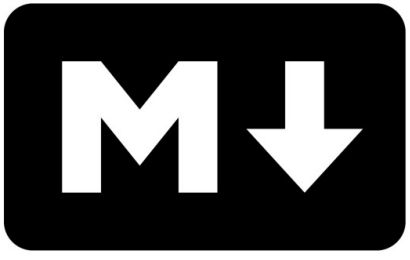
R markdown tutorial

at useR! 2015 – Aalborg, Denmark



Gergely Daroczi

@daroczig

2015-06-30

```
> sessionInfo()
[1] "June 30 - July 3, 2015"
[2] "Aalborg, Denmark"
```

Yes, I've created this with markdown

It looks like LaTeX, or to be more precise, it's a **Beamer** presentation, but this slide was created like:

```
## Yes, I've created this with markdown

It looks like \LaTeX, or to be more precise,
it's a **Beamer** presentation,
but this slide was created like:
```

Yes, I've created this with markdown

It looks like LaTeX, or to be more precise, it's a **Beamer** presentation, but this slide was created like:

```
## Yes, I've created this with markdown

It looks like \LaTeX, or to be more precise,
it's a **Beamer** presentation,
but this slide was created like:
```

Have you use any document markup language (e.g. LaTeX) before?

Yes, I've created this with markdown

It looks like LaTeX, or to be more precise, it's a **Beamer** presentation, but this slide was created like:

```
## Yes, I've created this with markdown

It looks like \LaTeX, or to be more precise,
it's a **Beamer** presentation,
but this slide was created like:
```

Have you use any document markup language (e.g. LaTeX) before?

Is it much more readable than `\textbf{tex}`, right?

No, I've not created this with markdown

```
\begin{frame}[fragile]
\frametitle{What is ``pander''?}
\framesubtitle{A collection of helper functions to print markdown syntax}
\small \begin{Verbatim}[commandchars=\\\{\}]
> ?pandoc.(footnote|header|horizontal.rule|image|link|p)(.return)?
> ?pandoc.(emphasis|strikeout|strong|verbatim)(.return)?

> \textcolor{orange}{pandoc.strong}(\textcolor{cyan}{'foobar'})
\textcolor{darkgray}{**foobar**}

> \textcolor{orange}{pandoc.strong.return}(\textcolor{cyan}{'foobar'})
\textcolor{darkgray}{[1] "**foobar**"}

> \textcolor{orange}{pandoc.header}(\textcolor{cyan}{'foobar'}, \textcolor{darkgreen}{level} = \textcolor{red}{2})

\textcolor{darkgray}{## foobar}

> \textcolor{orange}{pandoc.header}(\textcolor{cyan}{'foobar'}, \textcolor{darkgreen}{style} = \textcolor{cyan}{'setext'})

\textcolor{darkgray}{foobar}
\textcolor{darkgray}{======}
\end{Verbatim}
\end{frame}
```

No, I've not created this with markdown

```
\begin{frame}[fragile]
\frametitle{What is ``pander''?}
\framesubtitle{A collection of helper functions to print markdown syntax}
\small \begin{Verbatim}[commandchars=\\\{\}]
> ?pandoc.(footnote|header|horizontal.rule|image|link|p)(.return)?
> ?pandoc.(emphasis|strikeout|strong|verbatim)(.return)?

> \textcolor{orange}{pandoc.strong}(\textcolor{cyan}{'foobar'})
\textcolor{darkgray}{**foobar**}

> \textcolor{orange}{pandoc.strong.return}(\textcolor{cyan}{'foobar'})
\textcolor{darkgray}{[1] "**foobar**"}

> \textcolor{orange}{pandoc.header}(\textcolor{cyan}{'foobar'}, \textcolor{darkgreen}{level} = \textcolor{red}{2})

\textcolor{darkgray}{## foobar}

> \textcolor{orange}{pandoc.header}(\textcolor{cyan}{'foobar'}, \textcolor{darkgreen}{style} = \textcolor{cyan}{'setext'})

\textcolor{darkgray}{foobar}
\textcolor{darkgray}{======}
\end{Verbatim}
\end{frame}
```

Although I miss the `\framesubtitle` command.

Yes, I've created this with markdown

It looks like LaTeX, or to be more precise, it's a **Beamer** presentation, but this slide was created like:

```
## Yes, I've created this with markdown

It looks like \LaTeX, or to be more precise,
it's a **Beamer** presentation,
but this slide was created like:
```

Have you use any document markup language (e.g. LaTeX) before?

Is it much more readable than `\textbf{tex}`, right?

Have you used markdown before?

Yes, I've created this with markdown

It looks like LaTeX, or to be more precise, it's a **Beamer** presentation, but this slide was created like:

```
## Yes, I've created this with markdown

It looks like \LaTeX, or to be more precise,
it's a **Beamer** presentation,
but this slide was created like:
```

Have you use any document markup language (e.g. LaTeX) before?

Is it much more readable than `\textbf{tex}`, right?

Have you used markdown before?

And what about GitHub, StackOverflow, reddit?

Requirements

Install packages:

- knitr
- rmarkdown
- pander
- rapport
- shiny

Or use Docker:

https://registry.hub.docker.com/u/cardcorp/r-pandoc

```
> sapply(c(
+     'knitr',
+     'rmarkdown',
+     'pander',
+     'rapport',
+     'shiny'
+ ), require, character.only = TRUE)
```

Optional IDEs: RStudio, Emacs/ESS etc.

Markdown basics

markdown	LaTeX	HTML
`# Header 1`	`\section{Header 1}`	`<h1>Header 1</h1>`
`## Header 2`	`\subsection{Header 2}`	`<h2>Header 2</h2>`
`*italics*`	`\textit{italics}`	`<i>italics</i>`
`**bold**`	`\textbf{bold}`	`<b>italics</b>`

Markdown basics

markdown	LaTeX	HTML
`# Header 1`	`\section{Header 1}`	`<h1>Header 1</h1>`
`## Header 2`	`\subsection{Header 2}`	`<h2>Header 2</h2>`
`*italics*`	`\textit{italics}`	`<i>italics</i>`
`**bold**`	`\textbf{bold}`	`<b>italics</b>`

```
pander(rbindlist(list(
  list(markdown = '# Header 1',
    LaTeX = '\\section{Header 1}',    HTML = '<h1>Header 1</h1>'),
  list(markdown = '## Header 2',
    LaTeX = '\\subsection{Header 2}', HMTL = '<h2>Header 2</h2>'),
  list(markdown = '*italics*',
    LaTeX = '\\textit{italics}',      HTML = '<i>italics</i>'),
  list(markdown = '**bold**',
    LaTeX = '\\textbf{bold}',         HTML = '<b>italics</b>')
  )))
```

How to run the demos?

```
library(shiny)
## http://bit.ly/shiny-preview-markdown
runGist('e56e0e8ad3e8a3eed31e') # (ShinyApps on next slide)
```

Try markdown

```
## Hello markdown!

This is a demo created by

* *shiny*
* `pandoc` and
* **pander**.

+-----+-----+-----+
|  a  |  b  |  c  |
+=====+=====+=====+
|  1  |  4  |  7  |
+-----+-----+-----+
|  2  |  5  |  8  |
+-----+-----+-----+
|  3  |  6  |  9  |
+-----+-----+-----+
```

Hello markdown!

This is a demo created by

- *shiny*
- `pandoc` and
- **pander**.

a	b	c
1	4	7
2	5	8
3	6	9

How to run the demos?

https://daroczig.shinyapps.io/preview-markdown (*)

shinyapps.io Powered by RStudio

```
## Hello markdown!

This is a demo created by

* *shiny*
* `pandoc` and
* **pander**.

+-----+-----+-----+
|  a  |  b  |  c  |
+=====+=====+=====+
|  1  |  4  |  7  |
+-----+-----+-----+
|  2  |  5  |  8  |
+-----+-----+-----+
|  3  |  6  |  9  |
+-----+-----+-----+
```

Hello markdown!

This is a demo created by

- *shiny*
- `pandoc` and
- **pander**.

a	b	c
1	4	7
2	5	8
3	6	9

Some common markdown examples #1 (pardon the pun)

```
It does not
really
matters when you break
the line. Except for two line breaks in a

row.

This is a paragraph.
```

It does not really matters when you break the line. Except for two line breaks in a

row.

This is a paragraph.

Some common markdown examples #2 (emphasize, lists)

```
You can emphasize and strongly emphasize any text
with the stars:

* use one star for *italics*,
* double stars for **bold** font and
* three stars to ***mix those*** styles.
```

You can emphasize and strongly emphasize any text with the stars:

- use one star for *italics*,
- double stars for **bold** font and
- three stars to ***mix those*** styles.

Some common markdown examples #2 (emphasize, lists)

```
You can emphasize and strongly emphasize any text
with the stars:

* use one star for *italics*,
* double stars for **bold** font and
* three stars to ***mix those*** styles.
```

You can emphasize and strongly emphasize any text with the stars:

- use one star for *italics*,
- double stars for **bold** font and
- three stars to ***mix those*** styles.

Or by underscores, like `_italics_` and `__bold__`

Use `tilde` to kill stuff, like `~~strikethrough~~`: ~~strikethrough~~

Some common markdown examples #3 (ordered lists)

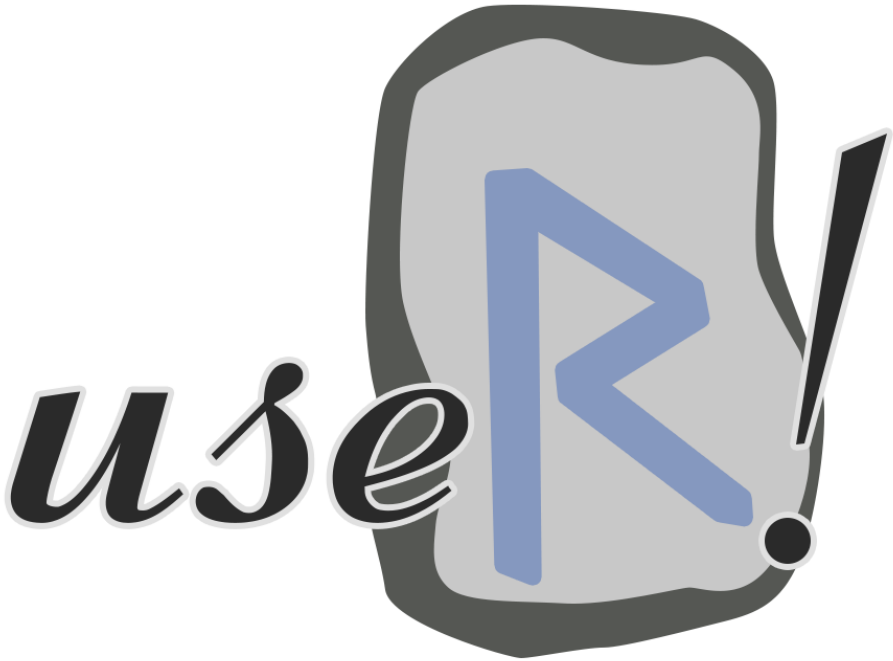
```
Using numbers instead of the stars results in an ordered list:

1. where it does not really matters
1. if you increment the numbers or
3. not, but you can create nested lists by indenting
    1. your list elements
    2. by 4 spaces
        * or by more for further levels
        * for further levels
```

1. where it does not really matters
2. if you increment the numbers or
3. not, but you can create nested lists by indenting
    1. your list elements
    2. by 4 spaces
        - or by more
        - for further levels

Some common markdown examples #4 (link, image)

```
![alt text, such as lorem ipsum](images/useR2015.png)
```




```
> sessionInfo()
[1] "June 30 - July 3, 2015"
[2] "Aalborg, Denmark"
```

Some common markdown examples #5 (tables)

```
   Grid/Org-mode      Multiline  Pipe/PHP Extra/rmarkdown   Simple

+-----+-----+-----+   ----------   | a | b | c |    a   b   c
| a | b | c |    a   b   c    |:---:|:---:|:---:|   --- --- ---
+=====+=====+=====+   --- --- ---   | 1 | 4 | 7 |    1   4   7
| 1 | 4 | 7 |    1   4   7    | 2 | 5 | 8 |    2   5   8
+-----+-----+-----+                | 3 | 6 | 9 |    3   6   9
| 2 | 5 | 8 |    2   5   8
+-----+-----+-----+
| 3 | 6 | 9 |    3   6   9
+-----+-----+-----+
```

a	b	c
1	4	7
2	5	8
3	6	9

Some common markdown examples #5 (tables)

```
----------------------
left    center    right
------ -------- -------
1         4       **7**

2         5          8

3         6          9
----------------------

Table: How to align cells?
```

Table 3:How to align cells?

left	center	right
1	4	**7**
2	5	8
3	6	9

Some common markdown examples #6 (comments)

No official comments, but this works:

```
<!---
HTML like comment
for multi-line stuff
which will be auto-removed by pandoc
-->
```

And this one as well:

```
[](URL hack comment for one-liners)
```

Some common markdown examples #7 (HTML, PDF)

- No general support for page-breaks, but LaTeXcommands work:

```
\pagebreak
```

- Similarly, you can fine-tune the resulting PDF document via:

```
\usepackage[margin=1cm]{geometry}
\includegraphics[width=..., keepaspectratio]{...}
\usepackage{fancyhdr}
```

- Use any HTML tag with HTML output:

```
<iframe src="..."></iframe>
<div style="..."></div>
<script>...</script>
```

Some common markdown examples #8 (code)

```
Use *backticks* for inline `code`,
```

Use *backticks* for inline `code`, and enable automatic syntax highlighting via fenced code blocks by referring the language:

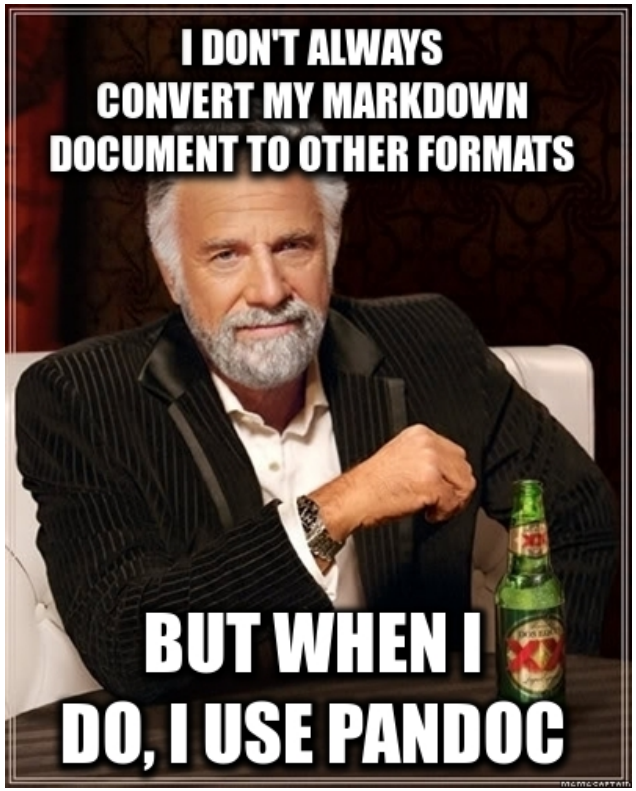
````
```r
require(graphics)
pairs(mtcars, main = "mtcars data")
coplot(mpg ~ disp | as.factor(cyl), data = mtcars)
```
````

```r
require(graphics)
pairs(mtcars, main = "mtcars data")
coplot(mpg ~ disp | as.factor(cyl), data = mtcars)
```

You can also use dash/tilde instead of backticks or indented code.

The great markdown test

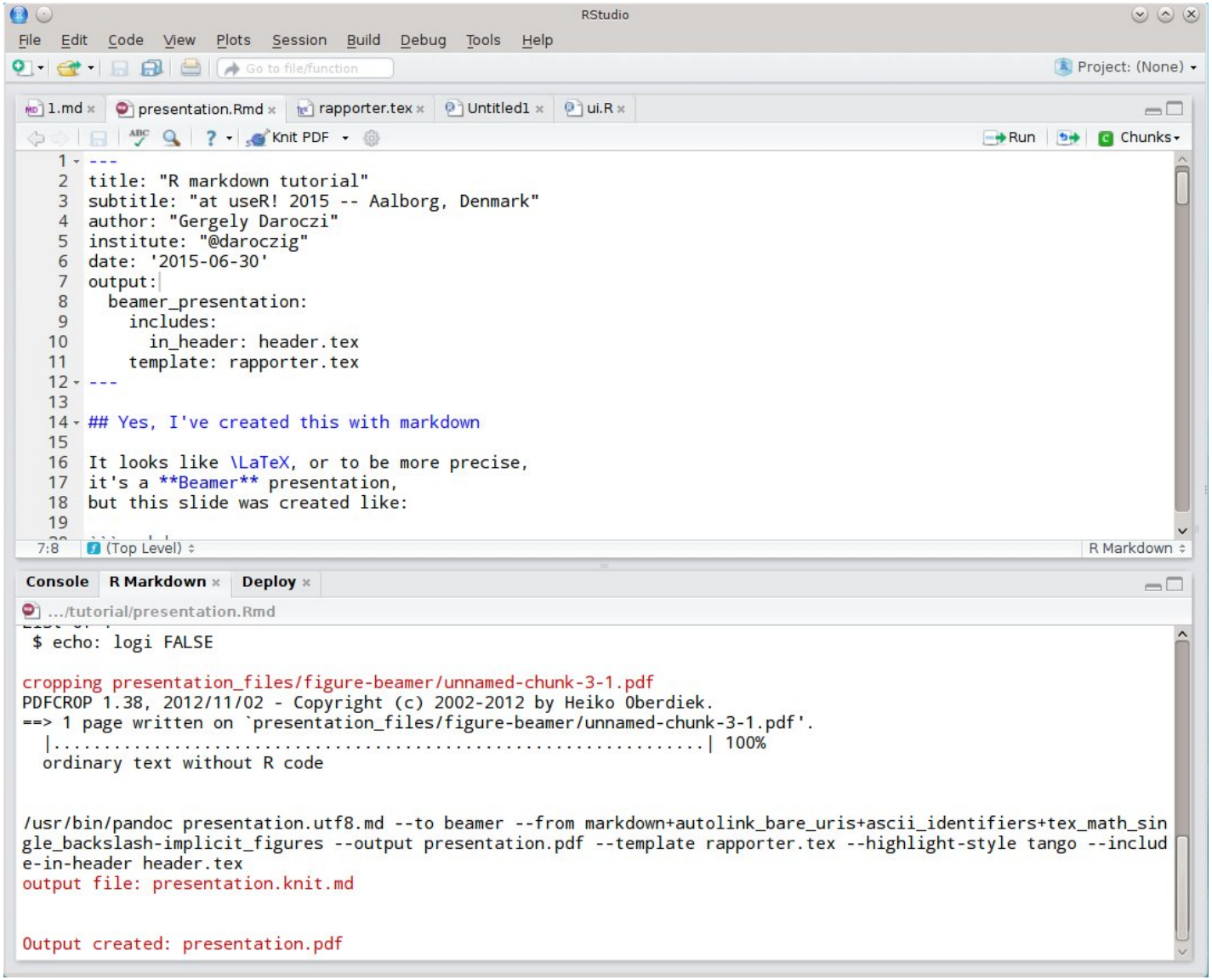
Writing **markdown** is easy. To convert it to *HTML*, use `pandoc` after reading the docs:



Replicate this by:

```
devtools::install_github('leeper/meme')
```

RStudio's R Markdown: knitr + pandoc




```
rmarkdown::render('foobar.Rmd')
```

R Markdown (knitr) code blocks

````
This is **not** an ordinary markdown document,
but can eval `r paste(letters[c(3, 15, 4:5)], collapse = '')`
inline and as standalone chunks as well:

```{r}
summary(mtcars$hp)
```
````

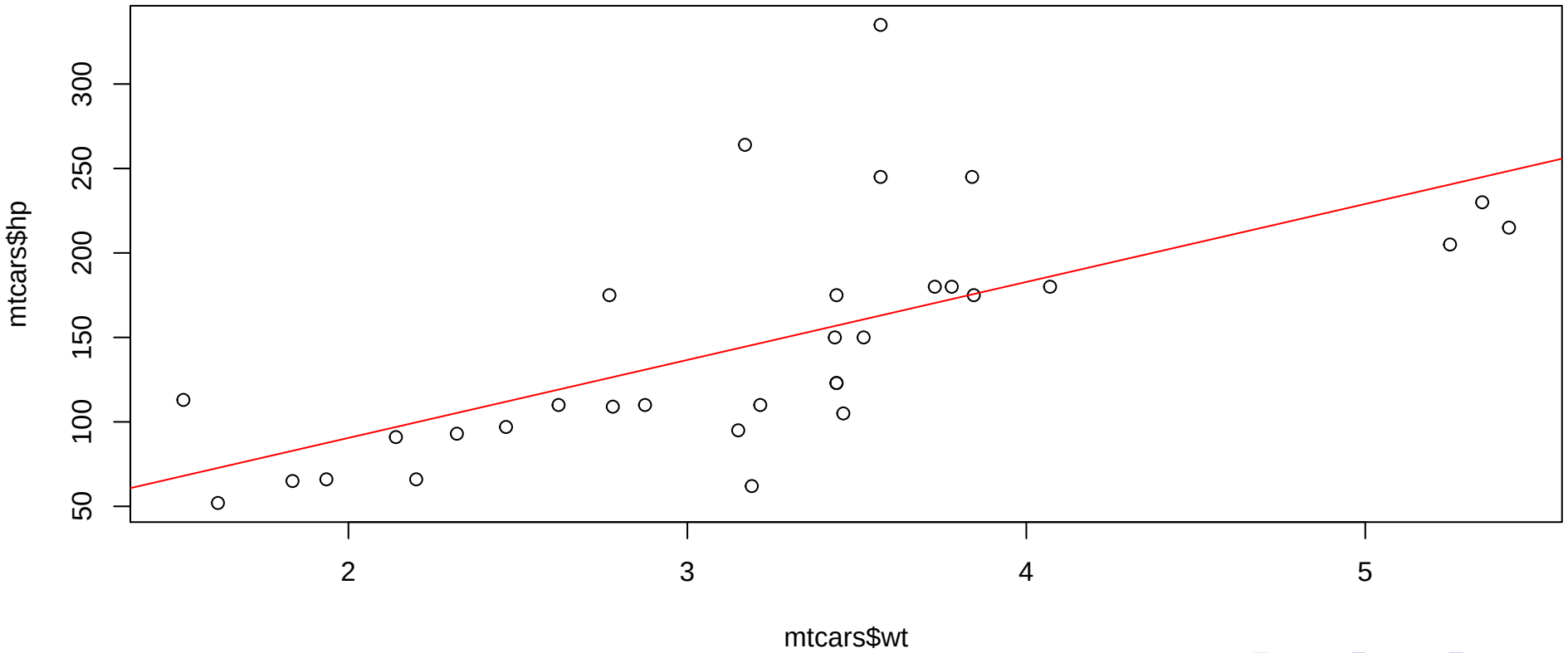
This is **not** an ordinary markdown document, but can eval code inline and as standalone chunks as well:

```
summary(mtcars$hp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    52.0    96.5   123.0   146.7   180.0   335.0
```

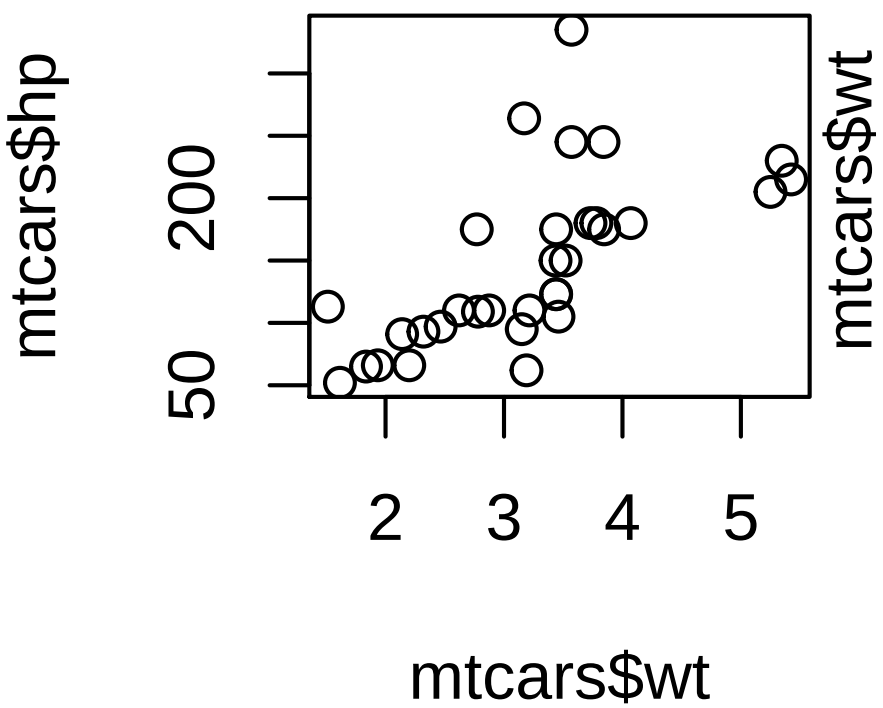
R Markdown (knitr) code blocks with plots

```
```{r, echo = FALSE, fig.height = 5}
plot(mtcars$wt, mtcars$hp)
abline(lm(hp ~ wt, mtcars), col = 'red')
```
```



R Markdown (knitr) code blocks with multiple plots

````
```{r, echo = FALSE, fig.height = 3, fig.width = 3, dpi = 300}
plot(mtcars$wt, mtcars$hp)
plot(mtcars$hp, mtcars$wt)
```
````



R Markdown (knitr) code blocks with tabular objects

````
```{r, echo = FALSE}
head(iris)
```
````

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

R Markdown (knitr) code blocks with tabular objects

````
```{r, echo = FALSE}
xtable::xtable(head(iris))
```
````

```
% latex table generated in R 3.2.0 by xtable 1.7-4 package
% Tue Jun 30 10:42:16 2015
\begin{table}[ht]
\centering
\begin{tabular}{rrrrrl}
  \hline
 & Sepal.Length & Sepal.Width & Petal.Length & Petal.Width & Species \\
  \hline
1 & 5.10 & 3.50 & 1.40 & 0.20 & setosa \\
  2 & 4.90 & 3.00 & 1.40 & 0.20 & setosa \\
  3 & 4.70 & 3.20 & 1.30 & 0.20 & setosa \\
  4 & 4.60 & 3.10 & 1.50 & 0.20 & setosa \\
  5 & 5.00 & 3.60 & 1.40 & 0.20 & setosa \\
  6 & 5.40 & 3.90 & 1.70 & 0.40 & setosa \\
   \hline
\end{tabular}
```

R Markdown (knitr) code blocks with tabular objects

````
```{r, echo = FALSE}
knitr::kable(head(iris))
```
````

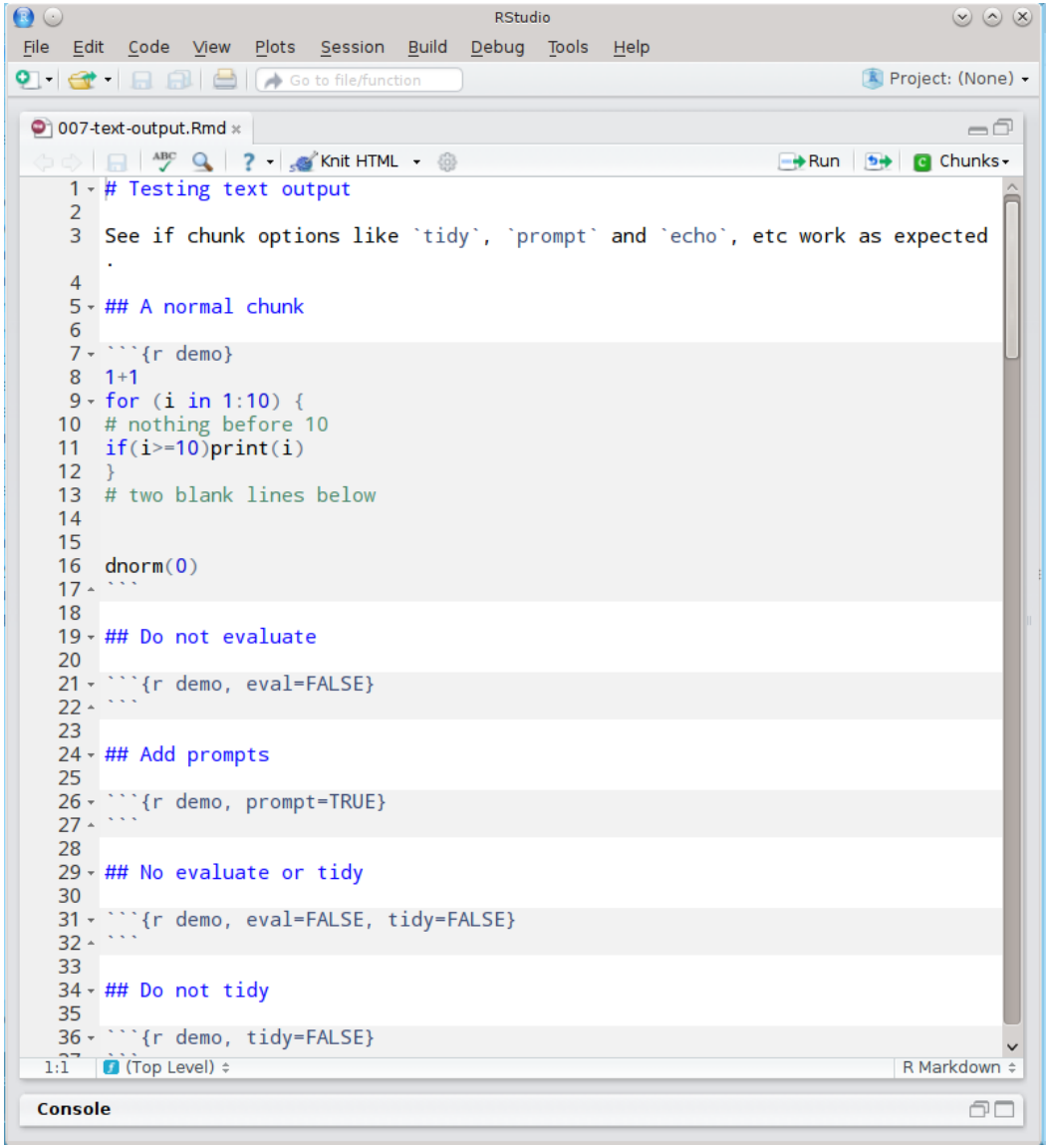
Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

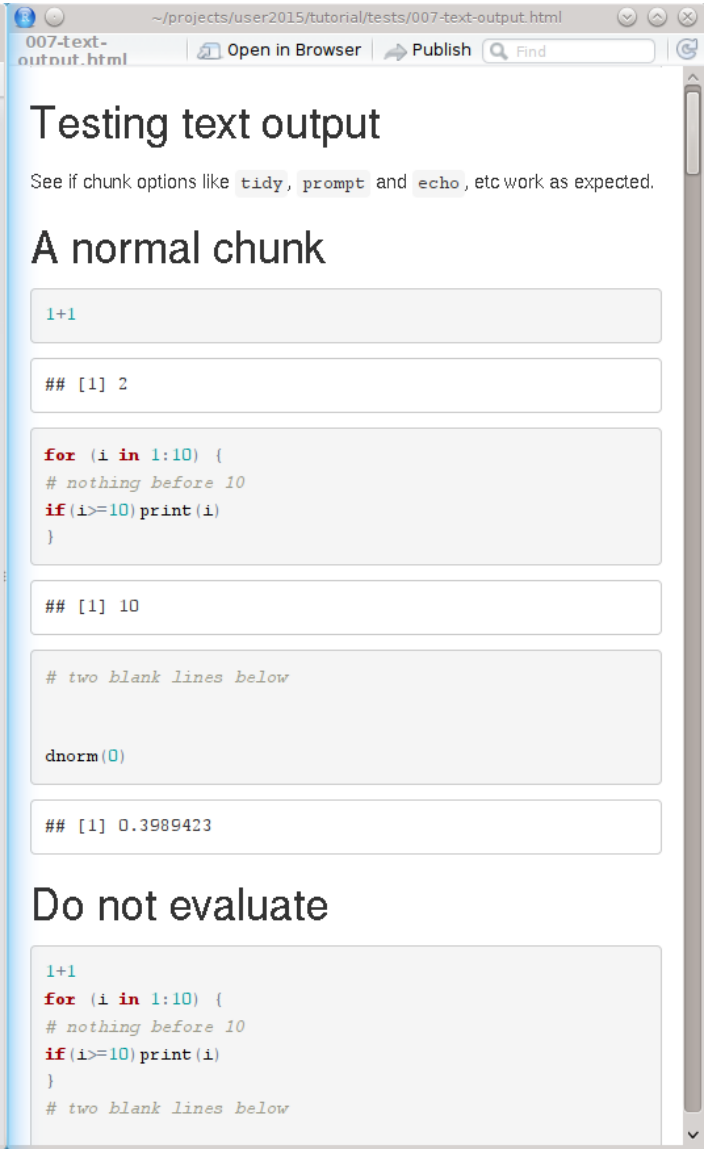
R Markdown (knitr) code blocks with tabular objects

````
```{r, echo = FALSE, ~~results = 'asis'~~}
knitr::kable(head(iris))
```
````

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

R Markdown (knitr) chunk options





https://github.com/yihui/knitr-examples – 007-text-output.Rmd

R Markdown (knitr) global options

```
```{r global_options, include = FALSE}
library(knitr)
opts_chunk$set(
   fig.width  = 8,
   fig.height = 8,
   dpi        = 300,
   fig.path   = 'plots/foobar',
   echo       = FALSE,
   warning    = FALSE,
   message    = FALSE)
```
```

http://yihui.name/knitr/options/

R Markdown (knitr) engines

````
```{r engine='bash'}
lsb_release -a
```
````

```
lsb_release -a
```

```
## LSB Version: 1.4
## Distributor ID:  Arch
## Description: Arch Linux
## Release: rolling
## Codename:    n/a
```

http://yihui.name/knitr/demo/engines/

The great knitr test

Create a HTML report on `mtcars` including

- descriptive stats on `hp`, `wt` and `gear`
- cross table of `gear` and `am`
- a paragraph on how to compute the standard deviation including
  - a formula
  - custom R function to compute that
  - demo run on `hp`
- scatterplot on `hp` and `wt` with a linear trend line

Resources:

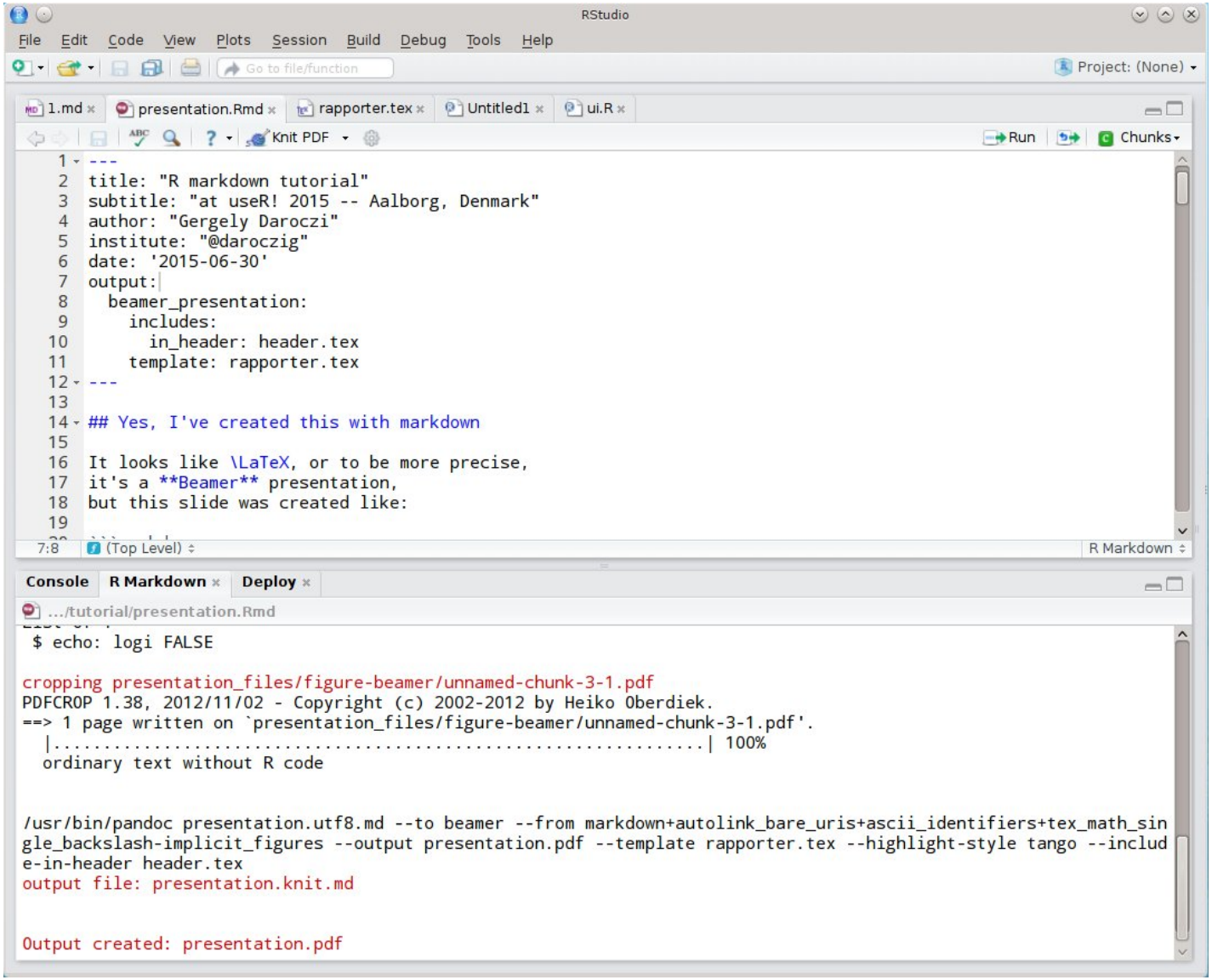
- Quick-R: http://www.statmethods.net/stats
- `knitr` options: http://yihui.name/knitr/options/
- `knitr` examples: https://github.com/yihui/knitr-examples (especially look for `007-text-output.Rmd`)

The great knitr test

Now create a PDF report

with the same content!

RStudio's R Markdown: knitr + pandoc




```
rmarkdown::render('foobar.Rmd')
```

RStudio ships pandoc

- Linux:

```
/usr/lib/rstudio/bin/pandoc/pandoc
```

- Mac:

```
/Applications/RStudio.app/Contents/MacOS/pandoc
```

- Windows:

```
c:\Program Files\RStudio\bin\pandoc
```

What is pandoc in R Markdown?

- Haskell software to convert text documents (swiss army knife)
- supports markdown flavors, HTML, LaTeX, docx, odt etc.
- markdown extensions, like references and bibliography
- great table support (although no col-row spanning)
- metadata in YAML header

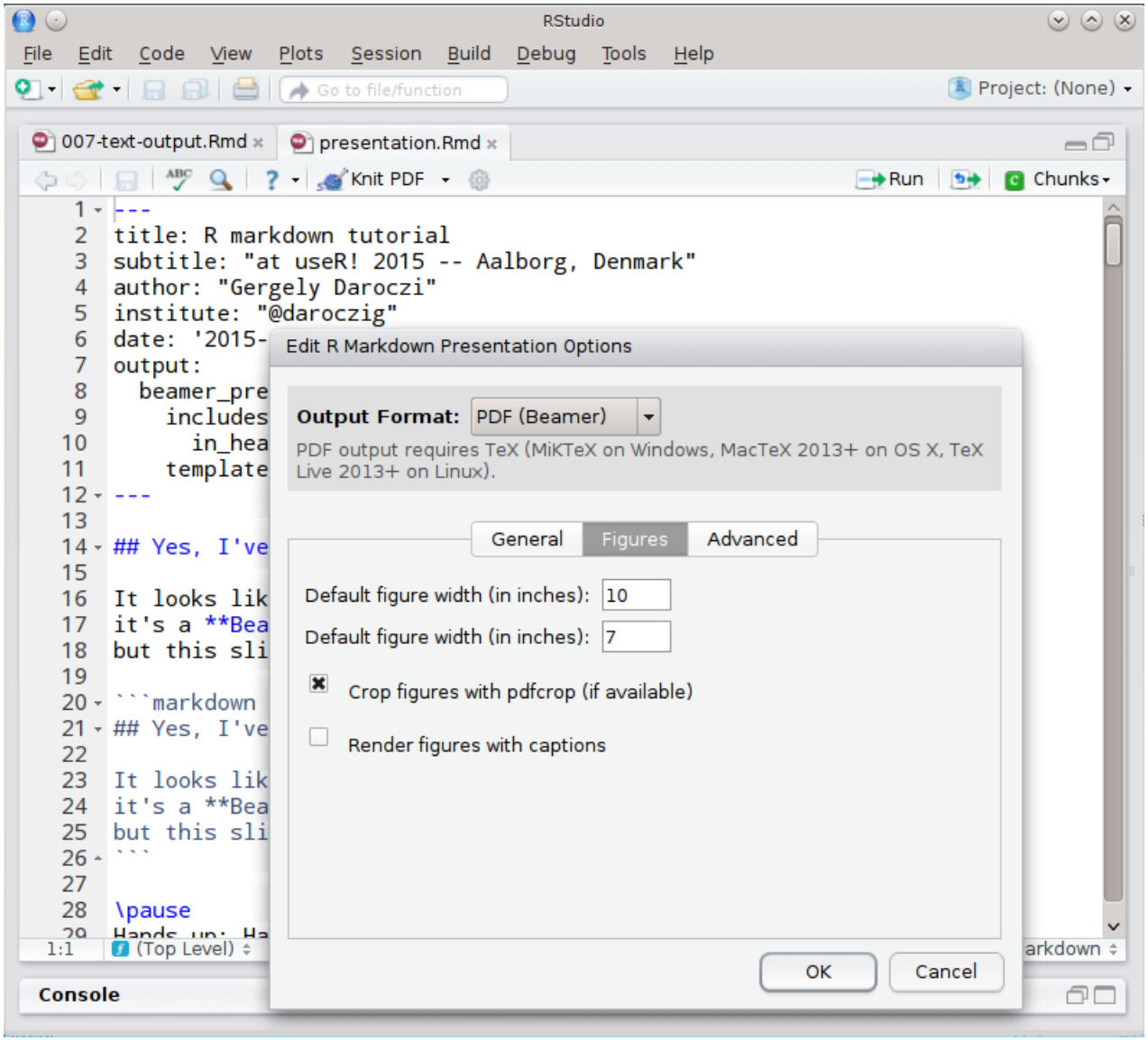
```
---
title:  'This is the title: it contains a colon'
author:
- name: Author One
affiliation: University of Somewhere
- name: Author Two
affiliation: Industry at Somewhere else
tags: [foo, bar]
---
```

- mathematical formulas via LaTeX syntax (in HTML, PDF, docx)
- http://commonmark.org/

R Markdown YAML header

```
---
title: "Title of the document"
author: "Gergely Daroczi"
date: `r Sys.time()`
output:
  beamer_presentation:
    includes:
      in_header: "header.tex"
    template: "custom.tex"
    toc: true
    keep_tex: true
    pandoc_args: [
      "--no-tex-ligatures"
    ]
    highlight: tango
---
```

R Markdown (pandoc) YAML header



R Markdown HTML themes

```
title: foobar
output:
  html_document:
    theme: readable
    highlight: espresso
```

foobar

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:

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
```

R Markdown HTML themes

```
title: foobar
output:
  html_document:
    theme: flatly
    highlight: kate
```

foobar

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:

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

R Markdown custom CSS

```
title: foobar
output:
  html_document:
    theme: flatly
    highlight: kate
    css: custom.css
```

foobar

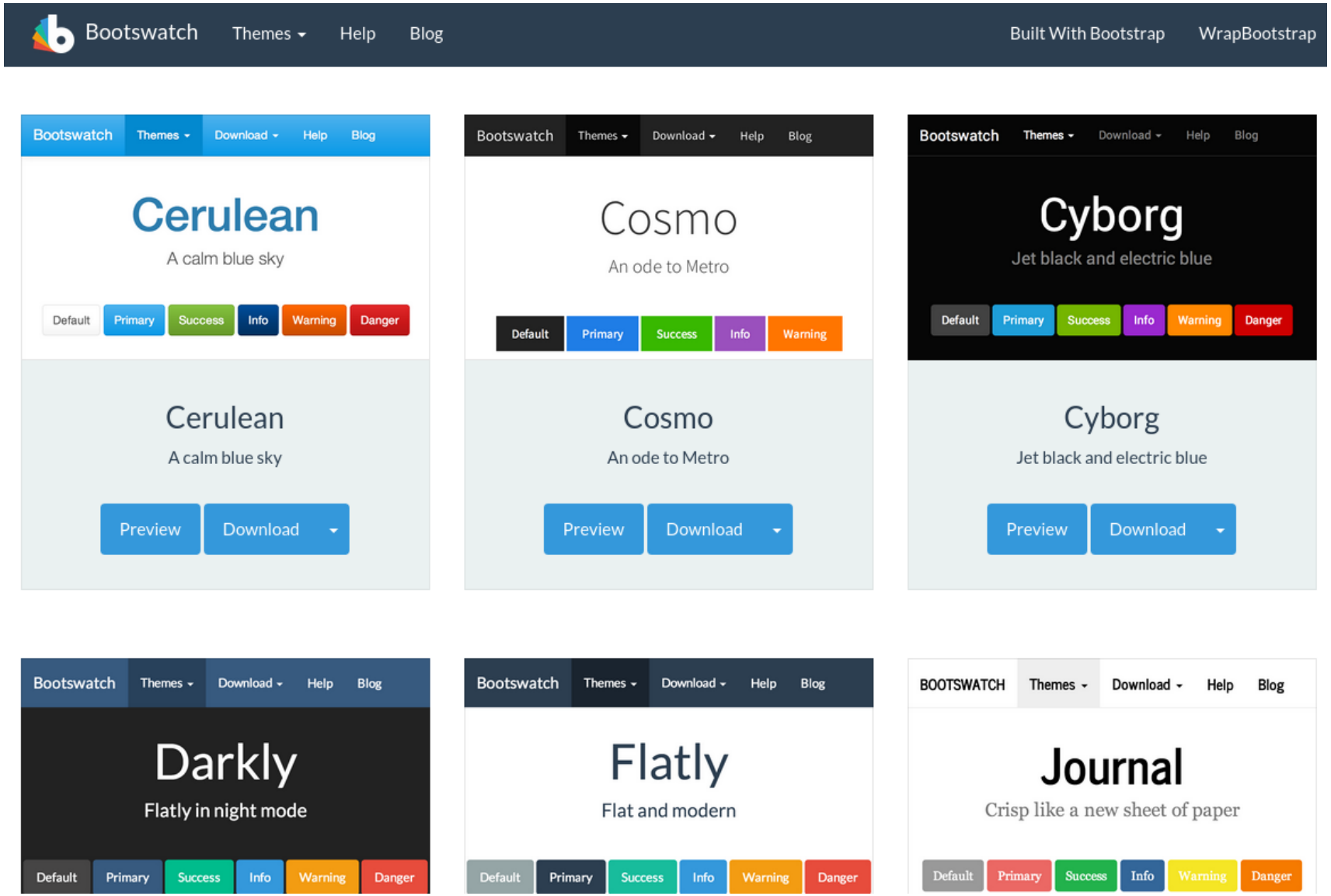
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

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:

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
```

R Markdown more themes, documentation



http://rmarkdown.rstudio.com/

Congrats, You Know R Markdown!

http://rmarkdown.rstudio.com/

But there are some other tools to improve this workflow!

The R Markdown toolbox

- literate programming engine (`knitr` , `brew` , `pander` etc.)
- document converter (**pandoc**, AsciiDoc, Textile, reStructuredText etc.)
- document templates, stylesheets (CSS, JavaScript, LaTeX, docx)
- version control (git, SubVersion etc.) – always do this
- document storage, publisher (GitHub, rpubs, shinyapps.io etc)

The R Markdown toolbox

- literate programming engine (`knitr` , `brew` , `pander` etc.)
- document converter (**pandoc**, AsciiDoc, Textile, reStructuredText etc.)
- document templates, stylesheets (CSS, JavaScript, LaTeX, docx)
- version control (git, SubVersion etc.) – always do this
- document storage, publisher (GitHub, rpubs, shinyapps.io etc)

Tools to pre-process *Rmd* files:

- tool to transform R objects into markdown (`kable` , `pander`)
- templating languages handling loops, conditional expressions and child documents in knitr (`brew` , `pander` , `R.rsp`)
- reusable markdown templates in `rapport`
- GNU Make: text file to describe build workflow
- `remake` package: Make-like declarative workflows in R

The R Markdown toolbox

- literate programming engine (`knitr` , `brew` , `pander` etc.)
- document converter (**pandoc**, AsciiDoc, Textile, reStructuredText etc.)
- document templates, stylesheets (CSS, JavaScript, LaTeX, docx)
- version control (git, SubVersion etc.) – always do this
- document storage, publisher (GitHub, rpubs, shinyapps.io etc)

Tools to pre-process *Rmd* files:

- **tool to transform R objects into markdown (`kable` , `pander`)**
- templating languages handling loops, conditional expressions and child documents in knitr (`brew` , `pander` , `R.rsp`)
- reusable markdown templates in `rapport`
- GNU Make: text file to describe build workflow
- `remake` package: Make-like declarative workflows in R

pander: An R pandoc writer for header and hyperlinks

```
> ?pandoc.(footnote|header|image|link|p)(.return)?

> pandoc.horizontal.rule()
---

> pandoc.horizontal.rule.return()
[1] "\n---\n"

> pandoc.header('foobar', level = 2)          # ATX style
## foobar

> pandoc.header('foobar', style = 'setext') # underlined
foobar
======

> pandoc.link('example.com', 'Most popular URL')
[Most popular URL](example.com)

> pandoc.image('http://image.url', 'image caption')
![image caption](http://image.url)
```

pander: An R pandoc writer for formatted text

```
> ?pandoc.(emphasis|strikeout|strong|verbatim)(.return)?

> pandoc.strong('foobar')
**foobar**

> pandoc.strong.return('foobar')
[1] "**foobar**"

> pandoc.emphasis('foobar')
*foobar*

> pandoc.strikeout('foobar')
~~foobar~~

> pandoc.verbatim('foobar')
`foobar`
```

pander: An R pandoc writer for lists

```
> pandoc.list(c('foo', 'bar'))
```

```
* foo
* bar

<!-- end of list -->
```

```
> l <- list('First list element', paste0(1:2, '. subelement'),
+     'Second element', list('F', 'B', 'I', c('phone', 'pad', 'talics')))
> pandoc.list(l, 'roman')
```

```
I. First list element
    I. 1. subelement
    II. 2. subelement
II. Second element
    I. F
    II. B
    III. I
        I. phone
        II. pad
```

pander: An R pandoc writer for lists

```
> pander(as.list(rownames(mtcars)))
```

* Mazda RX4
* Mazda RX4 Wag
* Datsun 710
* Hornet 4 Drive
* Hornet Sportabout
* Valiant
* Duster 360
* Merc 240D
* Merc 230
* Merc 280
* Merc 280C
* Merc 450SE
* Merc 450SL
* Merc 450SLC
* Cadillac Fleetwood
* Lincoln Continental
* Chrysler Imperial
* Fiat 128

pander: An R pandoc writer for tables

```
> pandoc.table(head(cars))
```

```
-------------
 speed   dist
------- ------
   4       2

   4       10

   7       4

   7       22

   8       16

   9       10
-------------
```

pander: An R pandoc writer for tables

```
> pandoc.table(head(cars), style = 'rmarkdown')
```

speed	dist
4	2
4	10
7	4
7	22
8	16
9	10

pander: An R pandoc writer for tables

```
> pandoc.table(head(mtcars))
```

	mpg	cyl	disp	hp	drat
Mazda RX4	21	6	160	110	3.9
Mazda RX4 Wag	21	6	160	110	3.9
Datsun 710	22.8	4	108	93	3.85
Hornet 4 Drive	21.4	6	258	110	3.08
Hornet Sportabout	18.7	8	360	175	3.15
Valiant	18.1	6	225	105	2.76

Table: Table continues below

	wt	qsec	vs	am	gear
Mazda RX4	2.62	16.46	0	1	4
Mazda RX4 Wag	2.875	17.02	0	1	4
Datsun 710	2.32	18.61	1	1	4
Hornet 4 Drive	3.215	19.44	1	0	3
Hornet Sportabout	3.44	17.02	0	0	3

pander: An R pandoc writer for tables

```
> pandoc.table(head(mtcars), split.table = Inf)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear
Mazda RX4	21	6	160	110	3.9	2.62	16.46	0	1	4
Mazda RX4 Wag	21	6	160	110	3.9	2.875	17.02	0	1	4
Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1	4
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3
Hornet Sportabout	18.7	8	360	175	3.15	3.44	17.02	0	0	3
Valiant	18.1	6	225	105	2.76	3.46	20.22	1	0	3

pander: An R pandoc writer for tables

```
> df <- iris
> df$Species <- as.character(df$Species)
> df$Species[4] <- 'foos and bars'
> names(df) <- gsub('.', ' ', names(df), fixed = TRUE)

> pandoc.table(head(df, 4), split.table = Inf, split.cells = 5,
+   style = 'grid', justify = 'center')
```

```
+---------+--------+---------+--------+----------+
|  Sepal  | Sepal  |  Petal  | Petal  | Species  |
| Length  | Width  | Length  | Width  |          |
+=========+========+=========+========+==========+
|   5.1   |  3.5   |   1.4   |  0.2   | setosa   |
+---------+--------+---------+--------+----------+
|   4.9   |   3    |   1.4   |  0.2   | setosa   |
+---------+--------+---------+--------+----------+
|   4.7   |  3.2   |   1.3   |  0.2   | setosa   |
+---------+--------+---------+--------+----------+
|   4.6   |  3.1   |   1.5   |  0.2   |  foos    |
|         |        |         |        |   and    |
|         |        |         |        |  bars    |
+---------+--------+---------+--------+----------+
```

pander: An R pandoc writer for tables

```
> I_can_justify <- function(df)
+   ifelse(sapply(df, is.numeric), 'right', 'left')

> pandoc.table(head(df, 4), split.table = Inf, split.cells = 5,
+   style = 'grid', justify = I_can_justify(head(df, 4)))
```

+---------+--------+---------+--------+----------+
| Sepal | Sepal | Petal | Petal | Species |
| Length | Width | Length | Width | |
+=========+========+=========+========+==========+
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
+---------+--------+---------+--------+----------+
| 4.9 | 3 | 1.4 | 0.2 | setosa |
+---------+--------+---------+--------+----------+
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
+---------+--------+---------+--------+----------+
4.6	3.1	1.5	0.2	foos
				and
				bars
+---------+--------+---------+--------+----------+

Why isn't this default? Was pander created by "marketing people"? – Waldir Leoncio May 29 at 20:00

Do not use `pandoc.table` directly!

Use `pander` instead.

pander: An R pandoc writer with global options

```
> panderOptions('table.alignment.default',
+     function(df) ifelse(sapply(df, is.numeric), 'right', 'left'))
> panderOptions('table.style', 'grid')
> panderOptions('table.split.cells', 5)
> panderOptions('table.split.table', Inf)

> pander(df)
```

+---------+--------+---------+--------+-----------+
| Sepal | Sepal | Petal | Petal | Species |
| Length | Width | Length | Width | |
+=========+========+=========+========+===========+
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
+---------+--------+---------+--------+-----------+
| 4.9 | 3 | 1.4 | 0.2 | setosa |
+---------+--------+---------+--------+-----------+
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
+---------+--------+---------+--------+-----------+
4.6	3.1	1.5	0.2	foos
				and
				bars
+---------+--------+---------+--------+-----------+
| 5 | 3.6 | 1.4 | 0.2 | setosa |
+---------+--------+---------+--------+-----------+

pander: An R pandoc writer with global options

panderOptions:

- digits
- round
- decimal mark
- big mark
- date format
- keep trailing zeros
- keep line breaks
- list style
- header style
- table style
- split tables, cells
- emphasize row names
- ...

Graph options:

- margin
- font family, color, size
- grid color, lty
- legend position
- color palette
- axis angle
- symbol

evalsOptions:

- cache
- hooks
- output
- graph width, height, res
- save graph env, recordplot

pander: An R pandoc writer with a general S3 method

```
> methods(pander)
```

```
 [1] pander.anova*           pander.aov*             pander.aovlist*
 [4] pander.call*            pander.cast_df*         pander.character*
 [7] pander.clogit*          pander.coxph*           pander.CrossTable*
[10] pander.data.frame*      pander.Date*            pander.default*
[13] pander.density*         pander.describe*        pander.evals*
[16] pander.factor*          pander.formula*         pander.ftable*
[19] pander.function*        pander.glm*             pander.htest*
[22] pander.image*           pander.list*            pander.lm*
[25] pander.lme*             pander.logical*         pander.matrix*
[28] pander.microbenchmark*  pander.mtable*          pander.NULL*
[31] pander.numeric*         pander.option           pander.POSIXct*
[34] pander.POSIXlt*         pander.prcomp*          pander.randomForest*
[37] pander.rapport*         pander.return           pander.rlm*
[40] pander.sessionInfo*     pander.smooth.spline*   pander.stat.table*
[43] pander.summary.aov*     pander.summary.aovlist* pander.summary.glm*
[46] pander.summary.lm*      pander.summary.lme*     pander.summary.prcomp*
[49] pander.summary.table*   pander.survdiff*        pander.survfit*
[52] pander.table*           pander.tabular*         pander.ts*
[55] pander.zoo*
```

pander: An R pandoc writer with a general S3 method

```
> pander(1:10)
_1_, _2_, _3_, _4_, _5_, _6_, _7_, _8_, _9_ and _10_

> panderOptions('p.copula', ' og ') # Danish
> pander(1:10)
_1_, _2_, _3_, _4_, _5_, _6_, _7_, _8_, _9_og_10_

> pander(Sys.time())
2015/06/29 12:25:59 PM

> pander(sessionInfo())
**R version 3.2.0 (2015-04-16)**

**Platform:** x86_64-unknown-linux-gnu (64-bit)

**locale:**
_LC_CTYPE=en_US.UTF-8_, _LC_NUMERIC=C_, _LC_TIME=en_US.UTF-8_, _LC_COLLATE=en_US.UT

**attached base packages:**
_stats_, _graphics_, _grDevices_, _utils_, _datasets_, _methods_ and _base_

**other attached packages:**
pander(v.0.5.3)
```

pander: An R pandoc writer with a general S3 method

```
> pander(sessionInfo())
```

R version 3.2.0 (2015-04-16)

Platform: x86_64-unknown-linux-gnu (64-bit)

locale: *LC_CTYPE=en_US.UTF-8*, *LC_NUMERIC=C*, *LC_TIME=en_US.UTF-8*, *LC_COLLATE=en_US.UTF-8*, *LC_MONETARY=en_US.UTF-8*, *LC_MESSAGES=hu_HU.utf8*, *LC_PAPER=hu_HU.utf8*, *LC_NAME=C*, *LC_ADDRESS=C*, *LC_TELEPHONE=C*, *LC_MEASUREMENT=hu_HU.utf8* and *LC_IDENTIFICATION=C*

attached base packages: *stats*, *graphics*, *grDevices*, *utils*, *datasets*, *methods* and *base*

other attached packages: pander(v.0.5.3)

loaded via a namespace (and not attached): *tools(v.3.2.0)*, *Rcpp(v.0.11.6)* and *digest(v.0.6.8)*

pander: An R pandoc writer with a general S3 method

```
> pander(sessionInfo(), compact = FALSE, locale = FALSE)
```

R version 3.2.0 (2015-04-16)

Platform: x86_64-unknown-linux-gnu (64-bit)

attached base packages:

- stats
- graphics
- grDevices
- utils
- datasets
- methods
- base

other attached packages:

- data.table(v.1.9.4)
- rmarkdown(v.0.6.1)
- pander(v.0.5.3)

loaded via a namespace (and not attached):

- Rcpp(v.0.11.6)

pander: An R pandoc writer with a general S3 method

```
> pander(head(cars, 2))
```

speed	dist
4	2
4	10

```
> pander(summary(mtcars$hp))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
52	96.5	123	146.7	180	335

```
> pander(table(mtcars$am, mtcars$gear))
```

	3	4	5
0	15	4	0
1	0	8	5

pander: An R pandoc writer for list of tables

```
> pander(list(top3 = head(cars, 3), bottom3 = tail(cars, 3)))
```

```
* **top3**:

  -------------
   speed   dist
  ------- ------
     4       2

     4       10

     7       4
  -------------

* **bottom3**:

  ----------------------
       speed   dist
  -------- ------- ------
   **48**    24      93

   **49**    24      120
```

pander: An R pandoc writer for list of tables

```
> pander(list(top3 = head(cars, 3), bottom3 = tail(cars, 3)))
```

- **top3**:

speed	dist
4	2
4	10
7	4

- **bottom3**:

	speed	dist
48	24	93
49	24	120
50	25	85

pander: An R pandoc writer for statistical models

```
> pander(chisq.test(table(mtcars$am, mtcars$gear)))
```

```
-------------------------------------
 Test statistic    df        P value
---------------- ---- -----------------
     20.94         2    _2.831e-05_ * * *
-------------------------------------

Table: Pearson's Chi-squared test: `table(mtcars$am, mtcars$gear)`

 **WARNING**^[Chi-squared approximation may be incorrect]
```

Table 7:Pearson's Chi-squared test: `table(mtcars$am, mtcars$gear)`

Test statistic	df	P value
20.94	2	*2.831e-05* * * *

WARNING[1]

[1] Chi-squared approximation may be incorrect

pander: An R pandoc writer for statistical models

```
> ## Dobson (1990) Page 93: Randomized Controlled Trial
> counts    <- c(18, 17, 15, 20, 10, 20, 25, 13, 12)
> outcome   <- gl(3, 1, 9)
> treatment <- gl(3, 3)
> m <- glm(counts ~ outcome + treatment, family = poisson())
> pander(m)
```

	Estimate	Std. Error	z value	Pr(>|z|)
outcome2	-0.4543	0.2022	-2.247	0.02465
outcome3	-0.293	0.1927	-1.52	0.1285
treatment2	1.338e-15	0.2	6.69e-15	1
treatment3	1.421e-15	0.2	7.105e-15	1
(Intercept)	3.045	0.1709	17.81	5.427e-71

pander: An R pandoc writer for statistical models

```
> pander(anova(m))
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL	NA	NA	8	10.58
outcome	2	5.452	6	5.129
treatment	2	2.665e-15	4	5.129

Table: Analysis of Deviance Table

pander: An R pandoc writer for statistical models

```
> panderOptions('missing', '')
> pander(anova(m))
```

```
-------------------------------------------------------
             Df    Deviance    Resid. Df    Resid. Dev
-------------- ---- ---------- ----------- ------------
   **NULL**                          8          10.58

  **outcome**     2      5.452          6          5.129

 **treatment**    2   2.665e-15         4          5.129
-------------------------------------------------------

Table: Analysis of Deviance Table
```

pander: An R pandoc writer for statistical models

```
> pander(aov(m))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
outcome	2	92.67	46.33	2.224	0.2242
treatment	2	8.382e-31	4.191e-31	2.012e-32	1
Residuals	4	83.33	20.83		

Table: Analysis of Variance Model

pander: An R pandoc writer for statistical models

```
> pander(prcomp(USArrests))
```

	PC1	PC2	PC3	PC4
Murder	0.0417	-0.04482	0.07989	-0.9949
Assault	0.9952	-0.05876	-0.06757	0.03894
UrbanPop	0.04634	0.9769	-0.2005	-0.05817
Rape	0.07516	0.2007	0.9741	0.07233

Table: Principal Components Analysis

pander: An R pandoc writer for statistical tables

```
> pander(descr::CrossTable(mtcars$cyl, mtcars$gear))
```

\ mtcars$cyl	mtcars$gear\ 3	\ 4	\ 5	\ Total
4\	\	\	\	\
N\	1\	8\	2\	11\
Chi-square\	3.3502\	3.6402\	0.0460\	\
Row(%)\	9.0909%\	72.7273%\	18.1818%\	34.3750%\
Column(%)\	6.6667%\	66.6667%\	40.0000%\	\
Total(%)	3.125%	25.000%	6.250%	
6\	\	\	\	\
N\	2\	4\	1\	7\
Chi-square\	0.5003\	0.7202\	0.0080\	\
Row(%)\	28.5714%\	57.1429%\	14.2857%\	21.8750%\
Column(%)\	13.3333%\	33.3333%\	20.0000%\	\
Total(%)	6.250%	12.500%	3.125%	
8\	\	\	\	\
N\	12\	0\	2\	14\
Chi-square\	4.5054\	5.2500\	0.0161\	\
Row(%)\	85.7143%\	0.0000%\	14.2857%\	43.7500%\
Column(%)\	80.0000%\	0.0000%\	40.0000%\	\
Total(%)	37.500%	0.000%	6.250%	
Total\	15\	12\	5\	32\
	46.875%	37.5%	15.625%	

---

pander: An R pandoc writer for statistical tables

```
> pander(descr::CrossTable(mtcars$cyl, mtcars$gear))
```

mtcars$cyl	mtcars$gear 3	4	5	Total
4				
N	1	8	2	11
Chi-square	3.3502	3.6402	0.0460	
Row(%)	9.0909%	72.7273%	18.1818%	
Column(%)	6.6667%	66.6667%	40.0000%	34.3750%
Total(%)	3.125%	25.000%	6.250%	
6				
N	2	4	1	7
Chi-square	0.5003	0.7202	0.0080	
Row(%)	28.5714%	57.1429%	14.2857%	
Column(%)	13.3333%	33.3333%	20.0000%	21.8750%
Total(%)	6.250%	12.500%	3.125%	
8				
N	12	0	2	14
Chi-square	4.5054	5.2500	0.0161	
Row(%)	85.7143%	0.0000%	14.2857%	
Column(%)	80.0000%	0.0000%	40.0000%	43.7500%
Total(%)	37.500%	0.000%	6.250%	
Total	15	12	5	32
	46.875%	37.5%	15.625%	

pander: An R pandoc writer for statistical tables

```
> library(tables)
> pander(tabular(as.factor(am) ~ (mpg+hp+qsec) * (mean+median),
+   data = mtcars))
```

\	mpg\	\	hp\	\	qsec\	\
as.factor(am)	mean	median	mean	median	mean	median
0	17.15	17.3	160.3	175	18.18	17.82
1	24.39	22.8	126.8	109	17.36	17.02

	mpg		hp		qsec	
as.factor(am)	mean	median	mean	median	mean	median
0	17.15	17.3	160.3	175	18.18	17.82
1	24.39	22.8	126.8	109	17.36	17.02

pander: An R pandoc writer for tables of statistical models

```
> library(memisc)
> berk  <- Aggregate(Table(Admit,Freq) ~ .,data = UCBAdmissions)
> berk0 <- glm(cbind(Admitted,Rejected) ~ 1, data = berk, family = "binomial")
> berk1 <- glm(cbind(Admitted,Rejected) ~ Gender, data = berk, family = "binomial")
> berk2 <- glm(cbind(Admitted,Rejected) ~ Gender + Dept, data = berk, family = "binomial")
> pander(mtable(berk0, berk1, berk2, coef.style = 'horizontal',
+   summary.stats = c('Deviance', 'AIC', 'N')), style = 'grid')
```

	berk0	berk1	berk2
(Intercept)	-0.457***\ (0.031)	-0.220***\ (0.039)	0.582***\ (0.069)
Gender: Female/Male	\	-0.610***\ (0.064)	0.100\ (0.081)
Dept: B/A	\	\	-0.043\ (0.110)
Dept: C/A	\	\	-1.263***\ (0.107)
Dept: D/A	\	\	-1.295***\ (0.106)
Dept: E/A	\	\	-1.739***\ (0.126)
Dept: F/A	\	\	-3.306***\ (0.170)

pander: An R pandoc writer for tables of statistical models

```
> library(memisc)
> berk  <- Aggregate(Table(Admit,Freq) ~ .,data = UCBAdmissions)
> berk0 <- glm(cbind(Admitted,Rejected) ~ 1, data = berk, family = "binomial")
> berk1 <- glm(cbind(Admitted,Rejected) ~ Gender, data = berk, family = "binomial")
> berk2 <- glm(cbind(Admitted,Rejected) ~ Gender + Dept, data = berk, family = "binomial")
> pander(mtable(berk0, berk1, berk2, coef.style = 'horizontal',
+   summary.stats = c('Deviance', 'AIC', 'N')), justify  = 'left')
```

	berk0	berk1	berk2
(Intercept)	-0.457***	-0.220***	0.582***
	(0.031)	(0.039)	(0.069)
Gender: Female/Male		-0.610***	0.100
		(0.064)	(0.081)
Dept: B/A			-0.043
			(0.110)
Dept: C/A			-1.263***
			(0.107)
Dept: D/A			-1.295***
			(0.106)
Dept: E/A			-1.739***
			(0.126)
Dept: F/A			-3.306***
			(0.170)
Deviance	877.056	783.607	20.204
AIC	947.996	856.547	103.144
N	4526	4526	4526

pander: An R pandoc writer with a general S3 method

```
> methods(pander)
```

```
 [1] pander.anova*           pander.aov*             pander.aovlist*
 [4] pander.call*            pander.cast_df*         pander.character*
 [7] pander.clogit*          pander.coxph*           pander.CrossTable*
[10] pander.data.frame*      pander.Date*            pander.default*
[13] pander.density*         pander.describe*        pander.evals*
[16] pander.factor*          pander.formula*         pander.ftable*
[19] pander.function*        pander.glm*             pander.htest*
[22] pander.image*           pander.list*            pander.lm*
[25] pander.lme*             pander.logical*         pander.matrix*
[28] pander.microbenchmark*  pander.mtable*          pander.NULL*
[31] pander.numeric*         pander.option           pander.POSIXct*
[34] pander.POSIXlt*         pander.prcomp*          pander.randomForest*
[37] pander.rapport*         pander.return           pander.rlm*
[40] pander.sessionInfo*     pander.smooth.spline*   pander.stat.table*
[43] pander.summary.aov*     pander.summary.aovlist* pander.summary.glm*
[46] pander.summary.lm*      pander.summary.lme*     pander.summary.prcomp*
[49] pander.summary.table*   pander.survdiff*        pander.survfit*
[52] pander.table*           pander.tabular*         pander.ts*
[55] pander.zoo*
```

pander: An R pandoc writer with some useful features

- several table tweaks described in `pandoc.table`
- captions for tables and images with `set.caption`
- temporarily override default `panderOptions` with `set.alignment`
- unified graphs
- `knitr` support

pander: Advanced table features

```
> pander(formals(pandoc.table.return))
```

$t

$caption

$digits
panderOptions("digits")

$decimal.mark
panderOptions("decimal.mark")

$big.mark
panderOptions("big.mark")

$round
panderOptions("round")

$missing
panderOptions("missing")

$justify

$style
c("multiline", "grid", "simple", "rmarkdown")

$split.tables
panderOptions("table.split.table")
...

pander: Advanced table features

```
> pander(formals(pandoc.table.return))
```

- **t**:
- **caption**:
- **digits**: `panderOptions("digits")`
- **decimal.mark**: `panderOptions("decimal.mark")`
- **big.mark**: `panderOptions("big.mark")`
- **round**: `panderOptions("round")`
- **missing**: `panderOptions("missing")`
- **justify**:
- **style**: `c("multiline", "grid", "simple", "rmarkdown")`
- **split.tables**: `panderOptions("table.split.table")`
- **split.cells**: `panderOptions("table.split.cells")`
- **keep.trailing.zeros**: `panderOptions("keep.trailing.zeros")`
- **keep.line.breaks**: `panderOptions("keep.line.breaks")`
- **plain.ascii**: `panderOptions("plain.ascii")`
- **use.hyphening**: `panderOptions("use.hyphening")`
- **emphasize.rownames**: `panderOptions("table.emphasize.rownames")`
- **emphasize.rows**:

pander: Advanced table features (captions)

```
> set.caption('Set your caption before printing.')
> set.alignment('right')
> pander(head(iris, 3), split.table = Inf)
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa

Table: Set your caption before printing.

pander: Advanced table features (hyphening)

```
> pander(data.frame(A = 'The hyphen (-) is a punctuation mark used to join words and
+   split.cells = 24,
+   use.hyphening = TRUE)
```

A
The hyphen (-) is a punctuation mark used to join words and to sepa- rate syllables of a sin- gle word. The use of hy- phens is called hyphen- ation.

pander: Advanced table features (emphasize cells)

```
> emphasize.cols(1)
> emphasize.rows(2)
> pander(head(cars))
```

---

speed	dist
4	2
4	*10*
7	4
7	22
8	16
9	10

---

pander: Advanced table features (emphasize cells)

```
> emphasize.strong.cells(which(head(cars) %% 2 == 0, arr.ind = TRUE))
> pander(head(cars))
```

---

speed	dist
4	**2**
4	**10**
7	**4**
7	**22**
8	**16**
9	**10**

---

pander: Advanced table features (knitr integration)

Gergely Daróczi
@daroczig

Thanks to @hadleywickham, now it's even easier to use "pander" inside of #knitr to generate #markdown tables: blog.rapporter.net/2014/09/pander… #rstats

RETWEETS 3 FAVORITES 18

1:30 AM - 19 Sep 2014

https://twitter.com/daroczig/status/512745401342885889

pander: Some further helper functions to eval

```
> evals('chisq.test(mtcars$am, mtcars$gear)[[1]]')

$src
[1] "chisq.test(mtcars$am, mtcars$gear)[[1]]"

$result
X-squared
 20.94467

$output
[1] "X-squared " " 20.94467 "

$type
[1] "numeric"

$msg
$msg$messages
NULL

$msg$warnings
[1] "Chi-squared approximation may be incorrect"

$msg$errors
NULL


$stdout
NULL

attr(,"class")
[1] "evals"
```

pander: Some nice features of evals

```
> x <- mtcars$hp
> y <- 1e5
> system.time(evals('sapply(rep(x, y), mean)'))
```

pander: Some nice features of evals

```
> x <- mtcars$hp
> y <- 1e5
> system.time(evals('sapply(rep(x, y), mean)'))
```

```
   user  system elapsed
 27.735   0.213  28.241
```

```
> system.time(evals('sapply(rep(x, y), mean)'))
```

pander: Some nice features of evals

```
> x <- mtcars$hp
> y <- 1e5
> system.time(evals('sapply(rep(x, y), mean)'))
```

```
   user  system elapsed
 27.735   0.213  28.241
```

```
> system.time(evals('sapply(rep(x, y), mean)'))
```

```
   user  system elapsed
  0.002   0.000   0.003
```

pander: Some nice features of evals

```
## sapply(rep(mtcars$hp, 1e5), mean)
> f <- sapply
> g <- rep
> h <- mean
> X <- mtcars$hp * 1
> Y <- 1000
> system.time(evals('f(g(X, Y), h)'))
```

pander: Some nice features of evals

```
## sapply(rep(mtcars$hp, 1e5), mean)
> f <- sapply
> g <- rep
> h <- mean
> X <- mtcars$hp * 1
> Y <- 1000
> system.time(evals('f(g(X, Y), h)'))
```

```
   user  system elapsed
  0.233   0.000   0.233
```

```
> system.time(evals('f(g(X, Y), h)'))
```

```
   user  system elapsed
  0.003   0.000   0.002
```

pander: Some nice features of evals

Caching algorithm:

- Each chunk is `parse`d to single R **expressions**.
- Each parsed expression's **part** (function, variable, constant etc.) is evaluated separately to a `list`. This list describes the unique structure *and* the content of the passed R expressions.
- The list of these R objects is serialized, then an `SHA-1` hash is computed.
- A **hash** is computed of each list element and *cached* as well.
- Return cached results or evaluated.
- The results and the modified R objects of the environment are optionally saved to cache.

pander for literate programming on its own

```
> x <- Pandoc.brew(text = '
+ Pi equals to <%=pi%>, and the best damn cars are:
+ <%=head(mtcars, 2)%>
+ ')
```

```
Pi equals to _3.142_, and the best damn cars are:

-------------------------------------------------------------------------------
                  mpg   cyl   disp   hp   drat   wt     qsec   vs   am   gear   c
------------------ ----- ----- ------ ---- ------ ----- ------ ---- ---- ------ ---
   **Mazda RX4**     21     6    160   110   3.9   2.62  16.46   0    1     4      4

 **Mazda RX4 Wag**   21     6    160   110   3.9   2.875 17.02   0    1     4      4
-------------------------------------------------------------------------------
```

pander for literate programming on its own

```
> str(x)
List of 3
 $ :List of 4
  ..$ type  : chr "text"
  ..$ text  :List of 2
  .. ..$ raw : chr "\nPi equals to <%=pi%>, and the best damn cars are:\n"
  .. ..$ eval: chr "\nPi equals to _3.142_, and the best damn cars are:\n"
  ..$ chunks:List of 2
  .. ..$ raw : chr "<%=pi%>"
  .. ..$ eval: chr "_3.142_"
  ..$ msg   :List of 3
  .. ..$ messages: NULL
  .. ..$ warnings: NULL
  .. ..$ errors  : NULL
 $ :List of 2
  ..$ type   : chr "block"
  ..$ robject:List of 6
  .. ..$ src   : chr "head(mtcars, 2)"
  .. ..$ result:'data.frame':   2 obs. of  11 variables:
  .. .. ..$ mpg : num [1:2] 21 21
  .. .. ..$ cyl : num [1:2] 6 6
  .. .. ..$ disp: num [1:2] 160 160
  .. .. ..$ hp  : num [1:2] 110 110
  .. .. ..$ drat: num [1:2] 3.9 3.9
  .. .. ..$ wt  : num [1:2] 2.62 2.88
  .. .. ..$ qsec: num [1:2] 16.5 17
  .. .. ..$ vs  : num [1:2] 0 0
  .. .. ..$ am  : num [1:2] 1 1
  .. .. ..$ gear: num [1:2] 4 4
  .. .. ..$ carb: num [1:2] 4 4
  .. ..$ type  : chr "data.frame"
  .. ..$ msg   :List of 3
  .. .. ..$ messages: NULL
```

The R Markdown toolbox

- literate programming engine (`knitr` , `brew` , `pander` etc.)
- document converter (**pandoc**, AsciiDoc, Textile, reStructuredText etc.)
- document templates, stylesheets (CSS, JavaScript, LaTeX, docx)
- version control (git, SubVersion etc.) – always do this
- document storage, publisher (GitHub, rpubs, shinyapps.io etc)

Tools to pre-process *Rmd* files:

- tool to transform R objects into markdown (`kable` , `pander`)
- **templating languages handling loops, conditional expressions and child documents in knitr (`brew` , `pander` , `R.rsp`)**
- markdown templates in `rapport`
- GNU Make: text file to describe build workflow
- `remake` package: Make-like declarative workflows in R

pander for literate programming on its own

```
<% for (v in names(mtcars)) { %>
The mean of <%= pandoc.verbatim.return(v) %> is <%= mean(mtcars[, v]) %>
<% } %>
```

```
> Pandoc.brew('demo.brew')
```

The mean of `mpg` is *20.09*

The mean of `cyl` is *6.188*

The mean of `disp` is *230.7*

The mean of `hp` is *146.7*

The mean of `drat` is *3.597*

The mean of `wt` is *3.217*

The mean of `qsec` is *17.85*

The mean of `vs` is *0.4375*

The mean of `am` is *0.4062*

The mean of `gear` is *3.688*

pander for literate programming on its own

```
<% for (v in names(mtcars)) { %>

<% if (nchar(v) == 3) { %>

## <%= v %>

The mean is <%= mean(mtcars[, v]) %>

<% }} %>
```

```
> Pandoc.brew('demo.brew')
```

```
## mpg

The mean is _20.09_

## cyl

The mean is _6.188_
```

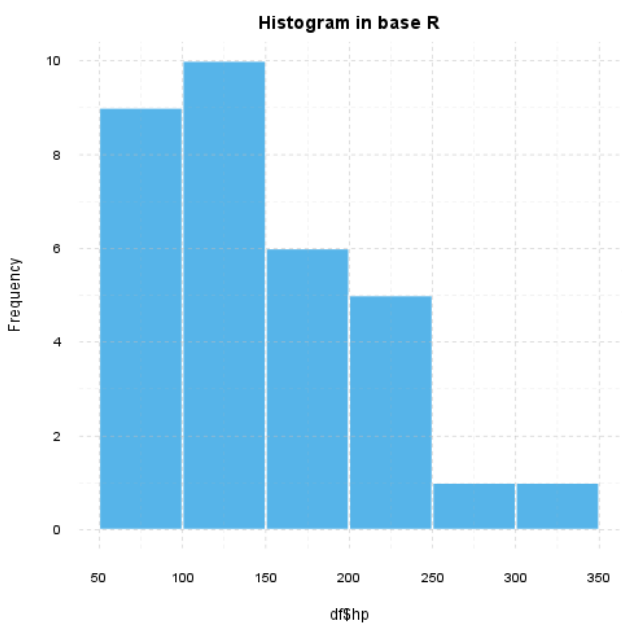
pander for literate programming with unified graphs

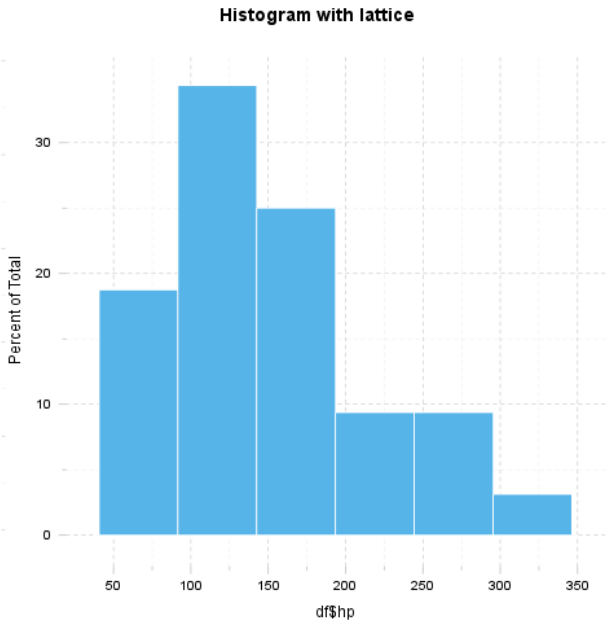
- https://github.com/Rapporter/pander/blob/master/inst/examples/graphs.brew
- http://rapporter.github.io/pander/graphs.html

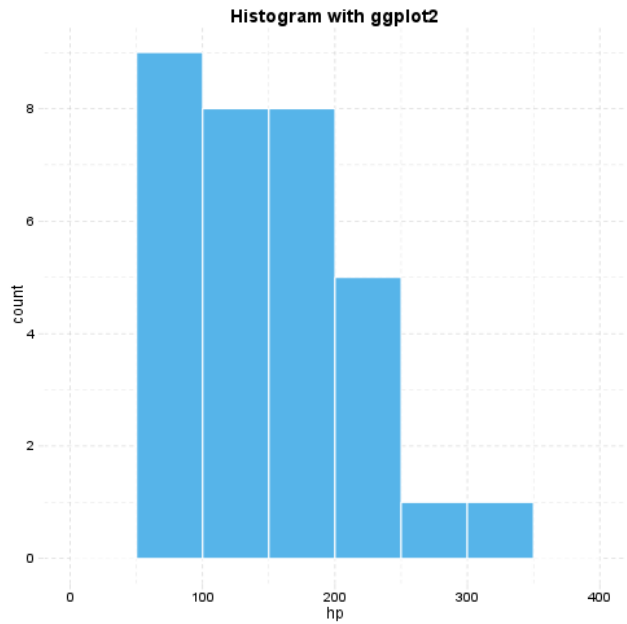
What graphic library was used to generate the following plot?

pander for literate programming with unified graphs

```
> hist(df$hp, main = "Histogram in base R")
> histogram(df$hp, main = "Histogram with lattice")
> ggplot(df) + geom_histogram(aes(x = hp), binwidth = 50) +
+   ggtitle("Histogram with ggplot2")
```

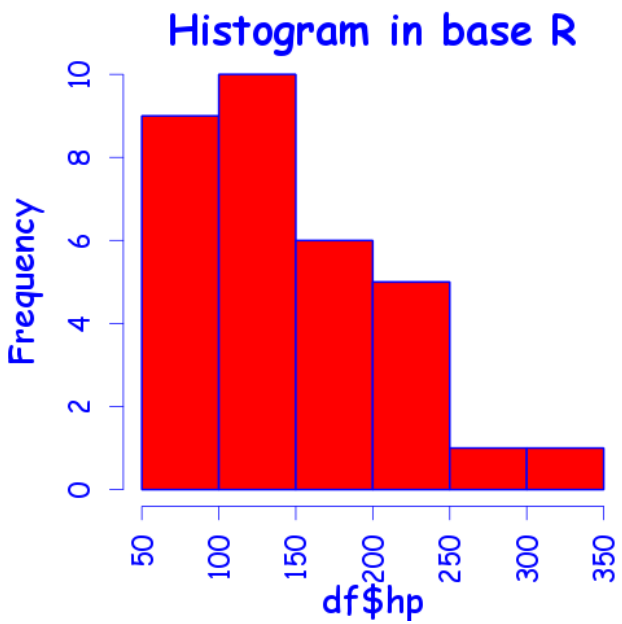


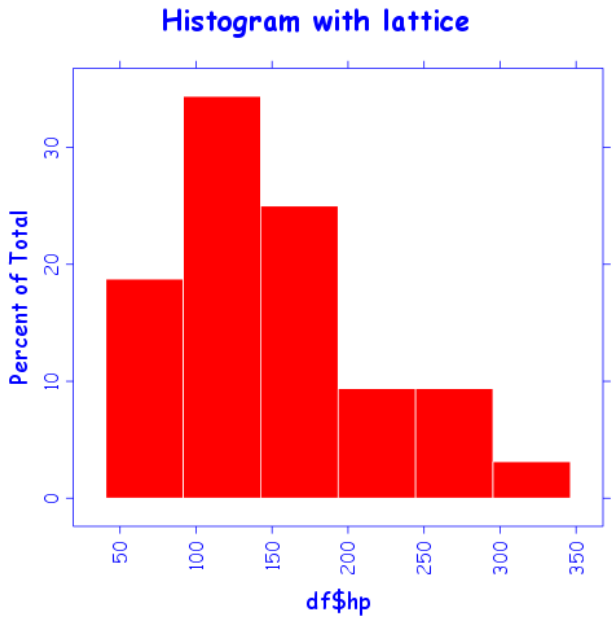


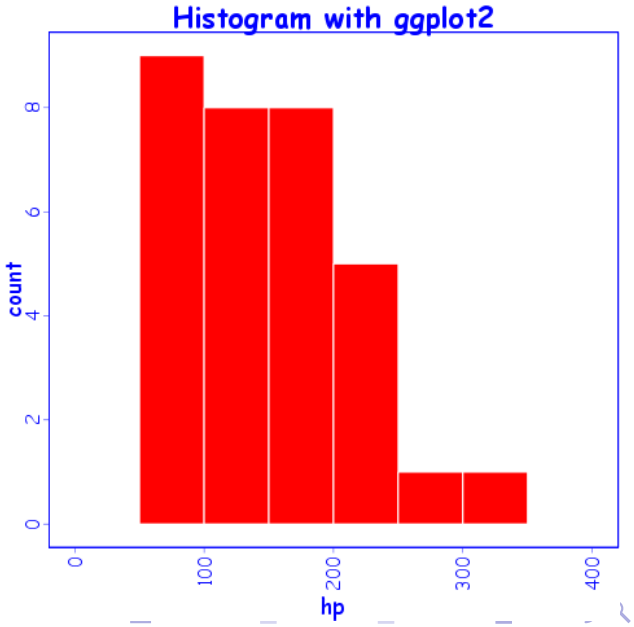


pander for literate programming with unified graphs

```
panderOptions('graph.fontfamily', "Comic Sans MS")
panderOptions('graph.fontsize', 18)
panderOptions('graph.fontcolor', 'blue')
panderOptions('graph.grid.color', 'blue')
panderOptions('graph.axis.angle', 3)
panderOptions('graph.boxes', T)
panderOptions('graph.legend.position', 'top')
panderOptions('graph.colors', rainbow(5))
panderOptions('graph.grid', FALSE)
panderOptions('graph.symbol', 22)
```







Highlighted pander features

- `brew` loops and conditional parts of a report,
- unify and theme plots with glovbal options,
- render all R objects automatically in Pandoc's markdown,
- recording all warning/error messages besides the raw R objects along with anything printed to stdout and the printed results,
- custom caching mechanism to disk or RAM with auto-dependecy,
- convert to HTML/pdf/odt/docx at one go,
- no chunk options (only workaround),
- building reports also in interactive session with an R5 reference class.

The Pandoc R5 reference class

```
> myReport <- Pandoc$new('Gergely Daróczi', 'Demo')

> myReport$add.paragraph('Hello there, this is a really short tutorial!')

> fit <- with(lm(mpg ~ hp + wt), data = mtcars)
> myReport$add(fit)

> myReport$add(plot(fit))
```

```
Demo
====
 written by *Gergely Daróczi* at *Mon Jun 29 23:06:51 2015*

  This report holds 4 block(s).

---


Hello there, this is a really short tutorial!

-------------------------------------------------------------
              Estimate   Std. Error   t value   Pr(>|t|)
---------------- ---------- ------------ --------- ----------
     **hp**        -0.03177    0.00903     -3.519    0.001451

     **wt**         -3.878      0.6327     -6.129    1.12e-06

 **(Intercept)**    37.23       1.599       23.28    2.565e-20
-------------------------------------------------------------

Table: Fitting linear model: mpg ~ hp + wt
```

The Pandoc R5 reference class

```
myReport$export()
```

Gergely Daroczi

Demo

Mon Jun 29 23:06:51 2015

Hello there, this is a really short tutorial!

Table 1: Fitting linear model: mpg ~ hp + wt

	Estimate	Std. Error	t value	Pr(>|t|)
hp	-0.03177	0.00903	-3.519	0.001451
wt	-3.878	0.6327	-6.129	1.12e-06
(Intercept)	37.23	1.599	23.28	2.565e-20

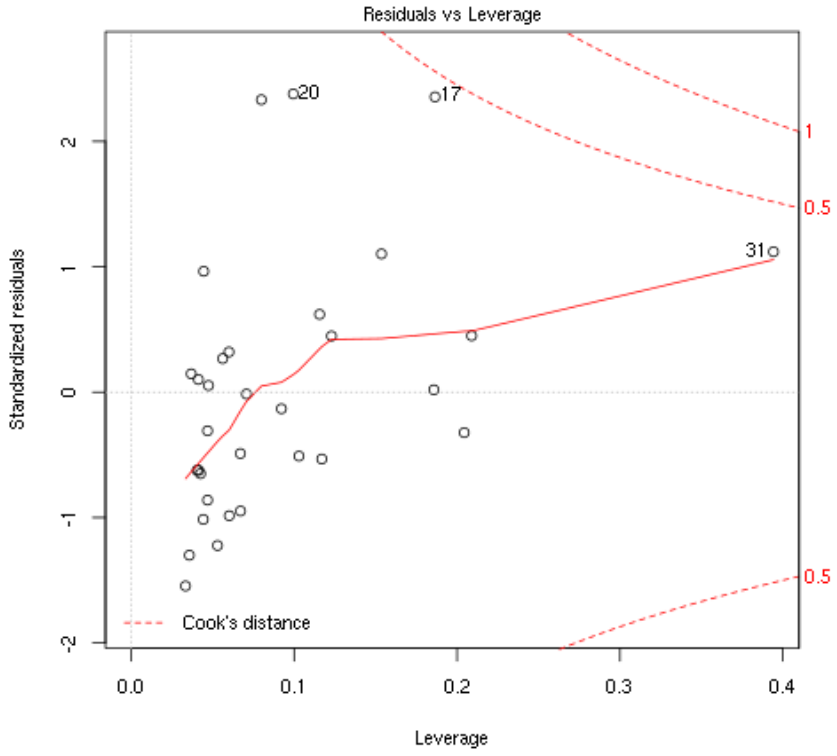
The Pandoc R5 reference class

```
myReport$export('html')
```

Hello there, this is a really short tutorial!

Fitting linear model: mpg ~ hp + wt

	Estimate	Std. Error	t value	Pr(>|t|)
hp	-0.03177	0.00903	-3.519	0.001451
wt	-3.878	0.6327	-6.129	1.12e-06
(Intercept)	37.23	1.599	23.28	2.565e-20



The Pandoc R5 reference class

```
myReport$export('docx')
```

Gergely Daroczi

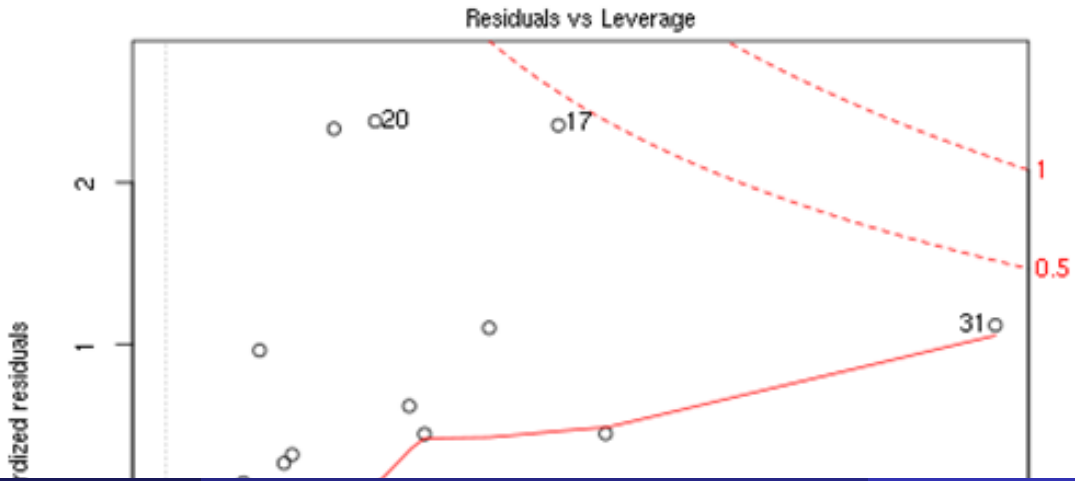
Demo

Mon Jun 29 23:06:51 2015

Hello there, this is a really short tutorial!

Fitting linear model: mpg ~ hp + wt

	Estimate	Std. Error	t value	Pr(>|t|)
hp	-0.03177	0.00903	-3.519	0.001451
wt	-3.878	0.6327	-6.129	1.12e-06
(Intercept)	37.23	1.599	23.28	2.565e-20



pander: An R Pandoc Writer

The main aim of the *pander* R package is to provide a minimal and easy tool for rendering **R objects** into Pandoc's **markdown**. The package is also capable of exporting/converting complex Pandoc documents (reports) in various ways. Regarding the difference between `pander` and other packages for exporting R objects to different file formats, please refer to this section.

Installation

The stable version can be found on CRAN and can be installed easily in the R console like any other package:

```
install.packages('pander')
```

On the other hand, I welcome everyone to use the most recent version of the package with quick-fixes, new features and probably new bugs. It's currently hosted on GitHub

Current build and test coverage status: build passing coverage 75%

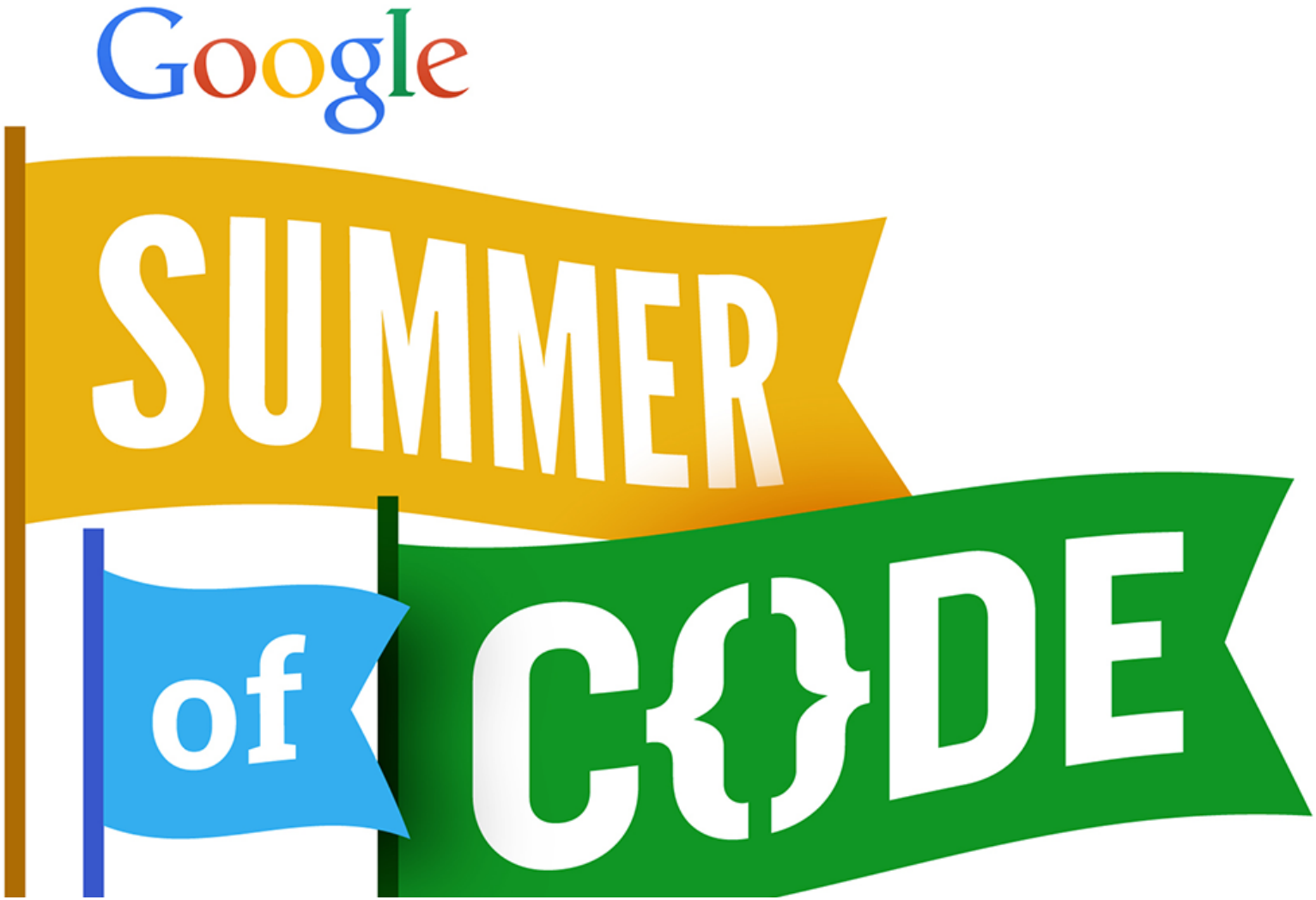
It can be installed easily with the nifty function of the `devtools` package from CRAN:

```
library(devtools)
install_github('pander', 'Rapporter')
```

Or download the sources and build manually. If you're running R on Windows, you need to install Rtools.

http://rapporter.github.io/pander/

pander: Documentation



Roman Tsegelskyi: 188 commits / 7,660 ++ / 4,347 --

The great pander/brew test

Create a HTML report on `mtcars` including

- descriptive stats on `hp`, `wt` and `gear`
- cross table of `gear` and `am`
- a paragraph on how to compute the standard deviation including
  - a formula
  - custom R function to compute that
  - demo run on `hp`
- scatterplot on `hp` and `wt` with a linear trend line

Resources:

- http://www.statmethods.net/stats
- http://rapporter.github.io/pander
  - Creating Pandoc documents
  - General options
  - Markdown tables

The R Markdown toolbox

- literate programming engine (`knitr` , `brew` , `pander` etc.)
- document converter (**pandoc**, AsciiDoc, Textile, reStructuredText etc.)
- document templates, stylesheets (CSS, JavaScript, LaTeX, docx)
- version control (git, SubVersion etc.) – always do this
- document storage, publisher (GitHub, rpubs, shinyapps.io etc)

Tools to pre-process *Rmd* files:

- tool to transform R objects into markdown (`kable` , `pander`)
- templating languages handling loops, conditional expressions and child documents in knitr (`brew` , `pander` , `R.rsp`)
- **markdown templates in** `rapport`
- GNU Make: text file to describe build workflow
- `remake` package: Make-like declarative workflows in R

“A n00b-friendly interface to statistical report creation” (c) 2011

http://rapport-package.info

(And that’s why we created `pander`.)

The overall structure of reusable rapport templates

```
<!--head
meta:
  title: ...
  author: ...
  description: ...
  packages:
  ...
inputs:
- name: ...
  class: ...
head-->

<% for (...) { # loop %>
## Subtitle with <%= inline code chunk %>
<%= R.code.transformed.to.markdown(...) %>
<% } %>
```

The YAML header of reusable rapport templates

```
<!--head
meta:
  title: Rapport demo
  author: daroczig
  description: This is POC demo rapport templates
  packages:
  - ggplot2
  - descr
inputs:
- name: v
  label: Variable
  description: A variable
  class: numeric
  length:
    min: 1
    max: 1
  value: ~
  required: TRUE
  standalone: FALSE
-->
```

The report body of reusable rapport templates

```
# A quick analyis on <%= v.name %>

The mean of <%= v.name %> is <%= mean(v) %> and the
standard deviation is <%= sd(v) %>. Let us also
check Tukey's five number summary:

<%= summary(v) %>

## Tables are boring!

<%=
set.caption(paste('Histogram of', v.name))
hist(v, xlab = v, main = '')
%>
```

Running rapport templates

```
> library(rapport)
> rapport('demo.rapport', v = 'hp', data = mtcars)
```

```
# A quick analyis on hp

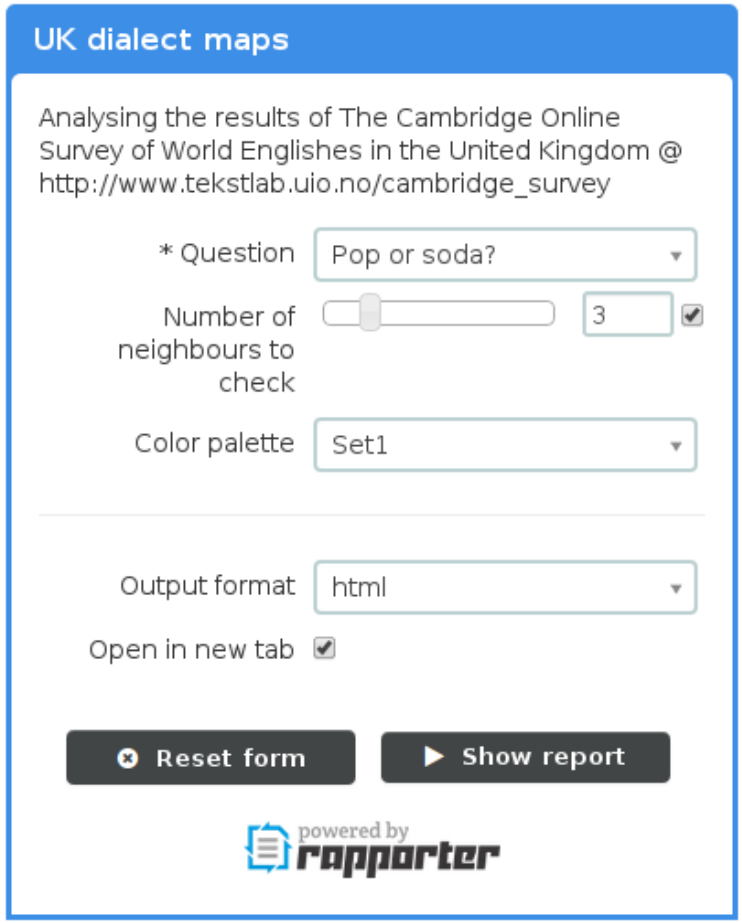

The mean of hp is _146.7_ and the
standard deviation is _68.56_. Let us also
check Tukey's five number summary:

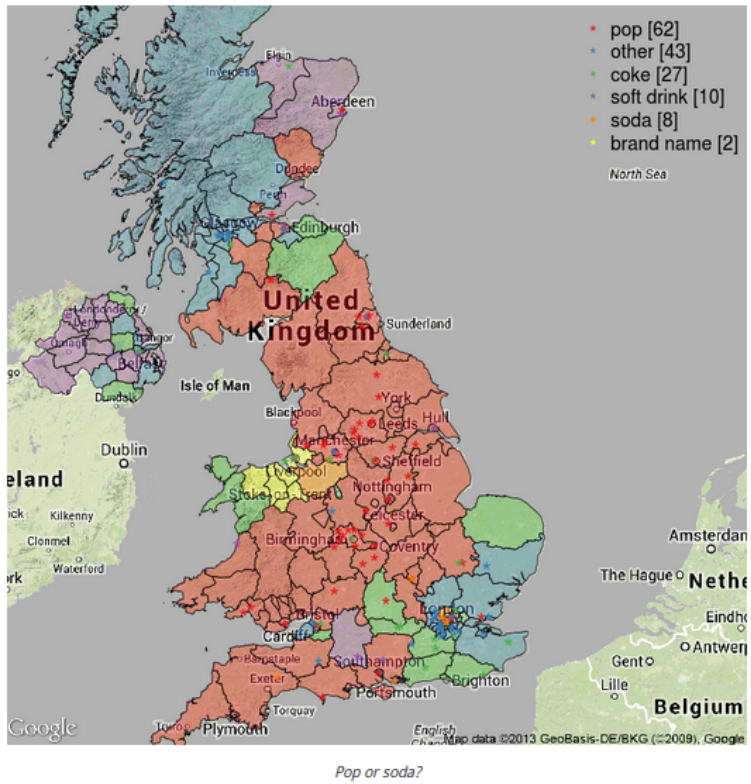

-------------------------------------------------
 Min.   1st Qu.   Median   Mean   3rd Qu.   Max.
------ --------- -------- ------ --------- ------
  52      96.5      123    146.7     180      335
-------------------------------------------------


## Tables are boring!

![Histogram of hp](plots/rapport-demo.rapport-7-1.png)
```

A more complex rapport template





Pop or soda?

http://blog.rapporter.net/2013/07/uk-dialect-maps.html

Summary

The **most popular category** in the United Kingdom was <<*pop*>> for <<*Pop or soda?*>> chosen by *four tenth* of the respondents.

And the most important differences between the countries can be summarised as:

- it seems, that *two tenth* of Brittish people disagree with <<*other*>> that is low comparing to e.g. Scottish people
- eventually, *less then one tenth* of Brittish people tends to dislike the answer <<*soft drink*>> that is low compared to lets say Northern Irish people
- it seems, that *one half* of people living in Northern Ireland tends to like the answer <<*soft drink*>> that is high comparing to e.g. Welsh citizens
- it seems, that *two tenth* of Scottish people tends to dislike the answer <<*pop*>> that is low compared to the average
- it seems, that *five tenth* of Scottish people love the answer <<*other*>> that is high compared to lets say Brittish people

http://blog.rapporter.net/2013/07/uk-dialect-maps.html

The great rapport test

Create a `rapport` template that takes a `data.frame`, a `variable` and `color` name as inputs, then render

- some descriptive stats inline and via tables as well,
- a histogram with customizable `color`,
- a detailed (step by step) description on how to compute the standard deviation including intermediate results.

Resources:

- http://www.statmethods.net/stats
- http://rapporter.github.io/pander
- http://rapport-package.info/

Congrats, you made it!

Further resources

On Markdown:

- http://markdowntutorial.com/
- http://pandoc.org/
- http://commonmark.org/

On pandoc:

- http://pandoc.org/README.html

On `knitr` :

- http://yihui.name/knitr/
- https://github.com/yihui/knitr-examples

On `pander` :

- https://github.com/Rapporter/pander
- http://rapporter.github.io/pander/
- http://stackoverflow.com/questions/tagged/pander

On `rapport` :

- http://rapport-package.info

RStudio cheat sheets:

- http://www.rstudio.com/resources/cheatsheets/