Scripts define HOW

The report defines WHAT & WHY

Mikhail Dozmorov
Fall 2017

Literate programming

> Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, **let us concentrate rather on explaining to humans what we want the computer to do.** Basic idea - use human- and computer-readable chunks.

–Donald E. Knuth, Literate Programming, 1984

RMarkdown/knitR

Writing reports

- **HTML**: HyperText Markup Language, used to create web pages. Developed in 1993
- **LaTeX**: a typesetting system for production of technical/scientific documentation, PDF output. Developed in 1994
- **Sweave**: a tool that allows embedding of the R code in LaTeX documents, PDF output. Developed in 2002
- **Markdown**: a lightweight markup language for plain text formatting syntax. Easily converted to HTML

HTML example

- HTML files have .html extension
- Pairs of tags define content/formatting

```
<h1> Header level 1 </h1>
<a href="http://www.."> Link </a>
<p> Paragraph </p>
```

HTML example

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>
<body>
<h1>Markdown example</h1>
<p>This is a simple example of a Markdown document.</p>
You can emphasize code with <strong>bold</strong> or <em>italics</em>,
or <code>monospace</code> font.
</body>
</html>
```

LaTeX example

- LaTeX files usually have a .tex extension
- LaTeX commands define appearance of text, and other formatting structures

http://www.electronics.oulu.fi/latex/examples/example_1

LaTeX example

~~~
\documentclass{article}
\usepackage{graphicx}
\begin{document}
\title{Introduction to \LaTeX{}}
\author{Author's Name}
\maketitle
\begin{abstract}
This is abstract text: This simple document shows very basic features
\LaTeX{}````.
\end{abstract}
\section{Introduction}
~~~

Sweave example

- Sweave files typically have `.Rnw` extension
- LaTeX syntax for text, `<<chunk_name>>= <code> @` syntax outlines code blocks

```
\documentclass{article}
\usepackage{amsmath}
\usepackage{natbib}
\usepackage{indentfirst}
\DeclareMathOperator{\logit}{logit}
% \VignetteIndexEntry{Logit-Normal GLMM Examples}
\begin{document}
First we attach the dat
<<booth>>=
library(bernor)
data(booth)
attach(booth)
@
```

Markdown syntax

`*italic*` OR `_italic_` *italics*

`**bold**` OR `__bold__` **bold**

Headers

`# Header 1`

`## Header 2`

`### Header 3`

Markdown syntax

Lists

Unordered List

```
* Item 1

* Item 2

  \t \t * Item 2a
```

(To achieve a subpoint, tab twice)

Markdown syntax

Lists

Ordered List

```
1. Item 1

2. Item 2

3. Item 3

  \t \t + Item 3a

  \t \t + Item 3b
```

Markdown syntax

`superscript^2^` x^2

`~~strikethrough~~` ~~powerpoint~~

Horizontal Rule / Page Break

```
******
```

```
------
```

Blockquotes

A friend once said:

```
> It's always better to give

> than to receive.
```

Markdown syntax

Links

```
http://example.com
```

```
[linked phrase](http://example.com)
```

Images

```
![](http://example.com/logo.png)
```

```
![optional caption text](figures/img.png)
```

```
<center><img src="img/clustering.png" height="370px" />
</center>
```

```
knitr::include_graphics("img/knitr.png")
```

Markdown syntax

Tables

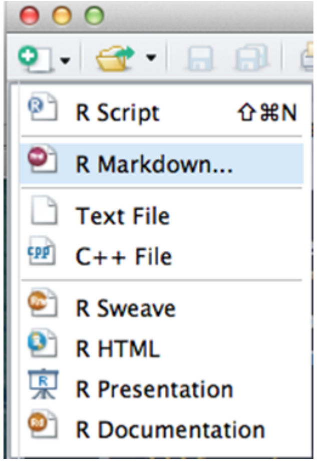
```
First Header  | Second Header
------------- | -------------
Content Cell  | Content Cell
Content Cell  | Content Cell
```

First Header	Second Header
Content Cell	Content Cell
Content Cell	Content Cell

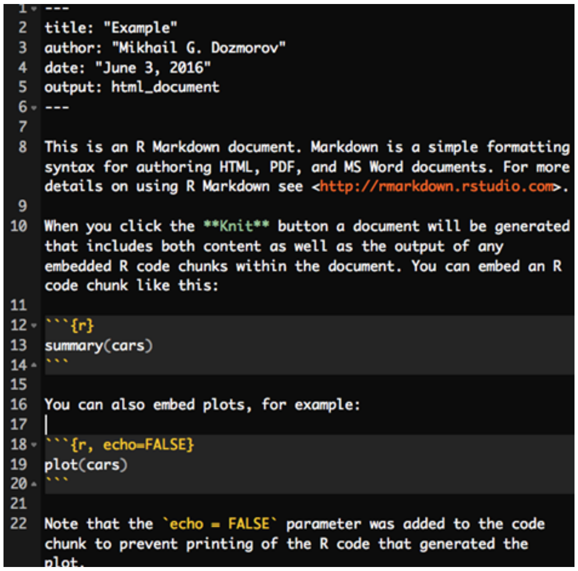
http://www.tablesgenerator.com/markdown_tables

Creating R markdown document

- Regular file with .Rmd extension
- Use RStudio



Creating R markdown document



~~~
---
title: "Example"
author: "Mikhail G. Dozmorov"
date: "June 3, 2016"
output: html_document
---

This is an R Markdown document. Markdown is a simple formatting
syntax for authoring HTML, PDF, and MS Word documents. For more
details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated
that includes both content as well as the output of any
embedded R code chunks within the document. You can embed an R
code chunk like this:

```{r}
summary(cars)
~~~

You can also embed plots, for example:

```{r, echo=FALSE}
plot(cars)
```

Note that the `echo = FALSE` parameter was added to the code
chunk to prevent printing of the R code that generated the
plot.
```

Mix and match HTML tags

`<font color="blue">Change the color of text</font>`

`<center>Center things</center>`

```
<div style="font-size: small;">
Smaller font section
</div>
```
```

# YAML header (think settings)

- YAML: YAML Ain't Markup Language
- YAML is a simple text-based format for specifying data, like JSON

```
---
title: "Untitled"
author: "Your Name"
date: "Current date"
output: html_document
---
```

`output` is the critical part - it defines the output format. Can be `pdf_document` or `word_document`

Comments # allowed - create a very detailed YAML header and comment out what's not needed

http://rmarkdown.rstudio.com/html_document_format.html

R Markdown

Code embedding

- Chunks of code are labeled

1. with single backticks, \`<code>\`, rendered in a monospace font, non-executable. A simple code formatting option
2. with single backticks, \` r <code>\`, for inline code. **r** indicates executable R code. Instead of hard coding numbers, the inline code allows evaluation of variables in real time.

- There are \` r paste(nrow(my_data))\` rows
- The estimated correlation is \` r cor(x, y)\`

https://support.rstudio.com/hc/en-us/articles/205368677-R-Markdown-Dynamic-Documents-for-R

Large code chunks

- Marked with triple backticks

~~~
```{r chunk_name, eval=FALSE}
x = Inf + .Machine$xmin
x
```
~~~

- The chunk name is optional
- By default, the code AND its output are displayed in the final report

Chunk options, comma-separated

- `echo=FALSE` (Default: TRUE): hides the code, but not the results/output.
- `results='hide'` (Default: 'asis') hides the results/output. 'hold' - hold all the output until the end of a chunk.
- `eval=FALSE` (Default: TRUE): disables code execution.
- `cache=TRUE` (Default: FALSE): turn on caching of calculation-intensive chunk.
- `fig.width=##`, `fig.height=##`, `fig.align="center"`: customize the size of a figure generated by the code chunk

Global chunk options

- Some options you would like to set globally, instead of typing them for each chunk

~~~
```{r global_options, eval=FALSE}
knitr::opts_chunk$set(fig.width=12, fig.height=8, fig.path='Figs/',
                      echo=FALSE, warning=FALSE, message=FALSE)
```
~~~

- `warning=FALSE` and `message=FALSE` suppress any R warnings or messages from being included in the final document
- `fig.path='Figs/'` the figure files get placed in the Figs subdirectory. (Default: not saved at all)

https://github.com/mdozmorov/MDmisc

An example of R Markdown document

- Stand-alone code chunk

  `{r libraries, echo=TRUE} library(ggplot2)`

- Inline R code

```
There are ` r paste(length(LETTERS))` letters in English alphabet.
```

An example of R Markdown document, continued

~~~
```{r count_combinations, echo=TRUE}
max_number_of_combinations <- 5
count_combinations <- list()
for (i in 1:max_number_of_combinations) {
  count_combinations <- c(count_combinations, ncol(combn(length(LETTERS), i)))
}
```
~~~

````
A total of ` r paste(count_combinations[[2]])` pairwise combinations of them
can be selected. Or, ` r paste(count_combinations[[3]]) ` combinations of
three letters can be selected.
````

Displaying data as tables

- `knitR` has built-in function to display a table

```
data(mtcars)
```

```
knitr::kable(head(mtcars))
```

- `pander` package allows more customization

```
pander::pander(head(mtcars))
```

- `xtable` package has even more options

```
xtable::xtable(head(mtcars))
```

- `DT` package, an R interface to the DataTables library

```
DT::datatable(mtcars)
```

Creating the final report

- Markdown documents `*.md` can be converted to HTML using

```
markdown::markdownToHTML('markdown_example.md',
'markdown_example.html')
```

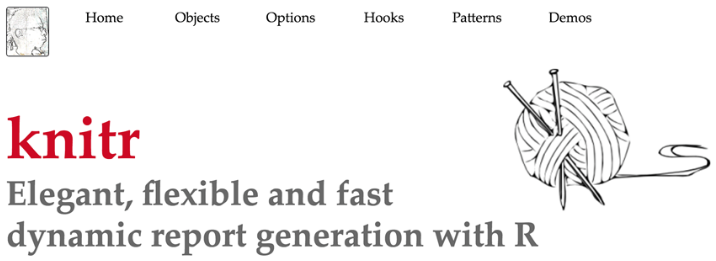
- Another option is to use:

```
rmarkdown::render('markdown_example.md')
```

At the backend it uses `pandoc` command line tool, installed with Rstudio http://pandoc.org/

Creating the final report: KnitR

- KnitR: a package for dynamic report generation written in R Markdown. PDF, HTML, DOCX output. Developed in 2012
- Available at: https://github.com/yihui/knitr
- Available for installation from CRAN, using: `install.packages('knitr', dependencies = TRUE)`

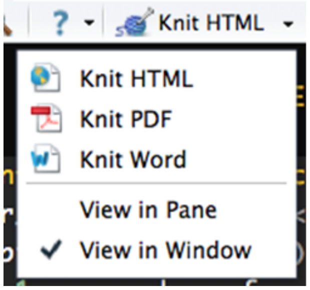


knitr

Elegant, flexible and fast dynamic report generation with R

Creating the final report

- Rstudio: one button
- `knit2html()`, `knit2pdf`



- **Note**: KnitR compiles the document in an R environment separate from yours (think Makefile). Do not use **./Rprofile** file.

R Markdown best practices

- At the beginning, include a code chunk named `libraries`, and load all the packages in this chunk. Generally, it is good to load `dplyr` and `pander` packages by default.
- Include a `settings` code chunk, add any cutoff variables or boolean switches that control the behavior of the main code base.
- e.g. `pval_adj_cutoff <- 0.1 # Cutoff for FDR-adjusted filtering`
- An important settings affecting `data.frame` behavior to include is `stringsAsFactors = FALSE`
- `set.seed(12345)`: initialize random number generator

R Markdown best practices

- At the end of the document, include session information: outputs all packages/versions used

  ```
  {r session_info, results='hide', message=FALSE}
  library("dplyr")
  ```

```
## Warning: package 'dplyr' was built under R version 3.4.1
```

~~~
```{r session_info, results='hide', message=FALSE}
library("pander")
diagnostics <- devtools::session_info()
platform <- data.frame(diagnostics$platform %>% unlist, stringsAsFactors = FALSE)
colnames(platform) <- c("description")
pander(platform)
packages <- as.data.frame(diagnostics$packages)
pander(packages[ packages$`*` == "*", ])
```
~~~

Bibliography

BibTex

```
@article{Berkum:2010aa,
    Abstract = {The three-dimensional folding of chromosomes ...},
    Author = {van Berkum, Nynke L and Lieberman-Aiden, Erez and Williams, Louise and
    Date-Added = {2016-10-08 14:26:23 +0000},
    Date-Modified = {2016-10-08 14:26:23 +0000},
    Doi = {10.3791/1869},
    Journal = {J Vis Exp},
    Journal-Full = {Journal of visualized experiments : JoVE},
    Mesh = {Chromosome Positioning; Chromosomes; DNA; Genomics; Nucleic Acid Conform
    Number = {39},
    Pmc = {PMC3149993},
    Pmid = {20461051},
    Pst = {epublish},
    Title = {Hi-C: a method to study the three-dimensional architecture of genomes},
    Year = {2010},
    Bdsk-Url-1 = {http://dx.doi.org/10.3791/1869}}
```

BibTex managers

- `JabRef` for Windows, http://www.jabref.org/
- `BibDesk` for Mac, http://bibdesk.sourceforge.net/

Save references in `.bib` text file

BibTex and RMarkdown

Add to YAML header

```
bibliography: 3D_refs.bib
```

Insert into RMarkdown as

```
The 3D structure of the human genome has proven to be highly organized
[@Dixon:2012aa; @Rao:2014aa]. This organization starts from distinct
chromosome territories [@Cremer:2010aa], following by topologically associated
domains (TADs) [@Dixon:2012aa; @Jackson:1998aa; @Ma:1998aa; @Nora:2012aa; @Sexton:20
smaller "sub-TADs" [@Phillips-Cremins:2013aa; @Rao:2014aa] and, on the
most local level, individual regions of interacting chromatin [@Rao:2014aa; @Dowen:2
```

Format your BibTex references

Add to YAML header

```
csl: genomebiology.csl
```

Get more styles at https://www.zotero.org/styles

Format your Word output

- If knitting into Word output, you may want to have fonts, headers, margins other than default.
- Create a Word document with the desired formatting. Change font styles by right-clicking on the font (e.g., "Normal") and select "Modify"
- Include it into YAML header

```
output:
  word_document:
    reference_docx: styles.doc/NIH_grant_style.docx
```

https://github.com/mdozmorov/presentations/tree/master/ioslides_template