25

robCompositions: an R-package for robust statistical analysis of compositional data

Matthias Templ[1,2], Karel Hron[3] and Peter Filzmoser[1]
[1]*Department of Statistics and Probability Theory, Vienna University of Technology, Austria*
[2]*Methods Unit, Statistics Austria, Vienna, Austria*
[3]*Department of Mathematical Analysis and Applications of Mathematics, Palacký University, Czech Republic*

25.1 General information on the R-package `robCompositions`

The programming language and software environment `R` (R Development Core Team 2010) is nowadays one of the most widely used and most popular software tools for statistics and data analysis. It is free and open-source (GPL 2) and it can be downloaded for all computer platforms from the comprehensive R archive network (`http://cran.r-project.org`). It is enhanceable via *packages* which consist of code and structured standard documentation explaining the input and output arguments (and more) of each function including code application examples.

Two contributed packages for compositional data analysis are currently available with R version 2.10.1.: the package `compositions` (van den Boogaart *et al*. 2008), and the package `robCompositions` (Templ *et al*. 2010). While `compositions` is devoted in particular

Compositional Data Analysis: Theory and Applications, First Edition. Edited by Vera Pawlowsky-Glahn and Antonella Buccianti.
 Published 2011 by John Wiley & Sons, Ltd.

to classical statistical procedures, `robCompositions` provides tools for a robust statistical analysis of compositional data together with corresponding graphical tools. A comprehensive overview is available using the command in R shown in Listing 25.1.

```
R> help(package ='robCompositions')
```

Listing 25.1 Information on the functions and data included in the R-package robCompositions.

The prefix `R>` in the code listings means that a command is applied in R. Text after the symbol # denotes comments.

25.1.1 Data sets included in the package

To explicitly display the data sets available one can use the commands shown in Listing 25.2 (the package has to be loaded first). Note that we do not intend to provide a comprehensive overview of the data sets included in the package. They are introduced only briefly in order to give the reader an idea about their meaning. Some of these data sets are used in papers on robust methods, and they can now be used to check the results presented there. All data sets from Aitchison (1986) on compositional data are available in the package `compositions`.

```
R> require(robCompostions)          # loads the package
R> data(package='robCompositions') # lists the data in the package

Data sets in package robCompositions:
arcticLake              Arctic lake sediment data
coffee                  Coffee data
expenditures            Household expenditures data
expendituresEU          Mean consumption expenditures data
haplogroups             Haplogroups data
machineOperators        Machine operators data set
phd                     PhD students in the EU
skyeLavas               Aphyric Skye lavas data
```

Listing 25.2 List of data sets in the R-package robCompositions.

Listing 25.2 outlines the names of the data sets available in the package robCompositions. By typing `help(rcdata)` in R a structured help file of the data set `rcdata` (or the named function) pops up. For example, `help(expendituresEU)` shows the help file of the `expendituresEU` data set. In the following we give a short description of the data included in the package:

- arcticLake: Sand, silt, clay compositions of 39 sediment samples at different water depths in an Arctic lake (Aitchison 1986, p. 359).
- coffee: A subset of three compositional parts of 27 commercially available coffee samples of different origins (Korhoňová *et al*. 2009).

- expenditures: The expenditure data set (Aitchison 1986, p. 395) contains household expenditures (in former Hong Kong dollars) on five commodity groups of 20 single men.
- expendituresEU: Mean consumption expenditures (in Euro) of households on 12 domestic year costs in all 27 member states of the European Union (Eurostat 2008).
- haplogroups: Distribution of European Y-chromosome DNA (Y-DNA) of 12 haplogroups by region in percentages (Eupedia 2010).
- machineOperators: Compositions of 8-h shifts of 27 machine operators, spent on four classified activities (Aitchison 1986, p. 382).
- phd: Classification of PhD students by different kinds of studies (in %) (Eurostat 2009).
- skyeLavas: AFM compositions of 23 Aphyric Skye lavas (Aitchison 1986, p. 360).

25.1.2 Design principles

Almost all functions in the package robCompositions make use of function overloading and the method dispatch of R. Each function returns an object of a certain class. Print, summary and plot methods are implemented for objects of these classes. The method dispatch of R has the advantage that the user may apply simple and standardized functions on the result objects, such as `print`, `summary` and `plot`. The corresponding method for printing and plotting is then selected automatically depending on the class of the object. This principle is shown below in practical applications.

The package robCompositions currently depends on five other packages, namely the packages `utils`, `robustbase`, `rrcov`, `car` and `MASS` from where few functions are imported. robCompositions makes use of the name space management system for R-packages, which does not allow to redefine or overwrite functions that are defined in robCompositions. This is desirable because the user can be sure that if another function (e.g. from a different package) with the same name as used in robCompositions is loaded beforehand, always the function from robCompositions will be used.

Compiled code is used for computationally intensive procedures such as calculating Aitchison distances between matrices. The compiled code has been made accessible by the R/C interface. This allows much faster computations than using interpreted R code.

25.2 Expressing compositional data in coordinates

Three different possibilities to express compositions in coordinates are implemented in the package robCompositions: the additive log-ratio transformation (alr), the centred log-ratio transformation (clr), both defined in Aitchison (1986), and the isometric log-ratio transformation (ilr) (Egozcue *et al*. 2003). Although these transformations are already implemented in the package `compositions`, our implementation differs because variable names and absolute values are preserved.

Note that in this package only one specific ilr transformation according to Hron *et al*. (2010) is available. However, this is fully sufficient for all methods described in this chapter and in related papers about robust compositional data analysis.

To show the usage of the corresponding R functions, the expenditures data set (Aitchison 1986) is loaded, and the first three observations are shown in Listing 25.3.

```
R>  data(expenditures)          # loads the data
R>  head(expenditures, 3)       # prints the first three observations
   housing foodstuff alcohol tobacco other
1      640       328     147     169   196
2     1800       484     515    2291   912
3     2085       445     725    8373  1732
```

Listing 25.3 Display of the first three observations of the expenditures data set.

The mentioned transformations can then be applied by using Listing 25.4.

```
R> alr(expenditures) # by default, the last col. is chosen as ratioing variable
R> clr(expenditures)
R> ilr(expenditures)
```

Listing 25.4 Applying log-ratio transformations with the R-package robCompositions.

Also the inverse transformations are implemented. Their use is shown in Listing 25.5 for the alr transformation.

```
R> a <- alr(expenditures, 3)       # choose 3rd column as ratioing variable
R> class(a)                        # display the class of object "a"
[1] "alr"                          # the class of object "a"
R> x <- invalr (a)                 # inverse alr for object "a"
R> expenditures[1,1]/expenditures[1,2] # ratio of the first 2 parts,
[1] 1.951220                       # first composition
R> x[1,1]/x[1,2]                   # check
[1] 1.951220
R> head(x, 3)                      # check
  housing foodstuff alcohol tobacco other
1     640       328     147     169   196
2    1800       484     515    2291   912
3    2085       445     725    8373  1732
```

Listing 25.5 Transformation and back-transformation.

Listing 25.5 shows some special and user-friendly features which are provided by the package robCompositions. The first thing to note is that the `invalr()` needs no information about the chosen ratioing variable, needed to generate the object `a`, as well as no information about the original names of the expenditures data (expenditures has 5 column names, object `a` only 4) to exactly reproduce the original data set (see Listings 25.4 and 25.5). Also the absolute values are preserved after applying the inverse transformation. Note that the functions `alr()` and `invalr()` allow to set specific parameters so that only the transformed data is returned without additional information. This is especially useful for simulations because of a reduction of computational costs.

Note that the data used can be expressed in percentages by using the function shown in Listing 25.6 that normalizes the expenditures data to a chosen sum (the default is 1).

```
R> ConstSum (expenditures, const=100)
```

Listing 25.6 Data mapped to a constant sum.

25.3 Multivariate statistical methods for compositional data containing outliers

In the following, the application of some popular (robust) methods for a statistical analysis of compositions are described. The first two methods are devoted to outlier detection and principal component analysis, where the following functions are implemented:

outCoDa: used for outlier detection;
print.outCoDa: print method for objects of class '`outCoDa`';
plot.outCoDa: plot method for objects of class '`outCoDa`';
pcaCoDa: apply (robust) principal component analysis;
print.pcaCoDa: print method for objects of class '`pcaCoDa`';
plot.pcaCoda: plot method for objects of class '`pcaCoDa`' (compositional biplot).

In addition to code explanations, the functions are illustrated by the `expenditureEU` data set (Eurostat 2008).

25.3.1 Multivariate outlier detection

Potential multivariate data outliers are identified by using robust Mahalanobis distances with the function `outCoDa()`. The Mahalanobis distance is defined for regular $(D-1)$-dimensional data $\mathbf{x}_i$, $i = 1, \ldots, n$, as

$$\mathrm{MD}(\mathbf{x}_i) = \left[(\mathbf{x}_i - T)^\top C^{-1}(\mathbf{x}_i - T)\right]^{1/2}$$

and represents a popular tool for outlier detection (Maronna *et al*. 2006; Filzmoser and Hron 2008). Here, the estimated covariance structure is used to assign a distance to each observation indicating how far the observation is from the centre of the data cloud with respect to the covariance structure. The choice of the location estimator T and the scatter estimator C is crucial. In the case of multivariate normal distribution, the (squared) Mahalanobis distances based on the classical estimators arithmetic mean and sample covariance matrix follow approximately a χ^2 distribution with $D-1$ degrees of freedom. In the presence of outliers, however, only robust estimators of T and C lead to a Mahalanobis distance being reliable for outlier detection. A popular choice for robust location and covariance estimation is the Minimum Covariance Determinant (MCD) estimator (Rousseeuw and van Driessen 1999). Usually, also in this case a χ^2 distribution with $D-1$ degrees of freedom is used as an approximate distribution, and a certain quantile, like the quantile 0.975, is used as a cut-off value for outlier identification: observations with larger (squared) robust Mahalanobis distance are considered as potential outliers.

The procedure described above follows the concept of Filzmoser and Hron (2008), and the corresponding function `outCoDa()` uses the isometric log-ratio transformed

compositions to search for outliers in real space. The function includes four arguments: the data `x`, the quantile `quantile` of the χ^2_{D-1} distribution, the `method` used (either 'standard' for classical estimation or 'robust' for MCD estimation – the latter is the default method), and `h` as the size of the subsets used for the MCD estimator (Filzmoser and Hron 2008). The latter three function arguments have sensible defaults, but they can be set by the user as well.

Outlier detection for the example data set `expendituresEU` is done with the commands shown in Listing 25.7. Since the parameters for `quantile` and `method` are the default parameters, the code in Listing 25.7 could be shortened to `outCoDa(expendituresEU)`.

```
R> data(expendituresEU)
R> outRob <- outCoDa(expendituresEU, quantile=0.975, method ="robust")
R> outRob
--------------------
[1] "8 out of 27 observations are detected as outliers."
--------------------
```

Listing 25.7 Outlier detection using robust methods.

As mentioned above, almost all functions in the package robCompositions make use of function overloading and the method dispatch of R, which now can be easily illustrated considering the comments in Listing 25.7. The function `outCoDa()` returns an object of class '`outdect`'. By typing the name of the result object (here: `outRob`) in the R console, the corresponding print method [`print.outCoDa()`] is selected automatically. Within our example, the print result simply reports that 8 out of 27 observations are detected as outliers.

Generally, the resulting Mahalanobis distances are stored in an object of class '`outCoDa`' (`mahalDist`), displayed in Listing 25.8, where a logical vector indicating outliers and non-outliers (`outlierIndex`) is shown as well.

```
R> outRob $mahalDist
 [1]  1.672683  2.374380  2.622180   1.942029  2.109558  2.195739  2.357287
 [8]  2.117132  7.134956  2.059403  10 .310637 1.896691  2.446645  2.191185
[15]  8.872333  2.506627 14 .812581  2.317635  1.623112 11 .410675 2.275384
[22] 16 .414259 2.247400  1.487761   1.798884 23.614927  9.142826
R> outRob $outlierIndex
 [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE
[13] FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE
[25] FALSE  TRUE  TRUE
R> which(outRob $outlierIndex) # index of outlying compositions
[1]  9 11 15 17 20 22 26 27
```

Listing 25.8 Accessing list elements: robust Mahalanobis distances and index of outliers.

Note that by typing `names(object)` into the R console, information about the list of objects is shown, with `object` equal to `outRob` in our example shown in Listings 25.7 and 25.8.

The resulting Mahalanobis distances (ordered according to the index of the observations), together with the corresponding cut-off value, can be graphically presented using the plot function `plot.outCoDa()`. R first searches for objects of class '`outCoDa`' if a function

called `plot.outCoDa()` exists. Therefore, the user again only needs to know the name of the generic function, `plot()`, which is simple to keep in mind. The outliers (observations 9, 11, 15, 17, 20, 22, 26 and 27) are marked with the symbol '+' by default (see Listing 25.9).

```
R> outStand <- outCoDa (expendituresEU, quantile=0.975, method="standard")
R> plot(outStand) # plots classical estimates of Mahalanobis distance
R> plot(outRob)  # plots robust estimates of Mahalanobis distance
```

Listing 25.9 Diagnostic of objects from class '`outCoDa`'

Using standard (classical) estimates for outlier detection (here stored in object `outStand`, see Listing 25.9), all the outliers are masked, and no observation is detected as outlier (see Listing 25.10). Two observations are placed very close to the cut-off value which equals the 97.5% quantile of the χ^2_d distribution ($d = 1$ in our example). Observations with (squared) Mahalanobis distance above such a cut-off value are considered as potential outliers (Rousseeuw and Van Zomeren 1990). Therefore, using standard estimates for outlier detection, which are themselves driven from outliers, may report that no artifacts are included in the data. However, observations 9, 11, 15, 17, 20, 22, 26 and 27 may highly influence further estimations based on standard estimates.

```
R> outStand
--------------------
[1] "0 out of 27 observations are detected as outlier."
--------------------
```

Listing 25.10 Outlier masking when using standard estimators.

Figure 25.1 can be easily produced by the commands shown in Listing 25.9.

25.3.2 Principal component analysis and the robust compositional biplot

The function `pcaCoDa()` computes scores and loadings in the clr space. By setting the function parameter `method` one can choose between standard principal component analysis ('standard') or its robust counterpart ('robust' – the default method). Function `plot.pcaCoDa()` then creates a (robust) compositional biplot (see Listing 25.11) according to Aitchison and Greenacre (2002) and Filzmoser *et al*. (2009a). The results obtained by standard and robust estimates are shown in Figure 25.2.

```
R> resStand <- pcaCoDa (expendituresEU, method="standard")
R> resRob <- pcaCoDa (expendituresEU)
R> plot(resStand); plot(resRob)
```

Listing 25.11 Robust principal component analysis and the robust compositional biplot.

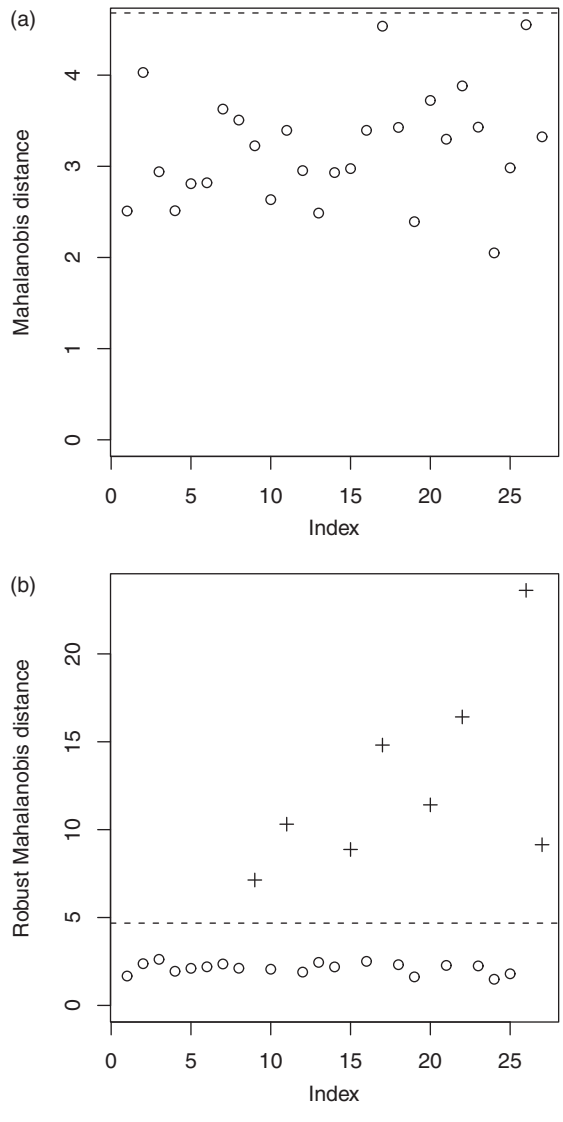

Figure 25.1 Graphical display of objects from class 'outCoDa'.

Outliers can play an important role whenever classical estimates are used. For PCA, outliers may affect the estimates of the correlation. This is no longer the case when robust PCA is applied to the transformed data [Figure 25.2(b)], because the correlations are estimated robustly. In this example, the observations (approximated by the scores) and the correlations between the variables do not differ drastically for the first two principal components. However, the classical (standard) version of the compositional biplot [Figure 25.2(a)] would indicate a different relation for example between the variables *food* and *health*.

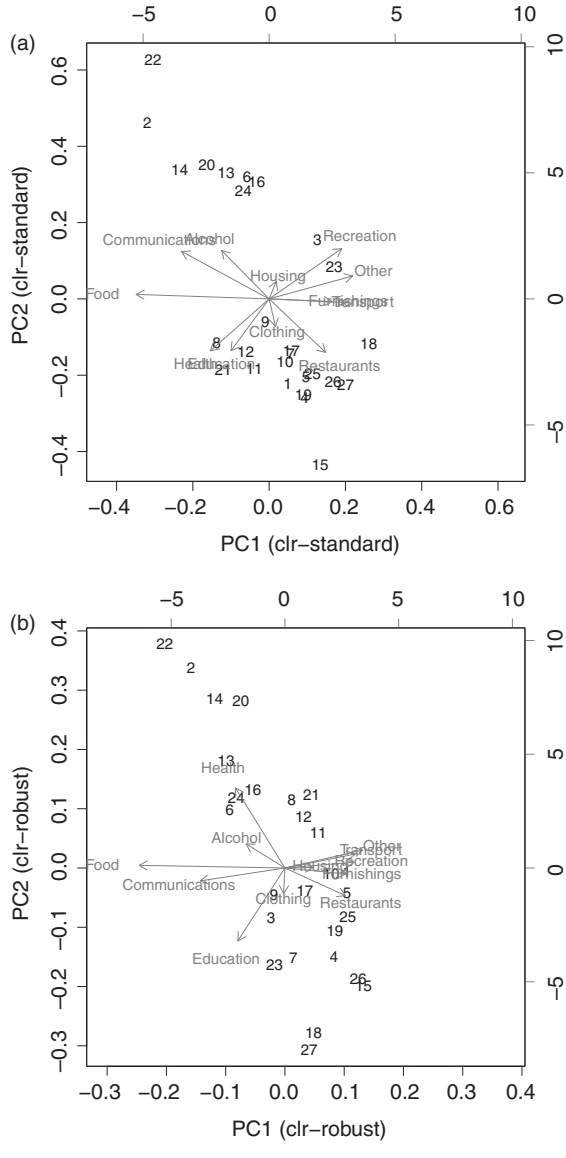

Figure 25.2 Compositional biplots using classical (a) and robust (b) estimates.

Although the robust compositional biplot [Figure 25.2(b)] (as well as the classical one) represents only rank-2 approximations of the multivariate data structure, there are also some artifacts according to outlier detection visible, e.g. positions of observations 15 (Luxembourg) and 18 (Netherlands). The top-to-bottom ordering of observations determines economic positions of the EU countries, starting with 22 (Romania) and 2 (Bulgaria). The direction of arrows shows that in poorer countries the variable *alcohol* plays an important role in the overall expenditures, contrary to higher expenditures on *clothing* and *restaurants* in countries

with higher economic position. The interpretation of the compositional biplot according to Aitchison and Greenacre (2002) enables also to conclude that there are quite strong relations (stable ratios), e.g. between the variables (expenditures on) *education*, *clothing* and *housing*, *furnishings*, respectively. In particular the latter relation reflects a similar proportion between the corresponding costs in all the EU countries.

The same idea as for principal component analysis was used for robust factor analysis for compositional data (Filzmoser *et al*. 2009b). The package robCompositions provides the function `pfa()` for standard and robust factor analysis. The utility function `factanal.fit.principal1()` which is called by `pfa()` internally is doing the actual estimation with the parameters defined via `pfa()`. The main difference to usual implementations of factor analysis is that uniquenesses are no longer of diagonal form (for details, see Filzmoser *et al*. 2009b). This kind of factor analysis is designed for centred log-ratio transformed compositional data. A robust version where the covariance matrix is estimated from the isometric log-ratio transformed compositions could be chosen [for details, see the examples in the corresponding R-help file of the package robCompositions (Templ *et al*. 2010) or in Filzmoser *et al*. (2009b)].

25.3.3 Discriminant analysis

To demonstrate robust discriminant analysis with the package robCompositions, the `coffee` data set (Korhoňová *et al*. 2009) is loaded and robust Fisher discriminant analysis (Filzmoser *et al*. 2009c) is applied (Listing 25.12). Three compositional parts are available in the data set: 1-hydroxy-2-propanone, 2-methylpyrazine and 5-methylfurfural. As a result, two natural groups are formed by Arabica coffee (16 samples) and various blends of the Arabica and Robusta coffee (11 samples). The information about the sort is saved in the fourth variable (`sort`) of this data set (Listing 25.12).

```
R> data(coffee)
R> head(coffee, 2)            # display the first 2 compositions
   Metpyr 5-Met furfu    sort
1 12.50 8.51 6.2 arabica
2  5.33 11.80 17.8 arabica
R> dres <- daFisher (coffee [,1:3], grp=coffee [,4], method
="robust", plotScore=TRUE)
```

Listing 25.12 Discriminant analysis with standard and robust methods.

In the last line of the code shown in Listing 25.12 (robust) discriminant analysis is applied. By setting the function argument `plotScore` to `TRUE`, a plot showing the discriminant scores will be generated (see Figure 25.3). Note that in the two-group case, the second direction does not include information on the group separation (Filzmoser *et al*. 2009c).

The discriminant rules rely on the assumption of normal distribution of compositions (Mateu-Figueras and Pawlowsky-Glahn 2008). Therefore, it is necessary to have a possibility of a proper multivariate normality test. In the package, a battery of Anderson–Darling tests according to Aitchison (1986) for data in orthonormal coordinates is available via the function `adtestWrapper()`. Robust multivariate methods such as robust discriminant analysis are allowing for certain deviations from normality, and only the main bulk of the data needs

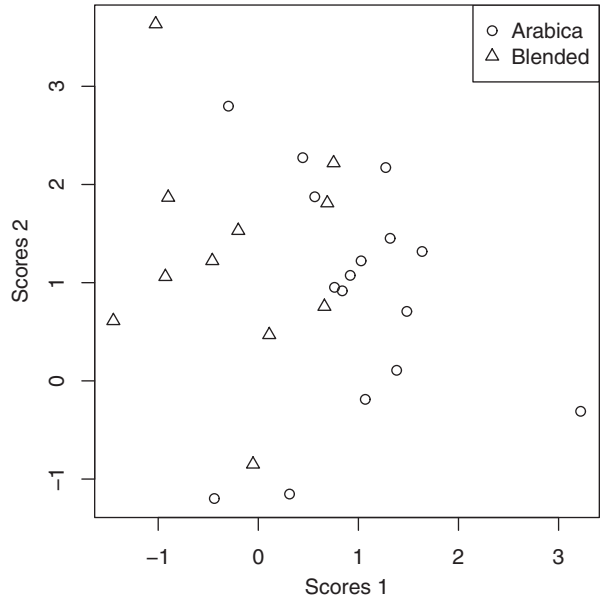

Figure 25.3 Fisher discriminant scores of the two sorts of coffee.

to follow multivariate normality. Therefore, a robust version based on the MCD estimator (Rousseeuw and van Driessen 1999) of the Anderson–Darling test for multivariate normality is implemented in the function `adtestWrapper()` [more information can be found in the manual by Templ *et al*. (2010)].

25.4 Robust imputation of missing values

Real-world compositional data sets often include missing values that need to be imputed before applying statistical methods. Since the relevant information of compositional data is contained in the ratios between the parts, special care for the imputation methodology is needed.

Two new imputation algorithms for estimating missing values in compositional data are introduced in Hron *et al*. (2010). The first proposal uses the k-nearest neighbour (k-nn) procedure based on the Aitchison distance. Hron *et al*. (2010) outlined that it is important to adjust the estimated missing values to the overall size of the compositional parts of the neighbours. As a second proposal, an iterative model-based imputation technique is introduced. This method initializes the missing values by the k-nearest neighbour procedure. The method is based on iterative and sequential regressions, hereby accounting for the whole multivariate data information. Sequentially means that in each step one variable is used as response while the other variables are used as predictors. Before a regression is applied, the data are expressed in orthonormal coordinates (ilr-transformed data) using a special basis in each step [for details, see Hron *et al*. (2010)]. The whole procedure is continued until the imputed values stabilize or a maximum number of iterations is reached.

```
R> expenditures[1,3]
147
R> expenditures[1,3] <- NA                    # set one value to NA
R> imputed1 <- impKNNa (expenditures)         # imputes the missing values using knn
R> imputed1
---------------------------------------
[1] "1 missing value was imputed"
---------------------------------------
R> imputed $xImp[1,3]
152 .1033
R> imputed2 <- impCoda (expenditures, method="ltsReg") # imputes the missing values
R> imputed2                                           # using model -based imputation
---------------------------------------
[1] "1 missing value was imputed"
[1] "2 iterations were needed"
[1] "the last change was 0"
---------------------------------------
R> imputed2 $xImp [1,3]
0.1016139
R> adjust(imputed2)$xImp [1,3]                        # preserves absolute values
150 .7718
```

Listing 25.13 *k*-nn and model-based imputation.

In Listing 25.13 one value of the expenditures data is set to missing to demonstrate the imputation methods. The original value is 147 (see Listing 25.2) whereas the values imputed with *k*-nn and model-based imputation are 152.1033 and 150.7718, respectively.

Note that the imputation method from Palarea-Albaladejo and Martín-Fernández (2008), which is based on EM-based regression using the alr transformations, is reimplemented as well in the function `alrEM()`.

Also diagnostics for imputed data can be performed where the goal is to visualize the imputed values in order to graphically evaluate the quality of the imputation. The package provides three diagnostic plot methods. In Listing 25.14 the commands to create a multiple scatterplot and a ternary diagram are shown (Aitchison 1986).

```
R> plot(imputed2, which=1)
R> plot(imputed2, which=3, seg1=FALSE)
warning:
In plot.imp(xi, which = 3, seg1 = FALSE) :
ternary diagram is only visible for 3 variables, you have 5
only the first three variables are selected for plotting
```

Listing 25.14 Diagnostic plots for imputed data.

Figure 25.4 shows the results obtained by the code in Listing 25.14. The multiple scatterplot in Figure 25.4(a) shows pairwise bivariate scatterplots of the original compositions, where it is easy to see that the imputed data point fits well to the main part of the data, where the imputed values are highlighted by the symbol +. Note that this plot can also be generated for instance for data in coordinates, taking care of the compositional nature of the data.

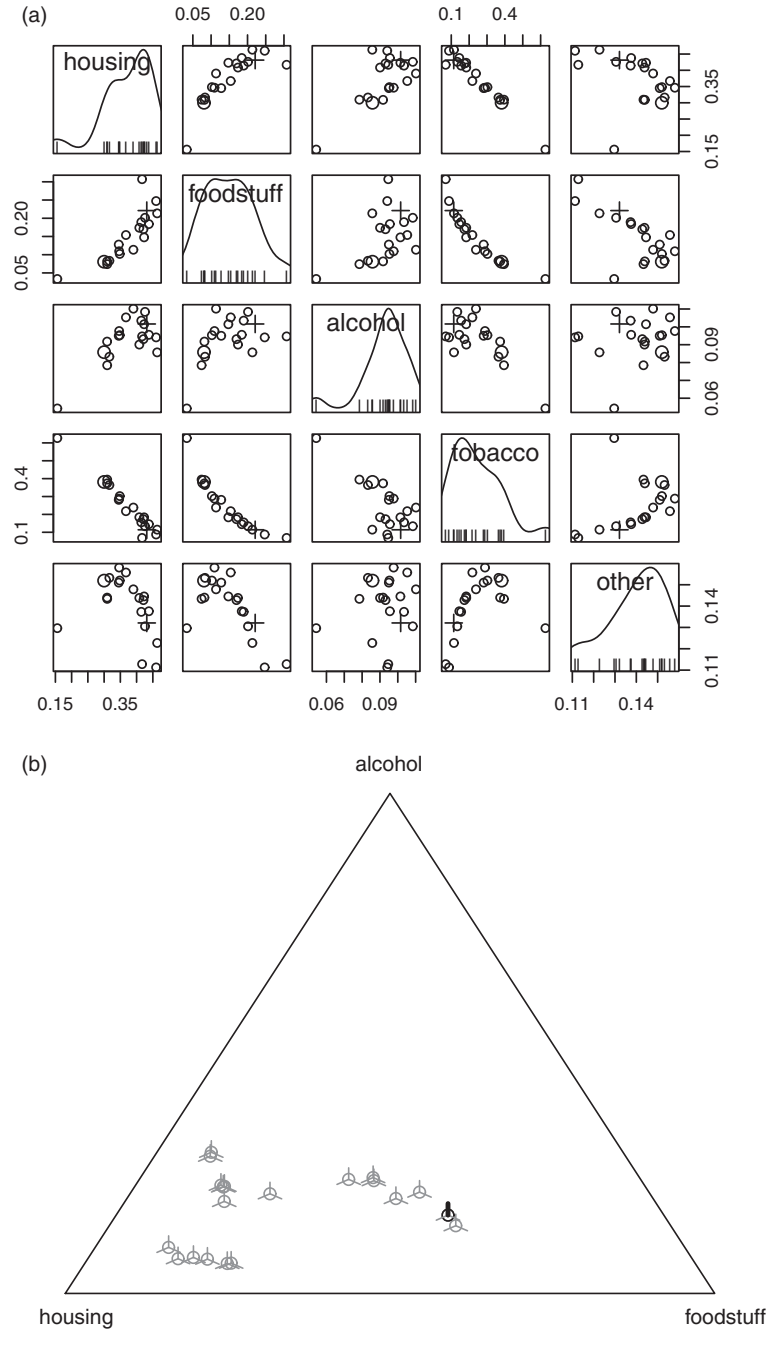

Figure 25.4 Multiple scatterplot (a) where the imputed value is marked by the symbol + and ternary diagram (b) (with thick-lined spike for the missing value) for graphical evaluation of the imputation results.

The ternary diagram in Figure 25.4(b) can be generated by `plot(xImp2, which=3, seg1=FALSE)`. Naturally, only three dimensions can be displayed in a ternary diagram (see the warning in Listing 25.14). Three-part compositions are presented by three spikes, pointing in the directions of the corresponding three parts. The spikes of the imputed values are highlighted by thick lines. This presentation allows a three-dimensional view of the data to be gained, being helpful for interpreting possible irregularities of imputed values.

A parallel coordinate plot which highlights the missing values can be generated with the function argument `which=2` in the corresponding plot method.

For all these plot methods the colour for highlighting imped and nonimputed values, their symbols and other graphical parameters can be specified.

Note that a package *vignette* for imputation is also included in the package. A package vignette is a document explaining parts or all of the functionality of a package in a more informal way than the strict format of reference help pages. It can be opened in the default pdf viewer of the operating system using Listing 25.15.

```
R> vignette ("imputation")
```

Listing 25.15 Package vignette for robust imputation of compositional data.

25.5 Summary

The design principles and the usage of the R-package robCompositions were shown using practical data examples. The examples are intended as a guidance for the user in order to successfully apply the functions included in the package robCompositions to their own data sets.

Compositional data, such as expenditures, income components, tax components, chemical concentrations, etc. virtually always contain outliers in the simplex. The robust methods included in the package fit the main part of the data minimizing the effect of data outliers and measurement errors. Most of the functions for multivariate analysis have an argument where classical (standard) methods can be compared with their robust counterparts. This allows possible differences to be seen which are due to outlying observations.

The package provides additional functionality for special tasks which were not shown in this chapter. For example, distances between compositions (or matrices) are provided by the function `aDist()`, where the function internally calls `C`-code by using the `R-C` interface. This is important whenever time complexity plays a role, i.e. when working with moderate or large data sets or when running simulations. Some other useful functions for a robust analysis of compositional data are provided, for example, for the estimation of a variation matrix using robust methods. The package will be extended with further functionality also in the future.

The package is under GPL 2. Therefore it can be used for free and the code is open source, whereas the intellectual rights on the code are preserved by this license.

References

Aitchison J 1986 *The Statistical Analysis of Compositional Data*. Monographs on Statistics and Applied Probability. Chapman and Hall Ltd (reprinted 2003 with additional material by The Blackburn Press), London (UK). 416 p.

Aitchison J and Greenacre M 2002 Biplots for compositional data. *Journal of the Royal Statistical Society, Series C (Applied Statistics)* **51**(4), 375–392.

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

Eupedia 2010 Distribution of European Y-chromosome DNA (Y-DNA) haplogroups by region in percentage. http://www.eupedia.com/europe/european_y-dna_haplogroups.shtml.

Eurostat 2008 Mean consumption expenditures (in euro) of households on 12 domestic year costs in all 27 member states of the European Union (2005). http://epp.eurostat.ec.europa.eu/statistics_explained/index.php/Household_consumption_expenditure.

Eurostat 2009. *Europa in Zahlen - Eurostat Jahrbuch 2009 - Europäische Statistiken von A bis Z*. Eurostat Pressmitteilung 135/2009.

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

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

Filzmoser P, Hron K, Reimann C and Garrett R 2009b Robust factor analysis for compositional data. *Computers and Geosciences* **35**, 1854–1861.

Filzmoser P, Hron K and Templ K 2009c Discriminant analysis for compositional data and robust estimation. Technical Report SM-2009-3, Department of Statistics and Probability Theory, Vienna University of Technology, Austria. 27 p.

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

Korhoňová M, Hron K, Klimčíková D. Müller L, Bednář P and Barták P 2009 Coffee aroma – statistical analysis of compositional data. *Talanta* **80**(82), 710–715.

Maronna R, Martin R and Yohai V 2006 *Robust Statistics: Theory and Methods*. John Wiley & Sons Ltd, New York, NY (USA). 436 p.

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

Palarea-Albaladejo J and Martín-Fernández J 2008 A modified EM alr-algorithm for replacing rounded zeros in compositional data sets. *Computers & Geosciences* **34**(8), 902–917–1861.

R Development Core Team 2010 *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna (Austria).

Rousseeuw P and van Driessen K 1999 A fast algorithm for the minimum covariance determinant estimator. *Technometrics* **41**, 212–223.

Rousseeuw P and Van Zomeren B 1990 Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association* **85**(411), 633–651.

Templ M, Hron K and Filzmoser P 2010 *robCompositions: Robust Estimation for Compositional Data*. Manual and package, version 1.4.1.

van den Boogaart G, Tolosana-Delgado R and Bren M 2008 *Compositions: Compositional Data Analysis*. R package version 1.01-1.