

PACKAGE CRS FAQ

JEFFREY S. RACINE

CONTENTS

1. Overview and Current Version	2
2. Frequently Asked Questions	2
2.1. How do I cite the <code>crs</code> package?	2
2.2. I have never used R before. Can you direct me to some introductory material that will guide me through the basics?	3
2.3. How do I keep all R packages on my system current?	3
2.4. It seems that there are a lot of packages that must be installed in order to conduct econometric analysis (<code>tseries</code> , <code>lmtest</code> , <code>np</code> , etc.). Is there a way to avoid having to individually install each package individually?	3
2.5. Is there a ‘gentle guide’ to the <code>crs</code> package that contains some easy to follow examples?	4
2.6. I noticed you have placed a new version of the <code>crs</code> package on CRAN. How can I determine what has been changed, modified, fixed etc?	4
2.7. How can I read data stored in various formats such as Stata, SAS, Minitab, SPSS etc. into the R program?	4
2.8. Where can I get some examples of R code for the <code>crs</code> package in addition to the examples in the help files?	4
2.9. I would like more/less information displayed when conducting search using the NOMAD routines...	5
2.10. When I have a large number of regressors/data the function <code>crs</code> just ‘sits there’ when conducting cross-validation via NOMAD...	5
2.11. How can I save a PDF of a plot created with the option <code>persp.rgl=TRUE</code> ?	6
References	7
Changes from Version 0.14-9 to 0.15-0 [23-Jun-2011]	8
Changes from Version 0.14-8 to 0.14-9 [20-Jun-2011]	8
Changes from Version 0.14-7 to 0.14-8 [10-Jun-2011]	8
Version 0.14-7 [09-Jun-2011]	8

Date: July 15, 2011.

1. OVERVIEW AND CURRENT VERSION

This set of frequently asked questions is intended to help users who are encountering unexpected or undesired behavior when trying to use the `crs` package.

Kindly report any issues you encounter to me, and please include your code, data, version of the package and version of R used so that I can help track down any such issues (racinej@mcmaster.ca). And, of course, if you encounter an issue that you think might be of interest to others, kindly email me the relevant information and I will incorporate it into this FAQ.

This FAQ refers to the most recent version, which as of this writing is 0.15-0. Kindly update your version should you not be using the most current (from within R, `update.packages()` ought to do it, though also see 2.3 below.). See the appendix in this file for cumulative changes between this and previous versions of the `crs` package.

2. FREQUENTLY ASKED QUESTIONS

2.1. How do I cite the `crs` package? Once you have installed the `crs` package (`install.packages("crs")`), if you load the `crs` package (`library("crs")`) and type `citation("crs")` you will be presented with the following information.

```
> citation("crs")
```

To cite package `crs` in publications use:

```
Jeffrey S. Racine <racinej@mcmaster.ca> and Zhenghua Nie
<nierz@mcmaster.ca> (2011). crs: Categorical Regression Splines. R
package version 0.14-7.
```

A BibTeX entry for LaTeX users is

```
@Manual{,
  title = {crs: Categorical Regression Splines},
  author = {Jeffrey S. Racine and Zhenghua Nie},
  year = {2011},
  note = {R package version 0.14-7},
}
```

2.2. I have never used R before. Can you direct me to some introductory material that will guide me through the basics? There are many excellent introductions to the R environment with more on the way. First, I would recommend going directly to the R website (<http://www.r-project.org>) and looking under Documentation/Manuals (<http://cran.r-project.org/manuals.html>) where you will discover a wealth of documentation for R users of all levels. See also the R task views summary page (<http://cran.nedmirror.nl/web/views/index.html>) for information grouped under field of interest. A few documents that I mention to my students which are tailored to econometricians include <http://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf>, Cribari-Neto & Zarkos (1999) [1], Racine & Hyndman (2002) [5] and Farnsworth (2006) [3], to name but a few.

Those looking for exemplar data sets outside of those contained in the `crs` package are directed to the `Ecdat` [2] and `AER` [4] packages.

Often the best resource is right down the hall. Ask a colleague whether they use or know anyone who uses R, then offer to buy that person a coffee and along the way drop something like “I keep hearing about the R project... I feel like such a Luddite...”

2.3. How do I keep all R packages on my system current? Run the command `update.packages(checkBuilt=TRUE,ask=FALSE)`, which will not only update all packages that are no longer current, but will also update all packages built under outdated installed versions of R, if appropriate.

2.4. It seems that there are a lot of packages that must be installed in order to conduct econometric analysis (tseries, lmtest, np, etc.). Is there a way to avoid having to individually install each package individually? Certainly. The Comprehensive R Archive Network (CRAN) is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. The CRAN task view for computational econometrics might be of particular interest to econometricians. The econometric task view provides an excellent summary of both parametric and nonparametric econometric packages that exist for the R environment and provides one-stop installation for these packages.

See cran.r-project.org/web/views/Econometrics.html for further information.

To automatically install a task view, the `ctv` package first needs to be installed and loaded, i.e.,

```
install.packages("ctv")
library("ctv")
```

The econometric task view can then be installed via `install.views()` and updated via `update.views()` (which first assesses which of the packages are already installed and up-to-date), i.e.,

```
install.views("Econometrics")
```

or

```
update.views("Econometrics")
```

2.5. Is there a ‘gentle guide’ to the `crs` package that contains some easy to follow examples? Perhaps the most gentle introduction is contained in the `crs` package itself in the form of a ‘vignette’. To view the vignette run R, install the `crs` package (`install.packages("crs")`), then type `vignette("crs",package="crs")` to view or print the vignette.

See also `vignette("spline_primer",package="crs")` for a vignette that presents a ‘gentle’ introduction to regression splines.

For a listing of all routines in the `crs` package type: `library(help="crs")`.

2.6. I noticed you have placed a new version of the `crs` package on CRAN. How can I determine what has been changed, modified, fixed etc? See the CHANGELOG on the CRAN site (<http://cran.r-project.org/web/packages/crs/ChangeLog>), or go to the end of this document where the CHANGELOG is provided for your convenience.

2.7. How can I read data stored in various formats such as Stata, SAS, Minitab, SPSS etc. into the R program? Install the foreign library via `install.packages("foreign")` then do something like

```
mydat <- read.dta("datafile.dta"),
```

where `datafile.dta` is the name of your Stata data file. Note that, as of version 0.8-34, the foreign package function `read.dta` supports reading files directly over the Internet making for more portable code. For instance, one could do something like

```
mydat <- read.dta(file="http://www.principlesofeconometrics.com/stata/mroz.dta")
```

as one could always do with, say, `read.table()`.

2.8. Where can I get some examples of R code for the `crs` package in addition to the examples in the help files? Start R then type `demo(package="crs")` and you will be presented with a list of demos for constrained estimation, inference, and so forth. To run one of these demos type, for example, `demo(radial_rgl)` (note that you must first install the `rgl` package to run this particular demo).

To find the location of a demo type `system.file("demo", "radial_rgl.R", package="crs")` for example, then you can take the source code for this demo and modify it for your particular application.

2.9. I would like more/less information displayed when conducting search using the NOMAD routines... This is accomplished by feeding the argument `opts=list("DISPLAY_DEGREE"=x)` to `crs` where x is a non-negative integer. Setting $x=0$ produces no information whatsoever while integers $x \geq 1$ provide successively more information.

2.10. When I have a large number of regressors/data the function crs just ‘sits there’ when conducting cross-validation via NOMAD... First, if you are concerned that the code is indeed just ‘sitting there’, you can verify that search is progressing by changing the "DISPLAY_DEGREE" setting in the `opts` list along the lines of the following:

```
opts <- list("MAX_BB_EVAL"=10000,
            "EPSILON"=.Machine$double.eps,
            "INITIAL_MESH_SIZE"="r1.0e-01",
            "MIN_MESH_SIZE"=paste("r",sqrt(.Machine$double.eps),sep=""),
            "MIN_POLL_SIZE"=paste("r",sqrt(.Machine$double.eps),sep=""),
            "DISPLAY_DEGREE"=3)
```

```
model <- crs(...,opts=opts)
```

will print out in gory detail exactly what the search engine is doing.

However, if this reveals that there something odd going on (i.e. you are seeing a lot of `inf` function values being printed out), then you might wish to begin by restricting the dimension of the combinatoric search process. By default `degree.max=10` for each predictor and `segments.max=10` as well. So this can lead to a basis with 21 columns for one predictor and when using `basis="tensor"` or `basis="auto"` (which computes both the tensor and additive bases) the dimension of the basis can swamp the number of observations in the sample (e.g. with 4 regressors we can have a tensor product multivariate basis that has up to $21^4=194481$ columns when using the default `degree.max=10` for each predictor and `segments.max=10`). For this illustration, the cross-validation function can approach ∞ if the sample size approaches 194491 from above and the search process will be searching for a non- ∞ value in order to proceed or terminate.

So, in such cases begin by restricting the dimension of the spline basis matrix by setting, for instance, `degree.max=2` and `segments.max=2`. Or begin by searching only over the spline

degree by setting `complexity="degree"` (the default is `complexity="degree-knots"`). The routine will throw a warning if you have a solution that hits the maximum value of `degree.max` or `segments.max` and offer some practical advice in these cases.

Alternatively, restrict attention to additive (semiparametric) splines by setting `basis="additive"` (e.g. with 4 regressors we can have a tensor product multivariate basis that has up to $21 \times 4 = 84$ columns when using the default `degree.max=10` for each predictor and `segments.max=10`) at the cost of imposing additivity which can be restrictive.

Alternatively, consider kernel regression that does not suffer from this computational limitation (see e.g. the `np` package).

2.11. How can I save a PDF of a plot created with the option `persp.rgl=TRUE`?

Version 0.15-1 has added support for RGL via the `rgl` package which is a 3D real-time rendering device driver system for R using OpenGL. These plots are dynamic so you can spin them and resize them using your keypad/mouse. However, they are not standard graphics objects that can be saved using R commands such as `pdf()`. But they can be saved as a PDF by first calling `rgl` and then issuing the command `rgl.postscript("foo.pdf", "pdf")` where `foo.pdf` is the desired name of your PDF file as the following illustrates:

```
n <- 1000
x1 <- sort(rnorm(n))
x2 <- rnorm(n)
y <- x1^3 + rnorm(n,sd=.1)
model <- crs(y~x1+x2)
plot(model,mean=T,persp.rgl=T)
rgl.postscript("foo.pdf","pdf")
```

However, this pdf driver does not support some features such as transparency etc. A better alternative is to create a `png` file as follows:

```
n <- 1000
x1 <- sort(rnorm(n))
x2 <- rnorm(n)
y <- x1^3 + rnorm(n,sd=.1)
model <- crs(y~x1+x2)
plot(model,mean=T,persp.rgl=T)
rgl.snapshot("foo.png")
```

and then include this in your \LaTeX document using `\includegraphics[scale=.5]{foo.png}`.

REFERENCES

- [1] Francisco Cribari-Neto and Spyros G Zarkos. R: Yet another econometric programming environment. *Journal of Applied Econometrics*, 14(3):319–29, May-June 1999. Available at <http://ideas.repec.org/a/jae/japmet/v14y1999i3p319-29.html>.
- [2] Yves Croissant. *Ecdat: Data sets for econometrics*, 2006. R package version 0.1-5.
- [3] Grant V. Farnsworth. Econometrics in R. Technical report, June 2006. Available at <http://cran.r-project.org/doc/contrib/Farnsworth-EconometricsInR.pdf>.
- [4] Christian Kleiber and Achim Zeileis. *Applied Econometrics with R*. Springer-Verlag, New York, 2008. ISBN 978-0-387-77316-2.
- [5] J. S. Racine and R. Hyndman. Using R to teach econometrics. *Journal of Applied Econometrics*, 17(2):175–189, 2002.

CHANGES FROM VERSION 0.14-9 TO 0.15-0 [23-JUN-2011]

- Thanks to Professor Brian Ripley, compile on Solaris system issues are resolved, and check/examples are reduced in run time to alleviate the excessive check times by the R development team. Many thanks to them for their patience and guidance.
- Minor changes to `radial_rgl` demo

CHANGES FROM VERSION 0.14-8 TO 0.14-9 [20-JUN-2011]

- Cleaned up issues for creating binary for windows
- Setting seed in `snomadr.cpp` via `snomadr.R` for starting points when `nmulti > 0`
- Increased default `MAX_BB_EVAL` from 500 to 10000 (makes a difference for difficult problems) and modified default `EPSILON` in `NOMAD` along with other parameters (`MIN_MESH_SIZE`, `MIN_POLL_SIZE`) to reflect actual machine precision (using R's `.Machine$double.eps` where `NOMAD` fixed `EPSILON` at `1e-13`)
- Zhenghua added help functionality for retrieving help via `snomadr`
- Now default number of restarts in `crs` is 5 (zero is not reliable and I want sensible defaults in this package - higher is better but for many problems this ought to suffice)
- Corrected glitches in interactive demos where options were not being passed, updated docs to reflect demos

CHANGES FROM VERSION 0.14-7 TO 0.14-8 [10-JUN-2011]

- `crsiv` now returns a `crs` model object that supports residuals, fitted, predict and other generic functions. Note that this approach is based on first computing the model via regularization and then feeding a transformed response to a `crs` model object. You can test how close the two approaches are to one another by comparing `model$phihat` with `fitted(model)` via
`all.equal(as.numeric(fitted(model)), as.numeric(model$phihat))`

VERSION 0.14-7 [09-JUN-2011]

- Initial release of the `crs` package on CRAN.