



rEDM: An R package for Empirical Dynamic Modeling and Convergent Cross Mapping

Hao Ye

University of
Florida

Adam Clark

German Centre for
Integrative Biodiversity
Research (iDiv)

Ethan Deyle

University of
California San Diego

George Sugihara

University of
California San Diego

Abstract

Empirical dynamic modeling (EDM) is an emerging non-parametric framework for modeling nonlinear dynamic systems. EDM is based on the mathematical theory of reconstructing attractor manifolds from time series data (Takens 1981). The **rEDM** package collects several EDM methods, including simplex projection (Sugihara and May 1990), S-map (Sugihara 1994), multivariate embeddings (Dixon, Milicich, and Sugihara 1999), convergent cross mapping (Sugihara, May, Ye, Hsieh, Deyle, Fogarty, and Munch 2012), and multiview embedding (Ye and Sugihara 2016). Here, we introduce the basic underlying theory, and describe the functionality of **rEDM**, using examples from both model simulations and real data.

Keywords: JSS, style guide, comma-separated, not capitalized, R.

1. Introduction: Count data regression in R

The introduction is in principle “as usual”. However, it should usually embed both the implemented *methods* and the *software* into the respective relevant literature. For the latter both competing and complementary software should be discussed (within the same software environment and beyond), bringing out relative (dis)advantages. All software mentioned should be properly `\cite{}`d. (See also Appendix B for more details on `BIBTEX`.)

For writing about software JSS 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 `LATEX` command as well. Below we also illustrate how abbreviations should be introduced and citation commands can be

employed. See the `LATEX` code for more details.

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** (?).

2. Models and software

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 $E[y_i | x_i] = \mu_i$ on the regressors x_i is then specified via a log link and a linear predictor

$$\log(\mu_i) = x_i^\top \beta, \quad (1)$$

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

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.

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.

3. Illustrations

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.

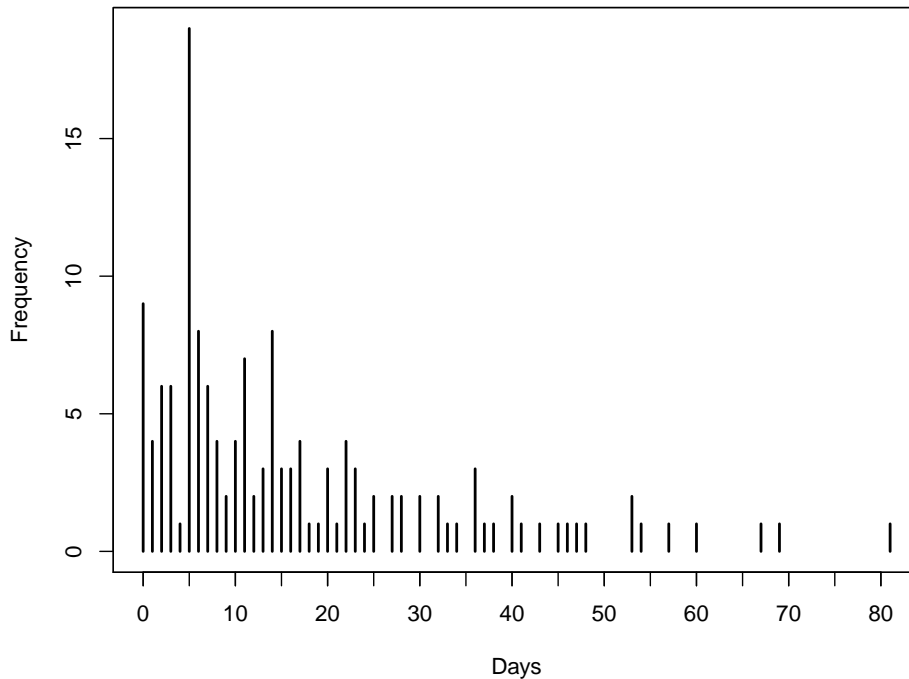


Figure 1: Frequency distribution for number of days absent from school.

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 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

```

4. 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 3.4.4 with the **MASS** 7.3.49 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

- Dixon PA, Milicich M, Sugihara G (1999). “Episodic fluctuations in larval supply.” *Science*, **283**, 1528–1530.
- Sugihara G (1994). “Nonlinear forecasting for the classification of natural time series.” *Philosophical Transactions: Physical Sciences and Engineering*, **348**(1688), 477–495.
- Sugihara G, May R, Ye H, Hsieh CH, Deyle E, Fogarty M, Munch S (2012). “Detecting causality in complex ecosystems.” *Science*, **338**, 496–500.
- Sugihara G, May RM (1990). “Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series.” *Nature*, **344**, 734–741.

- Takens F (1981). “Detecting strange attractors in turbulence.” *Dynamical Systems and Turbulence, Lecture Notes in Mathematics*, **898**, 366–381.
- Ye H, Sugihara G (2016). “Information leverage in interconnected ecosystems: Overcoming the curse of dimensionality.” *Science*, **353**(6302), 922–925.

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 BibT_EX

References need to be provided in a BibT_EX 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. In case you are not familiar with these commands see the JSS style FAQ for details.

Cleaning up BibT_EX 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:

Achim Zeileis
Journal of Statistical Software
and
Department of Statistics
Faculty of Economics and Statistics
Universität Innsbruck
Universitätsstr. 15
6020 Innsbruck, Austria
E-mail: Achim.Zeileis@R-project.org
URL: <https://eeecon.uibk.ac.at/~zeileis/>