Guide No: 48

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.

February 15, 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.

The listings class used here allows within-document style changes. We expect that report documents will not be customized by most authors, but guide documents are less formal. In order to make output fit within the indicated space, it may be necessary to fiddle with the font size, for example. Here are the highlights:

1. The font size and colors of R input chunks are controlled by LaTeX settings "Rsize", "Rbackground" and "Rcolor". Output displays depend on "Routsize", "Routbackground", and "Routcolor".

2. The fong can be adjusted by declarations like this

```
\def\Rsize{\huge\ttfamily}
\def\Routsize{\huge}
```

   These can be placed at the very beginning of the document to control all following chunks, but they can be placed immediately before any chunk to adjust just that one chunk.

3. Colors can be specified in many ways

```
\def\Rbackground{\color[gray]{0.90}}
\def\Routbackground{\color[gray]{0.40}}
\def\Rcolor{\color[gray]{0.60}
\def\Routcolor{\color[rgb]{0.9, 0.1, 0.1}]}
\def\Rcommentcolor{\color{green}}
```

To demonstrate this customization, compare these two chunks. The first uses the defaults:

```
x <- rnorm(100)
mean(x)
```

```
[1] 0.2451972
```

While the second offers a shockingly beautiful offering (emphasis on shockingly).

```
x <- rnorm(100)
mean(x)
```

```
[1] 0.2451972
```

Note we use a LaTeX group here–the squiggly braces–to confine the beautifying impact of the change to the immediately following output.

```
{
\def\Rbackground{\color[rgb]{0.4, 0.7, 0.6}}
\def\Routbackground{\color[rgb]{0.6, 0.5, 0.8}}
\def\Routcolor{\color{blue}}
\def\Routsize{\huge}
\input{tmpout/t-rnorm1.tex}
}
```

Otherwise, at least in this document type, the change applies to all following chunks.

3.2 Raw R input and output

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
-2.82337  -0.78233  -0.00208   0.70944   2.65257

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.94496    0.03784  51.395   <2e-16 ***
x           -0.01239    0.03763  -0.329    0.742
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 99.277  on 99  degrees of freedom
Residual deviance: 99.169  on 98  degrees of freedom
AIC: 476.03

Number of Fisher Scoring iterations: 4
```

3.3 Controlling display of chunk input and output

The chunk option "include" can be used to regulate whether the input and output appear immediately in the document. When combined with "echo" and "results", we can have a great deal of control.

In this chunk, we create a regression table object, but we hide everything:

To display the object `or` to the reader, we have two options.

The standard Sweave approach is to include another chunk, and then cause the LaTeX markup for the object `or` to be woven directly into the document (depends on "results=tex").

	My Poisson	
	Estimate	(S.E.)
(Intercept)	1.945***	(0.038)
A Normal Predictor	-0.012	(0.038)
N	100	
Deviance	99.169	
$-2LLR(Model\chi^2)$	0.108	

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

That chunk might be placed in a LaTeX floating table object, of course.

A second option, which is more convenient, is to simply use LaTeX to input the saved file. When the code chunk "pois11" is excuted, it creates a file named "tmpout/t-pois11.tex". In Table 1 we demonstrate how that can be included in a numbered floating table.

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". To show that code, there are (at least) 2 ways.

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 standard 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.

Table 1: A Poisson Regression

	My Poisson	
	Estimate	(S.E.)
(Intercept)	1.945***	(0.038)
A Normal Predictor	-0.012	(0.038)
N	100	
Deviance	99.169	
$-2LLR(Model\chi^2)$	0.108	

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

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] -179295.98  244619.31  116946.00 -325827.19  -52020.41  160447.16
```

```
mean(x)
```

```
[1] 28585.68
```

```
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_0.54

```

```
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
```