RRRR Ch 11: Presenting with *knitr* and LaTeX

Andy Kaempf

September 21, 2016

Outline I

Outline II

Intro to LaTeX

LaTeX is a markup language and document preparation program (macro package) that utilizes the TeX typesetting system.

RStudio is a fully-functioning LaTeX editor from which you can compose a LaTeX source file. Source files are compiled to produce a PDF (or EPS) typeset document.

Other LaTeX editors are listed and compared here: http://en.wikipedia.org/wiki/List_of_text_editors.

Gandrud used the 'Sublime Text' editor when writing LaTeX markup for his book (Gandrud 2015). Some popular LaTeX editors include: TeXmaker, TeXstudio, TeXworks (included in MiKTeX), Overleaf, and the TeXlipse plug-in for the Eclipse IDE.

LaTeX document types

LaTeX **document classes** are assigned with the \documentclass command and specify how the compiled output will be formatted. This command is the first line of the LaTeX source file.

Types of document classes:

- article (no chapters)
- report
- book
- letter
- beamer (slideshow)

Document class **options** are specified inside of brackets:
\documentclass[(option1),(option2)]{class_name}

Document structure

LaTeX documents consist of 2 main parts: **preamble** and **body**

1. **preamble**: required in all LaTeX documents, this is where you define the document class, load packages, and set styles
   - include \setbeamertemplate{caption}[numbered] in the preamble to show caption numbers in 'float' environments

2. **body**: this is where the content of your document is entered. The body is represented by the *document* environment (see slide 8) and delimited by the commands \begin{document} and \end{document}

LaTeX Packages

A **LaTeX package** is composed of one or more files (usually .sty) containing environment definitions and macros (Rensselaer Polytechnic Institute 2007)

Although many packages come installed with TeX distributions, **extra packages** can be loaded for additional functionality

Packages are **loaded** with the command: \usepackage[options]{package_name}. Place these commands under the \documentclass{} line in the preamble

Examples of packages loaded for this .Rnw file:

- ▶ graphicx – for inserting image files
- ▶ fancyvrb – needed for *Verbatim* enviro.
- ▶ biblatex – for bibliographies with "biber" backend

See the Comprehensive TeX Archive Network package page or the Wikibooks page for lists of packages with short descriptions

LaTeX Environments

Environments are text blocks inside of which special formatting and features exist. Click here for a list of environments.

Begin an environment like this: \begin{environment_name}

End an environment like this: \end{environment_name}

LaTeX environment examples:

- *document* to define a document's body
- *frame* for a new slideshow frame
- *itemize* and *enumerate* for lists
- *flushleft*, *center*, *flushright* to justify text or displays
- *tabular* to generate (hard-code) tables
- *table* and *figure* are 'float' environments
- *quotation* and *quote* for text indented on both sides

General syntax: part 1

- The LATEX **comment** character is a percent sign: %

- Like in R, the LATEX **escape** character is a backslash: \
  - to print a backslash \, use \textbackslash command

- The **new line** character is \\. Add **vertical space** with the command \vspace{distance}, where an example 'distance' is 2mm. I have added 2 mm of vertical space after this line.

- Create **negative space** with commands \! (thin minus space) and \kern-1em (larger 'quad' minus space)

- Add **horizontal space** between words with the command \hspace{distance}. In this sentence I have added 5 mm of horizontal space between these last two   words.

General syntax: part 2

- LATEX commands begin with a **backslash** \ and arguments are included inside of **curly braces** {}. Unlike in R, a command/function can be invoked without including the braces after its name if no arguments are being passed.

- To print LATEX syntax without it being compiled use **verb**, where the command or special character you want to display but not execute is placed between a pair of the same characters, such as #
  - eg, to display the opening command in a Beamer slideshow, include the code: \verb#\documentclass{beamer}#

- To print and format blocks of text **ignored by LATEX** use the *Verbatim* environment from the *fancyvrb* package

Document layout commands: part 1

Section-level headings are added to the **navigation bar**[1] with the \section{} command. In Beamer docs, this command does **not** create a frame heading (the \frametitle{} command does)

- ► the section heading command for this frame is:
  \section{Layout commands: part 1}
- ► the frame title command is:
  \frametitle{Document layout commands: part 1}

Sub-section-level headings are included with these commands

- ► \subsection{}
- ► \subsubsection{}
- ► \paragraph{}
- ► \subparagraph{}

'subsection' headings also appear in the **navigation bar**

---

[1]called 'Bookmarks' in Adobe Acrobat and SumatraPDF

Document layout commands: part 2

New paragraphs are created in the compiled PDF by adding a **blank line** in between the markup document lines or using the \par command.

However, adding more than one blank line in the markup will not increase the space between paragraphs in the compiled document. Instead, use the **vspace**{} command

The command **hrulefill** creates horizontal lines in your compiled document like this:

Use **hline** to create a horizontal line inside the $tabular$ environment

Lists and adding pauses

There are three main **list environments**:

Lists and adding pauses

There are three main **list environments**:

Here is the code that created the above (enumerated) list:

```
\begin{enumerate}
\item<3 -> \texttt{\emph{enumerate}}
\item<4 -> \texttt{\emph{itemize}}
\item<5 -> \texttt{\emph{description}}
\end{enumerate}
```

Note the "< ->" after the \item commands. Numbers inside these angle brackets specify the order that list items appear. Pauses (like before 'Here is the code...') in a frame are added with \pause

Lists and adding pauses

There are three main **list environments**:

1. *enumerate*

Here is the code that created the above (enumerated) list:

```
\begin{enumerate}
\item<3 -> \texttt{\emph{enumerate}}
\item<4 -> \texttt{\emph{itemize}}
\item<5 -> \texttt{\emph{description}}
\end{enumerate}
```

Note the "< ->" after the \item commands. Numbers inside these angle brackets specify the order that list items appear. Pauses (like before 'Here is the code...') in a frame are added with \pause

Lists and adding pauses

There are three main **list environments**:

1. *enumerate*
2. *itemize*

Here is the code that created the above (enumerated) list:

```
\begin{enumerate}
\item<3 -> \texttt{\emph{enumerate}}
\item<4 -> \texttt{\emph{itemize}}
\item<5 -> \texttt{\emph{description}}
\end{enumerate}
```

Note the "< ->" after the \item commands. Numbers inside these angle brackets specify the order that list items appear. Pauses (like before 'Here is the code...') in a frame are added with \pause

Lists and adding pauses

There are three main **list environments**:

1. *enumerate*
2. *itemize*
3. *description*

Here is the code that created the above (enumerated) list:

```
\begin{enumerate}
\item<3 -> \texttt{\emph{enumerate}}
\item<4 -> \texttt{\emph{itemize}}
\item<5 -> \texttt{\emph{description}}
\end{enumerate}
```

Note the "< ->" after the \item commands. Numbers inside these angle brackets specify the order that list items appear. Pauses (like before 'Here is the code...') in a frame are added with \pause

Text display: part 1

Font options

- *Italicize text* with either of theses commands:
    - \emph{Italicize text}
    - \textit{Italicize text}

- Give text a **bold font** like so: \textbf{bold font}

- Specify font size by wrapping the **size declaration** in curly braces like so: {\size_name{...}}

- List of **font size names**: Huge, huge, LARGE, Large, large, normalsize, small, footnotesize, scriptsize, tiny.

    - make text LARGE with command: {\LARGE{LARGE}}
    - make text tiny with command: {\tiny{tiny}}

Text display: part 2

Font Styles

- use command \textrm{} for roman font
- use command \textsf{} for sans serif font
- use command \texttt{} for typewriter font
- use command \uppercase{} for UPPER CASE TEXT

Diacritics

- \' for an **accute accent**, such as in the word: financé
- \c for a **cedilla**, such as in the word: façade
- \~ for a **tilde**, such as over the 'n' in the word: piñata
- \" for a **diaeresis**, such as over the 'i' in the word: naïve

Resource: LaTeX Wikibook page on diacritics

Math in LaTeX: part 1

Insert **inline math equations** in 3 ways:

1. place the formula between a pair of dollar "$" signs
2. place the formula after a "\(" and before a "\)"
3. use the $math$ environment by placing the short formula between the commands: \begin{math} and \end{math}

The **inline equation** for sample variance is written as
$s^{2} = \frac{\sum(x - \bar{x})^2}{n - 1}$ and
displayed as: $s^2 = \frac{\sum(x-\bar{x})^2}{n-1}$

Math in LaTeX: part 2

Insert **block math equations** by placing the formula:

1. between a pair of double dollar "$$" signs
2. between a "\[" and a "\]"
3. inside the $displaymath$ or $equation$ environments
   - $equation$ adds an equation number on the right edge

The following block equation is **written** as:

```
\[
\frac{d}{dx}\left(\int_{0}^{x} f(u)\,du\right)=f(x)
\]
```

...and is **printed** in the typeset PDF file as:

$$\frac{d}{dx}\left(\int_0^x f(u)\,du\right) = f(x)$$

Math in LaTeX: part 3

Here's an example of typesetting a **block math equation** using the $displaymath$ environment.

$$\lim_{x \to a} \frac{f(x) - f(a)}{x - a}$$

The syntax to type this definition of a derivative is:

```
\begin{displaymath}
\lim_{x \to a} \frac{f(x) - f(a)}{x - a}
\end{displaymath}
```

Math in LaTeX: part 4

List of selected **math commands**:

bold text:	`$\mathbf{A}$`	$\longrightarrow$	$\mathbf{A}$
bold symbol:	`$\boldsymbol{\alpha}$`	$\longrightarrow$	$\boldsymbol{\alpha}$
'Hat' notation:	`$\widehat{a}$`	$\longrightarrow$	$\widehat{a}$
'Proportional to':	`$\propto$`	$\longrightarrow$	$\propto$
'Not equal to':	`$\neq$` or `$\ne$`	$\longrightarrow$	$\neq$
'If and only if':	`$\iff$`	$\longrightarrow$	$\Longleftrightarrow$
Definite integral:	`$\int_{a}^{b}$`	$\longrightarrow$	$\int_{a}^{b}$
Indef. integral:	`$\int^{+\infty}_{-\infty}$`	$\longrightarrow$	$\int^{+\infty}_{-\infty}$
Summation:	`$\sum_{i=1}^{n}$`	$\longrightarrow$	$\sum_{i=1}^{n}$
Product of terms:	`$\prod_{i=1}^{n}$`	$\longrightarrow$	$\prod_{i=1}^{n}$

See Appendix B (page 165) of *Practical LaTeX* by Grätzer (2014) for math symbol tables

Footnotes

LaTeX **footnotes** are inserted using the \footnote{} command at the place in the markup text where you want the footnote number to appear.

Here's an example: Footnotes are discussed on page 225 of the Gandrud textbook.[2] The footnote's text (displayed at the bottom of the slide) is the argument for this command.

---

[2]"Reproducible Research with R and RStudio". Christopher Gandrud. CRC Press, 2nd ed. 2015.

Cross-referencing an image: part 1

Create a figure **cross-reference** with \label{} and reference its number and page with \ref{} and \pageref{}, respectively

Insert a **non-knitted image** like Figure 1 with template code \includegraphics[options]{file path}. From within the $figure$ environment, a title is given with command \caption{}

Figure 1: Walrus on an ice floe

Source: Wikipedia 2016

Cross-referencing an image: part 2

The walrus image on slide 25 was inserted with the code:

```
\begin{figure}
    \begin{center}
        \caption{Walrus on an ice floe}
        \label{walrus}
        \includegraphics[width=2in]{Walrus.jpg}
    \end{center}
    {\footnotesize{Source: \cite{walrus_image}}}
\end{figure}
```

Note 1: the \cite{} command is introduced on slide 36 when discussing BibTeX and .bib file entries

Note 2: tables (within *table* enviro.) and document locations (eg, Beamer slides) can also be **cross-referenced** with \label{}

Inserting a non-knitted image

Placing an image within a *figure* environment is not necessary and is less common for the **beamer** class than for an **article** or **report**. The following image is inserted in the *frame* environment with two commands: \centering and \includegraphics{}

LaTeX commands for links

Use command **url**{**http...**} to include a clickable hyperlink for a webpage. The lone argument is the URL. For example, the GitHub page for *Reproducible Research with R and RStudio* is here: (https://github.com/christophergandrud/Rep-Res-Examples/)

Use command **href**{**http...**}{**some text**} for text to be a hyperlink. The Netherlands Tex users group has collated a **comprehensive list** of LaTeX commands. If you hover over 'comprehensive list' you will see the URL of the link

R inline code

Insert **inline R code** in a .Rnw file with the **Sexpr**{} command:

For example, using the 'mpg' dataset loaded with the *ggplot2* package, the average fuel economy for city driving among 4-cylinder cars is 21.01 miles per gallon.

The R code passed as an argument to **Sexpr**{} that computes and displays this average fuel economy is:

```
round(mean(mpg[mpg$cyl==4,]$cty),digits=2)
```

R code chunks

Insert **R code blocks**, called **chunks**, in a .Rnw file by placing the code between "<< >>=" and "@". Code chunk labels and options are placed within the double angle brackets like this:

```
<<random_norm, echo=2:4, collapse=TRUE>>=
```

```
sample <- rnorm(100)
round(mean(sample),digits=3)

## [1] 0.039

round(sd(sample),digits=3)

## [1] 1.058
```

```
@
```

When inserting code chunks in Beamer frames, make sure to specify the 'fragile' option like so: \begin{frame}[fragile]

Knitting a table with xtable

Use R package *xtable* to create **Table 1** of regression coeff. using 'Cars93' dataset that comes loaded with *MASS* package.

Each observation is a car on sale in the U.S. in 1993. Fit a linear reg. model where the dependent varible is car price and the regressors are city mpg, drive train (3 levels), and engine size.

```
model <- lm(Price ~ MPG.city + DriveTrain + EngineSize,
            data=Cars93)
print(xtable(model, caption="Linear regression results",
      label='xtable', digits=2), size='small')
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	24.95	6.66	3.74	0.00
MPG.city	-0.63	0.19	-3.26	0.00
DriveTrainFront	2.09	2.50	0.84	0.41
DriveTrainRear	7.77	3.08	2.52	0.01
EngineSize	2.20	1.10	1.99	0.05

Table 1: Linear regression results

Table 2 estimates pertain to modeling U.S. state populations in 1975. The regressors are Frost (mean days with min temp below freezing), Area (sq miles of land), and Income (per capita)

	Model 1	Model 2	Model 3
(Intercept)	7226.42***	7092.14***	−2365.52
	(1361.61)	(1442.25)	(4443.20)
Frost	−28.53*	−28.74*	−34.48**
	(11.69)	(11.83)	(11.64)
Area		0.00	−0.00
		(0.01)	(0.01)
Income			2.36*
			(1.05)
R^2	0.11	0.11	0.20
Adj. R^2	0.09	0.07	0.15
Num. obs.	50	50	50
RMSE	4254.66	4295.36	4122.76

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Table 2: *texreg* table of nested models

Knitting a table with texreg: part 2

The **code chunk** that created the *texreg* Table 2 is shown below. Important **options** for the chunk are:

- results = 'asis'
- message = FALSE

```
state_df <- data.frame(state.x77)
M1 = lm(Population ~ Frost, data=state_df)
M2 = lm(Population ~ Frost + Area, data=state_df)
M3 = lm(Population ~ Frost + Area + Income, data=state_df)
texreg(list(M1,M2,M3),
       caption='\\emph{texreg} table of nested models',
       scalebox=0.85, label='texreg')
```

The texreg() '**scalebox**' value shrinks the table text by 15%.

The '**label**' value is the text to use as the argument for the \ref{} command to cite the table's number

Knitting a figure with base R

This chunk inserts default R box-and-whisker plots (Figure 2)

```
<<boxplots, out.width='3in', out.height='2in'>>=
boxplot(mpg ~ cyl, data=mtcars, main="")
@
```
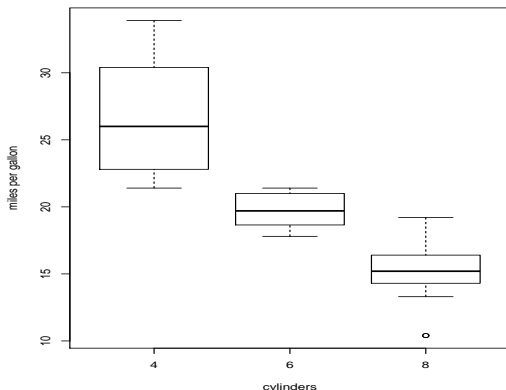

Figure 2: Fuel economy by cylinder number

Knitting a figure with ggplot2

Knit and scale a scatterplot with fitted loess curve (Figure 3)

```
<<scatter, out.width='2.4in', out.height='1.8in'>>=
ggplot(data=mpg, aes(x = displ, y=hwy)) +
        geom_point() + geom_smooth(span=0.7)
@
```
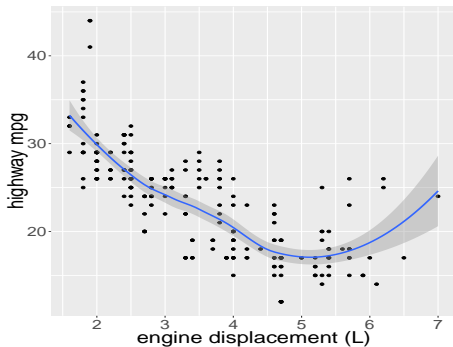

Figure 3: Scatterplot of engine size (x-axis) vs. mpg (y-axis)

BibTeX citations: part 1

BibTeX is a tool for creating databases of citation information.

BibTeX bibliographies are stored in **.bib files**, which are plain-text files that can be created with any text editor. In the preamble, type \addbibresource{file_name.bib} to access the BibTeX file whose entries you plan on citing.

Use **cite** (and other commands[3]) to embed citations in the compiled document. The argument to this command is the unique **citation key** defined in a .bib file entry (after the @).

References utilized in this document include a *ggplot2* article (Wickham 2011) and LaTeX book (Grätzer 2014), both listed in the **Bibliography** slide by using \printbibliography

---

[3]\citetitle{}, \citeauthor{}, \citeyear{}

BibTeX citations: part 2

Here is a **BibTeX entry** written in a .bib file, which can be made accessible via the preamble commands **addbibresource**{} or \bibliography{}, where the argument is the .bib file

```
@book{Gandrud2015,
title={Reproducible Research with R and RStudio},
author={Gandrud, Christopher},
publisher={CRC Press},
year={2015}
}
```

cite{**Gandrud2015**} prints this citation: Gandrud 2015 because 'style=authoryear' was specified when loading *biblatex*

Other citation 'style' options for the *biblatex* package include 'numeric', 'authortitle', and 'verbose'.

BibTeX citations: part 3

A separate **BibTeX** (.bib) file was created for citing the R
packages used in this .Rnw file by including the following
commands in a **code chunk** from an earlier frame:

```r
# vector of R packages used in knittable document
R_packages <- c("ggplot2","knitr","xtable","texreg")

# load R packages into memory
lapply(R_packages, library, character.only=TRUE)

# create a BibTeX file for R package citations
knitr::write_bib(R_packages, file = "Packages.bib")
```

This **BibTeX** file is created in the folder that the .Rnw resides in.

Bibliography

Gandrud, Christopher (2015). *Reproducible Research with R and RStudio*. 2nd. CRC Press.

Grätzer, George (2014). *Practical LaTeX*. Springer.

Rensselaer Polytechnic Institute (2007). *Text Formatting with LaTeX*. `http://www.rpi.edu/dept/arc/docs/latex/latex-intro.pdf`. Academic and Research Computing.

Wickham, Hadley (2011). "ggplot2". In: *Computational Statistics* 3.2, "180–185".

Wikipedia (2016). *Walrus*. `https://en.wikipedia.org/wiki/Walrus`. Page last modified Aug 20, 2016.