

Journal of Data Science, Statistics, and Visualisation

MMMMMM YYYY, Volume VV, Issue II.

doi: *XX.XXXXXX/jdssv.v000.i00*

Guidelines for the Journal of Data Science, Statistics, and Visualisation

Ying Chen

National University of Singapore

Patrick J.F. Groenen

Erasmus University Rotterdam

Kwan-Liu Ma

University of California, Davis

Ann Maharaj

Monash University

Stefan Van Aelst

KU Leuven

Abstract

This short article illustrates how to write a manuscript for the *Journal of Data Science, Statistics and Visualisation* (JDSSV) using its \LaTeX style files. Please follow JDSSV's style guidelines precisely. Also, it is recommended to keep the \LaTeX code as simple as possible, that is, avoid inclusion of packages/commands that are not necessary.

Keywords: JDSSV, style guidelines, comma-separated, not capitalized, R.

1. Mission

JDSSV¹ is an international refereed journal that creates a forum to present recent progress and ideas in the different disciplines of data science, statistics, and visualisation. It welcomes contributions to data science, statistics, and visualisation, and in particular, those aspects which link and integrate these subject areas. Articles can cover topics such as machine learning and statistical learning, the visualisation and verbalisation of data, visual analytics, big data infrastructures and analytics, interactive

¹This document is an adaptation of the style guide of the Journal of Statistical Software (Zeileis 2017).

learning, and advanced computing. Papers that discuss two or more research areas of the journal are favoured. Scientific contributions should be of a high standard. Articles should be oriented towards a wide scientific audience of statisticians, data scientists, computer scientists, data analysts, etc.

The journal welcomes original contributions that are not being considered for publication elsewhere and contain a high level of novelty. Papers with a thorough but concise review of a certain topic with the potential to provide new insights are also welcome. Manuscripts submitted to the journal should be publishable as is, implying that the quality of the submission, in content, style, attention for details in figures, tables, references, etc., should be very high. We recommend that non-English speakers utilise one of the large language models accessible online to verify that their papers adhere to well-crafted English language standards prior to submission.

Manuscripts submitted to the journal generally are accompanied by supplementary material that containing software code, technical derivations or detailed explanations, additional examples, etc. All submitted material will be reviewed by the assigned associate editors. Manuscripts may have a substantial theoretical component, but it is expected that all manuscripts contain at least one application on empirical or simulated data. The journal emphasizes the reproducibility of the results presented its papers. Therefore, all data and software code that is necessary to reproduce the empirical results in the manuscript should be made available in a user friendly manner. If the empirical data cannot be released for reasons of confidentiality or otherwise, then a generated dataset with comparable properties should be provided.

All submitted material will be reviewed by two assigned associate editors. Authors are promised publication within three months. Every effort will be made to have a decision within one month; at most, one month will be allowed for revision, and the manuscript will be published within one month from acceptance. To be able to guarantee fast review times, sufficiently concise papers are needed. To be able to guarantee fast review times sufficiently concise papers are needed. Therefore, the maximum length of a paper is set to 20 pages including references but excluding appendices. The paper should be self-explanatory and not depend on supplementary material (which do not contribute to the 20-page limit).

2. Preparing your Manuscript for Submission

All submissions to JDSSV should be written in $\{\text{\LaTeX}\}$ using the JDSSV style files provided on the journal website <https://jdssv.org>. For initial submission it suffices providing the pdf version of the manuscript together with all the necessary supplementary material including software code to reproduce all results in the manuscript. The final version of accepted manuscripts should adhere to all the style guidelines and should incorporate all changes requested by the production editor. All $\{\text{\LaTeX}\}$ source files of the final manuscript should be submitted and accepted manuscripts are only published if these files comply with all guidelines and instructions provided by the journal.

3. Software

The journal expects that submissions contain accompanying software with the aim of reproducibility of the results and application of the proposed methodology to other data by the reader. All existing software used in the paper should be properly referenced. Code should be delivered in an easily readable manner with clear instructions on its use, and preferably is accompanied by instructive examples of its use. We highly recommend to provide code that can be used in open-source software such as R (R Core Team 2019), Python (Python Software Foundation 2019), Julia (Bezanson et al. 2012), Octave (Eaton et al. 2017), etc.

To make code widely accessible, we advise making it available in a repository such as <https://zenodo.org> where it will receive a permanent Digital Object Identifier (DOI) which can be included in the manuscript.

The provided code should at least consist of a file or set of files for the functions that execute the core of the method. For reproducibility, another necessary file is the script that creates the results (tables and figures) of the paper. For methods that run very long, provide a toy example that runs sufficiently fast and highlights the properties of the method. Non-proprietary data should be provided or a permanent link to these data should be given. The corresponding script to analyse the data should read these data.

For increased readability, please apply the following naming conventions for code: Start function names with a verb, e.g., `set_initialisation()`, `compute_loss()`, `update_X()`, etc. when appropriate. Give objects descriptive names or closely follow the notation in the paper. For programming conventions (particularly in R), see Hadley Wickham’s guidelines (<http://r-pkgs.had.co.nz/style.html>) and the use of the **styler** package is recommended. For **python**, consider Google’s recommendations (<https://github.com/google/styleguide/blob/gh-pages/pyguide.md>) Sufficient comments should be added to the code to make it understandable. For **Octave** and **MatLab** (The MathWorks 2018), consider Richard Johnson’s MatLab Style Guidelines 2.0 (<https://www.mathworks.com/matlabcentral/fileexchange/46056-matlab-style-guidelines-2-0>).

For writing about software JDSSV requires authors to use the markup `\proglang{}` (programming languages and large programmable systems), `\pkg{}` (software packages), `\code{}` (functions, commands, arguments, etc.). If there is such markup in (sub)section titles (as above), a plain text version has to be provided in the $\{\text{\LaTeX}\}$ command as well. In the example in Section 5, we also illustrate how abbreviations should be introduced and citation commands can be employed. See the $\{\text{\LaTeX}\}$ code for more details.

4. Review process

All manuscripts should be submitted online at <https://jdssv.org>. JDSSV uses a single blind review process. Upon submission of their manuscript, authors will have the opportunity to provide a short list of names of researchers that preferably should not be contacted for reviewing the manuscript. All submitted manuscripts will undergo automatic checking for plagiarism and will not be considered for further review in case of plagiarism. The manuscript will be assigned to one of the editors who will make an

Table 1: Part of the table with word counts describing data science by different web sources as collected by [Lubbe \(2018\)](#).

	dsc.test	dsctechniques	datadiversity	dsc.tips	...	wikipedia
statistics	6	2	2	0	...	2
big data	2	0	3	5	...	0
analytics	1	0	2	0	...	2
database	2	0	0	2	...	1
insight	0	0	4	0	...	2
⋮	⋮	⋮	⋮	⋮	⋮	⋮
programming	7	0	0	0	...	0

initial screening to check the quality of the submitted work. After positive screening, the manuscript will be assigned to two associate editors who will review the manuscripts, and submit their report with their recommendations to the assigned editor. The initial review process should normally not take more than one month. All communication between the authors and the journal will be administered by the journal editors. It is expected that an accepted manuscript will be print ready with all typographical and grammatical errors removed. Authors may utilise one of the large language models accessible online to check that their manuscripts are print ready.

5. Correspondence Analysis

As an example, we provide a simple implementation of correspondence analysis (see, e.g., [Greenacre 2010](#)). For the analysis of bivariate categorical data or other tables that contain counts, correspondence analysis can be a useful technique to visualize important relations between the categories. Consider word count data on explanations of data science by several websources provided by [Lubbe \(2018\)](#), see Table 1.

Let \mathbf{F} be the matrix containing the values in Table 1. The most simple model to fit such a table is the independence model in matrix \mathbf{E} defined as

$$\mathbf{E} = n^{-1} \mathbf{D}_r \mathbf{1} \mathbf{1}^\top \mathbf{D}_c$$

with $\mathbf{1}$ a vector of ones of appropriate length, $\mathbf{D}_r = \text{Diag}(\mathbf{F}\mathbf{1})$ the diagonal matrix with row sums of \mathbf{F} , $\mathbf{D}_c = \text{Diag}(\mathbf{F}^\top \mathbf{1})$ the diagonal matrix with column sums of \mathbf{F} , and $n = \mathbf{1}^\top \mathbf{F} \mathbf{1}$ be the sum of all elements in \mathbf{F} . The goal of correspondence analysis is to find a matrices of row and column coordinates \mathbf{R} and \mathbf{C} such that

$$L(\mathbf{R}, \mathbf{C}) = \|\mathbf{D}_r^{-1/2}(\mathbf{F} - \mathbf{E} - \mathbf{D}_r \mathbf{R} \mathbf{C} \mathbf{D}_c^\top) \mathbf{D}_c^{-1/2}\|^2$$

is minimized (see, for example, [Van de Velden et al. 2009](#)). It may be verified that the least-squares optimal solution is obtained for by computing the singular value decomposition (SVD)

$$\mathbf{D}_r^{-1/2}(\mathbf{F} - \mathbf{E}) \mathbf{D}_c^{-1/2} = \mathbf{U} \mathbf{D} \mathbf{V}^\top$$

and computing

$$\begin{aligned}\mathbf{R} &= n^{1/2} \mathbf{D}_r^{-1/2} \mathbf{U} \mathbf{D}^\alpha \\ \mathbf{C} &= n^{1/2} \mathbf{D}_c^{-1/2} \mathbf{V} \mathbf{D}^{1-\alpha}\end{aligned}$$

for some α . Note that the least-squares optimal rank p solution is obtained by taking the first p columns of \mathbf{R} and \mathbf{C} . If $\alpha = 1$ the so-called row principal solution is obtained where the row points are the weighted centroids of the column points, $\alpha = 0$ gives the column principal solution with column points being the weighted average of the row points, and $\alpha = 1/2$ results in the symmetric solution. The importance measure of the dimensions are given by the inertia, that is, the diagonal elements of \mathbf{D}^2 . For more details, see, for example, [Greenacre \(2010\)](#).

A simple implementation of correspondence analysis in R is given by the function `corana()` given by

```
R> corana <- function(dat, alpha = 0.5){
+   # Perform correspondence analysis
+   # Input:
+   ## dat numeric matrix dat with nonnegative entries
+   ## alpha = 1 is row principal standardisation
+   ##          = 0 is column principle standardisation
+   ##          = 0.5 is symmetric standardisation
+   dat <- as.matrix(dat)
+   Dr <- rowSums(dat)
+   Dc <- colSums(dat)
+   n <- sum(Dr)
+   # Compute expected values under the independence model
+   E <- outer(Dr, Dc)/n
+   tt <- svd(diag(Dr^-0.5) %*% (dat - E) %*% diag(Dc^-0.5))
+   ## Remove dimensions with singular value zero
+   ind <- tt$d > 1e-10
+   tt$d <- tt$d[ind]
+   tt$u <- tt$u[, ind]
+   tt$v <- tt$v[, ind]
+   ## Compute row scores R and column scores C
+   R <- n^0.5 * diag(Dr^-0.5) %*% tt$u %*% diag(tt$d^alpha)
+   C <- n^0.5 * diag(Dc^-0.5) %*% tt$v %*% diag(tt$d^(1 - alpha))
+   rownames(R) <- rownames(dat)
+   rownames(C) <- colnames(dat)
+   ## Compute relative contribution to inertia per dimension
+   row.inert <- outer(Dr/n, tt$d^(2*alpha), "/" ) * R^2
+   col.inert <- outer(Dc/n, tt$d^(2*(1 - alpha)), "/" ) * C^2
+   ## Compute reconstructed Chi-square distance for row and column points
+   row.dist <- R^2 %*% diag(tt$d^(2 - 2*alpha))
+   row.dist <- row.dist / outer(rowSums(row.dist), rep(1, ncol(R)))
+   col.dist <- C^2 %*% diag(tt$d^(2 - 2*(1 - alpha)))
+   col.dist <- col.dist / outer(rowSums(col.dist), rep(1, ncol(C)))
+   ## Prepare list of results
+   out <- list(R = R, C = C, sing.val = tt$d,
```

```
+           row.inert = row.inert, col.inert = col.inert,
+           row.dist  = row.dist,  col.dist  = col.dist)
+   class(out) <- "corana"
+   return(out)
+ }
```

A plot is given by the `plot.corana()` method

```
R> plot.corana <- function(out, dims = 1:2, ...){
+   R <- out$R[, dims]
+   C <- out$C[, dims]
+   ## Set up coordinate system
+   coord <- rbind(R, C)
+   plot(coord[, 1], coord[, 2], type = "n", asp = 1, las = 1,
+        xlab = paste0("Dim ", dims[1]), ylab = paste0("Dim ", dims[2]), ...)
+   abline(h = 0, v = 0, col = "gray")
+   ## Use reconstructed distance as importance measure for transparency and size
+   row.alpha <- rowSums(out$row.dist[, dims])
+   row.alpha <- (row.alpha/max(row.alpha))^0.7
+   col.alpha <- rowSums(out$col.dist[, dims])
+   col.alpha <- (col.alpha/max(col.alpha))^0.7
+   ## Plot row and column points
+   points(R[, 1], R[, 2], pch = 20, col = rgb(1, 0, 0, row.alpha))
+   points(C[, 1], C[, 2], pch = 20, col = rgb(0, 0, 1, col.alpha))
+   ## Write text labels
+   text(R[, 1], R[, 2], rownames(R), pos = compute.pos(R), cex = 2*row.alpha,
+        col = rgb(1, 0, 0, ifelse(row.alpha > 0.1, row.alpha, 0)))
+   text(C[, 1], C[, 2], rownames(C), pos = compute.pos(R), cex = 2*col.alpha,
+        col = rgb(0, 0, 1, ifelse(col.alpha > 0.1, col.alpha, 0)))
+ }
```

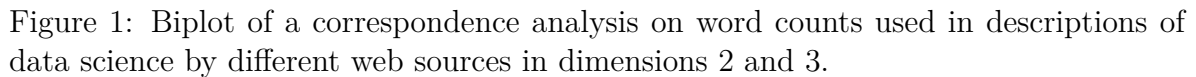
Assuming that `ds.word.cnt` contains the full matrix of which a part is shown in Table 1, then the following code does a correspondence analysis on these data:

```
R> load("ds.word.cnt.RData")
R> source("compute.pos.R")
R> out.corana <- corana(ds.word.cnt)
R> plot(out.corana, dims = 2:3)
```

that yields the plot of Dimensions 2 and 3 in Figure 1. In this plot, the points and labels are made more transparent and the labels decrease in size as they are represented worse in these two dimensions.

Computational Details

If necessary or useful, information about certain computational details such as version numbers, operating systems, or compilers could be included in an unnumbered section.



The results in this paper were obtained using R~3.5.1. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at <https://CRAN.R-project.org/>.

All acknowledgments should be collected in this unnumbered section before the references. It may contain the usual information about funding and feedback from colleagues/reviewers/etc. Furthermore, information such as relative contributions of the authors may be added here (if any).

Generative AI Statement

A statement on the use of generative AI is required at the end of the paper. Authors are allowed to use a chatbot, AI, or similarly solely to enhance the language and readability of their. Code generation is also allowed. With all use of AI, authors remain fully accountable for the content of their paper, including its accuracy, originality, and proper attribution of all sources. Chatbots cannot be listed as authors.

If an AI tool, such as a chatbot, is utilized for analytical tasks, generating research results, creating tables or figures, or writing computer code, this must be disclosed within the paper. To support scientific scrutiny, replication, and detection of falsification, authors should provide the full prompt used to generate the results, along with the query's time and date, the AI tool used, and its version.

A. More Technical Details

Appendices can be included after the bibliography (with a page break). Each section within the appendix should have a proper section title (rather than just *Appendix*).

B. Using BibTeX

References need to be provided in a BibTeX file (`.bib`). All references should be made with `\cite`, `\citet`, `\citep`, `\citealp` etc. (and never hard-coded). These commands yield different formats of author-year citations and allow to include additional details (e.g., pages, chapters, ...) in brackets.

Cleaning up BibTeX files is a somewhat tedious task – especially when acquiring the entries automatically from mixed online sources. However, it is important that information is complete and presented in a consistent style to avoid confusions. JDSSV requires the following format.

- Specific markup (`\proglang`, `\pkg`, `\code`) should be used in the references.
- Titles should be inserted in title case.
- Journal titles should not be abbreviated and in title case.
- DOIs should be included where available.
- Software should be properly cited as well. For R packages `citation("pkgname")` typically provides a good starting point.

References

- Bezanson, J., Karpinski, S., Shah, V., and Edelman, A. (2012). Julia: A fast dynamic language for technical computing. In *Lang.NEXT*. <http://arxiv.org/abs/1209.5145>.
- Eaton, J. W., Bateman, D., Hauberg, S., and Wehbring, R. (2017). *GNU Octave Version 4.4.1 Manual: a High-Level Interactive Language for Numerical Computations*, <https://www.gnu.org/software/octave/doc/v4.4.1/>.
- Greenacre, M. J. (2010). *Biplots in Practice*. Fundacion BBVA, <https://www.fbbva.es/en/publicaciones/biplots-in-practice-7/>.
- Lubbe, S. (2018). Personal communication.
- Python Software Foundation (2019). *Python Language Reference, version 3.7.2*, <https://www.python.org/>.
- R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, <https://www.R-project.org/>.
- The MathWorks, I. (2018). *MATLAB Release 2018b*. Natick, Massachusetts, United States, <https://nl.mathworks.com/>.
- Van de Velden, M., Groenen, P. J. F., and Poblome, J. (2009). Seriation by constrained correspondence analysis: A simulation study. *Computational Statistics & Data Analysis*, 53(8):3129–3138, DOI: <https://doi.org/10.1016/j.csda.2008.08.020>.
- Zeileis, A. (2017). *JSS: A Document Class for Publications in the Journal of Statistical Software*, <https://www.jstatsoft.org/public/journals/1/jss-style-only.zip>.

Affiliation:

Ying Chen

National University of Singapore

Department of Mathematics

National University of Singapore

10 Lower Kent Ridge Road

Singapore 119076

E-mail: matcheny@nus.edu.sg

URL: <https://blog.nus.edu.sg/matcheny/>

Patrick J.F. Groenen

Erasmus University Rotterdam

Econometric Institute, Erasmus School of Economics

Erasmus University Rotterdam

P.O. Box 1738

3000 DR Rotterdam, The Netherlands

E-mail: groenen@ese.eur.nl

URL: <https://personal.eur.nl/groenen/>

Kwan-Liu Ma

University of California, Davis

Department of Computer Science

University of California, Davis

One Shields Avenue

Davis, CA 95616, United States

E-mail: klma@ucdavis.edu

URL: <https://www.cs.ucdavis.edu/~ma>

Ann Maharaj

Monash University

Department of Econometrics and Business Statistics

Monash University

Wellington Road, Clayton

Victoria, 3800, Australia

E-mail: ann.maharaj@monash.edu

URL: <https://research.monash.edu/en/persons/ann-maharaj>

Stefan Van Aelst

KU Leuven

Department of Mathematics

KU Leuven

Celestijnenlaan 200B

3001 Leuven, Belgium

E-mail: Stefan.VanAelst@kuleuven.be

URL: <https://wis.kuleuven.be/stat/robust>