

# Strata+ Hadoop

---

WORLD

PRES EN TED BY

O'REILLY®

cloudera®

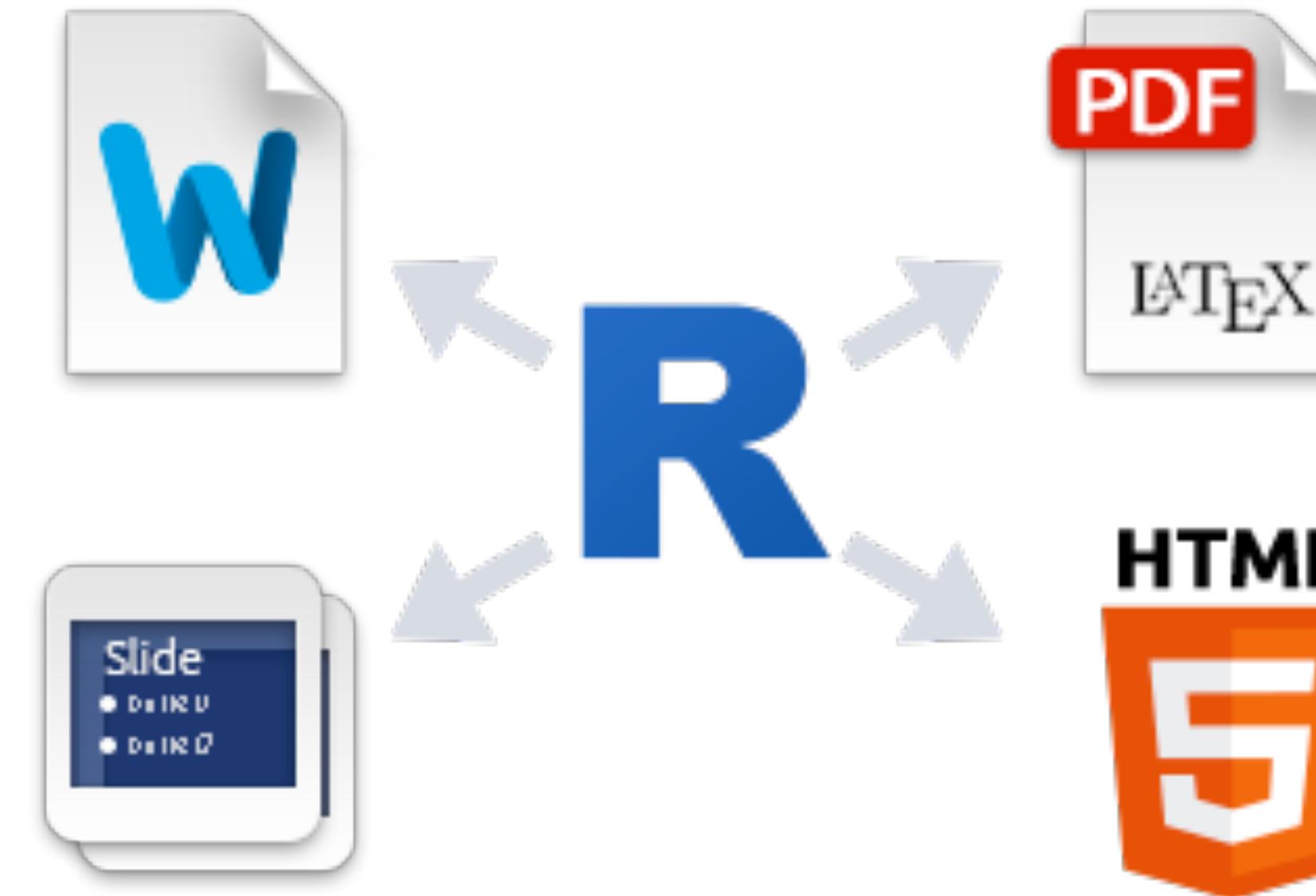


[strataconf.com](http://strataconf.com)

#StrataHadoop

# Reproducible Reports with Big Data

One format to rule them all



Garrett Grolemund

Professional Educator

March 2016

Email: [garrett@rstudio.com](mailto:garrett@rstudio.com)

Stock symbol: FB

Recommendation: BUY

FB will increase in price during the next trading period.

#### Price History

The price of FB has increased in recent years.

The chart below is made with the quantmod R package, a widely used package for collecting and visualizing financial data in R. You can learn more about quantmod at the website [www.quantmod.com](http://www.quantmod.com).



#### Method

This forecast was predicted with the *recency algorithm*, a simple---probably useless---method for determining stock prices. The recency algorithm predicts that the next price movement,  $M_j$ , will be in the same direction as the most recent price movement,  $M_i$ , where  $M_i = \text{Close}_i - \text{Open}_i$ .

#### Raw Data

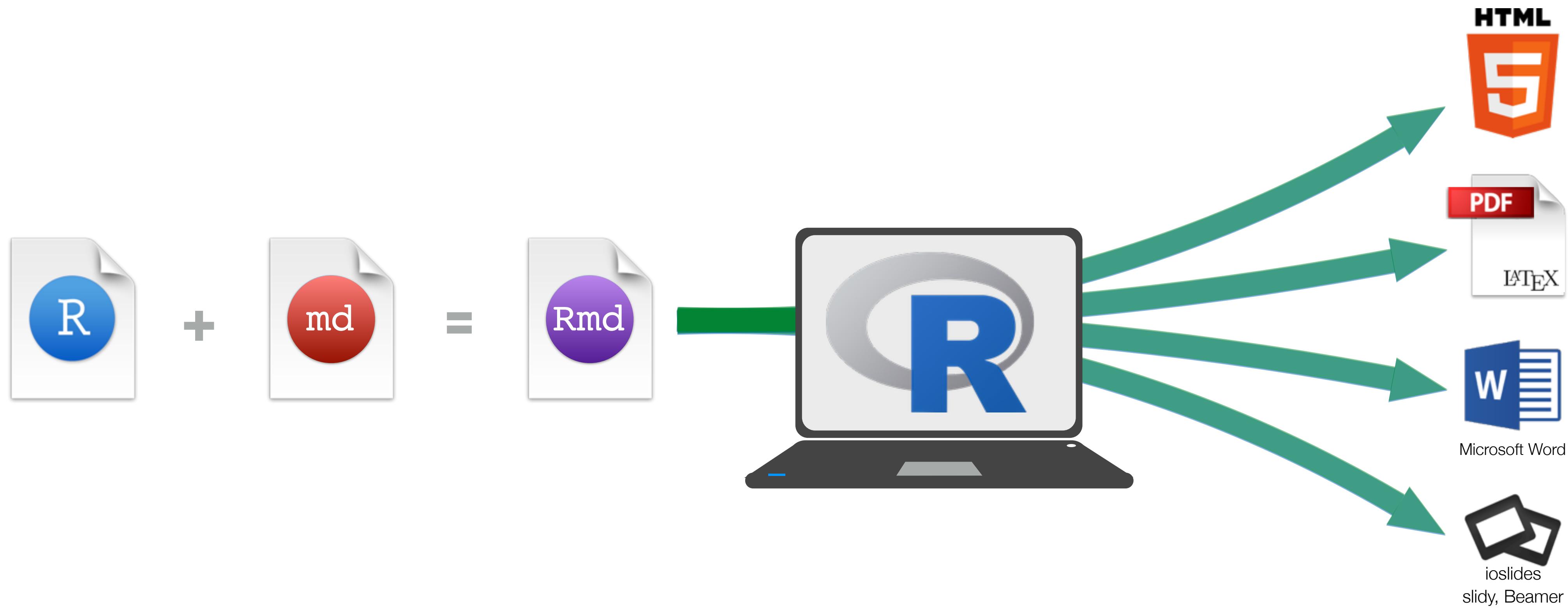
The table below displays the daily price data for FB.

	FB.Open	FB.High	FB.Low	FB.Close
2012-05-18	42.05	45.00	38.00	38.23
2012-05-21	36.53	36.66	33.00	34.03
2012-05-22	32.61	33.59	30.94	31.00
2012-05-23	31.37	32.50	31.36	32.00
2012-05-24	32.95	33.21	31.77	33.03

# How can you update the results?

**Ctrl + C (Copy)**  
**Ctrl + V (Paste)**

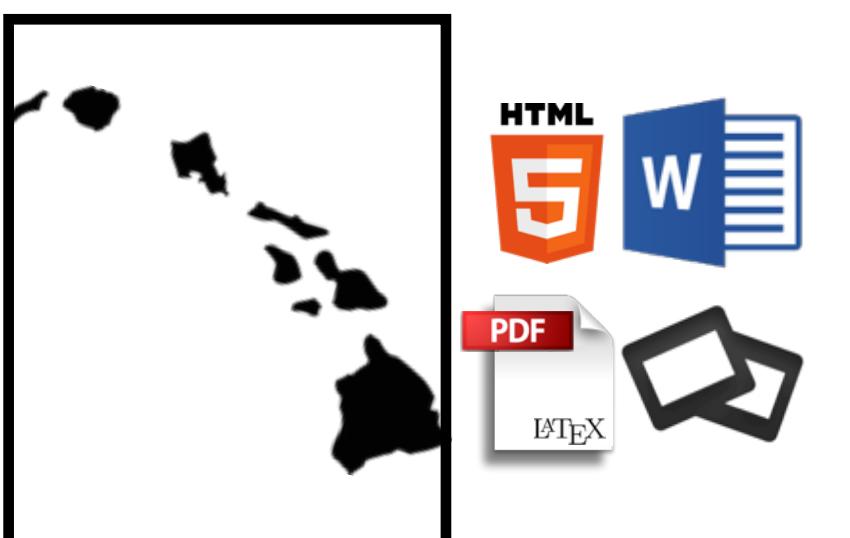
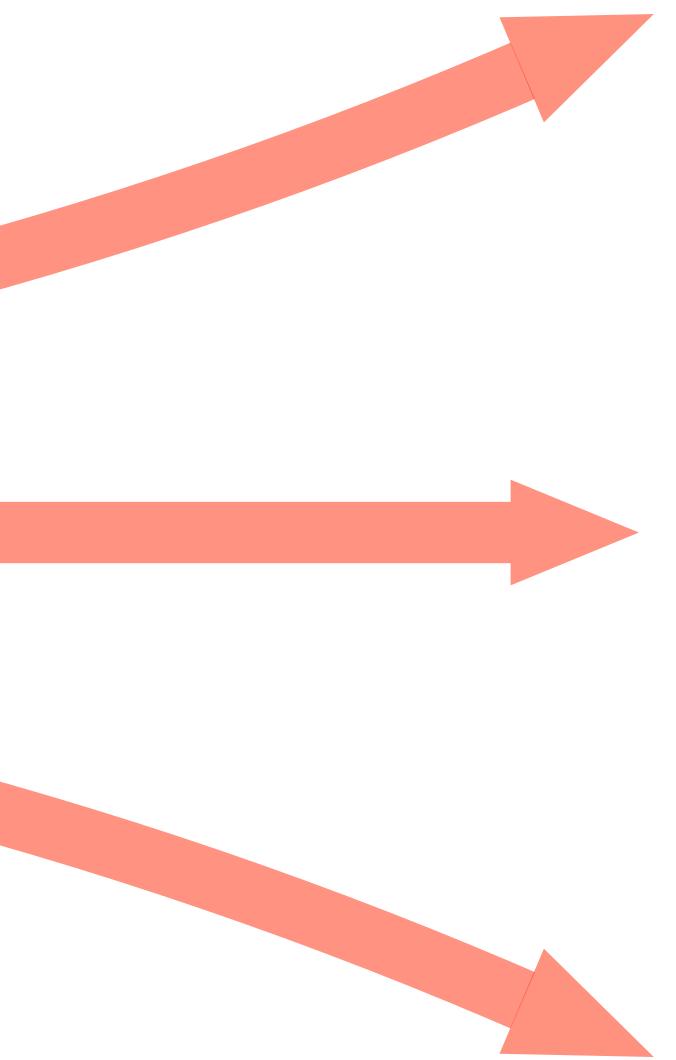
# R Markdown



# R Markdown

## Parameters

- + State = Florida
- + State = Georgia
- + State = Hawaii



**HELLO**

my name is

**Garrett**



@StatGarrett



# Outline

- 1. Write in Markdown**
- 2. Embed code with knitr**
- 3. Parameterize your report**
- 4. Choose output format**

# Warm Up

Open the **RMarkdownReports** project. Then open the **demos/01-flights.R** script in your working directory. Run the code and determine what it does.

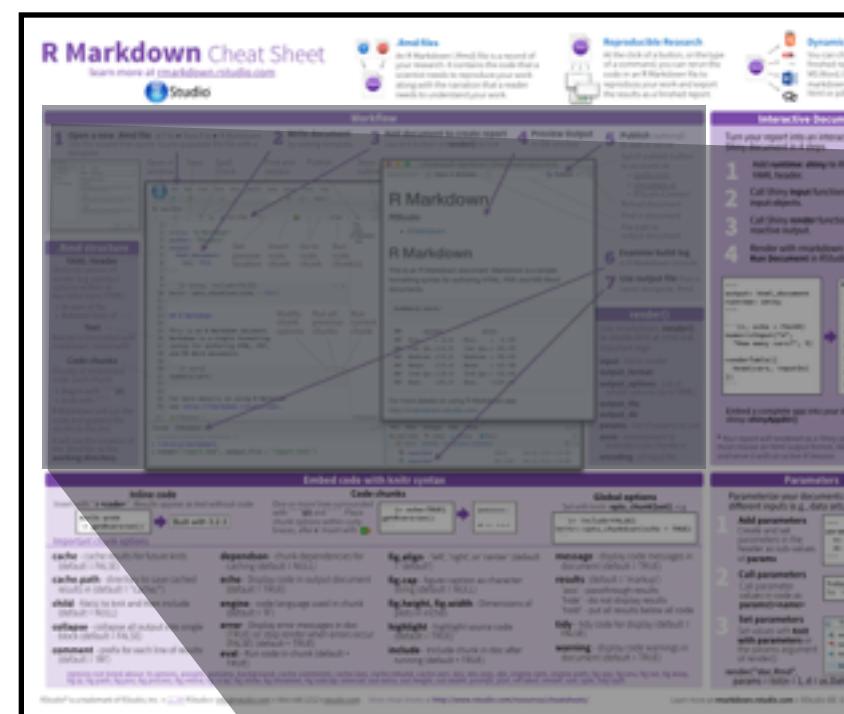


# Orientation

# R Markdown Cheatsheet

Help ► Cheatsheets ► R Markdown Cheatsheet

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



**Workflow**

- 1 Open a new .Rmd file** at File ▶ New File ▶ R Markdown. Use the wizard that opens to pre-populate the file with a template
- 2 Write document** by editing template
- 3 Knit document to create report** Use knit button or `render()` to knit
- 4 Preview Output** in IDE window
- 5 Publish** (optional) to web or server  
Sync publish button to accounts at • [rpubs.com](#), • [shinyapps.io](#), RStudio Connect  
Reload document  
Find in document  
File path to output document
- 6 Examine build log** in R Markdown console
- 7 Use output file** that is saved alongside .Rmd

**.Rmd structure**

**YAML Header**  
Optional section of `render` (e.g. `pandoc`) options written as key:value pairs (YAML).  
• At start of file  
• Between lines of `---`

**Text**  
Narration formatted with markdown, mixed with:

**Code chunks**  
Chunks of embedded code. Each chunk:  
• Begins with ````{r}`  
• ends with `````  
R Markdown will run the code and append the results to the doc.  
It will use the location of the .Rmd file as the **working directory**

**Code Editor**

```

1 ---
2 title: "R Markdown"
3 author: "RStudio"
4 output:
5   html_document:
6     toc: TRUE
7 ---
8
9 `r setup, include=FALSE`
10 knitr::opts_chunk$set(echo = TRUE)
11
12
13 ## R Markdown
14
15 This is an R Markdown document.
16 Markdown is a simple formatting
17 syntax for authoring HTML, PDF,
18 and MS Word documents.
19
20 `r cars`
21 summary(cars)
22
23
24 For more details on using R Markdown
25 see <http://rmarkdown.rstudio.com>.

```

**Console**

```

> library(rmarkdown)
> render("report.Rmd", output_file = "report.html")

```

**Output Preview**

**R Markdown**

**RStudio**

- R Markdown

**R Markdown**

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents.

```

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

For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

**Output Browser**

**Report.html**

	File	Plots	Packages	Help	Viewer
New Folder	Delete	Rename	More		
Home	Desktop	R-Markdown-Cheatsheet			
report.Rmd					398 B Feb 26, 2016, 3:36 PM
report.html					581.3 KB Feb 26, 2016, 3:36 PM

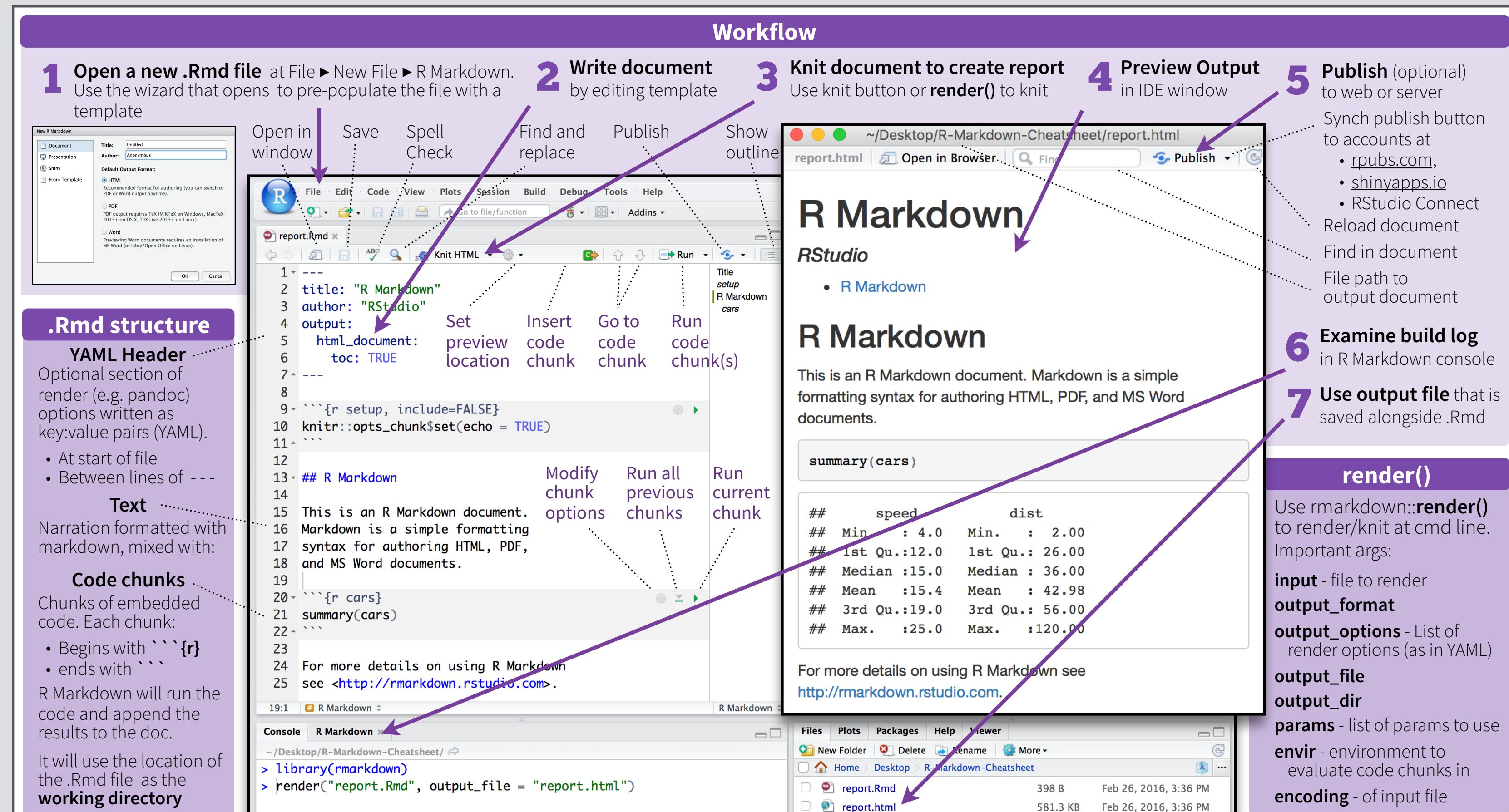
**render()**

Use `rmarkdown::render()` to render/knit at cmd line. Important args:

- input** - file to render
- output\_format**
- output\_options** - List of render options (as in YAML)
- output\_file**
- output\_dir**
- params** - list of params to use
- envir** - environment to evaluate code chunks in
- encoding** - of input file

# Your turn

Launch RStudio, and open your first R Markdown script.  
Make sure you can knit the script into finished HTML output.



# Markdown

## Web sites that use markdown

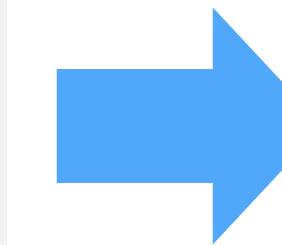
- \* **GitHub** [www.github.com](http://www.github.com)
- \* **StackOverflow** [www.stackoverflow.com](http://www.stackoverflow.com)
- \* **Reddit** [www.reddit.com](http://www.reddit.com)
- \* **Meteor** [www.meteor.com](http://www.meteor.com)
- \* many more

# Headers

Use # to create headers.

Multiple #'s create lower level headers.

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



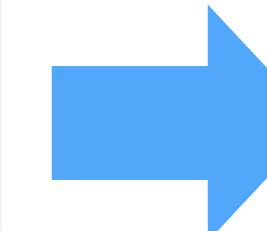
<b>Header 1</b>
<b>Header 2</b>
<b>Header 3</b>
<b>Header 4</b>
<b>Header 5</b>
<b>Header 6</b>

# 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

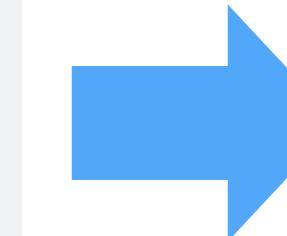
Use asterisks to make bullet points.  
Use numbers to make numbered lists.

## Bullets

- \* bullet 1
- \* bullet 2

## Numbered list

1. item 1
2. item 2



## Bullets

- bullet 1
- bullet 2

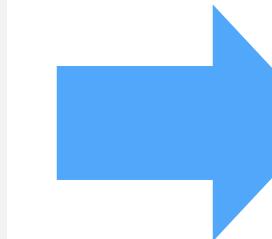
## Numbered list

1. item 1
2. item 2

# Hyperlinks

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

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

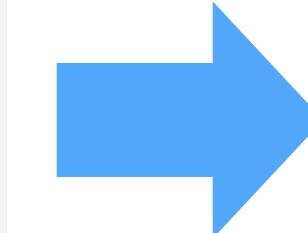


**This is a link.**

# Equations

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

According to Einstein,  
 $E=mc^2$



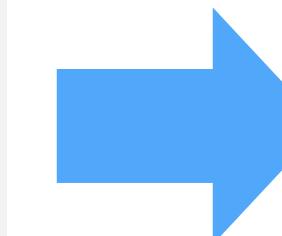
According to Einstein,  $E = mc^2$

# Equation blocks

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

According to Einstein,

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



According to Einstein,

$E = mc^2$

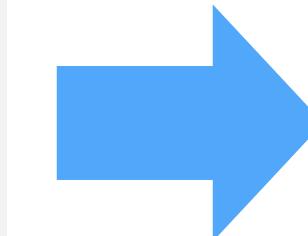
# Images

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*

# Your Turn

Open demos/02-text.Rmd. Read the report, then use R Markdown to format the report with

- headers
- three bullet points
- italicized text
- bold text
- a web link, etc.



# knitr

# code chunks

# Quiz

How should we put our graph in the report?

# Embed code

Insert a chunk of R code with

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

```
```
```

When you render the report, R Markdown will run the code and include its results. R Markdown will also remove the ````{r}` and `````.

# chunk options

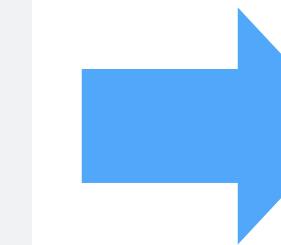
By default, R markdown includes both the code and its results

Here's some code

```
```{r}
```

```
dim(iris)
```

```
```
```



Here's some code

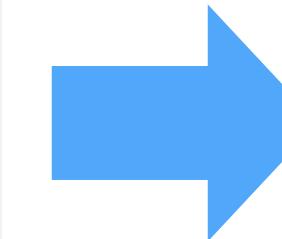
```
dim(iris)
```

```
## [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)  
```
```

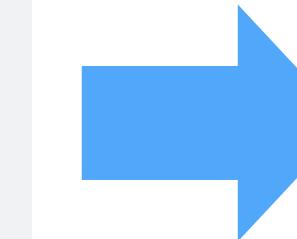


```
Here's some code  
## [1] 150 5
```

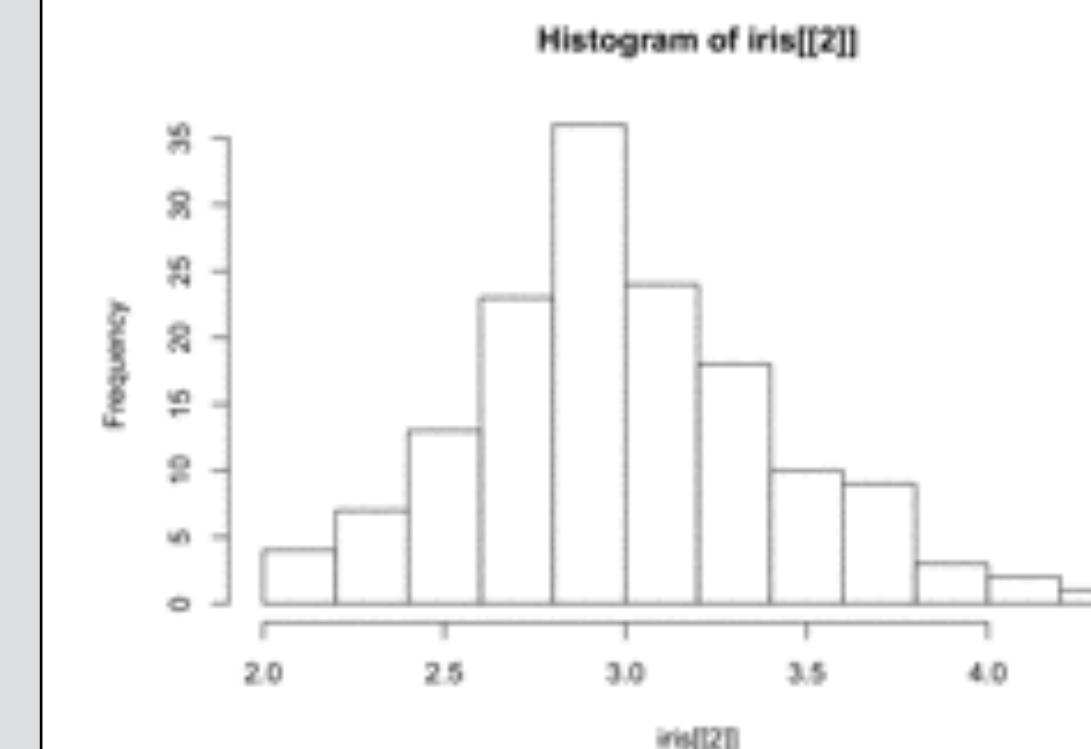
# echo

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

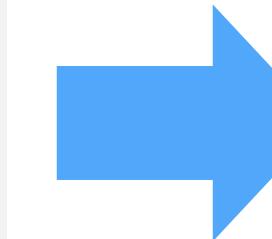


This is very useful  
for plots

# 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)  
```



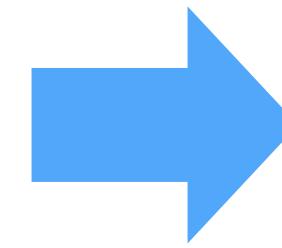
Here's some code  
`dim(iris)`

# fig.height, fig.width

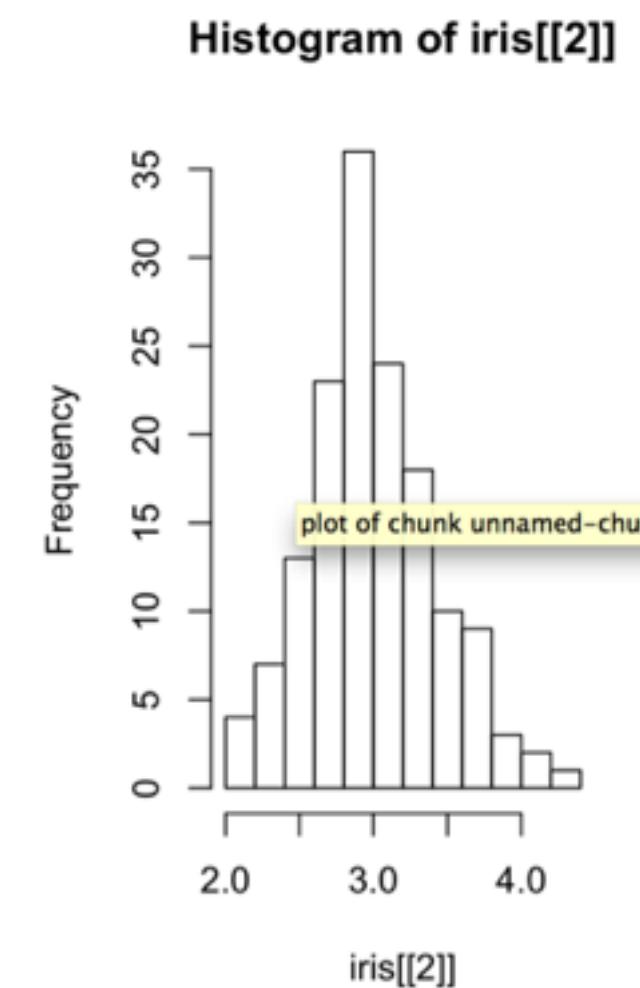
Specify the dimension of plots (in inches) with `fig.width` and `fig.height`. Separate multiple arguments with commas.

Here's a plot

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



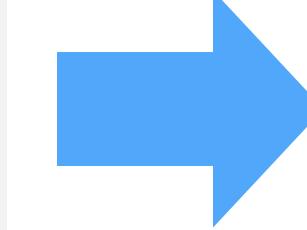
Here's a plot



# message

**message = FALSE** suppresses messages from appearing in the output.

```
Here's some code  
```{r message=FALSE}  
library(forecast)  
```
```



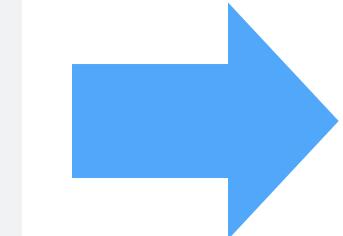
```
Here's some code  
library(forecast)
```

# warning

**warning = FALSE** supresses warnings from appearing in the output.

```
Here's some code  
```{r warning=FALSE}  
warning("Don't!")  
```
```

```
Here's some code  
warning("Don't!")
```



# Logistics

1

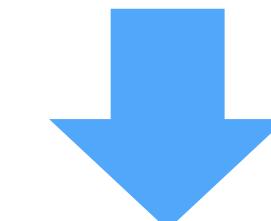
Knitr runs the document in a fresh R session,  
which means you need to load the libraries  
that the document uses *in the document*

# 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}
delays <- flights
...``
```



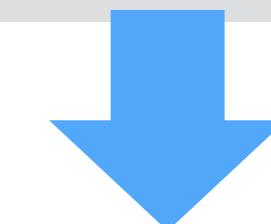
Error : object 'flights' not found

# 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(nycflights13)
delays <- flights
```
```



No Error (flights is loaded with nycflights13)

# Logistics

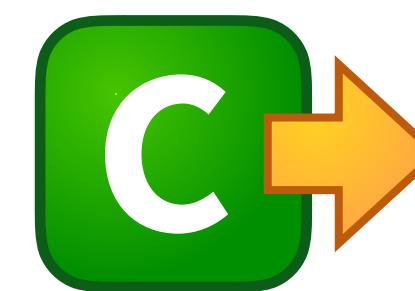
1

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

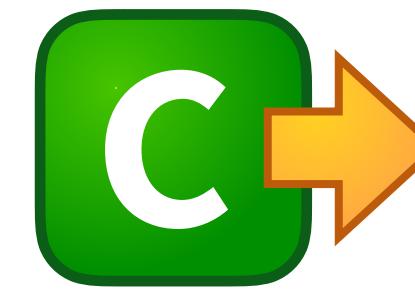
2

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

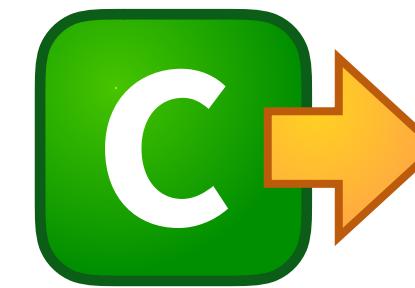
# Recap



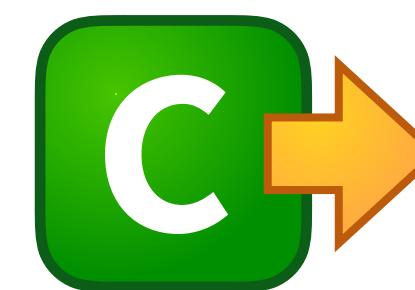
Place a **code chunk** where you would like code results to appear in your document



Begin a chunk with ````{r}`



End a chunk with `````



Add **named arguments** inside braces to customize execution of the code / display of results

# Your Turn

Add the code from flights.R to your R Markdown file.

Arrange for the table and graph to appear in your report.

Prevent the library() and ggplot2 commands (and messages, and warnings) from appearing in the finished report.

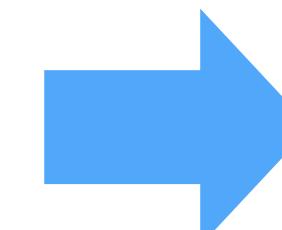


# Advanced knitr

# 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:

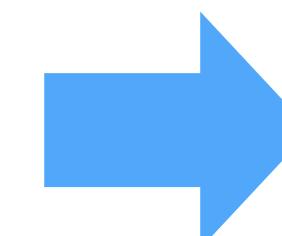
```
x = 'hello, python world!'  
print(x)  
print(x.split(' '))
```

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

# engine

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

```
Some python code,  
```{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

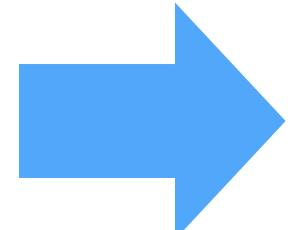
|             |             |               |      |
|-------------|-------------|---------------|------|
| asis        | coffee      | <b>python</b> | sed  |
| asy         | dot         | Rcpp          | sh   |
| awk         | gawk        | Rscript       | tikz |
| <b>bash</b> | haskell     | <b>ruby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

# cache

R Markdown will cache the result of the code chunk to reuse (and thus avoid computation) when **cache = TRUE**

Here's some code  
that takes a "long"  
time to run.

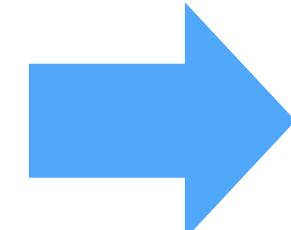
```
```{r cache=TRUE}  
Sys.sleep(5)  
rnorm(1)  
```
```



Here's some code that  
takes a "long" time to run.

```
Sys.sleep(5)  
rnorm(1)
```

```
## [1] 1.582407
```



Here's some code that  
takes a "long" time to run.

```
Sys.sleep(5)  
rnorm(1)
```

```
## [1] 1.582407
```

# dependson

What if the cached code depends on another chunk that may have updated?

```
```{r c1, cache=TRUE}
m <- 5
```
```
```{r c2, cache=TRUE}
n <- 1
```
```
```{r cache=TRUE, dependson=c("c1", "c2")}
Sys.sleep(m)
rnorm(n)
```
```

```

Give your chunks labels.  
Then set dependson.

First, unnamed argument becomes the chunk's label

This chunk depends on the chunks labelled c1 and c2

# global options

Use **knitr::opts\_chunk\$set()** to set global options (options that apply to ALL chunks)

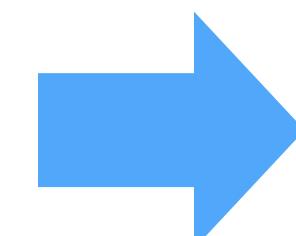
```
```{r echo = FALSE}  
knitr::opts_chunk$set(eval = FALSE)  
```
```

Will this evaluate?

```
```{r}  
1 + 1  
```
```

Will this?

```
```{r}  
1 + 2  
```
```



Will this evaluate?

```
1 + 1
```

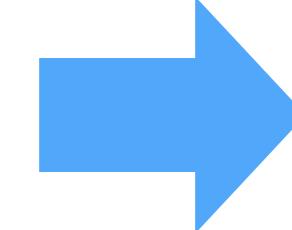
Will this?

```
1 + 2
```

# 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

# Your Turn

Use inline R code to programmatically generate the following whenever they appear in your report (use the code in flights.R):

1. The Day of the Week
2. The airport with the shortest average delay
3. The length of the average delay

*By embedding the code that returns the name (instead of typing the name) you expand the ways you can reuse the report.*



# tables

```
## Source: local data frame [3 x 2]

##          origin mean_delay
##          (chr)      (dbl)
## 1        EWR    7.680089
## 2        JFK    9.957611
## 3        LGA    4.226548
```

# knitr::kable

```
```{r results = 'asis'}  
knitr::kable(weekday)  
```
```

Creates table for all output formats

Use with `results = 'asis'`

`kable()` comes in the `knitr` package

| origin | mean_delay |
|--------|------------|
| EWR    | 7.680089   |
| JFK    | 9.957610   |
| LGA    | 4.226548   |

| origin | mean_delay |
|--------|------------|
| EWR    | 7.680089   |
| JFK    | 9.957610   |
| LGA    | 4.226548   |

PDF

MS Word

| origin | mean_delay |
|--------|------------|
| EWR    | 7.680089   |
| JFK    | 9.957610   |
| LGA    | 4.226548   |

# captions

```
```{r results = 'asis'}
```

```
knitr::kable(weekday, caption = "Results by day")
```

```
```
```

See **xtable**, **stargazer**, or **DT** packages for advanced alternatives.

# parameters

# Pop Quiz

## **What is the IMRAD format?**

Standard report structure for experimental science.

### **Introduction**

What hypothesis was tested and why?

### **Methods**

How was the study done?

### **Results**

What answer was discovered?

### **And Discussion**

What does the answer imply?

# Pop Quiz

## What is the IMRAD format?

Standard report structure for experimental science.

## Why?

### Introduction

What hypothesis was tested and why?

### Methods

How was the study done?

### Results

What answer was discovered?

### And Discussion

What does the answer imply?

# Pop Quiz

What would a person need to reproduce  
your report?

# YAML

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

```
---
```

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

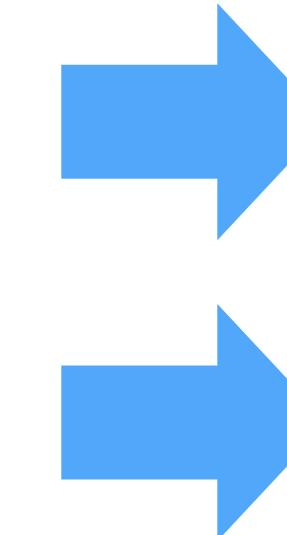
```
---
```

Text of document

# Parameters

A list of values that you can call in R code chunks

**params list**  
**elements and  
values**



```
---
```

```
title: "Untitled"
```

```
output: html_document
```

```
params:
```

```
  filename: "data.csv"
```

```
  symbol: "FB"
```

```
---
```

colon

New line. Indented  
two spaces

Access as **params\$filename** and **params\$symbol**

# Data types

R Markdown will automatically parse **characters**, **numerics**, **integers**, and **logicals**. Create other types with an R expression.

```
---
```

```
title: "Untitled"
```

```
output: html_document
```

```
params:
```

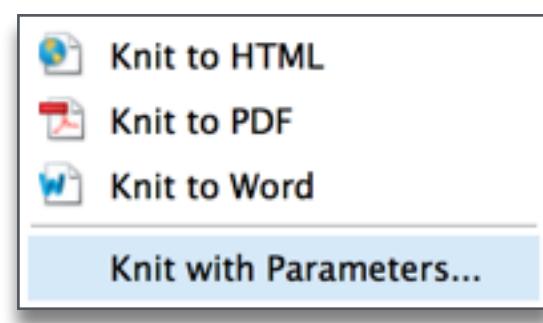
```
  date: !r as.Date("2015-01-01")
```

```
--
```

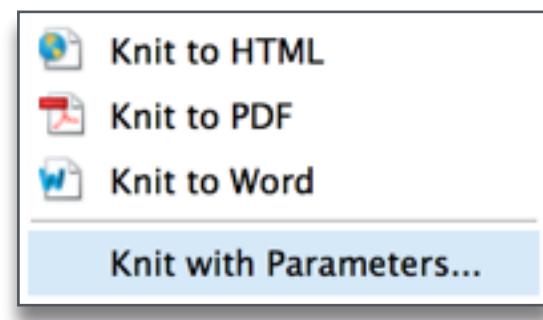
! r

R expression to  
coerce type

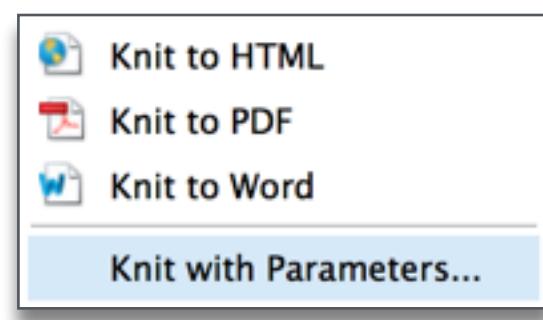
# Recap



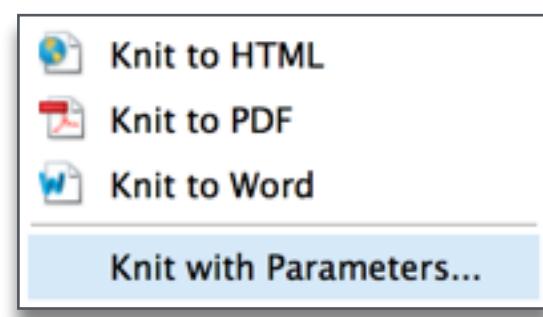
Place metadata in a **YAML header** at the very start of the file. Key value pairs surrounded by - - -



Add a **params** key and sub-keys to create parameter defaults



Access parameters in code as elements of **params\$**



Render with **knit with parameters** or **render()** combined with a params argument

# Your Turn

Add a day of the week parameter to your report.

Adjust your code to make the report update whenever the parameter changes. You will need to redefine **dow** in your code

Render a report for each of the 7 days.



# output formats

Untitled

RStudio  
April 16, 2015

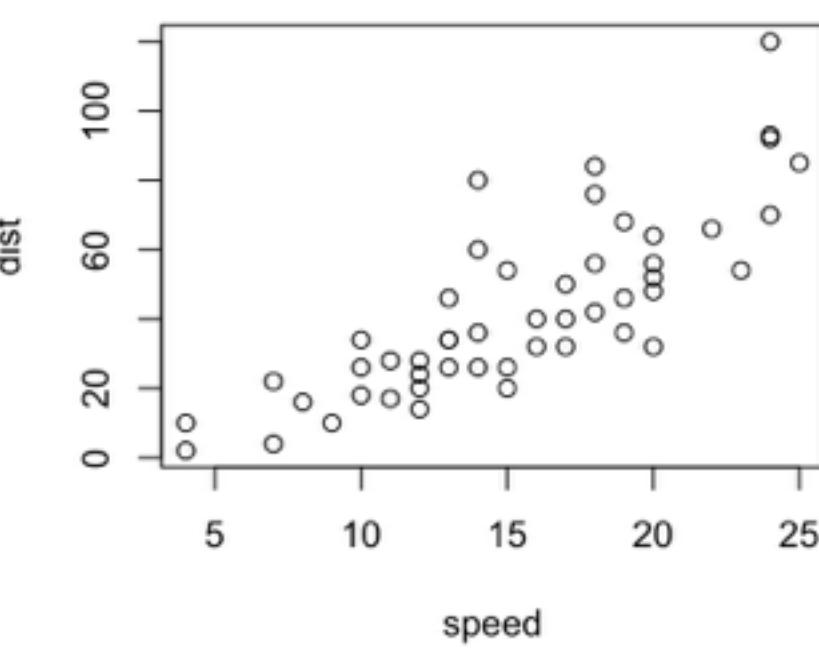
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

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

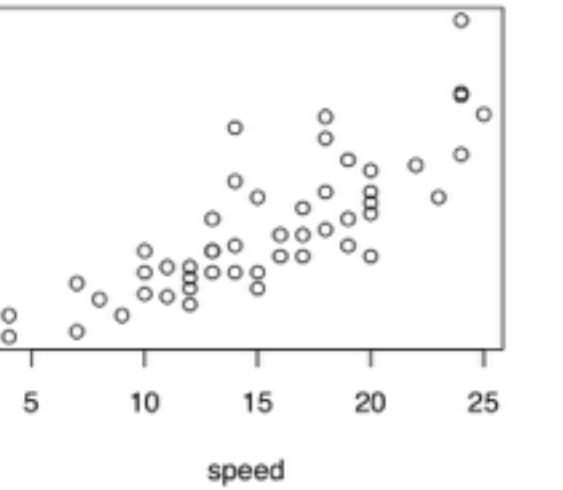
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

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"

Untitled

RStudio  
April 16, 2015

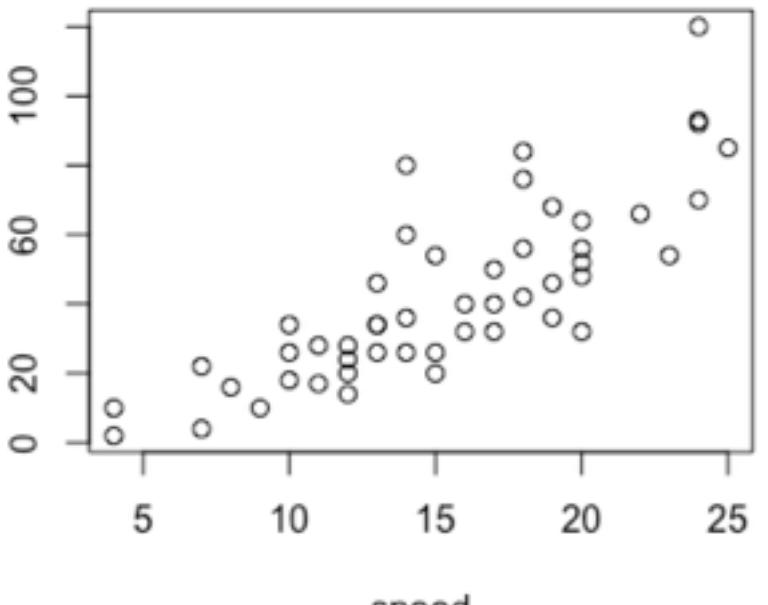
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

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
```

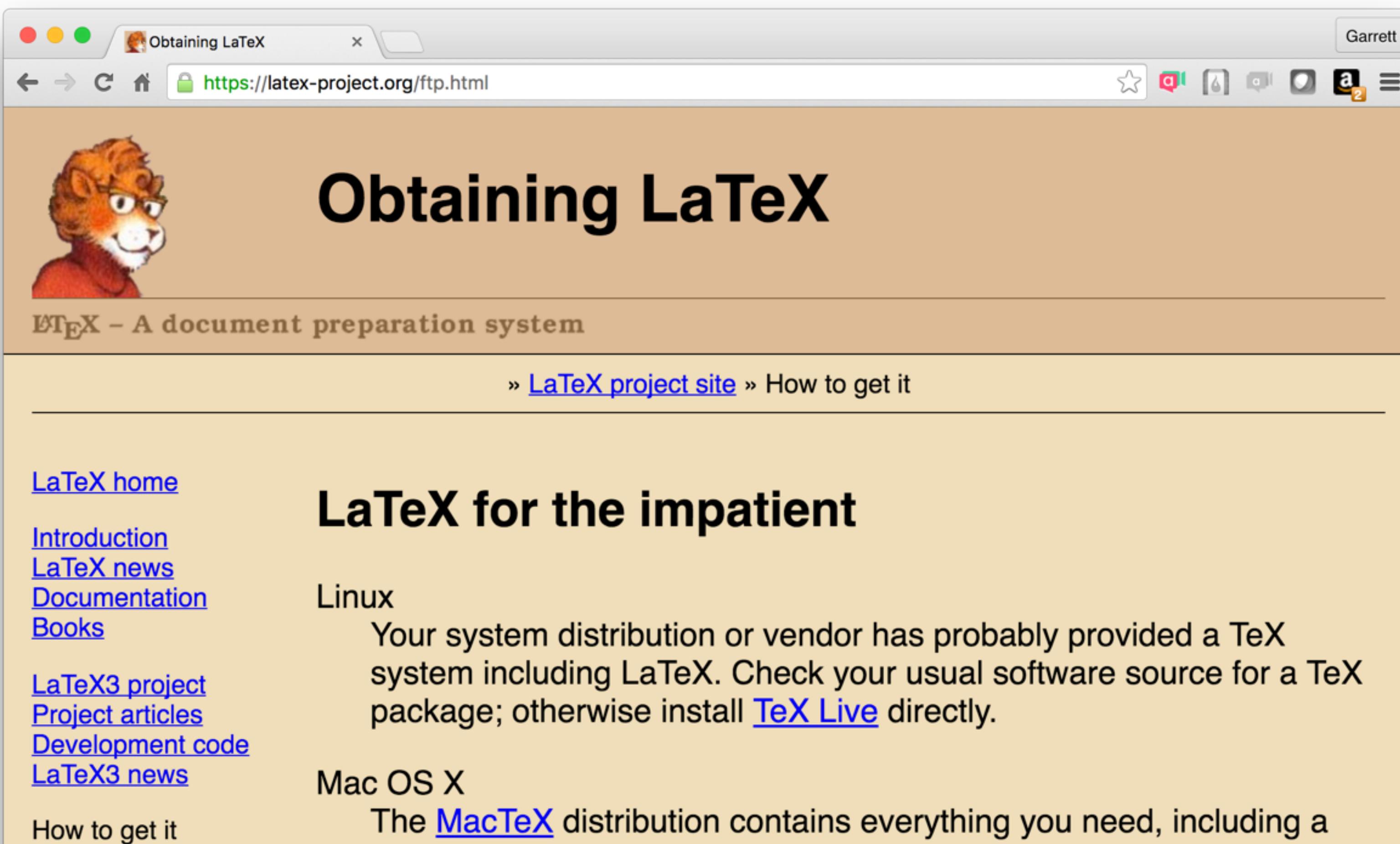
You can also embed plots, for example:



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

# LaTeX

<https://latex-project.org/ftp.html>

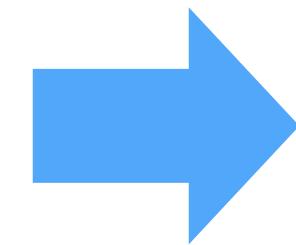


The screenshot shows a web browser window titled "Obtaining LaTeX". The address bar contains the URL <https://latex-project.org/ftp.html>. The page itself has a light orange background. On the left, there is a cartoon illustration of a brown lion wearing glasses. To the right of the lion, the word "Obtaining LaTeX" is written in a large, bold, black sans-serif font. Below the title, the text "LaTeX – A document preparation system" is displayed. At the top of the main content area, there is a breadcrumb navigation bar with the links "» LaTeX project site" and "» How to get it". The main content section is titled "LaTeX for the impatient". It includes a heading "Linux" and a paragraph explaining that TeX systems are often included in system distributions, with a note to check software sources or install [TeX Live](#) directly. There is also a section for "Mac OS X" which is partially visible at the bottom. On the far left, a sidebar lists various links: "LaTeX home", "Introduction", "LaTeX news", "Documentation", "Books", "LaTeX3 project", "Project articles", "Development code", and "LaTeX3 news". At the very bottom of the page, there is a link "How to get it".

# 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                                     |
| pdf_document          | → | pdf                                      |
| word_document         | → | Microsoft Word (.docx)                   |
| odt_document          | → | Open Document Text                       |
| rtf_document          | → | Rich Text Format                         |
| md_document           | → | markdown (converts R output to markdown) |
| ioslides_presentation | → | ioslides (HTML 5 slideshow)              |
| slidy_presentation    | → | slidy (HTML 5 slideshow)                 |
| beamer_presentation   | → | beamer (pdf slideshow)                   |

# Your Turn

Use output: and the knit button (or render()) to create an ioslides slide show from your report.



slides  
syntax

# Slide divisions

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

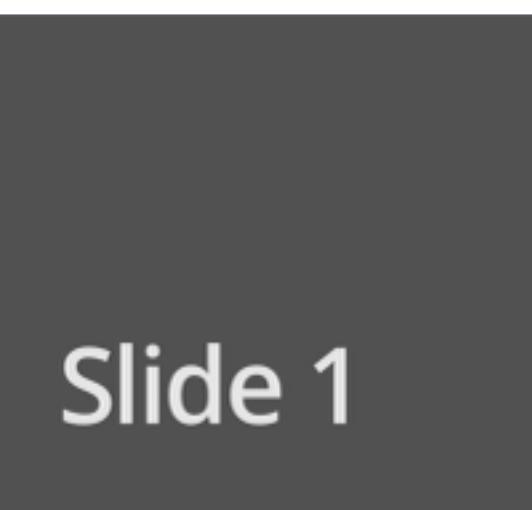
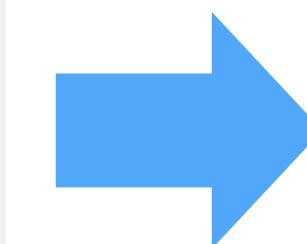
\* default for *ioslides*

```
# Slide 1  
text
```

```
## Slide 2  
text
```

\*\*\*

Slide 3



# Slide bullets

Start bullets with -

Incremental bullets with >-

```
## Slide 1
```

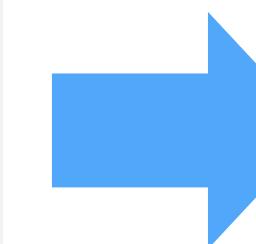
```
- Bullet 1
```

```
- Bullet 2
```

```
## Slide 2
```

```
>- Bullet A
```

```
>- Bullet B
```



Slide 1

- Bullet 1
- Bullet 2

Slide 2

**customizing  
output**

# sub options

```
> args(rmarkdown::html_document)
function (toc = FALSE, toc_depth = 3, number_sections = FALSE,
fig_width = 7, fig_height = 5, fig_retina = if (!fig_caption) 2,
fig_caption = FALSE, smart = TRUE, self_contained = TRUE,
theme = "default", highlight = "default", mathjax = "default",
template = "default", css = NULL, includes = NULL, keep_md = FALSE,
lib_dir = NULL, pandoc_args = NULL, ...)
```

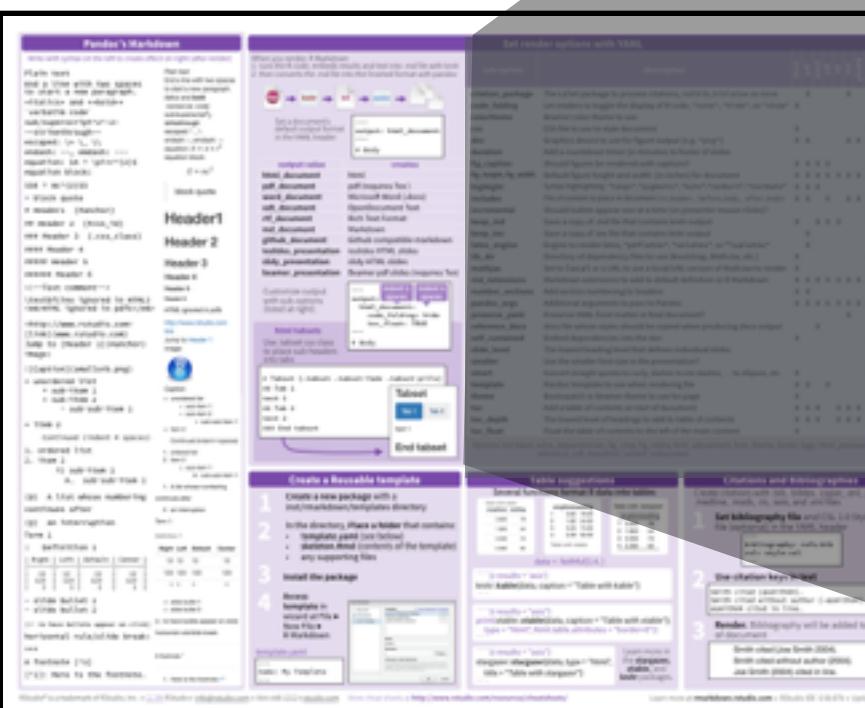
# R Markdown Cheatsheet

# Help ►

# Cheatsheets ►

# R Markdown Cheatshee

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



## Set render options with YAML

| sub-option            | description  | html | pdf | word | odt | rtf | md | gitbook | ioslides | slidy | beamer |
|-----------------------|--|------|-----|------|-----|-----|----|---------|----------|-------|--------|
| citation_package      | The LaTeX package to process citations, natbib, biblatex or none             | X    |     | X    |     |     |    |         |          |       | X      |
| code_folding          | Let readers to toggle the display of R code, "none", "hide", or "show"       | X    |     |      |     |     |    |         |          |       |        |
| colortheme            | Beamer color theme to use  |      |     |      |     |     |    |         |          |       | X      |
| css                   | CSS file to use to style document  | X    |     |      |     |     |    |         | X        | X     |        |
| dev                   | Graphics device to use for figure output (e.g. "png")                        | X    | X   |      |     |     |    |         | X        | X     | X      |
| duration              | Add a countdown timer (in minutes) to footer of slides                       |      |     |      |     |     |    |         |          |       | X      |
| fig_caption           | Should figures be rendered with captions?                                    | X    | X   | X    | X   |     |    |         | X        | X     | X      |
| fig_height, fig_width | Default figure height and width (in inches) for document                     | X    | X   | X    | X   | X   | X  | X       | X        | X     | X      |
| highlight             | Syntax highlighting: "tango", "pygments", "kate", "zenburn", "textmate"      | X    | X   | X    |     |     |    |         |          | X     | X      |
| includes              | File of content to place in document (in_header, before_body, after_body)    | X    | X   |      | X   |     |    | X       | X        | X     | X      |
| incremental           | Should bullets appear one at a time (on presenter mouse clicks)?             |      |     |      |     |     |    |         |          | X     | X      |
| keep_md               | Save a copy of .md file that contains knitr output                           | X    |     | X    | X   | X   |    |         |          | X     | X      |
| keep_tex              | Save a copy of .tex file that contains knitr output                          |      |     |      |     |     | X  |         |          |       | X      |
| latex_engine          | Engine to render latex, "pdflatex", "xelatex", or "lualatex"                 |      |     |      |     | X   |    |         |          |       | X      |
| lib_dir               | Directory of dependency files to use (Bootstrap, MathJax, etc.)              | X    |     |      |     |     |    |         | X        | X     |        |
| mathjax               | Set to local or a URL to use a local/URL version of MathJax to render        | X    |     |      |     |     |    |         | X        | X     |        |
| md_extensions         | Markdown extensions to add to default definition of R Markdown               | X    | X   | X    | X   | X   | X  | X       | X        | X     | X      |
| number_sections       | Add section numbering to headers   | X    | X   |      |     |     |    |         |          |       |        |
| pandoc_args           | Additional arguments to pass to Pandoc                                       | X    | X   | X    | X   | X   | X  | X       | X        | X     | X      |
| preserve_yaml         | Preserve YAML front matter in final document?                                |      |     |      |     |     |    |         |          | X     |        |
| reference_docx        | docx file whose styles should be copied when producing docx output           |      |     |      |     |     | X  |         |          |       |        |
| self_contained        | Embed dependencies into the doc  |      |     |      |     | X   |    |         |          | X     | X      |
| slide_level           | The lowest heading level that defines individual slides                      |      |     |      |     |     |    |         |          |       | X      |
| smaller               | Use the smaller font size in the presentation?                               |      |     |      |     |     |    |         |          |       | X      |
| smart                 | Convert straight quotes to curly, dashes to em-dashes, ... to ellipses, etc. | X    |     |      |     |     |    |         |          | X     | X      |
| template              | Pandoc template to use when rendering file                                   | X    | X   |      | X   |     |    |         |          | X     | X      |
| theme                 | Bootswatch or Beamer theme to use for page                                   | X    |     |      |     |     |    |         |          |       | X      |
| toc                   | Add a table of contents at start of document                                 | X    | X   | X    |     |     | X  | X       | X        |       | X      |
| toc_depth             | The lowest level of headings to add to table of contents                     | X    | X   | X    |     |     | X  | X       | X        |       |        |
| toc_float             | Float the table of contents to the left of the main content                  | X    |     |      |     |     |    |         |          |       |        |

Options not listed: extra\_dependencies, fig\_crop, fig\_retina, font\_adjustment, font\_theme, footer, logo, html\_preview, reference\_odt, transition, variant, widescreen

# sub options

```
---
```

```
title: "Untitled"
output: html_document
```

```
---
```

# sub options

---

```
title: "Untitled"  
output:  
  html_document
```

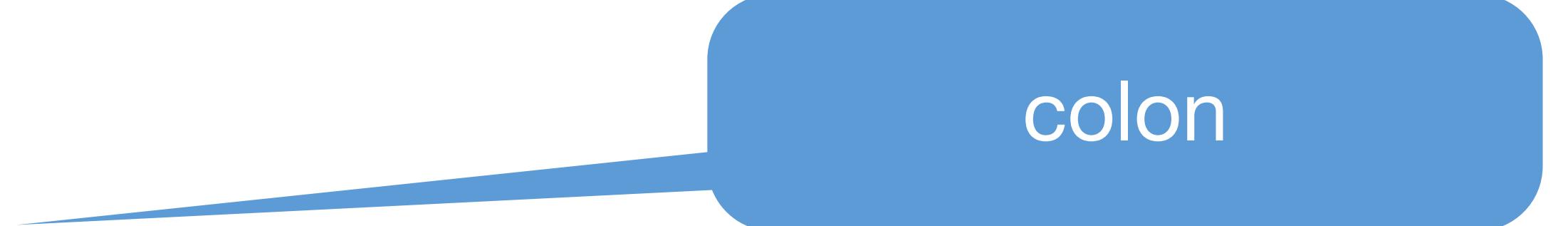
---

New line. Indented  
two spaces

# sub options

---

```
title: "Untitled"  
output:  
  html_document:
```



colon

---

# sub options

```
---
```

```
title: "Untitled"
```

```
output:
```

```
  html_document:
```

```
    toc: true
```

```
    theme: flatly
```

```
---
```

New lines. Indented  
two more spaces

Table of contents

pre-made CSS theme

# HTML themes

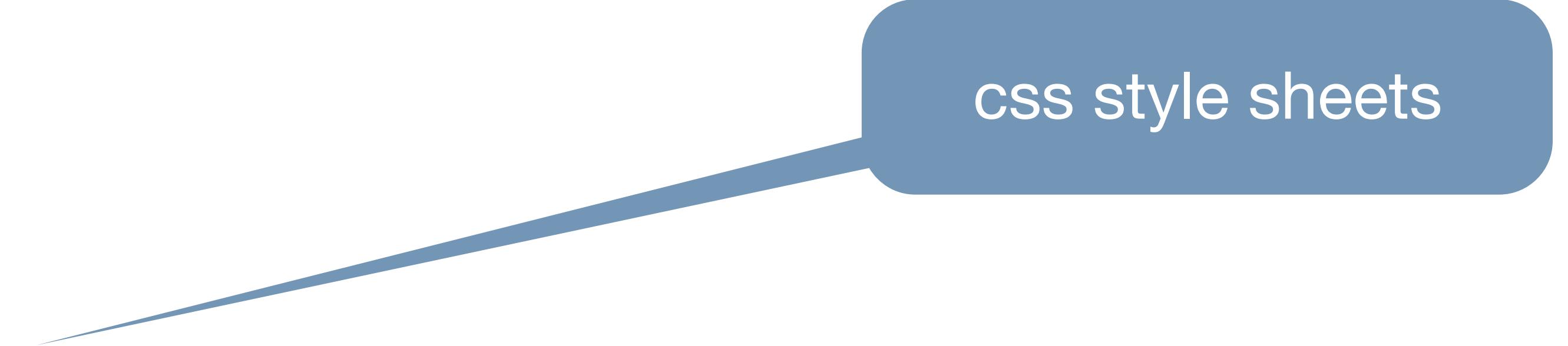
default  
cerulean  
journal  
flatly  
readable  
spacelab  
united  
cosmo

# follow up ideas

---

```
title: "Untitled"  
output:  
  html_document:  
    css: my-file.css
```

---



css style sheets

# Your Turn

default

cerulean

journal

flatly

readable

spacelab

united

cosmo

Revert your report to an  
html\_document.

Then add a table of contents to your  
report.

Then make the table of contents *float*,  
and choose your favorite style theme.



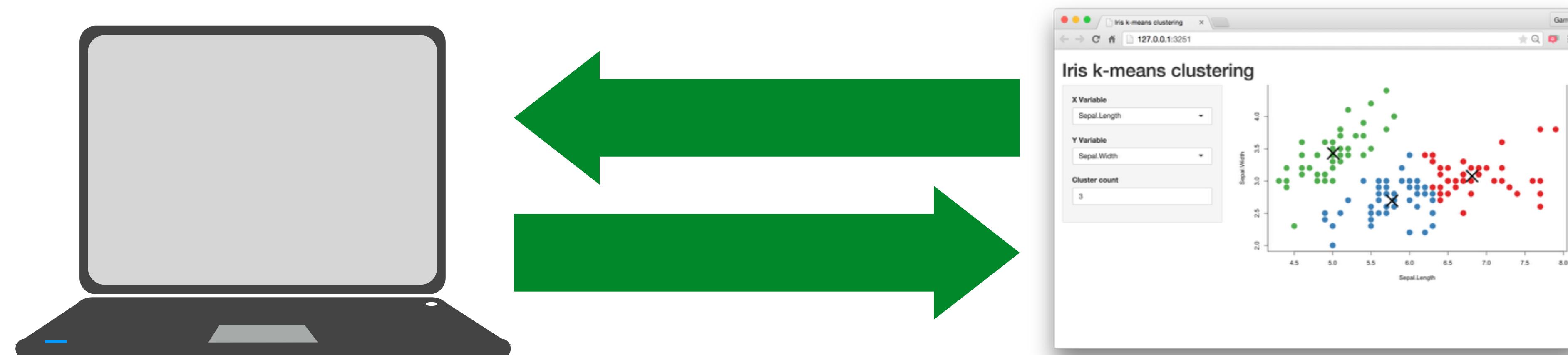
# Interactive documents



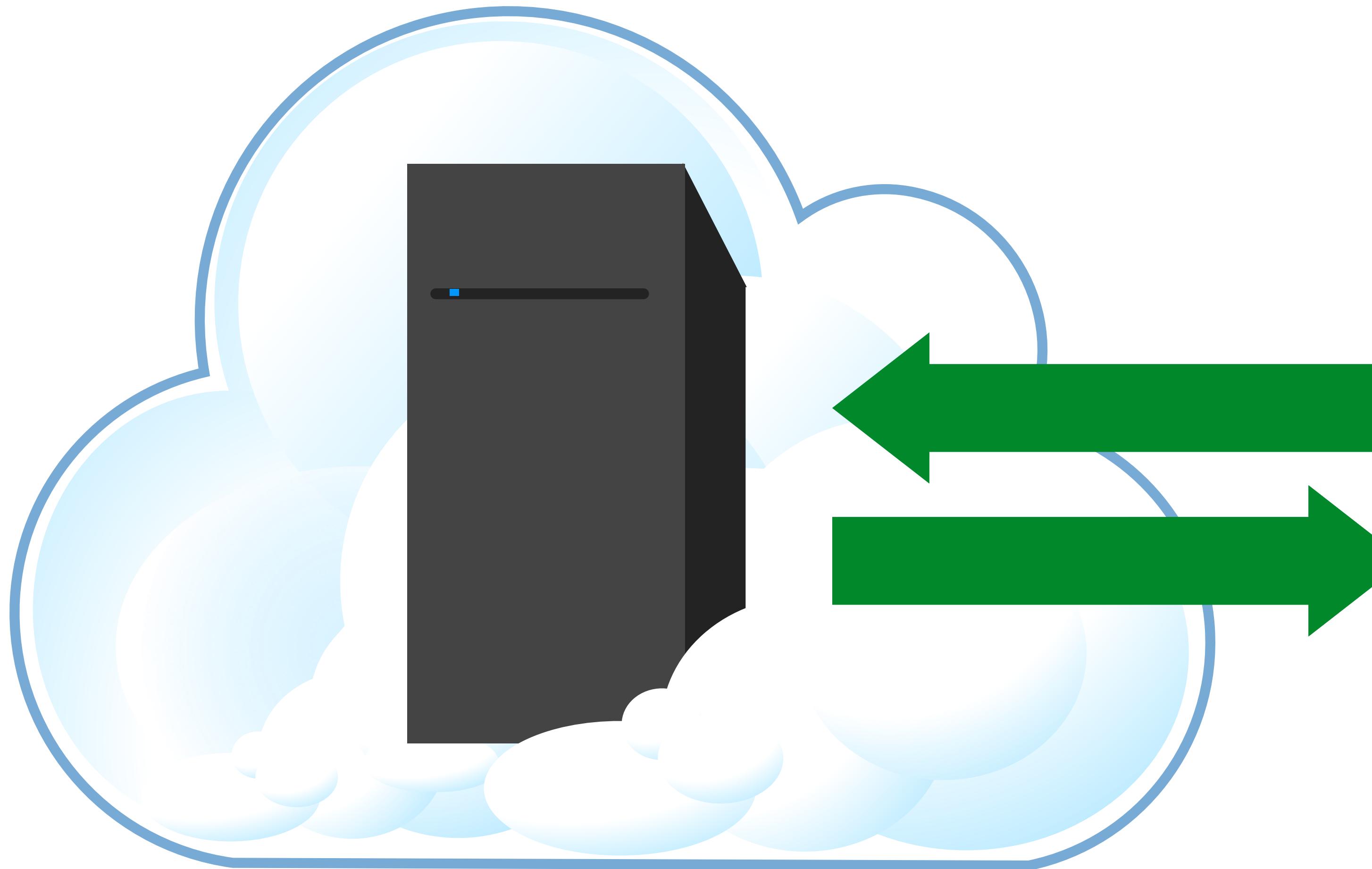
An R package that let's you build  
interactive HTML based apps with R.

```
install.packages("shiny")
```

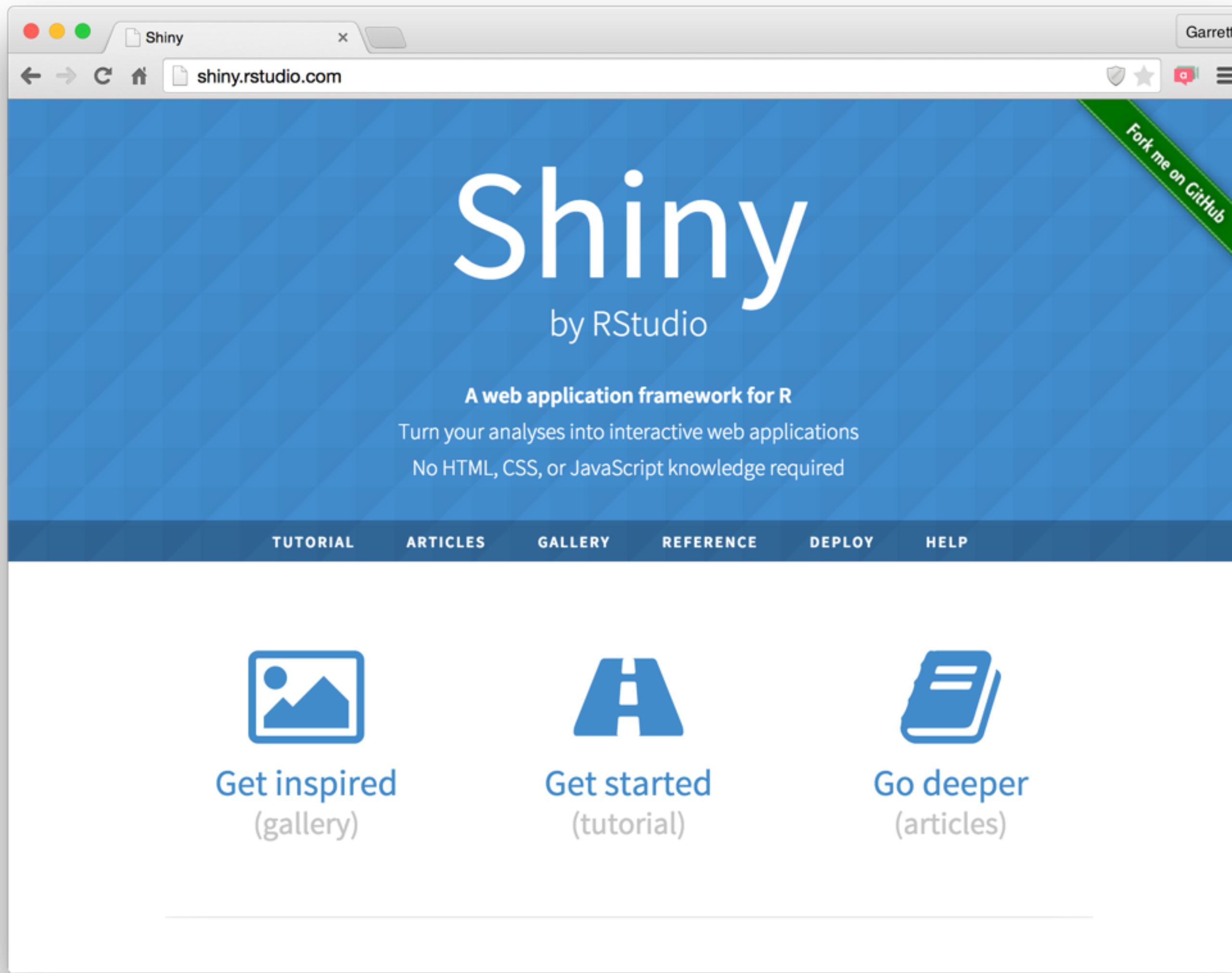
Every Shiny app is maintained by a computer running R



Every Shiny app is maintained by a computer running R



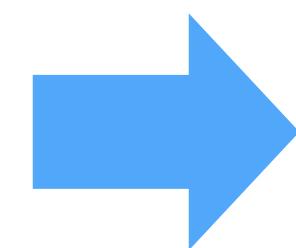
Learn more at  
[shiny.rstudio.com](http://shiny.rstudio.com)



# runtime:

Add **runtime: shiny** to any document that contains Shiny components.

**Arranges for  
document to run  
as a shiny app**



```
---
```

```
title: "Untitled"
```

```
author: "RStudio"
```

```
output: html_document
```

```
runtime: shiny
```

```
---
```

```
Text of document
```

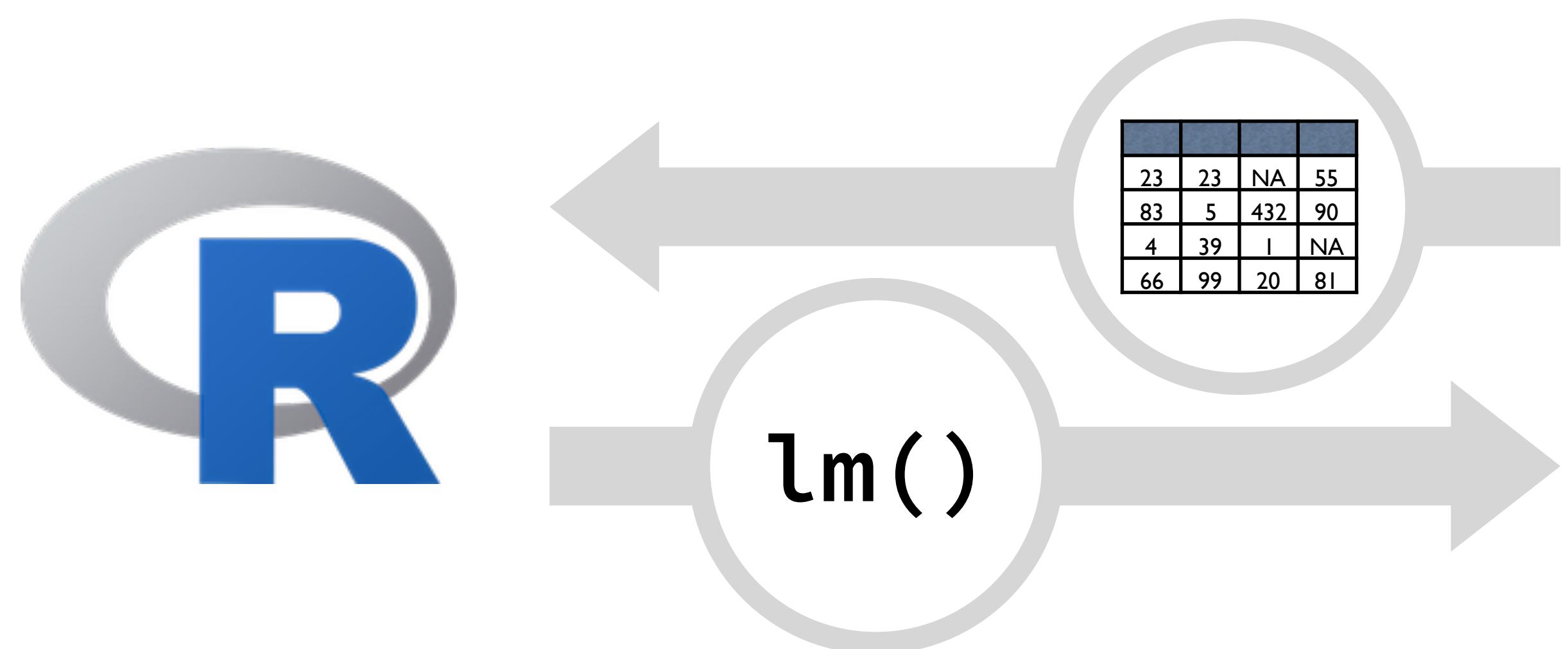
**Must use an html  
based output**  
(e.g. html\_document,  
ioslides\_presentation,  
slidy\_presentation)\_

# **Big Data**

# General Strategy

Store big data in a data warehouse

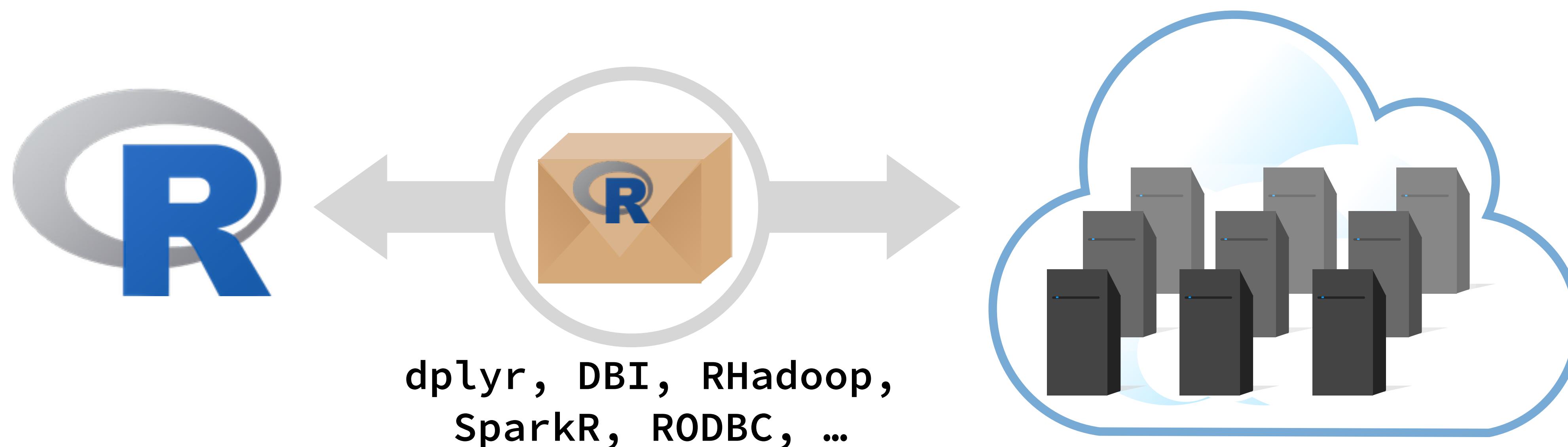
1. Pass subsets of data from warehouse to R
2. Transform R code, pass to warehouse.



# General Strategy

Store big data in a data warehouse

1. Pass subsets of data from warehouse to R
2. Transform R code, pass to warehouse.



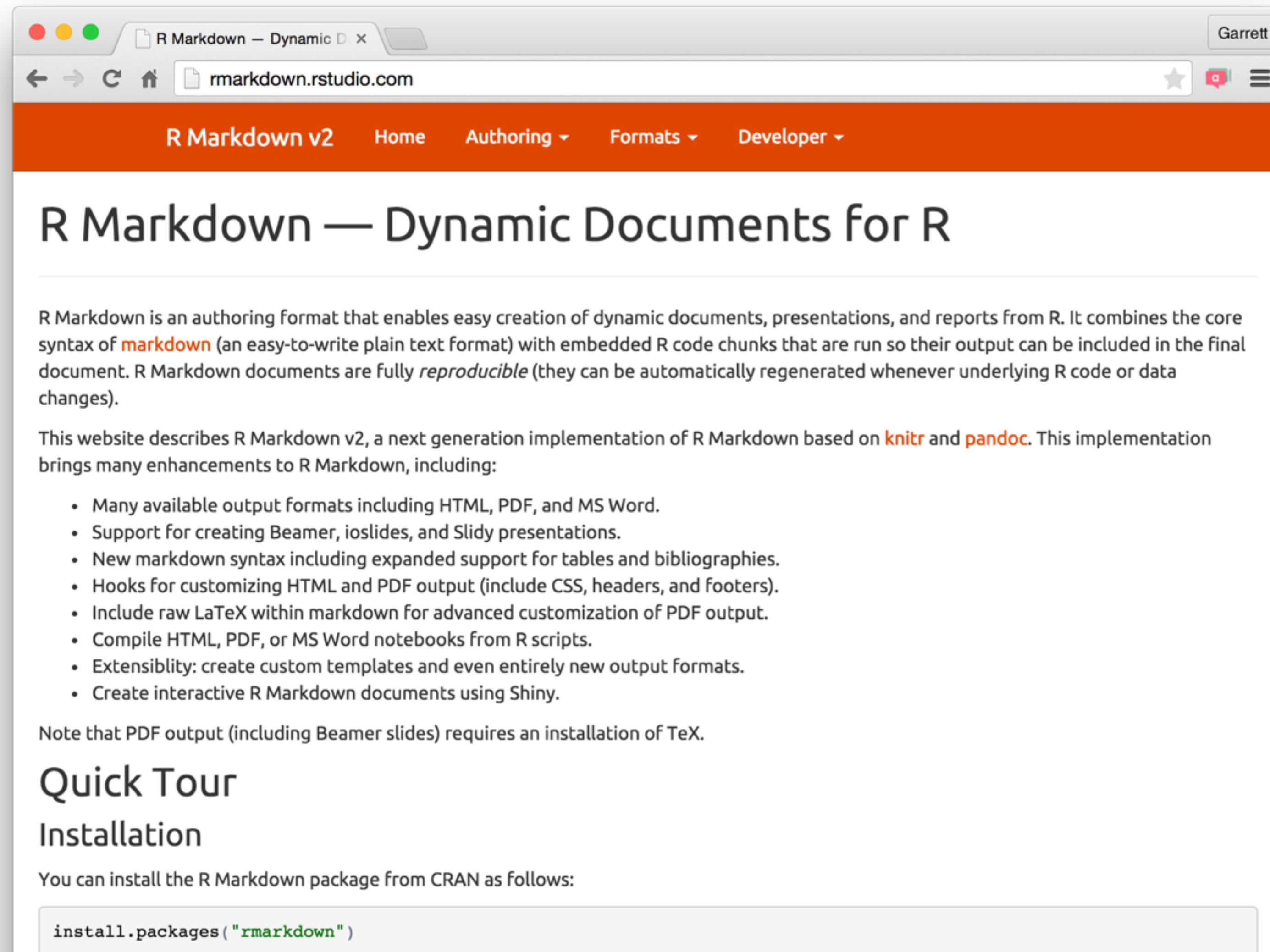
# Recall

1. R code chunks are run "as is" in a fresh R session
2. Use the same techniques that you would use at the command line to access big, out of memory data.

# Resources

# The R Markdown Development Center

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

A screenshot of a web browser displaying the R Markdown website. The title bar shows 'R Markdown — Dynamic' and the URL '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 what R Markdown is, followed by a section about the R Markdown v2 implementation and its enhancements. A bulleted list details these enhancements. At the bottom, there's a note about TeX requirements and two sections: 'Quick Tour' and 'Installation'.

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.

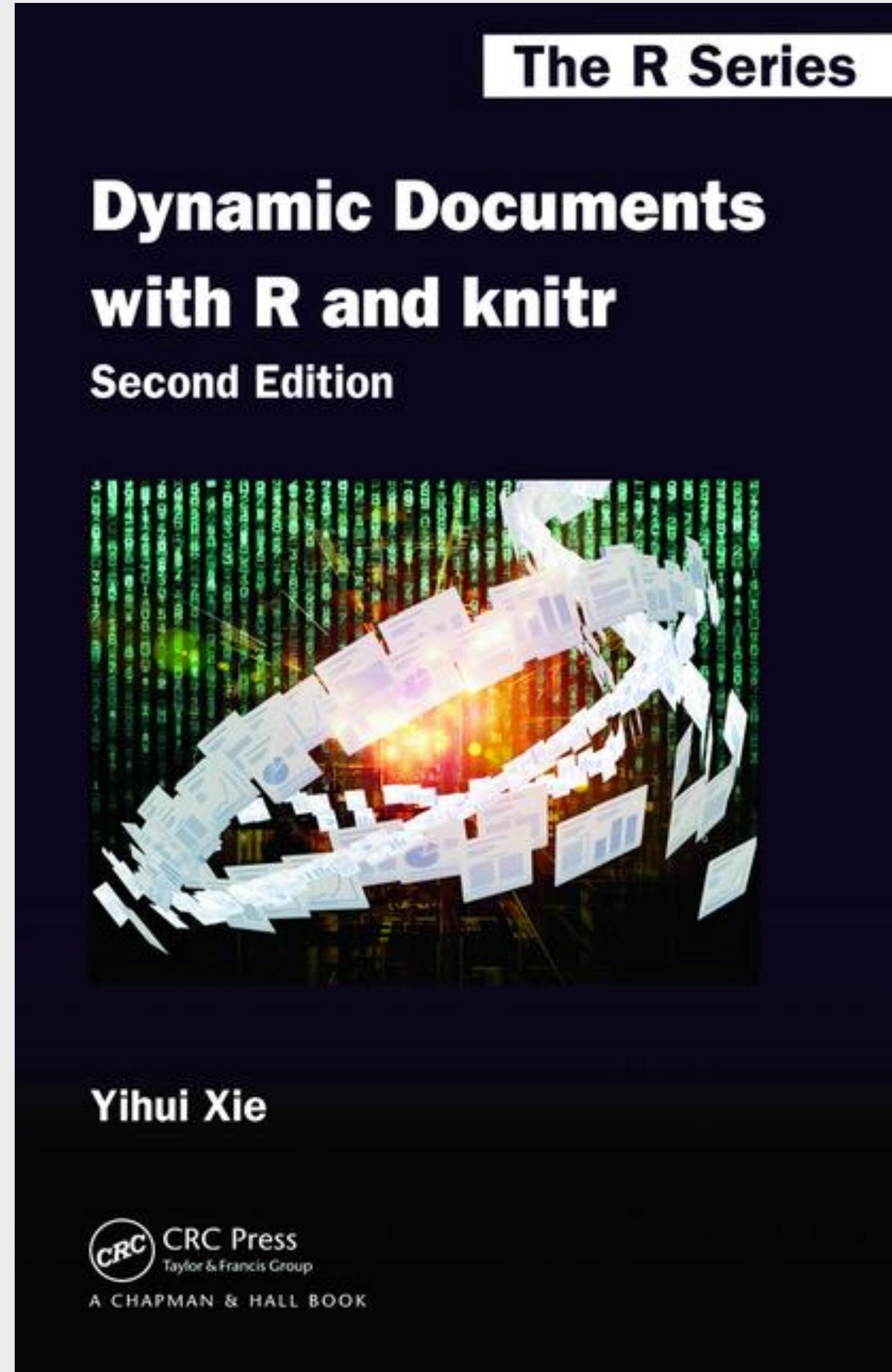
Note that PDF output (including Beamer slides) requires an installation of TeX.

## Quick Tour

## Installation

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

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



## 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/1498716962](http://amzn.com/1498716962)

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{Rout}` and `\end{Rout}`; 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