Instructions for guides using knitted code chunks

Johnson Paul, Directior, CRMDA
<pauljohn@ku.edu>
Second Author, CRMDA <second@ku.edu>

Guide No: 48

Keywords: stationery, guide, knitr
See https://crmda.ku.edu/guides for updates.

February 22, 2018

Abstract

This is an abstract. Please include a terse, yet descriptive statement here of less than 200 words. It should avoid colloquialisms and polysyllabic profundities. An abstract is required.

Contents

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 the skeleton document *as is* to test your setup
2. Make revisions incrementally, and re-compile often to make sure nothing has been broken.

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

2 LyX: Cautions

The document preamble has manual settings for margins (geometry) as well as hyperlinks (PDF hyperref). Don't use the LyX pull down menu to revise them. It is necessary to edit settings in the preamble manually.

3 What to edit

Title and author information

The first block in the document has the title and author information.

Footer information

The footer in this document uses data that is provided in a file named "addressFooter.tex". After the document is compiled for the first time, that document should be available in the theme folder.

About the theme folder

The theme folder should be empty when the initProject() function is run.

There is an R code chunk above called "texcopy". It will copy configuration files from the package into the theme folder. After running this for the first time, those files will not be automatically replaced by the scripts.

That means authors are free to edit them to fit their needs.

If the author makes an error in editing a theme file, it is safe to delete the erroneous file and run the compile script again. That will copy a fresh version of the theme file into the directory.

4 Check our documentation

There are several vignettes distributed with this package. Please review them.

1. "stationery": the package framework overview
2. "code_chunks": discusses display of code in LaTeX documents

5 Compile as usual, or with rnw2pdf

If you are editing these files in LyX, it is sufficient to simply compile as usual. That will handle the chore of converting a sequence of document types to arrive at PDF.

If not using LyX, then the author is probably editing the Rnw file. The Rnw file we provide is produced by LyX, it is an intermediate step in the document production sequence. A two step

compilation procedure is necessary. First, one must convert the "Rnw" file to "pdf" (with knit), and then the knitted tex file is compiled into pdf by pdflatex (or one of the other LaTeX compilers).

We provide a shell script that can handle this, `rnw2pdf.sh` script (which is included with the skeleton file). It is also possible to use our R function `rnw2pdf`.

6 Code Chunk Check

6.1 knitr issues/features

The original R (R Core Team, 2017) approach to combining code, output within the document is called Sweave. The knitr approach is an alternative package. knitr is helpful especially because it can be used with R markdown to produce HTML web pages.

Styling of knitr code chunks is different than Sweave. It appears we lose line-wrap entirely. I can't figure how to make Sweavel "listings" environments take over the knitr presentation.

We have not yet learned the ins-and-outs of correcting the knitr code and R output chunks to match our desired style. That is one reason to prefer the sweave-based templates we offer.

6.2 Code Chunks

If you choose to use knitr rather than Sweave as the chunk-processor–and by using this template that is what you did decide–you should study the knitr chunk options. These are discussed in some detail in the vignette on code chunks.

Here is an example of a data frame being created and a glm is estimated:

```
set.seed(234234)
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
-2.2212  -0.7115  -0.1668   0.4829   2.4448

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.93581    0.03821  50.658   <2e-16 ***
x            0.03175    0.04596   0.691     0.49
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 111.39  on 99  degrees of freedom
Residual deviance: 110.91  on 98  degrees of freedom
AIC: 485.89

Number of Fisher Scoring iterations: 4
```

Some code that might be used to create a regression table using outreg from the rockchalk package is the following. As you see, the output "splats" into the document, it is not in a floating figure or table.

```
library(rockchalk)
or <- outreg(list("My Poisson"= m1), varLabels = c("x" = "A Normal Predictor"), tight = FALSE)
cat(or, file="tmpout/t-pois15.tex")
cat(or)
```

	My Poisson	
	Estimate	(S.E.)
(Intercept)	1.936***	(0.038)
A Normal Predictor	0.032	(0.046)
N	100	
Deviance	110.909	
$-2LLR(Model\chi^2)$	0.477	

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

It is possible, however, to input the same chunk into a table, as we have done in Table 1. That is the typically-preferred format for presentation of tables in guides.

Table 1: A Poisson Regression

	My Poisson	
	Estimate	(S.E.)
(Intercept)	1.936***	(0.038)
A Normal Predictor	0.032	(0.046)
N	100	
Deviance	110.909	
$-2LLR(Model\chi^2)$	0.477	

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

7 References

References

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

8 Session Info

Please leave this section for guide documents. It is important for replication.

```
Error:  <text>:4:16:  unexpected symbol
3:    print("Warnings:")
4:    warnings()}utreg
^
```