

Meghan Sullivan, CRMDA <[first@ku.edu](mailto:first@ku.edu)>  
Second Author, CRMDA <[second@ku.edu](mailto:second@ku.edu)>  
Third Author, i <[third@ku.edu](mailto:third@ku.edu)>  
February 21, 2018

---

## 1 Introduction

This is a LaTeX “noweb” report generated with the knitr engine. We had a working document using Sweave and the listings class (via the Sweavel LaTeX package). This is migrated from that to use knitr. Some elements are still not well integrated, it is very tough for me to be sure which argument in the preamble are actually relevant to knitr and which are relevant to listings by itself.

This shows how we use R [?] to make report 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.

For what it is worth, we have found that the knitr framework is not so easy to customize and manage as Sweave and, if you are planning to produce a PDF document, it seems likely that the

## 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.

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.

## 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. “crmda”: 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

The syntax for knitr code chunks is slightly different than the syntax for Sweave chunks. Many of the options are identical. The syntax for a code chunk in both cases is like so

```
\input{tmpout/t-chunkname.tex}
```

where “T” stands for TRUE and it can be changed to “F” for FALSE. These are explained in the vignette “code\_chunks”. Briefly:

1. **echo:** when the chunk is processed, the R code will be displayed along with the output
2. **include:** the chunk is evaluated, but neither the echo nor the results are displayed
3. **eval:** if F, the chunk is not sent to the R session, but it is checked for correctness of code at compile time.

The difference with knitr is partly due to a larger set of legal flags but also because knitr provides different functions to handle the document processing. One benefit of knitr is that, at least in theory, it can be used to write about other kinds of programs. I’ve used it to write about the BASH shell, for example.

### Reports and the Non-use of Code Chunks

Here is the main issue. The “automatic” “self-documenting” report is almost never exactly correct. It requires some “finger painting”. Perhaps that would not be true in the perfect world, if we were perfect programmers, but I don’t live in a perfect world. In that case, I am willing to settle for a very good second best alternative. I have learned to export result files from R which can be included in documents easily, so the documents are easily updated and revised. They may not be completely automatic because I leave open the possibility that I might need to adjust a figure’s size or a part of a table’s markup.

The magic bullet in that is the “split” option for Sweave, which knitr for LaTeX also supports. If split=T, then the individual code chunks and their results are saved in separate files. Here we save them in a directory called “tmpout”.

What is the difference between a guide and a report? Simply put, a report document does not reveal source code and it should not distract the reader with code. A report document might just as well be typed by hand, if we could be sure all the numbers would be typed correctly and they could easily revised.

In a report, we do not usually have R output in the raw form

```
dat <- data.frame(x = rnorm(100), y = rnorm(100))
m1 <- lm(y ~ x, data = dat)
summary(m1)
```

Call:

```
lm(formula = y ~ x, data = dat)
```

```

Residuals:
      Min       1Q   Median       3Q      Max
-2.20347 -0.60278 -0.01114  0.61898  2.60970

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.02205     0.10353   0.213   0.832
x            0.09454     0.09114   1.037   0.302

Residual standard error: 1.011 on 98 degrees of freedom
Multiple R-squared:  0.01086, Adjusted R-squared:  0.0007664
F-statistic: 1.076 on 1 and 98 DF,  p-value: 0.3022

```

That is output from knitr’s code decoration setup, incidentally. I’d rather not have the color coded R code and I don’t want light gray output. Those are settings we need to fix before we go much further down this road.

In a report, we do not want R code or output, we want a professional-looking table. This table may not be perfect by APA standards, but it is certainly good enough for our reports.

	First Model	
	Estimate	(S.E.)
(Intercept)	0.022	(0.104)
Excellent Predictor	0.095	(0.091)
N	100	
RMSE	1.011	
$R^2$	0.011	
$*p \leq 0.05$ $**p \leq 0.01$ $***p \leq 0.001$		

In the kutils package, we made a function `semTable` that is intended to help with nice looking. Usually, we want a bigger table in a floating element. Please see Table 1. In the preamble of this document, I have some boilerplate code that causes content in floating tables and figures to be centered automatically. This looks nice!

## 7 Session Information

Leave the code chunks below. But the visible words and section name should be removed. Session Information is usually not written into a report, but an output file is created by the following pieces.

Table 1: A Confirmatory Factor Analysis Table  
Model

	Estimate	Std. Err.	z	p
<u>Factor Loadings</u>				
<u>visual</u>				
x1	0.90	0.08	11.13	.000
x2	0.50	0.08	6.43	.000
x3	0.66	0.07	8.82	.000
<u>textual</u>				
x4	0.99	0.06	17.47	.000
x5	1.10	0.06	17.58	.000
x6	0.92	0.05	17.08	.000
<u>speed</u>				
x7	0.62	0.07	8.90	.000
x8	0.73	0.07	11.09	.000
x9	0.67	0.07	10.30	.000
<u>Latent Variances</u>				
visual	1.00 <sup>+</sup>			
textual	1.00 <sup>+</sup>			
speed	1.00 <sup>+</sup>			
<u>Fit Indices</u>				
RMSEA	0.09			

<sup>+</sup>Fixed parameter