`micompr`: An R Package for Multivariate Independent Comparison of Observations

Nuno Fachada[1], João Rodrigues[2], Vitor V. Lopes[3], Rui C. Martins[4], and Agostinho C. Rosa[1]

[1]*Institute for Systems and Robotics, LARSyS, Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal*
[2]*École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland*
[3]*UTEC - Universidad de Ingeniería & Tecnología, Lima, Perú*
[4]*Life and Health Sciences Research Institute, School of Health Sciences, University of Minho, Braga, Portugal*

Abstract

The R package `micompr` implements a procedure for assessing if two or more multivariate samples are drawn from the same distribution. The procedure uses principal component analysis to convert multivariate observations into a set of linearly uncorrelated statistical measures, which are then compared using a number of statistical methods. This technique is independent of the distributional properties of samples and automatically selects features that best explain their differences. The procedure is appropriate for comparing samples of time series, images, spectrometric measures or similar high-dimension multivariate observations.

1   Introduction

The aim of this paper is to present the `micompr` package for R (R Core Team, 2015), which implements a procedure for comparing multivariate samples associated with different factor levels or groups. The research goal is to differentiate among pre-specified, well-defined classes or groups of sampling entities generating highly multivariate observations in which the dimensions or dependent variables are correlated, and to test for significant differences among groups. The procedure uses principal component analysis (PCA) (Jolliffe, 2002) to convert multivariate observations into a set of linearly uncorrelated statistical measures, which are then compared using a number of statistical methods, such as hypothesis tests and score plots.

This technique has several desirable attributes: a) it automatically selects observation features that best explain sample differences; b) it does not depend on the distributional properties of samples; and, c) it simplifies the researchers' work, as it can be used directly on multivariate observations. The procedure is appropriate for comparing samples of multivariate observations with highly correlated and similar scale dimensions, such as time series, images or spectrometric measures. However, the `micompr` package goes one step further by also accommodating the simultaneous comparison of multiple observation types, i.e., multiple *outputs* from a given "system". In this context, a "system" can be defined as an abstract entity capable of generating one or more stochastic data streams, i.e., *outputs*. Thus, `micompr` can determine if two or more instances of such a "system" display the same behavior by comparing observations of their *outputs*.

The remainder of this paper is organized as follows. First, in Section 2, commonly used techniques for differentiating samples of multivariate observations are discussed. The methodology employed by `micompr` is described in Section 3. Section 4 introduces the software and its architecture, namely the available objects and functions. Several concrete application examples, and how the "system"-*output* terminology fits each one, are presented in Section 5. The paper closes with Section 6, in which the overall approach and the R package are summarized.

2 Testing for significant differences in multivariate samples

Two-sample or multi-sample hypothesis tests are commonly used for assessing statistically dissimilarity in univariate samples, i.e., samples composed of scalar observations. If samples are drawn from normally distributed populations, the t (two samples) and ANOVA (n-samples) tests are adequate (Montgomery and Runger, 2014). Non-parametric tests are more appropriate if population normality cannot be assumed. The Mann-Whitney U test (Gibbons and Chakraborti, 2010) and the Kolmogorov-Smirnov test (Massey Jr., 1951) are typically employed for comparing two samples. The Kruskal-Wallis test (Kruskal and Wallis, 1952) extends the former for the n-sample case.

Multivariate analysis of variance (MANOVA) (Krzanowski, 1988; Tabachnick and Fidell, 2013) can be used as a statistical test for comparing multivariate samples. In this context, samples are composed of multi-dimensional observations, for which each dimension is a dependent variable (DV). However, MANOVA is not appropriate for cases with highly correlated DVs and when the number of DVs or dimensions is higher than the number of observations. Additionally, MANOVA is a parametric method which makes a series of assumptions on the underlying data which are not always met in practice.

Analogous non-parametric tests exist, but they are not as widespread and are commonly oriented towards specific research topics. Multiple Response Permutation Procedures (MRPP) (Mielke Jr et al., 1976) and associated permutation-based methods, such as ANOSIM (Clarke, 1993) or permutational MANOVA (Anderson, 2001), test for differences in distances between observations from each group. These tests are implemented in the `vegan` package (Oksanen et al., 2016), typically used in Ecology studies. The `Blossom` package (Talbert et al., 2016) also provides MRPP and other distance-function based permutation tests. In a similar note, Székely and Rizzo (2004) proposed a multi-sample test for equality of multivariate distributions based on the Euclidean distance between sample elements. The test statistic belongs to a class of multivariate statistics (energy statistics) proposed by the same authors. The `energy` package (Rizzo and Szekely, 2016) implements this test and other energy statistics-related functionality. The cross-match test is another distance-based test (Rosenbaum, 2005), with the particularity of not requiring permutation techniques. It is available for the R environment via the `crossmatch` package (Heller et al., 2012). In turn, the `cramer` package (Franz, 2014) provides a multivariate implementation of the non-parametric two-sample Cramér test, originally proposed by Baringhaus and Franz (2004). The critical value of the test can be determined with bootstrap (ordinary or permutation-based) or eigenvalue-based methods. Another test which avoids permutation was proposed by Duong et al. (2012). It is a kernel-based test, originally developed to assess the statistical differences between two cellular topologies. The test is implemented in the `ks` package (Duong, 2016), although limited to six-dimensional data.

An alternative to multivariate tests is to extract a number of statistical summaries (e.g., averages or extreme values) or specific points from individual multivariate observations, and then perform a univariate test for each summary measure. This approach also has its issues: a) it does not provide a single answer, i.e., it will yield as many p-values as there are summary measures; b) the choice of summary is problem-dependent and empirically driven, and consequently, error-prone, in the sense that the chosen summaries may not be representative of the original multivariate observations. While only careful analysis can minimize the latter issue, the former problem can be addressed with a multiple comparison adjustment procedure, such as the Bonferroni correction (Shaffer, 1995).

3 Independent comparison of multivariate observations

Given a matrix $\mathbf{X}_{(n \times m)}$ of n observations and m variables or dimensions, PCA can be used to obtain matrix $\mathbf{T}_{(n \times r)}$, which is the representation of $\mathbf{X}_{(n \times m)}$ in the principal components (PCs) space, and vector $\boldsymbol{\lambda}_{(1 \times r)}$, containing the eigenvalues of the covariance matrix of the original mean-centered data. Rows of $\mathbf{T}$ directly correspond to the observations of the original samples, while columns correspond to PCs. Columns are ordered by decreasing variance, i.e., the first column corresponds to the first PC, and so on. Variance is given by the eigenvalues in vector $\boldsymbol{\lambda}$, which are likewise ordered, each eigenvalue corresponding to the variance of the columns of $\mathbf{T}$. The percentage of variance explained by each PC can be obtained by dividing the respective eigenvalue with the sum of all eigenvalues. At this stage, PCA-reshaped observations associated with different groups can be compared using statistical methods. More specifically, hypothesis tests can be used to check if the sample projections on the PC space are drawn from populations with the same distribution. There are two possible lines of action:

1. Apply a MANOVA test to the samples, where each observation has q-dimensions, corresponding to the first q PCs (dimensions) such that these explain a user-defined minimum percentage of variance.

2. Apply a univariate test to observations in individual PCs. Possible tests include the t-test and the Mann-Whitney U test for comparing two samples, or ANOVA and Kruskal-Wallis test, which are the respective parametric and non-parametric versions for comparing more than two samples.

The MANOVA test yields a single p-value from the simultaneous comparison of observations along multiple PCs. An equally succinct answer can be obtained with a univariate test using the Bonferroni correction or a similar method for handling p-values from multiple comparisons. However, both approaches will not prioritize dimensions, even though the first PCs can be more important for characterizing an output, as they explain more variance. In the univariate case one can prioritize PCs according to the explained variance using the weighted Bonferroni procedure (Rosenthal and Rubin, 1983).

Conclusions concerning whether samples are statistically similar can be drawn by analyzing the p-values produced by the employed statistical tests, which should be below the typical 1% or 5% when samples are significantly different. In such case, less PCs should be required to explain the same percentage of variance than when, in the same context, no significant differences are found. The scatter plot of the first two PC dimensions can also provide visual, although subjective feedback on sample similarity.

While the procedure is most appropriate for comparing multivariate observations with highly correlated and similar scale dimensions, assessing the similarity of "systems" with multiple outputs of different scales is also possible. The simplest approach would be to apply the proposed method to samples of individual outputs, and analyze the results in a multiple comparison context. An alternative approach consists in concatenating, observation-wise, all outputs, centered and scaled to the same order of magnitude, thus reducing a "system" with k outputs to a "system" with one output. The proposed method would then be applied to samples composed of concatenated observations encompassing the existing outputs. This technique is described in detail by Fachada et al. (2017a) in the context of comparing simulation outputs.

4   The `micompr` package

4.1   Overview

The `micompr` package for the R statistical computing environment implements the methodology proposed in Section 3. Here we describe version 1.1.2 of the package, which is available at `https://cran.r-project.org/package=micompr`. The development version is hosted at `https://github.com/nunofachada/micompr`. The package is covered by the MIT license.

The `micompr` package is built upon two functions, `cmpoutput` and `micomp`. The former compares two or more samples of multivariate observations collected from one output. The latter is used for comparing multiple outputs and/or comparing outputs in multiple contexts. `grpoutputs` is a helper function for loading data from two or more set of files and preparing the data to be processed by the `cmpoutput` and/or `micomp` functions. `assumptions` is a generic function for assessing the assumptions of the parametric tests used in sample comparisons.

4.2   Architecture

`micompr` is structured according to the S3 object-oriented system. The `cmpoutput`, `micomp` and `grpoutputs` functions produce S3 objects with the same name. The package also provides the generic function `assumptions`, and two concrete implementations of methods for "cmpoutput" and "micomp" objects, which return objects of class "assumptions_cmpoutput" and "assumptions_micomp", respectively. All classes have method implementations of the common S3 generic functions `print`, `summary` and `plot`. Additionally, method implementations of the `toLatex` function, for producing user-configurable LaTeX tables with information about the performed comparisons, are provided for "cmpoutput" and "micomp" objects.

4.2.1   `grpoutputs`

This function groups outputs from sets of files containing multiple observations into samples. It returns a list of output matrices, ready to be processed by `micomp`. Alternatively, individual output matrices can

be handled by `cmpoutput`. Separate files contain one multivariate observation of one or more outputs, one column per output, one row per dimension or variable. Each specified set of files is associated with a different sample. The function is also able to create an additional concatenated output, composed from the centered and scaled original outputs.

The plot method for "grpoutputs" objects shows k plots, one per output. Output observations are plotted on top of each other, with different samples colored distinctively. The `summary` method for "grpoutputs" objects returns a list containing two elements: a) the $n \times m$ dimensions of each output matrix; and, b) the sizes of individual samples. The `print` method for "grpoutputs" objects simply outputs the summary in a more adequate presentation format.

4.2.2 `cmpoutput`

The `cmpoutput` function is at the core of `micompr`. It compares two or more samples of multivariate observations using the technique described in Section 3. It accepts an output matrix, $\mathbf{X}_{(n \times m)}$, with n observations and m variables or dimensions, a factor vector of length n, specifying the group associated with each observation, and a vector of explained variances with which to determine the number of PCs to use in the MANOVA test (alternatively, the number of PCs can also be directly specified). The function returns matrix $\mathbf{T}_{(n \times r)}$ of PCA scores and the p-values for the performed statistical tests, namely: a) a MANOVA test for each explained variance (or number of PCs); and, b) parametric (t-test or ANOVA) and non-parametric (Mann-Whitney or Kruskal-Wallis) univariate tests for each PC. Regarding the latter, the function also returns p-values adjusted with the weighted Bonferroni correction, using the percentages of explained variance by PC as weights.

The `plot` implementation for "cmpoutput" objects shows six sub-plots, namely a scatter plot with the PC1 vs. PC2 scores and five bar plots. The horizontal scale of the latter consists of the r PCs, and the vertical bars represent the explained variance (one plot) or univariate parametric and non-parametric p-values, before and after weighted Bonferroni correction (four plots). The `summary` method for "cmpoutput" objects returns a list with the following items: a) percentage of variance explained by each PC; b) p-values of the MANOVA test or tests; c) p-values of the parametric test, per PC, before and after weighted Bonferroni correction; d) p-values of the non-parametric test, per PC, before and after weighted Bonferroni correction; and, e) name of the parametric and non-parametric univariate tests employed (either t-test and Mann-Whitney U test for comparing two samples, or ANOVA and Kruskal-Wallis for more than two samples). The `print` method for "cmpoutput" objects shows the information provided by the `summary` implementation, but the p-values of the univariate tests are only shown for the first PC.

4.2.3 `micomp`

The `micomp` function performs one or more comparisons of multiple outputs, invoking `cmpoutput` for each comparison/output combination. It accepts a list of comparisons, where individual comparisons can have one of two configurations: a) a vector of folders and a vector of file sets containing data in the format required by `grpoutputs`, where each file set corresponds to a different sample; and, b) a "grpoutputs" object, passed directly. The returned objects, of class "micomp", are basically two-dimensional lists of "cmpoutput" instances, with rows associated with individual outputs, and columns with separate comparisons.

The `plot` method for "micomp" objects shows the PC1 vs. PC2 score plots for each comparison/output combination. The `summary` implementation for "micomp" objects returns a list of comparisons, each one containing a $a \times k$ matrix of p-values or number of PCs, associated with $a \geq 6$ measures and k outputs. Four rows represent the p-values of the parametric and non-parametric univariate tests for the first PC, before and after weighted Bonferroni correction. The remaining pairs of rows are associated with the MANOVA test for a given percentage of variance to explain. One row shows the p-values, and the other displays the number of PCs required to explain the specified percentage of variance for the given output. As with other `micompr` objects, the `print` method for "micomp" objects also shows the summary with a better presentation.

4.2.4 `assumptions`

`assumptions` is a generic function which performs a number of statistical tests concerning the assumptions of the parametric tests performed by the package functions. Implementations of this generic function

exist for "cmpoutput" and "micomp" objects. The former method returns objects of class "assumptions_-cmpoutput" containing results of the assumptions tests for a single output comparison. The latter returns a two-dimensional list of "assumptions_cmpoutput" objects, with rows associated to individual outputs, and columns to separate comparisons. These objects are tagged with the "assumptions_micomp" class attribute.

The following assumptions are checked: a) observations are normally distributed within each sample along individual PCs (Shapiro-Wilk test); b) observations follow a multivariate normal distribution within each sample for all PCs used in MANOVA (Royston test); c) samples have homogeneous variance along individual PCs (Bartlett test); and, d) samples have homogeneous covariance matrices for all PCs used in MANOVA (Box's M test). Assumptions a) and c) should be verified for the parametric test applied to each PC, while assumptions b) and d) should be verified for individual MANOVA tests performed for each variance to explain (or, alternatively, for each specified number of PCs).

The `plot` implementations for classes "assumptions_cmpoutput" and "assumptions_micomp" display a number of bar plots for the p-values of the performed tests. These are more detailed for "assumptions_cmpoutput" objects, showing the p-values of the univariate test for all PCs. For "assumptions_micomp" objects, one bar plot is shown per output/comparison combination, but in the case of the univariate tests only the p-values of the first PC are shown. Implementations of `summary` return a list of tabular data containing the p-values of the assumption tests. The `summary` method for "assumptions_cmpoutput" objects returns a list with two matrices of p-values, one for the MANOVA tests, another for the univariate tests. The `summary` method for "assumptions_micomp" objects follows the approach taken by the `summary` method for "micomp" objects, returning a list of p-value matrices, one matrix per comparison. Rows of individual matrices correspond to the assumptions tests, and columns to outputs. The `print` methods for "assumptions_cmpoutput" objects and for "assumptions_micomp" objects again show the summary information in a printable format.

4.2.5 `toLatex` methods for "cmpoutput" and "micomp" objects

These methods are implementations of the `toLatex` generic function, and convert "cmpoutput" and "micomp" objects to character vectors representing LATEX tables. The generated tables are configurable via function arguments, with sensible defaults. Tables can present the following data for each output/comparison combination: a) number of principal components required to explain a user-specified percentage of variance; b) MANOVA p-value for a user-specified percentage of variance to explain or number of PCs; c) parametric test p-value for a given PC, before and/or after weighted Bonferroni correction; d) non-parametric test p-value for a given PC, before and/or after weighted Bonferroni correction; e) variance explained by a specific PC; and, f) a score plot with the output projection on the first two PCs.

4.2.6 Other functions

The `micompr` package is bundled with additional functions whose purpose is to aid the main package methods do their job. However, some of these may be useful in other contexts.

The `concat_outputs` function concatenates outputs collected from multiple observations. It accepts two arguments, namely a list of output matrices, and the centering and scaling method. Several centering and scaling methods, such as "range", "iqrange", "vast" or "pareto" (Berg et al., 2006), are recognized in the second argument. The function returns an $n \times p$ matrix of n observations with length p, which is the sum of individual output lengths. Lower-level centering and scaling of individual outputs is performed by the `centerscale` function, which accepts a numeric vector and returns a new vector, centered and scaled with the specified method.

The `pvalf` generic function formats p-values for LATEX. A concrete default implementation is used by the `micompr toLatex` implementations. This implementation underlines and double-underlines p-values lower than 0.05 and 0.01, respectively, although these limits are configurable, and underlining can be turned off by setting both limits to zero. It is also possible to specify a limit below which p-values are capped. For example, if this limit is set to 1×10^{-5}, a p-value equal to 1×10^{-6} would be displayed as "$< 1e^{-5}$". The default method of `pvalf` will format p-values lower than 5×10^{-4} using scientific E notation, which is more compact and thus a better fit for tables. p-values between 5×10^{-4} and 1 are formatted using regular decimal notation with three decimal places. This aspect is not configurable.

However, another implementation of `pvalf` can be passed to the `micompr toLatex` implementations if different formatting is desired.

Simple TikZ 2D scatter plots, as the ones produced by the `micompr toLatex` implementations, can be generated with the `tikzscat` function. The function accepts the data to plot, an n x 2 numeric matrix, of n observations and 2 dimensions, and a factor vector specifying the levels or groups associated with each observation. Several plot characteristics, such as mark types, scale and axes color, are configurable via function arguments. `tikzscat` returns a string containing the TikZ figure code for plotting the specified data.

4.3   Included data

The package includes test data produced by several implementations of the Predator-Prey for High Performance Computing (PPHPC) simulation model (Fachada et al., 2015). The data is provided in `rdata` format, and is readily available on loading the package. The same data is also provided in TSV format. This is a limited subset of the complete data, and is included for package testing and exemplification purposes. The example discussed in Section 5.1 uses a superset of this data, which is available for public download, but could not be included with the package due to its large size.

4.4   Dependencies

`micompr` has a number of optional dependencies, not required for package installation and for using most of its functionality. The `biotools` (da Silva, 2016) and `MVN` (Korkmaz et al., 2014) packages are required by the `assumptions` functions, providing the statistical tests for assessing MANOVA and t-test assumptions. If these functions are invoked without the presence of the specified packages, they will inform the user of that fact, and terminate cleanly. The `testthat` (Wickham, 2011), `knitr` (Xie, 2015) and `roxygen2` (Wickham et al., 2015) packages are required for package development. The `deseasonalize` package (McLeod and Gweon, 2013) is required for building one of the vignettes[1].

5   Examples

In this section we discuss four concrete application examples for the `micompr` package. The complete scripts used in these examples are available at `https://github.com/nunofachada/micompr-examples`.

5.1   Simulation model with multiple outputs

The replication of a simulation model in a new context highlights differences between the conceptual and implemented models, as well as inconsistencies in the conceptual model specification (Edmonds and Hales, 2003), promoting model verification, model validation (Wilensky and Rand, 2007), and model credibility (Thiele and Grimm, 2015). Some argue that a simulation model is untrustworthy until it has been successfully replicated (Edmonds and Hales, 2003; David, 2013). Model parallelization is a an illustrative example of the importance of replication, as it is often required for simulating large models in practical time frames (Fachada et al., 2017b). By definition, model parallelization implies a number of changes, or even full reimplementation, of the original model, such that a robust comparison methodology, as provided by the `micompr` package, is required in order to make sure a parallelized model faithfully reproduces the behavior of the original serial model.

PPHPC is a reference model for studying and evaluating implementation strategies for spatial agent-based models, capturing important characteristics such as agent movement and local agent interactions (Fachada et al., 2015). The model describes a prototypical predator-prey system, and has six outputs, namely prey population, P^s, predator population, P^w, cell-bound food quantity, P^c, mean prey energy, $\overline{E}^s$, mean predator energy, $\overline{E}^w$, and mean cell-bound food levels, $\overline{C}$. Since outputs are collected once per iteration, each simulation run yields six time series, associated with the individual outputs. With several open source implementations publicly available, the model provides a good test case for multivariate comparison purposes.

---

[1]The `deseasonalize` package (McLeod and Gweon, 2013) has since been removed from CRAN, therefore the respective examples have been commented out of this vignette.

Here we show the main comparison cases discussed in a previous article (Fachada et al., 2017a), in which the model implementations are parameterized with size 400 and parameter set 1 (Fachada et al., 2015). A canonical PPHPC realization, implemented in NetLogo (Wilensky, 1999), is compared with three configurations of a parallel Java implementation (Fachada et al., 2017b). The NetLogo implementation and the first Java configuration follow the PPHPC conceptual model and the specified parameters. The second Java configuration disables agent shuffling prior to agent actions, which is explicitly mandated in the conceptual model description. The third Java configuration performs a minimal change in one of the parameters specified by parameter set 1. As such, we define three comparison cases:

Case I Compare the NetLogo implementation with the first Java configuration. These should yield distributionally equivalent results.

Case II Compare the NetLogo implementation with the second Java configuration. A small misalignment is to be expected.

Case III Compare the NetLogo implementation with the third Java configuration. There should be a mismatch in the outputs.

Independent samples of the six model outputs were obtained from $n = 30$ replications for each implementation or configuration, in a total of $4n = 120$ runs. Each replication $r = 1, \ldots, 4n$ was performed with a PRNG seed obtained by taking the MD5 checksum of r, guaranteeing independence between seeds, and consequently, between replications. The data generated by this computational experiment, as well as the scripts used to set up the experiment, are made available to other researchers at `https://zenodo.org/record/46848`.

The following script performs these comparisons. Note that the `concat = TRUE` option of the `micomp` function specifies that an additional concatenated output, $\widetilde{A}$, should be generated from the original outputs and analyzed in a similar fashion. The `dir_data` variable specifies the location of the dataset.

```r
# Load library
library(micompr)

# Output names
outputs <- c("$P^s$", "$P^w$", "$P^c$", "$\\mean{E}^s$",
             "$\\overline{E}^w$", "$\\overline{C}$",
             "$\\widetilde{A}$")

# Outputs from the NetLogo implementation
dir_nl_ok <- paste0(dir_data, "nl_ok")
# Outputs from the Java implementation, first configuration
dir_jex_ok <- paste0(dir_data, "j_ex_ok")
# Outputs from the Java implementation, second configuration
dir_jex_noshuff <- paste0(dir_data, "j_ex_noshuff")
# Outputs from the Java implementation, third configuration
dir_jex_diff <- paste0(dir_data, "j_ex_diff")

# Files for model size 400, parameter set 1
filez <- glob2rx("stats400v1*.txt")

# Perform the three comparison cases
mic <- micomp(outputs,
              ve_npcs = 0.75,
              list(list(name = "I",
                        folders = c(dir_nl_ok, dir_jex_ok),
                        files = c(filez, filez),
                        lvls = c("NLOK", "JEXOK")),
                   list(name = "II",
                        folders = c(dir_nl_ok, dir_jex_noshuff),
                        files = c(filez, filez),
                        lvls = c("NLOK", "JEXNS")),
```

```
                    list(name = "III",
                         folders = c(dir_nl_ok, dir_jex_diff),
                         files = c(filez, filez),
                         lvls = c("NLOK","JEXDIF"))),
               concat = T)
```

The `mic` object can be inspected at the R prompt using the common S3 generic functions `print`, `summary` and `plot`. For publication purposes, the `toLatex` method for "micomp" objects produces L^AT_EX tables with user-specified information. For example, to generate a table similar to Table 4 of our previous work (Fachada et al., 2017a), `toLatex` is invoked as follows::

```
toLatex(mic,
        booktabs = T,
        data_show = c("npcs-1", "mnvp-1", "parp-1", "scoreplot"),
        data_labels = c("$\\#$PCs", "MNV", "$t$-test", "PCS"),
        col_width = T,
        pvalf_params = list(minval = 1e-8, na_str = "*"),
        label = "tab:pphpc",
        caption = paste("Comparison of a NetLogo implementation of",
                        "the PPHPC model against three configurations",
                        "of a parallel Java implementation."))
```

This call produces Table 1 with booktabs (Fear, 2005) table style (`booktabs = TRUE`) and width set to document column width (`col_width = TRUE`), since the table is somewhat large. The `label` and `caption` parameters set the label and caption of the L^AT_EX table, respectively, while the `pvalf_params` argument accepts a list of options for formatting p-values. The `data_show` parameter specifies what data to show, which in this case is: 1) `npcs-1`, the number of PCs for the first specified variance (the `micomp` function accepts and performs output comparison with one or more specified variances); 2) `mnvp-1`, the MANOVA p-value for the first specified variance; 3) `parp-1`, t-test p-value for the first PC; and, 4) score plot for the first two PCs.

In terms of comparison, the method does not find significant differences in case I. However, it successfully differentiates the configurations compared in cases II and III. This is in line with what would be expected, and is discussed in further detail by Fachada et al. (2017a). While not shown here, the `assumptions(mic)` command reveals that most assumptions of the MANOVA and t-tests are verified.

5.2   Monthly sunspots

This example uses the monthly sunspot data (WDC, 2016), included with R, which contains the monthly numbers of sunspots from 1749 to the present day. The solar cycle is an approximate 11-year period of changes in the number of sunspots and other associated phenomena. Thus, we divide the data into 11-year (132-month) periods, and consider each period to be an observation. In practice this is an oversimplification, since the cycles can be a bit longer or shorter than 11 years.

Given the data, we define two samples of 10 observations each, over a period of 110 years or 1320 months. The first sample includes solar cycles from 1749 to 1859, while the second encompasses cycles from 1902 to 2012. We can now ask the following question: were the solar cycles during the 1749–1859 interval significantly different from the more recent observations? The following code compares observations from the two time intervals, and attempts to provide an answer:

```
# Load library
library(micompr)

# Months in the 1749-1859 interval (110 years)
# Months in the 1902-2012 interval (110 years)
m <- sunspot.month[c(1:1320, 1837:3156)]
m <- matrix(m, nrow = 20)

# Factor vector, two levels:
```

Comp.	Data	Outputs						
		P^s	P^w	P^c	$\overline{E}^s$	$\overline{E}^w$	$\overline{C}$	$\widetilde{A}$
I	#PCs	13	18	14	22	28	14	22
	MNV	0.325	0.640	0.462	0.052	0.796	0.463	0.523
	t-test	0.530	0.836	0.804	0.784	0.378	0.805	0.879
	PCS							
II	#PCs	13	15	13	1	28	14	21
	MNV	4e-05	<1e-08	0.002	*	0.089	0.003	2e-07
	t-test	0.042	<1e-08	0.108	<1e-08	0.017	0.109	0.390
	PCS							
III	#PCs	7	1	2	2	20	1	14
	MNV	<1e-08	*	<1e-08	<1e-08	<1e-08	*	<1e-08
	t-test	<1e-08	<1e-08	<1e-08	<1e-08	<1e-08	<1e-08	<1e-08
	PCS							

Table 1 – Comparison of a NetLogo implementation of the PPHPC model against three configurations of a parallel Java implementation.

```
# a) ten 11-year cycles from 1749 to 1859
# b) ten 11-year cycles from 1902 to 2012
groups <- factor(c(rep("A", 10), rep("B", 10)))

# Compare the two groups, use 9 PCs for MANOVA
cmp <- cmpoutput("SunSpots", 9, m, groups)
```

The `cmp` object can now be analyzed:

```
cmp

## Output name: SunSpots
## Number of PCs which explain 85.0% of variance: 9
## P-Value for MANOVA along 9 dimensions: 3.40755e-06
## P-Value for t-test (1st PC): 2.713985e-06
## P-Value for Mann-Whitney U test (1st PC): 4.330035e-05
## Adjusted p-Value for t-test (1st PC): 6.513579e-06
## Adjusted p-Value for Mann-Whitney U test (1st PC): 0.0001039211
```

The MANOVA p-value is significant, as well as the t-test and Mann-Whitney PC1 p-values, before and after weighted Bonferroni correction. As such, it is possible to conclude that solar cycles from 1749 to 1859 were significantly different from cycles between 1902 and 2012. However, is the data in accordance with the assumptions for the MANOVA and t-test? This can be checked with the `assumptions` function:

```
assumptions(cmp)

## === MANOVA assumptions ===
##                 NPCs=9
## Royston (A) 2.190796e-01
## Royston (B) 3.627858e-01
## Box's M     1.567180e-08
##
## === T-test assumptions ===
##                       PC1
## Shapiro-Wilk (A) 0.7739058
## Shapiro-Wilk (B) 0.3791168
## Bartlett         0.9353299
```

Only Box's M test, which checks for homogeneity of variance-covariance matrices, is significant. However, this test is prone to false positives, and this assumption is not critical when samples are of the same size (Tabachnick and Fidell, 2013). Given this information, it seems plausible to consider the results provided by the parametric tests in our final decision, i.e., that there is in fact a significant difference between samples. A good way to visualize the overall results is to plot the "cmpoutput" object:

```
plot(cmp)
```

This command generates the plots shown in Figure **??**. The score plot shows the samples to be distinctly separated, and the variance explained by PC decreases abruptly from the first PC to the second. Univariate p-values for PC1 are visibly significant, though not very much for the remaining PCs.

5.3 Saugeen river flow

This example uses the Saugeen River daily flow data (Hipel and McLeod, 1994), included with the `deseasonalize` package. This data consists of a time series of the rivers' daily flow (m^3/s) from 1915 to 1979. Considering one year as an observation, there are a total of 65 observations. We can, for example, define two samples of 30 observations each, with the first and last 30 years of records, and ask the following question: is there any statistical difference between the flow dynamics during the 1915–1944 and 1950–1979 periods (perhaps due to climate change or some other factor)?

Update 23-05-2022: The `deseasonalize` package has been removed from CRAN, therefore this example no longer runs. However, the example is still available in `micompr`'s original paper at `https://doi.org/10.32614/RJ-2016-055`.

5.4 PH^2 database of dermoscopic images

In this example we use the tools provided by the `micompr` package to study the PH^2 database of dermoscopic images (Mendonça et al., 2013). This image database contains a total of 200 dermoscopic images of melanocytic lesions, including, from benign to more serious, 80 common nevi, 80 atypical nevi, and 40 melanomas. These are 8-bit RGB color images, with a resolution of purportedly 768×560 pixels. We have found, however, that resolutions vary between 761×570 and 769×577. As such, we resized all images to 760×570 prior to our analysis. The goal is to verify if images of the three types of lesions form statistically distinguishable samples, i.e., this is not a classification exercise such as performed by Barata et al. (2014).

Each image is considered an observation of three outputs, **r**ed, **g**reen and **b**lue, corresponding to the respective color channels. The concatenation of all outputs, i.e., channels, provides a 4$^{\text{th}}$ output. The three lesion samples are compared pairwise, as follows:

1v2 Common nevi and atypical nevi.

1v3 Common nevi and melanomas.

2v3 Atypical nevi and melanomas.

The following code reads the image dataset from disk and compares images grouped by lesion type. The `imgfolder` variable specifies the path containing the images (resized to 760×570), while the `grpsfile` variable specifies the path to the file containing the sample to which each image belongs.
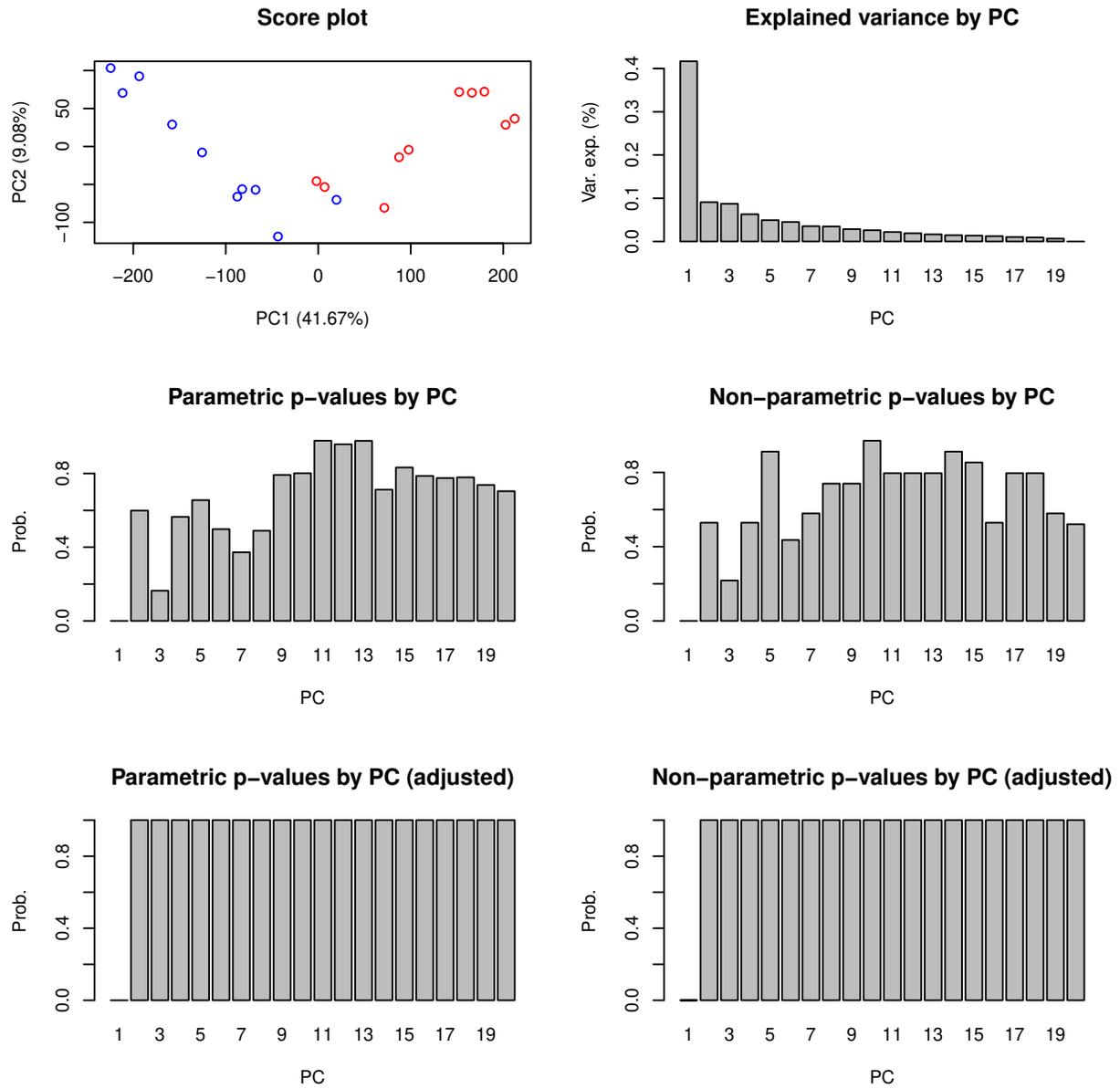
Figure 1 – Plots produced by sunspots example.

```r
# Load libraries
library(bmp)
library(micompr)

# Image definitions
imgs <- dir(imgfolder)
nimgs <- length(imgs)
npixels <- 760 * 570

# Specify image groups (Common nevi, atypical nevi,
# melanomas).
f <- read.table(grpsfile, row.names = 1)
grps <- f[order(row.names(f)), ]

# Read images from disk
# Use different color channels as outputs, and also
# use a concatenated output
rimgs <- matrix(nrow = nimgs, ncol = npixels)
gimgs <- matrix(nrow = nimgs, ncol = npixels)
bimgs <- matrix(nrow = nimgs, ncol = npixels)
rgbimgs <- matrix(nrow = nimgs, ncol = npixels * 3)

for (i in 1:nimgs) {

  cimg <- read.bmp(paste0(imgfolder, imgs[i]))
  rimgs[i, ] <- c(cimg[ , , 1])
  gimgs[i, ] <- c(cimg[ , , 2])
  bimgs[i, ] <- c(cimg[ , , 3])
  rgbimgs[i, ] <- c(cimg[ , , 1], cimg[ , , 2], cimg[ , , 3])

}

# Perform multivariate independent comparison of images
mic <-
  micomp(outputs = c("R", "G", "B", "RGB"),
         ve_npcs = 0.9,
         comps = list(
           list(name = "1v2",
                grpout = list(
                  data = list(R = rimgs[grps != 3, ],
                              G = gimgs[grps != 3, ],
                              B = bimgs[grps != 3, ],
                              RGB = rgbimgs[grps != 3, ]),
                  obs_lvls = factor(grps[grps != 3]))),
           list(name = "1v3",
                grpout = list(
                  data = list(R = rimgs[grps != 2, ],
                              G = gimgs[grps != 2, ],
                              B = bimgs[grps != 2, ],
                              RGB = rgbimgs[grps != 2, ]),
                  obs_lvls = factor(grps[grps != 2]))),
           list(name = "2v3",
                grpout = list(
                  data = list(R = rimgs[grps != 1, ],
                              G = gimgs[grps != 1, ],
                              B = bimgs[grps != 1, ],
```

Comp.	Data	Outputs			
		R	G	B	RGB
1v2	t-test	2e-04	0.001	0.033	0.002
	U test	8e-05	5e-04	0.024	0.001
	PCS				
1v3	t-test	$\leq$1e-08	$\leq$1e-08	7e-07	$\leq$1e-08
	U test	$\leq$1e-08	$\leq$1e-08	7e-06	$\leq$1e-08
	PCS				
2v3	t-test	$\leq$1e-08	$\leq$1e-08	3e-04	$\leq$1e-08
	U test	6e-08	$\leq$1e-08	0.001	8e-07
	PCS				

Table 2 – Comparison of PH^2 dataset images grouped by lesion type.

```
                    RGB = rgbimgs[grps != 1, ]),
        obs_lvls = factor(grps[grps != 1]))))))
```

As in the 5.1 example, the `mic` object can be inspected at the R prompt using the common S3 generic functions. Likewise, the `toLatex` function produces LaTeX tables summarizing the object. The following code generates Table 2:

```
toLatex(mic,
        booktabs = T,
        data_show = c("parp-1", "nparp-1", "scoreplot"),
        data_labels = c("$t$-test", "$U$ test", "PCS"),
        pvalf_params = list(minval = 1e-8, na_str = "*"),
        label = "tab:ph2",
        caption = paste("Comparison of PH$^2$ dataset images",
                        "grouped by lesion type."))
```

Note that we did not request the MANOVA p-values in the `data_show` parameter, as in this case the required assumptions do not appear to be verified. However, assumptions for the t-test on the first PC seem to hold. In any case, and to complement the information provided by the t-test, we specified the `"nparp-1"` option to the `data_show` argument, such that the table shows the p-value of the Mann-Whitney U test on the first PC.

Results in Table 2 show that images of different lesions have statistically significant differences, when compared either by individual color channels or with the three channels concatenated. The latter seems to provide better differentiation, with the common nevi and melanoma samples (1v3 comparison) appearing to be the most dissimilar.

6 Summary

In this vignette we presented the R package `micompr`, which implements a procedure for comparing multivariate samples associated with different factor levels or groups. The package architecture and its core components were discussed and four examples were examined.

References

M. J. Anderson. A new method for non-parametric multivariate analysis of variance. *Austral Ecology*, 26 (1):32–46, Feb. 2001. doi: 10.1111/j.1442-9993.2001.01070.pp.x.

C. Barata, M. Ruela, M. Francisco, T. Mendonça, and J. S. Marques. Two systems for the detection of melanomas in dermoscopy images using texture and color features. *IEEE Systems Journal*, 8(3): 965–979, Sept. 2014. doi: 10.1109/JSYST.2013.2271540.

L. Baringhaus and C. Franz. On a new multivariate two-sample test. *Journal of Multivariate Analysis*, 88(1):190–206, jan 2004. doi: 10.1016/s0047-259x(03)00079-4.

R. A. Berg, H. C. Hoefsloot, J. A. Westerhuis, A. K. Smilde, and M. J. Werf. Centering, scaling, and transformations: improving the biological information content of metabolomics data. *BMC Genomics*, 7(1):142, 2006. doi: 10.1186/1471-2164-7-142.

K. R. Clarke. Non-parametric multivariate analyses of changes in community structure. *Australian Journal of Ecology*, 18(1):117–143, mar 1993. doi: 10.1111/j.1442-9993.1993.tb00438.x. URL `http://dx.doi.org/10.1111/j.1442-9993.1993.tb00438.x`.

A. R. da Silva. *biotools: Tools for Biometry and Applied Statistics in Agricultural Science*, 2016. URL `https://CRAN.R-project.org/package=biotools`. R package version 3.0.

N. David. Validating simulations. In *Simulating Social Complexity*, Understanding Complex Systems, pages 135–171. Springer Berlin Heidelberg, 2013. ISBN 978-3-540-93812-5. doi: 10.1007/978-3-540-93813-2_8.

T. Duong. *ks: Kernel Smoothing*, 2016. URL `https://CRAN.R-project.org/package=ks`. R package version 1.10.4.

T. Duong, B. Goud, and K. Schauer. Closed-form density-based framework for automatic detection of cellular morphology changes. *Proceedings of the National Academy of Sciences*, 109(22):8382–8387, may 2012. doi: 10.1073/pnas.1117796109.

B. Edmonds and D. Hales. Replication, replication and replication: Some hard lessons from model alignment. *Journal of Artificial Societies and Social Simulation*, 6(4):11, 2003. URL `http://jasss.soc.surrey.ac.uk/6/4/11.html`.

N. Fachada, V. V. Lopes, R. C. Martins, and A. C. Rosa. Towards a standard model for research in agent-based modeling and simulation. *PeerJ Computer Science*, 1:e36, Nov. 2015. doi: 10.7717/peerj-cs.36.

N. Fachada, V. V. Lopes, R. C. Martins, and A. C. Rosa. (2017b) Parallelization strategies for spatial agent-based models. *International Journal of Parallel Programming*, 45(3):449–481, Jun. 2017. doi: 10.1007/s10766-015-0399-9.

N. Fachada, V. V. Lopes, R. C. Martins, and A. C. Rosa. (2017a) Model-independent comparison of simulation output. *Simulation Modelling Practice and Theory*, 72:131–149, Mar. 2017. ISSN 1569-190X. doi: 10.1016/j.simpat.2016.12.013. URL `http://www.sciencedirect.com/science/article/pii/S1569190X16302854`.

S. Fear. *Publication quality tables in LATEX*, Apr. 2005. URL `https://www.ctan.org/pkg/booktabs`.

C. Franz. *cramer: Multivariate nonparametric Cramer-Test for the two-sample-problem*, 2014. URL `https://CRAN.R-project.org/package=cramer`. R package version 0.9-1.

J. D. Gibbons and S. Chakraborti. *Nonparametric statistical inference*. Statistics: Textbooks & Monographs. Chapman and Hall/CRC, Boca Raton, FL, USA, fifth edition, July 2010.

R. Heller, D. Small, and P. Rosenbaum. *crossmatch: The Cross-match Test*, 2012. URL `https://CRAN.R-project.org/package=crossmatch`. R package version 1.3-1.

K. W. Hipel and A. I. McLeod. *Time series modelling of water resources and environmental systems.* Elsevier, 1994.

I. Jolliffe. *Principal component analysis.* Springer Series in Statistics. Springer, second edition, 2002. doi: 10.1007/b98835.

S. Korkmaz, D. Goksuluk, and G. Zararsiz. MVN: An R package for assessing multivariate normality. *The R Journal*, 6(2):151–162, 2014. URL http://journal.r-project.org/archive/2014-2/korkmaz-goksuluk-zararsiz.pdf.

W. H. Kruskal and W. A. Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621, 1952. doi: 10.1080/01621459.1952.10483441.

W. J. Krzanowski. *Principles of Multivariate Analysis: A User's Perspective.* Oxford University Press, New York, USA, 1988.

F. J. Massey Jr. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951. doi: 10.1080/01621459.1951.10500769.

A. I. McLeod and H. Gweon. Optimal deseasonalization for monthly and daily geophysical time series. *Journal of Environmental Statistics*, 4(11), 2013. URL http://www.jenvstat.org/v04/i11.

T. Mendonça, P. M. Ferreira, J. Marques, A. R. S. Marcal, and J. Rozeira. PH^2 - a dermoscopic image database for research and benchmarking. In *35th International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 5437–5440. IEEE, July 2013. doi: 10.1109/EMBC.2013.6610779.

P. W. Mielke Jr, K. J. Berry, and E. S. Johnson. Multi-response permutation procedures for a priori classifications. *Communications in Statistics - Theory and Methods*, 5(14):1409–1424, jan 1976. doi: 10.1080/03610927608827451.

D. C. Montgomery and G. C. Runger. *Applied statistics and probability for engineers.* John Wiley & Sons, sixth edition, 2014.

J. Oksanen, F. G. Blanchet, M. Friendly, R. Kindt, P. Legendre, D. McGlinn, P. R. Minchin, R. B. O'Hara, G. L. Simpson, P. Solymos, M. H. H. Stevens, E. Szoecs, and H. Wagner. *vegan: Community Ecology Package*, 2016. URL https://CRAN.R-project.org/package=vegan. R package version 2.4-1.

R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2015. URL https://www.R-project.org/.

M. L. Rizzo and G. J. Szekely. *energy: E-Statistics: Multivariate Inference via the Energy of Data*, 2016. URL https://CRAN.R-project.org/package=energy. R package version 1.7-0.

P. R. Rosenbaum. An exact distribution-free test comparing two multivariate distributions based on adjacency. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(4):515–530, sep 2005. doi: 10.1111/j.1467-9868.2005.00513.x.

R. Rosenthal and D. B. Rubin. Ensemble-adjusted p values. *Psychological Bulletin*, 94(3):540–541, 1983. doi: 10.1037/0033-2909.94.3.540.

J. P. Shaffer. Multiple hypothesis testing. *Annual Review of Psychology*, 46:561–584, Feb. 1995. doi: 10.1146/annurev.ps.46.020195.003021.

G. J. Székely and M. L. Rizzo. Testing for equal distributions in high dimension. *InterStat*, 5:1–6, Nov. 2004.

B. G. Tabachnick and L. S. Fidell. *Using multivariate statistics.* Pearson, sixth edition, July 2013.

M. Talbert, J. Richards, P. Mielke, and B. Cade. *Blossom: Statistical Comparisons with Distance-Function Based Permutation Tests*, 2016. URL https://CRAN.R-project.org/package=Blossom. R package version 1.4.

J. C. Thiele and V. Grimm. Replicating and breaking models: good for you and good for ecology. *Oikos*, 124(6):691–696, 2015. doi: 10.1111/oik.02170.

Sunspot Number. WDC-SILSO, Solar Influences Data Analysis Center (SIDC), Royal Observatory of Belgium, Brussels, Belgium, Apr. 2016. URL `http://www.sidc.be/silso/datafiles`.

H. Wickham. testthat: Get started with testing. *The R Journal*, 3:5–10, 2011. URL `http://journal.r-project.org/archive/2011-1/RJournal_2011-1_Wickham.pdf`.

H. Wickham, P. Danenberg, and M. Eugster. *roxygen2: In-Source Documentation for R*, 2015. URL `https://CRAN.R-project.org/package=roxygen2`. R package version 5.0.1.

U. Wilensky. *NetLogo*. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, USA, 1999. URL `http://ccl.northwestern.edu/netlogo/`.

U. Wilensky and W. Rand. Making models match: replicating an agent-based model. *Journal of Artificial Societies and Social Simulation*, 10(4):2, 2007. URL `http://jasss.soc.surrey.ac.uk/10/4/2.html`.

Y. Xie. *Dynamic Documents with R and knitr*. CRC Press, second edition, 2015.