

All Training materials are provided "as is" and without warranty and RStudio disclaims any and all express and implied warranties including without limitation the implied warranties of title, fitness for a particular purpose, merchantability and noninfringement.

The Training Materials are licensed under the Creative Commons Attribution-Noncommercial 3.0 United States License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/us/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

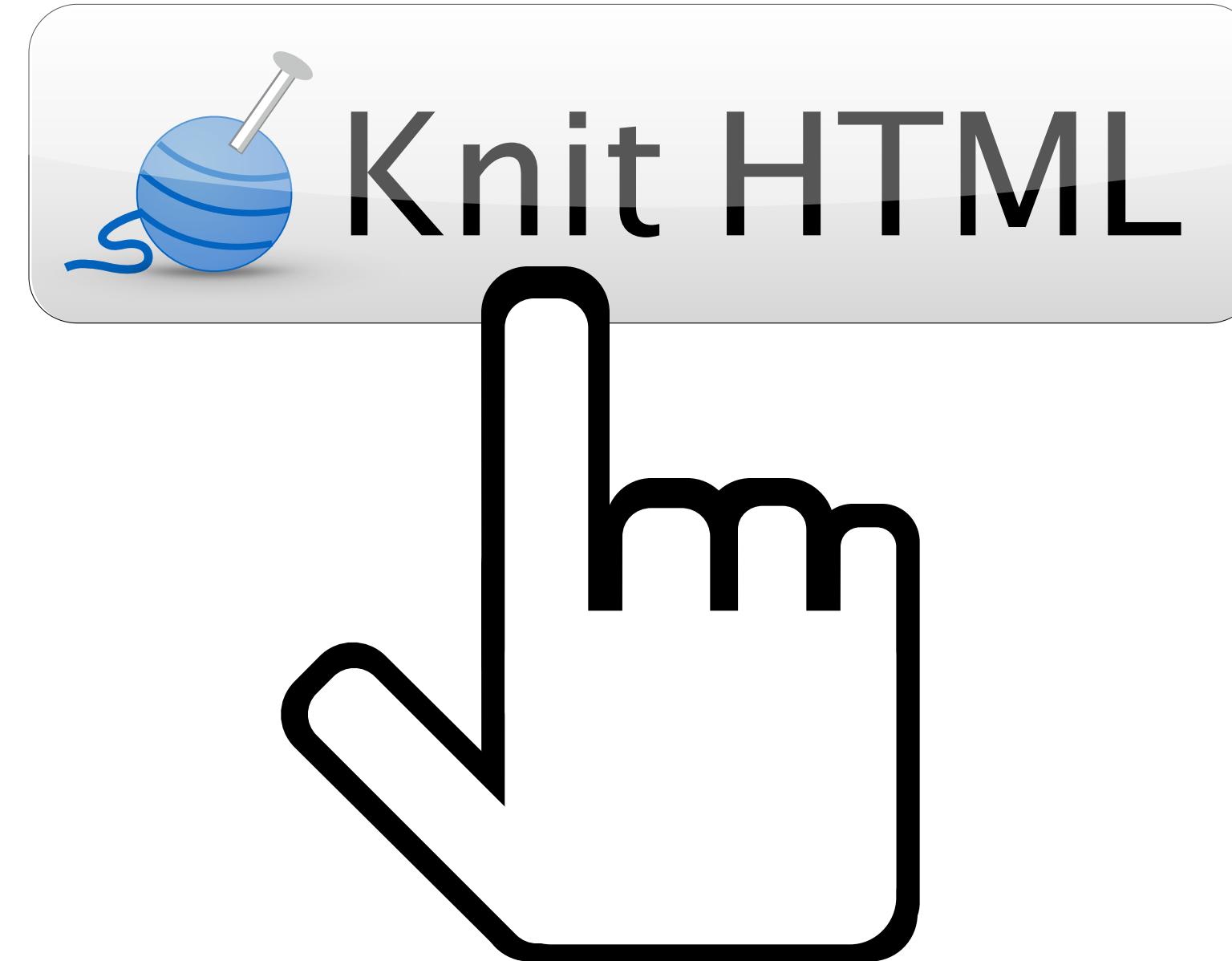
# Prereqs

```
install.packages("reportsWS",
repos = c(
  "http://cran.rstudio.com",
  "http://training.rstudio.com"
))
library(reportsWS)
```



# Reproducible reports with R Markdown

One format to rule them all



Joe Cheng

Software Engineer  
September 2015  
Email: [joe@rstudio.com](mailto:joe@rstudio.com)



RStudio IDE  
RPubs  
httpuv  
Shiny  
Shiny Server  
packrat  
htmlwidgets  
leaflet  
d3heatmap



@jcheng

Slides at: [bit.ly/EARL2015-Reports](https://bit.ly/EARL2015-Reports)

O'REILLY®



# Hands-On Programming with R

WRITE YOUR OWN FUNCTIONS AND SIMULATIONS

Garrett Grolemund  
Foreword by Hadley Wickham

O'REILLY®



# Introduction to Data Science with R

*Garrett Grolemund*

**VIDEO**

O'REILLY®

# Expert Data Wrangling with R

Streamline Your Work  
with `tidyR`, `dplyr`,  
and `ggvis`

*Garrett Grolemund*

**VIDEO**

Slides at: [bit.ly/EARL2015-Reports](http://bit.ly/EARL2015-Reports)

© 2015 RStudio, Inc. All rights reserved.



# DataCamp

[www.datacamp.com](http://www.datacamp.com)



**Data Visualization  
in R with ggvis**



**Reporting with  
R Markdown**



**Data Manipulation  
in R with dplyr**



**How to use the  
RStudio IDE**

Why is RStudio IDE greyed out?

Slides at: [bit.ly/EARL2015-Reports](http://bit.ly/EARL2015-Reports)

Today's assistants:

Tareef Kawaf – RStudio

Aimee Gott – Mango Solutions

Steph Locke – Mango Solutions

# Warm Up

TALK. Introduce yourself to the person on your left and the person on your right.

1. Where do you work?
2. What do you do with data?



- 
1. Reproduce reports (R Markdown)
  2. Reproduce computing environment (~~Packrat~~)
  3. Reproduce R expertise (Shiny)

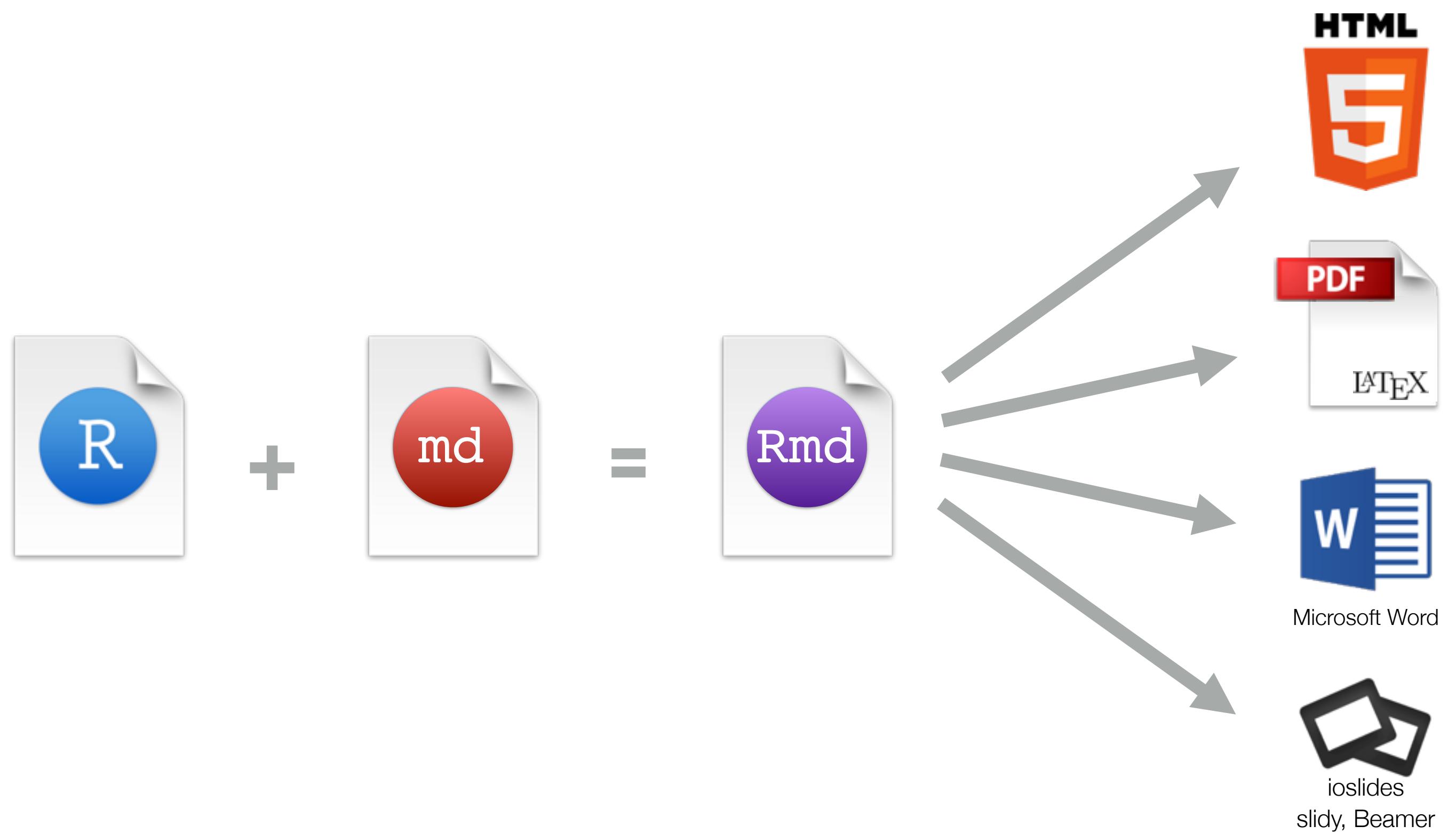
**Reproducible  
reports**

# How can you update results?

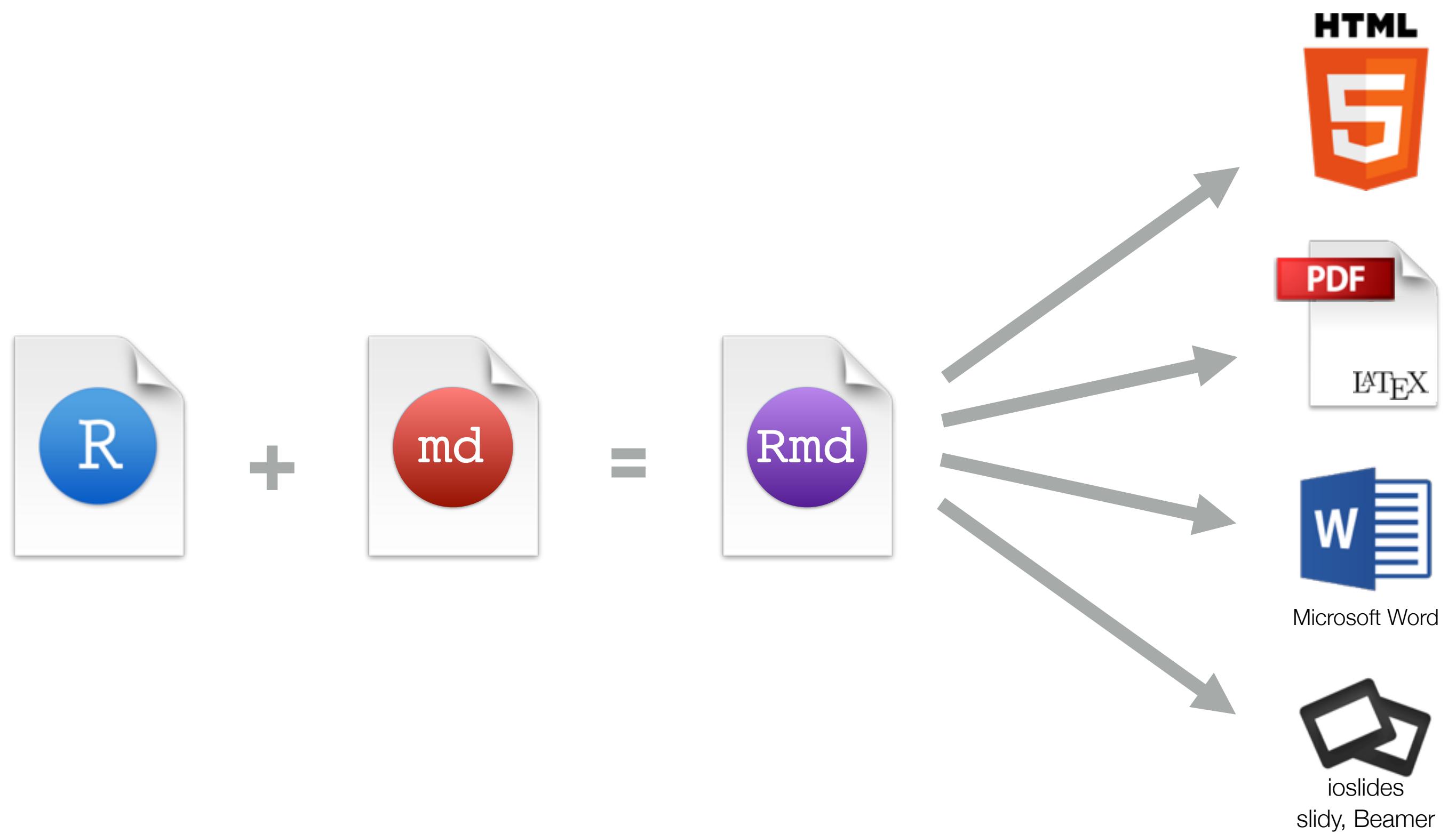
**Ctrl** + **C**      (Copy)

**Ctrl** + **V**      (Paste)

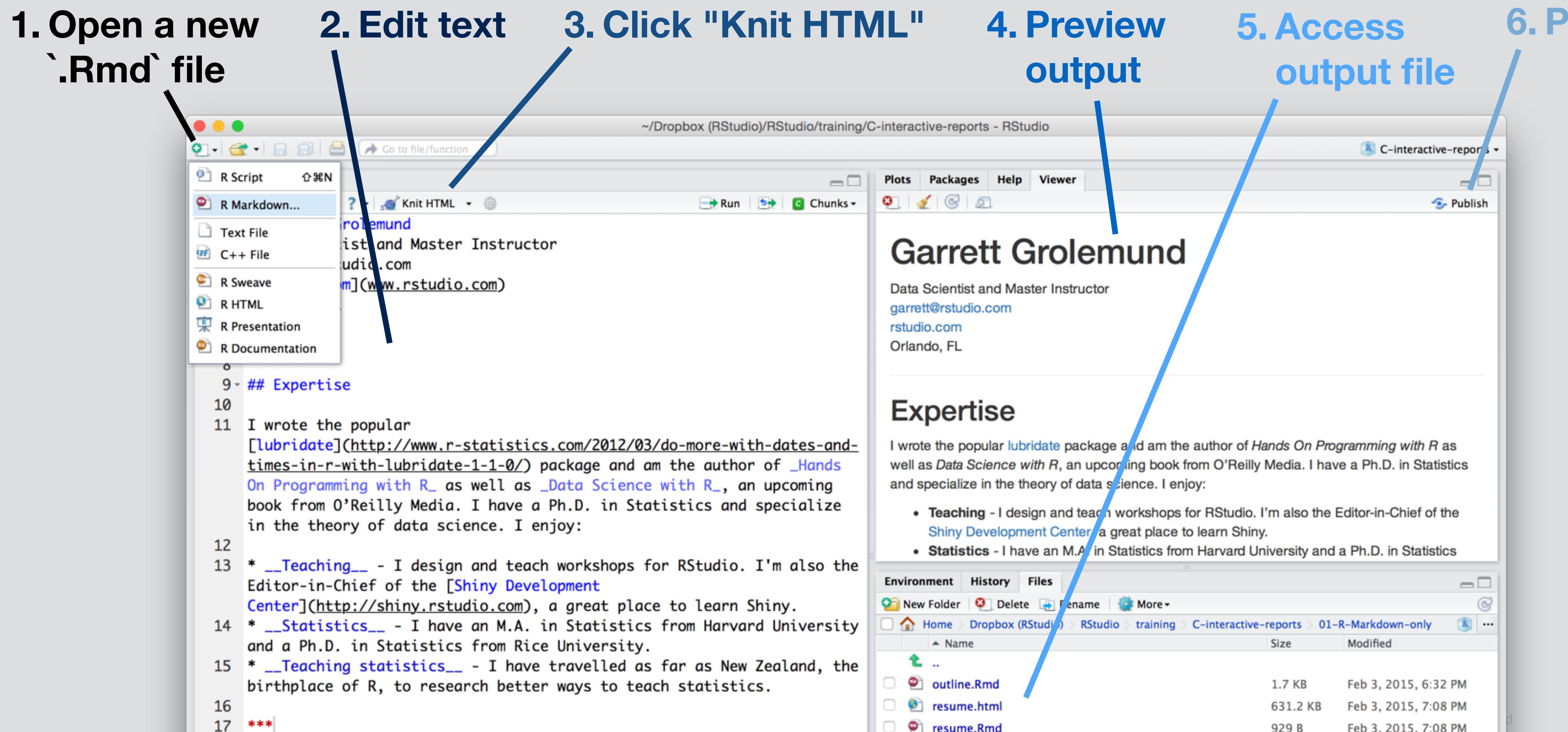
# R Markdown



# R Markdown



# demo



# Markdown

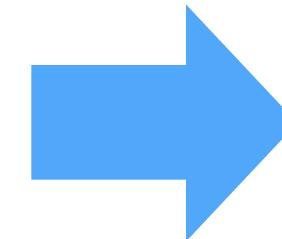
# Markdown

A simple way to write HTML.  
Mostly plain english (like an email)

```
# Say Hello  
To my little  
friend, markdown.
```

Markdown is

- \* easy to use
- \* simple
- \* fun?



## Say Hello

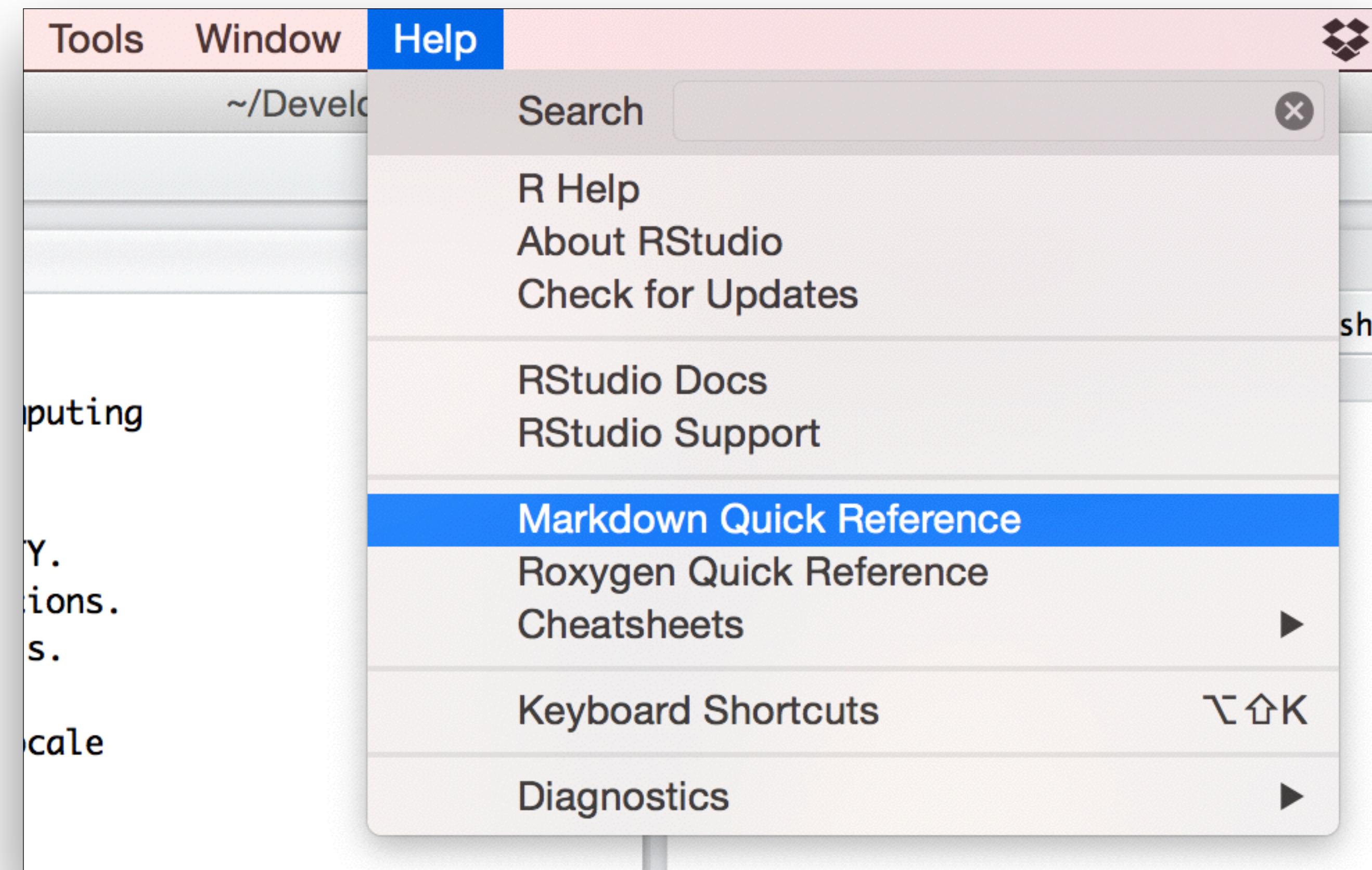
To my little friend, markdown.  
Markdown is

- easy to use
- simple
- fun?

# Web sites that use markdown

- \* **GitHub** [www.github.com](http://www.github.com)
- \* **StackOverflow** [www.stackoverflow.com](http://www.stackoverflow.com)
- \* **Reddit** [www.reddit.com](http://www.reddit.com)
- \* **Meteor** [www.meteor.com](http://www.meteor.com)
- \* many more

# Markdown Quick Reference



# A\* Markdown Cheatsheet

\* *incomplete*

```
library(reportsWS)  
exercise(2)
```

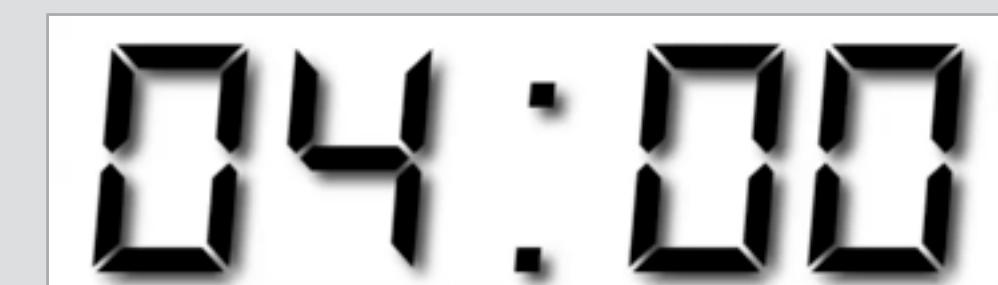
# Your Turn

exercise(2)

Pair up with your neighbor. Choose a writer.

1. Choose a line in the cheatsheet.
2. Rewrite it to display the Markdown formatting that it describes.
3. Re-render the file to see if it worked.

How many can you do?

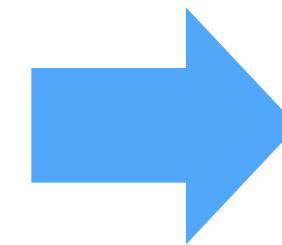


# Headers

Use # to create headers.

Multiple #'s create lower level headers.

```
# Header 1  
## Header 2  
### Header 3  
#### Header 4  
##### Header 5  
##### Header 6
```



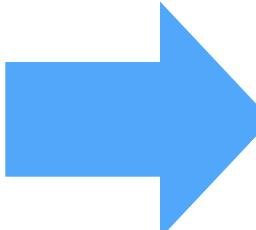
**Header 1**  
**Header 2**  
**Header 3**  
**Header 4**  
**Header 5**  
**Header 6**

# Text

Add two spaces at  
the end of a line to  
start a new line

Text is rendered as plain text. Use underscores (\_) to make italics, two underscores (\_\_) to make bold, back ticks to make code.

Text  
italics  
bold  
`code`



**Text**  
*italics*  
**bold**  
`code`

# Lists

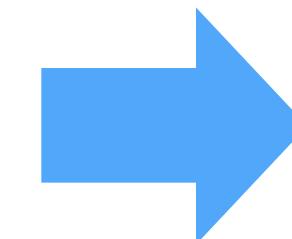
Use asterisks to make bullet points.  
Use numbers to make numbered lists.

## Bullets

- \* bullet 1
- \* bullet 2

## Numbered list

1. item 1
2. item 2



## Bullets

- bullet 1
- bullet 2

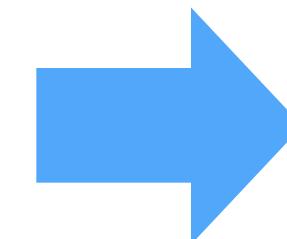
## Numbered list

1. item 1
2. item 2

# Hyperlinks

Use brackets to denote a link. Place the URL in parentheses.

This is a  
[link](http://vw.com).

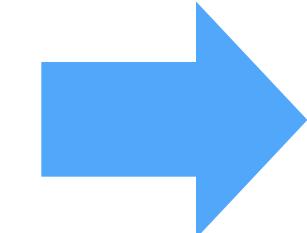


**This is a link.**

# Equations

Write equations with latex math commands and surround them in \$'s.

According to Einstein,  
 $E=mc^2$



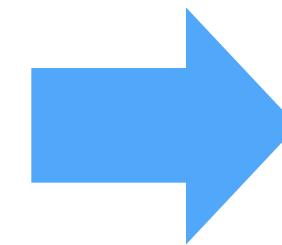
According to Einstein,  $E = mc^2$

# Equation blocks

Use two \$'s to make centered equation blocks.

According to Einstein,

$\$\$E=mc^{\{2\}}\$\$$



According to Einstein,

$$E = mc^2$$

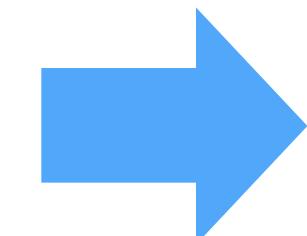
# Images

Use a link preceded by an ! to insert an image.

*The link text should be*

- *a URL (if the image is hosted online)*
- *a file path (if the image is saved as a file)*

  
The RStudio logo.



*Note: You must save your .Rmd file before the preview will find the image*

# R Markdown Reference Guide

[www.rstudio.com/resources/cheatsheets/](http://www.rstudio.com/resources/cheatsheets/)

The screenshot shows the R Markdown Reference Guide website. It features a large R logo on the left, followed by the title "R Markdown Reference Guide". Below the title are two links: "Learn more about R Markdown at [rmarkdown.rstudio.com](#)" and "Learn more about Interactive Docs at [shiny.rstudio.com/articles](#)". To the right is a "Contents" sidebar with three items: "1. Markdown Syntax", "2. Knitr chunk options", and "3. Pandoc options". The main content area is divided into two columns: "Syntax" and "Becomes". The "Syntax" column lists various R Markdown syntax elements, and the "Becomes" column shows how they are rendered. Examples include:

Syntax	Becomes
Plain text	Plain text
End a line with two spaces to start a new paragraph.	End a line with two spaces to start a new paragraph.
*italics* and _italics_	<i>italics</i> and <i>italics</i>
**bold** and __bold__	<b>bold</b> and <b>bold</b>
superscript^2^	superscript <sup>2</sup>
~~strikethrough~~	strikethrough
[link](www.rstudio.com)	<a href="#">link</a>
# Header 1	<h1>Header 1</h1>
## Header 2	<h2>Header 2</h2>
### Header 3	<h3>Header 3</h3>
#### Header 4	<h4>Header 4</h4>
##### Header 5	<h5>Header 5</h5>
###### Header 6	<h6>Header 6</h6>
endash: --	endash: –
emdash: ---	emdash: —
ellipsis: ...	ellipsis: ...
inline equation: \$A = \pi * r^2\$	inline equation: $A = \pi * r^2$
image:	image:
horizontal rule (or slide break):	horizontal rule (or slide break):
***	***

# knitr

# code chunks

# Embed code

Insert a chunk of R code with

```
```{r}  
# some code
```

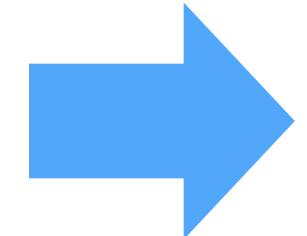
```
```
```

When you compile, R markdown will run the code and include its results. R markdown will also remove the ```{r} and ```.

# inline code

Place code in a sentence with `r #code`. R Markdown will replace the code with its results.

Today is  
`r Sys.Date()`.



Today is 2015-04-16.

You cannot add options  
to inline code

# exercise(19)

```
plot(mtcars$disp, mtcars$mpg)  
mod <- lm(mpg ~ disp, data = mtcars)  
coef(mod)
```

# Your Turn

Write a simple report that embeds the code in  
`exercise(19)` and explains the results of the code.

Check your work by knitting the document.



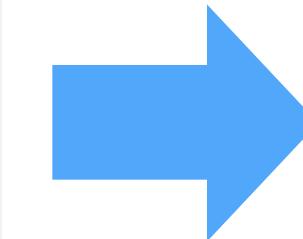
solution(19)

# chunk options

By default, R markdown includes both the code and its results

Here's some code

```
```{r}  
dim(iris)  
```
```



Here's some code

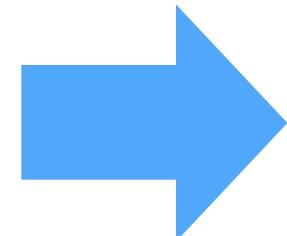
```
dim(iris)
```

```
## [1] 150 5
```

# echo

Add options in the brackets after r.  
**echo = FALSE** hides the code.

```
Here's some code  
```{r echo=FALSE}  
dim(iris)  
```
```

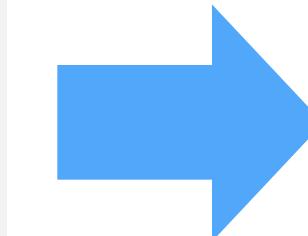


```
Here's some code  
## [1] 150 5
```

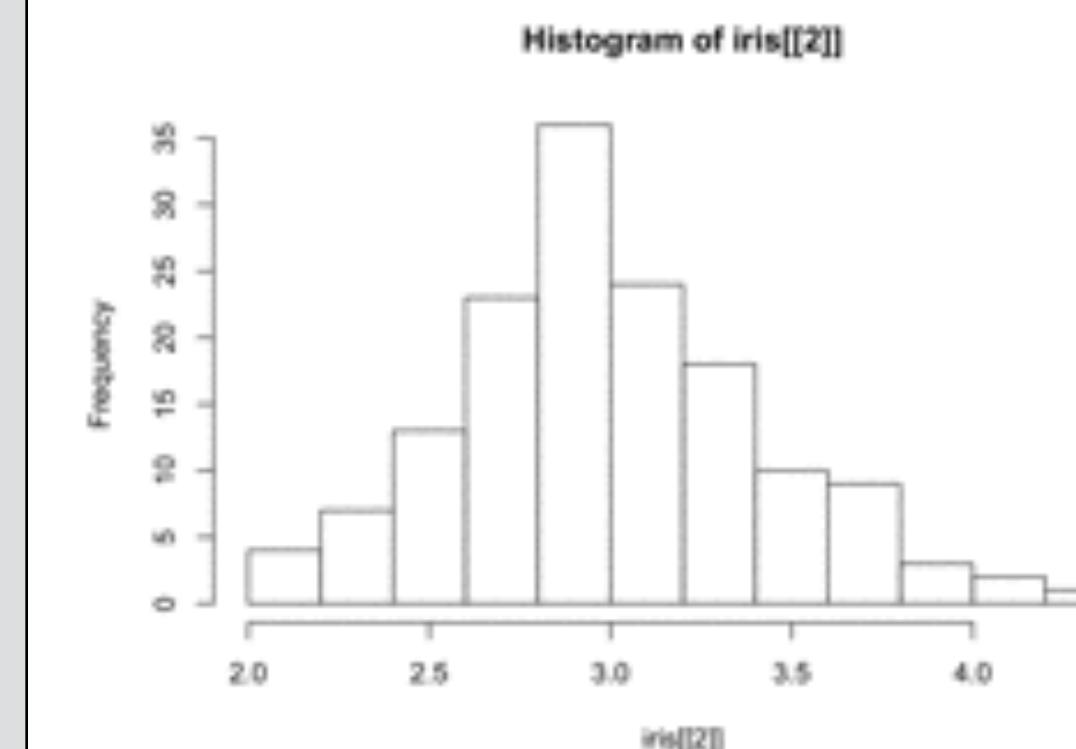
# echo

Add options in the brackets after r.  
**echo = FALSE** hides the code.

```
Here's a plot  
```{r echo=FALSE}  
hist(iris[[2]])  
```
```



Here's a plot

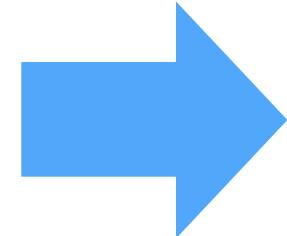


This is very useful  
for plots

# eval

**eval = FALSE** prevents the code from being run.  
As a result, no results will be displayed with the code

Here's some code  
```{r eval=FALSE}  
dim(iris)  
```

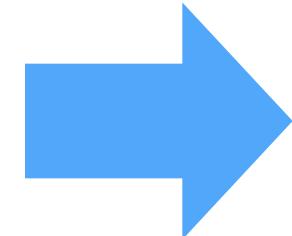


Here's some code  
`dim(iris)`

# results

**results = 'hide'** hides the results of the chunk.  
However, knitr still evaluates the code.

```
Here's some code  
```{r results = 'hide'}  
dim(iris)  
```
```



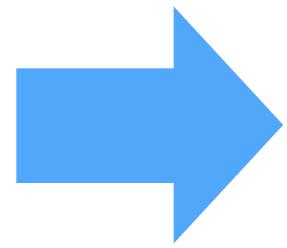
```
Here's some code  
dim(iris)
```

# fig.height, fig.width

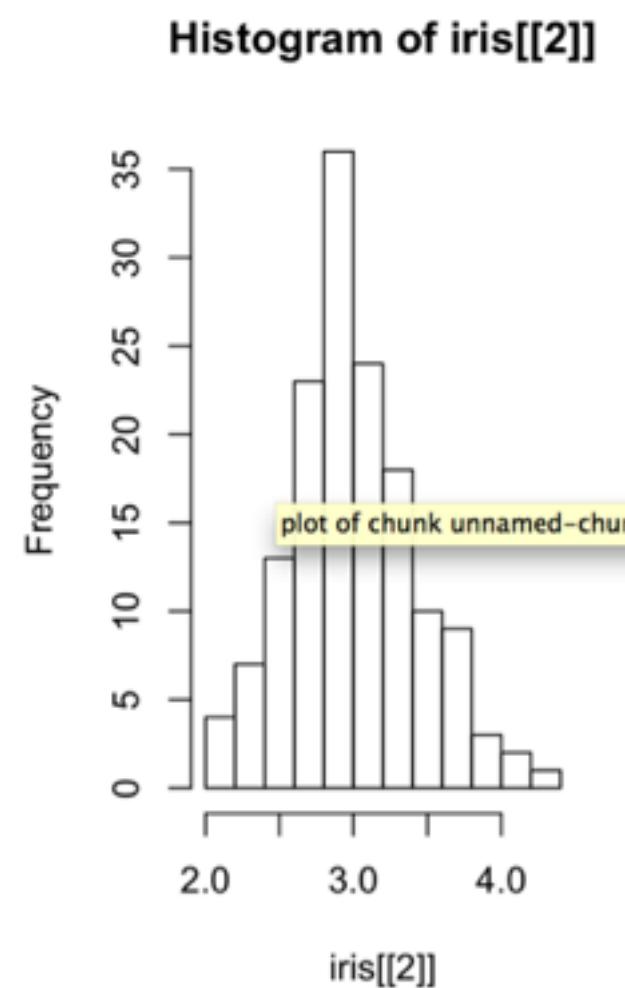
Specify the dimension of plots (in inches) with `fig.width` and `fig.height`.  
Separate multiple arguments with commas.

Here's a plot

```
```{r echo=FALSE, fig.width=3, fig.height=5}
hist(iris[[2]])
```
```



Here's a plot



# Your Turn

Return to your report. Separate this line into its own code chunk:

```
plot(mtcars$disp, mtcars$mpg)
```

1. Set a width and a height for the plot, and
2. prevent the code from appearing above the plot

Check your work by knitting the document.



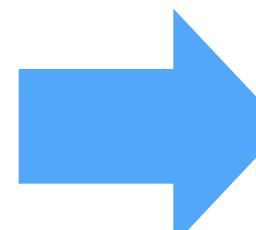
```
```{r echo=FALSE, fig.width=3, fig.height=5}
plot(mtcars$disp, mtcars$mpg)
```
```

# engine

To embed non R code, set the engine option to the language you want to embed.

Some python code,

```
```{r engine='python'}
x = 'hello, python
world!'
print(x)
print(x.split(' '))
```
```



Some python code:

```
x = 'hello, python world!'
print(x)
print(x.split(' '))
```

```
## hello, python world!
## ['hello,', 'python', 'world!']
```

knitr comes with engines for the following languages, and can be extended to other languages

|             |             |               |      |
|-------------|-------------|---------------|------|
| asis        | coffee      | <b>python</b> | sed  |
| asy         | dot         | Rcpp          | sh   |
| awk         | gawk        | Rscript       | tikz |
| <b>bash</b> | haskell     | <b>ruby</b>   | zsh  |
| <b>c</b>    | highlight   | <b>sas</b>    |      |
| cat         | <b>perl</b> | scala         |      |

# Logistics

1

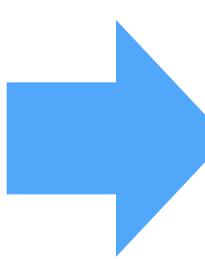
Knitr runs the document in a fresh R session,  
which means you need to load the libraries  
that the document uses *in the document*

# Logistics

1

Knitr runs the document in a fresh R session,  
which means you need to load the libraries  
that the document uses *in the document*

```
```{r}
library(reportsWS)
get_babyname("Ted", "M")[1,3:4]
```
```



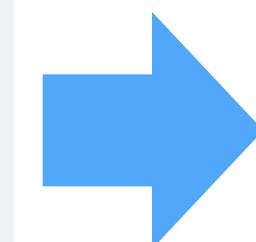
```
##   name  n
## 1 Ted 13
```

# Logistics

1

Knitr runs the document in a fresh R session,  
which means you need to load the libraries  
that the document uses *in the document*

```
```{r}
get_babynames("Ted", "M")[1,3:4]
```
```



Error :  
could not find  
function  
"get\_babynames"

# Logistics

1

Knitr runs the document in a fresh R session,  
which means you need to load the libraries  
that the document uses *in the document*

2

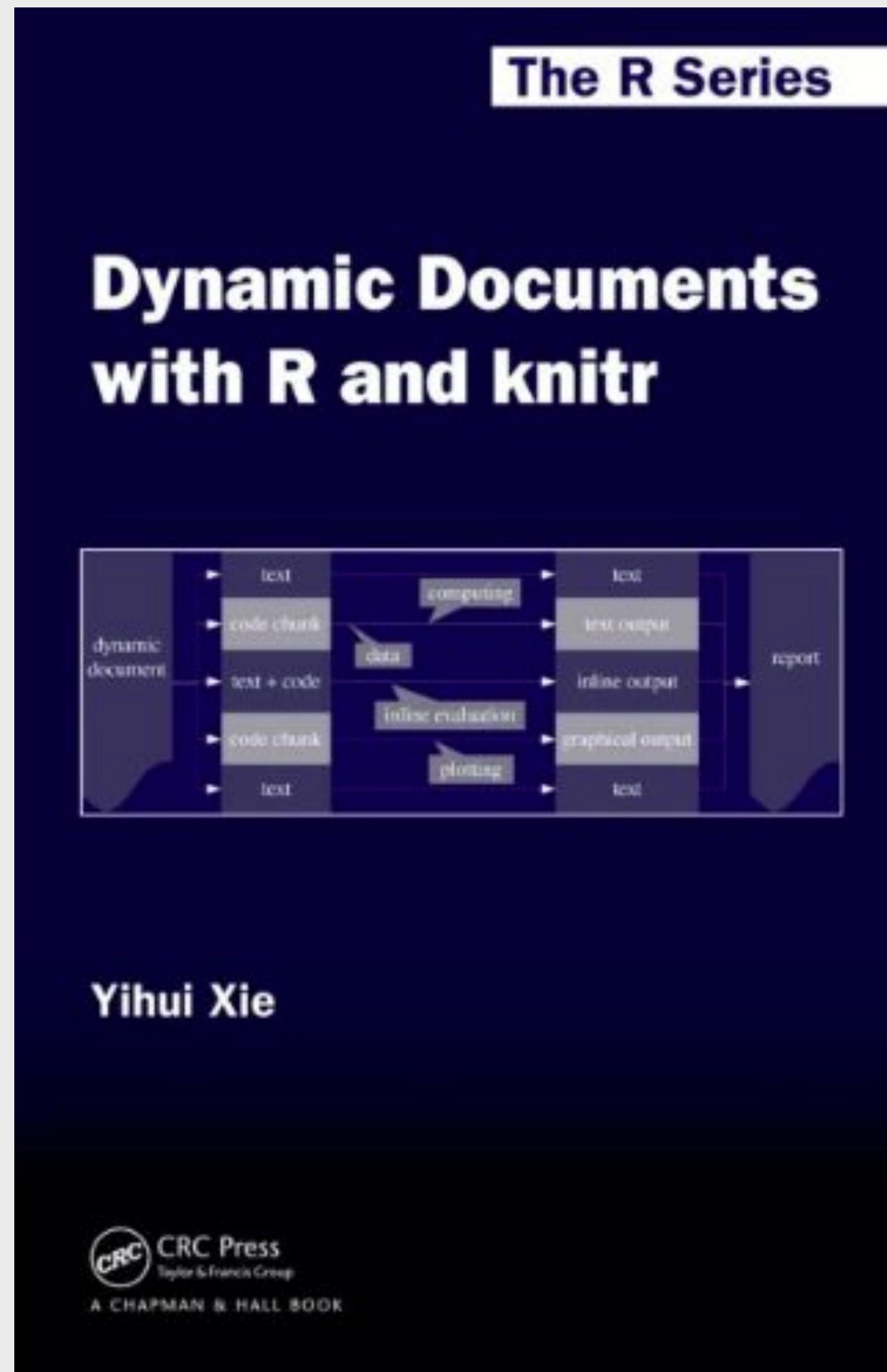
Objects made in one code chunk will be  
available to code in later code chunks.

# R Markdown Reference Guide

[www.rstudio.com/resources/cheatsheets/](http://www.rstudio.com/resources/cheatsheets/)

The screenshot shows the R Markdown Reference Guide page. At the top left is the RStudio logo. To its right is the title "R Markdown Reference Guide". Below the title are two links: "Learn more about R Markdown at [rmarkdown.rstudio.com](#)" and "Learn more about Interactive Docs at [shiny.rstudio.com/articles](#)". To the right of these links is a "Contents" sidebar with three items: "1. Markdown Syntax", "2. Knitr chunk options", and "3. Pandoc options". The main content area is divided into two columns: "Syntax" and "Becomes". Under "Syntax", there are examples for code chunks with three back ticks, inline code with single back ticks, and chunk options within braces. Each example includes R code and its resulting output. Below the examples is a link to "chunk options" at <http://yihui.name/knitr/options>. At the bottom is a "Chunk options" table with two sections: "Code evaluation" and "Results".

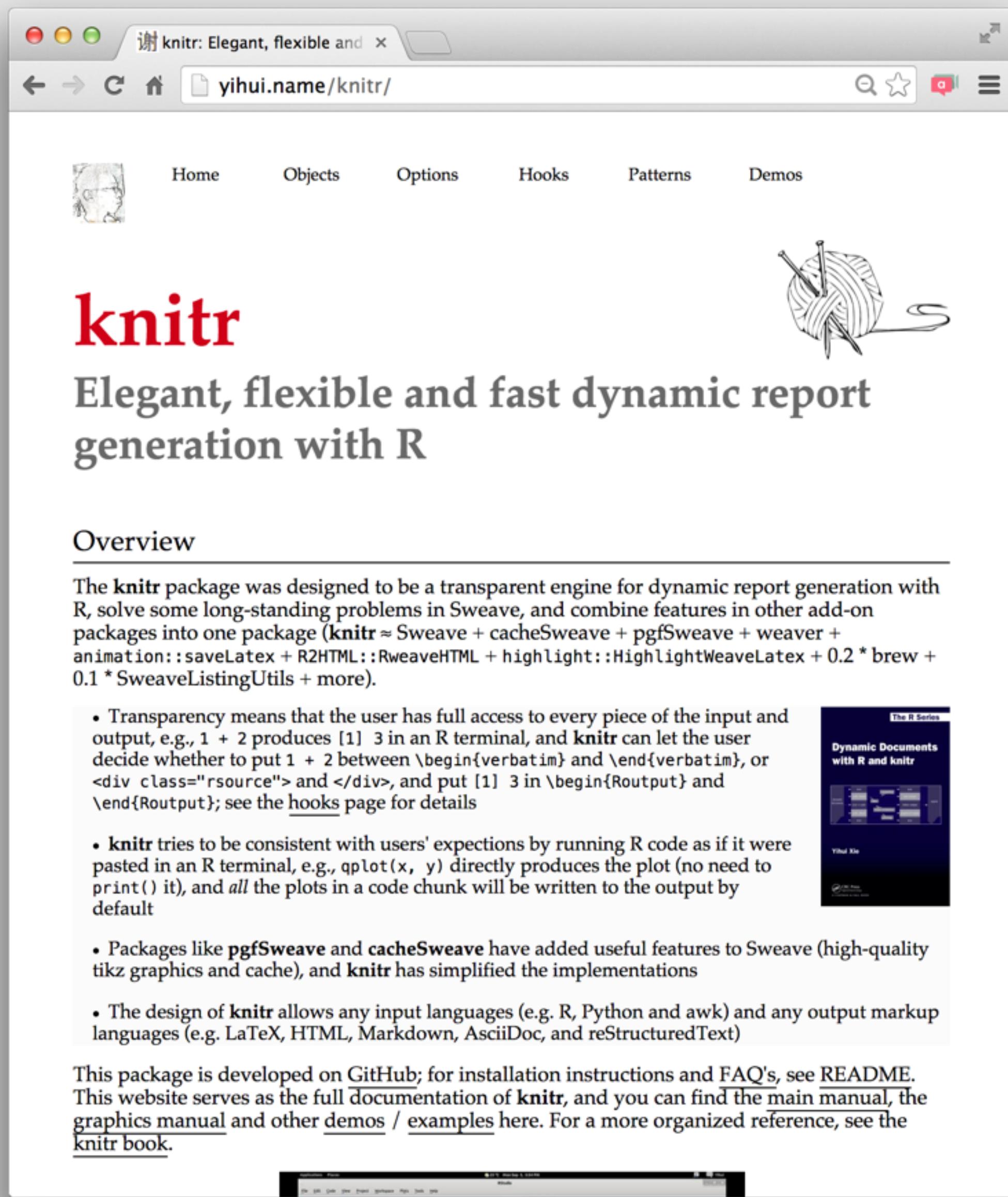
| option                 | default value | description   |
|------------------------|---------------|---|
| <b>Code evaluation</b> |               |   |
| <code>child</code>     | NULL          | A character vector of filenames. Knitr will knit the files and place them into the main document.   |
| <code>code</code>      | NULL          | Set to R code. Knitr will replace the code in the chunk with the code in the code option.   |
| <code>engine</code>    | 'R'           | Knitr will evaluate the chunk in the named language, e.g. <code>engine = 'python'</code> . Run <code>names(knitr::knit_engines\$get())</code> to see supported languages. |
| <code>eval</code>      | TRUE          | If FALSE, knitr will not run the code in the code chunk.  |
| <code>include</code>   | TRUE          | If FALSE, knitr will run the chunk but not include the chunk in the final document.   |
| <code>purl</code>      | TRUE          | If FALSE, knitr will not include the chunk when running <code>purl()</code> to extract the source code.   |
| <b>Results</b>         |               |   |
| <code>collapse</code>  | FALSE         | If TRUE, knitr will collapse all the source and output blocks created by the chunk into a single block.   |
| <code>echo</code>      | TRUE          | If FALSE, knitr will not display the code in the code chunk above it's results in the final document.   |
|                        |               | If ' <code>hide</code> ' knitr will not display the code's results in the final document. If ' <code>hold</code> ', knitr will delay displaying all output                |



## Dynamic Documents with R and knitr by Yihui Xie

Explains knitr and literate programming. Demonstrates how to extend knitr to other languages (e.g., python) and other formats (e.g. latex)

[amzn.com/1482203537](http://amzn.com/1482203537)



The screenshot shows a web browser window with the URL [yihui.name/knitr/](http://yihui.name/knitr/). The page title is "谢 knitr: Elegant, flexible and fast dynamic report generation with R". The main content features a red header "knitr" with a small portrait icon to its left, followed by the subtitle "Elegant, flexible and fast dynamic report generation with R". Below the subtitle is a section titled "Overview". A detailed text block explains the design of the knitr package, mentioning its transparency, consistency with R terminals, and support for various input and output languages. To the right of this text is a small thumbnail image of the book "Dynamic Documents with R and knitr" by Yihui Xie. At the bottom of the page, there is a note about development on GitHub and links to the README, main manual, graphics manual, and demos/examples.

谢 knitr: Elegant, flexible and fast dynamic report generation with R

**knitr**

Elegant, flexible and fast dynamic report generation with R

## Overview

The **knitr** package was designed to be a transparent engine for dynamic report generation with R, solve some long-standing problems in Sweave, and combine features in other add-on packages into one package (**knitr** ≈ Sweave + cacheSweave + pgfSweave + weaver + animation:::saveLatex + R2HTML:::RweaveHTML + highlight:::HighlightWeaveLatex + 0.2 \* brew + 0.1 \* SweaveListingUtils + more).

- Transparency means that the user has full access to every piece of the input and output, e.g., 1 + 2 produces [1] 3 in an R terminal, and **knitr** can let the user decide whether to put 1 + 2 between `\begin{verbatim}` and `\end{verbatim}`, or `<div class="rsource">` and `</div>`, and put [1] 3 in `\begin{Routput}` and `\end{Routput}`; see the [hooks](#) page for details
- knitr** tries to be consistent with users' expectations by running R code as if it were pasted in an R terminal, e.g., `qplot(x, y)` directly produces the plot (no need to `print()` it), and *all* the plots in a code chunk will be written to the output by default
- Packages like **pgfSweave** and **cacheSweave** have added useful features to Sweave (high-quality tikz graphics and cache), and **knitr** has simplified the implementations
- The design of **knitr** allows any input languages (e.g. R, Python and awk) and any output markup languages (e.g. LaTeX, HTML, Markdown, AsciiDoc, and reStructuredText)

This package is developed on [GitHub](#); for installation instructions and FAQ's, see [README](#). This website serves as the full documentation of **knitr**, and you can find the [main manual](#), the [graphics manual](#) and other [demos / examples](#) here. For a more organized reference, see the [knitr book](#).

<http://yihui.name/knitr/>

Demonstrates every knitr chunk option and has a gallery of example knitr projects

# output formats

Untitled

RStudio  
April 16, 2015

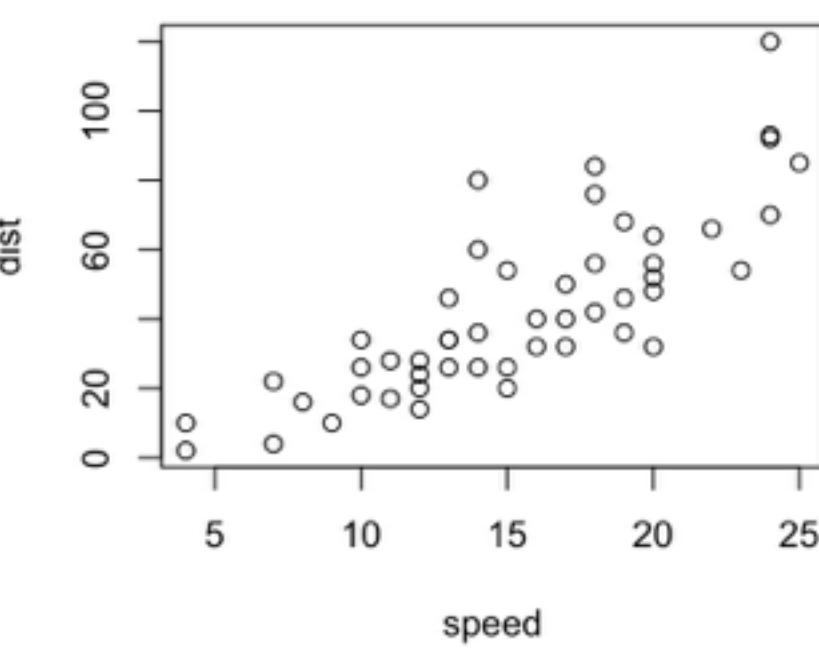
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

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
```

You can also embed plots, for example:



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

# HTML

# PDF

\* preview requires *tex*

Untitled

RStudio  
April 16, 2015

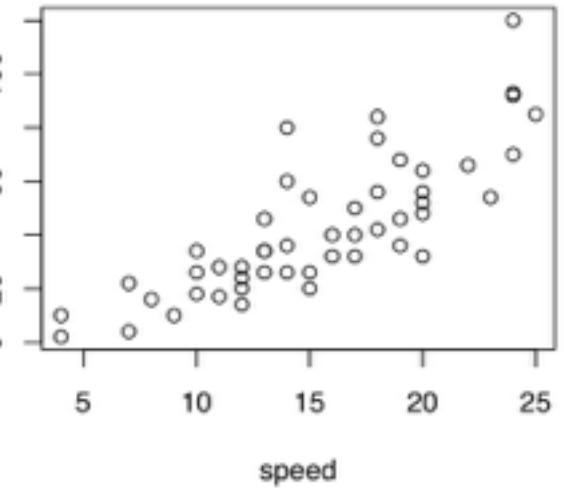
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

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
```

You can also embed plots, for example:



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

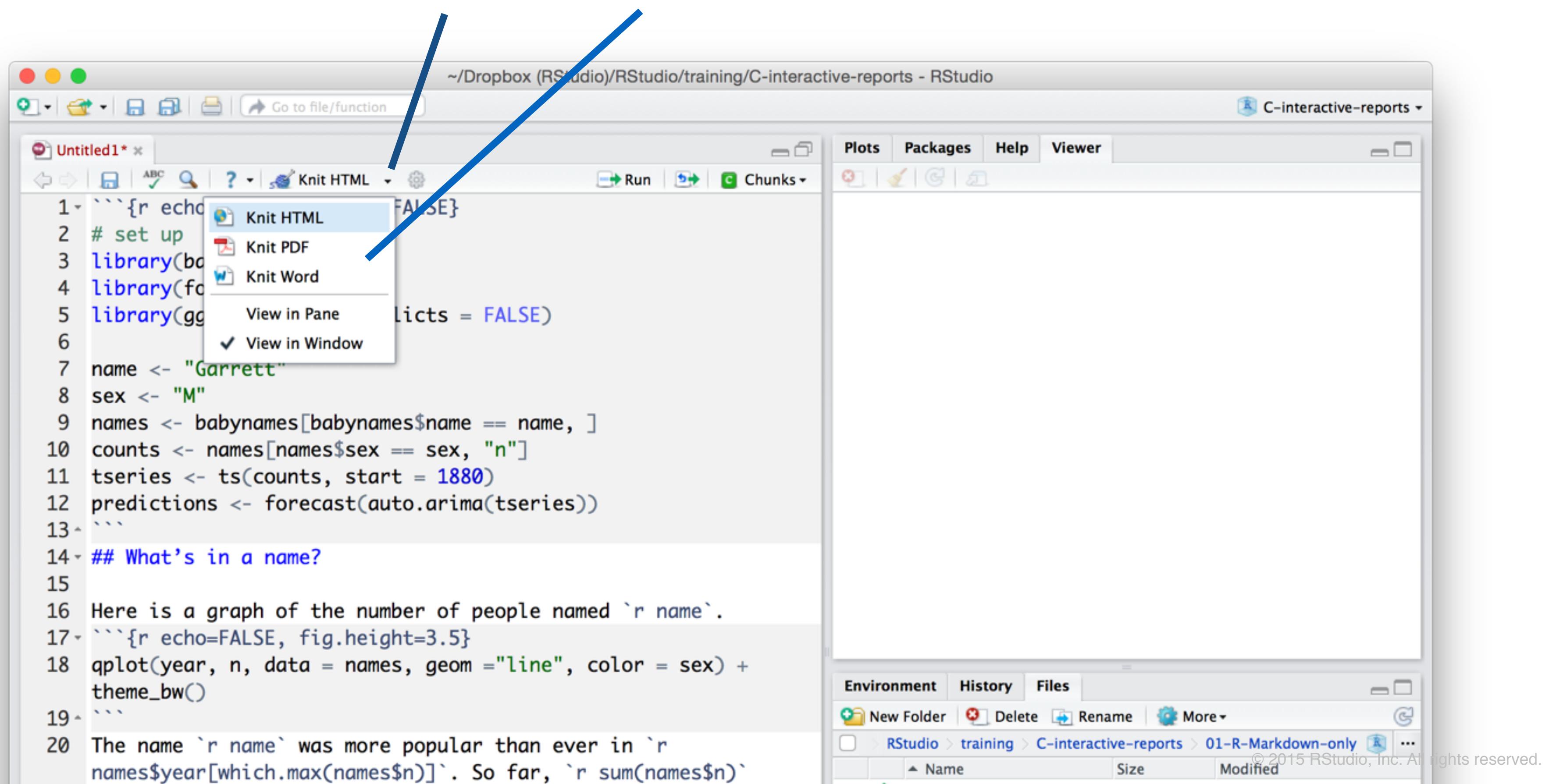
# MS Word

\* preview requires "Word"

# Output

1. Open drop down menu

2. Select format



# YAML

A section of key:value pairs separated by dashed lines — — —

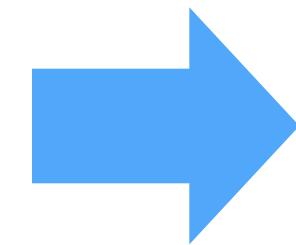
```
---  
title: "Untitled"  
author: "RStudio"  
date: "February 4, 2015"  
output: html_document  
---
```

Text of document

# YAML

A section of key:value pairs separated by dashed lines — — —

**Most important  
option**



```
---  
title: "Untitled"  
author: "RStudio"  
date: "February 4, 2015"  
output: html_document  
---  
Text of document
```

# output templates

|                       |   |
|-----------------------|---|
| html_document         | HTML document                                 |
| pdf_document          | pdf document                                  |
| word_document         | Microsoft .docx file                          |
| md_document           | markdown file (converts R output to markdown) |
| ioslides_presentation | ioslides HTML 5 slideshow                     |
| slidy_presentation    | slide HTML 5 slideshow                        |
| beamer_presentation   | beamer pdf slideshow                          |

# Slide divisions

R Markdown will start a new slide at each first  
(and second\*) level header, and horizontal rule

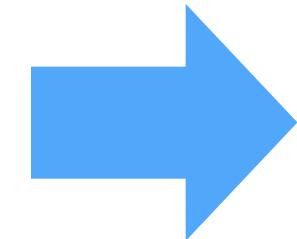
\* *ioslides only*

```
# Slide 1  
text
```

```
## Slide 2  
text
```

\*\*\*

Slide 3



# Slide bullets

Start bullets with -

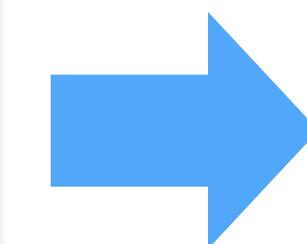
Incremental bullets with >-

```
## Slide 1
```

- Bullet 1
- Bullet 2

```
## Slide 2
```

- >- Bullet A
- >- Bullet B



Slide 1

- Bullet 1
- Bullet 2

Slide 2

# rmarkdown::render

Render at the command line with YAML options

```
> render("doc.Rmd")
```

Render at the command line, override output format.

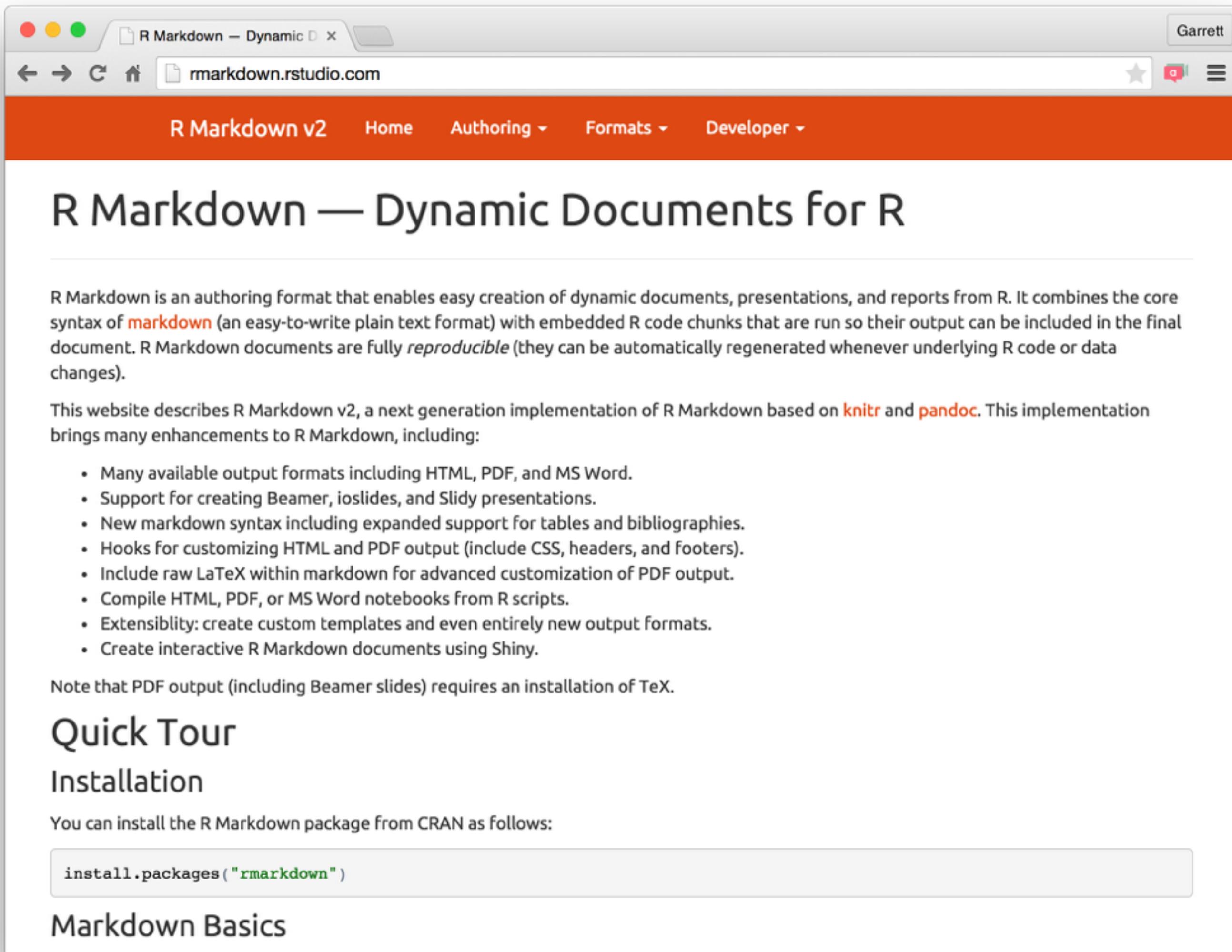
```
> render("doc.Rmd", "html_document")
```

Render at the command line to multiple formats.

```
> render("doc.Rmd", c("html_document", "pdf_document"))
```

# The R Markdown Development Center

## [rmarkdown.rstudio.com](http://rmarkdown.rstudio.com)



The screenshot shows a web browser window displaying the R Markdown v2 website. The title bar reads "R Markdown — Dynamic D X" and the address bar shows "rmarkdown.rstudio.com". The page has a red header with navigation links for "R Markdown v2", "Home", "Authoring", "Formats", and "Developer". The main content area features a large heading "R Markdown — Dynamic Documents for R". Below it is a paragraph about R Markdown's purpose and benefits. A bulleted list details the enhancements in R Markdown v2. A note about TeX installation is present, followed by sections for "Quick Tour" and "Installation". A code snippet for installing the package is shown, along with a "Markdown Basics" section.

R Markdown is an authoring format that enables easy creation of dynamic documents, presentations, and reports from R. It combines the core syntax of [markdown](#) (an easy-to-write plain text format) with embedded R code chunks that are run so their output can be included in the final document. R Markdown documents are fully *reproducible* (they can be automatically regenerated whenever underlying R code or data changes).

This website describes R Markdown v2, a next generation implementation of R Markdown based on [knitr](#) and [pandoc](#). This implementation brings many enhancements to R Markdown, including:

- Many available output formats including HTML, PDF, and MS Word.
- Support for creating Beamer, ioslides, and Slidy presentations.
- New markdown syntax including expanded support for tables and bibliographies.
- Hooks for customizing HTML and PDF output (include CSS, headers, and footers).
- Include raw LaTeX within markdown for advanced customization of PDF output.
- Compile HTML, PDF, or MS Word notebooks from R scripts.
- Extensibility: create custom templates and even entirely new output formats.
- Create interactive R Markdown documents using Shiny.

Note that PDF output (including Beamer slides) requires an installation of TeX.

## Quick Tour

## Installation

You can install the R Markdown package from CRAN as follows:

```
install.packages("rmarkdown")
```

## Markdown Basics