

Austrian Journal of Statistics

AUSTRIAN STATISTICAL SOCIETY

Volume 45, Number 1, 2016

Special Issue on R



Österreichische Zeitschrift für Statistik

ÖSTERREICHISCHE STATISTISCHE GESELLSCHAFT



Austrian Journal of Statistics; Information and Instructions

GENERAL NOTES

The Austrian Journal of Statistics is an open-access journal with a long history and is published approximately quarterly by the Austrian Statistical Society. Its general objective is to promote and extend the use of statistical methods in all kind of theoretical and applied disciplines. Special emphasis is on methods and results in official statistics.

Original papers and review articles in English will be published in the Austrian Journal of Statistics if judged consistently with these general aims. All papers will be refereed. Special topics sections will appear from time to time. Each section will have as a theme a specialized area of statistical application, theory, or methodology. Technical notes or problems for considerations under Shorter Communications are also invited. A special section is reserved for book reviews.

All published manuscripts are available at

<http://www.ajs.or.at>

(old editions can be found at <http://www.stat.tugraz.at/AJS/Editions.html>)

Members of the Austrian Statistical Society receive a copy of the Journal free of charge. To apply for a membership, see the website of the Society. Articles will also be made available through the web.

PEER REVIEW PROCESS

All contributions will be anonymously refereed which is also for the authors in order to getting positive feedback and constructive suggestions from other qualified people. Editor and referees must trust that the contribution has not been submitted for publication at the same time at another place. It is fair that the submitting author notifies if an earlier version has already been submitted somewhere before. Manuscripts stay with the publisher and referees. The refereeing and publishing in the Austrian Journal of Statistics is free of charge. The publisher, the Austrian Statistical Society requires a grant of copyright from authors in order to effectively publish and distribute this journal worldwide.

OPEN ACCESS POLICY

This journal provides immediate open access to its content on the principle that making research freely available to the public supports a greater global exchange of knowledge.

ONLINE SUBMISSIONS

Already have a Username/Password for Austrian Journal of Statistics?

Go to <http://www.ajs.or.at/index.php/ajs/login>

Need a Username/Password?

Go to <http://www.ajs.or.at/index.php/ajs/user/register>

Registration and login are required to submit items and to check the status of current submissions.

AUTHOR GUIDELINES

The original L^AT_EX-file guidelinesAJS.zip (available online) should be used as a template for the setting up of a text to be submitted in computer readable form. Other formats are only accepted rarely.

SUBMISSION PREPARATION CHECKLIST

- The submission has not been previously published, nor is it before another journal for consideration (or an explanation has been provided in Comments to the Editor).
- The submission file is preferable in L^AT_EX file format provided by the journal.
- All illustrations, figures, and tables are placed within the text at the appropriate points, rather than at the end.
- The text adheres to the stylistic and bibliographic requirements outlined in the Author Guidelines, which is found in About the Journal.

COPYRIGHT NOTICE

The author(s) retain any copyright on the submitted material. The contributors grant the journal the right to publish, distribute, index, archive and publicly display the article (and the abstract) in printed, electronic or any other form.

Manuscripts should be unpublished and not be under consideration for publication elsewhere. By submitting an article, the author(s) certify that the article is their original work, that they have the right to submit the article for publication, and that they can grant the above license.

Austrian Journal of Statistics

Volume 45, Number 1, 2016

Special Guest Editors: Andreas ALFONS, Rainer STÜTZ

Editor-in-chief: Matthias TEMPL

<http://www.ajs.or.at>

Published by the AUSTRIAN STATISTICAL SOCIETY

<http://www.osg.or.at>

Österreichische Zeitschrift für Statistik

Jahrgang 45, Heft 1, 2016

ÖSTERREICHISCHE STATISTISCHE GESELLSCHAFT



Impressum

- Editor: Matthias Templ, Statistics Austria & Vienna University of Technology
- Editorial Board: Peter Filzmoser, Vienna University of Technology
Herwig Friedl, TU Graz
Bernd Genser, University of Konstanz
Peter Hackl, Vienna University of Economics, Austria
Wolfgang Huf, Medical University of Vienna, Center for Medical Physics and Biomedical Engineering
Alexander Kowarik, Statistics Austria, Austria
Johannes Ledolter, Institute for Statistics and Mathematics, Wirtschaftsuniversität Wien & Management Sciences, University of Iowa
Werner Müller, Johannes Kepler University Linz, Austria
Josef Richter, University of Innsbruck
Milan Stehlik, Department of Applied Statistics, Johannes Kepler University, Linz, Austria
Wolfgang Trutschnig, Department for Mathematics, University of Salzburg
Regina Tüchler, Austrian Federal Economic Chamber, Austria
Helga Wagner, Johannes Kepler University Linz, Austria
Walter Zwirner, University of Calgary, Canada
- Book Reviews: Ernst Stadlober, Graz University of Technology
- Printed by Statistics Austria, A-1110 Vienna

Published approximately quarterly by the Austrian Statistical Society, C/o Statistik Austria
Guglgasse 13, A-1110 Wien

© Austrian Statistical Society

Further use of excerpts only allowed with citation. All rights reserved.

Contents

	Page
<i>Andreas ALFONS, Rainer STÜTZ: Editorial</i>	1
<i>Marc BILL, Beat HULLIGER: Treatment of Multivariate Outliers in Incomplete Business Survey Data</i>	3
<i>Kevin JAKOB, Matthias FISCHER: GCPM: A Flexible Package to Explore Credit Portfolio Risk</i>	25
<i>Jan-Philipp KOLB: Geovisualisation: Possibilities with R</i>	45
<i>Sebastian WARNHOLZ, Timo SCHMID: Simulation Tools for Small Area Estimation: Introducing the R Package saeSim</i>	55
<i>Andreas ALFONS, Christophe CROUX, Peter FILZMOSE: Robust Maximum Association Between Data Sets: The R Package ccaPP</i>	71
<i>Thomas MENDLIK, Georg HEINRICH, Andreas GOBIET, Armin LEUPRECHT: From Climate Simulations to Statistics – Introducing the wux Package</i>	81
<i>Matthias TEMPL, Valentin TODOROV: The Software Environment R for Official Statistics and Survey Methodology</i>	97

Editorial

Since its early beginnings in 1993, the statistical computing environment and programming language **R** has grown rapidly and has been the lingua franca of the statistics research community for many years. Besides being embraced by the research community, **R** is also widely used in, e.g., IT companies, the banking and insurance sector, the biotech and pharma industry, and even newspapers and periodicals. Much of **R**'s success can be attributed to its open source implementation, the fact that it is freely available, as well as its possibility to be extended with user-contributed packages. This allowed an active community to grow, resulting in currently more than 9000 packages on the community package repositories CRAN (the Comprehensive R Archive Network, <http://CRAN.R-project.org>) and Bioconductor (<http://www.bioconductor.org>), and many more on development platforms such as R-Forge (<http://R-Forge.R-project.org>) and GitHub (<http://www.GitHub.com>). The versatility of **R** is also reflected in the wide array of topics that are covered by the articles in this Special Issue.

In the paper “Treatment of Multivariate Outliers in Incomplete Business Survey Data”, Marc Bill and Beat Hulliger shed some light onto the difficult task of outlier detection with complex survey data in the presence of missing values. The paper guides users of the **R** package **modi** through every step of such an analysis, demonstrating how to apply different functions for outlier detection and imputation that are available in the package.

The following paper, “**GCPM**: A Flexible Package to Explore Credit Portfolio Risk” by Kevin Jakob and Matthias Fischer, illustrates how the package **GCPM** can be used to model credit portfolio risk in the banking industry. The package furthermore allows to perform sensitivity analysis with respect to model assumptions, and offers several approaches to speed up computations.

In the paper “Geovisualisation: Possibilities with **R**”, Jan-Philip Kolb shows how to use **R** to download and modify geographical information from the community driven online map service OpenStreetMap (OSM). Moreover, the paper demonstrates how to incorporate this geographical information into **R**'s powerful facilities for data visualization.

The paper “Simulation Tools for Small Area Estimation: Introducing the **R** Package **saeSim**” by Sebastian Warnholz and Timo Schmid presents a simulation platform that is designed specifically for research in small area estimation. The key feature of the package **saeSim** is that every step in a simulation study is treated as a data manipulation process, which allows to write code for reproducible research that is easy to read and to modify.

In the paper “Robust Maximum Association Between Data Sets: The **R** Package **ccaPP**”, Andreas Alfons, Christophe Croux and Peter Filzmoser illustrate how to use the package **ccaPP** to compute intuitive projection-based measures of association between data sets. Furthermore, they show how to assess the significance of those measures via permutation tests and analyze their computation time.

Thomas Mendlik, Georg Heinrich, Andreas Gobiet and Armin Leuprecht address the statistical analysis of climate model simulations in the paper “From Climate Simulations to Statistics – Introducing the **wux** Package”. They thereby explain every step from reading in climate model output, processing the data for various meteorological parameters, to statistical analysis of multi-model ensembles.

Finally, the paper “The Software Environment R for Official Statistics” by Matthias Templ and Valentin Todorov discusses the advantages and possibilities of R for statistical offices, and provides an overview of relevant R packages for official statistics and survey methodology. In addition, it presents practical examples for using some of those packages, as well as for accessing important international databases from within R.

Both of us felt honored to be chosen as guest editors of this Special Issue on R of the Austrian Journal of Statistics, and it was a pleasure to collaborate on putting it into place. We are grateful to all the people who contributed to this Special Issue: to the authors for submitting such interesting articles, to the reviewers for their valuable comments, and in particular to the editor-in-chief Matthias Templ for giving us this opportunity and for his support.

The R logo on the title page is © 2016 The R Foundation, distributed under the CC-BY-SA 4.0 license (see <http://creativecommons.org/licenses/by-sa/4.0/>). No changes were made.

Andreas Alfons, Rainer Stütz
(Guest Editors)

Erasmus Universiteit Rotterdam
Erasmus School of Economics
PO Box 1738
3000DR Rotterdam
The Netherlands

AIT Austrian Institute of Technology
Mobility Department
Giefinggasse 2
1210 Wien

Vienna/Rotterdam, February 2016

Treatment of Multivariate Outliers in Incomplete Business Survey Data

Marc Bill
FHNW School of Business

Beat Hulliger
FHNW School of Business

Abstract

The distribution of multivariate quantitative survey data usually is not normal. Skewed and semi-continuous distributions occur often. In addition, missing values and non-response is common. All together this mix of problems makes multivariate outlier detection difficult. Examples of surveys where these problems occur are most business surveys and some household surveys like the Survey for the Statistics of Income and Living Condition (SILC) of the European Union. Several methods for multivariate outlier detection are collected in the R package **modi**. This paper gives an overview of the package **modi** and its functions for outlier detection and corresponding imputation. The use of the methods is explained with a business survey data set. The discussion covers pre- and post-processing to deal with skewness and zero-inflation, advantages and disadvantages of the methods and the choice of the parameters.

Keywords: outlier detection, missing value, zero-inflation, imputation, R package.

1. Introduction

In surveys on monetary values, often several monetary variables are collected in order to capture the economic situation of an entity. This holds for business surveys, where many particular types of expenditures may be asked. Examples are surveys on expenditures for research and development or investments and expenditures for environment protection. Also for household or person surveys on the economic situation, several economic variables are needed. Examples are surveys on the economic situation of students, where sources of financing, expenditures for dwelling, travelling, food etc. are needed, or household surveys like the Statistics on Income and Living Conditions (SILC) of the European Union where various income sources are collected. Of course also non-monetary quantitative variables may be collected like various health indicators in a health survey or physical production parameters in a business survey or in a survey on livestock of farms. All these surveys have some common features: They have a complex sample design including stratification and possibly sub-sampling; they have elaborated questionnaires; they have unit and item non-response, and they typically have zero inflated distributions because of the multi-faceted economic situation. Zero-inflation occurs because a particular entity usually only needs a subset of the possible dimensions to describe its situation. For example in the SILC surveys, retired persons

usually (but not always) do not have labour income. Or in a business survey on environment protection, expenditures of an educational institution may not have investments into waste water treatment.

It is challenging to deal with outliers in these situations because, in addition to the above problems, the size effect of families, farms or businesses may yield heavily skewed distributions. There may be outliers which are correct observations and which must be taken into account when population totals, like waste water treatment expenditures of a branch of economy, must be estimated. However, taking the outliers into account introduces a large variability and, if actually the values are not correct, entail a large bias, too. [Chambers \(1986\)](#) introduced the notion of a representative outlier to conceptualise this dilemma.

In sample surveys, the definition of an outlier usually cannot rely on a parametric model. The outlier generating mechanisms may depend on other variables and therefore are not simple mixture models as in classical robust statistics (see, e.g., [Hampel, Ronchetti, Rousseeuw, and Stahel 1986](#)). Therefore, [Béguin and Hulliger \(2008\)](#) introduced the notion of an outlier at random, and [Hulliger and Schoch \(2013\)](#) discuss a full model of outlier generating mechanisms and the connections between missingness and outlyingness.

It is important to note that the results in the following example differ widely whether weights are used or not. In general, the outlyingness, the missingness mechanisms (item non-response), the sample design and the unit non-response mechanism are correlated and cannot be ignored. The package **modi** ([Hulliger 2015](#)) for the statistical environment R ([R Core Team 2014](#)) can handle outlyingness, missingness and sample design issues at once and in a multivariate framework.

To the best of our knowledge, the R package **modi** is at the moment the only package that at the same time deals with missing values and survey weights. The package **rrcovNA** ([Todorov 2014b](#)) is an extension of the package **rrcov** ([Todorov and Filzmoser 2009](#); [Todorov 2014a](#)) with methods that cope with missing values but does not take survey weights into account. Therefore, a comparison is difficult. A notable paper comparing the outlier detection algorithms of an earlier version of the packages **rrcovNA** and **modi** is [Todorov, Templ, and Filzmoser \(2011\)](#).

Section 2 gives an overview of the package **modi**. Section 3 introduces the SEPE data set which is used in Section 4 to show the application of different methods of the package. Section 5 gives a conclusion.

2. Overview of the **modi** package

Several multivariate outlier detection and imputation procedures are contained in Version 1.6 of the package **modi**. The BACON-EEM algorithm ([Béguin and Hulliger 2008](#), function **BEM()**) detects outliers under the assumption of a multivariate normal distribution. The BACON-EEM algorithm starts from an outlier free subset, uses the EM-algorithm to estimate the center and scatter of the observations of this subset and then judges outlyingness of the full data set by the Mahalanobis distance (MD) in order to define a new outlier free subset. This procedure is iterated until convergence. Sampling weights are taken into account. The Transformed Rank Correlation (TRC) algorithm ([Béguin and Hulliger 2004](#), function **TRC()**) uses Spearman rank correlations and, similar to [Maronna and Zamar \(2002\)](#), an orthogonal transformation to arrive at a robust estimate of the mean and covariance. Before the transformation, missing values are imputed provisionally by simple robust regression. The Epidemic Algorithm (EA) ([Béguin and Hulliger 2004](#)) is based on a type of data depth. It is run forward to detect outliers (function **EAdet()**) and backward to impute for outliers (function **EAimp()**). The GIMCD algorithm ([Béguin and Hulliger 2008](#), function **GIMCD()**) uses non-robust Gaussian imputation, i.e. an EM-algorithm, followed by a highly robust minimum covariance determinant (MCD) algorithm ([Rousseeuw and Van Driessen 1999](#)). The GIMCD algorithm is similar to the poor man's algorithm of [Todorov et al. \(2011\)](#). The nearest neigh-

Table 1: Algorithms of the package **modi**.

Algorithm	Function	Use
BACON-EEM	<code>BEM()</code>	Detection of outliers
Epidemic for detection	<code>EAdet()</code>	Detection of outliers
Epidemic for imputation	<code>EAImp()</code>	Imputation of outliers and NA's
Transformed Rank Correlation	<code>TRC()</code>	Detection of outliers
ER	<code>ER()</code>	Detection of outliers
Gaussian imputation and MCD	<code>GIMCD()</code>	Detection of outliers
Nearest Neighbour Imputation	<code>POEM()</code>	Imputation of outliers and NA's
Winsorization and Gaussian imputation	<code>Winsimp()</code>	Imputation of outliers and NA's

bour imputation algorithm POEM can cope with outliers (Charlton 2003, function `POEM()`). Another algorithm of **modi** for imputation is based on winsorization followed by a multivariate normal model (function `Winsimp()`). Also the first robust multivariate detection algorithm which was adapted to missing values by Little and Smith (1987) is included (function `ER()`). A list of the algorithms of the **modi** package is shown in Table 1.

In addition to the algorithms, the **modi** package contains a set of utility functions which are mostly internal. To mention are a plot for Mahalanobis distances (function `PlotMD()`) based on the χ^2 -distribution or the F-distribution, implementing the proposal by Little and Smith (1987), function `weighted.var()` to calculate weighted variances analogue to the base function `weighted.mean()`, and function `MDmiss()` to calculate Mahalanobis distances when missing values occur.

The package **modi** contains two data sets. The bushfire data (Campbell 1989) with a version that contains missing values (`bushfirem`) and a set of (fictive) weights (`bushfire.weights`). The bushfire data set is used in the examples of the package documentation. The second data set `sepe` stems from a real survey on environment expenditures of private companies carried out by the Swiss Federal Statistical Office. It is explained in detail in Section 3.

3. The SEPE data set

The `sepe` data set is an anonymised sample of the pilot survey on environment protection expenditures of the Swiss private economy conducted in 1993 by the Swiss Federal Statistical Office. The units are enterprises and the monetary variables are in thousand Swiss Francs (CHF). The data contain 675 observations on 23 variables overall. The sample design is stratified according to branch of economy and size. The sampling rate increases with the size class of the strata. For confidentiality reasons, a random subsample was chosen from the original sample and certain enterprises were excluded specifically. In addition, random small perturbation was added to certain variables, and some categories have been collapsed. The data set has missing values where in the original data collection the respondent indicated that the value was a guess rather than copied from records. The `sepe` data set has first been prepared for the FP5 project EUREDIT (Charlton 2003) and later been used as protected data for educational purposes. For this demonstration of the **modi** package, we focus on 8 variables representing the most important expenditure-areas (`exp`) and investment-areas (`inv`). In particular, the areas are water protection(`wp`), waste management (`wm`) and air protection (`ap`). The variables for noise protection (`np`) and other protection areas (`op`) are excluded from the following examples. The chosen variables are `totinvwp`, `totinvwm`, `totinvap`, `totinvto`, `totexpwp`, `totexpwm`, `totexpap` and `totexppto`, where the prefix `tot` indicates that the variables are the aggregates over all types of investment or expenditures. The variable naming includes the abbreviated type of spending in the middle and the area at the end. The variables `totinvto` and `totexppto` are the overall total expenditure and

Table 2: Summary statistics of the SEPE data (original scale).

	Mean	Std-Dev	Min	Med	Max	No. Miss
<code>totinvwp</code>	23.68	212.44	0	0	8400	48
<code>totinvwm</code>	15.57	342.70	0	0	18108	53
<code>totinvap</code>	78.13	1331.12	0	0	88248	53
<code>totinvto</code>	136.66	1490.55	0	0	88359	82
<code>totexpwp</code>	22.38	176.93	0	0	8800	90
<code>totexpwm</code>	30.42	236.87	0	3	6490	118
<code>totexpap</code>	8.35	125.86	0	0	8430	72
<code>totexppto</code>	58.11	340.41	0	7	16440	161

investment in all environmental protection areas, respectively. Since these overall totals have been collected, the balance condition (sub-totals adding to totals) does not always hold. These inconsistencies are not dealt with in the present analysis but methods for their treatment can be found in [Luzi, De Waal, Hulliger, Di Zio, Pannekoek, Kilchmann, Guarnera, Hoogland, Manzari, and Tempelman \(2007\)](#) and [Charlton \(2003\)](#). The sampling weight (`weight`) has been adjusted for non-response in the stratum by using the ratio of population size divided by the net sample size.

The descriptive univariate statistics in Table 2 show the strong asymmetry present in the data. To handle this, in a first step we logarithmize the data set by applying the transformation $\log(x+1)$.

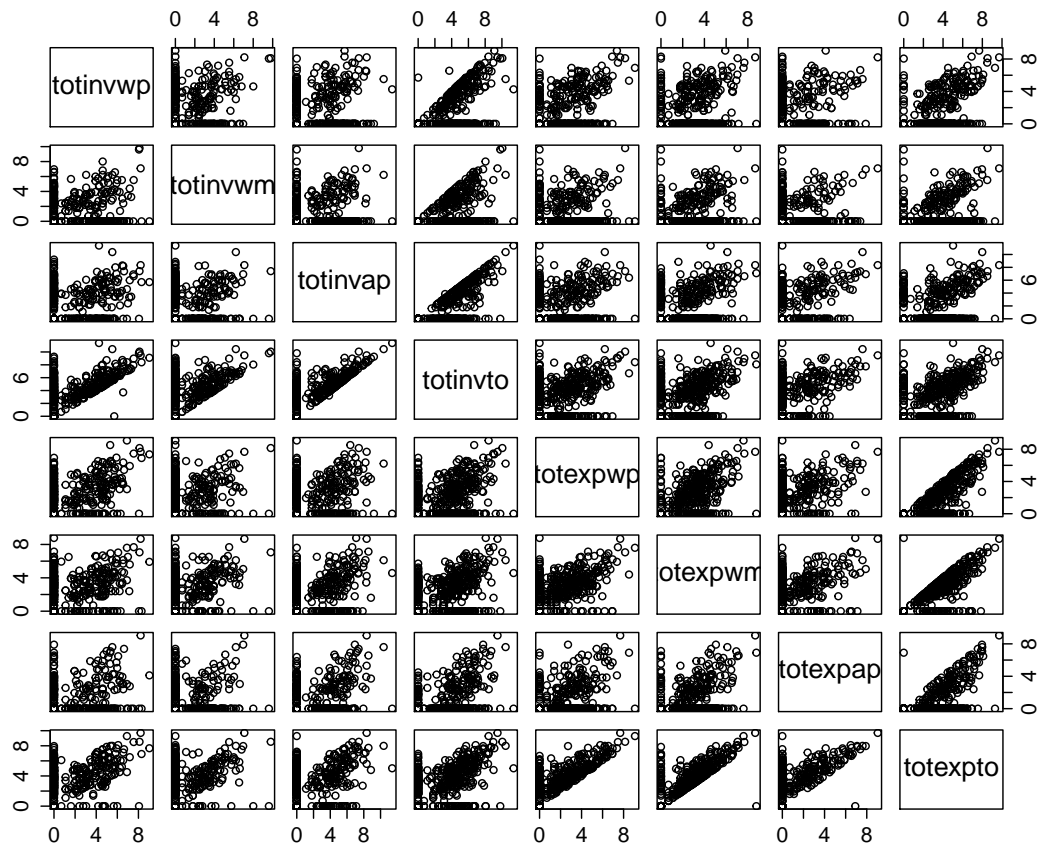
```
> data("sepe")
> sepevar8 <- c("totinvwp", "totinvwm", "totinvap", "totinvto",
+             "totexpwp", "totexpwm", "totexpap", "totexppto")
> data <- log(sepe[, sepevar8] + 1)
```

The `pairs()` plot in Figure 1 of the transformed data reveals the previously discussed positive correlation between the overall totals (`totexppto` and `totinvto`) and the subcategories. A large part of the observations lies on the axes, reflecting zeros, but there is always a part of the data forming a more or less elliptically shaped bivariate distribution. Univariate and bivariate outliers are visible. Besides 2'595 items containing the value zero, we have 677 missing values out of 5'400 items considered here. Hence, for the algorithms BACON-EEM, TRC and GIMCD, we need an additional step in data preparation. Because these algorithms have an underlying distributional assumption of multivariate normality, they have difficulties handling data sets with zero inflated distributions. Declaring the zeros as missing values, brings the data into the elliptical-like shape needed to run these functions.

```
> sepenozero <- recode(sepe, "0=NA")
> datanozero <- log(sepenozero[, sepevar8] + 1)
```

If we do not set the zeros to missing values, additional problems occur: The values of the median absolute deviation (MAD) may be 0 and standardisation is not possible then or the covariance matrix is singular. In the first case, the MADs can be substituted by a user specified probability quantile of the absolute deviations from the median. For the second case, we would have to reduce the dimension of the data set by omitting variables.

In the sections on the specific functions, we give recommendations at which stage it is preferable to reintroduce the removed information of zeros. Except for the Epidemic Algorithm, it is indispensable for outlier detection to declare zeros as missing values or to treat the zero-inflation in some way.

Figure 1: `pairs()` plot of the logarithmized `sepe` data.

4. Applying the methods

After running an outlier detection algorithm, a suitable imputation method should fill in the missing data to allow for the analysis of a “clean” and complete data set. The **modi** package contains four detection and three imputation functions. Detection with `BEM()`, `TRC()` and `GIMCD()` is based on the assumption of multivariate normality (of the bulk of the data) and thus we might want to use the same assumption for the imputation. Then `Winsimp()` or `POEM()` would be suitable choices, and here we stick to `Winsimp()`. After detecting outliers by `EAdet()`, it seems logical to use `EAimp()` for the imputation. In the following, these combinations are tested on the `sepe` data. The nearest neighbour imputation function `POEM()` is suitable with all detection algorithms since it combines local and global features. Nevertheless, it is not part of the present analysis, but `POEM()` has been extensively applied and tested in [Charlton \(2003\)](#).

4.1. `BEM()` – BACON-EEM

Outlier detection by `BEM()`

The BACON-EEM algorithm, developed in [Béguin and Hulliger \(2008\)](#), is implemented in the function `BEM()`. The BACON-EEM algorithm starts with a relatively small subset of good observations, i.e. observations which are not outliers. The mean and the covariance matrix of the good observations are estimated with the EM-algorithm. The estimates in the EM-algorithm must take into account the sample design, which gives reason for the name BACON-EEM. Then the Mahalanobis distance (MD) of all data points is calculated and the observations which are below the cutpoint defined through a χ^2 -quantile are the new

good subset. The algorithm iterates further until convergence. The most important control parameters in `BEM()` are the size of the initial good subset as a multiple `c0` of the dimension of the data, and the probability `alpha` to determine the quantile of the χ^2 -distribution for the cutpoint. Running `BEM()` on `sepe` with its default values for the starting subset fails, since the covariance-matrix used to calculate the MD is singular. The reason is that the starting subset for the algorithm is too small and contains too many missing values.

```
> BEM(data, sepe$weight)
Warning: missing observations 193 244 301 374 375 546 559
564 618 619 661 removed from the data

Error in qr.default(EM.var.good) :
  NA/NaN/Inf in foreign function call (arg 1)
```

A remedy is to set the zeros in the data set to missing values:

```
> sepenozero <- recode(sepe, "0=NA")
> datanozero <- log(sepenozero[, sepevar8] + 1)
```

However, `BEM()` returns the identical error as before even though it removes all data with completely missing observations. When increasing the initial subset size by setting `c0=5`, the algorithm works.

```
> BEM(datanozero, sepe$weight, c0 = 5)
Warning: missing observations 2 4 11 12 41 57 ...
... 655 656 657 659 661 662 665 666 removed from the data

BEM has detected 385 outlier(s) in 1.92 seconds.
```

The function `BEM()` returns a list whose first component `output` contains the summary information. Only 517 out of the 675 observations of `sepe` can be used. Hence, 158 observations were dropped due to complete missingness. The initial subset size contains 40 observations (`c0 * ncol(data)`). The algorithm used 1.92 seconds for the computation¹. The cutpoint of 21.16 is the minimum (squared) MD of the observations which are considered outliers. In general, the cutpoint is the (squared) MD which is above $c_{Npr} \cdot \chi_{p,1-\alpha}^2$, where the constant c_{Npr} is approximately 1 and α is a small proportion $0 < \alpha < 0.5$ with default `alpha=0.01` (Béguin and Hulliger 2008, p. 94). All observations with a (squared) MD larger or equal than the cutpoint are declared outliers. Before discussing this parameter more specifically, a reflection on the plausibility of the detected outliers is appropriate. Almost 75% of the observations (namely 385 out of 517) have been declared as outlying. The reliability of this result is highly questionable. This reveals an issue with large data sets: the choice of the tuning parameter `alpha`.

Alpha

The tuning parameter α (argument `alpha` of the function `BEM()`) represents a probability, indicating the level $1 - \alpha$ of the cut-off quantile based on a χ_p^2 distribution for good observations. If the analysis does return unreasonable amounts of outliers, one should decrease α . A rule of thumb is to divide a conventional value of α like the default $\alpha = 0.01$ by the number of observations (`alpha=0.01/nrow(data)`) for sample sizes above 100 observations. This strategy returns plausible results: Among the 517 observations which are usable for the analysis, 89 outliers have been detected in 1.37 seconds.

¹All computations have been executed on a 2.90GHz Intel Core i7 CPU with 8.00 GB RAM.


```
> BEM.r <- BEM(datanozero, sepe$weight, c0 = 5, alpha = 0.01/nrow(datanozero))
> PlotMD(BEM.r$dist, ncol(datanozero), alpha = 0.95)
```

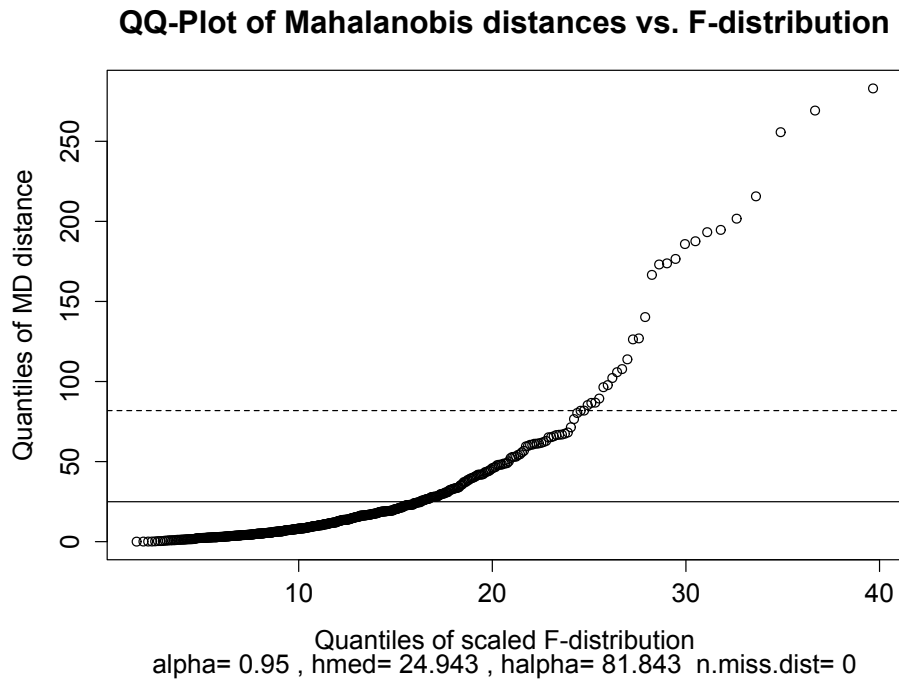


Figure 2: BEM() – QQ-plot of the Mahalanobis distances.

Cutpoint

The corresponding QQ-Plot in Figure 2 created by `PlotMD()` compares the (squared) MD with the F-distribution as proposed in (Little and Smith 1987). The Figure is practically the same (including the values) for the χ^2 distribution which can be chosen by setting the argument `chisquare=TRUE`. The default cutpoint is the minimal (squared) MD of the outliers. The value 37.14 of the default cutpoint is not appropriate and a higher value would fit the data better. There are two alternatives to set a more appropriate cutpoint. One can a) try to identify a substantial change in the distribution plot of the MD, or b) define a certain fraction of the data to be outlying. A good cutpoint in Figure 2 seems to be around 70 because there is a distinct upwards bend at that point. Implementing this cutpoint leads to a reduction of the number of outliers from 89 to 31.

```
> outind <- (ifelse(BEM.r$dist > 70, 1, 0))
> outind <- as.logical(recode(outind, "NA=0"))
> sum(outind)
[1] 31
```

Choosing option b), the default cutpoint is replaced by declaring a specific fraction of the observations with highest Mahalanobis distance as outliers. To preserve comparability between different algorithms, 5% of the total number of observations are declared outliers. Assuming that the fraction of outliers is identical in missing and non-missing data, the 5% is a fraction of the total number of usable observations. For `BEM()`, the number of usable observations is 517 and thus 26 observations are declared outliers. The cutpoint is 82.53.

```
> (cutpoint5 <- quantile(BEM.r$dist, 0.95, na.rm = TRUE))
95%
82.52671
```

```

> outind5 <- (BEM.r$dist > cutpoint5)
> outind5 <- as.logical(recode(outind5, "NA=0"))
> sum(outind5)
[1] 26

```

To get an idea of how the algorithm selected the outliers and what effect a different cutpoint might have, Figure 3 shows total expenditure against total investment. Note that all zeros have been set to missing values during the detection step, but now have been re-inserted for the plot. In Figure 3, the good items are marked as black circles and the 89 outliers detected while using the default values with red crosses. The blue rectangles represent the 31 outliers determined by the visual inspection of the QQ-Plot (cutpoint = 70). The 5% most extreme points are a further subset of the blue rectangles. Naturally, this two-dimensional plot is not fully adequate for multivariate analysis. This becomes apparent in Figure 3, where increasing the cutpoint does declare points to the upper-right as good points which would rather be outliers in the two-dimensional plot.

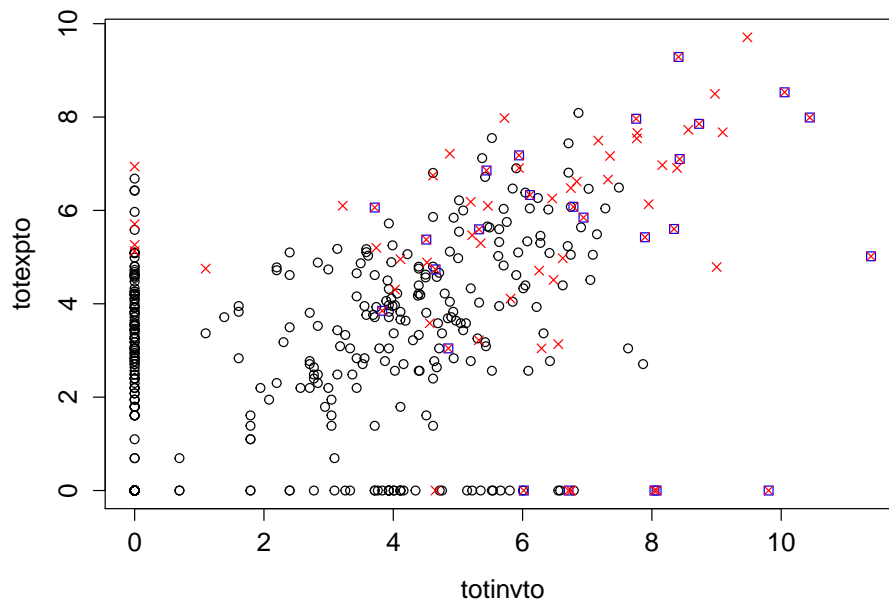


Figure 3: BEM() – Plot of total expenditures and investments. Outliers are marked as \times (default cutpoint=37.14) and as \square (visually chosen cutpoint=70); good observations are marked as \circ .

Imputation by Winsimp()

For imputation the visually determined cutpoint of 70 is applied. The next issue to be solved is at what stage the initial zeros should be re-inserted, as described in Section 3. There are two possibilities: re-insertion of the zeros before or after the imputation. There are arguments for both options. Re-inserting zeros before imputation yields imputations which are overall more coherent with multivariate normality since the multivariate normal relation among the variables is maintained also for those observations with 0 values. Re-insertion of zeros after imputation takes into account that the zeros did not contribute to the multivariate normal model implicitly used for the detection. However, the multivariate normal relation between the variables is broken. Both options are tested here. The imputed data sets are saved in the output object `Winsimp.r$imputed.data`. Unfortunately, the imputation gives negative values in expenditure and investment data. Since we assume those to be strictly positive, we censor the negative values to zero.


```

> # re-inserting zeros before imputation
> Winsimp.r <- Winsimp(data, BEM.r$output$center,
+                       BEM.r$output$scatter, outind)
> Winsimp.r$imputed.data <- ifelse(Winsimp.r$imputed.data < 0, 0,
+                                   Winsimp.r$imputed.data)
> zeros_before <- Winsimp.r$imputed.data

> # re-inserting zeros after imputation
> Winsimp.r <- Winsimp(datanozero, BEM.r$output$center,
+                       BEM.r$output$scatter, outind)
> zeros_after <- Winsimp.r$imputed.data
> zeros <- (ifelse(sepe[, sepevar8] == 0, 1, 0))
> zeros <- recode(zeros, "NA=0")
> zeros_after <- ifelse(zeros == 1, 0, zeros_after)

```

Note that function `Winsimp()` does not recalculate the center and scatter of the data but uses the results of `BEM()`. The resulting weighted means of the eight variables and the corresponding determinant of the covariance-matrix are shown in Table 3. Means and covariance-matrices are all weighted by the sampling design. For calculations with the original (raw) data, missing values are left out list-wise (column ‘Original’ in Table 3). Calculations with the non-robust EM-algorithm are also shown (column ‘Normal’). At which stage the zeros are re-inserted seems to play a significant role for the final data set, see the last two columns of the table. Re-insertion after imputation yields 63 negative values compared to 105 when re-insertion is before imputation. Negative values are replaced by zero. It is difficult to judge the shift of the means towards higher values after outlier detection and imputation (see Table 3). Also the change of the determinant of the covariance matrix does not allow a neat conclusion. It seems that re-insertion before imputation yields a very compact covariance matrix. Re-insertion of the zeros after imputation does inflate the scatter more, but still less than with the non-robust EM-imputation with package **norm** (Novo and Schafer 2013). The small determinant for re-inserted zeros before imputation shows evidence that extreme observations have been declared as outliers. The fact that the determinant for re-insertion after imputation is smaller than with **norm** shows that the robust imputation still yields a more compact determinant, which is less influenced by outliers.

Table 3: `BEM()` – Means and determinant of covariance matrix for original, normally imputed and robustly imputed data after re-insertion of zeros in different steps.

	Original	Normal	Before	After
<code>totinvwp</code>	0.71	0.73	0.78	0.87
<code>totinvwm</code>	0.47	0.56	0.72	0.69
<code>totinvap</code>	0.88	1.00	1.04	1.12
<code>totinvto</code>	1.51	1.81	1.69	1.88
<code>totexpwp</code>	0.99	1.05	1.04	1.11
<code>totexpwm</code>	1.53	1.62	1.48	1.61
<code>totexpap</code>	0.48	0.47	0.45	0.51
<code>totexpto</code>	2.01	2.12	1.96	2.19
Determinant	4.86	16.01	4.22	10.75

Notes: ‘Original’ refers to the original logarithmized data. For the calculations missing values are removed list-wise. ‘Normal’ refers to data imputed with the (non-robust) EM-algorithm. ‘Before’ and ‘After’ refers to re-insertion of zeros before or after imputation with the robust method `Winsimp()`. Means and determinants of the covariance-matrix are calculated with the package **survey** (Lumley 2004, 2014) and are hence weighted for the sample design.

4.2. TRC() – Transformed Rank Correlation

Outlier detection by TRC()

The algorithm of the transformed rank correlation (TRC), which is described in detail in [Béguin and Hulliger \(2004\)](#), is implemented in the function `TRC()`. The initial covariance matrix is calculated with Spearman-correlations which are standardised to be consistent at the multivariate normal distribution. The TRC algorithm uses an orthogonal transformation of the data into the space of eigenvectors, where the center and scatter is recalculated with robust univariate estimators (medians and median absolute deviations). For this transformation, complete data is needed and the TRC algorithm uses a provisional imputation of missing values based on the best simple robust regression available in the data. This provisional imputation is discarded later on, when the Mahalanobis distances are calculated. Nevertheless it is a critical step because contrary to the EM-algorithm, TRC uses just one—in principal the best—predictor for each variable. Thus important control parameters for the function `TRC()` are the minimal proportion `gamma` of the observations needed to determine an imputation model, and the minimal correlation `mincorr` needed to use a regressor in the provisional imputation. The first trial with `TRC()` uses the original data set (log-scale) and the default values. The algorithm returns the following warning:

```
> TRC.r <- TRC(data, sepe$weight)
Warning: missing observations 193 244 301 374 375
546 559 564 618 619 661 removed from the data

Number of missing items: 589 , percentage of missing items: 0.110881
Some mads are 0. Using 0.75 quantile absolute deviations!
The following variable(s) have 0.75 quantile absolute deviations equal to 0 :
1 2 7
Error in TRC(data, sepe$weight) :
  Remove these variables or increase the quantile probability
```

To run the algorithm with all variables, either the probability quantile must be set to at least 0.81 or the zeros must be set to missing values. The second option is preferable, since increasing the quantile would not solve the problem that TRC is based on the assumption of multivariate normal data, which is not the case when the zeros are left in the data. A run of the algorithm with its default values and `monitor=TRUE` shows all computational steps:

```
> TRC(data = datanozero, weight = sepe$weight, monitor = TRUE)
Warning: missing observations 2 4 11 12 ...
638 644 646 648 655 656 657 659 661 662 665 666 removed from the data

Number of missing items: 2008 , percentage of missing items: 0.4854932
End of preprocessing in 0 seconds
Computing Spearman Rank Correlations :
...
Spearman Rank Correlations (truncated and standardized):
      [,1]      [,2]      [,3]      [,4]
[1,] 1.0000000 0.133597112 0.633970052 0.8621370
...
[8,] 0.8640993 0.8396824 0.8141143 1.0000000
End of Spearman rank correlations estimations in 0.14 seconds
Regressors correlations
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1.110223e-16 0
...
 0 0 0 0 0 0 0 0 0 0 0.8396824 0 -1.110223e-16
Variable 1 :
 68 obs imputed using regressor 2 (cor= 0.1335971 slope= 0 intercept= 3.258097 )
 80 obs imputed using regressor 3 (cor= 0.6339701 slope= 0 intercept= 3.258097 )
```

```

...
Variable 8 :
  48 obs imputed using regressor 6 (cor= 0.8396824 slope= 0.812663 intercept= 1.173297 )
...
  3 obs imputed using regressor 7 (cor= 0.8141143 slope= 0 intercept= 3.044522 )
End of imputation in 0.03 seconds

TRC has detected 147 outlier(s) in 0.19 seconds.

```

Gamma

The monitored values of the Spearman rank correlations (`cor`) seem reasonable, but the simple robust regressions for the imputation of missing values to construct a positive semi-definite covariance-matrix have a slope of 0 (`slope=0`). This is a consequence of the large number of missing values and the default requirement that at least half of the observations must determine the correlation in order to be used for imputation (`gamma=0.5`). To lower this requirement the tuning parameter `gamma` may be set to a lower value, e.g. assuming that a sufficient number to run the regression is 30 observations.

```

> TRC.r <- TRC(data = datanozero, weight = sepe$weight,
+             monitor = TRUE, gamma = 30/TRC.r$output$sample.size)
> PlotMD(TRC.r$dist, ncol(datanozero))

```

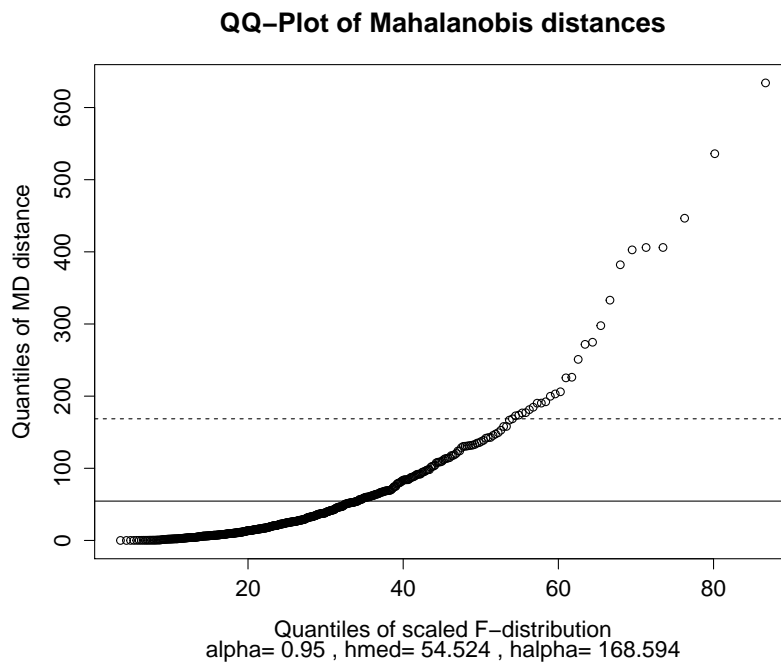


Figure 4: `TRC()` – QQ-plot of the Mahalanobis distances.

Now the imputation regression runs smoothly with non-zero slopes and the outlier detection works without a warning. `TRC()` drops 158 observations due to complete missingness. Out of the remaining 517 observations, 146 are declared as outliers. 48.5% of the items are missing values. The computation time is 0.18 seconds. The default cutpoint is

$$\text{median}(MD) \cdot F^{-1}(1 - \alpha, p, n - p) / F^{-1}(0.5, p, n - p), \quad (1)$$

where F^{-1} is the quantile function of the F-distribution. In this case, the default cutpoint is 54.52. The QQ-plot in Figure 4 created by `PlotMD()` shows the (squared) MD at the

corresponding F-distribution. It is easily seen that the proposed cutpoint of 54.52 is not appropriate and a higher value would fit the data better.

Cutpoint

An appropriate cutpoint is determined by inspection of Figure 4. The first upward-jump visible in the QQ-plot is at about 210. Setting all observations with an MD above 210 to outliers returns 14 outliers compared to 146 by default. Defining 5% of the data to be outlying returns the same number of outliers (26) as BACON-EEM. However, even if the same number of outliers is identified, the individual outliers may be different (see Table 7).

Imputation by Winsimp()

For the imputation of the TRC-detected outliers, function `Winsimp()` is used. As already discussed in section 4.1.4, an issue is at what stage we re-insert the zeros. Imputed negative values are censored to zero. Re-insertion after imputation yields 44 and re-insertion before imputation 123 negative values. The resulting means and determinants of the original and imputed data are shown in Table 4. Means differ between the re-insertion methods and tend to be higher in the imputed data compared to the original. The deviation is statistically not significant. Since the determinant of the covariance-matrix is smaller than for the original data with re-insertion of the zeros before imputation, it seems that that outliers have successfully been detected and imputed to a more appropriate value. However, the determinant is inflated considerably if the zeros are re-inserted after imputation. Therefore, re-insertion of zeros before imputation is preferable.

Table 4: `TRC()` – Means and determinant of covariance matrix for original and imputed data after re-insertion of zeros in different steps.

	Original	Normal	Before	After
<code>totinvwp</code>	0.71	0.73	0.78	0.87
<code>totinvwm</code>	0.47	0.56	0.72	0.69
<code>totinvap</code>	0.88	1.00	1.04	1.12
<code>totinvto</code>	1.51	1.81	1.69	1.88
<code>totexpwp</code>	0.99	1.05	1.04	1.11
<code>totexpwm</code>	1.53	1.62	1.48	1.61
<code>totexpap</code>	0.48	0.47	0.45	0.51
<code>totexpto</code>	2.01	2.12	1.96	2.19
Determinant	4.86	16.01	6.33	12.73

Notes: See Table 3.

4.3. `GIMCD()` – Gaussian Imputation and Minimum Covariance Determinant

Outlier detection by GIMCD()

The GIMCD algorithm (Béguin and Hulliger 2008) is implemented in function `GIMCD()`. The GIMCD algorithm first uses a non-robust Gaussian imputation, followed by a highly robust minimum covariance determinant (MCD) algorithm to detect outliers. The only control parameter is the probability α to determine the quantile for the cutpoint. Compared to the other discussed algorithms, weights cannot be used with GIMCD since at the moment there is no implementation of MCD available which takes weights into account. At the first attempt with the `sepe` data, the algorithm fails to finish. Again the problem is the large number of zeros which result in a singular covariance matrix.

```
> GIMCD(data, seedem = 234567819, seedmcd = 4097)
Error in solve.default(cov, ...) :
  Lapack routine dgesv: system is exactly singular: U[2,2] = 0
```

Recoding the zeros to missing values enables the algorithm to compute all needed parameters. The parameter **alpha** determines a threshold value for the cut-off for the outlier Mahalanobis distances. This cutpoint is the same as (1). The larger the value of **alpha**, the more outliers will be detected. The default value **alpha**=0.05 seems reasonable for the **sepe** data. The parameter **seedem** sets a starting value for the random number generator used in the first step for the Gaussian imputation with the EM-algorithm, which is executed by the **norm** package (Novo and Schafer 2013). The default is **seedem**=234567819. Since in the second step the MCD again uses a random number generator, the seed of MCD can be set separately with **seedmcd**. The default value for MCD is taken from the system and varies for each run if not set explicitly.

```
> GIMCD.r <- GIMCD(datanozero, seedem = 234567819, seedmcd = 4097)
GIMCD has detected 57 outliers in 1.46 seconds.
> PlotMD(GIMCD.r$dist, ncol(datanozero))
```

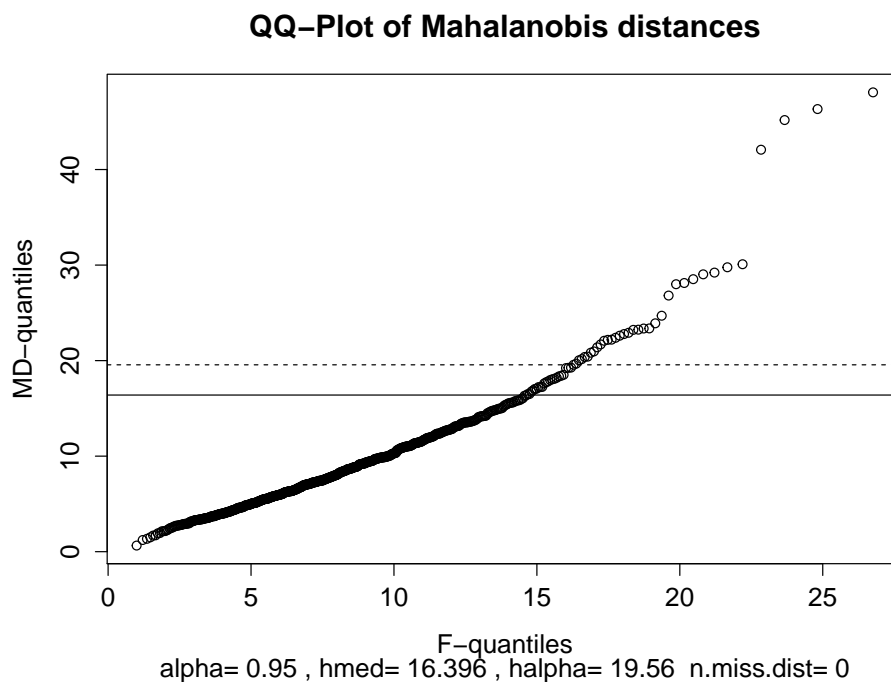


Figure 5: GIMCD() – QQ-plot of the Mahalanobis distances.

Cutpoint

The default cutpoint is defined according to (1) and evaluates to 16.4 for the **sepe** data. Visual inspection of the QQ-plot in Figure 5 favours a slightly different cutpoint, namely at 24. With this new cutpoint, only 13 outliers are detected. Cutting out 5% outliers yields 34 outlying data points. Note that GIMCD() also imputes values for the completely missing observations, unlike BEM() and TRC().

Imputation by Winsimp()

For the imputation the visually identified cutpoint of 24 is used. The imputation can be computed without any problems with function Winsimp(). Re-insertion before imputation

yields 49 negative values and re-insertion after yields 97 negative values, which are recoded to 0. A comparison of the results with re-insertion of the zeros before and after the imputation is given in Table 5. When re-inserting the zeros after imputation, the determinant is inflated more than when using the non-robust normal imputation. The means are shifted towards higher values than for the other methods. Here the lack of weighting may have had its impact.

Table 5: `GIMCD()` – Means and determinant of covariance matrix for original and imputed data after re-insertion of zeros in different steps.

	Original	Normal	Before	After
<code>totinvwp</code>	0.71	0.73	0.83	0.91
<code>totinvwm</code>	0.47	0.56	0.72	0.71
<code>totinvap</code>	0.88	1.00	1.08	1.15
<code>totinvto</code>	1.51	1.81	1.77	1.94
<code>totexpwp</code>	0.99	1.05	1.04	1.18
<code>totexpwm</code>	1.53	1.62	1.60	1.72
<code>totexpap</code>	0.48	0.47	0.47	0.55
<code>totexpto</code>	2.01	2.12	2.04	2.31
Determinant	4.86	16.01	6.47	20.81

Notes: See Table 3.

4.4. EA – Epidemic Algorithm

Outlier detection with `EAdet()`

The Epidemic Algorithm (EA) is implemented for detection in function `EAdet()` and for imputation in function `EAimp()`. Compared to the previous methods, the Epidemic Algorithm has no underlying distributional assumption. It is based on distance measures, which are described in detail in Béguin and Hulliger (2004). The basic idea of the Epidemic Algorithm is to simulate an epidemic which starts at a central point, actually a spatial median, and then infects points in the neighbourhood of the infected ones in a stepwise manner. The last infected points are nominated outliers. The exact dependence of the infection probability from the distance is determined by several functional forms with different tuning parameters. The default cutpoint is set at time $t = \text{median}(t, w) + 3 \cdot \text{MAD}(t, w)$, where the median and the median absolute deviation (MAD) are both weighted. The cutpoint may be adjusted after the calculation of the infection times by `EAdet()`. The detection function `EAdet()` can cope with the original data set in spite of the 48% items with value zero. The algorithm gives a warning because the median absolute deviation is replaced by the 90% quantile of absolute deviations, but the results can be calculated nonetheless. All points are infected within 0.38 seconds. For the `sepe` data the cutpoint results in only 7 outliers. However, there is a further set of 11 points which have never been infected. This may happen because they have too many missing values or because they are too far outlying. The function value `outind` is a logical vector with `TRUE` for the outliers (late infected), `FALSE` for the good observations (early infected) and `NA` for the never infected observations. In the present case the 11 never infected observations consist entirely of missing values.

```
> EAdet.r <- EAdet(data, sepe$weight)
```

```
Some mads are 0. Standardizing with 0.9 quantile absolute deviations!
EA detection has finished with 664 infected points in 0.38 seconds.
```

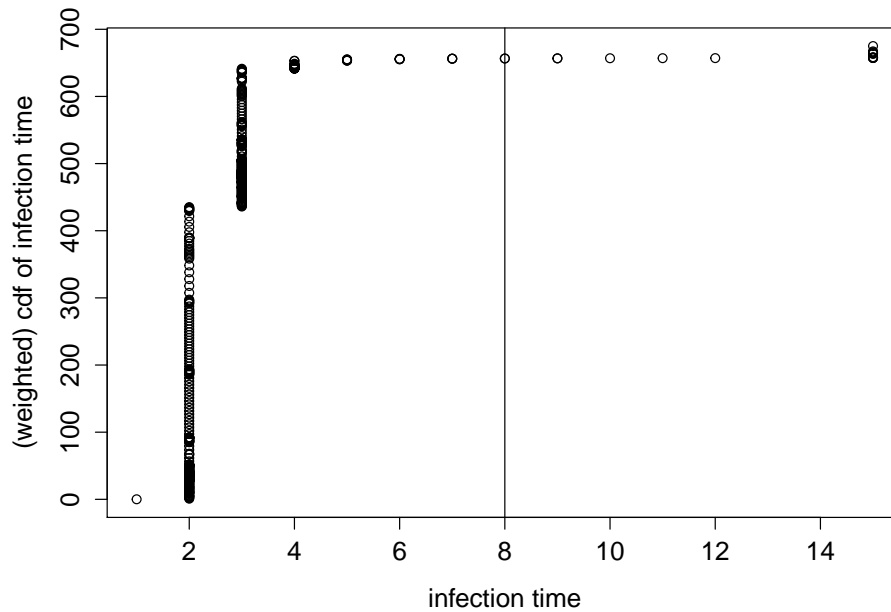


Figure 6: EA – Infection times and default cutpoint.

By default, the (weighted) cumulative distribution function of the infection times is plotted, see Figure 6. `EAdet()` detects less outliers than the other algorithms with the default cutpoint. Some tuning might be necessary even if the algorithm works with default parameters. Setting the parameter `monitor=TRUE` shows the exact number of infected observations at every step. Starting from the spatial median, which for `sepe` is a point with zero investments and only missing values in expenditures, the algorithm infects more than one third of the points in the first step. In the second step, already over 90% of the points are infected. Closer inspection shows that the Epidemic Algorithm suffers from the many zeros and missing values because the inter-point distances are calculated on the basis of the jointly observed values only. As a result, the infection probabilities do not discriminate enough between observations to be infected next, and the epidemic stops after 12 steps, i.e. at infection time 12. Also the transformation of the data has its influence on the duration of the epidemic. Points which should not be declared outliers might catch the epidemic when the transformation over-corrects the skewness of the data. It is of course possible to discard completely missing observations (you may set the parameter `rm.missing=TRUE`) and it is also possible to set zeros to missing before running `EAdet()`. Since the algorithm worked nevertheless we may use it as is and judge later on about the result.

Cutpoint

Also with `EAdet()` setting a good cutpoint needs some care. The default outlier rule declares observations which are larger or equal to 8 as outliers. Looking at the (weighted) cumulative distribution function of the infection times in Figure 6, it seems more reasonable to set the cutpoint to 5:

```
> outind <- EAdet.r$infection.time >= 5
> sum(outind, na.rm = TRUE)
[1] 20
> sum(is.na(outind))
[1] 11
```

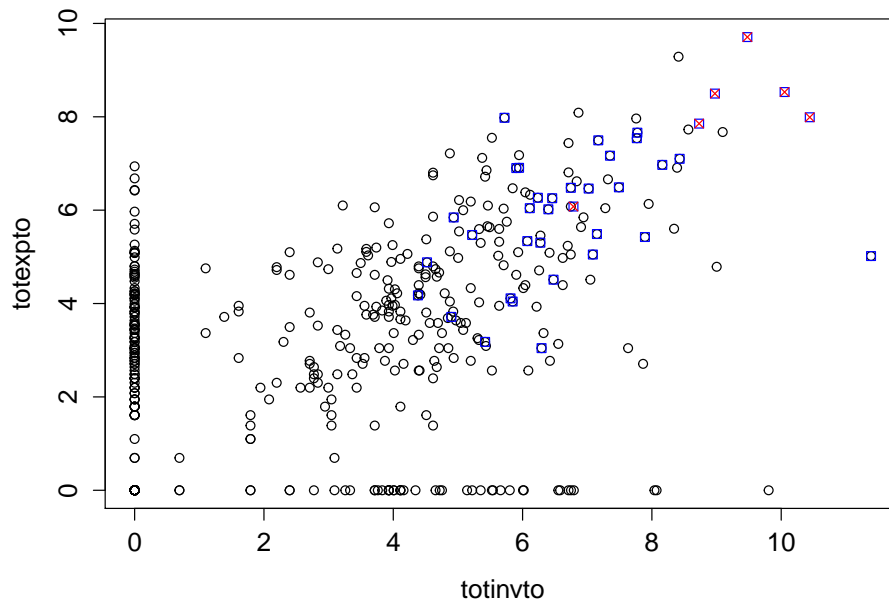


Figure 7: EA – Outliers with default cutpoint (red crosses) and visually set cutpoint (blue rectangles).

Using the cutpoint 5 yields 20 outliers. The corresponding plot in Figure 7 shows a different picture than we know from the other algorithms. First of all, there are substantially less outliers. Second, no points on the axes are outlying, and third, the point with largest `totinvto`—which the other algorithms declared an outlier—is not an outlier. `EAdet()` seems to be able to handle the multivariate outlier problem, because most outliers are not clearly outlying in one dimension. However, `EAdet()` has problems with outliers that contain many zeros.

Imputation by `EAimp()`

For imputation after detection with `EAdet()` and with the visually chosen cutpoint, the function `EAimp()` is used. Since the zeros were not set to missing, the re-insertion is not an issue now. `EAimp()` uses the distances calculated in `EAdet()` and starts an epidemic at each observation to be imputed until donors for the missing values are infected. Then a donor is selected randomly. The imputation uses the visually chosen cutpoint 5 and takes 2.76 seconds. Table 6 contains a comparison between the original, normally imputed and EA-imputed center and scatter measures. Most of the means of EA-imputed data are between the original and the normally-imputed means. Using the visually chosen cutpoint, the determinant of the imputed data is inflated but far from the determinant of the normally imputed data.

```
> EAimp.r <- EAimp(data, weights = sepe$weight, outind = EAdet.r$outind,
+                  duration = EAdet.r$output$duration)
```

Missing values in outlier indicator set to FALSE.

```
Dimensions (n,p): 675 8
Number of complete records 449
Number of records with maximum p/2 variables missing 633
Number of imputands is 230
Reach for imputation is max
```


Number of remaining missing values is 0

Table 6: EA – Means and determinant of covariance-matrix for original and imputed data.

	Original	Normal	EA
totinvwp	0.71	0.73	0.73
totinvwm	0.47	0.56	0.55
totinvap	0.88	1.00	0.97
totinvto	1.51	1.81	1.78
totexpwp	0.99	1.05	1.03
totexpwm	1.53	1.62	1.49
totexpap	0.48	0.47	0.42
totexpto	2.01	2.12	2.00
Determinant	4.86	16.01	10.69

Notes: See Table 3.

5. Conclusion

Multivariate outlier detection starts before running outlier detection algorithms such as the ones implemented in the **modi** package. Every data set has its unique issues which need to be solved before detection. Balance rules, missing value patterns as well as distributions have to be checked. E.g. the **sepe** data set has a zero inflated distribution and hence needs to be prepared to satisfy the distributional assumptions of the parametric algorithms. When the assumptions are satisfied, the parameters of the outlier detection function need to be chosen. Even though the algorithms in package **modi** have a high power in detecting multivariate outliers, user-intervention to choose the cutpoint is necessary. Checking of the imputed data is also necessary, e.g. to censor imputed data to positive values.

Choosing an appropriate method is difficult since all presented methods have advantages and disadvantages. Table 7 shows an individual comparison of the detected outliers. The number of outliers is approximately 5% (26, 26, 34, 49 for BACON-EEM, TRC, GIMCD, EA respectively). Only 5 points are selected by all four methods. The parametric methods (BACON-EEM, TRC, GICMD) have further 9 outliers in common. In terms of the Jaccard-distances between the outlier sets, BACON-EEM and TRC are closest, GIMCD somewhat detached from BACON-EEM and TRC, and EA has a large distance to all three parametric methods.

Table 8 summarizes the results of the four algorithms illustrated in the paper. The number of outliers detected by default varies strongly between the methods. However, if the cutpoint is chosen by the user, the different algorithms yield more similar numbers of outliers. Still we have to keep in mind that even though the absolute number of outliers is similar, the effectively detected points are not (see Table 7).

When applying the functions in **modi** to bigger data sets, computation time becomes a more and more crucial criteria. The fastest algorithm is TRC with only 0.28 seconds for detection and imputation in this example. EA is the slowest, since the imputation uses an epidemic for each outlier, which slows down the computational speed.

Looking at the determinants of the covariance-matrices lets us favour to re-insert the zeros before we run the imputation algorithms for the parametric methods. Determinants are smaller when zeros are re-inserted before imputation compared to the re-insertion after imputation. In any case, all four methods (except GIMCD with re-insertion after imputation) have a

Table 7: Outliers according to the four methods.

ID	BACON-EEM	TRC	GIMCD	EA	C	ID	BACON-EEM	TRC	GIMCD	EA	C
3	0	0	1	0	1	330	1	1	1	1	4
6	1	0	0	0	1	333	0	0	0	1	1
13	0	1	0	0	1	340	1	1	1	0	3
14	0	1	0	0	1	341	0	0	0	1	1
18	0	0	0	1	1	344	0	0	0	1	1
21	1	1	0	0	2	380	0	0	0	1	1
25	0	0	0	1	1	381	0	0	0	1	1
31	1	1	1	0	3	382	0	1	0	0	1
38	0	0	1	0	1	383	0	0	0	1	1
59	0	1	0	0	1	391	1	1	1	0	3
68	0	0	1	0	1	396	0	0	0	1	1
78	0	0	1	0	1	406	0	0	0	1	1
91	0	0	0	1	1	408	0	0	0	1	1
93	0	0	0	1	1	414	0	0	0	1	1
101	0	0	0	1	1	421	0	0	0	1	1
102	0	0	0	1	1	424	0	0	0	1	1
128	0	0	0	1	1	425	1	0	1	0	2
133	1	1	1	0	3	431	1	0	0	1	2
134	0	1	1	1	3	437	0	0	1	0	1
137	0	0	1	0	1	439	0	0	1	0	1
154	0	0	0	1	1	441	1	0	0	1	2
156	0	0	0	1	1	448	1	1	1	0	3
157	0	0	0	1	1	449	0	0	1	1	2
165	1	1	1	1	4	456	0	0	0	1	1
173	1	1	0	0	2	459	0	0	0	1	1
178	0	0	1	0	1	460	0	0	0	1	1
186	1	1	1	1	4	461	1	1	1	1	4
192	0	1	0	1	2	468	0	1	1	0	2
194	1	1	1	1	4	471	0	0	1	0	1
200	0	0	0	1	1	475	0	0	0	1	1
206	0	0	0	1	1	480	1	0	0	0	1
238	0	0	1	0	1	484	0	0	0	1	1
241	0	0	0	1	1	485	0	0	0	1	1
257	0	0	0	1	1	489	1	0	0	0	1
265	0	0	1	0	1	491	0	0	0	1	1
267	0	0	1	0	1	517	1	0	0	0	1
273	1	1	1	0	3	555	0	1	0	0	1
282	1	1	1	0	3	561	0	0	0	1	1
288	0	0	0	1	1	587	0	0	0	1	1
291	0	0	1	0	1	624	0	0	1	0	1
307	1	1	0	0	2	626	1	0	0	0	1
309	1	1	1	1	4	641	0	0	0	1	1
324	1	1	1	0	3	648	0	0	1	0	1
328	0	0	0	1	1	654	1	1	1	0	3

Notes: ID is the observation number. An entry 1 in the columns BACON-EEM to EA indicates that the corresponding method nominates the observation as an outlier. C is the number of coinciding methods.

Table 8: Summary of results for all functions.

Algorithm	No. Outliers		Time		Determinant	
	Default	Visual	Det.	Imp.	0's Before	0's After
BEM() – Winsimp()	89	31	1.4	0.07	4.22	10.75
TRC() - Winsimp()	146	14	0.23	0.05	6.33	12.73
GIMCD() - Winsimp()	57	20	1.48	0.05	6.47	20.81
EAdet() - EAimp()	7	20	0.36	2.54	10.69	

smaller determinant of the covariance matrix than when imputing in a non-robust way with the EM-algorithm. This is desirable since the outlying data points are normally far away of the center and blow up the determinant.

The distribution of the **sepe** data set is far from multivariate normal. Nevertheless, methods with an underlying assumption on multivariate normality may return usable results when the distribution is uni-modal apart from the zero-inflation, which must be treated explicitly. BACON-EEM and TRC extract sufficient information from the data to detect relevant outliers (they identify 14 identical outliers out of 26). GIMCD seems to be able to cope with the structure of the data in spite of not taking into account the survey weights. However the zeros must be re-inserted before imputation to preserve robustness. EA does not rely on the global structure of the data and does not need special treatment of the zero values. However, it needs careful tuning in order to differentiate outliers sufficiently well from good points.

Multivariate outliers are difficult to detect. If in addition missing values and zero-inflation occur, there are few algorithms that cope with such data. The presented algorithms are capable of doing this. However, their use is not straightforward and careful evaluation of the results is needed.

The package **modi** is currently available in Version 1.6 on R -Forge and will be submitted to CRAN (the Comprehensive R Archive Network, <http://CRAN.R-project.org>). The use of a deterministic MCD algorithm (function `covMcd()` in package **robustbase** (Rousseeuw, Croux, Todorov, Ruckstuhl, Salibián-Barrera, Verbeke, Koller, and Maechler 2014)) and a better tuning of the progression of the Epidemic Algorithm will be evaluated for the next version of the package.

Acknowledgement

The research for this paper was supported by the Swiss Federal Office for Education and Science through the European Union FP6 project EUREDIT, by the European Union FP7 project AMELI (EU FP7-SSH-2007-217322) and by FHNW School of Business. Thanks go to two anonymous referees and to the editors for their valuable comments.

References

- Béguin C, Hulliger B (2004). “Multivariate Outlier Detection in Incomplete Survey Data: the Epidemic Algorithm and Transformed Rank Correlations.” *Journal of the Royal Statistical Society, Series A: Statistics in Society*, **167**(2), 275–294.
- Béguin C, Hulliger B (2008). “The BACON-EEM Algorithm for Multivariate Outlier Detection in Incomplete Survey Data.” *Survey Methodology*, **34**(1), 91–103.
- Campbell N (1989). “Bushfire Mapping Using NOAA AVHRR Data.” *Technical report*, Commonwealth Scientific and Industrial Research Organisation.
- Chambers R (1986). “Outlier Robust Finite Population Estimation.” *Journal of the American Statistical Association*, **81**(396), 1063–1069.
- Charlton J (ed.) (2003). *Towards Effective Statistical Editing and Imputation Strategies – Findings of the Euredit project*, volume 1 and 2. EUREDIT consortium. URL <http://www.cs.york.ac.uk/euredit/results/results.html>.
- Hampel FR, Ronchetti EM, Rousseeuw PJ, Stahel WA (1986). *Robust Statistics: The Approach Based on Influence Functions*. John Wiley & Sons.

- Hulliger B (2015). **modi**: *Multivariate Outlier Detection and Imputation for Incomplete Survey Data*. R package version 1.6, URL <http://R-Forge.R-project.org/projects/modi/>.
- Hulliger B, Schoch T (2013). “Mechanisms for Multivariate Outliers and Missing Values.” In *Proceedings of the NTTS 2013 Conference*. Brussels, Belgium.
- Little R, Smith P (1987). “Editing and Imputation for Quantitative Survey Data.” *Journal of the American Statistical Association*, **82**(397), 58–68.
- Lumley T (2004). “Analysis of complex survey samples.” *Journal of Statistical Software*, **9**(1), 1–19.
- Lumley T (2014). **survey**: *Analysis of Complex Survey Samples*. R package version 3.30-3, URL <http://CRAN.R-project.org/package=survey>.
- Luzi O, De Waal T, Hulliger B, Di Zio M, Pannekoek J, Kilchmann D, Guarnera U, Hoogland J, Manzari A, Tempelman C (2007). “Recommended Practices for Editing and Imputation in Cross-Sectional Business Surveys.” *Technical report*, ISTAT, CBS, SFSO, Eurostat.
- Maronna R, Zamar R (2002). “Robust Estimates of Location and Dispersion for High-Dimensional Datasets.” *Technometrics*, **44**(4), 307–317.
- Novo AA, Schafer JL (2013). **norm**: *Analysis of Multivariate Normal Datasets with Missing Values*. R package version 1.0-9.5, URL <https://CRAN.R-project.org/package=norm>.
- R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>.
- Rousseeuw P, Croux C, Todorov V, Ruckstuhl A, Salibián-Barrera M, Verbeke T, Koller M, Maechler M (2014). **robstbase**: *Basic Robust Statistics*. R package version 0.91-1, URL <http://CRAN.R-project.org/package=robustbase>.
- Rousseeuw PJ, Van Driessen K (1999). “A Fast Algorithm for the Minimum Covariance Determinant Estimator.” *Technometrics*, **41**(3), 212–223.
- Todorov V (2014a). **rrcov**: *Scalable Robust Estimators with High Breakdown Point*. R package version 1.3-8, URL <http://CRAN.R-project.org/package=rrcov>.
- Todorov V (2014b). **rrcovNA**: *Scalable Robust Estimators with High Breakdown Point for Incomplete Data*. R package version 0.4-7, URL <http://CRAN.R-project.org/package=rrcovNA>.
- Todorov V, Filzmoser P (2009). “An Object-Oriented Framework for Robust Multivariate Analysis.” *Journal of Statistical Software*, **32**(3), 1–47.
- Todorov V, Templ M, Filzmoser P (2011). “Detection of Multivariate Outliers in Business Survey Data with Incomplete Information.” *Advances in Data Analysis and Classification*, **5**(1), 37–56.

Affiliation:

Beat Hulliger

FHNW School of Business

University of Northwestern Switzerland

4600 Olten, Switzerland

E-mail: beat.hulliger@fhnw.ch

URL: <http://www.fhnw.ch/people/beat-hulliger>

GCPM: A Flexible Package to Explore Credit Portfolio Risk

Kevin Jakob
University of Augsburg

Matthias Fischer
University of Erlangen-Nuremberg

Abstract

In this article, we introduce the novel **GCPM** package, which represents a **g**eneralized **c**redit **p**ortfolio **m**odel framework. The package includes two of the most popular modeling approaches in the banking industry, namely the CreditRisk⁺ and the CreditMetrics model, and allows to perform several sensitivity analyses with respect to distributional or functional forms assumptions. Therefore, besides the pure quantification of credit portfolio risk, the package can be used to explore certain aspects of model risk individually for every arbitrary credit portfolio. The way the package is implemented combines a high level of flexibility and performance together with a maximum of usability. Furthermore, the package offers the possibility to apply simple pooling techniques to speed up calculations for large portfolios as well as the opportunity to combine simulation models with a user specified importance sampling approach. The article concludes with a comprehensive example demonstrating the flexibility of the package.

Keywords: credit risk, portfolio model, model risk, R, Monte Carlo simulation, pooling, CreditRisk⁺, CreditMetrics.

1. Introduction

Banks apply credit portfolio models in order to quantify the amount of economic capital which must be withheld in order to cover unexpected losses caused by credit defaults. As the financial crisis had shown very impressively, the use of quantitative models is always accompanied by a certain amount of model risk which has to be taken into account whenever decisions or price evaluations are based on them. Nowadays, banks are explicitly requested by supervisors to validate their quantitative models and to quantify model risk (see [Board of Governors of the Federal Reserve System 2011](#)). Ignoring model risk can lead to wrong management decisions and an underestimation of the true risk. The **GCPM** package addresses both of these issues – quantification of credit risk and an analysis of the underlying model risk.

A great advantage of **GCPM** over other available packages for R ([R Core Team 2014](#)), like **QRM** ([Pfaff and McNeil 2014](#)) or **CreditMetrics** ([Wittmann 2007](#)), is that it utilizes an object oriented approach, where one object consists of a specified model together with all portfolio information and risk figures (once the portfolio loss distribution was estimated). Therefore, it is easy to handle different models (or portfolios) simultaneously without jeopardizing their

consistency. As the example in Section 5 will show, performing comparison or sensitivity studies is very simple. In addition, the package is able to deal with large portfolios. On the one hand, portfolios with several thousands of counterparties can be used, whereas in our tests the **CreditMetrics** package was unable to handle more than one hundred portfolio positions. On the other hand, and in contrast to the **QRM** package¹, risk parameters like the probability of default, the loss ratio in case of a default, the exposure and the assignment to a specific industry sector and country affecting the default dependencies can be defined individually for each counterparty. Together with a C++ implementation of the simulation framework, which takes advantage of modern multi-core systems, the package combines flexibility regarding counterparty characteristics and distributional assumptions with good performance and makes it suitable for practical applications. Moreover, for advanced users, simulation models can be combined with self-defined importance sampling techniques and counterparty pooling approaches in order to stabilize simulation results and to increase performance furthermore.

Please note that we will not address any questions regarding the parametrization of the models. In contrast, in order to guarantee a maximum flexibility regarding the distributional assumptions, we have to leave this task up to the user. However, we will provide several examples and demonstrate how already existing packages and basic R functions can be used to construct a parametrization (i.e. a sample from the multivariate sector distribution). For those who are interested in this topic, we refer to [Hamerle and Rösch \(2006\)](#). Please also note that the package focuses on credit risk only with respect to default events, i.e. migration risk is not considered.

The article is organized as follows. A short overview of credit portfolio models together with common notation is given in Section 2. Afterwards, we present the simulation framework and the derivation of risk contributions. The last section contains a hypothetical example, explaining how the package **GCPM** can be used to quantify credit and model risk. Here, starting from the basic CreditRisk⁺ model (see [Credit Suisse First Boston International 1997](#)), which is characterized by certain distributional assumptions, we show how risk figures might change if these assumptions are modified. Along with this, the available functions of the package are introduced including a simple pooling technique which will be useful for homogeneous portfolios (e.g. retail portfolios).

2. Credit portfolio modeling

2.1. Input data, loss distribution and risk figures

The key function of a classical bank is to hand out loans to enterprises or private persons. For reason of simplicity, let us assume that the loan portfolio consists of M loans given to M different counterparties or obligors. In this situation the bank faces the risk that one or more obligors default which means that they are not able or willing to pay back the out-standing amounts (principal and interests) which, in turn, leads to financial losses. The main purpose of a credit portfolio model is to forecast the portfolio loss distribution for the underlying loan portfolio and a fixed time interval, usually one year. Regardless of the specific modeling approach (two of them are introduced in the subsequent sections), every model requires the following set of information on each counterparty i :

- The exposure at the time of default (EAD_i),
- the probability of default (PD_i) for the given time horizon, usually one year,
- the loss given default rate (LGD_i) or recovery rate ($RR_i = 1 - LGD_i$, amount recovered

¹The **QRM** package also provides the possibility to evaluate so-called Bernoulli mixture models but only with respect to the number of defaults. Therefore, analyzing a portfolio with different default probabilities, exposures and sector affiliations is not possible.

through foreclosure or bankruptcy procedures in the event of default, expressed as a percentage of EAD_i) and

- the assignment of the obligor to predefined industry and/or country sectors in order to rebuild the dependence structure of the portfolio.

With this notation, the overall portfolio loss L reads as

$$L := \sum_{i=1}^M L_i = \sum_{i=1}^M D_i \cdot LGD_i \cdot EAD_i,$$

where $D_i \sim \text{Ber}(PD_i)$ is the default indicator² for obligor i (i.e. $PD_i = P(D_i = 1)$).

Under the assumption that the parameters LGD and EAD are deterministic and the loss distribution F_L has already been derived, the following key figures are required for the bank's risk reporting and management information (see also Figure 1 for a graphical representation):

- Expected loss $\mathbb{E}(L) = \sum_{i=1}^M PD_i \cdot LGD_i \cdot EAD_i$.
- Standard deviation $SD(L) = \left[\sum_{i,j=1}^M EAD_i \cdot EAD_j \cdot LGD_i \cdot LGD_j \cdot \text{Cov}(D_i, D_j) \right]^{1/2}$.
- Value at Risk $\text{VaR}_\alpha := \inf\{l | F_L(l) \geq \alpha\}$ for a specified level $\alpha \in (0, 1)$.
- Economic capital $\text{EC}_\alpha := \text{VaR}_\alpha - \mathbb{E}(L)$.
- Expected shortfall or expected tail loss $\text{ES}_\alpha := \mathbb{E}(L | L \geq \text{VaR}_\alpha)$.

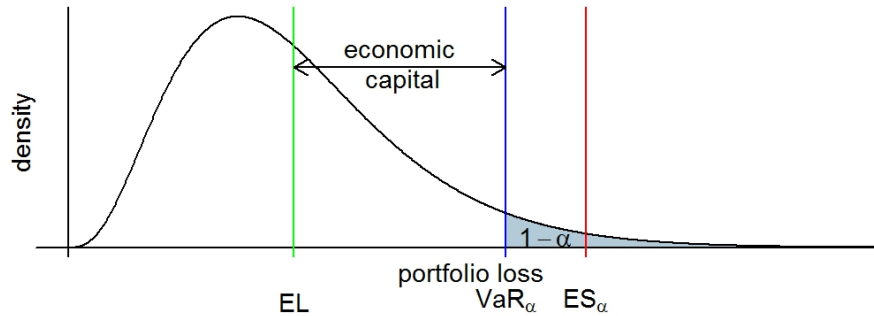


Figure 1: General portfolio loss distribution with risk figures.

In practice, VaR_α and EC_α constitute the relevant risk measures. For example, in the regulatory framework of Basel II (see [Basel Committee on Banking Supervision 2006](#)), a loss level of $\alpha = 0.999$ is used to quantify the economic capital.

Whereas the expected loss can be calculated directly from the raw portfolio data, the calculation of the loss distribution in general is a crucial issue. It requires the knowledge of the dependence structure (so-called "default correlations") between the M default indicators D_1, \dots, D_M , where M is typically large. To simplify this problem and reduce the dimension, every counterparty is assigned to one or more out of $K \ll M$ industry and/or country sectors such that dependence between obligors can be traced back to the belonging to same sectors and to the dependence structure between them. The sectors themselves are modeled via a (multivariate) latent variable \mathbf{S} which is distributed according to some K -dimensional distribution³ on \mathbb{R}^K .

² $\text{Ber}(p)$ denotes the Bernoulli distribution with success probability $p \in (0, 1)$.

³How the concrete sector distribution looks like depends on the type of portfolio model (i.e. on the link function) and the calibration, which will be discussed on the following pages.

The **GCPM** package deals with two of the most popular credit portfolio models, namely CreditRisk⁺ and CreditMetrics, which are briefly summarized in the following subsections. Whereas CreditRisk⁺ and its generalizations provide an analytic solution under certain restrictive distributional assumptions, CreditMetrics calculates the portfolio loss distribution within a simulation framework which is more flexible but also more time-consuming. For further details on these portfolio models we also refer to Crouhy, Galai, and Mark (2000) or Gordy (2000) who provide an excellent comparative analysis of these models.

2.2. The CreditRisk⁺ model

The CreditRisk⁺ model was developed by the Financial Products division of Credit Suisse in 1997, see Credit Suisse First Boston International (1997) for a detailed documentation. It belongs to the class of so-called Poisson mixture models where the intensity of the Poisson distribution (which approximates the Bernoulli distribution of the default indicator D_i) itself is driven by Gamma-distributed random variables. Relying on these specific stochastic assumptions and a discretization of the exposures, it is possible to express the probability mass function of the portfolio loss (or, equivalently, its probability generating function⁴) in a closed analytical form, which is a great advantage of CreditRisk⁺ and its major difference to its competitors. Hence, even for larger portfolios the risk figures can be obtained within a reasonable run-time.

More formally, the basic idea of the model can be summarized as follows: In a first step, a discretization parameter L_0 , called loss unit is introduced. All exposures are approximated by an integer multiple of this unit via $\nu_i = \max \left\{ \left\lceil \frac{\text{EAD}_i \cdot \text{LGD}_i}{L_0} \right\rceil, 1 \right\}$, where $\lceil x \rceil$ denotes the nearest integer value to x . The default probabilities are adjusted such that the discretization does not affect the expected loss. The adjusted PD is given by

$$\widetilde{\text{PD}}_i = \frac{\text{EAD}_i \cdot \text{LGD}_i \cdot \text{PD}_i}{\nu_i \cdot L_0}. \quad (1)$$

As for the calculation of the loss distribution, the loss unit represents the width of the exposure bands on which the marginal probabilities are calculated. For more details, please see Credit Suisse First Boston International (1997, para A 3.2).

Secondly, a further key assumption is to replace the default indicator D_i (naturally Bernoulli distributed) with a Poisson distributed random variable \tilde{D}_i with intensity parameter λ_i . This assumption is necessary in order to compute the portfolio loss distribution analytically. Because, in most cases, λ_i will be very small, the approximation error is not substantial. But if credit quality decreases, the effect of multiple defaults becomes crucial.

Finally, the intensity parameter of each obligor is mapped onto one or more (economic) sectors in order to introduce dependence between the counterparties belonging to the same sector via sector weights. Given a sector realization $\mathbf{s} = (s_1, \dots, s_K)^T$ of \mathbf{S} , the conditional default intensity reads as:

$$\lambda_i^{\mathbf{S}} := \widetilde{\text{PD}}_i \left(w_{i,0} + \sum_{k=1}^K w_{i,k} s_k \right), \quad (2)$$

with

- the individual adjusted $\widetilde{\text{PD}}_i$,
- individual sector weights $w_{i,k} \in [0, 1]$ for obligor i with respect to sector k such that $\sum_{k=1}^K w_{i,k} \leq 1$ and the idiosyncratic weight $w_{i,0} = 1 - \sum_{k=1}^K w_{i,k}$,

⁴For a discrete random variable X with values in \mathbb{N} , the probability generating function (PGF) is defined as $G(z) := \mathbb{E}(z^X)$.

- sector variables S_1, \dots, S_K which are assumed to be mutually independent and Gamma distributed with variance⁵ σ_k^2 and $\mathbb{E}(S_k) = 1$ such that $\mathbb{E}(\lambda_i^S) = \text{PD}_i = \lambda_i$.

Under these assumptions, the default correlation between obligor i and j reads as:

$$\text{Cor}(\tilde{D}_i, \tilde{D}_j) = \frac{\sqrt{\tilde{\text{PD}}_i \cdot \tilde{\text{PD}}_j}}{\sqrt{(1 - \tilde{\text{PD}}_i)(1 - \tilde{\text{PD}}_j)}} \sum_{k=1}^K w_{i,k} w_{j,k} \sigma_k^2.$$

In order to calculate the probability mass function (PMF) of the portfolio loss, a modified⁶ version of the algorithm given in [Haaf, Reiss, and Schoenmakers \(2003\)](#) is used. The algorithm calculates the marginal probabilities that the portfolio loss is equal to $\nu \cdot L_0$ with $\nu \in \mathbb{N}_0$. It stops if a desired level of the cumulative distribution function (CDF) has been reached.

In order to keep the notation simple and comparable to the CreditMetrics model, we will denote the adjusted PD with PD_i as well, instead of $\tilde{\text{PD}}_i$, in the remainder of this article. Switching back to the original notation does not imply that this approximation is unimportant. Please bear in mind that, if an inappropriately large loss unit L_0 is used, the discretized PDs and hence also the risk figures may be changed noticeably.

2.3. The CreditMetrics model

The CreditMetrics model, described in [Gupton, Finger, and Bhatia \(1997\)](#), is a typical representative of so-called threshold models. The fundamental idea grounds on the firm value model of [Merton \(1974\)](#). For each counterparty i an asset value variable is defined as

$$A_i := \mathbf{R}_i^T \mathbf{S} + \sqrt{1 - \mathbf{R}_i^T \Sigma \mathbf{R}_i} \epsilon_i, \quad (3)$$

where $\mathbf{R}_i \in \{[-1, 1]^K \mid \mathbf{R}_i^T \mathbf{R}_i < 1\}$ determines the correlation of i 's asset value to the systemic factors $\mathbf{S} \sim \mathcal{N}_K(\mathbf{0}, \Sigma)$ ⁷. The idiosyncratic risk is expressed by $\epsilon_i \sim \mathcal{N}(0, 1)$ which are independent from each other as well as from \mathbf{S} . A default occurs if the asset value A_i falls below the default threshold, defined by $\Phi^{-1}(\text{PD}_i)$ where Φ denotes the distribution function of a standard normal variable. Conditioning on a realization \mathbf{s} of the systemic factor \mathbf{S} the probability of default is given by

$$\text{PD}_i^S = \frac{\Phi^{-1}(\text{PD}_i) - \mathbf{R}_i^T \mathbf{s}}{\sqrt{1 - \mathbf{R}_i^T \Sigma \mathbf{R}_i}}. \quad (4)$$

Using formula (3), the default correlation between two counterparties reads as:

$$\text{Cor}(D_i, D_j) = \Phi_2(\Phi^{-1}(\text{PD}_i), \Phi^{-1}(\text{PD}_j), \mathbf{R}_i^T \Sigma \mathbf{R}_j),$$

where $\Phi_2(x_1, x_2, r)$ denotes the distribution function of a bivariate normal distribution with correlation parameter $r \in [-1, 1]$ and standard normal margins. The loss distribution is achieved via a Monte Carlo simulation, as described in the next section.

3. Simulation models

Alternatively to the analytical version of the CreditRisk⁺ model, one can also use a simulation setting. In this case, several distributional assumptions can be modified in order to analyze model sensitivities. By changing the link function (i.e. replacing (2) by (4)), one can also

⁵The variance σ_k^2 can either be estimated from historical default data or using analytical approximations based on the rating specific standard deviation of the PD, see [Gundlach \(2003\)](#).

⁶The loop-structure of the algorithm has been changed to calculate the CDF and the PMF simultaneously.

⁷ $\mathcal{N}_K(\mathbf{a}, \Sigma)$ denotes the K dimensional normal distribution with mean \mathbf{a} and correlation matrix Σ .

switch to a CreditMetrics-like model. Consequently, an analysis of the risk figure sensitivities with respect to the specific link function is also possible. Please take care that the sector drawings (argument `random.numbers` of the `init()` function, see Table 1) meet the correct distributional assumptions of the chosen model, defined via the link function, described in Sections 2.2 and 2.3. E.g. normally distributed sectors are not compatible with the CreditRisk⁺ setting. For each counterparty, the distribution of the default indicator D_i can be chosen individually between “Bernoulli” (natural choice) or “Poisson” (CreditRisk⁺- setting) within the portfolio data (see Table 2). Depending on these three elements (sector distribution, link function and default distribution), the basic idea of the simulation framework is to simulate N different portfolio losses. Given these losses, the portfolio loss distribution and risk figures can be estimated via the empirical loss distribution.

3.1. General simulation framework

Given a set of $N \in \mathbb{N}_{>0}$ (multivariate) sector drawings $\mathbf{s}^{(1)}, \dots, \mathbf{s}^{(N)} \in \mathbb{R}^K$ and a portfolio of M counterparties, the general simulation framework of the **GCPM** package is as follows:

Algorithm 1 Basic simulation algorithm

```

For  $n = 1, \dots, N$  #(simulation loop)
  For  $i = 1, \dots, M$  #(counterparty loop)
    Calculate conditional PD:
    If link.function == "CRP" then
       $\overline{\text{PD}}_i^{(n)} = \text{PD}_i \cdot (w_{i,0} + w_i^T \mathbf{s}^{(n)})$ 
    If link.function == "CM" then
       $\overline{\text{PD}}_i^{(n)} = \Phi \left( \frac{\Phi^{-1}(\text{PD}_i) - \mathbf{R}_i^T \mathbf{s}^{(n)}}{\sqrt{1 - \mathbf{R}_i^T \Sigma \mathbf{R}_i}} \right)$ 
    Draw default:
    If defaulti == "Bernoulli" then
       $D_i \sim \text{Bern}(\overline{\text{PD}}_i^{(n)})$ 
    If defaulti == "Poisson" then
       $D_i \sim \text{Pois}(\overline{\text{PD}}_i^{(n)})$ 
    Determine counterparty loss:
     $L_i^{(n)} = D_i \cdot \text{EAD}_i \cdot \text{LGD}_i$ 
  Determine portfolio loss:
   $L^{(n)} = \sum_{i=1}^M L_i^{(n)}$ 

```

After the simulation, the portfolio losses $L^{(n)}$ are discretized with respect to the loss unit L_0 , in order to group losses for the calculation of the probability mass function. The distribution is estimated based on the discretized simulated portfolio losses $\tilde{L} = \left(\tilde{L}^{(1)}, \dots, \tilde{L}^{(N)} \right)^T$, i.e.

$$f(L = l) = \frac{1}{N} \sum_{n=1}^N 1_{\{m \mid \tilde{L}^{(m)} = l\}}(n), \quad (5)$$

where 1_A denotes the indicator function on set A . For reasons of performance, the simulation algorithm is implemented in C++ and linked to the package via the **Rcpp** package (see Eddelbuettel and François 2011). In order to show the progress status, the **RcppProgress** package (see Forner 2013) is needed as well. In order to avoid errors during the simulation, please ensure that R can allocate enough memory from your operating system, by using the R functions `memory.size()` and `memory.limit()`. In order to increase performance within simulation models, one can also take advantage of multi-core systems. For this purpose, the **parallel** package is required (see Section 5.3.4).

3.2. Adaption of importance sampling techniques

In most cases, the risk figures are based on extreme scenarios with a low probability of occurrence. For instance, if the $ES_{0.999}$ should be estimated on a basis of 10^3 relevant scenarios (in order to achieve a reliable estimation), one has to perform 10^7 simulations. If portfolios include thousands of counterparties, the simulation will be very time-consuming and it will need lots of memory. With the help of importance sampling techniques, one can “manipulate” the simulation such that extreme scenarios occur more often and tail measures can be calculated on a higher number of simulated losses. Mathematically, importance sampling is just a change of the probability measure from P to P_{IS} . Instead of drawing random numbers from P , one can draw from P_{IS} where the probability of relevant scenarios is higher. The only restriction is that

$$\text{supp}(f) \subset \text{supp}(f_{IS}) \text{ and } f_{IS}(\mathbf{x}) > 0, \forall \mathbf{x} \in A,$$

where $\text{supp}(f_{IS})$ denotes the support of the corresponding density functions and A is the set of scenarios the risk measure is calculated on. In order to get an estimator with respect to original measure P , the standard estimator (e.g. for the mean) has to be adjusted by the so-called likelihood ratio

$$\text{LHR}(\mathbf{x}_{IS}) := \frac{f(\mathbf{x}_{IS})}{f_{IS}(\mathbf{x}_{IS})}, \quad \text{with } \mathbf{x}_{IS} \sim P_{IS}.$$

In our case, the standard estimator of the density function (5) changes to

$$f(L = l) = \frac{1}{\sum_{n=1}^N \text{LHR}(L_{IS}^{(n)})} \sum_{n: L_{IS}^{(n)} = l} \text{LHR}(L_{IS}^{(n)}). \quad (6)$$

Since a credit portfolio model in general contains a lot of different distributions, also the range of application for an importance sampling algorithm is very wide. For example, one could concentrate on the sector copula. Here, different approaches are possible. For instance, one can simply strengthen the overall level of dependence by increasing the entries of the dispersion matrix of a t-copula or by rising the degrees of freedom (e.g. see [Mai and Scherer 2012](#)). Another approach could be to concentrate on those sector drawings where extreme scenarios (e.g. exceeding the 95%-quantile) occur jointly across different sectors (see [Arbenz, Cambou, and Hofert 2014](#)). Additionally, one can also use importance sampling on the marginal distributions by shifting the mean or increasing the variance and higher moments or use a more sophisticated approach, see [Glasserman and Li \(2005\)](#).

Please note that, since the sector distribution itself can be defined arbitrarily by the user and the possibilities of importance sampling are manifold, the package does not perform any kind of importance sampling on its own. Instead, the sector drawings (`random.numbers`, see Table 1) can be simulated with a user defined importance sampling approach and passed to a portfolio model together with a vector of likelihood ratios, which will be respected when the loss distribution is calculated. In this way, as in case of the `random.numbers` matrix, the user has maximum flexibility to choose which approach is suitable in his or her situation.

For a more detailed introduction to importance sampling in general we refer to [Rubino and Tuffin \(2009\)](#).

4. Identification of risk drivers

For a portfolio manager, it is important to know which obligors within the portfolio are riskier than others. In order to identify such risk drivers, we briefly introduce different measures which are available in the package for counterparty risk contributions, i.e. contributions to standard deviation σ of the portfolio loss, value at risk, economic capital and expected shortfall. For a detailed derivation of the corresponding formulas in case of the analytical

CreditRisk⁺ model, please refer to [Credit Suisse First Boston International \(1997\)](#) and [Haaf and Tasche \(2002\)](#).

4.1. Analytical CreditRisk⁺ model

On counterparty level the following risk contributions (RC) can be calculated:

- **standard deviation:** $RC_i^\sigma = \frac{PL_i \cdot PD_i}{\sigma} \left(PL_i + \sum_{k=1}^K \sigma_k^2 w_{i,k} \ell_k \right)$, with σ_k denoting the standard deviation of sector k and $\ell_k := \sum_i w_{i,k} \cdot PD_i \cdot PL_i$ denoting the expected loss with respect to sector k ,
- **VaR_α:** $RC_i^{VaR_\alpha} = PD_i \cdot PL_i \frac{\sum_{k=1}^K w_{i,k} \mathbb{P}(L_k = VaR_\alpha - PL_i)}{\mathbb{P}(L = VaR_\alpha)}$, where L_k denotes the loss in sector k , and
- **ES_α:** $RC_i^{ES_\alpha} = PD_i \cdot PL_i \frac{\sum_{l=VaR_\alpha-PL_i}^{\bar{M}} \sum_{k=1}^K w_{i,k} \mathbb{P}(L_k=l)}{\sum_{l=VaR_\alpha}^{\bar{M}} \mathbb{P}(L=l)}$, where \bar{M} is the maximum portfolio loss a probability is calculated on (depending on `alpha.max`, see Table 1).

Please note that depending on the loss unit L_0 used for exposure discretization and the number of obligors within the portfolio, VaR contributions may be zero for some counterparties because they do not default in the single VaR-event. Therefore, it is reasonable to consider contributions to ES rather than VaR. Because ES is based on the upper tail of the loss distribution rather than a single loss level, the mentioned problem does not occur using ES contributions.

Finally, for all these measures it holds that the individual contributions sum up to the measure calculated on portfolio level. Therefore, one can also analyze contributions, for example on sector level (e.g. business lines or countries) by simply aggregating the corresponding counterparty contributions.

4.2. Simulation models

Within the simulation framework, expected shortfall contributions can be calculated. For this purpose, one has to define a loss threshold `loss.thr` > 0, which should be lower to the corresponding VaR but not too low in order to stress memory usage not too much. If the portfolio loss $L^{(n)}$ in scenario n is above `loss.thr`, all counterparty losses $L_i^{(n)}$ are stored. Counterparty risk contributions to ES on level $\alpha \in (0, 1)$ are then calculated as:

$$RC_i^{ES_\alpha} = \frac{1}{\sum_{n \in N_\alpha} LHR(L_{IS}^{(n)})} \sum_{n \in N_\alpha} LHR(L_{IS}^{(n)}) \cdot L_i^{(n)}, \quad (7)$$

where $N_\alpha := \{n = 1, \dots, N \mid L^{(n)} \geq VaR_\alpha\}$ denotes the set of all ES_α-relevant scenarios. Similar to the analytical CreditRisk⁺ model it holds that $\sum_{i=1}^M RC_i^{ES_\alpha} = ES_\alpha$.

For other tail measures (VaR and EC) the risk contributions are calculated with the same approach but with respect to another level $\tau \in (0, 1)$ such that $ES_\tau = VaR_\alpha$ or $ES_\tau = EC_\alpha$, respectively. Therefore, risk contributions to VaR and EC are approximated by risk contributions to ES but on a lower level τ . Using the ES approach instead of a direct calculation with respect to VaR or EC, risk contributions are much more stable because of the higher number of scenarios used for the calculation.

Since the portfolio loss distribution is not continuous, level τ for VaR/EC contributions is chosen such that ES_τ is as close as possible to VaR_α or EC_α , respectively. If deviations are greater or equal to 0.01% an appropriate message comes up.

5. The GCPM package

The main component of the package is the S4 class `GCPM`. Besides this class there are some additional functions, in particular for object creation. The class represents the whole portfolio model framework. It contains all model specifications as well as the portfolio and the loss distribution once it is estimated. In case of a simulation model, losses on counterparty level are also stored depending on a predefined threshold `loss.thr` (see Table 1)

In the next sections we give a detailed overview of the most important features. A complete list of all slots is available in the help pages of the package (see `?GCPM`). The following examples are based on the CreditRisk⁺ framework. Please note that the same analysis can be also performed within a CreditMetrics framework.

5.1. General structure

The overall structure of the package is very intuitive. At first, one has to initialize a new model using the `init()` function. The process of creation is as follows. Passing the input parameters for a new model to the function creates a new object of class `GCPM` with the specified settings (after some plausibility checks). For example:

```
library("GCPM")
sec.var <- c(0.2, 0.3, 0.4) #arbitrary sector variances
names(sec.var) <- c("A", "B", "C") #assign sector names to variances
CRP.classic <- init(model.type = "CRP", loss.unit = 50000, alpha.max = 0.9999,
  sec.var = sec.var)

##      Generalized Credit Portfolio Model
##      Copyright (C) 2015 Kevin Jakob & Dr. Matthias Fischer
##
##      This program is free software; you can redistribute it and/or
##      modify it under the terms of the GNU General Public License
##      version 2 as published by the Free Software Foundation.
##
##      This program is distributed in the hope that it will be useful,
##      but WITHOUT ANY WARRANTY; without even the implied warranty of
##      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
##      GNU General Public License for more details.
##
##      You should have received a copy of the GNU General Public License
##      along with this program; if not, write to the Free Software
##      Foundation, Inc., 51 Franklin Street, Fifth Floor,
##      Boston, MA 02110-1301, USA.
```

The above code generates a `GCPM` model, named `CRP.classic` with the given attributes. For some slots of the `GCPM` class, default values (e.g. for `alpha.max`) are provided, but they are not necessarily the best choice. Considering this, one should better choose them individually for each portfolio according to exposures, number of counterparties, and hardware restrictions. Depending on the `model.type`, different arguments have to be provided. A summary is given in Table 1 below.

After creating a new portfolio model, one can analyze a credit portfolio using the `analyze()` method. In case of an analytical CreditRisk⁺ model, the loss distribution will be calculated by using the algorithm described in Haaf *et al.* (2003). For simulation models, the simulation described in Algorithm 1 is used. If loss levels are provided via the parameter `alpha`, tail measures are calculated automatically with respect to those levels. Otherwise, one can calculate those measures afterwards with the corresponding methods as shown in the following examples. The portfolio data frame has to follow the structure described in Table 2.

model type	Parameter	Description
CRP	<code>alpha.max</code>	...is a numeric value between 0 and 1 defining the maximum CDF-level which will be computed.
	<code>sec.var</code>	...is a named numeric vector defining the sector variances. The names have to correspond to the sector names given in the portfolio.
simulative	<code>link.function</code>	... is a character value, specifying the type of the link function ("CRP" corresponds to equation (2) and "CM" to (4)).
	<code>N</code>	... is a numeric value, defining the number of simulations. If <code>N</code> is greater than the number of scenarios provided via <code>random.numbers</code> , scenarios are reused.
	<code>seed</code>	... is a numeric value used to initialize the random number generator. If <code>seed</code> is not provided, a value based on the current system time will be used. Therefore, the results are truly random in this case.
	<code>loss.thr</code>	... is a numeric value specifying a lower bound for portfolio losses to be stored in order to derive counterparties' risk contributions.
	<code>random.numbers</code>	... is a matrix with sector drawings. The columns represent the different sectors, whereas the rows represent the scenarios. The column names must correspond to the sector names used in the portfolio.
	<code>LHR</code>	... is a numeric vector of length equal to <code>nrow(random.numbers)</code> defining the likelihood ratio of each scenario. If not provided, all scenarios are assumed to be equally likely.
	<code>max.entries</code>	... is the number of scenarios stored to calculate risk contributions. The value should be set in consideration of the amount of available memory.

Table 1: Arguments for `init()` in case of a simulation and an analytical model.

Number	Name	Business	Country	EAD	LGD	PD	Default	A	B	C
1	Name 1	Energy	US	358475	0.989	0.001	Bernoulli	1	0	0
2	Name 2	IT	DE	1089819	0.608	0.003	Bernoulli	0	1	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 2: Structure of the portfolio data frame.

5.2. Analyzing credit risk: A first example

Based on a portfolio distributed with the package (in the package's `data` folder) consisting of 3000 counterparties and three industrial sectors, we offer an example to show how the package works. We start from the `CRP.classic` model defined in the previous section.

```
# first example
library("GCPM")
# importing portfolio
data("portfolios")
# analyzing the portfolio (Poisson defaults)
CRP.classic <- analyze(CRP.classic, portfolio.pois)

## Importing portfolio data....
## 3 sectors ...
## 3000 counterparties (0 removed due to EAD=0 (0), lgd=0 (0), pd<=0 (0) pd>=1 (0))
##
## Portfolio statistics....
## Loss unit: 50 K
## Portfolio EAD:1.5 B
```



```
## Portfolio potential loss:772.28 M
## Portfolio expected loss:130.69 M(analytical)
## Diversifiable risk: 7.67 M Systematic risk: 41.41 M
## Portfolio standard deviation:42.11 M(analytical)
## Calculate the loss distribution till 0.9999-confidence level is reached.

##
## Calculation completed...
## Reached level of confidence: 0.9999001591125 ( iterations actually done: 7073 )
##
## Calculating risk measures from loss distribution....
## Expected loss from loss distribution: 130.65 M
(deviation from EL calculated from portfolio data: -0.03%)
## Exceedance Probability of the expected loss:0.454820959078588
## Portfolio mean expected loss exceedance: 167.18 M
## Portfolio loss standard deviation:42.04 M
```

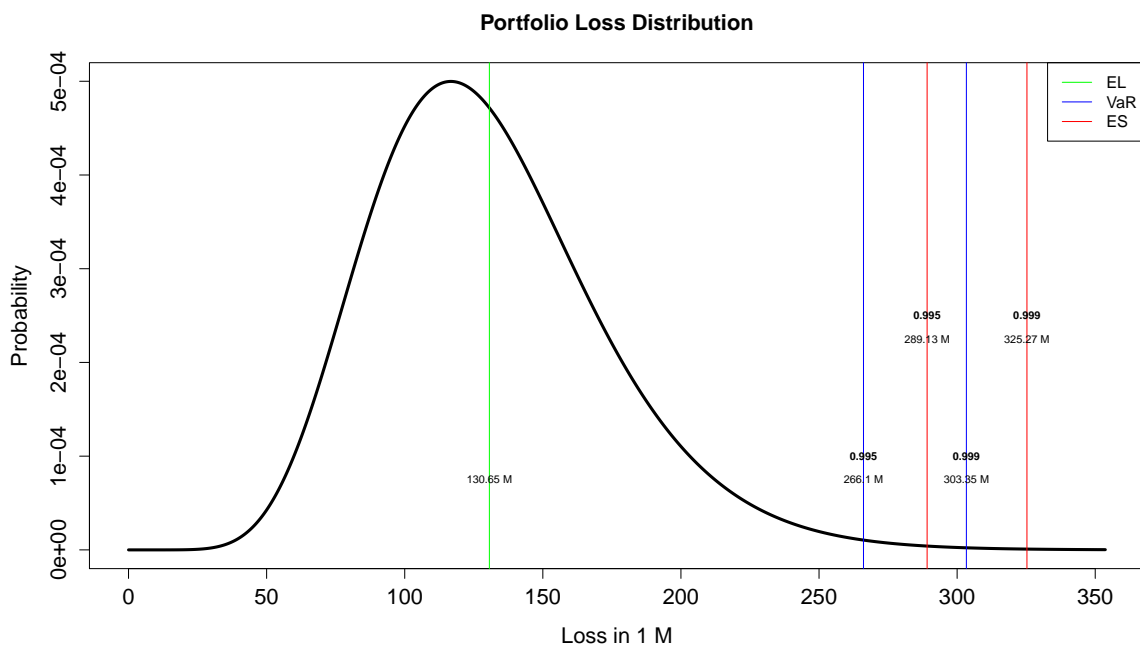
After deriving (or simulating) the loss distribution, risk measures like VaR, EC or ES can be calculated with the help of the corresponding methods.

```
alpha <- c(0.995, 0.999) #levels for tail measures
VaR(CRP.classic, alpha)

## [1] 266100000 303350000
```

The probability mass function for the loss distribution together with indicators for tail measures can be plotted by using the `plot()` function. The second argument defines the scale of the horizontal axis.

```
plot(CRP.classic, 1e+06, alpha)
```



Finally, one can calculate risk contributions in order to identify particular positions in the portfolio driving EC, VaR or ES.

```
RC.cont <- data.frame(Name = name(CRP.classic), EC.cont(CRP.classic, alpha),
  VaR.cont(CRP.classic, alpha), ES.cont(CRP.classic, alpha))
RC.cont[1:3, ]

##      Name EC.0.995 EC.0.999 VaR.0.995 VaR.0.999 ES.0.995 ES.0.999
## 1 Name  1 125170.64 170476.28 312088.98 378068.30 352293.31 419188.26
## 2 Name  2  15620.00 18351.49 22802.99 23955.69 23547.25 24518.83
## 3 Name  3  15006.43 20350.73 37055.76 44838.89 41798.39 49689.53
```

5.3. Modifying distributional assumptions

Besides the pure quantification of credit risk, the package assists in analyzing different aspects of model risk related to distributional or functional forms assumptions. Starting from the classic CreditRisk⁺ model (example of Section 5.2) we will show how the package can be used to build much more flexible models and how to quantify model risk similar to the analyses done by [Jakob and Fischer \(2014\)](#), [Fischer and Mertel \(2012\)](#) or [Fischer and Kaufmann \(2014\)](#).

A key element for this is the `random.numbers` matrix which represents the (multivariate) sector distribution. Since the dimension of this matrix depends on the portfolio, i.e. on the number of sectors used, the matrix has to be defined by the user. Additionally, the sector distribution (expressed by `random.numbers`) also heavily depends on the economic sectors it is associated with. I.e. the sector copula and the marginal distributions may be very different across geographical regions and industries. Since this is a very crucial issue, which has also a significant impact on the risk figures, this matrix must be defined by the user (i.e. no default value is provided). Furthermore, in this way, the user has maximum flexibility to define the sector distribution according to his or her needs. However, a few examples are given on the following pages. For more information about the question of sector parametrization we refer to [Hamerle and Rösch \(2006\)](#) or [Dorffleitner, Fischer, and Geidosch \(2012\)](#).

Checking for simulation error

At first we will check if the results of the simulation model correspond to the analytical one. Therefore, we create a matrix of random numbers, which are independently Gamma distributed with mean equals one and variance given by `sec.var`, which we can pass to the argument `random.numbers` of the `init()` function.

```
# generating random numbers for sector distribution
N <- 1e+05 # number of simulations
set.seed(1) # for reproducibility
rn.indep.gamma <- matrix(NA, N, 3, dimnames = list(1:N, c("A", "B", "C")))
for (i in 1:3) rn.indep.gamma[, i] <- rgamma(N, shape=1/sec.var[i], scale=sec.var[i])
```

Now we switch to a simulation model but with the same distributional assumptions as in the classic model.

```
CRP.pois <- init(model.type = "simulative", link.function = "CRP", N = N,
  loss.unit = 1000, random.numbers = rn.indep.gamma, seed = 1)

## Warning in init(model.type = "simulative", link.function = "CRP", N = N, :
## No LHR provided for simulative model, assuming equally likelihood for all szenarios.
## Warning in init(model.type = "simulative", link.function = "CRP", N = N, :
## loss.thr is not finite. Risk contributions (to EC, VaR and ES) will be not available.
```

Because we did not provide a vector with likelihood ratios, a corresponding warning is displayed. Similarly, we get another warning because the parameter `loss.thr` was not set

(default value: infinity). Hence no counterparty specific losses are stored, which means that risk contributions will be not available. Since in this example we only want to calculate risk figures on the overall portfolio level, we can proceed to analyze the given portfolio.

```
CRP.pois <- analyze(CRP.pois, portfolio.pois)

## Importing portfolio data....
## 3 sectors ...
## 3000 counterparties (0 removed due to EAD=0 (0), lgd=0 (0), pd<=0 (0) pd>=1 (0))
##
## Portfolio statistics....
## Loss unit: 1 K
## Portfolio EAD:1.5 B
## Portfolio potential loss:772.28 M
## Portfolio expected loss:130.69 M(analytical)
## Starting simulation (1e+05simulations )
## Simulation finished
##
## Calculating loss distribution...
## Calculating risk measures from loss distribution....
## Expected loss from loss distribution: 130.6 M
## (deviation from EL calculated from portfolio data: -0.06%)
## Exceedance Probability of the expected loss:0.45422
## Portfolio mean expected loss exceedance: 167.06 M
## Portfolio loss standard deviation:41.97 M
```

A comparison of risk figures shows that the simulation error is less than 1% in our example.

```
VaR(CRP.classic, alpha) / VaR(CRP.pois, alpha) # check if risk figures are close

## [1] 1.005308 1.008863
```

Quantifying the “Poisson effect”

Since the classic CreditRisk⁺ model assumes that counterparties’ defaults are Poisson and not Bernoulli distributed, there is a tendency to overestimated risk figures, especially for portfolios of bad quality⁸. To quantify this effect, we switch the default distribution within the portfolio data frame.

```
# Quantifying the Poisson effect
portfolio.bern <- portfolio.pois #copy portfolio
portfolio.bern$Default <- "Bernoulli" #change to Bernoulli distributed defaults
CRP.bern <- CRP.pois #duplicate model framework
CRP.bern <- analyze(CRP.bern, portfolio.bern) #analyze Bernoulli portfolio

## Importing portfolio data....
## 3 sectors ...
## 3000 counterparties (0 removed due to EAD=0 (0), lgd=0 (0), pd<=0 (0) pd>=1 (0))
##
## Portfolio statistics....
## Loss unit: 1 K
## Portfolio EAD:1.5 B
## Portfolio potential loss:772.28 M
## Portfolio expected loss:130.69 M(analytical)
## Starting simulation (1e+05simulations )
```

⁸In general, the Poisson distributions serves as a good approximation of the Bernoulli distribution only if the intensity parameter is very low.

```
## Simulation finished
##
## Calculating loss distribution...
## Calculating risk measures from loss distribution...
## Expected loss from loss distribution: 130.51 M
(Deviation from EL calculated from portfolio data: -0.13%)
## Exceedance Probability of the expected loss: 0.45437
## Portfolio mean expected loss exceedance: 166.67 M
## Portfolio loss standard deviation: 41.5 M
```

In our case, the overestimation due to the Poisson effect is around 2% - 3%.

```
VaR(CRP.pois, alpha) / VaR(CRP.bern, alpha) #compare risk figures

## [1] 1.017682 1.028880
```

Introducing sector dependencies

One of the most crucial assumptions of the classic CreditRisk⁺ model is the assumption of independent sectors. Within an analytical framework extensions to correlated sectors are proposed by Fischer and Dietz (2011) and Giese (2003). Here, we use dependent random variables (`random.numbers` matrix) to introduce dependence between sectors. Before we continue with our examples, a brief introduction to the concept of copulas is given, which will be used within the following example.

A copula is a multivariate distribution function on the d -dimensional unit hypercube with uniform one-dimensional margins. By using copulas, an arbitrary multivariate distribution can be decomposed into its one-dimensional margins and the dependence structure. Following Sklar's Theorem (see Sklar 1959) it holds that for any multivariate distribution function F on \mathbb{R}^d with univariate margins F_i a unique function $C : \times_{i=1}^d \text{Im}(F_i) \rightarrow [0, 1]$ exists, such that $F(\mathbf{x}) = C(F_1(x_1), \dots, F_d(x_d))$ for all $\mathbf{x} \in \mathbb{R}^d$. In reverse, if F_i are arbitrary univariate distribution functions and C is a copula function, then the function F defines a valid multivariate distribution function. Famous representatives of copulas are the Gaussian and the t-copula. For further details on this topic we refer to Joe (1997) and Nelson (2006).

Within our next example, the `copula` package (see Hofert, Yan, Maechler, and Kojadinovic 2014) is used in order to create an exchangeable Gaussian copula for the sector drawings. The margins are again Gamma distributed with parameters equal to the former example.

```
# Introducing sector dependencies
require("copula")
require("methods")
gauss <- normalCopula(param = 0.7, dispstr = "ex", dim = 3) # define copula
paramMargins <- list() # define margins
for (i in 1:3) paramMargins[[i]] <- list(shape = 1/sec.var[i], scale = sec.var[i])
# define multivariate sector distribution
mvdf <- mvdc(copula = gauss, margins = rep("gamma", 3), paramMargins = paramMargins)
rn.gauss.gamma <- rMvdc(N, mvdc = mvdf)
colnames(rn.gauss.gamma) <- c("A", "B", "C")
```

With the help of the new matrix `rn.gauss.gamma` we can simulate a model with dependent sectors.

```
CRP.bern.gauss <- init(model.type = "simulative", link.function = "CRP",
  N = N, loss.unit = 1000, random.numbers = rn.gauss.gamma, seed = 1)
CRP.bern.gauss <- analyze(CRP.bern.gauss, portfolio.bern)
```

As one would expect, the comparison of VaR figures shows that the risk clearly rises (by over 30% in our example) in case of dependent sectors.

```
VaR(CRP.bern.gauss, alpha) / VaR(CRP.bern, alpha) #compare risk figures

## [1] 1.330193 1.363756
```

A great advantage of the package is that one can use any arbitrary portfolio with any possible dependence structure and quantify the markup in his or her special case.

Exchanging both the sector copula and the margins

In our next example, we demonstrate how the sensitivity of risk figures with respect to distributional assumptions (i.e. sector copula and margins) can be quantified. The possibilities are only restricted by the set of distributions (univariate and multivariate) available in R. In order to increase performance, we use multiple cores (i.e. 4 cores) for the Monte Carlo simulation. Therefore, the package **parallel** is required.

```
# using a T-copula with Gamma margins
tcop <- tCopula(param = 0.7, dispstr = "ex", df = 4, dim = 3)
paramMargins <- list()
for (i in 1:3) paramMargins[[i]] <- list(shape = 1/sec.var[i], scale = sec.var[i])
mvdf <- mvdc(copula = tcop, margins = rep("gamma", 3), paramMargins = paramMargins)
rn.t.gamma <- rMvdc(N, mvdc = mvdf)
colnames(rn.t.gamma) <- c("A", "B", "C")
# initialize models and analyze portfolio
CRP.bern.t <- init(model.type = "simulative", link.function = "CRP", N = N,
  loss.unit = 1000, random.numbers = rn.t.gamma, seed = 1)
CRP.bern.t <- analyze(CRP.bern.t, portfolio.bern, Ncores = 4)
```

Again, a comparison of both models shows that specific assumptions of the sector copula may affect the risk figures. In our case, the markup is around 5% if a t-copula with 4 degrees of freedom is used instead of a Gaussian copula.

```
VaR(CRP.bern.t, alpha) / VaR(CRP.bern.gauss, alpha) #compare risk figures

## [1] 1.032988 1.057561
```

For a more detailed analysis regarding the sector copula within the CreditRisk⁺ and the CreditMetrics framework we also refer to [Fischer and Jakob \(2015\)](#). When exchanging sector distributions, please take care of the specific model assumptions, e.g. that the mean equals one within the CreditRisk⁺ framework or the quantification of the default threshold $\Phi^{-1}(\text{PD})$ in a CreditMetrics type model.

In the next step, we switch the marginal sector distributions from a Gamma distribution to a log-normal distribution.

```
# using a T-copula with logN margins
paramMargins <- list()
for (i in 1:3) paramMargins[[i]] <- list(meanlog = -0.5 * log(1 + sec.var[i]),
  sdlog = sqrt(log(1 + sec.var[i])))
mvdf <- mvdc(copula = tcop, margins = rep("lnorm", 3), paramMargins = paramMargins)
rn.t.logN <- rMvdc(N, mvdc = mvdf)
colnames(rn.t.logN) <- c("A", "B", "C")
CRP.bern.t.logN <- init(model.type = "simulative", link.function = "CRP",
  N = N, loss.unit = 1000, random.numbers = rn.t.logN, seed = 1)
CRP.bern.t.logN <- analyze(CRP.bern.t.logN, portfolio.bern, Ncores = 4)
```

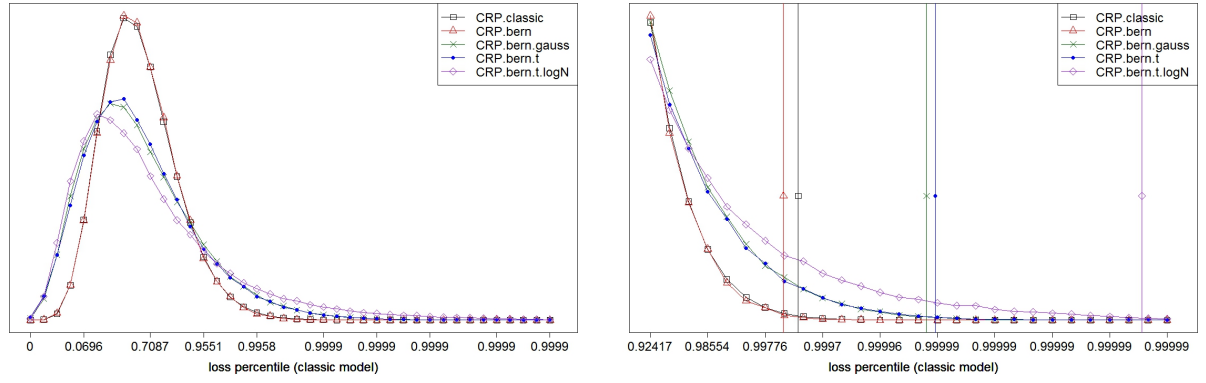


Figure 2: Loss distributions of example models together with indicators for $\text{VaR}_{0.999}$.

Since in contrast to the Gamma distribution, the log-normal distribution is heavy tailed, the values for $\text{VaR}_{0.995}$ and $\text{VaR}_{0.999}$ increase by approximately 7% and 10%, respectively.

```
VaR(CRP.bern.t.logN, alpha) / VaR(CRP.bern.t, alpha) #compare risk figures
```

```
## [1] 1.068731 1.094763
```

Please note that the same analysis can be carried out within a CreditMetrics like default model by using `link.function="CM"`.

The loss distributions of the examples are shown in Figure 2. The x-axis of both charts represent the loss percentile in the classic CreditRisk⁺ model. The right one exhibits the upper tail of all distributions together with vertical lines indicating the value of $\text{VaR}_{0.999}$ in each model, clearly demonstrating how risk increases if assumptions related to the sector distribution are modified.

5.4. Pooling

Finally, we show how a simple pooling approach can be used in order to speed up calculations. For this purpose, the package's `data` folder contains a prepared portfolio containing three pools (see Table 3). Here, all counterparties within the same sector and a potential loss ($\text{PL} = \text{EAD} \cdot \text{LGD}$) below 200,000 are grouped into one pool.

Let M_{Pool} denote the number of counterparties within one pool. Then for each pool, the values for EAD, LGD and PD are determined via the following formulas:

- $\text{EAD}_{\text{Pool}} = \frac{1}{M_{\text{Pool}}} \sum_{i \in \text{Pool}} \text{EAD}_i$ (average EAD per counterparty)
- $\text{LGD}_{\text{Pool}} = \frac{\sum_{i \in \text{Pool}} \text{EAD}_i \text{LGD}_i}{\text{EAD}_{\text{Pool}} M_{\text{Pool}}}$ (weighted average LGD per counterparty)
- $\text{PD}_{\text{Pool}} = \frac{\sum_{i \in \text{Pool}} \text{EAD}_i \text{LGD}_i \text{PD}_i}{\text{EAD}_{\text{Pool}} \text{LGD}_{\text{Pool}}}$ (average number of defaults within the pool)

Since the pooling criteria (i.e. potential loss threshold, sector membership) depend on the underlying portfolio as well as the desired accuracy, we have to leave this task up to the user. Additionally, in order to achieve good approximation results for the risk figures, advanced users may consider more sophisticated pooling techniques, for example based on certain PD and PL ranges, the pool loss standard deviation or the well-known Herfindahl index regarding the counterparty exposures as presented in Gordy (2003). Please note that in case of a CreditMetrics-like link function (i.e. if `link.function="CM"`), which includes the distribution function of a standard normal distribution, default intensities greater or equal to one are not supported.

Number	Name	Business	Country	EAD	LGD	PD	Default	A	B	C
:	:	:	:	:	:	:	:	:	:	:
100000	Pool A	misc	misc	342298,63	0,246	91,44	Poisson	1	0	0
200000	Pool B	misc	misc	332533,03	0,243	90,29	Poisson	0	1	0
300000	Pool C	misc	misc	334227,25	0,237	87,76	Poisson	0	0	1

Table 3: Structure of the pooled portfolio data frame.

With the help of this technique, we can reduce the number of portfolio positions in our example by over 50%. Using the model with a t-copula and Gamma distributed margins, we check the accuracy of risk figures if the pooled portfolio is used.

```
# Pooling
CRP.bern.t.pool <- analyze(CRP.bern.t, portfolio.pool, Ncores = 4)

## Importing portfolio data....
## 3 sectors ...
## 1483 counterparties (0 removed due to EAD=0 (0), lgd=0 (0), pd<=0 (0) pd>=1 (0))
##
## Portfolio statistics....
## Loss unit: 1 K
## Portfolio EAD:991.99 M
## Portfolio potential loss:648.54 M
## Portfolio expected loss:130.69 M(analytical)
## Starting simulation (1e+05simulations )
## Parallel computing on 4 cores (no progress bar)
## Simulation finished
##
## Calculating loss distribution...
## Calculating risk measures from loss distribution....
## Expected loss from loss distribution: 130.78 M
(deviation from EL calculated from portfolio data: 0.07%)
## Exceedance Probability of the expected loss:0.43296
## Portfolio mean expected loss exceedance: 187.4 M
## Portfolio loss standard deviation:63.37 M
```

Although the simulation of Poisson random variables is more time-consuming than those of Bernoulli ones, the simulation time (using `Ncores=1`) can be reduced by around 50% on our computer (Intel Core i7, 3.6GHz, calculation time: 7s to 3.8s). In combination with the option `Ncores=4` we can reduce the computation time by another 70% such that a simulation that needs 7s (without pooling, single core) can be done in just 1s.

```
VaR(CRP.bern.t.pool, alpha) / VaR(CRP.bern.t, alpha) #compare risk figures

## [1] 0.9974762 1.0047332
```

Comparing the risk figures of the pooled version with those of the ordinary simulation on two loss levels, we observe that the deviations are not substantial (around 1%) for our hypothetical portfolio.

Please note that the criteria and thresholds for the pooling have to be determined individually for each portfolio and model in order obtain tolerable approximation errors.

6. Summary

Quantifying credit portfolio risk is an essential part of risk controlling of financial institutions. For this purpose, the **GCPM** package offers the opportunity to choose between a CreditRisk^+

and a CreditMetrics-type model within a default framework. The examples show that, because of the flexible structure, the package helps to analyze the sensitivity of risk figures if distributional assumptions are modified and therefore to quantify aspects of model risk as well. In order to increase the performance further, simulation models can be combined with user specific importance sampling techniques and pooling approaches. The combination of these possibilities and a fast implementation of the simulation core in C++ together with the capability of parallel computing makes the package a powerful tool which also allows to perform calculations on portfolios with a large number of counterparties.

For more information about the package, especially about the individual methods, please have a look at the help pages provided in the package (e.g. `?init`).

Session Info

```
## R version 3.2.1 (2015-06-18)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 7 x64 (build 7601) Service Pack 1
##
## locale:
## [1] LC_COLLATE=German_Germany.1252 LC_CTYPE=German_Germany.1252
## [3] LC_MONETARY=German_Germany.1252 LC_NUMERIC=C
## [5] LC_TIME=German_Germany.1252
##
## attached base packages:
## [1] methods stats graphics grDevices utils datasets base
##
## other attached packages:
## [1] copula_0.999-13 GCPM_1.2 knitr_1.10.5
##
## loaded via a namespace (and not attached):
## [1] ADGofTest_0.3 Rcpp_0.11.6 lattice_0.20-31
## [4] mvtnorm_1.0-2 stabledist_0.7-0 pspline_1.0-16
## [7] grid_3.2.1 stats4_3.2.1 formatR_1.2
## [10] magrittr_1.5 evaluate_0.7 highr_0.5
## [13] stringi_0.5-2 Matrix_1.2-1 tools_3.2.1
## [16] stringr_1.0.0 RcppProgress_0.2.1 parallel_3.2.1
## [19] gsl_1.9-10
```

Acknowledgment

We would like to thank two anonymous referees for their helpful comments on an early version, which helped to improve this article.

References

- Arbenz P, Cambou M, Hofert M (2014). “An Importance Sampling Algorithm for Copula Models in Insurance.” *Submitted manuscript*. URL <http://arxiv.org/pdf/1403.4291>.
- Basel Committee on Banking Supervision (2006). “International Convergence of Capital Measurement and Capital Standards: A Revised Framework Comprehensive Version.”
- Board of Governors of the Federal Reserve System (2011). “Guidance on Model Risk Management.” *Technical report*, Federal Reserve System. URL <http://www.federalreserve.gov/bankinfo/reg/srletters/sr1107.htm>.

- Credit Suisse First Boston International (1997). “CreditRisk⁺ A Credit Risk Management Framework.” *Technical report*. URL <http://www.csfb.com/institutional/research/assets/creditrisk.pdf>.
- Crouhy M, Galai D, Mark R (2000). “A Comparative Analysis of Current Credit Risk Models.” *Journal of Banking & Finance*, **24**(1), 59–117.
- Dorflleitner G, Fischer M, Geidosch M (2012). “Specification Risk and Calibration Effects of a Multifactor Credit Portfolio Model.” *The Journal of Fixed Income*, **22**(1), 7–24.
- Eddelbuettel D, François R (2011). “**Rcpp**: Seamless R and C++ Integration.” *Journal of Statistical Software*, **40**(8), 1–18. URL <http://CRAN.R-project.org/package=Rcpp>.
- Fischer M, Dietz C (2011). “Modeling Sector Correlations with CreditRisk⁺ : The Common Background Vector model.” *The Journal of Credit Risk*, **7**(4), 23–43.
- Fischer M, Jakob K (2015). “Copula-Specific Credit Portfolio Modeling.” In K Glau, M Scherer, R Zagst (eds.), *Innovations in Quantitative Risk Management*, pp. 129–145. Springer-Verlag.
- Fischer M, Kaufmann F (2014). “An Analytic Approach to Quantify the Sensitivity of CreditRisk⁺ with Respect to its Underlying Assumptions.” *The Journal of Risk Model Validation*, **8**(2), 23–37.
- Fischer M, Mertel A (2012). “Quantifying Model Risk within a CreditRisk⁺ Framework.” *The Journal of Risk Model Validation*, **6**(1), 47–76.
- Forner K (2013). *RcppProgress: An Interruptible Progress Bar with OpenMP Support for C++ in R packages*. R package version 0.1, URL <http://CRAN.R-project.org/package=RcppProgress>.
- Giese G (2003). “Enhancing CreditRisk⁺.” *RISK*, **16**(4), 73–77.
- Glasserman P, Li J (2005). “Importance Sampling for Portfolio Credit Risk.” *Management Science*, **51**(11), 1643–1656.
- Gordy MB (2000). “A Comparative Anatomy of Credit Risk Models.” *Journal of Banking & Finance*, **24**(1), 119–149.
- Gordy MB (2003). “A Risk-Factor Model Foundation for Ratings-Based Bank Capital Rules.” *Journal of Financial Intermediation*, **12**(3), 199–232.
- Gundlach VM (2003). “Basics of CreditRisk⁺.” In M Gundlach, F Lehrbass (eds.), *CreditRisk⁺ in the Banking Industry*, pp. 7–24. Springer-Verlag.
- Gupton GM, Finger CC, Bhatia M (1997). “CreditMetrics: Technical Document.” *Technical report*. URL http://www.defaultrisk.com/_pdf6j4/creditmetrics_techdoc.pdf.
- Haaf H, Reiss O, Schoenmakers J (2003). “Numerically Stable Computation of CreditRisk⁺.” In M Gundlach, F Lehrbass (eds.), *CreditRisk⁺ in the Banking Industry*, pp. 69–77. Springer-Verlag.
- Haaf H, Tasche D (2002). “Credit Portfolio Measurements.” *GARP Risk Review*, **7**, 43–47.
- Hamerle A, Rösch D (2006). “Parameterizing Credit Risk Models.” *Journal of Credit Risk*, **2**(4), 101–122.
- Hofert M, Yan J, Maechler M, Kojadinovic I (2014). *copula: Multivariate Dependence with Copulas*. R package version 0.999-12, URL <http://CRAN.R-project.org/package=copula>.

- Jakob K, Fischer M (2014). “Quantifying the Impact of Different Copulas in a Generalized CreditRisk+ Framework: An Empirical Study.” *Dependence Modeling*, **2**(1), 1–21.
- Joe H (1997). *Multivariate Models and Dependence Concepts*. Chapman & Hall/CRC.
- Mai JF, Scherer M (2012). *Simulating Copulas*. Imperial College Press.
- Merton RC (1974). “On the Pricing of Corporate Debt: The Risk Structure of Interest Rates.” *The Journal of Finance*, **29**(2), 449–470.
- Nelson RB (2006). *An Introduction to Copulas*. Springer-Verlag.
- Pfaff B, McNeil A (2014). *QRM: Provides R-language Code to Examine Quantitative Risk Management Concepts*. R package version 0.4-10, URL <http://CRAN.R-project.org/package=QRM>.
- R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>.
- Rubino G, Tuffin B (2009). *Rare Event Simulation Using Monte Carlo Methods*. John Wiley & Sons.
- Sklar A (1959). “Fonctions de Répartition à n Dimensions et leurs Marges.” *Publications de l’Institut de Statistique de l’Université de Paris*, **8**, 229–231.
- Wittmann A (2007). *CreditMetrics: Functions for Calculating the CreditMetrics Risk Model*. R package version 0.0-2, URL <http://CRAN.R-project.org/package=CreditMetrics>.

Affiliation:

Kevin Jakob
 Universität Augsburg
 86135 Augsburg, Germany
 E-mail: Kevin.Jakob@Student.Uni-Augsburg.de

Matthias Fischer
 Department of Statistics and Econometric
 Universität Erlangen-Nürnberg
 91054 Nürnberg, Germany
 E-mail: Matthias.Fischer@fau.de
 URL: <http://www.statistik.wiso.uni-erlangen.de/>

Geovisualisation: Possibilities with R

Jan-Philipp Kolb
GESIS

Abstract

The processing of information related to a geographic location has long been difficult due to the lack of (pertinent) data sources and computational power. However, the recent developments of web-based technologies like OpenStreetMap (OSM) and Google Maps change this fundamentally. With R it is possible to process a large amount of data and produce appropriate visualisations. The challenge is to find the necessary spatial information, like appropriate polygons and data corresponding to these polygons. In this paper ways are presented to access this information via internet and to combine and visualise these information.

Keywords: spatial visualisation, choropleths, polygons, geocoding, R.

1. Introduction

Geographical visualisation facilitate the understanding of social phenomena. Often we compare the information depicted on the map with the perception of the real situation in our working or living environment. To enforce this process it is good to have a map which is as accurate as possible. In the past, it was often complicated to produce highly detailed maps with publicly available data (e.g. scientific or public use files). Mostly, complications were related with the disclosure control of data that included personal characteristics.

In the past few years, a massive amount of information has been uploaded on the internet and its volume is still growing. This information has an ubiquitous nature. In the course of this development, much information related to geographic dispositions has been published.

Primarily, this was caused by the introduction of new web based services and collaborative mapping. For the past 10 years, OpenStreetMap has offered crowd-sourced information (see [Haklay 2010](#) and [Neis, Zielstra, and Zipf 2011](#)) and in February 2005 Google Maps was introduced, which offers a fast-loading, tiled map display and a deep user interface (c.f. [Gibson and Erle 2006](#)).

As a result, a huge amount of spatial information is now freely accessible, for example through application programming interfaces (APIs). Thus, there is a big analytic potential but the information is often unstructured or only semi-structured (e.g. web documents, news archives). In addition, this information is often very heterogeneous and not intended for geographic purposes but contains geographic information implicitly (Web 2.0). Often there is little or no metadata available.

In the following sections, examples will be presented which combine polygons and the data available from APIs to produce choropleth maps. Additionally, simple methods will be described to combine this geodata. To do so it is initially necessary to find appropriate sources for polygons. These are presented in the next section, that will explain how to access spatial information available on the internet.

Recently, numerous R packages have been published, that allow data processing and visualisation of geographical information. A short overview of the scope of these packages and the opportunities R offers for spatial information is part of Section 3. In Section 4, possibilities to visualise this information will be described. Hence, two examples of application will be used to highlight the potential of information gained from OSM for visualising spatial data. The examples will be presented in the form of choropleth maps. The paper will conclude with a closing discussion in Section 5.

2. Information access

A choropleth map shows distributions by area (Pitzl 2004). These kind of maps are often used to visualise spatial information in the social sciences. The information is then often linked to an administrative entity. Administrative entities are organized in an hierarchical manner. To create a choropleth map it has to be clarified on which level the displayed information is available. The easiest example are values related to different countries. In this example the most simple way to visualise spatial information is to use the R package **maps** (Becker, Wilks, Brownrigg, and Minka 2013) which also contains country data. Alternatively it is possible to use the R package **choroplethr** (Lamstein and Johnson 2015) or the new **tmap** package (Tennekes 2015) especially for creating thematic maps such as choropleths. Both packages contain polygons on country level.

If the information is available below the country level, it is advisable to use the R package **maptools** (Lewin-Koh, Bivand, Pebesma, Archer, Baddeley, Bibiko, Dray, Forrest, Friendly, Giraudoux *et al.* 2011). Here, the information of areas is organised in polygons (Leipzig and Li 2011). If the administrative entity is not implemented in R packages, it is necessary to download the information from an external source. For example, a world borders dataset is available on thematicmapping.org. To import shapefiles in R the package **rgdal** can be used. It has the advantage, that it can handle projection information (Keitt, Bivand, Pebesma, and Rowlingson 2011).

One of the most comprehensive sources is the Global Administrative Areas database (GADM, www.gadm.org), which also provides downloadable information in the format of RData-files which contain objects of class **SpatialPolygonsDataFrame**. The package **raster** provides the function `getData()` with which it is possible to download the data automatically.

The data can also be downloaded with the following commands:

```
con <- url("http://biogeo.ucdavis.edu/data/gadm2/R/DEU_adm3.RData")
print(load(con))
close(con)
```

For the example above, polygons for NUTS3 level¹ in Germany have been downloaded.² This is the most detailed information published in the GADM database and includes smaller regions and large cities. For other countries only information on NUTS1 or 2 level is available.

The availability of data necessary to visualise information that is more detailed than the NUTS3 level differs highly by country. Germany, for example, provides information on communities at geodatenzentrum.de.³ Every country has its specific sources of information. The

¹The Nomenclature of Territorial Units for Statistics (NUTS) is a standard for referencing the subdivisions of countries, see Eurostat (1995) for more details.

²For more information on how to work with shapefiles see Kennedy (2013).

³www.geodatenzentrum.de/geodaten/gdz_rahmen.gdz_div?gdz_spr=deu&gdz_akt_zeile=5&gdz_anz_zeile=1&gdz_unt_zeile=14&gdz_user_id=0

US Census Bureau offers lots of information in the TIGER/Line program⁴ (see for example [Almquist 2010](#), p. 2) and this information can easily be linked with other data available from the US Census Bureau.

The number of publications which refer to this data source is an indication of how promising the program of publishing spatial information is. In Europe, the establishment of the infrastructure for spatial information (INSPIRE - website <http://inspire.ec.europa.eu>) accelerated the exchange about this topic.

Beside the possibility to plot choropleth maps for administrative areas, other areas such as electoral districts or the zip-code might be of interest as well. In Germany the *Bundesnetzagentur* offers a dataset of the prefix zones.⁵

But, as already stated, very detailed information is often not available due to reasons of disclosure control. The user has to decide on the degree of refinement necessary for her/his application. Sometimes, information is not present as data that corresponds to an administrative entity, but as address information. Then, an additional step can be interposed which is the geocoding of the information. In this situation the package **ggmap** is very useful ([Kahle and Wickham 2013](#)). The package can for example be used to geocode (function `geocode()`) points of interests (POI). It is possible to get latitude and longitude referenced to the World Geodetic System 1984 (WGS84) ellipsoid ([Lovelace and Cheshire 2014](#), pp. 7). The query `mapdist()` provides results on the distance between two points of interest. In both cases, it is necessary to give an exact address of the location. The number of requests is limited per day and you have to pay attention to data security issues. The problem is not that important when institutions are geocoded, but might be more considerable for interviewees, especially if the questionnaire content is delicate.

Some other packages are available for geocoding, like the **geocodeHERE** package, which is a Wrapper for Nokia's HERE geocoding API ([Nissen 2014](#)). There are some possibilities to download related information from crowdsourced services like wikipedia.org. This information can be accessed using for example the **geonames** package (see for example [Ceolin, Moreau, O'Hara, Schreiber, Sackley, Fokkink, van Hage, and Shadbolt 2013](#) or [Van Hage, Van Erp, and Malaisé 2012](#)). Other packages like the development version of the R package **tmap**⁶ are using the OSM-service Nominatim ([Warden 2011](#), p. 25).

In Germany, the block sides are often discussed as the smallest administrative area for which it would be possible to publish data. A block side is an area with the same street name which is restricted by intersections or similar geographical limitations. In fact, buildings are the smallest entities for which polygons might be used. The necessary information can be accessed via the package **osmar** ([Eugster and Schlesinger 2013](#)). In combination with the R package **sp** it is possible to transform this information into classes for points, lines, and polygons. With the **osmar** package it is possible to get information from user-generated street maps. The package uses the API provided by OpenStreetMap⁷ (see for example [Haklay and Weber 2008](#)). With this API it is possible to access the offered map data that is free to use, editable, and licensed under new copyright regulations.⁸

The package **osmar** is designed to get raw OSM data. The usage of the package is described in [Schlesinger \(2011\)](#). There are three important steps in order to download information from OpenStreetMap via **osmar**.

- First, you have to provide the information about the API of OpenStreetMap.
- Second, the bounding box must be defined.

⁴<http://census.gov/geo/maps-data/data/tiger-line.html>

⁵http://bundesnetzagentur.de/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/Nummerierung/Rufnummern/ONVerzeichnisse/GISDaten_ONBGrenzen/ONBGrenzen_Basepage.html

⁶A stable development version can be installed with: `devtools::install_github("mtennekes/tmap/pkg", ref = "45855fa")`.

⁷<http://api.openstreetmap.org/api/0.6/>

⁸http://rstudio-pubs-static.s3.amazonaws.com/12696_9fd49fb7055c40ff9b3a3ea740e13ab3.html

- and the third step is the download of the information.

The bounding box can be defined with `center_bbox()`, one has to specify the latitude, longitude and the size of the box as arguments. The boxsize is rendered in metres. In the example below, we have a box of 500×500 meters. The download time and the object size depend very much on the size of the chosen bounding box and of course on the number of objects in the bounding box.

```
library("ggmap")
library("osmar")
src <- osmsource_api()
Ma_Schloss <- geocode("Mannheim Schloss")
bb_MA_S <- center_bbox(Ma_Schloss$lon, Ma_Schloss$lat, 800, 800)
ua_MA_S <- get_osm(bb_MA_S, source = src)
```

However, the information downloadable by **osmar** is limited by boxsize. As a maximum an area of 0.25 square degrees can be queried (Eugster and Schlesinger 2013, p. 2). Another interesting option which is available for R is the usage of some services that offer the download of data extracts from OpenStreetMap. One of these services is geofabrik.de, where general information is available for free and specialised requests require a fee.

With OpenStreetMap it is possible to combine geographic information and social web contents. This is possible because more and more objects are geotagged (see for instance Scharl and Tochtermann 2009). A good example is the website flickr.com where many photos are geotagged. That means that the photos are associated with latitude and longitude and the user generated content can therefore be used for social analysis (Yee and Moodle 2008, p. 245). In this case, information is used that was not originally created for this purpose. You, DesArmo, and Joo (2013) mine for example user-generated text on Flickr to measure the happiness of US citizens. Sizov (2010) uses Flickr data to evaluate an algorithm for content management, retrieval and sharing.

The R package **twitterR** enables the access to geographic information from the twitter.com API (Gentry 2015). Kaczmirek, Mayr, Vatrapsu, Bleier, Blumenberg, Gummer, Hussain, Kinder-Kurlanda, Manshaei, Thamm *et al.* (2013) show how to use Twitter data for social and political research. They monitor the campaigns for the 2013 German Bundestag elections in social media.

Also data from OpenStreetMap can be a valuable source of information. With this data it is for example possible to analyse the number of services available for children in one zip-code area. Furthermore it is possible to compute the floor area of the buildings.

3. Information processing

Numerous R packages are available to process the spatial information gained from the sources described above. A good overview of these packages is available at the CRAN Task View about Analysis of Spatial Data from Roger Bivand.⁹ The book of Bivand, Pebesma, and Gómez-Rubio (2013) can also be recommended in this coherence. An overview of the implementation of spatial data analysis software tools in R is available in Bivand (2006). Most of the important packages are listed there, some will be described in the following.

The **maptools** package was already mentioned as a source for polygons. But this package is also very useful for the information processing of spatial data (Bivand 2011, p. 18). The **maptools** package has been adapted to use **sp** classes, and in combination with the **sp** package (Pebesma, Bivand, Rowlingson, and Gomez-Rubio 2013), it provides a good set of tools to process spatial data.

If data sets from different sources are combined, it is important to ensure that the same map projection is used. If that is not the case, transformation can be done for example with the

⁹<http://cran.r-project.org/web/views/Spatial.html>

function `spTransform` from package `sp`. See (Pebesma 2012, pp. 20) or (Rossiter 2012, pp. 9) for examples.

These and other packages enable the usage of the R language as a geographic information system (GIS). Here the package `rgeos` is very useful (Bivand and Rundel 2013) for basic topology operations.

`rgdal` for example can be used to bridge information from the geospatial data abstraction library (GDAL, www.gdal.org) to R (Keitt *et al.* 2011). To import shapefiles, `rgdal` has the advantage that it takes care of the projection, which is not the case for `maptools`. Alternatively the `read_shape`-command from `tmap` can be used.

The `raster` package is for reading, writing, manipulating, analysing and modeling of gridded spatial data (Hijmans and van Etten 2014). The package `rgrass7` provides an interface between the geographic information system GRASS 7.0 and R (Bivand 2015). The `rgeos` package is also useful to edit polygons.

In case of the example depicted in this article, the shapefiles downloaded for example from geodatenzentrum.de or other sources can be bridged to R using the `readOGR` command from the `rgdal` package.

```
library("rgdal")
zip <- readOGR(".", "post_pl")
```

It is then possible to plot and edit the polygons with the `spplot`-function from the `sp` package (Bivand *et al.* 2013, pp. 73). The data downloaded from OpenStreetMap via the package `osmar` has to be edited (for example with the `sp` package) to get such polygons. First steps to the processing of the information from OpenStreetMap are described in (Eugster and Schlesinger 2013). To convert the OpenStreetMap information to polygons, the following commands can be used:

```
ua_ids <- find(ua_MA_S, way(tags(k == "building")))
ua_ids <- find_down(ua_MA_S, way(ua_ids))
bg <- subset(ua_MA_S, ids = ua_ids)
bg_erg <- as_sp(bg, "polygons")
```

A wrapper around these functions is provided in the development version of the `tmap` package:

```
bb_schloss <- bb(q="Mannheim Schloss")
sp_schloss <- read_osm(bb_schloss, building = osm_poly("building"),
                     castle = osm_poly("historic=castle"))
qtm(sp_schloss$building, fill = "ivory", borders = "snow4") +
  qtm(sp_schloss$castle, fill = "royalblue")
```

As already described, it is only possible to download the data set for a small area. To get complete information about areas on NUTS3-level, many queries have to be realised. It is therefore advisable to choose a top-down approach. In the present case, this implies the download of information from geofabrik.de and the division into smaller entities. The information is not provided for entities more detailed than the administrative district. In the example it is the administrative district Karlsruhe. It is not advisable to plot all the information in one map. Hence, the available information must be processed. Due to the limited scope of this article the procedure will not be described. For details on this procedure please refer to the GitHub Repository <http://github.com/Japhilko/GeoData>.

4. Visualisation of information

Much potential is available to visualise information related to spatial data. Every type of geographical presentation is feasible in R if the conditions are satisfied. The most important condition is the availability of information necessary for the desired visualisation. In some research fields it is easier to get the data, in others it is more complicated. Topographic maps

for example can be visualised with **GEOmap** (Lees 2008). In the following, visualisations for social science data are addressed.

The **sp** package offers many fascinating possibilities to visualise spatial data (e.g. Endel and Filzmoser 2012). One type of spatial visualisations that is of particular interest in the social sciences are choropleth maps. This type of maps can be used to display area values. Often values are depicted on NUTS3 level and upwards. But, the finer the scale, the more interesting the map can be. The difficulty to access such detailed information makes the combination of choropleth maps and OSM data so interesting. In the given example, the information from OpenStreetMap is used to produce a choropleth map. Polygons of semi-administrative areas (zip-code areas) are connected with the aggregated information from OpenStreetMap.

As already described, polygons for entities below the community level are available for download using the **osmar** package. The R code for this minimal working example is presented in Appendix A. As a result, we get the buildings of the city of Mannheim with its castle highlighted in blue.

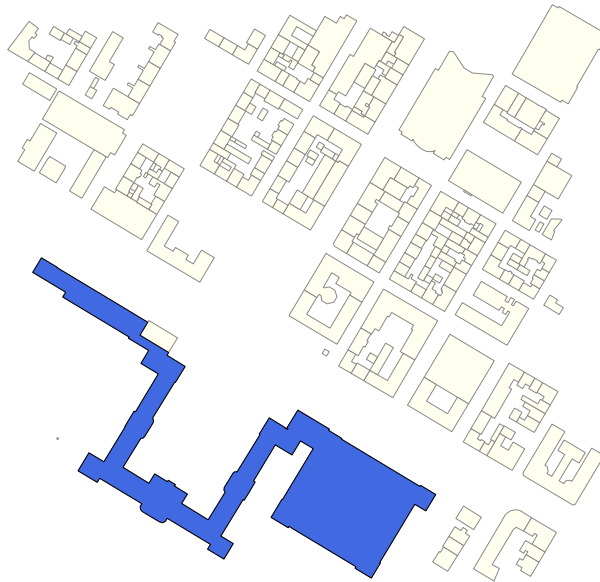


Figure 1: Buildings around the castle of Mannheim.

In a next step the user may want to combine the information gained from web-services like OpenStreetMap with polygons. In Figure 2 this is done for schools.

The OSM community did an amazing job in collecting all the information. But there are of course still white spots in regions, where not so many OSM members are active or the public authorities do not provide necessary information. In addition, one has to keep in mind that the situation is changing very fast. In Figure 1 for example the district court is missing, which is located across the castle. Haklay (2010) provides a list of criteria to evaluate geographical information.

The information downloaded from OpenStreetMap can of course also be used for other purposes than only visualising a social coherence. For example Behrisch, Bieker, Erdmann, and Krajzewicz (2011) show how to use the information for a geographic simulation model. Lovelace and Cheshire (2014) give an introduction to visualise spatial data in R.

Without a question, the analysis of geodata and context-related information offers a great potential for scientific purposes. However, the publication of geodata might be in conflict with data security. In the present case, this is a minor problem because it is hardly possible to extract any sensitive information from this choropleth map.

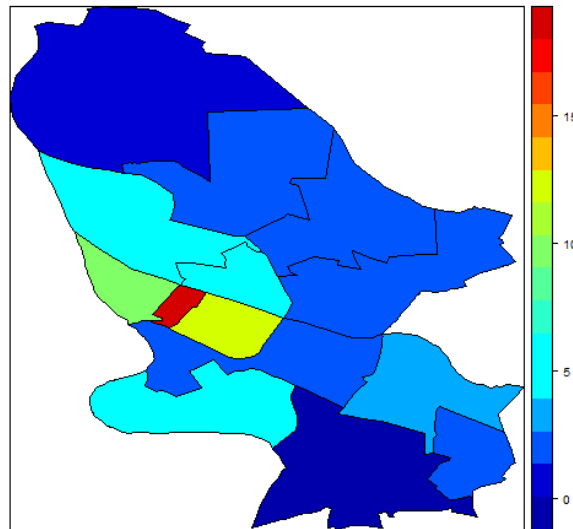


Figure 2: Number of schools per zip-code area in Mannheim.

5. Conclusion

The location of a POI is of special interest for the social sciences. But in practice, the sample size is often too small to create choropleth maps on NUTS1 level with survey-information, while hardly any other official dataset with geolocated information was available for researchers. That changed due to crowd-sourced information which is nowadays available. It is possible to access this information by using the so called API. Many services, like Google Maps, OpenStreetMap, Flickr, Twitter or Wikipedia do provide such API and the R environment offers many possibilities to use them.

Shapefiles are available for administrative areas but the level differs widely across the countries. Self-created polygons for non-administrative areas can be created using the R package **osmar**. This package uses information from OpenStreetMap which is a prime example for volunteered geographic information.

In general, one can say that the type of entities depends very much on the area of research. The challenge is to find and visualise context-related information. The next challenge is to show this information in reasonable visualisations. In this situation, the advantages of R can be used to shed light into the dark.

A. Transfer and plot data from OpenStreetMap

```
library("osmar")
library("ggmap")

src <- osmsource_api()
Ma_Schloss <- geocode("Mannheim Schloss")
bb_MA_S <- center_bbox(Ma_Schloss$lon, Ma_Schloss$lat, 800, 800)
ua_MA_S <- get_osm(bb_MA_S, source = src)

# filter buildings and convert to sp object
ua_ids <- find(ua_MA_S, way(tags(k == "building")))
ua_ids2 <- find_down(ua_MA_S, way(ua_ids))
bg <- subset(ua_MA_S, ids = ua_ids2)
bg_sp <- as_sp(bg, "polygons")

id <- ua_MA_S$ways$tags$id
vs <- ua_MA_S$ways$tags$v
```

```
id_s <- id[grepl("Schloss Mannheim", vs)]
plot(bg_sp, col = "ivory", border = "snow4")
plot(subset(bg_sp, id %in% id_s), col="royalblue", add = TRUE)
```

B. Subset POI from Geofabrik extracts

```
# Download postal code polygons from
# http://datahub.io/de/dataset/postal-codes-de

library("sp")
library("rgdal")
PLZ <- readOGR(".", "post_pl")

# Download OSM information from geofabrik.de
points <- readOGR(".", "points")

# Filter data sets
MA <- subset(PLZ, PLZORT99 == "Mannheim")
MA$PLZ99 <- droplevels(MA$PLZ99)
schools <- subset(points, type == "school")
proj4string(MA) <- proj4string(points)

# Get number of schools per ZIP-code area
tmp <- over(schools, MA)
school_freq <- data.frame(freq = tapply(tmp$PLZ99, tmp$PLZ99, length))
school_freq[is.na(school_freq)] <- 0
MA@data <- merge(MA@data, school_freq, by.x = "PLZ99", by.y = 0, all = TRUE)
spplot(MA, "freq")
```

The information from Geofabrik can be downloaded using the following link:

<http://download.geofabrik.de/europe/germany/baden-wuerttemberg-latest.shp.zip>

References

- Almquist ZW (2010). "US Census Spatial and Demographic Data in R: The **UScensus2000** Suite of Packages." *Journal of Statistical Software*, **37**, 1–31.
- Becker RA, Wilks AR, Brownrigg R, Minka TP (2013). *maps: Draw Geographical Maps*. URL <http://CRAN.R-project.org/package=maps>.
- Behrisch M, Bieker L, Erdmann J, Krajzewicz D (2011). "SUMO – Simulation of Urban Mobility – An Overview." In *SIMUL 2011, The Third International Conference on Advances in System Simulation*, pp. 55–60.
- Bivand R (2011). "Geocomputation and Open Source Software: Components and Software Stacks." *NHH Dept. of Economics Discussion Paper*, **23**. doi:10.2139/ssrn.1972280.
- Bivand R (2015). *rgrass7: Interface Between GRASS 7 Geographical Information System and R*. R package version 0.1-2, URL <http://CRAN.R-project.org/package=rgrass7>.
- Bivand R, Rundel C (2013). "**rgeos**: Interface to Geometry Engine - Open Source (GEOS)." *R package version 0.3-3*. URL <http://cran.r-project.org/web/packages/rgeos/index.html>.
- Bivand RS (2006). "Implementing Spatial Data Analysis Software Tools in R." *Geographical Analysis*, **38**(1), 23–40.

- Bivand RS, Pebesma EJ, Gómez-Rubio V (2013). *Applied Spatial Data Analysis with R*. Second edition. Springer, New York. ISBN 978-1-4614-7617-7.
- Ceolin D, Moreau L, O'Hara K, Schreiber G, Sackley A, Fokkink W, van Hage WR, Shadbolt N (2013). *Reliability Analyses of Open Government Data*. Proceedings of the 9th International Workshop on Uncertainty Reasoning for the Semantic Web. URL <http://eprints.soton.ac.uk/357160/1/paper6.pdf>.
- Endel F, Filzmoser P (2012). "R & GIS: Geospatial Plotting." In *Mathematical Modelling*, volume 7, pp. 618–623.
- Eugster MJ, Schlesinger T (2013). "osmar: OpenStreetMap and R." *The R Journal*, **5**(1), 53–63.
- Eurostat (1995). "Nomenclature of Territorial Units for Statistics." URL <http://ec.europa.eu/eurostat/web/nuts/>.
- Gentry J (2015). *twitterR: R Based Twitter Client*. URL <http://CRAN.R-project.org/package=twitterR>.
- Gibson R, Erle S (2006). *Google Maps Hacks*. O'Reilly Media, Inc. ISBN 978-0-596-10161-9.
- Haklay M (2010). "How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets." *Environment and Planning B, Planning & Design*, **37**(4), 682–703.
- Haklay M, Weber P (2008). "OpenStreetMap: User-Generated Street Maps." *IEEE Pervasive Computing*, **7**(4), 12–18.
- Hijmans RJ, van Etten J (2014). *raster: Geographic Data Analysis and Modeling*. URL <http://CRAN.R-project.org/package=raster>.
- Kaczmirek L, Mayr P, Vatraru R, Bleier A, Blumenberg M, Gummer T, Hussain A, Kinder-Kurlanda K, Manshaei K, Thamm M, *et al.* (2013). "Social media monitoring of the campaigns for the 2013 german bundestag elections on facebook and twitter." *pre-print*. URL <http://arxiv.org/abs/1312.4476>.
- Kahle D, Wickham H (2013). "ggmap: Spatial Visualization with ggplot2." *R Journal*, **5**(1).
- Keitt TH, Bivand R, Pebesma E, Rowlingson B (2011). *rgdal: bindings for the Geospatial Data Abstraction Library*. URL <http://CRAN.R-project.org/package=rgdal>.
- Kennedy MD (2013). *Introducing Geographic Information Systems with ArcGIS: A Workbook Approach to Learning GIS*. John Wiley & Sons. ISBN 978-1-118-15980-4.
- Lamstein A, Johnson BP (2015). *choroplethr: Simplify the Creation of Choropleth Maps in R*. R package version 3.1.0, URL <http://CRAN.R-project.org/package=choroplethr>.
- Lees J (2008). *GEOmap: Topographic and Geologic Mapping*. URL <http://CRAN.R-project.org/package=GEOmap>.
- Leipzig J, Li XY (2011). *Data Mashups in R*. O'Reilly Media, Inc. ISBN 978-0-596-55964-9.
- Lewin-Koh NJ, Bivand R, Pebesma E, Archer E, Baddeley A, Bibiko H, Dray S, Forrest D, Friendly M, Giraudoux P, *et al.* (2011). *maptools: Tools for Reading and Handling Spatial Objects*. URL <http://CRAN.R-project.org/package=maptools>.
- Lovelace R, Cheshire J (2014). "Introduction to Visualising Spatial Data in R." *Technical report*, National Centre for Research Methods. URL <http://eprints.ncrm.ac.uk/3295/>.

- Neis P, Zielstra D, Zipf A (2011). “The Street Network Evolution of Crowdsourced Maps: OpenStreetMap in Germany 2007–2011.” *Future Internet*, **4**(1), 1–21.
- Nissen C (2014). *geocodeHERE: Wrapper for Nokia’s HERE Geocoding API*. URL <http://CRAN.R-project.org/package=geocodeHERE>.
- Pebesma E (2012). “**spacetime**: Spatio-temporal Data in R.” *Journal of Statistical Software*, **51**(7), 1–30.
- Pebesma E, Bivand R, Rowlingson B, Gomez-Rubio V (2013). *sp: Classes and Methods for Spatial Data*. URL <http://CRAN.R-project.org/package=sp>.
- Pitzl GR (2004). *Encyclopedia of Human Geography*. Greenwood Publishing Group. ISBN 978-0313320101.
- Rossiter D (2012). *Applied Geostatistics Exercise 9: R and GIS*. URL <http://tw.rpi.edu/media/latest/ex9.pdf>.
- Scharl A, Tochtermann K (2009). *The Geospatial Web: How Geobrowsers, Social Software and the Web 2.0 are Shaping the Network Society*. Springer, London. ISBN 1-84628-826-6.
- Schlesinger T (2011). “OpenStreetMap in R - Freie Räumliche Daten für geostatistische Analysen.” Thesis (Bachelor). URL <http://epub.ub.uni-muenchen.de/12463/>.
- Sizov S (2010). “GeoFolk: Latent Spatial Semantics in Web 2.0 Social Media.” In *Proceedings of the third ACM international conference on Web search and data mining*, pp. 281–290. ACM.
- Tennekes M (2015). *tmap: Thematic Maps*. R package version 1.0, URL <http://CRAN.R-project.org/package=tmap>.
- Van Hage WR, Van Erp M, Malaisé V (2012). “Linked Open Piracy: A Story about e-Science, Linked Data, and Statistics.” *Journal on Data Semantics*, **1**(3), 187–201.
- Warden P (2011). *Data Source Handbook*. O’Reilly Media, Inc. ISBN 978-1-4493-0314-3.
- Yee R, Moodle M (2008). *Pro Web 2.0 Mashups*. Apress, Berkeley, CA; New York. ISBN 978-1590598580.
- You S, DesArmo J, Joo S (2013). “Measuring happiness of US cities by mining user-generated text in Flickr.com: A pilot analysis.” *Proceedings of the American Society for Information Science and Technology*, **50**(1), 1–4.

Affiliation:

Jan-Philipp Kolb
 GESIS - Leibniz Institute for the Social Sciences
 Survey Design and Methodology
 68072 Mannheim
 E-mail: Jan-Philipp.Kolb@gesis.org
 URL: www.gesis.org

Austrian Journal of Statistics
 published by the Austrian Society of Statistics
 Volume 45
 March 2016

<http://www.ajs.or.at/>
<http://www.osg.or.at/>
 Submitted: 2014-10-31
 Accepted: 2015-08-26

Simulation Tools for Small Area Estimation: Introducing the R Package **saeSim**

Sebastian Warnholz
Freie Universität Berlin

Timo Schmid
Freie Universität Berlin

Abstract

The demand for reliable regional estimates from sample surveys has substantially grown over the last decades. Small area estimation provides statistical methods to produce reliable predictions when the sample sizes in specific regions are too small to apply direct estimators. Model- and design-based simulations are used to gain insights into the quality of the methods utilized. In this article we present a framework which may help to support the reproducibility of simulation studies in articles and during research. The R package **saeSim** is adjusted to provide a simulation environment for the special case of small area estimation. The package may allow the prospective researcher during the research process to produce simulation studies with minimal coding effort.

Keywords: Package, R, reproducible research, simulation study, small area estimation.

1. Introduction

The demand for reliable small area statistics from sample surveys has substantially grown over the last decades due to their use in public and private sectors. In this paper we present a framework for simulation studies within the field of small area estimation. This tool might be useful for the prospective researcher or data analyst to provide reproducible research.

Reproducible research has become a widely discussed topic. In the field of statistics many open source tools like the R language (R Core Team 2014) and L^AT_EX, dynamic reporting packages like **knitr** (Yihui 2013), **Sweave** (Leisch 2002) and more recently **rmarkdown** (Allaire, McPherson, Xie, Wickham, Cheng, and Allen 2014), make the integration of text and source code for statistical analysis possible. Publishing source code and data alongside research results draws special attention to authoring the analysis. However, the requirements for source code are different from the written words in the article itself.

Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do. (Knuth 1992, p.99)

Besides the combination of text and source code, reproducible research aims at the availability of the full academic research, which is the paper combined with the full computational

environment, like data and source code. However, real data are often very sensitive and governed by strict confidentiality rules. Synthetic data generation mechanisms as discussed in [Alfons, Kraft, Templ, and Filzmoser \(2011\)](#) or [Kolb \(2013\)](#) can be used to provide safe data which are publicly available to enable the community to reproduce the analysis and results. [Burgard, Kolb, and Münnich \(2014\)](#) interpreted this as an open research philosophy. Such synthetic data sets can be used to test newly proposed statistical methods in a close-to-reality framework. In general, simulation studies in statistics can be divided into two concepts:

- Design-based: The simulation study is based on true or synthetic data of a fixed population. Then, samples are selected repeatedly from the underlying finite population and different estimation methods are applied in each replication. The estimates so obtained are compared to the true values of the population, for instance, in terms of relative bias (RB) or relative root mean squared error (RRMSE).
- Model-based: The simulation study uses data drawn from certain distributions. In each iteration, the population is generated from a model and a sample is selected according to a specific sampling scheme. The sample is used to estimate the quantity of interest for which quality measures (like RB and RRMSE) are derived.

Further discussion regarding model- and design-based simulations is available in [Münnich, Schürle, Bihler, Boonstra, Knotterus, Nieuwenbroek, Haslinger, Laaksonen, Eckmair, Quatember, Wagner, Renfer, Oetliker, and Wiegert \(2003\)](#), [Salvati, Chandra, Giovanna-Ranalli, and Chambers \(2010\)](#) or [Alfons, Templ, and Filzmoser \(2010\)](#).

[Alfons et al. \(2010\)](#) provide the R package **simFrame** which helps to conduct simulation studies in a reproducible environment. It includes a wide range of features (like data generation, sampling schemes, outlier contamination mechanisms and missing values) to conduct simulation studies. **simFrame** was originally developed for simulations in the context of survey statistics but is now designed to be as general as possible (cf. [Alfons et al. 2010](#)). Furthermore the package **simPop** ([Meindl, Templ, Alfons, Kowarik, and with contributions from Ribatet M 2014](#)) supports the generation of synthetic population data. This can be a suitable environment in scenarios where the reproducibility of results and confidentiality issues play an important role.

Survey statistics are used, for example, in order to deliver specific indicators as a basis for economic and political decision processes. Of special interest here are regional or group specific comparisons (cf. [Schmid and Münnich 2014](#)). Surveys which are utilized to report these regional indicators, however, are generally designed for larger areas. Hence sample information on more detailed levels, e.g. municipalities, is hardly available so that classical estimation methods (direct estimators) may lead to high variances of the estimates (cf. [Ghosh and Rao 1994](#)). In this case small area estimation methods may reveal highly improved results for the target estimates. Small area estimation has become more and more attractive over the last decade:

In 2002, small area estimation (SAE) was flourishing both in research and applications, but my own feeling then was that the topic has been more or less exhausted in terms of research and that it will just turn into a routine application in sample survey practice. As the past 9 years show, I was completely wrong; not only is the research in this area accelerating, but it now involves some of the best known statisticians... Pfeffermann (2013)

However, simulation studies in the context of small area estimation are often presented very briefly. Thus there is a need to have a suitable framework to guarantee the reproducibility of analysis. To the best of our knowledge, there is no R package or framework adjusted for the special case of small area estimation which provides a simulation environment.

The aim of this article is to introduce a new R package, **saeSim**, which supports the process of making simulation studies in the field of small area estimation reproducible. To be more precise, the suggested package has three main objectives: First, to provide tools for data generation. Second, to unify the process of simulation studies. Third, to make the source code of simulation studies available, such that it supports the conducted research in a transparent manner.

This paper is organised as follows. In Section 2 we give a short introduction to small area estimation focusing mainly on unit-level (Battese, Harter, and Fuller 1988) and area-level models (Fay and Herriot 1979). Section 3 introduces a framework for simulation studies and how it is supported by the R package **saeSim**. To illustrate some of the features of the package we present a model- and design-based simulation study in Section 4. We conclude the paper in Section 5 by summarising the main findings and by providing some avenues for further research.

2. Small area estimation

The aim of small area estimation is to produce reliable statistics (means, quantiles, proportions, etc.) for domains where few or no sampled units are available. Groups may be areas or other entities defined, for example, by socio-economic characteristics. The demand for such estimators is increasing as they are used for fund allocation, educational and health programs (Pfeffermann 2013). As direct estimation of such statistics are considered to be unreliable (with respect to MSE), methods in small area estimation try to improve the domain predictions by borrowing strength from neighbouring or *similar* domains. This can be achieved by using additional information from census data or registers to assist the prediction for non-sampled domains or domains with small sample sizes.

For the purpose of this article we will introduce two basic models frequently used in small area estimation, the unit-level model introduced by Battese *et al.* (1988) and the area-level model introduced by Fay and Herriot (1979). The unit level model (Battese *et al.* 1988) can be expressed as:

$$\begin{aligned} y_{ij} &= x_{ij}^\top \beta + v_i + e_{ij} \\ v_i &\stackrel{iid}{\sim} N(0, \sigma_v^2) \\ e_{ij} &\stackrel{iid}{\sim} N(0, \sigma_e^2), \end{aligned}$$

where $i = 1, \dots, D$ and $j = 1, \dots, n_i$. The population of size N is divided into D non-overlapping small areas of sizes N_i and into n sampled and $N - n$ non-sampled units, denoted by s and r respectively. y_{ij} is the dependent variable for domain i and unit j , and x_{ij} are the corresponding auxiliary information for that unit. Furthermore v and e are independent. Let $\hat{\beta}$ denote the best linear unbiased estimator (BLUE) of β and \hat{v}_i the best linear unbiased predictor (BLUP) of v_i (cf. Henderson 1950 or Searle 1971). The empirical best linear unbiased predictor (EBLUP) for the mean in small area i in the Battese-Harter-Fuller model is then given by

$$\hat{y}_i^{BHF} = N_i^{-1} \left\{ \sum_{j \in s_i} y_{ij} + \sum_{j \in r_i} (x_{ij}^\top \hat{\beta} + \hat{v}_i) \right\}. \quad (1)$$

Due to reasons of confidentiality unit-level information is not always available. Instead only aggregates for the domains or direct estimators may be supplied. In this case the feasible direct estimates are known to be unreliable in the case of small sample sizes. Here area-level models can be valuable. The area-level model introduced by Fay and Herriot (1979) is built on a sampling model:

$$y_i = \mu_i + e_i,$$

where y_i is a direct estimator of a statistic of interest μ_i for an area i . The sampling error e_i is assumed to be independent and normally distributed with known variances $\sigma_{e,i}^2$, i.e. $e_i|\mu_i \sim N(0, \sigma_{e,i}^2)$. The model assumes a linear relationship between the true area statistic μ_i and some auxiliary variables x_i :

$$\mu_i = x_i^\top \beta + v_i,$$

with $i = 1, \dots, D$. The model errors v_i are assumed to be independent and normally distributed, i.e. $v_i \sim N(0, \sigma_v^2)$. Furthermore e_i and v_i are assumed to be independent. Combining the sampling model and the linking model leads to:

$$y_i = x_i^\top \beta + v_i + e_i. \quad (2)$$

The Fay-Herriot (FH) model in (2) is effectively a random-intercept model where the distribution of the error term e_i is heterogeneous and known. The EBLUP of the small area mean in the FH model is given by

$$\hat{y}_i^{FH} = x_i^\top \hat{\beta} + \hat{v}_i. \quad (3)$$

3. A simulation framework

In this section we will present the simulation framework implemented in **saeSim**. The framework relies strongly on the idea to describe a simulation as a process of data manipulation. Independent of simulation studies, Wickham and Francois (2015) and Wickham (2014) strongly promote this idea by providing tools for cleaning and transforming data. In those frameworks every defined function takes a **data.frame** as input and returns it modified. This leads to a natural connection between all defined functions since the result of one function can be directly passed to the next as an argument. The symbioses of these packages with the pipe operator (`%>%`) from the package **magrittr** (Bache and Wickham 2014) only emphasizes the process of data manipulation. To avoid nested function calls the operator can be used to improve the readability as expressions can be read from left to right (cf. Section 4).

In **saeSim** we extend this approach to simulation studies in the field of small area estimation. The main focus lies in the description of a simulation as a process of data manipulation. Each step in this process can be defined as a self contained component (function) and thus can be easily replaced, extended and most importantly reused. Before we go into the details of the functionality of the package we discuss the process behind simulation studies and how **saeSim** maps this process into R.

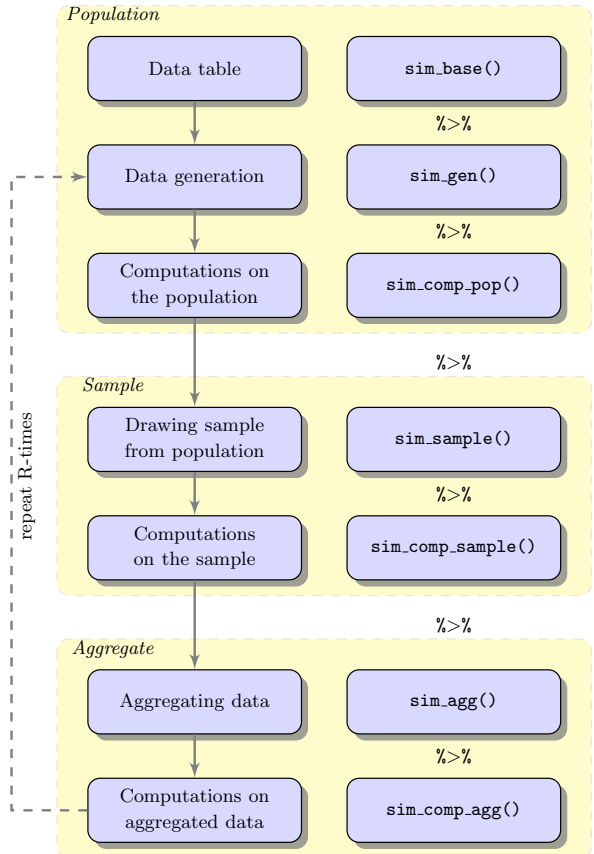


Figure 1: Process of simulation. Left column are the steps in a simulation. Right column are the corresponding function names to represent those steps in R.

Simulation studies in small area estimation address three different levels; these are the population, the sample and data on aggregated level. Figure 1 illustrates these levels. The left column describes the steps of data manipulation, the right column presents the function names to define the corresponding steps. The **population-level** defines the data on which a study is conducted and may be a true population, synthetic population data or randomly generated variates from a model. We see two different points of view to define a population: Firstly, *design-based* simulations, which means that a simulation study is based on true or synthetic data of *one* population. Secondly, *model-based* simulations, which have changing random populations drawn from a model.

The scope of this article is not to opt for viewpoints. The aim is to incorporate the different simulation concepts in a common framework. The *base* (first component in Figure 1) of a simulation study is a data table; here the question is whether these data are *fixed* or *random* over simulations. Or from a more technical point of view, is the data generation (the second step in Figure 1) repeated in each simulation run or omitted. Depending on the choice of a fixed or random population it is necessary to re-compute the population domain-statistics like domain means and variances, or other statistics of interest (third component in Figure 1).

The **sample-level** is necessary when domain predictions are conducted for unit-level models. Independently of how the population is treated - whether as fixed or random - this phase consists of two steps: Firstly, drawing a sample according to a specific sampling scheme. Secondly, conducting computations on the samples (fourth and fifth component in Figure 1). Given a sample, small area methods are applied. Of interest are, for instance, estimated model parameters, domain predictions or measures of uncertainty (MSE) for the estimates.

Since the sample-level is necessary when unit-level models are applied, the **aggregate-level** is conducted when area-level models are applied (the seventh and last component in Figure 1). Area-level models in small area estimation typically only use information available for domains (in contrast to units). Thus the question for simulation studies for area-level methods is whether the data are generated on unit-level and is used after the aggregation (sixth component in Figure 1) or whether the data are generated directly on area-level, i.e. drawn from an area-level model. Depending on whether or not unit-level data and sampling are part of the simulation process, the aggregate-level follows the generation of the population or is based on the aggregated sample.

Depending on the topic of research, some steps in this simulation framework can be more relevant than others. From our perspective, these steps are more a complete list of phases one can conduct. Single components may be omitted if not relevant in specific applications. For example *data generation* is not relevant if you have population data, or the *sample-level* is not used, when the sample is directly drawn from the model.

Seen this way, **saeSim** maps the different steps into R. Two layers with separate responsibilities need to be discussed. The first is *how* different simulation components can be combined, and the second is *when* they are applied. Regarding the first, in **saeSim** we put a special emphasis on the interface of each component. To be precise, we use functions which take a `data.frame` as argument and have a `data.frame` as return value. The return value of one component is the input of the next. This definition of interfaces is used for all existing tools in **saeSim**. The second column in Figure 1 shows how the different steps in a simulation can be accessed. It is important to note that the functions in Figure 1 control the process, the second layer, i.e. *when* components are applied. Each of these functions take a simulation setup object to be modified and a function with the discussed interface as arguments. Hence, the pipe operator (`%>%`) can be used to combine separate components to a simulation setup.

4. Case studies

We present two applications of **saeSim**, one model-based simulation in Section 4.1 and a

design-based simulation in Section 4.2. First, though, we introduce some basic functionalities as the pipe operator (`%>%`) needs some explanation. The pipe operator is designed to make otherwise nested expressions more readable as a line can be read from left to right, instead from inside out (Bache and Wickham 2014). As a simple example see the following lines which are equivalent with respect to their functionality:

```
> library("magrittr")
> colMeans(matrix(rnorm(10), ncol = 2))
> rnorm(10) %>% matrix(ncol = 2) %>% colMeans
```

In **saeSim**, we rely on this operator. Although all functions can be used without it, we strongly recommend its usage. The following example shows some of the aspects of the package:

```
> library("saeSim")
> setup1 <- sim_base_lm() %>% sim_sample(sample_number(5))
> setup2 <- sim_base_lm() %>% sim_sample(sample_fraction(0.05))
```

Without knowing anything about the setup defined in `sim_base_lm()` we notice that `setup1` and `setup2` only differ in the applied sampling scheme. `sim_sample()` is responsible as a control when a function is applied (after the population-level) and `sample_number(5)` and `sample_fraction(0.05)` define the explicit way of drawing samples. Separating the responsibility of each component into what is applied and when it is applied makes it possible to add new components to any step in the process. The composition of a simulation in that manner will focus on the definition of components and hide control structures. Any function can be passed to `sim_sample()` which has a `data.frame` both as input and as return value. The only responsibility of that function is to draw a sample, which makes it easy to find, understand and reuse when published. The operator `%>%` is used to add new components to the setup.

4.1. Model-based simulation

In the following we show one way how to construct a simulation in a model-based setting. The aim is to estimate the domain predictions under a FH model. Involved components are *data generation* and *computing on aggregated data* (cf. Figure 1). The first step is to generate the data under the model:

$$y_i = 100 + 2 \cdot x_i + v_i + e_i,$$

where $x_i \stackrel{iid}{\sim} N(0, 4^2)$, $v_i \stackrel{iid}{\sim} N(0, 1)$ and $e_i \stackrel{indep}{\sim} N(0, \sigma_i^2)$ with $\sigma_i^2 = 0.1, 0.2, \dots, 4$ and $i = 1, \dots, 40$ as index for the domains. x_i , v_i and e_i are independent from each other. The area-level data for the simulation are generated in each Monte Carlo replication.

In this case the *base-component* is a data frame with an id variable named `idD` and constructed with the function `base_id()`. Any random number generator in R can be used. However, we have normally distributed variates, for which some predefined functions are available in the package. For the reproducibility of the following results we also set the seed to 1. The seed is not part of a simulation setup in **saeSim** but needs to be defined by the researcher.

```
> set.seed(1)
> setup <- base_id(nDomains = 40, nUnits = 1) %>%
+   sim_gen_x(mean = 0, sd = 4) %>%
+   sim_gen_v(mean = 0, sd = 1)
> setup
```

```
data.frame [40 x 3]
```

	idD	x	v
1	1	-2.5058152	-0.1645236
2	2	0.7345733	-0.2533617
3	3	-3.3425144	0.6969634
4	4	6.3811232	0.5566632
5	5	1.3180311	-0.6887557
6	6	-3.2818735	-0.7074952
...

Note that if you print a simulation setup to the console, as in the above example, one simulation run is performed and only the first rows (the head) of the resulting data table are printed. This enables interactivity with the object itself; however, it hides the fact that the setup object is a collection of functions to be called. In this model the error component e_i has different variances which is not covered by a predefined function. Thus, as a *generator component*, we define a function which takes a `data.frame` as input and returns it after adding a variable named `vardir` with the variances and the variable `e` with the generated random numbers:

```
> gen_e <- function(dat) {
+   dat$vardir <- seq(0.1, 4, length.out = nrow(dat))
+   dat$e <- rnorm(nrow(dat), sd = sqrt(dat$vardir))
+   dat
+ }
> setup <- setup %>% sim_gen(gen_e)
> setup
```

```
data.frame [40 x 5]
```

	idD	x	v	vardir	e
1	1	-2.2746749	-0.5059575	0.1	0.1344285
2	2	-0.5407145	1.3430388	0.2	-0.1067262
3	3	4.7123480	-0.2145794	0.3	0.5797550
4	4	-6.0942672	-0.1795565	0.4	0.5606229
5	5	2.3757848	-0.1001907	0.5	-0.4378710
6	6	1.3318015	0.7126663	0.6	1.7088396
...

The last step in data generation is to construct the response variable which is named `y` and is added to the data. Furthermore, we add the *true* area statistic under the model to the data:

```
> setup <- setup %>%
+   sim_resp_eq(y = 100 + 2 * x + v + e) %>%
+   sim_comp_pop(comp_var(trueStat = y - e))
```

To add the area-level predictions from a Fay-Herriot model we need to define another component. The function takes a `data.frame` as input and returns the modified version. For the estimation of the EBLUP under the FH model we use the function `eb_lupFH()` from the package `sae` (Molina and Marhuenda 2013). To avoid naming conflicts between the dependencies of `sae` and the package `dplyr` (Wickham and Francois 2015) we make use of the double colon operator in order to call the function `eb_lupFH()` without attaching the package. Hence we define a function named `comp_FH()` and add it to the process:

```

> comp_FH <- function(dat) {
+   modelFH <- sae::eblupFH(y ~ x, vardir, data = dat)
+   dat$FH <- as.numeric(modelFH$eblup)
+   dat
+ }
> setup <- setup %>% sim_comp_agg(comp_FH)
> setup

data.frame [40 x 8]

   idD      x      v vardir      e      y trueStat      FH
1    1 1.637607 0.7073107    0.1 0.1258998 104.10843 103.98253 104.06121
2    2 6.755493 1.0341077    0.2 -0.1822523 114.36284 114.54509 114.23876
3    3 6.346354 0.2234804    0.3 0.7253263 113.64151 112.91619 113.45213
4    4 -1.323631 -0.8787076    0.4 -0.4434978 96.03053 96.47403 96.45416
5    5 -9.140942 1.1629646    0.5 -0.4105563 82.47052 82.88108 82.46045
6    6 9.990646 -2.0001649    0.6 -0.7754272 117.20570 117.98113 118.08379
.. ...      ...      ...      ...      ...      ...      ...

```

The object `setup` stores all necessary information to run one iteration of the simulation. In the following $R = 100$ repetitions are performed. The result is a list of `data.frames`. The function `bind_rows()` from the package **dplyr** is used to combine the resulting list:

```

> library("dplyr")
> simResults <- sim(setup, R = 100) %>% bind_rows
> simResults %>% select(idD, idR, simName, trueStat, y, FH)

```

Source: local data frame [4,000 x 6]

```

   idD idR simName trueStat      y      FH
1    1    1      107.87088 108.21065 108.30528
2    2    1      113.29033 114.13809 113.80733
3    3    1      105.88208 105.55180 105.63108
4    4    1       84.61171  84.36450  84.63272
5    5    1      103.68692 103.39260 103.42371
6    6    1      104.26130 103.97032 103.79131
7    7    1       97.85114  97.54439  97.43298
8    8    1      101.93187 101.66741 101.49500
9    9    1       92.51517  93.88300  93.67397
10   10    1      104.07055 103.37302 104.01407
.. ... ..

```

An additional variable `idR` is automatically added as an ID-variable for the iteration as well as a variable `simName` to distinguish between scenarios. In **saeSim** we do not provide further tools to process the resulting data as there are many tools readily available in R. In the design-based scenario we show how to process the result data into graphs with only a few lines of code.

4.2. Design-based simulation

In the design-based simulation we illustrate the use of **saeSim** under a fixed population. For this purpose we use a synthetic population generated from Austrian EU-SILC (European Union Statistics on Income and Living Conditions) data. The data consist of 25 thousand households. It is published alongside the R package **simFrame** ([Alfons *et al.* 2010](#)) where it is

also used as an example data set. To keep this study as simple as possible, we further restrict the data to the main income holder and only use some of the available auxiliary information.

```
> data(eusilcP, package = "simFrame")
> simDat <- eusilcP %>%
+   mutate(agesq = age^2, eqIncome = as.numeric(eqIncome)) %>%
+   filter(main) %>%
+   select(region, eqIncome, age, agesq, gender)
> head(simDat)
```

	region	eqIncome	age	agesq	gender
1	Upper Austria	11128.45	25	625	male
2	Styria	19694.85	53	2809	male
3	Styria	5066.24	30	900	female
4	Upper Austria	31480.01	32	1024	male
5	Vienna	17813.40	77	5929	female
6	Lower Austria	13501.53	35	1225	male

Using this data set as population, we repeatedly draw samples from it. Then we predict the domain means by using a direct estimator and a unit-level model. The sampling design is to draw a 10 per cent sample from each region with simple random sampling. For each region the direct estimator for income and the EBLUP under the BHF model is computed. Although the data offer some more information, we only use **gender**, **age** and **agesq** as covariates. The function `eblupBHF()` from the package **sae** is an implementation of the BHF estimator. This function takes three data objects and returns the domain predictions. The three objects are the sampled data, the population means of the auxiliary variables and the population sizes in each domain.

Before we begin to construct the simulation setup, we store these data frames as attributes to the population data. This allows us to process meta data alongside the main data frame. It is important to note that not all functions for manipulating data frames in R preserve these attributes. Users of **saeSim** have to keep this in mind when they implement new functions. Defining the interfaces between components differently is one possibility to avoid the usage of attributes. This can be done, for example, by using generic vectors or S4 classes instead of data frames. However this will add complexity to the process of data manipulation underlying the package which we try to avoid by following the paradigm: *data frame in, data frame out*. Thus all functions in **saeSim** preserve the attributes of the main data frame.

```
> attr(simDat, "popMeans") <- group_by(simDat, region) %>%
+   summarise(age = mean(age),
+             agesq = mean(agesq),
+             genderFemale = mean(as.integer(gender) - 1),
+             trueStat = mean(eqIncome))
> attr(simDat, "popMeans")
```

Source: local data frame [9 x 5]

	region	age	agesq	genderFemale	trueStat
1	Burgenland	54.50063	3269.677	0.3366708	22005.42
2	Lower Austria	51.95259	3009.934	0.3777874	19813.37
3	Vienna	46.98310	2486.448	0.4662797	20395.84
4	Carinthia	51.81428	2995.735	0.3540337	19486.18
5	Styria	50.64087	2886.845	0.3573538	19335.39
6	Upper Austria	50.18644	2795.804	0.3443871	20517.29

```

7      Salzburg 51.44943 2965.268    0.4189108 19890.33
8      Tyrol 51.76707 2995.451    0.3975648 19350.89
9      Vorarlberg 49.06904 2697.382    0.3583756 22156.12

```

```

> attr(simDat, "popN") <- group_by(simDat, region) %>% summarise(N = n())
> attr(simDat, "popN")

```

Source: local data frame [9 x 2]

	region	N
1	Burgenland	799
2	Lower Austria	4619
3	Vienna	5857
4	Carinthia	1723
5	Styria	3386
6	Upper Austria	4071
7	Salzburg	1671
8	Tyrol	1889
9	Vorarlberg	985

Before we come to the estimation, the first step is to add a sampling scheme. As stated earlier, the starting point of a simulation setup is to provide a **data.frame** as *base-component* which, in this case, is the population data. Then the sampling component is added, where the definition is to draw 10 per cent samples of the observations from each domain with simple random sampling.

```

> setup <- simDat %>%
+   sim_sample(sample_fraction(0.1, groupVars = "region"))
> setup

```

data.frame [2,500 x 5]

	region	eqIncome	age	agesq	gender
1	Burgenland	23572.28	67	4489	male
2	Burgenland	24056.37	26	676	male
3	Burgenland	21613.73	46	2116	male
4	Burgenland	11750.30	40	1600	male
5	Burgenland	9664.08	80	6400	female
6	Burgenland	19369.91	63	3969	female
..

In the next step, we define components which add the estimates of interest to the data. Here we compute the direct estimator of the mean income in each domain and the EBLUP under the BHF model. Although this could be done in one step, we separate the two computations to illustrate how to combine several estimations and how to define each component independently of the others. This automatically organises the simulation and each component is arranged using the simulation framework. Hence we define two functions, one for adding the direct estimates and one for adding the EBLUP.

```

> comp_direct <- function(dat) {
+   attr(dat, "sampleMean") <-
+     dat %>% group_by(region) %>% summarise(direct = mean(eqIncome))
+   dat
+ }

```

```

> comp_BHF <- function(dat) {
+   popMeans <- select(attr(dat, "popMeans"), -trueStat)
+   modelBHF <-
+     sae::eblupBHF(eqIncome ~ age + agesq + gender, region,
+                   meanxpop = popMeans, popnsize = attr(dat, "popN"),
+                   data = dat)
+   attr(dat, "BHF") <- modelBHF$eblup
+   dat
+ }

```

A positive aspect of the above definitions is that the code for each step is relatively short and the purpose is clearly defined. This may help to improve readability and to reproduce the research. Finally, the simulation results are combined in an *aggregation-component*, possibly followed by the application of area-level models. The result of this aggregation step is a `data.frame` with one row for each region.

```

> agg_results <- function(dat) {
+   cbind(attr(dat, "sampleMean"),
+         BHF = attr(dat, "BHF")$eblup,
+         trueStat = attr(dat, "popMeans")$trueStat)
+ }

```

To combine the simulation setup and the defined components we arrange them using the function `sim_comp_sample()` to ensure that the direct estimator and the EBLUP are computed on the sampled data, and `sim_agg()` to add the above aggregation step.

```

> setup <- setup %>%
+   sim_comp_sample(comp_BHF) %>%
+   sim_comp_sample(comp_direct) %>%
+   sim_agg(agg_results)
> setup

```

data.frame [9 x 4]

	region	direct	BHF	trueStat
1	Burgenland	21933.95	21255.53	22005.42
2	Lower Austria	18885.08	19036.29	19813.37
3	Vienna	20734.08	20720.48	20395.84
4	Carinthia	19211.55	19377.21	19486.18
5	Styria	18704.75	18955.96	19335.39
6	Upper Austria	20636.94	20504.37	20517.29
..

To repeat the simulation $R = 50$ times the simulation setup is passed to the function `sim()`. The resulting list is directly combined using `bind_rows()`.

```

> simResults <- setup %>% sim(R = 50) %>% bind_rows
> simResults

```

Source: local data frame [450 x 6]

	region	direct	BHF	trueStat	idR	simName
1	Burgenland	23083.61	20800.16	22005.42	1	

2	Lower Austria	20414.91	20264.56	19813.37	1
3	Vienna	20756.78	20422.85	20395.84	1
4	Carinthia	19581.64	19912.42	19486.18	1
5	Styria	19443.11	19757.42	19335.39	1
6	Upper Austria	20417.38	20356.38	20517.29	1
7	Salzburg	19900.44	20009.83	19890.33	1
8	Tyrol	18493.29	19540.25	19350.89	1
9	Vorarlberg	21166.20	20444.35	22156.12	1
10	Burgenland	21340.42	20510.92	22005.42	2
..

To further process the simulation results we present two plots using the package **ggplot2** (Wickham 2009) and **reshape2** (Wickham 2007) for further reshaping of the data. Figure 2 and 3 show the Monte Carlo BIAS and MSE for each region in Austria and for each estimator. Keep in mind that the BHF was applied to illustrate the simulation framework. There are a number of issues with regard to model choice, variable selection and outliers, which we will not discuss in this context.

```
> ggDat <- reshape2::melt(
+   simResults,
+   id.vars = c("region", "trueStat"),
+   measure.vars = c("BHF", "direct"),
+   variable.name = "method",
+   value.name = "prediction")

> library("ggplot2")
> ggplot(ggDat, aes(x = region, y = prediction - trueStat, fill = method)) +
+   geom_boxplot() + theme(legend.position = "bottom")
> ggplot(ggDat,
+   aes(x = region, y = (prediction - trueStat)^2, fill = method)) +
+   geom_boxplot() + scale_y_log10() + theme(legend.position = "bottom")
```

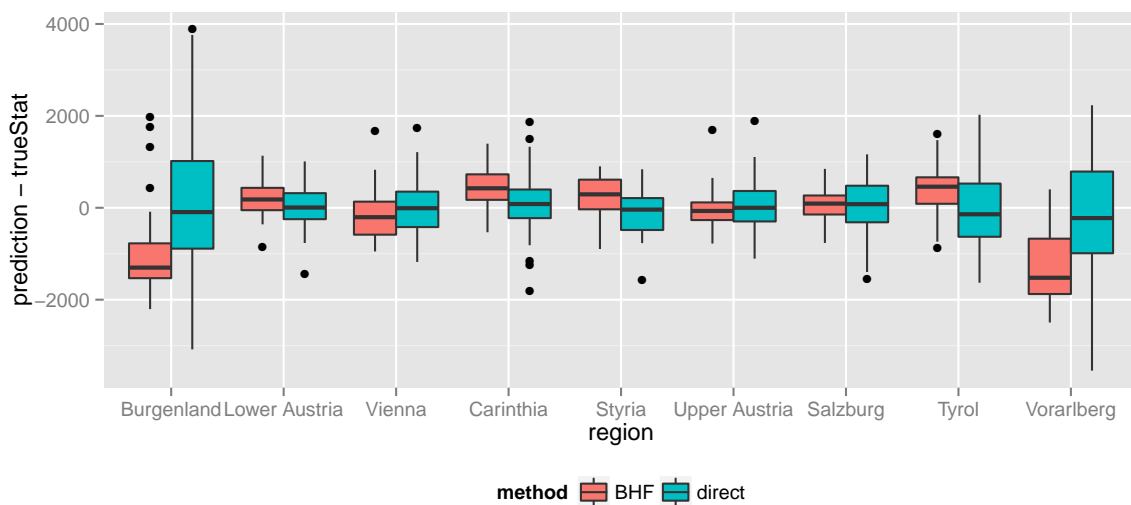


Figure 2: Monte Carlo BIAS of direct vs. BHF predictor. 50 predictions for each region and estimation technique.

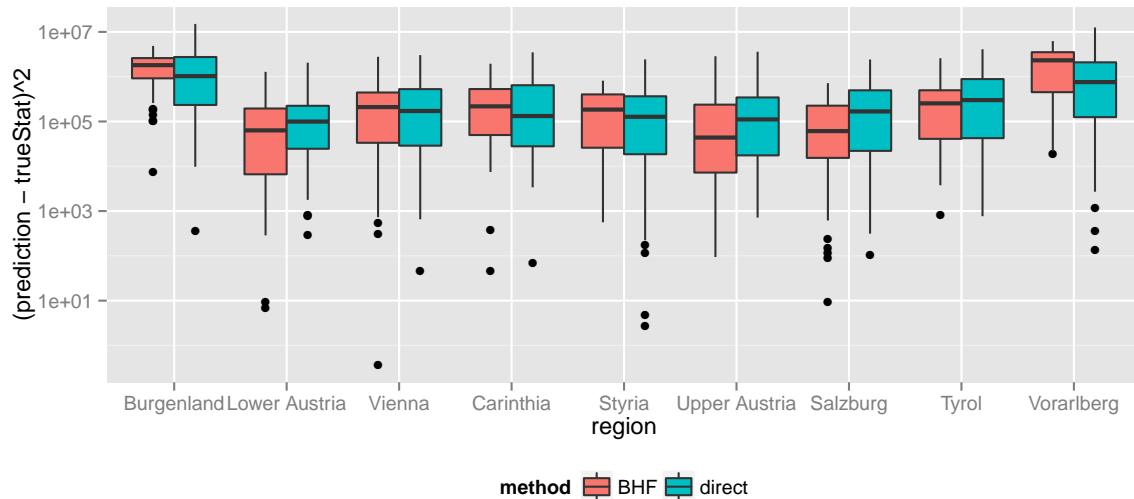


Figure 3: Monte Carlo MSE of direct vs. BHF predictor. 50 predictions for each region and estimation technique.

5. Outlook

From our perspective, there is a need for sharing tools for data generation and simulation among the scientific community in order to guarantee the reproducibility of research. **saeSim** may provide an adequate framework for pursuing this aim in the field of small area estimation. By defining the steps of a simulation we may promote a reasonable way to communicate results in academic articles and during research.

The package source is available on CRAN (<http://CRAN.R-project.org/package=saeSim>) and the repository for development on GitHub (<https://github.com/wahani/saeSim>). As GitHub allows to share and contribute source code using version control, it is open for submissions. Apart from the availability of specific utility functions, we may promote and support the design of source code for simulation studies. One aspect is the design of simulations as processes of data. Furthermore, we encourage the definition of small and self contained components, i.e. functions. This reduces the lines of code necessary to be read in order to understand its purpose.

The package provides more features than are introduced in this article. One may mention in this context the support of outlier contaminated data. Currently only representative outliers (outlying observations in the population) are supported (cf. [Chambers 1986](#)). However, we plan to extend this feature to non-representative outliers (outliers are part of the sample but not the population, e.g., incorrectly recorded values). Furthermore, an interface to random number generators in R is available. The user can also generate group effects as needed in mixed models. As the response is created by an R expression, any form of non-linearity in the relationship between response and auxiliary variables as well as error components can be modeled.

A more technical feature is a back-end for parallel computations which is a link to the **parallelMap** package in R ([Bischl and Lang 2015](#)). Tools to process result data after the simulation, i.e. summaries or plotting methods, are avenues for further research. Already available are some simple plots for the simulation setups as well as a summary method to get information on the expected run time and structure of the resulting data.

References

- Alfons A, Kraft S, Templ M, Filzmoser P (2011). “Simulation of Close-to-Reality Population Data for Household Surveys with Application to EU-SILC.” *Statistical Methods & Applications*, **20**(3), 383–407.
- Alfons A, Templ M, Filzmoser P (2010). “An Object-Oriented Framework for Statistical Simulation: The R Package **simFrame**.” *Journal of Statistical Software*, **37**(3), 1–36.
- Allaire J, McPherson J, Xie Y, Wickham H, Cheng J, Allen J (2014). **rmarkdown**: *Dynamic Documents for R*. R package version 0.3.3, URL <http://CRAN.R-project.org/package=rmarkdown>.
- Bache SM, Wickham H (2014). **magrittr**: *A Forward-Pipe Operator for R*. R package version 1.0.1, URL <http://CRAN.R-project.org/package=magrittr>.
- Battese GE, Harter RM, Fuller WA (1988). “An Error-Components Model for Prediction of County Crop Areas Using Survey and Satellite Data.” *Journal of the American Statistical Association*, **83**(401), 28–36.
- Bischl B, Lang M (2015). **parallelMap**: *Unified Interface to Some Popular Parallelization Back-Ends for Interactive Usage and Package Development*. R package version 1.2, URL <http://CRAN.R-project.org/package=parallelMap>.
- Burgard JP, Kolb JP, Münnich R (2014). “Generation of Synthetic Universes for Micro-Simulations in Survey Statistics.” *Working Paper*.
- Chambers R (1986). “Outlier Robust Finite Population Estimation.” *Journal of the American Statistical Association*, **81**(396), 1063–1069.
- Fay R, Herriot R (1979). “Estimation of Income for Small Places: An Application of James-Stein Procedures to Census Data.” *Journal of the American Statistical Association*, **74**(366), 269–277.
- Ghosh M, Rao JNK (1994). “Small Area Estimation: An Appraisal.” *Statistical Science*, **9**(1), 55–93.
- Henderson CR (1950). “Estimation of Genetic Parameters.” *Annals of Mathematical Statistics*, **21**(2), 309–310.
- Knuth DE (1992). *Literate Programming*. CSLI.
- Kolb JP (2013). *Generation of Synthetic Universes*. Ph.D. thesis, University of Trier. URL <http://ubt.opus.hbz-nrw.de/volltexte/2013/816/>.
- Leisch F (2002). “**Sweave**, Part I: Mixing R and L^AT_EX.” *R News*, **2**(3), 28–31.
- Meindl B, Templ M, Alfons A, Kowarik A, with contributions from Ribatet M (2014). **simPop**: *Simulation of Synthetic Populations for Survey Data Considering Auxiliary Information*. R package version 0.2.6, URL <http://CRAN.R-project.org/package=simPop>.
- Molina I, Marhuenda Y (2013). **sae**: *Small Area Estimation*. R package version 1.0-2, URL <http://CRAN.R-project.org/package=sae>.
- Münnich R, Schürle J, Bihler W, Boonstra H, Knotterus P, Nieuwenbroek N, Haslinger A, Laaksonen S, Eckmair D, Quatember A, Wagner H, Renfer J, Oetliker U, Wiegert R (2003). “Monte Carlo Simulation Study of European Surveys.” DACSEIS Deliverables D3.1 and D3.2. URL https://www.uni-trier.de/fileadmin/fb4/projekte/SurveyStatisticsNet/Dacseis_Deliverables/DACSEIS-D3-1-D3-2.pdf.

- Pfeffermann D (2013). “New Important Developments in Small Area Estimation.” *Statistical Science*, **28**(1), 40–68.
- R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>.
- Salvati N, Chandra H, Giovanna-Ranalli M, Chambers R (2010). “Small Area Estimation Using Nonparametric Model-Based Direct Estimator.” *Computational Statistics & Data Analysis*, **54**(9), 2159–2171.
- Schmid T, Münnich R (2014). “Spatial Robust Small Area Estimation.” *Statistical Papers*, **55**(3), 653–670.
- Searle SR (1971). *Linear Models*. John Wiley & Sons, New York.
- Wickham H (2007). “Reshaping Data with the **reshape** Package.” *Journal of Statistical Software*, **21**(12), 1–20.
- Wickham H (2009). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag.
- Wickham H (2014). *tidyr: Easily tidy data with spread and gather functions*. R package version 0.1, URL <http://CRAN.R-project.org/package=tidyr>.
- Wickham H, Francois R (2015). *dplyr: A Grammar of Data Manipulation*. R package version 0.4.1, URL <http://CRAN.R-project.org/package=dplyr>.
- Yihui X (2013). *Dynamic Documents with R and knitr*. Chapman & Hall/CRC.

Affiliation:

Timo Schmid

Department of Economics

Freie Universität Berlin

D-14195 Berlin, Germany

E-mail: Timo.Schmid@fu-berlin.de

URL: <http://www.wiwiss.fu-berlin.de/fachbereich/vwl/Schmid>

Sebastian Warnholz

Department of Economics

Freie Universität Berlin

D-14195 Berlin, Germany

E-mail: Sebastian.Warnholz@fu-berlin.de

URL: <http://www.wiwiss.fu-berlin.de/fachbereich/vwl/Schmid/Team/Warnholz.html>

Robust Maximum Association Between Data Sets: The R Package ccaPP

Andreas Alfons
Erasmus Universiteit Rotterdam

Christophe Croux
KU Leuven

Peter Filzmoser
Vienna University of Technology

Abstract

An intuitive measure of association between two multivariate data sets can be defined as the maximal value that a bivariate association measure between any one-dimensional projections of each data set can attain. Rank correlation measures thereby have the advantage that they combine good robustness properties with good efficiency. The software package **ccaPP** provides fast implementations of such maximum association measures for the statistical computing environment R. We demonstrate how to use **ccaPP** to compute the maximum association measures, as well as how to assess their significance via permutation tests.

Keywords: multivariate analysis, outliers, projection pursuit, rank correlation, R.

1. Introduction

Projection pursuit allows to introduce intuitive and therefore appealing association measures between two multivariate data sets. Suppose that the data sets \mathbf{X} and \mathbf{Y} consist of p and q variables, respectively. A measure of multivariate association between \mathbf{X} and \mathbf{Y} can be defined by looking for linear combinations $\mathbf{X}\boldsymbol{\alpha}$ and $\mathbf{Y}\boldsymbol{\beta}$ having maximal association. Expressed in mathematical terms, we define an estimator

$$\hat{\rho}_R(\mathbf{X}, \mathbf{Y}) = \max_{\|\boldsymbol{\alpha}\|=1, \|\boldsymbol{\beta}\|=1} \hat{R}(\mathbf{X}\boldsymbol{\alpha}, \mathbf{Y}\boldsymbol{\beta}), \quad (1)$$

where \hat{R} is an estimator of a bivariate association measure R such as the Pearson correlation, or the Spearman or Kendall rank correlation. Using the projection pursuit terminology, \hat{R} is the *projection index* to maximize. The projection directions corresponding to the maximum association are called *weighting vectors* and are estimated by

$$(\hat{\boldsymbol{\alpha}}_R(\mathbf{X}, \mathbf{Y}), \hat{\boldsymbol{\beta}}_R(\mathbf{X}, \mathbf{Y})) = \underset{\|\boldsymbol{\alpha}\|=1, \|\boldsymbol{\beta}\|=1}{\operatorname{argmax}} \hat{R}(\mathbf{X}\boldsymbol{\alpha}, \mathbf{Y}\boldsymbol{\beta}). \quad (2)$$

Alfons, Croux, and Filzmoser (2016) developed the *alternate grid algorithm* for the computation of such maximum association estimators and studied their theoretical properties for various association measures. It turns out that the Spearman and Kendall rank correlation yield maximum association estimators with good robustness properties and good efficiency.

This paper is a companion paper to Alfons *et al.* (2016) that demonstrates how to apply the maximum association estimators in the statistical environment R (R Core Team 2015) using the add-on package **ccaPP** (Alfons 2015). The package is freely available on CRAN (Comprehensive R Archive Network, <http://CRAN.R-project.org>).

Note that using the Pearson correlation as the projection index of the maximum association estimator corresponds to the first step of canonical correlation analysis (CCA; see, e.g., Johnson and Wichern 2002), hence the package name **ccaPP**. Since CCA is a widely applied statistical technique, various algorithms and extensions are implemented in R packages on CRAN. Two important examples are briefly discussed in the following. The package **CCA** (González, Déjean, Martin, and Baccini 2008; González and Déjean 2012) extends the built-in R function `cancor()` with additional numerical and graphical output. Moreover, it provides a regularized version of CCA for data sets containing a large number of variables. Bayesian models and inference methods for CCA are implemented in the package **CCAGFA** (Klami, Virtanen, and Kaski 2013; Virtanen, Leppaaho, and Klami 2015).

The remainder of the paper is organized as follows. In Section 2, the design and implementation of the package are briefly discussed. Section 3 demonstrates how to compute the maximum association estimators, and Section 4 illustrates how to test for their significance. A comparison of computation times is given in Section 5. The final Section 6 concludes the paper.

2. Design and implementation

Various bivariate association measures and the alternate grid algorithm for the maximum association estimators are implemented in C++, and integrated into R via the package **RcppArmadillo** (Eddelbuettel and Sanderson 2014; Eddelbuettel, François, and Bates 2015). The following bivariate association measures are available in the package **ccaPP**:

`corPearson()`: Pearson correlation

`corSpearman()`: Spearman rank correlation

`corKendall()`: Kendall rank correlation, also known as Kendall's τ

`corQuadrant()`: Quadrant correlation (Blomqvist 1950)

`corM()`: Association based on a bivariate M-estimator of location and scatter with a Huber loss function (Huber and Ronchetti 2009)

It should be noted that these are barebones implementations without proper handling of missing values. Hence the first three functions come with a substantial speed gain compared to R's built-in function `cor()`. Moreover, the fast $O(n \log(n))$ algorithm for the Kendall correlation (Knight 1966) is implemented in `corKendall()`, whereas `cor()` uses the naive $O(n^2)$ algorithm.

The alternate grid algorithm for the maximum association estimators is implemented in the function `maxCorGrid()`. Any of the bivariate association measures above can be used as projection index, with the Spearman rank correlation being the default. We do not recommend to use the quadrant correlation since its influence function is not smooth, which may result in unstable estimates of the weighting vectors. For more details on the theoretical properties of the maximum association estimators, the reader is referred to Alfons *et al.* (2016).

To assess the significance of a maximum association estimate, a permutation test is provided via the function `permTest()`. Parallel computing to increase computational performance is implemented via the package **parallel**, which is part of R since version 2.14.0.

3. Maximum association measures

In this section, we show how to apply the function `maxCorGrid()` from the package **ccaPP** to compute the maximum association estimators. We thereby use the classic **diabetes** data (Andrews and Herzberg 1985, page 215), which are included as example data in the package. First we load the package and the data. All measurements are taken for a group of $n = 76$ persons.

```
library("ccaPP")
data("diabetes")
x <- diabetes$x
y <- diabetes$y
```

Component **x** consists of $p = 2$ variables measuring *relative weight* and *fasting plasma glucose*, while component **y** consists of $q = 3$ variables measuring *glucose intolerance*, *insulin response to oral glucose* and *insulin resistance*. It is of medical interest to establish a relation between the two data sets.

The function `maxCorGrid()` by default uses the Spearman rank correlation as projection index.

```
spearman <- maxCorGrid(x, y)
spearman
##
## Call:
## maxCorGrid(x = x, y = y)
##
## Maximum correlation:
## [1] 0.5346995
```

The estimated weighting vectors can be accessed through components **a** and **b** of the returned object, respectively.

```
spearman$a
## [1] -0.2560459 0.9666646
spearman$b
## [1] 9.999999e-01 3.630496e-04 4.520212e-05
```

With the argument **method**, another bivariate association measure can be set as projection index, e.g., the Kendall rank correlation, the M-association or the Pearson correlation.

```
maxCorGrid(x, y, method = "kendall")
##
## Call:
## maxCorGrid(x = x, y = y, method = "kendall")
##
## Maximum correlation:
## [1] 0.3969117
maxCorGrid(x, y, method = "M")
##
## Call:
## maxCorGrid(x = x, y = y, method = "M")
##
## Maximum correlation:
## [1] 0.5342933
```

```
maxCorGrid(x, y, method = "pearson")
##
## Call:
## maxCorGrid(x = x, y = y, method = "pearson")
##
## Maximum correlation:
## [1] 0.4887632
```

Note that the Spearman and Kendall rank correlation estimate different population quantities than the Pearson correlation. Thus the above values of the different maximum association measures are not directly comparable. The argument `consistent` can be used for the former two methods to get consistent estimates of the maximum correlation under normal distributions.

```
maxCorGrid(x, y, consistent = TRUE)
##
## Call:
## maxCorGrid(x = x, y = y, consistent = TRUE)
##
## Maximum correlation:
## [1] 0.5526498
maxCorGrid(x, y, method = "kendall", consistent = TRUE)
##
## Call:
## maxCorGrid(x = x, y = y, method = "kendall", consistent = TRUE)
##
## Maximum correlation:
## [1] 0.5838538
```

The M-association measure is consistent at the normal model and estimates the same population quantity as the Pearson correlation.

4. Permutation tests

To assess the significance of maximum association estimates, permutation tests can be performed with the function `permTest()`. The number of random permutations to be used can be set with the argument `R`, which defaults to 1000. On machines with multiple processor cores, only the argument `nCores` needs to be set to take advantage of parallel computing in order to reduce computation time. If `nCores` is set to `NA`, all available processor cores are used.

In the examples in this section, we use 2 processor cores. Furthermore, we set the seed of the random number generator via the argument `seed` for reproducibility of the results. Since we employ parallel computing, **ccaPP** uses random number streams (L'Ecuyer, Simard, Chen, and Kelton 2002) from the package **parallel** rather than the default R random number generator.

```
permTest(x, y, nCores = 2, seed = 2014)
##
## Permutation test for no association
##
## r = 0.534699, p-value = 0.001000
## R = 1000 random permutations
## Alternative hypothesis: true maximum correlation is not equal to 0
```


Again, the Spearman rank correlation is used as projection index by default. A different bivariate association measure can be specified via the argument `method`, which is passed down to the function `maxCorGrid()`.

```
permTest(x, y, method = "kendall", nCores = 2, seed = 2014)
##
## Permutation test for no association
##
## r = 0.396912, p-value = 0.001000
## R = 1000 random permutations
## Alternative hypothesis: true maximum correlation is not equal to 0
permTest(x, y, method = "M", nCores = 2, seed = 2014)
##
## Permutation test for no association
##
## r = 0.534293, p-value = 0.001000
## R = 1000 random permutations
## Alternative hypothesis: true maximum correlation is not equal to 0
permTest(x, y, method = "pearson", nCores = 2, seed = 2014)
##
## Permutation test for no association
##
## r = 0.488764, p-value = 0.000000
## R = 1000 random permutations
## Alternative hypothesis: true maximum correlation is not equal to 0
```

Clearly, all four tests strongly reject the null hypothesis of no association between the two data sets.

Since the focus of **ccaPP** is on robustness, we introduce an outlier into the `diabetes` data as in [Taskinen, Kankainen, and Oja \(2003\)](#). More precisely, we replace the value 0.81 of the first observation of variable *glucose intolerance* by 8.1, i.e., by a simple shift of the comma.

```
y[1, "GlucoseIntolerance"] <- 8.1
```

Now we repeat the four permutation tests with the contaminated data.

```
permTest(x, y, nCores = 2, seed = 2014)
##
## Permutation test for no association
##
## r = 0.487536, p-value = 0.003000
## R = 1000 random permutations
## Alternative hypothesis: true maximum correlation is not equal to 0
permTest(x, y, method = "kendall", nCores = 2, seed = 2014)
##
## Permutation test for no association
##
## r = 0.361116, p-value = 0.003000
## R = 1000 random permutations
## Alternative hypothesis: true maximum correlation is not equal to 0
```

```
permTest(x, y, method = "M", nCores = 2, seed = 2014)
##
## Permutation test for no association
##
## r = 0.509974, p-value = 0.001000
## R = 1000 random permutations
## Alternative hypothesis: true maximum correlation is not equal to 0
permTest(x, y, method = "pearson", nCores = 2, seed = 2014)
##
## Permutation test for no association
##
## r = 0.267837, p-value = 0.337000
## R = 1000 random permutations
## Alternative hypothesis: true maximum correlation is not equal to 0
```

The test based on the maximum Pearson correlation is highly influenced by the outlier and no longer rejects the null hypothesis. The tests based on the maximum Spearman and Kendall rank correlation, as well as the test based on maximum M-association, remain stable.

5. Computation times

This section analyzes the computation times of the methods implemented in **ccaPP**. All computations are performed in R version 3.2.2 on a machine with an Intel Xeon X5670 CPU. The computation times are recorded with the R package **microbenchmark** (Mersmann 2014).

First, we compare the barebones implementations of the Pearson, Spearman and Kendall correlations (functions `corPearson()`, `corSpearman()` and `corKendall()` in **ccaPP**) with their counterparts from the base R function `cor()`. We also include the M-association measure from the function `corM()` in the comparison. The bivariate association measures are computed for 10 random draws from a bivariate normal distribution with true correlation $\rho = 0.5$ and sample size $n = 100, 1\,000, 10\,000, 100\,000$. For each random sample, computation times from 10 independent runs are recorded.

Table 1 contains the average computation times of the bivariate association measures. Clearly, the fast $O(n \log(n))$ algorithm for the Kendall correlation (Knight 1966) in **ccaPP** is a huge improvement over the naive $O(n^2)$ algorithm in base R. Time savings for the Spearman and Pearson correlation are also substantial, considering that they are only due to a lack of missing data handling. For the M-association, the computation time is somewhat higher than that of the Spearman and Kendall correlation.

Since the projection pursuit algorithm for the maximum association measures involves computing a large number of bivariate associations (see Alfons *et al.* 2016), the faster barebones implementations are crucial to keep the computation of the maximum association feasible.

Table 1: Average computation time (in milliseconds) of the bivariate association measures in base R and the package **ccaPP**.

n	Base R			Package ccaPP			
	Spearman	Kendall	Pearson	Spearman	Kendall	Pearson	M
100	0.20	0.40	0.08	0.03	0.03	0.01	0.11
1 000	0.41	18.73	0.08	0.18	0.16	0.01	0.32
10 000	3.38	1761.71	0.19	2.06	1.84	0.05	2.42
100 000	54.88	176431.15	1.34	25.21	22.46	0.41	27.42

Table 2: Average computation time (in seconds) of the maximum association measures in package **ccaPP**, as well as association measures based on corresponding full correlation matrix.

n	p	q	Package ccaPP				Full scatter matrix			
			Spearman	Kendall	Pearson	M	Spearman	Kendall	Pearson	MCD
100	5	1	0.014	0.011	0.001	0.036	0.001	0.005	0.001	0.020
100	5	5	0.073	0.049	0.006	0.244	0.002	0.010	0.001	0.038
100	10	1	0.030	0.023	0.003	0.088	0.002	0.012	0.001	0.044
100	10	5	0.114	0.083	0.012	0.473	0.002	0.021	0.001	0.075
100	10	10	0.180	0.107	0.021	0.658	0.003	0.037	0.001	0.130
100	50	1	0.174	0.137	0.047	0.641	0.007	0.224	0.003	0.926
100	50	5	0.588	0.429	0.365	5.777	0.008	0.259	0.003	1.096
100	50	10	0.692	0.435	0.426	8.249	0.009	0.307	0.003	1.348
100	50	50	1.257	0.824	0.993	33.368	0.013	0.839	0.005	
1 000	5	1	0.189	0.152	0.005	0.219	0.002	0.324	0.001	0.075
1 000	5	5	1.143	0.961	0.035	1.280	0.003	0.860	0.001	0.143
1 000	10	1	0.408	0.342	0.018	0.532	0.004	1.034	0.001	0.165
1 000	10	5	1.837	1.620	0.072	2.239	0.005	1.890	0.001	0.271
1 000	10	10	2.567	2.145	0.110	3.693	0.006	3.320	0.001	0.459
1 000	50	1	2.285	2.055	0.293	3.567	0.019	21.126	0.005	2.805
1 000	50	5	8.728	7.611	1.188	14.019	0.020	24.544	0.006	3.264
1 000	50	10	10.264	8.661	1.271	16.524	0.024	29.184	0.006	3.938
1 000	50	50	21.192	16.785	3.448	39.227	0.038	80.656	0.011	14.740
10 000	5	1	1.933	1.895	0.043	1.472	0.018	32.153	0.002	0.115
10 000	5	5	12.136	10.695	0.251	8.958	0.036	85.527	0.004	0.214
10 000	10	1	4.783	4.113	0.140	3.223	0.032	102.857	0.003	0.234
10 000	10	5	19.922	19.365	0.539	17.111	0.043	188.259	0.004	0.369
10 000	10	10	32.188	24.658	0.856	22.533	0.063	330.891	0.006	0.618
10 000	50	1	28.747	26.078	3.150	29.440	0.153	2107.029	0.028	3.374
10 000	50	5	116.614	100.885	9.538	114.121	0.160	2448.142	0.032	3.917
10 000	50	10	134.916	103.590	10.014	123.863	0.179	2910.402	0.035	4.706
10 000	50	50	244.389	209.834	20.318	224.293	0.320	8045.749	0.082	16.556

We employ the same procedure as above to record the computation time of the maximum association measures, except that each of the random samples is drawn from a multivariate normal distribution such that the true maximum correlation is $\rho = 0.5$ and the corresponding weighting vectors are $\alpha = (1, 0, \dots, 0)'$ and $\beta = (1, 0, \dots, 0)'$. The sample size is set to $n = 100, 1\,000, 10\,000$, the dimension of \mathbf{X} is $p = 5, 10, 50$, and the dimension of \mathbf{Y} is $q = 1, 5, 10, 50$.

Inspired by canonical correlation analysis (CCA), we also compute other association measures for comparison. In CCA, the first canonical correlation is given by the square root of the largest eigenvalue of the matrix

$$\Sigma_{XX}^{-1} \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX}, \quad (3)$$

where $\Sigma_{XX} = \text{Cov}(\mathbf{X})$, $\Sigma_{YY} = \text{Cov}(\mathbf{Y})$, $\Sigma_{XY} = \text{Cov}(\mathbf{X}, \mathbf{Y})$ and $\Sigma_{YX} = \Sigma'_{XY}$ (see, e.g., [Johnson and Wichern 2002](#)). This is of course identical to the maximum association measure with the Pearson correlation as projection index. Other association measures are obtained by plugging different scatter matrices into (3). However, such a measure is in general different from the maximum association measure based on the corresponding bivariate association, with the maximum association being much easier to interpret. Here we plug in scatter matrices

corresponding to the Pearson, Spearman and Kendall correlation. For the Pearson correlation, the corresponding scatter matrix is the sample covariance matrix. For the Spearman and Kendall correlation, the scatter matrices are given by the respective pairwise associations multiplied with scale estimates of the corresponding variables. Furthermore, since a multivariate M-estimator of the covariance matrix is not robust, we instead use the minimum covariance determinant estimator (MCD; see Rousseeuw and Van Driessen 1999).

Table 2 lists average computation times for various values of n , p and q . The function `maxCorGrid()` is thereby used with the default values for all control parameters of the algorithm (see the corresponding R help file). For the maximum association measures, the number of bivariate associations that have to be computed clearly takes a toll on computation time compared to the association measures based on the full scatter matrices. Note that the Kendall correlation is the exception, as the computation of the full scatter matrix uses R's built-in `cor()` function, and therefore the naive $O(n^2)$ algorithm. Also note that computing the full MCD scatter matrix requires more observations than variables, i.e., $n > p + q$, hence it cannot be computed for $n = 100$ and $p = q = 50$.

For the Pearson correlation, the projection pursuit algorithm to find the maximum association cannot be recommended since the first canonical correlation is much faster to compute. However, the focus of **ccaPP** is on the Spearman and Kendall rank correlation, for which the maximum association measures are much more intuitive than the association measures based on the full scatter matrix. In our opinion, the gain of easy interpretability outweighs the increased computational cost. In any case, the maximum association measures are still reasonably fast to compute for many problem sizes due to our C++ implementation.

It is also worth noting that the association measures based on a full scatter matrix require the number of observations to be larger than the number of variables in each of the two data sets, i.e., $n > \max(p, q)$. The maximum association measures do not have this limitation, although computation time increases considerably in high dimensions.

6. Conclusions

The package **ccaPP** provides functionality for the statistical computing environment R to compute intuitive measures of association between two data sets. These maximum association measures seek the maximal value of a bivariate association measure between one-dimensional projections of each data set. We recommend the maximum Spearman and Kendall rank correlation measures because of their good robustness properties and efficiency. For details on the theoretical properties of the estimators, as well as the alternate grid algorithm and extensive numerical results, the reader is referred to Alfons *et al.* (2016).

Due to our C++ implementation, the maximum association measures are reasonably fast to compute. The significance of maximum association estimates can be assessed via permutation tests, which allow for parallel computing to decrease computation time. In addition, the corresponding functions in **ccaPP** are easy to use.

References

- Alfons A (2015). *ccaPP: (Robust) Canonical Correlation Analysis via Projection Pursuit*. R package version 0.3.1, URL <http://CRAN.R-project.org/package=ccaPP>.
- Alfons A, Croux C, Filzmoser P (2016). "Robust Maximum Association Estimators." *Journal of the American Statistical Association*. doi:10.1080/01621459.2016.1148609. In press.
- Andrews D, Herzberg A (1985). *Data*. Springer-Verlag, New York. ISBN 978-1-4612-5098-2.
- Blomqvist N (1950). "On a Measure of Dependence Between Two Random Variables." *The Annals of Mathematical Statistics*, **21**(4), 593–600.

- Eddelbuettel D, François R, Bates D (2015). *RcppArmadillo: Rcpp Integration for Armadillo Templated Linear Algebra Library*. R package version 0.6.200.2.0, URL <http://CRAN.R-project.org/package=RcppArmadillo>.
- Eddelbuettel D, Sanderson C (2014). “RcppArmadillo: Accelerating R with High-Performance C++ Linear Algebra.” *Computational Statistics & Data Analysis*, **71**, 1054–1063.
- González I, Déjean S (2012). *CCA: Canonical Correlation Analysis*. R package version 1.2, URL <http://CRAN.R-project.org/package=CCA>.
- González I, Déjean S, Martin P, Baccini A (2008). “CCA: An R Package to Extend Canonical Correlation Analysis.” *Journal of Statistical Software*, **23**(12), 1–14.
- Huber P, Ronchetti E (2009). *Robust Statistics*. 2nd edition. John Wiley & Sons, New York. ISBN 978-0-470-12990-6.
- Johnson R, Wichern D (2002). *Applied Multivariate Statistical Analysis*. 5th edition. Prentice Hall, Upper Saddle River, New Jersey. ISBN 978-0-130-92553-4.
- Klami A, Virtanen S, Kaski S (2013). “Bayesian Canonical Correlation Analysis.” *Journal of Machine Learning Research*, **14**(Apr), 965–1003.
- Knight W (1966). “A Computer Method for Calculating Kendall’s Tau with Ungrouped Data.” *Journal of the American Statistical Association*, **61**(314), 436–439.
- L’Ecuyer P, Simard R, Chen E, Kelton W (2002). “An Object-Oriented Random-Number Package with Many Long Streams and Substreams.” *Operations Research*, **50**(6), 1073–1075.
- Mersmann O (2014). *microbenchmark: Accurate Timing Functions*. R package version 1.4-2, URL <http://CRAN.R-project.org/package=microbenchmark>.
- R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>.
- Rousseeuw PJ, Van Driessen K (1999). “A Fast Algorithm for the Minimum Covariance Determinant Estimator.” *Technometrics*, **41**(3), 212–223.
- Taskinen S, Kankainen A, Oja H (2003). “Sign Test of Independence Between Two Random Vectors.” *Statistics and Probability Letters*, **62**(1), 9–21.
- Virtanen S, Leppaaho E, Klami A (2015). *CCAGFA: Bayesian Canonical Correlation Analysis and Group Factor Analysis*. R package version 1.0.7, URL <http://CRAN.R-project.org/package=CCAGFA>.

Affiliation:

Andreas Alfons
 Erasmus School of Economics
 Erasmus Universiteit Rotterdam
 PO Box 1738
 3000DR Rotterdam
 E-mail: alfons@ese.eur.nl
 URL: <http://people.few.eur.nl/alfons/>

Austrian Journal of Statistics
 published by the Austrian Society of Statistics

<http://www.ajs.or.at/>
<http://www.osg.or.at/>

Volume 45
 March 2016

Submitted: 2014-11-05
 Accepted: 2015-12-11

From Climate Simulations to Statistics – Introducing the wux Package

Thomas Mendlik
Wegener Center

Georg Heinrich
Wegener Center

Andreas Gobiet
ZAMG

Armin Leuprecht
Wegener Center

Abstract

We present the R package **wux**, a toolbox to analyze projected climate change signals by numerical climate model simulations and the associated uncertainties. The focus of this package is to automatically process big amounts of climate model data from multi-model ensembles in a user-friendly and flexible way. For this purpose, climate model output in common binary format (NetCDF) is read in and stored in a data frame, after first being aggregated to a desired temporal resolution and then being averaged over spatial domains of interest. The data processing can be performed for any number of meteorological parameters at one go, which allows multivariate statistical analysis of the climate model ensemble.

Keywords: climate research, climate uncertainty, multi-model ensembles, data processing, R.

1. Introduction

The human influence on the climate system is often assessed using numerical climate simulations (General Circulation models or GCMs). These are models representing physical processes in the atmosphere, ocean, cryosphere and land surface. However, due to their relative coarse resolution in the order of several hundred kilometers, they are not able to cover important processes at smaller spatial scales. For a more regional analysis, the models can be refined using regional climate models (RCMs) (Giorgi and Mearns 1991) and statistical down-scaling techniques (Maraun, Wetterhall, Ireson, Chandler, Kendon, Widmann, Brienen, Rust, Sauter, Themeßl, Venema, Chun, Goodess, Jones, Onof, Vrac, and Thiele-Eich 2010). Under certain assumptions of future greenhouse gas (GHG) emissions, those models can project climate into future periods. We define a climate change signal of a particular meteorological parameter (from climate simulations) as the measure of change between a future climate projection and the past climate.

However, those projected climates are subject to different sources of uncertainty stemming from the natural variability of the climate system, unknown future GHG emissions, and errors and simplifications in GCMs and from regionalization methods. The resulting uncertainties can be partly assessed by analyzing so-called multi-model ensembles (i.e. climate projections which are generated by various GCMs and RCMs), which aim to sample the various sources of uncertainty. However, those ensembles do not systematically sample components of model

uncertainty (e.g. physical parametrizations), and thus do not stem from an experimental design in a statistical sense (Knutti, Furrer, Tebaldi, Cermak, and Meehl 2010). They cannot be expected to represent unbiased distributions of possible future climate states. Also, interdependence between GCMs may induce additional biases in the sample, which makes a proper statistical analysis even more difficult. Several publications address those problems, for example Tebaldi and Knutti (2007); Smith, Tebaldi, Nychka, and Mearns (2009); Pirtle, Meyer, and Hamilton (2010); Bishop and Abramowitz (2012); Collins, Chandler, Cox, Huthnance, Rougier, and Stephenson (2012); Fischer, Weigel, Buser, Knutti, Künsch, Liniger, Schür, and Appenzeller (2012); Kang, Cressie, and Sain (2012); Stephenson, Collins, Rougier, and Chandler (2012); Chandler (2013); Rougier, Goldstein, and House (2013); Stephenson *et al.* (2012); Mendlik and Gobiet (2015).

This paper introduces the R package **wux** (Wegener Center Climate Uncertainty Explorer) (Mendlik, Heinrich, and Leuprecht 2015), a toolbox which enables multi-model handling for statistical analysis of climate scenarios. It is intended to be used to interpret climate model output and provide uncertainty information for the end-user of the climate simulations. Having in mind the heterogeneous target audience, we want this tool to perform following tasks:

1. Enable easy statistical *descriptive analysis* of user-defined climate model ensembles.
2. Be *expandable* to any kind of statistical analysis (to push the development of new statistical methods for climate multi-model analysis).
3. Easily *process climate simulations* to a common data format usable for statistical analysis. This enables reproducing data for any analysis needed.

Descriptive statistics of climatic changes from ensembles (point 1) are crucial to understand the underlying data. In practice people sometimes tend to forget this important step and prefer to directly address their complex research questions without having an overview of the data beforehand. A lot of valuable information lies in this analysis. Having some ready-to-use tools already implemented in **wux** should encourage users to perform this sort of analysis more often.

However, such a tool should not restrict the user to a pre-defined set of standard methods, on the contrary, development of new methods for statistical inference on climate simulations should be strongly supported, as this is still ongoing research (Knutti *et al.* 2010). Having set up this tool directly in R, allows to explore an extremely broad pool of ready-to-use methods, also from other disciplines using different approaches (point 2).

One of the most time consuming and frustrating tasks when analyzing climate simulations can be the step of processing data (point 3). The user of this tremendously big amount of datasets will find him-/herself challenged, when trying to aggregate them to the desired format (typically some sort of data frame) or get the desired statistics of the ensemble for certain geographical regions of interest. The challenge here is definitely a technical one: Processing ensembles of data in a binary-format usually requires dedicated programming work. The upside is that the data comes in the handy NetCDF file format¹, where a lot of meta-information about the data is stored in its header, however, life is more complicated in practice. Quite often it happens that meta-information between individual climate simulation output files differ substantially. For this reason it quickly becomes a nuisance when treating large samples of these files in an automated way. Up to now, no such tool is available which processes user-defined climate simulations in an automated way and which allows sophisticated statistical analysis. Furthermore, it is very difficult to reproduce statistical analysis from the scientific community when either the data set from the publication is not available, or the user wishes to apply the method with his/her own climate data. Providing a software which takes this burden, allows the user to solely focus on the interpretation of the climate model output

¹<http://www.unidata.ucar.edu/software/netcdf/>

without spending too many resources on technicalities. We consider it a great strength of this package to perform this task in an automated way.

Several powerful tools already exist to process climate model outputs, such as CDO², NCO (Zender 2008), climate explorer³ (van Oldenborgh, Drijfhout, van Ulden, Haarsma, Sterl, Severijns, Hazeleger, and Dijkstra 2009) and NCL⁴. All of those tools are designed to perform some sort of descriptive analysis and/or process the data to a desired format, however, none of those tools combines both easy multi-model handling and flexibility in statistical analysis. For example the climate explorer allows very straight forward processing of multi-model ensembles without any programming work. The user specifies what climate models to analyze simply by clicking on their names and the desired statistics. Such web-based tools however, being simple to use, lack of flexibility for a real programming interface. In addition it is not possible to extend those tools for own climate simulations which are not implemented. Also, statistical analysis is restricted to available methods. More programming-oriented tools like CDO and NCO also provide possibilities to analyze ensembles of climate simulations. However, the user has to specify the location of the data each time when calling a function and the data have to be pre-formatted for the program to understand its meaning. Changing local NetCDF files too much is a restriction to reproducible research. Even though programming is possible, we are restricted to pre-defined CDO statistics operators. The main difference of **wux** compared to those tools is the easy way it can read in a multitude of climate simulations and simply the fact that this tool is embedded in R, which allows to apply a very broad range of sophisticated statistical tools and is not restricted only by methods implemented in the toolbox itself.

The structure of this paper is as follows. Section 2 gives an overview of the functionalities of the **wux** package. Section 3 describes how climate data are being processed to a suitable data frame step-by-step. We introduce the statistical functionalities implemented in the package in Section 4 and provide an example application in Section 5 to show possible extensions of the implemented statistical functionalities. We conclude in Section 6.

2. Package overview

wux is meant to be an interfacing toolbox for scientists performing statistical analysis on climate models. Its focus is to provide a simple data frame for the user to make statistical inference on the ensemble. In particular, this package performs following actions, which are depicted in Figure 1 and described in Table 1:

Climate data processing. The function `models2wux` reads output of climate model simulations from NetCDF files, extracts subregions of interest, and writes climate change signals or time series to a data frame. Specific meta-information, like file locations, are stored in a `modelinput` input argument, which allows to simple processing of the simulations. For any new climate simulation it is enough to specify those meta-information without having to actually program a new input routine.

Statistical analysis of climate change signals. Based on the data frame returned by `models2wux`, we implemented various plotting options and summarizing utilities for a descriptive analysis of the projected climate change signals (e.g. scatterplots of temperature and precipitation). In addition, reconstruction tools allow to fill up missing climate simulations by multiple imputation methods. Based on such a reconstructed data frame (here termed as `rwux.df`), the user can assess for variance components via the implemented ANOVA tools or perform exploratory data analysis.

²CDO 2014: Climate Data Operators. Available at: <https://code.zmaw.de/projects/cdo>

³<http://climexp.knmi.nl>

⁴The NCAR Command Language (Version 6.2.1) [Software]. (2014). Boulder, Colorado: UCAR/NCAR/-CISL/VETS. <http://dx.doi.org/10.5065/D6WD3XH5>

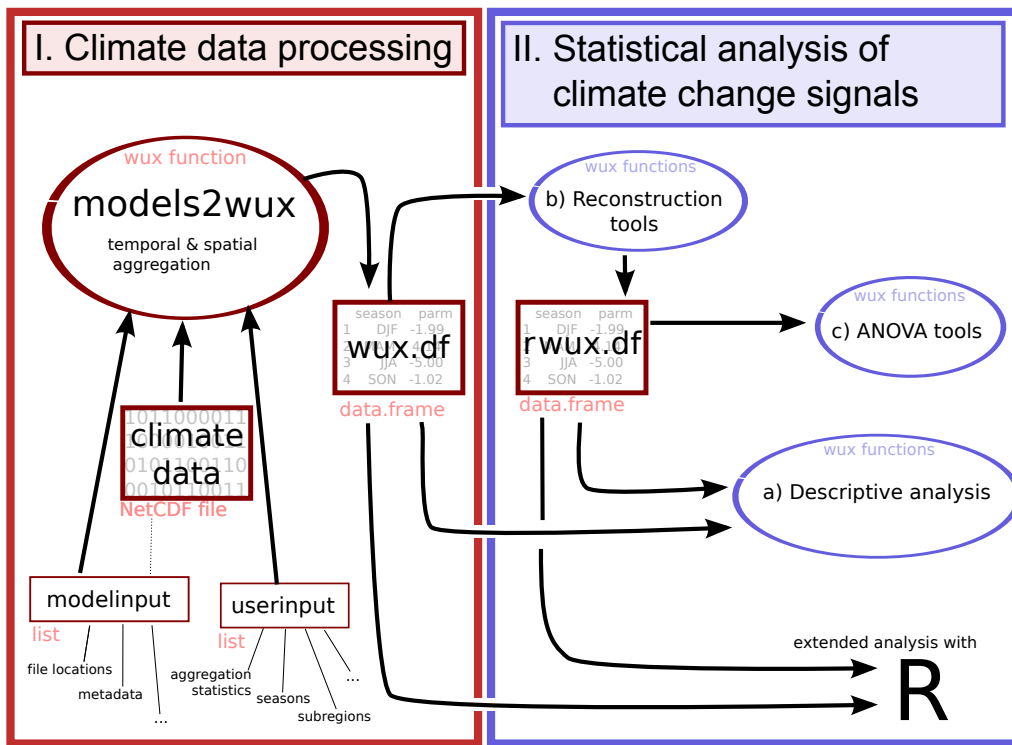
I. Climate Data Processing (Section 3)

Function	Input	Output	Description
<code>modelS2wux</code>	NetCDF files	<code>wux.df</code>	Reads NetCDF climate model output, processes it, and writes the results to a data frame which is the backbone of all further wux analyses.
<code>read.wux.table</code>	<code>wux.df</code> files	<code>wux.df</code>	Reads data frame files produced by <code>modelS2wux</code> .

II. Statistical Analysis of Climate Change Signals (Section 4)

Function	Input	Output	Description
a) Descriptive analysis (Section 4.1)			
<code>summary</code>	<code>wux.df/rwux.df</code>	summary statistics	Summary statistics of the wux data frame (<code>wux.df</code> object).
<code>plot</code>	<code>wux.df/rwux.df</code>	figure	Scatter plot
<code>plotAnnualCycle</code>	<code>wux.df/rwux.df</code>	figure	Annual cycle plot
<code>hist</code>	<code>wux.df/rwux.df</code>	figure	Density plot
b) Reconstruction tools (Section 4.2)			
<code>reconstruct</code>	<code>wux.df</code>	<code>rwux.df</code>	Filling missing values of an unbalanced climate model design matrix in order to avoid biased ensemble estimates. Currently, the underlying reconstruction technique is based on an ANOVA using various methods for estimation. Returns reconstructed <code>wux.data.frame</code> of class <code>rwux.df</code> .
c) Analysis of variance components (Section 4.2)			
<code>aovwux</code>	<code>rwux.df</code>	<code>wux.aov</code>	Extracts variance components of multiple climate model simulations using an ANOVA. Data must be balanced, so a reconstruction preprocessing is necessary.
<code>plot</code>	<code>wux.aov</code>	figure	Barchart for <code>aovwux</code> output.

Table 1: Most important functionalities of the **wux** package.

Figure 1: Basic functionalities of the **wux** package.

3. Climate data processing

The central role of the **wux** package is to automatically read in binary climate model output data from NetCDF files and process them to a data frame for statistical analysis. This task is performed by the function `models2wux`. The resulting data frame (further called `wux.df`, as it is technically a `wux.df` object) contains the climate change signals for user-specified periods, regions, seasons, and parameters for each of the climate models. One example `wux.df` is shown at the end of Section 3.1. Alternatively, also time series data can be obtained.

3.1. From climate model output to wux data frame

This is what `models2wux` is doing for each specified climate model:

1. Read in a three dimensional array (longitude, latitude, time) from binary climate model output.
2. *Temporal aggregation* of the fields according to user-specified climate periods and seasons. Aggregation statistics can also be specified by the user.
3. *Spatial aggregation* (arithmetic mean) over geographical domain.
4. Computing climate change signal for specified periods.

The resulting climate change signals for each climate model are returned to a data frame.

Temporal aggregation can be performed several times serially, going from fine temporal resolution to coarser resolution, each time using another statistic for aggregation. For example, daily temperature of a climate model output could first be aggregated to monthly resolution using the `mean` function and as a second step the warmest month in the year can be calculated with `max`. This would result in a climate change signal of the warmest monthly averages. We can thus calculate a vast amount of sufficient statistics to explore the climate data. Also, the user has the possibility to retrieve the full time-series of the climate model instead of the

climate change signal. This can, however, result in quite a large data frame. The lowest time resolution currently implemented for time-series data is on monthly basis.

Being able to flexibly perform spatial aggregation over a specified domain is one of the key strengths of this program. Several ways exist for the user to identify the region of interest. For example a rectangular region defined by the longitude-latitude corners can be specified. For more flexibility, polygons can be defined using ESRI shapefiles⁵ to cut out and aggregate over the desired subregion domain. The spatial aggregation is always performed using the arithmetic mean over geographical regions of any complexity. However, this process is not as trivial as it first may seem. One problem lies in the geographical projection of the climate model. Averaging over pixels of a model on a Mercator projection (angle preserving) will result in a different value than averaging over pixels in an area-preserving projection. GCMs usually do not come on an area-preserving projection. Therefore, the pixels should be weighted by the cosine of their latitudes, otherwise areas near the poles would gain much more weight than areas near the equator. When aggregating over a certain subregion, another problem arises from the gridpoints which are associated with the subregion. Instead of either considering a gridpoint to be within a region or not (0 and 1 weight), we may want to weight all the model cells that contribute even partly to the considered subregion, i.e. seize the fraction of the cell corresponding to the area covered by the subregion.

3.2. Setting up `models2wux`

To process a climate multi-model ensemble of your choice, `models2wux` needs two input arguments `userinput` and `modelinput`, each being a named list object or a file containing a named list.

`modelinput` stores general information about your climate data, i.e. the locations of the NetCDF files and their filenames. It also saves certain meta-information for the specific climate simulations (e.g. a unique acronym for the simulation, the developing institution, the radiative forcing). Usually the `modelinput` information should be stored in a single file on your system and should be updated when new climate simulations come in. It is advisable to share this file with your colleagues if you work with the same NetCDF files on a shared IT infrastructure.

The second input argument, `userinput`, defines which meteorological parameters of which climate simulations defined in `modelinput` should be analyzed. This is simply done by calling the models acronym, as all meta-information is already stored in the `modelinput` file. Also the geographical regions of interest and the temporal statistics are specified in this file. This file typically changes depending on the type of analysis performed.

3.3. Getting started

We explain `models2wux` in more detail by considering an example of a typical workflow for climate data processing. We start with downloading a couple of global climate simulations (GCMs) from the CMIP5 project (Taylor, Stouffer, and Meehl 2012), then we specify their meta-information and the output statistics and finally we run `models2wux` to process the binary data to an object of class `wux.df`.

To obtain CMIP5 climate simulations you can get started with downloading some example NetCDF files directly from an ESGF (Earth System Grid Federation) node⁶ or using the `CMIP5fromESGF` function from the **wux** package (Linux only).

```
> ## I) Load wux functions and example datasets...
> library("wux")

> ## II) obtain some climate simulations
> CMIP5fromESGF(save.to = "~/tmp/CMIP5/",
```

⁵<http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>

⁶e.g. from the data node <http://pcmdi9.llnl.gov>

```
models = c("NorESM1-M", "CanESM2"),
variables = c("tas", "pr"),
experiments= c("historical", "rcp85"))
```

Here, we download the 2m air temperature and surface precipitation files (`tas` and `pr`) from two simulations `NorESM1-M` and `CanESM2` for the `historical` period (here 1850–2005) and the future projection (2006–2100), assuming a strong change in future radiative forcing (`rcp85`, see Taylor *et al.* (2012)). The data will be downloaded into a temporary directory `~/tmp/CMIP5/` which can take a while. You need a valid account at any ESGF node for this function to run. In order to run `models2wux`, you need to specify the two input arguments explained above: A `modelinput` file to define which climate simulations you have on your hard-disk and a `userinput` file which controls `models2wux` itself. An example for the model specification can be obtained in the package itself:

```
> ## III) Meta-information on downloaded data for models2wux.
> data(modelinput_test)
> str(modelinput_test)
List of 2
 $ CanESM2-r1i1p1_rcp85 :List of 11
 ..$ rcm                : chr ""
 ..$ gcm                : chr "CanESM2"
 ..$ gcm.run            : num 1
 ..$ institute          : chr "CCCma"
 ..$ emission.scenario: chr "rcp85"
 ..$ file.path.alt      :List of 2
 .. ..$ air_temperature :List of 2
 .. .. ..$ historical    : chr "~/tmp/CMIP5/CanESM2/historical"
 .. .. ..$ scenario      : chr "~/tmp/CMIP5/CanESM2/rcp85"
 .. ..$ precipitation_amount:List of 2
 .. .. ..$ historical    : chr "~/tmp/CMIP5/CanESM2/historical"
 .. .. ..$ scenario      : chr "~/tmp/CMIP5/CanESM2/rcp85"
 ..$ file.name          :List of 2
 .. ..$ air_temperature :List of 2
 .. .. ..$ historical    : chr "tas_Amon_CanESM2_historical_r1i1p1_185001-200512.nc"
 .. .. ..$ scenario      : chr "tas_Amon_CanESM2_rcp85_r1i1p1_200601-210012.nc"
 .. ..$ precipitation_amount:List of 2
 .. .. ..$ historical    : chr "pr_Amon_CanESM2_historical_r1i1p1_185001-200512.nc"
 .. .. ..$ scenario      : chr "pr_Amon_CanESM2_rcp85_r1i1p1_200601-210012.nc"
 ..$ gridfile.path      : chr "~/tmp/CMIP5/CanESM2/historical"
 ..$ gridfile.filename: chr "tas_Amon_CanESM2_historical_r1i1p1_185001-200512.nc"
 ..$ resolution         : chr ""
 ..$ what.timesteps     : chr "monthly"
 $ NorESM1-M-r1i1p1_rcp85:List of 11
 ...
```

This input specifies the simulations which have just been downloaded. It is a named list with the name being an unique acronym of the climate simulation. The example input here specifies two simulations, but for the sake of brevity we only display the first one, being the `CanESM2-r1i1p1_rcp85` model. As this is a GCM, the `rcm` tag has no entry. The other tags specify the model in more detail: This simulation is run number 1 of the GCM `CanESM2` and has been developed by the `CCCma` institution⁷. The corresponding anthropogenic forcing is `rcp85`. `file.path.alt` defines the file locations for both temperature and precipitation files as well as for historical runs and future scenario projections. In this case the historical and the future scenario runs are located in different directories, whereas both meteorological parameters are saved in the same path. `file.name` gives information for the corresponding file names. The files which are necessary to define the geographical longitude and latitude information are specified in `gridfile.path` and `gridfile.filename`. The data is on a monthly timescale, which is defined in `what.timesteps` and the horizontal resolution is not specified here as it is optional.

It is advisable to store this list as a single file on your system. You should share this file with colleagues using the same IT infrastructure to use synergies. Such a file can also be

⁷Canadian Centre for Climate Modelling and Analysis (www.ec.gc.ca/ccmac-cccma)

created in an automated way using the function `CMIP5toModelinput`, for data obtained with `CMIP5fromESGF` (see the manual for more details).

Next, we want to tell `models2wux` to get climate change signals of both simulations we just defined above. In this example we are specifically interested in the temperature changes for the Alpine area at the end of the 21st century. Therefore we specify a user input file which contains a named list with all the necessary information:

```
> ## IV) Input argument controlling models2wux.
> data(userinput_CMIP5_changesignal)
> str(userinput_CMIP5_changesignal)
List of 9
 $ parameter.names      : chr "air_temperature"
 $ area.fraction        : logi TRUE
 $ reference.period     : chr "1971-2000"
 $ scenario.period      : chr "2071-2100"
 $ temporal.aggregation:List of 1
 ..$ stat.level.1:List of 3
 .. ..$ period      :List of 4
 .. .. ..$ DJF: chr [1:3] 12 1 2
 .. .. ..$ MAM: chr [1:3] 3 4 5
 .. .. ..$ JJA: chr [1:3] 6 7 8
 .. .. ..$ SON: chr [1:3] 9 10 11
 .. ..$ statistic   : chr "mean"
 .. ..$ time.series: logi FALSE
 $ subregions          :List of 1
 ..$ AL: num [1:4] 5 15 48 44
 $ plot.subregion      :List of 4
 ..$ save.subregions.plots: chr "/tmp/"
 ..$ xlim              : num [1:2] 0 20
 ..$ ylim              : num [1:2] 40 50
 ..$ cex                : num 10
 $ save.as.data        : chr "/tmp/wuxexample"
 $ climate.models      : chr [1:2] "CanESM2-r11p1_rcp85", "NorESM1-M-r11p1_rcp85"
```

The `userinput` argument tells `models2wux` to process `air_temperature` (`parameter.names`) for both models `CanESM2-r11p1_rcp85` and `NorESM1-M-r11p1_rcp85` (`climate.models` tag). We define our base period (tag `reference.period`) to be 1971–2000 and the projected future period of interest (tag `scenario.period`) for the climatic change to be 2071–2100. We want the data to be aggregated to seasons summer (June, July, August: `JJA`), autumn (`SON`), winter (`DJF`) and spring (`MAM`). For each of those seasons `models2wux` returns the climate change signal defined by the user by calculating `scenario.period` minus `reference.period` (for precipitation, changes are in addition calculated relative to `reference.period`). When setting the attribute `time.series` to `TRUE`, the output would be a transient time series instead of climate change.

We want to aggregate over the spatial extend of the Alpine area (`AL`, see [Christensen and Christensen \(2007\)](#)), which is defined in the `subregions` tag. Here it is a named vector of longitude and latitude coordinates and it defines a rectangular region (western, eastern, northern and southern coordinates of the corners). There are plenty of other ways to define a subregion, like reading in shapefiles. To analyze which model grid cells lie within the specified region, we can specify `plot.subregion` (see Figure 2). We usually want to aggregate all model cells which lie within the specified region, however, sometimes we would like to down-weight those cells which only partly contribute to the considered region. Setting `area.fraction` as `TRUE` weights the cells corresponding to the area covered by the subregion (Figure 2). Furthermore, `area.fraction=TRUE` is necessary, if the size of the subregion is in the same order of magnitude as the grid cell. Such cases should be handled with care, since the grid point interpretation of climate models is problematic. In most cases, the analyzed subregions should be much larger than the grid size of the models and the error produced by setting `area.fraction` to `FALSE` is negligible and processing gains a massive speed up. The data frame will also be saved as a comma-separated file to `/tmp/wuxexample`.

Finally we run `models2wux` with the input arguments explained above to obtain the temperature climate change signals (`delta.air_temperature`) for both simulations aggregated over the Alpine region and four seasons. Columns besides `subreg`, `season` and the temperature

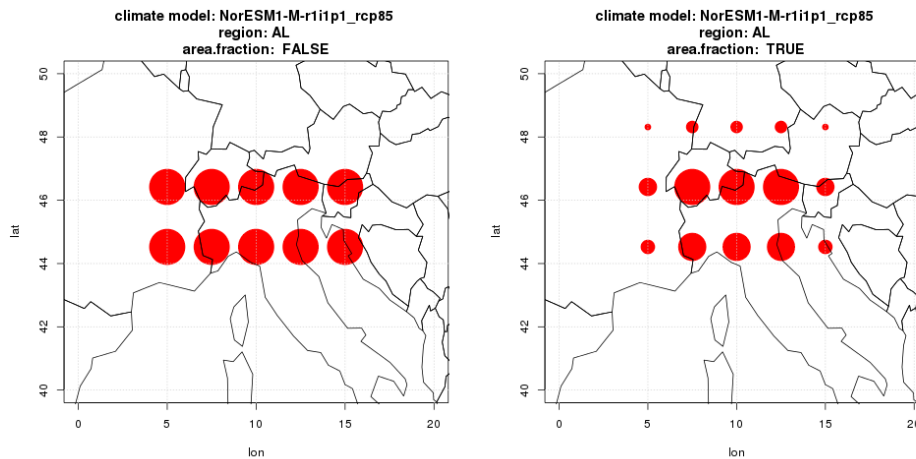


Figure 2: Grid cells of the NorESM1-M climate model being aggregated. On the left side `area.fraction` is switched off, taking all cells with their centroids lying within the AL region and weight them equally. The right figure has `area.fraction` on: The smaller the circles, the smaller the coverage of the model cells and the smaller their weight.

change parameter are meta-information of the climate data and derived from the `modelinput` input argument.

```
> ## V) Process NetCDF files
> climchange.df <- models2wux(userinput = userinput_CMIP5_changesignal,
>                             modelinput = modelinput_test)
> climchange.df
  subreg season      acronym institute      gcm gcm.run em.scn
1      AL   DJF CanESM2-r1i1p1_rcp85   CCCma CanESM2      1 rcp85
2      AL   JJA CanESM2-r1i1p1_rcp85   CCCma CanESM2      1 rcp85
3      AL   MAM CanESM2-r1i1p1_rcp85   CCCma CanESM2      1 rcp85
4      AL   SON CanESM2-r1i1p1_rcp85   CCCma CanESM2      1 rcp85
13     AL   DJF NorESM1-M-r1i1p1_rcp85    NCC NorESM1-M      1 rcp85
14     AL   JJA NorESM1-M-r1i1p1_rcp85    NCC NorESM1-M      1 rcp85
15     AL   MAM NorESM1-M-r1i1p1_rcp85    NCC NorESM1-M      1 rcp85
16     AL   SON NorESM1-M-r1i1p1_rcp85    NCC NorESM1-M      1 rcp85
  period ref.per resolution corrected delta.air_temperature
1 2071-2100      no          NA       no          4.066630
2 2071-2100      no          NA       no          8.041165
3 2071-2100      no          NA       no          4.261498
4 2071-2100      no          NA       no          5.686222
13 2071-2100      no          NA       no          3.336806
14 2071-2100      no          NA       no          5.378479
15 2071-2100      no          NA       no          3.922325
16 2071-2100      no          NA       no          3.787082
```

4. Statistical analysis of climate change signals

Several functions are available to analyze the processed climate change signals created by `models2wux`.

4.1. Descriptive analysis

The `summary` function gives a descriptive overview of the climate model ensemble which has been processed. On the one hand it calculates categorical statistics (counting climate models, emission scenarios, RCM-GCM cross-tables, ...) and on the other hand it returns statistics of continuous climate change signals (mean, standard deviation, coefficient of variation and quantiles) split by season, emission scenario, meteorological parameters and subregions. Let us consider the climate change signals from 1961–1990 until 2021–2050 in the Greater Alpine Region (GAR) of a multi-model ensemble consisting of 22 RCMs from the ENSEMBLES project (van der Linden and Mitchell 2009).

```

> ## VI b) Analyze climate change data - summary statistics
> data(ensembles)
> # consider Greater Alpine Region (GAR) only
> wuxtest.df <- droplevels(subset(ensembles, subreg == "GAR"))
> ## summary statistics
> summary(wuxtest.df)

```

```

-----
----- FREQUENCIES BY SCENARIO -----
-----
A1B:
8 GCMs (disregarding runs)
22 models total
Number of GCMs used:
      ARPEGE   BCCR-BCM2.0      CGCM3 ECHAM5/MPI-OM      HadCM3Q0
           3             3           1             5           5
      HadCM3Q16   HadCM3Q3   IPSL-CM4
           2             2           1
Number of RCM runs:
      CLM   CRCM   HIRHAM   HadRM3   PROMES   RACMO   RCA   RCA3   REMO   RM4.5   RM5.1
        2     1     5       3       1       1       3     1     1     1       1
      RRCM   RegCM
        1     1
Number of RCMs: 13

```

```

-----
----- CLIMATE MODEL STATISTICS BY SUBREGION -----
-----
----- GAR -----
perc.delta.precipitation_amount:
[A1B]
      n      mean    sd    coefvar min      max      med    q25    q75
DJF: 22      2.88   5.09    1.77   -8.96   10.25    3.81    1.54    5.8
JJA: 22     -2.82   6.87    2.44  -12.42   10.71   -3.7    -7.19    1.61
MAM: 22     -0.64   4.99    7.83   -9.41    6.61    0.7    -5.52    2.87
SON: 22      0.76   5.7     7.51  -12.16   12.46    0.77   -2.09    3.65

```

```

delta.air_temperature:
[A1B]
      n      mean    sd    coefvar min      max      med    q25    q75
DJF: 22      1.66   0.51    0.31    0.92    2.41    1.56    1.19    2.13
JJA: 22      1.7    0.65    0.38    0.47    2.79    1.88    1.31    2.18
MAM: 22      1.25   0.53    0.43   -0.02    2.26    1.21    0.91    1.55
SON: 22      1.57   0.55    0.35    0.61    2.88    1.64    1.27    1.8

```

For the sake of brevity, we do not show all parts of the output. The **FREQUENCIES** output shows that $n = 22$ climate simulations driven by 8 GCMs forced with one emission scenario (A1B) have been processed and shows the count of the specific RCMs and GCMs used in the analysis. The **CLIMATE MODEL STATISTICS** output shows a descriptive analysis of the continuous variables in the data set based on all $n = 22$ climate simulations available. In this case the continuous variables are the relative change of precipitation (**perc.delta.precipitation_amount**) in percent and the absolute change of temperature (**delta.air_temperature**) in °C. The precipitation change in the GAR is not significant for either season, but there is a tendency in DJF for a slight increase of total precipitation. In contrast to that, the change signal for temperature is significant for all seasons showing quite an uniform warming, where MAM seems to have the smallest trend.

Also, functions for a graphical overview of the climate model ensemble are available in **wux**. The method **plot** for a **wux.df** object draws one or more scatterplots containing climate change signals of selected meteorological parameters.

```

> ## VI b) Analyze climate change data - scatterplots
> plot(ensembles, "perc.delta.precipitation_amount",
>      "delta.air_temperature", boxplots = TRUE,
>      xlim = c(-40,40), ylim = c(0, 4),
>      xlab = "Precipitation Amount [%]", ylab = "2-m Air Temperature [K]",
>      main = "Scatterplot", subreg.subset = c("GAR"))

```

This draws a simple scatterplot which accounts for certain meta-information of the climate change data frame and allows to highlight certain models. One of the scatterplots produced by this call is shown on the left side of Figure 3. This is a very useful plot as it gives a

good overview on the model behavior and the climate change uncertainty. In our example, some models project an increase in precipitation change, whereas some project a decline. No correlation between temperature and precipitation change is visible on this small spatial scale.

4.2. Data reconstruction methods

Due to limited computational capacities, even in large-scale climate modeling projects such as CMIP5 or CORDEX (Jacob, Petersen, Eggert, Alias, Christensen, Bouwer, Braun, Collette, Déqué, Georgievski, Georgopoulou, Gobiet, Menut, Nikulin, Haensler, Hempelmann, Jones, Keuler, Kovats, Kröner, Kotlarski, Kriegsmann, Martin, Meijgaard, Moseley, Pfeifer, Preuschmann, Radermacher, Radtke, Rechid, Rounsevell, Samuelsson, Somot, Soussana, Teichmann, Valentini, Vautard, Weber, and Yiou 2013) only a limited number of climate simulations can be realized and it is a question of the experimental design which uncertainty components are primarily tackled within the ensemble. Therefore, missing realizations within climate projection ensembles are a common problem and even simple ensemble estimates such as mean and variability for e.g. temperature changes are potentially biased due to unequal sampling of the uncertainty components. In order to avoid such biases, Déqué, Rowell, Lüthi, Giorgi, Christensen, Rockel, Jacob, Kjellström, Castro, and Hurk (2007) introduced an iterative data reconstruction method which assumes additivity between uncertainty components in order to estimate the missing climate change signals. This reconstruction method was further applied in several studies in order to obtain a balanced design for the analysis of variance components (Déqué *et al.* 2007; Heinrich, Gobiet, and Mendlik 2014; Prein, Gobiet, and Truhetz 2011; Déqué, Somot, Sanchez-Gomez, Goodess, Jacob, Lenderink, and Christensen 2011; Mendlik and Gobiet 2015). In **wux**, we implemented the method of Déqué *et al.* (2007) for a two-factorial design (**reconstruct**) such as realized in the ENSEMBLES project (van der Linden and Mitchell 2009). In ENSEMBLES, a set of 21 high resolution RCM simulations with a horizontal grid spacing of about 25 km was produced. The ensemble consists of 8 GCMs and 16 RCMs only forced by the A1B emission scenario, but due to limited computational resources, only a small fraction (16.4 % of the possible GCM-RCM combinations) could be realized. The result of such a reconstruction is shown in Figure 3. In that case, filling up the missing GCM-RCM combinations does not alter the distribution of temperature and precipitation change. However, as the method relies on an implicit formulation of the uncertainty components, it cannot be used to extend the ensemble to GCMs that have not been used as driver for any RCM in the ensemble. Further reconstruction methods which are able to extend the ensemble to GCMs outside of the original design are investigated in Heinrich *et al.* (2014).

5. Example: Further statistical analysis

It is one of the key strengths of this package to be directly implemented in R and for that reason to have direct access to a huge magnitude of statistical methods to analyze climate data. We provide an example application in this section to show possible extensions based fully on the **wux.df**. We use a linear mixed effects model from the **lme4** package (Bates, Mächler, Bolker, and Walker 2015) to estimate the average summer temperature trend over the Greater Alpine Region based on individual time-series of 16 GCMs from the CMIP5 ensemble under a moderate stabilization scenario (RCP 4.5).

To generate the appropriate **wux.df**, the **timeseries** tag in the **userinput** file was set **TRUE** (see Section 3). The aim here is to get an average linear trend while accounting for the unbalanced model design. Several of the GCMs were run a couple of times (up to 10 times) with different initial conditions, which induces a dependency structure in the data set. We assess for this dependency by putting random effects in the linear model:

$$Y_{ijk} = \beta_0 + \beta_1 \text{year}_{jk} + b_{0i} + b_{1i} \text{year}_{jk} + \epsilon_{ijk}$$

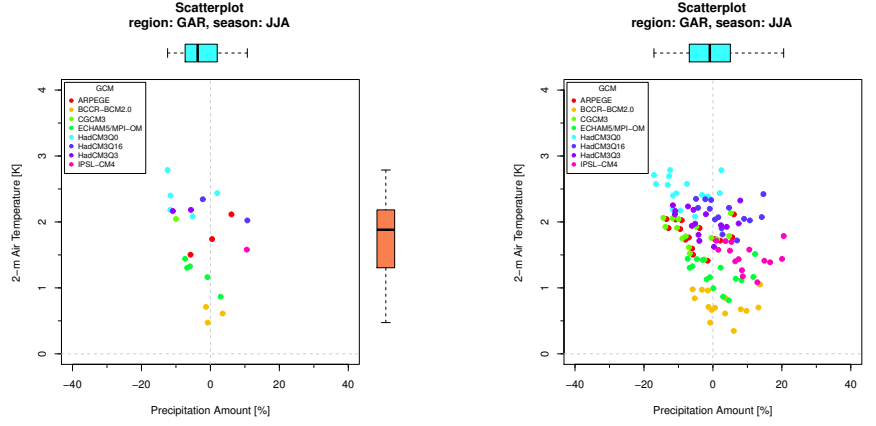


Figure 3: Projected changes of summer precipitation and temperature of the ENSEMBLES models from 1961–1990 to 2021–2050 in the Greater Alpine Region. The left plot shows the originally available 22 RCMs, whereas the right plot depicts a reconstructed dataset filled up with the function `reconstruct`.

where Y_{ijk} is the average summer temperature projected by $i = 1, \dots, 16$ GCMs with $j = 1, \dots, n_i$ runs per GCM and $k = 1, \dots, 130$ yearly time steps. The random effects are defined as

$$\begin{pmatrix} b_{0i} \\ b_{1i} \end{pmatrix} \sim N \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{gcm}^2 & 0 \\ 0 & \sigma_{gcm.t}^2 \end{pmatrix} \right) \quad \text{and} \quad \epsilon_{ijk} \stackrel{iid}{\sim} N(0, \sigma_y^2).$$

We use the `lmer` function from the **lme4** package for our analysis to estimate the fixed effects $\hat{\beta}_0, \hat{\beta}_1$ and to predict the individual random effects $\hat{b}_0 = (\hat{b}_{0,1}, \dots, \hat{b}_{0,16})'$, $\hat{b}_1 = (\hat{b}_{1,1}, \dots, \hat{b}_{1,16})'$. The time-series data and the trends are shown in Figure 4 plotted with the **lattice** package (Sarkar 2008).

```
> data(alpinesummer)
> ## pick just a few GCMs for this example - for a more compact display
> gcms.sub <- c("ACCESS1-3", "BCC-CSM1-1", "CESM1-CAM5", "CMCC-CM",
+              "CNRM-CM5", "CSIRO-Mk3-6-0", "EC-EARTH", "FGOALS-g2",
+              "GFDL-CM3", "HadGEM2-ES", "INM-CM4", "IPSL-CM5A-LR",
+              "MIROC5", "MPI-ESM-LR", "MRI-CGCM3", "NorESM1-M")
> alpinesummer.sub <- droplevels(subset(alpinesummer, gcm %in% gcms.sub))
> ## transform for better convergence
> alpinesummer.sub$time <- alpinesummer.sub$year - 1971
> lmm.fit <- lmer(air_temperature ~ 1 + time + (1 | gcm) + (0 + time | gcm),
+               data = alpinesummer.sub)
> summary(lmm.fit)
Linear mixed model fit by REML ['lmerMod']
Formula: air_temperature ~ 1 + time + (1 | gcm) + (0 + time | gcm)
Data: alpinesummer.sub

REML criterion at convergence: 16472.2

Scaled residuals:
    Min       1Q   Median       3Q      Max
-4.0410 -0.6150 -0.0321  0.5766  4.5612

Random effects:
Groups   Name             Variance Std.Dev.
gcm      (Intercept)    2.5671124  1.60222
gcm.1    time           0.0001318  0.01148
Residual             1.2482244  1.11724
Number of obs: 5330, groups: gcm, 16

Fixed effects:
              Estimate Std. Error t value
(Intercept)  16.49168    0.40257   40.97
time          0.03443    0.00292   11.79
```

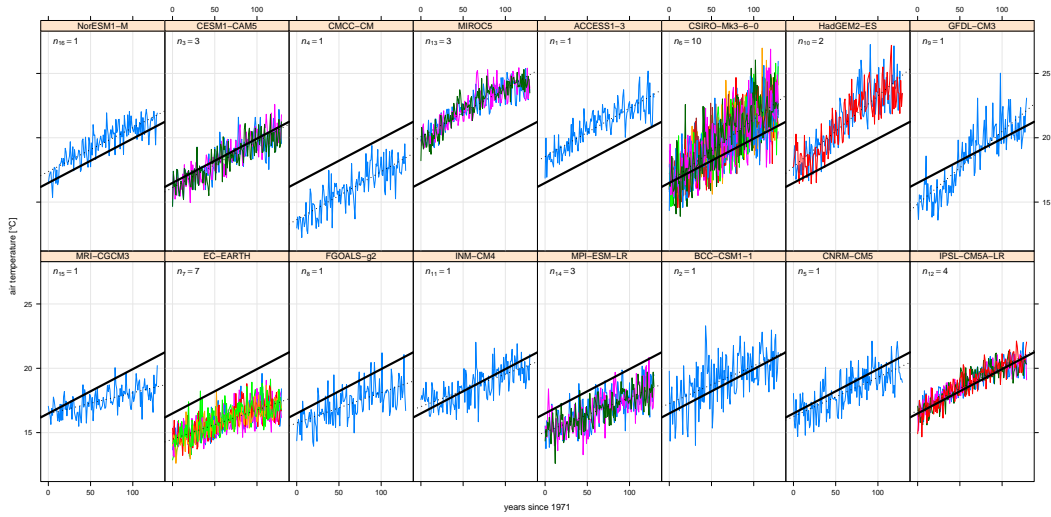


Figure 4: Time-series of GCMs from the CMIP5 ensemble for summer temperature in the Alpine region. The estimated average trend $\hat{\beta}_1$ is shown as a bold line, the predicted random effects trends are shown as a dashed line. The simulations are ordered from low trend (lower left panel) to high trend (upper right panel).

```
Correlation of Fixed Effects:
(Intr)
time -0.016
> ## prints the first random effects
> head(coef(lmm.fit)$gcm)
              (Intercept)          year
ACCESS1-3      18.53855  0.03755274
BCC-CSM1-1     17.26063  0.02928145
CESM1-CAM5     16.11973  0.03574971
CMCC-CM        13.62953  0.03733811
CNRM-CM5       16.25376  0.03042184
CSIRO-Mk3-6-0  16.55908  0.04872848
```

The average slope $\hat{\beta}_1 = 0.34^\circ\text{C}/\text{decade}$ ($0.034^\circ\text{C}/\text{y}$) is highly significant and the individual slopes of the GCMs reach from slowly warming simulations $\hat{b}_{1,1} = 0.16^\circ\text{C}/\text{decade}$ to very sensitive simulations $\hat{b}_{1,16} = 0.56^\circ\text{C}/\text{decade}$ assuming linear temperature evolution over 130 years from 1971–2100. The residual standard deviation is $\hat{\sigma}_y = 1.12^\circ\text{C}$, which in this case can be interpreted as the average year-to-year natural variability.

6. Conclusion

It is crucial in climate research not only to analyze outcomes of single climate models, but to consider entire multi-model ensembles, as it is virtually demanded in every climate impact related study to assess the associated uncertainties of the projected changes. There is, however, definitely a technical challenge to process large amount of climate simulations at once and not many tools exist to assess this problem. Another more general problem arises from the measure of uncertainty in multi-model ensembles. It is somewhat uncomfortable to make statistical inference on multi-model ensembles, as they do not stem from a designed experiment (Knutti *et al.* 2010), are utterly unbalanced (Déqué *et al.* 2007), and are known to be biased (Maraun *et al.* 2010; Themeßl, Gobiet, and Leuprecht 2011).

The focus here is not to show solutions for sophisticated statistical analyses of climate datasets, but merely to present a flexible and easy-to-use tool which is able to pre-process the datasets for further statistical analysis. This way, the user can focus on solving the grand challenges of statistical inference of multi-model datasets and does not need to spend valuable resources on technical data issues. The function `models2wux` fulfills exactly this task by processing magnitudes of binary climate model data to a R data frame of climate change signals. Subse-

quently, the user can take advantage of the vast amount of methods available in R, to analyze this data set.

However, this package also provides some functions for a first exploratory data analysis, as e.g. a **summary** function and some plotting routines. Such simple analysis provide very valuable information on the multi-model ensemble. In addition, we also provide a couple of methods to address the issue of unbalanced experiment designs. Several methods from literature are implemented to fill up the incomplete data matrix (Déqué *et al.* 2007; Heinrich *et al.* 2014).

It should be kept in mind, that also other software packages exist which partly fulfill similar tasks (e.g. climate explorer, CDO, NCL). The climate explorer can be a very convenient way to have a quick descriptive analysis of a multi-model ensemble. It is easy to use, but it is also restricted to a non-programming environment. Also, one can analyze only models which are implemented in the system, and the statistical methods are restricted as well. It should be noted, that no spatial analysis is currently possible within **wux**, as the emphasize lies on averaged domains. For spatial maps, tools as CDO or NCL are far better suited. Another limitation can be the hardware needed to process large datasets. R is not the most memory-efficient environment and one can run into trouble when reading climate simulations with a very high spatial resolution.

To sum it up, **wux** is a very flexible tool dealing with different aspects of climate model uncertainty in climate change impact investigations and enables a quick analysis of climate scenario uncertainty, which typically demands a considerable technical effort as well as fundamental knowledge about climate modeling. It can be used to achieve a quick overview on the involved uncertainties to identify the most important sources of uncertainty or to select representative sub-ensembles to be used as input for impact studies. **wux** is fully flexible regarding the meteorological parameter and region under consideration and is able to assess uncertainties based on multiple user-defined parameters.

Acknowledgments

We acknowledge the World Climate Research Programme’s Working Group on Coupled Modelling, which is responsible for CMIP, and we thank the climate modeling groups for producing and making available their model output. For CMIP the U.S. Department of Energy’s Program for Climate Model Diagnosis and Intercomparison provides coordinating support and led development of software infrastructure in partnership with the Global Organization for Earth System Science Portals. The ENSEMBLES data used in this work was funded by the EU FP6 Integrated Project ENSEMBLES (Contract number 505539) whose support is gratefully acknowledged.

References

- Bates D, Mächler M, Bolker B, Walker S (2015). “Fitting Linear Mixed-Effects Models Using **lme4**.” *Journal of Statistical Software*, **67**(1), 1–48. doi:10.18637/jss.v067.i01.
- Bishop CH, Abramowitz G (2012). “Climate model dependence and the replicate Earth paradigm.” *Climate Dynamics*, pp. 1–16. doi:10.1007/s00382-012-1610-y.
- Chandler RE (2013). “Exploiting strength, discounting weakness: combining information from multiple climate simulators.” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **371**(1991). doi:10.1098/rsta.2012.0388.
- Christensen JH, Christensen OB (2007). “A summary of the PRUDENCE model projections of changes in European climate by the end of this century.” *Climatic Change*, **81**(1), 7–30. ISSN 0165-0009. doi:10.1007/s10584-006-9210-7.

- Collins M, Chandler RE, Cox PM, Huthnance JM, Rougier J, Stephenson DB (2012). “Quantifying future climate change.” *Nature Climate Change*, **2**(6), 403–409. doi: [10.1038/nclimate1414](https://doi.org/10.1038/nclimate1414).
- Déqué M, Rowell DP, Lüthi D, Giorgi F, Christensen JH, Rockel B, Jacob D, Kjellström E, Castro M, Hurk B (2007). “An intercomparison of regional climate simulations for Europe: assessing uncertainties in model projections.” *Climatic Change*, **81**(S1), 53–70. doi: [10.1007/s10584-006-9228-x](https://doi.org/10.1007/s10584-006-9228-x).
- Déqué M, Somot S, Sanchez-Gomez E, Goodess CM, Jacob D, Lenderink G, Christensen OB (2011). “The spread amongst ENSEMBLES regional scenarios: regional climate models, driving general circulation models and interannual variability.” *Climate Dynamics*. doi: [10.1007/s00382-011-1053-x](https://doi.org/10.1007/s00382-011-1053-x).
- Fischer AM, Weigel AP, Buser CM, Knutti R, Künsch HR, Liniger MA, Schür C, Appenzeller C (2012). “Climate change projections for Switzerland based on a Bayesian multi-model approach.” *International Journal of Climatology*, **32**(15), 2348–2371. ISSN 1097-0088. doi: [10.1002/joc.3396](https://doi.org/10.1002/joc.3396).
- Giorgi F, Mearns LO (1991). “Approaches to the simulation of regional climate change: A review.” *Reviews of Geophysics*, **29**(2), 191–216. ISSN 1944-9208. doi: [10.1029/90RG02636](https://doi.org/10.1029/90RG02636).
- Heinrich G, Gobiet A, Mendlik T (2014). “Extended regional climate model projections for Europe until the mid-twentyfirst century: combining ENSEMBLES and CMIP3.” *Climate Dynamics*, **42**(1-2), 521–535. doi: [10.1007/s00382-013-1840-7](https://doi.org/10.1007/s00382-013-1840-7).
- Jacob D, Petersen J, Eggert B, Alias A, Christensen OB, Bouwer LM, Braun A, Colette A, Déqué M, Georgievski G, Georgopoulou E, Gobiet A, Menut L, Nikulin G, Haensler A, Hempelmann N, Jones C, Keuler K, Kovats S, Kröner N, Kotlarski S, Kriegsmann A, Martin E, Meijgaard E, Moseley C, Pfeifer S, Preuschmann S, Radermacher C, Radtke K, Rechid D, Rounsevell M, Samuelsson P, Somot S, Soussana JF, Teichmann C, Valentini R, Vautard R, Weber B, Yiou P (2013). “EURO-CORDEX: new high-resolution climate change projections for European impact research.” *Regional Environmental Change*, pp. 1–16. ISSN 1436-3798. doi: [10.1007/s10113-013-0499-2](https://doi.org/10.1007/s10113-013-0499-2).
- Kang EL, Cressie N, Sain SR (2012). “Combining outputs from the North American Regional Climate Change Assessment Program by using a Bayesian hierarchical model.” *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **61**(2), 291–313. ISSN 1467-9876. doi: [10.1111/j.1467-9876.2011.01010.x](https://doi.org/10.1111/j.1467-9876.2011.01010.x).
- Knutti R, Furrer R, Tebaldi C, Cermak J, Meehl GA (2010). “Challenges in Combining Projections from Multiple Climate Models.” *Journal of Climate*, **23**. doi: [10.1175/2009JCLI3361.1](https://doi.org/10.1175/2009JCLI3361.1).
- Maraun D, Wetterhall F, Ireson AM, Chandler RE, Kendon EJ, Widmann M, Brienen S, Rust HW, Sauter T, Themeßl M, Venema VKC, Chun KP, Goodess CM, Jones RG, Onof C, Vrac M, Thiele-Eich I (2010). “Precipitation downscaling under climate change: Recent developments to bridge the gap between dynamical models and the end user.” *Reviews of Geophysics*, **48**(3), 1–34. doi: [10.1029/2009RG000314](https://doi.org/10.1029/2009RG000314).
- Mendlik T, Gobiet A (2015). “Selecting climate simulations for impact studies based on multivariate patterns of climate change.” *Climatic Change*. Submitted.
- Mendlik T, Heinrich G, Leuprecht A (2015). *wux: Wegener Center Climate Uncertainty Explorer*. R package version 1.2-3, URL <http://CRAN.R-project.org/package=wux>.
- Pirtle Z, Meyer R, Hamilton A (2010). “What does it mean when climate models agree? A case for assessing independence among general circulation models.” *Environmental Science*, **13**(5), 351 – 361. ISSN 1462-9011. doi: [10.1016/j.envsci.2010.04.004](https://doi.org/10.1016/j.envsci.2010.04.004).

- Prein AF, Gobiet A, Truhetz H (2011). “Analysis of uncertainty in large scale climate change projections over Europe.” *Meteorologische Zeitschrift*, **20**(4), 383–395. doi:[10.1127/0941-2948/2011/0286](https://doi.org/10.1127/0941-2948/2011/0286).
- Rougier J, Goldstein M, House L (2013). “Second-Order Exchangeability Analysis for Multimodel Ensembles.” *Journal of the American Statistical Association*, **108**(503), 852–863. doi:[10.1080/01621459.2013.802963](https://doi.org/10.1080/01621459.2013.802963).
- Sarkar D (2008). *lattice: Multivariate Data Visualization with R*. Springer, New York. ISBN 978-0-387-75968-5.
- Smith RL, Tebaldi C, Nychka D, Mearns LO (2009). “Bayesian modeling of uncertainty in ensembles of climate models.” *Journal of the American Statistical Association*, **104**(485), 97–116. doi:[10.1198/jasa.2009.0007](https://doi.org/10.1198/jasa.2009.0007).
- Stephenson DB, Collins M, Rougier JC, Chandler RE (2012). “Statistical problems in the probabilistic prediction of climate change.” *Environmetrics*, **23**(5), 364–372. ISSN 1099-095X. doi:[10.1002/env.2153](https://doi.org/10.1002/env.2153).
- Taylor KE, Stouffer RJ, Meehl GA (2012). “An Overview of CMIP5 and the Experiment Design.” *Bulletin of the American Meteorological Society*, **93**(4), 485–498. ISSN 0003-0007. doi:[10.1175/BAMS-D-11-00094.1](https://doi.org/10.1175/BAMS-D-11-00094.1).
- Tebaldi C, Knutti R (2007). “The use of the multi-model ensemble in probabilistic climate projections.” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **365**(1857), 2053–2075. doi:[10.1098/rsta.2007.2076](https://doi.org/10.1098/rsta.2007.2076).
- Thiemeßl M, Gobiet A, Leuprecht A (2011). “Empirical-statistical downscaling and error correction of daily precipitation from regional climate models.” *International Journal of Climatology*, **31**(10), 1530–1544. ISSN 1097-0088. doi:[10.1002/joc.2168](https://doi.org/10.1002/joc.2168).
- van der Linden P, Mitchell JFB (2009). *ENSEMBLES: Climate Change and its Impacts: Summary of research and results from the ENSEMBLES project*. Met Office Hadley Centre.
- van Oldenborgh GJ, Drijfhout S, van Ulden A, Haarsma R, Sterl A, Severijns C, Hazeleger W, Dijkstra H (2009). “Western Europe is warming much faster than expected.” *Climate of the Past*, **5**(1), 1–12. doi:[10.5194/cp-5-1-2009](https://doi.org/10.5194/cp-5-1-2009).
- Zender CS (2008). “Analysis of Self-describing Gridded Geoscience Data with netCDF Operators (NCO).” *Environmental Modelling & Software*, **23**(10), 1338–1342. doi:[10.1016/j.envsoft.2008.03.004](https://doi.org/10.1016/j.envsoft.2008.03.004).

Affiliation:

Thomas Mendlik
 Wegener Center for Climate and Global Change
 University of Graz
 Brandhofgasse 5, 8010 Graz, Austria
 E-mail: thomas.mendlik@uni-graz.at

The Software Environment R for Official Statistics and Survey Methodology

Matthias Templ
TU WIEN &
Statistics Austria

Valentin Todorov
UNIDO

Abstract

The open-source programming language and software environment R is currently one of the most widely used and popular software tools for statistics and data analysis. This contribution provides an overview of important R packages used in official statistics and survey methodology and discusses the usefulness of R in the daily work of a statistical office. Examples of activities and developments in R related projects in several national and international statistical offices are given. The focus is not only on the internal infrastructure that national and international statistical offices provide for using R but also on some interesting R related projects carried out in those institutes. Two particular packages (laeken and sdcMicro) and one data set (Statistics on Earnings Survey) are used to illustrate the usefulness (and the user-friendliness) of R and to present methods available in R. In addition, the access to international statistical databases like WDI of World Bank and UN COMTRADE with R is illustrated.

Keywords: official statistics, survey methodology, R.

1. Some general statements on R

The R Core Team (<http://www.r-project.org/>) defines R as an environment rather than a statistical system or programming language. R is an integrated suite of software facilities for data manipulation, calculation and graphical display, which includes:

- a suite of operators for calculations on arrays, mostly written in C and integrated in R;
- comprehensive, coherent and integrated collection of methods for data analysis;
- can be extended with (add-on) packages;
- graphical facilities for data analysis and display, either on-screen or in hard copy. It features trellis graphics (Sarkar 2008) and an implementation of the grammar of graphics book (Wilkinson and Wills 2005; Wickham 2009);
- a well-developed, simple and effective programming language which includes conditional statements, loops, user-defined recursive functions and input and output facilities;
- a flexible object-oriented system facilitating code reuse;

- high performance computing with interfaces to compiled code and facilities for parallel and grid computing;
- an environment that allows communication with many other software tools.

Each R package provides a structured standard documentation including code application examples. Further documents (so called vignettes [Leisch 2003](#)) are also available showing more applications of the packages and illustrating dependencies between the implemented functions and methods.

R has become an essential tool for statistics and data science ([Godfrey 2013](#)). Also companies in the area of social media (Google, Facebook, Twitter, Mozilla Corporation), in the banking world (Bank of America, ANZ Bank, Simple), in the food and pharmaceutical area (FDA, Merck, Pfizer), finance (Lloyd, London, Thomas Cook), technology companies (e.g. Revolution Analytics as subsidiary of Microsoft), car construction and logistic companies (Ford, John Deere, Uber), newspapers (The New York Times, New Scientist), and companies in many other areas use R in a professional context (see also [Gentlemen 2009](#); [Tippmann 2015](#)).

Outline of the paper

This contribution focuses on the use of R in national and international statistical organizations, giving the readers an overview of the usefulness of R in the area of official statistics. In Section 2 we first give some impression about systems and software used in daily work at the statistical office. This shows which variety of software is used in the daily work and how R could serve as a mediator between and within these software products.

Survey methodology/statistics is one large part of official statistics. Typically, the data in this area are of complex nature, having hierarchical structures and sampled from finite populations using complex sampling designs. Due to non-response, measurement errors and additional available information, many of the methods are focused on pre-processing of the data while the main part of other methods are related to data summarization and dissemination. A summary of available tools in different topics in survey methodology is therefore given in Section 3, which enhances the CRAN Task View on Official Statistics that is maintained by the authors of this paper.

Having presented a list of useful packages, the use of these packages and the strategy of using R in national statistical offices as well as in international statistical offices are presented in Section 4. This should give the readers an impression of the current status of the usage of R and ideas how to integrate R into the production system. However, it is out of scope to give a complete picture of the usage of R in each institution. Therefore a sample of selected offices is taken and reported.

Two examples of applying R on real data from official statistics cannot show all the useful packages and functions in the area of official statistics, but they show how efficiently R can be used for a specific task. In the first example of Section 5, the estimation of indicators is in focus while the second example shows the application of statistical disclosure control methods. Both topics are of main interest in official statistics. In Section 5.3 the access to international databases is shown. For the access to the world development indicators an R package is used while JavaScript object notation (JSON) format is used to access data from UN COMTRADE.

2. R in the statistical office

National and international statistical offices' main responsibility is to collect and publish empirical information about our society and economy, which become an important economic and social factor. The huge information requirements of our society have led to the development of sophisticated methods to collect, process, analyze and supply information. It is the role of national statistical offices to provide reliably collected and expertly analyzed political, social

and economic information. This information provides a basis for political decision-making; Its application and use for this purpose has become of increasing importance for the general public.

For data processing and data analysis in national or international statistical organization, several well-established statistical software packages are often available (see also [Todorov and Templ 2012](#)):

- (i) **SAS**[®] because of its traditional position in these organizations (if the necessary funds are available), its ability to handle large data sets and its availability on any platform, including mainframes;
- (ii) **SPSS**[®] is considered user friendly because of its point-and-click interface (albeit still providing the so-called syntax);
- (iii) **Stata**[®] is likely the most cost-effective among the three and, in terms of its design, is particularly suitable for handling data generated from complex survey designs (as is the case in many NSOs).

However, if the objective is

- flexibility in reading, manipulating and writing data,
- availability of recent statistical methodology,
- versatile presentation capabilities for generating tables and graphics which can readily be used in text processing systems such as \LaTeX (or Microsoft Word),
- creating dynamical reports using, e.g., **rmarkdown** ([Allaire, McPherson, Xie, Wickham, Cheng, and Allen 2014](#)), **Sweave** ([Leisch and Rossini 2003](#)), **brew** ([Horner 2011](#)) or the **knitr** ([Xie 2013](#)) package,
- to build web-based applications using **shiny** ([RStudio Inc. 2014](#)), and last but not least,
- a particularly economical solution,

R is the answer (see also [Todorov and Templ 2012](#)). The integration of all types of modern tools for scientific computing, programming and management of data and files into one environment is possible. Such an environment combines the capacities of R with editors that allow syntax highlighting and code completion, the use of modern version control systems for code and file management or a modern document markup languages such as \LaTeX or Mark-down, or interfaces to general purpose programming languages such as C, C++ or Java as well as easy-to-use (automatic) connections to powerful workstations. There exist editors that provide a complete programming environment for R. For example, **eclipse** with the extension **STATET**, an eclipse interface for R (see <http://www.walware.de/goto/statet>) or the modified eclipse IDE from *Open Analytics* called **Architect** (<http://www.openanalytics.eu/architect>), provide not only syntax highlighting, a defined project philosophy and interaction with R, but also integrate C++, Java, \LaTeX , **Sweave**, Subversion, server-connection facilities and many more. A very popular IDE for R nowadays is **RStudio** (see <http://support.rstudio.com>), which includes these features and additionally includes an integration of the packages **shiny** and **rmarkdown** (and related tools for creating slides in HTML – *RPresentation*), i.e. it provides a modern scientific computing environment, well designed and easy to use.

Although R has powerful statistical analysis features, data manipulation tools and versatile presentation capabilities, it has not yet become a standard statistical package in national and international statistical offices. This is mainly attributable to the widespread opinion that R is difficult to learn and has a very steep learning curve, which is true for learners without any programming skills. However, GUI's which display the underlying produced code are available as well ([Fox 2005](#)), and interfaces to all popular GUI toolkits are available.

In the daily work of a national or international statistical organization, for example, at Statistics Austria or at UNIDO, different systems for data analysis and data processing are used. Data exchange between statistical systems (like SAS®, SPSS®, EViews, Stata®, Microsoft Excel), database systems (like Microsoft Access, MySQL, IBM DB2) or output formats (like HTML, XML) is often required. Also note that the importance of statistical data and meta-data exchange format (like SDMX) is continuously growing. In this respect, R offers very flexible import and export interfaces either through its base installation or through add-on packages which are available from CRAN. For example, the packages **XML** (Temple Lang 2013) and **xml2** (Wickham 2015b) allow to read XML files. For importing delimited files, fixed width files and web log files it is worth mentioning the package **readr** (Wickham and Francois 2015) which is supposed to be faster than the available functions in base R. See also the **fread** function from package **data.table** (Dowle, Short, Lianoglou, and Srinivasan 2014) that is also substantially faster than **read.csv** from base R. The packages **XLConnect** (Mirai Solutions GmbH 2015) and **readxl** (Wickham 2015a) import Microsoft Excel files (.xls and .xlsx) into R. The packages **foreign** (R Core Team 2015) and a newer promising package called **haven** (Wickham and Miller 2015) allow to read SPSS®, Stata® and SAS® files from within R.

The connection to all major database systems is easily established with the packages **ROracle**, **RMySQL**, **RSQLite**, **RmSQL**, **RPgSQL**, **RODBC** and **RJDBC**. The **DBI** package provides an extra abstraction layer, used by the former mentioned packages. The integration of other statistical systems is made possible through packages like **RWeka**, **X12** and **RExcel**, while data exchange among these systems is facilitated by the package **foreign**. For more specific applications like web page generation or data and metadata exchange between organizations, the packages **R2HTML**, **sdmxr** or **RSDMX** can be used.

Data manipulation – in general but in any case with large data – can be best done with the package **dplyr** (Wickham and Francois 2014) or the **data.table** (Dowle *et al.* 2014) package. Functions for data manipulation in **dplyr** are implemented in C++ and thus the computational speed is often much faster than the data manipulation functions of the base packages. The syntax of **dplyr** is easy to learn and it is possible to write **dplyr** syntax in *data pipelines* that is internally provided by package **magrittr** (Bache and Wickham 2014). **data.table** (consisting of C code) even slightly outperforms **dplyr** but the syntax is not easy to learn, since especially indexing is done differently from the base packages.

In the case of large data files which exceed available RAM, interfaces to (relational) database management systems might be useful. The large data set problem can also be resolved by using either the **filehash**, **LaF**, **ff**, **ffbase** or **bigmemory** packages or by connecting to powerful workstations with **Rserve** (see <http://www.rforge.net/Rserve/>). Note that the RStudio server enables you to provide a browser based interface (the RStudio IDE) to a version of R running on a remote Linux server. This has some advantages such as to access your R workspace from any location, to allow multiple users to access powerful hardware and having installation of R packages and related features centralized.

R is designed to allow for parallel computing using multiple cores or CPU's and the recommended package for this task is the R package **parallel**. Several other packages can also be used for parallel computing, such as **snow** or **foreach**. Note that for medium or large data sets, parallel computing can be very slow on Microsoft Windows machines since the *fork* system call is only available on POSIX compliant platforms (e.g. Unix based operating systems). There exist user friendly interfaces for integrating compiled code (for example, the package **Rcpp**, **inline** and **rJava**) and features for code profiling (the package **profr** and **proftools**). Worth to mention is also an integration of scripting languages such as JavaScript, see for example (amongst others) package **shinyjs** (Attali 2016). In addition, R offers interfaces to almost all commercial and open-source linear program solvers which are often needed for special tasks in survey methodology.

3. R for survey statistics

R includes several methods that are helpful for data processing and survey methodology in statistical offices and organizations which usually deal with complex data sets from finite populations. The CRAN task view on *Official Statistics and Survey Methodology* (<http://cran.r-project.org/view=OfficialStatistics>) contains a list of packages which include methods typically used in official statistics and survey methodology. Below we list those packages and briefly outline their functionalities.

3.1. Complex survey designs

The package **sampling** includes various algorithms (Brewer, Midzuno, pps, systematic, Sampford, balanced (cluster or stratified) sampling via the cube method, etc.) for drawing survey samples, as well as functionality to calibrate the design weights (see, e.g., Tillé 2006).

For estimation purposes and to work with already drawn survey samples, the package **survey** (Lumley 2010) – the standard package for that task – can be used once the given survey design has been specified (stratified sampling design, cluster sampling, multi-stage sampling and pps sampling with or without replacement). The resulting object can be used to estimate (Horvitz-Thompson-) totals, means, ratios and quantiles for domains or the entire survey sample, and to apply regression models. Variance estimation for means, totals and ratios can either be done by Taylor linearization or resampling (BRR, jackknife, bootstrap or user-defined).

As an add-on package, **ReGenesees** uses facilities from and extends the **survey** package (not on CRAN, see <http://www.istat.it/en/tools/methods-and-it-tools/processing-tools/regenesees>). This package also includes a GUI.

The package **EVER** (Estimation of Variance by Efficient Replication) provides variance estimation for complex designs by delete-a-group jackknife replication for (Horvitz-Thompson-) totals, means, absolute and relative frequency distributions, contingency tables, ratios, quantiles and regression coefficients, even for domains.

The **laeken** package (Alfons and Templ 2013) implements functions to estimate certain social inclusion indicators (at-risk-of-poverty rate, quintile share ratio, relative median risk-of-poverty gap, Gini coefficient), including their variance for domains and strata based on (calibrated) bootstrap resampling.

To perform simulation studies in official statistics, the package **simFrame** (Alfons, Templ, and Filzmoser 2010) provides a framework for comparing different point and variance estimators under different survey designs as well different scenarios for missing values, representative and non-representative outliers.

Other packages are available for selecting samples with specific designs: **pps**, **sampling** and **SamplingStrata**.

3.2. Calibration

To calibrate the sampling weights to precisely match the population characteristics and/or to calibrate for unit non-responses, the package **survey** allows for post-stratification, generalized raking/calibration, generalized regression (GREG) estimation and trimming of weights.

The **EVER** package includes facilities (function `kottcalibrate()`) for calibrating on known population characteristics, on marginal distributions or joint distributions of categorical variables, or on totals of quantitative variables.

The `calib()` function in the package **sampling** allows to calibrate for non-response (with response homogeneity groups) for stratified samples. The implementation in **laeken** (and package **simPop**) (function `calibWeights()`) is similar but possibly faster.



3.3. Editing and visual inspection of microdata

The package **editrules** (de Jonge and van der Loo 2012) provides tools for editing and error localization using the Fellegi-Holt principle and categorical constraints. It converts readable linear (in)equalities into matrix form, which can then be applied for editing the given data set. The package **deducorrect** depends on the package **editrules** and applies deductive correction of simple rounding, typing and sign errors based on balanced edits. Values are changed so that the given balanced edits are complete.

The package **rrcovNA** (Todorov, Templ, and Filzmoser 2011) provides robust location and scatter estimation and robust principal component analysis with a high breakdown point for incomplete data. It can thus be used to find representative and non-representative outliers.

The package **VIM** (Templ, Alfons, and Filzmoser 2012) can be used for visual inspection of microdata. It is possible to visualize missing values using suitable plot methods and to analyze missing values structure in microdata using univariate, bivariate, multiple and multivariate plots. The information on missing values from specified variables is highlighted in selected variables. **VIM** can also evaluate imputations visually. Moreover, the package **VIMGUI** (Schopfhauser, Templ, Alfons, Kowarik, and Prantner 2014) provides a point and click graphical user interface (GUI).

The package **tabplot** (Tennekens, de Jonge, and Daas 2013) entails the table plot visualization method, which is used to profile or explore large statistical data sets. Up to a dozen variables are shown column-wise as bar charts (numeric variables) or stacked bar charts (factors). Hierarchies can be visualized with the package **treemap**.

Visual analysis of data is important to understand the main characteristics, main trends and relationships in data sets and it can be used to assess the data quality. Using the R package **sparkTable** (Kowarik, Meindl, and Templ 2014a), statistical tables holding quantitative information can be enhanced by including spark-type graphs such as sparklines  and sparkbars . These kind of graphics have initially been proposed by Tufte (2001) and are considered as simple, intense and illustrative graphs that are small enough to fit in a single line. Thus, they can easily enrich tables and texts with additional information in a comprehensive visual way. The R-package **sparkTable** uses a clean S4-class design and provides methods to create different types of sparkgraphs that can be used in webpages, presentations and text documents. With the GUI, graphical parameters can be interactively changed, variables can be sorted, and graphs can be added/removed in an interactive manner. Thereby it is possible to produce custom-tailored graphical tables – standard tables that are enriched with graphs – that can be displayed in a browser and exported to various formats.

3.4. Imputation

A distinction is made between interactive model-based methods, k -nearest neighbor methods (k NN) and miscellaneous methods. However, the criteria for using a given method depend on the scale of the data, which in official statistics are typically a mixture of continuous, semi-continuous, binary, categorical and count variables. Note that only few imputation methods can deal with mixed types of variables and semi-continuous ones, and only the methods in the package **VIM** account for robustness issues.

Expectation-Maximization (EM) based imputation methods are offered by the packages **mi** (Yu-Sung, Gelman, Hill, and Yajima 2011), **mice** (van Buuren and Groothuis-Oudshoorn 2011), **Amelia** (Honaker, King, and Blackwell 2011), **VIM** and **mix** (Schafer 1997). The package **mi** provides the iterative EM-based multiple Bayesian regression imputation of missing values and checking of the regression models used, whereas the regression models for each variable can also be defined by the user. The data set may consist of continuous, semi-continuous, binary, categorical and/or count variables. The package **mice** (van Buuren and Groothuis-Oudshoorn 2011) provides iterative EM-based multiple regression imputation as well, and the data set may consist of continuous, binary, categorical and/or count variables.

Multiple imputation in which first bootstrap samples are drawn for EM-based imputation can be carried out with **Amelia** (Honaker *et al.* 2011). It is also possible to impute longitudinal data. The package **VIM** offers EM-based multiple imputation (function `irmi()`) using robust estimations (Templ, Kowarik, and Filzmoser 2011), which adequately deal with data including outliers. It can handle data consisting of continuous, semi-continuous, binary, categorical and/or count variables.

Nearest neighbor (NN) imputation methods are also included in the **VIM** package. It provides implementation of the popular sequential and random (within a domain) hot-deck algorithm, and also a fast k NN algorithm which can be performed for large data sets. It uses a modification of the *Gower Distance* to deal with a mixture of numerical, categorical, ordered, continuous and semi-continuous variables.

3.5. Statistical disclosure control

Data from statistical agencies and other institutions are often confidential in its raw form. One of the main tasks of data providers is to modify the original data in order to guarantee that no statistical unit can be re-identified and, at the same time, to minimize the loss of information. For microdata perturbation, the package **sdcMicro** (Templ, Meindl, Kowarik, and Chen 2014c; Templ, Kowarik, and Meindl 2015) can be used to generate confidential (micro)data, i.e. to generate public- and scientific-use files. All methods are implemented in C++ to allow fast computations. The package **sdcMicroGUI** (Templ, Meindl, and Kowarik 2014b) also provides a GUI (Kowarik, Templ, Meindl, and Fontenau 2014c; Templ *et al.* 2014b).

To simulate synthetic data, the package **simPop** (Alfons, Kraft, Templ, and Filzmoser 2011b; Meindl, Templ, Alfons, and Kowarik 2014) offers methods for the simulation of synthetic, confidential, close-to-reality populations for surveys based on sample data. Such population data can then be used for extensive simulation studies in official statistics using, for example, the package **simFrame** (Alfons *et al.* 2010). Another package, **synthpop**, offers also methods to simulate populations, but it is not designed for dealing with hierarchical structures of the data.

For tabular data, the package **sdcTable** (Templ and Meindl 2010) can be used to provide confidential (hierarchical) tabular data. It includes techniques to solve the secondary cell suppression problem. A method is included that protects the complete hierarchical, multi-dimensional table at once, and therefore it is only suitable for small problems. The package also offers interfaces to various commercial and open-source linear program solvers.

3.6. Time series analysis and seasonal adjustment

For a general time series methodology, we refer to the CRAN Task View *TimeSeries* on CRAN. Specifically for survey methodology, the decomposition of time series can be done using the function `decompose()`. For a more advanced decomposition the functions `stl()` and `StructTS()` can be used. All these functions are available in the base **stats** package. Many powerful tools for seasonal adjustment can be accessed via the R package **x12** and **x12gui** (Kowarik, Meraner, Templ, and Schopfhauser 2014b). It provides a wrapper function and GUI for the **X12 binaries**, which have to be installed first. Another package, available on CRAN is **seasonal**, which supports also *SEATS Specification Files* from TRAMO-SEATS, a software for seasonal adjustment developed by the Bank of Spain.

3.7. Statistical matching and record linkage

The package **StatMatch** (D’Orazio, Di Zio, and Scanu 2006) provides functions for conducting statistical matching between two data sources sharing a number of common variables. It creates a synthetic data set after matching of two data sources via a likelihood approach or hot-deck. **MatchIt** allows nearest neighbor matching, exact matching, optimal matching and

full matching, among other matching methods.

The package **RecordLinkage** (Borg and Sariyar 2015) provides functions for linking and de-duplicating data sets. It can be used to perform and evaluate different record linkage methods. A stochastic framework is implemented which calculates weights through an EM algorithm. Machine learning methods are utilized, including decision trees (package **rpart**), adaboost (package **ada**), neural nets (package **nnet**) and support vector machines (see package **e1071** and package **kernlab**). The generation of record pairs and comparison patterns from single data items are provided as well. Comparison patterns can be chosen to be binary or based on some string metrics. In order to reduce computation time and memory usage, blocking can be used.

It is worth mentioning the package **stringdist** (van der Loo 2014), which implements various methods for string comparison.

3.8. Small area estimation

The package **hbsae** (Boonstra 2012) provides functions to compute small area estimates based on a basic area or unit-level model. The model is fit using restricted maximum likelihood (REML), or in a hierarchical Bayesian way. Auxiliary information can be either counts resulting from categorical variables or means from continuous population information.

The package **rsae** (Schoch 2014) provides functions to estimate the parameters of the basic unit-level small area estimation (SAE) model (aka nested error regression model) by means of ML or robust ML. On the basis of the estimated parameters, robust predictions of the area-specific means are computed (incl. MSE estimates; parametric bootstrap). However, the current version (rsae 0.1-5) does not allow for categorical independent variables.

The package **nlme** provides facilities to fit Gaussian linear and nonlinear mixed-effects models and **lme4** includes facilities to fit linear and generalized linear mixed-effects model, both used in small area estimation. With package **JoSAE** (Breidenbach 2013), point and variance estimation for the generalized regression (GREG) and a unit level empirical best linear unbiased prediction (EBLUP) estimators can be made at domain level. It basically provides wrapper functions to the **lme** package that is used to fit the basic random effects models. Package **saeSim** (Warnholz and Schmid 2015) provides tools for simulation of synthetic data for small area related tasks.

3.9. Indices and indicators

A comprehensive collection of indicator methodology is included in the package **laeken** which helps to estimate popular risk-of-poverty and inequality indicators (at-risk-of-poverty rate, quintile share ratio, relative median risk-of-poverty gap, Gini coefficient, gender pay gap). In addition, standard and robust methods for tail modeling of Pareto distributions are provided for the semi-parametric estimation of indicators from continuous univariate distributions such as income variables. Classes for the resulting indicators are defined as well as print, summary, plotting and subsetting methods for objects of these classes. The variances of the indicators can be estimated via a calibrated bootstrap approach (Alfons and Templ 2013).

Various indicators of poverty, concentration and inequality are included in the package **ineq** (Zeileis 2014). It provides some basic tools like Lorenz curves and Pen's parade graph. However, it is not designed to deal with sampling weights directly (these could only be approximately emulated via `rep(x, weights)`). The package **IC2** includes three inequality indices: extended Gini, Atkinson and Generalized Entropy. It can deal with sampling weights and subgroup decomposition is also supported.

The function `priceIndex()` from the package **micEcon** can be used to calculate price indices. For visualization purposes, the package **sparkTable** offers tools to produce scalable sparklines for webpages and reports, as already mentioned in Section 3.3. It also contains visualization tools for presenting indicators in checker plots—a grid-based representation of possible com-

plex indicators, bar plots and time series in thematic maps (Templ, Hulliger, Kowarik, and Fürst 2013).

3.10. Dependencies between packages

Figure 1 shows the dependencies between packages listed at the CRAN Task View on Official Statistics and Survey Methodology. In general, each package may depend on others using functionality already implemented there. As soon as a package depends to another one as depending on it in the package description, the whole *namespace* of this package is imported. Too many dependencies often may cause packages instability over time, since any change in a package has an influence. On the other hand, dependencies highlight packages connections. This makes the functions design and usage more consistent. For example, several packages rely on the **survey** package—the **samplingbook** package uses existing functionality of the **survey** package while the **VIMGUI** package allows imputation of **survey** objects. In general,

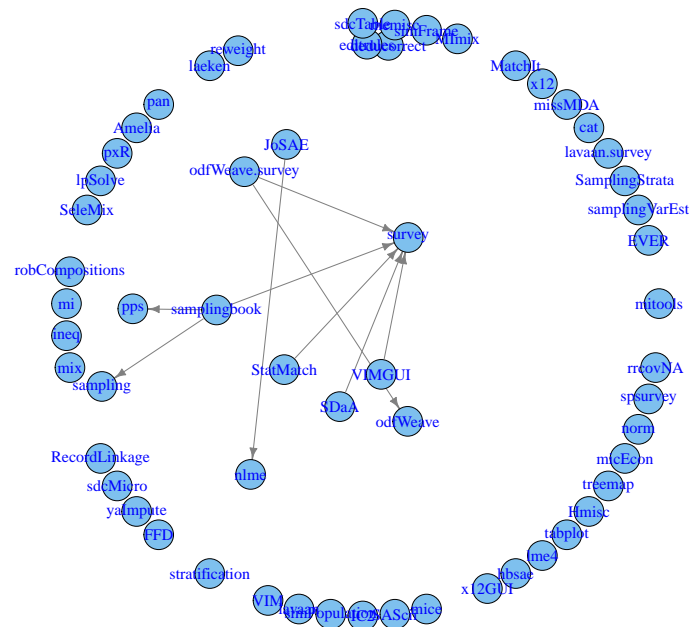


Figure 1: Dependencies between packages listed at the CRAN Task View on Official Statistics and Survey Methodology.

not many connections between the highlighted packages are visible. On the one hand, this makes the packages independent but the consistency of code and code usage might increase as soon as packages allow for dealing with similar structures and objects belonging to other packages.

The choropleth map in Figure 2 presents each country according to the relative downloads of R packages included in the CRAN Task View on Official Statistics and Survey Methodology. Since no download statistics about downloads from all (approximately 90) CRAN mirrors and internal repositories exist, the download of *RStudio*'s '0-cloud' CRAN log files from October 2012 till March 2014 are used to show the distribution of R over the world. Since the absolute number of downloads is dominated by countries with high population, the absolute download statistics per country are divided by the 2005 population count. North America, Europe and Australia are leading in the usage of R packages listed on the CRAN Task

View on Official Statistics and Survey Methodology. Figure 3 also presents the download statistics from RStudio's repository and the most popular packages listed on the CRAN Task View on Official Statistics and Survey Methodology (note that a similar analysis was carried out at <http://www.r-bloggers.com/finally-tracking-cran-packages-downloads/>). The figures underestimate the download by not considering all CRAN mirrors and on the other hand overestimation is caused by counting also package updates. The packages **Hmisc**, **nlme** and **lme4** are downloaded with highest frequency. Note that these packages are also used for tasks not related to the official statistics and survey methodology. Therefore, the most widely used packages for survey tasks might be **survey** and **mice** (more than 250 times per week on average). In any case, the download peak refers to packages that include imputation methods (**mice**, **mitools**, **Amelia**, **VIM**, **mi**), which suggests the extensive use of this methodology.

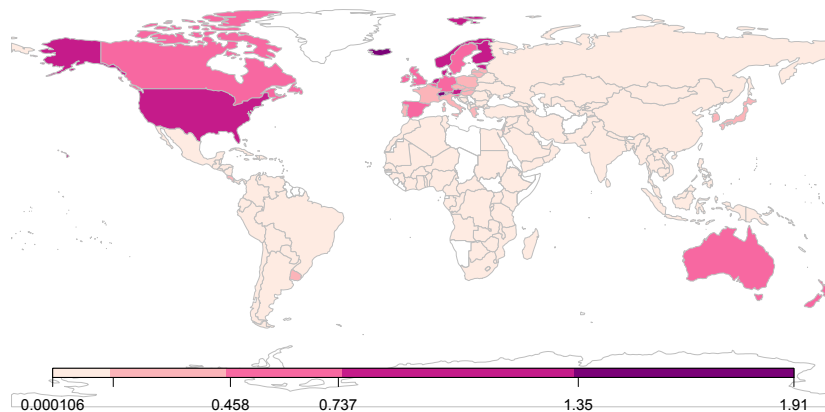


Figure 2: Choropleth map showing the downloads of R packages included in the CRAN Task View on Official Statistics and Survey Methodology over the world. The number of downloads are presented per capita (i.e. normalized by dividing by the population counts).

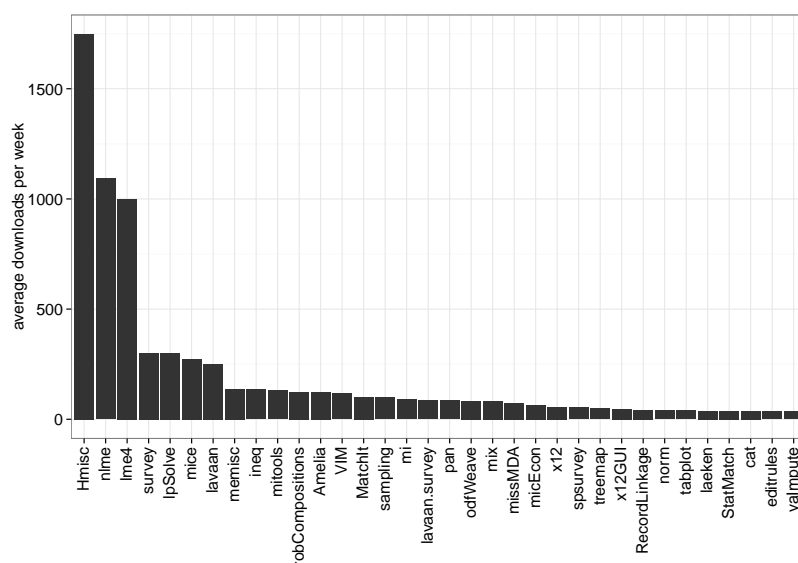


Figure 3: Average number of downloads per week (from October 2012 till March 2014) for the most downloaded packages listed on the CRAN Task View on Official Statistics and Survey Methodology.

4. The use of R in selected national statistical offices

Statistical organizations need a strategy for using and distributing software to their employees. The national statistical offices of Canada, Austria, Netherlands, Italy, USA, UK, the statistical offices of a few other countries and several international agencies are quite active in using R. This is recognized by the number of contributions at international conferences, see for example the official statistics sessions at the useR! 2013 conference in Albacete, the R session at the International Conference on Establishment Surveys in Montreal 2012, as well as the UNECE data editing work sessions in 2012, 2013 and 2014. The latter had sessions focused on the use of R in data editing and imputation. Tutorials at the NTTS 2015 and the Q2014 on the use of R in the statistical office were presented.

Moving to R seems to be slow due to the lack of strong programming knowledge and availability of many legacy code developed in other statistical software environments. However, new and innovative projects using R are carried out, some of them are listed in the following subsections. Examples will be given of how R is used in selected national statistical offices.

4.1. Use of R at Statistics Austria

At Statistics Austria, R is currently installed on more than 65 computers and on virtual servers. This is useful for tasks involving large memory requirements or to put content on the web, e.g. via **shiny** (RStudio Inc. 2014).

The leading R-team at Statistics Austria consists of three methods division experts. In addition, each department has nominated one person as first contact in case of questions and problems. Furthermore, the following organizational setup has been chosen:

- the R-experts at the methods unit (the administrators) take care that the version used is always up-to-date and they also decide on a GUI front-end which packages should be installed with the default installation. They place all necessary files (R, RStudio, packages, documentation, examples) on a particular server, in the following termed *container*;
- the IT department takes these files and deploys the R-installation including the front-end (RStudio) to users. This ensures that only one standardized software package is installed on all computers;
- the general R-support is centralized through a mailing list (apart from direct questions that can be answered by first-contact persons);
- an internal wiki was created and is used to collect know-how;
- the administrators define – together with the IT department – access rights for users on the servers, the mailing list, wiki, file depot, etc. In this situation, basically two user groups exist:
 1. R-administrators: have read and write access and are responsible for the folder containing all software package, documentation, wiki, etc. Additionally, administrators have full access to the mailing list “R-Support”.
 2. R-team: team-members have read-only access to the R container and read and write access to the wiki.

Currently five R courses are offered for the employees at Statistics Austria. The introductory course consists of 12 hours tutorial (3×4 hours), the aim is to reach a certain level of knowledge for all participants. The target group is constituted by beginners and regular R users who learned R in self-study. For the latter group many fundamental insights to the software are presented which are mostly new even for experienced users. The course covers topics on data types, data import/export (including database connections), syntax, data manipulation

(including presentation of important add-on packages such as **dplyr** and **data.table**) and basic object-orientation features. Ex-cathedra teaching is followed by exercises and R sessions in which the trainers interactively give additional insights.

The advanced training course consists of four-module courses between 4 and 8 hours each. This course covers topics on graphics (**graphics**, **grid**, **ggplot2**, **ggmap**, **ggvis**), classes and object-orientation (**S3** classes as well as a brief introduction to **S4** classes), dynamic reporting (**Sweave**, **knitr**, **brew**, **rmarkdown**), R development issues (profiling, debugging, benchmarking, basic packaging), web-applications (**shiny**), data manipulation (**dplyr**, **data.table**) and survey statistics using **survey**. The courses give an overview of other useful packages for key-tasks in official statistics.

At Statistics Austria, methodological training is offered to staff. In these courses, R is intensively used for teaching purposes, but participants do not get in direct touch with R. The reason was that for these courses no requirements with respect to any software or programming language should be set. Thus, a blended learning feedback system was developed (Dinges, Kowarik, Meindl, and Templ 2011). At the start of a course, participants fill out an online questionnaire. The collected data are then automatically used in the exercises and are also incorporated into the presentation slides. Participants are able to identify their data in various graphics, tables and other output. At various times, participants have to do interactive exercises using point and click directly in the browser. An online client/server based tool was developed which includes (among others) single- and multiple choice questions, animated and interactive examples. All clicks and answers from the participants are automatically saved on the server and aggregated statistics (feedback) are generated automatically in order to show clearly if the examples were correctly solved. In a teacher-interface, trainers can select certain examples which are then available for course participants. They also get feedback about how many participants have already solved which question and the correctness of the solutions. In the student-interface, the current exercises are available and listed ready to be selected.

Beside the development of the packages **laeken**, **sparkTable**, **sdcTable**, **sdcMicro**, **sdcMicroGUI**, **VIM**, **VIMGUI**, **x12** and **x12GUI**, R is applied in the production process for sampling, editing and imputation, estimation, analysis and output generation for several surveys. One example could be the estimation regarding PEAAC (Program for the International Assessment of Adult Competencies) or even the tourism statistics as well as the higher education forecast where R is used for everything, also for producing reports (see, e.g., Radinger, Nachtmann, Peterbauer, Reif, Hanika, Kowarik, and Lehner 2014).

4.2. Use of R at the national statistical office at UK

The National Statistical Office of UK started to use R version 2.0.1. in 2004. Employees training is done since 2005. In 2011 an R testing group was established consisting of members of the methods and IT department, with the aim of testing disclosure control tools and standard graphics using R. Since 2012 an R development group has been installed, whose objective was to test if R, and related specialized packages, are ready for use in the production environment. Several applications were investigated in the past. R was used to call **X12ARIMA**. In addition, the **dlnm** package (Petrus 2010) was used to calculate the unemployment statistics. The **survey** package is used for the Labor Force Survey (panel design). The survey error correlation problem induced by this rotating panel is considered, specifying a multivariate model according to time series of five waves (a selected household will be surveyed five times in row, 1/5 of households are exchanged) of the rotating panel. The **spatstat** package and its kernel smoothing features (Baddeley and Turner 2005) are used to visualize crime data at postcode level. The **MortalitySmooth** package (Camarda 2012) has been used for mortality rates estimation (Brown, Mills, Ayoubkhani, and Gallop 2013). Hereby, the (smoothed) mortality against age per year are presented in heat maps.

Currently, the **survey** and **ReGenesee**s packages are tested and other statistical functions are implemented for the national accounts database *CORD*.

4.3. Use of R at the national statistical institute of Romania

The National Statistical Institute of Romania established the Romanian R team in 2013. They organized already two international workshops on *New Challenges for Statistical Software - The Use of R in Official Statistics*, the last held in Bucharest in April 2015.

One course was given with the title *Introduction in SAE estimation techniques with application in R* whereas the **JoSAE** package was used. Several other courses on R are planned.

Various R packages play an important role for the Business Register Department, due to the need of using of administrative data in the production of business statistics. Hereby, R is used because of its flexibility and powerful tools for editing, validation of data and data imputation. The quality of data is in general evaluated and reported using existing tools in R. Probabilistic record linkage methods are applied to merge enterprises from different data sources on the basis of names and address information. The Department of Indicators on Population and International Migration uses the package **JoSAE** for the application of small area estimation techniques. It is used to produce data on annual international migrant stocks. The Department of Social Statistics tests the use of R for sampling. Particularly packages **ReGenesees**, **vardpoor** and **survey** are tested for use in the production according to variance estimation for household surveys.

More information can be found in (Dobre and Adam 2014).

4.4. Use of R at the national statistical institute of Serbia

The Statistical Office of Serbia uses the **laeken** package for poverty estimation. They compared the results obtained with **laeken** to those obtained with SAS[®] Macros provided by Eurostat. They also use R for estimations on the monthly retail trade survey. They use the package **XLConnect** to read and write Microsoft Excel files from within R.

4.5. Use of R at Statistics Netherlands

Statistics Netherlands started using R in a systematic manner already in 2010, see van der Loo (2012). A knowledge center was built up, an internal wiki provides code examples and serves as a platform for knowledge sharing. Every employee who wants to use R participates in training courses that are offered internally in the office. Statistics Netherlands distinguishes between three installations of R:

- the production installation is the smallest one,
- the analysts installation includes more packages and
- the researchers installation includes all tools which are useful for the development of R code including the *RTools* (<https://cran.r-project.org/bin/windows/Rtools/>) which facilitate the infrastructure for R package building on Microsoft Windows platforms.

Currently R is installed on approximately 160 individual computers and it is used mainly for data manipulation tasks, regression analysis, visualizations and data editing. Examples of R use in the statistical production process include the estimation of the Dutch Hospital Standardized Mortality Ratio (HSMR), the estimation of certain unemployment figures, estimation of tourist accommodations, and manipulation of supply and demand tables for National Accounts, see van der Loo (2012). Statistics Netherlands uses R also for data collection and web crawling with web robots, and for collecting data for compilation of price statistics.

Several packages were developed by Statistics Netherlands. The **editrules** package for data editing and the **deducorrect** package for deductive correction and deductive imputation, see Section 3.3. With the **rspa** package numerical records can be modified to satisfy edit rules.

These packages are used for automatic data editing system in child care center statistics. .NET/SQL is used to communicate between R and the database.

Also packages for visualization were developed by Statistics Netherlands and submitted to CRAN: **treemap**, **tableplot** and **tabplotd3**. The package **LaF** can import large ASCII files. In addition, Statistics Netherlands also developed the packages **stringdist**, **docopt**, **ffbase**, **daff** and **whisker**.

4.6. Use of R at UNIDO

The statistical business process of international organizations is slightly different from that in the national statistical offices. International organizations like the United Nations Industrial Development Organisation (UNIDO) do not carry out surveys, but collect and aggregate data from national authorities (statistical offices, ministries, governmental departments) and create multivariate cross-sectional time series for their analysis and for further dissemination. The statistical activities of UNIDO are defined by its responsibility to provide the international community with global industrial statistics and meet internal data requirements to support the development and research program of the organization. Currently, UNIDO maintains an industrial statistics database, which is regularly updated with the data, collected from national statistical offices and OECD (for OECD member countries). UNIDO also collects national accounts-related data from the National Accounts Main Aggregates Database of UNSD, the World Development Indicators of the World Bank and other secondary sources. Such data are primarily used to compile statistics related to Manufacturing Value Added (MVA); its growth rate and share in gross domestic product (GDP) in various countries and regions. UNIDO disseminates industrial data through its publication of the International Yearbook of Industrial Statistics, CD products and through the newly developed online portal at <http://stat.unido.org>.

The statistics team in UNIDO started using R already in 2008, when the migration of the complete statistical system from a mainframe to a client/server architecture was completed. While the main production line is still in SAS® and .NET, all new applications and tools are developed in R. UNIDO has published two research papers on the topic *R in the Statistical Office* (Todorov 2010; Todorov and Templ 2012). These papers present an overview of R, which focuses on the strengths of this statistical environment for the typical tasks performed in national and international statistical offices and outline some of the advantages of R using examples from the statistical production process of UNIDO where certain steps were either migrated or newly developed in R. One example application emphasizes the graphical excellence of R as applied for generating publication quality graphics included in the *International Yearbook of Industrial Statistics* (UNIDO 2014). The graphics together with the related text are typeset in L^AT_EX using the R tool for dynamic reporting **Sweave**. Another application illustrates the analytical and modeling functions available in R and within add-on packages (see Boudt, Todorov, and Upadhyaya 2009). These are used to implement a *nowcasting* tool for Manufacturing Value Added (MVA) to generate estimates for UNIDO publications. Functions from the package for robust statistics **robustbase** (Rousseeuw, Croux, Todorov, Ruckstuhl, Salibian-Barrera, Verbeke, and Maechler 2013) are used for this purpose.

UNIDO has developed several packages for internal use, but some of the packages are also available online. The package **CIttools** provides tools for computation and evaluation of composite indicators. It was developed for computing and analysis of the UNIDO's *Competitiveness Industrial Performance Index* (UNIDO 2013) and is available from *R-Forge* (<https://r-forge.r-project.org/projects/cia/>).

The World Input-Output Database (WIOD) (Timmer, Erumban, Gouma, Los, Temurshoev, de Vries, Arto, Genty, Neuwahl, Rueda-Cantuche, Villanueva, Francois, Pindyuk, Poschl, Stehrer, and Streicher 2012) is a new public data source which provides time-series of world input-output tables for the period from 1995 to 2009. The package **rwiot** developed by UNIDO provides analytical tools for exploration of the various dimensions of the internationalization

of production through time and across countries using input-output analysis. The package contains functions for basic (Leontief and Goshian inverse, backward and forward linkage, impact analysis) as well as advanced (vertical specialization) input-output analysis. Compositional data analysis techniques (Facevicová, Hron, Todorov, Guo, and Templ 2014) can be applied to study the interregional intermediate flows by sector and by region.

UNIDO's technical assistance to developing countries and countries with economies in transition is aimed at either creating a new industrial database or improving the existing statistical system. Through its technical support, UNIDO promotes the quality assurance of industrial statistics by statistical projects that are designed to assist in producing accurate, complete and internationally comparable statistical data. As a main statistical tool for data management and analysis the R environment is promoted. Training materials, example data sets and packages are prepared and courses are carried out, but still a lot of work is necessary to make R the statistical software of choice in the developing countries.

5. Examples

In the following section we provide two examples of R in official statistics. The first one illustrates the estimation of gender pay gap while the second one gives a short overview of anonymization of data.

5.1. Indicators and models from SES

In the European Union the gender pay gap is estimated from the *Structure of Earnings Survey* (SES) which is conducted in nearly all European countries. Similar surveys are collected all over the world. SES data includes information on enterprise and employment level. Generally, such linked employer-employee data are used to identify the determinants/differentials of earnings, but some indicators are also directly derived from hourly earnings, like the gender pay gap or the Gini coefficient (Gini 1912). SES is a complex survey of enterprises and establishments with more than 10 employees (e.g. 11.600 enterprises in Austria), NACE (an European industry standard classification system consisting of a 6 digit code) C-O, including a large sample of employees (in Austria: 207.000). The sampling units can be selected using the **sampling** (Tillé 2006) or the **simFrame** (Alfons *et al.* 2011b) package, for example. In many countries, a two-stage design is used, whereby a stratified sample of enterprises and establishments on the NACE 1-letter section level, *NUTS1* (Nomenclature of Territorial Units for Statistics) and employment size range is used in the first stage, and large enterprises may have higher inclusion probabilities. In stage 2, systematic sampling or simple random sampling is applied on each enterprise to sample employees. Often, unequal inclusion probabilities regarding employment size range categories are included here.

Unit non-responses can be considered by applying calibration through **survey** (Lumley 2010), **sampling** or **ReGenesees** packages. It is also important to understand the behavior of non-response items. To analyze the missing values structure using exploratory and visual tools, the package **VIM** (Templ *et al.* 2012) can be used. **VIM** also provides model-based imputation methods built on robust estimates that can deal with all kinds of variables. For example, one of the functions **irmi** or **kNN** can be used to impute item non-responses in the data. Calibration is applied to represent some population characteristics corresponding to NUTS 2 and NACE 1-digit level, but calibration is also carried out for the gender characteristic (number of males and females in the population). Here, the package **survey**, **sampling** or the **calibWeights** function from the package **laeken** (Alfons and Templ 2013) or **simPop** can be used.

Our example focuses on the gender pay gap as implemented in the R-package **laeken** (Alfons, Holzer, and Templ 2011a; Alfons and Templ 2013). The classical estimates – presented here as breakdown by education – are obtained by the **gpg()** function, assuming that the analyzed data are stored as a data frame. The **print** method for the resulting objects displays the estimates – the overall estimate and the estimates from the chosen breakdown. Not

surprisingly, the sampling weights were specified by the `weights` argument. The code for estimation of the Gender Pay Gap using the package **laeken** is as follows:

```
> library("laeken")
> data("ses")
> g1 <- gpg(inc = "earningsHour", method = "median",
  gender = "sex", weights = "GrossingUpFactor.x",
  breakdown = "education", data = ses)
> g1
g1
Value:
[1] 0.1938192
```

```
Value by stratum:
      stratum      value
1 ISCED 0 and 1 0.2086474
2      ISCED 2 0.1487547
3 ISCED 3 and 4 0.1695580
4      ISCED 5A 0.2974547
5      ISCED 5B 0.2198194
```

The variance of these point estimates are estimated using the syntax below. Here, a calibrated bootstrap is applied for variance estimation ([Alfons and Templ 2013](#)).

```
variance("earningsHour", weights = "GrossingUpFactor.x",
  gender="Sex", data = x, indicator = g1,
  X = calibVars(x$Location), breakdown="education", seed = 123)
```

```
Value:
[1] 0.1938192
```

```
Variance:
[1] 2.078831e-05
```

```
Confidence interval:
      lower      upper
0.2253051 0.2439922
```

```
Value by stratum:
      stratum      value
1 ISCED 0 and 1 0.2086474
2      ISCED 2 0.1487547
3 ISCED 3 and 4 0.1695580
4      ISCED 5A 0.2974547
5      ISCED 5B 0.2198194
```

```
Variance by stratum:
      stratum      var
1 ISCED 0 and 1 1.362429e-03
2      ISCED 2 9.451208e-05
3 ISCED 3 and 4 2.191488e-05
4      ISCED 5A 2.218666e-04
5      ISCED 5B 2.475571e-04
```

Confidence interval by stratum:

	stratum	lower	upper
1	ISCED 0 and 1	0.1772622	0.3200365
2	ISCED 2	0.1525004	0.1888864
3	ISCED 3 and 4	0.1951637	0.2142648
4	ISCED 5A	0.3024255	0.3652755
5	ISCED 5B	0.1728143	0.2363946

However, since the gender pay gap is very sensitive to outliers, it is recommended to apply robust methods (Hulliger, Alfons, Filzmoser, Meraner, Schoch, and Templ 2011). The following code snippet shows how to estimate robustly the gender pay gap using semi-parametric modeling (for details, see, Alfons, Templ, Filzmoser, and Holzer 2011c; Hulliger *et al.* 2011) and replacing the outliers in the tail with estimates from the modeled Pareto distribution.

```
ts <- paretoScale(x$earningsHour, w = x$GrossingUpFactor.x)

# estimate shape parameter
fit <- paretoTail(x$earningsHour, k = ts$k,
  w = x$GrossingUpFactor.x)

# replacement of outliers
earningsHour <- replaceOut(fit)

# fit of the gender pay gap
gpg(earningsHour, weights = x$GrossingUpFactor.x)
```

To illustrate model-based estimations and to show (a very simple) model-fitting functionality of R, we choose a model described in Marsden (2010) which was applied in the PiEP Lissy project (http://cordis.europa.eu/docs/publications/8260/82608181-6_en.pdf). This model is also used in Dybczak and Galuscak (2010). OLS regression models are fitted and the gross log hourly earnings of workers in enterprises are modeled, see below.

```
lm1 <- lm(log(earningsHour) ~ Sex + age + I(age^2) + education +
  Occupation, data=x)
summary(lm1)
```

The predicted values for the hourly earnings can then be used to estimate the gender pay gap. With the `summary()` method, the effect of gender, age, education and occupation on the hourly earnings can be displayed. A bunch of diagnostic plots are available with `plot(lm1)`.

5.2. Anonymization of SES

Typically, statistical data are published on the web in tabular form. However, the tabulated information may allow to identify individuals. To avoid the disclosure of individuals in tabular data, primary and secondary suppressions can be made with the help of the package `sdcTable` (Meindl 2014). Once the hierarchies are specified, the function `protectTable()` allows to apply various algorithms to anonymize the tables.

However, researchers from various institutions often need microdata for more detailed analysis. If linkage of the SES data with externally released data sources is successfully based on a number of identifiers (key variables), the intruder will have access to all the information related to a specific corresponding unit in the released data. This means that a subset of critical variables can be exploited to disclose everything about a unit in the data set.

The code listed below shows how to set up the disclosure scenario for the SES data. First an object of class `sdcMicroObj` is created by using function `createSdcObj()` (from package

sdcMicro) that includes all information on the disclosure scenario. For example, the disclosure risk of our data corresponding to the key variables is already available in the resulting object. We selected categorical and continuous key variables as well as we specified the variable holding information on the sampling weights.

```
library("sdcmicro")
sdc <- createSdcObj(x,
  keyVars = c('Size', 'age', 'Location', 'economicActivity'),
  numVars = c('earningsHour', 'earnings'),
  weightVar = 'GrossingUpFactor.y')
print(sdc, "freq")
```

Number of observations violating

```
- 2-anonymity: 4979
- 3-anonymity: 11291
```

Percentage of observations violating

```
- 2-anonymity: 2.49 %
- 3-anonymity: 5.65 %
```

```
print(sdc, "risk")
```

```
-----
0 obs. with higher risk than the main part
Expected no. of re-identifications:
4956.19 [ 2.48 %]
-----
```

From this output it is easy to see the large number of unique combinations from cross-tabulating the categorical key variables. However, relative to the size of the data (199909 observation) this number is not that large as it looks at the first view, it is about 2.5% of the observations. Nevertheless, certain number of observations may have a considerable higher individual risk, see the last line in the previous code snippet. For details on risk estimation (also on other methods available in the package) we refer to [Franconi and Polettini \(2004\)](#) and [Templ *et al.* \(2015\)](#). It is necessary to recode some categories of the key variables to receive a lower number of uniqueness as well as to apply local suppression as well as anonymizing the continuous key variables. Functions `globalRecode` and `groupVars` (for global recoding), `localSuppression` (heuristic algorithm performing local suppression), `microaggregation`, and many other methods can be directly applied on the object `sdc`.

As an example, below the code is shown how to add correlated noise ([Brand 2004](#)) to continuously scaled key variables. The parameter `noise` determines how many noise (in percentages) is added.

```
sdc <- addNoise(sdc, noise=100)
```

In the object `sdc` all information about disclosure risk and data utility is saved (see the corresponding print methods from the manual). For further reading we refer to [Templ *et al.* \(2015\)](#) and ([Templ, Kowarik, and Meindl 2014a](#)).

5.3. Accessing international statistical databases with R

When conducting economics studies, like competitiveness analysis or benchmarking, it is necessary to access different sources of data. Many international organizations maintain statistical databases which cover certain types of data: COMTRADE, UNCTAD and WTO for international trade data, World Development Indicators (WDI) from the World bank, World Economic Outlook (WEO) and International Financial statistics (IFS) from the International Monetary Fund (IMF), the Industrial statistics databases (INDSTAT) by UNIDO and many more. Some of these organizations already provide an application programming interface (API) for accessing the data which tremendously facilitates the use of these databases. Here we will consider several examples of accessing such databases using code written in R. For some of these APIs R packages are already available, for others only examples of R code are provided.

World Development Indicators

The flagship publication of the World Bank, *World Development Indicators*, presents a comprehensive collection of cross-country comparable development indicators, compiled from officially-recognized international sources. The database contains more than 1300 time series for more than 200 economies. The countries and areas are presented in more than 30 groups, with data for many indicators going back more than 50 years. The statistical tables are available online and are consistently updated based on revisions to the World Development Indicators database. An API is provided by the World Bank which offers direct access to information contained in the World Bank databases. The data can be accessed by country, by type, by topic and more. The CRAN package **WDI** makes it easy to search and download data from the WDI. The package contains essentially two functions – `WDIsearch()` for searching for data and `WDI()` for downloading the selected data into a data frame. Let us search for example for indicators related to CO2 emissions:

```
library("WDI")
co2list <- WDIsearch("CO2 emissions")
head(co2list)
```

indicator	name
[1,] "EN.ATM.CO2E.CP.KT"	"CO2 emissions from cement production (thousand metric tons)"
[2,] "EN.ATM.CO2E.FF.KT"	"CO2 emissions from fossil-fuels, total (thousand metric tons)"
[3,] "EN.ATM.CO2E.FF.ZS"	"CO2 emissions from fossil-fuels (% of total)"
[4,] "EN.ATM.CO2E.GF.KT"	"CO2 emissions from gaseous fuel consumption (kt) "
[5,] "EN.ATM.CO2E.GF.ZS"	"CO2 emissions from gaseous fuel consumption (% of total) "
[6,] "EN.ATM.CO2E.GL.KT"	"CO2 emissions from gas flaring (thousand metric tons)"

The function `WDIsearch()` uses `grep`, which allows to use (case-insensitive) regular expressions. For example if searching for manufacturing value added at constant prices we could write:

```
WDIsearch("manufacturing.*value.*constant")
```

indicator	name
[1,] "NV.IND.MANF.KD"	"Manufacturing, value added (constant 2005 US\$)"
[2,] "NV.IND.MANF.KN"	"Manufacturing, value added (constant LCU)"

Having identified the indicator we need, e.g. `NV.IND.MANF.KD` for *Manufacturing, value added (constant 2005 US\$)* we can download the data for one or more countries. Let us download also the total population (indicator `'SP.POP.TOTL'`) for the same countries and compute MVA per capita.

```
df.mva <- WDI("NV.IND.MANF.KD", country=c("IN","MY","ID"), start=1970, end=2013)
df.pop <- WDI("SP.POP.TOTL", country=c("IN","MY","ID"), start=1970, end=2013)
df <- merge(df.pop, df.mva)
names(df)[4:5] <- c("POP", "MVA")
df$MVACAP <- round(df$MVA/df$POP)
head(df)
```

	iso2c	country	year	POP	MVA	MVACAP
1	ID	Indonesia	1970	114066887	2750076402	24
2	ID	Indonesia	1971	116996006	2836927088	24
3	ID	Indonesia	1972	119974444	3265345458	27
4	ID	Indonesia	1973	123002081	3763262762	31
5	ID	Indonesia	1974	126080548	4371172606	35
6	ID	Indonesia	1975	129210098	4909605912	38

Figure 4 shows the MVA for the selected countries.

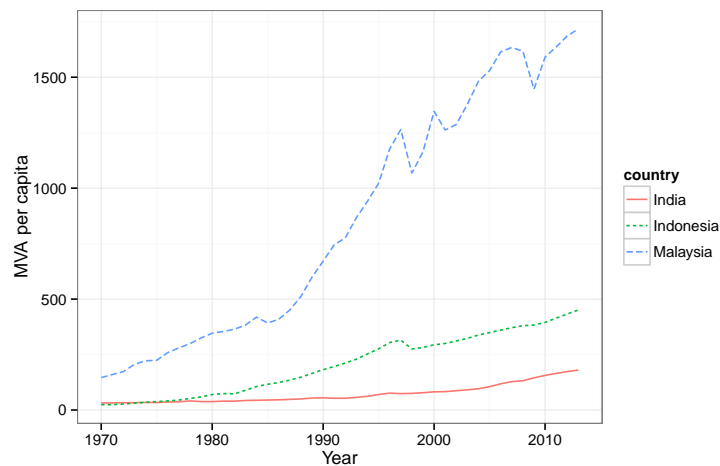


Figure 4: Manufacturing value added (MVA) per capita in India, Indonesia and Malaysia, using data from World Development Indicators.

UN COMTRADE

The United Nations Commodity Trade Statistics Database (UN COMTRADE) contains annual international trade data for over 170 reporter countries, detailed by commodities and partner countries. This is the largest repository of international trade data. All values are converted from national currency into current US dollars using exchange rates supplied either by the reporter countries, or derived from monthly market rates and volume of trade. The data are reported in current classification and revision – Harmonized system (HS) in most cases – and are converted all the way down to the earliest classification SITC revision 1. The time series start as far back as 1962 and go up to the most recent completed year. The data are available from Internet at <http://comtrade.un.org/db/default.aspx>, but recently a new API was developed (still in beta version). A HTTP GET request is sent to the URL <http://comtrade.un.org/api/get?...> and the output may be (currently) in comma-separated values (CSV) or JSON format. In the following will be shown how to use this API for simple data queries. First let us get the list of all reporting countries and areas and find the codes of some countries – which we will use later for retrieving the data.

```
## Read the list of all reporting countries/areas
```

```
library("rjson")
jsonFile <- "http://comtrade.un.org/data/cache/reporterAreas.json"
repCountry <- fromJSON(file=jsonFile)
repCountry <- as.data.frame(do.call(rbind, repCountry[[2]]))
colnames(repCountry) <- c("code", "name")
head(repCountry)
```

	code	name
1	all	All
2	4	Afghanistan
3	8	Albania
4	12	Algeria
5	20	Andorra
6	24	Angola

```
subset(repCountry, name %in% c("Austria", "Bulgaria"))
```

	code	name
13	40	Austria
33	100	Bulgaria

The code of Bulgaria is 100 and of Austria is 40. Let us now build a query and retrieve all trade flows (imports and exports) from Bulgaria to Austria in 2010. Only the total for all commodities will be requested. All other parameters remain defaults.

```
comtradeURL <- "http://comtrade.un.org/api/get?"
ps <- "2010"      # one year
r <- 100          # reporting country=Bulgaria
p <- 40           # partner = Austria
rg <- "all"       # trade flow
cc <- "TOTAL"     # total of all commodities

comtradeURL <- paste0(comtradeURL,
                      "ps=", ps, "&",    # time period
                      "r=", r, "&",      # reporting area
                      "p=", p, "&",      # partner country
                      "rg=", rg, "&",    # trade flow
                      "cc=", cc, "&",    # commodities
                      "fmt=csv",        # format is CSV
                      sep = "")
dd <- read.csv(comtradeURL, header=TRUE)
dd
```

	Classification	Year	Period	Period.Desc.	Aggregate.Level
1	H3	2010	2010	2010	0
2	H3	2010	2010	2010	0
	Is.Leaf.Code	Trade.Flow.Code	Trade.Flow	Reporter.Code	Reporter
1	0	1	Import	100	Bulgaria
2	0	2	Export	100	Bulgaria
	Reporter.ISO	Partner.Code	Partner	Partner.ISO	Commodity.Code
1	BGR	40	Austria	AUT	TOTAL
2	BGR	40	Austria	AUT	TOTAL
	Commodity	Qty.Unit.Code	Qty.Unit	Qty	Netweight..kg.
1	All Commodities	1	No Quantity	NA	NA

2	All Commodities	1	No Quantity	NA	NA
	Trade.Value..US..	Flag			
1	882147923	0			
2	388469998	0			

Similarly, data can be downloaded from COMTRADE using JSON. Other formats (Microsoft Excel, **SDMX**) are planned. The maximum number of records is limited to 50.000 per data query. For authorized users batch mode download through the API will be available. As already mentioned, this is still an experimental API and for production is recommended to use the legacy API.

6. Summary and conclusions

There is an increasing demand for statistical tools, which combine easy to use traditional software packages with newest analytical methods and one very popular such tool is the statistical programming language R. In this contribution, we briefly described its usefulness in the daily work of statistical offices, listed and briefly presented the most popular R packages for survey methodology.

The development of R packages for specific areas of official statistics is growing quickly. For example, R provides many more methods for statistical disclosure control than any other software package. The situation is similar in indicators methodologies and survey statistics, especially new developments using robust methodology are available (see for example the robust estimation of gender pay gap in the listing in Section 5). The development of such new tools was strongly supported by international activities such as the AMELI project (Münnich, Alfons, Bruch, Filzmoser, Graf, Hulliger, Kolb, Lehtonen, Lussmann, Meraner, Nedyalkova, Schoch, Templ, Valaste, Veijanen, and Zins 2011).

The usefulness and power of various R packages can be shown, whereas the field of application is manifold. First, R can be used to work efficiently with data, either computing in memory using new packages like **dplyr** and **data.table** or by connecting to databases. Specialized packages allow the user-friendly application of R to many specific fields in official statistics and survey methodology, as shown in the example section for survey statistics and remote access to statistical databases. Infrastructure for R is provided in various national and international statistical offices; this includes the distribution of R, the internal organizational support, the development of packages and holding training courses on R. Our contribution gives an outline of the usefulness of R for statisticians, methodologists, subject matter specialists and statistical stakeholders in the area of official statistics and survey methodology.

And finally note that R is not only freeware, free software and open-source, one of the greatest advantages of R is its online support (Tippmann 2015).

Acknowledgment

We would like to thank Ana Maria Dobre for providing us further details on the use of R in Statistics Romania. Special thanks goes to Rainer Stütz for numerous helpful comments and improvements of the text.

The views expressed herein are those of the authors and do not necessarily reflect the views of the United Nations Industrial Development Organization (UNIDO).

References

Alfons A, Holzer J, Templ M (2011a). *laeken: Laeken Indicators for Measuring Social Cohesion*. R package version 0.2.2, URL <http://CRAN.R-project.org/package=laeken>.

- Alfons A, Kraft S, Templ M, Filzmoser P (2011b). “Simulation of Close-to-Reality Population Data for Household Surveys with Application to EU-SILC.” *Statistical Methods & Applications*, **20**(3), 383–407. doi:10.1007/s10260-011-0163-2.
- Alfons A, Templ M (2013). “Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**.” *Journal of Statistical Software*, **54**(15), 1–25.
- Alfons A, Templ M, Filzmoser P (2010). “An Object-Oriented Framework for Statistical Simulation: The R Package **simFrame**.” *Journal of Statistical Software*, **37**(3), 1–36.
- Alfons A, Templ M, Filzmoser P, Holzer J (2011c). “Robust Pareto Tail Modeling for the Estimation of Indicators on Social Exclusion using the R Package **laeken**.” *Research Report CS-2011-2*, Department of Statistics and Probability Theory, Vienna University of Technology.
- Allaire J, McPherson J, Xie Y, Wickham H, Cheng J, Allen J (2014). **rmarkdown**: *Dynamic Documents for R*. R package version 0.3.3, URL <http://rmarkdown.rstudio.com>.
- Attali D (2016). **shinyjs**: *Perform Common JavaScript Operations in shiny Apps using Plain R Code*. R package version 0.4.0, URL <https://CRAN.R-project.org/package=shinyjs>.
- Bache S, Wickham H (2014). **magrittr**: *A Forward-Pipe Operator for R*. R package version 1.5, URL <http://CRAN.R-project.org/package=magrittr>.
- Baddeley A, Turner R (2005). “**spatstat**: An R Package for Analyzing Spatial Point Patterns.” *Journal of Statistical Software*, **12**(6), 1–42.
- Boonstra H (2012). **hbsae**: *Hierarchical Bayesian Small Area Estimation*. R package version 1.0, URL <http://CRAN.R-project.org/package=hbsae>.
- Borg A, Sariyar M (2015). **RecordLinkage**: *Record Linkage in R*. R package version 0.4-7, URL <http://CRAN.R-project.org/package=RecordLinkage>.
- Boudt K, Todorov V, Upadhyaya S (2009). “Nowcasting Manufacturing Value Added for Cross-Country Comparison.” *Statistical Journal of the IAOS: Journal of the International Association of Official Statistics*, **26**, 15–20.
- Brand R (2004). “Microdata Protection Through Noise Addition.” In *Privacy in Statistical Databases. Lecture Notes in Computer Science*, pp. 347–359. Springer, Berlin, Heidelberg.
- Breidenbach J (2013). **JoSAE**: *Functions for Some Unit-Level Small Area Estimators and their Variances*. R package version 0.2.2, URL <http://CRAN.R-project.org/package=JoSAE>.
- Brown G, Mills J, Ayoubkhani D, Gallop A (2013). “Smoothing Mortality Rates Using R.” *Research report*, ONS.
- Camarda C (2012). “**MortalitySmooth**: An R Package for Smoothing Poisson Counts with P-Splines.” *Journal of Statistical Software*, **50**(1), 1–24. ISSN 1548-7660.
- de Jonge E, van der Loo M (2012). **editrules**: *R Package for Parsing and Manipulating Edit Rules*. R package version 2.2-0, URL <http://CRAN.R-project.org/package=editrules>.
- Dinges G, Kowarik A, Meindl B, Templ M (2011). “An Open Source Approach for Modern Teaching Methods: The Interactive TGUI System.” *Journal of Statistical Software*, **39**(7), 1–19.
- Dobre A, Adam R (2014). “The Progress of R in Romanian Official Statistics.” *Romanian Statistical Review*, **2**, 45–54.

- D’Orazio M, Di Zio M, Scanu M (2006). *Statistical Matching: Theory and Practice*. Wiley Series in Survey Methodology. John Wiley & Sons, Chichester, England; Hoboken, NJ. ISBN 9780470023549.
- Dowle M, Short T, Lianoglou S, Srinivasan A (2014). **data.table**: *Extension of data.frame*. R package version 1.9.2, URL <http://CRAN.R-project.org/package=data.table>.
- Dybczak K, Galuscak K (2010). “Changes in the Czech Wage Structure: Does Immigration Matter?” *Working paper series no 1242*, European Central Bank. Wage dynamics network.
- Facevicová K, Hron K, Todorov V, Guo D, Templ M (2014). “Logratio Approach to Statistical Analysis of 2x2 Compositional Tables.” *Applied Statistics*, **41**(5), 944–958.
- Fox J (2005). “The R Commander: A Basic Statistics Graphical User Interface to R.” *Journal of Statistical Software*, **14**(9), 1–42.
- Franconi L, Polettini S (2004). “Individual Risk Estimation in μ -Argus: A Review.” In J Domingo-Ferrer (ed.), *Privacy in Statistical Databases, Lecture Notes in Computer Science*, pp. 262–272. Springer, Berlin, Heidelberg.
- Gentlemen R (2009). “Data Analysts Captivated by R’s Power.” URL <http://www.nytimes.com/2009/01/07/technology/business-computing/07program.html>.
- Gini C (1912). “Variabilità E Mutabilità: Contributo Allo Studio Delle Distribuzioni E Delle Relazioni Statistiche.” *Studi Economico-Giuridici della R. Università di Cagliari*, **3**, 3–159.
- Godfrey A (2013). “Statistical Analysis from a Blind Person’s Perspective.” *The R Journal*, **5**(1), 73–80.
- Honaker J, King G, Blackwell M (2011). “Amelia II: A Program for Missing Data.” *Journal of Statistical Software*, **45**(7), 1–47.
- Horner J (2011). **brew**: *Templating Framework for Report Generation*. R package version 1.0-6, URL <http://CRAN.R-project.org/package=brew>.
- Hulliger B, Alfons A, Filzmoser P, Meraner A, Schoch T, Templ M (2011). “Robust Methodology for Laeken Indicators.” *Research Project Report WP4 – D4.2*, FP7-SSH-2007-217322 AMELI. URL <http://ameli.surveystatistics.net>.
- Kowarik A, Meindl B, Templ M (2014a). **sparkTable**: *Sparklines and Graphical Tables for T_EX and HTML*. R package version 0.12.0, URL <http://CRAN.R-project.org/package=sparkTable>.
- Kowarik A, Meraner A, Templ M, Schopfhauser D (2014b). “Seasonal Adjustment with the R Packages **x12** and **x12GUI**.” *Journal of Statistical Software*, **62**(2), 1–21.
- Kowarik A, Templ M, Meindl B, Fontenau F (2014c). “Graphical User Interface for Package **sdcmicro**.” *Technical report*, International Household Survey Network. URL <http://www.ihnsn.org/home/sites/default/files/resources/sdcMicroGUI.pdf>.
- Leisch F (2003). “**Sweave**, Part II: Package Vignettes.” *R News*, **3**(2), 21–24.
- Leisch F, Rossini A (2003). “Reproducible Statistical Research.” *Chance*, **16**(2), 46–50.
- Lumley T (2010). *Complex Surveys: A Guide to Analysis Using R*. Wiley, Hoboken, NJ. ISBN 9780470284308.
- Marsden D (2010). “Pay Inequalities and Economic Performance.” *Technical Report PiEP Final Report V4*, Centre for Economic Performance London School of Economics, London.

- Meindl B (2014). **sdcTable**: *Methods for Statistical Disclosure Control in Tabular Data*. R package version 0.13.0, URL <http://CRAN.R-project.org/package=sdcTable>.
- Meindl B, Templ M, Alfons A, Kowarik A (2014). **simPop**: *Simulation of Synthetic Populations for Surveys Based On Auxiliary Data*. R package version 0.2.6, URL <http://CRAN.R-project.org/package=simPop>.
- Mirai Solutions GmbH (2015). **XLConnect**: *Excel Connector for R*. R package version 0.2-11, URL <http://CRAN.R-project.org/package=XLConnect>.
- Münnich R, Alfons A, Bruch C, Filzmoser P, Graf M, Hulliger B, Kolb JP, Lehtonen R, Lussmann D, Meraner A, Nedyalkova D, Schoch T, Templ M, Valaste M, Veijanen A, Zins S (2011). “Policy Recommendations and Methodological Report.” *Research Project Report WP10 – D10.1/D10.2*, FP7-SSH-2007-217322 AMELI. URL <http://ameli.surveystatistics.net>.
- Petris G (2010). “An R Package for Dynamic Linear Models.” *Journal of Statistical Software*, **36**(12), 1–16.
- Radinger R, Nachtmann G, Peterbauer J, Reif M, Hanika A, Kowarik A, Lehner D (2014). *Hochschulprognose 2014*. URL http://www.statistik.at/web_de/static/hochschulprognose_2014_063538.pdf.
- R Core Team (2015). **foreign**: *Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, Weka, dBase, . . .*. R package version 0.8-63, URL <http://CRAN.R-project.org/package=foreign>.
- Rousseeuw PJ, Croux C, Todorov V, Ruckstuhl A, Salibián-Barrera M, Verbeke T, Maechler M (2013). **robustbase**: *Basic Robust Statistics*. R package version 0.4-5, URL <http://CRAN.R-project.org/package=robustbase>.
- RStudio Inc (2014). **shiny**: *Web Application Framework for R*. R package version 0.10.2.1, URL <http://CRAN.R-project.org/package=shiny>.
- Sarkar D (2008). **lattice**: *Multivariate Data Visualization with R*. Springer, New York. ISBN 978-0-387-75968-5.
- Schafer J (1997). *Analysis of Incomplete Multivariate Data*. Chapman & Hall, London.
- Schoch T (2014). “Robust Unit-Level Small Area Estimation: A Fast Algorithm for Large Datasets.” *Austrian Journal of Statistics*, **41**(4), 243–265.
- Schopfhaue D, Templ M, Alfons A, Kowarik A, Prantner B (2014). **VIMGUI**: *Visualization and Imputation of Missing Values*. R package version 0.9.0, URL <http://CRAN.R-project.org/package=VIMGUI>.
- Templ M, Alfons A, Filzmoser P (2012). “Exploring Incomplete Data Using Visualization Techniques.” *Advances in Data Analysis and Classification*, **6**(1), 29–47. doi:10.1007/s11634-011-0102-y.
- Templ M, Hulliger B, Kowarik A, Fürst K (2013). “Combining Geographical Information and Traditional Plots: The Checkerplot.” *International Journal of Geographical Information Science*, **27**(4), 685–698.
- Templ M, Kowarik A, Filzmoser P (2011). “Iterative Stepwise Regression Imputation using Standard and Robust Methods.” *Computational Statistics & Data Analysis*, **55**(10), 2793–2806.

- Templ M, Kowarik A, Meindl B (2014a). “Statistical Disclosure Control Methods for Anonymization of Microdata and Risk Estimation.” *Technical report*, International Household Survey Network. URL <http://www.ihsn.org/home/sites/default/files/resources/sdcMicro.pdf>.
- Templ M, Kowarik A, Meindl B (2015). “Statistical Disclosure Control for Micro-Data Using the R Package **sdcMicro**.” *Journal of Statistical Software*, **67**(4), 1–36. doi:[10.18637/jss.v067.i04](https://doi.org/10.18637/jss.v067.i04).
- Templ M, Meindl B (2010). “Practical Applications in Statistical Disclosure Control Using R.” In J Nin, J Herranz (eds.), *Privacy and Anonymity in Information Management Systems*, Advanced Information and Knowledge Processing, pp. 31–62. Springer, London.
- Templ M, Meindl B, Kowarik A (2014b). “Tutorial for **sdcMicroGUI**.” *Technical report*, International Household Survey Network. URL <http://www.ihsn.org/home/sites/default/files/resources/Tutorial%20sdcMicroGUI%20v6.pdf>.
- Templ M, Meindl B, Kowarik A, Chen S (2014c). “Introduction to Statistical Disclosure Control (SDC).” *Technical Report IHSN Working Paper No 007*, International Household Survey Network. URL <http://www.ihsn.org/home/sites/default/files/resources/ihsn-working-paper-007-Oct27.pdf>.
- Temple Lang D (2013). *XML: Tools for Parsing and Generating XML within R and S-Plus*. R package version 3.98-1.1, URL <http://CRAN.R-project.org/package=XML>.
- Tennekes M, de Jonge E, Daas P (2013). “Visualizing and Inspecting Large Datasets with Tableplots.” *Journal of Data Science*, **11**(1), 43–58.
- Tillé Y (2006). *Sampling Algorithms*. Springer Series in Statistics. Springer, New York. ISBN 9780387308142.
- Timmer M, Erumban AA, Gouma R, Los B, Temurshoev U, de Vries GJ, Arto I, Genty VAA, Neuwahl F, Rueda-Cantuche JM, Villanueva A, Francois J, Pindyuk O, Poschl J, Stehrer R, Streicher G (2012). “The World Input-Output Database (WIOD): Contents, Sources and Methods.” WIOD Background document, URL www.wiod.org.
- Tippmann S (2015). “Programming tools: Adventures with R.” *Nature*, pp. 109–110. doi:[10.1038/517109a](https://doi.org/10.1038/517109a).
- Todorov V (2010). “R in the Statistical Office: The UNIDO Experience.” *Working Paper 03/2010 1*, United Nations Industrial Development.
- Todorov V, Templ M (2012). “R in the Statistical Office: Part II.” *Working paper 1/2012*, United Nations Industrial Development.
- Todorov V, Templ M, Filzmoser P (2011). “Detection of Multivariate Outliers in Business Survey Data with Incomplete Information.” *Advances in Data Analysis and Classification*, **5**(1), 37–56.
- Tufte ER (2001). *The Visual Display of Quantitative Information*. 2nd edition. Graphics Press, Cheshire, CT. ISBN 0961392142.
- UNIDO (2013). “The Industrial Competitiveness of Nations: Competitive Industrial Performance Report 2012/2013.” *Technical report*, UNIDO, Vienna.
- UNIDO (2014). *International Yearbook of Industrial Statistics*. Edward Elgar Publishing Ltd, Glensanda House, Montpellier Parade, Cheltenham Glos GL50 1UA, UK.

- van Buuren S, Groothuis-Oudshoorn K (2011). “**mice**: Multivariate Imputation by Chained Equations in R.” *Journal of Statistical Software*, **45**(3), 1–67. URL <http://www.jstatsoft.org/v45/i03/>.
- van der Loo M (2012). “The Introduction and use of R Software at Statistics Netherlands.” In *Proceedings of the Third International Conference of Establishment Surveys (CD-ROM)*. American Statistical Association, Montréal, Canada. URL <http://www.amstat.org/meetings/ices/2012/papers/302187.pdf>.
- van der Loo M (2014). “The **stringdist** Package for Approximate String Matching.” *The R Journal*, **6**, 111–122.
- Warnholz W, Schmid T (2015). **saeSim**: *Simulation Tools for Small Area Estimation*. R package version 0.7.0, URL <http://CRAN.R-project.org/package=saeSim>.
- Wickham H (2009). **ggplot2**: *Elegant Graphics for Data Analysis*. Springer, New York. ISBN 978-0-387-98140-6.
- Wickham H (2015a). **readxl**: *Read Excel Files*. R package version 0.1.0, URL <http://CRAN.R-project.org/package=readxl>.
- Wickham H (2015b). **xml2**: *Parse XML*. R package version 0.1.0, URL <http://CRAN.R-project.org/package=xml2>.
- Wickham H, Francois F (2014). **dplyr**: *A Grammar of Data Manipulation*. R package version 0.2.0.9000, URL <https://github.com/hadley/dplyr>.
- Wickham H, Francois R (2015). **readr**: *Read Tabular Data*. R package version 0.1.0, URL <http://CRAN.R-project.org/package=readr>.
- Wickham H, Miller E (2015). **haven**: *Import SPSS, Stata and SAS Files*. R package version 0.2.0, URL <http://CRAN.R-project.org/package=haven>.
- Wilkinson L, Wills G (2005). *The Grammar of Graphics*. Springer, New York. ISBN 0387245448.
- Xie Y (2013). *Dynamic Documents with R and knitr*. Chapman & Hall/CRC The R Series. Taylor & Francis. ISBN 9781482203530.
- Yu-Sung S, Gelman A, Hill J, Yajima M (2011). “Multiple Imputation with Diagnostics (Mi) in R: Opening Windows into the Black Box.” *Journal of Statistical Software*, **45**(2), 1–31.
- Zeileis A (2014). **ineq**: *Measuring Inequality, Concentration, and Poverty*. R package version 0.2-13, URL <http://CRAN.R-project.org/package=ineq>.

Affiliation:

Matthias Templ
CSTAT – Computational Statistics
Institute of Statistics & Mathematical Methods in Economics
Vienna University of Technology
Wiedner Hauptstr. 8–10
1040 Vienna, Austria
Tel. +43 1 58801 10562
e-mail: matthias.templ@tuwien.ac.at
<http://institute.tuwien.ac.at/cstat/>

Valentin Todorov
United Nations Industrial Development Organization (UNIDO),
Vienna International Centre, P.O. Box 300,
A-1400 Vienna, Austria
e-mail: v.todorov@unido.org

Contents

	Page
<i>Andreas ALFONS, Rainer STÜTZ</i> : Editorial	1
<i>Marc BILL, Beat HULLIGER</i> : Treatment of Multivariate Outliers in Incomplete Business Survey Data	3
<i>Kevin JAKOB, Matthias FISCHER</i> : GCPM : A Flexible Package to Explore Credit Portfolio Risk	25
<i>Jan-Philipp KOLB</i> : Geovisualisation: Possibilities with R	45
<i>Sebastian WARNHOLZ, Timo SCHMID</i> : Simulation Tools for Small Area Estimation: Introducing the R Package saeSim	55
<i>Andreas ALFONS, Christophe CROUX, Peter FILZMOSER</i> : Robust Maximum Association Between Data Sets: The R Package ccaPP	71
<i>Thomas MENDLIK, Georg HEINRICH, Andreas GOBIET, Armin LEUPRECHT</i> : From Climate Simulations to Statistics – Introducing the wux Package	81
<i>Matthias TEMPL, Valentin TODOROV</i> : The Software Environment R for Official Statistics and Survey Methodology	97