Guide Sweave Instructions

Paul Johnson, CRMDA <pauljohn.edu>
Meghan Sullivan, CRMDA <second@ku.edu>

Keywords: single-authoring, just one
See https://crmda.ku.edu/guides for updates.

CENTER FOR
RESEARCH METHODS
& DATA ANALYSIS
College of Liberal Arts
& Sciences

Guide No: 48

February 13, 2018

1 Introduction

This shows how we use R (R Core Team, 2017) to make guide documents using the CRMDA style.

We suggest you

1. Compile this document *as is* to test your setup
2. Compare the document with the pdf output so see the impact of the settings.

Do put your title & name in the top block.

Don't change the code chunks above or the last chunks below.

Don't change margins or geometry with Lyx pull down menus. Edit preamble or config files for that.

Repeat **CAUTION**: Dont change the page margins or settings for hyperlinks with pull down menus.

2 Check our documentation

1. "rnw2pdf-guide-instructions" should be available in folder
2. "crmda": the package framework overview
3. "code_chunks": discusses display of code in LaTeX documents

3 Code Chunk Check

Illustrative R code can be included in the document. The author has a good deal of control over how, and at which, the input and output are displayed. Correctly formatted LaTeX code can be written by R functions and it can appear in the document. The vignette "code_chunks" has full details. This is a brief highlight.

3.1 The listings package

The document preamble includes settings for the LaTeX package listings, which is used to display code input and output. Inline references to `code` can be marked for highlighting (by LaTeX macro “\code”) that will mimic the color styling of the code displays.

One advantage of using our Sweave-based LaTeX documents is the listings class can handle very long lines (allows linewrap) and also lets us have fine grained control over the display of code input and output. In guide documents, we have ine numbers turned on. is used. Among its benefits, we get “line wrap” on long lines.

3.2 Regression example

Consider a regression.

```
dat <- data.frame(x = rnorm(100), y = rpois(100, lambda = 7))
m1 <- glm(y ~ x, data = dat, family = "poisson")
summary(m1)
```

```
Call:
glm(formula = y ~ x, family = "poisson", data = dat)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.35756  -0.80601  -0.01817   0.63433   2.78849

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.97217    0.03806  51.814   <2e-16 ***
x           -0.03514    0.03368  -1.043    0.297
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 112.78  on 99  degrees of freedom
Residual deviance: 111.70  on 98  degrees of freedom
AIC: 489.94

Number of Fisher Scoring iterations: 4
```

3.3 Controlling display of chunk input and output

We now want to conduct a poisson regression, one that treats `y` as a discrete variable. We will hide the estimation code from the reader for the moment. The chunk, which is in the paragraph following in the document, runs a regression and saves the LaTeX formatted output in an R object named `or` . However, this is not shown to the reader because the chunk option “include=F” is set. But the object `or` was created.

To display the object `or` to the reader, we can display it in a chunk using the standard Sweave approach, which depends on “results=tex”.

Table 1: A Poisson Regression

	My Poisson	
	Estimate	(S.E.)
(Intercept)	1.972***	(0.038)
A Normal Predictor	-0.035	(0.034)
N	100	
Deviance	111.697	
$-2LLR(Model\chi^2)$	1.087	

$*p \leq 0.05 ** \, p \leq 0.01 *** p \leq 0.001$

	My Poisson	
	Estimate	(S.E.)
(Intercept)	1.972***	(0.038)
A Normal Predictor	-0.035	(0.034)
N	100	
Deviance	111.697	
$-2LLR(Model\chi^2)$	1.087	

$*p \leq 0.05 ** \, p \leq 0.01 *** p \leq 0.001$

There are other ways, however, to display the same output code. We could, for example, simply use LaTeX to input the saved file from the chunk. Another possibility is that we might created a separate numbered “floating table”, as the reader would see in Table 1. There are plenty of other strategies for controlling the display of code and output chunks, as we illustrate in the next paragraph.

The reader has not yet had a chance to see the code chunk that calculated the regression. The code chunk that ran the regression was marked “include=F, results=hide”. We might want to show the reader what commands were run at some later point. There are (at least) 2 ways we do this on a regular basis.

1. The chunk is named “pois10”. Because we have the R Sweave argument split=T in the above, the code file is written separately and we can retrieve it with an input statement:

```
library(rockchalk)
or <- outreg(list("My Poisson"= m1), varLabels = c("x" = "A
    Normal Predictor"), tight = FALSE)
```

2. An old-school approach using Sweave itself, is to create another chunk and then display it inside double “<<>>” brackets. Here we turn off evaluation (set eval=F) to prevent R from re-running the code chunk:

```
library(rockchalk)
or <- outreg(list("My Poisson"= m1), varLabels = c("x" = "A
    Normal Predictor"), tight = FALSE)
```

Those two code displays should be identical in the document.

3.4 Line wrap

This chunk shows what happens if the R input long. The linewrapping prevents code input from running into the margin.

```
## Show very long variable names
Nisthesamplesamplesize <- 100
Misformuorthemean <- 10123
Sisforsigma <- 234234
x <- rnorm(Nisthesamplesamplesize, mean = Misformuorthemean, s =
    Sisforsigma)
head(x)
```

```
[1] -369178.75  138576.44   55864.72 -178786.25  -15320.49  -48657.23
```

```
mean(x)
```

```
[1] 28813.7
```

```
dataFrame <- data.frame(x1 = rnorm(100, m = 13, s = 23), x2 =
    rnorm(100, m = 13, s = 23), x3 = rnorm(100, m = 13, s = 23))
```

4 References

References

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

5 Session Info

```
R version 3.4.3 (2017-11-30)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 17.10

Matrix products: default
BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1
LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.7.1

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C               LC_TIME=en_US.UTF-8
 [4] LC_COLLATE=en_US.UTF-8     LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C                  LC_ADDRESS=C
[10] LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats     graphics  grDevices utils     datasets  base

other attached packages:
[1] rockchalk_1.8.110 crmda_0.53

```

```
loaded via a namespace (and not attached):
 [1] Rcpp_0.12.15       compiler_3.4.3     nloptr_1.0.4       plyr_1.8.4
     methods_3.4.3
 [6] tools_3.4.3        digest_0.6.15      lme4_1.1-15        evaluate_0.10.1
     nlme_3.1-131
[11] lattice_0.20-35    mgcv_1.8-23        openxlsx_4.0.17    Matrix_1.2-12
    parallel_3.4.3
[16] SparseM_1.77       pbivnorm_0.6.0     stringr_1.2.0      knitr_1.19
    MatrixModels_0.4-1
[21] stats4_3.4.3       rprojroot_1.3-2    nnet_7.3-12        grid_3.4.3
    foreign_0.8-69
[26] rmarkdown_1.8      lavaan_0.5-23.1097 minqa_1.2.4        car_2.1-6
    magrittr_1.5
[31] backports_1.1.2    htmltools_0.3.6    MASS_7.3-48        kutils_1.34
    splines_3.4.3
[36] pbkrtest_0.4-7     mnormt_1.5-5       xtable_1.8-2       quantreg_5.35
    quadprog_1.5-5
[41] stringi_1.1.6
```