Reproducible reports with Markdown, knitr

Mikhail Dozmorov

Summer 2018

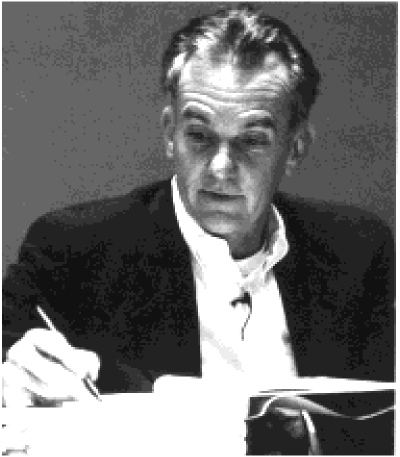
Literate programming

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.**

– Donald E. Knuth, Literate Programming, 1984

Name to know: Edward Tufte

"Design cannot rescue failed content."



Tufte's Rules

1. Show the data
2. "Induce the viewer to think about the substance rather than about methodology, graphic design, the tech of graphic production, or something else."
3. Avoid Distorting the Data
4. Present Many Numbers in a Small Space
5. Make Large Datasets Coherent
6. Encourage Eyes to Compare Data
7. Reveal Data at Several Levels of Detail
8. Serve a Reasonably Clear Purpose
9. Be Closely Integrated with Statistical and Verbal Descriptions of the Dataset

http://www.sealthreinhold.com/school/tuftes-rules/rule_one.php

Document formatting

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 `.htm` or `.html` extensions
- Pairs of tags define content/formatting
  - `<h1> Header level 1 </h1>`
  - `<a href="http://www....."> Link </a>`
  - `<p> Paragraph </p>`

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ut
</head>

<body>
<h1>Markdown example</h1>

<p>This is a simple example of a Markdown document.</p>
```

LaTeX example

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

```
\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 f
```

Sweave example

- Sweave files typically have an `.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}
```

Markdown

- Markdown is a markup language, like HTML and LaTeX, but designed to be as lightweight as possible
- The goal is still to separate form and content, but also to prioritize human-readability, even at the cost of fancy features
- You can learn Markdown in about 5 minutes. If you can write an email, you can write Markdown
- Or, use a desktop Markdown editor like `MarkdownPad` (Windows) or `MacDown` (Mac)

http://bioconnector.github.io/markdown

http://markdownpad.com/

http://macdown.uranusjr.com/

Basic Markdown Syntax

Regardless of your chosen output format, some basic syntax will be useful:

- Section headers
- Text emphasis
- Lists
- R code

Section Headers

To set up different sized header text in your document, use # for Header 1, ## for Header 2, and ### for Header 3.

- In a presentation, this creates a new slide.

Text emphasis

- *Italicize* text via `*Italicize*` or `_Italicize_`.
- **Bold** text via `**Bold**` or `__Bold__`.

Unordered Lists

This code

```
* Item 1
* Item 2
    + Item 2a
    + Item 2b
```

Renders these bullets (sub-lists need 1 tab or 4 spaces!)

- Item 1
- Item 2
    - Item 2a
    - Item 2b

Ordered Lists

This code

```
1. Item 1
2. Item 2
    + Item 2a
    + Item 2b
```

Renders this list (be advised - the bullets may not look great in all templates)

1. Item 1
2. Item 2

- Item 2a
- Item 2b

Inline R Code

- To use R within a line, use the syntax, wrapped in single forward ticks `r dim(mtx)`
- This can be useful to refer to estimates, confidence intervals, p-values, etc. in the body of an article/homework without worrying about copy errors.

Markdown syntax

```
superscript^2^
~~strikethrough~~

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

Images
![](http://example.com/logo.png)
![optional caption text](figures/img.png)

Blockquotes
A friend once said:
> It's always better to give
> than to receive.
```

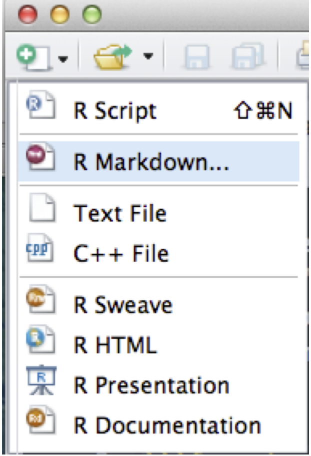
Large code chunks

Marked with triple backticks

~~~
```{r optionalChunkName, echo=TRUE, results='hide'}
# R code here
```
~~~

Creating R markdown document

- Regular text file with `.Rmd` extension
- Create manually, or use RStudio



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`

https://github.com/mdozmorov/MDmisc

YAML header for a PDF presentation

```
---
title: "Reproducible reports with Markdown, knitr"
author: "Mikhail Dozmorov"
date: "Summer 2018"
output:
  beamer_presentation:
    # colortheme: seahorse
    colortheme: dolphin
    fig_caption: no
    fig_height: 6
    fig_width: 7
    fonttheme: structurebold
    # theme: boxes
    theme: AnnArbor
---
```

YAML header for a Word document

```
---
bibliography: [3D_refs.bib,brain.bib]
csl: styles.ref/genomebiology.csl
output:
  word_document:
    reference_docx: styles.doc/NIH_grant_style.docx
  pdf_document: default
  html_document: default
---
```

Modifying the behavior of R code chunks

Chunk options, comma-separated

- `echo=FALSE` - hides the code, but not the results/output. Default: TRUE
- `results='hide'` - hides the results/output. `markup` (the default) takes the result of the R evaluation and turns it into markdown that is rendered as usual, `hold` – `hold` will hold all the output pieces and push them to the end of a chunk. Useful if you're running commands that result in lots of little pieces of output in the same chunk, `hide` will hide results, `asis` writes the raw results from R directly into the document. Only really useful for tables
- `eval=FALSE` - disables code execution. Default: TRUE
- `cache=TRUE` - turn on caching of calculation-intensive chunk. Default: FALSE
- `fig.width=##`, `fig.height=##` - customize the size of a figure generated by the code chunk
- `include`: (TRUE by default) if this is set to `FALSE` the R code is still

Global chunk options

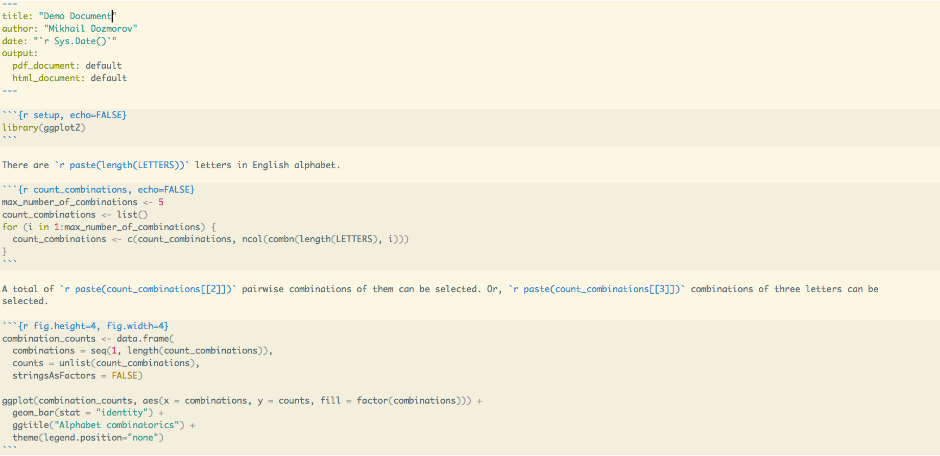
- Some options you would like to set globally, instead of typing them for each chunk

```
knitr::opts_chunk$set(fig.width=12, fig.height=8, fig.path='im
```

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

A special note about **caching**: The cache= option is automatically set to FALSE. That is, every time you render the Rmd, all the R code is run again from scratch. If you use cache=TRUE, for this chunk, knitr will save the results of the evaluation into a directory that you specify, e.g., cache.path='cache/'. When you re-render the document, knitr will first check if there are previously cached results under the cache directory before really evaluating the chunk; if cached results exist and this code

An example of R Markdown document

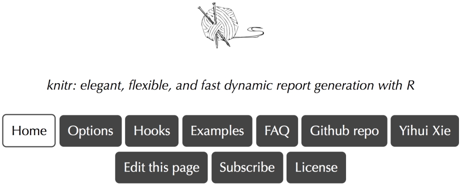


~~~
---
title: "Demo Document"
author: "Mikhail Dozmorov"
date: "`r Sys.Date()`"
output:
  pdf_document: default
  html_document: default
---

```{r setup, echo=FALSE}
library(ggplot2)
```

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

```{r count_combinations, echo=FALSE}
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.

```{r fig.height=4, fig.width=4}
combination_counts <- data.frame(
  combinations = seq(1, length(count_combinations)),
  counts = unlist(count_combinations),
  stringsAsFactors = FALSE)

ggplot(combination_counts, aes(x = combinations, y = counts, fill = factor(combinations))) +
  geom_bar(stat = "identity") +
  ggtitle("Alphabet combinatorics") +
  theme(legend.position="none")
```
~~~

KnitR

- KnitR – Elegant, flexible, and fast dynamic report generation written in R Markdown. PDF, HTML, DOCX output. Developed in 2012

```
install.packages('knitr', dependencies = TRUE)
```



knitr: elegant, flexible, and fast dynamic report generation with R

https://github.com/yihui/knitr, http://yihui.name/knitr/

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)
```

Including figures

- Plots may be generated by R code and displayed in the output document
- Existing image files like `*.jpg`, `*.png`, may be inserted like:

```
![](http://example.com/logo.png)
![optional caption text](figures/img.png)
```

- Alternatively, use `knitr` capabilities:
  ```
  {r, out.width = '300px', echo=FALSE}
  knitr::include_graphics('img/bandThree2.png')
  ```
- For PDF output, use LaTeX syntax:

```
\begin{center}
\includegraphics[height=170px]{img/bioinfo3.png}
\end{center}
```

Customizing Figures: Captions

The `fig.cap` option allows you to specify the caption for the figure generated by a given chunk:

~~~
```{r caption, fig.cap="I am the caption"}
plot(pressure)
```
~~~

Customizing Figures: Size

The fig.height and fig.width options let you specify the dimensions of your plots:

~~~
```{r caption, fig.height = 4, fig.width = 8}
plot(pressure)
```
~~~

Creating the final report

- Markdown documents (*.md or *.Rmd) 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.
- Rstudio – one button. knit2html(), knit2pdf() functions

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

http://pandoc.org/

Things to include in your final report

set.seed(12345) – initialize random number generator

Include session_info() at the end – outputs all packages/versions used

~~~
```{r sessionInfo}
diagnostics <- devtools::session_info()
platform <- data.frame(diagnostics$platform %>% unlist, strin
colnames(platform) <- c('description')
pander(platform)
packages <- as.data.frame(diagnostics$packages)
pander(packages[ packages$`*` == '*', ])
```
~~~

Making default RMarkdown document on your own

Altering the default Rmarkdown file each time you write a homework, report, or article would be a pain.

- Fortunately, you don't have to!

Templates

You can create your own templates which set-up packages, fonts, default chunk options, etc.

- http://rmarkdown.rstudio.com/developer_document_templates.html
- Some packages (e.g `rticles`) provide templates that meet journal requirements or provide other.

Parameters

You may also set parameters in your document's YAML header

```
---
output: html_document
params:
  date: "2017-11-02"
---
```

or pass new values with the `render` function.

- This creates a read-only list `params` containing the values declared.
- e.g. `params$date` returns 2017-11-02.

Bibliography

BibTex

```
@article{Berkum:2010aa,
    Abstract = {The three-dimensional folding of chromosomes
    Author = {van Berkum, Nynke L and Lieberman-Aiden, Erez an
    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
    Number = {39},
    Pmc = {PMC3149993},
    Pmid = {20461051},
    Pst = {epublish},
    Title = {Hi-C: a method to study the three-dimensional arc
    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

Convert anything to BibTex

- doi2bib - BibTex from DOI, arXiv, biorXiv. https://www.doi2bib.org/
- ZoteroBib - create a bibliography from a URL, ISBN, DOI, PMID, arXiv ID, or title. Download as BibTex and more. https://zbib.org/

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 c
[@Dixon:2012aa; @Rao:2014aa]. This organization starts from di
chromosome territories [@Cremer:2010aa], following by topologi
domains (TADs) [@Dixon:2012aa; @Jackson:1998aa; @Ma:1998aa; @M
smaller "sub-TADs" [@Phillips-Cremins:2013aa; @Rao:2014aa] and
most local level, individual regions of interacting chromatin
```

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

Math formulas

Markdown Code: MathJax

- Markdown supports **MathJax JavaScript engine** to render mathematical equations and formulas
- Inline equations - use single “dollar sign” $ to specify MathJax coding

```
$s^{2} = \frac{\sum(x-\bar{x})^2}{n-1}$
```

$$s^2 = \frac{\sum(x-\bar{x})^2}{n-1}$$

Check out this online tutorial http://meta.math.stackexchange.com/questions/5020/mathjax-basic-tutorial-and-quick-reference

https://github.com/ohsu-knight-cancer-biostatistics/reproducible-research/blob/32bba6a78e347d64745982fb6245915cecb1b7c3/slides-info-reproducible-research/study-group-2016/Chpt%2013%20Web%20Presentations/MathJax_2.Rmd

Centering you equations

Insertion of two dollar signs $$ centers your equations. Other examples, off set and centered - notice double dollar signs:

```
$ \sum_{i=0}^n i^2 = \frac{(n^2+n)(2n+1)}{6} $

$$ \sum_{i=0}^n i^2 = \frac{(n^2+n)(2n+1)}{6} $$
```

Inline equation $\sum_{i=0}^n i^2 = \frac{(n^2+n)(2n+1)}{6}$ on the same line. Or, self-standing equation on a separate line

$$\sum_{i=0}^n i^2 = \frac{(n^2+n)(2n+1)}{6}$$

More Interesting Codes:

Greek Letters

```
$\alpha$      $\beta$       $\gamma$      $\chi$
$\Delta$      $\Sigma$      $\Omega$
```

Greek Letters: (not all capitalized Greek letters available)

$\alpha\ \beta\ \gamma\ \chi$

$\Delta\ \Sigma\ \Omega$

superscripts (^) and subscripts (_)

$x_i^2\ log_2x$

Grouping with Brackets

Use brackets $\{\ldots\}$ to delimit a formula containing a superscript or subscript. Notice the difference the grouping makes:

```
${x^y}^z$
$x^{y^z}$
$x_i^2$
$x_{i^2}$
```

${x^y}^z$ x^{y^z} x_i^2 x_{i^2}

Scaling:

Add the scaling code `\left(...\right)` to make automatic size adjustments

```
$(\frac{\sqrt x}{y^3})$
$\left(\frac{\sqrt x}{y^3}\right)$
```

$(\frac{\sqrt x}{y^3})$ $\left(\frac{\sqrt x}{y^3}\right)$

Sums and Integrals

Subscript (_) designates the lower limit; superscript (^) designates upper limit:

```
$\sum_1^n$           $\sum_{i=0}^\infty i^2$
```

$\sum_1^n \sum_{i=0}^\infty i^2$

Other notable symbols:

```
- $\prod$            $\infty$
- $\bigcup$          $\bigcap$
- $\int$             $\iint$
```

$\prod \infty \bigcup \bigcap \int \iint$

Radical Signs

Use 'sqrt' code to adjust the size of its argument. Note the change in size of the square root function based on the code

```
1. $sqrt{x^3}$
2. $sqrt[3]{\frac xy}$
 and for complicated expressions use  brackets
3. ${...}^{1/2}$
```

1. $\sqrt{x^3}$
2. $\sqrt[3]{\frac{x}{y}}$
3. $\ldots^{1/2}$

You can also change fonts!

```
$\mathbb or $Bbb for 'Blackboard bold"
$\mathbf for boldface
$\mathtt for 'typewritter' font
$\mathrm for roman font
$\mathsf for sans-serif
$\mathcal for 'caligraphy'
$\mathscr for script letter:
$\mathfrak for "Fraktur" (old German style)
```

$\mathbb{ABCDEFG}$ $\mathbf{ABCDEFG}$ $\mathtt{ABCDEFG}$ $\mathrm{ABCDEFG}$ $\mathsf{ABCDEFG}$ $\mathcal{ABCDEFG}$

You can also change fonts!

Some special functions such as “lim” “sin” “max” and “ln” are normally set in roman font instead of italic. Use `\lim`, `\sin` to make these (roman):

```
$\sin x$   (roman)  vs  $sin x$  (italics)
```

$\sin x$ (roman) vs $sin x$ (italics)

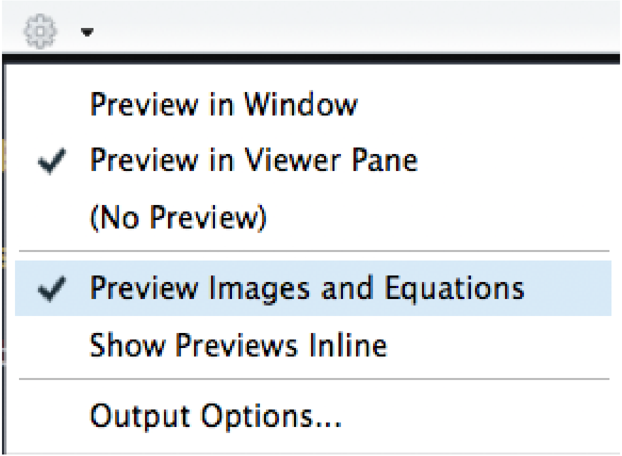
And, add curly brackets

```
$$\begin{cases}
\widehat{IF_{1D}} = IF_{1D} - f(D)/2 \\
\widehat{IF_{2D}} = IF_{2D} + f(D)/2
\end{cases} \ (1)$$
```

$$\begin{cases} \widehat{IF_{1D}} = IF_{1D} - f(D)/2 \\ \widehat{IF_{2D}} = IF_{2D} + f(D)/2 \end{cases} \quad (1)$$

RStudio bonus

Inline preview of forumlas and images in an RMarkdown document



Preview in Window

✓ Preview in Viewer Pane

(No Preview)

✓ Preview Images and Equations

Show Previews Inline

Output Options...

LaTeX and Markdown

- Rendering Markdown as a pdf requires a LaTeX installation.
- You will additionally need to install Pandoc from http://pandoc.org/
- With LaTeX, many customizations are possible.

LaTeX Customization, 1

- You can include additional LaTeX commands and content.
- Use the `includes` option as follows to add your favorite style files for the preamble, title/abstract, bibliography, etc...

```
---
title: 'A More Organized Person's Document'
output:
  beamer_presentation:
    includes:
      in_header: header.tex
      before_body: doc_prefix.tex
      after_body: doc_suffix.tex
---
```

LaTeX Customization, 2

- If you prefer a self-contained document, you may opt for the `header-includes` option over the modular approach:

```
---
title: 'BIOST 691: Reproducible Research Tools'
author: "Author Name"
date: "November 2, 2017"
header-includes:
   - \usepackage{graphicx}
output:
  beamer_presentation:
    theme: "Frankfurt"
---
```

Note: LaTeXin Text

- In Markdown, "`\LaTeX rocks`" renders as "LaTeXrocks" (no space!).
- Use "`\LaTeX\ rocks`" to render "LaTeX rocks", instead.
- This can be especially important when using new commands.