Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. doi: 10.18637/jss.v000.i00

A **Python/Zig** optimized and customizable implementation for the ρ_{DCCA} and DMC_x^2 methods

Fernando Ferraz Ribeiro
Universidade Federal da Bahia

Gilney Figueira Zebende
Universidade Estadual de Feira de Santana

Abstract

This short article illustrates how to write a manuscript for the *Journal of Statistical Software* (JSS) using its LaTeX style files. Generally, we ask to follow JSS's style guide and FAQs precisely. Also, it is recommended to keep the LaTeX code as simple as possible, i.e., avoid inclusion of packages/commands that are not necessary. For outlining the typical structure of a JSS article some brief text snippets are employed that have been inspired by **?**, discussing count data regression in R. Editorial comments and instructions are marked by vertical bars.

Keywords: ρ_{DCCA}, DMC_x^2, optimization, Python, Zig.

1. introduction

The ρ_{DCCA} (Zebende 2011) is a widely used coefficient that measures the cross-correlation between tow non-stationary time series. It´s an extension of the Detrended Fluctuation Analysis (DFA) (Peng, Buldyrev, Havlin, Simons, Stanley, and Goldberger 1994) and the Detrended Cross-correlation Analysis ($DCCA$) (Podobnik and Stanley 2008): while the DFA calculates the self-affinity and long-memory properties of a time series data, and the $DCCA$ analyses power-law cross correlations between two different non-stationarity time series, the ρ_{DCCA} coefficient quantifies this cross-correlation in simple values ranging from -1 to 1, where -1 indicates a perfect anti-correlation between the series, 1 a perfect correlation and zero (0) no correlation at all.

The Detrended Multiple Cross-Correlation Coefficient (Zebende and Silva 2018) (DMC_x^2) is a generalization of the ρ_{DCCA} coefficient that correlates one time series (dependent variable) a number of time series (independent variables). The DMC_x^2 values ranges from zero (0), indicating no correlation to 1, meaning perfect correlation or anti-correlation between the dependent and the independent variables.

This paper presents the **Zebende** Python package, an implementation of the DFA, $DCCA$, ρ_{DCCCA}, DMC_x^2 and utility functions related to the methods. In section 2 the steps for calculating the ρ_{DCCCA} and DMC_x^2 are presented and discussed. Section 3 shows how this library was implemented, the optimization technics and the recommended steps to use the library. In Section 4 the **Zebende** package is compared with other packages for Python and R that calculates the ρ_{DCCA} in terms of performance and usability, leading to the conclusions in 5.

2. Algorithms of the coefficients

$$DMC_x^2 \equiv \rho_{Y,X_i}(n)^T \times \rho^{-1}(n) \times \rho_{Y,X_i}(n) \tag{1}$$

The DMC_x^2 ranges from zero (0), where the time-series do not correlate, to one (1) meaning that the time-series have either a perfect correlation or a perfect anti-correlation. In this equation, $\rho^{-1}(n)$ represent the inverse matrix of all possible combinations of ρ_{DCCA} between the independent variables, in other words:

$$\rho^{-1}(n) = \begin{pmatrix} 1 & \rho_{X_1,X_2}(n) & \rho_{X_1,X_3}(n) & \dots & \rho_{X_1,X_j}(n) \\ \rho_{X_2,X_1}(n) & 1 & \rho_{X_2,X_3}(n) & \dots & \rho_{X_2,X_j}(n) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \rho_{X_j,X_1}(n) & \rho_{X_j,X_2}(n) & \rho_{X_j,X_3}(n) & \dots & 1 \end{pmatrix}^{-1} \tag{2}$$

$$\rho_{Y,X_k}(n)^T = [\rho_{Y,X_1}(n), \rho_{Y,X_2}(n), \cdots, \rho_{Y,X_j}(n)] \tag{3}$$

$$\rho_{DCCA}(n) = \frac{F^2_{DCCA\ (x\alpha,\ x\beta)}(n)}{F_{DFA\ (x\alpha)}(n) \times F_{DFA\ (x\beta)}(n)} \tag{4}$$

Taking a time series $\{x_i\}$ with i ranging from 1 to N, the integrated series X_k is calculated by $X_k = \sum_{i=1}^{k} [x_i - \langle x \rangle]$ with k also ranging from 1 to N;

the X_k series is divided in $N - n$ boxes of size n(time scale),each box containing $n+1$ values, starting in i up to $i + n$;

for each box, a polynomial (usually of degree 1) best fit is calculated, getting $\widetilde{X}_{k,i}$ with $i \leq k \leq (i + n)$ (detrended values);

in each box is calculated: $f^2_{DFA}(n, i) = \frac{1}{1+n} \sum_{k=i}^{i+n} (X_k - \widetilde{X}_{k,i})^2$

for all the boxes of a time scale, the DFA is calculated as:

$F_{DFA}(n) = \sqrt{\frac{1}{N-n} \sum_{i=1}^{N-n} f^2_{DFA}(n, i)}$;

for a number of different timescales (n), with possible values $4 \leq n \leq \frac{N}{4}$ the F_{DFA} function is calculated to find a relation among $F_{DFA} \times n$

Taking two time series with the same length $\{x\alpha_i\}$ and $\{x\beta_i\}$ with i ranging from 1 to N, the integrated series $X\alpha_k$ and $X\beta_k$ are calculated by $X_k = \sum_{i=1}^{k} [x_i - \langle x \rangle]$ for each series, with k also ranging from 1 to N;

$X\alpha_k$ and $X\beta_k$ series are divided in $N - n$ boxes of size n(time scale), each box containing $n + 1$ values, starting in i up to $i + n$;

for each box, a polynomial (usually of degree 1) best fit is calculated, getting $\widetilde{X\alpha}_{k,i}$ and $\widetilde{X\beta}_{k,i}$, for series $\{x\alpha_i\}$ and $\{x\beta_i\}$ respectively, with $i \leq k \leq (i + n)$ (detrended values);

in each box is calculated: $f^2_{DCCA}(n, i) = \frac{1}{1+n} \sum_{k=i}^{i+n} (X\alpha_k - \widetilde{X\alpha}_{k,i}) \times (X\beta_k - \widetilde{X\beta}_{k,i})$

for all the boxes of a time scale, the DFA is calculated as:

$F^2_{DCCA}(n) = \frac{1}{N-n} \sum_{i=1}^{N-n} f^2_{DCCA}(n, i)$;

for a number of different timescales (n), with possible values $4 \leq n \leq \frac{N}{4}$ the F^2_{DCCA} function is calculated to find a relation among $F^2_{DCCA} \times n$

Modeling count variables is a common task in economics and the social sciences. The classical Poisson regression model for count data is often of limited use in these disciplines because empirical count data sets typically exhibit overdispersion and/or an excess number of zeros. The former issue can be addressed by extending the plain Poisson regression model in various directions: e.g., using sandwich covariances or estimating an additional dispersion parameter (in a so-called quasi-Poisson model). Another more formal way is to use a negative binomial (NB) regression. All of these models belong to the family of generalized linear models (GLMs). However, although these models typically can capture overdispersion rather well, they are in many applications not sufficient for modeling excess zeros. Since **?** there is increased interest in zero-augmented models that address this issue by a second model component capturing zero counts. An overview of count data models in econometrics, including hurdle and zero-inflated models, is provided in **?**.

In R (**?**), GLMs are provided by the model fitting functions `glm()` in the **stats** package and `glm.nb()` in the **MASS** package (**?**, Chapter 7.4) along with associated methods for diagnostics and inference. The manuscript that this document is based on (**?**) then introduced hurdle and zero-inflated count models in the functions `hurdle()` and `zeroinfl()` in the **pscl** package (**?**). Of course, much more software could be discussed here, including (but not limited to) generalized additive models for count data as available in the R packages **mgcv** **?**, **gamlss** (**?**), or **VGAM** (**?**).

3. Zebende package: implementation and optimization

The basic Poisson regression model for count data is a special case of the GLM framework **?**. It describes the dependence of a count response variable y_i ($i = 1, \dots, n$) by assuming a Poisson distribution $y_i \sim \text{Pois}(\mu_i)$. The dependence of the conditional mean $\mathsf{E}[y_i \,|\, x_i] = \mu_i$ on the regressors x_i is then specified via a log link and a linear predictor

$$\log(\mu_i) \quad = \quad x_i^\top \beta, \tag{5}$$

where the regression coefficients β are estimated by maximum likelihood (ML) using the iterative weighted least squares (IWLS) algorithm.

> Note that around the `{equation}` above there should be no spaces (avoided in the LaTeX code by `%` lines) so that "normal" spacing is used and not a new paragraph started.

R provides a very flexible implementation of the general GLM framework in the function `glm()` (?) in the **stats** package. Its most important arguments are

```
glm(formula, data, subset, na.action, weights, offset,
family = gaussian, start = NULL, control = glm.control(...),
model = TRUE, y = TRUE, x = FALSE, ...)
```

where `formula` plus `data` is the now standard way of specifying regression relationships in R/S introduced in ?. The remaining arguments in the first line (`subset`, `na.action`, `weights`, and `offset`) are also standard for setting up formula-based regression models in R/S. The arguments in the second line control aspects specific to GLMs while the arguments in the last line specify which components are returned in the fitted model object (of class '`glm`' which inherits from '`lm`'). For further arguments to `glm()` (including alternative specifications of starting values) see `?glm`. For estimating a Poisson model `family = poisson` has to be specified.

> As the synopsis above is a code listing that is not meant to be executed, one can use either the dedicated `{Code}` environment or a simple `{verbatim}` environment for this. Again, spaces before and after should be avoided.
>
> Finally, there might be a reference to a `{table}` such as Table 1. Usually, these are placed at the top of the page (`[t!]`), centered (`\centering`), with a caption below the table, column headers and captions in sentence style, and if possible avoiding vertical lines.

4. Results

For a simple illustration of basic Poisson and NB count regression the `quine` data from the **MASS** package is used. This provides the number of `Days` that children were absent from school in Australia in a particular year, along with several covariates that can be employed as regressors. The data can be loaded by

```
R> data("quine", package = "MASS")
```

and a basic frequency distribution of the response variable is displayed in Figure 1.

> For code input and output, the style files provide dedicated environments. Either the "agnostic" `{CodeInput}` and `{CodeOutput}` can be used or, equivalently, the environments `{Sinput}` and `{Soutput}` as produced by `Sweave()` or **knitr** when using the `render_sweave()` hook. Please make sure that all code is properly spaced, e.g., using `y = a + b * x` and *not* `y=a+b*x`. Moreover, code input should use "the usual" command prompt in the respective software system. For R code, the prompt `"R> "` should be used

Type	Distribution	Method	Description
GLM	Poisson	ML	Poisson regression: classical GLM, estimated by maximum likelihood (ML)
		Quasi	"Quasi-Poisson regression": same mean function, estimated by quasi-ML (QML) or equivalently generalized estimating equations (GEE), inference adjustment via estimated dispersion parameter
		Adjusted	"Adjusted Poisson regression": same mean function, estimated by QML/GEE, inference adjustment via sandwich covariances
	NB	ML	NB regression: extended GLM, estimated by ML including additional shape parameter
Zero-augmented	Poisson	ML	Zero-inflated Poisson (ZIP), hurdle Poisson
	NB	ML	Zero-inflated NB (ZINB), hurdle NB

Table 1: Overview of various count regression models. The table is usually placed at the top of the page (`[t!]`), centered (`centering`), has a caption below the table, column headers and captions are in sentence style, and if possible vertical lines should be avoided.

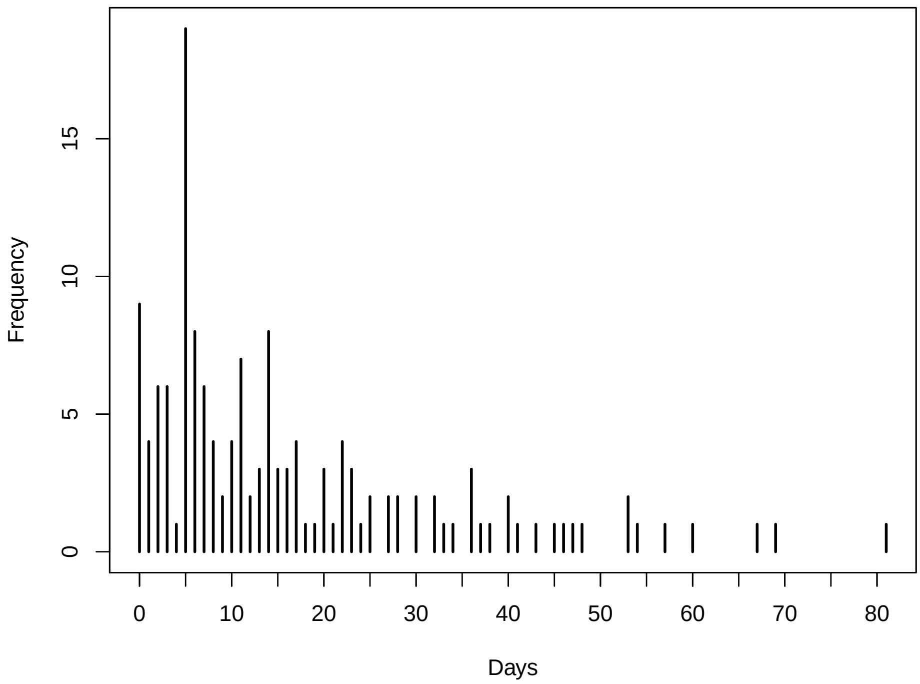

Figure 1: Frequency distribution for number of days absent from school.

with `"+ "` as the continuation prompt. Generally, comments within the code chunks should be avoided – and made in the regular LaTeX text instead. Finally, empty lines before and after code input/output should be avoided (see above).

As a first model for the quine data, we fit the basic Poisson regression model. (Note that JSS prefers when the second line of code is indented by two spaces.)

```
R> m_pois <- glm(Days ~ (Eth + Sex + Age + Lrn)^2, data = quine,
+    family = poisson)
```

To account for potential overdispersion we also consider a negative binomial GLM.

```
R> library("MASS")
R> m_nbin <- glm.nb(Days ~ (Eth + Sex + Age + Lrn)^2, data = quine)
```

In a comparison with the BIC the latter model is clearly preferred.

```
R> BIC(m_pois, m_nbin)

df      BIC
m_pois 18 2046.851
m_nbin 19 1157.235
```

Hence, the full summary of that model is shown below.

```
R> summary(m_nbin)

Call:
glm.nb(formula = Days ~ (Eth + Sex + Age + Lrn)^2, data = quine,
init.theta = 1.60364105, link = log)

Deviance Residuals:
Min       1Q   Median       3Q      Max
-3.0857  -0.8306  -0.2620   0.4282   2.0898

Coefficients: (1 not defined because of singularities)
Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.00155    0.33709   8.904  < 2e-16 ***
EthN        -0.24591    0.39135  -0.628  0.52977
SexM        -0.77181    0.38021  -2.030  0.04236 *
AgeF1       -0.02546    0.41615  -0.061  0.95121
AgeF2       -0.54884    0.54393  -1.009  0.31296
AgeF3       -0.25735    0.40558  -0.635  0.52574
LrnSL        0.38919    0.48421   0.804  0.42153
EthN:SexM    0.36240    0.29430   1.231  0.21818
EthN:AgeF1  -0.70000    0.43646  -1.604  0.10876
EthN:AgeF2  -1.23283    0.42962  -2.870  0.00411 **
EthN:AgeF3   0.04721    0.44883   0.105  0.91622
EthN:LrnSL   0.06847    0.34040   0.201  0.84059
SexM:AgeF1   0.02257    0.47360   0.048  0.96198
SexM:AgeF2   1.55330    0.51325   3.026  0.00247 **
```

```
SexM:AgeF3   1.25227    0.45539   2.750  0.00596 **
SexM:LrnSL   0.07187    0.40805   0.176  0.86019
AgeF1:LrnSL -0.43101    0.47948  -0.899  0.36870
AgeF2:LrnSL  0.52074    0.48567   1.072  0.28363
AgeF3:LrnSL       NA         NA      NA       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(1.6036) family taken to be 1)

Null deviance: 235.23  on 145  degrees of freedom
Residual deviance: 167.53  on 128  degrees of freedom
AIC: 1100.5

Number of Fisher Scoring iterations: 1


Theta:  1.604
Std. Err.:  0.214

2 x log-likelihood:  -1062.546
```

5. Summary and discussion

As usual …

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. Also, auxiliary packages (say, for visualizations, maps, tables, …) that are not cited in the main text can be credited here.

The results in this paper were obtained using R 3.4.1 with the **MASS** 7.3.47 package. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/.

Acknowledgments

All acknowledgments (note the AE spelling) 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).

References

Peng CK, Buldyrev SV, Havlin S, Simons M, Stanley HE, Goldberger AL (1994). "Mosaic Organization of DNA Nucleotides." **49**(2), 1685–1689.

Podobnik B, Stanley HE (2008). "Detrended cross-correlation analysis: A new method for analyzing two nonstationary time series." *Physical Review Letters*, **100**(8). ISSN 00319007. doi:10.1103/PhysRevLett.100.084102. 0709.0281.

Zebende GF (2011). "DCCA cross-correlation coefficient: Quantifying level of cross-correlation." *Physica A: Statistical Mechanics and its Applications*, **390**(4), 614–618. ISSN 03784371. doi:10.1016/j.physa.2010.10.022. URL http://dx.doi.org/10.1016/j.physa.2010.10.022.

Zebende GF, Silva AM (2018). "Detrended Multiple Cross-Correlation Coefficient." *Physica A*, **510**, 91–97. ISSN 0378-4371. doi:10.1016/j.physa.2018.06.119.

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*).

For more technical style details, please check out JSS's style FAQ at `https://www.jstatsoft.org/pages/view/style#frequently-asked-questions` which includes the following topics:

- Title vs. sentence case.
- Graphics formatting.
- Naming conventions.
- Turning JSS manuscripts into R package vignettes.
- Trouble shooting.
- Many other potentially helpful details…

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). This commands yield different formats of author-year citations and allow to include additional details (e.g., pages, chapters, …) in brackets. In case you are not familiar with these commands see the JSS style FAQ for details.

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

- JSS-specific markup (`\proglang`, `\pkg`, `\code`) should be used in the references.
- Titles should be 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.

Affiliation:

Fernando Ferraz Ribeiro
Universidade Federal da Bahia
Faculty of Achitecture
Universität Innsbruck
Universitätsstr. 15
6020 Innsbruck, Austria
E-mail: fernando.ribeiro@ufba.br
and
Centro Universitário Senai-Cimatec

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd