under the name **NEWCODA** in Matlab. Also Savazzi and Reyment (1999) presented some routines in C++ and FORTRAN 90, and Reynolds and Billheimer (2005) implemented basic transformations and plots in SPLUS/R. Almost at the same time, the software tool **CoDaPack** was developed (Thió-Henestrosa et al. 2003, 2005) using Visual Basic and Excel (the acronym CoDaPack comes from Compositional Data Package). This software is now available in version 2 (Comas-Cufí and Thió-Henestrosa 2011) based on a new implementation in Java. Also in 2005, the first version of the R package **compositions** (van den Boogaart et al. 2014) was available on CRAN. In 2009, the R package **robCompositions** (Templ et al. 2011a,b) has been developed, currently available in version 2.0. The package

P. Filzmoser et al., *Applied Compositional Data Analysis*, Springer Series in Statistics, https://doi.org/10.1007/978-3-319-96422-5_2

ggtern (Hamilton 2016) for plotting ternary diagrams is available since 2013. The package **zCompositions** (Palarea-Albaladejo and Martín-Fernández 2015) provides several functionalities for the imputation of zeros and non-detects in compositional data. In the following, some of the packages are briefly described. The package **robCompositions** is discussed in more detail since it is the basis for this book.

2.1.1 compositions

The philosophy of the implementation of the CRAN R package **compositions** (van den Boogaart et al. 2014) is to consider different multivariate scales, namely

rplus: the data are supposed to be non-compositional.
rcomp: the data are supposed to be of compositional nature, but the analysis is done in the original scale.
acomp: the data are supposed to be of compositional nature, and the analysis is done in the relative geometry.
aplus: the total amount is meaningful, and the data are analyzed in the relative geometry.

The *rplus* approach is thus equivalent to a classical non-compositional analysis of non-compositional data. The *rcomp* approach might just be used for comparison reasons, and *aplus* is mainly used internally in the package. Thus *acomp* is the interesting supported class of the package **compositions** for analyzing compositional data. The function `acomp` applied on compositional data produces an object of class *acomp*. Methods are defined for this class, ranging from plotting methods like ternary diagrams to tests over outlier detection, multivariate statistical methods such as cluster and discriminant analysis, principal component analysis and regression methods.

2.1.2 robCompositions

The CRAN R package **robCompositions** includes methods for the analysis of compositional data including robust methods, algorithms for imputation, methods to replace rounded zeros, outlier detection for compositional data, classical and robust multivariate methods for compositional data, such as principal component analysis and discriminant analysis for compositional data, robust regression with compositional predictors, and Anderson-Darling normality tests for compositional data. Several options to express compositions in coordinates or coefficients are available, together with the corresponding inverse mappings. In addition, visualization and diagnostic tools are implemented as well as high- and low-level plot

functions for the ternary diagram. The examples in this book are based on the package **robCompositions**.

Table 2.1 presents the most important functions of the R package **robCompositions**.

Table 2.1 Most important functions of the R package **robCompostions** for compositional data analysis

Function	Aim	References
`addLR` and `addLRinv`	Additive logratio coordinates and inverse mapping	Aitchison (1986)
`aDist`	Aitchison distance between two compositions or pairwise between two data sets	Aitchison (1986)
`adtest`	Anderson-Darling normality test	Anderson and Darling (1952)
`cenLR` and `cenLRinv`	Centered logratio coefficients and inverse mapping	Aitchison (1986)
`compareMahal`	Compares Mahalanobis distances from two approaches	
`constSum`	Closure operation	
`daFisher`	Discriminant analysis by Fisher's rule	Filzmoser et al. (2012)
`gm`	Geometric mean	
`impCoda`	Robust imputation of missing values (EM algorithm)	Hron et al. (2010)
`impKNNa`	Imputation of missing values (k nearest neighbor approach)	Hron et al. (2010)
`imputeBDLs`	Imputation of rounded zeros	Martín-Fernández et al. (2012), Templ et al. (2016, for high-dimensional methods)
`pivotCoord` and `pivotCoordInv`	Pivot coordinates as a special choice of isometric logratio coordinates and inverse mapping	Egozcue et al. (2003), Fišerová and Hron (2011)
`lmCoDaX`	Regression with compositional explanatory variables	Hron et al. (2012)
`missPatterns` and `zeroPatterns`	Missing values and zeros pattern structure	
`outCoDa`	Outlier detection	Filzmoser and Hron (2008)
`pcaCoDa`	Robust principal component analysis	Filzmoser et al. (2009)
`ternaryDiag`	Ternary diagram	Aitchison (1986)
`variation`	Variation matrix	Aitchison (1986)

Additionally, a number of different kinds of data sets are included in the package, listed below.

```
data(package = "robCompositions")
```

```
Data sets in package 'robCompositions':


ageCatWorld        child, middle and eldery population
alcohol            alcohol consumptions by country and type of alcohol
alcoholreg         regional alcohol consumption by WHO region
arcticLake         Arctic lake sediment data
cancer             Hospital discharges on cancer and distribution of age
cancerMN           Malignant neoplasms cancer
chorizonDL         C-horizon of the Kola data with rounded zeros
coffee             coffee data set
economy            economic indicators
educFM             education level of father (F) and mother (M)
election           election data
electionATbp       Austrian presidential election data
employment         employment in different countries by gender and status
employment_df      employment in different countries by gender and status
expenditures       synthetic household expenditures toy data set
expendituresEU     mean consumption expenditures data
GDPsatis           GDP satisfaction
gemas              GEMAS geochemical data set
govexp             government spending
haplogroups        haplogroups data
instw              value added, output and input for different ISIC codes
isic32             ISIC codes by name
laborForce         labour force by status in employment
lifeExpGdp         life expectancy and GDP (2008) for EU-countries
machineOperators   machine operators
mcad               metabolomics MCAD data set
mortality          mortality and life expectancy in the EU
mortality_tab      mortality table
nutrients          nutrient contents
nutrients_branded  nutrient contents (branded)
payments           special payments
phd                PhD students in the EU
precipitation      table containing counts for 24-hour precipitation
production         production split by nationality on enterprise level
rcodes             codes for UNIDO tables
skyeLavas          aphyric Skye lavas data
socExp             social expenditures
teachingStuff      teaching stuff
trondelagC         regional geochemical survey of soil C in Norway
trondelagO         regional geochemical survey of soil O in Norway
unemployed         unemployment of young people
```

Most of these data sets are used for illustrating the theoretical concepts presented in this book.

2.1.3 ggtern

The CRAN R package **ggtern** (Hamilton 2016) is an extension to **ggplot2** (Wickham 2009) for plotting ternary diagrams. It is possible to put a great variety of symbols, error bars, lines, and ellipses into ternary diagrams. For examples and documentation, see the **ggtern** website (http://www.ggtern.com/).

2.1.4 zCompositions

The CRAN R package **zCompositions** (Palarea-Albaladejo and Martín-Fernández 2015) offers several possibilities for the imputation of left-censored data by considering the compositional data approach. Methods for imputation are treated in Chap. 13, where also references to the implemented methods will be made.

2.1.5 mvoutlier, StatDA

The CRAN R package **mvoutlier** (Filzmoser and Gschwandtner 2017) includes programs for multivariate outlier detection for compositional data, as well as tools for visualizing the outliers. This package also contains data sets from geochemistry, and there are several more geochemical data sets available in the R package **StatDA** (Filzmoser 2015).

2.1.6 CoDaPack

The freeware package **CoDaPack** can be downloaded from the web site http://ima.udg.edu/CoDaPack. This point and click user interface relies on the Java Virtual Machine. It includes the basic logratio coordinate systems, ternary plots, biplots, summaries, and the basic mathematical operations such as powering and perturbation.

2.1.7 *compositionsGUI*

Also the package **compositionsGUI** (Eichler et al. 2014) represents a point and click user interface and includes basic plots, logratio coordinates, and multivariate methods that are called from the package **robCompositions** or **compositions**. It is not further developed and archived on CRAN.

2.2 The Statistics Environment R

R (R Development Core Team 2018) was founded by Ross Ihaka and Robert Gentlemen in 1995. It is based on S, a programming language developed by John Chambers (Bell Laboratories) and Scheme. Since 1997 it is internationally developed and distributed from Vienna over the Comprehensive R Archive Network (CRAN, cran.r-project.org). R nowadays belongs to the most popular and most used software environments in the statistics world. In addition, R is free and open-source (under the GPL2). R is not only a software for doing statistics, it is an environment for interactive computing with data supporting facilities to produce high-quality graphics. The exchange of code with others is easy since everybody may download R. This might also be one reason why modern methods are often exclusively developed in R. R is an object-oriented programming language and has interfaces to many other software products such as C, C++, Java, and interfaces to databases.

Useful information can be found at:

- Homepage: http://www.r-project.org/ and CRAN http://cran.r-project.org for download
- Lists with frequently asked questions (FAQ) on CRAN
- Manuals and *contributed* manuals
- Task-views on CRAN

The basic installation of R is extendable with approximately 10,000 *add-on* packages.

For R programming it is advisable to write the code in a well-developed editor. An editor should allow syntax highlighting, code completion, and interactive communication with R. For beginners but also for advanced users, R-Studio (http://www.rstudio.org/) is one choice. Experts might also use the combination of Eclipse + its add-on STATET (https://marketplace.eclipse.org/content/statet-r).

2.3 Basics in R

R can be used as an overgrown calculator. All operations of a calculator can be very easily used also in R. For instance, addition is done with +, subtraction

with -, division with /, exponential with exp(), logarithm with log(), square-root using sqrt(), sinus with sin(), etc. All operations work as expected. As an example, the following expression is parsed by R, inner brackets are solved first, multiplication and division operators have precedence over the addition and subtraction operators, etc.

```
0.5 + 0.2 * log(0.15^2)

## [1] -0.258848
```

R is a function and object-oriented language. Functions can be applied to objects. The syntax is as shown in the following example, where the add-on package **robCompositions** (Templ et al. 2011a,b) is loaded first.

```
library("robCompositions")
gm(runif(10, 0, 1))

## [1] 0.4132139
```

With the function runif, 10 numbers are randomly drawn from a uniform distribution, in our case values in the interval [0,1]. Afterwards, the geometric mean using the function gm is calculated for these 10 numbers. Functions typically have function arguments that can be set. The syntax for calling a function has the general structure:

```
res1 <- name_of_function(v1) # one input argument
res2 <- name_of_function(v1, v2) # two input arguments
res3 <- name_of_function(v1, v2, v3) # three input arguments
# ...
```

Functions often have additional function arguments with default values. It is possible to get access to all function arguments with args().

```
args(gm)

## function (x)
## NULL

args(runif)

## function (n, min = 0, max = 1)
## NULL
```

Allocations to objects are made by <- or =, and the generated object can be printed with object name, followed by typing ENTER.

```
x <- runif(10, 0, 1)
x

##  [1] 0.829134482 0.858681638 0.057063767 0.005798472
##  [5] 0.316049618 0.158100595 0.497237901 0.342189933
##  [9] 0.075772098 0.980245832
```

Note that R is case sensitive.

2.3.1 Installation of R and Updates

If R is already installed on the computer, ensure that it is the current version. If the software is not installed, go to http://cran.r-project.org/bin/ and choose your platform. For Windows, just download the executable file and follow the on-screen instructions.

2.3.2 Install robCompositions

Open R on your computer and type:

```
install.packages("robCompositions")
```

This command installs the package **robCompositions** from the CRAN server, provided that the computer has access to the Internet. Installation is needed only once.

Typing `update.packages()` into R searches for possible updates and installs new versions of packages if those are available.

If your organization uses a proxy server to connect to the internet, automatic access of R is usually restricted, but users can access the necessary internet connection from within R. If you have a proxy server, the following command, typed into the R-console, might help:

```
setInternet2(TRUE)
```

This may allow you to install the packages. Otherwise, contact your IT department for the permission so that R can connect to the CRAN servers.

The previous information was about to install the stable CRAN version of the packages. However, latest changes are only available in the development version of the package. This is hosted on https://github.com/matthias-da/robCompositions and includes test batteries to ensure that the package keeps stable when modifying parts of the package. From time to time, a new version is uploaded to CRAN.

To install the latest development version, the installation of the package **devtools** (Wickham and Chang 2015) is needed. After calling the **devtools** package, the development version can be installed via `install_github()`.

```
## if not installed, install package devtools:
if(!require(devtools)){
    install.packages("devtools")
}
## load the devtools package
library("devtools")
## install package from github
install_github("matthias-da/robCompositions")
```

2.3.3 Help

It is crucial to have basic knowledge about getting help. With

```
help.start()
```

your browser opens, and the help pages (and more) get available.

The browsable help index of the package can be accessed by typing the following command into R:

```
help(package = robCompositions)
```

To find specific help for a function, say `name`, one can use `help(name)` or `?name`. As an example, we look at the help file of the function `outCoDa`, which is included in the package **robCompositions**:

```
?outCoDa
```

Data in the package can be loaded via the `data()` function, e.g. in case of the `phd` data set from the package **robCompositions**:

```
data("phd")
```

`help.search()` can be used to find functions for which the exact name is not known by heart. For instance,

```
help.search("pca coda")
```

will search your local R installation for functions approximately matching the character string `"pca coda"` in the (file) name, alias, title, concept or keyword entries. With the function `apropos` one can find and list objects by (partial) name.

For example, to list all objects with partial name match `coda`:

```
apropos("coda")

##  [1] "clustCoDa"           "clustCoDa_qmode"
##  [3] "CoDaDendrogram"      "corCoDa"
##  [5] "daCoDa"              "impCoda"
##  [7] "lmCoDaX"             "mvoutlier.CoDa"
##  [9] "outCoDa"             "pcaCoDa"
## [11] "plot.mvoutlierCoDa"
```

It can be seen that several functions are listed that may be helpful in the context of compositional data analysis.

To search help pages, vignettes or task views, using the search engine at http://search.r-project.org and to view them in your web browser, you can use

```
RSiteSearch("isometric logratio")
```

which reports all search results for the character string `"isometric logratio"`.

2.3.4 The R Workspace and the Working Directory

Created objects are available in the workspace of R and loaded in the memory of your computer. The collection of all created objects is called *workspace*. To list the objects in the workspace, type:

```
ls()

## character(0)
```

When importing or exporting data, the working directory must be defined. To show the current working directory, the function `getwd` can be used:

```
getwd()

## [1] "/home/filz/latex/papers/hron/buch/ver12/codabook/book"
```

To change the working directory, the function `setwd` is the choice:

```
# paste creates a string
p <- paste(getwd(), "/data", sep = "")
p

## [1] "/home/filz/latex/papers/hron/buch/ver12/codabook/book/data"
```

```
# now change the working directory
setwd(p)
```

2.3.5 *Data Types*

The most important data types in R are:

- vectors/factors
- lists
- data frames
- special data types: missing values, NULL-objects, NaN, -/+ Inf

Vectors are the simplest data structure in R. A vector is a sequence of elements of the same type such as numerical vectors, character vectors, or logical vectors. Vectors are often created with the function `c()`, e.g.:

```
v.num <- c(0.1, 0.3, 0.5, 0.9, 0.7)
v.num

## [1] 0.1 0.3 0.5 0.9 0.7

is.numeric(v.num)

## [1] TRUE
```

The command `is.numeric` checks if the vector is of class numeric. Note that characters are written with parenthesis.

Logical vectors are often created indirectly from numerical/character vectors:

```
v.num > 0.3

## [1] FALSE FALSE  TRUE  TRUE  TRUE
```

Many operations on vectors are performed element-wise, e.g. logical comparisons or arithmetic operations with vectors. A common error source is when the lengths of the vectors differ. Then the shorter one is repeated (*recycling*):

```
v1 <- c(0.1, 0.2, 0.3)
v2 <- c(0.4, 0.5)
v1 + v2

## [1] 0.5 0.7 0.7
```

One should also be aware that R coerces internally to meaningful data types automatically. For example:

```
v2 <- c (100, TRUE, "A", FALSE)
v2

## [1] "100"   "TRUE"  "A"     "FALSE"

is.numeric (v2)

## [1] FALSE
```

Here, the lowest common data type is a string and therefore all entries of the vector are coerced to character. Note, to create vectors, the functions `seq` and `rep` are very useful.

Often it is necessary to subset vectors. The selection is made using the `[]` operator. A selection can be done in three ways:

positive: a vector of positive integers that specifies the position of the desired elements,
negative: a vector with negative integers indicating the position of the non-required elements,
logical: a logic vector with selected (TRUE) and not selected (FALSE) elements.

```
data("gemas")
# extract a subset of the variable sand from the gemas data
sand <- gemas[1:10, "sand"]
sand

##  [1] 69.7 47.4 69.4 61.4 83.3 26.9 43.2 50.0 50.5 60.5

# positive indexing:
sand[c(3, 6, 7)]

## [1] 69.4 26.9 43.2

# negative indexing:
sand[-c(1, 2, 4, 5, 8:10)]

## [1] 69.4 26.9 43.2

# logical indexing:
sand < 30

##  [1] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
## [10] FALSE

# a logical expression can be written directly in []
sand[sand < 30]

## [1] 26.9
```

A list in R is an *ordered* collection of objects whereas each object is part of the list and where the data types of the individual list elements can be different (vectors, matrices, data frames, lists, etc.). The dimension of each list item can be different. Lists can be used to group and summarize various different objects in a new object. There are (at least) three ways to access elements of a list: the `[]`-operator, the operator `[[]]`, the `$`-operator and the name of a list item. With `str()`, you can view the structure of a list, with `names()` you get the names of the list elements.

```
## compute clr coefficients
clr <- cenLR(gemas[, 12:29])
## result is a list
class(clr)

## [1] "clr"

str(clr)

## List of 2
##  $ x.clr:'data.frame': 2108 obs. of  18 variables:
##   ..$ Al: num [1:2108] 3.62 4.23 3.91 3.85 3.79 ...
##   ..$ Ba: num [1:2108] -0.694 -0.303 -0.583 -0.666 -0.852 ...
##   ..$ Ca: num [1:2108] 1.57 1.47 2.13 2.11 2.81 ...
##   ..$ Cr: num [1:2108] -3.01 -3.88 -3.7 -3.37 -3.8 ...
##   ..$ Fe: num [1:2108] 2.4 3.05 2.71 2.71 2.74 ...
##   ..$ K : num [1:2108] 2.96 2.92 3.28 3.26 2.43 ...
##   ..$ Mg: num [1:2108] 0.581 1.682 1.332 1.479 1.607 ...
##   ..$ Mn: num [1:2108] -0.847 -0.632 -0.78 -1.045 -0.714 ...
##   ..$ Na: num [1:2108] 1.77 2.6 2.44 1.79 2.67 ...
##   ..$ Nb: num [1:2108] -4.15 -4.66 -4.75 -4.29 -4.69 ...
##   ..$ P : num [1:2108] 0.1512 -0.3154 0.0521 -0.297 -0.0861 ...
##   ..$ Si: num [1:2108] 6.71 5.57 6.1 6.26 5.65 ...
##   ..$ Sr: num [1:2108] -2.38 -1.43 -1.99 -2.3 -1.37 ...
##   ..$ Ti: num [1:2108] 1.287 1.256 0.727 1.182 0.999 ...
##   ..$ V : num [1:2108] -3.05 -2.84 -3.3 -3.19 -3.28 ...
##   ..$ Y : num [1:2108] -3.66 -4.42 -3.65 -3.59 -3.59 ...
##   ..$ Zn: num [1:2108] -2.83 -2.86 -2.62 -3.09 -3.17 ...
##   ..$ Zr: num [1:2108] -0.446 -1.423 -1.296 -0.794 -1.141 ...
##  $ gm   : num [1:2108] 506 1167 812 728 980 ...
##  - attr(*, "class")= chr "clr"

names(clr)

## [1] "x.clr" "gm"

## access elements from the named list with the dollar sign
summary(clr$gm)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   117.8   815.3  1113.0  1069.0  1336.0  2441.0
```

Factors in R are of special importance. They are used to represent nominal or ordinal data. More precisely, unordered factors for nominally scaled data and ordered factors for ordinally scaled data. Factors can be seen as special vectors. They are internally coded integers from 1 to n (# of occurrences) which are all associated with a name (label). So why should or can numeric or character variables be used as factors? Basically, factors have to be used for categorical information in order to get the correct number of degrees of freedom and correct design matrices in statistical modeling. In addition, the implementation of graphics for factors versus numerical or character vectors differs. Also, factors are more efficient for storing character vectors. However, factors have a more complex data structure, since factors include a numerically coded data vector and labels for each level/category.

```
class(gemas$soilclass)

## [1] "factor"

levels(gemas$soilclass)

## [1] ""   "l"  "ll" "m"  "s"  "ss"

summary(gemas$soilclass)

##       l  ll   m   s  ss
##   5 583 415 766 329  10
```

We note that the output of `summary` is different for factors. Internally, R applies a method dispatch for generic functions like `summary`, searching in our case if a function `summary.factor` exists. If yes, this function is applied; if not, `summary.default` is used.

Data frames (in R `data.frame`) are the most important data type. This corresponds to the rectangle data format, well-known from other software packages, with *rows* corresponding to observation units and *columns* to variables. A `data.frame` is like a `list` whereas all list elements are vectors/factors but with the restriction that all list elements have the same number of elements (equal length). For example, data from external sources to be read are often stored as data frames, i.e. data frames are usually created by reading data but they can also be constructed with the function `data.frame()`.

A lot of opportunities exist to subset a data frame; one possibility is to use `[` *index rows*`,` *index columns* `]`. Again, positive, negative, and logical indexing is possible and the type of indexing may be different for row index and column index. The access to individual columns is possible by the `$`-operator (like lists).

```
## select a subset of observations:
w <- gemas$soilclass == "ss" & gemas$MeanTemp < 14
dim(gemas[w, ])

## [1]  3 30

## select a subset of variables
cn <- colnames(gemas) %in% c("COUNTRY", "longitude", "latitude")
gemas[w, cn]

##      COUNTRY longitude latitude
## 732      CRO   17.5772  45.1733
## 783      ITA   11.6917  42.9817
## 1044     SPA   -2.6106  40.2847
```

A few helpful functions that can be used in conjunction with data frames are `dim()`, reporting the dimension (number of rows and columns), `head()`, the first (default 6) rows of a data frame, `colnames()`, the columns/variable names.

Missing values are frequently present in the data. The default representation of a missing value in R is the symbol `NA`. A very useful function to check if data values are missing is `is.na`. It returns a logical vector or data frame, depending on if the input is a vector or data frame indicating missingness. To calculate the number of

missing values, one could sum up the TRUE's (interpreted as 1, while FALSE is interpreted as 0).

```
sum(is.na(gemas))

## [1] 75
```

All in all, 75 values are missing.

To analyze the structure of missing values, the R package **VIM** (Templ et al. 2012) can be used. In the package **robCompositions**, a useful function is `missPatterns` that shows the structure of missing values.

```
m <- missPatterns(gemas)
names(m)

## [1] "groups"       "cn"           "tabcomb"      "tabcombPlus"
## [5] "rsum"         "rindex"

## patterns of missingness:
m$tabcombPlus

##      X1   X2   X3   X4   X5   X6   X7   X8    X9   X10   X11
## 1 TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  TRUE  TRUE  TRUE
## 2 TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE
##    X12  X13  X14  X15  X16  X17  X18  X19  X20  X21  X22  X23
## 1 TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## 2 TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##    X24  X25  X26  X27  X28  X29  X30 csum
## 1 TRUE TRUE TRUE TRUE TRUE TRUE TRUE 2083
## 2 TRUE TRUE TRUE TRUE TRUE TRUE TRUE   25

## observations with missings:
which(is.na(gemas$sand), arr.ind = TRUE)

##  [1]   84  191  306  308  445  521  568  592  693  735  916
## [12]  937 1044 1052 1060 1241 1448 1452 1547 1559 1565 1631
## [23] 1661 1741 2089

## e.g.
gemas[2089,]

##      COUNTRY longitude latitude  Xcoord  Ycoord MeanTemp
## 2089     HUN   19.3569  46.5875 5036660 2653419     10.9
##      AnnPrec soilclass sand silt clay    Al  Ba    Ca Cr   Fe
## 2089     547        ll   NA   NA   NA 22440 199 17567 18 7974
##         K   Mg  Mn   Na Nb   P     Si Sr   Ti  V  Y Zn Zr LOI
## 2089 9273 3256 294 6083  4 498 399894 91 1343 18 13 34 85 3.6
```

It can be seen that 25 missing values occur in the variable `sand`; the same holds for the variables `silt` and `clay`. A similar function (`zeroPatterns()`) exists to check for zeros in the data set.

2.3.6 Generic Functions, Methods and Classes

R has different class systems, the most important ones are S3 and S4 classes. Programming with S3 classes is lazy living, it is much easier than S4. However, S4 is more *clean* and the use of S4 can make packages very user-friendly.

In any case, in R each object is assigned to a class (attribute *class*). Classes allow object-oriented programming and *overloading* of *generic functions*. Generic functions produce different output for objects from different classes if methods are written for such classes. This sounds complex, but with the following example it should get clearer.

As an example of a generic function, we use the function `summary`. This is a generic function used to produce result summaries. The function invokes particular methods which depend on the class of the first argument.

```
## how often "summary" exists for methods summarizing certain classes
length(methods(summary))

## [1] 199

class(gemas$soilclass)

## [1] "factor"

summary(gemas$soilclass)

##       l  ll   m   s  ss
##   5 583 415 766 329  10

## just to see the difference, convert to class character:
summary(as.character(gemas$soilclass))

##    Length     Class      Mode
##      2108 character character
```

From this previous example one can see that the summary is different, depending on the class of the object. R internally looks if a method is implemented for the given class of the object. If yes, this function is used, if not, the function `summary.default` is used. This procedure is called *method dispatch*.

In the previous example, last line, R looks if a function `summary.factor` is available, which was true.

Note that–even not touched in this introduction–one can easily write own generic functions, and define print, summary, and plot functions for objects of certain classes.

The package **robCompositions** is used and applied in the example sections of the following chapters.

References

J. Aitchison, *The Statistical Analysis of Compositional Data* (Chapman & Hall, London, 1986). Reprinted in 2003 with additional material by The Blackburn Press

T.W. Anderson, D.A. Darling, Asymptotic theory of certain goodness-of-fit criteria based on stochastic processes. Ann. Math. Stat. **23**, 193–212 (1952)

M. Comas-Cufí, S. Thió-Henestrosa, CoDaPack 2.0: a stand-alone, multi-platform compositional software, in *CoDaWork'11: 4th International Workshop on Compositional Data Analysis, Sant Feliu de Guíxols*, ed. by J.J. Egozcue, R. Tolosana-Delgado, M.I Ortego (2011). ISBN 978-84-87867-76-7

J.J. Egozcue, V. Pawlowsky-Glahn, G. Mateu-Figueras, C. Barceló-Vidal, Isometric logratio transformations for compositional data analysis. Math. Geol. **35**(3), 279–300 (2003)

J. Eichler, K. Hron, R. Tolosana-Delgado, K.G. van den Boogaart, M. Templ, P. Filzmoser, Graphical User Environment for Compositional Data Analysis. R package version 1.40-1 (2014), https://cran.r-project.org/src/contrib/Archive/compositionsGUI/

P. Filzmoser, StatDA: Statistical Analysis for Environmental Data. R package version 1.6.9 (2015), https://CRAN.R-project.org/package=StatDA

P. Filzmoser, M. Gschwandtner, mvoutlier: Multivariate Outlier Detection Based on Robust Methods. R package version 2.0.8 (2017), https://CRAN.R-project.org/package=mvoutlier

P. Filzmoser, K. Hron, Outlier detection for compositional data using robust methods. Math. Geosci. **40**(3), 233–248 (2008)

P. Filzmoser, K. Hron, C. Reimann, Principal component analysis for compositional data with outliers. Environmetrics **20**, 621–632 (2009)

P. Filzmoser, K. Hron, M. Templ, Discriminant analysis for compositional data and robust parameter estimation. J. Comput. Stat. **27**(4), 585–604 (2012)

E. Fišerová, K. Hron, On interpretation of orthonormal coordinates for compositional data. Math. Geosci. **43**(4), 455–468 (2011)

N. Hamilton, ggtern: An Extension to 'ggplot2', for the Creation of Ternary Diagrams. R package version 2.0.1 (2016), https://CRAN.R-project.org/package=ggtern

K. Hron, M. Templ, P. Filzmoser, Imputation of missing values for compositional data using classical and robust methods. Comput. Stat. Data Anal. **54**(12), 3095–3107 (2010)

K. Hron, P. Filzmoser, K. Thompson, Linear regression with compositional explanatory variables. J. Appl. Stat. **39**(5), 1115–1128 (2012)

J. Martín-Fernández, K. Hron, M. Templ, J. Palarea-Albaladejo, Model-based replacement of rounded zeros in compositional data: Classical and robust approaches. Comput. Stat. Data Anal. **56**(9), 2688–2704 (2012)

J. Palarea-Albaladejo, J.A. Martín-Fernández, zCompositions — R package for multivariate imputation of left-censored data under a compositional approach. Chemom. Intell. Lab. Syst. **143**, 85–96 (2015)

R Development Core Team, *R: A Language and Environment for Statistical Computing* (R Foundation for Statistical Computing, Vienna, 2017). http://www.R-project.org/, ISBN 3-900051-07-0.

J.H. Reynolds, D. Billheimer, Basic compositional data analysis functions for S+/R (2005), http://faculty.washington.edu/dmck/feradata/compositions.txt

E. Savazzi, R.A. Reyment, *Aspects of Multivariate Statistical Analysis in Geology* (Elsevier, Amsterdam, 1999)

M. Templ, K. Hron, P. Filzmoser, robCompositions: Robust Estimation for Compositional Data. R package version 1.5.0 (2011a), http://CRAN.R-project.org/package=robCompositions

M. Templ, K. Hron, P. Filzmoser, robCompositions: an R-package for robust statistical analysis of compositional data, in *Compositional Data Analysis: Theory and Applications*, ed. by V. Pawlowsky-Glahn, A. Buccianti (Wiley, Chichester, 2011b), pp. 341–355

M. Templ, A. Alfons, P. Filzmoser, Exploring incomplete data using visualization techniques. Adv. Data Anal. Classif. **6**(1), 29–47 (2012)

M. Templ, K. Hron, P. Filzmoser, A. Gardlo, Imputation of rounded zeros for high-dimensional compositional data. Chemom. Intell. Lab. Syst. **155**, 183–190 (2016)

S. Thió-Henestrosa, C. Barceló-Vidal, J.A. Martín-Fernández, V. Pawlowsky-Glahn, CoDaPack. An Excel and Visual Basic based software for compositional data analysis. Current version and discussion for upcoming versions, in *Compositional Data Analysis Workshop – CoDaWork'03, Proceedings, Girona*, ed. by S. Thió-Henestrosa, J.A. Martín-Fernández (2003)

S. Thió-Henestrosa, R. Tolosana-Delgado, O. Gómez, New features of CoDaPack—a compositional data package, in *Proceedings of IAMG'05 – The X. Annual Conference of the Internationl Association for Mathematical Geology*, ed. by Q. Cheng, G. Bonham-Carter, vol. 2 (2005), pp. 1171–1178

K.G. van den Boogaart, R. Tolosana-Delgado, M. Bren, compositions: Compositional Data Analysis. R package version 1.40-1 (2014), https://CRAN.R-project.org/package=compositions

H. Wickham, *ggplot2: Elegant Graphics for Data Analysis* (Springer, New York, 2009)

H. Wickham, W. Chang, devtools: Tools to Make Developing R Packages Easier. R package version 1.7.0 (2015), http://CRAN.R-project.org/package=devtools

Chapter 3
Geometrical Properties of Compositional Data

Abstract For an appropriate statistical processing it is essential to consider the inherent geometrical properties of the sample space of observations. In case of compositional data, this space is represented by equivalence classes of proportional vectors, possibly represented on the simplex, endowed with the Aitchison geometry. Its Euclidean vector space structure enables to construct coordinates with respect to a basis, eventually coefficients of a generating system. Here, isometric logratio coordinates, real coordinates with respect to an orthonormal basis in the Aitchison geometry, are preferable. As their name indicates, they are isometric with the Aitchison geometry, which makes it possible to proceed with standard statistical analyses in a meaningful way. For interpretation purposes, pivot coordinates that extract relative information about a compositional part in just one coordinate are taken as first option. In addition, also other alternatives are considered: symmetric pivot coordinates, which are suitable for a bivariate analysis, and particularly balance coordinates, which are interpretable in the sense of balances between groups of compositional parts. They can be intuitively constructed using sequential binary partitioning, and they form a family of general isometric logratio coordinates; also the preferable pivot coordinates can be taken as a special case.

3.1 Motivation

A frequent argument for the necessity of a special treatment of compositional data is that this kind of data is not coherent with the usual Euclidean geometry. Rather, compositional data follow the so-called Aitchison geometry on the simplex, see Sect. 3.2. But how is the "usual" Euclidean geometry defined, and what is the simplex?

In analytical geometry, a Euclidean space is associated with a vector space. Starting from an origin in the Euclidean space, one can reach a specific point by a vector in terms of an arrow, connecting the origin with this point. It is then possible to measure distances and angles in the Euclidean space with the help of the lengths of the arrows and the angles between them. This generates a vector space with a scalar product.

P. Filzmoser et al., *Applied Compositional Data Analysis*, Springer Series in Statistics, https://doi.org/10.1007/978-3-319-96422-5_3

Consider two vectors $\mathbf{x} = (x_1, \ldots, x_p)'$ and $\mathbf{y} = (y_1, \ldots, y_p)' \in \mathbb{R}^p$. Formally, it is necessary first to define the addition of two vectors

$$\mathbf{x} + \mathbf{y} = (x_1 + y_1, \ldots, x_p + y_p)'$$

and the multiplication of a vector by a real number α

$$\alpha\mathbf{x} = (\alpha x_1, \ldots, \alpha x_p)'.$$

Accordingly, the difference between two vectors results in

$$\mathbf{x} - \mathbf{y} = \mathbf{x} + (-1) \cdot \mathbf{y} = (x_1 - y_1, \ldots, x_p - y_p)'.$$

Both operations of addition and multiplication are geometrically very intuitive. Addition can be easily obtained using the well-known triangle rule: the tail of the second arrow is positioned on the head of the first. Their sum has the tail of the first as its tail and the head of the second as its head. Multiplication by a scalar can be interpreted in terms of stretching/shrinkage of the arrow.

The **(Euclidean) inner product** between $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sqrt{\mathbf{x}'\mathbf{y}} = \sqrt{\sum_{i=1}^{p} x_i y_i}.$$

As a special case, the inner product of $\mathbf{x}$ is defined as

$$\|\mathbf{x}\| = \sqrt{\mathbf{x}'\mathbf{x}} = \sqrt{\sum_{i=1}^{p} x_i^2}.$$

The expression $\|\mathbf{x}\|$ is also called the **Euclidean norm** of $\mathbf{x}$, which in fact is the length of the vector $\mathbf{x}$.

The **Euclidean distance** between $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\| = \sqrt{\sum_{i=1}^{p} (x_i - y_i)^2}.$$

The inner product function together with addition of two vectors and multiplication of a vector by a real number is sufficient to define the Euclidean geometry. In simple words, this Euclidean geometry corresponds to the geometrical space of our intuition. The most usual statistical methods are designed for this space, i.e., they rely on definitions of norm and distance following the Euclidean geometry (Eaton 1983).

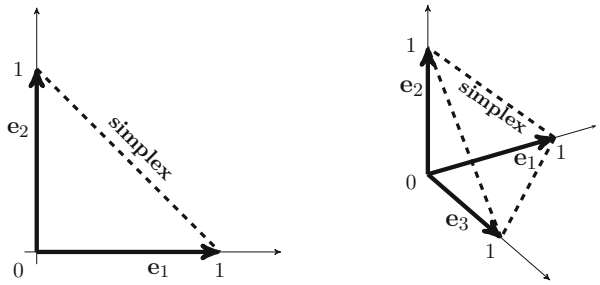


Fig. 3.1 1-standard simplex in $\mathbb{R}^2$ (left), and 2-standard simplex in $\mathbb{R}^3$ (right), shown by the dashed lines

In order to define the Aitchison geometry on the simplex, the meaning of a **simplex** needs to be explained. A simplex can be seen as a generalization of the notion of a triangle or a tetrahedron to higher dimensions. Consider a very specific simplex, the so-called $(D-1)$-standard simplex, a subset of $\mathbb{R}^D$, which is defined by

$$\left\{ \mathbf{x} = (x_1, \ldots, x_D)' \in \mathbb{R}^D \mid x_i \geq 0, \ \sum_{i=1}^{D} x_i = 1 \right\}. \tag{3.1}$$

The D vertices of this simplex are the unit vectors $\mathbf{e}_1 = (1, 0, \ldots, 0)'$, $\mathbf{e}_2 = (0, 1, 0, \ldots, 0)', \ldots, \mathbf{e}_D = (0, 0, \ldots, 0, 1)'$, which are vectors in $\mathbb{R}^D$. The $(D-1)$-standard simplex then forms a $(D-1)$-dimensional subset of $\mathbb{R}^D$.

The 1-standard simplex in $\mathbb{R}^2$ is visualized in Fig. 3.1 (left), and the 2-standard simplex in $\mathbb{R}^3$ is shown in Fig. 3.1 (right). Both representations are frequently used to illustrate the concepts of compositional data. Particularly, the 2-standard simplex leads to the **ternary diagram**, a very useful plot in this context—see Chap. 4.

The simplex is of particular interest for compositional data analysis, because it is widely referred to be the sample space of compositional data. Think about election data, where the votes for the different parties are expressed as proportions for a specific region. Then the numbers have to be non-negative, and they sum up to one. Geometrically, this observation is then located in a standard simplex.

This definition of the sample space would still be too narrow, since compositional data do not necessarily sum up to one, as thoroughly discussed in Chap. 1: The votes could as well be reported in absolute numbers, or simply some parts from the proportional representation could have been omitted. For this reason a more thorough definition of the sample space of compositions needs to be provided.

Consider a composition with D parts, say $\mathbf{x} = (x_1, \ldots, x_D)'$. For example, if the concentrations of $D = 10$ chemical elements in soil samples are measured, then $\mathbf{x}$ contains the concentrations of these compounds in a specific sample. If the data are expressed in mg/kg, the sum of the concentrations will typically not be one million mg/kg, since not all existing elements could be measured. Moreover, the sum for

another soil sample can be very different. Denote the sum of the concentrations of sample $\mathbf{x}$ by κ, so $\sum_{i=1}^{D} x_i = \kappa$, where κ can be any arbitrary positive real number, here treated as a constant value. Then the D-part simplex S^D is defined as

$$S^D = \left\{ \mathbf{x} = (x_1, \ldots, x_D)' \in \mathbb{R}^D \mid x_i > 0, \sum_{i=1}^{D} x_i = \kappa \right\}. \tag{3.2}$$

This space would thus cover all observations $\mathbf{x}$, the parts of which sum up to κ. For a practical data set, this might only be valid for one particular observation—the other observations will be characterized by other sums.

Note that there are some important differences to the definition in (3.1). Definition (3.1) assumes that the components of the observations, the compositional parts, sum up to one, while the definition in (3.2) is more general. The reason for that will be discussed below. Furthermore, in (3.1) the parts are non-negative, meaning that they can also be zero, while in (3.2) the compositional parts must be strictly positive. This is for sure a limitation, but it does not imply that compositional data analysis cannot deal with zeros—this problem will be treated in Chap. 13. The definition with strictly positive values is rather a convenience for a standard methodological treatment based on logratios, where zeros would lead to ill-defined values.

One might ask now why the sum κ is relevant here. In fact, as it will be shown in the following, the sum constraint κ is irrelevant, as a consequence of scale invariance, described in Chap. 1. It is always possible to rescale compositional data by multiplication of the parts with a positive constant without changing the information for the compositional analysis, contained in the logratios between the parts. In that sense, expressing compositional data with a constant sum constraint is just a matter of convenience. As an example, when plotting compositional data in the ternary diagram, see Chap. 4, the observations are often rescaled to have sum one. This allows for a comparison in the plot. There are some peculiarities resulting from the Aitchison geometry and relative nature of compositions, discussed in Chap. 4. Nevertheless, the important point is that this rescaling does not change the information contents for compositional data analysis, as it will be shown later.

Rescaling of compositions can be formalized by the so-called *closure operator* $\mathcal{C}$. Consider a composition $\mathbf{x} = (x_1, \ldots, x_D)' \in \mathbb{R}_+^D$, where $\mathbb{R}_+^D$ denotes the D-dimensional real space with strictly positive elements, so $x_i > 0$ for $i = 1, \ldots, D$. The *closure* of $\mathbf{x}$ to any positive number κ is defined as

$$\mathcal{C}_\kappa(\mathbf{x}) = \left(\frac{\kappa \cdot x_1}{\sum_{i=1}^{D} x_i}, \ldots, \frac{\kappa \cdot x_D}{\sum_{i=1}^{D} x_i} \right)'. \tag{3.3}$$

Thus, applying the closure operation to the composition $\mathbf{x}$ leads to a new compositional vector with the same number of elements. The parts of this new vector sum up to κ, the desired constant which has been selected to rescale $\mathbf{x}$. By setting $\kappa = 1$,

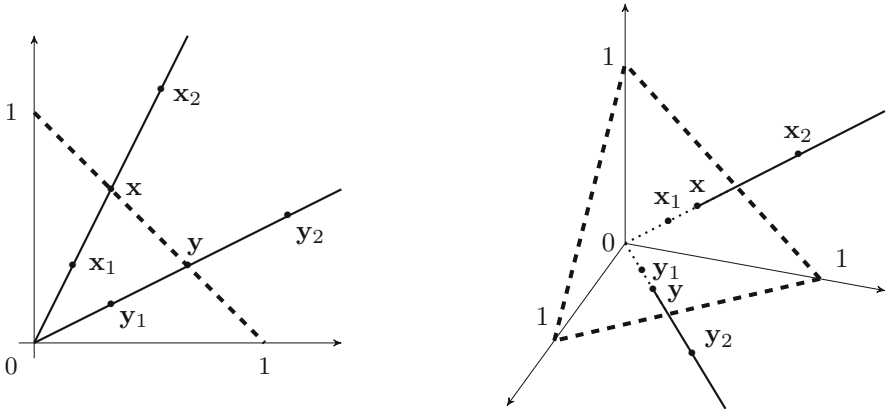


Fig. 3.2 Two-part (left) and three-part (right) compositions; the compositions $\mathbf{x}$, $\mathbf{x}_1$, and $\mathbf{x}_2$, as well as the compositions $\mathbf{y}$, $\mathbf{y}_1$, and $\mathbf{y}_2$, are compositionally equivalent, since they are located on one and the same ray. The projection of the vectors on the simplex, indicated by dashed lines, results in the compositions $\mathbf{x}$ and $\mathbf{y}$

it is now straightforward to rescale a composition $\mathbf{x}$ with any arbitrary sum of parts to a composition $\mathcal{C}_1(\mathbf{x})$ with component sum equal to one.

Consider two compositions $\mathbf{x}$ and $\mathbf{y}$, i.e. vectors in $\mathbb{R}_+^D$ that have any (possibly different) sum of compositional parts. Rescaling them with the same constant κ leads to the compositions $\mathcal{C}_\kappa(\mathbf{x})$ and $\mathcal{C}_\kappa(\mathbf{y})$, which now both have component sum κ. In general, the new composition $\mathcal{C}_\kappa(\mathbf{x})$ will be different from the composition $\mathcal{C}_\kappa(\mathbf{y})$. However, if $\mathcal{C}_\kappa(\mathbf{x}) = \mathcal{C}_\kappa(\mathbf{y})$, then the original compositions $\mathbf{x}$ and $\mathbf{y}$ are "equal" and differ only by a constant (scale factor). In that case, $\mathbf{x}$ and $\mathbf{y}$ are called **compositionally equivalent**.

The concept of compositional equivalence is illustrated in Fig. 3.2. The left picture shows two-part compositions, while the right picture explains the concept with three-part compositions. In both representations, the compositions denoted with "$\mathbf{x}$" are compositionally equivalent, and the same is true for those compositions denoted with "$\mathbf{y}$". So, $\mathcal{C}_1(\mathbf{x}_1) = \mathcal{C}_1(\mathbf{x}_2) = \mathbf{x}$ and $\mathcal{C}_1(\mathbf{y}_1) = \mathcal{C}_1(\mathbf{y}_2) = \mathbf{y}$. Accordingly, the sum of the parts of the compositions $\mathbf{x}$ and $\mathbf{y}$ is one. Compositional equivalence refers to any composition which is located on one ray through the origin. The projection of the rays onto the simplex defined by the unit vectors leads to the compositions $\mathbf{x}$ and $\mathbf{y}$ with sum one, and they can be viewed as proper representations of the corresponding equivalent compositions. Clearly, $\mathbf{x}$ and $\mathbf{y}$ are different from each other—just their sum of parts is the same. The projection onto the simplex makes it easier to compare the compositions, either on the line segment (Fig. 3.2, left) or in the triangle (Fig. 3.2, right) forming the ternary diagram.

Using the index κ with the closure operator $\mathcal{C}$ should emphasize the chosen representation with the constant κ. Note that the same could be done even for the D-part simplex S^D, see Eq. (3.2), where κ plays the role of a parameter as well. In subsequent sections it will turn out that the particular choice of κ is

irrelevant, and for this reason the operator is usually used without index in the literature. In fact, an emphasis on the constant sum constraint may sometimes even lead to confusion among practitioners, since this gives the impression that all concepts for compositional data analysis are just valid in case of constant sum of the compositional parts. Formally, this can be avoided by re-defining the sample space of compositions. Instead of the definition in (3.2) of the D-part simplex S^D as sample space of representations of compositions with a prescribed sum constraint κ, a new definition is given by

$$\tilde{S}^D = \left\{ \mathbf{x} = (x_1, \ldots, x_D)' \in \mathbb{R}_+^D \mid x_i > 0,\ \forall \kappa > 0\ \exists! \lambda > 0 :\ \mathbf{x} = \lambda \mathcal{C}_\kappa(\mathbf{x}) \right\}. \tag{3.4}$$

Interpreting Equation (3.4) in terms of Fig. 3.2 means that the sample space of a composition consists of the set of all complete rays from the origin, such that the parts are strictly positive. For example, $\mathbf{x} \in \tilde{S}^D$ refers to $\mathbf{x}_1$, to $\mathbf{x}_2$ and to $\mathbf{x}$ in Fig. 3.2—they are indistinguishable from a compositional point of view. The constant κ does not matter at all. In other words, the space $\tilde{S}^D$ refers to $\mathbb{R}_+^D$, decomposed according to equivalence classes of compositionally equivalent vectors. This concept will also be used in the following, which allows to avoid the closure operator.

3.2 Aitchison Geometry on the Simplex

It was argued in the previous section that compositional data do not follow the usual Euclidean geometry. The sample space of compositions is the simplex (in the sense of Eq. (3.4)), and thus an appropriate geometrical concept needs to be developed. In the pioneering work of Aitchison (1986) the geometrical perspective of compositional data analysis was not considered. This book follows Pawlowsky-Glahn and Egozcue (2001) and Egozcue et al. (2003), for which the geometrical structure of compositions is referred to as the *Aitchison geometry*. The aim is to define a vector space structure of the simplex, and for that some basic operations are needed. These correspond to the addition of two vectors, i.e. the shifting operation, and multiplication of a vector by a real number in the Euclidean geometry. In order to underline the difference between both geometries, also a special notation is applied.

- **Perturbation:** Consider two compositions $\mathbf{x}$ and $\mathbf{y}$ from the simplex sample space $\tilde{S}^D$. Then the perturbation of $\mathbf{x}$ by $\mathbf{y}$ is a composition defined as

$$\mathbf{x} \oplus \mathbf{y} = (x_1 y_1, x_2 y_2, \ldots, x_D y_D)'. \tag{3.5}$$

- **Powering:** The power transformation of a composition $\mathbf{x} \in \tilde{S}^D$ by a constant $\alpha \in \mathbb{R}$ is defined as

$$\alpha \odot \mathbf{x} = (x_1^\alpha, x_2^\alpha, \ldots, x_D^\alpha)'. \tag{3.6}$$

These operations are indeed sufficient to obtain a vector space, and the usual properties (commutative, associative, distributive) hold, see, e.g., Pawlowsky-Glahn et al. (2015). Particularly, a composition with identical parts forms the neutral element in $\tilde{S}^D$, denoted as $\mathbf{n}$ in the following. As a consequence, $\mathbf{n}$ has all pairwise logratios equal to zero and corresponds to the zero vector in the Euclidean geometry.

By applying both perturbation and powering, it is also possible to define the perturbation difference as

$$\mathbf{x} \ominus \mathbf{y} = \mathbf{x} \oplus [(-1) \odot \mathbf{y}] = (x_1/y_1, x_2/y_2, \ldots, x_D/y_D)'.$$

It follows that the difference between the same composition results in the neutral element, i.e.

$$\mathbf{x} \ominus \mathbf{x} = (x_1/x_1, x_2/x_2, \ldots, x_D/x_D)' = (1, 1, \ldots, 1)' = \mathbf{n}.$$

A Euclidean vector space structure can be obtained by defining norm, inner product, and distance in the Aitchison sense:

- **Aitchison inner product:** The inner product of two compositions $\mathbf{x} = (x_1, \ldots, x_D)'$ and $\mathbf{y} = (y_1, \ldots, y_D)'$ from $\tilde{S}^D$ is defined as

$$\langle \mathbf{x}, \mathbf{y} \rangle_A = \frac{1}{2D} \sum_{i=1}^{D} \sum_{j=1}^{D} \ln \frac{x_i}{x_j} \ln \frac{y_i}{y_j}. \tag{3.7}$$

- **Aitchison norm:** The norm of a composition $\mathbf{x} = (x_1, \ldots, x_D)' \in \tilde{S}^D$ is defined via the inner product of $\mathbf{x}$ with itself,

$$\|\mathbf{x}\|_A = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle_A} = \sqrt{\frac{1}{2D} \sum_{i=1}^{D} \sum_{j=1}^{D} \left(\ln \frac{x_i}{x_j} \right)^2}. \tag{3.8}$$

- **Aitchison distance:** The distance between $\mathbf{x}$ and $\mathbf{y} \in \tilde{S}^D$ is defined as

$$d_A(\mathbf{x}, \mathbf{y}) = \sqrt{\frac{1}{2D} \sum_{i=1}^{D} \sum_{j=1}^{D} \left(\ln \frac{x_i}{x_j} - \ln \frac{y_i}{y_j} \right)^2}. \tag{3.9}$$

These definitions lead to a Euclidean linear vector space structure, and in the literature this is simply denoted by the *Aitchison geometry*. The definitions are based on logarithms of ratios (logratios) between the compositional parts, and thus one refers to the logratio methodology for compositional data analysis. It is interesting to compare the definitions of Euclidean inner product, norm and distance, see Sect. 3.1, with the counterparts from the Aitchison geometry: in the former geometry

the original variables are used in the definitions, while in the latter case pairwise logratios are employed.

A further consequence of using logratios is that indeed the sum of the compositional parts is irrelevant: The compositions $\mathbf{x} = (x_1, \ldots, x_D)'$ and $\mathbf{x}_\lambda = (\lambda x_1, \ldots, \lambda x_D)'$, for any $\lambda > 0$, lead to the same logratios, and thus $\langle \mathbf{x}, \mathbf{x}_\lambda \rangle_A = \langle \mathbf{x}, \mathbf{x} \rangle_A = \|\mathbf{x}\|_A^2$, and $d_A(\mathbf{x}, \mathbf{x}_\lambda) = d_A(\mathbf{x}, \mathbf{x}) = 0$. Similar to the metrical concepts, also the results of perturbation and powering do not depend, up to constant sum representation, on the initial scaling of the input compositions $\mathbf{x}$ and $\mathbf{y}$.

It was already mentioned in Sect. 1.1 that using logratios instead of ratios themselves simplifies the interpretation and mathematical treatment. Using the property $\ln \frac{x_i}{x_j} = \ln(x_i) - \ln(x_j)$, the Aitchison distance (3.9) can be rewritten as

$$d_A(\mathbf{x}, \mathbf{y}) = \sqrt{\frac{1}{2D} \sum_{i=1}^{D} \sum_{j=1}^{D} \left(\ln \frac{x_i}{y_i} - \ln \frac{x_j}{y_j} \right)^2}, \tag{3.10}$$

which supports the concept of relative scale. Accordingly, the sources of difference between the compositions $\mathbf{x}$ and $\mathbf{y}$ are contained in the logratios between the corresponding parts. Together with (3.9) it is also visible that any such difference between the parts automatically influences the relations between the other parts through logratios within single compositions.

The next example motivates a further important point: the dimensionality of compositions.

Example Consider a composition with three parts, say $\mathbf{x} = (1, 2, 10)'$. This could correspond to a recipe of a fresh drink, with 1 "unit" lemon, 2 "units" sugar, and 10 "units" cold water. The composition determines the taste of the drink, but not the absolute amount or the size of the "units". In that sense, the sum of the compositional parts does not matter for the taste. What matters is the proportions between the compositional parts, i.e. the ratio lemon to sugar (1/2), the ratio lemon to water (1/10), and the ratio sugar to water (2/10). The latter ratio is already determined by the first two ratios, since lemon/sugar divided by lemon/water is equal to water/sugar. In other words, without any loss of information it is possible to express information of 3 parts by only 2 ratios—in the Aitchison geometry these will not be ratios but logratios.

In general, for a D-part composition $\mathbf{x}$ there exist $D(D-1)$ combinations of nonzero logratios. Since $\ln \frac{x_i}{x_j} = -\ln \frac{x_j}{x_i}$, this number can be reduced to $D(D-1)/2$ different (up to sign) combinations of logratios. However, it is always possible to find $D-1$ logratios such that all the remaining logratios can be expressed, using the relation

$$\ln \frac{x_i}{x_k} = \ln \frac{x_i}{x_j} + \ln \frac{x_j}{x_k}, \quad \text{for } i, j, k = 1, \ldots, D.$$

An important example to define these $D-1$ logratios is

$$\ln \frac{x_1}{x_D}, \ln \frac{x_2}{x_D}, \ldots, \ln \frac{x_{D-1}}{x_D},$$

which is used in the next section in the context of logratio coordinates. Another example is

$$\ln \frac{x_1}{x_2}, \ln \frac{x_2}{x_3}, \ldots, \ln \frac{x_{D-1}}{x_D}.$$

This feature can also intuitively explain why the sum of $D(D-1)$ nonzero logratios in the Aitchison inner product (norm, distance) is divided by D: the total outcome is simply "penalized" by redundant information. Moreover, there are also geometrical implications: the D-part simplex S^D demonstrates that D-part compositions can always be represented within a $(D-1)$-dimensional subspace, without any loss of information.

Summarizing, in case of D-part compositions, the Aitchison geometry has dimension $D-1$ (Pawlowsky-Glahn et al. 2015), i.e., any basis formed by the compositions contains $D-1$ elements. This fact turns out to be crucial for the next section: D-part compositions will be expressed as coordinates in a $(D-1)$-dimensional real space.

3.3 Coordinate Representations of Compositions

Compositional data analysis is frequently associated with applying an appropriate transformation first, and then employing the standard statistical methodology as usual. Although from a practical point of view this is true in many cases, the difficulty with this kind of thinking is the interpretation of the results. After applying a transformation, one does no longer work with the original compositions but with transformations thereof, and the interpretation of the results has to be adapted accordingly. For this purpose, however, one needs to understand the meaning and the purpose of the transformation.

A transformation can also be viewed as expressing the compositions in a coordinate system with respect to the Aitchison geometry. This geometrical view helps a lot to understand the interpretation and limitations of various coordinate systems. It forms also a principal difference to other classes of transformations, like those mentioned in Aitchison (1986) and those which have been presented recently in the literature (Scealy and Welsh 2011, 2015; Stewart and Field 2011). Note that even in Aitchison (1986) the coordinate-based approach has not yet been discussed as it was not possible until the Aitchison geometry has been introduced.

Accordingly, the goal of this section is to explain transformations in terms of coordinate representations. Here the focus is on the so-called *logratio coordinates*

that express compositional data, driven by the Aitchison geometry, in the usual Euclidean geometry of the real space. Due to the close link of the logratio coordinates to the Aitchison geometry, the principles of compositional data analysis (Sect. 1.3) are automatically fulfilled. Though, one should be aware, as the name of the coordinates indicates, that the new variables will contain exclusively logratios of the original compositional parts. It forms a source of certain peculiarities, thoroughly discussed in the following.

In this approach, not just pairwise logratios, as considered up to now, are used, but also aggregations. For example, by summing up the logratios $\ln \frac{x_1}{x_3}$ and $\ln \frac{x_2}{x_4}$, one gets a new variable $z_1 = \ln \frac{x_1 x_2}{x_3 x_4}$. The sum of $\ln \frac{x_1}{x_2}$, $\ln \frac{x_1}{x_3}$, and $\ln \frac{x_1}{x_4}$ results in $z_2 = \ln \frac{x_1^3}{x_2 x_3 x_4}$. Both z_1 and z_2 represent special cases of a logcontrast (Aitchison 1986), which is a linear combination $\sum_{i=1}^{D} c_i \ln x_i$ of log-transformed compositional parts such that $c_1 + c_2 + \ldots + c_D = 0$. For z_1, the coefficients are $c_1 = c_2 = 1$, $c_3 = c_4 = -1$, while in the case of z_2 they are $c_1 = 3$, $c_2 = c_3 = c_4 = -1$. Accordingly, any of the following coordinate representations can be expressed in terms of logcontrasts as well. Nevertheless, in order to keep the practical focus of the book, this idea will not be further developed.

For historical reasons, the overview of different choices of coordinates starts with additive logratio coordinates and centered logratio coefficients, introduced already in Aitchison (1982, 1983) and summarized in Aitchison (1986). Nevertheless, the main focus will be devoted to isometric logratio coordinates that will be used for most methods presented in the book.

3.3.1 Additive Logratio (alr) Coordinates

This is a mapping from $\tilde{S}^D$ to $\mathbb{R}^{D-1}$, and the result for an observation $\mathbf{x} \in \tilde{S}^D$ are coordinates $\mathbf{x}^{(j)} \in \mathbb{R}^{D-1}$ with

$$\mathbf{x}^{(j)} = \mathrm{alr}_j(\mathbf{x}) = (x_1^{(j)}, \ldots, x_{D-1}^{(j)})' = \left(\ln \frac{x_1}{x_j}, \ldots, \ln \frac{x_{j-1}}{x_j}, \ln \frac{x_{j+1}}{x_j}, \ldots, \ln \frac{x_D}{x_j} \right)'. \tag{3.11}$$

If an $n \times D$ matrix $\mathbf{X}$ of compositional data is given, with the compositions $\mathbf{x}_i' = (x_{i1}, \ldots, x_{iD})$ in the rows of $\mathbf{X}$, for $i = 1, \ldots, n$, then the matrix of alr coordinates is formed by the rows

$$\left(\mathbf{x}_i^{(j)}\right)' = \left(\mathrm{alr}_j(\mathbf{x}_i)\right)' = \left(\ln \frac{x_{i1}}{x_{ij}}, \ldots, \ln \frac{x_{i,j-1}}{x_{ij}}, \ln \frac{x_{i,j+1}}{x_{ij}}, \ldots, \ln \frac{x_{iD}}{x_{ij}} \right). \tag{3.12}$$

The index $j \in \{1, \ldots, D\}$ refers to the variable that is chosen as ratioing variable in the coordinates. This choice usually depends on the context, but also

on the suitability of the results for visualization and data exploration. The main practical disadvantages of alr coordinates are the subjectivity of the choice of the ratioing variable, and the fact that alr leads to a non-orthogonal coordinate system (Pawlowsky-Glahn et al. 2015). Accordingly, although alr coordinates move the operations of perturbation and powering to the standard vector addition and multiplication,

$$\mathrm{alr}_j(\mathbf{x} \oplus \mathbf{y}) = \mathrm{alr}_j(\mathbf{x}) + \mathrm{alr}_j(\mathbf{y}), \quad \mathrm{alr}_j(c \odot \mathbf{x}) = c \cdot \mathrm{alr}_j(\mathbf{x})$$

for $\mathbf{x}, \mathbf{y} \in \tilde{S}^D$, $c \in \mathbb{R}$ and any $j \in \{1, \ldots, D\}$, this is in general not fulfilled for the Aitchison inner product, norm and distance, e.g., $\langle \mathbf{x}, \mathbf{y} \rangle_A \neq \langle \mathrm{alr}_j(\mathbf{x}), \mathrm{alr}_j(\mathbf{y}) \rangle$. Moreover, the interpretation of alr coordinates would be misleading if they were interpreted in terms of the original parts. For example, the first component of $\mathbf{x}^{(j)}$ is $\ln \frac{x_1}{x_j}$, and it contains relative information of x_1 only to the j-th part, but not to all the other parts. From another perspective, $\mathbf{x}^{(j)}$ contains the relative information of part x_j to all remaining parts (note that $\ln \frac{x_i}{x_j} = -\ln \frac{x_j}{x_i}$). This information, however, is distributed among all components of $\mathbf{x}^{(j)}$ and not devoted to just one coordinate, and thus the interpretation of a particular coordinate cannot be made in terms of one part.

Since the alr coordinates form a one-to-one mapping from $\tilde{S}^D$ to $\mathbb{R}^{D-1}$, it is also possible to get back to the original compositional data as

$$\begin{aligned} x_i &= \exp\left(x_i^{(j)}\right) \quad \text{for } i = 1, \ldots, D,\ i \neq j, \\ x_j &= 1 \quad \text{for } j \in \{1, \ldots, D\}. \end{aligned} \tag{3.13}$$

Note that the sum of the back-transformed parts $x_1 + \ldots + x_D$ will in general not be equal to the sum of the original parts, neither will it be 1. However, since the sum does not matter from a compositional data analysis point of view, scaling is omitted in Eq. (3.13).

As already indicated, alr coordinates are mentioned here rather for historical reasons and for completeness. They will not be in focus in the subsequent chapters.

3.3.2 Centered Logratio (clr) Coefficients

A composition $\mathbf{x} \in \tilde{S}^D$ is expressed by a vector $\mathbf{y} \in \mathbb{R}^D$, with

$$\mathbf{y} = \mathrm{clr}(\mathbf{x}) = (y_1, \ldots, y_D)' = \left(\ln \frac{x_1}{\sqrt[D]{\prod_{k=1}^D x_k}}, \ldots, \ln \frac{x_D}{\sqrt[D]{\prod_{k=1}^D x_k}} \right)'. \tag{3.14}$$

For an $n \times D$ matrix $\mathbf{X}$ of compositional data, with the compositions $\mathbf{x}_i' = (x_{i1}, \ldots, x_{iD})$ in the rows of $\mathbf{X}$, for $i = 1, \ldots, n$, the matrix of clr coefficients $\mathbf{Y}$ is formed by the rows

$$\mathbf{y}_i' = (\text{clr}(\mathbf{x}_i))' = \left(\ln \frac{x_{i1}}{\sqrt[D]{\prod_{k=1}^{D} x_{ik}}}, \ldots, \ln \frac{x_{iD}}{\sqrt[D]{\prod_{k=1}^{D} x_{ik}}} \right). \tag{3.15}$$

The denominator used in (3.14) is called the **geometric mean**,

$$g_m(\mathbf{x}) = \sqrt[D]{\prod_{k=1}^{D} x_k} = \exp\left(\frac{1}{D} \sum_{k=1}^{D} \ln\, x_k \right). \tag{3.16}$$

In the sample formulation (3.15), the geometric mean used in the denominator is calculated for each individual observation.

At first glance, the difference between alr and clr is that clr avoids the subjectivity of the choice of the denominator by using the geometric mean, which treats the components symmetrically. Further, clr ends up with D components instead of only $D - 1$ for alr. However, these D components sum up to zero, since

$$\begin{aligned} \sum_{j=1}^{D} y_j &= \sum_{j=1}^{D} \ln \frac{x_j}{\exp\left(\frac{1}{D} \sum_{k=1}^{D} \ln\, x_k \right)} = \sum_{j=1}^{D} \left(\ln\, x_j - \frac{1}{D} \sum_{k=1}^{D} \ln\, x_k \right) \\ &= \sum_{j=1}^{D} \ln\, x_j - \frac{1}{D} D \sum_{k=1}^{D} \ln\, x_k = 0. \end{aligned}$$

In order to emphasize this peculiarity, we refer to clr *coefficients* (instead of coordinates). From a geometrical point of view, there is one more composition than necessary to form the basis in the Aitchison geometry, being just of dimension $D-1$. Therefore, $\mathbf{y}$ represents coefficients with respect to a *generating system* (instead of a basis). For details see, e.g., Pawlowsky-Glahn et al. (2015). This means that there is not a unique possibility how to form coefficients with respect to the same system of compositions. This feature has one important consequence: it is not possible to consider just one of the clr coefficients for the analysis without taking also the others into account. This is a serious limitation, e.g. for univariate data analysis, that can be overcome by using coordinates with respect to an orthonormal basis.

A practical implication of the zero sum of clr coefficients is that when using them one ends up with constrained data. Thinking about a data matrix of compositions, then after expressing each observation in clr coefficients, the resulting matrix has not full rank in the columns and the corresponding covariance matrix is singular.

Looking more closely at the resulting clr coefficients, one can see that logratios to all parts are involved. For example, coefficient y_1 can be written as

$$y_1 = \ln \frac{x_1}{\sqrt[D]{\prod_{i=1}^{D} x_i}} = \ln \frac{x_1}{x_1^{\frac{1}{D}} x_2^{\frac{1}{D}} \cdots x_D^{\frac{1}{D}}}$$
$$= \ln \frac{x_1^{\frac{1}{D}}}{x_1^{\frac{1}{D}}} + \ln \frac{x_1^{\frac{1}{D}}}{x_2^{\frac{1}{D}}} + \ldots + \ln \frac{x_1^{\frac{1}{D}}}{x_D^{\frac{1}{D}}} = \frac{1}{D} \left(\ln \frac{x_1}{x_2} + \ldots + \ln \frac{x_1}{x_D} \right).$$

Thus, the logratios of part x_1 to all other parts are involved in terms of an average, with a "scaling factor" $1/D$. Accordingly, each logratio contributes with the same weight to the first coefficient y_1—similarly for the other components. Among different coefficients, however, there is "overlap": For example, part x_1 is not exclusively associated with y_1, but there is also logratio information with x_1 in $y_2, y_3, \ldots, y_D$. In other words, although all relative information about one compositional part within a given composition can be exclusively devoted to one particular clr coefficient (as it will be seen later on), one cannot interpret $y_1, \ldots, y_D$ in terms of the compositional parts $x_1, \ldots, x_D$ *simultaneously*. This is a frequent mistake made in practice! As a prominent example, the analysis of the correlation structure of clr coefficients might become completely misleading. This will be made clearer when introducing special isometric logratio coordinates, called pivot coordinates (see next subsection).

There is another interesting view of the clr coefficients: The first coefficient y_1, for instance, can be represented as

$$y_1 = \ln \frac{x_1}{g_m(\mathbf{x})} = \ln x_1 - \ln g_m(\mathbf{x}) = \ln x_1 - \frac{1}{D} \sum_{i=k}^{D} \ln x_k.$$

Thus, one obtains essentially a log transformation (by ln), but the resulting observation is centered. Thinking about a compositional data matrix $\mathbf{X}$ with the compositions in the rows, then the clr coefficients are essentially $\ln(\mathbf{X})$, with subsequent row-centering. This might be an interesting aspect for communities like geochemistry, where the log transformation is frequently applied to symmetrize the right-skewed data distributions.

Also the clr coefficients represent a one-to-one mapping, and thus it is possible to come back again to the original parts—up to a scaling factor,

$$x_j = \exp(y_j) \quad \text{for } j = 1, \ldots, D. \tag{3.17}$$

The link between log transformation and clr coefficients has been outlined above. One can formalize this differently. Consider the matrix

$$\mathbf{W} = \begin{pmatrix} 1 & 0 \cdots 0 \\ 0 & 1 \quad \vdots \\ \vdots & \ddots \\ 0 & 0 \cdots 1 \end{pmatrix} - \frac{1}{D} \begin{pmatrix} 1 & 1 \cdots 1 \\ 1 & 1 \cdots 1 \\ \vdots & \vdots \quad \vdots \\ 1 & 1 \cdots 1 \end{pmatrix} = \mathbf{I}_D - \frac{1}{D} \mathbf{1}_D \mathbf{1}_D', \tag{3.18}$$

where $\mathbf{I}_D$ is the identity matrix of dimension D, and $\mathbf{1}_D = (1, \ldots, 1)'$ a vector of length D with entries of 1. It can be proven (Aitchison 1986) that the relation

$$\mathbf{y} = \text{clr}(\mathbf{x}) = \mathbf{W} \cdot \ln(\mathbf{x})$$

holds.

The clr coefficients also fulfill further important properties (Egozcue et al. 2003). For two compositions $\mathbf{x}_1$ and $\mathbf{x}_2 \in \tilde{S}^D$ and $c \in \mathbb{R}$ it holds that

(1) $\text{clr}(\mathbf{x}_1 \oplus \mathbf{x}_2) = \text{clr}(\mathbf{x}_1) + \text{clr}(\mathbf{x}_2), \quad \text{clr}(c \odot \mathbf{x}_1) = c \cdot \text{clr}(\mathbf{x}_1);$

(2) $\langle \mathbf{x}_1, \mathbf{x}_2 \rangle_A = \langle \text{clr}(\mathbf{x}_1), \text{clr}(\mathbf{x}_2) \rangle, \quad \|\mathbf{x}_1\|_A = \|\text{clr}(\mathbf{x}_1)\|;$

(3) $d_A(\mathbf{x}_1, \mathbf{x}_2) = d(\text{clr}(\mathbf{x}_1), \text{clr}(\mathbf{x}_2)).$

While property (1), representing the linearity of the mapping, was fulfilled already with the alr coordinates, now also other important features are valid. Property (2) refers to the Aitchison inner product between the two compositions, see Eq. (3.7), which is the same as the usual inner product of the clr-quantities (similarly also for the Aitchison norm). Property (3) says that the Aitchison distance, see Eq. (3.9), between the two compositions is the same as the Euclidean distance between the compositions in clr coefficients. These two properties are important since they show that the clr coefficients represent an **isometry**: all metric concepts in the simplex are maintained after taking the clr coefficients.

3.3.3 Isometric Logratio (ilr) and Pivot Coordinates

While the clr coefficients map a composition $\mathbf{x}$ from $\tilde{S}^D$ to a $(D-1)$-dimensional hyperplane in $\mathbb{R}^D$, the class of isometric logratio (ilr) coordinates aims at building an orthonormal basis in this hyperplane and expressing the composition therein. The resulting vector $\mathbf{z}$ is in $\mathbb{R}^{D-1}$, and the practical implication is that one avoids the singularity issue that occurred with clr coefficients. In other words, ilr coordinates set up an orthonormal basis in the hyperplane formed by clr coefficients, and there are infinitely many possibilities to define such an orthonormal basis system. For this

reason, ilr is considered as a *class* of coordinates and it is also common to refer to *orthonormal* (logratio) coordinates. One particular choice of a basis leads to

$$\text{ilr}(\mathbf{x}) = \mathbf{z} = (z_1, \ldots, z_{D-1})'$$

with

$$z_j = \sqrt{\frac{D-j}{D-j+1}} \ln \frac{x_j}{\sqrt[D-j]{\prod_{k=j+1}^{D} x_k}} \quad \text{for} \quad j = 1, \ldots, D-1. \tag{3.19}$$

(Fišerová and Hron 2011). From now on, these ilr coordinates will be referred to as *pivot* (logratio) *coordinates*. The reason for such a notation is intuitive: one part (here x_1) is set to be a pivot, it is contained just in the first coordinate. As it becomes clear soon, such a choice has also a primary importance for the coordinate system as a whole.

For an $n \times D$ matrix $\mathbf{X}$ of compositional data, with the compositions $\mathbf{x}_i' = (x_{i1}, \ldots, x_{iD})$ in the rows of $\mathbf{X}$, for $i = 1, \ldots, n$, the $n \times (D-1)$ matrix of pivot coordinates $\mathbf{Z}$ is formed by the elements with index (i, j)

$$z_{ij} = \sqrt{\frac{D-j}{D-j+1}} \ln \frac{x_{ij}}{\sqrt[D-j]{\prod_{k=j+1}^{D} x_{ik}}}. \tag{3.20}$$

Throughout the rest of the book, the notation ilr($\mathbf{x}$) and the letter "$\mathbf{z}$" for the resulting coordinates refer to the pivot coordinates as defined in (3.19), or—depending on the context—also to general ilr (orthonormal) coordinates.

The definition of pivot coordinates (3.19) can be made more explicit,

$$\begin{aligned}
z_1 &= \sqrt{\frac{D-1}{D}} \ln \frac{x_1}{\sqrt[D-1]{\prod_{k=2}^{D} x_k}} \\
z_2 &= \sqrt{\frac{D-2}{D-1}} \ln \frac{x_2}{\sqrt[D-2]{\prod_{k=3}^{D} x_k}} \\
&\vdots \\
z_{D-2} &= \sqrt{\frac{2}{3}} \ln \frac{x_{D-2}}{\sqrt{x_{D-1} x_D}} \\
z_{D-1} &= \sqrt{\frac{1}{2}} \ln \frac{x_{D-1}}{x_D}.
\end{aligned}$$

As mentioned above, the pivot coordinates have the feature that part x_1 only appears in coordinate z_1. This is not the case for other parts; x_2, for example, appears in z_1 and in z_2. Isolating one part into one coordinate is appealing, since z_1 summarizes now all relative information (logratios) about x_1,

$$z_1 = \sqrt{\frac{D-1}{D}} \ln \frac{x_1}{\sqrt[D-1]{\prod_{k=2}^{D} x_k}} = \sqrt{\frac{1}{D(D-1)}} \left(\ln \frac{x_1}{x_2} + \ln \frac{x_1}{x_3} + \ldots + \ln \frac{x_1}{x_D} \right), \tag{3.21}$$

and can thus be interpreted as the relative dominance of x_1 within the given composition. In other words, by considering the resulting form of z_1 with the geometric mean in the denominator, this coordinate expresses the level of dominance of part x_1 with respect to the other parts "on average." Accordingly, for positive values of z_1, the first part dominates in the composition with respect to an "average part" (formed by the geometric mean) and vice versa for $z_1 < 0$. Finally, $z_1 = 0$ indicates a balanced state between x_1 and an average behavior of the other parts in the given composition.

No other part can be interpreted in such a manner, and thus the definition of pivot coordinates in (3.19) is specifically designed in favor of an interpretation for the first part. The scaling constants in (3.19) guarantee orthonormality of the resulting coordinate system.

Note that the first pivot coordinate z_1, see (3.21), and the first clr coefficient y_1, see (3.14), are proportional up to a scaling factor depending just on the dimension D, i.e.

$$z_1 = \sqrt{\frac{D}{D-1}} y_1.$$

Thus, also y_1 can be interpreted like z_1 in terms of the relative dominance of x_1 in the composition. Note, however, that $\mathbf{z} = (z_1, z_2, \ldots, z_{D-1})'$ are coordinates of an orthonormal basis, which is not the case for the coefficients $\mathbf{y} = (y_1, y_2, \ldots, y_D)'$. Moreover, the part x_1 is not contained exclusively in y_1, but also in the other clr coefficients. This leads to an overlap of the relative information conveyed by the clr variables, and intuitively also to the mentioned singularity of their covariance matrix. For this reason, by considering y_1 separately, an important feature, formed by the zero sum constraint of the clr coefficients, is omitted. This is not the case for z_1 as variable in an orthonormal coordinate system. Therefore, although the relation between z_1 and y_1 might seem to be just a matter of scaling, this feature has also important consequences for further methodological developments.

Like for clr coefficients, also pivot coordinates represent a one-to-one mapping. It is thus possible to come back to the original parts by

$$
\begin{aligned}
x_1 &= \exp\left(\frac{\sqrt{D-1}}{\sqrt{D}} z_1\right), \\
x_j &= \exp\left(-\sum_{k=1}^{j-1} \frac{1}{\sqrt{(D-k+1)(D-k)}} z_k + \frac{\sqrt{D-j}}{\sqrt{D-j+1}} z_j\right), \; j = 2, \ldots, D-1, \\
x_D &= \exp\left(-\sum_{k=1}^{D-1} \frac{1}{\sqrt{(D-k+1)(D-k)}} z_k\right),
\end{aligned} \tag{3.22}
$$

up to a scaling factor.

As mentioned previously, clr coefficients map a composition $\mathbf{x} \in \tilde{S}^D$ to a hyperplane $\mathcal{H} : \; y_1 + \ldots + y_D = 0$, i.e., to a subspace of $\mathbb{R}^D$, and the ilr coordinates are formed by coefficients expressing $\mathbf{x}$ in an orthonormal basis of this hyperplane. The orthonormal basis vectors corresponding to the pivot coordinates defined in Eq. (3.19) are

$$
\mathbf{v}_{\cdot j} = \sqrt{\frac{D-j}{D-j+1}} \left(0, \ldots, 0, 1, -\frac{1}{D-j}, \ldots, -\frac{1}{D-j}\right)' \tag{3.23}
$$

for $j = 1, \ldots, D-1$, with $j-1$ zero entries. These vectors, collected as columns in a $D \times (D-1)$ matrix $\mathbf{V} = (\mathbf{v}_{\cdot 1}, \ldots, \mathbf{v}_{\cdot D-1})$, are formed by clr coefficients of the original compositional basis. Then one immediately gets the relations between clr coefficients and pivot coordinates as

$$
\mathbf{y} = \mathbf{V}\mathbf{z} \quad \text{and} \quad \mathbf{z} = \mathbf{V}'\mathbf{y}, \tag{3.24}
$$

see Egozcue et al. (2003). Of course, such a linear relation holds in general also for any ilr coordinates, not just for the pivot ones.

Like clr coefficients, also ilr coordinates represent an *isometry* (Egozcue et al. 2003). For two compositions $\mathbf{x}_1$ and $\mathbf{x}_2 \in \tilde{S}^D$ and $c \in \mathbb{R}$ it holds that

1. $\operatorname{ilr}(\mathbf{x}_1 \oplus \mathbf{x}_2) = \operatorname{ilr}(\mathbf{x}_1) + \operatorname{ilr}(\mathbf{x}_2), \quad \operatorname{ilr}(c \odot \mathbf{x}_1) = c \cdot \operatorname{ilr}(\mathbf{x}_1)$;

2. $\langle \mathbf{x}_1, \mathbf{x}_2 \rangle_A = \langle \operatorname{ilr}(\mathbf{x}_1), \operatorname{ilr}(\mathbf{x}_2) \rangle, \quad \|\mathbf{x}_1\|_A = \|\operatorname{ilr}(\mathbf{x}_1)\|$;

3. $d_A(\mathbf{x}_1, \mathbf{x}_2) = d(\operatorname{ilr}(\mathbf{x}_1), \operatorname{ilr}(\mathbf{x}_2))$.

Thus, all metric concepts in the simplex are maintained after taking the ilr coordinates.

3.3.4 Special Coordinate Systems: Generalization of Pivot Coordinates

The pivot coordinates introduced in Sect. 3.3.3 support an interpretation especially of the first compositional part x_1, because the first coordinate exclusively describes all relative information about x_1. This special role of the first part is not necessarily given in a practical data set. However, it can still be of interest to obtain a specific interpretation for a single part within a given composition which is not the first part. In that case, one can simply permute the compositional parts in a way that the part of interest is placed at the first position, and the pivot coordinates (3.19) are constructed for the permuted composition.

Suppose that the interest for the interpretation is in part x_l, where $l \in \{1, \ldots, D\}$. Then the original composition $\mathbf{x} = (x_1, \ldots, x_D)'$ is replaced by the permuted composition

$$\mathbf{x}^{(l)} = (x_l, x_1, \ldots, x_{l-1}, x_{l+1}, \ldots, x_D)' =: (x_1^{(l)}, x_2^{(l)}, \ldots, x_l^{(l)}, x_{l+1}^{(l)}, \ldots, x_D^{(l)})'.$$

The pivot coordinates corresponding to Eq. (3.19) for the permuted composition are

$$z_j^{(l)} = \sqrt{\frac{D-j}{D-j+1}} \ln \frac{x_j^{(l)}}{\sqrt[D-j]{\prod_{k=j+1}^{D} x_k^{(l)}}} \quad \text{for} \quad j = 1, \ldots, D-1, \tag{3.25}$$

defining the coordinates $\mathbf{z}^{(l)} = (z_1^{(l)}, \ldots, z_{D-1}^{(l)})'$. From the Definition (3.19) it is clear that $z_j^{(1)} = z_j$, for $j = 1, \ldots, D-1$. Note that here only part x_l is put to the first position, and the order of the remaining parts is unchanged. In fact, this order of the remaining parts is irrelevant, since the focus here is on $z_1^{(l)}$, which explains all relative information about part x_l.

Similar as before, also for the (generalized) pivot coordinates their sample version can be presented. For an $n \times D$ matrix $\mathbf{X}$ of compositional data, with the compositions $\mathbf{x}_i' = (x_{i1}, \ldots, x_{iD})$ in the rows of $\mathbf{X}$, for $i = 1, \ldots, n$, the $n \times (D-1)$ matrix of pivot coordinates $\mathbf{Z}^{(l)}$ with emphasis on part x_l, $l = 1, \ldots, D$, is formed by the elements with index (i, j),

$$z_{ij}^{(l)} = \sqrt{\frac{D-j}{D-j+1}} \ln \frac{x_{ij}^{(l)}}{\sqrt[D-j]{\prod_{k=j+1}^{D} x_{ik}^{(l)}}}, \tag{3.26}$$

where $x_{ij}^{(l)}$ is the element with index (i, j) of the i-th row of the permuted data matrix,

$$(x_{il}, x_{i1}, \ldots, x_{i,l-1}, x_{i,l+1}, \ldots, x_{iD}) =: (x_{i1}^{(l)}, x_{i2}^{(l)}, \ldots, x_{il}^{(l)}, x_{i,l+1}^{(l)}, \ldots, x_{iD}^{(l)}).$$

The above permutation can be obtained via a permutation matrix $\mathbf{P}^{(l)}$, which is a $D \times D$ matrix with entries of 0/1. For example, if the composition $\mathbf{x} = (x_1, x_2, x_3)'$ is considered, and the interest of interpretation is in part x_3, then the permuted composition $\mathbf{x}^{(3)} = (x_3, x_1, x_2)'$ is obtained by

$$\mathbf{x}^{(3)} = \begin{pmatrix} x_3 \\ x_1 \\ x_2 \end{pmatrix} = \mathbf{P}^{(3)}\mathbf{x} = \begin{pmatrix} 0\,0\,1 \\ 1\,0\,0 \\ 0\,1\,0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

With this permutation matrix it is possible to define the new basis for the pivot coordinates $\mathbf{z}^{(l)}$. The orthonormal basis for the unpermuted composition was defined in (3.23), and the basis for the permuted composition is

$$\mathbf{V}^{(l)} = \mathbf{P}^{(l)}\mathbf{V}. \tag{3.27}$$

Thus, the rows of $\mathbf{V}$ are permuted in the same way as the parts. Consider the matrix

$$\mathbf{Q}^{(l)} = (\mathbf{V}^{(l)})'\mathbf{V}; \tag{3.28}$$

this is an orthonormal matrix, $(\mathbf{Q}^{(l)})'\mathbf{Q}^{(l)} = \mathbf{Q}^{(l)}(\mathbf{Q}^{(l)})' = \mathbf{I}_{D-1}$ (Egozcue et al. 2003). Then the coordinates for the permuted composition are

$$\mathbf{z}^{(l)} = \mathbf{Q}^{(l)}\mathbf{z} = \mathbf{V}'(\mathbf{P}^{(l)})'\mathbf{V}\mathbf{z}. \tag{3.29}$$

Equation (3.29) shows an elegant way how the basis can be changed to express compositions in a different orthonormal basis system that allows for a concise interpretation of the l-th compositional part. This will be used in the methodological chapters, when the interest is in the interpretation of single parts within a given composition. Moreover, Eq. (3.29) demonstrates that another choice of the pivot coordinate system is just a rotation of the original one. This important property holds also in general for any two ilr coordinate systems.

Similar as in the previous section for the special case of z_1, also the relation between $z_1^{(l)}$ and clr coordinates y_l can be generalized as

$$z_1^{(l)} = \sqrt{\frac{D}{D-1}}\, y_l. \tag{3.30}$$

It "supports" another temptation that frequently occurs in practice, namely to analyze the relation between different clr coefficients, e.g., in terms of correlations. This should be avoided as one deals with coefficients with respect to a generating system. Accordingly, the covariance structure of clr coefficients is driven by the zero sum constraint,

$$\mathrm{cov}(y_l, y_1) + \ldots + \mathrm{cov}(y_l, y_{l-1}) + \mathrm{cov}(y_l, y_{l+1}) + \ldots + \mathrm{cov}(y_l, y_D) = -\mathrm{var}(y_l),$$

for $l = 1, \ldots, D$ (Aitchison 1986). This leads to the so-called negative bias (overabundance of negative covariances/correlations), similar as for proportional or any other constrained data. Additionally, the coordinates $z_1^{(l)}$, for $l = 1, \ldots, D$, come from different coordinate systems, indicating clearly that analyzing relations between clr coordinates definitely cannot be recommended. An alternative is proposed in the next subsection.

3.3.5 *Special Coordinate Systems: Symmetric Pivot Coordinates*

If one is interested in the relation between two compositional parts, a possible alternative is to construct such coordinates that would treat the dominance of both parts in a given composition. This can be achieved by constructing two pivot coordinate systems (3.25), but in a symmetric manner. Without loss of generality, we are interested in identifying the relation between the parts x_1 and x_2. Accordingly, two pivot coordinate systems $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$ resulting from the permutation of the parts in (3.19) are taken and the focus is on the role of x_1 and x_2, respectively. It is obvious that the first two coordinates from each system, (3.31) and (3.32), fully describe the subcomposition $(x_1, x_2)'$ within the given composition:

$$z_1^{(1)} = \sqrt{\frac{D-1}{D}} \ln \frac{x_1}{\sqrt[D-1]{\prod_{i=2}^{D} x_i}}, \quad z_2^{(1)} = \sqrt{\frac{D-2}{D-1}} \ln \frac{x_2}{\sqrt[D-2]{\prod_{i=3}^{D} x_i}}, \tag{3.31}$$

$$z_1^{(2)} = \sqrt{\frac{D-1}{D}} \ln \frac{x_2}{\sqrt[D-1]{x_1 \prod_{i=3}^{D} x_i}}, \quad z_2^{(2)} = \sqrt{\frac{D-2}{D-1}} \ln \frac{x_1}{\sqrt[D-2]{\prod_{i=3}^{D} x_i}}. \tag{3.32}$$

On the other hand, neither (3.31) nor (3.32) can be considered as treating x_1 and x_2 in a symmetric manner. Coordinates (3.31) clearly highlight the role of x_1, because in the second coordinate the dominance of x_2 over the aggregated remaining components without x_1 is expressed. A similar interpretation can be derived for the coordinates (3.32) and for part x_2. A natural idea is thus to "average" both couples of coordinates, just by taking care about orthonormality of the resulting coordinates (Kynčlová et al. 2017). These considerations lead to

$$z_1^{(1,2)} = \sqrt{\frac{D-1+\sqrt{D(D-2)}}{2D}} \ln \frac{x_1}{x_2^{\frac{1}{D-1+\sqrt{D(D-2)}}} \left(x_3 x_4 \cdots x_D\right)^{\frac{\sqrt{D-2}+\sqrt{D}}{\sqrt{D-2}(D-1+\sqrt{D(D-2)})}}} \tag{3.33}$$

and

$$z_2^{(1,2)} = \sqrt{\frac{D-1+\sqrt{D(D-2)}}{2D}} \ln \frac{x_2}{x_1^{\frac{1}{D-1+\sqrt{D(D-2)}}} \left(x_3 x_4 \cdots x_D\right)^{\frac{\sqrt{D-2}+\sqrt{D}}{\sqrt{D-2}(D-1+\sqrt{D(D-2)})}}}. \tag{3.34}$$

The construction of the above coordinates guarantees that $z_1^{(1,2)}, z_2^{(1,2)}, z_3^{(1)}, \ldots, z_{D-1}^{(1)}$, or alternatively $z_1^{(1,2)}, z_2^{(1,2)}, z_3^{(2)}, \ldots, z_{D-1}^{(2)}$, form orthonormal coordinates of the composition $\mathbf{x}$ (Kynčlová et al. 2017). In the sequel, any of such choices are referred to as **symmetric pivot coordinates**. Of course, for practical purposes mostly just the first two coordinates are needed. The interpretation of the resulting symmetric pivot coordinates (by considering now just the first two coordinates out of the whole coordinate system) is indeed as expected: they both capture the dominance of x_1 and x_2, respectively, with respect to the other components in a symmetric manner. Although the coefficients in the denominator of (3.33) and (3.34) seem to be quite complicated, one does not need to take care about them in practice, because they are just resulting from the normalization required to achieve orthonormality of the coordinates. More important is the weighting of x_2 in $z_1^{(1,2)}$ (and x_1 in $z_2^{(1,2)}$) that is different from the remaining parts, which reflects the compromise resulting from symmetrizing the input coordinates (3.31) and (3.32). Nevertheless, it is easy to see that the ratio of both weights,

$$\frac{\frac{1}{D-1+\sqrt{D(D-2)}}}{\frac{\sqrt{D-2}+\sqrt{D}}{\sqrt{D-2}(D-1+\sqrt{D(D-2)})}} = \frac{\sqrt{D-2}}{\sqrt{D-2}+\sqrt{D}},$$

is stabilized quite soon with an increasing number of parts to approximately one half in favor of the remaining parts (Kynčlová et al. 2017).

In the next step, symmetric pivot coordinates can be generalized to any couple of parts x_k and x_l, for $k \neq l$; particularly,

$$z_1^{(k,l)} = C \cdot \ln \frac{x_k}{x_l^{\frac{1}{D-1+\sqrt{D(D-2)}}} \left(x_1 \cdots x_{k-1} x_{k+1} \cdots x_D\right)^{\frac{\sqrt{D-2}+\sqrt{D}}{\sqrt{D-2}(D-1+\sqrt{D(D-2)})}}} \tag{3.35}$$

and

$$z_2^{(k,l)} = C \cdot \ln \frac{x_l}{x_k^{\frac{1}{D-1+\sqrt{D(D-2)}}} \left(x_1 \cdots x_{l-1} x_{l+1} \cdots x_D\right)^{\frac{\sqrt{D-2}+\sqrt{D}}{\sqrt{D-2}(D-1+\sqrt{D(D-2)})}}} \tag{3.36}$$

with $C = \sqrt{\frac{D-1+\sqrt{D(D-2)}}{2D}}$. Similarly, also their sample versions could be introduced. It can be seen that now the first two symmetric pivot coordinates can be

taken for a bivariate analysis, highlighting the relative roles of x_k and x_l within a given composition without the danger of negative bias, as it was the case for clr coefficients.

Symmetric pivot coordinates are used in the following chapter for bivariate plotting, and particularly in Chap. 8 in the context of correlation analysis.

3.3.6 Special Coordinate Systems: Balances

In practice it is not only of interest to interpret the relative dominance of single parts, but also the behavior of (non-overlapping) **groups of compositional parts** within the composition. Suppose that the major effects within a composition are caused by two groups of parts. Then one is interested in constructing coordinates that allow for an interpretation of the two groups in terms of relative information. Such coordinates are called **balances**, since they refer to the balance between the groups. The procedure for their construction is called *sequential binary partitioning* (SBP) (Egozcue and Pawlowsky-Glahn 2005). As this name indicates, not only the balance between two groups is investigated with such a procedure, but the relative dominance of groups of parts is considered in a sequential manner. The balances for groups of parts are sequentially constructed as follows (Egozcue and Pawlowsky-Glahn 2005). For the k-th step of the procedure, denote the indices of the compositional parts of one group by $i_1, i_2, \ldots, i_{p_k}$, coded by "+", and those of the second group by $j_1, j_2, \ldots, j_{m_k}$, coded by "−". Thus, the first group consists of p_k parts, and the second group has m_k parts. In the first step of the procedure, $p_k + m_k = D$, but in subsequent steps not all parts are involved because the initial groups will be split up into smaller groups. The parts which are not involved are coded by "0". The corresponding balance is defined as

$$\tilde{z}_k = \sqrt{\frac{p_k m_k}{p_k + m_k}} \ln \frac{\left(x_{i_1} x_{i_2} \cdots x_{i_{p_k}}\right)^{1/p_k}}{\left(x_{j_1} x_{j_2} \cdots x_{j_{m_k}}\right)^{1/m_k}}. \tag{3.37}$$

The total number of steps required is $D - 1$, and the resulting coordinates $\tilde{z}_1, \ldots, \tilde{z}_{D-1}$ correspond to an orthonormal basis in $\tilde{S}^D$. Similar to Eq. (3.23), the basis vector corresponding to the coordinate of Eq. (3.37) is $\tilde{\mathbf{v}}_{.k} = (\tilde{v}_{1k}, \ldots, \tilde{v}_{Dk})'$, the k-th column of $\tilde{\mathbf{V}}$, and it is given by

$$\begin{aligned}
\tilde{v}_{lk} &= \frac{1}{p_k}\sqrt{\frac{p_k m_k}{p_k + m_k}} \quad \text{for } l \in \{i_1, i_2, \ldots, i_{p_k}\} \\
\tilde{v}_{lk} &= -\frac{1}{m_k}\sqrt{\frac{p_k m_k}{p_k + m_k}} \quad \text{for } l \in \{j_1, j_2, \ldots, j_{m_k}\} \\
\tilde{v}_{lk} &= 0 \quad \text{for all remaining indices.}
\end{aligned}$$

With this basis vector one can see that the coordinate $\tilde{z}_k$ can also be expressed as log-contrast, see earlier in this section, since one gets linear combinations $\sum_{l=1}^{D} \tilde{v}_{lk} \ln x_l$ of log-transformed compositional parts such that $\tilde{v}_{1k} + \tilde{v}_{2k} + \ldots + \tilde{v}_{Dk} = 0$. Note that the result of sequential binary partitioning of a compositional data set can be displayed also graphically together with basic descriptive characteristics using the so-called CoDa-dendrogram, see, e.g., Pawlowsky-Glahn and Egozcue (2011) for details.

As an example, consider a five-part composition $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)'$, which should be split into the groups x_1, x_2, x_3 and x_4, x_5, because from the contents it is clear that the parts forming both groups are belonging together in some sense. One can imagine that such a composition corresponds to household expenditures, consisting of components *foodstuff* (x_1), *housing* (x_2), *clothing* (x_3), *recreation* (x_4), and *restaurants* (x_5). The first three parts can be considered as *basic* expenditures, while the remaining two as *complementary* ones. The primary interest is to evaluate the dominance of the first group of parts (represented by their geometric mean) with respect to an "average" behavior of the complementary components. The sequential binary partitioning can be done as indicated in Table 3.1.

The matrix $\tilde{\mathbf{V}}$ corresponding to the partition of Table 3.1 has the structure

$$\tilde{\mathbf{V}} = (\tilde{\mathbf{v}}_{\cdot 1}, \tilde{\mathbf{v}}_{\cdot 2}, \tilde{\mathbf{v}}_{\cdot 3}, \tilde{\mathbf{v}}_{\cdot 4}) = \begin{pmatrix} \frac{1}{3}\sqrt{\frac{6}{5}} & \sqrt{\frac{2}{3}} & 0 & 0 \\ \frac{1}{3}\sqrt{\frac{6}{5}} & -\frac{1}{2}\sqrt{\frac{2}{3}} & \sqrt{\frac{1}{2}} & 0 \\ \frac{1}{3}\sqrt{\frac{6}{5}} & -\frac{1}{2}\sqrt{\frac{2}{3}} & -\sqrt{\frac{1}{2}} & 0 \\ -\frac{1}{2}\sqrt{\frac{6}{5}} & 0 & 0 & \sqrt{\frac{1}{2}} \\ -\frac{1}{2}\sqrt{\frac{6}{5}} & 0 & 0 & -\sqrt{\frac{1}{2}} \end{pmatrix}.$$

It can be observed that the sums of the columns of $\tilde{\mathbf{V}}$ are indeed zero. The coordinates are then obtained in analogy to Eq. (3.24) as

$$\tilde{\mathbf{z}} = (\tilde{z}_1, \tilde{z}_2, \tilde{z}_3, \tilde{z}_4)' = \tilde{\mathbf{V}}'\mathbf{y} = \tilde{\mathbf{V}}'\mathrm{clr}(\mathbf{x}),$$

Table 3.1 Sequential binary partitioning of a five-part composition into the two groups x_1, x_2, x_3 and x_4, x_5

	$k =$			
	1	2	3	4
x_1	+	+	0	0
x_2	+	−	+	0
x_3	+	−	−	0
x_4	−	0	0	+
x_5	−	0	0	−
p_k	3	1	1	1
m_k	2	2	1	1

Table 3.2 Sequential binary partitioning corresponding to the ilr basis of pivot coordinates defined in Eq. (3.19)

	$k =$				
	1	2	3	...	$D-1$
x_1	+	0	0	...	0
x_2	−	+	0	...	0
x_3	−	−	+	...	0
⋮	⋮	⋮	⋮		⋮
x_{D-1}	−	−	−	...	+
x_D	−	−	−	...	−
p_k	1	1	1	...	1
m_k	$D-1$	$D-2$	$D-3$	...	1

explicitly as

$$\tilde{z}_1 = \sqrt{\frac{6}{5}} \ln \frac{\sqrt{x_1 x_2 x_3}}{\sqrt[3]{x_4 x_5}}, \quad \tilde{z}_2 = \sqrt{\frac{2}{3}} \ln \frac{x_1}{\sqrt{x_2 x_3}}, \quad \tilde{z}_3 = \sqrt{\frac{1}{2}} \ln \frac{x_2}{x_3}, \quad \tilde{z}_4 = \sqrt{\frac{1}{2}} \ln \frac{x_4}{x_5}.$$

According to the construction, $\tilde{z}_1$ is the balance between the two groups of parts x_1, x_2, x_3 and x_4, x_5. The coordinates $\tilde{z}_2$ and $\tilde{z}_3$ describe all relative information of the first group, and they could have been constructed also in a different order. Coordinate $\tilde{z}_4$ describes all relative information of the second group, here involving just the logratio between the parts x_4 and x_5.

The pivot coordinates defined in Eq. (3.19) are constructed in a way that z_1 describes all relative information about part x_1. Thus, z_1 can also be seen as a balance between the "groups" x_1 and $x_2, \ldots, x_D$. Accordingly, one could even alternatively refer to pivot *balances*. Moreover, with sequential binary partitioning it is immediate to construct the corresponding orthonormal basis, see Table 3.2. The above procedure for the construction of the basis vectors results exactly in the matrix $\mathbf{V}$ defined in Eq. (3.23).

Remark 1 One should be aware that the value of the pivot coordinate z_1, constructed to extract the relative information about x_1, strongly depends on the other parts of the actual composition. For example, while $\mathbf{x} = (4, 2, 3, 40)'$ results in $z_1 = \sqrt{\frac{3}{4}} \ln \frac{4}{\sqrt[3]{2 \cdot 3 \cdot 40}} = -0.382$, for its three-part subcomposition $\mathbf{x}^* = (4, 2, 3)'$ the corresponding coordinate yields $z_1^* = \sqrt{\frac{2}{3}} \ln \frac{4}{\sqrt{2 \cdot 3}} = 0.400$. Thus, while the first part dominates on average in the original composition, just the opposite conclusion can be made for the subcomposition. Consequently, this result might lead to the temptation to denote the above effect as to violate subcompositional coherence, introduced in Sect. 1.3. However, this would be a principal misunderstanding of the concept. Subcompositional coherence refers to parts of the original composition, and particularly to its underlying geometrical properties that are valid when passing from the parent composition to its subcomposition. On the other hand, here one deals purely with logratios, where for the construction of z_1 all those logratios

containing x_1 were aggregated. It is logical that removing one or more parts leads to a change of the input information for this coordinate. Therefore, it is recommended to consider a knowledge-driven selection of the parts whenever possible that helps to prevent irresponsible (and even misleading) results of the analysis.

Remark 2 It might be confusing to the reader, why the natural logarithm "ln" is permanently used to construct the coordinates, and not, for example, the logarithm to the base 10, i.e. $\log_{10}$. Note that for a general basis "b", there is the relation $\log_b x = \frac{\ln x}{\ln b}$, and thus the relation between "ln" and $\log_{10}$ is

$$\log_{10} x = \frac{\ln x}{\ln 10} \approx 0.4343 \cdot \ln x.$$

Thus, $\log_{10}$ could be used as well for the construction of the coordinates; this just refers to rescaling compared to the natural logarithm. In case of ilr coordinates, they would still refer to an orthonormal basis. $\log_{10}$ is probably easier for the interpretation. For example, if $\log_{10} \frac{x_1}{x_2} = 1$, then x_1 dominates x_2 by a factor of $10^1 = 10$; if $\log_{10} \frac{x_1}{x_2} = 2$, then x_1 dominates $10^2 = 100$ times the part x_2, etc.

3.4 Examples

Some of the theoretical concepts presented in this chapter are illustrated in the following using the R package **robCompositions**.

An important concept of the logratio approach is that the row sums must not be equal for the compositions; even more: the analysis of compositional data does not rely on the constant sum constraint. Consider again the *phd* data from Table 1.2 of Sect. 1.2, and restrict only to the study groups, forming the parts of the composition. These data are stored in the R object "phd_totals". The number of students for the listed countries is very different (these numbers slightly changed because the percentage data were re-expressed as absolute numbers of students):

```
data("phd_totals")
rowSums(phd_totals)

##     BE     BG     CZ     DK     EE     IE     GR     ES
##   7501   5200  22602   4800   1999   5101  22499  77099
##     FR     IT     LV     LT     HU     AT     PL     PT
##  69800  38299   1801   2899   8000  16800  32700  20500
##     RO     SI     SK     FI     SE     UK     CR     TK
##  21699   1100  10700  22100  21401  94200   1300  32600
##     NO     CH     JP     US
##   5000  17200  75000 388699
```

For instance, the ratios between the parts *human* and *health* are:

```
phd_totals$human / phd_totals$health

##  [1]  0.9577329  1.7567568  0.9683457  0.5752066  2.1428571
##  [6]  2.4977778 10.2828283  1.1783976  8.1832393  0.9615576
## [11]  2.3846154  1.3651877  1.5276074  5.1936709  3.3158245
## [16]  1.6081081  0.5522686  1.1250000  0.9738142  2.2298421
## [21]  0.3987567  1.4117322  1.2170213  1.9231778  0.5204918
## [26]  1.0193182  0.4197451  1.3785337
```

The compositions can now be represented with constant sum one, using the function `constSum` of the package **robCompositions**:

```
phd_totals1 <- constSum(phd_totals, const = 1)
all(rowSums(phd_totals1) == 1) # OK

## [1] TRUE
```

Now consider the same ratio as above, for the new object "phd_totals1":

```
phd_totals1$human / phd_totals1$health

##  [1]  0.9577329  1.7567568  0.9683457  0.5752066  2.1428571
##  [6]  2.4977778 10.2828283  1.1783976  8.1832393  0.9615576
## [11]  2.3846154  1.3651877  1.5276074  5.1936709  3.3158245
## [16]  1.6081081  0.5522686  1.1250000  0.9738142  2.2298421
## [21]  0.3987567  1.4117322  1.2170213  1.9231778  0.5204918
## [26]  1.0193182  0.4197451  1.3785337
```

It can be seen that the ratios do not change even if each observation is multiplied by a different constant. In this case, the constant for multiplication led to the constant sum one for all observations. The values are in the interval from zero to infinity, and taking the logarithm symmetrizes the data around zero, theoretically into the interval from minus to plus infinity:

```
log(phd_totals1$human / phd_totals1$health)

##  [1] -0.04318630  0.56346936 -0.03216611 -0.55302598
##  [5]  0.76214005  0.91540145  2.33047535  0.16415555
##  [9]  2.10208807 -0.03920083  0.86903785  0.31129194
## [13]  0.42370270  1.64744075  1.19870630  0.47505840
## [17] -0.59372081  0.11778304 -0.02653472  0.80193076
## [21] -0.91940392  0.34481743  0.19640630  0.65397890
## [25] -0.65298114  0.01913395 -0.86810761  0.32102040
```

Scale invariance is also valid for the Aitchison distance. Computing the Aitchison distances between the first four observations of the object "phd_totals" is done with

the function `aDist`, and results in:

```
aDist(phd_totals[1:4, ])

##           BE        BG        CZ
## BG 0.8289179
## CZ 0.5007665 0.5580570
## DK 0.7620140 1.2034696 0.7714111
```

The distance matrix is represented as lower diagonal matrix, and it is the same as computing the Aitchison distances between the observations represented by constant sum 1:

```
aDist(phd_totals1[1:4, ])

##           BE        BG        CZ
## BG 0.8289179
## CZ 0.5007665 0.5580570
## DK 0.7620140 1.2034696 0.7714111
```

On the other hand, the Euclidean distances are not scale invariant:

```
dist(phd_totals[1:4, ])

##            BE         BG          CZ
## BG  1539.8711
## CZ  8393.2825  9800.3517
## DK  1831.1218   910.4153 10059.3724

dist(phd_totals1[1:4, ])

##            BE         BG         CZ
## BG 0.11877222
## CZ 0.05138784 0.11703089
## DK 0.14976808 0.17546305 0.13215412
```

As a next step, the compositions are expressed in alr coordinates. The corresponding function in the package **robCompositions** is `addLR` (and `addLRinv` for its inverse to get back to the original compositions). Figure 3.3 shows a plot of the first two alr coordinates, if (a) the third part is used as the "ratioing" variable, and if (b) the fourth part is used in the denominator of Eq. (3.11).

The first two alr coordinates shown in Fig. 3.3 reveal that the results depend quite a lot on the choice of the ratioing variable. While in Fig. 3.3b the relation between the variables seems to be positive, this is no longer true for Fig. 3.3a. Thus note that the coordinate system is not symmetrical in the components, and the main problem with the additive logratio coordinates is the non-isometric character of this coordinate system. This is also seen when computing distances. For an isometric mapping, the Euclidean distances between the alr coordinates should be equal to the Aitchison distances between the compositional parts. For the two different alr

```
a1 <- addLR(phd_totals, ivar = 3)
ggplot(a1$x.alr[, 1:2],
  aes(x=technical,
      y = socio.economic.law)) +
  geom_point(size = 2) +
  xlab("log(technical / human)") +
  ylab("log(socio economic law /
       human)")
```

```
a2 <- addLR(phd_totals, ivar = 4)
ggplot(a2$x.alr[, 1:2],
  aes(x=technical,
      y = socio.economic.law)) +
  geom_point(size = 2) +
  xlab("log(technical / health)") +
  ylab("log(socio economic law /
       health)")
```

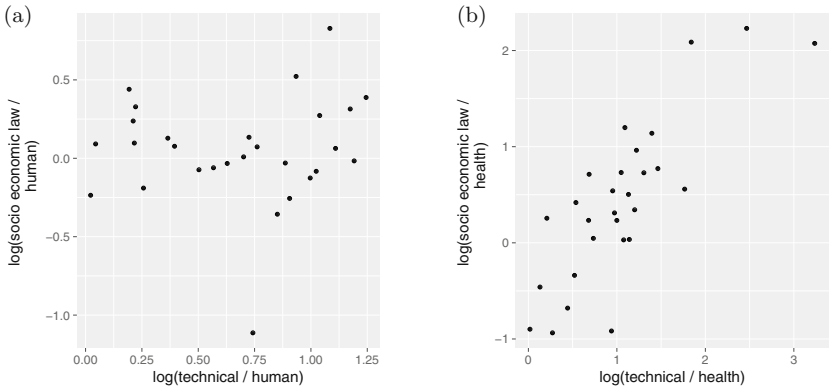


Fig. 3.3 Additive logratio coordinates. (**a**) *human* as ratioing variable. (**b**) *health* as ratioing variable

coordinate systems one obtains:

```
dist(a1$x.alr[1:4, ])

##           BE        BG        CZ
## BG 1.5315528
## CZ 0.6914512 0.9844924
## DK 0.7668354 1.7009175 0.8648189

dist(a2$x.alr[1:4, ])

##           BE        BG        CZ
## BG 0.8317580
## CZ 0.6746951 0.7633451
## DK 1.4434255 1.7675542 1.0925844
```

So, in both cases different results are obtained, and they differ from the Aitchison distances shown earlier. As a conclusion, alr coordinates are still frequently used in many applications, but they should be taken with caution, last but not least due to their lack of isometry (see also Pawlowsky-Glahn et al. 2015).

A better choice is to use the centered logratio coefficients or isometric logratio coordinates, which both are isometric mappings. Both are implemented in the package **robCompositions** as functions named `cenLR` and `pivotCoord`, with their corresponding inverse functions `cenLRinv` and `pivotCoordinv`. In fact, the function `pivotCoord` corresponds to specific isometric logratio coordinates, namely to the pivot coordinates defined in Eq. (3.19).

```
## centered logratio coefficients
c1 <- cenLR(phd_totals)$x.clr
ggplot(c1, aes(x = technical,
  y = socio.economic.law)) +
  geom_point(size = 2) +
  xlab("technical (clr)") +
  ylab("socio economic law (clr)")
```

```
## isometric logratio coordinates
i1 <- pivotCoord(phd_totals)
names(i1) <- paste0("z", 1:ncol(i1))
ggplot(i1, aes(x = z1, y = z2)) +
  geom_point(size = 2) +
  xlab("technical / rest") +
  ylab("socio economic law / rest
(excluding technical)")
```

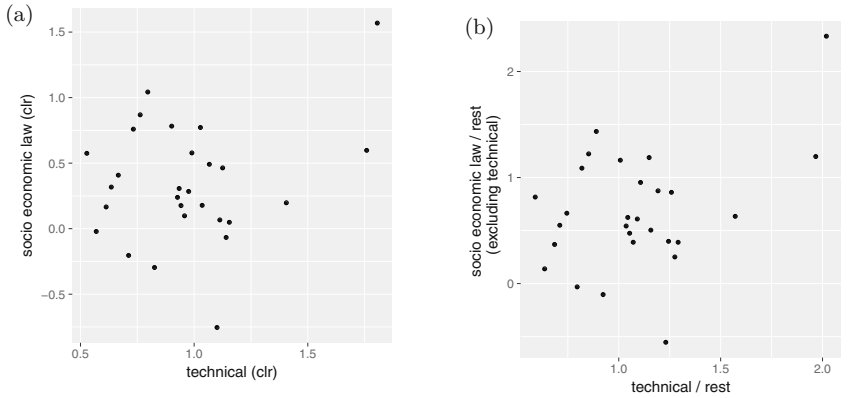


Fig. 3.4 Scatterplot of the first two coefficients (coordinates). (**a**) Centered logratio coefficients. (**b**) Isometric logratio coordinates

It can be seen from Fig. 3.4 that both plots are very similar—up to scaling in the x-coordinate and taking into account that this is no more the case for the y-coordinate, since the part "technical" is no more contained in the second pivot coordinate. Both coordinate representations are isometric, which can be seen by computing Euclidean distances of the clr coefficients and ilr coordinates, respectively, here just for the first four observations:

```
dist(c1[1:4, ])

##           BE        BG        CZ
## BG 0.8289179
## CZ 0.5007665 0.5580570
## DK 0.7620140 1.2034696 0.7714111

dist(i1[1:4, ])

##           1         2         3
## 2 0.8289179
## 3 0.5007665 0.5580570
## 4 0.7620140 1.2034696 0.7714111
```

The results match with the Aitchison distances shown earlier, computed for the original compositions.

The centered logratio coefficients, saved in the object `c1`, and the isometric logratio coordinates, saved as `i1`, are linked by an orthonormal matrix `V`, see Eq. (3.24). This matrix can be obtained using the function `orthbasis`.

```
V <- orthbasis(ncol(phd_totals))
```

The outcome is:

```
V

## $V
##             [,1]       [,2]       [,3]       [,4]
## [1,]  0.8944272  0.0000000  0.0000000  0.0000000
## [2,] -0.2236068  0.8660254  0.0000000  0.0000000
## [3,] -0.2236068 -0.2886751  0.8164966  0.0000000
## [4,] -0.2236068 -0.2886751 -0.4082483  0.7071068
## [5,] -0.2236068 -0.2886751 -0.4082483 -0.7071068
##
## $basisv
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]   -1    1    0    0
## [3,]   -1   -1    1    0
## [4,]   -1   -1   -1    1
## [5,]   -1   -1   -1   -1
```

The list element $V contains the matrix with orthonormal columns, and $basisv represents the sequential binary partitioning for this ilr representation, see Table 3.2. This matrix supports the interpretation of the pivot coordinates: the first coordinate describes all relative information about the first compositional part, the second coordinate includes all relative information of the second part to the parts 3–5, etc.

The centered logratio coefficients can be multiplied with the matrix $\mathbf{V}$ and compared with the outcome from object `i1`:

```
i1b <-  as.matrix(c1) %*% V$V
head(i1b, 2)

##         [,1]      [,2]      [,3]      [,4]
## BE 1.157554 0.5045118 0.2387930 0.4746762
## BG 1.044335 0.6236336 0.9552852 0.8577365

head(i1, 2)

##          z1        z2        z3        z4
## 1 1.157554 0.5045118 0.2387930 0.4746762
## 2 1.044335 0.6236336 0.9552852 0.8577365
```

It can be seen that exactly the same results are obtained, see also Eq. (3.24).

Now the compositional parts are reordered as follows:

```
phd_totals3 <- phd_totals[, c(3,4,5,1,2)]
```

and (generalized) pivot coordinates are constructed, where the focus is on the third part.

```
i3 <- pivotCoord(phd_totals3)
```

Thus, the first coordinate of `i3` contains all relative information of the third part to the other parts in the composition, see Eq. (3.25). In the next step, the permutation matrix is constructed in order to obtain the basis for the permuted composition, see Eq. (3.27).

```
P5 <- cbind(c(0,0,1,0,0), c(0,0,0,1,0), c(0,0,0,0,1),
            c(1,0,0,0,0), c(0,1,0,0,0))
print(P5)

##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    0    1    0
## [2,]    0    0    0    0    1
## [3,]    1    0    0    0    0
## [4,]    0    1    0    0    0
## [5,]    0    0    1    0    0

V3 <- t(P5) %*% V$V
```

Then the matrix $\mathbf{Q}^{(3)}$ is obtained, as in Eq. (3.28).

```
Q3 <- t(V3) %*% V$V
```

This matrix is orthogonal, and it can be used to switch from a particular ilr representation to another one, as shown in Eq. (3.29), but here for the sample version.

```
i3b <- as.matrix(i1) %*% Q3
head(i3b, 2)

##            [,1]       [,2]      [,3]      [,4]
## [1,] -0.2342318 -0.2525247 -1.179287 0.6061776
## [2,]  0.4096915 -0.1217289 -1.657794 0.4437230

head(i3, 2)

##   human_he-ag-te-so health_ag-te-so agriculture_te-so
## 1        -0.2342318       -0.2525247         -1.179287
## 2         0.4096915       -0.1217289         -1.657794
##   technical_so
## 1    0.6061776
## 2    0.4437230
```

It can be seen that the new pivot coordinates of `i3` can indeed be obtained through the original pivot coordinates of `i1`. This works not only between different (generalized) pivot coordinate systems, but for any ilr representation.

In a next step, the use of symmetric pivot coordinates is illustrated. The PhD data set is employed again, and the interest is in the relation between the study subjects "technical" and "health" (as always in this context throughout the book, both parts are mentioned in the sense of their dominance to the other parts in the given composition). The first two symmetric pivot coordinates are computed using Eqs. (3.33)

```
s1 <- pivotCoord(phd_totals[c(1,4,2,3,5)], method = "symm")
s1 <- data.frame(s1)
names(s1) <- c("technical","health")
ggplot(s1, aes(x = technical, y = health, label = coun)) +   geom_text() +
  xlab("technical (symm. balance)") + ylab("health (symm. balance)")
```

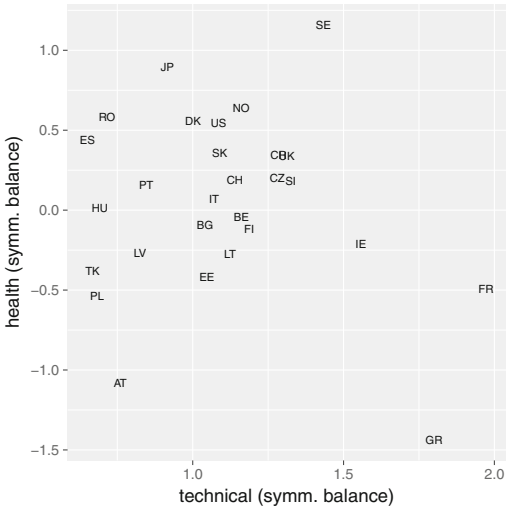


Fig. 3.5 Symmetric balances for the subjects "technical" and "health"

and (3.34). Since "technical" and "health" do not form the first two columns of the data set, the order has to be rearranged, or one could also use the generalized Eqs. (3.35) and (3.36). Note that when using the parameter `method = "symm"` in the function `pivotCoord`, just the first two coordinates are obtained, but it would not be difficult to compute a whole (symmetric pivot) coordinate representation.

The outcome is shown in Fig. 3.5, where no clear relation between the two subjects is visible. However, some atypical observations can be seen: Greece (GR), for instance, is quite dominant in technical studies (with respect to the remaining study areas in the composition), but shows very low dominance for studies related to health.

As a final exercise, a specific balance will be constructed. Suppose that the interest is in separating the relative information of the more technical studies (technical, agriculture) from the remaining studies (soc-eco-law, human, health), see Table 1.2. This refers to a balance of the first and last part versus the three remaining studies. With sequential binary partitioning one needs to define the following matrix, see also Table 3.1.

```
Y <- data.frame(c(1,-1,-1,-1,1), c(1,0,0,0,-1),
                c(0,1,-1,-1,0), c(0,0,1,-1,0))
names(Y) <- paste0("e", 1:4)
print(Y)
```

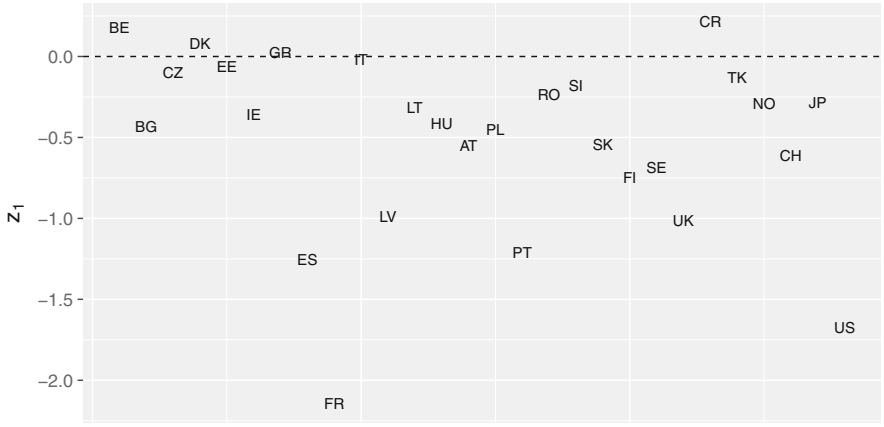


Fig. 3.6 First balance distinguishing relative information of the technical from the non-technical study subjects

```
##   e1 e2 e3 e4
## 1  1  1  0  0
## 2 -1  0  1  0
## 3 -1  0 -1  1
## 4 -1  0 -1 -1
## 5  1 -1  0  0
```

Figure 3.6 presents a plot of the first balance (first column of the object `ib`). Positive values refer to countries with a dominance of the technical study subjects.
The first column defines the balance between the study groups, while the remaining columns describing the relative information within the groups could also be defined differently. The orthonormal basis can be defined with the function `balances`, and the corresponding matrix needs to be multiplied with the clr coefficients to obtain the corresponding ilr coordinates, or get them directly from the list element `$balances`.

```
b <- balances(phd_totals, Y)
ib <- as.matrix(c1) %*% b$V
```

References

J. Aitchison, The statistical analysis of compositional data (with discussion). J. R. Stat. Soc. Ser. B (Stat Methodol.) **44**(2), 139–177 (1982)
J. Aitchison, Principal component analysis of compositional data. Biometrika **70**(1), 57–65 (1983)
J. Aitchison, *The Statistical Analysis of Compositional Data* (Chapman & Hall, London, 1986). Reprinted in 2003 with additional material by The Blackburn Press
M.L. Eaton, *Multivariate Statistics. A Vector Space Approach* (Wiley, New York, 1983)

J.J. Egozcue, V. Pawlowsky-Glahn, Groups of parts and their balances in compositional data analysis. Math. Geol. **37**(7), 795–828 (2005)

J.J. Egozcue, V. Pawlowsky-Glahn, G. Mateu-Figueras, C. Barceló-Vidal, Isometric logratio transformations for compositional data analysis. Math. Geol. **35**(3), 279–300 (2003)

E. Fišerová, K. Hron, On interpretation of orthonormal coordinates for compositional data. Math. Geosci. **43**(4), 455–468 (2011)

P. Kynčlová, P. Filzmoser, K. Hron, Correlation between compositional parts based on symmetric balances. Math. Geosci. **49**(6), 777–796 (2017)

V. Pawlowsky-Glahn, J.J. Egozcue, Geometric approach to statistical analysis on the simplex. Stoch. Env. Res. Risk A. **15**(5), 384–398 (2001)

V. Pawlowsky-Glahn, J.J. Egozcue, Exploring compositional data with the CoDa-dendrogram. Aust. J. Stat. **40**(1–2), 103–113 (2011)

V. Pawlowsky-Glahn, J.J. Egozcue, R. Tolosana-Delgado, *Modeling and Analysis of Compositional Data* (Wiley, Chichester, 2015)

J.L. Scealy, A.H. Welsh, Regression for compositional data by using distributions defined on the hypersphere. J. R. Stat. Soc. Ser. B (Stat Methodol.) **73**(3), 351–375 (2011)

J.L. Scealy, A.H. Welsh, Robust principal component analysis for power transformed compositional data. J. Am. Stat. Assoc. **110**(509), 136–148 (2015)

C. Stewart, C. Field, Managing the essential zeros in quantitative fatty acid signature analysis. J. Agric. Biol. Environ. Stat. **16**(1), 45–69 (2011)

Chapter 4
Exploratory Data Analysis and Visualization

Abstract Standard descriptive characteristics like arithmetic mean, variance, or covariance are not appropriate for compositional data since they cannot cope with their scale invariance principle. Instead, geometric mean (center) and variation matrix, containing the variances of all pairwise logratios, are considered. The scale invariance of compositions has also serious implications for graphical visualization. Univariate plotting of single parts is no longer possible, and reasonable alternatives are plots of knowledge driven logratios, pairwise logratios, first pivot coordinates, or data driven logratio coordinates. For bivariate plotting, when considering a pair of parts of the original composition, either a univariate pairwise logratio plot or symmetric pivot coordinates representing the dominance of the pair with respect to the remaining parts are recommendable. Three-part compositions can be displayed with the traditional ternary diagram. One just needs to be aware that the data to be visualized are driven by the Aitchison geometry, and thus there is the danger of misleading conclusions from this graphical tool. The overall preferred option is to display compositional data in any interpretable orthonormal (ilr) coordinates, free of possible caveats resulting from an inappropriate treatment of observations carrying relative information.

4.1 Descriptive Statistics of Compositional Data

In standard descriptive statistics, one is interested in summarizing information about the given data set. Characteristics like arithmetic mean and variance/standard deviation for one variable, or covariance matrix in the multivariate case, are prominent tools to capture information on location and covariance of the data set. Nevertheless, their use in the case of compositional data is limited to logratio coordinates, where the Euclidean geometry applies. Attempts to apply them for the original compositions that obey the Aitchison geometry is rather problematic. Therefore, if any such descriptive characteristics should be interpretable directly in terms of compositional parts, alternative approaches are necessary.

To illustrate one possible caveat, caused by an inappropriate use of the standard characteristics, consider a simple example of two two-part compositions, $\mathbf{x}_1=(1, 3)'$

P. Filzmoser et al., *Applied Compositional Data Analysis*, Springer Series in Statistics, https://doi.org/10.1007/978-3-319-96422-5_4

and $\mathbf{x}_2 = (5, 3)'$. Applying the arithmetic mean, one would get $\overline{\mathbf{x}} = (\mathbf{x}_1 + \mathbf{x}_2)/2 = (3, 3)'$, where the ratio of the parts equals to one. Due to scale invariance of compositions, also for proportional representations of the resulting arithmetic mean, $\mathcal{C}_1(\overline{\mathbf{x}})$, this key information remains unaltered. Naturally, it is expected that by any other representation of the input data the same result (up to a scaling constant) will be obtained. Nevertheless, by considering $\mathcal{C}_1(\mathbf{x}_1) = (0.25, 0.75)'$ and $\mathcal{C}_1(\mathbf{x}_2) = (0.625, 0.375)'$ the result $\overline{\mathbf{x}}^* = (0.438, 0.562)$ is obtained. The arithmetic mean fails in preserving the ratio between the parts, because $0.438/0.562 = 0.779$.

According to Pawlowsky-Glahn and Egozcue (2002), the proper alternative to the arithmetic mean as a characteristic of location with respect to the Aitchison geometry is represented by the component-wise geometric mean. In this context it is also called *center*, because it characterizes the center of the distribution of the sample at hand. Formally, it is defined for an $n \times D$ compositional data matrix $\mathbf{X} = (x_{ij})$ as a composition

$$\mathbf{g}_{\mathbf{x}} = (g_1, \ldots, g_D)', \tag{4.1}$$

where $g_j = (\prod_{i=1}^{n} x_{ij})^{1/n}$. The center follows the principles of compositional data analysis including the mentioned scale invariance. This feature can be demonstrated directly using the above example, where the center results for both representations of the input compositions in

$$\mathcal{C}_1(\mathbf{g}_{\mathbf{x}}) = \mathcal{C}_1(\sqrt{1 \cdot 5}, \sqrt{3 \cdot 3}) = \mathcal{C}_1(\sqrt{0.25 \cdot 0.625}, \sqrt{0.75 \cdot 0.375}) = (0.427, 0.573),$$

expressed in proportions for the sake of comparison. The ratio $0.427/0.573 = 0.745$ differs from both previous ratios, produced by the arithmetic mean.

Centering of the data can be performed directly with the original compositional data. For any composition $\mathbf{x}_i = (x_{i1}, \ldots, x_{iD})'$, forming the i-th row of the matrix $\mathbf{X}$, its centered counterpart is given as $\mathbf{x}_i^c = \mathbf{x}_i \ominus \mathbf{g}_{\mathbf{x}}$. Due to the properties of the coordinate systems in Sects. 3.3.1–3.3.4, this indeed corresponds to standard mean-centering in logratio coordinates. For example, the center of the centered compositions is the neutral element $\mathbf{n}$. As a consequence, centering of the original compositions can also suppress the effect of relative scale. This effect is stronger close to the border of the simplex, and less relevant in the center, i.e. around the neutral element. This fact is frequently used for visualization purposes, like for three-part compositions in the ternary diagram (see Sect. 4.4), to reveal the compositional data structure masked near the border of the simplex (von Eynatten et al. 2002). Nevertheless, one should be aware that the real data structure can hardly be recognized without expressing the compositions in orthonormal coordinates.

As a measure of variability of compositional data there is no such counterpart as in case of the center that could be constructed directly for the original compositions. Instead, the attention is traditionally directed to source information in a composition, to pairwise logratios. This results in the co-called *variation matrix* (Aitchison 1986), which is formed by the variances of all pairwise logratios. Concretely, for the

compositional data matrix $\mathbf{X} = (x_{ij})$, the variation matrix is defined as

$$\mathbf{T} = \begin{pmatrix} t_{11} & t_{12} & \dots & t_{1D} \\ t_{21} & t_{22} & \dots & t_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ t_{D1} & t_{D2} & \dots & t_{DD} \end{pmatrix}, \tag{4.2}$$

where t_{jk}, $j, k = 1, \dots, D$, are sample variances of pairwise logratios between x_j and x_k, i.e.

$$t_{jk} = \frac{1}{n-1} \sum_{i=1}^{n} (z_{jk}^i - \overline{z}_{jk})^2$$

with

$$\{z_{jk}^i = \ln \frac{x_{ij}}{x_{ik}},\ i = 1, \dots, n\}$$

and

$$\overline{z}_{jk} = \frac{1}{n} \sum_{i=1}^{n} z_{jk}^i.$$

Alternatively, by adding a normalizing constant $\frac{1}{\sqrt{2}}$ to the above logratios one gets, up to sign, the only coordinate for the respective two-part composition. In this case, it is denoted as *normalized variation matrix*

$$\mathbf{T}^* = (t_{jk}^*) = \frac{1}{2}\mathbf{T}$$

(Pawlowsky-Glahn et al. 2015b). Both matrices $\mathbf{T}$ and $\mathbf{T}^*$ are by construction symmetric with diagonal elements of zero. The elements of the variation matrix can be interpreted in terms of variability of the ratio between the corresponding parts. Namely, for values t_{jk} (t_{jk}^*) close to zero the ratios between x_j and x_k in a given sample are almost constant, so that nearly a perfect proportionality of these parts is achieved. Consequently, instead of the rather problematic correlation coefficient computed for the original compositions, the variability of the respective pairwise logratio was adopted as a measure of strength of association between two compositional parts. In order to enhance interpretability, the elements of the variation matrix can be normed to the range (0, 1] as

$$\tau_{jk} = \exp(-\text{var}(t_{jk}^*)) \quad \text{for } 1 \le j, k \le D,\ j \ne k$$

(Buccianti and Pawlowsky-Glahn 2005; Filzmoser et al. 2010). High variability of the logratio then tends to a result approaching zero and, conversely, small variability is reflected by values of τ_{jk} close to one with the limiting case of perfect proportionality. Other possibilities for a better interpretation of the variation matrix elements are discussed in Egozcue et al. (2013) and Pawlowsky-Glahn et al. (2015b).

Nevertheless, the concept of variation matrix as a tool to reveal associations between compositional parts is not completely free of controversy. It results from the fact that interpretability in the sense of *positive* and *negative* association, known from the correlation coefficient, is lost. Moreover, the coefficients τ_{jk} exhibit a non-linear behavior because of the use of the exponential function. As a consequence, it is not very clear, which values are high (low) in comparison to the correlation coefficient. Unfortunately, statistical inference like hypotheses testing, that could help for this purpose, is problematic and can be performed only indirectly (Egozcue et al. 2013). One possible way out is to use symmetric pivot balances (3.35) and (3.36) which treat the dominances of x_j and x_k with respect to the other parts in a given composition in a symmetric manner, and to perform bivariate analysis (including the computation of the correlation coefficient) there. This issue will be further developed in Sect. 4.3, where bivariate plots are discussed, and particularly in Chap. 8 on correlation analysis in logratio coordinates.

Finally, for theoretical purposes it is important to make sure that the total variability of a compositional data set does not depend on a particular coordinate representation. It is not a big surprise that the measure of total variability, the *total variance* (Pawlowsky-Glahn and Egozcue 2001) is defined as a (scaled) sum of all elements of the variation matrix,

$$\text{totvar}(\mathbf{X}) = \frac{1}{2D}\sum_{j=1}^{D}\sum_{k=1}^{D} t_{jk} = \frac{1}{D}\sum_{j=1}^{D}\sum_{k=1}^{D} t^*_{jk}.$$

Indeed, it can be shown using Pawlowsky-Glahn and Egozcue (2001) and the relation between clr coefficients and ilr coordinates (3.24) that totvar($\mathbf{X}$) equals also to the sum of the diagonal elements of the covariance matrix of any orthonormal coordinate representation or clr coefficients. Moreover, the total variance itself induces interesting properties like asymptotic normality (Hron and Kubáček 2011) that can be used for further theoretical developments.

Finally, powering a (centered) composition with totvar($\mathbf{X}$)$^{-1/2}$ can serve as a counterpart to *standardization* (scaling) of real variables by a standard deviation (Pawlowsky-Glahn et al. 2015b). Accordingly, a data set with unit total variance is obtained. Because all parts share the same units (unlike the usual case with real multivariate data), their relative contribution to the total variation is a rich information that an individual standardization of each part would remove, apart from the fact that the ratios between the parts would be altered. However, this kind of standardization is rather of theoretical importance and is rarely used in practical data processing.

4.2 Univariate Graphics

After all the reasoning from previous chapters that refers to the purely multivariate character of compositional data, it might seem inconsistent to talk now about univariate visualization. Nevertheless, as univariate plotting of single compositional parts is of high interest in practice, one cannot simply omit this topic without providing a reasonable alternative. In natural sciences, like in geochemistry or chemometrics, it is common that compositional data are log-transformed prior to further statistical processing with the idea that the *closure problem* (basically negative bias of the covariances resulting as an undesired effect of the constant sum of parts) was removed. Although the log-transformation is definitely easy to handle and interpret, it tells just a part of the story. According to Mateu-Figueras and Pawlowsky-Glahn (2008) and Tolosana-Delgado and Pawlowsky-Glahn (2007) it removes just the effect of relative scale—this is necessary to proceed with processing of absolute-valued information using standard multivariate statistics. On the other hand, the log-transformation violates the scale invariance principle, because this transformation is inherently linked to a given sum of components (Pawlowsky-Glahn et al. 2015a). There are definitely situations in practice, where such treatment seems to be fully satisfactory, because the nature of the data is apparently not purely compositional. One example are geochemical data, where also the absolute abundance of elements may yield important information (Reimann et al. 2012). Nevertheless, even in such cases, taking just relative contributions of elemental concentrations into account can reveal further interesting features (Filzmoser et al. 2009; Reimann et al. 2012; McKinley et al. 2016). From a mathematical perspective, any univariate analysis of compositional data should be performed in coordinates, preferably in those with respect to a basis rather than to a generating system that brings ambiguity to the coordinate representation of compositions. Consequently, clr coordinates are not recommended for the purpose of univariate analysis (see also discussion in Sect. 3.3.2).

For this reason, even the univariate information in compositional data should be conveyed exclusively in terms of logratios. In McKinley et al. (2016) several options were discussed that honor this requirement: knowledge driven logratios, pairwise logratios, pivot coordinates $z_1^{(l)}$, $l = 1, \ldots, D$ from (3.25) capturing relative information on a compositional part within one orthonormal coordinate system, and data driven logratio coordinates. Although these options were developed for the particular case of geochemical mapping, they have also universal validity.

Knowledge driven logratios benefit from the expertise in a given field suggesting that it may be useful to analyze a simple logratio, or a certain balance related to a concrete question of interest. For example, the ratio between expenditures on food and services indicates the living standard of a household. Balance $(\sqrt{2}/\sqrt{3}) \ln(MN/\sqrt{MM \cdot NN})$, formed by the MN, MM, NN genotypes in the MN system of blood groups, helps to reveal a genetic (Hardy-Weinberg) equilibrium (Graffelman and Egozcue 2011). For soils and sediments that have been weathered or mixed/diluted with other material, no element shows the same percentages as

in the background or source rock, but the ratios of elements unaffected by the mixing (or by pollution or weathering) are preserved (McKinley et al. 2016). Just *pairwise logratios* between components are easy to handle and interpret, as long as information in terms of the original compositional parts is not of primary interest. In this case, single pairwise logratios provide too much elemental information for the purpose. If one is interested in all pairwise logratios to one specific part, it would be necessary to consider $D-1$ such logratios in case of a D-part composition, and draw conclusions out of this information. Note that there are, up to sign, $D(D-1)/2$ such logratios in the composition: it means six of them for a four-part composition, but already 45 for a composition with 10 parts and 4950 for data, where 100 components are considered! For many chemometric applications this is still a lower bound of the number of parts to be analyzed.

The simplest solution thus seems to aggregate all pairwise logratios with the part of interest, leading (after proper scaling) to pivot coordinates $z_1^{(l)}$, $l = 1, \ldots, D$ from (3.25). Nevertheless, in line with Remark 1 at the end of Sect. 3.3.4, one should be aware that some of these logratios can be dominant and affect substantially the resulting coordinate values. This happens particularly when parts with small values are involved. Unfortunately, just these parts are frequently burdened by measurement errors, thus one should carefully consider whether to involve them into the input composition, or not. This can be decided on a knowledge-based level (as proposed in Sect. 3.3.4), but also in a data-driven manner. In the latter case, elements of the variation matrix, introduced in Sect. 4.1, can serve as a rule of thumb, which *remaining* components should be considered. As an example, suppose that the part x_1 is of primary interest, represented through the respective coordinate $z_1^{(l)}$. The idea is that parts with weak relations to x_1 bring rather no valuable input, when the dominance of the first component to the rest of the parts is under consideration (Hron et al. 2017). Practically, after proper permutation of the parts in (3.25), this would mean that not the first coordinate, but any of the others, showing dominance of x_1 to the reduced rest of the components, would be taken instead. The pre-selection of the *remaining* parts also prevents from the case of including components with data quality problems, for which weaker relations to x_1 are typical. This situation often happens in geochemistry or in chemometrics, where measurement devices tend to produce erroneous values around detection limits of variables. Based on a discussion concerning the variation matrix in Pawlowsky-Glahn et al. (2015b), one possible rule is to exclude elements, for which $t_{1j} > \frac{2 \cdot \text{totvar}(\mathbf{X})}{D-1}$, $j = 2, \ldots, D$. This means that the logratio between x_1 and x_j contributes to the total variance with a share larger than the average logratio—in other words that the strength of the relation between x_1 and x_j is weaker than the average one. On the other hand, with increasing number of parts, the geometric mean in coordinates $z_1^{(l)}$ gets quickly robust enough against possible data quality problems or systematic patterns arising from one or more compositional parts (Mert et al. 2016). Thus it is recommended to always check first, whether the intended reduction of parts to be involved for the univariate analysis of x_l using $z_1^{(l)}$, $l = 1, \ldots, D$, respectively, would indeed lead to significantly different results.

Data driven logratio coordinates profit from the possibility of defining natural non-overlapping groups of parts in a data driven manner, used further to define a sequential binary partition and interpretable balances. One option is to apply Q-mode clustering of compositional parts, discussed in detail in Sect. 6.6. The resulting coordinates can be analyzed either univariately, or in a multivariate context.

Finally, it is important to note that some of the compositions seem to be univariate already from their definition. A typical example is the unemployment rate in given regions of a country—seemingly just one component is available. On the other hand, for compositional data analysis there must be always at least two parts in a composition. For this reason it is necessary to form the "rest" part, represented by the complement to the whole. If the unemployment rate is denoted by x, the rest component is clearly $1 - x$. In general, instead of 1, any positive constant κ can be considered, like $\kappa = 100$ (percentages) or $\kappa = 10^6$ (mg/kg). Even in such a case of "univariate" compositions, it is necessary to express them in the respective coordinate, here

$$z = \frac{1}{\sqrt{2}} \ln \frac{x}{\kappa - x} \tag{4.3}$$

(Filzmoser et al. 2009), prior to further visualization. Note that (4.3) corresponds, up to a scaling constant, to the well-known logit transformation. The resulting exploratory tools (boxplot, histogram, etc.) can even be expressed in terms of the original scale (for a given κ) as

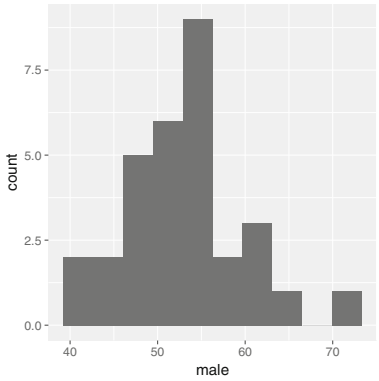
$$x = \frac{\kappa \cdot \exp(\sqrt{2}z)}{1 + \exp(\sqrt{2}z)}. \tag{4.4}$$

On the other hand, it would be very dangerous to get inspired by the simple interpretability of (4.3) and to perform such "univariate analysis" for each part of a D-part composition separately, by merging (amalgamating) all remaining parts together or forming simply an artificial complementary part. This approach can be used exclusively when information about the other parts in a composition is not available, like in the mentioned example with the unemployment rate. Namely, by performing any of the above operations, the variability arising from the aggregation of pairwise logratios in (3.21), that provides additional information about relations of the part of interest to the remaining components, would be irreversibly lost. An additional argument is that summing up compositional parts (amalgamation (Aitchison 1986)) is not consistent with the Aitchison geometry (Egozcue and Pawlowsky-Glahn 2005).

Example Consider the PhD student data from Table 1.2 of Sect. 1.2. We are interested in the percentages of male PhD students, and study the data distribution of this variable by a histogram and a QQ-plot. Figure 4.1 presents the raw percentage data. With the exception of one upper outlier, the distribution seems to be rather symmetric, and probably even close to normality.

```
ggplot(phd, aes(x=male)) +
  geom_histogram(bins = 10)
```

```
library("car")
qqPlot(phd$male)
```



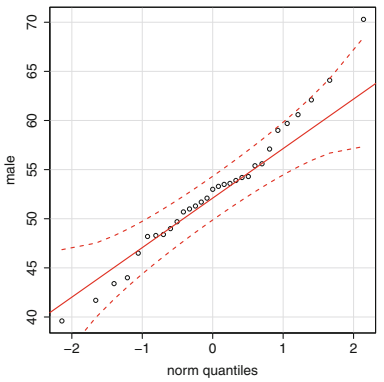
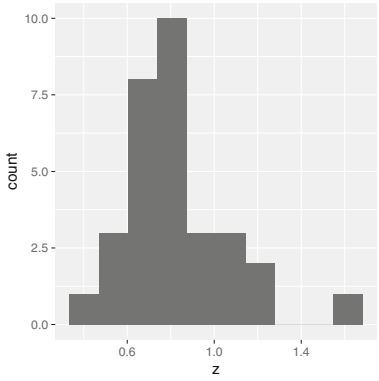


Fig. 4.1 Percentages of male PhD students: original data

```
ggplot(phd, aes(x = z)) +
  geom_histogram(bins = 10)
```

```
qqPlot(phd$z)
```



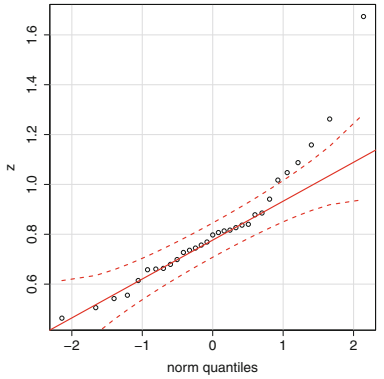


Fig. 4.2 Percentages of male PhD students: coordinate representation

Figure 4.2 shows the data expressed by a coordinate according to Equation (4.3). This coordinate is added to the `phd` data set in the following.

```
phd$z <- 1 / sqrt(2) * phd$male / (100 - phd$male)
```

Comparing Fig. 4.2 with Fig. 4.1, a different impression is obtained. Although the changes are rather small, it can be observed that the distribution is more skewed and also some additional observations deviating from the normality assumption were revealed by the QQ-plot. Note that, in general, both the histogram and the QQ-plot rely on the Euclidean geometry, and thus only the coordinate representation gives

the appropriate picture. Also the values of this coordinate can be interpreted: a value of 0 means equal proportion of females and males, values bigger than zero refer to dominance of the males, and values smaller than zero refer to dominance of females among the PhD students.

4.3 Bivariate Plotting

In general, for bivariate analysis any two ilr coordinates can be used. Here, bivariate plotting refers to visualizations that can be interpreted in the sense of a couple of the original compositional parts, either themselves or in terms of their dominance with respect to the rest of the components.

Getting inspired by the variation matrix (4.2), the simplest bivariate plotting is to consider univariate plots of pairwise logratios. Accordingly, the level of proportionality between two compositional parts can be checked, together with possible occurrences of deviating values leading to higher variability, reflected by the respective element of the variation matrix.

As already indicated in Sect. 4.1, it is desirable in many applications to get an interpretation in terms of positive and negative association, which is not possible with the variation matrix approach (Filzmoser et al. 2010). Therefore, as an alternative symmetric pivot coordinates (3.35) and (3.36) can be employed. Although one should take into account that not the parts themselves, but just their dominances with respect to the averaged rest of components are displayed, the visualization of symmetric pivot coordinates provides a clear value added by considering it as a counterpart to the pairwise logratio plot. Note that bivariate plots with symmetric pivot coordinates for each couple of parts in a composition can also be displayed in form of a matrix plot; each of those plots corresponds to an own coordinate system.

If the original data are not clearly driven by a constant sum representation, like in cases of proportions or percentages, so that also absolute information might be relevant, a bivariate plot of log-transformed parts can be considered as a complementary tool. It accounts for the relative scale of compositions, but the scale invariance principle is violated. Therefore, it is definitely not recommended to use log-transformed data as a sole tool for bivariate plotting of compositional parts.

Example Along a transect of 120 km length crossing the city Oslo, nine different plant materials have been collected and analyzed for the geochemical concentration of various chemical elements. For each plant material, 40 samples are available, and the composition consists of 25 parts. The data have been used, for example, in Templ et al. (2008) for cluster analysis, and they are available in the R package **rrcov** as data set "OsloTransect". First, various elements are extracted and rows with missing values are excluded (see Chap. 13 for the procedure to handle missing values and values below detection limit).

```
data("OsloTransect")
X <- OsloTransect[, c(18,28,14:17,19:27,29:38)] # 18 is Ca, 28 is Mg
isna <- missPatterns(X)$rindex # rows with missing values
```

Next, the first two symmetric pivot coordinates are calculated using the function `pivotCoord` with method `"symm"`. The new coordinates are stored in the Oslo data set.

```
symm <- pivotCoord(X[!isna, ], method = "symm") # symm. coord. for Ca, Mg
colnames(symm) <- c("Ca_symm", "Mg_symm")
OsloTransect <- cbind(OsloTransect[!isna, ], symm)
```

In Fig. 4.3, the focus is on a visualization of the bivariate information of the elements Calcium (Ca) and Magnesium (Mg). The left plot shows the concentrations of these elements in log-scale, where the different colors and symbols refer to the nine plant materials, see also the legend. Some of the plant materials are clearly different concerning the concentrations, and the plants show clear clusters. The plot on the right shows the two symmetric pivot coordinates constructed for the elements Ca and Mg. A positive correlation is visible, but somewhat lower than a correlation of the log-transformed elements, where it is mainly induced by the big concentration differences of the groups. Note that for constructing the symmetric

```
g1 <- ggplot(OsloTransect, aes(x = Ca, y = Mg, colour = X.MAT,
        shape = X.MAT)) + geom_point() + coord_trans(x = "log", y = "log")

g2 <- ggplot(OsloTransect, aes(x = Ca_symm, y = Mg_symm, colour = X.MAT,
        shape = X.MAT)) + geom_point()

grid.arrange(g1, g2, ncol = 2)
```

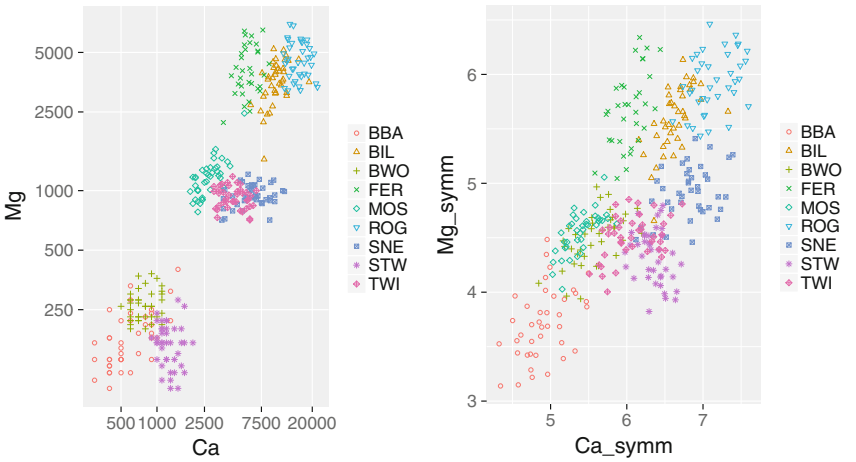


Fig. 4.3 Log-transformation (left) and symmetric pivot coordinates (right) of the elements Ca and Mg

pivot coordinates, information from all pairwise logratios of Ca and Mg to all other parts of the composition are considered. Smaller differences in logratios with Ca and Mg for different plant materials may be the reason for less pronounced grouping structure than it was the case for the absolute concentrations. On the other hand, there are interesting details visible: the observations from some plant materials are subdivided into smaller groups (e.g., FER and MOS). Such subgroups are not visible when inspecting the log-transformed concentrations. Moreover, groups like BBA show even better separation now.

It can be concluded that both plots in Fig. 4.3 contain valuable information. It might be interesting for the geochemist that some plant materials clearly differ in the concentration level of Ca and Mg, and even the actual concentrations (absolute values) may be of interest. One should, however, not draw conclusions from this plot concerning the correlation between the two elements. On the other hand, the plot of the symmetric pivot coordinates takes into account also the logratios to the remaining elements, and thus contains much more information. This leads to additional insight like further subgroups. The correlation between dominance of Ca and Mg within the given composition could be derived from this plot, but one could ask if a correlation at the basis of different plant materials is informative at all.

4.4 Multivariate Visualization

A widely known tool for plotting three-part compositions is the ternary diagram, introduced already in Sect. 3.1 (Fig. 3.1, right). It results from representing the input compositional data with a constant sum constraint. In other words, the plot shows a graphical visualization of the three-part simplex

$$S^3 = \left\{ \mathbf{x} = (x_1, x_2, x_3)' \in \mathbb{R}^3 \mid x_1 > 0, x_2 > 0, x_3 > 0,\ x_1 + x_2 + x_3 = \kappa \right\}$$

for a given constant κ, usually taken as 1 or 100 for the cases of proportions or percentages, respectively. The ternary diagram thus corresponds to a two-dimensional projection of three-part compositions. This can be used to illustrate the reduced dimensionality of compositional data, resulting from their scale invariance. From its construction, the ternary diagram is an equilateral triangle $X_1X_2X_3$ such that a composition $\mathbf{x} = (x_1, x_2, x_3)'$ is plotted at a distance x_1 from the opposite side of vertex X_1, at a distance x_2 from the opposite side of vertex X_2, and at a distance x_3 from the opposite side of vertex X_3. The sum of the distances remains constant for any choice of the parts of $\mathbf{x}$. The borders of the ternary diagram correspond to a value of zero of the part on the opposite vertex, the vertices themselves represent compositions with one part equal to κ (the whole is contained just in that part) and two zero parts. Nevertheless, both these cases fall out of S^3.

Example Within the GEMAS project, also the proportions of sand, silt, and clay have been measured in the samples. Figure 4.4 shows a ternary diagram of this

```
data("gemas")
isna <- missPatterns(gemas[, 9:11])$rindex # look for NAs
sc <- gemas$soilclass[!isna]   # soil class
ternaryDiag(gemas[!isna, 9:11], col = sc, pch = as.numeric(sc))
legend("topleft", c("l","ll","m","s","ss"), col = 1:5, pch = 1:5)
```

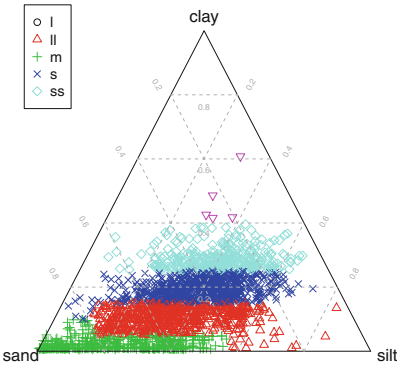


Fig. 4.4 Ternary diagram of the composition sand-silt-clay of the samples from the GEMAS project, distinguished according to different soil classes

composition. The grouping information corresponds to the soil class, as it has been defined in this project. The samples of soil class "m" have a very low proportion on clay, and some of them also consist of a low silt proportion and thus are composed almost exclusively of sand.

The ternary diagram is frequently used as a didactic instrument to explain the peculiarities of the Aitchison geometry. As the whole sample space needs to be contained in the triangle, its borders stand for infinity. This is a natural consequence of the relative scale property of compositions. The closer some observations are placed near the border, the closer are the values of one or two parts to zero, and the ratios between the corresponding parts explode by approaching infinity (by considering logratios instead, also minus infinity could be reached). On the contrary, for observations near to equilibrium of the triangle, represented by the neutral element **n** with all parts being the same, the scale of the compositions approaches the absolute one. It is also interesting to see how geometric figures like lines, circles, or ellipses look like in the projected sample space of compositional data. Such figures result simply from back-transforming the standard figures from any ilr coordinate representation to the original space. Consequently, the previous considerations are supported by the fact that circles and ellipses are minimally distorted near to the barycenter (neutral element), while close to the borders their expected shape is completely deformed. An interesting result is obtained, when two parallel lines are displayed. They intersect in infinity (vertices of the triangle), a very natural output from the perspective of projective geometry.

```
data("coffee")
x <- coffee[, 2:4]                # select 3 parts
x.ilr <- pivotCoord(x)            # construct pivot coordinates
SEQ <- seq(-10, 10, length = 1000) # sequence for prediction
library("oreg")
res.oreg <- oregMM(x.ilr)         # orthogonal regression
co <- res.oreg$coefficients
o <- co[1] + co[2] * SEQ          # expected values
oo <- cbind(SEQ, o)
par(mfrow = c(1,2))
ternaryDiag(x, grid = FALSE,
            name = c("acetic acid","methylpyrazine","furfural"))
ternaryDiagEllipse(x, tolerance = 0.975, locscatt = "MCD")
ternaryDiagPoints(pivotCoordInv(oo), type="l", col="blue")

cv <- robustbase::covMcd(x.ilr)  # robust estimation of center/covariance
chemometrics::drawMahal(x.ilr, cv$center, cv$cov, quantile = 0.975,
    xlab = expression(z[1]), ylab = expression(z[2]))
lines(oo, col = "blue")
```

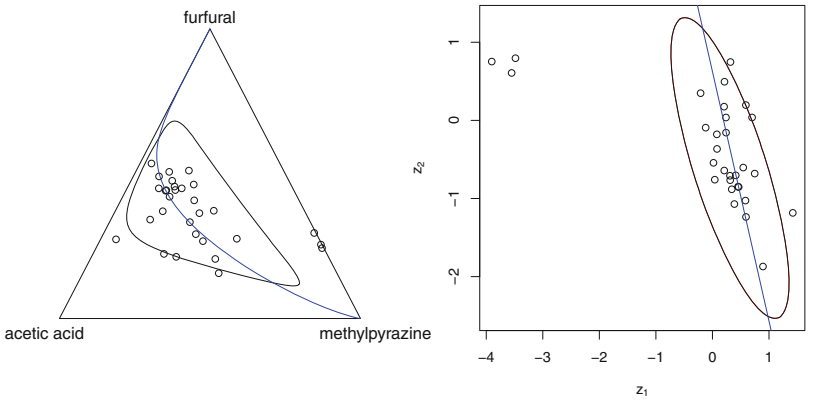


Fig. 4.5 Ternary diagram for three compounds of the coffee data set (left), and representation in ilr coordinates (right). The ellipse and the straight line are constructed in coordinates, and they are also shown in the ternary diagram

Example The package **robCompositions** contains data from 30 commercially available coffee samples of different origins. Here we look at the values of 3 chemical compounds. The ternary diagram in Fig. 4.5 left reveals three atypical coffees, with very low concentrations of acetic acid (sort "robusta"). This subcomposition is shown in ilr (pivot) coordinates on the right plot. Coordinate z_1 presents all relative information of acetic acid to the rest, and z_2 is proportional to the logratio of the remaining parts. In this plot, an ellipse is constructed based on the MCD estimator, yielding robust estimates of location and covariance, see Chap. 5. The points outside the ellipse would in fact represent outliers. The ellipse is also shown in the ternary diagram, where its shape looks very different. Moreover, a straight line is defined in

coordinates; this line is a robust orthogonal regression line, see Chap. 10. The same line is projected into the ternary diagram, where it gets the form of a curve, and reaches infinity at the vertices of the triangle.

The ternary diagram can provide a rough image about the data distribution, but it might also be misleading just due to the relative scale that pronounces near the border of the simplex. This effect can be suppressed by centering the original compositions (von Eynatten et al. 2002), but still it needs to be taken into account. The main problem is that the human brain is used to think in terms of the Euclidean distance, and not in the Aitchison distance that needs to be considered when looking at data in the ternary diagram. The only way out is to express the data in interpretable orthonormal coordinates, where the common thinking in terms of the absolute scale, provided by the Euclidean geometry, can be applied. Therefore, throughout this book, the ternary diagram will be used for didactic or illustrative purposes, but not to draw direct conclusions from an analysis. For readers who like to make use of this graphical tool, more details and possible extensions of ternary diagrams to matrix plots are provided in van den Boogaart and Tolosana-Delgado (2013).

Note that it is possible to visualize also four-part compositional data by a solid, regular tetrahedron, being a direct generalization of the three-part simplex. Accordingly, observations are displayed within a geometrical object with four vertices (corresponding to three zero parts), six edges (combinations of two zero parts), and four triangular faces (zero of the opposite vertex part). Visualization in the tetrahedron generalizes the properties of the ternary diagram, but the necessity to display and interpret the data in two dimensions makes this tool rather rarely used in practice.

Both the ternary diagram and the tetrahedron represent possibilities how to display the (projected) original compositional data. Therefore, by considering that compositions are driven by the Aitchison geometry, such graphical tools must necessarily have some limitations. On the other hand, they can still be used to get a raw impression about the data structure. In the following chapters, other possibilities of graphical displays of compositions are introduced, mainly those that result from a multivariate statistical technique. One prominent tool in exploratory data analysis is the biplot, showing loadings and scores from principal component analysis, see Sect. 7.3. All these statistical methods are performed in logratio coordinates, preferably in an ilr coordinate representation. Accordingly, it is also possible to add any further (non-compositional) variables in a consistent way to supplement information provided by the composition.

References

J. Aitchison, *The Statistical Analysis of Compositional Data* (Chapman & Hall, London, 1986). Reprinted in 2003 with additional material by The Blackburn Press

A. Buccianti, V. Pawlowsky-Glahn, New perspectives on water chemistry and compositional data analysis. Math. Geol. **37**(7), 703–727 (2005)

J.J. Egozcue, V. Pawlowsky-Glahn, Groups of parts and their balances in compositional data analysis. Math. Geol. **37**(7), 795–828 (2005)

J.J. Egozcue, D. Lovell, V. Pawlowsky-Glahn, Testing compositional association, in *Proceedings of the 5th International Workshop on Compositional Data Analysis, Vorau*, ed. by K. Hron, P. Filzmoser, M. Templ (2013)

P. Filzmoser, K. Hron, C. Reimann, Univariate statistical analysis of environmental (compositional) data: problems and possibilities. Sci. Total Environ. **407**, 6100–6108 (2009)

P. Filzmoser, K. Hron, C. Reimann, The bivariate statistical analysis of enviromental (compositional) data. Sci. Total Environ. **408**(19), 4230–4238 (2010)

J. Graffelman, J.J. Egozcue, Hardy-Weinberg equilibrium: a non-parametric compositional approach, in *Compositional Data Analysis: Theory and Applications*, ed. by V. Pawlowsky-Glahn, A. Buccianti (Wiley, Chichester, 2011), pp. 207–215

K. Hron, L. Kubáček, Statistical properties of the total variation estimator for compositional data. Metrika **74**(2), 221–230 (2011)

K. Hron, P. Filzmoser, P. de Caritat, E. Fišerová, A. Gardlo, Weighted pivot coordinates for compositional data and their application to geochemical mapping. Math. Geosci. **49**(6), 797–814 (2017)

G. Mateu-Figueras, V. Pawlowsky-Glahn, A critical approach to probability laws in geochemistry. Math. Geosci. **40**(5), 489–502 (2008)

J.M. McKinley, K. Hron, E.C. Grunsky, C. Reimann, P. de Caritat, P. Filzmoser, K.G. van den Boogaart, R. Tolosana-Delgado, The single component geochemical map: fact or fiction? J. Geochem. Explor. **162**, 16–28 (2016)

C. Mert, P. Filzmoser, K. Hron, Error propagation in compositional data analysis: theoretical and practical considerations. Math. Geosci. **48**(8), 941–961 (2016)

V. Pawlowsky-Glahn, J.J. Egozcue, Geometric approach to statistical analysis on the simplex. Stoch. Env. Res. Risk A. **15**(5), 384–398 (2001)

V. Pawlowsky-Glahn, J.J. Egozcue, BLU estimators and compositional data. Math. Geol. **34**(3), 259–274 (2002)

V. Pawlowsky-Glahn, J.J. Egozcue, D. Lovell, Tools for compositional data with a total. Stat. Model. **15**(2), 175–190 (2015a)

V. Pawlowsky-Glahn, J.J. Egozcue, R. Tolosana-Delgado, *Modeling and Analysis of Compositional Data* (Wiley, Chichester, 2015b)

C. Reimann, P. Filzmoser, K. Fabian, K. Hron, M. Birke, A. Demetriades, E. Dinelli, A. Ladenberger, The GEMAS Project Team, The concept of compositional data analysis in practice-total major element concentrations in agricultural and grazing land soils of Europe. Sci. Total Environ. **426**, 196–210 (2012)

M. Templ, P. Filzmoser, C. Reimann, Cluster analysis applied to regional geochemical data: problems and possibilities. Appl. Geochem. **23**(8), 2198–2213 (2008)

R. Tolosana-Delgado, V. Pawlowsky-Glahn, Kriging regionalized positive variables revisited: sample space and scale considerations. Math. Geol. **39**(6), 529–558 (2007)

K.G. van den Boogaart, R. Tolosana-Delgado, *Analyzing Compositional Data with R* (Springer, Heidelberg, 2013)

H. von Eynatten, V. Pawlowsky-Glahn, J.J. Egozcue, Understanding perturbation on the simplex: a simple method to better visualize and interpret compositional data in ternary diagrams. Math. Geol. **34**(3), 249–257 (2002)

Chapter 5
First Steps for a Statistical Analysis

Abstract Following consistently the principles of compositional data analysis has serious impacts for distributional modeling and statistical processing in general. Particularly, due to the lack of scale invariance, the known Dirichlet distribution is no longer the "must" as the underlying distribution of compositions. It is rather preferred to make use of the concept of normal distribution on the simplex, because the appropriateness of the distribution can be verified by using a standard normality test in coordinates, and the parameters are easy to interpret. Consequently, it can be utilized as the underlying distribution for a wide range of popular methods and tests, including Hotelling tests and MANOVA models in any orthonormal coordinate representation. Because compositional data frequently contain outliers, data inconsistencies, rounding effects, dependencies among the observations, etc., it is recommendable to apply robust counterparts to classical methods in practice. Either univariate or multivariate robust statistical processing can be performed, based on such logratio coordinate representation that serves the purpose of the analysis. Even the classical estimators of location and scale, the sample mean and the sample covariance matrix, are highly sensitive to outliers. As robust alternatives, affine equivariant estimators (like the MCD estimator) are preferred as they can be computed in any coordinate representation. Robust estimators of location and scale can then be used to compute Mahalanobis distances in order to identify multivariate outliers.

5.1 Distributions and Statistical Inference

Before the logratio methodology was introduced, a standard approach for modeling compositional data was based on the Dirichlet distribution, defined for their proportional representation through the corresponding density function as

$$f(\mathbf{x};\alpha_1,\ldots,\alpha_D)=\frac{\Gamma(\alpha_1+\ldots+\alpha_D)}{\Gamma(\alpha_1)\cdot\ldots\cdot\Gamma(\alpha_D)}x_1^{\alpha_1-1}x_2^{\alpha_2-1}\ldots x_D^{\alpha_D-1}, \tag{5.1}$$

P. Filzmoser et al., *Applied Compositional Data Analysis*, Springer Series in Statistics, https://doi.org/10.1007/978-3-319-96422-5_5

for $\mathbf{x} \in S^D$ and zero otherwise (Aitchison 1986). Here, $\Gamma(\cdot)$ denotes the Euler gamma function and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_D)'$ are positive parameters. The latter are used to derive, among others, the expectation and mode of the Dirichlet random vector. The Dirichlet distribution has many advantageous theoretical properties, like that marginal distributions are again Dirichlet distributions. Moreover, the Dirichlet distribution is widely applied, e.g., in Bayesian statistics, where it serves as prior and posterior distribution. Finally, the fixed proportional representation also allows to work with zero values in the composition. This seems like a great advantage for analyzing compositional data in practice. However, there is also a severe shortcoming: The approach based on the Dirichlet distribution is not scale invariant, a major principle in compositional data analysis, and this also implies that other important principles from Sect. 1.3 are violated. Scale invariance would mean that rescaling of the compositional parts results in rescaling of the parameters $\boldsymbol{\alpha}$, and in invariance of the derived estimates. An example for the lack of invariance is the mode of the Dirichlet distribution, which is defined as

$$\text{mode}(\mathbf{x}; \boldsymbol{\alpha}) = \left(\frac{\alpha_1 - 1}{\sum_{i=1}^{D} \alpha_i - D}, \ldots, \frac{\alpha_D - 1}{\sum_{i=1}^{D} \alpha_i - D} \right)'.$$

Suppose that $D = 3$, and $\boldsymbol{\alpha}_1 = (1, 3, 4)'$. Then the mode is $\text{mode}(\mathbf{x}; \boldsymbol{\alpha}_1) = (0, 0.4, 0.6)'$. Multiplication of $\boldsymbol{\alpha}_1$ by two leads to $\boldsymbol{\alpha}_2 = (2, 6, 8)'$, which results in $\text{mode}(\mathbf{x}; \boldsymbol{\alpha}_1) = (0.077, 0.385, 0.538)'$. Note also that (5.1) gives a density function only if the compositional parts sum up to one. All these intrinsic features of the Dirichlet distribution cannot be overcome even by re-defining it with respect to the Aitchison geometry (Monti et al. 2011; Pawlowsky-Glahn et al. 2015).

The case of the Dirichlet distribution shows that one needs to be careful when using seemingly established tools for modeling compositional data in their broader definition, where principles of scale and permutation invariance, respectively, and subcompositional coherence play a crucial role. And even further, it seems to be questionable, to which extent it is meaningful to develop distributions directly for the raw compositions, endowed with the Aitchison geometry, when it is possible to represent and analyze the compositions also in coordinates. Namely, any probability distribution of compositional data can be defined directly in (preferably) orthonormal coordinates, and when necessary, re-expressed for the original data. A prominent case is the **normal distribution on the simplex** (Mateu-Figueras and Pawlowsky-Glahn 2008) that is followed by a D-part random composition $\mathbf{x}$, if the random vector of its orthonormal coordinates follows a multivariate normal distribution on $\mathbb{R}^{D-1}$. Accordingly, the density for a given coordinate representation $\mathbf{z}$ of $\mathbf{x}$ is obtained as

$$f(\mathbf{z}; \boldsymbol{\mu}_\mathbf{z}, \boldsymbol{\Sigma}_\mathbf{z}) = \frac{1}{(2\pi)^{(D-1)/2} |\boldsymbol{\Sigma}_\mathbf{z}|^{1/2}} \cdot \exp\left[-\frac{1}{2} (\mathbf{z} - \boldsymbol{\mu}_\mathbf{z})' \boldsymbol{\Sigma}^{-1} (\mathbf{z} - \boldsymbol{\mu}_\mathbf{z}) \right], \qquad (5.2)$$

where $|\boldsymbol{\Sigma}_\mathbf{z}|$ denotes the determinant of $\boldsymbol{\Sigma}_\mathbf{z}$.

As indicated, the parameters $\boldsymbol{\mu}_{\mathbf{z}}$ and $\boldsymbol{\Sigma}_{\mathbf{z}}$, mean and covariance matrix of the random vector $\mathbf{z}$, depend on the concrete coordinate representation, defined through the matrix $\mathbf{V}$ of the basis compositions in clr coefficients as $\mathbf{z} = \mathbf{V}'\mathbf{y}$. For another choice of coordinates $\tilde{\mathbf{z}} = \tilde{\mathbf{V}}'\mathbf{y}$ and the relation $\tilde{\mathbf{z}} = \mathbf{Q}\mathbf{z}$ through an orthogonal matrix $\mathbf{Q} = \tilde{\mathbf{V}}'\mathbf{V}$ (see Sect. 3.3.3) one gets also normally distributed coordinates, whose parameters transform accordingly as

$$\boldsymbol{\mu}_{\tilde{\mathbf{z}}} = \mathbf{Q}\boldsymbol{\mu}_{\mathbf{z}}, \quad \boldsymbol{\Sigma}_{\tilde{\mathbf{z}}} = \mathbf{Q}\boldsymbol{\Sigma}_{\mathbf{z}}\mathbf{Q}'.$$

Consequently, if compositions in any ilr coordinate representation fulfill the assumption of normality, then it is preserved for any other choice of orthonormal coordinates. Any mean value parameter $\boldsymbol{\mu}_{\mathbf{z}}$ in coordinates can also be expressed using an inverse ilr mapping, like (3.22) for the case of pivot coordinates, as parameter $\boldsymbol{\mu}$ that stands for the (theoretical) compositional center (see Sect. 4.1).

It should be emphasized that the goal of a coordinate representation of compositional data is not to achieve any concrete distribution, like it is frequently done in practice, for example, by the log- or Box-Cox transformations, to obtain normality. The aim is purely to represent compositions in an appropriate sample space without any preliminary distributional assumptions. Depending on the resulting data matrix (and on the output of some statistical test on the data distribution) one should decide whether the assumption of a certain distribution can be used, or not.

5.1.1 Normality Testing

In case of normality, it turned out to be difficult to construct an overall "acceptable" test for multivariate normality in more than two dimensions because of the large number of things that can go wrong (Johnson and Wichern 2007). Therefore, the focus is usually on the behavior of the observations in one or two dimensions (e.g., marginal distributions and scatter plots) in addition to testing the input multivariate data. Although one cannot have a guarantee that some features were not missed that could be revealed only in higher dimensions, for many types of non-normality the focus on univariate and bivariate testing is fully sufficient. Since different orthonormal coordinates are mutual rotations of each other, the results of such univariate and bivariate tests will strongly depend on the actual data configuration. Therefore, it is desirable to find a testing procedure that would produce unambiguous results for any choice of ilr coordinates.

In Aitchison et al. (2004) a battery of tests based on singular value decomposition (SVD) of the $n \times (D-1)$ matrix $\mathbf{Z}$ of (any) mean-centered ilr coordinates is proposed. For a given number of components $p \leq \min\{D-1, n\}$, the SVD decomposes $\mathbf{Z}$ into three parts,

$$\mathbf{Z} = \mathbf{U}\mathbf{D}\mathbf{W}', \tag{5.3}$$

where $\mathbf{U}$ is an $n \times p$ orthogonal matrix containing the left singular vectors, $\mathbf{D}$ is a diagonal matrix ($p \times p$) containing the positive singular values, and $\mathbf{W}$ is a $(D-1) \times p$ orthogonal matrix containing the right singular vectors. Note that the SVD decomposition of $\mathbf{Z}$ is also closely connected with principal component analysis of compositional data, see Chap. 7. As a consequence, $\mathbf{UD}$ forms again a matrix of ilr coordinates whose variables are uncorrelated. When considering just the matrix $\mathbf{U} = \mathbf{ZWD}^{-1}$, uncorrelated and normed (nonzero) ilr coordinates are obtained. Under the assumption of normality of $\mathbf{Z}$, $\mathbf{U}$ thus follows a $(D-1)$-variate standard normal distribution with independent components. The mentioned battery of tests then consists of two basic levels:

1. Univariate tests for marginal normality are performed for the columns of the matrix $\mathbf{U}$. For this purpose, the Anderson-Darling test can be employed (Aitchison 1986), but also any common normality test including graphical evaluations (Q-Q plot) can be applied here.
2. Under the assumption of normality, the squared norm of the rows of $\mathbf{U}$ consist of $D-1$ independent squared standard normal variables, and should thus follow a χ^2 distribution with $D-1$ degrees of freedom. Similar as before, any standard distributional test (like the Kolmogorov-Smirnov test, or again the Q-Q plot) can be performed.

As originally proposed by Aitchison et al. (2004), also a series of bivariate normality tests can be applied to the columns of $\mathbf{U}$. Nevertheless, because the variables are uncorrelated, these tests seem to be rather redundant here.

5.1.2 Statistical Inference in Coordinates

As it is common in statistics, one initially starts with a random sample of compositions $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from a certain distribution and the aim is to deduce properties of the underlying distribution by analyzing this sample. Previous sections claimed that instead of analyzing the original compositions, their proper coordinate representations are preferred. Then the common tools for statistical inference, which are available in the statistical literature, covering estimation, hypotheses testing, etc., can be applied (see, e.g., Anderson 2003). The purpose of this book is not to list these methods here, but just to point out some specificities of their use, mostly linked to the interpretation of ilr coordinates.

Fortunately, almost no peculiarities can be expected when performing statistical inference for compositional data in orthonormal coordinates. This is due to the fact that most multivariate tests are invariant under rotation of the observations, i.e., their result does not depend on the particular choice of orthonormal coordinates.

As an example, this feature is demonstrated for the well-known one-sample Hotelling test, which aims at testing a hypothetic value of the mean vector under the assumption of normality. The null hypothesis can be formulated directly in compositional terms as $H_0 : \boldsymbol{\mu} = \boldsymbol{\mu}_0$, with a hypothetic value of the center $\boldsymbol{\mu}_0$,

against the alternative $H_A : \boldsymbol{\mu} \neq \boldsymbol{\mu}_0$. The composition $\boldsymbol{\mu}_0$ can also be equal to the neutral element $\mathbf{n}$, especially when pure random effects are of primary interest. After expressing the compositional sample in orthonormal coordinates, $\mathbf{z}_1, \ldots, \mathbf{z}_n$, the null and alternative hypotheses are reformulated accordingly, $H_0 : \boldsymbol{\mu}_\mathbf{z} = \boldsymbol{\mu}_{\mathbf{z}0}$ against $H_A : \boldsymbol{\mu}_\mathbf{z} \neq \boldsymbol{\mu}_{\mathbf{z}0}$. Hereat, $\boldsymbol{\mu}_{\mathbf{z}0} = \text{ilr}(\boldsymbol{\mu}_0)$ is simply any (ilr) coordinate representation of the original hypothetic center $\boldsymbol{\mu}_0$. Then the testing procedure can proceed as usual, i.e., sample mean and covariance matrix are computed,

$$\overline{\mathbf{z}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{z}_i, \quad \mathbf{S}_\mathbf{z} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{z}_i - \overline{\mathbf{z}})(\mathbf{z}_i - \overline{\mathbf{z}})',$$

and the final test statistic under the null hypothesis

$$F = \frac{n[n-(D-1)]}{(D-1)(n-1)}(\overline{\mathbf{z}} - \boldsymbol{\mu}_\mathbf{z})'\mathbf{S}_\mathbf{z}^{-1}(\overline{\mathbf{z}} - \boldsymbol{\mu}_\mathbf{z}) \sim F_{D-1,n-D+1}. \tag{5.4}$$

An important point is whether the test statistic would give the same result for a different coordinate representation of the compositions. In order to check the invariance of F it is sufficient to show that for any other coordinate representation $\mathbf{z}^* = \mathbf{Q}\mathbf{z}$ the following relation holds

$$(\overline{\mathbf{z}}^* - \boldsymbol{\mu}_\mathbf{z}^*)'\mathbf{S}_{\mathbf{z}^*}^{-1}(\overline{\mathbf{z}}^* - \boldsymbol{\mu}_{\mathbf{z}^*}) = (\mathbf{Q}\overline{\mathbf{z}} - \mathbf{Q}\boldsymbol{\mu}_\mathbf{z})'\mathbf{Q}\mathbf{S}_\mathbf{z}^{-1}\mathbf{Q}'(\mathbf{Q}\overline{\mathbf{z}} - \mathbf{Q}\boldsymbol{\mu}_\mathbf{z}) =$$

$$(\overline{\mathbf{z}} - \boldsymbol{\mu}_\mathbf{z})'\mathbf{Q}'\mathbf{Q}\mathbf{S}_\mathbf{z}^{-1}\mathbf{Q}'\mathbf{Q}(\overline{\mathbf{z}} - \boldsymbol{\mu}_\mathbf{z}) = (\overline{\mathbf{z}} - \boldsymbol{\mu}_\mathbf{z})'\mathbf{S}_\mathbf{z}^{-1}(\overline{\mathbf{z}} - \boldsymbol{\mu}_\mathbf{z}).$$

By doing that, two features were utilized. The first refers to the orthogonality of the matrix $\mathbf{Q}$, $\mathbf{Q}'\mathbf{Q} = \mathbf{Q}\mathbf{Q}' = \mathbf{I}_{D-1}$, the latter comes from the *orthogonal equivariance* of the sample mean and covariance matrix. In simple terms, this means that there is no privileged direction in the $(D-1)$-dimensional space that would allow one to bias the estimators in some specific directions. Formally, a location estimator $\mathbf{t}$ and a covariance estimator $\mathbf{C}$ share the property of orthogonal equivariance, if for any orthogonal matrix $\mathbf{Q}$ of full rank the conditions

$$\begin{aligned}\mathbf{t}(\mathbf{Q}\mathbf{z}_1, \ldots, \mathbf{Q}\mathbf{z}) &= \mathbf{Q}\mathbf{t}(\mathbf{z}_1, \ldots, \mathbf{z}_n),\\ \mathbf{C}(\mathbf{Q}\mathbf{z}_1, \ldots, \mathbf{Q}\mathbf{z}_n) &= \mathbf{Q}\mathbf{C}(\mathbf{z}_1, \ldots, \mathbf{z}_n)\mathbf{Q}'\end{aligned} \tag{5.5}$$

are fulfilled. Accordingly, the sample mean and sample covariance matrix estimators follow the usual properties of the expectation and (theoretical) covariance matrix, respectively. Note that orthogonal equivariance is a special case of affine equivariance (see Sect. 5.2.3), typically applied when arbitrary (also non-orthonormal) logratio coordinates are considered (Filzmoser and Hron 2008; Filzmoser et al. 2012b).

The case of the Hotelling test has demonstrated that statistical inference is straightforward if the data are expressed in coordinates. This is similar for many

other statistical methods and tests. In particular, for those methods which are object-oriented, like MANOVA, cluster analysis, or discriminant analysis, the choice of the orthonormal coordinates usually does not matter. This is no more the case, e.g., in regression analysis, if the explanatory variables come from a composition, and if tests on the regression parameters are required. Then more specific coordinate representations are needed that allow for an appropriate interpretation of the outcome. Details will be discussed in the following chapters.

5.2 Classical and Robust Statistical Analysis

Compositional data, like any other statistical data, can contain outliers, data inconsistencies, rounding effects, dependencies among the observations, etc. Many classical statistical methods rely on strict model assumptions, like independence or (multivariate) normal distribution, and violations of the assumptions can lead to biased results. Robust statistics offers a methodological approach that tolerates certain deviations from strict model assumptions (Hampel et al. 1986). The basic idea behind robust statistical methods is to fit a statistical model to the data majority, and not to satisfy every single data point with one and the same model. A prominent example are outliers in simple linear regression analysis, which can completely spoil the regression line for least-squares estimation, while a robust regression line fits those data points which form the majority and show a linear trend. Robust methods assign appropriate weights to the data points, usually in the range $[0, 1]$, where weights close to one refer to observations that fully support the model, and outliers that are deviating from the model obtain small weights (Maronna et al. 2006).

Various measures of robustness have been proposed in the literature. One important tool is the *influence function* of an estimator (Hampel et al. 1986), which investigates the behavior of the estimator under small (infinitesimal) amounts of contamination. In general, it is desirable that a robust estimator does not change arbitrarily in presence of contamination. For example, the arithmetic mean can go towards infinity if one observation is moved arbitrarily far away, while the median remains more or less stable.

Another concept is the *breakdown point* of an estimator, which measures the degree of robustness with respect to larger amounts of contamination. Loosely speaking, the breakdown point of an estimator corresponds to the minimal fraction of arbitrary contamination that drives the estimator beyond all bounds. For the arithmetic mean it is sufficient to move only one observation (out of n) arbitrarily far away in order to cause "break down." In the limit, for $n \longrightarrow \infty$, this corresponds to a breakdown point of zero. On the other hand, the median gives non-sense if at least half of the observations are replaced by arbitrary numbers, and thus the breakdown point of the median is 50%, the highest possible value.

There are still other features of a robust estimator, like Fisher consistency, that are important. Moreover, like in classical statistics, a good robust estimator should also achieve high efficiency (Maronna et al. 2006).

Robust estimators as a complement to classical ones will be mentioned throughout the book. These estimators are defined in the following. Note that the listed estimators are by far not exhaustive, and thus interested readers are referred to the literature on robust statistics (Maronna et al. 2006). It is important to note that the robust estimators used here are exclusively applied in logratio coordinates, and not in the original sample space of compositions.

5.2.1 Univariate Location

Although compositional data are by definition multivariate data, univariate estimation is important in some cases, for example when considering a single pairwise logratio, or an interpretable orthonormal coordinate. The classical estimator of the location parameter of a normal distribution is the arithmetic mean. In presence of outliers, leading to deviations from normality, a robust alternative is the median, the innermost value of the sorted data. The median has good robustness properties, but low efficiency—under normality one would need about one third more data to achieve the same efficiency (precision) as for the arithmetic mean. Increasing the efficiency requires using more of the available data information, i.e. more than just the order of the values in case of the median. The *trimmed mean* is a compromise, where a tuning parameter $\alpha \in (0, 0.5)$ regulates the amount of trimming. Another compromise are M-estimators of location, that combine good robustness properties with high efficiency (Maronna et al. 2006).

5.2.2 Univariate Scale

Consider a normal distribution $N(\mu, \sigma^2)$, then scale estimators considered here estimate the parameter σ. The classical scale estimator for given data $x_1, \ldots, x_n$ (not the original compositional data!) is the empirical standard deviation

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2},$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the arithmetic mean.

A robust counterpart is the **median absolute deviation**,

$$s_{\mathrm{MAD}} = 1.4826 \cdot \mathrm{median}_i |x_i - \tilde{x}|,$$

where $\tilde{x}$ denotes the median of the sample. The factor 1.4826 makes s_{MAD} a consistent estimator for σ, which means that due to this correction factor one indeed obtains an estimator for the parameter σ.

Another popular robust scale estimator is the **interquartile range**,

$$s_{\mathrm{IQR}} = 0.7413 \cdot (Q_{0.75} - Q_{0.25}),$$

where Q_k denotes the k-quantile. Similar as before, the normalizing constant serves for the purpose of consistency of the estimator.

5.2.3 *Multivariate Location and Covariance*

Consider multivariate non-compositional observations $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$, which form the rows of the $n \times p$ data matrix $\mathbf{X}$. In subsequent chapters, these observations will typically be the compositions, expressed in coordinates. The classical estimators for location and covariance are the arithmetic mean vector $\bar{\mathbf{x}}$ and the sample covariance matrix $\mathbf{S}_\mathbf{x}$, defined as

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \ , \quad \mathbf{S}_\mathbf{x} = \frac{1}{n-1}\sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})'.$$

Both estimators are highly sensitive to outliers—their breakdown point is zero (Maronna et al. 2006). However, they transform properly under affine transformations. An affine transformation of the data $\mathbf{X}$ is given by a non-singular $p \times p$ matrix $\mathbf{A}$ and a vector $\mathbf{b}$ of length p as

$$\mathbf{Y} = \mathbf{X}\mathbf{A} + \mathbf{1}_n \mathbf{b}',$$

where the outcome is denoted as $\mathbf{Y}$. So, $\mathbf{Y}$ can be any shifted, rotated, and rescaled version of $\mathbf{X}$. Accordingly, estimators of location $\mathbf{t}$ (in this context exceptionally as a row vector) and covariance $\mathbf{C}$ are called **affine equivariant** if they satisfy

$$\mathbf{t}(\mathbf{Y}) = \mathbf{t}(\mathbf{X})\mathbf{A} + \mathbf{b},$$
$$\mathbf{C}(\mathbf{Y}) = \mathbf{A}'\mathbf{C}(\mathbf{X})\mathbf{A},$$

where the matrices in brackets refer to the data sets the estimators are applied to. For the special case of considering just rotation of the initial data, the affine equivariance reduces to orthogonal equivariance, introduced in the previous section in the context of statistical inference in coordinates (5.5). Affine equivariant estimators enable to consider theoretically also coordinate representations different from orthonormal ones (like alr coordinates) for their computation. The definitions of arithmetic mean and sample covariance matrix reveal that these classical estimators share the equivariance property.

Nowadays, several robust counterparts are available. One popular estimator of multivariate location and covariance is the **minimum covariance determinant**

(MCD) estimator (Rousseeuw 1985; Rousseeuw and Van Driessen 1999). It is defined by that subset $\{\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_h}\}$ of h observations whose sample covariance matrix has the smallest determinant among all possible subsets of size h. The MCD location estimator $\mathbf{t}_{\text{MCD}}$ is given by the arithmetic mean of the h observations, and the MCD covariance estimator $\mathbf{C}_{\text{MCD}}$ by their sample covariance, multiplied by a factor c_{MCD} for consistency under normality,

$$\mathbf{t}_{\text{MCD}} = \frac{1}{h}\sum_{j=1}^{h}\mathbf{x}_{i_j},$$

$$\mathbf{C}_{\text{MCD}} = c_{\text{MCD}} \cdot \frac{1}{h-1}\sum_{j=1}^{h}(\mathbf{x}_{i_j} - \mathbf{t}_{\text{MCD}})(\mathbf{x}_{i_j} - \mathbf{t}_{\text{MCD}})'.$$

The number h has to refer to the data majority; it can be taken as an integer in the interval $[(n + p + 1)/2, n]$. The highest breakdown point of about 50% is achieved for the smallest value of h, but this also leads to low efficiency. A compromise in practice is to take $h \approx 0.75 \cdot n$. The MCD estimators $\mathbf{t}_{\text{MCD}}$ and $\mathbf{C}_{\text{MCD}}$ are affine equivariant (Rousseeuw 1985).

The MCD estimator became popular because of the availability of a fast algorithm for its computation (Rousseeuw and Van Driessen 1999). However, there are also limitations, like low efficiency at normal models, which can be overcome by a reweighting step, see Maronna et al. (2006). Another limitation is that the MCD estimator does not work for data sets with more variables than observations, because the determinant of the covariance matrix of any subset would always yield zero, and for the same reason also in clr coefficients. This is an issue especially for high-dimensional low sample size data. A robust estimator for this situation is the **orthogonalized Gnanadesikan-Kettenring (OGK)** estimator (Maronna and Zamar 2002), which is based on a robust pairwise estimation of the covariances, but ensures that the resulting covariance matrix is positive definite. The OGK estimator, however, is not affine equivariant.

5.2.4 Center and Variation Matrix

The variation matrix was introduced in Eq. (4.2) as the matrix consisting of the sample variances of all pairwise logratios. Theoretically, for a composition $\mathbf{x} = (x_1, \ldots, x_D)'$, here in terms of random variables, the element (j, k) of the variation matrix is defined as $\text{var}(\ln(x_j/x_k))$, where "var" denotes the variance, and $j, k \in \{1, \ldots, D\}$. This can be written as

$$\text{var}\left(\ln\frac{x_j}{x_k}\right) = \text{var}(\ln x_j - \ln x_k) = \text{var}(\ln x_j) + \text{var}(\ln x_k) - 2\text{cov}(\ln x_j, \ln x_k), \tag{5.6}$$

where "cov" denotes the covariance. On the other hand, consider clr coefficients $\mathbf{y} = (y_1, \ldots, y_D)'$, with $y_j = \ln x_j - \ln g_m(\mathbf{x})$, see (3.14), for $j = 1, \ldots, D$. Then

$$\begin{aligned}\operatorname{var}(\ln y_j) + \operatorname{var}(\ln y_k) - 2\operatorname{cov}(\ln y_j, \ln y_k) &= \operatorname{var}(\ln x_j - \ln g_m(\mathbf{x})) \\ &+ \operatorname{var}(\ln x_k - \ln g_m(\mathbf{x})) - 2\operatorname{cov}(\ln x_j - \ln g_m(\mathbf{x}), \ln x_k - \ln g_m(\mathbf{x})),\end{aligned}$$

which is equal to (5.6). This equality can be written in matrix notation. Denote $\mathbf{T}$ as the (theoretical) variation matrix and $\operatorname{Cov}(\mathbf{y})$ as the clr covariance matrix. Then the following relation is obtained,

$$\mathbf{T} = \mathbf{J}\operatorname{diag}(\operatorname{Cov}(\mathbf{y})) + \operatorname{diag}(\operatorname{Cov}(\mathbf{y}))\mathbf{J} - 2\operatorname{Cov}(\mathbf{y}), \tag{5.7}$$

where $\mathbf{J}$ denotes a $D \times D$ matrix of ones.

With this relation it is straightforward to robustly estimate the variation matrix. One simply has to plug-in a robust estimate of the covariance matrix in (5.7), but this needs to be done in coordinates. With the findings in the previous section, a robust covariance estimation, e.g., in pivot coordinates (3.19), is obtained as $\mathbf{C}_{\text{MCD}}$ by the MCD estimator, which yields a robust estimation of the variation matrix

$$\mathbf{T}_{\text{MCD}} = \mathbf{J}\operatorname{diag}(\mathbf{V}\mathbf{C}_{\text{MCD}}\mathbf{V}') + \operatorname{diag}(\mathbf{V}\mathbf{C}_{\text{MCD}}\mathbf{V}')\mathbf{J} - 2\mathbf{V}\mathbf{C}_{\text{MCD}}\mathbf{V}', \tag{5.8}$$

where the $D \times (D-1)$ matrix $\mathbf{V}$ is defined in (3.23).

For the center, a robust counterpart is obtained simply by taking the resulting subset of h original observations for computing the geometric mean $\mathbf{g}_{\text{MCD}}$ using Eq. (4.1) for this subset.

5.3 Outlier Detection

In Sect. 5.2 it was argued that data outliers can spoil classical estimators, and therefore robust counterparts are usually preferable. Outliers are widely present in real data sets (Barnett and Lewis 1994). In robust statistics, outliers are downweighted in order to reduce their effect on the estimation. The outlier weight is a result of the robust estimation procedure; depending on the procedure, the weight can be either zero (outlier) or one (regular data point), but it can also be a real number in the interval $[0, 1]$. There is a frequent misunderstanding: downweighting outlying data points does not imply that this observation is non-informative and should thus be discarded from the data. On the contrary, outliers are often the most interesting observations, because some atypical phenomenon is responsible for their presence; they are solely downweighted in order to get a model fit which accommodates the data majority. A subsequent inspection of the observations with small weight is an important step in the analysis.

The focus in this section is on identifying observations that deviate from an underlying model distribution. Here only the (multivariate) normal distribution is considered, which serves as an important model distribution for many statistical methods.

5.3.1 Univariate Outliers

It is assumed that the underlying univariate data follow a normal distribution with certain parameters. Possible outliers, however, come from a different distribution; either not from a normal distribution or from a normal distribution but with different mean and/or variance. The data analyst observes the joint data and thus the joint distribution, and depending on the position of the outliers, it will be difficult to identify them.

Note that the "univariate data" under consideration will not be compositional data, since compositions are by definition multivariate. Univariate data could be non-compositional variables, they could refer to a logratio, or to a coordinate, like the first pivot coordinate z_1 defined in Eq. (3.19) referring to all relative information about the first part x_1 within the composition considered.

A standard procedure to identify univariate outliers under the assumption of normality is the following. Assume that the regular observations are generated from the normal distribution $N(\mu, \sigma^2)$, where mean μ and variance σ^2 are unknown. From the normal theory it is known that the interval

$$[\mu - 1.96 \cdot \sigma, \mu + 1.96 \cdot \sigma] \tag{5.9}$$

contains the "inner" 95% of the distribution, because the left boundary corresponds to quantile $Q_{0.025}$ and the right one to $Q_{0.975}$ of the distribution $N(\mu, \sigma^2)$. Accordingly, if a data point falls outside this interval, which will happen in the limit in 5% of the cases, this observation is "unusual" and could be treated as an outlier. Following the above thoughts, it would be unclear if this outlier was indeed generated by a different distribution, or if it is located just in the extremes of the same distribution. In practice this is hard or even impossible to distinguish (Filzmoser et al. 2005).

For a sample $x_1, x_2, \ldots, x_n$, the interval boundaries in (5.9) need to be estimated appropriately. Since it is believed that outliers are present in the sample, they would potentially have an effect on the classical estimators $\bar{x}$ and s of μ and σ, respectively. This can be prevented by using robust counterparts as proposed in Sects. 5.2.1 and 5.2.2. So, if $\tilde{x}$ denotes the median of the sample, robust alternatives to (5.9) are

$$[\tilde{x} - 1.96 \cdot s_{MAD}, \tilde{x} + 1.96 \cdot s_{MAD}] \text{ or } [\tilde{x} - 1.96 \cdot s_{IQR}, \tilde{x} + 1.96 \cdot s_{IQR}]. \tag{5.10}$$

Note that for a normally distributed sample with very large n, hence data without any outliers, the resulting intervals would be essentially identical with the interval based on classical estimators, $[\bar{x} - 1.96 \cdot s, \bar{x} + 1.96 \cdot s]$. This will in general be different for small samples.

A further possibility for identifying univariate outliers is the Tukey boxplot (Tukey 1977). This boxplot uses the quantiles $Q_{0.25}$, $Q_{0.5}$ = median, and $Q_{0.75}$ for its construction. Outliers are those data points which are outside the interval

$$[Q_{0.25} - 1.5 \cdot \text{IQR}, Q_{0.75} + 1.5 \cdot \text{IQR}],$$

with the *interquartile range* $\text{IQR} = Q_{0.75} - Q_{0.25}$. A comparison of the behavior with the previously mentioned methods is shown, e.g., in Filzmoser et al. (2005).

Example The Austrian presidential election 2016 received quite some attention because the second round with the candidates Hofer and Van der Bellen had to be repeated, see:

```
https://en.wikipedia.org/wiki/Austrian_presidential_election,_2016
```

Finally, Van der Bellen was elected as the president, with 2,472,892 votes, while Hofer received 2,124,661 votes. The raw data of the votes in the Austrian communities are available as data set `electionATbp` in the package **robCompositions**. The data are first restructured.

```
data("electionATbp")
d <- electionATbp # short
bp <- data.frame("Votes" = c(d[, 8], d[, 10]),
           "Percentages" = c(d[, 9], d[, 11]),
      "candidate" = rep(c("Hofer", "Van der Bellen"),
                        each = nrow(d)))
head(bp, 3) # first three observations

##   Votes Percentages candidate
## 1  3753       45.86     Hofer
## 2   681       56.89     Hofer
## 3   580       55.34     Hofer
```

The numbers of votes for the two candidates in the communities are shown as boxplots in Fig. 5.1 (left). The distributions are right-skewed because of the cities (we used a log-scale). The medians thus do not correctly reflect the election result (but the mean does!), because it refers to the ordered absolute values in the communities, and Van der Bellen received more votes in the cities than Hofer. Figure 5.1 (right) shows the percentage data which are more symmetric. Again, the median does not reflect the election result, since it only expresses that Hofer had the majority in more than 50% of the communities.

If the interest is in the relative information, one can construct a coordinate according to (3.19) as $z = \ln(x_1/x_2)/\sqrt{2}$, where x_1 and x_2 are either the votes or the percentage for the two candidates—both lead to the same result. Here x_1 represents the votes for Hofer, and x_2 those for Van der Bellen, and thus positive values of z correspond to a "dominance" of the votes for Hofer.

```
ggplot(bp, aes(x = candidate,
  y = Votes)) + geom_boxplot() +
  coord_trans(y = "log10")
```

```
ggplot(bp, aes(x = candidate,
  y = Percentages)) + geom_boxplot()
```

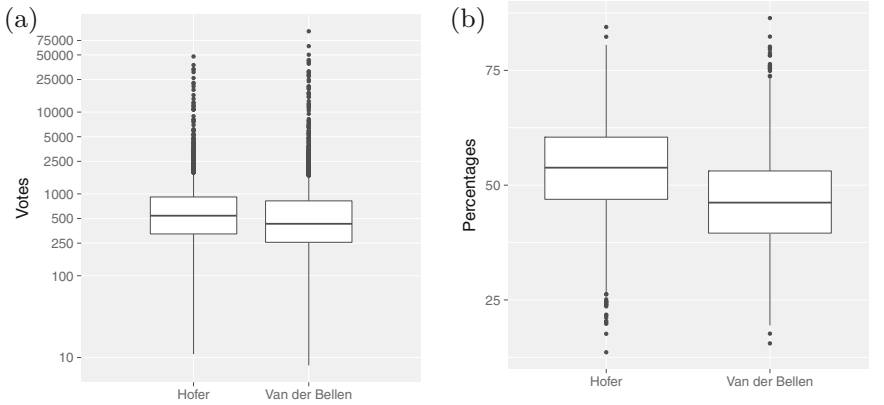


Fig. 5.1 Numbers (log-scale) of votes in the Austrian communities for the two presidential candidates (left), and resulting percentages (right). (**a**) Absolute values. (**b**) Log-transformed data

```
data("electionATbp")
z <- as.matrix(pivotCoord(electionATbp[, c(8,10)]))
# gives the same as
# z <- 1/sqrt(2)*log(x1/x2)
# Identify univariate outliers by:
outup <- median(z) + 1.96 * mad(z)
outlow <- median(z) - 1.96 * mad(z)
outindex <- which(z < outlow | z > outup) # index of outliers
```

Figure 5.2 shows this coordinate (vertical axis) against the index of the observations (horizontal axis), which in fact corresponds to the identity numbers of the communities, and these are sorted. The colors are according to the nine Austrian districts (see also legend below the plot). The bigger symbols are for communities where the number of valid votes was at least 20,000. Outlier rule (5.10) has been applied—the median is the dashed line, and the outlier boundaries are the dashed-dotted lines. It can be seen that several districts of Vienna (W), points on the very right hand-side of the plot, are lower outliers. These are districts with an exceptionally high percentage of votes for Van der Bellen. The most extreme lower outlier is located in Tyrol (T). This is in fact the community Kaunertal, the home village of Van der Bellen. A further quite unusual observation is in the lower range of the values from Carinthia (K). This is the community Zell, where almost 90% of the inhabitants are of Carinthian Slovenian descent. This is the highest percentage of all municipalities in the state of Carinthia.

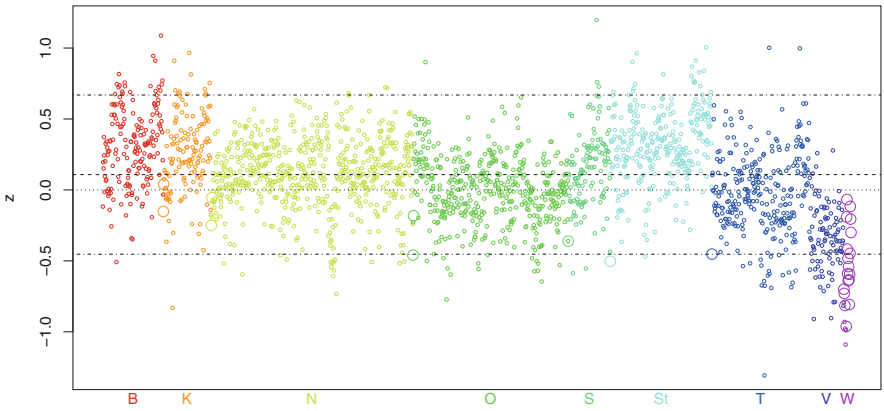


Fig. 5.2 Coordinate representation of the votes; positive values for z correspond to a higher proportion of votes for Hofer, negative values to a higher proportion of votes for Van der Bellen. The points show the communities in the different Austrian districts (color). The dashed-dotted horizontal lines are outlier cut-off values

5.3.2 Multivariate Outliers

Multivariate outliers are in general much harder to find than univariate outliers, because one can no longer rely on graphical inspection. Moreover, multivariate outliers are not necessarily extreme along one coordinate, but they could be located anywhere in the multivariate space.

In this section it is assumed that the multivariate data are compositions. Rather than identifying the outliers directly in the original space, it is common to first express the compositions in logratio coordinates, and then to apply the usual methods for multivariate outlier detection.

Nevertheless, before doing that it is important to realize, what are the sources of outlyingness in compositional data. In contrast to standard multivariate observations that rely on the absolute values of the components, deviating compositions arise due to aberrant (pairwise) logratios between their parts. As pairwise logratios are typically merged into logratio coordinates, preferably orthonormal ones, their appropriate choice can help to reveal, which parts are predominantly responsible for obtaining deviating logratios. On the other hand, for detecting multivariate outliers it is important to have such methods that are able to reveal deviating observations irrespective of the choice of the coordinate system. Another aspect is that due to the relative scale of the compositions, parts with low concentrations will in general tend to produce outliers more likely than dominating components. This feature is often accompanied with approaching the detection limit of measurement devices, leading to lower precision of the output values. Consequently, the resulting outliers can be even more “dangerous” than in the standard case. Graphical inspection of the raw compositional data is thus even more unreliable for the purpose of multivariate

outlier detection than for traditional multivariate data. The above thoughts will be further developed, directly or indirectly, in the whole sequel of the book.

Suppose that a compositional data set $\mathbf{X}$ is available, with n observations (rows) and D compositional parts (columns). Expressing $\mathbf{X}$ in coordinates, e.g. by applying Eq. (3.20) results in the $n \times (D-1)$ matrix $\mathbf{Z}$, with the observations $\mathbf{z}_1, \ldots, \mathbf{z}_n$.

Standard multivariate outlier detection procedures assume that the majority of the observations of $\mathbf{Z}$ are generated by a multivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. For a location estimator $\mathbf{t}$ and a covariance estimator $\mathbf{C}$, the squared Mahalanobis distances between the observations expressed in coordinates and the respective location estimator $\mathbf{t}$,

$$\mathrm{MD}(\mathbf{z}_i)^2 = (\mathbf{z}_i - \mathbf{t})' \mathbf{C}^{-1} (\mathbf{z}_i - \mathbf{t}), \qquad \text{for } i = 1, \ldots, n, \tag{5.11}$$

are approximately following a χ^2 distribution with $D-1$ degrees of freedom, χ^2_{D-1}. Similar to the univariate case, a certain quantile of this distribution, like the quantile 0.975, $\chi^2_{D-1;0.975}$, is used as a cut-off value to identify multivariate outliers as observations

$$\mathbf{z}_i \text{ with } \mathrm{MD}(\mathbf{z}_i)^2 > \chi^2_{D-1;0.975}; \tag{5.12}$$

the remaining observations are considered as regular data points.

It is clear that the classical arithmetic mean (vector) and the sample covariance matrix are inappropriate as estimators $\mathbf{t}$ and $\mathbf{C}$ in Eq. (5.11), since they could be spoiled themselves by the outliers, which would make outlier detection unreliable. Thus, $\mathbf{t}$ and $\mathbf{C}$ should be robust estimators, like the MCD estimators $\mathbf{t}_{\mathrm{MCD}}$ and $\mathbf{C}_{\mathrm{MCD}}$, defined in Sect. 5.2.3.

Example The Austrian presidential election example from the previous section is continued. In addition to the number of eligible votes for the two candidates, also the number of nonvoters in the different communities is considered as a third composition. The relative information is shown in a ternary diagram in Fig. 5.3 (right).

The left graphic shows the plot of the resulting two pivot coordinates. For these coordinates, mean and covariance matrix are estimated, once with classical arithmetic mean and sample covariance matrix, and once with the MCD estimator. The outlier cut-off value is $\chi^2_{2;0.975} = 7.378$. If $\mathbf{z}$ denotes any point in this plane, the squared Mahalanobis distance in (5.11) can be set equal to this cut-off value, and then the results of this equation define an ellipse. This ellipse is also called *97.5% tolerance ellipse*, since in case of bivariate normal distribution it will contain the innermost 97.5% of the data. The ellipses shown in the plot are constructed with the classical (red dashed line) and the robust (green solid line) estimators. All data points outside the (green) ellipse can be considered as multivariate outliers. One can see that the red ellipse differs slightly from the green one, and the reason are the outliers which inflated the classical covariance estimation. Using the inverse

```
par(mfrow = c(1,2))
Z <- pivotCoord(X)
plot(Z, col = "#0066FFFF")
library("ellipse")
mdclass <- ellipse::ellipse(cov(Z),
      centre = apply(Z, 2, mean), level = 0.975)
library("robustbase")
Z.mcd <- covMcd(Z)
mdrob <- ellipse::ellipse(Z.mcd$cov, centre = Z.mcd$center, level = 0.975)
lines(mdclass, col = 2, lty = 2, lwd = 2)
lines(mdrob, col = 3, lwd = 2)
mdclassinv <- pivotCoordInv(mdclass)
mdrobinv <- pivotCoordInv(mdrob)
ternaryDiag(X, col = "#0066FFFF")
ternaryDiagPoints(mdclassinv, col = 2, lty = 2, type = "l", lwd = 2)
ternaryDiagPoints(mdrobinv, col = 3, type = "l", lwd = 2)
legend("topleft", legend = c("Classical","Robust"),
       lty = c(2,1), col = c(2,3))
```

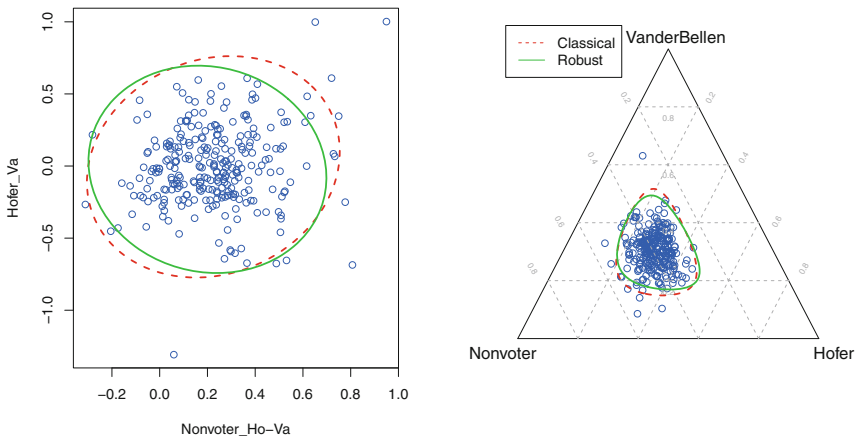


Fig. 5.3 Coordinate representation of the presidential election data (votes for the two candidates and nonvoters) with tolerance ellipses (left), and ternary diagram with tolerance ellipses (right)

mapping (3.22), both ellipses can be presented in the ternary diagram. Here it is easier to interpret the outliers. For example, the outlier in the upper part of the ternary diagram is Kaunertal. Several outliers with high proportion on nonvoters are visible.

Another composition of the presidential election data is used for Fig. 5.4: the number of votes for the two candidates, and the number of invalid votes. Here, all Austrian communities are taken, and the same symbols as in Fig. 5.2 are used. There is no big difference for the classical and robust estimators. As in the previous example, one could argue if the χ^2 distribution for the outlier cut-off is appropriate, because the data do not seem to be normally distributed. Still, the cut-off gives some impression about unusual observations, which are here several districts of Vienna

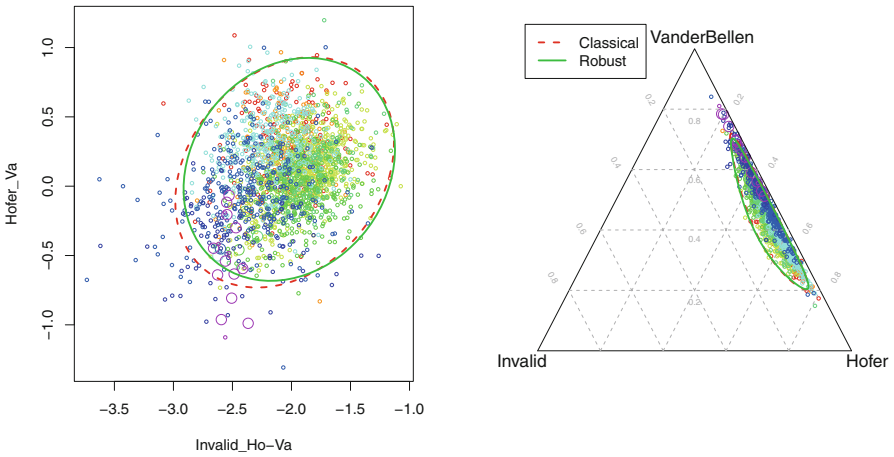


Fig. 5.4 Outlier detection for the valid and invalid presidential votes. The symbols are according to Fig. 5.2

(W) and many communities of Tyrol (T) and Vorarlberg (V). It is interesting to see that also for the proportion of invalid votes clear regional patterns are visible.

5.3.3 Interpretation of Multivariate Outliers

From the coordinate representation in Fig. 5.4 (left) it becomes clear that it is not so straightforward to interpret the plot, and in particular the reason for the outlyingness of some observations. Only when showing the data in the ternary diagram, this interpretation gets easier (Fig. 5.4, right). In the general case, however, one deals with more than three-part compositions, and graphical representations in terms of ternary diagrams are no longer possible. Therefore, Filzmoser et al. (2012a) have proposed some plots that support the interpretation of the reason for multivariate outlyingness.

The basic principle is to show all different pivot coordinates, specifically, the first coordinates from each of the D pivot coordinate systems, and using special symbols. The symbols have been proposed in Filzmoser et al. (2005), and their choice is explained by an artificial data set in Fig. 5.5. The ternary diagram (right plot) shows the three-part compositions. The coloring of the symbols is from blue in the center of the ternary diagram, where the proportions on the parts are very similar, to red on the boundary of the simplex, where the proportions get high on average. The left plot presents this information in terms of ilr coordinates. Similar as in the previous section, multivariate outlier detection is carried out, and the 97.5% tolerance ellipse is included. Points outside this ellipse are multivariate outliers, and the symbol is taken as a big $+$. The more the data points come to the center of the

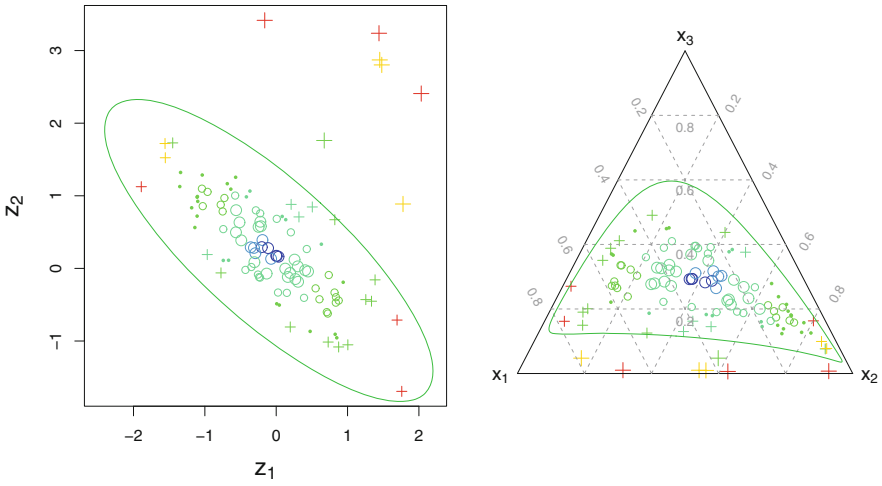


Fig. 5.5 Artificial data set with three-part compositions to explain the choice of color and symbol type for the interpretation of multivariate outliers (see text)

ellipse, the smaller are the Mahalanobis distances, with corresponding changes of the symbols. Now color and symbol type is defined, and this should be helpful for the interpretation. Going back again to the ternary diagram, the tolerance ellipse is presented, and it can be seen that most multivariate outliers have very low values for part x_3.

Color and symbol type, as well as robust Mahalanobis distances are computed as follows:

```
library(mvoutlier)
res <- mvoutlier.CoDa(X)
```

The resulting object `res` contains all this information, which can be visualized in different ways. Figure 5.6 shows two possibilities: The left plot shows pivot coordinates $z_1^{(l)}$, $l = 1, 2, 3$ according to (3.25) for the single compositional parts as parallel vertical axes; along the horizontal direction, random scattering is done to make the observations better visible. It can be seen that most multivariate outliers have high values in pairwise log-ratios with the original parts, aggregated into the respective coordinates (red symbol). This can be seen in the pivot coordinates for x_1 and x_2, but for x_3 the outliers are in the very low range. The plot on the right-hand side shows the pivot coordinates as parallel coordinates. The axes are again arranged vertically in parallel, but the values are normed now to the interval [0, 1], and the individual observations are shown by individual lines. One can see that in particular the lines for the multivariate outliers follow a very different pattern than the data majority.

```
plot(res, which = "uni",
     onlyout = FALSE, cex.main = 2)
```

```
plot(res, which = "parallel",
     onlyout = FALSE)
```

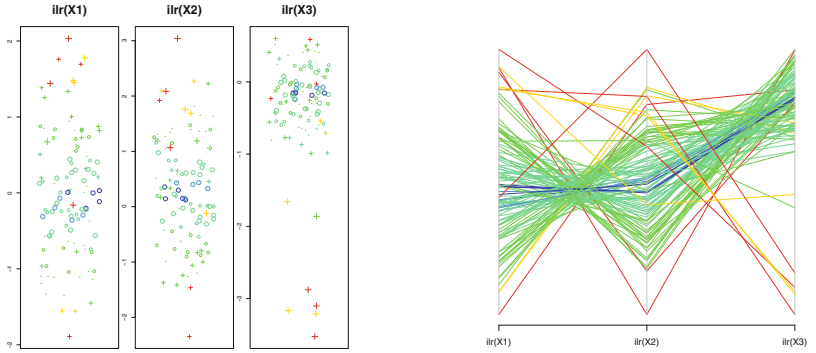


Fig. 5.6 Parallel pivot coordinate plots as scatter plots (left) and parallel coordinate plot (right), with the symbols and colors as defined in Fig. 5.5

5.4 Example

The R package **StatDA** contains data from the so-called Kola project, a multi-media geochemical mapping project carried out from 1993-1998 in the peninsula Kola by the Geological Surveys of Finland (GTK), Norway (NGU), and Central Kola Expedition (CKE) in Russia, see Reimann et al. (1998). More than 600 soil samples in five different layers were analyzed for the concentration of several chemical elements. The project area, which is located on the boundary of Norway, Finland and Russia, is interesting for geochemical mapping because it contains big smelters in Russia, as well as very pristine areas in Norway and in the Finish part.

Here the focus is on the organic surface soil (O-horizon), and in particular on the element concentrations of As, Cd, Co, Cu, Mg, Pb, and Zn. With the exception of Mg and Zn, these elements are in the emission spectrum of the Ni-smelters in the Russian cities Nikel/Zapolyarnij and Monchegorsk.

Multivariate outliers according to the definition in Eq. (5.12) can be flagged via the function `outCoDa` in the package **robCompositions**, see also the corresponding graphical output in Fig. 5.7, where robust Mahalanobis distances together with the cut-off line are plotted.

```
library("StatDA")
data("ohorizon")
X <- ohorizon[, c("As","Cd","Co","Cu","Mg","Pb","Zn")]
out <- outCoDa(X)
out

##
##  --------------------
## [1] "104 out of 617 observations are detected as outliers."
##
##  --------------------
```

```
plot(out) # to produce Fig 5.7.
```

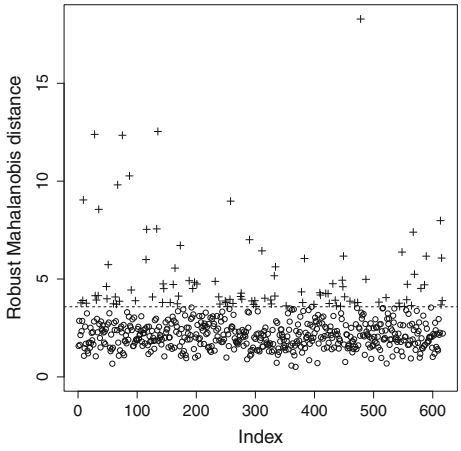


Fig. 5.7 Robust Mahalanobis distances for the selected variables of the O-horizon Kola data

```
res <- mvoutlier.CoDa(X)
plot(res, which = "map", coord = ohorizon[, 2:3], onlyout = FALSE)
pkb(add.plot = TRUE)
```

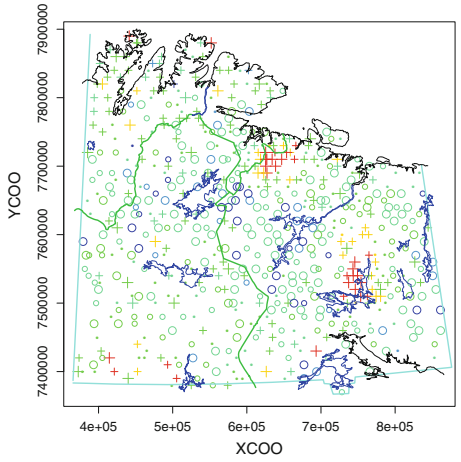


Fig. 5.8 Multivariate outlier plot for the Kola O-horizon data, here represented in the map of the project area

The methods in the package `mvoutlier` (Filzmoser and Gschwandtner 2017) give a more detailed view on the outliers and extend the outlier detection methods available in `robCompositions`. Figure 5.8 shows a map of the project area, together with the locations of the samples, already plotted by the corresponding

```
plot(res, which = "uni", onlyout = FALSE)
```

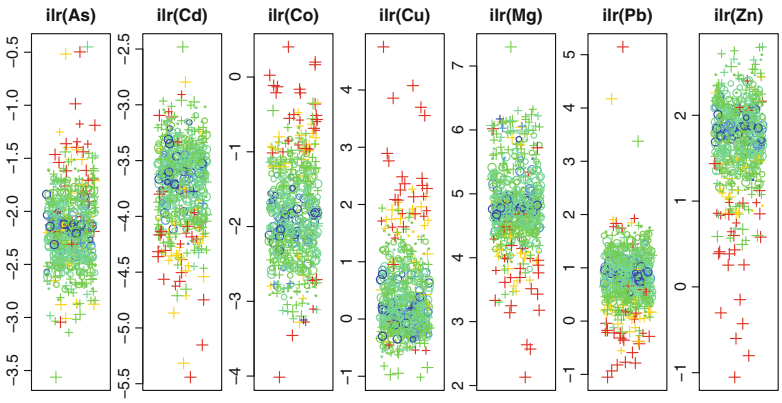


Fig. 5.9 Univariate scatter plots for pivot coordinates of the considered Kola data, with symbols and colors according to outlyingness

```
plot(res, which = "parallel", onlyout = TRUE)
```

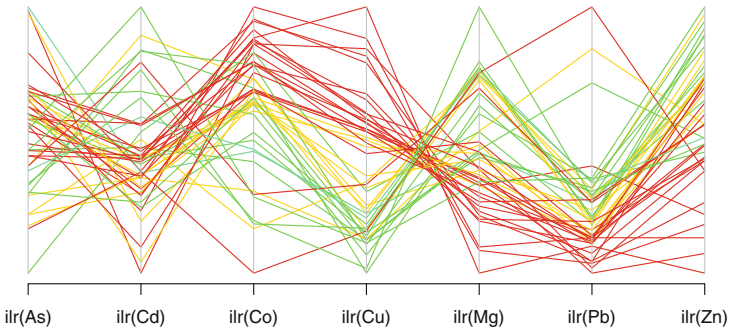


Fig. 5.10 Parallel coordinate plots for pivot coordinates of the considered Kola data, with symbols and colors according to outlyingness. Here only the multivariate outliers are shown

colors and symbols as explained in Fig. 5.5. Most multivariate outliers are located at the Ni-smelters, with high concentration values.

Figures 5.9 and 5.10 show more details about the outliers, and try to support their interpretation by showing pivot coordinates for the single compositional parts. These plots show that most multivariate outliers have high dominance of As, Co and Cu, but low dominance for Mg, Pb, and Zn. This characterizes the emission spectrum of the Ni-smelters. There is one outlier with a very high proportion on Pb. This observation is located on the coast in Norway, and the reason for this exceptional value is unclear (see Filzmoser et al. 2012a).

References

J. Aitchison, *The Statistical Analysis of Compositional Data* (Chapman & Hall, London, 1986). Reprinted in 2003 with additional material by The Blackburn Press

J. Aitchison, G. Mateu-Figueras, K.W. Ng, Characterisation of distributional forms for compositional data and associated distributional tests. Math. Geol. **35**(6), 667–680 (2004)

T.W. Anderson, *An Introduction to Multivariate Statistical Analysis* (Wiley, Chichester, 2003)

V. Barnett, T. Lewis, *Outliers in Statistical Data*, 3rd edn. (Wiley, New York, 1994)

P. Filzmoser, M. Gschwandtner, *mvoutlier: Multivariate Outlier Detection Based on Robust Methods*, 2017. https://CRAN.R-project.org/package=mvoutlier. R package version 2.0.8

P. Filzmoser, K. Hron, Outlier detection for compositional data using robust methods. Math. Geosci. **40**(3), 233–248 (2008)

P. Filzmoser, R.G. Garrett, C. Reimann, Multivariate outlier detection in exploration geochemistry. Comput. Geosci. **31**, 579–587 (2005)

P. Filzmoser, K. Hron, C. Reimann, Interpretation of multivariate outliers for compositional data. Comput. Geosci. **39**, 77–85 (2012a).

P. Filzmoser, K. Hron, M. Templ, Discriminant analysis for compositional data and robust parameter estimation. J. Comput. Stat. **27**(4), 585–604 (2012b)

F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, W. Stahel, *Robust Statistics. The Approach Based on Influence Functions* (Wiley, New York, 1986)

R.A. Johnson, D.W. Wichern, *Applied Multivariate Statistical Analysis*, 6th edn. (Prentice Hall, Upper Saddle River, 2007)

R.A. Maronna, R.H. Zamar, Robust estimation of location and dispersion for high-dimensional datasets. Technometrics **44**(4), 307–317 (2002)

R. Maronna, D. Martin, V. Yohai, *Robust Statistics: Theory and Methods* (Wiley, Chichester, 2006)

G. Mateu-Figueras, V. Pawlowsky-Glahn, A critical approach to probability laws in geochemistry. Math. Geosci. **40**(5), 489–502 (2008)

G.S. Monti, G. Mateu-Figueras, V. Pawlowsky-Glahn, Notes of the scaled Dirichlet distribution, in *Compositional Data Analysis: Theory and Applications*, ed. by V. Pawlowsky-Glahn, A. Buccianti (Wiley, Chichester, 2011), pp. 128–138

V. Pawlowsky-Glahn, J.J. Egozcue, R. Tolosana-Delgado, *Modeling and Analysis of Compositional Data* (Wiley, Chichester, 2015)

C. Reimann, M. Äyräs, V. Chekushin, I. Bogatyrev, R. Boyd, P. de Caritat, R. Dutter, T.E. Finne, J.H. Halleraker, Ø. Jæger, G. Kashulina, O. Letho, H. Niskavaara, V. Pavlov, M.L. Räisänen, T. Strand, T. Volden, *Environmental Geochemical Atlas of the Central Parts of the Barents Region* (Geological Survey of Norway, Trondheim, 1998)

P. Rousseeuw, Multivariate estimation with high breakdown point, in *Mathematical Statistics and Applications*, ed. by W. Grossmann, G. Pflug, I. Vincze, W. Wertz (Reidel Publishing Company, Dordrecht, 1985), pp. 283–297

P.J. Rousseeuw, K. Van Driessen, A fast algorithm for the minimum covariance determinant estimator. Technometrics **41**(3), 212–223 (1999)

J.W. Tukey, *Exploratory Data Analysis* (Addison-Wesley, Reading, 1977)

Chapter 6
Cluster Analysis

Abstract Cluster analysis is an exploratory statistical technique to group observations or variables in data sets. The main goal of cluster analysis is to achieve highly homogeneous clusters, i.e. the observations (or compositional parts—in Q-mode clustering) within a cluster should be very similar to each other. On the other hand, different clusters should be dissimilar, because otherwise they should have been merged into one cluster. With cluster analysis one typically aims to find elliptically shaped partitions in the data, but also more special structures in the data are sometimes of interest. Cluster analysis again needs to be adapted in the context of compositional data. The use of the Aitchison distance or the clustering after representing the data in ilr coordinates is crucial. Moreover, for clustering of compositional parts in Q-mode clustering the variation matrix, either classically or robustly estimated, is taken. For clustering observations (compositions), no particular methodological peculiarities occur; basically, any orthonormal logratio coordinates serve well for this purpose. In this chapter, some of the most popular methods are described in more detail: hierarchical clustering with different linkage methods, the k-means algorithm, model-based clustering as well as fuzzy clustering. Finally, also some cluster validity measures for evaluating the quality of the clustering result are presented.

6.1 Distance Measures and Dissimilarities

The input of a clustering procedure is typically not the raw data set but dissimilarities: a distance matrix when clustering compositions and, for example, the variation matrix when clustering variables.

Let the i-th composition be denoted by $\mathbf{x}_i = (x_{i1}, \ldots, x_{iD})'$ for $i = 1, \ldots, n$, with D the number of parts and n the number of observations of a compositional data set $\mathbf{X}$.

In case of clustering compositions, the dissimilarity/distance matrix $\mathbf{D}$ is of dimension $n \times n$, expressing the distance from each composition to any other composition. For latter use, let $d(i, j)$ be the distance between the i-th and the

P. Filzmoser et al., *Applied Compositional Data Analysis*, Springer Series in Statistics, https://doi.org/10.1007/978-3-319-96422-5_6

j-th composition. It is crucial how to define distances between compositions or their parts.

In statistical analysis of data carrying absolute information, the most popular distance measure for continuous variables is the Euclidean distance, but also many other distance measures are available such as the Manhattan distance. For binary and nominal variables, other distances are typically chosen, e.g. the Jaccard distance. A generalized distance that considers different kinds of variables is the Gower distance. However, for compositional data—typically positive continuous multivariate data—another kind of distance needs to be chosen. The Aitchison distance defined in Eq. (3.9) is appropriate for this purpose because it considers the special nature of compositional data. Alternatively, the compositional data can first be represented in orthonormal coordinates, and then standard distance measures including the Euclidean distance can be used.

First a very small data set is introduced that is suitable to explain how distances are calculated and how observations are combined using an agglomerative clustering approach.

```
data("alcoholreg")
alcoholreg

##                   region year recorded unrecorded
## 1                 Africa 2010      4.2        1.8
## 2               Americas 2010      7.2        1.2
## 3       South-East Asia 2010      1.8        1.6
## 4                 Europe 2010      9.0        1.9
## 5 Eastern Mediterranean 2010      0.3        0.4
## 6       Western Pacific 2010      5.1        1.7
```

These data describe the recorded and unrecorded alcohol consumption per capita (age 15+, in liters of pure alcohol), depending on the WHO region, and in this case for the year 2010. If this two-dimensional data set (recorded versus unrecorded consumption) is plotted, it can be seen that the distance between observations 1 and 6 is the smallest out of all pairwise distances, if the standard Euclidean distance is chosen as a distance measure, see Fig. 6.1.

This changes when considering logratios, and thus the Aitchison distance. Let $d_A(\mathbf{x}_i, \mathbf{x}_j)$ be the Aitchison distance between the i-th and j-th composition, defined in Eq. (3.9). The Aitchison distances between all pairs of observations can be computed with the function `aDist`.

```
x <- alcoholreg[, c("recorded", "unrecorded")]
aDist(x)

##           1         2         3         4         5
## 2 0.6678352
## 3 0.5158449 1.1836801
## 4 0.5006831 0.1671521 1.0165280
## 5 0.8025520 1.4703872 0.2867071 1.3032351
## 6 0.1777061 0.4901291 0.6935510 0.3229770 0.9802581
```

Using the Aitchison distance, the observations 2 and 4 have the smallest distance among all distances, which was not to be expected from Fig. 6.1. In the next

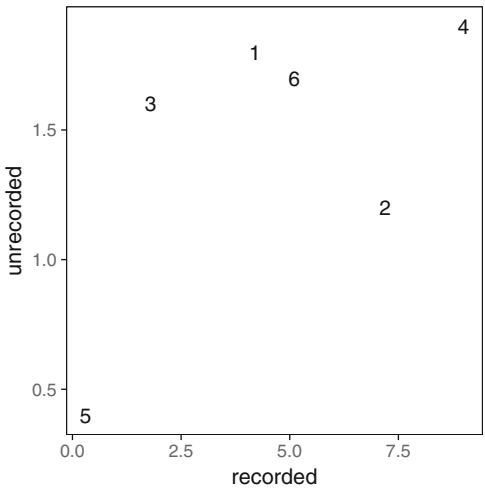


Fig. 6.1 Recorded and unrecorded alcohol consumption (in liters of pure alcohol) in WHO regions

step these two observations are joined and new distances are calculated. Here the geometric mean is used to join the observations, and the distances are computed as before. However, several other choices exist to compute distances between joined observations or clusters, which will be defined later on.

```
z <- apply(x[c(2,4), ], 2, gm)
y <- rbind(z, x[c(1,3,5:6), ])
rownames(y) <- c("2-4", "1", "3", "5", "6")
aDist(y)

##          2-4         1         3         5
## 1 0.5842592
## 3 1.1001040 0.5158449
## 5 1.3868112 0.8025520 0.2867071
## 6 0.4065530 0.1777061 0.6935510 0.9802581
```

In a further step, the observations 1 and 6 could be merged, since they have the smallest Aitchison distance. Proceeding in this way leads to a whole tree of clusters, which is the basic idea of hierarchical clustering. Since the algorithm started to build this hierarchy with single observations, the clusters get larger and larger, until all observations are finally merged into a single big cluster. This procedure is denoted as *agglomerative*; the reverse procedure would be called *divisive*, but it is not very commonly applied. This topic continues in the next section.

Distances can also be computed between variables (Q-mode clustering). This is useful if one is interested in clustering compositional parts rather than observations. One way to define distances between parts is to calculate a measure of association between the parts. Note that when absolute information in data is under consideration this could be the Pearson correlation coefficient (see Sect. 8.1). Typically such measures of association are normalized and a high value means strong relationship while a value around 0 means no dependency between the variables.

In the context of compositional data, the variation between the parts is suitable to express the association between the parts. Low values express a high association, and all ratios in a sample are nearly perfectly proportional to each other, while large values express that the ratios between the parts are very different from each other. The dissimilarities used as an input for a clustering method (typically for agglomerative clustering methods) are given by t_{jk}, which defines the variation matrix elements between the j-th and the k-th part, see Eq. (4.2).

6.2 Hierarchical Clustering Methods

The result of a hierarchical clustering procedure is composed of a sequence of clustering partitions. This sequence can visually be displayed by a clustering tree which is called *dendrogram*.

6.2.1 Agglomerative Clustering Algorithms

At the beginning, each object forms an own class, leading to n different clusters. At each step of the algorithm, the number of clusters is reduced by one, where the most similar classes are combined. The "similarity" of the combined pair can be measured, and a "height" is associated with this newly formed class. At the end of the process there is only one single cluster left.

Consider the classes C_i and C_j, which consist of indexes representing the observations of the i-th and j-th cluster, respectively. The similarity between these classes is expressed by a distance $d(C_i, C_j)$. If the classes C_i and C_j are combined (linked), a general scheme of evaluating the similarity between $C_i \cup C_j$ and some other class C_k can be defined as (Lance and Williams 1966)

$$\begin{aligned} d(C_i \cup C_j, C_k) = \alpha_i d(C_i, C_k) + \alpha_j d(C_j, C_k) + \beta d(C_i, C_j) \\ + \gamma |d(C_i, C_k) - d(C_j, C_k)|, \end{aligned} \tag{6.1}$$

where the parameters α_i, α_j, β, and γ are in general real numbers. Table 6.1 presents the most commonly used parameters. The methods for three of these choices of the parameters will be discussed below in detail.

In the first step of the algorithm, the cluster C_i consists only of object i and cluster C_j only of object j. The distances $d(C_i, C_j)$ for single element classes can be selected using the so-called linkage criteria (see also Table 6.1).

Table 6.1 Hierarchical clustering linkage strategies

Clustering criterion	α_i, α_j	β	γ
Single linkage	$\frac{1}{2}$	0	$-\frac{1}{2}$
Complete linkage	$\frac{1}{2}$	0	$\frac{1}{2}$
Average linkage	$\frac{1}{2}$	0	0
Centroid linkage	$\frac{n_i}{n_i+n_j}$	$\frac{n_i n_j}{(n_i+n_j)^2}$	0
Ward's method	$\frac{n_i+n_k}{n_i+n_j+n_k}$	$\frac{-n_k}{n_i+n_j+n_k}$	0

Parameters with respect to Eq. (6.1)
n_l is the number of observations in class C_l ($l = i, j, k$)

6.2.1.1 Single Linkage

In the first step, the two closest objects are combined. Hence, the combined objects generate a new group, say $C_i \cup C_j$. Equation (6.1) is used with the coefficients for single linkage (Table 6.1). Then this formula can be simplified to

$$d(C_i \cup C_j, C_k) = \min\{d(C_i, C_k), d(C_j, C_k)\}. \tag{6.2}$$

Thus, the minimum distance between the observations of a combined cluster to another object C_k is chosen. This can be further generalized. In case of already three formed clusters, the minimal distance between them (single linkage) is indicated with a black thick(er) solid line in Fig. 6.2, while all other distances from a cluster to another are in grey. Note that these data are already presented in orthonormal coordinates and thus $d_A(\mathbf{x}, \mathbf{y}) = d(\text{ilr}(\mathbf{x}), \text{ilr}(\mathbf{y}))$ applies, the latter being the Euclidean distance.

Single linkage tends to be unbalanced in the sense that big clusters are quickly combined. This procedure tends to produce many small groups and few large groups. Single linkage is also suitable to detect outliers.

As a simple illustration, the alcohol consumption data set, introduced at the beginning of this section, is used. The unrecorded and recorded alcohol consumptions are first expressed by an ilr coordinate with (3.20), then Euclidean distances (being here simply absolute differences between the coordinate values) are computed, and finally single linkage clustering is performed. The result is plotted and presented as a dendrogram in Fig. 6.3, where the "height" on the vertical axis corresponds to the level where the clusters are merged.

```
res <- hclust(dist(pivotCoord(x)), method = "single")
plot(res) # produces Figure 6.3
```

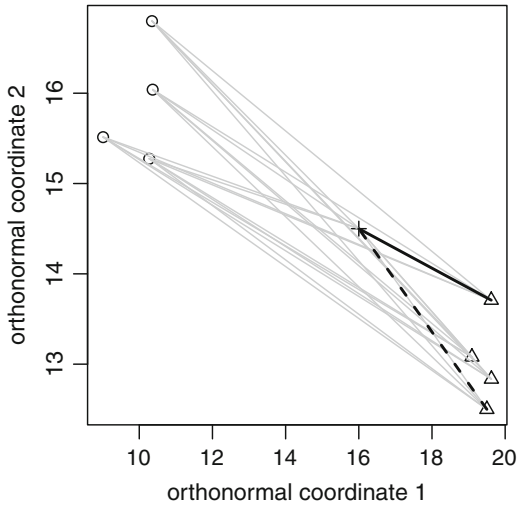


Fig. 6.2 Single linkage (solid black line) and complete linkage (dashed line) for three given clusters defined by the different symbols

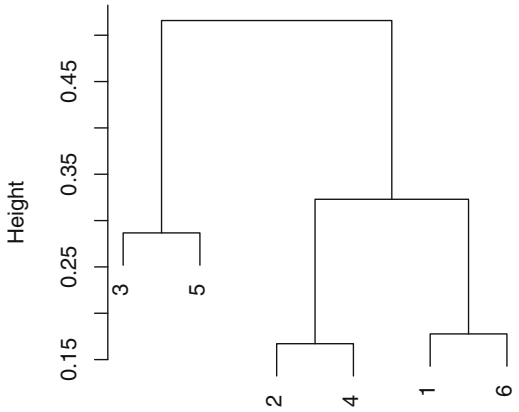


Fig. 6.3 Dendrogram from single linkage clustering of the recorded and unrecorded alcohol consumption data

6.2.1.2 Complete Linkage

Complete linkage clustering proceeds in much the same manner as single linkage clustering, with one important exception: not the smallest distances are considered but the biggest ones. So, the criterion for merging clusters is

$$d(C_i \cup C_j, C_k) = \max\{d(C_i, C_k), d(C_j, C_k)\}. \tag{6.3}$$

The complete linkage algorithm tends to produce a balanced dendrogram. This procedure is also illustrated in Fig. 6.2 (black dashed line). The maximum distance from the observations from one cluster to any other observation from any other cluster is considered, where the minimum over these distances is selected for the complete linkage criterion.

6.2.1.3 Average Linkage

The average linkage criterion was designed to take a middle road between single linkage and complete linkage and by far surpasses them in the sensitivity with respect to single objects. This method treats the distance between two clusters as the average distance between all pairs of objects where one member of a pair belongs to each cluster. If $\mathbf{z}_{(ij)}$ denotes an object (composition in orthonormal coordinates) from $C_i \cup C_j$, and $\mathbf{z}_k$ an object from C_k, then the average distance can be expressed as

$$d(C_i \cup C_j, C_k) = \frac{\sum_{(ij)} \sum_k d(\mathbf{z}_{(ij)}, \mathbf{z}_k)}{n_{(ij)} n_k}, \tag{6.4}$$

where $n_{(ij)}$ is the number of objects in $C_i \cup C_j$.

6.2.1.4 Ward's Method

Ward clustering can be implemented in a similar way as the methods mentioned before, but with the appropriate coefficients listed in Table 6.1. This can also be seen as a method, where at each step in the analysis, the union of every possible cluster pair is considered and the two clusters whose fusion results in a minimum increase of *information loss* are combined. In this context the error sum of squares (ESS) is often used; it can be expressed as sum of squared differences between the observed values and their predictions from a model, mostly represented by the sample mean of the samples within a given cluster. Accordingly, the criterion minimizes the total within-cluster variance.

As a simple example, consider univariate data with 10 observations (2, 6, 5, 2, 2, 2, 2, 0, 0, 0) with arithmetic mean 2.5. In this case, the original two-part compositions are already expressed by one coordinate. For ESS the sum of squared differences to the arithmetic mean is taken,

$ESS_{unclustered}: \quad (2-2.5)^2 + (6-2.5)^2 + \ldots + (0-2.5)^2 = 50.5.$

Now let the following four groups be formed, {0, 0, 0},{2, 2, 2, 2},{5},{6, 6}. Then $ESS_{4\,clusters} = ESS_1 + ESS_2 + ESS_3 + ESS_4 = 0$, i.e. no information loss occurred by joining them into these four clusters.

6.2.2 Tree Cutting

In hierarchical clustering, the clusters are defined as branches of a cluster tree. The partition into clusters itself is done by cutting the corresponding tree, represented as dendrogram. One can cut a tree into several groups either by specifying the desired number of groups or the cut height. Both variants correspond to a constant height cut-off value in the dendrogram. Naturally, this method exhibits

suboptimal performance on complicated dendrograms. Note that the R package **dynamicTreeCut** can be used for dynamic branch cutting depending on the shape of the dendrogram to better identify nested clusters. However, the authors' experience is that these methods often do not give good results or need a lot of work for parameter adjustment. Therefore, dynamic tree cutting is not further considered in the following.

6.3 Partitioning Methods

The probably most famous algorithm for clustering observations into groups is the k-means algorithm. It turns out that this algorithm is just a variant of the EM algorithm.

Given a compositional data set $\mathbf{X}$ with n objects, characterized by D parts. The aim is to partition the observations into n_c clusters $\{C_1, C_2, \ldots, C_{n_c}\}$ such that cluster C_k has $n_{(k)}$ members and each observation is assigned to one distinct cluster.

For simplicity the method is not described based on the original compositional data, but already for the data expressed in (any) ilr coordinates. Thus, denote the corresponding observations, expressed in $D-1$ coordinates, by $\mathbf{z}_1, \ldots, \mathbf{z}_n$, and the resulting data matrix with n rows and $D-1$ columns by $\mathbf{Z}$.

The mean vector (center, prototype), $\mathbf{v}_k$, of a cluster C_k is defined as the centroid of the cluster, and the components of the mean vector can be calculated by

$$\mathbf{v}_k (\in \mathbb{R}^{D-1}) = \left(\frac{1}{n_{(k)}} \sum_{i=1}^{n_{(k)}} z_{i1}^{(k)}, \ldots, \frac{1}{n_{(k)}} \sum_{i=1}^{n_{(k)}} z_{i,D-1}^{(k)} \right)', \tag{6.5}$$

where $\mathbf{z}_i^{(k)} = (z_{i1}^{(k)}, \ldots, z_{i,D-1}^{(k)})'$ is the i-th observation belonging to cluster C_k. For each cluster $C_1, \ldots, C_{n_c}$ the corresponding cluster means $\mathbf{v}_1, \ldots, \mathbf{v}_{n_c}$ are calculated.

At the beginning, the number of clusters n_c of the output partition needs to be determined. Starting from a given initial location of the n_c cluster centroids, the algorithm uses the data points to iteratively relocate the centroids and reallocate points to the closest centroid. The process is composed of the following steps:

1. Select an initial partition with n_c clusters.
2. E-step: (re)compute the cluster centers using the current cluster memberships.
3. M-step: assign each object to the closest cluster center $\rightarrow$ new memberships.
4. Go to step 2 until the cluster memberships and thus the cluster centroids do not change beyond a specified bound.

Accordingly, k-means clustering optimizes the objective function

$$J(\mathbf{Z}, \mathbf{V}, \mathbf{U}) = \sum_{k=1}^{n_c} \sum_{i=1}^{n} u_{ik} d^2(\mathbf{z}_i, \mathbf{v}_k), \tag{6.6}$$

where $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_{n_c})$ is the matrix of cluster centers (prototypes) of dimension $(D-1) \times n_c$ and $\mathbf{U} = (u_{ik})$ is an $n \times n_c$ matrix with the membership coefficients u_{ik} for observation $\mathbf{z}_i$ to a cluster C_k. The Euclidean distance d measures the distance between the observations and the cluster centers.

The k-means algorithm can be implemented as follows. Fix n_c, $2 \leq n_c < n$, and choose the termination tolerance $\delta > 0$, e.g., 0.001. Initialize $\mathbf{U}^{(0)}$, usually randomly.

REPEAT for $r = 1, 2, \ldots$

1. E-step: Calculate the centers of the clusters:

$$\mathbf{v}_k^{(r)} = \frac{\sum_{i=1}^{n} u_{ik}^{(r-1)} \cdot \mathbf{z}_i}{\sum_{i=1}^{n} u_{ik}^{(r-1)}}, \quad 1 \leq k \leq n_c \tag{6.7}$$

2. M-step: Update $\mathbf{U}^{(r)}$: Reallocate cluster memberships:

$$u_{ij}^{(r)} = 1 \text{ if } d(\mathbf{z}_i, \mathbf{v}_j^{(r)}) = \min_{1 \leq l \leq n_c} d(\mathbf{z}_i, \mathbf{v}_l^{(r)}), \text{ or } u_{ij}^{(r)} = 0 \text{ otherwise}$$

UNTIL the Frobenius matrix norm $\|\mathbf{U}^{(r)} - \mathbf{U}^{(r-1)}\|_F < \delta$, where

$$\|\mathbf{U}^{(r)} - \mathbf{U}^{(r-1)}\|_F = \sqrt{\sum_{i=1}^{n} \sum_{k=1}^{n_c} \left(u_{ik}^{(r)} - u_{ik}^{(r-1)}\right)^2}.$$

It is easy to see that in k-means clustering the E-step is the fitting step and the M-step is the assignment step. Iterating between the E- and M-step improves the solution, which means that $J(\mathbf{Z}, \mathbf{V}, \mathbf{U})$ gets smaller in each iteration. The procedure is stopped when the cluster assignments stabilize.

Consider for illustration an artificial compositional data set, which is already expressed in ilr coordinates. The original data consist of three parts, and thus the coordinate representation is two-dimensional, see Fig. 6.4 (upper left).

In the following the results of the algorithm after iteration 1, 2, and after convergence are plotted. Instead of the presented simplified implementation of the k-means algorithm, the default k-means implementation of R is chosen. There exist some variants of k-means, where the algorithm of *MacQueen* is chosen, but only for reasons of exploring the algorithm (the default method, *Hartigan-Wong* is converging too fast to show the steps of the algorithm). Note that the k-means algorithm starts with randomly chosen cluster centers. Thus it is necessary to set a *seed* to ensure the same starts in each call of the k-means algorithm.

```
set.seed(123456)
cl1 <- kmeans(Z, centers = 4, iter.max = 1, algorithm = "MacQueen")
set.seed(123456)
cl2 <- kmeans(Z, centers = 4, iter.max = 2, algorithm = "MacQueen")
set.seed(123456)
cl3 <- kmeans(Z, centers = 4, iter.max = 3, algorithm = "MacQueen")
set.seed(123456)
cl4 <- kmeans(Z, centers = 4, algorithm = "MacQueen")
```

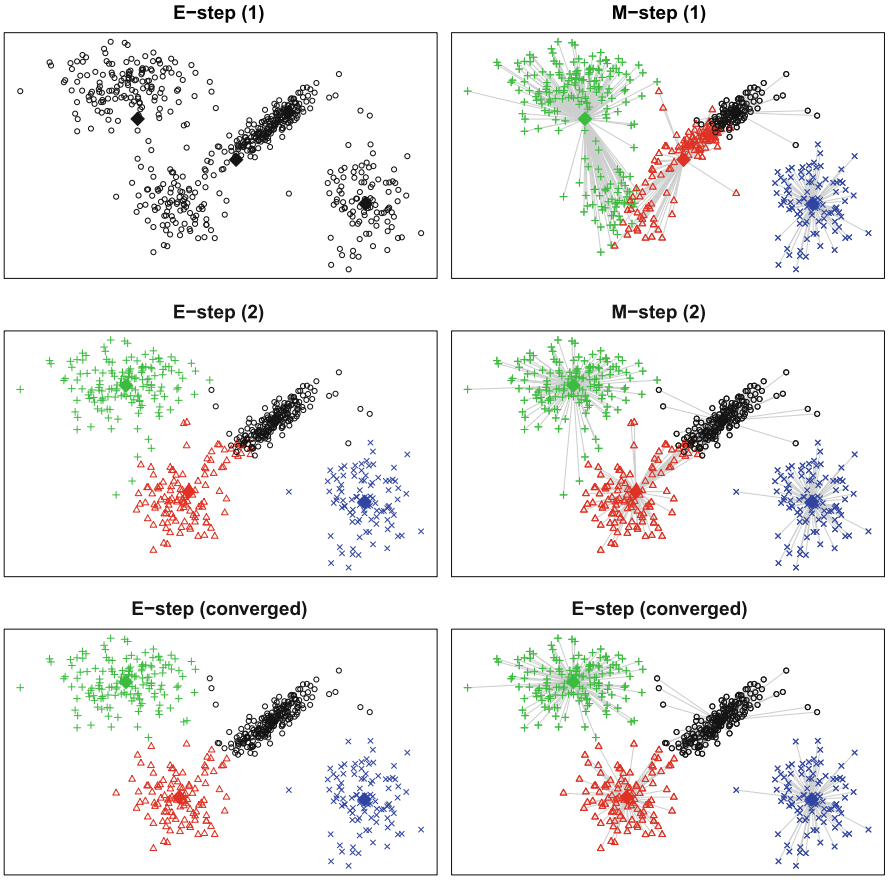


Fig. 6.4 Solutions of the k-means algorithm. Top left: initial centers, the solution of the E-step in iteration 1. Top right: first assignment of points to estimated centers, the solution of the M-step in iteration 1. Middle left: new centers at iteration 2. Middle right: new assignment. Bottom left: final solution of the centers. Bottom right: final assignment of observations to cluster centers

Then the results after the E-step and after the M-step for the first two iterations, but also for the final solution are plotted. This can be easily done by accessing the cluster centers from the k-means results, e.g. for the first solution after one iteration with:

```
cl1$centers

##          Z1          Z2
## 1  4.787137  4.65547187
## 2  2.555571  2.20578465
## 3 -1.590451  4.32789868
## 4  7.997304 -0.08258293
```

The calculated centers (E-step) and the allocation of the observation to their nearest cluster (M-step) are shown in detail in Fig. 6.4.

Note that the k-means algorithm only takes the centers into account and works with a distance function to calculate the distance from the observations to the cluster centers. A limitation of k-means clustering is that the resulting clusters tend to be spherically symmetric. An alternative approach is to incorporate also the shape of the clusters. This is implemented in the model-based clustering framework (Fraley and Raftery 2002). The model-based procedures usually give better clustering results (Templ et al. 2008) but they are computationally more complex since in each E-step also the covariance of each cluster needs to be estimated.

6.4 Model-Based Clustering

As the name indicates, model-based clustering makes use of a statistical model for the shape of the clusters. The standard "model" is multivariate normal distribution, i.e., the distribution of a compositional cluster is assumed to have the density of a multivariate normal distribution on the simplex (5.2), with a specific location and covariance. Considering this, it is clear that compositional data first need to be expressed in ilr coordinates before model-based clustering can be applied.

A detailed description of model-based clustering can be found in Fraley and Raftery (2002), and in many other sources of these authors. Here the focus is rather on practical aspects. Assume that the data consist of n_c clusters, generated by multivariate normal densities with expectation $\boldsymbol{\mu}_j$ and covariance $\boldsymbol{\Sigma}_j$, for $j = 1, \ldots, n_c$. Further, the class probabilities are given by the so-called mixing coefficients $\pi_1, \ldots, \pi_{n_c}$, where $\pi_1 + \ldots + \pi_{n_c} = 1$. All these parameters are unknown, and they are estimated using the EM algorithm. In case of D-part compositions, the covariance matrices of the data expressed in coordinates are of dimension $(D-1) \times (D-1)$. Thus, if D gets larger, many parameters need to be estimated from the available data, which can lead to instability. For this reason, the cluster "models" can be simplified, by imposing restrictions on the cluster covariance structures.

The simplest possibility for such restrictions is $\boldsymbol{\Sigma}_j = \sigma^2 \mathbf{I}$, for $j = 1, \ldots, n_c$, where $\mathbf{I}$ is the identity matrix and σ^2 is a parameter for the variance. This would imply that all clusters are spherical, with the same radius. The estimation of the covariances thus reduces to estimating only one parameter, the variance σ^2. A less restricted covariance structure is $\boldsymbol{\Sigma}_j = \sigma_j^2 \mathbf{I}$, for $j = 1, \ldots, n_c$. In this case, the clusters are still spherical, but their size can be different according to their variance σ_j^2, which needs to be estimated. Figure 6.5 illustrates different covariance structures.

In R, the package **mclust** can be used to apply model-based clustering by finite Gaussian mixture modelling fitted via an EM-algorithm. An optimal model (according to a BIC criterion (Schwarz 1978)) can be chosen using the function `Mclust`. Figure 6.6 shows the outcome of using `Mclust` for the data set from

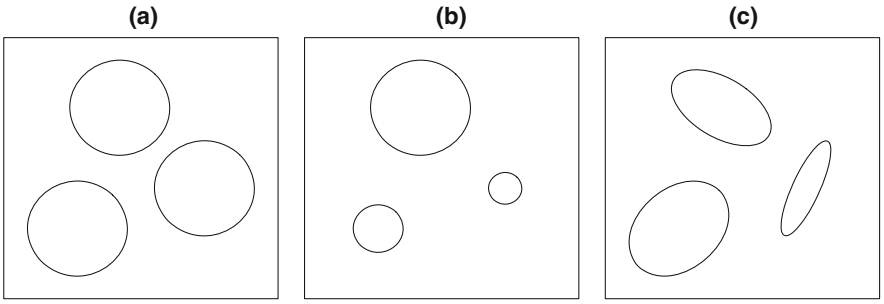


Fig. 6.5 Different covariances for three clusters: (**a**) $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_3 = \sigma^2\mathbf{I}$; (**b**) $\boldsymbol{\Sigma}_j = \sigma_j^2\mathbf{I}$, for $j = 1, 2, 3$; (**c**) all $\boldsymbol{\Sigma}_j$ different and of no special structure

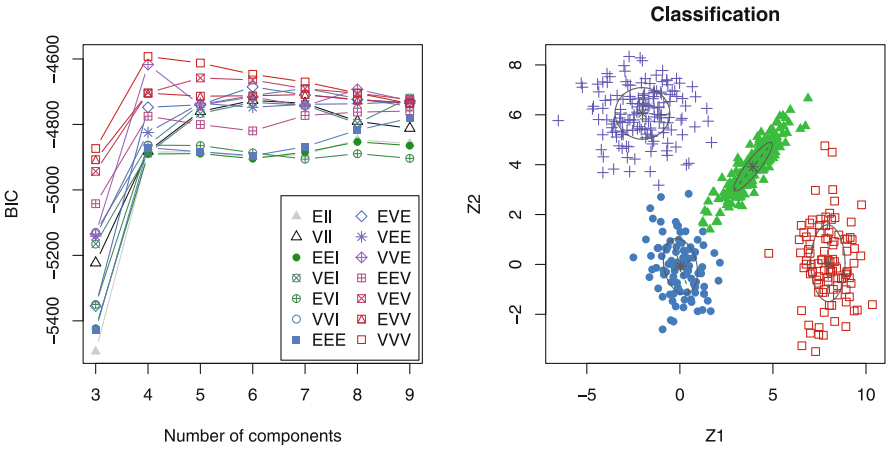


Fig. 6.6 Result of model-based clustering: Left: the choice of the optimal cluster model based on the BIC criterion; right: the resulting cluster assignments

the previous section, compare Fig. 6.4. The left plot shows the BIC values (vertical axis) for different numbers of clusters (horizontal axis). Different cluster models are used, corresponding to the structure of the covariance matrices. The maximum BIC value points at the optimal model, which is a model with four clusters, and covariance structure "VVV," meaning that all covariances are different from each other. The right plot presents the assignments of the observations to the different clusters, and also the shapes of the covariance structures together with the estimated group centers. The mixing coefficients correspond to the proportion of observations in the different clusters.

```
library("mclust")
res <- Mclust(Z, G = 3:9, verbose = FALSE)  # 3 to 9 mixture components
plot(res, what = "BIC")
plot(res, what = "classification")
```

There is an interesting difference to the results from k-means clustering: While for k-means some observations from other clusters have been assigned to the long-shaped cluster, see Fig. 6.4 (bottom right), this is not the case for model-based clustering. k-means solutions are typically spherically symmetric, while model-based clustering is more flexible and not limited to spherical symmetry.

6.5 Fuzzy Clustering

Fuzzy clustering has been described in various papers and monographs (see, e.g., Kaufman and Rousseeuw 1990), and therefore only the essential ideas are outlined. The basic difference to partitioning methods is that an observation is not assigned to only one cluster, but there is a proportional assignment to all clusters. Accordingly, a membership coefficient u_{ik} is introduced, assigning the i-th observation to the k-th cluster ($i = 1, \ldots, n; k = 1, \ldots, n_c$), with $u_{ik} \geq 0$ and $u_{i1} + \ldots + u_{in_c} = 1$, for all i.

For a fixed number of clusters n_c, the fuzzy cluster solution can be found by minimizing the objective function (6.6). For this purpose, the compositions first need to be expressed in orthonormal coordinates. Note that in case of k-means clustering, the values of u_{ik} in (6.6) were restricted to 0 and 1 ("hard" clustering), while in case of fuzzy clustering they are in the whole interval [0, 1].

An implementation of fuzzy clustering is available in the R package **e1071** as function `cmeans`. The four-cluster solution leads to the estimated membership coefficients which are shown in grey scale in Fig. 6.7. Note that, similar to k-means, also this algorithm works with a random initialization, and thus the results could differ when computed again. Further, note that as a result of minimizing the objective function (6.6), also here the procedure ends up with clusters which tend to be spherically shaped.

```
library("e1071")
groups <- 4
res <- cmeans(Z, groups)
for(i in seq_along(1:groups)){
  plot(Z, col = gray(1 - res$membership[, i])) # produces Fig. 6.7
}
```

6.6 Clustering Parts: Q-Mode Clustering

While previously the main interest was in grouping the observations (R-mode clustering), now the aim is to group the variables or compositional parts (Q-mode clustering). The key ingredients for cluster analysis are the distances or dissimilarities, and as already mentioned in Sect. 6.1, an appropriate way to measure the relatedness between parts is the variation matrix, see Eq. (4.2). The basic idea is described in more detail in the following.

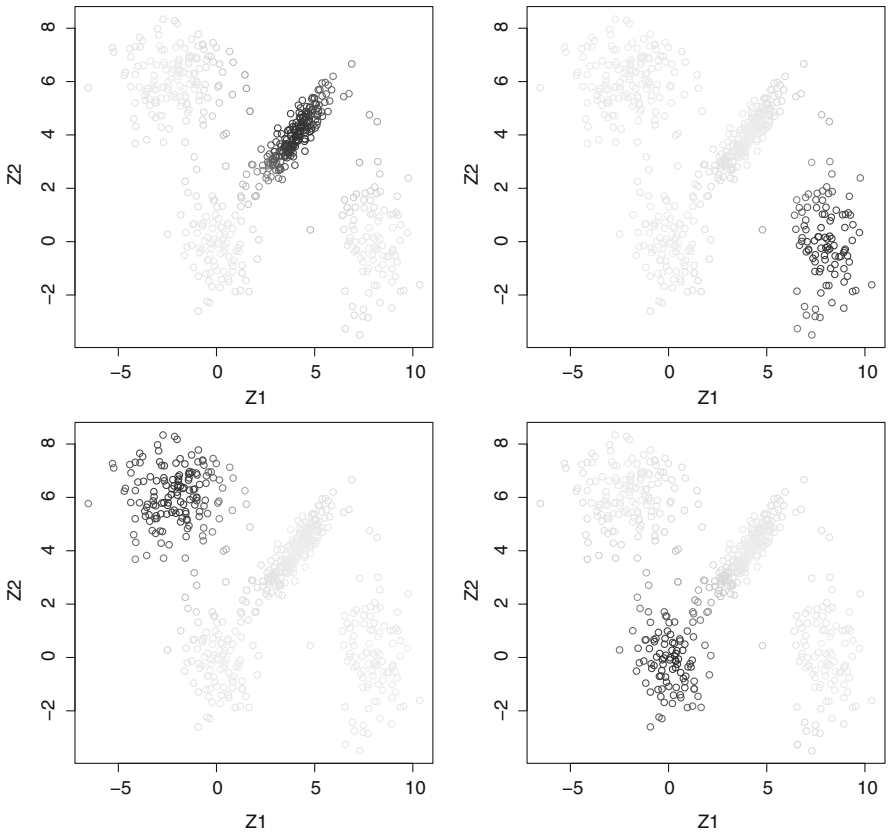


Fig. 6.7 Result of fuzzy clustering: The four plots show the resulting membership coefficients for each of the four clusters in grey scale; dark means high value, light means low value

Consider a compositional data matrix $\mathbf{X}$ with D parts and n observations $\mathbf{x}_i = (x_{i1}, \ldots, x_{iD})'$, for $i = 1, \ldots, n$. Remind that the variation matrix $\mathbf{T}$ is of dimension $D \times D$, and its elements t_{jk} are defined as

$$t_{jk} = \text{var}\left[\ln\left(\frac{x_{1j}}{x_{1k}}\right), \ln\left(\frac{x_{2j}}{x_{2k}}\right), \ldots, \ln\left(\frac{x_{nj}}{x_{nk}}\right)\right], \tag{6.8}$$

where $j, k = 1, \ldots, D$, and "var" denotes the variance. The elements of the variation matrix report the variability of the logratio of a pair of parts (pairwise logratio). The smaller the value of t_{jk} is, the more the logratio tends to be constant. In this case, the corresponding parts can be considered as being proportional. The variation matrix is symmetric and the diagonal elements are zero, and thus the elements of the matrix can be directly used as dissimilarity measure for clustering (van den Boogaart and Tolosana-Delgado 2013; McKinley et al. 2016). Note,

however, that the variation matrix does not possess the properties of a distance matrix, see Fačevicová et al. (2016).

An open issue is which estimator for the variance "var" should be used. The standard choice would be the sample variance, but also more robust variance estimators could be considered, like that resulting from the MCD estimator through the relation (5.8). The latter is the default in the function `variation` of the package **robCompositions**, while the sample variance is only used with the option `robust=FALSE`. Once this dissimilarity measure is defined and computed, it is straightforward to apply a clustering method, like hierarchical clustering.

As an illustration the data set `expendituresEU` from **robCompositions** is used, reporting the average expenditures in different countries of the European Union on various commodity groups. Both types of estimating the variation matrix are compared in hierarchical clustering, using the Ward's method (of course, also any other linkage method could be applied).

```
data("expendituresEU")
v.cla <- as.dist(variation(expendituresEU, robust = FALSE))
v.rob <- as.dist(variation(expendituresEU))
plot(hclust(v.cla, method = "ward.D"))   # produces Fig. 6.8 left
plot(hclust(v.rob, method = "ward.D"))   # produces Fig. 6.8 right
```

The resulting dendrograms are presented in Fig. 6.8. These solutions differ quite a lot, and it seems that the variable *education* contains outliers, which have been downweighted in the robust version. Overall, the solution for the robust version seems to be more logical, but more detailed diagnostics is recommended before drawing more general conclusions.

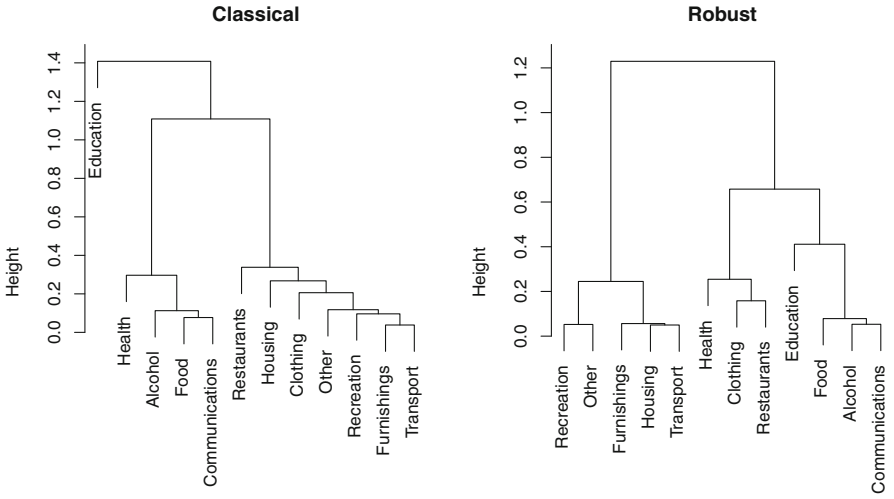


Fig. 6.8 Resulting dendrograms of Q-mode clustering for the expenditures data. Left is the solution for the classical estimation of the variation matrix elements, right the dendrogram for robust estimation

6.7 Evaluation

The difficulty with cluster analysis is not only that there are various different procedures how to perform the clustering (hierarchical, partitioning, fuzzy clustering, etc.), but that for each procedure there exist several different algorithms. Moreover, several cluster algorithms require input parameters, like the number of clusters, and depending on this choice, the results can differ quite a lot. Consequently, there is a need for comparing the outcomes, and this is done by using the so-called *cluster validity measures*.

The main goal of cluster analysis is to achieve highly homogeneous clusters, i.e. the observations (or variables, in Q-mode clustering) within a cluster should be very similar to each other. On the other hand, different clusters should be dissimilar, because otherwise they should have been merged into one cluster. In other words, heterogeneity between different clusters should be achieved. Heterogeneity can be measured by

$$B_{n_c} = \sum_{k=1}^{n_c} \|\mathbf{v}_k - \bar{\mathbf{v}}\|^2, \tag{6.9}$$

where $\|\cdot\|$ denotes the Euclidean norm, $\mathbf{v}_k$ is the k-th cluster center ($k = 1, \ldots, n_c$), and

$$\bar{\mathbf{v}} = \frac{1}{n_c} \sum_{k=1}^{n_c} \mathbf{v}_k$$

is the overall mean of the cluster centers. Note that the cluster centers have to be computed from the observations expressed in orthonormal coordinates, see Eq. (6.5). This term is also called the *between cluster sum of squares*. Homogeneity within the clusters can be defined by

$$W_{n_c} = \sum_{k=1}^{n_c} \sum_{i \in C_k} \|\mathbf{z}_i - \mathbf{v}_k\|^2, \tag{6.10}$$

where $\mathbf{z}_i$, $i = 1, \ldots, n$, are the observations expressed in coordinates. This term is called the *within cluster sum of squares*, since it considers squared Euclidean distances from the observations to their own cluster center.

While B_{n_c} should be large, W_{n_c} should be small. However, both measures depend on the number n_c of clusters, and thus this needs to be considered in a validity measure. Two prominent measures are the *Calinski-Harabasz index*

$$\mathrm{CH}_{n_c} = \frac{B_{n_c}/(n_c - 1)}{W_{n_c}/(n - n_c)}$$

and the *Hartigan index*

$$\mathrm{H}_{n_c} = \ln \frac{B_{n_c}}{W_{n_c}}.$$

Practically, one considers a range of values for the possible number of clusters and computes the validity measure(s) for each cluster solution. The largest value of the index determines the optimal number of clusters.

Another prominent validity measure is the *average silhouette width* (Kaufman and Rousseeuw 1990). Before computing this value, some definitions have to be provided first. The average dissimilarity of an observation $\mathbf{z}_i$ belonging to cluster C_k to all other observations of the same cluster is given by

$$d_{i,C_k} = \frac{1}{n_{(k)} - 1} \sum_{i,j \in C_k, i \neq j} d^2(\mathbf{z}_i, \mathbf{z}_j),$$

where $n_{(k)}$ is the number of observations in cluster C_k. The average dissimilarity of $\mathbf{z}_i$ to observations from another cluster C_l is given by

$$d_{i,C_l} = \frac{1}{n_{(l)}} \sum_{j \in C_l} d^2(\mathbf{z}_i, \mathbf{z}_j).$$

The smallest of these values is

$$d_{i,C} = \min_l d_{i,C_l},$$

and it corresponds to the smallest dissimilarity of the i-th observation to its "closest" cluster. The *silhouette value* is defined as

$$s_i = \frac{d_{i,C} - d_{i,C_k}}{\max(d_{i,C_k}, d_{i,C})}.$$

The values of s_i are within the interval $[-1, 1]$. If the value of s_i is close to 1, the observation is well classified, a value of zero means that the observation is in between two clusters, and a value of -1 refers to a poor classification. Observations with negative silhouette values are probably assigned to a wrong cluster. The average silhouette width is

$$\frac{1}{n} \sum_{i=1}^{n} s_i,$$

and the higher this value, the better the classification.

6.8 Examples

The data from the Kola project (Reimann et al. 1998) are again considered, but this time the interest is in the moss data set, which is available in the R package **StatDA** as `data(moss)`. Only a selection of the available variables is used: the concentration of the elements Cu, Ni, and Co, which are severely increased by the emissions of the smelters in Russia; the elements Al, Fe, V, because their concentrations will be strongly influenced by the input of dust to the moss; K, S, P, which might characterize biological processes in the mosses.

```
library("StatDA")
sel <- c("Cu","Ni","Co","Al","Fe","V","K","S","P")
X <- moss[, sel]
dim(X)

## [1] 598   9
```

The selected data set has 594 observations and 9 variables. Q-mode clustering is applied to see if indeed the selected variable groups form separate clusters. The dendrogram in Fig. 6.9 confirms this. With these clusters of variables, the practitioner may be able to identify processes in the region which are characterized by the groups of variables. In our case, the meaning of the groups is clear since the variables have just been selected according to some characteristics. Here, the function `clustCoDa_qmode` from the package **robCompositions** is used. The default method is Ward clustering, which is applied to the variation matrix (robustly estimated).

```
library("robCompositions")
cl <- clustCoDa_qmode(X)
plot(cl)
```

In a next step the observations are clustered. The number of clusters is unclear, and therefore hierarchical clustering is used first, with the complete linkage method. This method is applied to the Euclidean distances computed from the ilr coordinates.

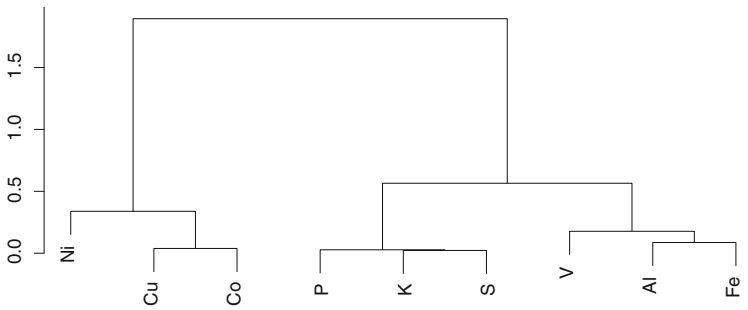


Fig. 6.9 Q-mode clustering result for the selected Kola moss data

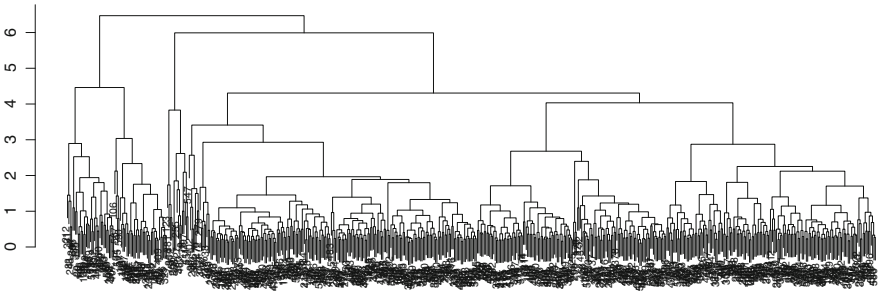


Fig. 6.10 Hierarchical clustering with complete linkage for the selected Kola moss data

Figure 6.10 shows the resulting dendrogram. The observation indexes are overplotted on the bottom of the dendrogram, but the main interest is in identifying a useful number of clusters.

```
X.ilr <- pivotCoord(X)                  # coordinates
res.hclust <- hclust(dist(X.ilr))       # dist computes Euclidean distances
plot(res.hclust)
```

It can be clearly seen that the dendrogram of Fig. 6.10 indicates grouping structure, and the results are inspected now by using seven clusters. This can be done by cutting the dendrogram at a level (height) which leads to seven clusters, i.e., by applying the function `cutree()` to the result object, with the desired number of clusters. Figure 6.11 (upper left) shows the results in form of the map of the Kola region, where the colors represent the different clusters for the coordinates of the sample locations.

Figure 6.11 also compares with the results from other cluster methods: with *k*-means clustering (upper right), model-based clustering (lower left), and fuzzy clustering (lower right). For better comparability, the number of clusters was always set to seven. In case of fuzzy clustering, a hard assignment of the observations to the clusters has been carried out, according to the largest value of the membership coefficients for each observation. The following code is used to make the comparison.

```
# to reproduce exactly the same result
set.seed(123)
# store the cluster memberships
moss$hclust <- cutree(res.hclust, 7)
moss$kmeans <- kmeans(X.ilr, 7)$cluster
moss$mclust <- Mclust(X.ilr, 7, verbose = FALSE)$class
moss$fuzzy <- cmeans(X.ilr, 7)$cluster
# make data tidy
dfl <- reshape2::melt(moss, id.vars = 2:3, measure.vars = 35:38)
dfl$value <- factor(dfl$value)
colnames(dfl)[4] <- "cluster"
```

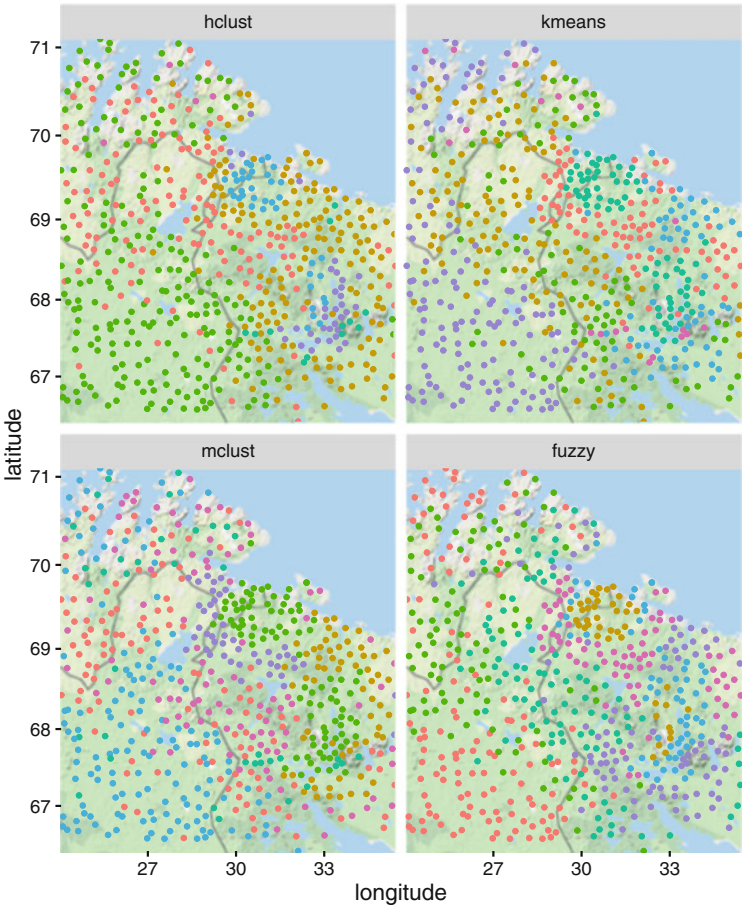


Fig. 6.11 Results of different cluster methods with 7 clusters for the selected Kola moss data

Figure 6.11 can now be easily produced with (code without background map):

```
ggplot(dfl, aes(x = XCOO, y = YCOO, color = cluster)) + geom_point() +
  facet_wrap(~variable) # produces Fig. 6.11.
```

Note that the sequence of the clusters, determining the color in the plots, can be arbitrary.

The clustering outcomes shown in Fig. 6.11 differ to some extent, although one can see similar patterns. For instance, all methods identify clusters around the Russian smelters Nikel/Zapolyarnij (about 2.5 longitude units left from Murmansk) and Monchegorsk (about one latitude unit below Murmansk; these locations are the outlier locations in Fig. 5.8).

For a better interpretation of the clusters it can be desirable to look at the dominance of each element in the composition of the cluster. This means that

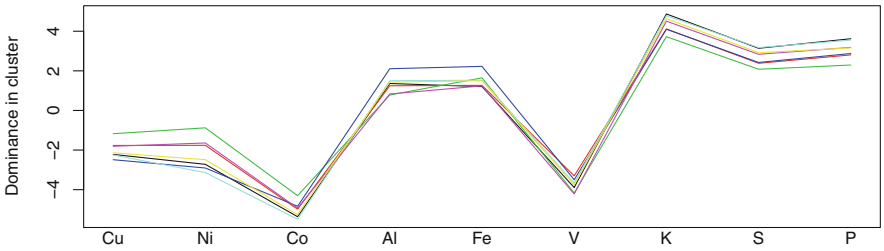


Fig. 6.12 Dominance of each element in each cluster for model-based clustering. The different colors of the lines are the same as those used to show the clusters in the map, see Fig. 6.11, lower left plot

for each compositional part a pivot coordinate is constructed—for each cluster separately. The results are shown in Fig. 6.12 for model-based clustering (Fig. 6.11, lower left plot). The colors of the lines are the same as the colors representing the clusters. The green cluster is just around the Russian smelters. Indeed, the elements Cu, Ni, and Co are dominant in this cluster, which confirms that these elements are typical emission elements. The dust elements Al, Fe, and V are dominant in the blue cluster, which in the map often follows some lines. For example, the blue cluster on the coast in the east is at the harbor of Murmansk. Finally, the black and pink cluster have dominance of the elements K, S, and P, and thus these refer to regions with biological processes in the mosses.

```
res.mclust <- Mclust(X.ilr, 7, verbose = FALSE)
cnter <- matrix(NA, nrow = 7, ncol = 9)
for (i in 1:7){ cnter[i,] <- apply(X[res.mclust$class==i, ], 2, gm) }
dom <- matrix(NA, nrow = 9, ncol = 7) # variables in rows
for (i in 1:nrow(dom)){
  Xi.ilr <- pivotCoord(cbind(X[,i], X[,-i]))
  for (j in 1:ncol(dom)){
    dom[i,j] <- mean(Xi.ilr[res.mclust$class == j, 1])
  }
}
matplot(dom, type = "l", lty = 1, xaxt = "n",
        ylab = "Dominance in cluster", col = 1:7)
mtext(sel, at = 1:9, side = 1)
```

The package **robCompositions** contains a convenient function called `clustCoDa()` which allows to call various clustering methods and different algorithms, and returns a unified output. As an illustration, hierarchical clustering with complete linkage is applied, as it has been done previously, see Fig. 6.10:

```
# Note that the input is X (selected moss variables in original scale.
# The function constructs coordinates by default.
res2.hclust <- clustCoDa(X, k = 7, method = "complete",
                 scale = "none", verbose = FALSE)
table(cutree(res.hclust, 7), res2.hclust$cluster) # comp. with res2.hclust

##
##       1   2   3   4   5   6   7
##   1 140   0   0   0   0   0   0
##   2   0 155   0   0   0   0   0
##   3   0   0 213   0   0   0   0
##   4   0   0   0  12   0   0   0
##   5   0   0   0   0  39   0   0
##   6   0   0   0   0   0  35   0
##   7   0   0   0   0   0   0   4
```

One can see that the cluster results are identical. For practitioners it might be convenient to use this function `clustCoDa()`, because this avoids to search for packages which implement the clustering algorithm one is interested in.

One issue that has not been discussed earlier is **scaling**. In the non-compositional case it is crucial for most cluster algorithms to first scale the variables in the data set to mean zero and variance one. This makes the variables comparable for their use with distance measures like the Euclidean distance. Is scaling necessary for compositional data? Obviously, the raw compositional data should not be scaled, because cluster analysis is applied in coordinates. So, the question is whether the coordinates should be scaled or not. The answer is immediate, since a basic requirement of a statistical analysis is invariance of the results with respect to the order of the input variables. If the sequence of the compositional parts in the data is changed—this corresponds to a rotation of the ilr coordinates given by the same formula, like for pivot coordinates (3.25)—the results from cluster analysis should be the same:

```
X3.ilr <- pivotCoord(X[, c(4:9,1:3)])  # change order of parts
res3.hclust <- hclust(dist(X3.ilr))
all.equal(res3.hclust$height, res.hclust$height)

## [1] TRUE
```

Indeed, the dendrogram information, stored in `$height`, is the same for the modified and the original sequence of the compositional parts. Now for scaled parts:

```
res4.hclust <- hclust(dist(scale(X3.ilr)))
res5.hclust <- hclust(dist(scale(X.ilr)))
all.equal(res4.hclust$height, res5.hclust$height)

## [1] "Mean relative difference: 0.2146178"
```

Here, the sequence of the variables makes a difference. Therefore, scaling should be strictly avoided.

In a final example the use of the *silhouette value*, introduced as validity criterion in Sect. 6.7, is illustrated. For this purpose the (selected) Oslo data set from Sect. 4.3 is used, which consists of element concentrations in nine different plant materials. It would be natural to find these plant materials as clusters in a cluster analysis. Plots

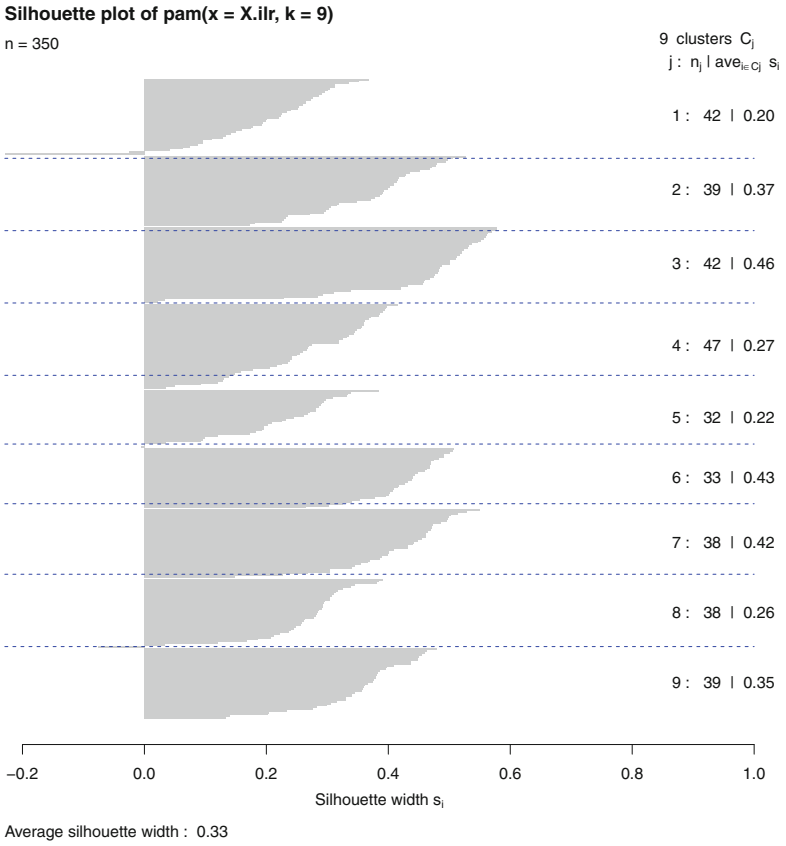


Fig. 6.13 Silhouette plot for the selected Oslo transect data, with horizontal dashed lines for the true groups (plant materials)

of these silhouette values are implemented in the package **cluster**. The algorithm `pam()` is used—it can be considered as a variant of k-means clustering which is more robust against data outliers, see Kaufman and Rousseeuw (1990) for details.

Figure 6.13 shows the resulting silhouette plot. The input data `X.ilr` are the same as used in the example of Sect. 4.3, and also the group information `grp` is as before. The plot presents all silhouette values as gray bars, and they are vertically arranged in the same order as the observations in the data set. Therefore, one can compare with the groups representing the different plant materials, using the information in `grp`: Observations of each group are arranged in blocks of the data matrix, and whenever the group label changes, a horizontal dashed line is added to the silhouette plot. Of course, it is not a must that groups and clusters are identical, but one would expect a close relation. Indeed, this is true for most groups, and only rarely some observations would have been better placed in other clusters

(negative silhouette value). The average silhouette width indicates the quality of the clustering, and this value could be compared with other cluster methods.

```
library("cluster")
res.pam <- pam(X.ilr, 9)
plot(res.pam, which.plots = 2)
abline(h = which(abs(diff(grp)) > 0), col = "blue", lty = 2)
```

References

K. Fačevicová, K. Hron, O. Bábek, T. Kumpan, Element chemostratigraphy of the Devonian/Carboniferous boundary – a compositional approach. Appl. Geochem. **75**, 211–221 (2016)

C. Fraley, A.E. Raftery, Model-based clustering, discriminant analysis, and density estimation. J. Am. Stat. Assoc. **97**(458), 611–631 (2002)

L. Kaufman, P.J. Rousseeuw, *Finding Groups in Data* (Wiley, New York, 1990)

G.N. Lance, W.T. Williams, Computer programs for hierachical polythetic classification (similarity analyses). Comput. J. **9**(1), 60–64 (1966)

J.M. McKinley, K. Hron, E.C. Grunsky, C. Reimann, P. de Caritat, P. Filzmoser, K.G. van den Boogaart, R. Tolosana-Delgado, The single component geochemical map: Fact or fiction? J. Geochem. Explor. **162**, 16–28 (2016)

C. Reimann, M. Äyräs, V. Chekushin, I. Bogatyrev, R. Boyd, P. de Caritat, R. Dutter, T.E. Finne, J.H. Halleraker, Ø. Jæger, G. Kashulina, O. Letho, H. Niskavaara, V. Pavlov, M.L. Räisänen, T. Strand, T. Volden, *Environmental Geochemical Atlas of the Central Parts of the Barents Region* (Geological Survey of Norway, Trondheim, 1998)

G.E. Schwarz, Estimating the dimension of a model. Ann. Stat. **6**(2), 461–464 (1978)

M. Templ, P. Filzmoser, C. Reimann, Cluster analysis applied to regional geochemical data: problems and possibilities. Appl. Geochem. **23**(8), 2198–2213 (2008)

K.G. van den Boogaart, R. Tolosana-Delgado, *Analyzing Compositional Data with R* (Springer, Heidelberg, 2013)

Chapter 7
Principal Component Analysis

Abstract Principal component analysis is a key tool in exploratory data processing to get an impression about the multivariate data structure. Its goal is to reduce dimensionality of the input data set by constructing new coordinates, called principal components, that seek for the highest possible explained variability. They can be derived by either using a singular value decomposition of the data matrix, or by an eigenvalue decomposition of the covariance matrix to get loadings (basis coefficients) and scores (coordinates) of the principal components. In case of compositional data these computations need to be done in orthonormal coordinates, preferably either in balances or pivot coordinate systems. The latter are closely related to clr coefficients that are historically preferred in case of principal component analysis. When the effect of outliers in any given orthonormal coordinate representation of the compositions needs to be suppressed, a robust covariance estimation can be used to get robust loadings and scores. Loadings and scores of the first two principal components are often visualized together using a planar graph called biplot that has a specific interpretation in case of clr coefficients.

7.1 Introductory Remarks

Principal component analysis (PCA) is one of the most popular and important multivariate statistical methods. Its goal is to reduce dimensionality of the input data set by constructing new coordinates, called principal components, which are used to capture the complex multivariate data structure. The dimension of the data set is reduced to the chosen number of principal components that are used either for a visual presentation of the information or for a subsequent statistical analysis. The principal components themselves are formed as linear combinations of the original variables with the aim to achieve maximum variability, with the constraint of mutual uncorrelatedness. Consequently, although standard algorithms for PCA extract all principal components, corresponding to the dimensionality of the data, just few of them are relevant for the subsequent analysis, namely those with the largest variance.

P. Filzmoser et al., *Applied Compositional Data Analysis*, Springer Series in Statistics, https://doi.org/10.1007/978-3-319-96422-5_7

It is not the goal of this section to describe PCA in every detail. There are many publications for this purpose (see, e.g., Johnson and Wichern 2007). Instead, the focus is on compositional aspects of PCA which may be of primary importance for readers of this book. One aspect is very intuitive: principal components, constructed in any orthonormal coordinate system, result in uncorrelated ilr coordinates. Moreover, for the purpose of PCA it is meaningful to also consider the respective orthonormal basis vectors, because their coefficients indicate which part, or group of parts, contributed (in the relative sense) most to the construction of the respective principal components.

Throughout the book, the analysis of compositional data is done with orthonormal coordinates, because they form the most natural way for a representation with respect to the Aitchison geometry. Traditionally, PCA is an exception with this respect in compositional data analysis, because clr coefficients are commonly applied instead. This dates back already to the work of Aitchison (1983). Here, a compromise is chosen: Although, due to methodological reasons, PCA will be developed in orthonormal coordinates, also the advantages of its computation and interpretation directly in clr coefficients will be shown. Namely, the latter approach turns out to be computationally much simpler and can thus be advantageously applied in practice.

7.2 Estimation of Principal Components

It frequently occurs in practice that due to a high initial number of compositional parts in the data set, it gets too complex to achieve an overview and an interpretation of the relations between the observations and/or compositional parts. In order to simplify the analysis, it is thus worth to investigate whether the initial components could be replaced by a smaller number of other, possibly artificial (latent) variables, that summarize the information about the original parts by accepting a minimum loss of information. Accordingly, PCA defines new variables, consisting of linear combinations of the original ones, in such a way that the first axis is in the direction containing most variation. Every subsequent new variable is orthogonal to the previous variables, but again in the direction containing most of the remaining variation. The new variables are termed principal components. From a practical perspective, PCA thus provides a direct mapping of the original, possibly high-dimensional data, into a lower-dimensional space capturing most of the information contained in the original data.

7.2.1 Estimation by SVD

Although there are several approaches to estimate principal components, the possibly most instructive one is based on singular value decomposition (Puntanen

et al. 2011). Assume that the compositional data set is given by the $n \times D$ data matrix $\mathbf{X}$. This information can be expressed in arbitrary interpretable orthonormal coordinates, for instance using Eq. (3.20). Denote the mean-centered coordinate data matrix by $\mathbf{Z}$. Mean-centering is essential here, because otherwise one would not obtain the directions maximizing the variance. To remind (see Sect. 5.1.1), SVD decomposes the $n \times (D-1)$ matrix $\mathbf{Z}$ into three parts,

$$\mathbf{Z} = \mathbf{U}\mathbf{D}\mathbf{W}', \tag{7.1}$$

where $\mathbf{U}$ is an $n \times p$ orthogonal matrix containing the left singular vectors, $\mathbf{D}$ is a diagonal matrix of order p containing the (positive) singular values $d_1, \ldots, d_p$, and $\mathbf{W}$ is a $(D-1) \times p$ orthogonal matrix containing the right singular vectors. Here, $p = \min(n, D-1)$ indicates the maximum number of principal components to be considered, and it is the minimum of the number of rows and columns of the data matrix $\mathbf{Z}$. Assume that the left and right singular vectors are sorted according to a decreasing order of the singular values, i.e. $d_1 \geq d_2 \geq \ldots \geq d_p > 0$. Rearranging (7.1) into

$$\mathbf{Z} = (\mathbf{U}\mathbf{D})\mathbf{W}' = \mathbf{Z}^*\mathbf{W}', \tag{7.2}$$

indicates the resulting PCA transformation. The coordinates of the samples in the new space, called *scores*, are contained in the matrix $\mathbf{Z}^* = (z^*_{ij})$. Note that due to orthogonal equivariance of PCA, the scores do not depend on the initial choice of orthonormal coordinates in $\mathbf{Z}$. The columns in $\mathbf{U}$ give the same scores in a normalized form: they have unit variances, whereas the variances of the columns in $\mathbf{Z}^*$ correspond to those of each particular principal component. These variances λ_i, for $i = 1, \ldots, p$, are proportional to the squares of the diagonal elements in the matrix $\mathbf{D}$,

$$\lambda_i = d_i^2/(n-1). \tag{7.3}$$

The sum of variances of the principal components is equal to the total variability of the compositional data set, i.e., $\sum_{i=1}^{p} \lambda_i = \text{totvar}(\mathbf{X})$ for $n \geq D-1$. Accordingly, the proportion of variance expressed by the i-th principal component is given by

$$P_i = \frac{\lambda_i}{\sum_{i=1}^{p} \lambda_i}.$$

There are many options how to set up the relevant number of principal components to be taken for the analysis. One popular choice is to examine the plot of λ_i against i, the so-called *scree plot* (scree diagram, Cattell (1966)). The number of components to be selected is the value of i corresponding to an "elbow" in the curve, i.e., a change of slope from "steep" to "shallow." The first point on the approximately straight line is then taken as the last component to be retained. Call this index $i = \tilde{p}$,

which now denotes the number of principal components one is interested in for the further analysis.

The columns of the matrix $\mathbf{W}$ (the right singular vectors) are called *loadings*, and $\mathbf{W}$ is called the loading matrix. Geometrically, the loadings form basis vectors of the principal components, or, even more specifically, orthonormal compositions (with respect to the Aitchison geometry), represented in the initial ilr coordinates, e.g., (3.20). From a practical perspective, the loadings can be seen as weights of the original ilr variables to determine the principal components. Therefore, from the perspective of the loadings it is crucial already how the initial coordinates are chosen. By using (3.20), the first element of the loading vector refers to the contribution of the relative dominance of x_1 in the composition for the construction of the given principal component. Similarly, by a permutation of the parts in the original composition, also the role of any other part in the construction of the principal components can be highlighted by using pivot coordinates (3.26), with the resulting loading matrix $\mathbf{W}^{(l)}$.

Finally, with the selected number $\tilde{p}$ of principal components, the input coordinate data matrix $\mathbf{Z}$ can be approximated by the $n \times (D-1)$ matrix $\tilde{\mathbf{Z}} = (\tilde{z}_{ij})$ as

$$\tilde{\mathbf{Z}} = \mathbf{Z}^*_{\tilde{p}} \mathbf{W}'_{\tilde{p}}, \tag{7.4}$$

where the matrices $\mathbf{Z}^*_{\tilde{p}}$ and $\mathbf{W}_{\tilde{p}}$ contain just the first $\tilde{p}$ columns of $\mathbf{Z}^*$ and $\mathbf{W}$, respectively. The approximation is considered in the least squares sense, which means that the Frobenius matrix norm

$$||\mathbf{Z} - \tilde{\mathbf{Z}}||_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{D-1} (z_{ij} - \tilde{z}_{ij})^2}$$

is minimal.

If an interpretable choice of orthonormal coordinates, like some expert knowledge based balances, is considered for PCA, a comprehensive information about groups of the original compositional parts is contained just in one coordinate system. Nevertheless, if an interpretability in the sense of single parts is preferable, one needs to compose the PCA output from D pivot coordinate systems. Particularly, for high-dimensional compositions, this is not always the best option due to a higher computational effort, and different alternatives have been proposed (Mert et al. 2015) with the aim to use just one interpretable coordinate system for the construction of principal components. An intuitive option is to take clr coefficients for PCA of compositional data. According to (3.30), there is a relation between the clr coefficients y_l and the first coordinates $z_1^{(l)}$ from (3.25), for $l = 1, \ldots, D$. By considering the singular value decomposition of the clr data matrix $\mathbf{Y}$,

$$\mathbf{Y} = \mathbf{U}_\mathbf{y} \mathbf{D}_\mathbf{y} \mathbf{W}'_\mathbf{y} = \mathbf{Y}^* \mathbf{W}'_\mathbf{y}, \tag{7.5}$$

it follows directly from (3.24) that the $D \times p$ clr loading matrix is equal to $\mathbf{W_y} = \mathbf{V}^{(l)}\mathbf{W}^{(l)}$. Specifically, (3.30) induces that the l-th row of $\mathbf{W_y}$ is equivalent to the first row of $\mathbf{W}^{(l)}$, differing only by the constant $\sqrt{\frac{D}{D-1}}$ (Kynčlová et al. 2016). Since the matrix of clr coefficients does not have full column rank, there is always at least one zero singular value in $\mathbf{D_y}$ for $n \geq D$, the other values coincide with those from $\mathbf{D}$. Because ilr coordinates form an orthonormal basis in the hyperplane induced by clr coefficients, the scores that correspond to the nonzero singular values are the same in both matrices $\mathbf{Y}^*$ and $\mathbf{Z}^*$.

These properties of PCA in clr coefficients are of particular practical importance. Namely, they refer to the fact that it is sufficient to proceed with PCA in clr coefficients, the score and loading information can be easily derived there. Particularly, the relative contributions of the original compositional parts (in the sense of clr variables, or equivalently, in the sense of the first pivot coordinates in (3.25)) to principal components can be analyzed directly from the clr loadings. Further peculiarities of the clr coefficients will be demonstrated in the context of compositional biplots for a graphical visualization of loadings and scores, introduced in Sect. 7.3. Nevertheless, it is worth to note that the relation between clr coefficients and orthonormal coordinates in the context of PCA is important also in the reverse direction. Using ilr coordinates for the construction of principal components justifies theoretically the possibility of including non-compositional variables into PCA, which is not straightforward for coefficients with respect to a generating system (Kynčlová et al. 2016).

7.2.2 *Estimation by Decomposing the Covariance Matrix*

An alternative algorithm to SVD for computing principal components is based on an estimation of the covariance matrix. Assume, like in the previous section, a mean-centered ilr coordinate data matrix $\mathbf{Z}$. Then the sample covariance matrix of $\mathbf{Z}$ is

$$\mathbf{S_z} = \frac{1}{n-1}\mathbf{Z}'\mathbf{Z}. \tag{7.6}$$

Using the SVD decomposition of (7.1) results in

$$\mathbf{S_z} = \frac{1}{n-1}\mathbf{WDU}'\mathbf{UDW}' = \frac{1}{n-1}\mathbf{WD}^2\mathbf{W}', \tag{7.7}$$

due to the orthonormality of the columns in $\mathbf{U}$. The diagonal matrix $\mathbf{D}^2$ contains the squared singular values $d_1^2, \ldots, d_p^2$ in its diagonal. Equation (7.7) shows that the columns of $\mathbf{W}$ are just the eigenvectors of $\mathbf{S_z}$ to the eigenvalues $d_i^2/(n-1)$, which are the variances of the principal components, see Eq. (7.3). In other words, an eigenvalue decomposition of the sample covariance matrix of the centered matrix

of ilr coordinates leads to the same solution as an SVD applied to these data. The loading matrix $\mathbf{W}$ is exactly the same in both approaches, and with the help of Eq. (7.2), one obtains the same matrix of scores, since

$$\mathbf{ZW} = \mathbf{UDW}'\mathbf{W} = \mathbf{UD} = \mathbf{Z}^*. \tag{7.8}$$

One could ask now, what is the advantage of this algorithm to obtain the PCA solution? While in the above approach the sample covariance matrix was used, see Eq. (7.6), one could also make use of a robustly estimated covariance, and proceed with the eigenvalue decomposition as explained above. A popular estimator for this purpose is the MCD estimator, mentioned already in Sect. 5.2.3. This estimator is applied to the uncentered coordinates, and yields robust estimates of the location $\mathbf{t}_{\text{MCD}}$ and covariance $\mathbf{C}_{\text{MCD}}$. As mentioned in Sect. 5.2.3, the MCD estimator is affine equivariant, and thus the specific choice of the ilr coordinates will not alter the resulting principal components.

The procedure to get principal components based on the MCD estimator (for the case $n \geq D-1$) is as follows (Filzmoser et al. 2009). The loading matrix is obtained by an eigenvalue decomposition of the MCD covariance matrix,

$$\mathbf{C}_{\text{MCD}} = \mathbf{W}_{\text{MCD}}\mathbf{D}_{\text{MCD}}\mathbf{W}'_{\text{MCD}},$$

with the matrix of eigenvectors $\mathbf{W}_{\text{MCD}}$, which is the loading matrix, and the diagonal matrix $\mathbf{D}_{\text{MCD}}$, containing the eigenvalues in its diagonal. These eigenvalues are the (robust) variances of the principal components. The principal component scores are obtained by first centering the coordinates matrix column-wise with the elements of $\mathbf{t}_{\text{MCD}}$. Denote these robustly centered coordinates by $\mathbf{Z}_{\text{MCD}}$. Then, following Eq. (7.8), the PCA scores are

$$\mathbf{Z}^*_{\text{MCD}} = \mathbf{Z}_{\text{MCD}}\mathbf{W}_{\text{MCD}}.$$

The resulting principal components are robust against outliers (Croux and Haesbroeck 2000). Practically, this means that the first most important principal components summarize the information described by the joint distribution of the data majority. For non-robust PCA it could happen that single outliers attract the first principal component directions, because these outliers lead to a large (non-robust) variance of those principal components. This is not desirable, since the purpose of PCA is not to identify outliers (PCA would also be unreliable for this purpose), but rather to summarize the information contained in the data majority in lower dimension.

As it was mentioned in Sect. 5.2.3, the computation of the MCD estimator is not possible if the determinant of the covariance matrix is zero, i.e. if the input matrix for PCA does not have full column rank. This is the case for clr coefficients. If a representation of PCA loadings and scores is preferred there, the MCD estimates of location and covariance need to be computed in any ilr coordinates and then expressed in the clr space in terms of loading and score matrices (Filzmoser et al.

2009). Because scores that correspond to nonzero singular values are the same for both clr and ilr representations, no further transformation of $\mathbf{Z}^*_{\mathrm{MCD}}$ is needed. For computing the loadings, the linear relation from Eq. (3.24) is used, where the columns of the matrix $\mathbf{V}$ are defined as in (3.23) for the pivot coordinates (3.20). Accordingly, the MCD estimate of the loading matrix in clr coordinates is given as

$$\mathbf{W}_{\mathbf{y},\mathrm{MCD}} = \mathbf{W}_{\mathrm{MCD}}\mathbf{V}'. \tag{7.9}$$

7.3 Compositional Biplot

The biplot is a two-dimensional graphical display of both objects (observations) and variables in one plot (Gabriel 1971). The term "bi" is not connected to the dimension two, but to the fact that both observations and variables are represented together. For our purpose, biplots are closely connected to the idea of PCA and singular value decomposition, although biplots can also be constructed for other methods (multidimensional scaling, correspondence analysis, nonlinear biplots, etc.), see Gower and Hand (1996). Obviously, two-dimensional plots are transparent and easy to handle. On the other hand, the projection of the data to two dimensions results from the assumption that the data set has approximately rank two, so that two dimensions explain the majority of the data variability. For a data matrix with a higher rank, the first two principal components should represent the data information sufficiently well.

Let the $n \times (D-1)$ matrix $\mathbf{Z}$ of mean-centered compositional data in ilr coordinates be decomposed as in (7.1) as $\mathbf{Z} = \mathbf{UDW}'$. A rank-two approximation of $\mathbf{Z}$ in the least squares sense, $\mathbf{Z}_{(2)}$, is obtained by taking the first two singular values d_{11}, d_{22} and the first two columns of $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_p)$ and $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_p)$,

$$\mathbf{Z} \approx \mathbf{Z}_{(2)} = (\mathbf{u}_1, \mathbf{u}_2) \begin{pmatrix} d_{11} & 0 \\ 0 & d_{22} \end{pmatrix} \begin{pmatrix} \mathbf{w}_1' \\ \mathbf{w}_2' \end{pmatrix}. \tag{7.10}$$

The matrix $\mathbf{Z}_{(2)}$ can also be expressed in the form

$$\mathbf{Z}_{(2)} = \mathbf{GH}' \tag{7.11}$$

with

$$\mathbf{G} = (\mathbf{u}_1, \mathbf{u}_2) \begin{pmatrix} d_{11} & 0 \\ 0 & d_{22} \end{pmatrix}^{1-c}$$

and

$$\mathbf{H} = (\mathbf{w}_1, \mathbf{w}_2) \begin{pmatrix} d_{11} & 0 \\ 0 & d_{22} \end{pmatrix}^c$$

for $c \in [0, 1]$. Depending on the choice of c, the first two singular values are "distributed" between the matrices $\mathbf{G}$ and $\mathbf{H}$, representing the observations and variables, respectively. The biplot is now formed just from the rows of the matrices $\mathbf{G}$ and $\mathbf{H}$, i.e. from $n + D - 1$ two-dimensional vectors. The rows of the matrix $\mathbf{G}$ define coordinates of the different observations, and they are shown as points in the biplot. The rows of $\mathbf{H}$ represent the variables and these points are the arrows-heads in the biplot, starting from the origin.

For the choice $c = 0.5$, the vectors for observations and variables would have the same scaling. With $c = 0$, the PCA transformation (7.2) for $p = 2$ components would be obtained. In such a case it is common to refer to the *form biplot* which favors the display of individuals (in terms of scores of principal components). Nevertheless, in practice the most common choice corresponds to $c = 1$, leading to the *covariance biplot*, which favors the display of the variables; this option will be discussed in the following. The principal component scores are now normed and the variability of the principal components is assigned to the loading vectors. Rescaling the matrices $\mathbf{G}$ and $\mathbf{H}$ leads to new matrices (though with the same notation),

$$\mathbf{G} = \sqrt{n-1}(\mathbf{u}_1, \mathbf{u}_2), \quad \mathbf{H} = \frac{1}{\sqrt{n-1}}(\mathbf{w}_1, \mathbf{w}_2) \begin{pmatrix} d_{11} & 0 \\ 0 & d_{22} \end{pmatrix}, \tag{7.12}$$

and to an intuitive interpretation of the biplot. Specifically, the inner product between the rows of $\mathbf{H}$, namely the matrix $\mathbf{H}'\mathbf{H}$, approximates the sample covariance matrix $\mathbf{S_z}$. Consequently, the length of the arrows approximates the standard deviation of the corresponding coordinates, and the cosine of the angle between two arrows indicates the correlation coefficient between the respective variables. It follows directly from the interpretation of the principal component loadings that the points in the direction of single arrows correspond to observations with high abundance of the respective variables. Finally, the Euclidean distance between the rows of $\mathbf{G}$ approximates the Mahalanobis distance between the observations. Thus, the distance of the point in the biplot from the origin (note that the data are centered) indicates the Mahalanobis distance of the data point from the mean, the result that can be useful also for outlier detection purposes.

Because biplots of compositional data cannot be constructed for the original observations and both individuals and variables are of mutual interest, the choice of interpretable coordinates is a key point to deliver a reasonable output of the biplot analysis. The first natural choice is to consider an interpretable SBP and proceed with the above interpretation of the biplot. In cases where such a coordinate system is not available, an alternative possibility is to take the set of pivot coordinate systems (3.25). Due to rotational invariance of SVD, the matrices $\mathbf{U}$ and $\mathbf{D}$ are

always the same. The only difference thus is in the loading matrices $\mathbf{W}^{(l)} = (\mathbf{w}_1^{(l)}, \ldots, \mathbf{w}_p^{(l)})$, $l = 1, \ldots, D$, where only the first two columns are used here. The first row of $\mathbf{W}^{(l)}$, or directly of the respective matrix

$$\mathbf{H}^{(l)} = \frac{1}{\sqrt{n-1}}(\mathbf{w}_1^{(l)}, \mathbf{w}_2^{(l)}) \begin{pmatrix} d_{11} & 0 \\ 0 & d_{22} \end{pmatrix},$$

contains the loading coefficients of $z_1^{(l)}$. Accordingly, the resulting arrows stand for the relative dominance of single parts within the composition. Of course, taking D biplots and considering always just one arrow there would not be very useful in practice. By following the case of PCA, the way out is to merge the arrows together into one biplot with the same score values (Kynčlová et al. 2016). Obviously, it would not be reasonable to consider correlations between the coordinates $z_1^{(l)}$, $l = 1, \ldots, D$, as they come from different coordinate systems. Moreover, due to the relation between clr coefficients and coordinates $z_1^{(l)}$ (3.30), one would also need to cope with the distorted covariance structure of clr variables. On the other hand, by considering orthonormal coordinates for the biplot construction it is straightforward to include also other compositional and non-compositional variables (the former in an appropriate coordinate representation), including correlation analysis with the newcoming variables.

The above considerations lead to a possibility to construct the biplot of compositional data directly in clr coefficients (Aitchison and Greenacre 2002), similar as for the case of PCA. In fact, this is still considered to be the default approach in the literature (van den Boogaart and Tolosana-Delgado 2013; Pawlowsky-Glahn et al. 2015), so it is frequently referred to as the *compositional biplot*. Because coefficients with respect to a generating system are employed there, certain limitations concerning the interpretation when compared to the standard biplot need to be taken into account. The lack of the possibility of correlation analysis between clr coefficients (in terms of angles between the arrows) is replaced by links between the vertices of rays corresponding to the clr variables y_j and y_k, $j, k \in \{1, \ldots, D\}$, $j \neq k$. The link approximates the variance of the pairwise logratio between the original compositional parts x_j and x_k, i.e. an element of the variation matrix (see Sect. 4.1). Accordingly, if two vertices coincide, or nearly so, the respective parts are proportional, or nearly so. Although also other interpretational tools for compositional biplots are available (Aitchison and Greenacre 2002), they are rather rarely used in practice. The only exception is the angle between two links, whose cosine approximates the correlation coefficient between two pairwise logratios. Accordingly, two uncorrelated logratios provide orthogonal rays; see, e.g., van den Boogaart and Tolosana-Delgado (2013) for details.

7.4 Examples

Principal component analysis can be carried out in R by the function `prcomp` (SVD-based, see Sect. 7.2.1) or by `princomp` (eigen-decomposition of the covariance matrix, see Sect. 7.2.2). The function `pcaCoDa` from the package **robCompositions** for PCA for compositional data uses internally the function `princomp`. This function works as described above: it represents the data in orthonormal coordinates before PCA is applied. For reasons of interpretation using the biplot, the results are then internally projected to clr coefficients. The function `pcaCoDa` has four important function arguments,

```
args(pcaCoDa)

## function (x, method = "robust", mult_comp = NULL, external = NULL)
## NULL
```

where `x` should be a compositional data set (as an object of class `data.frame`), `method` can be used to select alternatively a non-robust method (default is MCD-based robust PCA). The parameter `mult_comp` is required when the data consist of more than one composition which need to be analyzed jointly. Here, each composition is independently represented in orthonormal coordinates before PCA is applied. The function argument `external` is needed when data should be analyzed that consist of compositional parts and non-compositional variables. The function returns scores, loadings, eigenvalues, and a `princomp` object.

7.4.1 Representation of Principal Components in a Ternary Diagram

The first example shows the first principal component as a line in a ternary diagram, when the loadings (providing the direction) computed in ilr coordinates are re-expressed in the original sample space of the compositions, see Fig. 7.1. For this purpose, the data set `arcticLake` is used that consists of 39 three-part compositions on sand, silt, and clay in an Arctic lake, see Aitchison (1986). It is an example to use principal component analysis as a modeling tool to describe relationships between the variables and the trend in a data set if no response variable is available on which the variables might depend.

7.4.2 Example: Household Expenditures at EU Level

Principal component analysis is primarily a tool for dimension reduction. In the following, the `expenditureEU` data set is used which represents the mean consumption of 12 categories of expenditures of households at EU level for 27 countries, see also Sect. 6.6.

```
data("arcticLake")
ternaryDiag(arcticLake, line = "pca")
```

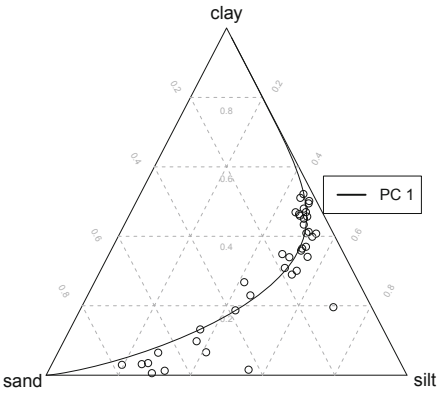


Fig. 7.1 First principal component direction represented in the original data space of the compositions

PCA for compositional data is carried out, and the non-robust approach is compared with the robust one.

```
data("expendituresEU")
## compositional PCA, non-robust
p_comp <- pcaCoDa(expendituresEU, method = "classical")
## compositional PCA, robust
set.seed(234) # to reproduce the result exactly
p_comp_rob <- pcaCoDa(expendituresEU, method = "robust")
```

The results are presented in Fig. 7.2 by compositional biplots. The non-robust and robust versions lead to slightly different results (note that the PCA results are unique up to the orientation of the axes). For instance, the variable *Health*, represented by the respective clr coefficient, has a different relation to the other variables in the robust analysis; in the non-robust case it is along the direction of relative dominance of *Food*, *Communications* and *Alcohol* within the given composition. The reason might be the effect of outliers like Sweden (S). Therefore, the robust analysis will be more reliable for the interpretation. One can see mainly former Eastern-European countries on the left-hand side of the plot, where the proportion of expenditures of *Food* is dominating. Cyprus (CY) has a high dominance of expenditures in *Education*, whereas for Sweden one can see a very low value.

The proportion of explained variance is also of interest. For the robust compositional PCA this can be seen by:

```
summary(p_comp_rob)

## Importance of components:
##                           Comp.1    Comp.2     Comp.3
## Standard deviation     1.3953487 0.8208709 0.55903864
```

```
par(mfrow=c(1,2), mar = c(4,4,2,2))
biplot(p_comp, xlabs = rownames(expendituresEU))
biplot(p_comp_rob, xlabs = rownames(expendituresEU))
```

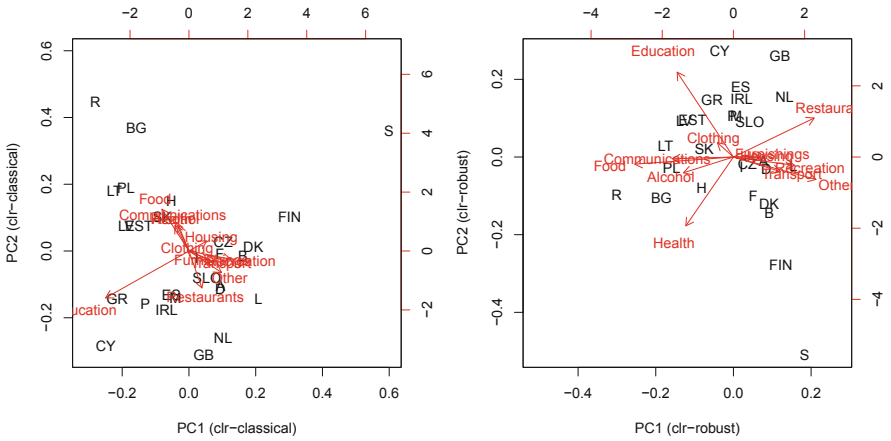


Fig. 7.2 Compositional biplot for non-robust (left) and robust (right) PCA

```
## Proportion of Variance 0.5596182 0.1936761 0.08982763
## Cumulative Proportion  0.5596182 0.7532943 0.84312191
##                            Comp.4     Comp.5      Comp.6
## Standard deviation     0.45989315 0.3301670 0.28497390
## Proportion of Variance 0.06079113 0.0313324 0.02334191
## Cumulative Proportion  0.90391303 0.9352454 0.95858735
##                            Comp.7     Comp.8       Comp.9
## Standard deviation     0.24635168 0.22835496 0.125850770
## Proportion of Variance 0.01744365 0.01498812 0.004552375
## Cumulative Proportion  0.97603100 0.99101912 0.995571496
##                           Comp.10      Comp.11
## Standard deviation     0.120623494 0.0292817685
## Proportion of Variance 0.004182059 0.0002464455
## Cumulative Proportion  0.999753554 1.0000000000
```

Note that the principal components are constructed in ilr coordinates, thus just 11 new variables are considered in the above list. After their transformation into clr coefficients the cumulative proportions would remain unchanged plus an additional component with zero standard deviation would be obtained.

The first two robust principal components explain about 75% of the total variance, and thus the biplot in Fig. 7.2 (right) is already meaningful. This information of explained variance can also be visualized in the scree plot, see Fig. 7.3. Indeed, the first two components seem to summarize the information contained in the data sufficiently well.

```
plot(p_comp_rob, type = "l")
```

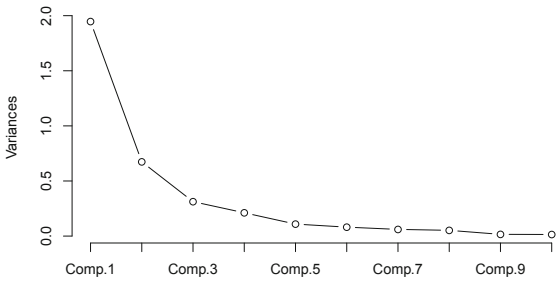


Fig. 7.3 Scree plot for robust compositional PCA of the expenditures data

```
res <- pcaCoDa(Beer, method = "classical")
par(mfrow = c(1,2), mar = c(4,4,2,2))
biplot(res, xlabs = Beer.age, xlim = c(-0.3,0.2))
biplot(res, xlabs = Beer.origin, xlim = c(-0.3,0.2))
```

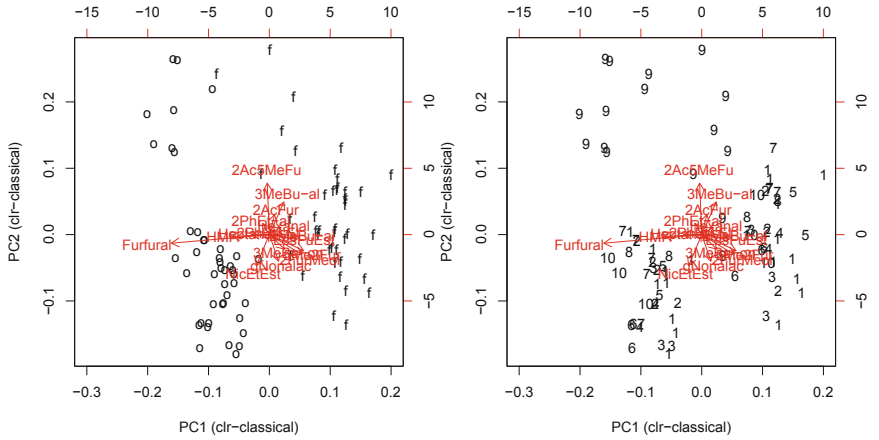


Fig. 7.4 Biplots for the beer data set. Left: observation are labeled according to the age of the beer (f-fresh, o-old); right: observation are labeled according to the producer

7.4.3 Example: Beer Data

In Sect. 1.2.2, some variants of PCA to a data set describing the composition of old and fresh beers have been presented. Here, the data set is treated as compositional data set, and the appropriate methodology is applied. Figure 7.4 shows the biplots resulting from non-robust compositional PCA. Because of the many variables, the biplots look a bit messy. The difference between the left and the right plot is only in the labels for the observations: The left biplot shows symbols “f” (fresh) and “o” (old) for the age of the beers, while the right biplot shows the numbers

of the different producers. Dominance in the variable *Furfural*, a toxic organic compound, seems to be a good indicator for old beer. However, producer 9 seems to produce a very different beer composition, with dominance, for instance, for variable *2Ac5MeFu*.

7.4.4 Example with Two Different Compositions

The Gemas data set (Reimann et al. 2012) is again considered, as it is available in `data(gemas)` of the package **robCompositions**. In fact, there are two different compositions available, measured on the same locations: the proportion of sand, silt, and clay in the soils, forming the first composition, and the concentration of various chemical elements, forming the second composition. Of course, the two compositions are related to each other, and these relations are investigated with PCA.

The function `pcaCoDa` has an argument *mult_comp*, which allows to provide a list with the column indexes of the different compositions. Then these compositions are extracted independently from the data, are expressed in coordinates, and are jointly treated in a PCA. Finally, the loadings are expressed in clr coefficients for the respective compositions in order to obtain an interpretation in the compositional (clr) biplot. Figure 7.5 shows the resulting biplot. High scores on the first principal component (PC1) refer to clay-dominant soils, which are also characterized by a dominance of Fe, V, and Cr, for instance. On the other hand, high values for PC2 are dominated by sand, also related to a dominance of Na and Sr. The scores are visualized in Fig. 7.6 in the European map, using symbols that result from a split of the distribution of the scores at the quantiles $q_{0.05}$, $q_{0.25}$, $q_{0.75}$, and $q_{0.95}$. The scores show clear regional patterns: Generally speaking, southern European soils are dominated by clay, while northern European soils are dominated by sand, with the corresponding chemical compositions.

7.4.5 Example for PCA Including External Non-compositional Variables

The article Kynčlová et al. (2016) has demonstrated how a biplot can be constructed if, in addition to a composition, external non-compositional variables are available. This is also implemented in the function `pcaCoDa`, where the external variables can be provided through the argument *external*.

As an illustration, data from the German federal election 2013 are used, where the election data in the different federal states are considered. The compositional

```
data("gemas")
# Index of rows with NA's (25 NA's in summary)
isna <- missPatterns(gemas)$rindex
# Index of rows with zeros (4 zeros in summary)
iszero <- zeroPatterns(gemas)$rindex
# exclude those
gemas <- gemas[!isna & !iszero, ]
# pca
res <- pcaCoDa(gemas, mult_comp = list(c(9:11), c(12:29)))
biplot(res, xlabs = rep(".", nrow(gemas)))  # obs. as dots
```

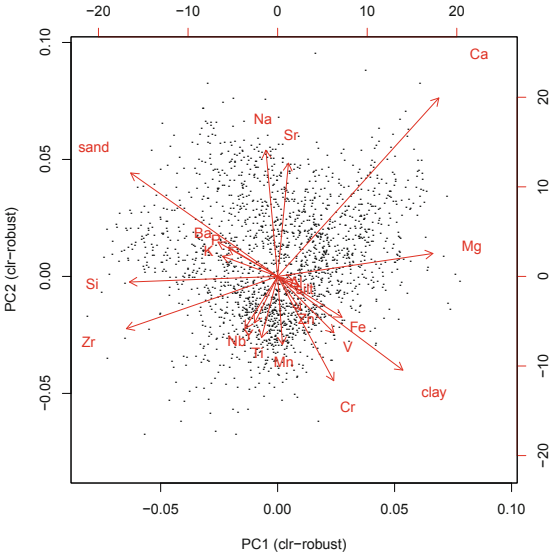


Fig. 7.5 Biplot for the two compositions in the data set `gemas`: the sand/silt/clay composition, and the composition with the element concentrations

parts are the parties CDU/CSU (Christian Democratic Union and Christian Social Union of Bavaria), SDP (Social Democratic Party), DIE LINKE (The Left), GRÜNE (Alliance '90/The Greens), FDP (Free Democratic Party) and other parties, forming the rest of the parties which participated in the elections. The votes are reported in absolute numbers of valid votes. In addition to this composition, for the same federal states the unemployment rate and the average monthly income in Euros are considered as external variables. The data are available as `data(election)`

```
library("StatDA")
par(mfrow = c(1,2), mar = c(0.1,0.1,0.1,0.1))
xc <- gemas[, 2]  # longitude
yc <- gemas[, 3]  # latitude
plot(xc, yc,
    type = "n", xaxt = "n", yaxt = "n", xlim = c(-11,37), ylim = c(33,72))
SymbLegend(xc, yc, res$sco[, 1],
    leg.position = "topleft", leg.title = "PC1 scores")
plot(xc, yc,
    type = "n", xaxt = "n", yaxt = "n", xlim = c(-11,37), ylim = c(33,72))
SymbLegend(xc, yc, res$sco[, 2],
    leg.position = "topleft", leg.title = "PC2 scores")
```

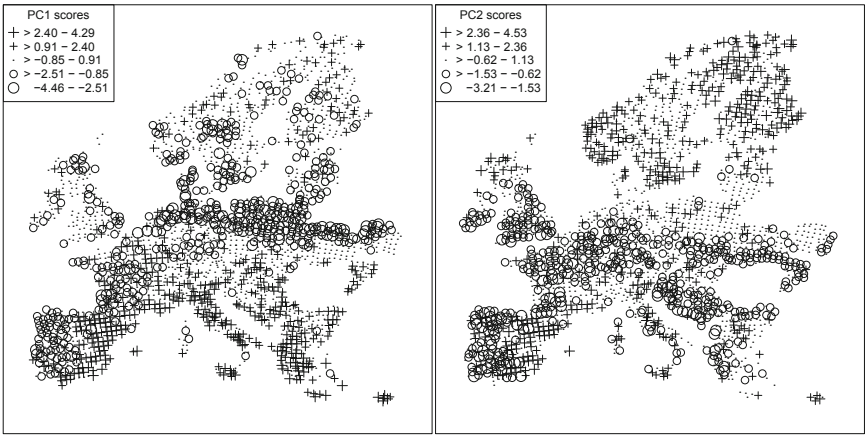


Fig. 7.6 Plots of the PC1 and PC2 scores from Fig. 7.5 in the European map

from the package **robCompositions**. After some transformation/scaling of the external variables, the output from robust PCA is obtained:

```
data("election")
# transform external variables
ue <-  election$unemployment / 100
ue.tr <- log((ue) / (1 - ue))      # logit transformation
inc <- scale(election$income)  # scale the Euro values
ext <- data.frame(Unemployment = ue.tr, Income = inc)
dimnames(ext)[[1]] <- dimnames(election)[[1]]
## PCA
res <- pcaCoDa(election[, 1:6], method = "robust", external = ext)
summary(res)
```

```
## Importance of components:
##                             Comp.1     Comp.2     Comp.3
## Standard deviation     2.0614949 0.66280436 0.38814335
## Proportion of Variance 0.8533327 0.08821137 0.03025089
## Cumulative Proportion  0.8533327 0.94154408 0.97179497
##                             Comp.4      Comp.5       Comp.6
## Standard deviation     0.30986292 0.174155774 0.117763961
## Proportion of Variance 0.01927938 0.006090173 0.002784702
```

```
## Cumulative Proportion  0.99107435 0.997164521 0.999949223
##                                        Comp.7
## Standard deviation     1.590219e-02
## Proportion of Variance 5.077708e-05
## Cumulative Proportion  1.000000e+00
```

Accordingly, the first two principal components explain much more than 90% of the total variance, and thus a biplot of these components will be very informative, see Fig. 7.7. A cluster of very similar observations for small values on PC1 gets immediately visible; these are the federal states Brandenburg (BB), Mecklenburg-Vorpommern (MV), Saxony (SN), Saxony-Anhalt (ST), and Thuringia (TH), which (with the exception of East Berlin) constitute the former East Germany. Not only the voting behavior is different in these states, but also the distribution on the external variables is different (e.g., lower income). It is also interesting to see the relation between the external variables and the votes.

```
biplot(res, scale = 0) # produces Fig. 7.7
```

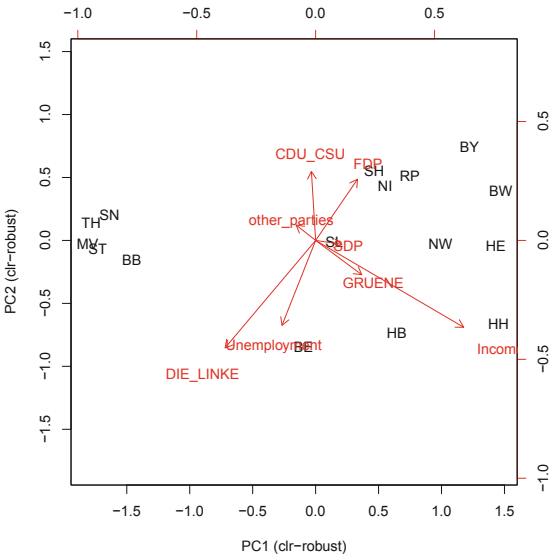


Fig. 7.7 Biplot for the German federal election data, joined with two external variables *Unemployment* and *Income*

References

J. Aitchison, Principal component analysis of compositional data. Biometrika **70**(1), 57–65 (1983)
J. Aitchison, *The Statistical Analysis of Compositional Data* (Chapman & Hall, London, 1986). Reprinted in 2003 with additional material by The Blackburn Press
J. Aitchison, M. Greenacre, Biplots of compositional data. J. R. Stat. Soc. Ser. C (Appl. Stat.) **51**(4), 375–392 (2002)
R.B. Cattell, The scree test for the number of factors. Multivar. Behav. Res. **1**, 245–276 (1966)
C. Croux, G. Haesbroeck, Principal components analysis based on robust estimators of the covariance or correlation matrix: influence functions and efficiencies. Biometrika **87**, 603–618 (2000)
P. Filzmoser, K. Hron, C. Reimann, Principal component analysis for compositional data with outliers. Environmetrics **20**, 621–632 (2009)
K.R. Gabriel, The biplot graphic display of matrices with application to principal component analysis. Biometrika **58**(3), 453–467 (1971)
J.C. Gower, D.J. Hand, *Biplots* (Chapman & Hall, London, 1996)
R.A. Johnson, D.W. Wichern, *Applied Multivariate Statistical Analysis*, 6th edn. (Prentice Hall, Upper Saddle River, 2007)
P. Kynčlová, P. Filzmoser, K. Hron, Compositional biplots including external non-compositional variables. Statistics **50**(5), 1132–1148 (2016)
C. Mert, P. Filzmoser, K. Hron, Sparse principal balances. Stat. Model. **15**(2), 159–174 (2015)
V. Pawlowsky-Glahn, J.J. Egozcue, R. Tolosana-Delgado, *Modeling and Analysis of Compositional Data* (Wiley, Chichester, 2015)
S. Puntanen, G.P.H. Styan, J. Isotalo, *Matrix Tricks for Linear Statistical Models* (Springer, Heidelberg, 2011)
C. Reimann, P. Filzmoser, K. Fabian, K. Hron, M. Birke, A. Demetriades, E. Dinelli, A. Ladenberger, The GEMAS Project Team, The concept of compositional data analysis in practice-Total major element concentrations in agricultural and grazing land soils of Europe. Sci. Total Environ. **426**, 196–210 (2012)
K.G. van den Boogaart, R. Tolosana-Delgado, *Analyzing Compositional Data with R* (Springer, Heidelberg, 2013)

Chapter 8
Correlation Analysis

Abstract The goal of correlation analysis is to quantify the strength of the relationship between a pair of variables or between groups of variables. In case of compositional data, it might be particularly misleading to compute correlation coefficients for the original data: due to scale invariance of the compositions, any correlation values could be obtained, depending on the representation of the compositional data in the respective equivalence classes. Therefore, a proper coordinate representation of compositions is again a must. A default setting are balance coordinates, for which either the standard Pearson correlation coefficient as a measure of strength of the linear association between the two balances or robust correlations can be computed. If an interpretation of the correlations in terms of a pair of parts is required, symmetric pivot coordinates capturing the dominance of these parts within the given composition are recommendable. Correlation analysis between two coordinates can also be extended to correlations between one coordinate and a set of coordinates, or to group correlations summarizing the relationships between balance representations of groups of compositional parts.

8.1 Correlation Measures

Since compositional data are expressed in orthonormal coordinates, like in balances $\tilde{z}_1, \ldots, \tilde{z}_{D-1}$ (3.37), any of the established correlation measures can be used to evaluate their association. For measuring the strength and direction of the linear relationship between two coordinates, the well-known Pearson correlation coefficient is widely used, defined as

$$\rho_{\tilde{z}_i,\tilde{z}_j} = \frac{\mathrm{cov}(\tilde{z}_i, \tilde{z}_j)}{\sqrt{\mathrm{var}(\tilde{z}_i)\,\mathrm{var}(\tilde{z}_j)}}, \quad i, j = 1, \ldots, D-1. \tag{8.1}$$

This measure is normed to the interval $[-1, 1]$, with 0 indicating no linear relation, and 1 (-1) for perfect positive (negative) linear relation. It is possible to express the

P. Filzmoser et al., *Applied Compositional Data Analysis*, Springer Series in Statistics, https://doi.org/10.1007/978-3-319-96422-5_8

squared correlation coefficient as

$$\rho^2_{\tilde{z}_i,\tilde{z}_j} = 1 - \frac{|\boldsymbol{\Sigma}|}{\mathrm{var}(\tilde{z}_i)\,\mathrm{var}(\tilde{z}_j)} \quad \text{with} \quad \boldsymbol{\Sigma} = \begin{pmatrix} \mathrm{var}(\tilde{z}_i) & \mathrm{cov}(\tilde{z}_i,\tilde{z}_j) \\ \mathrm{cov}(\tilde{z}_j,\tilde{z}_i) & \mathrm{var}(\tilde{z}_j) \end{pmatrix}, \tag{8.2}$$

where $|\boldsymbol{\Sigma}|$ denotes the determinant of $\boldsymbol{\Sigma}$.

Equation (8.1) refers to random variables $\tilde{z}_i$ and $\tilde{z}_j$, and to the theoretical covariance "cov" and theoretical variance "var". When a sample of n balance coordinates (3.37) is available, the covariance and variances are replaced by their sample counterparts. Either the classical estimators can be used or in presence of data outliers robust counterparts can be taken, e.g. based on the MCD estimator, see Sect. 5.2.3 for details. Correlation coefficients of coordinates $\tilde{z}_1, \ldots, \tilde{z}_{D-1}$ can be arranged in the correlation matrix $\mathbf{R} = (\rho_{\tilde{z}_i,\tilde{z}_j})$ of order $D-1$. By construction this is a symmetric, positive definite matrix with ones forming the main diagonal.

The interpretation of the correlation coefficient should acknowledge the fact that not the original compositional parts, but two balances are considered. In other words, as the coordinates (3.37) express balances between two groups of compositional parts, a positive value of the coefficient indicates that with increasing dominance of the group of parts in the numerator of $\tilde{z}_i$ over the parts in its denominator, also the dominance of the parts in the numerator of $\tilde{z}_j$ increases, and vice versa for negative correlation. This interpretation could be a bit tricky, if sequential binary partitioning is not defined according to a deeper knowledge of the inherent data processes. As an alternative, symmetric pivot coordinates (3.35) and (3.36) can be used, where the roles of the single compositional parts are highlighted. This case is briefly discussed in the next section. Note that clr coefficients are not recommended for the purpose of correlation analysis because of a negative bias of the covariance (correlation) structure (Sect. 3.3.4).

The Pearson correlation coefficient is definitely not the only option for measuring statistical relationships between two or more random variables or the observed data values. For example, an important class are rank correlations that aim to study the association between the ranks of different variables or different rankings of the same variable. Here one can use:

- Spearman's rank correlation coefficient, a measure of how well the relationship between two variables can be described by a monotonic function;
- Kendall's tau correlation coefficient, a measure of the portion of ranks that match between two variables;
- Goodman and Kruskal's gamma, a measure of the strength of association of the cross tabulated data when both variables are measured at the ordinal level.

Rank correlation is particularly recommendable in cases when the distribution of both variables is strongly deviating from normality. It is just important to note that even for these measures, when applied to compositional data, a proper coordinate representation is needed. On the other hand, rank correlation measures do not utilize the whole information contained in the data, resulting from ordering the coordinate

values. Therefore, throughout this book just classical and robust Pearson correlation is applied.

8.2 Relating Two Compositional Parts

The efforts to perform correlation analysis between the original compositional parts of a D-part composition $\mathbf{x} = (x_1, \ldots, x_D)'$ led to some confusion in the past (Pearson 1897; Chayes 1960). One reason is the negative bias of the covariance structure, when the proportional representation of compositional data is considered, namely

$$\mathrm{cov}(x_l, x_1) + \ldots + \mathrm{cov}(x_l, x_{l-1}) + \mathrm{cov}(x_l, x_{l+1}) + \ldots + \mathrm{cov}(x_l, x_D) = -\mathrm{var}(x_l),$$

for $l = 1, \ldots, D$. Similar as for clr coordinates, there is a tendency to negative covariances, and thus the correlations lose their predicative value. Moreover, the correlation structure of compositional data is not scale invariant: using arbitrary (generally different) representations with sum κ of the parts in (3.4) for each composition in the sample would lead to arbitrary values of the correlation coefficients. Therefore, a proper coordinate representation of compositions is a must.

It was advocated in the previous section that for measuring association of two compositional parts in terms of correlation analysis, symmetric pivot coordinates $z_1^{(i,j)}$ (3.35) and $z_2^{(i,j)}$ (3.36) are preferred. Nevertheless, also here one must be careful with the interpretation of the resulting correlation coefficient. Concretely, both symmetric pivot coordinates can be interpreted in terms of a dominance of both parts to the average behavior of the rest (Kynčlová et al. 2017). Hence, the remaining parts can influence the value of the correlation coefficient as well, which fully corresponds to the relative nature of compositional data. As a consequence, a positive correlation coefficient would mean that the dominances of the two amounts over the respective "average representatives" of the other parts increase simultaneously and vice versa for negative correlation. A zero coefficient would mean that the dominances of these two amounts are controlled by uncorrelated processes. Of course, part x_1 is contained in $z_2^{(i,j)}$ and, conversely, x_2 in $z_1^{(i,j)}$. Accordingly, it is interesting to see what happens if ratios with x_1 uniformly increase by a constant behavior of the other parts (and their ratios). By construction of both coordinates, while $z_1^{(i,j)}$ increases, $z_2^{(i,j)}$ slightly decreases (x_1 is contained with reduced power in its denominator), resulting in negative correlation. This reminds to the case of correlation between two original parts in a proportional representation, but now in a geometrically reasonable manner with orthonormal coordinates. Moreover, it is also a kind of logical result: if the dominance of one part (here x_1) increases, the dominance of another part (x_2) must necessarily decrease. However, the effect for the latter part cannot be the same: x_1 is just one out of $D-1$ parts to which the dominance of x_2 is related.

By summarizing all corresponding correlation coefficients between $z_1^{(i,j)}$ and $z_2^{(i,j)}$ into one matrix, the *pivot correlation matrix* $\mathbf{R}_P(\mathbf{x})$ of order D is obtained (Kynčlová et al. 2017). It is symmetric with unit diagonal as the standard correlation matrix. Moreover, any scaling and shifting in the compositional sense, i.e. by perturbing $\mathbf{x}$ with a non-random composition $\mathbf{b} = (b_1, \ldots, b_D)'$ and powering with a real constant a in order to get a composition $a \odot \mathbf{x} \oplus \mathbf{b} = (x_1^a b_1, \ldots, x_D^a b_D)'$, yields the same result, $\mathbf{R}_P(a \odot \mathbf{x} \oplus \mathbf{b}) = \mathbf{R}_P(\mathbf{x})$. Although experiments with data sets indicated some further interesting properties (like positive definiteness), it is crucial to realize that the elements of $\mathbf{R}_P(\mathbf{x})$ are formed by using $D(D-1)/2$ different coordinate systems that prevents from processing it simply as a whole, e.g. by computing principal components.

8.3 Multiple Correlation

The expression (8.2) of the squared correlation coefficient opens a possibility to consider more general correlation measures, appropriate in case of compositional data (Filzmoser and Hron 2009). A measure of linear relationship between a balance $\tilde{z}_i$ and a group of balances $\tilde{\mathbf{z}}_k$ is the *multiple correlation coefficient* $\rho_{\tilde{z}_i,\tilde{\mathbf{z}}_k}$. This measure returns a value in the interval $[0, 1]$, where 0 indicates no linear relationship and 1 perfect linear relation. The square of the multiple correlation coefficient is defined as

$$\rho^2_{\tilde{z}_i,\tilde{\mathbf{z}}_k} = \frac{\mathrm{cov}(\tilde{z}_i, \tilde{\mathbf{z}}_k)\boldsymbol{\Sigma}_k^{-1}\mathrm{cov}(\tilde{\mathbf{z}}_k, \tilde{z}_i)}{\mathrm{var}(\tilde{z}_i)}, \tag{8.3}$$

where $\boldsymbol{\Sigma}_k = \mathrm{cov}(\tilde{\mathbf{z}}_k)$,

$$\mathrm{cov}(\tilde{z}_i, \tilde{\mathbf{z}}_k) = (\mathrm{cov}(\tilde{z}_i, \tilde{z}_1), \ldots, \mathrm{cov}(\tilde{z}_i, \tilde{z}_{i-1}), \mathrm{cov}(\tilde{z}_i, \tilde{z}_{i+1}), \mathrm{cov}(\tilde{z}_i, \tilde{z}_{D-1}))$$

and $\mathrm{cov}(\tilde{\mathbf{z}}_k, \tilde{z}_i) = [\mathrm{cov}(\tilde{z}_i, \tilde{\mathbf{z}}_k)]'$. An equivalent formulation is

$$\rho^2_{\tilde{z}_i,\tilde{\mathbf{z}}_k} = 1 - \frac{|\boldsymbol{\Sigma}|}{|\boldsymbol{\Sigma}_k|\,\mathrm{var}(\tilde{z}_i)} \quad \text{with} \quad \boldsymbol{\Sigma} = \begin{pmatrix} \mathrm{var}(\tilde{z}_i) & \mathrm{cov}(\tilde{z}_i, \tilde{\mathbf{z}}_k) \\ \mathrm{cov}(\tilde{\mathbf{z}}_k, \tilde{z}_i) & \mathrm{cov}(\tilde{\mathbf{z}}_k) \end{pmatrix}.$$

Similar as for the Pearson correlation coefficient, the sample multiple correlation coefficient can be computed either classically using the sample covariance matrix or robustly by employing the MCD estimator, for instance.

Typically, instead of computing (8.3) with general balances (3.37), pivot coordinates (3.25) are utilized; $z_1^{(l)}$ is taken in place of z_i, and for $\mathbf{z}_k$ the remaining pivot balances $z_2^{(l)}, \ldots, z_{D-1}^{(l)}$ are considered. Accordingly, the resulting multiple correlation coefficient is a measure of strength of the linear relationship between relative information on x_l and the rest of the composition. Small values of the coefficient

indicate an exceptional behavior of dominance of the part x_l with respect to the other compositional parts. Finally, a large difference between the classical and robust versions of the coefficient indicates that the possible relation is driven by outliers.

Alternatively, the multiple correlation coefficient can also be used to measure the strength of the linear relationship between a non-compositional variable and a composition, expressed in (any) ilr coordinates. For affine equivariant estimators of location and covariance (classical, or robust) the resulting value is always the same.

8.4 Correlation Between Groups of Compositional Parts

It is possible to go even one step further and define the *group correlation coefficient* $\rho_{\tilde{\mathbf{z}}_k,\tilde{\mathbf{z}}_l}$ for two vectors $\tilde{\mathbf{z}}_k$ and $\tilde{\mathbf{z}}_l$ of balance coordinates, representing usually two non-overlapping groups of compositional parts of one composition, or two compositions. Its square is defined as

$$\rho^2_{\tilde{\mathbf{z}}_k,\tilde{\mathbf{z}}_l} = 1 - \frac{|\boldsymbol{\Sigma}^*|}{|\boldsymbol{\Sigma}_k|\,|\boldsymbol{\Sigma}_l|} \quad \text{with} \quad \boldsymbol{\Sigma}^* = \begin{pmatrix} \operatorname{cov}(\tilde{\mathbf{z}}_k) & \operatorname{cov}(\tilde{\mathbf{z}}_k, \tilde{\mathbf{z}}_l) \\ \operatorname{cov}(\tilde{\mathbf{z}}_l, \tilde{\mathbf{z}}_k) & \operatorname{cov}(\tilde{\mathbf{z}}_l) \end{pmatrix},$$

where $\boldsymbol{\Sigma}_k = \operatorname{cov}(\tilde{\mathbf{z}}_k)$, $\boldsymbol{\Sigma}_l = \operatorname{cov}(\tilde{\mathbf{z}}_l)$ and $\operatorname{cov}(\tilde{\mathbf{z}}_k, \tilde{\mathbf{z}}_l) = [\operatorname{cov}(\tilde{\mathbf{z}}_l, \tilde{\mathbf{z}}_k)]'$ stands for a matrix of covariances between coordinates from $\tilde{\mathbf{z}}_k$ and $\tilde{\mathbf{z}}_l$, see Anderson (2003). Similar as for the multiple correlation coefficient, this measure yields a value in the interval [0, 1], where 0 indicates no linear relationship and 1 perfect linear relation between the groups of compositional parts.

This straightforward extension is less known, but there is a link to the more frequently used canonical correlation analysis. This method not only measures the linear relationship between two multivariate data sets, but searches for latent variables—so-called *canonical variates*—in each of the data groups such that the scores on the latent variables have maximal correlation (see, e.g., Johnson and Wichern 2007). Nevertheless, these new variables must not necessarily be representative (in terms of explained variance) within the multivariate data structure of balance groups, e.g., like principal components. Therefore, the use of canonical correlation analysis is rather limited to more specific problems; an example of their use with compositional data in geochemistry is presented in Filzmoser and Hron (2009).

A natural question which might arise is, which values of the above coefficients are already high enough to represent a strong association between the coordinates or their groups. One possibility would be to apply statistical inference like hypotheses testing. However, it maybe more useful to state that this varies strongly depending on the concrete data and problem setting. While in natural sciences in general stronger associations are requested, it is frequently not the case with data coming from economics or social sciences, or for omics data containing many erroneous parts. Therefore, it is rather relevant to compare the values of the correlation coefficients mutually to see, which relationship is relatively stronger than the others.

8.5 Examples

8.5.1 *Example for Correlation Between Single Compositional Parts*

Consider the data set `data(phd)` from the package **robCompositions**, see also Table 1.2, with the numbers of PhD students in different subject areas. This data set has also been used in the examples of Chap. 3; Fig. 3.5 already presented symmetric pivot coordinates for the two subject areas "technical" and "health." This is the appropriate representation for computing the correlation between these two parts, because these orthonormal coordinates represent symmetrically all relative information of these parts of interest to the rest. Moreover, since these coordinates represent the information in the usual Euclidean geometry, standard correlation measures can be employed which rely on this geometry.

Figure 8.1 shows a scatterplot matrix, where for each plot in this matrix symmetric pivot coordinates were constructed and the first two of them were taken for the parts representing the axes. In other words, for each plot symmetric pivot coordinates need to be constructed individually, but there is of course symmetry around the main diagonal of the plot. Within the loop generating the plots, also the Pearson correlation and Spearman's rank correlation coefficient is computed.

The correlations can also be computed using the function `corCoDa` from the package **robCompositions**. The code is as follows:

```
Rp1 <- corCoDa(phdred)
Rs1 <- corCoDa(phdred,method="spearman")
```

The resulting objects `Rp1` and `Rs1` are identical with `Rp` and `Rs`, respectively, from the code in Fig. 8.1.

Figure 8.1 reveals several outliers, most visibly France (FR) where the proportions of PhD students are quite different to those for most other countries. Such outliers may affect the Pearson correlation, but they are less influential to a Spearman rank correlation since this measure is just based on the ranks of the observations, and not directly on their values of the symmetric pivot coordinates. One could also compute robust correlations using the MCD estimator. In that case, the R code to compute the (i, j)-th element of this matrix would be as follows:

```
Rr[i, j] <- covMcd(Z[, 1:2], cor = TRUE)$cor[1, 2]
```

The information contained in the pivot correlation matrix can be visually displayed in heatmaps, where the correlation coefficients in the interval $[-1, 1]$ are simply color coded, symmetrically around zero. In this way it is easy to compare the outcomes of different correlation estimators. This is shown in Fig. 8.2 for the Pearson (left) and the Spearman rank (right) correlation. One can see quite some changes in the correlation coefficients, and the reason are the outliers which have strong influence on the classical Pearson correlation. The highest positive correlation is between the symmetric coordinates for socio-economic and law studies and

```
D <- ncol(phdred)
par(mfrow = c(D, D), mar = c(0.1,0.1,0.1,0.1))
Rp <- Rs <- matrix(NA, ncol = D, nrow = D)
nam <- substr(names(phdred), 1, 5)  # shorter variable names
dimnames(Rp) <- dimnames(Rs) <- list(nam, nam)
diag(Rp) <- diag(Rs) <- rep(1, D)
for (i in 1:D){
  for (j in 1:D){
    if (i==j){
      plot(0, 0, type = "n", xaxt = "n", yaxt = "n")
      text(0, 0, names(phdred[i]))
    }
    else{
      Z <- pivotCoord(phdred[, c(i, j, (1:D)[-c(i, j)])], method = "symm")
      plot(Z[, 1:2],
           xaxt = "n", yaxt = "n", xlab = "", ylab = "", type = "n")
      text(Z[, 1], Z[, 2], coun, cex = 0.7)
      Rp[i,j] <- cor(Z[, 1:2])[1, 2]   # Pearson correlation
      Rs[i,j] <- cor(Z[, 1:2], method = "spearman")[1, 2] # Spearman corr.
    }
  }
}
```

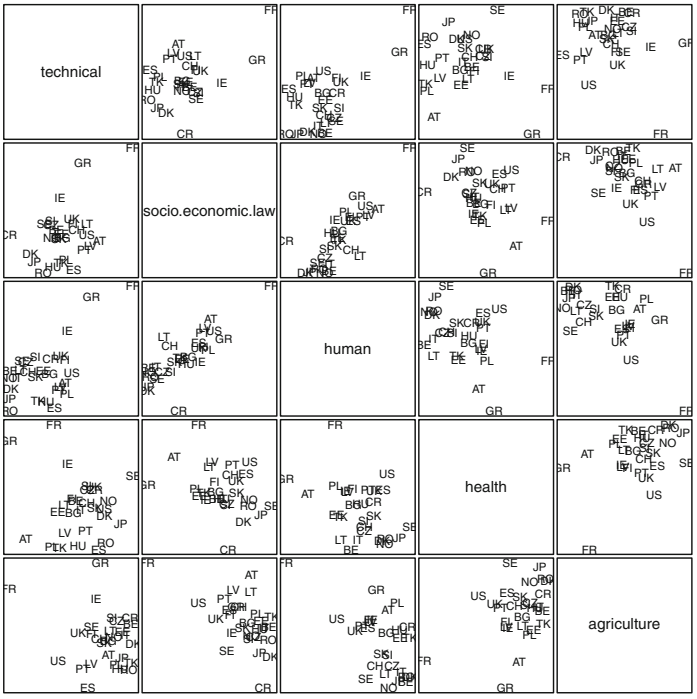


Fig. 8.1 Scatterplot matrix of the PhD data using symmetric pivot coordinates for each single plot

```
library("gplots")
rgbcol <- colorRampPalette(c("blue4","turquoise","white","orange","red4"),
                space = "rgb")
heatmap.2(as.matrix(Rp), Rowv = FALSE, symm = TRUE, col = rgbcol(256),
          key = TRUE, trace = "none", main = "Pearson correlation",
          margins = c(4, 4), cexRow = 1.2, cexCol = 1.2)
heatmap.2(as.matrix(Rs), Rowv = FALSE, symm = TRUE, col = rgbcol(256),
          key = TRUE, trace = "none", main = "Spearman correlation",
          margins = c(4, 4), cexRow = 1.2, cexCol = 1.2)
```

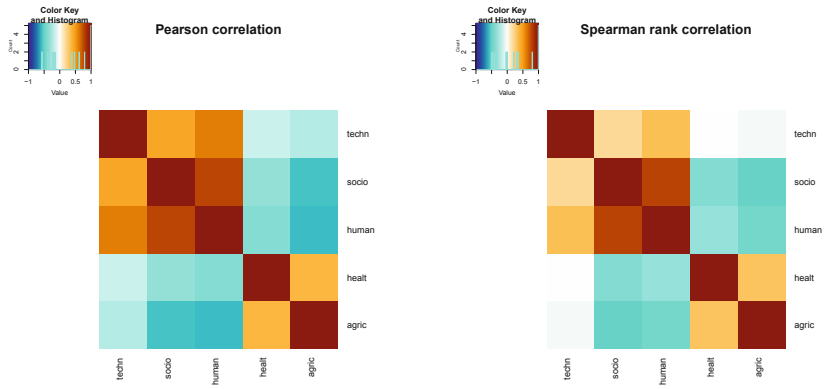


Fig. 8.2 Heatmaps for the Pearson and Spearman rank correlation for the PhD data using symmetric pivot coordinates

human science studies. This means that with respect to the average behavior of the remaining parts in the composition, the dominances in "socio-economic-law" and "human" increase simultaneously. On the contrary, for a negative correlation as between "human" and "agriculture," the respective dominances show a reverse behavior: while it increases for some countries in "human," it decreases for those countries in "agriculture," and vice versa (always with respect to the average behavior of the remaining parts in the composition).

Finally, by ignoring potential effects caused by the spatial dependence between the observations, it is also possible to test for uncorrelatedness. Consider the symmetric pivot coordinates constructed for the parts technical studies and socio-economic and law studies, which are the first two variables in the compositional data set.

```
Z <- pivotCoord(phdred[, c(1,2,3:5)], method = "symm")
```

The null hypothesis is that the relative information of these parts with respect to the remaining parts in the composition is uncorrelated. One can use the classical Pearson correlation for the test, or Spearman's rank correlation. In the first case the null hypothesis would be rejected because of the very small p-value, while in the second case it cannot be rejected. Looking again at Fig. 8.1 (upper left) gives the answer to the different conclusions: France (FR) and probably some

other countries deviate from the otherwise unstructured point cloud, which results in a significantly positive Pearson correlation. The more robust Spearman rank correlation downweights the effect of the outliers.

```
# Test for uncorrelatedness, based on Pearson correlation
cor.test(Z[, 1], Z[, 2], method = "pearson")

##
##  Pearson's product-moment correlation
##
## data:  Z[, 1] and Z[, 2]
## t = 2.7292, df = 26, p-value = 0.01124
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1199266 0.7184781
## sample estimates:
##       cor
## 0.4718903

# Test for uncorrelatedness, based on Spearman rank correlation
cor.test(Z[, 1], Z[, 2], method = "spearman")

##
##  Spearman's rank correlation rho
##
## data:  Z[, 1] and Z[, 2]
## S = 2904, p-value = 0.2934
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.2052545
```

8.5.2 *Example for Multiple Correlation*

The data set `data(gemas)` from the package **robCompositions** is again considered, and the interest is in associating the annual precipitation with the chemical composition in the agricultural soils. It can be assumed that precipitation has some effect on this composition, although this effect might not be too strong. Note that the variable "annual precipitation" is not a compositional part, and thus it is not necessary to compute any coordinate. However, since this variable is characterized by relative scale, its log-transformed version is used (Mateu-Figueras and Pawlowsky-Glahn 2008). The composition with the chemical element concentrations is treated as usual. This information needs to be expressed in ilr coordinates, but the choice of the orthonormal coordinates is not essential in the sense that it would not alter the resulting correlation coefficient.

The following code shows how the squared multiple correlation coefficient is computed, compare with Eq. (8.3).

```
data("gemas")
x <- log(gemas$AnnPrec) # log-transformed annual precipitation
X <- gemas[, 12:29]      # composition of element concentrations
```

```
Z <- pivotCoord(X)       # choose orthonormal coordinates
xZ <- cbind(x, Z)         # joint matrix
xZ.cov <- cov(xZ)        # classical covariance estimation
# compute squared multiple correlation coefficient:
1 - det(xZ.cov) / (det(xZ.cov[-1, -1]) * xZ.cov[1, 1])

## [1] 0.254522
```

The result shows that there seems to be some association between precipitation and "chemistry," but this association is rather weak. From this coefficient alone one cannot say anything about the form of this linear relationship, i.e. how the composition would change with a low or high level of precipitation.

In a next step, the interest is in the robustness of this correlation against data outliers. Since it is difficult to get an overview of the data by means of visualization, it is hard to say if there are outliers or not. In any case, the above coefficient is based on classical estimates of the covariance, and those might be affected by outliers. In the following code the MCD estimator is used to robustly estimate the covariance, and thus the result is a robust squared multiple correlation coefficient.

```
library("robustbase")
xZ.cov <- covMcd(xZ)$cov
# compute squared multiple correlation coefficient:
1 - det(xZ.cov) / (det(xZ.cov[-1, -1]) * xZ.cov[1, 1])

## [1] 0.3644017
```

It can be seen that the robust coefficient is considerably higher than the classical one, which confirms that outliers had a certain effect on the classical estimator.

8.5.3 *Example for Correlation Between Groups of Compositional Parts*

The data set `data(laborForce)` from the package **robCompositions** is used, which reports for 124 countries the percentages of female and male employees, employers, and own-account workers. Thus, the data contain two compositions, one for females and one for males. With correlation analysis, the interest can be in the similarity of both compositions.

```
data("laborForce")
Xf <- laborForce[, c(5,7,3)]   # female data
Xm <- laborForce[, c(6,8,4)]   # male data
```

The compositions are shown in ternary diagrams in Fig. 8.3. Each number corresponds to a certain country. One can see that the proportion of employers is very low in both cases, with the exception of few countries: Sierra Leone (98), Anguilla (3), South Africa (102). However, there are doubts if these data are really correct. A closer look at the data reveals that for African and Asian countries, the proportion of own-account workers is generally much higher than for countries

```
par(mfrow = c(1,2), mar = c(1,4,1,2))
ternaryDiag(Xf, type = "n", grid = FALSE)
    x <- Xf / rowSums(Xf)
    xp <- x[, 2] + x[, 3] / 2
    yp <- x[, 3] * sqrt(3) / 2
    text(xp, yp, 1:nrow(x))
ternaryDiag(Xm, type = "n", grid = FALSE)
    x <- Xm / rowSums(Xm)
    xp <- x[, 2] + x[, 3] / 2
    yp <- x[, 3] * sqrt(3) / 2
    text(xp, yp, 1:nrow(x))
```

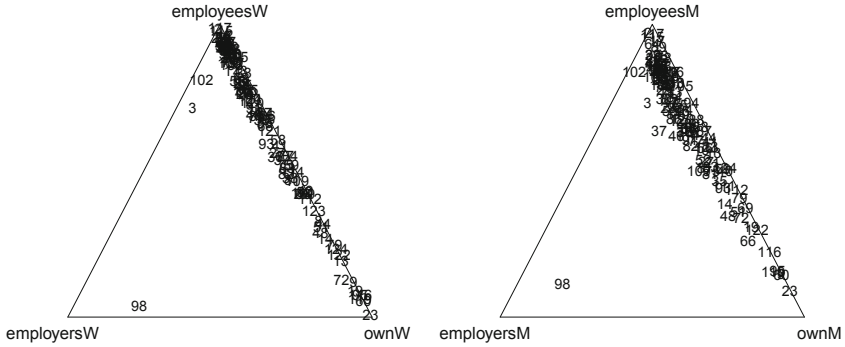


Fig. 8.3 Ternary diagrams for the labor force data; left: compositions for females, right: compositions for males

from other continents. Overall, the data structure for females (left) and males (right) seems to be quite similar.

One may argue that the compositions in both groups are close to the border of the simplex with a substantial effect of relative scale, and thus conclusions made from the ternary diagrams might be misleading, see discussion in Sect. 4.1. Accordingly, the observations were centered (with respect to the Aitchison geometry) to bring them closer to the neutral element and provide a more realistic picture, see Fig. 8.4. Note that the centering can easily be carried out in ilr coordinates, here in pivot coordinates (3.20), although this could also directly be done in the original sample space using the functions `perturbation` and `powering` from the **robCompositions** package. From Fig. 8.4, some minor differences are now clearly visible; particularly a slightly higher proportional representation of the employers in the male group together with possible outliers deviating from the main data cloud due to low proportions in the same part, e.g., Sao Tome and Principe (94) and Suriname (105) in both groups. On the other hand, as the classical (non-robust) center was used for the purpose of centering, some further outliers might still be masked. A definitively true picture of the data structure (in the Euclidean sense) would be obtained only if the data were expressed in orthonormal coordinates.

```
par(mfrow = c(1,2), mar = c(1,4,1,2))
ternaryDiag(Xf, type = "n", grid = FALSE)
#centering
Xfc <- pivotCoordInv(scale(pivotCoord(Xf), scale=FALSE))
x <- Xfc / rowSums(Xfc)
xp <- x[, 2] + x[, 3] / 2
yp <- x[, 3] * sqrt(3) / 2
text(xp, yp, 1:nrow(x))
ternaryDiag(Xm, type = "n", grid = FALSE)
#centering
Xmc <- pivotCoordInv(scale(pivotCoord(Xm), scale=FALSE))
x <- Xmc / rowSums(Xmc)
xp <- x[, 2] + x[, 3] / 2
yp <- x[, 3] * sqrt(3) / 2
text(xp, yp, 1:nrow(x))
```

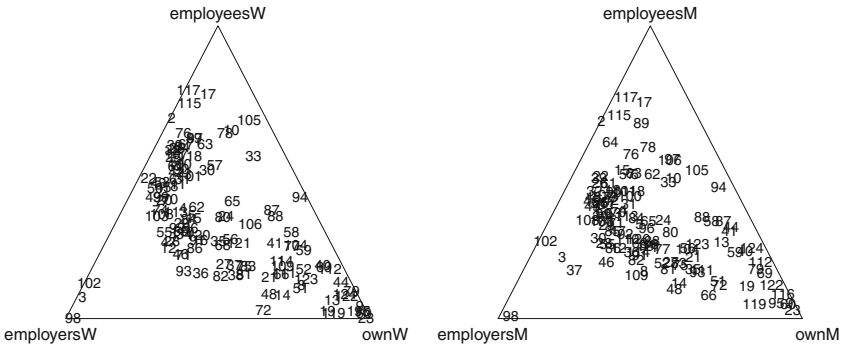


Fig. 8.4 Ternary diagrams for the centered labor force data; left: compositions for females, right: compositions for males

Now the group correlation between the compositions is computed, see Sect. 8.4. Here the classical sample covariance matrix is used to estimate the covariance, but one could also take robust counterparts.

```
Zf <- pivotCoord(Xf)
Zm <- pivotCoord(Xm)
Z <- cbind(Zf, Zm)
Z.cov <- cov(Z)
# compute group correlation coefficient:
1 - det(Z.cov) / (det(Z.cov[1:2, 1:2]) * det(Z.cov[3:4, 3:4]))

## [1] 0.9711786
```

```
cvx1 <- as.matrix(Zf) %*% res$xcoef[,1]
cvy1 <- as.matrix(Zm) %*% res$ycoef[,1]
plot(cvx1, cvy1, type = "n", xlab = "Canonical variate 1 (female)",
    ylab = "Canonical variate 1 (male)")
text(cvx1, cvy1, 1:nrow(Z))
```

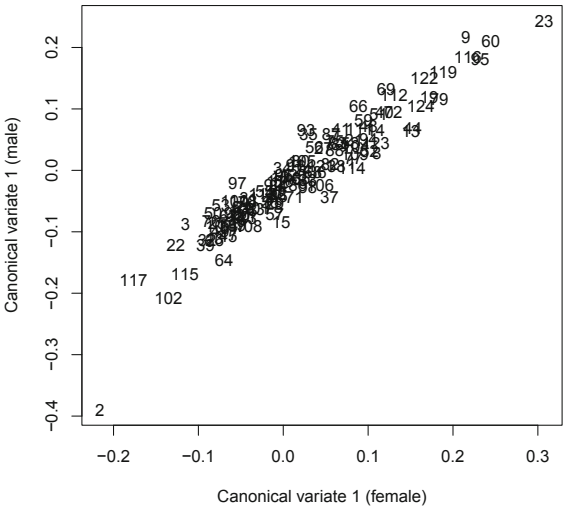


Fig. 8.5 Plot of the first canonical variates for the female and male labor force data. A strong linear trend is visible, resulting in a first canonical correlation coefficient of about 0.95

The resulting group correlation coefficient is 0.97, which is very close to 1, and thus it indicates high linear association between the two compositions.

As mentioned in Sect. 8.4, one could also use canonical correlation analysis to associate both compositions. This can be done as follows:

```
res <- cancor(Zf, Zm)
res$cor # canonical correlation coefficients

## [1] 0.9467533 0.8496806
```

Not only the first but also the second canonical correlation coefficient are very high, which confirms the strong linear association. The first canonical variates can also be shown visually, see Fig. 8.5, and their correlation is in fact equal to the first canonical correlation. One can see the strong linear trend for most observations. An exception is observation number 2 (American Samoa), where again the data might not be fully reliable.

References

T.W. Anderson, *An Introduction to Multivariate Statistical Analysis* (Wiley, Chichester, 2003)
F. Chayes, On correlation between variables of constant sum. J. Geophys. Res. **65**(12), 4185–4193 (1960)
P. Filzmoser, K. Hron, Correlation analysis for compositional data. Math. Geosci. **41**(8), 905–919 (2009)
R.A. Johnson, D.W. Wichern, *Applied Multivariate Statistical Analysis*, 6th edn. (Prentice Hall, Upper Saddle River, 2007)
P. Kynčlová, P. Filzmoser, K. Hron, Correlation between compositional parts based on symmetric balances. Math. Geosci. **49**(6), 777–796 (2017)
G. Mateu-Figueras, V. Pawlowsky-Glahn, A critical approach to probability laws in geochemistry. Math. Geosci. **40**(5), 489–502 (2008)
K. Pearson, Mathematical contributions to the theory of evolution. On a form of spurious correlation which may arise when indices are used in the measurement of organs. Proc. R. Soc. Lond. **60**, 489–502 (1897)

Chapter 9
Discriminant Analysis

Abstract In the setting of discriminant analysis it is assumed that the so-called training data belong to certain groups. The goal is to find classification rules that allow to assign new test data to one of the groups. Different discriminant methods have been introduced, such as linear discriminant analysis (LDA), with Fisher's method as a specific approach, and quadratic discriminant analysis (QDA). Both LDA and QDA utilize the information on prior class probabilities and heavily use the assumption of normality in ilr coordinates (normal distribution on the simplex) to represent group distributions. While for QDA individual group covariance matrices are assumed, a joint covariance matrix is computed for the case of LDA. These methods result in classification rules that allow to assign a new test set observation to one of the groups by taking the prior information on class pertinence into account. The Fisher discriminant rule aims for the same goal, but now no underlying distributions of the samples in the groups are assumed and the idea is to search for projection directions which allow for a maximum separation of the group means with respect to the spread of the projected data. As a consequence, also discriminant scores can be derived that are used to visualize relevant information for the group separation. All described procedures are invariant with respect to the choice of the orthonormal coordinates, and this also holds for the robust counterparts of the covariance-based methods if an affine equivariant location and covariance estimator (like the MCD estimator) is taken.

9.1 Introductory Remarks

Discriminant analysis goes back to the work of Fisher (1936), and it can be considered as one of the traditional methods for classification. Classification—in contrast to clustering—assumes prior knowledge of class memberships in addition to the multivariate data. Typically, this information is available for a training data set, and only the multivariate information without class membership information is available for a test data set. The task of discriminant analysis is to predict the class memberships for the test set observations. The prediction is based on a discriminant rule which is obtained through the training data. There are various ways to establish

P. Filzmoser et al., *Applied Compositional Data Analysis*, Springer Series in Statistics, https://doi.org/10.1007/978-3-319-96422-5_9

a rule; the most prominent rules are the *Bayes* and the *Fisher* rule (see, e.g., Johnson and Wichern 2007). These rules build on different assumptions; while the Bayes rule requires a specification of the distribution underlying the data, this is not explicitly required for the Fisher rule. However, only under certain assumptions one will obtain "optimality" of the rule in the sense of a minimal misclassification error (for the training data). An observation is misclassified if the true group label and the predicted group label are not identical.

Suppose that n observations are given from a training data set; these are compositions with D compositional parts. Moreover, the n observations originate from $g \geq 2$ different groups, and the sample sizes of the groups are $n_1, \ldots, n_g$, clearly with $n_1 + \ldots + n_g = n$. In order to distinguish the observations from the different groups, the notation $\mathbf{x}_{ij}$ will be used, which is the column vector of the compositional information of the ith observation, where $i = 1, \ldots, n_j$, for the jth group, with $j = 1, \ldots, g$.

Assume that the g groups originate from g underlying populations $\pi_1, \ldots, \pi_g$. For example, the groups could refer to different kinds of plants where the samples have been taken from. It is further assumed that the jth population has a certain prior probability p_j, with $p_1 + \ldots + p_g = 1$. Thus, p_j would be the probability that an observation comes from population π_j. If the training data reflect the structure of the populations, it is expected that n_j/n is close to p_j, for all groups.

Since the underlying data are compositions, they first need to be expressed in ilr coordinates, because the discriminant analysis methods are based on the usual Euclidean geometry. Accordingly, the vector of length D for the ith observation from the jth group, $\mathbf{x}_{ij} = (x_{i1}^{[j]}, x_{i2}^{[j]}, \ldots, x_{iD}^{[j]})'$, is expressed in pivot coordinates as shown in Eq. (3.20), by

$$z_{il}^{[j]} = \sqrt{\frac{D-l}{D-l+1}} \ln \frac{x_{il}^{[j]}}{\sqrt[D-l]{\prod_{k=l+1}^{D} x_{ik}^{[j]}}} \quad \text{for } l = 1, \ldots, D-1, \tag{9.1}$$

which results in the column vector $\mathbf{z}_{ij} = (z_{i1}^{[j]}, \ldots, z_{i,D-1}^{[j]})'$.

A further assumption that is usual in discriminant analysis is that the jth population π_j can be characterized by an underlying density function $\mathbf{f}_j$, for $j = 1, \ldots, g$, which is usually assumed to be a multivariate normal density with expectation $\boldsymbol{\mu}_j$ and covariance $\boldsymbol{\Sigma}_j$. In this context, these assumptions are made for the coordinate representation (referring to normal distribution on the simplex, see Sect. 5.1); thus, $\boldsymbol{\mu}_j$ is a vector of length $D-1$, and $\boldsymbol{\Sigma}_j$ is a matrix of dimension $(D-1) \times (D-1)$. Later on, these population quantities need to be estimated from the samples of the training data.

9.2 Bayes Discriminant Rule

The classification problem is considered as follows: Given an observation $\mathbf{z}$, already expressed in ilr coordinates, and the goal is to predict the class label, using a predictor $G(\mathbf{z})$. The result is a number from the discrete set $\{1, \ldots, g\}$. The above definitions imply that the class probability of the kth group is $P(G = k) = p_k$, for $k = 1, \ldots, g$. Now the interest is in the conditional probability that $G = k$, given the information of observation $\mathbf{z}$. This posterior probability is, according to Bayes' theorem, given by

$$P(G = k|\mathbf{z}) = \frac{\mathbf{f}_k(\mathbf{z})\, p_k}{\sum_{j=1}^{g} \mathbf{f}_j(\mathbf{z})\, p_j}. \tag{9.2}$$

The decision boundary between the kth and the lth group is defined by $P(G = k|\mathbf{z}) = P(G = l|\mathbf{z})$. Plugging in for the density $\mathbf{f}_j$ the multivariate normal density with the parameters $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$, for $j = 1, \ldots, g$, leads to the following equality,

$$\ln \frac{P(G = k|\mathbf{z})}{P(G = l|\mathbf{z})} = \delta_k^{QDA}(\mathbf{z}) - \delta_l^{QDA}(\mathbf{z}), \tag{9.3}$$

where

$$\delta_k^{QDA}(\mathbf{z}) = -\frac{1}{2} \ln |\boldsymbol{\Sigma}_k| - \frac{1}{2}(\mathbf{z} - \boldsymbol{\mu}_k)' \boldsymbol{\Sigma}_k^{-1} (\mathbf{z} - \boldsymbol{\mu}_k) + \ln p_k \tag{9.4}$$

is called the kth *quadratic discriminant function*, for $k = 1, \ldots, g$. Thus, at the decision boundary it holds that $\delta_k^{QDA}(\mathbf{z}) = \delta_l^{QDA}(\mathbf{z})$. On the other hand, $\mathbf{z}$ would be assigned to population π_k whenever $\delta_k^{QDA}(\mathbf{z}) > \delta_l^{QDA}(\mathbf{z})$. More generally, $\mathbf{z}$ is assigned to that group for which $\delta_j^{QDA}(\mathbf{z})$ is the largest, for $j = 1 \ldots, g$.

Indeed, $\delta_k^{QDA}(\mathbf{z})$ is quadratic in $\mathbf{z}$ which is the reason for the naming. Using this decision rule, one also talks about *quadratic discriminant analysis*, or simply QDA.

An important simplification of the QDA rule can be made if the assumption $\boldsymbol{\Sigma} := \boldsymbol{\Sigma}_1 = \ldots = \boldsymbol{\Sigma}_g$ holds. In that case, some terms in Eq. (9.4) can be simplified, and one obtains

$$\delta_k^{LDA}(\mathbf{z}) = \mathbf{z}' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k - \frac{1}{2} \boldsymbol{\mu}_k' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k + \ln p_k, \tag{9.5}$$

called the kth *linear discriminant function*, for $k = 1, \ldots, g$. The functions are linear in $\mathbf{z}$, and the resulting rule leads to the name *linear discriminant analysis*, or LDA. Similar as before, $\mathbf{z}$ is assigned to that group for which $\delta_j^{LDA}(\mathbf{z})$ is the largest, for $j = 1 \ldots, g$.

In practice it will be difficult to know if the group covariance matrices can be assumed to be equal. On the other hand, all population quantities, the parameters of

the normal distribution, need to be estimated from the training data. With QDA one ends up with many more parameters to estimate (the individual group covariances) than with LDA (the joint covariance matrix). From this point of view, QDA rather tends to overfitting, while LDA might have a tendency to underfitting. It will thus be important to evaluate the classification rule, e.g. by estimating the misclassification error. This can be done by computing the misclassification error directly for the training observations, when they are classified according to the established rules. A more realistic (and less optimistic) estimation is obtained in a resampling scheme, e.g. by using some cross-validation procedure (see, e.g., Hastie et al. 2009).

The population parameters can be estimated by the group means and empirical covariances of the groups. More precisely, the compositional observations are expressed in ilr coordinates, and $\boldsymbol{\mu}_j$ is estimated by

$$\bar{\mathbf{z}}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} \mathbf{z}_{ij}, \tag{9.6}$$

and $\boldsymbol{\Sigma}_j$ by

$$\mathbf{S}_j = \frac{1}{n_j - 1} \sum_{i=1}^{n_j} (\mathbf{z}_{ij} - \bar{\mathbf{z}}_j)(\mathbf{z}_{ij} - \bar{\mathbf{z}}_j)', \tag{9.7}$$

for $j = 1, \ldots, g$. One can plug in these estimates directly into (9.4) to obtain a decision rule based on the sample level. As mentioned above, p_k can be estimated by n_k/n.

Alternatives to the classical estimators (9.6) and (9.7) are robust estimators, like the *minimum covariance determinant (MCD)* estimator (Rousseeuw 1985; Rousseeuw and Van Driessen 1999), see Sect. 5.2.3. The impact of outliers on the decision rule will be reduced, leading to a robust decision rule.

In case of LDA, the joint covariance matrix $\boldsymbol{\Sigma}$ needs to be estimated. This can be done by centering the observations first with their group center, and then estimate the joint covariance. A classical estimator is

$$\mathbf{S} = \frac{1}{n - g} \sum_{j=1}^{g} \sum_{i=1}^{n_j} (\mathbf{z}_{ij} - \bar{\mathbf{z}}_j)(\mathbf{z}_{ij} - \bar{\mathbf{z}}_j)' = \frac{1}{n - g} \sum_{j=1}^{g} (n_j - 1)\mathbf{S}_j; \tag{9.8}$$

for robust estimation there are different options (see, e.g., Todorov and Filzmoser 2009).

There are some important details, reported in Filzmoser et al. (2012), where it has been shown that the discriminant rules for LDA and QDA are invariant to the choice of the orthonormal coordinates, and that this also holds for the robust counterparts if affine equivariant location and covariance estimators are taken. The MCD estimator fulfills these properties, see Sect. 5.2.3.

9.3 Fisher Discriminant Rule

The discriminant rule of Fisher (1936), and its extension to the multi-group case by Rao (1948), uses the idea of searching for projection directions which allow for a maximum separation of the group means with respect to the spread of the projected data. This has an interesting practical consequence, since the grouping structure can then be visually investigated in the projection. In the two-group case ($g = 2$), one considers a projection direction $\mathbf{a} \in \mathbb{R}^{D-1}$, with $\mathbf{a} \neq \mathbf{0}$. The idea is to project the D-part composition $\mathbf{x}$, expressed by $D-1$ coordinates as $\mathbf{z}$, to a univariate quantity $y = \mathbf{a}'\mathbf{z}$.

If the group means (expectations) are $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, then the projected group means are denoted as $\mu_{1,y} = \mathbf{a}'\boldsymbol{\mu}_1$ and $\mu_{2,y} = \mathbf{a}'\boldsymbol{\mu}_2$. An overall group mean, weighted by the prior probabilities, can also be projected to obtain

$$\mu_y := p_1\mu_{1,y} + p_2\mu_{2,y} = \mathbf{a}'(p_1\boldsymbol{\mu}_1 + p_2\boldsymbol{\mu}_2). \tag{9.9}$$

It is not difficult to see that

$$p_1(\mu_{1,y} - \mu_y)^2 + p_2(\mu_{2,y} - \mu_y)^2 = p_1p_2(\mu_{1,y} - \mu_{2,y})^2$$

holds. The projection direction $\mathbf{a}$ is then taken in such a way that the expression

$$\frac{p_1p_2(\mu_{1,y} - \mu_{2,y})^2}{\sigma_y^2} \tag{9.10}$$

is maximized, where $\sigma_y^2 = \mathbf{a}'\boldsymbol{\Sigma}\mathbf{a}$, and $\boldsymbol{\Sigma}$ denotes the joint group covariance matrix. The direction $\mathbf{a}$ maximizing (9.10) is given by

$$\mathbf{a} = \frac{1}{\sqrt{p_1p_2}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2). \tag{9.11}$$

The term $\sqrt{p_1p_2}$ is not important here since $\mathbf{a}$ is only unique up to scaling.

A new observation $\mathbf{z}$ would then be assigned to the first group if

$$y = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)'\boldsymbol{\Sigma}^{-1}\mathbf{z} \geq \mu_y = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)'\boldsymbol{\Sigma}^{-1}(p_1\boldsymbol{\mu}_1 + p_2\boldsymbol{\mu}_2) + \ln\frac{p_2}{p_1}.$$

The last term was added here in order to adjust the rule for differences in the prior probabilities. One obtains a rule that is closely related (but not identical) to the LDA rule (9.5). In case of equal prior probabilities $p_1 = p_2 = 1/2$, the rules are identical.

It is now straightforward to extend the ideas of the two-group case to multiple groups. The goal is to maximize the expression

$$\frac{\sum_{j=1}^{g} p_j(\mu_{j,y} - \mu_y)^2}{\sigma_y^2}, \tag{9.12}$$

with $\mu_{j,y} = \mathbf{a}'\boldsymbol{\mu}_j$, for $j = 1, \ldots, g$. The solution is given as follows. Define the overall mean as

$$\boldsymbol{\mu} = \sum_{j=1}^{g} p_j \boldsymbol{\mu}_j, \tag{9.13}$$

and

$$\mathbf{B} = \sum_{j=1}^{g} p_j (\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})' \tag{9.14}$$

as the matrix describing the variation *between the groups*. Further, define

$$\mathbf{W} = \sum_{j=1}^{g} p_j \boldsymbol{\Sigma}_j \tag{9.15}$$

as the *within groups* covariance matrix. Maximization problem (9.12) can be expressed as

$$\frac{\mathbf{a}'\mathbf{B}\mathbf{a}}{\mathbf{a}'\mathbf{W}\mathbf{a}} \quad \text{for } \mathbf{a} \in \mathbb{R}^{D-1},\ \mathbf{a} \neq \mathbf{0}. \tag{9.16}$$

The solution of this maximization problem is given by the eigenvectors $\mathbf{a}_1, \ldots, \mathbf{a}_l$ of the matrix $\mathbf{W}^{-1}\mathbf{B}$, which are scaled according to $\mathbf{a}_j{}'\mathbf{W}\mathbf{a}_j = 1$ for $j = 1, \ldots, l$. The number l of strictly positive associated eigenvalues is $l \leq \min\{g - 1, D - 1\}$.

The *Fisher discriminant functions* can now be defined as

$$y_j = \mathbf{a}_j'\mathbf{z} \quad \text{for } j = 1, \ldots, l, \tag{9.17}$$

and they show projections of a new observation $\mathbf{z}$ along the directions $\mathbf{a}_j$. It will be of particular interest to investigate the projections along $\mathbf{a}_1$ and $\mathbf{a}_2$, since bivariate views are easy to inspect, and since the first two directions contain the most relevant information for the group separation. Note that for a three-group problem ($g = 3$), the number of projection directions is at most two, since $l \leq \min\{g - 1, D - 1\}$.

For obtaining a classification rule it is useful to compute the *Fisher discriminant score* for each group. For a new observation $\mathbf{z}$, this score is defined for the kth group as

$$\delta_k^F(\mathbf{z}) = \sum_{j=1}^{l} (y_j - \mu_{k,y_j})^2 - 2\ln p_k = \sum_{j=1}^{l} (\mathbf{a}_j'(\mathbf{z} - \boldsymbol{\mu}_k))^2 - 2\ln p_k, \tag{9.18}$$

for $k = 1, \ldots, g$. Here, $\mu_{k,y_j} = \mathbf{a}_j'\boldsymbol{\mu}_k$, and thus one obtains a measure of deviation of $\mathbf{z}$ from the kth group center in the discriminant space. A new observation $\mathbf{z}$ is

therefore assigned to population π_k if $\delta_k^F(\mathbf{z})$ is the smallest among all discriminant scores $\delta_1^F(\mathbf{z}), \ldots, \delta_g^F(\mathbf{z})$.

The eigenvectors $\mathbf{a}_1, \ldots, \mathbf{a}_l$ can be collected as columns in the matrix $\mathbf{A}$. Then the scores (9.18) can be written as

$$\delta_k^F(\mathbf{z}) = (\mathbf{z} - \boldsymbol{\mu}_k)'\mathbf{A}\mathbf{A}'(\mathbf{z} - \boldsymbol{\mu}_k) - 2\ln p_k, \tag{9.19}$$

which corresponds to a squared Mahalanobis distance in the original space of the coordinates. Note that in the space of the discriminants, which is the space to visualize the problem, one simply works with squared Euclidean distances.

Finally, note that the adjustment with the prior probability $(-2\ln p_k)$ brings the rule closer to the Bayes rule (9.5) that minimizes the total probability of misclassifications. It can be shown that

$$\delta_k^F(\mathbf{z}) = \mathbf{z}'\boldsymbol{\Sigma}^{-1}\mathbf{z} - 2\delta_k^{LDA}(\mathbf{z}), \tag{9.20}$$

see Johnson and Wichern (2007).

For a practical application of the Fisher rule, the population parameters need to be estimated. Similar to the Bayes rule, one can use the classical estimators, i.e. the arithmetic group means to estimate $\boldsymbol{\mu}_j$, and the group sample covariances to estimate $\boldsymbol{\Sigma}_j$, for $j = 1, \ldots, g$. This allows to estimate the matrices $\mathbf{B}$ from (9.14) and $\mathbf{W}$ from (9.15). For robust estimation one can use the MCD estimator, as it has been done for the Bayes rule, but there are also several other options. For details, see Filzmoser et al. (2006).

Note that, like for the Bayes rule, also the Fisher rule is invariant with respect to the choice of orthonormal coordinates. Moreover, also a robustified Fisher rule is invariant if affine equivariant estimators for location and covariance are used, see Filzmoser et al. (2012).

9.4 Examples

9.4.1 Example for LDA and QDA

The R package **rrcov** contains the data set *fish* with different body measurements of 159 fish, which are classified into seven different species: Bream (1), Whitewish (2), Roach (3), Parkki (4), Smelt (5), Pike (6), Perch (7). For illustrative purposes, only three variables are used: length from the nose to the beginning of the tail, length from the nose to the notch of the tail, and length from the nose to the end of the tail. When considering these data as compositional, one is only interested in analyzing the logratios between the variables. In the following, the data are prepared and pivot coordinates are computed.

```
plot(Z, col = fish$Species, pch = fish$Species,
     xlab = expression(z[1]), ylab = expression(z[2]))
legend("topleft", legend = 1:7, col = 1:7, pch = 1:7)
```

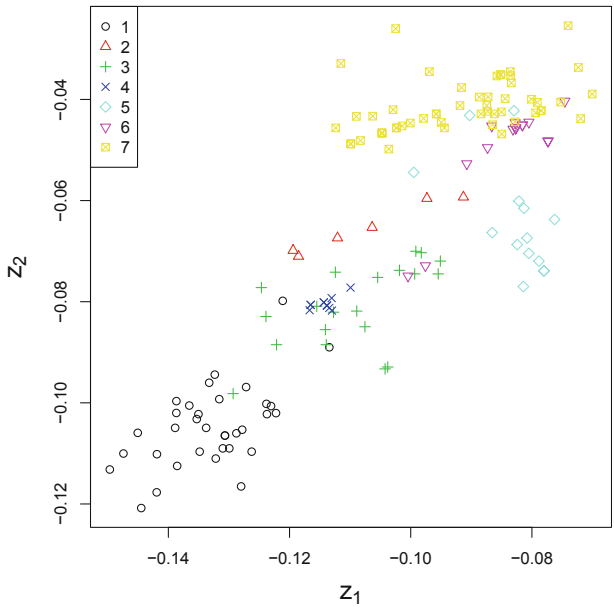


Fig. 9.1 Body measurements from different fish species. The species are shown in this coordinate presentation by different colors

```
library("rrcov")
data("fish")
fish <- fish[-14, ]             # remove observation with missing value
Z <- pivotCoord(fish[, 2:4])  # pivot coordinates for selected 3 variables
table(fish$Species)           # grouping variable

##
##  1  2  3  4  5  6  7
## 34  6 20 11 14 17 56
```

Since only three compositional parts are considered, one obtains two coordinates, which are represented in Fig. 9.1. The observations from the different groups are visualized by different colors.

The data in the different groups look quite heterogeneous, and it is questionable whether the assumption of equal group covariances is valid. Nevertheless, in a first attempt LDA is applied. The function `LdaClassic` from the package **rrcov** is used, but there are also other options, such as the function `lda` from the package **MASS**.

```
library("rrcov")
resLDA <- LdaClassic(Z, fish$Species)
predict(resLDA)

##
## Apparent error rate 0.2342
```

```
##
## Classification table
##       Predicted
## Actual  1  2  3  4  5  6  7
##      1 32  0  1  1  0  0  0
##      2  0  4  0  0  0  2  0
##      3  1  0 17  2  0  0  0
##      4  0  0 11  0  0  0  0
##      5  0  0  0  0 11  0  3
##      6  0  0  2  0  0  3 12
##      7  0  0  0  0  0  2 54
##
## Confusion matrix
##       Predicted
## Actual     1     2     3     4     5     6     7
##      1 0.941 0.000 0.029 0.029 0.000 0.000 0.000
##      2 0.000 0.667 0.000 0.000 0.000 0.333 0.000
##      3 0.050 0.000 0.850 0.100 0.000 0.000 0.000
##      4 0.000 0.000 1.000 0.000 0.000 0.000 0.000
##      5 0.000 0.000 0.000 0.000 0.786 0.000 0.214
##      6 0.000 0.000 0.118 0.000 0.000 0.176 0.706
##      7 0.000 0.000 0.000 0.000 0.000 0.036 0.964
```

The function `LdaClassic` calculates the LDA rules for the data set, while the `predict` command computes the error rate for the same data. Therefore, this error rate is called "apparent error rate," and because the data were not split into training and test data, this estimate of the error rate is usually too optimistic. The underlying classification table and the confusion matrix, where absolute frequencies from the classification table are replaced by relative ones, compare the true class membership (rows) with the predicted classes (columns). For some groups, the discrimination works well, for other groups not. Figure 9.2 shows the outcome of the resulting LDA rules by differently colored dots. Basically, each dot could be considered as a new test set observation, and the color indicates the predicted class membership. Naturally, the decision boundaries are linear, and here it is immediate for which groups the classification works well, while other groups contain outliers or show some overlap.

Now turn to the case of QDA, which is appropriate if the assumption of equal group covariances cannot be made. Note, however, that multivariate normal distribution of the different groups is still assumed. QDA from the package **rrcov** is performed as follows:

```
library("rrcov")
resQDA <- QdaClassic(Z, fish$Species)
predict(resQDA)

##
## Apparent error rate 0.1139
##
## Classification table
##       Predicted
## Actual  1  2  3  4  5  6  7
##      1 32  0  2  0  0  0  0
##      2  0  5  0  0  0  1  0
##      3  1  0 18  1  0  0  0
```

```
##       4  0  0  1 10  0  0  0
##       5  0  0  0  0 11  0  3
##       6  0  0  2  0  0 13  2
##       7  0  0  0  0  0  5 51
##
## Confusion matrix
##        Predicted
## Actual     1     2     3     4     5     6     7
##      1 0.941 0.000 0.059 0.000 0.000 0.000 0.000
##      2 0.000 0.833 0.000 0.000 0.000 0.167 0.000
##      3 0.050 0.000 0.900 0.050 0.000 0.000 0.000
##      4 0.000 0.000 0.091 0.909 0.000 0.000 0.000
##      5 0.000 0.000 0.000 0.000 0.786 0.000 0.214
##      6 0.000 0.000 0.118 0.000 0.000 0.765 0.118
##      7 0.000 0.000 0.000 0.000 0.000 0.089 0.911
```

The resulting apparent error rate is much lower than for LDA. Note, however, that this is again evaluated for the same data which are used to compute the classification rules, and thus possibly too optimistic. For a more realistic evaluation, one should consult some form of cross-validation.

Figure 9.3 shows the decision boundaries, which are no longer linear. It could well be that these decision boundaries are too much adjusted to the data at hand, and not necessarily suitable for new test set observations. It can again be seen that some of the training set observations are assigned to the wrong group.

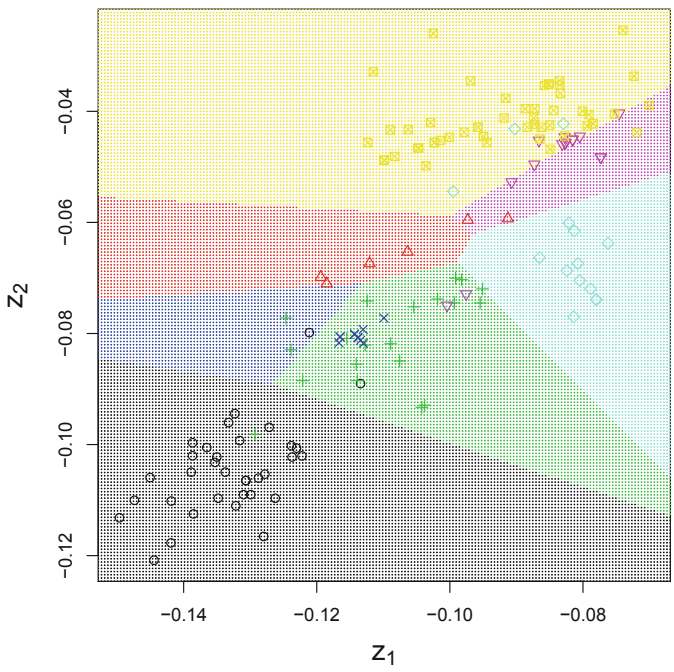


Fig. 9.2 Linear decision boundaries from LDA for the fish data set

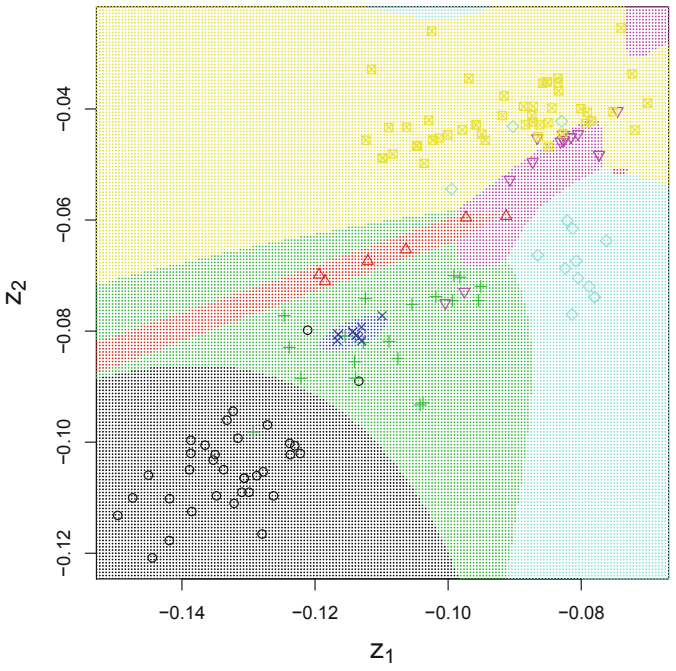


Fig. 9.3 Nonlinear decision boundaries from QDA for the fish data set

It is straightforward to apply robust versions of LDA and QDA, using the package **rrcov**:

```
library("rrcov")
resrLDA <- Linda(Z, fish$Species)   # robust LDA
predict(resrLDA)@ct                 # show classification table

##        Predicted
## Actual  1  2  3  4  5  6  7
##      1 32  0  1  1  0  0  0
##      2  0  6  0  0  0  0  0
##      3  3  0 16  1  0  0  0
##      4  0  0 11  0  0  0  0
##      5  0  0  0  0 11  1  2
##      6  0  0  2  0  0 13  2
##      7  0  0  0  0  0  4 52

resrQDA <- QdaCov(Z, fish$Species) # robust QDA
predict(resrQDA)@ct                # show classification table

##        Predicted
## Actual  1  2  3  4  5  6  7
##      1 32  0  2  0  0  0  0
##      2  0  4  2  0  0  0  0
##      3  1  0 17  2  0  0  0
##      4  0  0  1 10  0  0  0
##      5  0  0  0  0 11  0  3
##      6  0  0  2  0  1 10  4
##      7  0  0  0  0  0  3 53
```

Similar as before, robust QDA works better than robust LDA and there seems not to be a big difference to the performance of the non-robust counterparts.

9.4.2 Example for Fisher Discriminant Analysis

Consider the data set from Oslo with the chemical concentrations in nine different plant species, see also Sect. 4.3. Only a subset of the variables is used, containing those elements which are plant nutrients. For each observation, the knowledge of the corresponding plant species is used as the group membership to construct the discriminant rules. In the following, robust Fisher's LDA for the multi-group case is employed. This will also allow for a visualization of the problem in lower dimension.

Figure 9.4 shows the visualization of the first two Fisher scores, i.e., only the first two eigenvectors $\mathbf{a}_1$ and $\mathbf{a}_2$ are taken to compute the scores in (9.19). This visualization is the two-dimensional plot that best shows the group separation in terms of separation of the group centers. Note that this plot could also be directly generated as an outcome of the routine `daFisher`:

```
res <- daFisher(X, grp, method = "robust", plotScore = TRUE)
```

However, in Fig. 9.4 it is also shown which observations from which groups are misclassified, using filled symbols with the appropriate color. Only very few misclassifications can be seen. However, again the training data for establishing the discriminant rule have been used, and the rules have been employed to classify the same data, thus resulting in a possibly too optimistic misclassification rate.

9.4.3 Example with Appropriate Evaluation of the Error Rate

Only in rare cases it is possible to get training data to build the discriminant rules, and afterwards to collect independent test data to evaluate the rules. Rather, it is common that one data set is available, and it needs to be split into training and test data randomly. In the following, fivefold cross-validation (CV) is employed using stratified sampling. This means that the observations of each group are randomly split into five folds of about equal size, the rule is established on four folds, and evaluated of the fifth fold. This is done in turn, until each fold has been used as test set. In addition, the whole process is replicated 100 times by splitting the data always randomly into five folds. This procedure is implemented in the package **HiDimDA**.

As an illustration, data from the package **classifly** are used, namely the concentrations of fatty acids of olive oils, originating from three different areas in Italy: southern Italy (1), Sardinia (2), and northern Italy (3). As usual, the compositional data are expressed in ilr coordinates. However, some observations contain zeros in some of the variables, and thus, in a first step, these observations are omitted. In

```
X <- OsloTransect[, c("Cu","K","Mg","Mn","P","Sr","Zn")]
grp <- rep(1:9, table(OsloTransect$X.MAT)) # plant materials
isna <- (apply(is.na(X), 1, sum) > 0) # which observations contain NAs
X <- X[!isna, ]
grp <- grp[!isna]
table(grp)              # number of observations in the groups

## grp
##  1  2  3  4  5  6  7  8  9
## 40 40 39 33 38 40 40 40 40

res <- daFisher(X, grp, method = "robust")  # robust Fisher LDA

## Direct agreement: 9 of 9 pairs
## Cases in matched pairs: 97.71 %

res$mcrate              # misclassification rate

## [1] 0.02285714

colv <- rainbow(9)      # generate 9 colors
par(mar = c(4,4,0.1,0.1), cex.lab = 1.4)
plot(res$fdiscr[,1:2], col = colv[grp], pch = grp, xlim = c(-12,13.5),
     xlab = "First Fisher scores", ylab = "Second Fisher scores")
legend("topleft", levels(OsloTransect$X.MAT), pch = unique(grp), col = colv)
points(res$fdiscr[res$grppred != grp, 1:2], pch = 20,
     col = colv[res$grppred][res$grppred != grp])  # misclassified
```

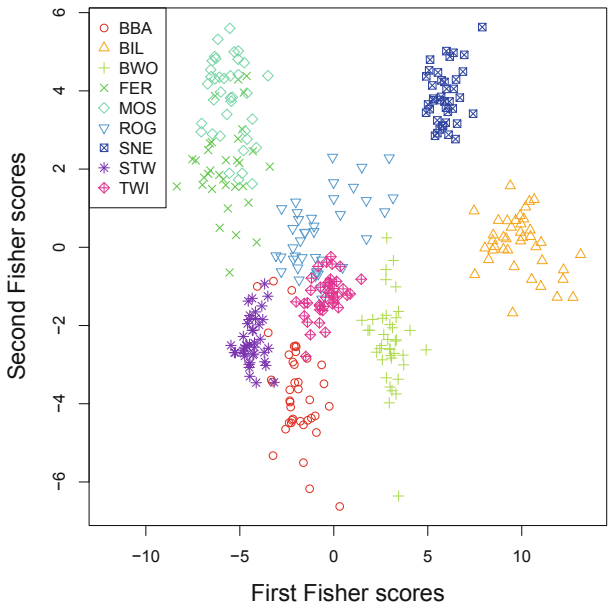


Fig. 9.4 Robust Fisher LDA for the selected data set from Oslo

cases like here, where not many zeros occur, this does not lead to a serious loss of information, contained in nonzero parts of the observations. On the other hand, this is definitely not a systematic approach. Moreover, here the zeros represent a value below the detection limit of the measurement device rather than a pure absence of the property (fatty acid). In such cases it is reasonable to impute each zero by a small nonzero value, being still below the detection limit. This is referred to as *rounded zeros*. Such methods will be discussed in detail in Sect. 13.3.

```
data("olives", package = "classifly")
X <- olives[, -c(1:2)]              # fatty acids
grp <- as.factor(olives[, 1])       # grouping variable
table(grp)

## grp
##   1   2   3
## 323  98 151

notzero <- apply(X == 0, 1, sum) == 0 # to omit zeros
Z1 <- pivotCoord(X[notzero, ])
grp1 <- grp[notzero]
table(grp1)

## grp1
##   1   2   3
## 323  98 114
```

The zeros appear only in the third group. Next, LDA is applied with repeated CV, i.e., five-fold cross-validation is repeated 100 times by using random splits.

```
library("HiDimDA")
set.seed(1234)
res1 <- DACrossVal(Z1, grp1, TrainAlg = lda, kfold = 5, CVrep = 100)
summary(res1[, , "Clerr"])

##        1           2           3
##  Min.   :0   Min.   :0   Min.   :0.00000
##  1st Qu.:0   1st Qu.:0   1st Qu.:0.00000
##  Median :0   Median :0   Median :0.00000
##  Mean   :0   Mean   :0   Mean   :0.00798
##  3rd Qu.:0   3rd Qu.:0   3rd Qu.:0.00000
##  Max.   :0   Max.   :0   Max.   :0.17391
```

The error rates resulting in the different folds and replications are summarized separately for the three groups. The error for the first two groups is zero, and that for the third group is at most 17%.

Now the zeros are imputed by assuming that these are values under the detection limit, using a model-based imputation algorithm, see Sect. 13.3. Only the observations from the third group are used for the imputation.

```
X[X == 0] <- NA
X2 <- impCoda(X[grp == 3, ], method = "lm")$xImp
Z2 <- pivotCoord(rbind(X[grp != 3, ], X2))
```

Finally, LDA is applied again with the above scheme for the evaluation.

```
set.seed(1234)
res2 <- DACrossVal(Z2, grp, TrainAlg = lda, kfold = 5, CVrep = 100)
summary(res2[, , "Clerr"])

##        1           2           3
##  Min.   :0   Min.   :0   Min.   :0.00000
##  1st Qu.:0   1st Qu.:0   1st Qu.:0.00000
##  Median :0   Median :0   Median :0.03226
##  Mean   :0   Mean   :0   Mean   :0.02851
##  3rd Qu.:0   3rd Qu.:0   3rd Qu.:0.03333
##  Max.   :0   Max.   :0   Max.   :0.20000
```

The error rates of the first two groups are still zero, while that for the third group slightly increased. However, at this point one cannot say if this is due to the imputation, or if the observations with zeros are more difficult to classify.

The three groups in the olive oil data contain subgroups: group 1 consists of samples from North and South Apulia, Calabria, and Sicily, group 2 has observations from inland and costal Sardinia, and group 3 is subdivided into Umbria, East and West Liguria. With these nine groups, LDA with repeated CV is again performed.

```
grp3 <- as.factor(olives[, 2])
table(grp3)

## grp3
##        Calabria  Coast-Sardinia    East-Liguria
##              56              33              50
## Inland-Sardinia    North-Apulia          Sicily
##              65              25              36
##    South-Apulia          Umbria    West-Liguria
##             206              51              50

set.seed(1234)
res3 <- DACrossVal(Z2, grp3, TrainAlg = lda, kfold = 5, CVrep = 100)
summary(res3[, , "Clerr"])[4, ]

##            Calabria      Coast-Sardinia        East-Liguria
## "Mean   :0.07355  " "Mean   :0.09571  "  "Mean   :0.1038  "
##     Inland-Sardinia        North-Apulia             Sicily
## "Mean   :0.01400  "  "Mean   :0.0752  "  "Mean   :0.5446  "
##        South-Apulia             Umbria        West-Liguria
## "Mean   :0.01530  " "Mean   :0.07987  "   "Mean   :0.022  "
```

The average group misclassifications are up to about 10%, with the exception of Sicily, where it is more than 50%. One can now compare with the apparent error rates from LDA.

```
res4 <- LdaClassic(Z2, grp3)
1 - diag(predict(res4)@ct) / table(grp3)

## grp3
##        Calabria  Coast-Sardinia    East-Liguria
##      0.05357143      0.09090909      0.10000000
## Inland-Sardinia    North-Apulia          Sicily
##      0.00000000      0.08000000      0.47222222
##    South-Apulia          Umbria    West-Liguria
##      0.01456311      0.05882353      0.02000000
```

```
par(mar = c(5,7,0.1,0.1), cex.lab = 1.3)
boxplot(res3[, , "Clerr"], xlab = "Error rate", horizontal = TRUE, las = 1)
```

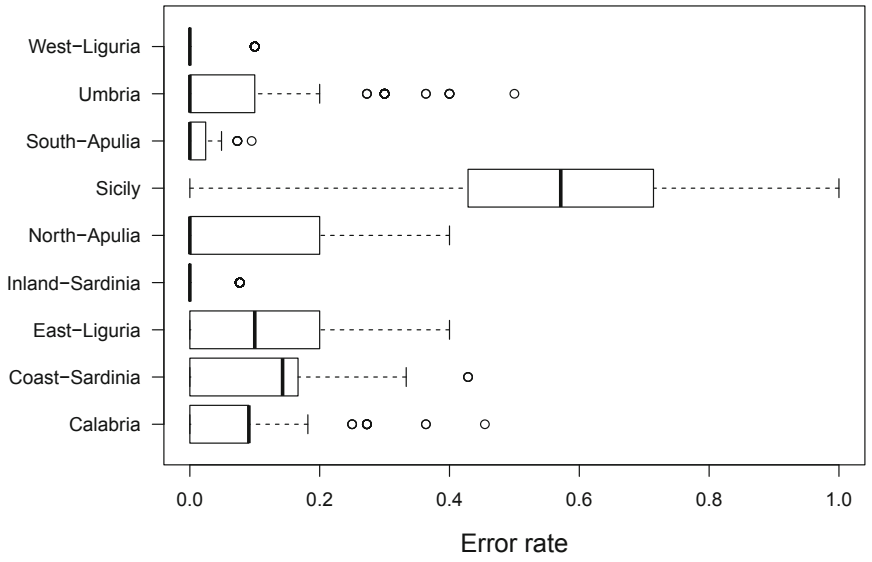


Fig. 9.5 Boxplots of the LDA error rates from repeated CV for the nine groups of the olive oil data

In this case, the error rates resulting from the training data are quite comparable and obviously not too optimistic. However, this is not necessarily always the case. On the other hand, repeated CV does not only give an average error rate, but one can look at all resulting errors in the folds by a boxplot, see Fig. 9.5. Thus one can also get an idea about the distribution of the error rates. Indeed, for Sicily one can see a big variation, and the error rates increase even up to 100%.

Note that almost all zeros are in the group West-Liguria, and none in group Sicily:

```
table(notzero, grp3)
```

```
##        grp3
## notzero Calabria Coast-Sardinia East-Liguria Inland-Sardinia
##   FALSE        0              0            3                0
##   TRUE        56             33           47               65
##        grp3
## notzero North-Apulia Sicily South-Apulia Umbria West-Liguria
##   FALSE            0      0            0      0           34
##   TRUE            25     36          206     51           16
```

References

P. Filzmoser, K. Joossens, C. Croux, Multiple group linear discriminant analysis: robustness and error rate, in *COMPSTAT 2006 - Proceedings in Computational Statistics*, ed. by A. Rizzi, M. Vichi (Physica-Verlag, Heidelberg, 2006), pp. 521–532

P. Filzmoser, K. Hron, M. Templ, Discriminant analysis for compositional data and robust parameter estimation. J. Comput. Stat. **27**(4), 585–604 (2012)

R.A. Fisher, The use of multiple measurements in taxonomic problems. Ann. Eugenics **7**(Part II), 179–188 (1936)

T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*, 2nd edn. (Springer, New York, 2009)

R.A. Johnson, D.W. Wichern, *Applied Multivariate Statistical Analysis*, 6th edn. (Prentice Hall, Upper Saddle River, 2007)

C.R. Rao, The utilization to multiple measurements in problems of biological classification. J. R. Stat. Soc. Ser. B **10**, 159–203 (1948)

P. Rousseeuw, Multivariate estimation with high breakdown point, in *Mathematical Statistics and Applications*, ed. by W. Grossmann, G. Pflug, I. Vincze, W. Wertz (Reidel Publishing Company, Dordrecht, 1985), pp. 283–297

P.J. Rousseeuw, K. Van Driessen, A fast algorithm for the minimum covariance determinant estimator. Technometrics **41**(3), 212–223 (1999)

V. Todorov, P. Filzmoser, An object-oriented framework for robust multivariate analysis. J. Stat. Softw. **32**(1), 1–47 (2009)

Chapter 10
Regression Analysis

Abstract Regression analysis is used to model the relationship between a response variable and one or more explanatory variables (covariates). In the compositional case, the proper choice of logratio coordinates matters, both due to the interpretation of the regression parameters and because of the properties of the regression models. And again, orthonormal coordinates, particularly in their pivot version, are preferable. Moreover, in case of regression with compositional response and real covariates, ilr coordinates enable to decompose the multivariate regression model into single multiple regressions. The coordinate representation of compositions is essential also for statistical inference like hypotheses testing, which is frequently of interest in the regression context. In this chapter, all basic regression cases are contained: the mentioned regression with compositional response and real covariates, the case of real response and compositional explanatory variables, regression between two compositions, and finally also regression between the parts within one composition. A further important task is considered: variable selection of relevant covariates by forward and backward selection. Robustness issues are also of particular importance in the regression context—outliers in the response or in the covariates will have limited effect for robust regression estimates.

10.1 Introductory Remarks

The aim of regression modeling is to analyze the functional relationship between the independent (explanatory) variable(s), also called covariate(s), and the response (dependent) variable(s) (Johnson and Wichern 2007). The results are primarily used for the following purposes:

- To quantify the change of the response with changes in each of the covariates.
- To forecast or predict the value of the response based on the values of the covariates.

P. Filzmoser et al., *Applied Compositional Data Analysis*, Springer Series in Statistics, https://doi.org/10.1007/978-3-319-96422-5_10

In the context of compositional data, one can distinguish four main cases:

- The dependent variables are compositional and the covariates are real (non-compositional) values. This case also includes categorical explanatory variables that lead to analysis of variance (ANOVA) models.
- The response consists of real variable(s), and the explanatory variables are compositional.
- Both the response and the covariates are of compositional nature.
- Regression analysis has to be performed between compositional parts, or between two groups of variables within one composition.

Although the above regression models might be developed directly for the original compositions with respect to the Aitchison geometry, the main interest will be devoted to their ilr coordinate representations that allows to employ standard tools of regression analysis. Consequently, regression with both compositional response and covariates is an immediate result of the previous cases and does not need to be built from scratch. A further important task is variable selection, i.e. the selection of covariates which are relevant for the explanation of the response(s); this is briefly discussed in Sect. 10.5.

It is popular in regression analysis to assume that the explanatory variables are non-random, while the response is supposed to contain random effects. Although this assumption is kept in the following, the authors of the book are aware that in many practical situations it might become too restrictive. A proper alternative in such cases would be to employ orthogonal regression (Fuller 1987) that deals with errors in both dependent and independent variables. On the other hand, the accompanying statistical inference (hypothesis testing, confidence intervals) in orthogonal regression is just asymptotic unlike for the standard least-squares (LS) method, and it is usually performed indirectly through bootstrap methods. Therefore, this approach is applied only for regression within a composition, where it is simply inevitable.

10.2 Regression with Compositional Response

The case of regression with compositional response models situations, when relative information, carried by a composition $\mathbf{x} = (x_1, \ldots, x_D)'$, is influenced by one or more real (non-compositional) variables $\mathbf{t} = (t_0, t_1, t_2, \ldots, t_r)'$ (Egozcue et al. 2012). Suppose that n samples are available, where the ith record is made of a compositional response $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iD})'$ and the values of r covariates are arranged in a vector $\mathbf{t}_i = (t_{i0}, t_{i1}, t_{i2}, \ldots, t_{ir})'$, where $t_{i0} = 1$ is equal for each record. The multiple regression model within the framework of the Aitchison geometry can be defined as follows:

$$\mathbf{x}_i = \mathbf{b}_0 \oplus \bigoplus_{k=1}^{r} (t_{ik} \odot \mathbf{b}_k) \oplus \mathbf{e}_i \ , \ i = 1, \ldots, n \ , \tag{10.1}$$

where the D-part compositions $\mathbf{e}_i$ form an additive-perturbation error. Moreover, the covariate $t_{i0} = 1$ provides a constant term of the regression model used for the intercept. Similar as in standard regression analysis, the task is to estimate the regression parameters (D-part compositions) $\mathbf{b}_k$, $k = 0, \ldots, r$ in an optimal sense. This is traditionally represented by the least-squares method, where the problem is to find estimates $\widehat{\mathbf{b}}_k$ minimizing the sum of square-norms of the perturbation-difference between the observed and the predicted values of the response (residual sum of squares, RSS),

$$\text{RSS} = \sum_{i=1}^{n} \|\mathbf{e}_i\|_A^2 = \sum_{i=1}^{n} \|\mathbf{x}_i \ominus [\mathbf{b}_0 \oplus \bigoplus_{k=0}^{r} (t_{ik} \odot \mathbf{b}_k)]\|_A^2 \,. \tag{10.2}$$

It is worth to note that for the case of a regression model consisting of the absolute term covariate only, the center as a measure of location in compositional data (see Sect. 4.1) would be obtained.

The least-squares problem can be efficiently solved by expressing the compositional responses in orthonormal coordinates, preferably directly in proper balances (3.37). If the coordinate representation of compositions involved in (10.1) is denoted with a tilde (as in (3.37)), the transformed model is

$$\tilde{\mathbf{x}}_i = \tilde{\mathbf{b}}_0 + \sum_{k=1}^{r} (t_{ik} \cdot \tilde{\mathbf{b}}_k) + \tilde{\mathbf{e}}_i \,, \; i = 1, \ldots, n, \tag{10.3}$$

and

$$\text{RSS} = \sum_{i=1}^{n} \|\tilde{\mathbf{e}}_i\|^2 = \sum_{i=1}^{n} \sum_{j=1}^{D-1} (\tilde{e}_{ij})^2. \tag{10.4}$$

Equation (10.4) is a consequence of the isometric character of ilr coordinates: the Aitchison norm of a composition is equal to the ordinary real Euclidean norm of its coordinates. In the expression of RSS (10.4), the order of the sums can be inverted and, being all terms non-negative, the minimization of RSS in coordinates is equivalent to the separate minimization of the $D - 1$ terms

$$\text{RSS}_j = \sum_{i=1}^{n} \tilde{e}_{ij}^2 = \sum_{i=1}^{n} \left(\tilde{x}_{ij} - \sum_{k=0}^{r} t_{ik} \tilde{b}_{kj} \right)^2 , \; j = 1, \ldots, D-1, \tag{10.5}$$

where $\tilde{b}_{kj}$ is the jth coordinate of the compositional coefficient $\mathbf{b}_k$. Comparing (10.4) and (10.5), the Pythagorean decomposition $\sum_{j=1}^{D-1} \text{RSS}_j = \text{RSS}$ is easily obtained. For the jth coordinate, (10.5) implies the ordinary least-squares solution of the real regression model

$$\tilde{x}_{ij} = \sum_{k=0}^{r} t_{ik} \tilde{b}_{kj} + \tilde{e}_{ij} \,, \; i = 1, 2, \ldots, n \,, \tag{10.6}$$

where $\tilde{e}_{ij}$ is the jth coordinate of the compositional error $\mathbf{e}_i$. The interpretation of the regression parameters $\tilde{b}_{kj}$, $k = 1, \ldots, r$, follows the case of standard multiple regression and corresponds to a change in the response based on a one-unit change in the corresponding explanatory variable keeping all other variables fixed.

Also the usual statistical inference on the regression parameters can be performed without any further restrictions. This also holds for the popular T- and F-statistics that are used for significance testing of single parameters and the whole parameter vector (except of the intercept term), respectively.

Equations (10.4) and (10.5) imply that the original compositional least-squares regression problem (10.1), (10.2) is equivalent to $D - 1$ ordinary least-squares problems in orthonormal coordinates. Remarkably, the least-squares problems for the coordinates can be solved independently. Moreover, the prediction capabilities of the regression model are independent of the concrete coordinate representation: although the coordinates of the estimated coefficients $\widehat{\mathbf{b}}_k$ and the residuals $\mathbf{r}_i = \mathbf{x}_i \ominus \widehat{\mathbf{x}}_i$, where

$$\widehat{\mathbf{x}}_i = \widehat{\mathbf{b}}_0 \oplus \bigoplus_{k=1}^{r} (t_{ik} \odot \widehat{\mathbf{b}}_k), \tag{10.7}$$

depend on the selected basis, the back-transformed compositional coefficients and residuals do not (Egozcue et al. 2012). For this purpose, e.g., pivot coordinates (3.19) and the respective inverse mapping (3.22) can be used.

Expressing the results of the regression modeling in coordinates back by means of the original compositions might be useful for interpretational purposes by considering the peculiarities of the Aitchison geometry. Nevertheless, any statistical inference concerning the regression parameters needs to be performed in coordinates. Here, the choice of interpretable balances for the compositional response becomes of primary importance. For example, by taking pivot coordinates (3.25), it is possible to aggregate pairwise logratios with a part of interest into one single coordinate $z_1^{(l)}$, $l = 1, \ldots, D$.

Along with the above developments, it is possible to consider D regression models, where the response is formed by the respective coordinate $z_1^{(l)}$. The lth model can be formulated as follows:

$$z_{i1}^{(l)} = b_0^{(l)} + t_{i1} b_1^{(l)} + \cdots + t_{ir} b_r^{(l)} + e_i^{(l)}, \; i = 1, \ldots, n. \tag{10.8}$$

The situation differs from the above case, when one multivariate regression model was decomposed into $D - 1$ simple multiple regressions. Now from each such decomposition, based on pivot coordinates in their sample version (3.26), always only one of the models is taken for the further considerations.

In this case, the interpretation of the regression parameters gets linked to the single original compositional parts. For example, if $t_2, \ldots, t_r$ are fixed, then for each change of one unit in t_1, the response $z_1^{(l)}$ changes by $b_1^{(l)}$ units on average (by fixed values of the other covariates). Nevertheless, as the orthonormal coordinates (3.25)

have to be interpreted in terms of *scaled* logratios under the natural logarithm, the interpretation of these "units" and thus also of the values of the regression parameters might still get rather complex for practical purposes. For this reason, Müller et al. (2018) proposed to switch to easier interpretable *orthogonal* coordinates. In other words, the aim is to suppress the scaling of the coordinates and replace the rather strange natural logarithm by another base (such as the decadic or binary logarithm) that is easier to handle for the application at hand, see also Remark 2 in Sect. 3.3.4. By doing so, nothing from the above properties of the regression modeling in coordinates is lost, while at the same time a substantial simplification for the interpretation of the parameters is gained. In particular, the values of the T- and F-statistics under the assumption of normality of the errors remain unchanged, as well as the decomposition features of regression with compositional response (Egozcue et al. 2012). Following (3.25), these considerations lead to orthogonal (pivot) coordinates

$$z_i^{(l)*} = \log_2 \frac{x_i^{(l)}}{\sqrt[D-i]{\prod_{j=i+1}^{D} x_j^{(l)}}}, \; i = 1, \ldots, D-1, \tag{10.9}$$

for $l = 1, \ldots, D$, where the normalizing constants are omitted and the original natural logarithm is replaced by the binary one. This results in regression models

$$z_{i1}^{(l)*} = b_0^{(l)*} + t_{i1} b_1^{(l)*} + \cdots + t_{ir} b_r^{(l)*} + e_i^{(l)*}, \; i = 1, \ldots, n, \tag{10.10}$$

for $l = 1, \ldots, D$. From the properties of LS estimation and the relation between logarithms of different bases it is possible to get a straightforward relation between the regression parameters of the models (10.8) and (10.10),

$$b_j^{(l)*} = \log_2(\mathrm{e}) \sqrt{\frac{D}{D-1}} b_j^{(l)}, \; j = 0, \ldots, r.$$

Accordingly, $b_j^{(l)*}$ is the additive increment of the logratio response $z_1^{(l)*}$ when adding one to an explanatory variable t_j, $j = 1, \ldots, r$, (at constant values of the other covariates)

$$b_j^{(l)*} = \Delta z_1^{(l)*} = \log_2 \frac{x_1^{(l)}}{\sqrt[D-1]{\prod_{i=2}^{D} x_i^{(l)}}} \delta - \log_2 \frac{x_1^{(l)}}{\sqrt[D-1]{\prod_{i=2}^{D} x_i^{(l)}}} = \log_2 \delta,$$

where $\delta = 2^{b_j^{(l)*}}$ is the multiplicative increase in the relative dominance of the original compositional response x_l. So, for a unit additive change in t_j, the ratio of $x_1^{(l)}$ to the "average representative" of the other compositional responses grows $\delta = 2^{b_j^{(l)*}}$ times.

10.3 Regression with Compositional Covariates

For regression with compositional response it was very instructive to introduce the model in terms of the Aitchison geometry, because this helped to reveal the necessity of taking orthonormal coordinates in order to decompose the multivariate model into simple multiple regressions. In the case of regression with compositional explanatory variables, just the coordinate representation seems to be fully sufficient for the practical purpose, although the model might also be expressed for the original compositions together with a corresponding interpretation (Tolosana-Delgado and van den Boogaart 2011; van den Boogaart and Tolosana-Delgado 2013). Regression with compositional covariates corresponds to the well-known experiments with mixtures (Scheffé 1958), where the fixed-sum representations of compositions were considered, thus ignoring the scale invariance property of compositional data (Aitchison 1986).

10.3.1 Real Response

Accordingly, when a D-part composition $\mathbf{x} = (x_1, \ldots, x_D)'$ is expressed in balances (3.37), or even more specifically, in one of D coordinate systems (3.25), it is possible to form a standard multiple regression model that can be used for further estimations by the least-squares method. Some robust alternatives for parameter estimation are discussed in Sect. 10.6. When n measurements of the covariates are taken together with those of the real response variable Y, the resulting models can be written as follows:

$$Y_i = b_0^{(l)} + b_1^{(l)} z_{i1}^{(l)} + \cdots + b_{D-1}^{(l)} z_{i,D-1}^{(l)} + \varepsilon_i,\ i = 1, \ldots, n;\ l = 1, \ldots, D. \qquad (10.11)$$

Using orthogonal transformations between pivot coordinate systems (3.29), it can be shown (Hron et al. 2012) that LS estimates of the parameters $b_0^{(l)} \equiv b_0$ are the same for all $l = 1, \ldots, D$. The same holds for the prediction of the response variable and for further model characteristics. The list includes RSS, given as sum of squared differences between observed and predicted values of the response, the well-known coefficient of determination

$$R^2 = 1 - \frac{RSS}{\sum_{i=1}^{n} (Y_i - \overline{Y})^2} \in [0, 1]$$

($\overline{Y}$ being the arithmetic mean of Y_i, $i = 1, \ldots, n$) indicating the proportion of the variance in the dependent variable that is predictable from the covariates, and the F-statistic

$$F = \frac{R^2}{1 - R^2} \frac{n - D}{D - 1}. \qquad (10.12)$$

The statistic (10.12) follows a Fisher F distribution with $D-1$ and $n-D$ degrees of freedom under the assumption of normality and independence of the errors ε_i, common for all models from (10.11). The reason for such nice properties is again based on a rotation between the orthonormal coordinate systems. In addition to the absolute term parameter, also the parameters $b_1^{(l)}$ are of particular interest due to the interpretation of the first coordinates in (3.25). Because the remaining coordinates $z_2^{(l)}, \ldots, z_{D-1}^{(l)}$ fully represent the remaining parts $x_2^{(l)}, \ldots, x_D^{(l)}$, they cannot be omitted from the model, although the corresponding regression parameters are rather rarely taken for interpretation purposes.

The estimates of the parameters $b_0, b_1^{(1)}, \ldots, b_1^{(D)}$ together with their further characteristics (standard errors, values of the T-statistics and the respective p-values) are usually jointly presented in a table, as they all would result from one regression model. However, it is important to realize that the outputs come from D regression models, therefore, attempts like to construct the F-statistics for all parameters $b_1^{(1)}, \ldots, b_1^{(D)}$ simultaneously would not be reasonable. Note that it would be possible to consider also just one common model instead of D regressions by taking centered logratio coefficients of the explanatory composition $\mathbf{x}$ (Bruno et al. 2015); the regression coefficients would differ only by a constant multiple between the respective coordinates (3.30). Though, such a model cannot be considered in general as a way out of the above problem with joint consideration of the parameters $b_1^{(1)}, \ldots, b_1^{(D)}$. Since the covariates sum up to the constant zero, the LS estimates cannot vary freely, which also affects the interpretability of the model itself (Rao and Mitra 1971; Fišerová et al. 2007).

Particularly for explanatory compositions with a higher number of parts, the computation of the parameter estimates and the corresponding statistical inference in D regression models might become computationally intensive. Then it is possible to use the orthogonal transformation matrix from Eq. (3.29), adapted to the general case describing the relation between two pivot coordinate systems from (3.25). For $\mathbf{z}^{(l)} = (z_1^{(l)}, \ldots, z_{D-1}^{(l)})'$ and $\mathbf{z}^{(k)} = (z_1^{(k)}, \ldots, z_{D-1}^{(k)})'$, $k \neq l$, one gets

$$\mathbf{z}^{(l)} = \mathbf{Q}^{(lk)}\mathbf{z}^{(k)} = (\mathbf{V}^{(l)})'\mathbf{V}^{(k)}\mathbf{z}^{(k)}.$$

Consequently, when the respective regression coefficients in (10.11) are written in vector form, $\mathbf{b}^{(l)} = (b_1^{(l)}, \ldots, b_{D-1}^{(l)})'$ and $\mathbf{b}^{(k)} = (b_1^{(k)}, \ldots, b_{D-1}^{(k)})'$, they are related as

$$\mathbf{b}^{(l)} = \mathbf{Q}^{(lk)}\mathbf{b}^{(k)}. \tag{10.13}$$

Such a strategy will be particularly useful in Chap. 11, when high-dimensional compositions will be analyzed in the context of partial least-squares regression.

Finally, in order to enhance the interpretability of the regression parameters using pivot coordinates (3.25), it is possible to employ orthogonal coordinates $z_1^{(l)*}, \ldots, z_{D-1}^{(l)*}$ (10.9). Accordingly, the model (10.11) is adjusted to

$$Y_i = b_0^{(l)*} + b_1^{(l)*} z_{i1}^{(l)*} + \cdots + b_{D-1}^{(l)*} z_{i,D-1}^{(l)*} + \varepsilon_i, \ i = 1, \ldots, n \tag{10.14}$$

for $l = 1, \ldots, D$. The regression parameters between both models follow the relations

$$b_0^* = b_0, \ b_1^{(l)*} = \ln(2)\sqrt{\frac{D-1}{D}} b_1^{(l)},$$

generally

$$b_i^{(l)*} = \ln(2)\sqrt{\frac{D-i}{D-i+1}} b_i^{(l)}, \ i = 1, \ldots, D-1,$$

and similarly for their estimates and the respective standard errors. The interpretation of the parameters gets simpler now: $b_1^{(l)*}$ stands for an additive increase in the response Y that corresponds to increasing $z_1^{(l)*}$ by one (i.e., increasing the dominance of x_l with respect to the other components twice), while keeping everything else fixed.

10.3.2 Compositional Response

In the previous case of regression with a real response and compositional covariates, as well as for regression with compositional response and real covariates, discussed in Sect. 10.2, the key point was to express the compositional variables in proper orthonormal/orthogonal coordinates before starting with regression modeling. The same strategy can be chosen now, when both the response and the covariates are of compositional nature. Specifically, if pivot coordinates (3.25) are taken for this purpose, only the first coordinates that correspond to both compositions with D_1 and D_2 parts are interpretable. Accordingly, although the full multivariate model is formed with $D_1 - 1$ coordinates $z_1^{1(k)}, \ldots, z_{D_1-1}^{1(k)}$, $k = 1, \ldots, D_1$, for the response and $D_2 - 1$ coordinates $z_1^{2(l)}, \ldots, z_{D_2-1}^{2(l)}$, $l = 1, \ldots, D_2$, for the explanatory composition (plus possibly a parameter for the absolute term), just the regression parameter that corresponds to $z_1^{1(k)}$ and $z_1^{2(l)}$ is considered. In line with the interpretation of (3.25), these relate the dominance of the kth and lth parts within the respective compositions. Together, $D_1 \cdot D_2$ multiple regression models are necessary to cover all possible combinations of the response and explanatory parts (Chen et al. 2017). Particularly for compositions with a higher number of parts this can lead to quite a computational effort. Therefore, alternatively the relation (10.13)

can be used to get the parameter estimates for each of the response coordinates within one regression model.

10.4 Regression Within a Composition

In addition to cases where the composition plays the role of the response, the covariates, or both, also such situations occur where the interest is to find a relation between the parts of a composition. If two groups of parts are considered, it is possible to assign balances to both of them using SBP and then proceed with the regression analysis. However, it seems to be a peculiar problem using the logratio methodology to build up a regression model, if one of the parts is explained by the remaining parts in the composition. The reason is that now at least two parts, the response and the explanatory ones, are of simultaneous interest within the composition. Due to scale invariance it is not possible to fix a concrete representation, say the proportional one, and express simply the response part by subtracting the others from 1. Moreover, the positions of the parts are not equivalent, like in correlation analysis (see Chap. 8) because one of them is explained by the other(s). Regression within a composition is not an exceptional case in practice as it might seem at a glance. For example, in a household survey it might be interesting to see, how relative contributions of services to the overall expenditures are influenced by aliquots of foodstuff, housing and clothing.

For two-part compositions, all relative information is contained in the respective pairwise logratio, the regression problem thus gets trivial here. In the general case of $D \geq 3$, one has to rethink the pivot coordinates (3.25) once again. For this purpose, an additional upper index is introduced,

$$z_i^{(lk)} = \sqrt{\frac{D-i}{D-i+1}} \ln \frac{x_i^{(lk)}}{\sqrt[D-i]{\prod_{j=i+1}^{D} x_j^{(lk)}}}, \ i = 1, \ldots, D-1. \tag{10.15}$$

Here, $(x_1^{(lk)}, \ldots, x_D^{(lk)})'$ stands for a permutation of the parts $(x_1, \ldots, x_D)'$, such that always the lth compositional part fills the first position and the kth part the second one, $(x_l, x_k, x_1, \ldots, x_i, \ldots, x_D)'$, $i \notin \{l, k\}$. In such a configuration, the first pivot coordinate $z_1^{(lk)}$ explains, as usual, all the relative information (logratios) about the original compositional part x_l, the coordinates $z_2^{(lk)}, \ldots, z_{D-1}^{(lk)}$ then explain the remaining logratios in the composition.

Assume that x_l stands for the response and the remaining parts in the actual composition form the explanatory variables. Then the response part is well represented by the coordinate $z_1^{(lk)}$. Further, an appropriate coordinate representation for the explanatory subcomposition $(x_1, \ldots, x_{l-1}, x_{l+1}, \ldots, x_D)'$ needs to be identified. In the above notation, e.g., $z_2^{(lk)}, \ldots, z_{D-1}^{(lk)}$ can serve for this purpose. Like in Sect. 10.3 the problem arises, whether it is possible to treat $D-1$ compositional covariates

simultaneously, represented by the respective coordinates. Unfortunately, this is not the case. The reason is that there would be an overlap of information, conveyed by pairwise logratios, used to construct the resulting coordinates. To see this, a pair of covariates x_k and x_m is considered and the respective pairwise logratios from the explanatory subcomposition are aggregated like in (3.21). Obviously, up to the sign, both of the resulting first pivot coordinates contain $\ln(x_k/x_m)$, so it would not be possible to construct orthonormal coordinates out of that, required for a meaningful and interpretable statistical processing. For a single particular part x_k, however, one can continue to use coordinates (10.15), and the constructed $z_2^{(lk)}$ would exactly fit for the aim of the analysis. It is just not possible to consider both x_k and x_m (or even all covariates) simultaneously in one regression model.

Consequently, in order to analyze the influence of single explanatory parts (or, more precisely, their dominance within the given composition in terms of the respective logratios) to the response, $D-1$ multiple regression models according to the coordinate representations (10.15) need to be constructed. In each of such models, the response is represented by the coordinate $z_1^{(lk)}$ to capture the relative information about x_l. Note that this coordinate is the same for any $k \in \{1,\ldots,D\}$, $k \neq l$. To each of the explanatory parts x_k, $k \neq l$, the coordinates $z_2^{(lk)},\ldots,z_{D-1}^{(lk)}$ are assigned according to the reordered subcomposition $(x_k, x_1,\ldots,x_i,\ldots,x_D)'$, $i \notin \{k,l\}$, $k \in \{1,\ldots,D\}$. Similar as before, the coordinate $z_2^{(lk)}$ explains all the relative information about part x_k in the resulting subcomposition. Considering the range of k, finally $D-1$ regression models are obtained (Hrůzová et al. 2016),

$$z_{i1}^{(lk)} = b_1^{(lk)} + b_2^{(lk)} z_{i2}^{(lk)} + \ldots + b_{D-1}^{(lk)} z_{i,D-1}^{(lk)} + \varepsilon_i,\ i = 1,\ldots,n, \tag{10.16}$$

for $l,k \in \{1,\ldots,D\}$, $l \neq k$ (ε_i stands for the error term resulting from n observations of the response); these models are assigned to single explanatory compositional parts. The interpretation of the above regression models results from the interpretability of pivot coordinates, i.e., in each model just the absolute term parameter and the parameter corresponding to the coordinate $z_2^{(lk)}$ are used for further interpretation and for statistical inference (confidence intervals, hypothesis testing).

Since both the response and the explanatory variables originate from one composition, it cannot be assumed that the covariates represent errorless variables like in the case of a real valued response (Hron et al. 2012). Consequently, the use of an ordinary multiple regression model is inappropriate and can even lead to biased results. Therefore, an orthogonal regression model (or, equivalently, a total least-squares (TLS) model) is applied for this purpose, which is a specific type of errors-in-variable (EIV) model (Fuller 1987). According to Markovsky and Van Huffel (2007), the regression estimates in a total least-squares model (10.16) are obtained using singular value decomposition of a joint matrix of the mean-centered response (i.e., its respective coordinate) and the coordinates of the explanatory composition. For n realizations of both the response and covariates, the joint matrix is given as $[\mathbf{Z}^{(lk)}, \mathbf{z}_1^{(lk)}]$, with $\mathbf{z}_1^{(lk)}$ being the vector of length n that stands

for the corresponding coordinate, and $\mathbf{Z}^{(lk)}$ an $n \times (D-2)$ matrix of coordinates $z_2^{(lk)}, \ldots, z_{D-1}^{(lk)}$. Due to mutual orthogonality of the coordinates (10.15), the SVD of $[\mathbf{Z}^{(lk)}, \mathbf{z}_1^{(lk)}]$ is given as (5.3),

$$[\mathbf{Z}^{(lk)}, \mathbf{z}_1^{(lk)}] = \mathbf{U}\mathbf{D}(\mathbf{W}^{(lk)})'. \tag{10.17}$$

The choice of the coordinates is reflected just in the matrix of the right singular vectors. Further, define the partitions

$$\mathbf{W}^{(lk)} = \begin{bmatrix} \mathbf{W}_{11}^{(lk)} & \mathbf{w}_{12}^{(lk)} \\ \mathbf{w}_{21}^{(lk)} & w_{22}^{(lk)} \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} \\ \mathbf{0} & d_{D-1} \end{bmatrix}, \tag{10.18}$$

where the matrices $\mathbf{W}_{11}^{(lk)}$ and $\mathbf{D}_1 = \mathrm{diag}(d_1, \ldots, d_{D-2})$ are of dimension $(D-2) \times (D-2)$. Then a TLS solution exists iff $w_{22}^{(lk)}$ is non-zero; moreover, it is unique iff $d_{D-2} \neq d_{D-1}$. In this case it is given by

$$\widehat{\mathbf{b}}^{(lk)} = (\widehat{b}_2^{(lk)}, \ldots, \widehat{b}_{D-1}^{(lk)})' = -\mathbf{w}_{12}^{(lk)}/w_{22}^{(lk)} \tag{10.19}$$

and the corresponding TLS error matrix equals to $-\mathbf{U}\mathrm{diag}(\mathbf{0}, d_{D-1})(\mathbf{W}^{(lk)})'$ (Markovsky and Van Huffel 2007), with $\mathbf{0}$ being a vector with $D-2$ zeros. Thus, when a unique solution $\widehat{\mathbf{b}}^{(lk)}$ exists, it is computed from the scaled right singular vector corresponding to the smallest singular value. The absolute term parameter $b_1^{(lk)}$, that equals to zero for mean-centered data, is estimated as

$$\widehat{b}_1^{(lk)} = \frac{(\mathbf{t}^{(lk)})'[(\mathbf{w}_{12}^{(lk)})', w_{22}^{(lk)}]'}{w_{22}^{(lk)}},$$

where $\mathbf{t}^{(lk)}$ stands for means of sampled coordinates $z_2^{(lk)}, \ldots, z_{D-1}^{(lk)}, z_1^{(lk)}$. Note that due to the close relation between SVD and principal component analysis, also an interpretation of the regression estimates in terms of the latter method can be considered (Hrůzová et al. 2016).

The regression analysis using models (10.16) naturally continues with an inference on the regression parameters in order to support the information obtained from their estimates. Particularly, significance testing including the respective p-values and confidence intervals on single regression parameters is frequently of interest in practice. Unfortunately, in case of orthogonal regression, this statistical inference is only possible with strict distributional assumptions, except for the case of three-part compositional data, where an alternative approach using a special linear regression model can be utilized (Fišerová and Hron 2012). Therefore, if those assumptions are not fulfilled, a better strategy is to apply resampling methods. In order to relax the assumptions about the distribution of the input data, in Hrůzová et al. (2016) the nonparametric bootstrap (Davison and Hinkley 1997) was chosen for this purpose. Generally, bootstrapping is based on building a sampling distribution for a statistic

by resampling from the data at hand. Consequently, the nonparametric bootstrap allows to estimate the sampling distribution of a statistic empirically without making assumptions about the distribution of the population, and without deriving the sampling distribution explicitly. The basic idea is that, after drawing a sample of size n from $\mathsf{S} = \{\mathbf{z}_1^{(lk)}, \ldots, \mathbf{z}_n^{(lk)}\}$, $\mathbf{z}_i^{(lk)} = (z_{i1}^{(lk)}, \ldots, z_{i,D-1}^{(lk)})'$, $i = 1, \ldots, n$ with replacement, the sample is considered as a representative sample of the whole population. This means that each element $\mathbf{z}_i$ of S is selected with probability $1/n$ to mimic the original sample S. This procedure is repeated R times, where R is a large number, to obtain a sufficient number of bootstrap samples. For bootstrap confidence intervals, several approaches are available in the literature; in Hrůzová et al. (2016), the simplest percentile intervals and their bias-corrected version with acceleration constant were considered.

Finally, also for regression within a composition, the orthogonal coordinates (10.9) with the notation adjusted according to (10.15) as

$$z_i^{(lk)*} = \log_2 \frac{x_i^{(lk)}}{\sqrt[D-i]{\prod_{j=i+1}^{D} x_j^{(lk)}}}, \; i = 1, \ldots, D-1 \tag{10.20}$$

can be applied to simplify the interpretation of the regression parameters. Their properties with respect to regression models (10.16) themselves and their parameters result as combination of the cases of regression with compositional response and compositional covariates, respectively. Accordingly, a twofold multiplicative increase in the relative dominance of x_k (or equivalently, a unit additive increment in coordinate $z_2^{(lk)*}$) leads to an increase in the relative dominance of the response x_l of $\delta = 2^{b_2^{(l)*}}$. Note that the proportionality coefficient δ stays the same irrespective of the base to which the logarithm was taken, as the factor 2 in the expression now stands for a twofold increase in dominance, and not for the logarithmic base (Müller et al. 2018).

10.5 Variable Selection

Variable selection is intended to select the "best" subset of covariates (predictors). This is a frequent task in applications, where many "candidate" predictors could be included in the regression model. Nevertheless, including them without any deeper consideration could lead to one of the following challenges:

- The aim is to explain the response in the simplest way—redundant predictors should be removed. The principle of Occam's Razor states that among several plausible explanations of a phenomenon, the simplest is the best. Applied to regression analysis, this implies that the smallest model that fits the data is preferable.
- Unnecessary predictors will add noise to the estimation of other quantities of interest.

- Collinearity, i.e. (almost) linear dependence between the covariates, which is caused by having too many variables trying to do the same job. This can distort the inference statistics.
- Cost: if the model is to be used for prediction, one can save time and/or money by not measuring redundant covariates.

Usually, no prior information on which variables to choose is available and thus, a model selection method in order to find a subset of the compositional/real covariates for the regression model under consideration is needed. This might be the case for any of the models introduced in Sects. 10.2–10.4, those from the latter two sections are considered already in a pivot coordinate representation (3.25).

A very intuitive method for finding good models in the set of possible submodels is stepwise variable selection. Here, the common Backward Stepwise (BS) and Forward Stepwise (FS) algorithms are considered. FS starts with a small model, for example with a model containing only a constant, and adds one covariate in each step of the procedure. BS starts with a large model, for example with the full model, and in each step one covariate is removed (see, e.g., Varmuza and Filzmoser 2009).

One possible criterion to select good from bad models is the Akaike Information Criterion—AIC (Akaike 1973), which is frequently used for model selection (Heritier et al. 2009). Given a collection of models for the data, the AIC estimates the prediction quality of each model, relative to each of the other models; the preferred model is the one with the minimum AIC value. The AIC value does not depend on a rotation of either the response or the covariates, and thus any choice of orthonormal coordinates for the representation of the corresponding compositions leads to the same conclusion.

Accordingly, in each step of the BS algorithm such a variable is removed, for which the resulting model has the lowest AIC value. In case of compositional covariates this is done by removing the coordinate $z_1^{(l)}$ for any x_l from the actual step because the other coordinates represent the resulting subcomposition. The algorithm stops when either the AIC value is no longer decreasing, or the minimum of one real covariate (or two explanatory compositional parts, resulting into one pivot coordinate) is reached.

Conversely, in the FS algorithm an explanatory variable is added if this leads to a lower AIC value; in this case such an explanatory variable is added, for which the resulting (richer) model has the lowest AIC value. In the compositional case, this is again done by using the "first" pivot coordinate $z_1^{(l)}$ which stands for a dominance of the newcoming part to the explanatory composition from the current step. The algorithm terminates either when the AIC value no longer decreases, or if the maximum of possible covariates (say D for compositional predictors) is reached. As an initial step, single real covariates are taken, or two-part compositions, resulting again into one real coordinate. Nevertheless, for the compositional case thus $\binom{D}{2}$ models have to be considered, which can lead to computational difficulties when D is large. Therefore, the FS algorithm is recommendable in general rather just for a regression with compositional response and real covariates.

10.6 Robustness Issues

Independent of the regression model considered so far, the regression coefficients need to be estimated using some criterion. The most widely used criterion considered in this context is the least-squares (LS) criterion. Consider in the following the model (10.11), where specific observations $y_1, \ldots, y_n$ are given for the (real) response variable, and the explanatory composition is expressed in a specific ilr coordinate system, leading to a vector $\mathbf{z}_i = (1, z_{i1}, \ldots, z_{i,D-1})'$, for $i = 1, \ldots, n$. The constant 1 is added here for the intercept term. The linear regression model is given by

$$y_i = \mathbf{z}_i' \mathbf{b} + \varepsilon_i \qquad \text{for } i = 1, \ldots, n, \tag{10.21}$$

where $\mathbf{b} = (b_0, b_1, \ldots, b_{D-1})'$ is the vector of regression coefficients, and ε_i represents the error term.

For a given regression estimator $\widehat{\mathbf{b}}$, the ith residual is

$$r_i = r_i(\widehat{\mathbf{b}}) = y_i - \mathbf{z}_i' \widehat{\mathbf{b}}.$$

Regression estimators typically minimize the size of the residuals. The estimated regression coefficients according to the LS criterion are given by

$$\widehat{\mathbf{b}}_{LS} = \underset{\mathbf{b}}{\operatorname{argmin}} \sum_{i=1}^{n} (r_i(\mathbf{b}))^2 , \tag{10.22}$$

where "argmin" refers to minimizing the argument (sum of squared residuals). Outliers can have a strong effect on the minimization of (10.22), because they can yield large (squared) residuals which will dominate the sum. Consequently, the solution $\widehat{\mathbf{b}}_{LS}$ can change, and from a robustness point of view this is not desirable.

Generally, there are two types of outliers in the regression context: outliers in the response, so-called vertical outliers, and outliers in the explanatory variables, so-called leverage points (Maronna et al. 2006). Robust regression should protect against both types.

A formal approach to robust regression started with the M-estimator for regression, defined as

$$\widehat{\mathbf{b}}_M = \underset{\mathbf{b}}{\operatorname{argmin}} \sum_{i=1}^{n} \rho\left(\frac{r_i(\mathbf{b})}{\widehat{\sigma}(\mathbf{b})}\right), \tag{10.23}$$

where $\rho(\cdot)$ is an appropriate function, which might be squared around zero, but bounded for large (absolute) values (Huber 1981). Further, $\widehat{\sigma}(\mathbf{b})$ is the estimated residual scale which depends on the unknown regression coefficients. Here one can already see the difficulty: The regression coefficients can only be estimated if the

residual scale is given, but the residual scale can only be estimated if the regression coefficients are known. The problem can be solved by an iterative algorithm, where it is essential to estimate the residual scale robustly, and to initialize the algorithm with robust starting values. This leads to the so-called MM-estimator which, depending on the tuning parameters, is highly robust and achieves high efficiency (Maronna et al. 2006). This estimator can be computed with the function `lmrob` from the R package **robustbase**.

Also other robust regression estimators are available, such as the LTS (least trimmed sum-of-squares) estimator, implemented as function `ltsReg` from the package **robustbase**. This estimator minimizes a trimmed sum of squared residuals, where the largest values of the squared residuals are trimmed (Rousseeuw 1984). The trimming proportion determines the robustness of the estimator, but also its efficiency. Generally, the LTS regression estimator has a (much) lower efficiency than the MM-estimator (Maronna et al. 2006).

Finally, the MM-estimator is also used to robustify orthogonal regression from Sect. 10.5. Here the link to PCA is employed instead of taking any robust version of SVD; robust PCA is obtained through a robust estimation (MM-estimation) of the covariance matrix (Rousseeuw and Hubert 2013). This has implications also for the statistical inference. Although bootstrap is a very useful tool, in case of robust estimators there are two problems: computational complexity of robust estimators, and the possible instability of the bootstrap in case of many outliers in some of the bootstrap samples. Therefore, for robust orthogonal regression, fast and robust bootstrap (Salibian-Barrera et al. 2006; Van Aelst and Willems 2013) is used which is based on the fact that the robust estimators (specifically the MM-estimator) can be represented by smooth fixed-point equations which allow to calculate a fast approximation of the estimates in each bootstrap sample.

10.7 Examples

In the following, the regression models introduced in Sects. 10.2–10.4 are illustrated with data from applications. Because for an interpretation of the results, concrete values of the regression parameters were mostly not needed, all computations were performed in orthonormal coordinates. Alternatively, of course, also their orthogonal counterparts could be utilized if a more detailed analysis would be preferable. We leave it as an option for an interested reader.

10.7.1 Example for Regression with Compositional Response

In the following, a data set referring to the European Union countries and to some other European countries is used to illustrate the case where the response is a composition and the explanatory variable(s) not. The explanatory variable (only

one) is the GDP per capita from the year 2012. The response composition refers to the financial situation of the households in these countries, where proportions in the categories "Very bad," "Bad," "Moderately bad," "Moderately good," "Good," and "Very good" are reported. This is also how the data in the composition are arranged. The data are available from Eurostat, http://ec.europa.eu/eurostat/.

```
library("robCompositions")
data("GDPsatis")
y <- GDPsatis[, 3:ncol(GDPsatis)]  # compositional parts of the response
x <- GDPsatis[, "gdp"]   # GDP as explanatory variable
```

The response composition is represented using pivot coordinates—but according to Eq. (10.8), this needs to be done for each of the D compositional parts separately. So, D regression models need to be computed. This can be done as follows:

```
# initialize empty list to collect results
allres <- vector("list", ncol(y))
# loop over all compositional parts
for (j in 1:ncol(y)){
  zj <- pivotCoord(y, pivotvar = j)
  # use only first coordinate
  res <- lm(zj[,1] ~ x)
  # result for the first coordinate
  allres[[j]] <- summary(res)
}
```

The object `allres` collects all the results of the summary statistics, but only for the first response pivot coordinate from each multivariate model. One option is to use the R package **broom**, which includes methods (e.g., the functions `tidy` and `glance`) to extract information from statistical models in an elegant way, and the package **data.table** which provides by far the fastest implementation (function `rbindlist`) to combine list elements.

```
library("broom")
library("data.table")
res <- rbindlist(lapply(allres, tidy))
res[, c(1:3, 5)] # selection of estimates

##            term     estimate   std.error      p.value
##  1: (Intercept) -1.047990520 0.231916371 9.638559e-05
##  2:           x -0.002085140 0.002070134 3.221461e-01
##  3: (Intercept) -0.022588977 0.125233365 8.581127e-01
##  4:           x -0.003046581 0.001117859 1.077482e-02
##  5: (Intercept)  0.794525260 0.147754683 8.890763e-06
##  6:           x -0.004229546 0.001318889 3.260439e-03
##  7: (Intercept)  1.494218165 0.138160419 1.083406e-11
##  8:           x -0.001599755 0.001233249 2.047919e-01
##  9: (Intercept)  0.547543470 0.174592964 3.904451e-03
## 10:           x  0.003486685 0.001558453 3.311901e-02
## 11: (Intercept) -1.765707399 0.279875365 6.849721e-07
## 12:           x  0.007474337 0.002498226 5.611997e-03
```

The intercept terms are not relevant here for the interpretation, but the signs of the slope parameters are of interest. For example, the first model refers to all relative information about the part "very bad," and the GDP is assigned a negative coefficient. This means that in countries with smaller GDP there is a "very bad" financial household situation. However, the p-value is around 0.3, and thus GDP is not significant in the model. Significance (at the level 0.05) is obtained only for models 2 ("bad"), 3 ("moderately bad"), 5 ("good"), and 6 ("very good"). The coefficients in the first two cases are negative, those in the last two cases are positive, which also corresponds to the intuition. Note that by using the orthogonal coordinates (see Sect. 10.2) one could further analyze concrete values of the regression coefficients.

One can also have a look at the R^2 measure and at the adjusted R^2, which penalizes for the size of the model, confirming the previous significance results:

```
rbindlist(lapply(allres, glance))[, 1:2]

##     r.squared adj.r.squared
## 1: 0.03380195  0.0004847734
## 2: 0.20390143  0.1764497593
## 3: 0.26178993  0.2363344133
## 4: 0.05484189  0.0222502360
## 5: 0.14719407  0.1177869683
## 6: 0.23586110  0.2095114822
```

10.7.2 Example for Regression with Compositional Covariates and Real Response

The R package **UsingR** contains the data set `fat` with several body measurements that can be used to predict the response variable "body.fat". A prediction model could offer an easy alternative to an underwater weighing technique. Note that there are also other variables like age available, but here only the body measurements are used for modeling. These can be regarded as composition because the "size" of the body is not relevant, only the relations (ratios) between the different body measurements contain the important information.

```
data("fat", package = "UsingR")
fat <- fat[-182, ] # removing a suspicious observation
sel <- c("neck","chest","abdomen","hip","thigh","knee","ankle","bicep",
         "forearm","wrist")
x <- fat[, sel] # explanatory composition
```

The response variable reports the percentage of body fat measured for the males. In order to turn the relative scale into an absolute one, this variable is logit-transformed.

```
y <- log(fat$body.fat / (100 - fat$body.fat)) # logit of fat-%
```

According to Eq. (10.11), D models need to be computed, where D is the number of compositional parts. Since pivot coordinates are used to represent the compositions, only the first coefficient in each model is of interest, together with the corresponding inference statistic. There is a convenient function `lmCoDaX` available in the package **robCompositions** which collects all these first coefficients and the results from the statistical tests in one inference table.

```
rescl <- lmCoDaX(y, x, method = "classical")$ilr
rescl

##
## Call:
## lm(formula = y ~ ., data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.48737 -0.19401  0.00627  0.23868  0.75495
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -4.7762     0.9007  -5.303 2.58e-07 ***
## X.neck       -1.9774     0.6599  -2.996  0.00302 **
## X.chest      -1.2722     0.7318  -1.738  0.08343 .
## X.abdomen     7.2994     0.5041  14.480  < 2e-16 ***
## X.hip        -3.7571     0.9326  -4.029 7.52e-05 ***
## X.thigh       1.0910     0.6395   1.706  0.08932 .
## X.knee       -0.3395     0.7087  -0.479  0.63231
## X.ankle      -0.2015     0.4438  -0.454  0.65015
## X.bicep       0.7868     0.4253   1.850  0.06552 .
## X.forearm     0.3568     0.4397   0.812  0.41785
## X.wrist      -1.9862     0.6993  -2.840  0.00489 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3285 on 240 degrees of freedom
## Multiple R-squared:  0.6835,Adjusted R-squared:  0.6717
## F-statistic: 57.84 on 9 and 241 DF,  p-value: < 2.2e-16
```

In this analysis, classical least-squares regression was used. This function also provides an option to perform robust regression. In order to reproduce the same result, a random seed is fixed.

```
set.seed(123)
res <- lmCoDaX(y, x, method = "robust")$ilr
res

##
## Call:
## ltsReg.formula(formula = y ~ ., data = d)
##
## Residuals (from reweighted LS):
##     Min      1Q  Median      3Q     Max
## -0.6339 -0.1630  0.0000  0.2050  0.6215
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## Intercept  -4.9016     0.7797  -6.286 1.60e-09 ***
```

```
## X.neck      -1.9235     0.5682  -3.385 0.000836 ***
## X.chest     -1.1487     0.6383  -1.800 0.073223 .
## X.abdomen    7.1643     0.4480  15.993  < 2e-16 ***
## X.hip       -3.5483     0.8046  -4.410 1.58e-05 ***
## X.thigh      1.3323     0.5618   2.371 0.018540 *
## X.knee      -0.3408     0.6456  -0.528 0.598046
## X.ankle     -1.2502     0.4682  -2.670 0.008111 **
## X.bicep      0.2747     0.3718   0.739 0.460675
## X.forearm    0.9389     0.3818   2.459 0.014676 *
## X.wrist     -1.6173     0.6046  -2.675 0.008001 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2803 on 234 degrees of freedom
## Multiple R-Squared: 0.7423,Adjusted R-squared: 0.7323
## F-statistic: 74.26 on 9 and 232 DF,  p-value: < 2.2e-16
```

When comparing the results of the inference statistics from classical and robust regression, some differences can be seen: `thigh`, `ankle`, and `forearm` are also significant in the robust version, which seems to be logical in this context. Moreover, the R^2 measure has slightly increased in the robust case, indicating a slightly better model fit of the robust model (to the data majority).

One could inspect various diagnostic plots, but here only the scaled residuals from both approaches are compared, see Fig. 10.1. Only for the classical fit, some

```
qplot(res$residuals / res$sigma, rescl$residuals / rescl$sigma,
  xlab = "scaled residuals (robust fit)",
  ylab = "scaled residuals (classical fit)") +
  geom_abline(intercept = 0, slope = 1)
```

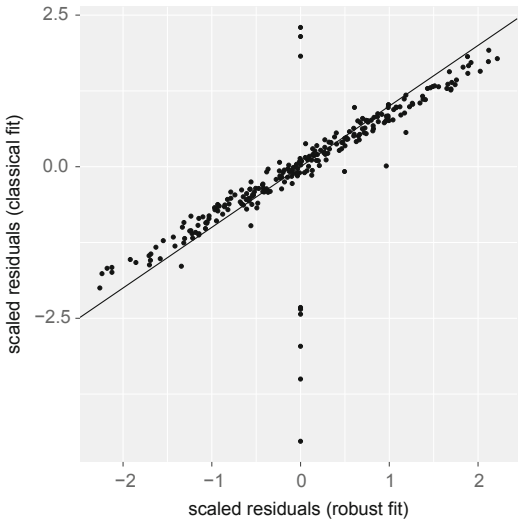


Fig. 10.1 Comparison of the scaled residuals from the classical and robust fit. The line indicates equal values on both axes

very large scaled residuals appear, indicating outliers for this approach. Surprisingly, these observations fit very well to the robust model, and thus one can conclude that classical LS-regression was influenced by them.

10.7.3 Example for Regression with Compositional Covariates and Compositional Response

One question relevant for the European Commission is whether the likelihood of poverty is inherited. Various databases exist to define indicators and instruments related to this question. One such database refers to the education level of father (F) and mother (M). Here, the percentages of low (l), medium (m), and high (h) education levels of father and mother are investigated. The data are available from Eurostat, http://ec.europa.eu/eurostat/, and included as data set `educFM` in the package **robCompositions**. The particular interest is in identifying relationships between the education level of the mother and of the father. This means that two three-part compositions are available and the relationships between them are of interest.

Figure 10.2 shows ternary diagrams of the two compositions. Their structure seems to be similar, but it is not immediate to identify any relationships.

```
data("educFM")
father <- educFM[,2:4]
mother <- educFM[,5:7]
par(mfrow = c(1,2), mar = c(0.1,2,0.1,2))
ternaryDiag(father, text = educFM$country)
ternaryDiag(mother, text = educFM$country)
```

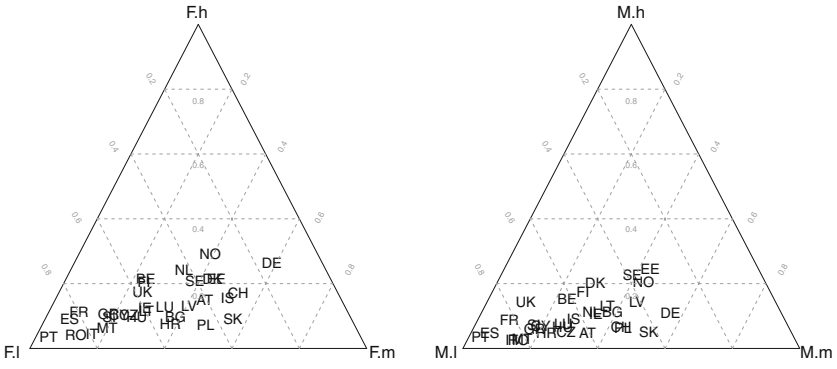


Fig. 10.2 Ternary diagrams of education levels for father (F) and mother (M) in European countries; for father (left plot): low (F.l), medium (F.m), high (F.h); for mother (right plot): low (M.l), medium (M.m), high (M.h)

In order to identify all pairwise relationships, pivot coordinates are constructed for both compositions separately, and LS-regression is performed. Note that because of the orthogonality of the pivot coordinates, multiple regressions instead of a multivariate regression can be carried out. Here the fathers are predicted from the mothers.

```
D <- 3
pval <- coef <- matrix(NA, ncol = D , nrow = D)
dimnames(pval) <- dimnames(coef) <- list(names(father), names(mother))
for (i in 1:D){
  for (j in 1:D){
    zfath <- as.matrix(pivotCoord(cbind(father[, i], father[, -i])))
    zmoth <- as.matrix(pivotCoord(cbind(mother[, j], mother[, -j])))
    res <- summary(lm(zfath[,1] ~ zmoth))$coefficients
    pval[i,j] <- res[2, 4]  # entry of the p-value in the matrix
    coef[i,j] <- res[2, 1]  # entry of the coefficient in the matrix
  }
}
```

The p-values and regression coefficients are summarized below:

```
round(pval, 3)  # all p-values

##     M.l   M.m   M.h
## F.l   0 0.000 0.045
## F.m   0 0.000 0.001
## F.h   0 0.058 0.000

round(coef, 3)  # all regression coefficients

##        M.l    M.m    M.h
## F.l  0.936 -0.671 -0.265
## F.m -0.525  0.862 -0.337
## F.h -0.411 -0.191  0.602
```

Almost all p-values are below 0.05, only the combination F.h–M.m is slightly above. This means that a medium education level of the mother (relative to the other education levels) is only weakly related to high education levels of father (relative to the rest). Also the regression coefficients are shown above. The diagonal of the table relates the same education levels of fathers and mothers, and since all coefficients are positive, there is a positive relation. So, if the mother has a certain education level, it is likely that the father has the same level of education. All other coefficients have negative sign, with a different interpretation. For example, for countries with a dominance of a medium education level of the mother (M.m), there is a deficiency of a low education level of the father (F.l). In other words and simplified, couples with different education levels are more unlikely.

10.7.4 Example for Regression Within a Composition

Coming back to the example from Sect. 10.7.2, one can be interested in the relation between some body measurements. The data set which has been used there contains

such measurements, and the interest here is in the relationship between the variable "abdomen" with the variables "chest," "wrist," and "forearm." It is quite obvious that there will be a relation between "abdomen" and "chest," but other relations are not so clear. Since only one composition is considered here, orthogonal regression needs to be applied for the purpose.

```
data("fat", package = "UsingR")
fat <- fat[-182, ] # removing a suspicious observation
sel <- c("abdomen", "chest", "wrist", "forearm")
x <- fat[, sel]   # considered composition
```

Classical as well as robust orthogonal regression has been implemented in the package **oreg**. Also an appropriate presentation in pivot coordinates is available there. In the following analysis, the part "abdomen" takes the role of the response part, and the remaining parts will be represented by other coordinates, where the second coordinate refers to all relative information of "chest" to the rest (without "abdomen"), and the third coordinate represents the logratio of "wrist" to "forearm."

```
library("oreg")
z1 <- do.ilr(x, ilr.type = "1") # second coordinate represents chest
oo1cl <- oregClassic(z1)
oo1rob <- oregMM(z1)
```

Both the classical and a robust version of orthogonal regression have been computed above, and instead of the usual `summary()`, only the most important results are collected and compared:

```
cbind(estimates = oo1cl$coefficients, oo1cl$coefCIperc, oo1cl$coef.Pval)

##             estimates  95% lower 95% higher
## Intercept  1.0841019  0.8536210  1.4105092 0.000
## Z2         1.4810664  1.3325681  1.7124898 0.000
## Z3        -0.2205848 -0.4629028  0.2163116 0.216

cbind(estimates = oo1rob$coefficients, oo1rob$coefCIperc, oo1rob$coef.Pval)

##             estimates  95% lower 95% higher
## Intercept 1.12646792  0.8415581  1.4721163 0.000
## Z2        1.58322726  1.3617387  1.8666929 0.000
## Z3        0.02771638 -0.6159734  0.7615994 0.992
```

The row "Z2" refers to the coordinate describing the relative information of "chest," and the regression coefficient is positive in both cases, implying a positive relationship to (all relative information of) "abdomen." The last column is the p-value, and in both cases significance is derived. This is also seen by the bootstrap confidence intervals. "Z3," the logratio of "wrist" to "forearm," is not significant for "abdomen," as it was expected.

Figure 10.3 shows some plots related to the robust solution. The left plot illustrates the problem: Since only four parts have been used, the problem can be visualized by the three resulting coordinates. Coordinate "Z1" represents the response, all relative information about "abdomen," "Z2" stands for the relative

```
par(mfrow = c(1,2), mar = c(.1,.1,.1,.1))
oreg::plot3d(oo1rob)

## [1] "Z2" "Z3" "Z1"

par(mar = c(4,4,2,2))
plot(z1[,3], oo1rob$fitted.values,
     xlab = "Measured Z1", ylab = "Estimated Z1")
abline(c(0,1))
```

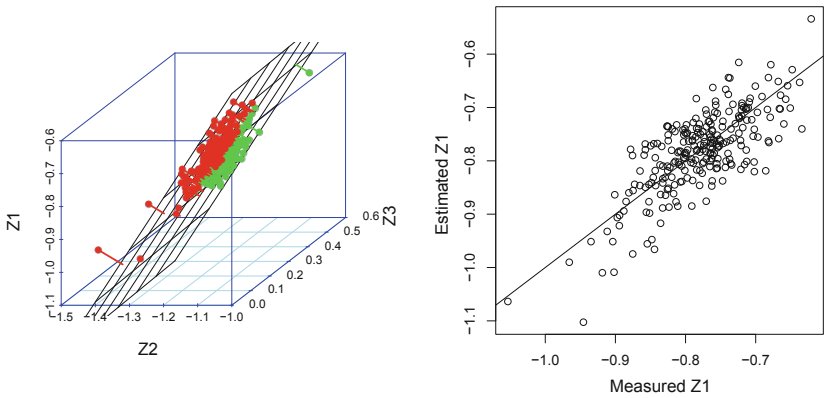


Fig. 10.3 Left: 3D plot of the orthogonal regression fit and the residuals (positive in red, negative in green); right: measured versus estimated variable "Z1"

information about "chest" (except of "abdomen"), and "Z3" contains the information of "wrist" versus "forearm." The regression plane shows the fit, and the residuals are visualized by the red (positive) and green (negative) points with orthogonal lines to the plane. The right plot compares the values of "Z1" and the fitted values, reflecting not an excellent but a moderate fit.

The explanatory variables can be represented in different coordinate systems. In particular, one can use pivot coordinates, where the second coordinate (here denoted by "Z2") represents all relative information about one of the explanatory variables. In the following, this role of "Z2" takes the part "wrist."

```
z2 <- do.ilr(x, ilr.type = "2")  # second coordinate represents wrist
oo2rob <- oregMM(z2)
cbind(estimates = oo2rob$coefficients, oo2rob$coefCIperc, oo2rob$coef.Pval)

##             estimates  95% lower 95% higher
## Intercept  1.1264679  0.8470024  1.4777993 0.002
## Z2        -0.7676105 -1.3144261 -0.1240528 0.036
## Z3         1.3849732  0.9526535  1.8941613 0.002
```

The relative information about "wrist" is still significant in the model, but the coefficient is negative. Thus, for males where the part "abdomen" is more dominant, "wrist" gets less dominant.

A similar conclusion is obtained below, where the role of "Z2" is taken by the part "forearm."

```
z3 <- do.ilr(x, ilr.type = "3")  # second coordinate represents forearm
oo3rob <- oregMM(z3)
cbind(estimates = oo3rob$coefficients, oo3rob$coefCIperc, oo3rob$coef.Pval)

##            estimates  95% lower 95% higher
## Intercept  1.1264679  0.8390529  1.4702319 0.00
## Z2        -0.8156167 -1.6292089 -0.2401641 0.02
## Z3         1.3572568  0.9721390  1.7135905 0.00
```

References

J. Aitchison, *The Statistical Analysis of Compositional Data* (Chapman & Hall, London, 1986); Reprinted in 2003 with additional material by The Blackburn Press)

H. Akaike, Information theory and an extension of the maximum likelihood principle, in *Proceedings of the Second International Symposium on Information Theory*, ed. by B. Petrov, F. Csaki (Akademiai Kiado, Budapest, 1973), pp. 267–281

F. Bruno, F. Greco, M. Ventrucci, Spatio-temporal regression on compositional covariates: modeling vegetation in a gypsum outcrop. Environ. Ecol. Stat. **22**(3), 445–463 (2015)

P. Bühlmann, S. van de Geer, *Statistics for High-Dimensional Data* (Springer, Heidelberg, 2011)

J. Chen, X. Zhang, S. Li, Multiple linear regression with compositional response and covariates. J. Appl. Stat. **44**(12), 2270–2285 (2017)

A. Davison, D. Hinkley, *Bootstrap Methods and Their Application* (Cambridge University Press, Cambridge, 1997)

J.J. Egozcue, J. Daunis-i-Estadella, V. Pawlowsky-Glahn, K. Hron, P. Filzmoser, Simplicial regression. The normal model. J. Appl. Probab. Stat. **6**(1–2), 87–108 (2012)

M.A. Engle, M. Gallo, K.T. Schroeder, N.J. Geboy, J.W. Zupancic, Three-way compositional analysis of water quality monitoring data. Environ. Ecol. Stat. **21**(3), 565–581 (2014)

E. Fišerová, K. Hron, Statistical inference in orthogonal regression for three-part compositional data using a linear model with type-II constraints. Commun. Stat. Theory Methods **41**(13–14), 2367–2385 (2012)

E. Fišerová, L. Kubáček, P. Kunderová, *Linear Regression Models: Reguarity and Singularities* (Academia, Praha, 2007)

W.A. Fuller, *Measurement Error Models* (Wiley, New York, 1987)

M. Gallo, Tucker3 model for compositional data. Commun. Stat. Theory Methods **44**(21), 4441–4453 (2015)

A. Gardlo, A. Smilde, K. Hron, M. Hrdá, R. Karlíková, T. Adam, Normalization techniques for PARAFAC modeling of urine metabolomics data. Metabolomics **12**, 117 (2016)

S. Heritier, E. Cantoni, S. Copt, M.P. Victoria-Feser, *Robust Methods in Biostatistics* (Wiley, Chichester, 2009)

K. Hron, P. Filzmoser, K. Thompson, Linear regression with compositional explanatory variables. J. Appl. Stat. **39**(5), 1115–1128 (2012)

K. Hrůzová, V. Todorov, K. Hron, P. Filzmoser, Classical and robust orthogonal regression between parts of compositional data. Stat. J. Theor. Appl. Stat. **50**(6), 1261–1275 (2016)

P.J. Huber, *Robust Statistics* (Wiley, New York, 1981)

R.A. Johnson, D.W. Wichern, *Applied Multivariate Statistical Analysis*, 6th edn. (Prentice Hall, Upper Saddle River, 2007)

P.M. Kroonenberg, *Applied Multiway Data Analysis* (Wiley, Hoboken, 2008)

I. Markovsky, S. Van Huffel, Overview of total least-squares methods. *Signal Processing* **87**(10), 2283–2302 (2007)
R. Maronna, D. Martin, V. Yohai, *Robust Statistics: Theory and Methods* (Wiley, Chichester, 2006)
C. Mert, P. Filzmoser, K. Hron, Error propagation in compositional data analysis: theoretical and practical considerations. Math. Geosci. **48**(8), 941–961 (2016)
I. Müller, K. Hron, E. Fišerová, J. Šmahaj, P. Cakirpaloglu, J. Vančáková, Interpretation of compositional regression with application to time budget analysis. Austrian J. Stat. **47**(2), 3–19 (2018)
C.R. Rao, S.K. Mitra, *Generalized Inverse of Matrices and Its Applications* (Wiley, New York, 1971)
P.J. Rousseeuw, Least median of squares regression. J. Am. Stat. Assoc. **79**, 871–880 (1984)
P. Rousseeuw, M. Hubert, High-breakdown estimators of multivariate location and scatter, in *Robustness and Complex Data Structures*, ed. by C. Becker, R. Fried, S. Kuhnt (Springer, Heidelberg, 2013), pp. 49–66
M. Salibian-Barrera, S. Van Aelst, G. Willems, Principal component analysis based on multivariate MM-estimators with fast and robust bootstrap. J. Am. Stat. Assoc. **101**(475), 1198–1211 (2006)
H. Scheffé, Experiments with mixtures. J. R. Stat. Soc. Ser. B Stat Methodol. **20**(2), 344–360 (1958)
A. Smilde, R. Bro, P. Geladi, *Multi-way Analysis: Applications in the Chemical Sciences* (Wiley, Chichester, 2004)
R. Tolosana-Delgado, K.G. van den Boogaart, Linear models with compositions in R, in *Compositional Data Analysis: Theory and Applications*, ed. by V. Pawlowsky-Glahn, A. Buccianti (Wiley, Chichester, 2011), pp. 356–371
S. Van Aelst, G. Willems, Fast and robust bootstrap for multivariate inference: the R package FRB. J. Stat. Softw. **53**(3), 1–32 (2013)
K.G. van den Boogaart, R. Tolosana-Delgado, *Analyzing Compositional Data with R* (Springer, Heidelberg, 2013)
K. Varmuza, P. Filzmoser, *Introduction to Multivariate Statistical Analysis in Chemometrics* (CRC Press, Boca Raton, 2009)
B. Walczak, P. Filzmoser, What can go wrong at the data normalization step for identification of biomarkers? J. Chromatogr. A **1362**, 194–205 (2014)
H. Wold, M. Sjöström, L. Eriksson, PLS regression: a basic tool of chemometrics Chemom. Intell. Lab. Syst. **58**, 109–130 (2001)

Chapter 11
Methods for High-Dimensional Compositional Data

Abstract With increasing dimensionality of compositional data much more care needs to be devoted to a reasonable coordinate representation and selection of methods to be used for their statistical processing. This situation frequently occurs with chemometric data, particularly when dealing with observations from "omics"-fields (genomics, proteomics, or metabolomics). In principle, all methods that are popular in the context of high-dimensional data, like principal component analysis and partial least squares regression, can also be used for compositional data with far more parts than observations. On the other hand, while pivot coordinates are still useful in terms of interpretation also in the high-dimensional context, this is not so clear for other types of balances: defining an interpretable sequential binary partition for compositions with hundreds or thousands of parts, where many of them may just be related to noise, is nearly impossible. Accordingly, it is meaningful here to consider even the elemental information, contained in pairwise logratios, to build up a relevant method for marker identification or for the detection of cell-wise outliers. The latter one can be used to reveal which observations are deviating from the majority in order to identify possible measurement errors or other artifacts. Moreover, it may be possible with these methods to identify parts or groups of parts that show a different behavior in all or in subsets of the observations.

11.1 Specific Problems of High-Dimensional Compositions

In applications from the field of chemometrics, and also in other related fields, it is common that the number of compositional parts is substantially higher than the number of observations available. While hundreds or even thousands of parts may be available, only tens or, in the best case, hundreds of observations may have been accessible. This is due to the fact that such samples often originate from measuring biological material, such as plants, animals, or humans. In addition to financial reasons, the sample size is thus limited also by ethical restrictions or by the rareness of a disorder to be analyzed.

P. Filzmoser et al., *Applied Compositional Data Analysis*, Springer Series in Statistics, https://doi.org/10.1007/978-3-319-96422-5_11

On the other hand, most traditional statistical methods, including their adaptations to compositional data, assume that the number of samples is higher than the number of variables (Bühlmann and van de Geer 2011). For instance, this is essential for the estimation of the covariance matrix: Although the theoretical covariance matrix is positive definite, the regularity cannot be achieved by its estimation from a sample where the number of observations is lower than the number of variables (parts). Singularity of the covariance matrix is a serious problem for many multivariate methods, where its inverse is required. This is the case for outlier detection based on Mahalanobis distances, for classification using LDA/QDA, and indirectly also for regression analysis with compositional response and/or covariates as introduced in Sect. 10. Finally, traditional methods of robust statistics also cannot deal with the high dimensionality of the observations. An example is the MCD estimator, where the determinant of the sample covariance matrices, computed from subsets of the whole data set, needs to be minimized (see Sect. 5.2.3); trivially, these determinants are zero and the optimization problem is ill-defined. For this reason, approaches that relax the assumption of a large sample size are preferred. Among them, singular value decomposition belongs to the most popular ones. This algorithm is used not just for the computation of principal components, but also for a range of other popular methods in high-dimensional data analysis, including partial least squares regression which is introduced below, and Parafac/Tucker3 models for the analysis of three-way data (Kroonenberg 2008; Smilde et al. 2004) that were adapted also for compositional data (Engle et al. 2014; Gallo 2015). Some popular methods in the context of high-dimensional data (including, e.g., Lasso regression) were discussed in Bühlmann and van de Geer (2011).

Specific problems with high-dimensional compositions have either computational origin, or they are related to the characteristics of the data themselves. The first issue is mostly connected to the numerical instability resulting from computing geometric means that occur in logratio coordinates. Note that a similar problem can also arise by determining the center of the distribution (Sect. 4.1) directly from the original compositions, when the absolute values of the parts are very high. Consequently, by multiplying the parts in $g_m(\mathbf{x})$ it can easily happen that the resulting product to be extracted exceeds the storage for the number representation in the computer. As a way out, the geometric mean can be computed by expressing the original compositional parts in log-scale (3.16), i.e.

$$g_m(\mathbf{x}) = \exp\left(\frac{1}{D}\sum_{j=1}^{D}\ln x_j\right).$$

By doing so, the numerical stability is achieved even for compositions with very high numbers of compositional parts.

The second problem deserves more attention. With high-dimensional data resulting from signal processing it frequently happens that some parts can be considered as random noise. For example, if two patient groups should be classified according to mass spectral measurements, there may be several parts that are not informative

for the group classification and thus be considered as noise variables. If a statistical method cannot filter such noise variables in an appropriate manner, the classifier may have poor accuracy. This is even more severe in compositional data analysis, where the construction of coordinates involves all or most of the parts (Walczak and Filzmoser 2014). Since it is usually not clear at the beginning of the analysis which are the uninformative noise parts, it is not possible to construct coordinates that would exclude those parts. Thus, alternative methods need to be consulted, see Sect. 11.3.

Another difficulty may arise from measurement problems, for example if the measurements of some parts are close to the detection limit of the device. As before, this may have severe consequences for such logratio coordinates where all or most parts are aggregated, because the problem is reflected also in the aggregated form. On the other hand, these effects are usually suppressed with an increasing number of parts, because possible trends eliminate each other and in the geometric mean more or less only pure noise remains (Mert et al. 2016; Gardlo et al. 2016), see Sect. 11.4.

Finally, also for high-dimensional compositions it is theoretically possible to construct a specific sequential binary partition to obtain balance coordinates (3.37). Nevertheless, due to the high number of compositional parts and the complexity of the relations between them, this approach is rarely used.

11.2 Partial Least Squares for Regression and Classification

One of the most prominent statistical methods for high-dimensional data with a range of applications in chemometrics and other fields is partial least squares (PLS) regression (Wold et al. 2001). Basically, this is a method to relate a set of explanatory variables to one or more responses by means of latent variables. It can be viewed as a variant of principal component analysis and multiple regression. PLS is used for both regression and classification tasks in practice, and it can be employed for reducing the dimensionality of the data. The intrinsic assumption of all PLS methods is that the observed data are generated by a system or process which is guided by a small number of latent variables which cannot be directly observed or measured. In the compositional context, PLS regression is used primarily instead of the standard LS regression method for the case of regression with compositional explanatory variables (Sect. 10.3), if the number of compositional parts (D) is higher than the number of observations (n). Although PLS estimators cannot provide such nice theoretical properties like those resulting from the LS regression estimation, they represent a well-justified alternative when the standard approach necessarily fails.

For the purpose of PLS modeling, both the response and the covariates are usually mean-centered. Similar as for the standard regression case, the resulting orthonormal coordinates can be relaxed to orthogonal ones in order to enhance the interpretation of the regression parameters. For a given $l \in \{1, \ldots, D\}$ and pivot coordinates (3.25), the sample values are recorded in the matrix $\mathbf{Z}^{(l)}$ of dimension $n \times (D-1)$. Without loss of generality, $l = 1$ is assumed in the following, and

the notation $\mathbf{Z}^{(1)}$ is simplified to $\mathbf{Z}$. In the matrix $\mathbf{Y}$ of size $n \times q$ the values of q properties (response variables) for the same n objects are collected, where q can be (much) bigger than n. In the classification case, denoted as PLS-DA (partial least squares discriminant analysis), the matrix $\mathbf{Y}$ consists of binary variables describing the different categories, e.g., zeroes and ones in the case of two categories (Pérez-Enciso and Tenenhaus 2003). The number of dependent variables is equal to the number of categories. The method is optimized for the balanced case, when the same amount of members in each category is considered.

Partial least squares regression applied to the multivariate case ($q > 1$) is also known under PLS2, whereas the case $q = 1$ is denoted by PLS1 (Varmuza and Filzmoser 2009). The aim of PLS2 regression is to find a linear relationship between the response and the explanatory variables, using a $(D-1) \times q$ matrix $\mathbf{B}$ of regression coefficients, and an error matrix $\mathbf{E}$,

$$\mathbf{Y} = \mathbf{ZB} + \mathbf{E}. \tag{11.1}$$

As mentioned above, the columns of $\mathbf{Z}$ and $\mathbf{Y}$ are assumed to be mean-centered. Instead of directly estimating the regression coefficients in the relation (11.1), $\mathbf{Z}$ and $\mathbf{Y}$ are modeled by linear latent variables according to the regression models

$$\mathbf{Z} = \mathbf{TP}' + \mathbf{E}_Z$$
$$\mathbf{Y} = \mathbf{UQ}' + \mathbf{E}_Y,$$

where $\mathbf{E}_Z$ and $\mathbf{E}_Y$ are matrices of residuals. The matrices $\mathbf{T}$ and $\mathbf{U}$ represent score matrices and the matrices $\mathbf{P}$ and $\mathbf{Q}$ are loading matrices, respectively. All these matrices have a columns, where $a \leq \min(D-1, q, n)$ is the number of PLS components, to be chosen by the user. The scores in $\mathbf{T}$ are linear combinations of the explanatory variables and can be considered as good summaries of these variables. The same relationship holds for the response variables and the matrix $\mathbf{U}$.

Then the relationship between the scores becomes

$$\mathbf{U} = \mathbf{TD} + \mathbf{H}, \tag{11.2}$$

where $\mathbf{D}$ is a diagonal matrix with elements $d_1, \ldots, d_a$, and $\mathbf{H}$ is the residual matrix (Varmuza and Filzmoser 2009). Since all quantities in (11.2) are unknown (latent variable problem), the parameter estimation needs to be based on an additional criterion. In case of PLS2, this criterion is the maximization of the covariance between the scores, corresponding to explanatory and response variables. The requirements of high (total) explained variance of $\mathbf{Z}$ and high correlation between $\mathbf{Z}$ and $\mathbf{Y}$ are both included in this criterion. Note that for PLS1, simply the covariance between the response, represented by the vector $\mathbf{y}$ of length n, and the scores of $\mathbf{Z}$ is maximized. Consider a weight vector $\mathbf{w}$ for the explanatory variables, $\mathbf{t} = \mathbf{Zw}$, and a weight vector $\mathbf{c}$ for the response variables, $\mathbf{u} = \mathbf{Yc}$. Then the maximization

problem can be written as

$$\max_{\|\mathbf{t}\|=\|\mathbf{u}\|=1} \operatorname{cov}(\mathbf{t}, \mathbf{u}) = \max_{\|\mathbf{Zw}\|=\|\mathbf{Yc}\|=1} \operatorname{cov}(\mathbf{Zw}, \mathbf{Yc}). \tag{11.3}$$

The solution of the maximization problem is formed by the first score vectors $\mathbf{t}_1$ and $\mathbf{u}_1$, the columns of the corresponding score matrices (their unit length is required for uniqueness of the solution). For the next score vectors, orthogonality constraints to the previous score vectors are imposed, i.e., $\mathbf{t}_j^T \mathbf{t}_l = 0$ and $\mathbf{u}_j^T \mathbf{u}_l = 0$ for $1 \leq j < l \leq a$. Finally, the score matrices $\mathbf{T}$ and $\mathbf{U}$, together with the matrices formed by the weight vectors $\mathbf{w}$ and $\mathbf{c}$, are used for the estimation of the regression parameters $\mathbf{B}$.

Unlike principal component analysis (Chap. 7), it is not possible to get both uncorrelated scores and loadings simultaneously in PLS modeling. There are several algorithms for solving the PLS problem by preserving uncorrelated scores, such as Kernel PLS, NIPALS, SIMPLS, or O-PLS (Varmuza and Filzmoser 2009). Since each additional score vector covers new variability, having uncorrelated scores might be preferable for prediction purposes. On the other hand, from the compositional perspective, it might be rather preferable to find uncorrelated loading vectors. By considering just the space of compositional covariates, such loadings form orthonormal basis vectors leading to coordinates that work in favor of the aim of PLS regression. This goal is followed by the eigenvalue algorithm (Hoeskuldsson 1988). Here the idea is to compute all eigenvectors to the largest a eigenvalues, where a is the desired number of PLS components. Specifically, $\mathbf{p}_1, \ldots, \mathbf{p}_a$ are orthogonal PLS loading vectors in the space of the covariates (logratio coordinates) given by the eigenvectors to the a largest eigenvalues of $\mathbf{Z}'\mathbf{Y}\mathbf{Y}'\mathbf{X}$. Orthogonal PLS loading vectors in the space spanned by the response variables, $\mathbf{q}_1, \ldots, \mathbf{q}_a$, are the eigenvectors to the a largest eigenvalues of $\mathbf{Y}'\mathbf{Z}\mathbf{Z}'\mathbf{Y}$. The scores for both the covariates and the response are found by projecting the data on the loading vectors, i.e., $\mathbf{t}_j = \mathbf{Z}\mathbf{p}_j$ and $\mathbf{u}_j = \mathbf{Y}\mathbf{q}_j$, respectively. Even though this approach does not solve the initial maximization problem, this price is worth to be payed for the desired orthogonality of the loadings.

For the estimation of the regression parameters, an analogous strategy concerning the choice of ilr coordinates can be used as for the LS approach developed in Sect. 10.3. It is recommended to use an orthogonal transformation matrix $\mathbf{Q}^{(lk)}$ from (10.13) in order to reduce the computation effort, necessary for obtaining scores and loadings using the PLS algorithms (Kalivodová et al. 2015). Alternatively, it is possible to replace the pivot coordinates by clr coefficients in the matrix $\mathbf{Z}$. The results in terms of a dominance of the single compositional parts with respect to the averaged rest of components, extracted from the first coordinates (columns) of the respective matrices $\mathbf{Z}^{(l)}$, $l = 1, \ldots, D$, are the same up to a scaling constant (3.30).

Similar as for PCA, the loadings and scores of the covariates corresponding to the first two PLS components can be jointly visualized in a biplot. In order to save computational effort, they can be obtained in clr coefficients, analogously to the case of Sect. 7.3. Consequently, the loadings can be assigned to the original parts in terms of an interpretation of clr coefficients without the necessity of constructing

D pivot coordinate systems. The interpretation of the PLS biplot follows only roughly the usual compositional biplot due to the different origin of the score and loading vectors from the PCA case. Nevertheless, the information about the response variables, utilized by the construction of loadings and scores, can form an important value added in applications. This holds particularly for PLS-DA that utilizes the class pertinence of samples.

As mentioned above, the key idea of PLS to identify the latent variables is covariance maximization, see Eq. (11.3). Traditionally, the sample covariance is considered for this purpose, which is quick and easy to compute. In case of outliers or heavy-tailed distributions, however, the resulting model may have poor prediction performance because the classical covariance estimation is affected by these non-ideal conditions. For this reason, several approaches exist with the aim of estimating the PLS model in a more robust way. The approach by Serneels et al. (2005) for PLS1 regression, called Partial Robust M-estimator (PRM), uses a weighted covariance, with weights derived in the score space and from the residuals. This estimator is highly robust and also quick to compute. A sparse version of this estimator has been developed in Hoffmann et al. (2015), and thus in addition to robustness one obtains variable selection according to the sparsity of the model. Based on these ideas, a sparse robust PLS method for a two-group classification problem (PLS-DA) has been developed in Hoffmann et al. (2016). All these robust methods are implemented in the R package **sprm**.

11.3 Marker Identification Using Pairwise Logratios

One of the natural tasks in the classification problem with high-dimensional compositions is significance testing. In the context of chemometrics, this is closely connected to marker identification, i.e. which of the predictor variables can be considered as typical (either because of its lack, or abundance) for a specific group of samples, like for patients suffering from a disease, or which variable allows to distinguish the groups or classes. In PLS-DA with compositional data, one possibility is to use a jack-knife procedure (Kalivodová et al. 2015); nevertheless, this is only recommendable for the balanced case of having about the same number of samples in each group.

Moreover, as it was indicated in Sect. 11.1, using pivot balances (3.25) for the marker identification might be a bit tricky, especially in high-dimensional settings. They tend to lead to false positive results (Walczak and Filzmoser 2014), which means that their "significance" is frequently just driven by several pairwise logratios, aggregated into $z_1^{(l)}$, for $l = 1, \ldots, D$. A possible way out is to give up a coordinate representation and rely just on pairwise logratios. Of course, one must be aware that there are $D(D-1)/2$ different variable pairs, so their number can explode quickly with increasing D. On the other hand, an appropriate treatment of pairwise logratios can lead to convincing results.

One such approach is proposed in Walach et al. (2017), where a method for marker identification in the most common case of two groups of observations is presented. It is based on the use of the variation matrix (4.2) that enables to reveal the proportionality between compositional parts. Consider an $n \times D$ compositional data matrix $\mathbf{X}$, where the observations originate from two groups. Let $\mathbf{X}^{[1]}$ denote the sub-matrix with the n_1 compositions in the rows from the first group, and $\mathbf{X}^{[2]}$ the corresponding matrix with n_2 observations of the second group, where $n_1 + n_2 = n$. The matrix elements of $\mathbf{X}^{[m]}$ are denoted by $x_{ij}^{[m]}$, for $i = 1, \ldots, n_m$, $j = 1, \ldots, D$, and $m = 1, 2$. Besides the variation matrix $\mathbf{T} = (t_{jk})$ of order D based on all observations jointly, the individual group variation matrices are considered as well. The symbol $\mathbf{T}^{(m)}$ denotes the variation matrix of group m, for $m = 1, 2$, with its elements defined as

$$t_{jk}^{[m]} = \operatorname{var}\left[\ln\left(\frac{x_{1j}^{[m]}}{x_{1k}^{[m]}}\right), \ln\left(\frac{x_{2j}^{[m]}}{x_{2k}^{[m]}}\right), \ldots, \ln\left(\frac{x_{n_m j}^{[m]}}{x_{n_m k}^{[m]}}\right)\right], \tag{11.4}$$

for $j, k = 1, \ldots, D$. Thus, the variation matrices of the individual groups consider only the observations from their own groups.

For marker identification, the following statistic V_j is proposed:

$$V_j = \sum_{k=1}^{D} \frac{(n_1 + n_2)\sqrt{t_{jk}}}{n_1 \cdot \sqrt{t_{jk}^{[1]}} + n_2 \cdot \sqrt{t_{jk}^{[2]}}}, \qquad \text{for } j = 1, \ldots, D. \tag{11.5}$$

If the jth part is not a marker, the jth column (and row) of all three sources of information $\mathbf{T}$, $\mathbf{T}^{(1)}$ and $\mathbf{T}^{(2)}$ will have similar structure. For this reason, each term of the sum in (11.5) will be approximately around one for all non-markers k. On the other hand, if the jth part is a marker, $t_{jk}^{(1)}$ and $t_{jk}^{(2)}$ will be different, and tentatively much smaller than t_{jk}, for all k. The resulting V_j will then be considerably higher than for non-markers. So, the higher the value of the statistic (11.5) is, the less similar the groups are with respect to this jth variable. For a normalized version of the statistic V_j,

$$V_j^* = \frac{V_j - \bar{V}}{s_V}, \qquad \text{for } j = 1, \ldots, D, \tag{11.6}$$

with the arithmetic mean

$$\bar{V} = \frac{1}{D}\sum_{k=1}^{D} V_k$$

and the empirical standard deviation

$$s_V = \sqrt{\frac{1}{D-1}\sum_{k=1}^{D}(V_k - \bar{V})^2},$$

standard normal distribution can be approximately assumed (Walach et al. 2017). Accordingly, values of (11.6) higher than a cut-off value formed by the standard normal quantile $u_{0.975} \approx 1.96$ can be considered as markers.

In presence of outliers, the values of the above variation matrices with "var" being represented by the sample variance can result in spoiled values of the statistics V_j. As a consequence, marker identification based on V_j^* would become unreliable. Similar as in Sect. 6.6, a robust version of the classical variation matrix is thus needed. Because the MCD estimator now necessarily fails, other alternatives are required. One of them would be to apply the OGK estimator (Sect. 5.2.3) together with Eq. (5.8) for this task. Nevertheless, as this estimator is not affine equivariant, it is even preferable here to go for a univariate estimator of the single elements of the variation matrix. One such possibility is to use the τ estimator of the variance (Yohai and Zamar 1998; Maronna and Zamar 2002). This estimator is highly robust, and it also attains a high efficiency, tunable with two constants c_1 and c_2. This is particularly important when dealing with small sample sizes. The estimator uses weights for the observations, defined for a univariate sample $\mathbf{y} = (y_1, \ldots, y_n)'$ as

$$w_i = \omega_{c_1}\left(\frac{y_i - \text{median}(\mathbf{y})}{s_0}\right) \qquad \text{for } i = 1, \ldots, n, \tag{11.7}$$

with the weight function

$$\omega_{c_1}(u) = \max\left(0, (1 - (u/c_1)^2)^2\right) \text{ and } s_0 = \text{MAD}(\mathbf{y})$$

for the MAD estimator defined in Sect. 5.2.2. Then the τ estimator of variance is defined as

$$\sigma_\tau^2 = \frac{s_0^2}{n}\sum_{i=1}^{n}\rho_{c_2}\left(\frac{y_i - \bar{y}_w}{s_0}\right), \tag{11.8}$$

where

$$\bar{y}_w = \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i} \quad \text{and} \quad \rho_{c_2}(u) = \min(c_2^2, u^2).$$

In order to combine good robustness properties with high efficiency, the recommended tuning parameters are $c_1 = 4.5$ and $c_2 = 3$. This leads to around 80% efficiency at normal distributions, while keeping the breakdown point at 50%. The resulting robust statistics V_j^* have shown very good performance both with simulated and real data compared to other possible approaches, even in unbalanced settings (Walach et al. 2017).

Moreover, the weights for the τ estimator, computed for all pairwise logratios in the single groups of observations, can be used for cell-wise outlier detection. Thus it is possible to reveal which observations are deviating from the majority in order to identify possible measurement errors or other artifacts, and also some parts or groups of parts that show different behavior in all or subsets of the observations are detected. Since all variable pairs $j, k = 1, \ldots, D$ are considered for estimating the variation matrix, one can store all weights (11.7) in a three-way array $\mathbf{W}^{[m]} = (w_{jki}^{[m]})$ with D rows, D columns, and n_m slices, $m = 1, 2$. Because the slices of $\mathbf{W}^{[m]}$ are symmetric, the weights can be averaged for each observation and each involved part,

$$p_{ij}^{[m]} = \frac{1}{D} \sum_{k=1}^{D} w_{jki}^{[m]}, \tag{11.9}$$

for $j = 1, \ldots, D$, $i = 1, \ldots, n_m$, and $m = 1, 2$. This information is stored in the $n_m \times D$ matrix $\mathbf{P}^{[m]}$, which can be represented graphically. All values are in the interval [0, 1], where small values indicate outlying cells (Walach et al. 2017).

11.4 Principal Balances

While pivot coordinates (3.25) can still be useful in the high-dimensional context, this is hardly the case for general balances (3.37): defining an interpretable sequential binary partition for compositions with hundreds or thousands of parts is nearly impossible. One way out is to define such (usually only few) balances that account for most of the variability contained in the compositional data set. The idea links to principal component analysis (see Sect. 7), which also inspired the name *principal balances* for such orthonormal coordinates (Pawlowsky-Glahn et al. 2011). In order to minimize computational costs and to enable for a direct interpretation of the loading vectors in terms of logcontrast coefficients (Sect. 3.3), principal component analysis is computed in clr coordinates. Three methods for the construction of principal balances were proposed in Pawlowsky-Glahn et al. (2011):

AP (angular proximity to principal components): In the first step of this recursive algorithm, all possible binary partitions of the full D-part composition are created. The balancing element with the smallest geometric angle with one of the principal components is stored and removed from the set of possible directions. The procedure is then applied to each group of the previously identified balance separately, where the geometric angle to one of the remaining principal components is minimized. This can be repeated step-by-step, until $D - 1$ balances are extracted, i.e., until a complete sequential binary partition is achieved.

HC (hierarchical clustering of components): The set of balances is constructed by Q-mode clustering from Sect. 6.6. The variance of the balance between two groups (3.37) is used as a criterion to link two clusters. It turns out that this corresponds to the Ward's method (Sect. 6.2.1.4) based on the variation matrix (4.2).

MV (maximum explained variance hierarchical balances): This sequential algorithm starts with the first principal component, and uses two groups with the signs of the loadings for constructing a balance. Let r denote the number of positive signs and s the number of negative signs. Then it is checked whether a change of one positive sign to the other group increases the explained variance. This check is also carried out for all combinations of $2, \ldots, r-1$ positive signs. The balance with the maximum explained variance is stored, a new principal component analysis is performed with the larger group, and so on.

The computation time of AP and MV explodes quickly with increasing dimension because of the exponentially growing number of possible combinations that are used as candidates. Creating all possible combinations also leads to memory allocation problems for larger D. The HC algorithm is just based on a $D \times D$ dissimilarity matrix, which is unproblematic even for larger dimension (Mert et al. 2015).

Even if only few first principal balances accounting for a reasonable portion of the total variability are used for the analysis, their interpretation is violated by the presence of all parts in the first balance. Subsequently, also the other principal balances contain always all parts of the subgroup for which the balance coordinate is constructed. This feature can be suppressed by using sparse principal balances (SPB) (Mert et al. 2015). This method allows for a tradeoff between maximizing explained variance and sparsity, where the latter is referring to the number of involved components $r + s \ll D$ in each of the new coordinates. Accordingly, an SPB should describe the information of only a few compositional parts with zero contribution from the other (majority of) parts. This is similar to the aim of sparse principal component analysis, where many of the entries of the loading matrix are forced to be zero (Zou et al. 2006). Nevertheless, even in case of sparse principal balances, a careful interpretation is needed, because with more balances derived, the danger increases that also some marginal (erroneous) effects can be captured.

11.5 Examples

11.5.1 Example for PLS for Two-Group Classification

Consider the data set `BrainSpectra` from the package **MetabolAnalyze** with NMR spectral data from brain tissue samples of rats. The NMR spectra consist of 164 spectral bins, and they are measured in parts per million (ppm), which already indicates the relative scale. It is known from which brain regions the samples have been taken. Here the samples of only two out of four available brain regions

are considered: region "Hippocampus" and region "Pre-frontal cortex," and the corresponding observations are coded with −1 and 1, respectively.

```
library("MetabolAnalyze")
data("BrainSpectra")
y <- BrainSpectra[[2]][17:33]        # brain regions 3 and 4
y[y == 3] <- -1
y[y == 4] <- 1
table(y)

## y
## -1  1
##  8  9

X <- BrainSpectra[[1]][17:33, ] + 0.1  # NMR spectra, baseline corrected
dim(X)

## [1]  17 164
```

The goal is to find a model which allows to accurately classify tissue samples to the two brain regions, and to identify which spectral bins are "significantly" different in the two groups. In a first attempt, pivot coordinates are constructed for the NMR spectra matrix, and PLS is applied to this two-group classification problem.

```
Z <- as.matrix(pivotCoord(X))
library(pls)
res.ilr <- mvr(y ~ Z, method = "simpls", validation = "LOO")
```

In this case, the algorithm "simpls" is used, and leave-one-out (LOO) cross-validation is performed to estimate the prediction error, see Fig. 11.1. For this

```
plot(res.ilr, "validation", val.type = "RMSEP", legendpos = "top")
```

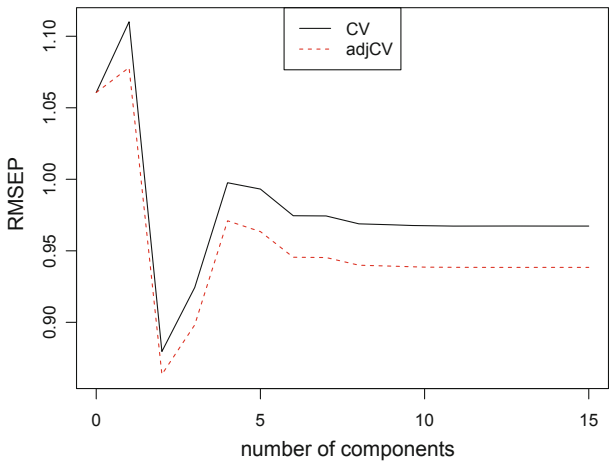


Fig. 11.1 Prediction error (root mean squared error of prediction, RMSEP) for the brain spectral data set with two groups, depending on the number of PLS components

plot, the root-MSE is computed based on the cross-validated predictions for the different numbers of PLS components. The root-MSE is the square-root of the mean squared error (MSE), where MSE is the average of the squared differences between the measured and the predicted response values. The plot also shows the RMSE adjusted for the number of parameters. The conclusion is that a model with two PLS components leads to the smallest prediction error. However, one should be careful with this plot: As mentioned before, prediction errors are based on the squared residuals $(y_i - \hat{y}_i)^2$, where y_i is the group label (here -1 or 1) and $\hat{y}_i$ is the predicted group label for the ith observation, $i = 1, \ldots, n$. The predicted group label is not necessarily a number in $\{-1, 1\}$, but in general it can be any real number, since a linear combination of the x-variables is used to compute this prediction. Therefore, it might be better to look at other criteria that are more appropriate in the classification context, such as misclassification errors of predictive abilities.

Since pivot coordinates have been used to construct the PLS model, it would also be difficult to draw conclusions concerning the "significance" of the variables (compositional parts) for distinguishing the two groups. This would only be appropriate for the first pivot coordinate which describes all relative information of the first compositional part (spectral bin) to the remaining parts.

Rather than constructing pivot coordinates for each single compositional part, it might be more convenient to compute the PLS model from clr coefficients and to perform inference for those coefficients.

```
X.clr <- as.matrix(cenLR(X)$x.clr)    # clr coefficients matrix
res.clr <- mvr(y ~ X.clr, method = "simpls",
               validation = "LOO", jackknife = TRUE)
```

The option `jackknife=TRUE` causes that the variance of the PLS regression coefficients is estimated by a jackknife prodedure. These estimates will be used further below for statistical inference about the clr coefficients. Before doing that, the "optimal" number of PLS components needs to be determined. This is done now based on the predictive ability, which is defined as the average of the proportions of correctly classified observations in the two groups (Varmuza and Filzmoser 2009). The class predictions are derived from the cross-validation scheme, using a PLS model with a certain number of PLS components.

Figure 11.2 shows that a PLS model with two components leads to the best predictive ability of about 0.83. In other words, on average 83% of the observations in the two groups are correctly classified with this model. Taking more or fewer PLS components would lead to a much worse predictive ability. Note that this outcome would be exactly the same for the PLS model based on pivot coordinates. Using similar reasoning as in Sect. 7.3 about compositional biplots, the clr coefficients can serve as workhorse for computational issues, but interpretation of results and, particularly, possible inference is done by having the first pivot coordinates $z_1^{(l)}$ for $l, \ldots, D$ from (3.25) in mind instead. The point is that each of such coordinates is assigned to its own orthonormal coordinate system, and thus enables to avoid an interrelation of clr coefficients.

```
pred <- drop(res.clr$validation$pred)       # LOO-CV predictions
ncomp <- dim(pred)[2]                       # max number of components
Pabil <- matrix(NA, ncol = 2, nrow = ncomp)  # Predictive abilities
for (i in 1:ncomp){
  class1 <- pred[y == -1, i]<0              # predicted as class -1
  class2 <- pred[y == 1, i] >= 0            # predicted as class 1
  Pabil[i,1] <- sum(class1) / sum(y == -1) # predictive ability class -1
  Pabil[i,2] <- sum(class2) / sum(y == 1)  # predictive ability class 1
}
Pab <- apply(Pabil, 1, mean)              # average predictive ability
par(mar = c(4,4,0.1,0.1), cex.lab = 1.3)
plot(1:ncomp, Pab, xlab = "Number of components",
     ylab = "Predictive ability", type = "b")
```

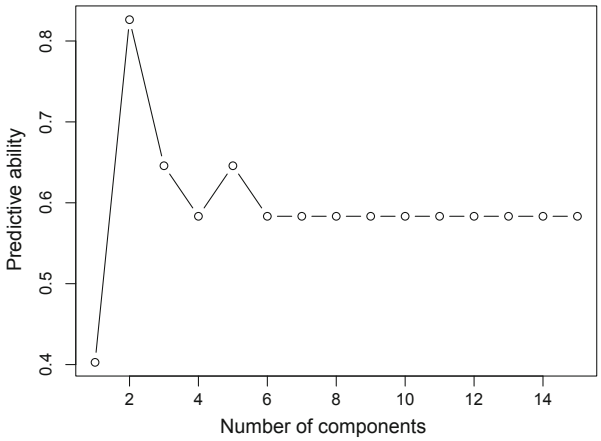


Fig. 11.2 Predictive ability for the brain spectral data set with two groups, depending on the number of PLS components

Here the interest is in inference for the PLS regression coefficients for the two-component model. These regression coefficients relate to the single clr coefficients, which carry all relative information of a specific spectral bin to the geometric mean of all bins. Using the variance estimates from the jackknife procedure, approximate t-tests for the single regression coefficients can be performed. Figure 11.3 shows the outcome of the test statistic for each coefficient. The boundaries at ± 2 can be considered as cut-off values: if a value of the test statistic exceeds this range $[-2, 2]$, the corresponding variable can be considered as significant in the model. This means that these variables are important for distinguishing the two brain regions.

It is also easy to extract the indexes of the significant coefficients. One can also distinguish between significant coefficients where the absolute value of the test statistic is in the interval $(2, 3]$, and highly significant coefficients with values exceeding 3:

```
abst <- abs(drop(restest$tvalues))       # absolute value of test statistic
ind2 <- which(abst > 2 & abst <= 3)      # significant
```

```
restest <- jack.test(res.clr, ncomp = 2)
par(mar = c(4,4,0.1,0.1), cex.lab = 1.4)
plot(1:ncol(X.clr), drop(restest$tvalues), cex = 0.7,
     xlab = "index of variable", ylab = "value of test statistic")
abline(h = 0)
abline(h = c(-2, 2), lty = "dashed")
```

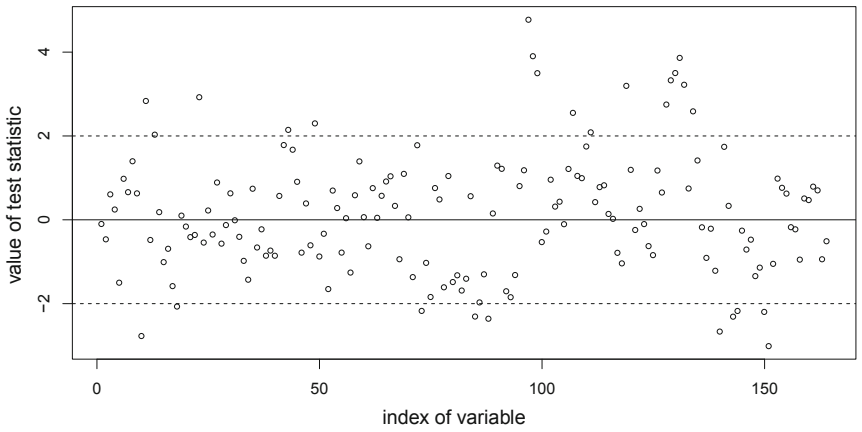


Fig. 11.3 Inference for the PLS regression coefficients for the brain spectral data set with two groups, for a model with two PLS components

```
as.numeric(ind2)

##  [1]  10  11  13  18  23  43  49  73  85  88 107 111 128 134
## [15] 140 143 144 150

ind3 <- which(abst > 3)                    # highly significant
as.numeric(ind3)

## [1]  97  98  99 119 129 130 131 132 151
```

Figure 11.4 shows the clr coefficients of the two groups, visualized in different colors. The significant regression coefficients are indicated by the vertical lines in yellow color, whereas orange is used for highly significant coefficients.

11.5.2 Example for Marker Identification

For reasons of comparability, the same data set as before is used. However, here the interest is only in identifying those markers which allow to distinguish the two groups, and not so much in a misclassification rate or a predictive ability of a classification model. For this purpose, the method of Walach et al. (2017) is used which is based on pairwise logratios, employing the variation matrix. In more detail, the variation matrix elements of all observations jointly are compared with those

```
par(mar = c(4,4,0.1,0.1), cex.lab = 1.4)
matplot(t(X.clr), col = y + 4, type = "l", lty = 1,
        xlab = "Index of variable", ylab="clr coefficient")
abline(v = ind2, col = "yellow")
abline(v = ind3, col = "orange")
matlines(t(X.clr), col = y + 3, type = "l", lty = 1)
legend("topleft", legend = c("Group -1", "Group 1"),
       col = c(2,4), bg = "white", lty = c(1,1))
```

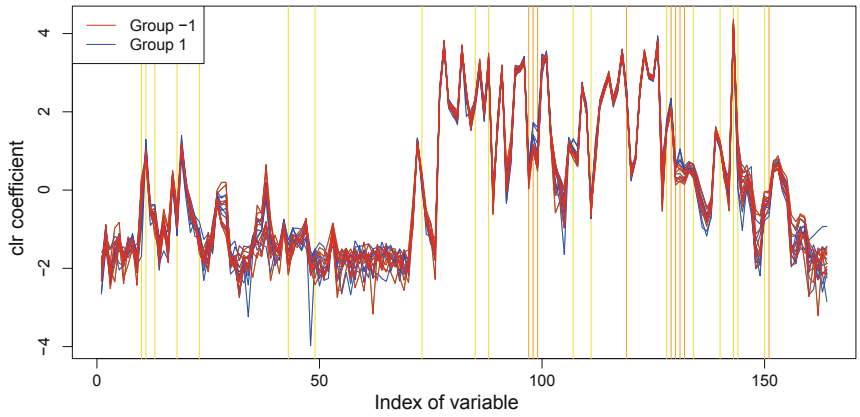


Fig. 11.4 Significance of the PLS regression coefficients for the brain spectral data set with two groups, for a model with two PLS components

computed from the single groups only, and big differences indicate potential marker variables. Big differences are expressed in terms of big values of the statistic V_j^* from Eq. (11.6).

The outcome will depend on the estimator of the variance "Var" for the variation matrix elements, see (11.4). The function `biomarker` in **robCompositions** has different options. First, the classical empirical variance is used, with the option `type="sd"`.

Figure 11.5 shows the outcome for the statistic V_j^* for each variable. Values that exceed the dotted horizontal line indicate potential biomarkers. These are the variables with the following indexes:

```
ind.sd <- which(res.sd$biom.ident$biomarkers)
ind.sd

##  [1]  10  49  86  88  98  99 119 129 131 143
```

This outcome can be compared to the significant variables identified with the PLS approach, see Fig. 11.4, and there is a strong overlap. However, now fewer variables are declared as potential biomarkers.

In a second approach, the robust τ estimator is used as estimator of the variance for the variation matrix, see (11.8), leading to the outcome in Fig. 11.6 for the V_j^* statistic, and to the following indexes of the biomarkers:

```
res.sd <- biomarker(X, g1 = which(y == 1), g2 = which(y == -1),
                    type = "sd", diag = FALSE)
plot(res.sd)
```

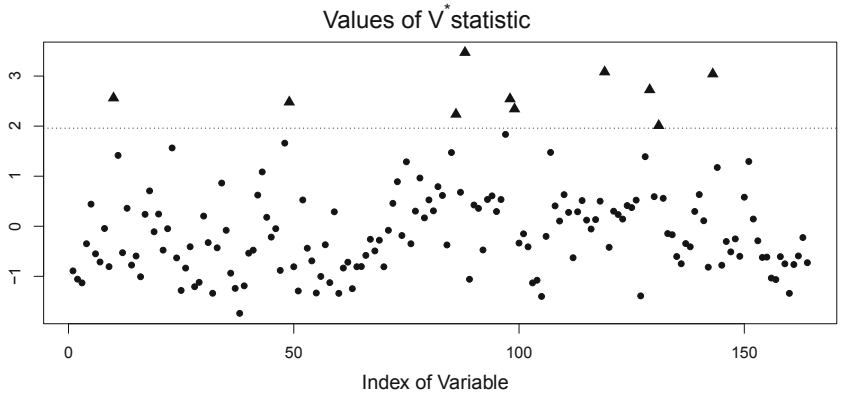


Fig. 11.5 Values of the statistic V_j^* for each variable, based on the classical empirical variance as estimator for the variation matrix elements

```
res.tau <- biomarker(X, g1 = which(y == 1), g2 = which(y == -1),
                     type = "tau")
plot(res.tau)
```

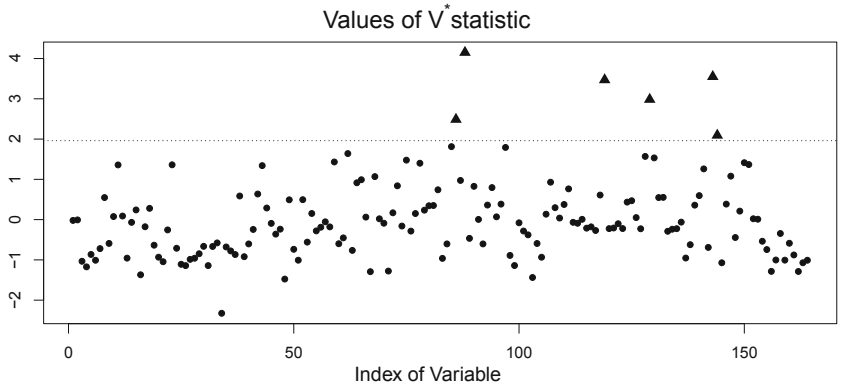


Fig. 11.6 Values of the statistic V_j^* for each variable, based on the robust τ estimator as estimator for the variation matrix elements

```
res.tau <- biomarker(X, g1 = which(y == 1), g2 = which(y == -1),
                     type = "tau", diag = FALSE)
ind.tau <- which(res.tau$biom.ident$biomarkers)
ind.tau

## [1]  86  88 119 129 143 144
```

```
plot(res.tau, type = "diag")   # outlier diagnostics plot
```

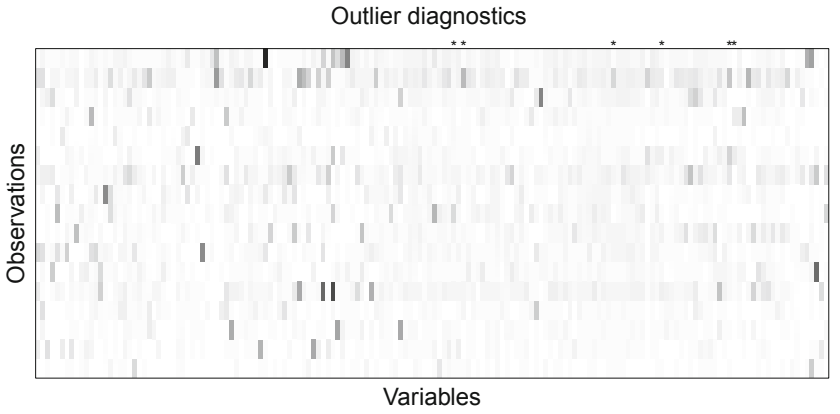


Fig. 11.7 Outlier diagnostics for the cells of the data matrix, based on the robust τ estimator as estimator for the variation matrix elements. The darker the cell, the smaller the weight and the more likely this cell is an outlier. Biomarkers are indicated with $*$

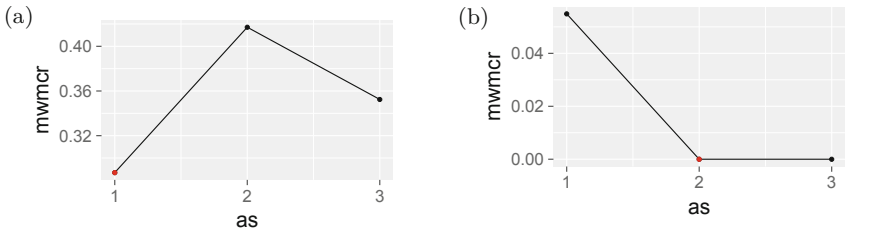
So, again fewer variables are identified as biomarkers, and the reason for the different answer might be due to outliers in the data. Since weights are computed for the τ estimator, these weights can be used to show a diagnostic plot for outlyingness of the individual cells of the data matrix, see Fig. 11.7. Indeed, there are some cells which seem to be inconsistent with the corresponding cells of the other observations in the group.

In practice, one can also be interested in the classification accuracy of models with the variable subsets corresponding to the identified biomarkers. However, for this purpose it is necessary to compare the models in a unified manner. Here, this comparison is made with PRM, a robust version of PLS, see Serneels et al. (2005), used in a cross-validation scheme for classification. This procedure is implemented in the package **sprm** as function `prmdaCV`. Compared are models based on all clr coefficients, for the significant variables from the PLS approach, and for the variation matrix approach using the significant variables based on the classical variance and the τ estimator, respectively. First, the data are prepared accordingly by calculating clr coefficients for the corresponding variable sets. Note that one could also express the information in ilr coordinates, without any difference in the model performance.

```
d.clr <- data.frame(y, X.clr)     # use all clr coefficients
Xsel.pls <- as.matrix(cenLR(X[, c(ind2,ind3)])$x.clr)
d.pls <- data.frame(y, Xsel.pls) # use significant variables from PLS
Xsel.sd <- as.matrix(cenLR(X[, ind.sd])$x.clr)
d.sd <- data.frame(y, Xsel.sd)    # classical variation matrix approach
Xsel.tau <- as.matrix(cenLR(X[, ind.tau])$x.clr)
d.tau <- data.frame(y, Xsel.tau) # robust variation matrix approach
library("sprm")                  # load package
set.seed(123)                    # set random seed for reproducibility
```

```
rescv.clr <- prmdaCV(y ~ .,
     data = d.clr, as = 1:3,
     nfold = 5)
```

```
rescv.pls <- prmdaCV(y~ . ,
     data = d.pls, as = 1:3,
     nfold = 5)
```




```
rescv.sd <- prmdaCV(y ~ .,
     data = d.sd, as = 1:3,
     nfold = 5)
```

```
rescv.tau <- prmdaCV(y~.,
     data = d.tau, as = 1:3,
     nfold = 5)
```

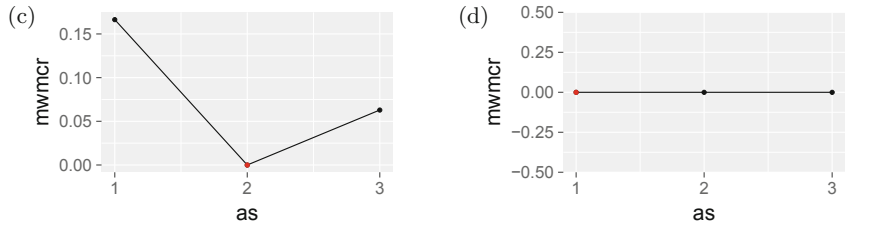


Fig. 11.8 Cross-validated misclassification rates (vertical axes) for different numbers of PLS components (horizontal axes), for the different variable (sub)sets. (**a**) Model based on all clr coefficients. (**b**) Model based on the significant variables for the PLS approach. (**c**) Model using the classical variation matrix approach. (**d**) Model using the robust variation matrix approach

Then, the procedure can be applied to the corresponding data sets, using fivefold cross-validation and up to three PLS components. The resulting misclassification rates (vertical axes) are shown in Fig. 11.8, depending on the number of PLS components (horizontal axes). It can be seen that the misclassification rate is high only if no variable selection is carried out. Otherwise, with models using two PLS components, the misclassification error is even zero. So, any of the corresponding variable subsets referring to potential biomarkers would lead to a perfect group separation. Still, for practitioners it is important to get a reliable indication of biomarkers, and because outliers seem to be present in certain data cells, the robust variation matrix approach seems to give the most reliable answer.

References

A. Hoeskuldsson, PLS regression methods. J. Chemom. **2**, 211–228 (1988)
I. Hoffmann, S. Serneels, P. Filzmoser, C. Croux, Sparse partial robust M regression. Chemom. Intell. Lab. Syst. **149**, 50–59 (2015)
I. Hoffmann, P. Filzmoser, S. Serneels, K. Varmuza, Sparse and robust PLS for binary classification. J. Chemom. **30**(4), 153–162 (2016)
A. Kalivodová, K. Hron, P. Filzmoser, L. Najdekr, H. Janečková, T. Adam, PLS-DA for compositional data with application to metabolomics. J. Chemom. **29**(1), 21–28 (2015)
R.A. Maronna, R.H. Zamar, Robust estimation of location and dispersion for high-dimensional datasets. Technometrics **44**(4), 307–317 (2002)
C. Mert, P. Filzmoser, K. Hron, Sparse principal balances. Stat. Model. **15**(2), 159–174 (2015)
V. Pawlowsky-Glahn, J.J. Egozcue, R. Tolosana-Delgado, Principal balances, in *Proceedings of the 4th International Workshop on Compositional Data Analysis*, St. Feliu de Guíxols, ed. by J.J. Egozcue, R. Tolosana-Delgado, M. Ortego, 2011
M. Pérez-Enciso, M. Tenenhaus, Prediction of clinical outcome with microarray data: a partial least squares discriminant analysis (PLS-DA) approach. Hum. Genet. **112**(5–6), 581–592 (2003)
S. Serneels, C. Croux, P. Filzmoser, P.J. Van Espen, Partial robust M-regression. Chemom. Intell. Lab. Syst. **79**(1–2), 55–64 (2005)
K. Varmuza, P. Filzmoser, *Introduction to Multivariate Statistical Analysis in Chemometrics* (CRC Press, Boca Raton, 2009)
J. Walach, P. Filzmoser, K. Hron, B. Walczak, Robust biomarker identification based on pairwise log-ratios. Chemom. Intell. Lab. Syst. **171**, 277–285 (2017)
B. Walczak, P. Filzmoser, What can go wrong at the data normalization step for identification of biomarkers? J. Chromatogr. A **1362**, 194–205 (2014)
V.J. Yohai, R.H. Zamar, High breakdown-point estimates of regression by means of the minimization of an efficient scale. J. Am. Stat. Assoc. **83**(402), 406–413 (1998)
H. Zou, T. Hastie, R. Tibshirani, Sparse principal component analysis. J. Comput. Graph. Stat. **15**(2), 265–286 (2006)

Chapter 12
Compositional Tables

Abstract Contingency and probability tables are well described in the literature, but the compositional nature of such tables is often not considered. We discuss compositional tables, a generalization of contingency tables that allow also for continuous values in the cells under the requirement of scale invariance. Compositional tables carry relative information about the relationships within and between row and column categories of the variables (factors). The assumption of the Aitchison geometry enables to decompose a compositional table orthogonally into independent and interactive parts. The independence table is formed by a product of row and column geometric marginals and can be considered as a relevant alternative to the independence case in a probability table. Consequently, the interaction table captures relative information about the relationships between factors. In the chapter primarily the special case of 2×2 compositional tables is discussed, being dominant in practical applications. It turns out that for a coordinate representation of compositional tables the sequential binary partitioning is in general not appropriate as it does not respect the two-factor nature of compositional tables. The general case of compositional tables reveals that balance coordinates are recommendable just for the representation of the independence table. For the interaction table coordinates the interpretation in terms of log odds ratios of parts and their groups (quaternary coordinates) is required.

12.1 Motivation and Geometry

Up to now, the methods considered in this book were designed for compositional data that can be expressed in terms of a vector with positive entries, referring to one certain whole. A natural extension is to consider two (marginal) wholes simultaneously, whose combinations form a new (joint) whole. In the two-way case, one can also refer to two (row and column) factors that are represented by the respective wholes and their parts.

This situation is well-known from the case of contingency tables (see, e.g., Agresti 2012), where two discrete distributions are analyzed jointly in form of a table with counts corresponding to various combinations of these factors. A

P. Filzmoser et al., *Applied Compositional Data Analysis*, Springer Series in Statistics, https://doi.org/10.1007/978-3-319-96422-5_12

contingency table and probability tables are examples (representations) of compositional tables (Egozcue et al. 2008), a generalization of contingency and probability tables. A compositional table expresses quantitatively relative contributions of factors to a common whole, i.e. also the important principle of scale invariance applies here. The cells of a compositional table thus may also contain continuous values rather than just counts.

Compositional tables, either in their continuous or discrete forms, arise in many real-world problems. We show examples of two compositional tables that are analyzed in Fačevicová et al. (2014). The first example shows the total intermediate flow of economic activities decomposed by sector (industrial and non-industrial) and by region (domestic and international). One aim was to study whether the intermediate flow between industrial and non-industrial sectors differs by region or not. All in all, 2×2 compositional tables of the 41 countries and regions are analyzed in Fačevicová et al. (2014). One example table belonging to China is shown in Table 12.1 (UNIDO 2009).

In the second example, the total number of employees in a country is decomposed according to working time (full- and part-time) and gender of employee with the aim to analyze relationships between these two factors (see Table 12.2 for the structure of employment in Germany) as well as differences among the countries (the complete data can be found in UNECE 2013).

The traditional analysis of contingency tables involves to analyze if two categorical response variables (two factors) are independent, i.e. to investigate the assumption that independent classifications in contingency tables are built up by multiplying the (standard) marginals, hereafter called arithmetic marginals. An equivalent description is that all joint probabilities are equal to the product of their respective marginal probabilities. It was shown in Egozcue et al. (2015) that this formulation with arithmetic mean marginals is problematic and the table cannot be decomposed orthogonally into its independent and interactive parts, an intuitive and expected requirement. Another result of Egozcue et al. (2015) is that the formulation of the well-known Pearson χ^2 test of independence has its limitations, and alternatives are formulated there. Similar as for contingency tables, the relationships between the factors including their possible independence belong to the main tasks of the analysis of compositional tables. Nevertheless,

Table 12.1 Distribution of intermediate flow between two sectors of China's economy in 2009 in million USD (left) and (as estimated probability table) in proportions (right) (UNIDO 2009)

China	Domestic	International	Domestic	International
Industrial	4,574,156	482,601	0.461	0.049
Non-industrial	4,321,198	555,852	0.435	0.056

Table 12.2 Distribution of employment in Germany (DEU) in 2011 in thousands of persons

DEU	Female	Male
Part-time	8377	2200.4
Full-time	9956.9	19,202.7

the assumption that independent classifications in contingency tables are built up by multiplying the standard arithmetic marginals is replaced by using geometric marginals. This makes it necessary to re-formulate the concept of independence of factors.

Note that for the statistical analysis of contingency tables also log-linear models (Agresti 2012) or correspondence analysis (Greenacre 2007) is frequently employed. In the latter case also zero counts are allowed and even a link to compositional data exists (Greenacre 2011) if the absolute values of counts are irrelevant, but correspondence analysis does not utilize all possibilities resulting from considering the Aitchison geometry of compositional tables. Moreover, similar as for compositional data, it is natural to consider an ensemble of compositional tables that can be analyzed with statistical tools described in Chaps. 4–10 (including exploratory data analysis and visualization, principal component analysis, clustering, classification, regression analysis, etc.). This is a distinct difference to the case of contingency tables, where such issues are usually not considered or are subject to specific approaches, like using three-way contingency tables and the respective log-linear models.

Generally, compositional tables are formed by I rows and J columns (Egozcue et al. 2008, 2015; Fačevicová and Hron 2015; Fačevicová et al. 2016). Nevertheless, the next section will focus first on the simplest case of 2×2 compositional tables that seems to have the greatest practical potential (Fačevicová et al. 2014).

12.2 Independent and Interaction Parts of Compositional Tables

This section follows closely Fačevicová et al. (2014) and also Fačevicová and Hron (2015), where the case of 2×2 compositional tables was discussed in detail. Accordingly, a 2×2 compositional table is given as

$$\mathbf{x} = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix},$$

see examples in Tables 12.1 and 12.2, which represents the relationship between the row and column factors. Any such table can be vectorized row-wise into a four-part composition $\mathrm{vec}(\mathbf{x}) = (x_{11}, x_{12}, x_{21}, x_{22})'$. This formulation will be useful to study the geometrical properties of compositional tables in the sample space $\tilde{\mathcal{S}}^4$ (see (3.4) for a general definition) without a necessity of introducing their own sample space.

In general, the Aitchison geometry can be easily extended to the case of compositional tables. Perturbation and powering for 2×2 compositional tables $\mathbf{x}$ and $\mathbf{y}$ and a real number α, respectively, result in

$$\mathbf{x} \oplus \mathbf{y} = \begin{pmatrix} x_{11}y_{11} & x_{12}y_{12} \\ x_{21}y_{21} & x_{22}y_{22} \end{pmatrix}, \quad \alpha \odot \mathbf{x} = \begin{pmatrix} x_{11}^{\alpha} & x_{12}^{\alpha} \\ x_{21}^{\alpha} & x_{22}^{\alpha} \end{pmatrix}.$$

Note that

$$\mathbf{n} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

denotes, as usual, the neutral element in $\tilde{\mathcal{S}}^4$ as sample space of vectorized compositional tables. It immediately follows that the dimension of $\tilde{\mathcal{S}}^4$ is three, which also determines the number of orthonormal coordinates to be assigned to any 2×2 compositional table. The Aitchison inner product of two compositional tables $\mathbf{x}$ and $\mathbf{y}$ is defined as $\langle \mathbf{x}, \mathbf{y} \rangle_A =$

$$\frac{1}{4}\left(\ln\frac{x_{11}}{x_{12}}\ln\frac{y_{11}}{y_{12}} + \ln\frac{x_{11}}{x_{21}}\ln\frac{y_{11}}{y_{21}} + \ln\frac{x_{11}}{x_{22}}\ln\frac{y_{11}}{y_{22}} + \ln\frac{x_{12}}{x_{21}}\ln\frac{y_{12}}{y_{21}} + \ln\frac{x_{12}}{x_{22}}\ln\frac{y_{12}}{y_{22}} + \ln\frac{x_{21}}{x_{22}}\ln\frac{y_{21}}{y_{22}}\right).$$

Similar as for standard compositional data,

$$\|\mathbf{x}\|_A = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle_A} \quad \text{and} \quad d_A(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} \ominus \mathbf{y}\|_A$$

represent the Aitchison norm of a table $\mathbf{x}$ and the distance between two tables $\mathbf{x}$ and $\mathbf{y}$, respectively.

For the computation of the inner product, norm and distance, the R package **robCompositions** can be used. For example, the Aitchison distance of the data from Germany shown in Table 12.2 and the corresponding table on employment in Austria can be calculated as follows:

```
germany <- c(8377, 2200.4, 9956.9, 19202.7)
austria <- c(833.3, 192.8, 1056.8, 1969.6)
names(germany) <- names(austria) <- c("part-time_female", "part-time_male",
                                      "full-time_female", "full-time_male")
germany

## part-time_female   part-time_male full-time_female
##           8377.0           2200.4           9956.9
##   full-time_male
##          19202.7

austria

## part-time_female   part-time_male full-time_female
##            833.3            192.8           1056.8
##   full-time_male
##           1969.6

aDist(germany, austria)

## [1] 0.1448979
```

12.2.1 Decomposition of 2 × 2 Compositional Tables

For the purpose of a structural analysis, a 2×2 compositional table can be decomposed into two orthogonal tables, the *independence table* $\mathbf{x}_{ind}$ and the *interaction table* $\mathbf{x}_{int}$. This idea of the independence table is inspired by probability tables and the independence of random variables, where the corresponding independence table is formed by a product of row and column marginals. In the compositional case, the (arithmetic) marginals are replaced by the geometric ones, i.e., instead of amalgamation of rows and columns, geometric means of them are computed. As a result, two two-part compositions (for row and column marginals, respectively) are obtained,

$$\mathbf{g}_r = (\sqrt{x_{11}x_{12}}, \sqrt{x_{21}x_{22}})', \ \mathbf{g}_c = (\sqrt{x_{11}x_{21}}, \sqrt{x_{12}x_{22}})',$$

instead of taking two arithmetic marginals of a probability table,

$$\mathbf{a}_r = (x_{11} + x_{12}, x_{21} + x_{22})', \ \mathbf{a}_c = (x_{11} + x_{21}, x_{12} + x_{22})'.$$

For example, the arithmetic and the geometric marginals of Table 12.2 are:

```
germanytab <- matrix(germany, ncol=2)
rownames(germanytab) <- c("female", "male")
colnames(germanytab) <- c("part-time", "full-time")
addmargins(germanytab, FUN = mean)

## Margins computed over dimensions
## in the following order:
## 1:
## 2:
##        part-time full-time     mean
## female    8377.0    9956.9  9166.95
## male      2200.4   19202.7 10701.55
## mean      5288.7   14579.8  9934.25

addmargins(germanytab, FUN = gmean)

## Margins computed over dimensions
## in the following order:
## 1:
## 2:
##        part-time full-time    gmean
## female  8377.000   9956.90 9132.850
## male    2200.400  19202.70 6500.279
## gmean   4293.338  13827.49 7704.938
```

In addition to the fact that the resulting geometric marginals are scale invariant, the geometric mean is a natural form of aggregation in compositional data analysis in general (Pawlowsky-Glahn and Egozcue 2002; Egozcue et al. 2008). It is easy to verify that $\mathbf{x}_{ind}$ can be explicitly expressed as

$$\mathbf{x}_{ind} = \begin{pmatrix} x_{11}\sqrt{x_{12}x_{21}} & x_{12}\sqrt{x_{11}x_{22}} \\ x_{21}\sqrt{x_{11}x_{22}} & x_{22}\sqrt{x_{12}x_{21}} \end{pmatrix}.$$

From its construction, the independence table captures relative information between the rows and columns of $\mathbf{x}$, respectively, i.e. excluding relationships within them. Consequently, the interaction table carries information about interactions between both factors in $\mathbf{x}$.

The independence and interaction tables result in the decomposition of the compositional table $\mathbf{x}$ as

$$\mathbf{x} = \mathbf{x}_{ind} \oplus \mathbf{x}_{int} \tag{12.1}$$

(Egozcue et al. 2008; Fačevicová et al. 2014). The subspace of independence tables, $\tilde{\mathcal{S}}^4_{ind}$, has dimension two, representing the row/column ratios. The independence table would thus be sufficient to reconstruct the original 2×2 compositional table, if the two corresponding row and column factors were independent in the above sense.

Using Eq. (12.1), the interaction table is defined as

$$\mathbf{x}_{int} = \mathbf{x} \ominus \mathbf{x}_{ind} = \begin{pmatrix} \frac{1}{\sqrt{x_{12}x_{21}}} & \frac{1}{\sqrt{x_{11}x_{22}}} \\ \frac{1}{\sqrt{x_{11}x_{22}}} & \frac{1}{\sqrt{x_{12}x_{21}}} \end{pmatrix} = \begin{pmatrix} \sqrt{x_{11}x_{22}} & \sqrt{x_{12}x_{21}} \\ \sqrt{x_{12}x_{21}} & \sqrt{x_{11}x_{22}} \end{pmatrix};$$

by construction, tables $\mathbf{x}_{ind}$ and $\mathbf{x}_{int}$ are orthogonal, i.e. $\langle \mathbf{x}_{ind}, \mathbf{x}_{int} \rangle_A = 0$. It is worth to note that in case of the independence table, both the geometric and the arithmetic marginals coincide, up to proportional representation. For the interaction table, $\mathbf{g}_r$ and $\mathbf{g}_c$ are neutral elements, but this is not fulfilled for arithmetic marginals (Egozcue et al. 2015).

The independence and interaction tables can be obtained in R as follows:

```
ind2x2(germanytab)

##          part-time full-time
## female 0.13841322 0.4457853
## male   0.09851519 0.3172863
## attr(,"class")
## [1] "ind2x2"

int2x2(germanytab)

##        part-time full-time
## female 0.3652162 0.1347838
## male   0.1347838 0.3652162
## attr(,"class")
## [1] "int2x2"
```

Note that the functions `ind2x2` and `int2x2` are short versions of the more general implementation of $I \times J$ tables in the functions `indTab` and `intTab`.

```
# results equivalent to previous calculations, thus output is suppressed
xind <- indTab(germanytab)
intTab(prop.table(germanytab), xind)
```

12.2.2 Coordinate Representation of Compositional Tables

The decomposition of 2×2 compositional tables into independent and interactive parts is now used to derive an orthonormal coordinate representation with respect to the Aitchison geometry. The new coordinates follow the specific structure of compositional tables and enable to visualize the coordinates in real space as well as to perform further statistical processing.

The focus on a coordinate representation reflects the fact that the aim here is primarily to analyze **a sample of** compositional tables. Thus, the interest is not only in analyzing one compositional table as, for example, given in Table 12.2, but to analyze a sample of compositional tables, e.g. factors of Table 12.2 given for several countries. Note, however, that in special cases the analysis of compositional tables can also be considered as an alternative to independence testing in contingency tables (Egozcue et al. 2015; Fačevicová et al. 2014).

As already mentioned, it is possible to assign orthonormal coordinates, like pivot coordinates (3.19), to a vectorized compositional table $\text{vec}(\mathbf{x}) = (x_{11}, x_{12}, x_{21}, x_{22})'$, or, more general, to apply a sequential binary partition that leads to coordinates (3.37). However, this approach has some limitations for compositional tables.

The aim of the analysis of compositional tables is to decompose the original table into an independent and an interactive part, $\mathbf{x}_{ind}$ and $\mathbf{x}_{int}$, and to support a simple interpretability of the results. Suppose that the original data which form the basis of the 2×2 compositional table $\mathbf{x}$ are given. Then it is expected that also the orthonormal coordinates $\mathbf{z}$ of these original data can be decomposed as

$$\mathbf{z} = \mathbf{z}_{ind} + \mathbf{z}_{int}, \tag{12.2}$$

and that these parts (vectors) reflect the dimensionality of $\mathbf{x}_{ind}$ and $\mathbf{x}_{int}$. Concretely, the numbers of nonzero coordinates in the three-component vectors $\mathbf{z}_{ind}$ and $\mathbf{z}_{int}$ should correspond to the dimensions of the respective subspaces. Unfortunately, this cannot be achieved for coordinates (3.19). By taking the more general sequential binary partitioning (3.37), the only possibility is given by the partitioning in Table 12.3, resulting in coordinates

$$\tilde{z}_1 = \frac{1}{2} \ln \frac{x_{11}x_{22}}{x_{12}x_{21}},\ \tilde{z}_2 = \frac{1}{\sqrt{2}} \ln \frac{x_{12}}{x_{21}},\ \tilde{z}_3 = \frac{1}{\sqrt{2}} \ln \frac{x_{11}}{x_{22}}. \tag{12.3}$$

Indeed, for $\mathbf{z}_{ind} = (0, \tilde{z}_2, \tilde{z}_3)'$ and $\mathbf{z}_{int} = (\tilde{z}_1, 0, 0)'$ the relation (12.2) holds. Unfortunately, these coordinates cannot be easily extended to the general case of compositional tables with I rows and J columns (Fačevicová et al. 2014). This can be achieved with coordinates

$$z^{int} = \frac{1}{2} \ln \frac{x_{11}x_{22}}{x_{12}x_{21}},\ z_1^{ind} = \frac{1}{2} \ln \frac{x_{11}x_{12}}{x_{21}x_{22}},\ z_2^{ind} = \frac{1}{2} \ln \frac{x_{11}x_{21}}{x_{12}x_{22}} \tag{12.4}$$

Table 12.3 Sequential binary partitioning of a vectorized 2×2 compositional table

	$k =$		
	1	2	3
x_{11}	+	0	+
x_{12}	−	+	0
x_{21}	−	−	0
x_{22}	+	0	−
p_k	2	1	1
m_k	2	1	1

(see also Fačevicová and Hron 2015), which are consistent with the general approach to a coordinate representation of compositional tables (Fačevicová et al. 2016, 2018). This representation is equivalent with the implementation in the function `ilr.2x2`. The output in `$z1` corresponds to the coordinates defined in Eq. (12.3).

```
ilr.2x2(germanytab)$z1

## [1]  0.9968179  1.0674672 -0.5865882
```

Although all coordinates in Eq. (12.4) have the same structure as logratios of two two-part groups of compositional parts, their interpretation should be accommodated according to independence and interaction tables they represent. While the coordinate z^{int}, assigned to the interaction table and contained also in (12.3), can be interpreted in terms of odds ratios (Agresti 2012), coordinates z_1^{ind} and z_2^{ind} of the independence table should be rather interpreted as balances. By doing so, the latter coordinates represent relative information between rows and columns of $\mathbf{x}$, respectively, except for the interactions between them that are left for coordinate z^{int}. In this context, z^{int} is called also *quaternary coordinate*, being of primary interest in practice. For the case of independence between factors, the quaternary coordinate would equal to zero. As a consequence, the further it deviates from zero (in a positive or negative) in a sample of compositional tables, the stronger the relation between both factors.

12.3 Extension to the General Case

For an $I \times J$ compositional table

$$\mathbf{x} = \begin{pmatrix} x_{11} & \cdots & x_{1J} \\ \vdots & \ddots & \vdots \\ x_{I1} & \cdots & x_{IJ} \end{pmatrix}$$

all concepts from Sect. 12.2 (geometrical properties, decomposition of the compositional table into independent and interactive parts) can be extended in a straightforward manner (Egozcue et al. 2008, 2015). Particularly, the dimensionality

of $\mathbf{x}$ now equals to $IJ-1$, i.e. one less than the number of cells (parts) of the table. The independence and interaction tables have dimensions $I+J-2$ and $(I-1)(J-1)$, respectively, and sum up to the dimension of the original table. For practical purposes, just the coordinate representation of general compositional tables (Fačevicová et al. 2016, 2018) is briefly discussed in the following.

It was stated in Sect. 12.2.2 that an interpretation in terms of balances is in general not suitable for compositional tables. Because their cells represent relationships between two factors, only considering two groups of parts into a coordinate would not take the two-factor nature of these observations into account. In fact, balances are suitable for extracting information from the two factors individually, thus dealing with the rows and columns of the table. To represent inter-factorial patterns, coordinates in the form of (log) odds ratios between four groups of parts are preferable, similar to the case of contingency tables (Agresti 2012).

For the construction of coordinates of an $I \times J$ independence table, first an SBP of the entire rows (columns) of a compositional table $\mathbf{x}$ is considered, which is denoted by SBPr (SBPc) in the following. This partition is in line with the nature of the levels of row (column) factors, and follows a standard SBP. In each of the $I-1$ $(J-1)$ steps, the levels with some common properties are separated from the others. The first $I+J-2$ coordinates z_i^r and z_j^c of the $I \times J$ compositional table $\mathbf{x}$ are given as

$$z_i^r = \sqrt{\frac{s \cdot t \cdot J}{s+t}} \ln \frac{\left[g(\mathbf{x}_{j_1.}) \cdots g(\mathbf{x}_{j_s.})\right]^{1/s}}{\left[g(\mathbf{x}_{k_1.}) \cdots g(\mathbf{x}_{k_t.})\right]^{1/t}}, \quad \text{for} \quad i = 1, 2, \ldots, I-1 \tag{12.5}$$

and

$$z_j^c = \sqrt{\frac{u \cdot v \cdot I}{u+v}} \ln \frac{\left[g(\mathbf{x}_{.l_1}) \cdots g(\mathbf{x}_{.l_u})\right]^{1/u}}{\left[g(\mathbf{x}_{.m_1}) \cdots g(\mathbf{x}_{.m_v})\right]^{1/v}}, \quad \text{for} \quad j = 1, 2, \ldots, J-1, \tag{12.6}$$

where s, t (u, v) are the numbers of rows (columns) involved in the ith (jth) step of the SBP, the indices $(j_1., \ldots, j_s.)$ and $(k_1., \ldots, k_t.)$, or $(.l_1, \ldots, .l_u)$ and $(.m_1, \ldots, .m_v)$ specify the rows/columns, and $g(.)$ stands for the geometric mean.

In addition to balance-like coordinates, also odds ratios for the interaction table must be determined to obtain $IJ-1$ orthonormal coordinates. These coordinates are called *quaternary (logratio)* coordinates. They are orthogonal to the first $I+J-2$ variables, and represent a generalization of sequential binary partitioning (Fačevicová et al. 2018). It is based on the partitioning of the parts of the compositional table into four groups (blocks) in a systematic manner that results in coordinates in form of a log odds ratio between these four groups (marked as A (upper left), B (upper right), C (lower left) and D (lower right))

$$z^{OR} = \sqrt{\frac{a \cdot d}{a+b+c+d}} \ln \frac{\left(x_{i_1} \cdots x_{i_a}\right)^{1/a} \left(x_{l_1} \cdots x_{l_d}\right)^{1/d}}{\left(x_{j_1} \cdots x_{j_b}\right)^{1/b} \left(x_{k_1} \cdots x_{k_c}\right)^{1/c}}, \tag{12.7}$$

where a, b, c, d are the numbers of parts in groups A, B, C, and D, respectively and $i_\cdot, j_\cdot, k_\cdot, l_\cdot$ are the indices of those parts. The construction of quaternary coordinates (12.7) for the interaction table is done using a combination of row and column SBP. For the first step of SBPr applied to the rows of the table, all $J-1$ steps of SBPc are performed. The first $J-1$ coordinates are obtained in accordance with (12.7). The next $J-1$ coordinates are obtained by applying the second step of SBPr to the rows and all of the steps of the SBPc to the columns, and so on, until all $I-1$ steps of the SBPr have been completed. All $(I-1)(J-1)$ coordinates of z^{OR} thus result from a successive application of all steps of the SBPr combined with repeated use of all steps of the SBPc, or conversely.

A specific choice of SBPr and SBPc according to pivot coordinates (3.19) leads to $(I-1)(J-1)$ *pivot quaternary coordinates* (Fačevicová et al. 2016),

$$z_{rc} = \frac{1}{\sqrt{r \cdot c \cdot (r-1) \cdot (c-1)}} \ln \frac{\prod_{i=1}^{r-1} \prod_{j=1}^{c-1} \left(x_{ij} x_{rc}\right)}{\prod_{i=1}^{r-1} \prod_{j=1}^{c-1} \left(x_{ic} x_{rj}\right)}. \tag{12.8}$$

Here, one group of the odds ratio is always formed by a single pivot part $x_{rc}, r = 2, \ldots, I$ and $c = 2, \ldots, J$, which determines the lower right corner of a partial table, represented by the respective coordinate. These coordinates can also be seen as a scaled sum of log odds ratios according to some logical scheme, all containing part x_{rc}. This follows from (12.8).

It is natural that quaternary coordinates of the interaction table, revealing relationships between both factors, are of primary interest in applications. For concrete examples of their use in statistical processing, have a look at Fačevicová et al. (2016, 2018).

The above coordinate representation can be practically applied using the function `coord` in the R package **robCompositions**. Nevertheless, for a better understanding, the coordinates will be defined directly in the R code of the following examples.

12.4 Examples

Compositional tables occur more frequently in practice than it might be expected. In Egozcue et al. (2015) and Fačevicová et al. (2014, 2016, 2018) concrete examples from official statistics, genetics, health and economics are discussed. Further typical examples in this context are input–output tables. They describe the sale and purchase relationships between producers and consumers within an economy, and the relative structure is of primary interest. When more countries are considered simultaneously, input–output tables provide important information about the international trade structure (Timmer et al. 2015). Here two other examples from the OECD statistics database (http://stats.oecd.org/) are taken. These data sets have been made available in the **robCompositions** package.

12.4.1 Gender Based Cancer Data

Many human disorders are directly related to the gender of the patients, either due to genetic reasons or because of their prevalent life style. The aim is to analyze how the role of gender (female, male) interacts with the relative structure of two main types of malignant neoplasms within a population, affecting colon and lung, respectively. For this purpose, data from 35 OECD countries were collected (OECD 2012). The data are accessible in the R package **robCompositions** as data set `cancerMN`.

```
# load the cancerMN data set
data("cancerMN")
# first three observations
head(cancerMN, 3)

##   country females-colon females-lung males-colon males-lung
## 1     AUS     0.2602574    0.1697059   0.3231618  0.2468750
## 2     AUT     0.2205291    0.1825397   0.2952381  0.3016931
## 3     BEL     0.2358439    0.1236269   0.2911331  0.3493961
```

For each country, the values can be arranged in a 2×2 table. Here, the rows of the tables will be "female" and "male", and the columns "colon" and "lung." The coordinate representation for the resulting 2×2 tables, see Eq. (12.4), is formed by one coordinate z^{int} for the interaction table and two coordinates z_1^{ind}, z_2^{ind} for the independence table. The following R code can be used to calculate coordinates for the interaction and independence tables.

```
# extend the data set cancerMN
cancerMN$Index <- 1:nrow(cancerMN)
# interaction coordinate
cancerMN$z_int <- 0.5 *
  log((cancerMN$`females-colon` * cancerMN$`males-lung`) /
      (cancerMN$`females-lung` * cancerMN$`males-colon`))
# independent coordinates
cancerMN$z_ind1 <- 0.5 *
  log((cancerMN$`females-colon` * cancerMN$`females-lung`) /
      (cancerMN$`males-colon` * cancerMN$`males-lung`))
cancerMN$z_ind2 <- 0.5 *
  log((cancerMN$`females-colon` * cancerMN$`males-colon`) /
      (cancerMN$`females-lung` * cancerMN$`males-lung`))
```

This provides information about the strength of the relationship between the factors female/male and colon/lung, and the balances within the single factors.

In Fig. 12.1, the values of the interaction coordinate are displayed. Higher values indicate departure from independence between both factors, and thus there would be an effect of gender and/or of the type of cancer. Regional differences can clearly be seen in this plot. For example, some Eastern European countries (Belarus, Estonia and Latvia), but also some countries from Southern Europe (Greece, Turkey, Spain, Portugal, Italy) have high values for this coordinate. On the other hand, the lowest values for the coordinate z^{int} occur for highly developed countries from

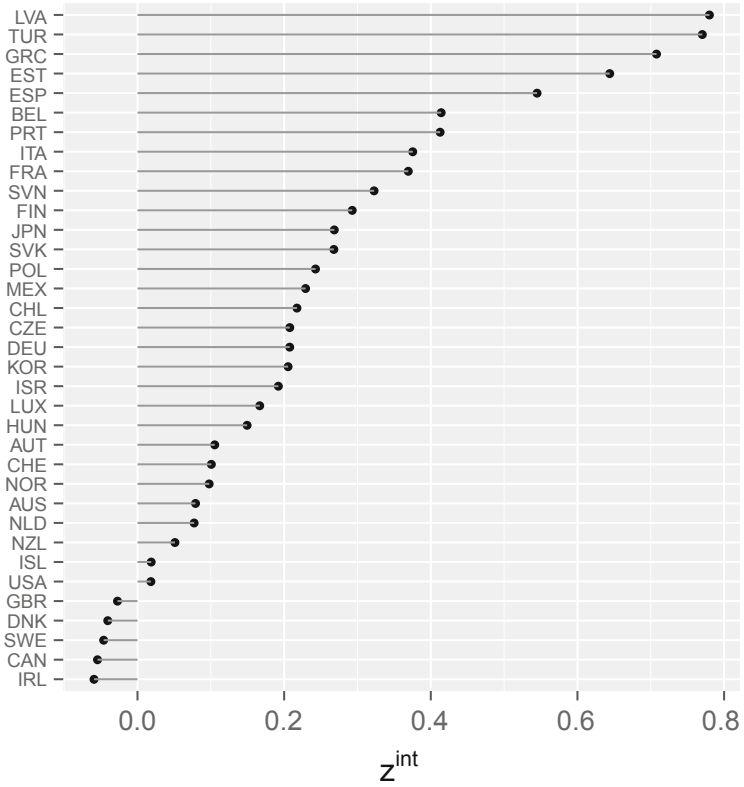


Fig. 12.1 Values of the interaction table coordinate for the different countries from the `cancerMN` data set using a special choice of coordinates, see Eq. (12.4)

Western and Northern Europe, accompanied by the USA, Canada, and Australia. Some reasoning for that can be derived from displaying the coordinates of the independence tables, see Fig. 12.2. Here, z_1^{ind} represents the balance between female and male contributions to the overall population of diseased by aggregating both types of cancer, while z_2^{ind} quantifies the balance between colon and lung malignant neoplasms for aggregated information about gender. Negative values on the first coordinate indicate that the departure from independence is caused by a dominance of cancer occurrence by men, which is particularly the case for some Southern European countries. Negative values on the second coordinate indicate that the departure from independence is caused by a dominance of lung cancer, and this happens for Greece and Turkey. On the other hand, in Spain and Portugal this is caused by a dominance of colon cancer.

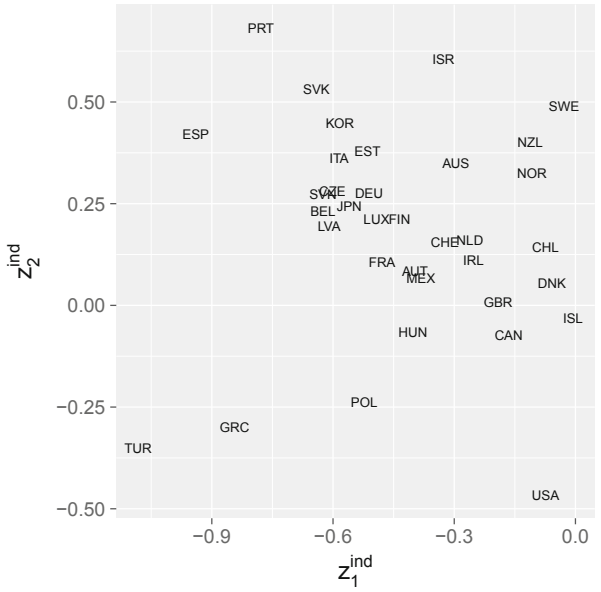


Fig. 12.2 Coordinates of independence tables obtained from the `cancerMN` data set using a special choice of coordinates, see Eq. (12.4)

12.4.2 Social Expenditures According to Funding Sources

In the second example, the social expenditures according to source (public, private) and three important branches (health, old age, incapacity related) are studied (OECD 2010). Accordingly, 2×3 compositional tables have to be analyzed. Due to the fact that not in all OECD countries both sources were considered (by law, or just by not recording them), altogether just 20 tables were collected as data set `socExp` in the **robCompositions** package. In order to get a coordinate representation of both interaction and independence tables, row and column sequential binary partitions (SBPr and SBPc, respectively) need to be considered first. Due to the definition of the sequential binary partition (see Sect. 3.3.5), it is sufficient for SBPr to separate public and private sources in one step. For SBPc two steps are needed. At first, the health branch is separated from the latter two and in the next step, old age and incapacity related branches are separated into one-part groups. As a result, the following coordinates of the independence table,

$$z^r = \sqrt{\frac{3}{2}} \ln \frac{\sqrt[3]{x_{11}x_{12}x_{13}}}{\sqrt[3]{x_{21}x_{22}x_{23}}},\ z_1^c = \sqrt{\frac{4}{3}} \ln \frac{x_{11}x_{21}}{\sqrt{\sqrt{x_{12}x_{22}}\sqrt{x_{13}x_{23}}}},\ z_2^c = \ln \frac{\sqrt{x_{12}x_{22}}}{\sqrt{x_{13}x_{23}}}, \tag{12.9}$$

and quaternary coordinates (those of the interaction table)

$$z_1^{OR} = \frac{1}{\sqrt{3}} \ln \frac{x_{11}\sqrt{x_{22}x_{23}}}{\sqrt{x_{12}x_{13}}x_{21}}, \; z_2^{OR} = \frac{1}{2} \ln \frac{x_{12}x_{23}}{x_{13}x_{22}} \tag{12.10}$$

are obtained. These can be computed in R as follows:

```
# load social expenditures data
data("socExp")
# first three observations
head(socExp, 3)

##   country     currency health-public old-public incap-public
## 1     AUT         Euro     0.2884688  0.5210435   0.10395608
## 2     BEL         Euro     0.3832669  0.3927777   0.13113643
## 3     CHL Chilean Peso     0.3421648  0.3031592   0.09110255
##   health-private old-private incap-private
## 1     0.02100887  0.02846553    0.03705719
## 2     0.02072542  0.05181970    0.02027389
## 3     0.11654890  0.12243393    0.02459059
```

```
x <- socExp[, 3:8]
# interaction coordinates
socExp$z_int1 <- (1 / sqrt(3)) * log((x[,1] * sqrt(x[,5] * x[,6])) /
                                     (x[,4] * sqrt(x[,2] * x[,3])))
socExp$z_int2 <- 0.5 * log((x[,2] * x[,6]) / (x[,3] * x[,5]))
# independent coordinates
socExp$z_r1 <- sqrt(3 / 2) * log(((x[,1] * x[,2] * x[,3])^(1/3)) /
                                 ((x[,4] * x[,5] * x[,6])^(1/3)))
socExp$z_c1 <- sqrt(4 / 3) * log(sqrt(x[,1] * x[,4]) /
                 sqrt(sqrt(x[,2] * x[,5]) * sqrt(x[,3] * x[,6])))
socExp$z_c2 <- log(sqrt(x[,2] * x[,5])/sqrt(x[,3] * x[,6]))
```

According to the construction of the coordinates, z^r can be interpreted as balance between public and private sources according to the aggregated branches. Further, z_1^c captures the dominance of health expenditures to old age and incapacity related ones conditional to aggregated sources. The third coordinate of the independence table, z_2^c, can be interpreted similarly, as the balance between old age and incapacity related expenditures by aggregated sources. Nonzero values of the first quaternary coordinate indicate a departure from the independence case related to health expenditures (as z_1^{OR} is the only quaternary coordinate containing them), while z_2^{OR} does the same within the sub-table determined by the cells x_{12}, x_{13}, x_{22}, x_{23}.

Because of the dimensionality of the interaction table (two coordinates) and the independence table (three coordinates), they both can still be displayed without the necessity of using any dimension reduction method, e.g., principal component analysis, see Chap. 7. From Fig. 12.3, where quaternary coordinates are displayed, it is easy to see that in countries like Sweden, Czech Republic, and Great Britain the relative expenditures in health (with respect to other branches) form an important source of dependence between both factors. Similarly, by considering the coordinate z_2^{OR} and the 2×2 subtable given by x_{12}, x_{13}, x_{22}, x_{23}, the dependence between the factors can be observed also for France, Greece, and Portugal on the one hand

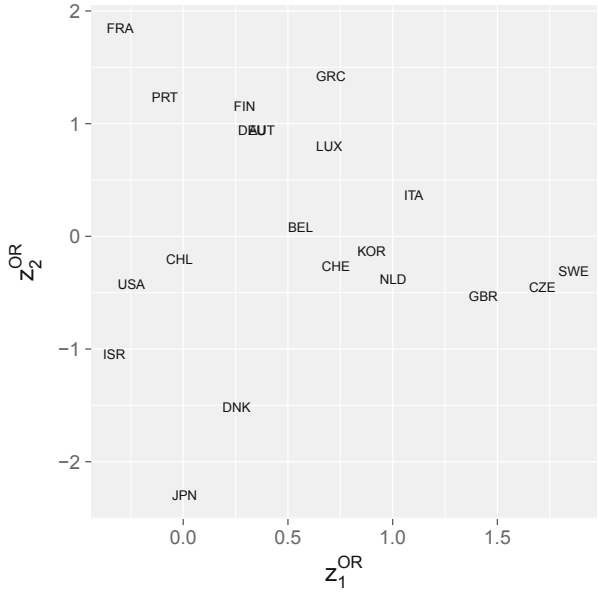


Fig. 12.3 Coordinates of interaction tables for the social expenditures quaternary coordinate representation based on Eq. (12.10)

(represented by high values of the coordinate), and for Denmark and Japan on the other hand (low values). Finally, countries like the USA and Chile, with almost zero values on both coordinates, approach independence between factors.

The coordinates of the independence table (Fig. 12.4) help to reveal possible reasons for either dependence or independence tendencies. In Sweden and the Czech Republic they are caused obviously by a dominance of expenditures on aggregated old age and incapacity related branches with respect to health (low values of z_1^c) in combination with dominating state sources (high values of z^r). Dominating state sources can also be observed for the countries Finland, Denmark, and Italy. Exceptionally dominating state sources are also present in Japan, accompanied by a remarkably high dominance of old age over incapacity related expenditures (z_2^c) and high overall health expenditures (z_1^c). This might be related to the age structure of the population in Japan. Another outlying observation is formed by the USA with higher relative health expenditures, covered mainly from private sources (very low value of z_r). However, here specific combinations of both factors do not influence the relative structure of the table. As a final note, remind that for the interpretation of the coordinates z^r, z_1^c and z_2^c in the sense of one factor, the aggregated values of the latter factor are assumed.

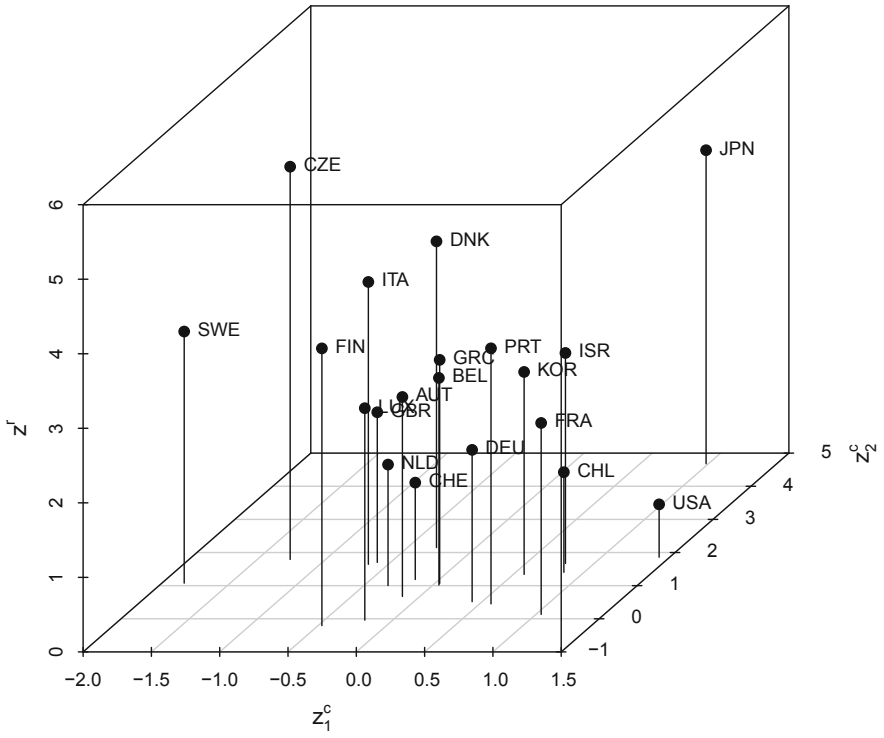


Fig. 12.4 Coordinates of the independence table calculated by Eq. (12.9)

References

A. Agresti, *Categorical Data Analysis*, 3rd edn. (Wiley, Chichester, 2012)

J.J. Egozcue, J.L. Díaz-Barrero, V. Pawlowsky-Glahn, Compositional analysis of bivariate discrete probabilities, in *Proceedings of CODAWORK'08, The 3rd Compositional Data Analysis Workshop*, ed. by J. Daunis-i-Estadella, J.A. Martín-Fernández (University of Girona, Girona, 2008)

J.J. Egozcue, V. Pawlowsky-Glahn, M. Templ, K. Hron, Independence in contingency tables using simplicial geometry. Commun. Stat. Theory Methods **44**(18), 3978–3996 (2015)

K. Fačevicová, K. Hron, Covariance structure of compositional tables. Austrian J. Stat. **44**(3), 31–44 (2015)

K. Fačevicová, K. Hron, V. Todorov, D. Guo, M. Templ, Logratio approach to statistical analysis of 2×2 compositional tables. J. Appl. Stat. **41**(5), 944–958 (2014)

K. Fačevicová, K. Hron, V. Todorov, M. Templ, Compositional tables analysis in coordinates. Scand. J. Stat. **43**(4), 962–977 (2016)

K. Fačevicová, K. Hron, V. Todorov, M. Templ, General approach to coordinate representation of compositional tables. Scand. J. Stat. (2018). https://doi.org/10.1111/sjos.12326

M. Greenacre, *Correspondence Analysis in Practice*, 2nd edn. (Chapman & Hall/CRC Press, Boca Raton, 2007)

M. Greenacre, Compositional data and correspondence analysis, in *Compositional Data Analysis: Theory and Applications*, ed. by V. Pawlowsky-Glahn, A. Buccianti (Wiley, Chichester, 2011), pp. 104–113

OECD, OECD.Stat, Social expenditure - aggregated data (2010). Accessed 30 June 2013

OECD. OECD.Stat, Health status: cancer (2012). Accessed 30 June 2013

V. Pawlowsky-Glahn, J.J. Egozcue, BLU estimators and compositional data. Math. Geol. **34**(3), 259–274 (2002)

M.P. Timmer, E. Dietzenbacher, B. Los, R. Stehrer, G.J. de Vries, An illustrated user guide to the world input–output database: the case of global automotive production. Rev. Int. Econ. **23**(3), 575–605 (2015)

UNECE, Statistical database of United Nations Economic Commission for Europe (2013). http://w3.unece.org/. Accessed 18 September 2013

UNIDO, *International Yearbook of Industrial Statistics* (United Nations Industrial Delevopment Organization, Vienna, 2009)

Chapter 13
Preprocessing Issues

Abstract In practice, for many compositional data sets it will not be possible to apply the outlined statistical methods immediately, and there will be a need for a preprocessing. This is the case in presence of missing values in some compositional parts, but also when zero values occur, which are basically excluded from the definition of compositional data. According to the nature of zero values, rounded and structural zeros need to be considered. Rounded zeros occur when either small values of components are rounded to zeros, or a measurement device has incorporated a detection limit (number) that automatically sets values below this limit to zero. Therefore, their replacement by a small positive value is reasonable. Count zeros are similar to rounded zeros. They result from insufficient sample size from the underlying distribution (mostly from a multinomial distribution) that drives compositions consisting of counts. For the imputation of missing values, rounded and count zeros, model-based algorithms have been developed and are available in R. In contrast, structural zeros are a result of a structural process, and thus imputing them to obtain a full data set is not meaningful. There are several approaches how compositional data with structural zeros can be (indirectly) processed using the logratio methodology. One possibility is to impute structural zeros in an auxiliary step to estimate the overall location and covariance, followed by a statistical analysis in groups of subcompositions according to the zero patterns.

13.1 Specific Problems with Data Preprocessing of Compositions

Your data set includes zeros and therefore you cannot apply logratio techniques, since this would result in a division by zero? The measurement unit did not work for all measured values and you have thus some non-measured values in your data set that do not allow to apply compositional methods in a straightforward manner (division by a non-available)? Concentrations of some chemical elements in some samples were too low and could not be measured and you are not sure if you should replace these concentrations with a positive constant? This chapter tries to give answers to these questions.

P. Filzmoser et al., *Applied Compositional Data Analysis*, Springer Series in Statistics, https://doi.org/10.1007/978-3-319-96422-5_13

Compositional data, like any other multivariate observations, are not free from imprecisions resulting from the measurement process. Though, due to their relative nature, there are some peculiarities that are worth to be considered in advance. Particularly, observations are rarely measured in terms of ratios, or any similar way that would invoke their compositional nature. Instead, almost always the absolute values are produced and it depends on the purpose of the analysis, whether the input data are considered as compositional or not. For example, in chemometrics it is usual to adjust measurements to an internal standard, provided in absolute values, or add a certain positive value to avoid negative and/or zero values resulting from calibration of the measurement device. Clearly, the latter adjustment could completely destroy the source information, conveyed by (log)ratios between the components. The influence can be particularly severe for small (absolute) concentrations, resulting from the relative scale of the compositions. For example,

$$\ln \frac{0.1}{0.2} = -0.69, \ \ln \frac{0.1+0.1}{0.2+0.1} = -0.41,$$

while for higher concentrations the distortion is not as big,

$$\ln \frac{1}{2} = -0.69, \ \ln \frac{1+0.1}{2+0.1} = -0.65.$$

Although such adaptations are often done with the aim to enable further processing using the logratio techniques, they should be suppressed whenever possible. They are definitely not needed in presence of zeros, because other (relevant) methods are available to cope with the issue.

Also from another perspective, absolute information in compositional data is relevant for preprocessing. For instance, for geochemical data or data from chemometrics, the detection limit of measurement devices needs to be taken into account. A detection limit is characterized as absolute number, and it may differ among different variables and even among different samples (coming, e.g., from different laboratories). Often, a value below the detection limit is simply set to zero, to half of the value of the detection limit, or to the negative value of the detection limit. Values below the detection limit are known under the name *rounded zeros* (Aitchison 1986; Martín-Fernández et al. 2003), or more precisely as *values below the detection limit*.

An example of data with rounded zeros is shown below. The moss layer of the soil samples of the Kola data was already used in Sect. 5.4. Now the C-horizon can be used, which represents the soil samples deeper under the surface. Some rounded zeros are present in the data, e.g. in row 234, the values of `As` and `S` are zero.

```
library("robCompositions")
data("chorizonDL")
# select variables with zeros included
vars <- which(colSums(chorizonDL == 0) > 0)
# exclude variable ASP (not of interest here)
vars <- names(vars[-length(vars)])
compNA <- chorizonDL[, vars]
# example zeros
compNA[233:234, ]

##        Ag  As    Bi Co_INAA    K Nd_INAA  S  Sc
## 233 0.006 0.2 0.035      13 1100      13 22 2.2
## 234 0.015 0.0 0.032       8  300      11  0 1.2

# number of zeros for each variable
colSums(compNA == 0)

##      Ag      As      Bi Co_INAA       K Nd_INAA       S
##       1      10      15       1       3       8       3
##      Sc
##       1

# detection limit for these variables
attributes(chorizonDL)$DL[vars]

##      Ag      As      Bi Co_INAA       K Nd_INAA       S
##   1e-03   1e-01   5e-03   1e+00   2e+02   5e+00   5e+00
##      Sc
##   1e-01
```

Even though the detection limits are usually treated as compositional parts, they implicitly include also absolute information (van den Boogaart and Tolosana-Delgado 2013). Obviously, this is inconsistent with the Aitchison geometry that assumes purely relative contributions of parts. One possibility would be to incorporate the absolute information into a model to impute rounded zeros by reasonable values under the detection limit, like the one proposed in van den Boogaart et al. (2015).

Preprocessing of multivariate data is often characterized by imputation, when absent or improper values are replaced by some acceptable alternatives. With compositional data, such imputation should produce values that reflect well the multivariate compositional data structure, formed by the logratios between the parts. For example, in the last code chunk there is a rounded zero value in row 234 of variable `As`; the value is simply coded with 0, and the detection limit for `As` is 0.1. After an appropriate imputation, the value should be in the interval $(0, 0.1)$ for the given representation of compositional data (here mg/kg), and the imputation should consider information of all other compositional parts. Furthermore, any logratio between other parts, like the logratio 0.015/0.032 between `Ag` and `Bi`, should remain unchanged after imputation of the rounded zeros.

For other data sets and problems, the imputation may be context dependent, i.e. other constraints might be important as well. One possible requirement is to preserve a constant sum constraint of the components (like 1 or 100, or, for example, the total consumption in household consumption data) even after the imputation step

is performed. Once the imputed value is considered as a concrete absolute number, it seems to be natural to adjust the other parts in the composition, preferably in a multiplicative way (Martín-Fernández et al. 2003, 2011). In this book, another philosophical perspective is followed, the imputed values are immediately considered as those carrying relative information, thus being subject to any representation within the respective equivalence class of proportional positive vectors. Such an approach simplifies further considerations and is inherently contained in all methods that follow.

13.2 Missing Values

In practice it often happens that values of compositional parts are not reported and the corresponding cells in the data matrix are empty, typically coded as symbol `NA` (not available) in R. This can happen, e.g., in household expenditure studies, where the respondents tend to omit controversial questions, like those on expenditures on alcohol/tobacco. Another source of missing values is the failure of a measurement device that erroneously did not report concentrations of element(s) in a composition. The latter case corresponds to *missings completely at random* (abbreviated as MCAR), where missing values do not depend on observed or unobserved measurements (Little and Rubin 2002). In other words, under MCAR, the analysis of only those units with complete data results in unbiased point estimates, but the variances of the estimators are in any case underestimated, because the sample size is smaller when only the complete cases are used.

The previous case of household expenditures corresponds rather to the case of *missings at random* (MAR), where—given the observed data—the missingness mechanism does not depend on the unobserved data. The particular case of MAR can also be related to non-compositional parts in the data set or it can be related to absolute information in compositions. For example, the combination of age, gender, and region may result in higher probabilities of missingness in expenditures. For example, in Austria the probability of missingness in income components of men in the age between 35 and 45 living in Vienna as the capital city with many job opportunities and high standard of living is higher than for elder people in the rather less economically developed region of Burgenland. Even cases of MNAR (*missing not at random*) can be present in data, for example when the probability of missingness in a wage component is dependent on the absolute value of the wage of a person. Here a possible solution could be to ask directly for relative values in proportions or percentages, and thus to "hide" the absolute wages, but it is not always feasible to influence the design of the study in advance.

Most statistical methods cannot be applied directly to compositional data containing missing values. Moreover, expressing data in logratio coordinates would lead to a further expansion of the missing information. At the same time, deleting such observations from the data set would result in an unacceptable loss of information and in biased estimates. Instead, the missing cells need to be imputed first and

then one can continue with further statistical processing. As compositional data are multivariate, it is not recommendable to replace missing values by the geometric mean of each compositional part. Also if the replacement would be done with a proper robust counterpart (see Sect. 5.2.4), this would possibly be useful only for data sets with a very small amount of missing values, otherwise it would lead to an artificial underestimation of the compositional variability. The more advisable multivariate methods are based on similarities among the objects and/or variables. Among them, a popular tool is distance-based k-nearest neighbor ($k$nn) imputation, where the information of the nearest $k \geq 1$ complete observations is used to estimate the missing values. Another well-known procedure is the EM (expectation maximization) algorithm (Dempster et al. 1977), which uses the relations between observations and variables for estimating the missing cells in a data matrix. Further details, as well as methods based on multiple regression and principal component analysis are described in Little and Rubin (2002) and Schafer (1997). Although most of these methods can cope with both types of missing values (MCAR and MAR), they ultimately need to be adapted for compositional data. Particularly, the imputation of missing values (in terms of the original compositional parts) cannot be done without considering logratios of the missing parts to other parts in the composition.

An obvious peculiarity of imputation with compositional data is the necessity to cope with imputed absolute values in a vector whose components represent relative contributions on a whole. According to the previous section, once the compositional parts are imputed, they are immediately considered as those carrying relative information, possibly with any appropriate representation of this information. This is principally different to Martín-Fernández et al. (2003), one of the rare references that deals with missing values in compositional data. Here the estimation of missing values in compositional data is done in the sense of the Aitchison geometry, but with a prescribed constraint of a constant sum of the parts. In any case, the imputed values need to be adapted according to the actual total of the observed parts, as discussed for two methods (cf. Hron et al. 2010) that are introduced below. They both use multivariate data information for imputation, though each of them from another perspective.

13.2.1 k-Nearest Neighbor (knn) Imputation

The idea of the $k$nn imputation method (Troyanskaya et al. 2001) is to use a distance measure for finding the k most similar observations to a composition containing missings, and to replace the missing values by using the available variable information of the neighbors.

In the context of compositional data, the Aitchison distance (defined in Sect. 3.9) seems to be the first choice for such a measure. When a composition contains missing values in several cells, the imputation is done sequentially (one cell after the other), by searching the k nearest neighbors among the observations where

all information corresponding to the non-missing cells plus the information in the variable to be imputed is available (Hron et al. 2010). This option is preferred among possible alternatives, because it enables that the k observations can change during the sequential imputation. In addition, more neighbors are considered for imputation, and requesting more information per observation thus leads to a more reliable imputation result. For imputing a missing part of a composition, the median of the corresponding cells of the k nearest neighbors is used. Because ratios between parts are the same for proportionally equivalent compositions, the cells first need to be adjusted according to the overall size of the parts.

Specifically, for a compositional data set $\mathbf{x}_i = (x_{i1}, \ldots, x_{iD})', i = 1, \ldots, n$, let $M_i \subset \{1, \ldots, D\}$ denote the set of indexes referring to the missing cells of $\mathbf{x}_i$. Then $O_i = \{1, \ldots, D\} \backslash M_i$ refers to the observed parts of $\mathbf{x}_i$. For imputing a missing cell x_{ij}, for any $j \in M_i$, among all remaining compositions those are considered which have non-missing parts at positions j and O_i, and the k nearest neighbors $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_k}$ to the composition $\mathbf{x}_i$ using the Aitchison distance are computed. The jth cell of all k nearest neighbors is of interest for imputation. First these cells have to be adjusted by factors comparing the size of the parts in O_i. The adjustment factors can be taken as

$$f_{ii_l} = \frac{\sum\limits_{o \in O_i} x_{io}}{\sum\limits_{o \in O_i} x_{i_l o}} \qquad \text{for } l = 1, \ldots, k. \tag{13.1}$$

Using these factors as weights for the observations makes the k nearest neighbors comparable. The imputed value replacing the missing cell x_{ij} is

$$x_{ij}^* = \text{median}\{f_{ii_1} x_{i_1 j}, \ldots, f_{ii_k} x_{i_k j}\}\,. \tag{13.2}$$

By taking the median, robustness to outliers in the jth parts of the k nearest neighbors is obtained.

Although the choice of the adjustments in Eq. (13.1) is coherent with the definition of compositional data, a more robust version could be preferable. This can be achieved by using the adjustment factors

$$f_{ii_l}^* = \frac{\underset{o \in O_i}{\text{median}}\, x_{io}}{\underset{o \in O_i}{\text{median}}\, x_{i_l o}} \qquad \text{for } l = 1, \ldots, k, \tag{13.3}$$

which leads to more stable results for contaminated data.

$k$nn imputation is numerically stable (no iterative scheme is required), but it has also some limitations (Hron et al. 2010). Particularly, the optimal number k of nearest neighbors has to be determined. This is usually done within a simulation, by randomly setting observed cells to missing, estimating these missings based on different choices for the number k, and measuring the error between the imputed

and the originally observed values. The k producing the smallest error can be considered as optimal. A further limitation concerns small sample sizes, where the Aitchison distance can lead to nearest neighbors that contain much worse information for estimating the missing values than data points being further away. Therefore, whenever small sample sizes relative to the number of parts occur one has to be aware of this problem with the $k$nn approach. This is especially true in case of high-dimensional data sets.

The implementation in function `impKNNa` (package **robCompositions**) matches the description of the method in this section.

Example This function is applied on a governmental expenditure data set from OECD, which is available in the package **robCompositions** or—originally—from https://data.oecd.org/ (OECD 2015). The general government sector consists of central, state and local governments, and the social security funds controlled by these units. The data are based on the *system of national accounts*, a set of internationally agreed concepts, definitions, classifications and rules for national accounting. The *classification of functions of government* (COFOG) is used as classification system. The COFOG expenditures are divided into the following ten categories: general public services; defense; public order and safety; economic affairs; environmental protection; housing and community amenities; health; recreation, culture and religion; education; and social protection. The central government spending by category is measured as a percentage of total expenditures.

The data are first loaded and restructured. The resulting data structure is indicated by printing the first three lines of the original and restructured data.

```
data("govexp")
# first three observations (long/tidy data format)
head(govexp, 3)

##   country category year value
## 1     AUS      DEF 2007  5.77
## 2     AUS      DEF 2008  5.49
## 3     AUS      DEF 2009  5.56

library("dplyr"); library("reshape2")
# from long format to wide format and year 2014
gov14 <- govexp %>%
  filter(year == 2014) %>%
  select(-year)  %>%
  reshape2::dcast(country ~ category, mean)
head(gov14, 3)

##   country  DEF ECOAFF  EDU ENVPROT GRALPUBSER HEALTH HOUCOMM
## 1     AUT 1.62  15.07 9.43    0.70       33.61   3.52    0.11
## 2     BEL 2.89   7.62 4.86    0.54       67.59   2.80      NA
## 3     CHE 7.37  20.76 9.90    1.67       25.53   0.45    0.04
##   PUBORD RECULTREL SOCPROT
## 1   3.27      0.85   31.83
## 2   3.82      0.25    9.63
## 3   1.57      0.79   31.93
```

The data set includes two missing values, which can be seen below in detail.

```
library("VIM")
a <- aggr(gov14, plot = FALSE)
a

##
##  Missings in variables:
##  Variable Count
##   ENVPROT     1
##   HOUCOMM     1

# one (of the two) missing values is for USA in category
# environmental protection
w <- is.na(gov14$ENVPROT)
gov14[w, ]

##    country   DEF ECOAFF  EDU ENVPROT GRALPUBSER HEALTH
## 32     USA 15.26   5.45 2.67      NA      13.64   27.4
##    HOUCOMM PUBORD RECULTREL SOCPROT
## 32     1.7    1.4      0.13   32.34
```

The two missing values are imputed in the following using k-nearest neighbor imputation. Note that the first variable contains information on the country of origin and should be omitted before imputation (= it is not a part of the composition).

```
gov14imp <- impKNNa(gov14[, 2:ncol(gov14)])$xImp
# the imputed value for USA on environmental protection
gov14imp[w, ]

##        DEF     ECOAFF        EDU    ENVPROT GRALPUBSER
##  15.260000   5.450000   2.670000   0.902492  13.640000
##     HEALTH    HOUCOMM    PUBORD  RECULTREL    SOCPROT
##  27.400000   1.700000   1.400000   0.130000  32.340000
```

It can be seen that the algorithm preserves also the ratios and values for all non-missing cells.

13.2.2 Iterative Model-Based Imputation

An alternative to distance-based $k$nn imputation is to use an imputation technique that would be able to capture also the multivariate structure of compositional data (Hron et al. 2010). One such option is to apply a regression-based iterative imputation procedure, where in each step of the iteration, one variable is used as a response variable and the remaining variables serve as the regressors. Thus, the multivariate information is used for imputation in the response variable. In the context of compositional data, the response variable is formed by the first pivot coordinate from Eq. (3.26), which contains all relative information concerning the lth part. A proper choice of these balances can also help to avoid a kind of error propagation effect. Namely, such a permutation of parts in Eq. (3.26) is required so that as few as possible coordinates are affected by the missing values. Because

for initializing the algorithm, a full data matrix is needed, the missing values are imputed first with $k$nn imputation, as described above, with the hope that the imputed values are further enhanced with the regression-based procedure. The outline of the algorithm is below:

Step 1: Initialize the missing values using the $k$nn algorithm based on Aitchison distances, as described in the previous section.

Step 2: Sort the parts according to the amount of missing values. In order to simplify the notation, it is assumed that the corresponding columns of the compositional data matrix, $\mathbf{x}_{\cdot 1}, \ldots, \mathbf{x}_{\cdot D}$, are already sorted, i.e. $\mathcal{M}(\mathbf{x}_{\cdot 1}) \geq \mathcal{M}(\mathbf{x}_{\cdot 2}) \geq \ldots \geq \mathcal{M}(\mathbf{x}_{\cdot D})$, where $\mathcal{M}(\mathbf{x}_{\cdot j})$ denotes the number of missing cells in part x_j.

Step 3: Set $l = 1$.

Step 4: Express compositional data in pivot coordinates (3.26) so that all pairwise logratios with x_l are aggregated in the first coordinate.

Step 5: Denote $m_l \subset \{1, \ldots, n\}$ the indices of the observations that were originally missing in column $\mathbf{x}_{\cdot l}$, and $o_l = \{1, \ldots, n\} \backslash m_l$ the indices corresponding to the observed cells of $\mathbf{x}_{\cdot l}$. Furthermore, $\mathbf{z}_{\cdot 1}^{o_l}$ and $\mathbf{z}_{\cdot 1}^{m_l}$ denote the first coordinate (column) with the observed and missing parts, respectively, corresponding to the part x_l. Let $\mathbf{Z}_{-1}^{o_l}$ and $\mathbf{Z}_{-1}^{m_l}$ denote the matrices with the remaining coordinates corresponding to the observed and missing cells of $\mathbf{x}_{\cdot l}$, respectively. Additionally, the first column of $\mathbf{Z}_{-1}^{o_l}$ and $\mathbf{Z}_{-1}^{m_l}$ consists of ones, taking care of an intercept term in the linear regression model

$$\mathbf{z}_{\cdot 1}^{o_l} = \mathbf{Z}_{-1}^{o_l}\mathbf{b} + \mathbf{e} \tag{13.4}$$

with unknown regression coefficients $\mathbf{b}$ and an error term $\mathbf{e}$.

Step 6: Estimate the regression coefficients $\mathbf{b}$ in (13.4), and use the estimated regression coefficients $\widehat{\mathbf{b}}$ to replace the missing parts $\mathbf{z}_{\cdot 1}^{m_l}$ by

$$\widehat{\mathbf{z}}_{\cdot 1}^{m_l} = \mathbf{Z}_{-1}^{m_l}\widehat{\mathbf{b}}. \tag{13.5}$$

Step 7: Use the updated coordinates for expressing them back in the original space using the sample version of (3.22)—initial reordering of parts according to (3.26) is preserved. As a consequence, the values that were originally missing in the cells m_l in column $\mathbf{x}_{\cdot l}$ are updated. Note that also the non-missing cells are updated, but the ratios between them do not change.

Step 8: Carry out Steps 4–7 in turn for each $l = 2, \ldots, D$.

Step 9: Repeat Steps 3–8 until the Frobenius norm of the difference between the sample covariance matrices computed, e.g. from the pivot coordinates (3.20) from the present and the previous iteration is smaller than a certain boundary.

Although there is no proof of convergence for this procedure, according to Hron et al. (2010) it usually converges in a few iterations. The choice of pivot coordinates (Eq. (3.26)) also guarantees that already for $l = 1$, the information of the column

$\mathbf{x}_{\cdot 1}$ with the highest amount of missings is only contained on the left-hand side of Eq. (13.4), but not in the explanatory variables on the-right hand side.

In order to suppress the influence of outlying observations, and simultaneously also to protect against poorly initialized missing values, the classical least squares regression can be replaced by a robust counterpart from Sect. 10.6. Another, theoretically sound but computationally more intensive possibility would also be to use classical and robust orthogonal regression, discussed in Sect. 10.4.

This regression-based imputation procedure can be easily adapted to the case of rounded and count zeros, and even to high-dimensional compositions. Details are provided in the following two sections.

Example (contd.) The function `impCoda` can be used to impute missing values with the iterative model-based procedure described above.

```
gov14imp <- impCoda(gov14[, 2:ncol(gov14)])$xImp
# the imputed value for USA on environmental protection
gov14imp[w, ] * 100

##         DEF   ECOAFF      EDU   ENVPROT GRALPUBSER   HEALTH
## 32 15.16102 5.41465 2.652682 0.6585555   13.55153 27.22228
##     HOUCOMM   PUBORD RECULTREL  SOCPROT
## 32 1.688973 1.390919 0.1291568 32.13024

# ratios are preserved, see e.g.:
gov14[w,"DEF"] / gov14[w, "ECOAFF"]

## [1] 2.8

gov14imp[w,"DEF"] / gov14imp[w, "ECOAFF"]

## [1] 2.8
```

Note that diagnostics can be made to evaluate if the imputed values are reasonable, by using the diagnostic plots of missing and imputed values implemented in the package **VIM** (Templ et al. 2012).

13.3 Rounded and Count Zeros

From the definition of compositional data it follows that all relevant information is contained in the (log)ratios between the parts. Accordingly, it is natural that zero values are in conflict with this concept. Also purely from a numerical perspective, the logarithm of a zero value is not valid. On the other hand, zero values frequently occur in compositional data sets across all applications. In some fields, like in official statistics or in omics data, they can even form the majority of the values in the data set. The crucial question to be answered in the rest of this chapter is how to cope with them without the danger that they would completely inhibit further statistical processing of compositional data using the logratio methodology.

Both rounded and count zeros cannot be considered as *true* zeros, but rather as a result of imprecision/detection limit issues (rounded zeros) or insufficient sample size (count zeros). In other words, it is fully meaningful to impute them with a reasonable small value and continue with processing of a complete data set. In order to guarantee imputation by a small value, an upper bound should be considered. Any relevant imputation procedure should then impute with values below this upper bound, usually denoted as *detection limit* (Martín-Fernández et al. 2003). This is clearly an absolute number, though any imputed value should respect the relative nature of compositions. It turns out that the regression-based iterative procedure, introduced for the purpose of missing values imputation, is able to incorporate detection limits into a sound imputation technique.

13.3.1 Rounded Zeros

Rounded zeros represent a prominent zero type in compositional data analysis. They occur frequently in environmental and chemical data, whenever either small values of components are rounded to zeros or a measurement device has incorporated a detection limit (number) that automatically sets values below the limit to zero. As mentioned already before, sometimes the values below the detection limit are stored as negative values of the detection limits. However, the meaning remains the same: the measurement unit cannot observe a value and thus we can consider it as a zero before imputation. Rounded zeros are also known under the name *censored values* in the literature (Helsel 2012; Millard et al. 2012), and here only the case of left-censored data is discussed, referring to a lower detection limit (also upper detection limits exist). Detection limits can differ among the parts, as it is possible that some parts can be measured with higher precision than others. An additional complication might occur when the detection limit of a component differs among samples, as a result of processing in different laboratories.

The logratio methodology has its origin in the field of geochemistry, and thus much interest was devoted to rounded zeros from the very beginning up to recent developments (Aitchison 1986; Martín-Fernández et al. 2003; Palarea-Albaladejo et al. 2007; Palarea-Albaladejo and Martín-Fernández 2008; Martín-Fernández et al. 2011, 2012; Palarea-Albaladejo and Martín-Fernández 2013; Palarea-Albaladejo et al. 2014). If the proportion of zeros is not too high, say less than 10% of the values in the data matrix, a non-parametric replacement strategy might be sufficient. The simplest option is to impute zeros with 65% of the detection limit (DL) as this minimizes the distortion of the covariance structure (Martín-Fernández et al. 2003, 2011). Obviously, for a higher proportion of replaced values this approach leads to an underestimation of the compositional variability. Moreover, the concrete value to be imputed across the data set (compositional part) is highly influential due to the relative scale of compositional data. The variability issue (in a univariate sense) can be overcome by replacing with values that are sampled from a uniform distribution in $(0, DL)$, or by using the assumption of lognormal distribution, truncated by

the threshold (Palarea-Albaladejo and Martín-Fernández 2013; Palarea-Albaladejo et al. 2014). Although the latter alternatives to the simplest replacement strategy can cope with distortion of variability, the univariate character of these approaches ignores the multivariate complexity of compositional data. They can be considered as a "quick and dirty" solution, but recent simulations (Martín-Fernández et al. 2012; Palarea-Albaladejo et al. 2014) show that multivariate imputation methods clearly outperform them.

Similar as for the imputation of missing values, a model-based algorithm can be developed also in case of rounded zeros that uses censored regression in order to guarantee that the imputed values do not exceed the threshold. By following the previous developments using the modified EM algorithm in alr coordinates (Palarea-Albaladejo et al. 2007; Palarea-Albaladejo and Martín-Fernández 2008), a recent approach with ilr coordinates was introduced in Martín-Fernández et al. (2012). Basically, this is an adapted algorithm from Sect. 13.2.2, just that the usual multiple regression is replaced by a truncated counterpart. Because all computations are performed in ilr coordinates, also the detection limit(s) need to be expressed there. The detection limit can differ among the compositional parts (different precision for measuring different parts), and even among the observations (effect of different laboratories), thus the respective limit values d_{il}, $i = 1, \ldots, n$, $l = 1, \ldots, D$, are in general different. Let $d_{i1}^{(l)} \equiv d_{il}$ be the thresholds in the lth compositional part of the original data set $\mathbf{X}$, $l = 1, \ldots, D$, reordered according to pivot coordinates (3.26). Then, the pivot coordinates $\mathbf{Z}^{(l)}$ of the rounded zeros, when $x_{i1}^{(l)} < d_{i1}^{(l)}$ occurs, result in unknown values $z_{i1}^{(l)}$ with the property $z_{i1}^{(l)} < \psi_{i1}^{(l)}$, where

$$\psi_{i1}^{(l)} = \sqrt{\frac{D-1}{D}} \ln \frac{d_{i1}^{(l)}}{\sqrt[D-1]{\prod_{j=2}^{D} x_{ij}^{(l)}}}. \tag{13.6}$$

For the sake of completeness, a detailed description of the algorithm (Martín-Fernández et al. 2012) follows:

Step 1: Initialize the rounded zeros using 65% of the detection limit, or by any alternative univariate method, e.g., from those listed above or in Palarea-Albaladejo et al. (2014).

Step 2: Sort the parts according to the amount of zero values. In order to simplify the notation, it is assumed that the corresponding columns of the compositional data matrix, $\mathbf{x}_{\cdot 1}, \ldots, \mathbf{x}_{\cdot D}$, are already sorted, i.e. $\mathcal{M}(\mathbf{x}_{\cdot 1}) \geq \mathcal{M}(\mathbf{x}_{\cdot 2}) \geq \ldots \geq \mathcal{M}(\mathbf{x}_{\cdot D})$, where $\mathcal{M}(\mathbf{x}_{\cdot j})$ denotes the number of zero cells in part x_j.

Step 3: Set $l = 1$.

Step 4: Express the compositional data in pivot coordinates (3.26) so that all pairwise logratios with x_l are aggregated in the first coordinate. Particularly, also the detection limit of the lth compositional part needs to be expressed in coordinates using (13.6).

Step 5: Denote $m_l \subset \{1, \ldots, n\}$ the indices of the observations that were originally zero in column $\mathbf{x}_{\cdot l}$, and $o_l = \{1, \ldots, n\} \backslash m_l$ the indices corresponding to the non-zero cells of $\mathbf{x}_{\cdot l}$. Furthermore, $\mathbf{z}_{\cdot 1}^{o_l}$ and $\mathbf{z}_{\cdot 1}^{m_l}$ denote the first coordinate (column) with the non-zero and zero parts, respectively, corresponding to the part x_l. Let $\mathbf{Z}_{-1}^{o_l}$ and $\mathbf{Z}_{-1}^{m_l}$ denote the matrices with the remaining coordinates corresponding to the observed and zero cells of $\mathbf{x}_{\cdot l}$, respectively. Additionally, the first column of $\mathbf{Z}_{-1}^{o_l}$ and $\mathbf{Z}_{-1}^{m_l}$ consists of ones, taking care of an intercept term in the linear regression model

$$\mathbf{z}_{\cdot 1}^{o_l} = \mathbf{Z}_{-1}^{o_l} \mathbf{b} + \mathbf{e} \tag{13.7}$$

with unknown regression coefficients $\mathbf{b}$ and an error term $\mathbf{e}$.

Step 6: Denote the ith row of $\mathbf{Z}_{-1}^{m_l}$ as $\mathbf{z}_{i,-1}^{m_l}$, being a column vector. Estimate the regression coefficients $\mathbf{b}$ in (13.7), and use the estimated regression coefficients $\widehat{\mathbf{b}}$ to replace the (originally) zero parts $\mathbf{z}_{\cdot 1}^{m_l}$ by the conditional expected values

$$\widehat{z}_{i1}^{m_l} = (\mathbf{z}_{i,-1}^{m_l})' \widehat{\mathbf{b}} - \widehat{\sigma} \frac{\phi\left(\frac{\psi_{i1}^{(l)} - (\mathbf{z}_{i,-1}^{m_l})' \cdot \widehat{\mathbf{b}}}{\widehat{\sigma}}\right)}{\Phi\left(\frac{\psi_{i1}^{(l)} - (\mathbf{z}_{i,-1}^{m_l})' \cdot \widehat{\mathbf{b}}}{\widehat{\sigma}}\right)}, \tag{13.8}$$

for all $i \in m_l$. Here, ϕ and Φ stand for the density and distribution function of the standard normal distribution, respectively; $\hat{\sigma}$ is the estimated conditional standard deviation of the variable $\mathbf{z}_{\cdot 1}^{o_l}$.

Step 7: Use the updated coordinates for expressing them back in the original space using the sample version of (3.22)—the initial reordering of the parts is preserved. As a consequence, the values that were originally zeros in the cells m_l in column $\mathbf{x}_{\cdot l}$ are updated. Note that also the non-zero cells are updated, but the ratios between them do not change.

Step 8: Carry out Steps 4–7 in turn for each $l = 2, \ldots, D$.

Step 9: Repeat Steps 3–8 until the Frobenius norm of the difference between the sample covariance matrices computed, e.g. from the pivot coordinates (3.20) from the present and the previous iteration is smaller than a certain boundary.

The concrete implementation of the algorithm in packages of the statistical software R (Palarea-Albaladejo and Martín-Fernández 2015; Templ et al. 2011) might slightly depart from the above description, particularly in terms of the adjustment of the imputed zeros with respect to the actual scale of the compositions.

Similar as for the imputation of missing values, also the final output of the above algorithm may depend on the initialization in Step 1. Therefore, it is recommended to use a method which is more sophisticated than just taking 65% of the detection limit. The estimation of the regression coefficients $\mathbf{b}$ in (13.7) can be done either by taking the classical least squares estimation or using an appropriate robust counterpart. The effect can be quite strong in case

of rounded zero imputation, because small absolute values (relative to the other values in the composition) tend to produce aberrant logratios, and consequently also outlying observations. Suppressing their influence can thus considerably improve the stability of the rounded zero imputation. Finally, note that the assumption of normal distribution of the first coordinate in (13.8), used to guarantee an estimation of zeros below the detection limit, is usually not restrictive in practice. Even for moderate departures from normality, the algorithm still yields reasonable values.

Example The data set `chorizonDL` was already used at the beginning of this chapter.

```
data("chorizonDL", package = "robCompositions")
dl <- attributes(chorizonDL)$DL
exclude <- c("*ID", "XCOO", "YCOO", "*COUN", "*ASP", "TOPC", "LITO")
w <- colnames(chorizonDL) %in% exclude
ch <- chorizonDL[, !w]
# amount of zeros in each column
vars0 <- apply(ch, 2, function(x) any(x == 0))
colSums(ch[, vars0] == 0)

##      Ag      As      Bi Co_INAA       K Nd_INAA       S
##       1      10      15       1       3       8       3
##      Sc
##       1

# index of zeros in As
w <- which(ch$As == 0)
w

##  [1]  64 164 224 231 234 293 361 372 386 388

# look at one zero in As (only few variables)
ch[w[1], 1:5]

##       Ag    Al Al_XRF As   Ba
## 64 0.011 12200      8  0 55.7
```

In the following, the rounded zeros are imputed using the robust model-based procedure described above.

```
chimp <- imputeBDLs(ch, method = "lmrob")
chimp
# look at one imputed zero in As
chimp$x[w[1], 1:5]

##       Ag    Al Al_XRF         As   Ba
## 64 0.011 12200      8 0.03333333 55.7

# note that the detection limit was
dl["As"]

##  As
## 0.1
```

13.3.2 Count Zeros

So far it has been implicitly assumed that compositional data contain values coming from a continuous distribution, or at least that the possibly underlying discrete distribution of compositions can be satisfactorily approximated by a continuous one. It is common to do so in compositional data analysis. In practice, however, the data generating process may be based on counts, and this is also referred to as *discrete (count) compositions* in the literature (van den Boogaart and Tolosana-Delgado 2013; Martín-Fernández et al. 2015). For example, although the structure of the population according to dwelling type (detached house, semi-detached house, flat, other), analyzed for different countries is purely of compositional nature, it is based on counts in each of the categories. Similarly, the structure of votes for political parties is again a composition, but the assumption of continuity of the values in the parts is clearly violated. Yet another example: Geochemical data are treated as continuous compositions, although the process how the concentrations are measured is often a discrete one, depending on the measurement technique or device.

One intuitive conclusion would be that discrete compositions can be modeled with a multinomial distribution, where the counts x_j in $\mathbf{x} = (x_1, \ldots, x_D)'$ are driven by parameters $p_1, \ldots, p_D$, $p_1 + \cdots + p_D = 1$ and N being the total number of counts (Hogg et al. 2005). In other words, it would be assumed that the *parameters* of the underlying discrete model are continuous. Although this idea was further developed in several directions (Martín-Fernández et al. 2015; Egozcue et al. 2015), it refers to a model that is not scale invariant. In order to be consistent with the previous methodology, it can be stated that vectors of counts are of compositional nature if the total number of counts is irrelevant from the perspective of subsequent statistical analysis, so that scale invariance can be taken as a relevant principle. If this would not be the case, the total would matter when characterizing the variability and uncertainty of the observations (with respect to a theoretical model). As a consequence, in view of the authors of this book, the logratio methodology can be recommended for count vectors with sufficiently high numbers of counts.

To give an example, the data sets `election` and `electionATbp` are based on counts:

```
head(election, 3)

##    CDU_CSU     SDP GRUENE    FDP DIE_LINKE other_parties
## SH  638756  513725 153137  91714     84177         146781
## HH  285927  288902 112826  42869     78296          82009
## NI 1825592 1470005 391901 185647    223935         348180
##    unemployment income
## SH          6.9   3157
## HH          7.4   3835
## NI          6.6   3229
```

Also the data set `GDPsatis`, or the data set `laborForce`, are originally measured as counts, thus the percentages given in the data sets are somehow discrete since they are calculated from counts (how many people are not satisfied, etc.). Such a discrete nature of the data is very common in practice.

```
head(GDPsatis, 3)

##   country gdp very.bad   bad moderately.bad moderately.good
## 1      SE 126    0.030 0.063          0.089           0.284
## 2      DK 125    0.029 0.055          0.081           0.378
## 3      BE 120    0.032 0.077          0.104           0.307
##    good very.good
## 1 0.388     0.146
## 2 0.329     0.128
## 3 0.388     0.092
```

Despite the above reasoning it can happen that some cells of the discrete compositions contain zeros, denoted as *count zeros* (Martín-Fernández et al. 2011). In view of the origin of the data, such zeros do not result from a pure absence of compositional parts, but rather from insufficient sample sizes in single observations. From this perspective, by considering an additional sampling, the zeros might be replaced by non-zero values. Therefore, it is meaningful to perform zero imputation using methods from the previous section. Obviously, the detection limit is at least 1, if data are reported in original counts, or $1/N_i$, if for a proportional representation of each composition $\mathbf{x}_i$ from the sample, $i = 1, \ldots, n$, also the total number of counts N_i is provided. Consequently, imputation techniques like the model-based replacement algorithm from Sect. 13.3.1 can be utilized (Martín-Fernández et al. 2015). Clearly, the imputed values are no more counts, just positive, continuous numbers below the detection limit. If the total number of counts is irrelevant, as required, this is not a limitation for the further processing using the logratio approach. Note that as an alternative also the Bayes-multiplicative treatment of zeros can be considered (Martín-Fernández et al. 2015). Although it enables to incorporate the imprecision resulting from low numbers of counts, it is based on a Dirichlet model (see Sect. 5.1) that implies a violation of the scale invariance principle. An additional feature of the Bayes-multiplicative treatment is that count zeros are replaced for each composition separately, the information from other observations in the data set is considered mostly within a Bayes prior. On the other hand, the model-based approach utilizes directly the whole sample of compositions, providing complete information on their multivariate structure.

13.4 Rounded Zeros in High-Dimensional Data

For data sets where the number of compositional parts is larger than the number of observations, the model-based algorithm for imputation from Sect. 13.3.1 cannot be used. The reason is the regression step of the algorithm (Step 6), because

neither the classical nor robust regression is designed to deal with high-dimensional covariates. A natural way out is to use partial least squares (PLS) regression from Sect. 11.2 instead. PLS regression is also used for imputation purposes in non-compositional contexts (Brás and Menezes 2006; Guyon and Pommeret 2011; Nguyen et al. 2004) and it usually outperforms $k$nn imputation in this context. Its implementation to the model-based algorithm, that also preserves the detection limit, is straightforward. For the performance of the imputation algorithm it is necessary to estimate an appropriate number of latent variables (PLS components), avoiding underfit as well as overfit. The respective procedure (Templ et al. 2016) is based on bootstrapping, and it consists of the following steps (here for imputing values in the first compositional part):

Step 1: Based on a compositional sample, coordinates with observations $(z_{i1}, z_{i2}, \ldots, z_{i,D-1})'$, for $i = 1, \ldots, n$ are computed. Without loss of generality, the pivot coordinates (3.20) are considered. R bootstrap data sets, each consisting of n samples with replacement, are taken, and split into paired data sets $(\mathbf{z}^*_{.1,r}, \mathbf{Z}^*_{-1,r})$, for $r = 1, \ldots, R$.

Step 2: PLS regression is applied to each pair, using $1, \ldots, k$ components. The PRedicted Error Sum of Squares (PRESS) criterion is computed, using a ten-fold cross-validation procedure (Filzmoser et al. 2009).

Step 3: For each number of components, the arithmetic mean of the PRESS values over all bootstrap samples is calculated. The minimum of these arithmetic means is chosen, and the standard error of the PRESS values is calculated for that number of components determining this minimum. A threshold is fixed given by this minimum plus one standard error.

Step 4: The final PLS model is determined with the smallest number of components, for which the mean PRESS value is still below the threshold. This ensures the selection of a parsimonious model that is not significantly worse than the possibly larger model with the smallest cross-validation prediction error.

The estimation of the optimal number of PLS components is done just once, for pivot coordinates where the response coordinate contains the part with the highest number of zeros to be imputed (initialized with a proper univariate method). As a result, a PLS model is obtained that is used further in the iterative regression scheme. Due to the repeated use of PLS regression, the algorithm itself can become quite time consuming for a high number of parts. Therefore, an alternative that saves some computational effort is discussed in Templ et al. (2016). The proposed algorithm makes use of the variation matrix (Sect. 4.1) for selecting variables to reduce the dimension of the data. A slightly modified algorithm is then used to replace rounded zeros. It is known that low values in the variation matrix indicate strong association between the parts in terms of their proportionality. When replacing rounded zeros in a particular compositional part, an optimal prediction model with a subcomposition of the remaining variables is identified, using a ranking from the variation matrix

elements. The number of predictor variables in the model is kept low, and thus the rounded zero imputation is based on ordinary least squares or robust (MM) regression. Another approach is to pre-select predictors based on Q-mode clustering (Sect. 6.6), see Chen et al. (2018). All three methods are available in the function `imputeBDLs`.

Example For an illustration how to use the imputation function `imputeBDLs` for high-dimensional compositional data, a data set from metabolomics is employed. The aim of the experiment was to ascertain novel biomarkers of MCAD (medium chain acyl-CoA dehydrogenase) deficiency. The data consist of 25 patients and 25 controls and the analysis was done by LC-MS; for details, see Najdekr et al. (2015). The rows represent patients and controls, and the columns refer to the chemical entities with their quantity. The columns are represented by m/z which is a chemical characterization of individual chemical components on exact mass measurements. All in all, the data set consists of 50 observations and 278 variables.

This data set does not contain rounded zeros, and thus, for demonstration reasons, an artificial detection limit is set equal to the 0.05-quantile for every 20th variable.

```
data("mcad")
# one patient group
mcad <- orig <- mcad[26:50, 2:ncol(mcad)]
dim(mcad)

## [1]  25 278

# set detection limit artificially
dl <- rep(0, ncol(mcad))
dl <- apply(mcad, 2, quantile, 0.05)
for (i in seq(1, ncol(mcad), 20)){
        mcad[mcad[,i] < dl[i], i] <- 0
}
system.time(
  replaced_lm <- imputeBDLs(mcad, dl = dl, eps = 1,
                            method = "lm", verbose = FALSE,
                            R = 50, variation = TRUE)$x)

##    user  system elapsed
##  47.239   4.920  52.252

system.time(
  replaced_plsfull <- imputeBDLs(mcad, dl = dl, eps = 1,
                            method = "pls", verbose = FALSE,
                            R = 50, variation = FALSE)$x)

##    user  system elapsed
## 162.736   8.068 171.358
```

The options `method = "lm"` and `variation = TRUE` use the ranking of the variation matrix elements, and only few of the predictor variables, depending on a cross-validated prediction error measure, are selected in order to perform least squares regression. On the other hand, for the second imputation, PLS regression is used with all available predictors. This explains why the PLS algorithm is slower.

The quality of the imputation can be evaluated in several ways. One possibility is to compute the compositional error distance, defined as a normalized Aitchison distance between two data sets, the original data set and the imputed data set. In this example, this measure can be calculated because the true values from our experiment are known.

```
ced <- function(x, y, ni){ # also available in robCompositon, see ?ced
        return(aDist(x, y) / ni)
}
ni <- sum(mcad == 0)
ced(orig, replaced_lm, ni)

## [1] 0.3136315

ced(orig, replaced_plsfull, ni)

## [1] 0.3860194
```

It can be seen that for this data set, the ordinary least squares regression approach outperforms the method based on PLS, but only in terms of this specific measure. Further error measures and comparisons can be found in Templ et al. (2016). In general, the PLS method leads to a better performance of the imputation.

13.5 Structural Zeros

Structural zeros, sometimes also called *essential* zeros, are definitely the most challenging type of zero values from those listed up to now. As it was pointed out in Aitchison and Kay (2003), "by an essential zero we mean a component which is truly zero, not something recorded as zero simply because the experimental design or the measuring instrument has not been sufficiently sensitive to detect a trace of the component." In other words, structural zeros are the result of a structural process and imputing them to obtain a full data set for further processing would not be meaningful. A common example of structural zeros are expenditures on alcohol and tobacco in teetotal households, but also many other sources of structural zeros can be considered: plant species that are not able to survive in a given soil type or climate, a political party that has no candidates in a region, or retirement pension in income budget of young employees.

From the essence of the problem, structural zeros cannot be analyzed directly within the logratio methodology, the essential zero structure always needs to be considered as a latent or external information. This is a principal difference to other approaches to compositional data processing that frequently define themselves just against this apparent lack of the logratio approach. This is the case, e.g., of the square root and the hyperspherical transformations (Butler and Gladsbey 2008; Scealy and Welsh 2011; Stewart and Field 2011; Wang et al. 2007), resulting from considering a fixed constant sum constraint 1 of the compositional parts instead of scale invariance as it is the case in the logratio methodology. Although these

transformations represent the concepts of dealing with compositional data that allow for zero parts, they fail (from the perspective of the logratio approach) in other important features like incorporating the relative scale of the compositions, or their subcompositional coherence (Egozcue 2009).

Several possibilities have been proposed to handle structural zeros in data. Probably the simplest one is *amalgamation* that aims to aggregate (add) the values of parts containing predominantly zero values to such part(s) that are thematically related and free of this effect. Similarly, it is possible to sum up (amalgamate) specific parts with related interpretation to one part. Examples are to amalgamate the parts alcohol and tobacco to one part *alcohol&tobacco* in foodstuff expenditures data, or merge single plant species to one part representing a more general class.

In the following example, the *European Union Statistics on Income and Living Conditions* (EU-SILC) survey is used to give practical insights to the problem of structural zeros. This is a very popular annual panel household survey conducted in EU member states and in most other European countries, and it serves as data basis for measuring risk-of-poverty and social cohesion in Europe, see EU-SILC (2009). Here a synthetic version (Alfons et al. 2011) of the Austrian EU-SILC 2006 data set is considered. The data set with 14,827 observations from 6000 households and 28 variables (household information and various income components) is available in the R package **laeken** as data set `eusilc`.

```
library("laeken")
data("eusilc")
# every row (observation) contains at least one zero
sum(apply(eusilc[, 9:24], 1, function(x) any(x == 0, na.rm = TRUE)))

## [1] 14827

head(eusilc, 2)

##   db030 hsize db040 rb030 age  rb090 pl030 pb220a   py010n
## 1     1     3 Tyrol   101  34 female     2     AT  9756.25
## 2     1     3 Tyrol   102  39   male     1  Other 12471.60
##   py050n py090n py100n py110n py120n py130n py140n hy040n
## 1      0      0      0      0      0      0      0 4273.9
## 2      0      0      0      0      0      0      0 4273.9
##    hy050n hy070n hy080n hy090n hy110n hy130n hy145n eqSS
## 1 2428.11      0      0  33.39      0      0      0  1.8
## 2 2428.11      0      0  33.39      0      0      0  1.8
##   eqIncome    db090    rb050
## 1 16090.69 504.5696 504.5696
## 2 16090.69 504.5696 504.5696
```

Since the income components contain (far too) many zeros, the parts are amalgamated according to Table 13.1 to obtain the four compositional parts (see also Templ et al. 2017) *workinc* (work income), *capinc* (capital income), *transh* (household transfers), and *transp* (personal transfers).

From the code below it can be seen that still almost every row in the amalgamated data set contains at least one zero, thus an analysis would only be possible in subsets of the data.

Table 13.1 Amalgamation of the income components from the `eusilc` data set

workinc	= py010n *(employee cash or near cash income)*	+ py050n *(cash benefit or losses from self-employment)*	
capinc	= [hy040n *(income from rental of a property or land)*	+ hy090n]/hhsize *(interests, dividends, profit from capital investments in unincorporated business)*	
transh	= [hy050n *(family/ children related allowances)* + hy080n *(inter-household cash transfers received)*	+ hy110n *(income received by people aged under 16)* – hy130n *(inter-household cash transfers paid)*	+ hy070n + *(housing allowances)* – hy145n]/hhsize *(payments/receipts for tax adjustments)*
transp	= py090n *(unemployment benefits)* + py100n *(old-age benefits)*	+ py110n *(survivor benefits)* + py120n *(sickness benefits)*	+ py130n *(disability benefits)* + py140n *(education related allowances)*

```
attach(eusilc)
workinc <- py010n + py050n
capinc <- hy040n + hy090n
transh <- hy050n + hy070n + hy080n + hy110n - hy130n - hy145n
transp <- py090n + py100n + py110n + py120n + py130n + py140n
detach(eusilc)
silc <- data.frame("workinc" = workinc,
                   "transp" = transp,
                   "capinc" = capinc,
                   "transh" = transh)
head(silc, 2)

##    workinc transp  capinc  transh
## 1  9756.25      0 4307.29 2428.11
## 2 12471.60      0 4307.29 2428.11

# all NA's should be structural zeros
silc[is.na(silc)] <- 0
# number of observations including zeros:
sum(apply(silc, 1, function(x) any(x == 0, na.rm = TRUE)))

## [1] 14189
```

After amalgamation, only ratios between the amalgamated parts can be computed—provided they do not contain any zeros. From such an analysis, however, it is not possible to draw conclusions about the original parts before amalgamation, and this may be an essential disadvantage of this procedure. Moreover, amalgamation is a nonlinear operation with respect to the Aitchison

geometry (Egozcue and Pawlowsky-Glahn 2005). Another approach, mentioned already in Aitchison (1986) and discussed further in later studies (Aitchison and Kay 2003; Bacon-Shone 2003; Martín-Fernández et al. 2011) was to interpret structural zeros in a certain part as indicators of two different subgroups: one group containing observations with a value of zero in the given component versus the other with observations taking a positive value instead. This implicitly assumes that the observations originate from two populations, with and without zero in the specific component, with possibly different distributions of the non-zero parts. This might well reflect practical situations: if a certain political party does not have a candidate in a certain region, it might significantly affect the distribution of the votes among the other parties. Consequently, both groups of observations are analyzed separately. Nevertheless, such an approach might be successful only in cases with very simple zero structures, like just mentioned. Nevertheless, this is rarely the case in practice. Usually, the zero structure is much more complex, and it would lead to far more than just the indicated two groups. If one would split the data set into subgroups corresponding to all possible patterns of zeros in the data, an inevitable consequence would be an insufficient sample size for the purpose of the statistical processing of the single groups. Moreover, the argument of different distributions of common non-zero parts for different zero patterns cannot be assumed as a general rule (Aitchison and Kay 2003).

A natural step further would be to build up a model that is able to cope with both the zero structure and the information contained in the non-zero parts. This idea is followed in Aitchison and Kay (2003), where a two-stage model is proposed: the first stage is used to determine where the zeros occur (using a binary matrix of zero values) and in the second step it is estimated how the observations are distributed within the non-zero parts. Technically, this strategy leads to a binomial conditional logistic normal model, where the parameters of the normal distribution on the simplex (one common distribution for all zero patterns is assumed) are estimated using the maximum likelihood technique. However, due to the apparent complexity of the likelihood function this model was never brought to wider practical use.

For this reason, Templ et al. (2017) propose another strategy. They use the principal assumption that an imputation of structural zeros in data should not add new information to the overall covariance structure. Of course, it would not be meaningful to impute zero values and continue to work with complete data in the usual way. However, it is reasonable to impute the zeros as an auxiliary step (using, e.g., the model-based algorithm from Sect. 13.2.2), just to estimate the overall location and covariance, and then proceed to an analysis in the groups of the subcompositions defined by the zero patterns. Using the imputation step, the problem with insufficient sample size by analyzing subcompositional groups separately is avoided. As a next step, the zero patterns themselves are analyzed through the respective binary matrix with methods designed for this specific type of data.

The general procedure is applied in Templ et al. (2017) for the case of outlier detection. At first, the overall covariance structure of the non-zero parts is estimated using the imputation step. Subsequently, non-zero parts of observations

are tested for outlyingness using Mahalanobis distances (Sect. 5.3.2). In order to see whether the zero patterns might indicate different distributions of non-zero parts, the resulting Mahalanobis distances can be compared with those resulting from estimating the covariance structure in the single zero patterns separately. Finally, possible outliers in the binary structure of the zero values are analyzed using PCA for binary data (de Leeuw 2006).

Example (EU-SILC contd.) As a first step, the zero structure of the amalgamated EU-SILC data is analyzed. The left plot in Fig. 13.1 shows the number of zeros

```
par(mar = c(10,3,0,10))
silcNA <- silc
silcNA[silc == 0] <- NA
aggr(silcNA, numbers = TRUE, prop = FALSE)
```

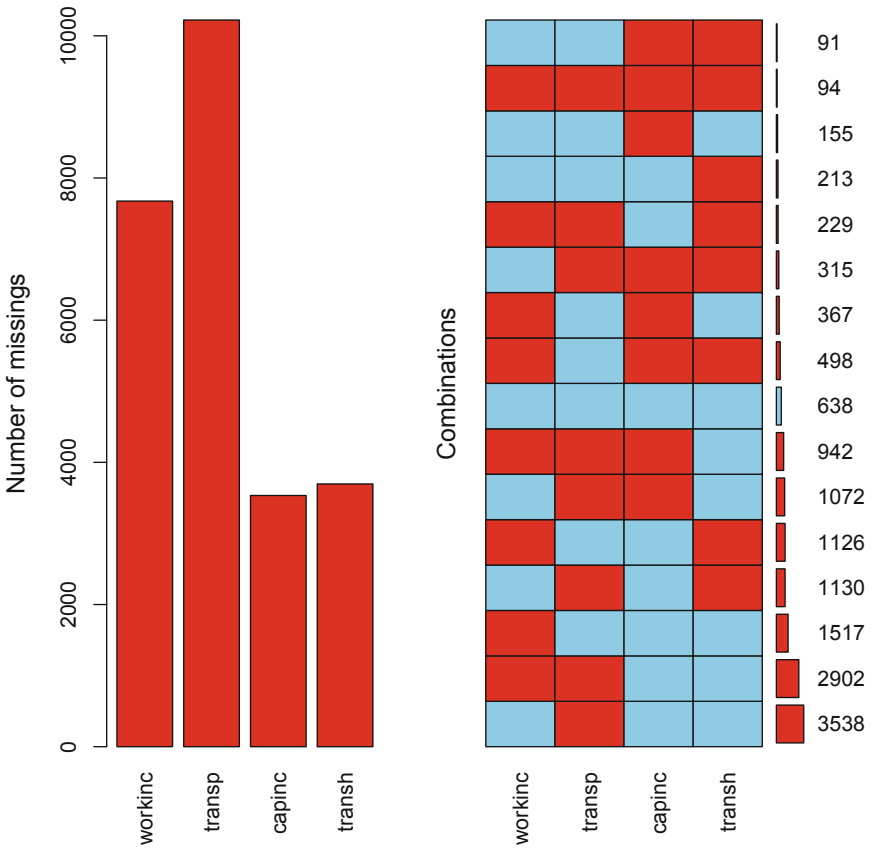


Fig. 13.1 Zero structure of the Austrian EU-SILC data. Left: the number of zeros for work income, capital income, household transfers and personal transfers. Right: combinations of zeros belonging to these four parts

in the four parts with barplots, and the right plot shows all combinations of zeros sorted according to their frequency in the data. A red rectangle indicates zeros in the corresponding parts, a blue rectangle represents non-zero data. These different combinations correspond to the previously mentioned zero patterns. Note that when using the package **VIM** one has to code the zeros as missing values. The frequencies of the different combinations are represented by a small bar plot and by numbers. These plots are adapted from the methodology described in Templ et al. (2012) to visualize missing values. For example, the bottom row in Fig. 13.1 (right) represents compositions with only zeros in the part personal transfers (`transp`), in this case the majority of the observations. Also, many compositions (observations) have zeros in both parts, `workinc` and `transp`. The least frequent combination is displayed in the top row: zeros that are present in the parts `capinc` and `transh`. These findings can be further helpful when considering results of principal component analysis of binary data.

Observations which only consist of zeros, or where only one non-zero value is present, are excluded from further analysis, as well as a few observations with negative income. This results in 10 different zero patterns which are further analyzed below.

After imputation of the zeros with *k*nn, the overall location and covariance is estimated and Mahalanobis distances can be computed. These Mahalanobis distances are compared with those resulting from each data subgroup corresponding to a zero pattern. Figure 13.2 shows the results for the ten different zero patterns, cf. text and graphics in Templ et al. (2017). The line in each graphic indicates equal Mahalanobis distance with both approaches, using the overall data information, and using only the information of the data defined by the corresponding zero pattern. It turns out that for most zero patterns, the data structure is very similar to the overall data structure. The largest deviations can be seen in the patterns "`0xx0`" (zeros in `workinc` and `transh`) and "`x0x0`" (zeros in `transp` and `transh`) but also these results reveal only slightly larger deviations for both methods. It can be concluded that methods described in detail in Templ et al. (2017) can be used for this data set, i.e. the analysis can be done not only strictly in each subcomposition (zero pattern) separately, but it is possible to use approaches that refer to an analysis of the whole data set. In case of outlier detection this means that first the structural zeros are imputed, the overall location and covariance is estimated, and the analysis can be done in the subcomposition by considering the jointly estimated location and covariance matrix (Templ et al. 2017). The gain is less uncertainty in parameter estimation due to larger sample size. This has been shown here for the estimated covariance matrix, which plays also an essential role when applying other multivariate statistical methods.

The distribution of the structural zeros may also contain valuable information. Figure 13.3 presents the results of a binary PCA (de Leeuw 2006; Tang and Tao 2006; Lee et al. 2010) applied to the different zero patterns in form of score and loading plots of the first two principal components. Since this analysis is not based on logratios, all zero patterns identified in Fig. 13.1 (right) can be used. These plots

```
silc <- silc[!silc$transh < 0, ]
v <- apply(silc, 1, function(x) sum(x == 0) %in% c(1,2))
silc <- silc[v, ]
mah <- compareMahal(silc, imp = "knn")
plot(mah)
```

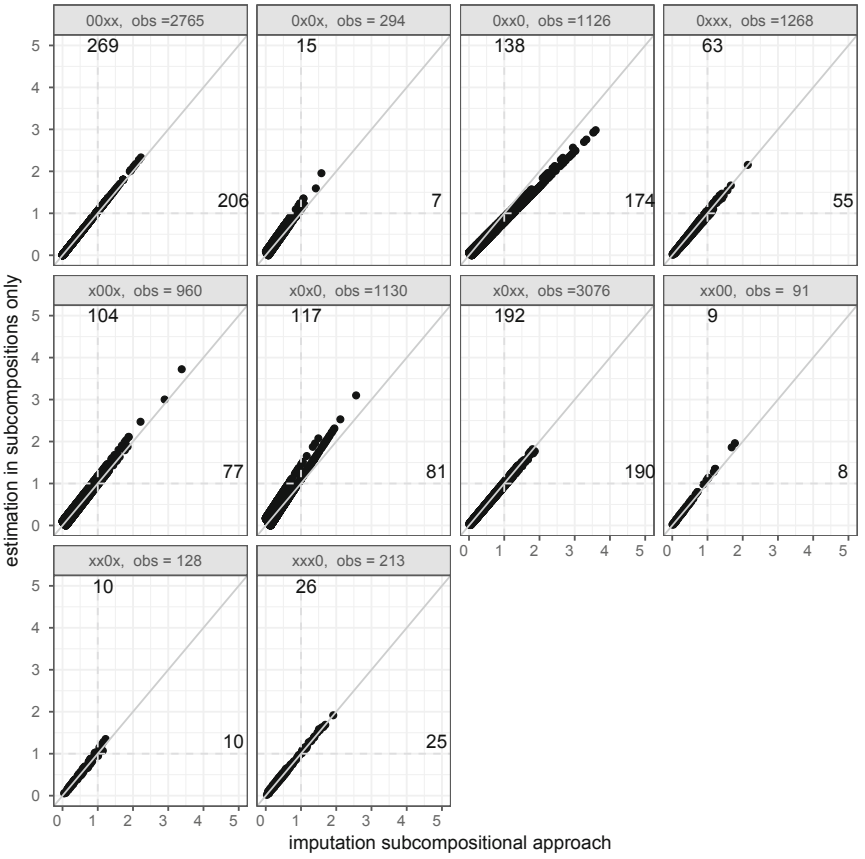


Fig. 13.2 Comparing Mahalanobis distances obtained from the imputation approach and from the estimation in subcompositions applied to the EU-SILC data, originally published in Templ et al. (2017). Published with kind permission of © Taylor & Francis, United Kingdom 2016. All Rights Reserved

can be jointly interpreted in the sense of the standard covariance biplot (Gabriel 1971). For example, the variable household transfer (`transh`) points to the upper left corner of the (loadings) plot, which means that patterns with observed values in this variable (indicated by `x`) are located in the upper left, and patterns with zeros in `transh` are in the opposite direction. According to the configuration in the loadings plot (Fig. 13.3, right), the patterns referring to `transh` and capital income (`capinc`) show similar behavior, which is also visible in Fig. 13.1. In contrast,

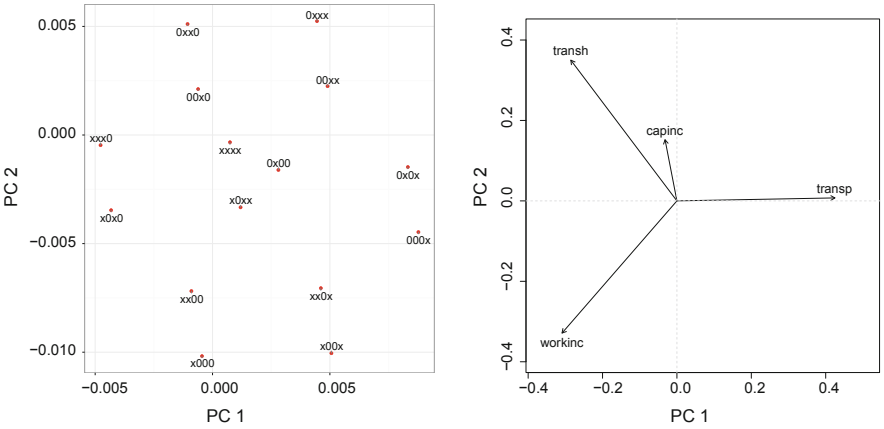


Fig. 13.3 Binary PCA plots for every zero pattern, originally published in Templ et al. (2017). Published with kind permission of © Taylor & Francis, United Kingdom 2016. All Rights Reserved

`transh` and `transp` (personal transfer) point at very different directions in the loadings plot, also the direction for capital income (`capinc`) is very different, and thus the occurrence of zeros in these variables is rather independent from each other (Templ et al. 2017).

There is no clear outlier visible in the scores plot (Fig. 13.3, left), i.e., none of the zero patterns shows completely different behavior. There are just some atypical patterns that tend to be located further from the origin. For example, the pattern "`x00x`"—the pattern expressed by observed positive values in the first and fourth variable, zeros in the second and third variable—is in the bottom right of the plot, further away from the origin, and this is the pattern which occurs only 91 times, see Fig. 13.1 and Templ et al. (2017).

References

J. Aitchison, *The Statistical Analysis of Compositional Data* (Chapman & Hall, London, 1986); Reprinted in 2003 with additional material by The Blackburn Press

J. Aitchison, J. Kay, Possible solution of some essential zero problems in compositional data, in *Proceedings of CoDaWork'03, The 1st Compositional Data Analysis Workshop*, ed. by S. Thió-Henestrosa, J.A. Martín-Fernández (University of Girona, Girona, 2003). CD-ROM

A. Alfons, S. Kraft, M. Templ, P. Filzmoser, Simulation of close-to-reality population data for household surveys with application to EU-SILC. Stat. Methods Appl. **20**(3), 383–407 (2011)

J. Bacon-Shone, Modelling structural zeros in compositional data, in *Proceedings of CoDaWork'03, The 1st Compositional Data Analysis Workshop*, ed. by S. Thió-Henestrosa, J.A. Martín-Fernández (University of Girona, Girona, 2003). CD-ROM

L.P. Brás, J.C. Menezes, Dealing with gene expression missing data. Syst. Biol. **153**(3), 105–119 (2006)

A. Butler, C. Gladsbey, A latent Gaussian model for compositional data with zeros. J. R. Stat. Soc.: Ser. C: Appl. Stat. **57**(5), 505–520 (2008)

J. Chen, X. Zhang, K. Hron, M. Templ, S. Li, Regression imputation with Q-mode clustering for rounded zero replacement in high-dimensional compositional data. J. Appl. Stat. **45**(11), 2067–2080 (2018)

J. de Leeuw, Principal component analysis of binary data by iterated singular value decomposition. Comput. Stat. Data Anal. **50**(1), 21–39 (2006)

A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood for incomplete data via the EM algorithm (with discussions). J. R. Stat. Soc. **39**, 1–38 (1977)

J.J. Egozcue, Reply to "On the Harker variation diagrams; . . . " by J.A. Cortés. Math. Geosci. **41**(7), 829–834 (2009)

J.J. Egozcue, V. Pawlowsky-Glahn, Groups of parts and their balances in compositional data analysis. Math. Geol. **37**(7), 795–828 (2005)

J.J. Egozcue, V. Pawlowsky-Glahn, M. Templ, K. Hron, Independence in contingency tables using simplicial geometry. Commun. Stat. Theory Methods **44**(18), 3978–3996 (2015)

EU-SILC, Algorithms to compute social inclusion indicators based on EU-SILC and adopted under the Open Method of Coordination (OMC). EU-SILC LC-ILC/39/09/EN-rev.1, Directorate F: Social and information society statistics Unit F-3: living conditions and social protection (European Commission, Eurostat, Luxembourg, 2009)

P. Filzmoser, B. Liebmann, K. Varmuza, Repeated double cross validation. J. Chemom. **230**(4), 160–171 (2009)

K.R. Gabriel, The biplot graphic display of matrices with application to principal component analysis. Biometrika **58**(3), 453–467 (1971)

E. Guyon, D. Pommeret, Imputation by PLS regression for linear mixed models. J. Soc. Fr. Stat. **152**(4), 30–46 (2011)

D.R. Helsel, *Statistics for Censored Environmental Data using Minitab and R*, 2nd edn. (Wiley, Hoboken, 2012)

R.V. Hogg, J.W. McKean, A.T. Craig, *Introduction to Mathematical Statistics*, 6th edn. (Prentice Hall, Upper Saddle River, 2005)

K. Hron, M. Templ, P. Filzmoser, Imputation of missing values for compositional data using classical and robust methods. Comput. Stat. Data Anal. **54**(12), 3095–3107 (2010)

S. Lee, J.Z. Huang, J. Hu, Sparse logistic principal components analysis for binary data. Ann. Appl. Stat. **4**(3), 1579–1601 (2010)

R.J.A. Little, D.B. Rubin, *Statistical Analysis with Missing Data*, 2nd edn. (Wiley, New York, 2002)

J.A. Martín-Fernández, C. Barceló-Vidal, V. Pawlowsky-Glahn, Dealing with zeros and missing values in compositional data sets using nonparametric imputation. Math. Geol. **35**(3), 253–278 (2003)

J.A. Martín-Fernández, J. Palarea-Albaladejo, R.A. Olea, Dealing with zeros, in *Compositional Data Analysis: Theory and Applications*, ed. by V. Pawlowsky-Glahn, A. Buccianti (Wiley, Chichester, 2011), pp. 43–58

J. Martín-Fernández, K. Hron, M. Templ, J. Palarea-Albaladejo, Model-based replacement of rounded zeros in compositional data: classical and robust approaches. Comput. Stat. Data Anal. **56**(9), 2688–2704 (2012)

J.A. Martín-Fernández, K. Hron, M. Templ, P. Filzmoser, J. Palarea-Albaladejo, Bayesian-multiplicative treatment of count zeros in compositional data sets. Stat. Model. **15**(2), 134–158 (2015)

S.P. Millard, N.K. Neerchal, P. Dixon, *Environmental Statistics with R*, 2nd edn. (CRC Press, Boca Raton, 2012)

L. Najdekr, A. Gardlo, L. Mádrová, D. Friedecký, H. Janečková, E.S. Correa, R. Goodacre, T. Adam, Oxidized phosphatidylcholines suggest oxidative stress in patients with medium-chain acyl-CoA dehydrogenase deficiency. Talanta **139**, 62–66 (2015)

D.V. Nguyen, N. Wang, R.J. Carroll, Evaluation of missing value estimation for microarray data. J. Data Sci. **2**, 347–370 (2004)

OECD, OECD.Stat, Government expenditure by function (COFOG) (2015). Accessed 28 May 2015

J. Palarea-Albaladejo, J.A. Martín-Fernández, A modified EM alr-algorithm for replacing rounded zeros in compositional data sets. Comput. Geosci. **34**(8), 902–917 (2008)

J. Palarea-Albaladejo, J.A. Martín-Fernández, Values below detection limit in compositional chemical data. Anal. Chim. Acta **764**, 32–43 (2013)

J. Palarea-Albaladejo, J.A. Martín-Fernández, zCompositions — R package for multivariate imputation of left-censored data under a compositional approach. Chemom. Intell. Lab. Syst. **143**, 85–96 (2015)

J. Palarea-Albaladejo, J.A. Martín-Fernández, J. Gómez-García, A parametric approach for dealing with compositional rounded zeros. Math. Geol. **39**(7), 625–645 (2007)

J. Palarea-Albaladejo, J.A. Martín-Fernández, R.A. Olea, A bootstrap estimation scheme for chemical compositional data with nondetects. J. Chemom. **28**(7), 585–599 (2014)

J.L. Scealy, A.H. Welsh, Regression for compositional data by using distributions defined on the hypersphere. J. R. Stat. Soc. Ser. B Stat Methodol. **73**(3), 351–375 (2011)

J.L. Schafer, *Analysis of Incomplete Multivariate Data* (Chapman & Hall, London, 1997)

C. Stewart, C. Field, Managing the essential zeros in quantitative fatty acid signature analysis. J. Agric. Biol. Environ. Stat. **16**(1), 45–69 (2011)

F. Tang, H. Tao, Binary principal component analysis, in *Proceedings of the British Machine Vision Conference*, vol. I (2006), pp. 377–386

M. Templ, K. Hron, P. Filzmoser, robCompositions: an R-package for robust statistical analysis of compositional data, in *Compositional Data Analysis: Theory and Applications*, ed. by V. Pawlowsky-Glahn, A. Buccianti (Wiley, Chichester, 2011), pp. 341–355

M. Templ, A. Alfons, P. Filzmoser, Exploring incomplete data using visualization techniques. Adv. Data Anal. Classif. **6**(1), 29–47 (2012)

M. Templ, K. Hron, P. Filzmoser, A. Gardlo, Imputation of rounded zeros for high-dimensional compositional data. Chemom. Intell. Lab. Syst. **155**, 183–190 (2016)

M. Templ, K. Hron, P. Filzmoser, Exploratory tools for outlier detection in compositional data with structural zeros. J. Appl. Stat. **44**(4), 734–752 (2017)

O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, R. Altman, Missing value estimation methods for DNA microarrays. Bioinformatics **17**(6), 520–525 (2001)

K.G. van den Boogaart, R. Tolosana-Delgado, *Analyzing Compositional Data with R* (Springer, Heidelberg, 2013)

K.G. van den Boogaart, R. Tolosana-Delgado, M. Templ, Regression with compositional response having unobserved components or below detection limit values. Stat. Model. **15**(2), 191–213 (2015)

H. Wang, Q. Liu, H.M.K. Mok, L. Fu, W. Man Tse, A hyperspherical transformation forecasting model for compositional data. Eur. J. Oper. Res. **179**(2), 459–468 (2007)

Software Versions Used in the Book

All computations in this book were performed using the following R session:

- R version 3.5.0 (2018-04-23), `x86_64-apple-darwin13.4.0`
- Base packages: base, datasets, graphics, grDevices, grid, methods, stats, tools, utils
- Other packages: bayesm 3.0-2, boot 1.3-18, broom 0.4.2, car 2.1-4, cluster 2.0.5, colorspace 1.3-2, compositions 1.40-1, data.table 1.10.4, devtools 1.12.0, dplyr 0.5.0, e1071 1.6-8, ellipse 0.3-8, energy 1.7-0, geoR 1.7-5.2, ggmap 2.7, ggplot2 2.2.1, gplots 3.0.1, gridExtra 2.2.1, gtools 3.5.0, HiDimDA 0.2-4, knitr 1.16, laeken 0.4.6, MASS 7.3-45, mclust 5.2.3, MetabolAnalyze 1.3, mvoutlier 2.0.8, mvtnorm 1.0-5, oreg 0.1-5, pls 2.6-0, reshape2 1.4.2, robCompositions 2.0.8, robustbase 0.92-7, rrcov 1.4-3, scales 0.4.1, scatterplot3d 0.3-38, sgeostat 1.0-27, sprm 1.2.2, StatDA 1.6.9, stringr 1.2.0, tensorA 0.36, VIM 4.7.0, xtable 1.8-2

All results are reproducible and the book is generated by using the R package **knitr** (Xie 2014a, 2013, 2014b).

P. Filzmoser et al., *Applied Compositional Data Analysis*, Springer Series in Statistics, https://doi.org/10.1007/978-3-319-96422-5

Bibliography

Y. Xie, *Dynamic Documents with R and knitr* (Chapman and Hall/CRC, Boca Raton, 2013)
Y. Xie, *knitr: A General-Purpose Package for Dynamic Report Generation in R* (2014a). http://yihui.name/knitr/. R package version 1.6
Y. Xie, knitr: A comprehensive tool for reproducible research in R, in *Implementing Reproducible Computational Research*, ed. by V. Stodden, F. Leisch, R.D. Peng (Chapman & Hall/CRC, Boca Raton, 2014b), pp. 3–32

P. Filzmoser et al., *Applied Compositional Data Analysis*, Springer Series in Statistics, https://doi.org/10.1007/978-3-319-96422-5

Index

P. Filzmoser et al., *Applied Compositional Data Analysis*, Springer Series in Statistics, https://doi.org/10.1007/978-3-319-96422-5