

Robust Imputation of Missing Values in Compositional Data Using the R-Package robCompositions

Matthias Templ^{1,2}, Peter Filzmoser¹, Karel Hron³

¹ Department of Statistics and Probability Theory, TU WIEN, Austria

² Department of Methodology, Statistics Austria

³ Department of Mathematical Analysis and Applications of Mathematics, Palacký University, Olomouc, Czech Republic

Brussels, Feb. 19, 2009



- 1 Compositional Data
- 2 Transformation
- 3 Imputation Methods
- 4 Simulation Results
- 5 R-package robCompositions
- 6 Conclusion

Compositional (Closed) Data

- **Multivariate data** that sum up to a constant (e.g. **100%**):

$$\mathbf{x} = (x_1, \dots, x_D)^t, \quad x_i > 0, \quad \sum_{i=1}^D x_i = \kappa$$

(the constant κ could be different for each observation as well)

- The set of all closed observations with positive values forms a **simplex sample space**.
- the **ratios** between the parts are of interest.

Compositional (Closed) Data

- **Multivariate data** that sum up to a constant (e.g. **100%**):

$$\mathbf{x} = (x_1, \dots, x_D)^t, \quad x_i > 0, \quad \sum_{i=1}^D x_i = \kappa$$

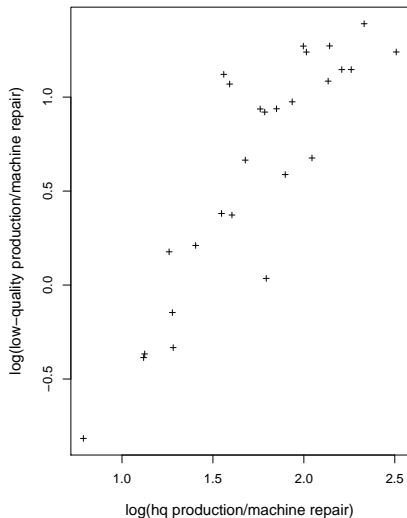
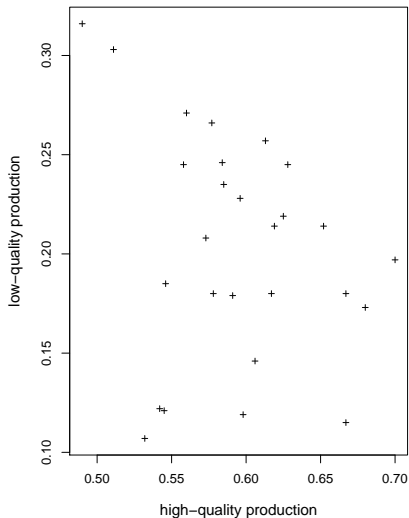
(the constant κ could be different for each observation as well)

- The set of all closed observations with positive values forms a **simplex sample space**.
- the **ratios** between the parts are of interest.

Key reference:

J. Aitchison. The Statistical Analysis of Compositional Data. Chapman and Hall, London, U.K., 1986.

Compositional Data: Example

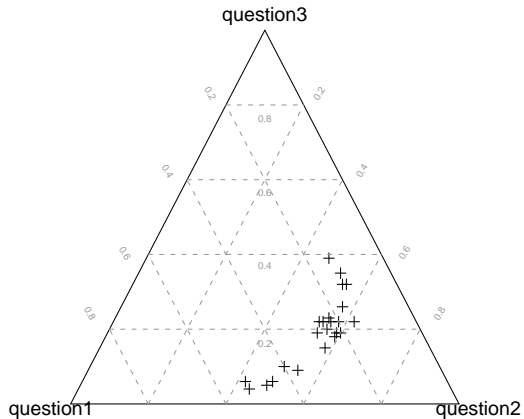


Compositional Data: Example

	qu1	qu2	qu3	sum
1	52	42	6	100%
2	52	44	4	100%
3	47	48	5	100%
⋮	⋮	⋮	⋮	⋮
22	14	47	39	100%
23	24	56	20	100%

Compositional Data: Example

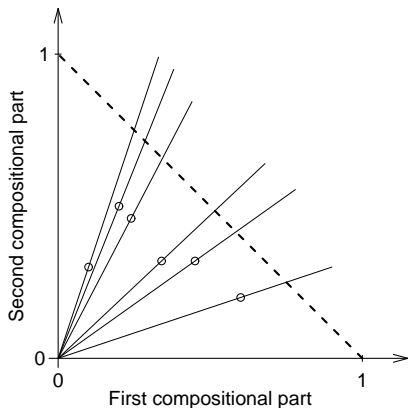
	qu1	qu2	qu3	sum
1	52	42	6	100%
2	52	44	4	100%
3	47	48	5	100%
⋮	⋮	⋮	⋮	⋮
22	14	47	39	100%
23	24	56	20	100%



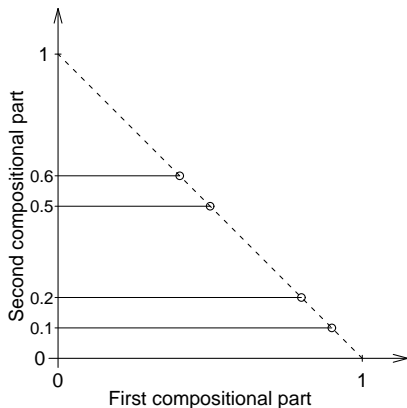
Compositional Data: Expenditures

	housing	foodstuff	alcohol	tobacco	other goods
1	640	328	147	169	196
2	1800	484	515	2291	912
3	2085	445	725	8373	1732
4	616	331	126	117	149
5	875	368	191	290	275
6	770	364	196	242	236
⋮	⋮	⋮	⋮	⋮	⋮

Compositional Data: Example



Left plot: Two-part compositional data **without** the constraint of constant sum. The points could be varied along the lines from the origin **without** changing the ratio of the compositional parts.



Right plot: The points at the boundary are more distant than the central points. The **Aitchison distance** accounts for this fact.

Aitchison Distance and the Simplex

A distance measure that is accounting for this relative scale property is the Aitchison distance (Aitchison, 1992, Aitchison et al., 2000), defined for two compositions $x = (x_1, \dots, x_D)^t$ and $y = (y_1, \dots, y_D)^t$ as

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

As an example, the boundary points in the previous Figure (right) have an Aitchison distance of **0.33**, whereas the central points have Aitchison distance **0.08**.

Replacing the Euclidean distance by the Aitchison distance is necessary because the simplex sample space has a **different** geometrical structure than the classical Euclidean space.

Logratio Transformations

Family of one-to-one transformations from the simplex to the real space (Aitchison, 1986):

- additive logratio (**alr**) transformation
- centred logratio (**clr**) transformation
- isometric logratio (**ilr**) transformation

alr Transformation

Divide all values by the j -th part:

$$\mathbf{x}^{(j)} = \left(x_1^{(j)}, \dots, x_{D-1}^{(j)} \right)^t = \left(\log \frac{x_1}{x_j}, \dots, \log \frac{x_{j-1}}{x_j}, \log \frac{x_{j+1}}{x_j}, \dots, \log \frac{x_D}{x_j} \right)^t$$

The index $j \in \{1, \dots, D\}$ refers to the “**ratioing**” variable.

clr Transformation

Divide all values by the geometric mean:

$$\mathbf{y} = (y_1, \dots, y_D)^t = \left(\log \frac{x_1}{\sqrt[D]{\prod_{i=1}^D x_i}}, \dots, \log \frac{x_D}{\sqrt[D]{\prod_{i=1}^D x_i}} \right)^t$$

Advantage: symmetric with respect to variables, easier interpretation

Disadvantage: singularity problem, because

$$\begin{aligned} \log \frac{x_1}{\sqrt[D]{\prod_{i=1}^D x_i}} + \dots + \log \frac{x_D}{\sqrt[D]{\prod_{i=1}^D x_i}} &= \\ \sum_{j=1}^D \log(x_j) - \frac{1}{D} \sum_{j=1}^D \sum_{i=1}^D \log(x_i) &= 0 \end{aligned}$$

ilr Transformation

Take an orthonormal basis $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_{D-1})$ (of dimension $D \times D - 1$) with

$$\mathbf{v}_i = \sqrt{\frac{i}{i+1}} \left(\frac{1}{i}, \dots, \frac{1}{i}, -1, 0, \dots, 0 \right) \quad \text{for } i = 1, \dots, D - 1,$$

in the hyperplane $\mathcal{H} : y_1 + \dots + y_D = 0$ in \mathbb{R}^D .

The ilr-transformed data are

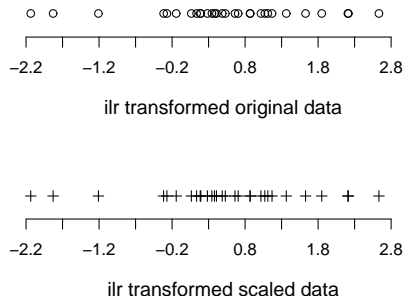
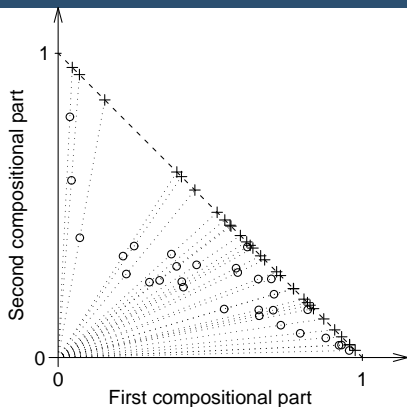
$$\mathbf{z} = (z_1, \dots, z_{D-1})^t = \mathbf{V}^t \mathbf{y}.$$

z_i are coefficients to the chosen basis.

Advantage: no singularity problem, good geometric properties

Disadvantage: z_i is not easy to interpret.

Properties of the ILR Transformation



Left plot: Two-part compositional data without the constraint of constant sum (symbols \circ), and projections on the line indicating a constant sum of 1 (symbols $+$).

Right plot: In the upper part the ilr transformed original data (with symbols \circ are shown). The lower plot shows the ilr transformed data with constant sum constraint (symbols $+$). This demonstrates that the constant sum constraint does not change the ilr transformed data.

Special Choice of ILR Variables

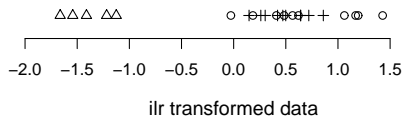
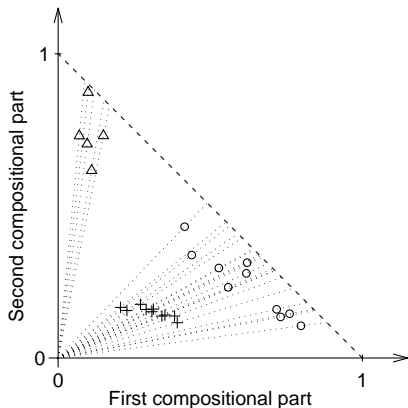
If, for example, the missing values are mainly contained in the first compositional part of the data, one can choose the ilr transformation as

$$ilr(x) = (z_1, \dots, z_{D-1})^t, \quad z_j = \sqrt{\frac{D-j}{D-j+1}} \ln \frac{\sqrt[D-j]{\prod_{l=j+1}^D x_l}}{x_j},$$

with $j = 1, \dots, D-1$.

Only this choice of the balances guarantees that missing values in x_1 does not affect z_2, \dots, z_{D-1} .

Outliers



Left plot: Two-part compositional data consisting of three groups. While the relative information of the groups with symbols \circ and $+$ is similar, the data points corresponding to the open triangles contain very different information.

Right plot: The ilr transformed data reveal that the group with open triangles are indeed different. They are potentially influencing non-robust statistical methods.

KNN Imputation

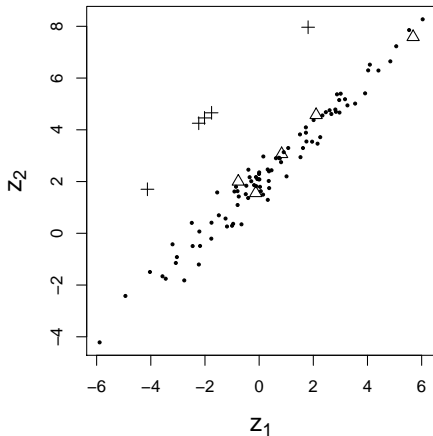
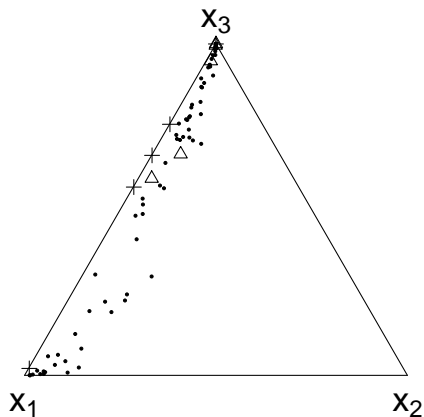
When imputing one missing value we

- use the **Aitchison distance** to find k nearest neighbors.
- **adjust** the corresponding cells according to the overall size of the parts.
- take the median of these cells to impute the missing.

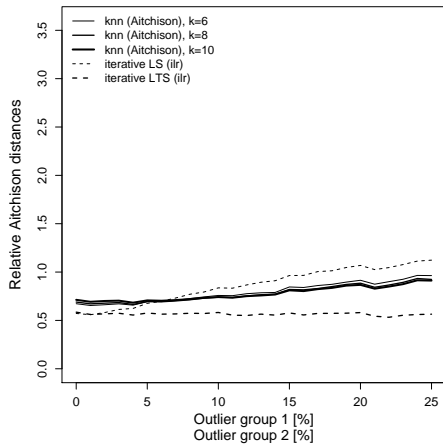
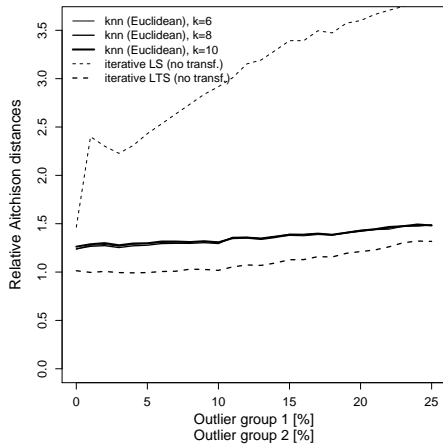
Iterative Model Based Imputation

- Start: knn solution.
- Order the data so that the first variable includes the highest amount of missing values, ...
- **Untill** convergence:
 - **For** i in $1 : D$
 - Apply a specific, well-defined ilr-transformation
 - Update former missing values in z_i by regression imputation in the ilr-space; z_i is chosen as the response variable.
 - back-transformation to the original space
 - **end** inner “loop”

Simulated Data



Results



Usage

```
http:  
//cran.r-project.org/web/packages/robCompositions/index.html  
  
> library(robCompositions)  
> help(package=robCompositions)
```

Description:

```
Package:      robCompositions  
Type:         Package  
Title:        Robust Estimation for Compositional Data.  
Version:      1.1  
Date:         2009-01-22  
Depends:      utils, e1071, robustbase, compositions, car, MASS  
Author:       Peter Filzmoser, Karel Hron, Matthias Templ  
Maintainer:   Matthias Templ <templ@tuwien.ac.at>  
Description:  This first version of the package includes methods for  
              imputation of compositional data including robust  
              methods and Anderson-Darling normality tests for  
              compositional data. The package will be enhanced with  
              other multivariate methods for compositional data in  
              near future.  
  
License:      GPL-2  
LazyLoad:     yes  
Built:        R 2.8.0; ; 2009-01-22 16:53:39; windows
```

Index:

Data

We use the randomly generated data as used in the previous Figure.

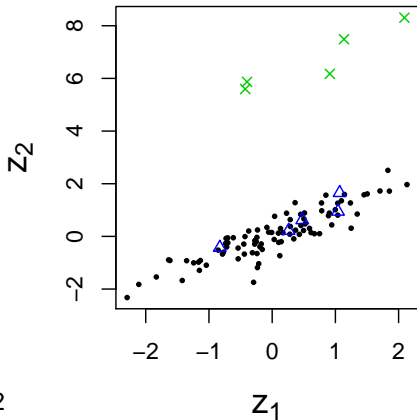
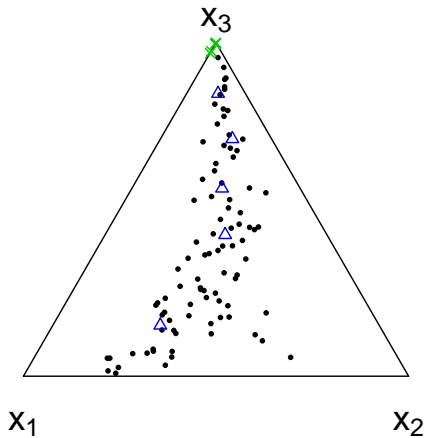
```
> head(x)
```

```
      [,1]      [,2]      [,3]  
[1,] 0.29395572 0.16181078 0.09169296  
[2,] 0.24290463 0.24092547 0.16041012  
[3,]          NA 0.05278444 0.51727452  
[4,]          NA 0.09599913 0.11838661  
[5,] 0.31172499 0.22095742 0.35843191  
[6,] 0.02038967 0.04858723 0.55728004
```

```
> dim(x)
```

```
[1] 100  3
```

Data

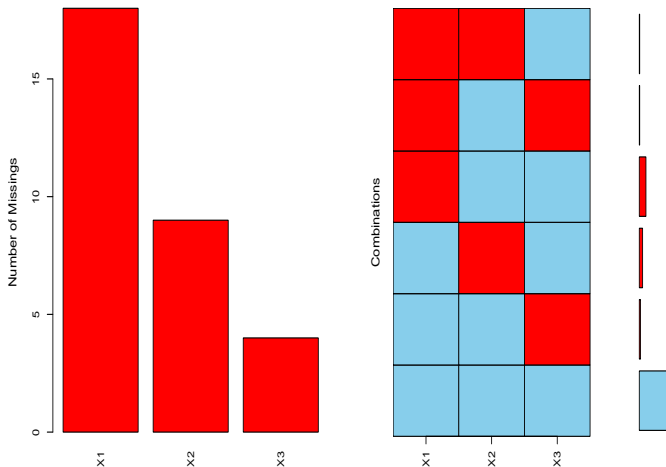


Missing Values

```
> library(VIM)
> plot(aggr(x))
```

Missing Values

```
> library(VIM)
> plot(aggr(x))
```



Imputation with robCompositions

```
> xImp <- impKNNa(x, k = 6)
> class(xImp)
```

Imputation with robCompositions

```
> xImp <- impKNNa(x, k = 6)
> class(xImp)
```

```
[1] "imp"
```

Imputation with robCompositions

```
> xImp <- impKNNa(x, k = 6)
> class(xImp)
```

```
[1] "imp"
```

```
> methods(class = "imp")
```

Imputation with robCompositions

```
> xImp <- impKNNa(x, k = 6)
> class(xImp)
```

```
[1] "imp"
```

```
> methods(class = "imp")
```

```
[1] plot.imp print.imp summary.imp
```

Imputation with robCompositions

```
> xImp <- impKNNa(x, k = 6)
> class(xImp)

[1] "imp"

> methods(class = "imp")

[1] plot.imp print.imp summary.imp

> xImp
```

Imputation with robCompositions

```
> xImp <- impKNNa(x, k = 6)
> class(xImp)
```

```
[1] "imp"
```

```
> methods(class = "imp")
```

```
[1] plot.imp print.imp summary.imp
```

```
> xImp
```

```
-----
[1] "31 missing vales were imputed"
-----
```

Imputation with robCompositions

```
> names(xImp)
```

Imputation with robCompositions

```
> names(xImp)
```

```
[1] "xOrig" "xImp" "criteria" "iter" "w" "wind" "metric"
```

Imputation with robCompositions

```
> names(xImp)
```

```
[1] "xOrig" "xImp" "criteria" "iter" "w" "wind" "metric"
```

```
> xImp$xImp[1, 3]
```

Imputation with robCompositions

```
> names(xImp)
```

```
[1] "xOrig" "xImp" "criteria" "iter" "w" "wind" "metric"
```

```
> xImp$xImp[1, 3]
```

```
[1] 0.09169296
```

Imputation with robCompositions

```
> names(xImp)
```

```
[1] "xOrig" "xImp" "criteria" "iter" "w" "wind" "metric"
```

```
> xImp$xImp[1, 3]
```

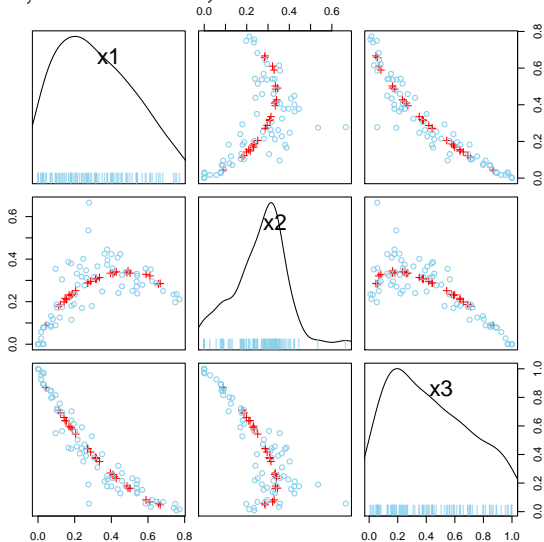
```
[1] 0.09169296
```

```
> xImp1 <- impCoda(x, method = "lm")
```

```
> xImp2 <- impCoda(x, method = "ltsReg")
```

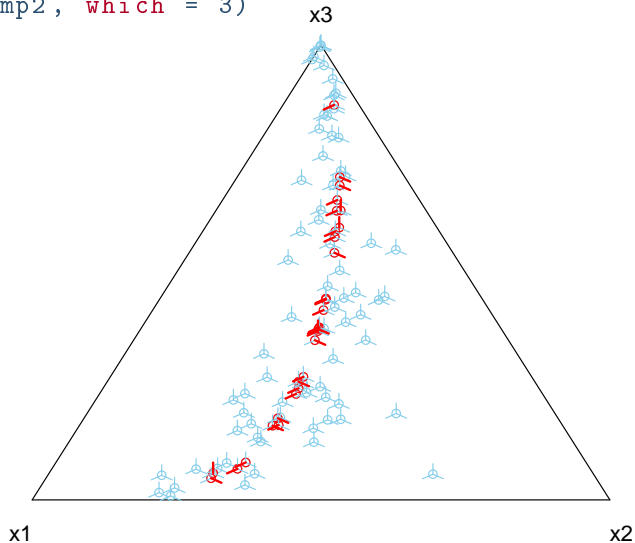
Diagnostics

```
> plot(xImp2, which = 1)
```



Diagnostics

```
> plot(xImp2, which = 3)
```



Conclusion

- We tested more than 20 imputation procedures which all were outperformed by our method (Hron, Templ, Filzmoser, 2008) for compositional data.
- Robustness is an issue. We proposed new robust imputation methods for compositional data.
- R-package `robCompositions` includes these methods, but other methods are implemented as well. Diagnostic tools are available within the package.
- A lot of important issues were not mentioned in this presentation, but they have been discussed in our NTTS-paper or in Hron, Templ, Filzmoser (2008).