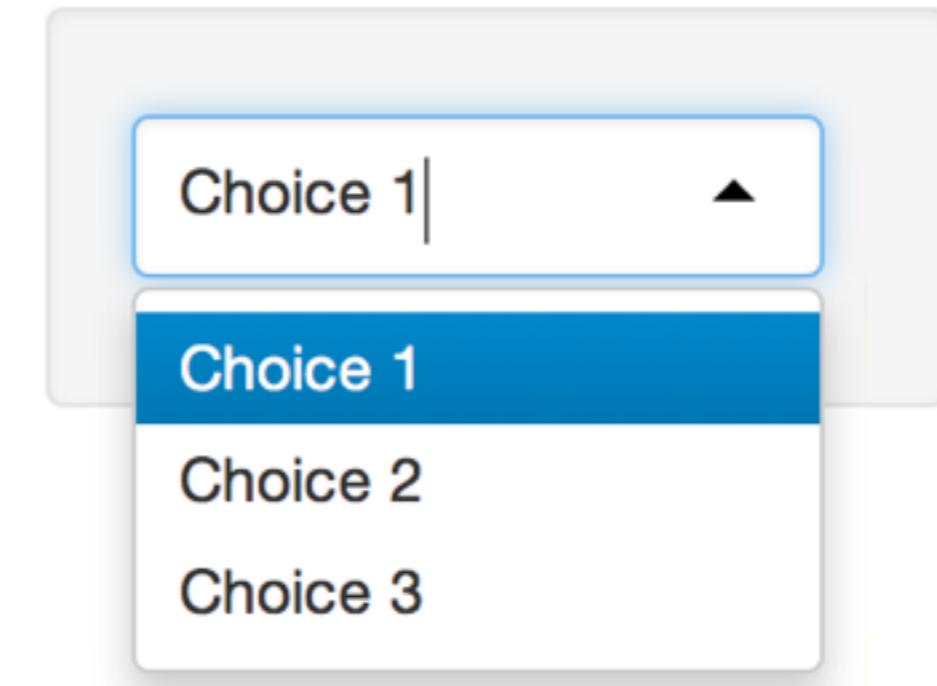


All Training materials are provided "as is" and without warranty and RStudio disclaims any and all express and implied warranties including without limitation the implied warranties of title, fitness for a particular purpose, merchantability and noninfringement.

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.

Interactive Data Display

with Shiny and R Markdown



Garrett Grolemund

Master Instructor, RStudio

June 2014

HELLO
my name is

Garrett



garrett@rstudio.com



@StatGarrett

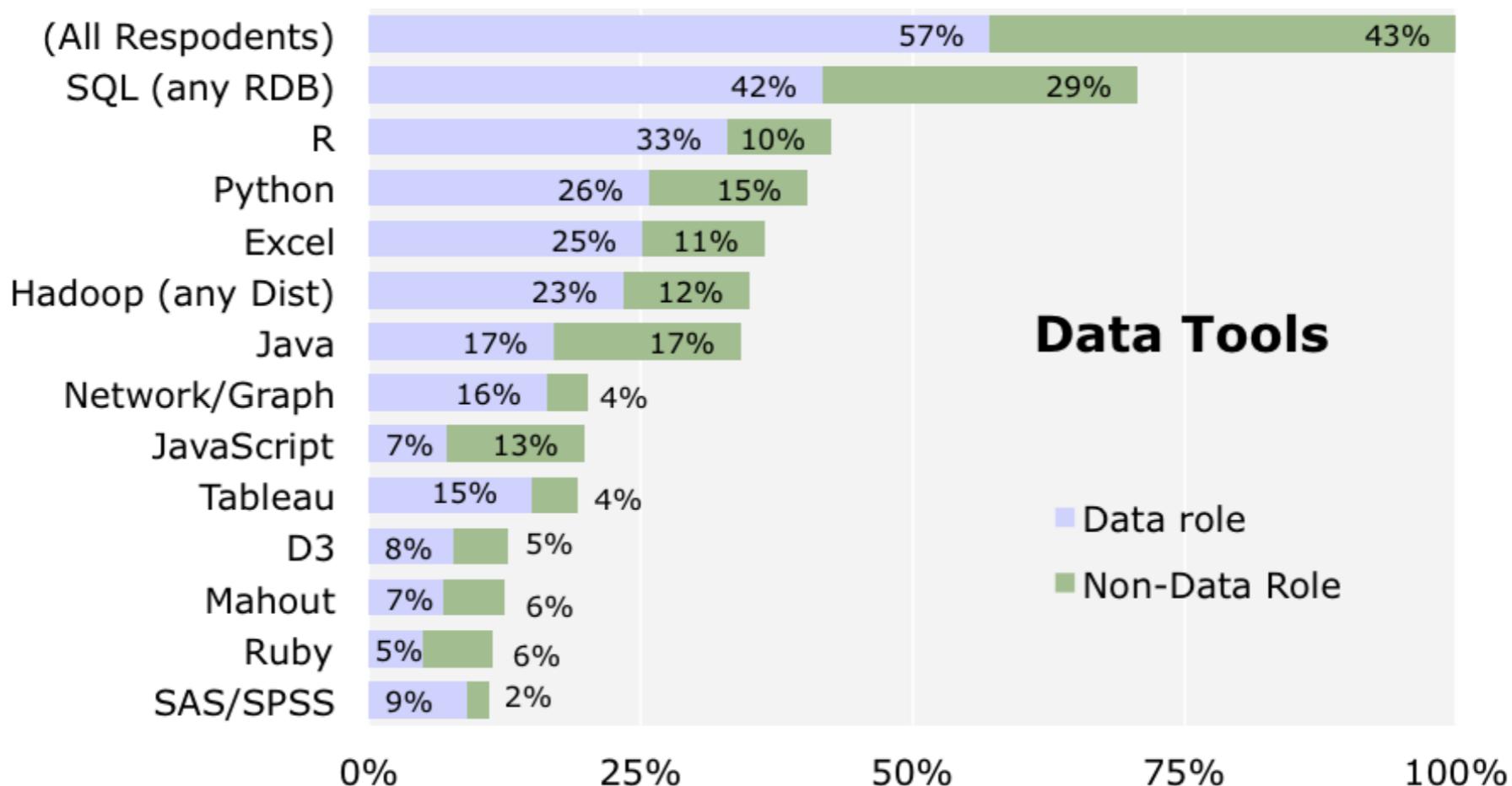
Warm up

Introduce yourself to your neighbor. Tell them

- * your name
- * your job
- * 2 to 3 interesting things about yourself

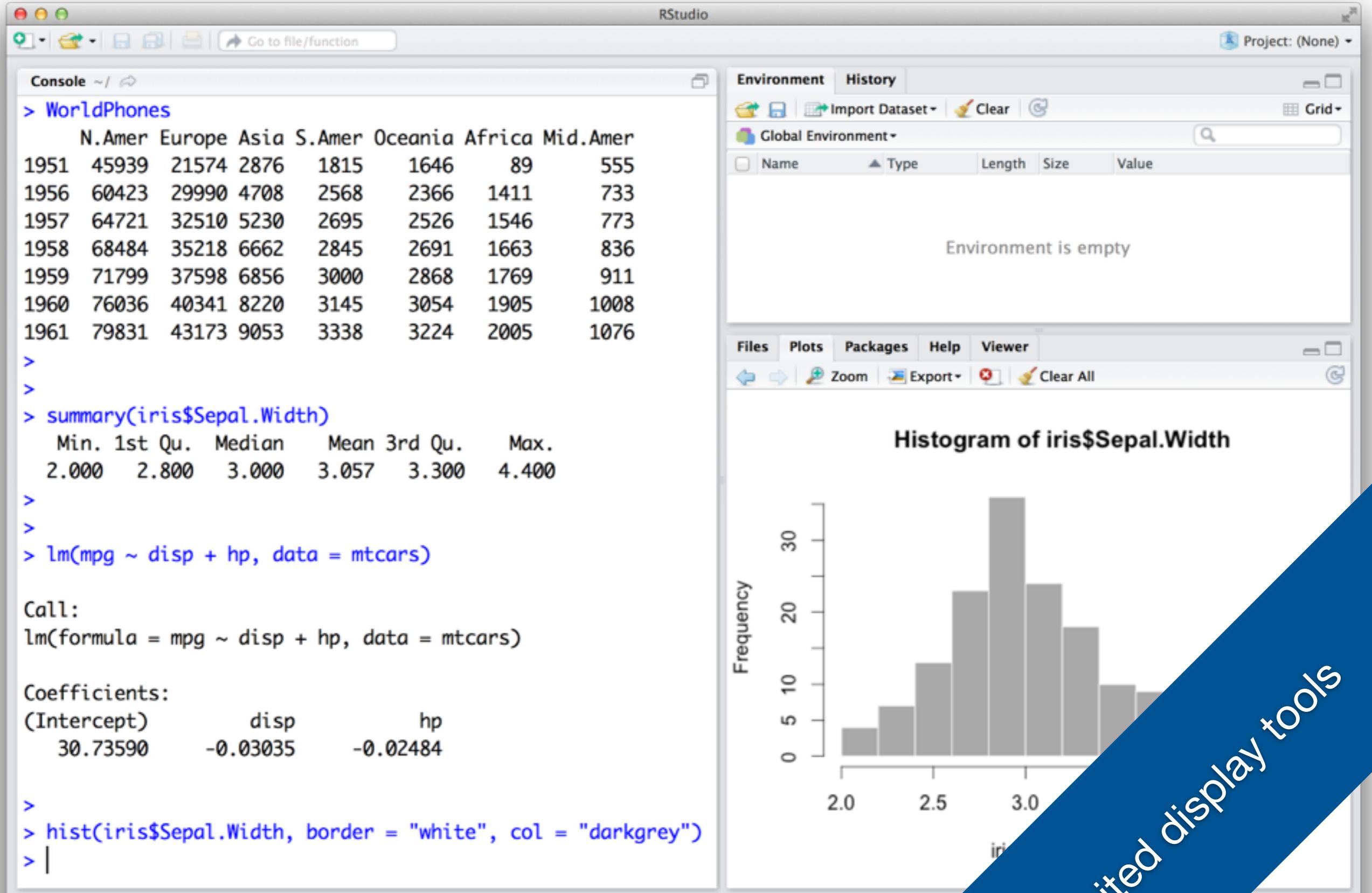


2013 Data Science Salary Survey



Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo





Limited display tools

Shiny



Analytic
Power



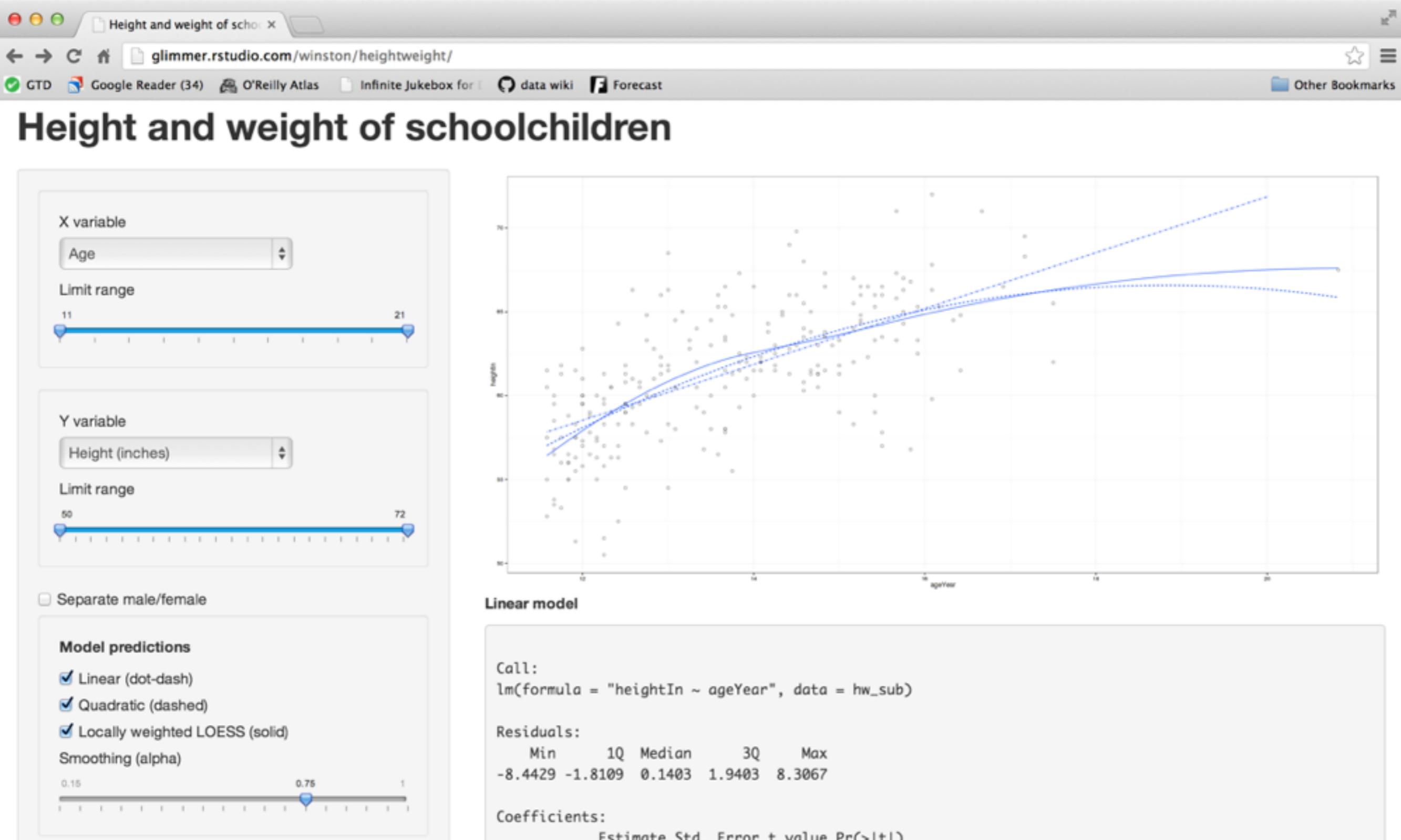
HTML



CSS



Display abilities
and interactivity



<http://glimmer.rstudio.com/winston/heightweight/>

Shiny

An R package that does two things

1. Creates reactive R objects
2. Builds HTML web pages

R Markdown



Analytic
Power



Microsoft Word



Reveal.js
ioslides, Beamer



Report generation

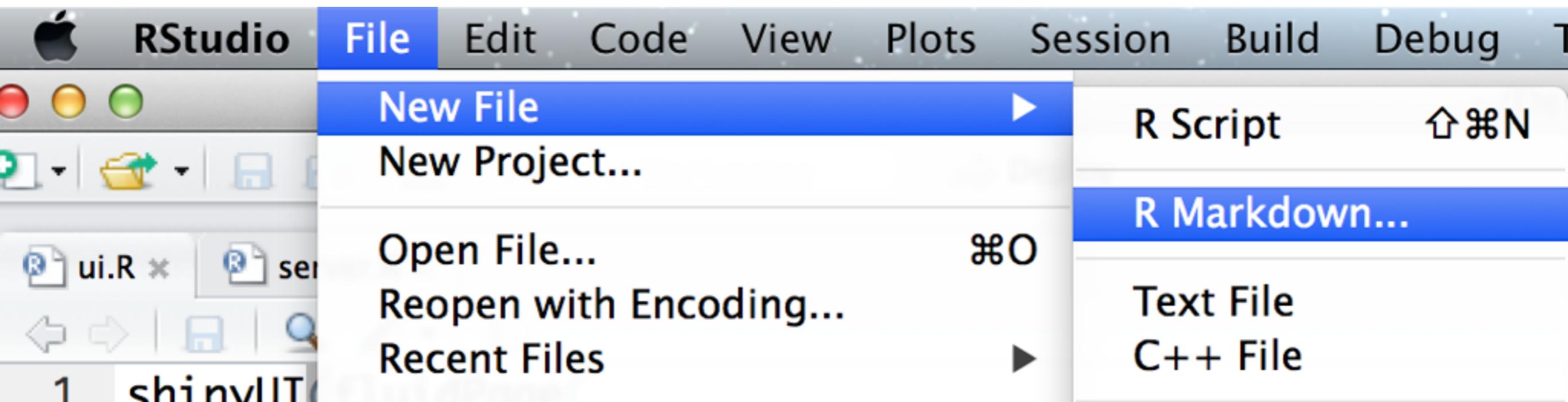
- 1. Reports in R**
- 2. Interactive Reports**
- 3. Web Apps**
- 4. Sharing**

RStudio IDE
text editor

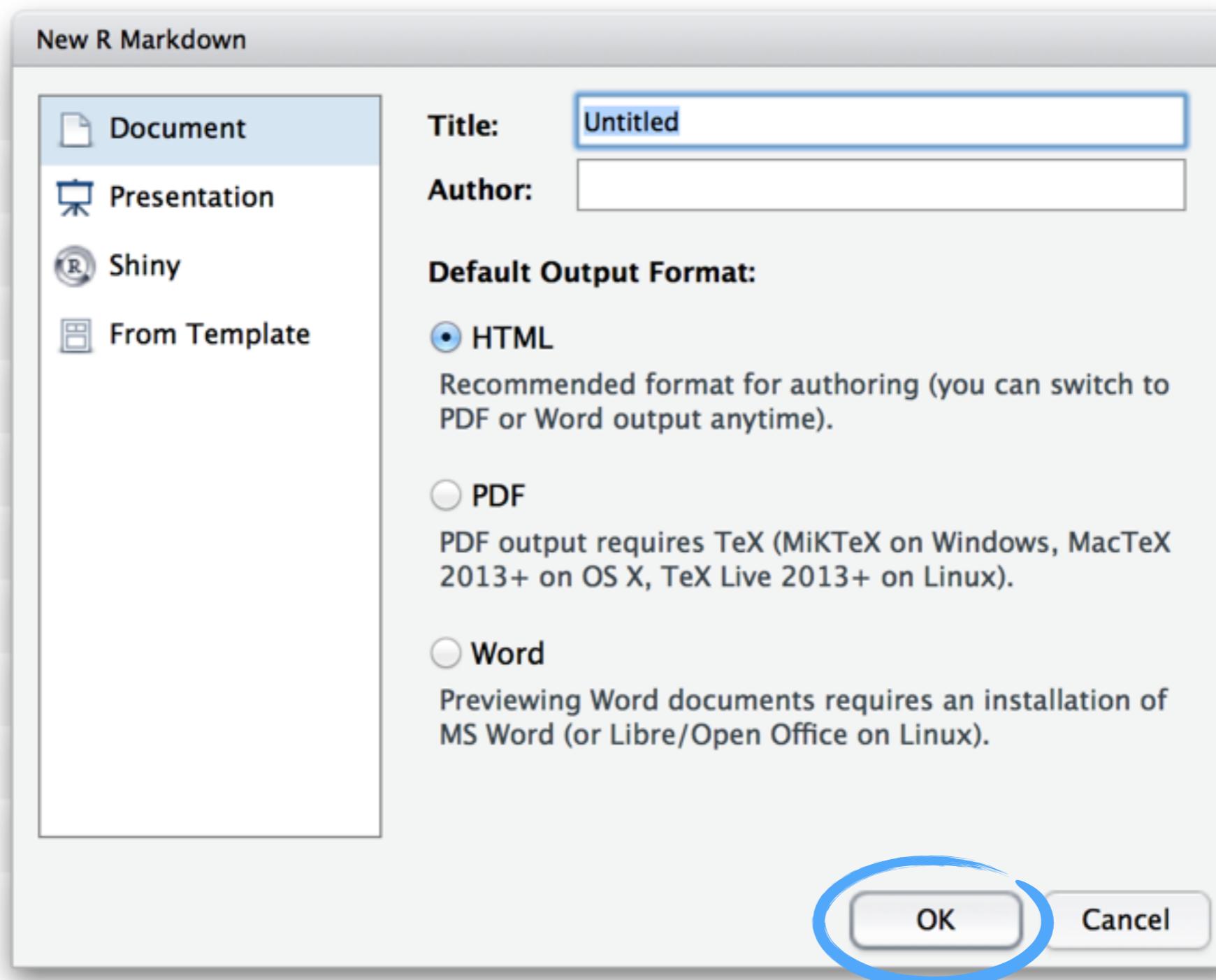
RStudio text editor

A dedicated pane for composing various types of documents

File > New File > ...



To open a basic R markdown document select "R Markdown" and then click OK in the pop up



~/Dropbox (RStudio)/RStudio/rstudio-training/interactive-reports – RStudio

Untitled1 x Go to file/function

Interactive-reports

Environment History

Knit HTML Run Chunks

Import Dataset Clear Grid

Global Environment

Name Type Length Size Value

Environment is empty

Files Plots Packages Help Viewer

Zoom Export Clear All

Untitled1 x ABC ? Knit HTML Run Chunks

1 - ---
2 title: "Untitled"
3 output: html_document
4 ---
5
6 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>>.
7
8 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:
9
10 - ``{r}
11 summary(cars)
12 ````

2:1 (Top Level) R Markdown

Console ~/Dropbox (RStudio)/RStudio/rstudio-training/interactive-reports/

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/Dropbox (RStudio)/RStudio/rstudio-training/int
eractive-reports/.RData]

> |

Your turn

Do you remember what you said to your neighbor?

Open a .Rmd file. Erase everything in it. Then write down:

- * your name
- * your job
- * 2 to 3 interesting things about yourself

Garrett Grolemund

Master Instructor

I wrote the lubridate R package

I'm the author of *_Hands On Programming with R_*.

I like to play the ukulele (but I'm not very good at it).

(Pandoc)
Markdown

Markdown

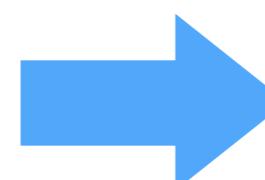
A simple way to write HTML.

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

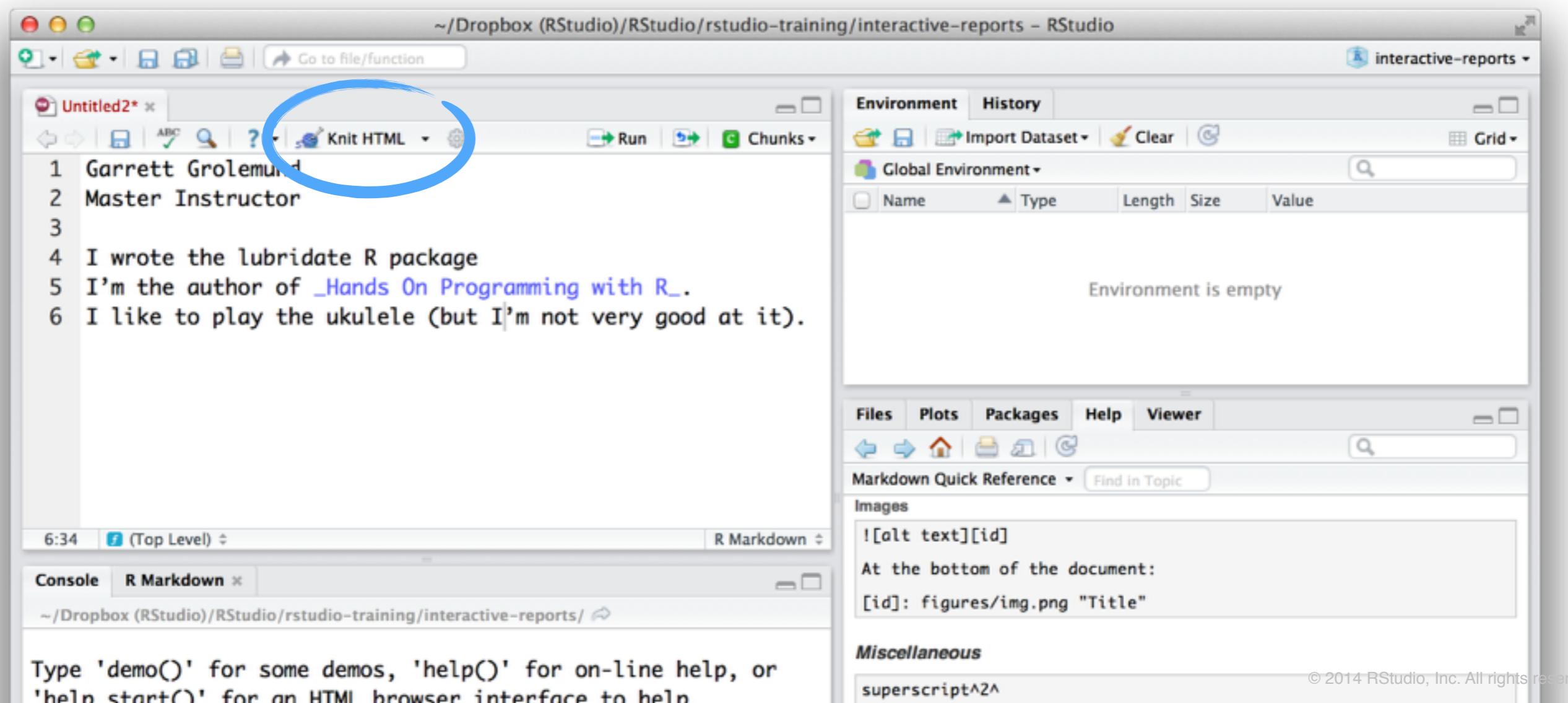
- easy to use
- simple
- fun?

Web sites that use markdown

- * **GitHub** www.github.com
- * **StackOverflow** www.stackoverflow.com
- * **Reddit** www.reddit.com
- * **Meteor** www.meteor.com
- * many more

To compile markdown into HTML, click the "Knit HTML" button.

Note your file must have the extension `.md` or `.Rmd`



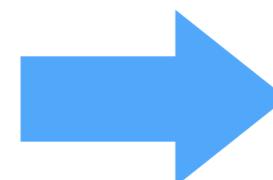


Basics

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
```

Basics

Text is rendered as plain text.

Surround words with underscores (_) to make italics.

Use two underscores (___) to make bold.

Use back ticks to make code.

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



Basics

Use asterisks to make bullet points.

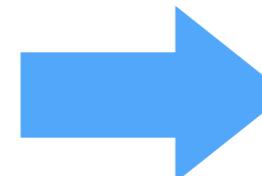
Use numbers to make numbered lists.

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

Markdown guide

The screenshot shows the RStudio interface with a focus on the Markdown guide. In the top-left corner, there's a blue circular icon with a white 'R' and the word 'Studio' next to it. The main title 'Markdown guide' is displayed prominently at the top.

The RStudio interface includes several panels:

- Code Editor:** Shows an R Markdown file named 'Untitled2.Rmd'. The code contains sections for 'Bullets' and 'Numbered list'. A blue circle highlights the question mark icon in the toolbar above the editor, which has a tooltip 'Using R Markdown'.
- Environment:** Shows an empty environment with the message 'Environment is empty'.
- Console:** Displays the message '[Workspace loaded from ~/Dropbox (RStudio)/RStudio/rstudio-training/interactive-reports/.RData]'
- Help:** A 'Markdown Quick Reference' panel is open, providing a quick reference for R Markdown syntax.

The 'Using R Markdown' link in the tooltip points to the following content in the 'Markdown Quick Reference' panel:

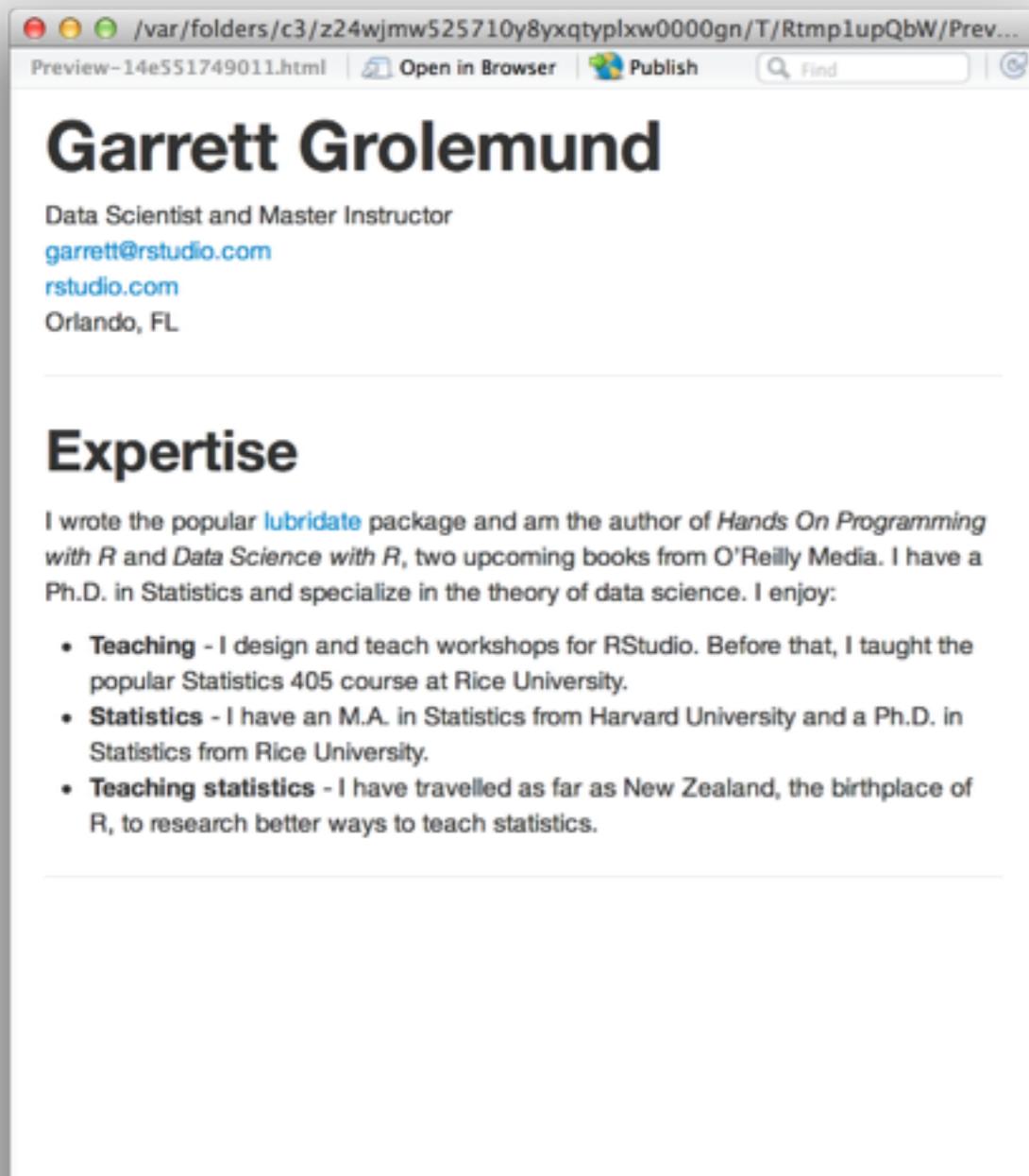
Markdown Quick Reference

R Markdown is an easy-to-write plain text format for creating dynamic documents and reports. See [Using R Markdown](#) to learn more.

Emphasis

```
*italic*    **bold**  
_italic_    __bold__
```

Your turn



The screenshot shows an RStudio preview window with the following content:

Garrett Grolemund
Data Scientist and Master Instructor
garrett@rstudio.com
rstudio.com
Orlando, FL

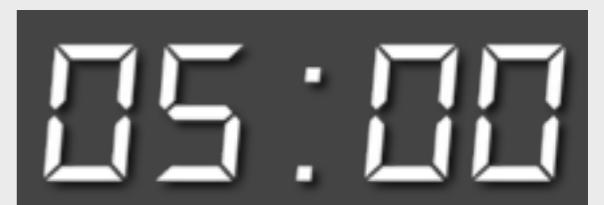
Expertise
I wrote the popular [lubridate](#) package and am the author of *Hands On Programming with R* and *Data Science with R*, two upcoming books from O'Reilly Media. I have a Ph.D. in Statistics and specialize in the theory of data science. I enjoy:

- **Teaching** - I design and teach workshops for RStudio. Before that, I taught the popular Statistics 405 course at Rice University.
- **Statistics** - I have an M.A. in Statistics from Harvard University and a Ph.D. in Statistics from Rice University.
- **Teaching statistics** - I have travelled as far as New Zealand, the birthplace of R, to research better ways to teach statistics.

Turn your text into an HTML resume. Use:

1. headings
2. italics and bold
3. a list
4. hyperlinks (challenge)
5. horizontal rules (challenge)

Open the RStudio markdown guide if you need help.



Garrett Grolemund
Data Scientist and Master Instructor
garrett@rstudio.com
[rstudio.com](www.rstudio.com)
Orlando, FL

Expertise

I wrote the popular [lubridate](http://www.r-statistics.com/2012/03/do-more-with-dates-and-times-in-r-with-lubridate-1-1-0/) package and am the author of _Hands On Programming with R_ and _Data Science with R_, two upcoming books from O'Reilly Media. I have a Ph.D. in Statistics and specialize in the theory of data science. I enjoy:

- * __Teaching__ - I design and teach workshops for RStudio. Before that, I taught the popular Statistics 405 course at Rice University.
- * __Statistics__ - I have an M.A. in Statistics from Harvard University and a Ph.D. in Statistics from Rice University.
- * __Teaching statistics__ - I have travelled as far as New Zealand, the birthplace of R, to research better ways to teach statistics.

Garrett Grolemund

← Header

Data Scientist and Master Instructor

garrett@rstudio.com

[rstudio.com](www.rstudio.com)

Orlando, FL

Expertise

I wrote the popular [lubridate](http://www.r-statistics.com/2012/03/do-more-with-dates-and-times-in-r-with-lubridate-1-1-0/) package and am the author of _Hands On Programming with R_ and _Data Science with R_, two upcoming books from O'Reilly Media. I have a Ph.D. in Statistics and specialize in the theory of data science. I enjoy:

- * __Teaching__ - I design and teach workshops for RStudio. Before that, I taught the popular Statistics 405 course at Rice University.
- * __Statistics__ - I have an M.A. in Statistics from Harvard University and a Ph.D. in Statistics from Rice University.
- * __Teaching statistics__ - I have travelled as far as New Zealand, the birthplace of R, to research better ways to teach statistics.

Garrett Grolemund
Data Scientist and Master Instructor
garrett@rstudio.com
[rstudio.com](www.rstudio.com)
Orlando, FL

Expertise

italics



I wrote the popular [lubridate](http://www.r-statistics.com/2012/03/do-more-with-dates-and-times-in-r-with-lubridate-1-1-0/) package and am the author of **_Hands On Programming with R_** and **_Data Science with R_**, two upcoming books from O'Reilly Media. I have a Ph.D. in Statistics and specialize in the theory of data science. I enjoy:

- * **_Teaching_** - I design and teach workshops for RStudio. Before that, I taught the popular Statistics 405 course at Rice University.
- * **_Statistics_** - I have an M.A. in Statistics from Harvard University and a Ph.D. in Statistics from Rice University.
- * **_Teaching statistics_** - I have travelled as far as New Zealand, the birthplace of R, to research better ways to teach statistics.

Garrett Grolemund
Data Scientist and Master Instructor
garrett@rstudio.com
[rstudio.com](www.rstudio.com)
Orlando, FL

Expertise

I wrote the popular [lubridate](http://www.r-statistics.com/2012/03/do-more-with-dates-and-times-in-r-with-lubridate-1-1-0/) package and am the author of _Hands On Programming with R_ and _Data Science with R_, two upcoming books from O'Reilly Media. I have a Ph.D. in Statistics and specialize in the theory of data science. I enjoy:

- 
- * **Teaching** - I design and teach workshops for RStudio. Before that, I taught the popular Statistics 405 course at Rice University.
 - * **Statistics** - I have an M.A. in Statistics from Harvard University and a Ph.D. in Statistics from Rice University.
 - * **Teaching statistics** - I have travelled as far as New Zealand, the birthplace of R, to research better ways to teach statistics.

Garrett Grolemund
Data Scientist and Master Instructor
garrett@rstudio.com
[rstudio.com](www.rstudio.com)
Orlando, FL

Expertise

I wrote the popular [lubridate](<http://www.R-statistics.com/2012/03/do-more-with-dates-and-times-in-r-with-lubridate-1-1-0/>) package and am the author of _Hands On Programming with R_ and _Data Science with R_, two upcoming books from O'Reilly Media. I have a Ph.D. in Statistics and specialize in the theory of data science. I enjoy:

list

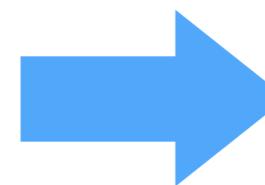


- * __Teaching__ - I design and teach workshops for RStudio. Before that, I taught the popular Statistics 405 course at Rice University.
- * __Statistics__ - I have an M.A. in Statistics from Harvard University and a Ph.D. in Statistics from Rice University.
- * __Teaching statistics__ - I have travelled as far as New Zealand, the birthplace of R, to research better ways to teach statistics.

Hyperlinks

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

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



This is a **link**.

Garrett Grolemund

Data Scientist and Master Instructor

garrett@rstudio.com

[rstudio.com](www.rstudio.com)



Orlando, FL

Expertise

I wrote the popular [lubridate](http://www.r-statistics.com/2012/03/do-more-with-dates-and-times-in-r-with-lubridate-1-1-0/) package and am the author of _Hands On Programming with R_ and _Data Science with R_, two upcoming books from O'Reilly Media. I have a Ph.D. in Statistics and specialize in the theory of data science. I enjoy:

- * __Teaching__ - I design and teach workshops for RStudio. Before that, I taught the popular Statistics 405 course at Rice University.
- * __Statistics__ - I have an M.A. in Statistics from Harvard University and a Ph.D. in Statistics from Rice University.
- * __Teaching statistics__ - I have travelled as far as New Zealand, the birthplace of R, to research better ways to teach statistics.

Horizontal Rule

Use three or more asterisks to make a horizontal rule (a horizontal dividing line).



Garrett Grolemund
Data Scientist and Master Instructor
garrett@rstudio.com
[rstudio.com](www.rstudio.com)
Orlando, FL

Horizontal Rule

Expertise

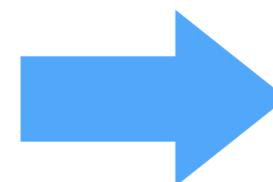
I wrote the popular [lubridate](http://www.r-statistics.com/2012/03/do-more-with-dates-and-times-in-r-with-lubridate-1-1-0/) package and am the author of _Hands On Programming with R_ and _Data Science with R_, two upcoming books from O'Reilly Media. I have a Ph.D. in Statistics and specialize in the theory of data science. I enjoy:

- * __Teaching__ - I design and teach workshops for RStudio. Before that, I taught the popular Statistics 405 course at Rice University.
- * __Statistics__ - I have an M.A. in Statistics from Harvard University and a Ph.D. in Statistics from Rice University.
- * __Teaching statistics__ - I have travelled as far as New Zealand, the birthplace of R, to research better ways to teach statistics.

Equations

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

According to
Einstein,
 $E=mc^2$



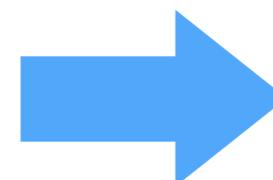
According to
Einstein, $E = mc^2$

Equations

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.

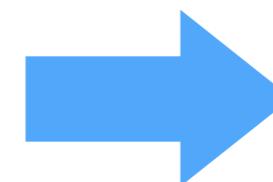
The link text should be

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

```

```

The RStudio logo.



The RStudio logo.

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

A screenshot of a web browser window displaying the Pandoc Markdown documentation. The title bar shows the URL `rmarkdown.rstudio.com/authoring_pandoc_markdown.html`. The page has a dark orange header with navigation links for "R Markdown v2", "Home", "Authoring", "Formats", and "Developer". The main content area features a large heading "Pandoc Markdown" and a bulleted list of Pandoc features. Below the list, there is a section titled "Overview" with a descriptive paragraph and a note about copyright.

Pandoc Markdown

- Overview
- Philosophy
- Paragraphs
- Headers
- Block quotations
- Verbatim (code) blocks
- Line blocks
- Lists
- Horizontal rules
- Tables
- YAML metadata block
- Backslash escapes
- Smart punctuation
- Inline formatting
- Math
- Raw HTML
- Raw TeX
- LaTeX macros
- Links
- Images
- Footnotes
- Citations
- Authors

Overview

Pandoc understands an extended and slightly revised version of John Gruber's original [markdown](#) syntax. This document explains the syntax, noting differences from standard markdown.

This document is a reproduction of most of the [pandoc markdown](#) documentation on the Pandoc website, and is © 2006-2013 John MacFarlane (jgm). All rights reserved.

Embed code

babynames

Popularity of baby names in USA since 1880. From SSA.

1,792,091 records

Five variables: year, sex, name, n, and prop.

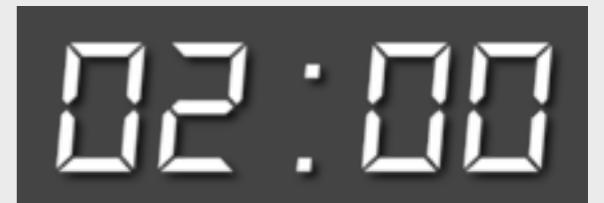
```
# devtools::install_github("babynames", "hadley")
library(babynames)
```

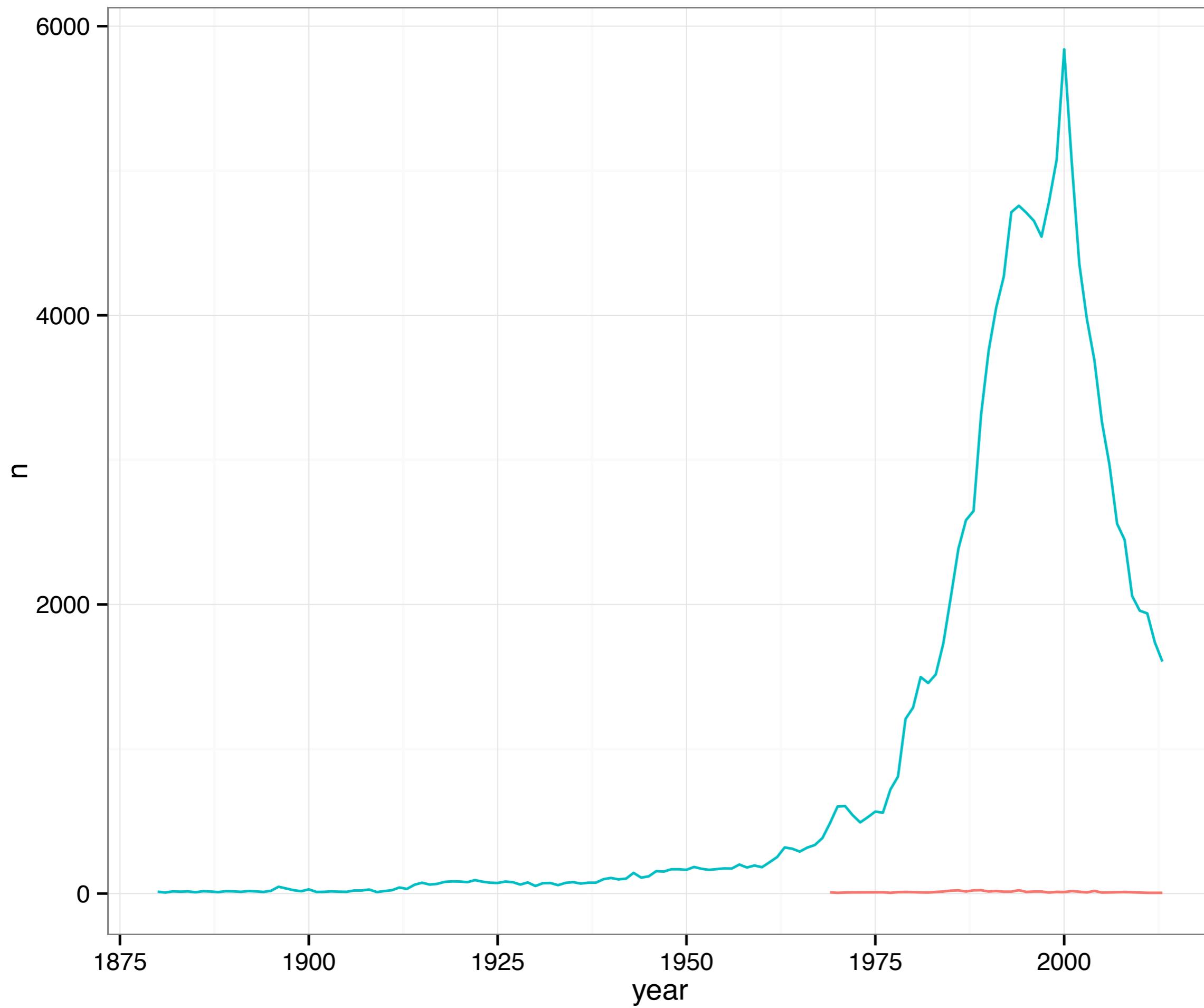
Warm up

Load the babynames package, and graph the history of your name.

```
# devtools::install_github("babynames", "hadley")
library(babynames)
library(ggplot2)

garrett <- subset(babynames, name == "Garrett")
qplot(year, n, data = garrett, geom = "line",
      color = sex) + theme_bw()
```





/var/folders/c3/z24wjmw525710y8yxqtyplxw0000gn/T/RtmpHz4Nsh/Preview-7837485df91.html

Preview-7837485df91.html | Open in Browser | Publish | Find | G

Garrett Grolemund

Data Scientist and Master Instructor
garrett@rstudio.com
rstudio.com
Orlando, FL

Expertise

I wrote the popular [lubridate](#) package and am the author of *Hands On Programming with R* and *Data Science with R*, two upcoming books from O'Reilly Media. I have a Ph.D. in Statistics and specialize in the theory of data science. I enjoy:

- **Teaching** - I design and teach workshops for RStudio. Before that, I taught the popular Statistics 405 course at Rice University.
- **Statistics** - I have an M.A. in Statistics from Harvard University and a Ph.D. in Statistics from Rice University.
- **Teaching statistics** - I have travelled as far as New Zealand, the birthplace of R, to research better ways to teach statistics.

What's in a name?

Here is a graph of the number of people named Garrett.

The graph displays the count of people named Garrett over time. The x-axis represents the year, ranging from 1875 to 2000. The y-axis represents the count, ranging from 0 to 6000. The male population (blue line) shows a sharp increase starting around 1950, peaking at approximately 5800 in 2000. The female population (red line) remains very low, near zero throughout the entire period.

Year	Males (count)	Females (count)
1875	~100	~10
1900	~100	~10
1925	~100	~10
1950	~200	~10
1975	~1000	~10
2000	~5800	~10

My name was more popular than ever in 2000. So far, 125065 people have been named Garrett.

© 2014 RStudio, Inc. All rights reserved.

Code chunks

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 ```.

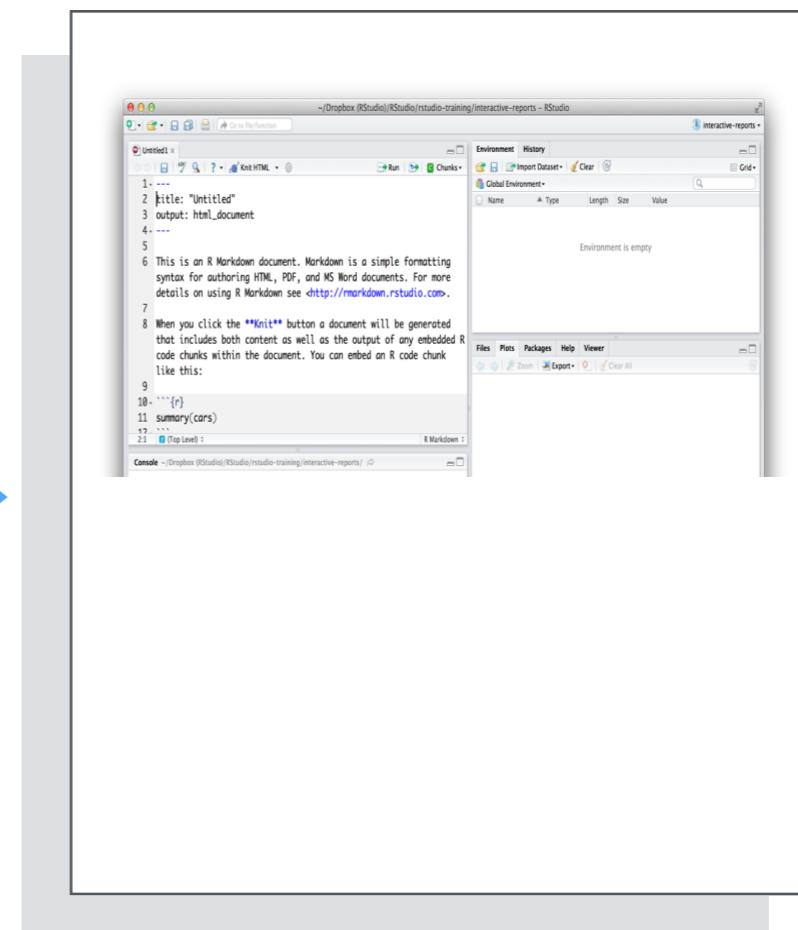
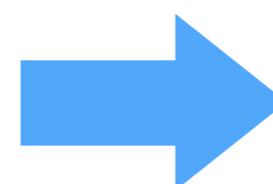
eval

Change the display by adding arguments in the brackets after r.

eval = FALSE prevents the code from being run.

Here's some code

```
```{r eval=FALSE}
dim(iris)
````
```

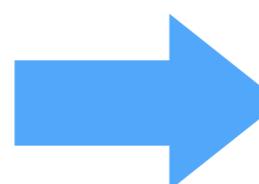


echo

`echo = FALSE` displays the results, but not the code.

Here's some code

```
```{r echo=FALSE}  
dim(iris)
```
```

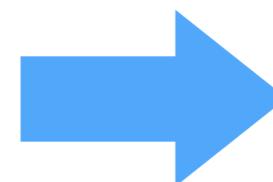


echo

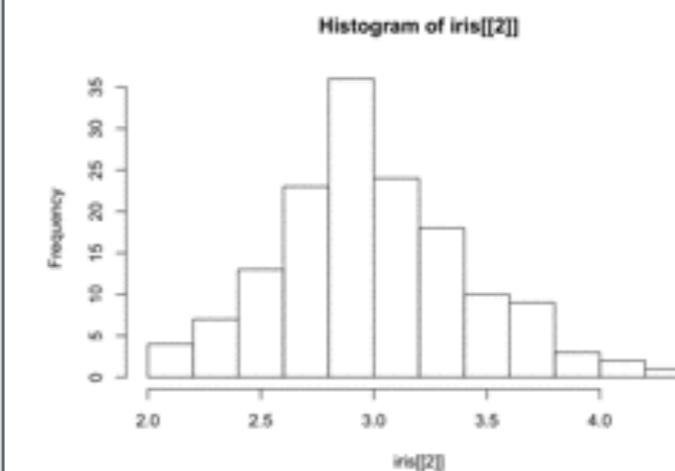
`echo = FALSE` displays the results, but not the code.

Here's a plot

```
```{r echo=FALSE}  
hist(iris[[2]])
```
```



Here's a plot

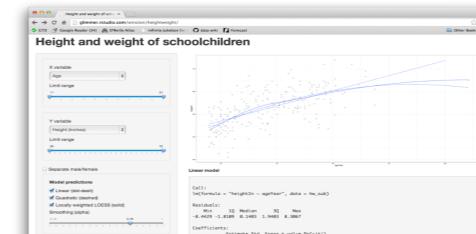
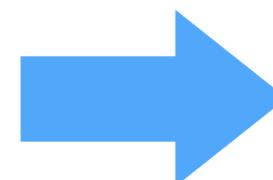


This is very useful
for plots

inline code

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

Two plus two equals `r 2 + 2`.

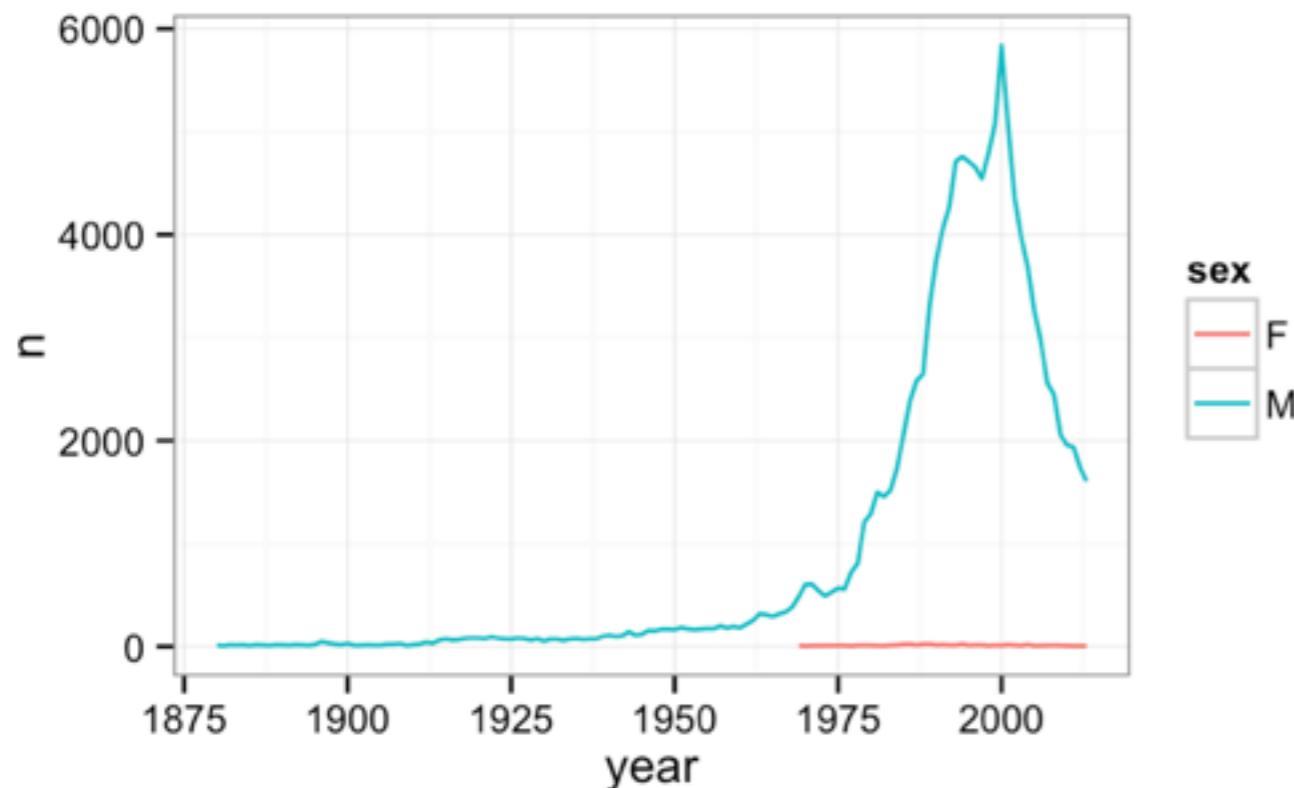


Your turn

Add a third section that describes your name.

What's in a name?

Here is a graph of the number of people named Garrett.



```
library(babynames)
library(ggplot2)
garrett$year[which.max(garrett$n)]
sum(garrett$n)
```

My name was more popular than ever in 2000. So far, 125065 people have been named Garrett.

...

```
## What's in a name?
```

Here is a graph of the number of people named Garrett.

```
```{r echo=FALSE, fig.height=3, fig.width=5}
library(babynames)
library(ggplot2)

garrett <- subset(babynames, name == "Garrett")
qplot(year, n, data = garrett, geom ="line", color = sex) +
theme_bw()
```

```

My name was more popular than ever in `r garrett\$year[which.max(garrett\$n)]`. So far, `r sum(garrett\$n)` people have been named Garrett.

...

```
## What's in a name?
```

Here is a graph of the number of people named Garrett.

```
```{r echo=FALSE, fig.height=3, fig.width=5}
```

```
library(babynames)
```

```
library(ggplot2)
```

```
garrett <- subset(babynames, name == "Garrett")
```

```
qplot(year, n, data = garrett, geom = "line", color = sex) +
```

```
theme_bw()
```

```
```
```



**Code chunk
embeds plot**

My name was more popular than ever in `r garrett\$year[which.max(garrett\$n)]`. So far, `r sum(garrett\$n)` people have been named Garrett.

...

```
## What's in a name?
```

Here is a graph of the number of people named Garrett.

```
```{r echo=FALSE, fig.height=3, fig.width=5}
library(babynames)
library(ggplot2)

garrett <- subset(babynames, name == "Garrett")
qplot(year, n, data = garrett, geom ="line", color = sex) +
theme_bw()
...``
```

**inline code**



My name was more popular than ever in `r garrett  
\$year[which.max(garrett\$n)]`. So far, `r sum(garrett\$n)` people have  
been named Garrett.

# fig.width, fig.height

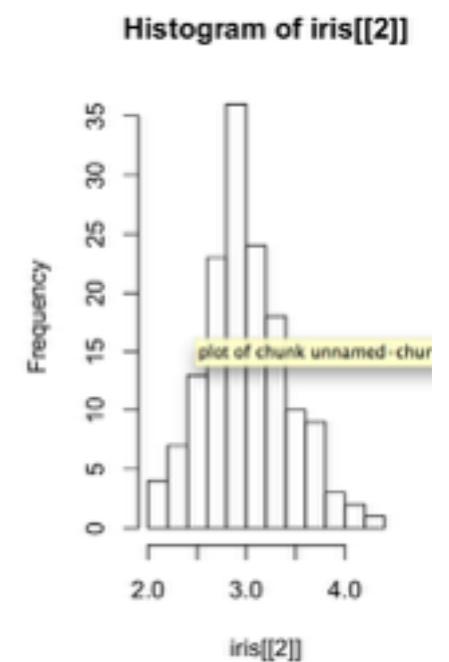
Specify the dimension of plots (in inches) with `fig.width` and `fig.height`.

Separate multiple arguments with commas.

Here's a plot

```
```{r echo=FALSE, fig.width=1, fig.height=2}  
hist(iris[[2]])  
```
```

Here's a plot



# For more on chunk options

[yihui.name/knitr/](http://yihui.name/knitr/)

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

The screenshot shows the homepage of the knitr website. At the top, there's a navigation bar with links for Home, Objects, Options, Hooks, Patterns, and Help. Below the navigation is a large red header with the word "knitr" in white. Underneath the header, there's a sub-header "Elegant, flexible and fast dynamic report generation with R". A decorative image of a skein of yarn and knitting needles is centered below the sub-header. The main content area contains a brief overview of the package's purpose and its relationship to other packages like Sweave and R2HTML.

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).

The screenshot shows a page from the rmarkdown.rstudio.com website titled "R Markdown v2" under the "R Code Chunks" section. It starts with a heading "Basic Usage" and a note about using R code chunks to render output or display code. It includes examples of R code blocks with the "echo=TRUE" option. Below that, it shows how to display output without the underlying code using "echo=FALSE". It also explains how to render plots while omitting the code used to generate them using "echo=TRUE" and "eval=FALSE". The page concludes with a section on "Table Output".

R code chunks can be used as a means render R output into documents or to simply display code for illustration. Here is a simple R code chunk that will result in both the code and it's output being included:

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

To display the output of a code chunk but not the underlying R code, you specify the `echo=FALSE` option:

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

Note that R code chunks can also be used to render plots. To display a plot while omitting the code used to generate the plot you'd do this:

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

To display R code without evaluating it, you specify the `eval=FALSE` chunk option:

```
```{r, eval=FALSE}
summary(cars)
```
```

**Table Output**

By default data frames and matrixes are output as they would be in the R terminal (in a monospaced font). However, if you prefer that data be displayed with additional formatting you can use the `knitr::kable` function. For

# **packrat**

## **(Reproducible Research)**

# packrat

<http://rstudio.github.io/packrat/>

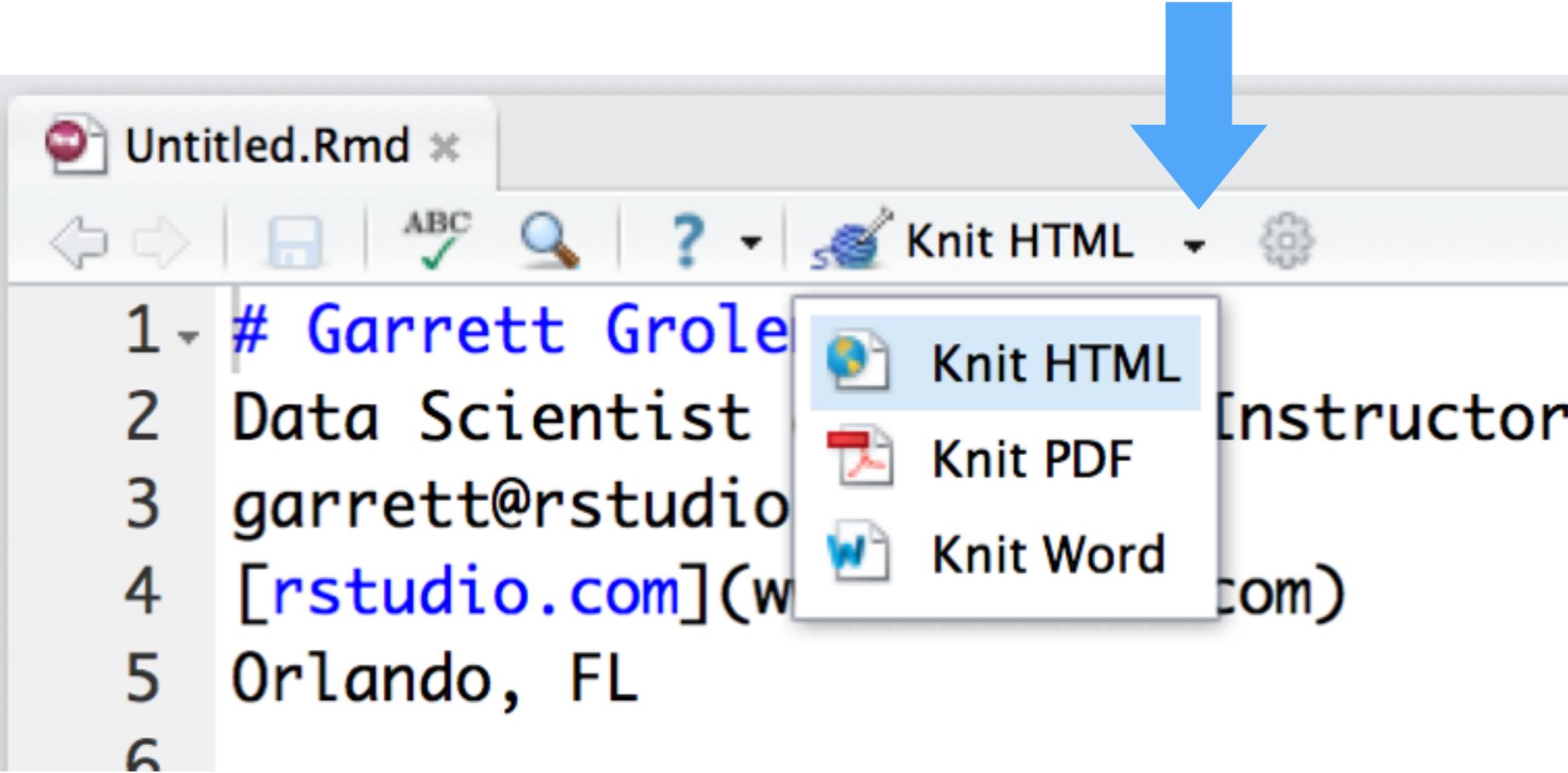
Packrat is a dependency management system for R. Use packrat to make your R projects more:

- **Isolated:** Installing a new or updated package for one project won't break your other projects, and vice versa. That's because packrat gives each project its own private package library.
- **Portable:** Easily transport your projects from one computer to another, even across different platforms. Packrat makes it easy to install the packages your project depends on.
- **Reproducible:** Packrat records the exact package versions you depend on, and ensures those exact versions are the ones that get installed wherever you go.

**Multiple  
output formats**

R Markdown can generate three preset file formats from a .Rmd file

- HTML
- PDF (*Must have latex installed on your computer*)
- MS Word (*Must have Word installed on your computer*)



Untitled.html Open in Browser Publish

# Garrett Grolemund

Data Scientist and Master Instructor  
[garrett@rstudio.com](mailto:garrett@rstudio.com)  
[rstudio.com](http://rstudio.com)  
 Orlando, FL

## Expertise

I wrote the popular [lubridate](#) package and am the author of *Hands On Programming with R* and *Data Science with R*, two upcoming books from O'Reilly Media. I have a Ph.D. in Statistics and specialize in the theory of data science. I enjoy:

- **Teaching** - I design and teach workshops for RStudio. Before that, I taught the popular Statistics 405 course at Rice University.
- **Statistics** - I have an M.A. in Statistics from Harvard University and a Ph.D. in Statistics from Rice University.
- **Teaching statistics** - I have travelled as far as New Zealand, the birthplace of R, to research better ways to teach statistics.

## What's in a name?

Here is a graph of the number of people named Garrett.

My name was more popular than ever in 2000. So far, 125065 people have been named Garrett.

# Garrett Grolemund

Data Scientist and Master Instructor  
[garrett@rstudio.com](mailto:garrett@rstudio.com)  
[rstudio.com](http://rstudio.com)  
 Orlando, FL

---

## Expertise

I wrote the popular [lubridate](#) package and am the author of *Hands On Programming with R* and *Data Science with R*, two upcoming books from O'Reilly Media. I have a Ph.D. in Statistics and specialize in the theory of data science. I enjoy:

- **Teaching** - I design and teach workshops for RStudio. Before that, I taught the popular Statistics 405 course at Rice University.
- **Statistics** - I have an M.A. in Statistics from Harvard University and a Ph.D. in Statistics from Rice University.
- **Teaching statistics** - I have travelled as far as New Zealand, the birthplace of R, to research better ways to teach statistics.

---

## What's in a name?

Here is a graph of the number of people named Garrett.

My name was more popular than ever in 2000. So far, 125065 people have been named Garrett.

# Garrett Grolemund

Data Scientist and Master Instructor  
[garrett@rstudio.com](mailto:garrett@rstudio.com)  
[rstudio.com](http://rstudio.com)  
 Orlando, FL

---

## Expertise

I wrote the popular [lubridate](#) package and am the author of *Hands On Programming with R* and *Data Science with R*, two upcoming books from O'Reilly Media. I have a Ph.D. in Statistics and specialize in the theory of data science. I enjoy:

- **Teaching** - I design and teach workshops for RStudio. Before that, I taught the popular Statistics 405 course at Rice University.
- **Statistics** - I have an M.A. in Statistics from Harvard University and a Ph.D. in Statistics from Rice University.
- **Teaching statistics** - I have travelled as far as New Zealand, the birthplace of R, to research better ways to teach statistics.

---

## What's in a name?

Here is a graph of the number of people named Garrett.

My name was more popular than ever in 2000. So far, 125065 people have been named Garrett.

# HTML

# PDF

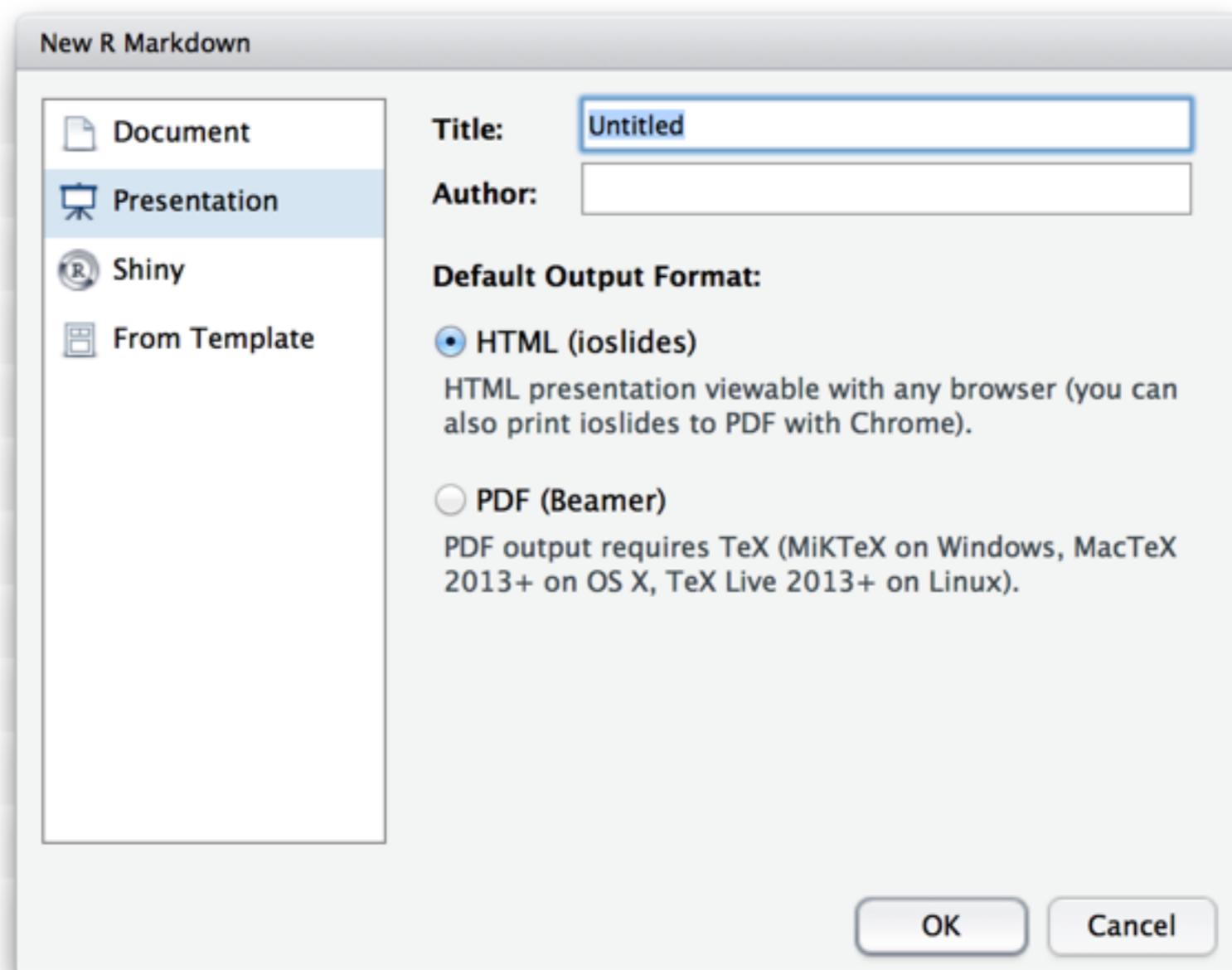
# MS Word

\* requires *tex*

\* requires "Word"

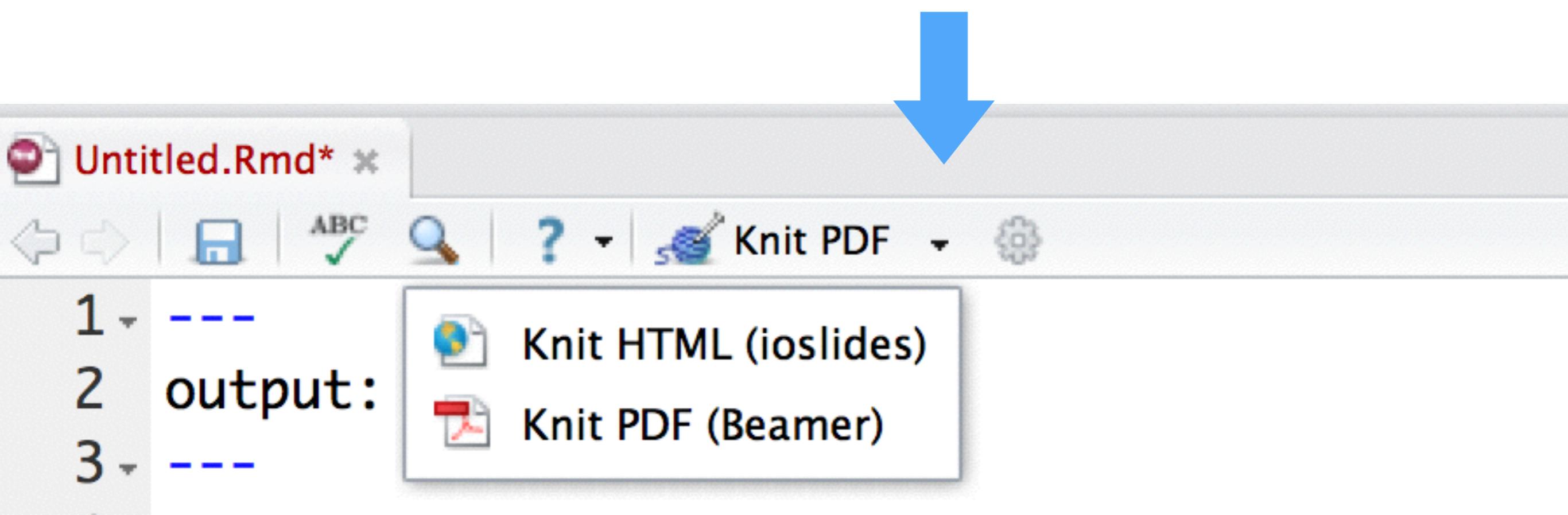
# Slide shows

Build slide presentations from a .Rmd  
(each \*\*\* denotes a new slide).



# Slide shows

Build slide presentations from a .Rmd  
(each \*\*\* denotes a new slide).



Data Scientist and Master Instructor  
garrett@rstudio.com  
rstudio.com  
Orlando, FL

## Garrett Grolemund

### Expertise

I wrote the popular [lubridate](#) package and am the author of *Hands On Programming with R* and *Data Science with R*, two upcoming books from O'Reilly Media. I have a Ph.D. in Statistics and specialize in the theory of data science. I enjoy:

- Teaching - I design and teach workshops for RStudio. Before that, I taught the popular Statistics 405 course at Rice University.
- Statistics - I have an M.A. in Statistics from Harvard University and a Ph.D. in Statistics from Rice University.
- Teaching statistics - I have travelled as far as New Zealand, the birthplace of R, to research better ways to teach statistics.

Garrett Grolemund

Data Scientist and Master Instructor  
garrett@rstudio.com  
rstudio.com  
Orlando, FL

Navigation icons: back, forward, search, etc.

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

ioslides  
(HTML)

### Expertise

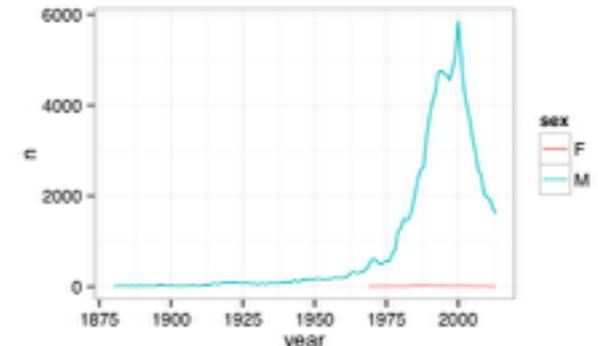
I wrote the popular [lubridate](#) package and am the author of *Hands On Programming with R* and *Data Science with R*, two upcoming books from O'Reilly Media. I have a Ph.D. in Statistics and specialize in the theory of data science. I enjoy:

- ▶ **Teaching** - I design and teach workshops for RStudio. Before that, I taught the popular Statistics 405 course at Rice University.
- ▶ **Statistics** - I have an M.A. in Statistics from Harvard University and a Ph.D. in Statistics from Rice University.
- ▶ **Teaching statistics** - I have travelled as far as New Zealand, the birthplace of R, to research better ways to teach statistics.

Navigation icons: back, forward, search, etc.

### What's in a name?

Here is a graph of the number of people named Garrett.



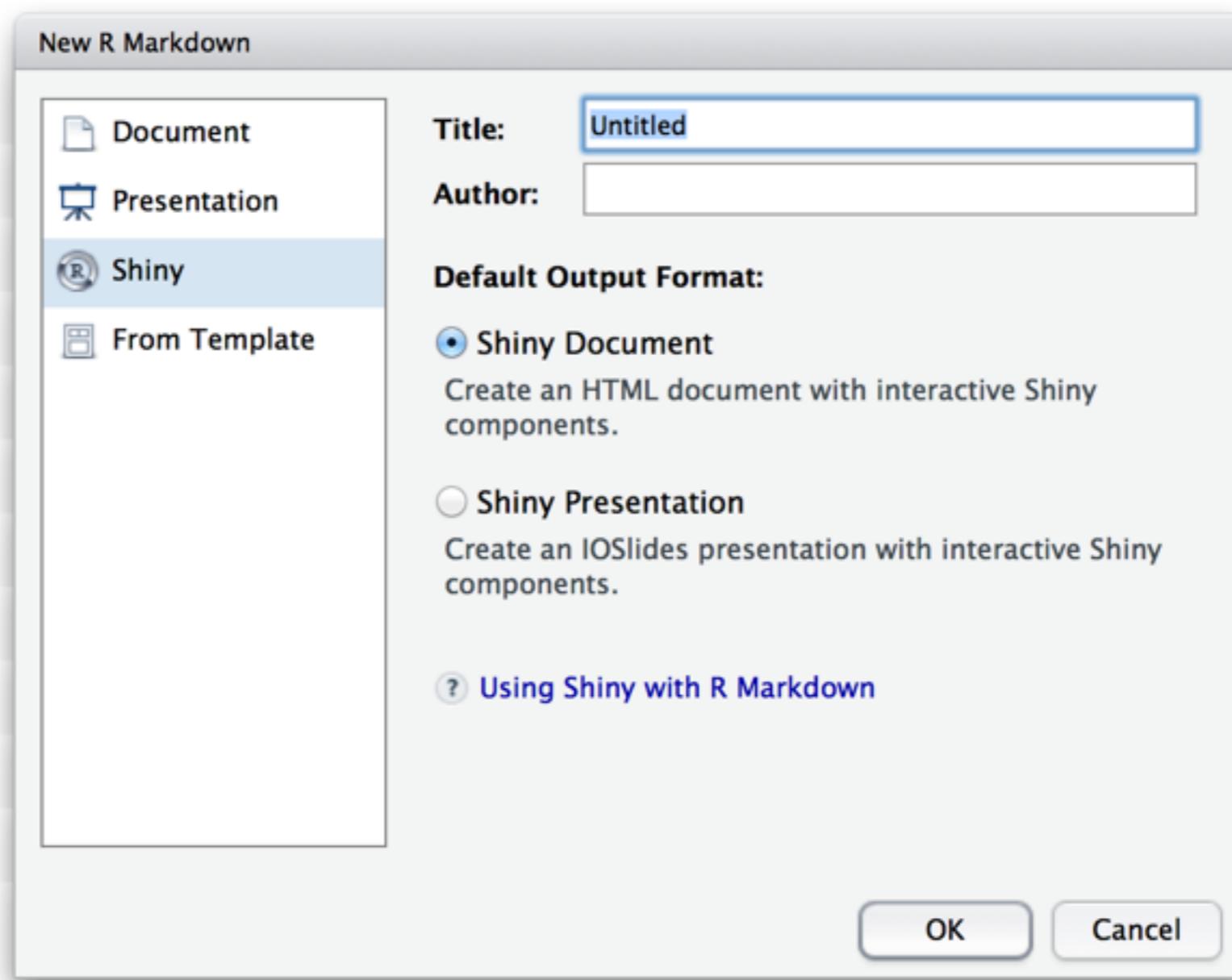
My name was more popular than ever in 2000. So far, 125065 people have been named Garrett.

Navigation icons: back, forward, search, etc.

Beamer  
(PDF)

# Interactive Documents

Select Shiny to add live Shiny elements to code chunks



**Shiny**

# Goal: make graphs and text *interactive*

**What's in a name?**

Please select a name to explore.

Name:

Garrett

Here is a graph of the number of people named Garrett.

Garrett was more popular than ever in 2000. So far, 125065 people have been named Garrett.

**What's in a name?**

Please select a name to explore.

Name:

Hadley

Here is a graph of the number of people named Hadley.

Hadley was more popular than ever in 2013. So far, 16866 people have been named Hadley.

# Shiny

An R package that does two things

1. Creates reactive R objects
2. Builds HTML web pages

# Shiny

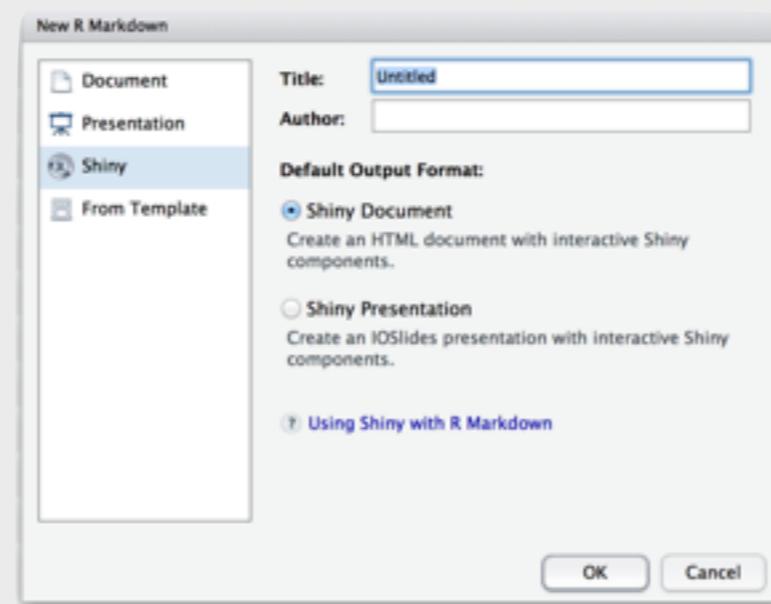
An R package that does two things

- 1. Creates reactive R objects**
2. Builds HTML web pages

# Get Ready

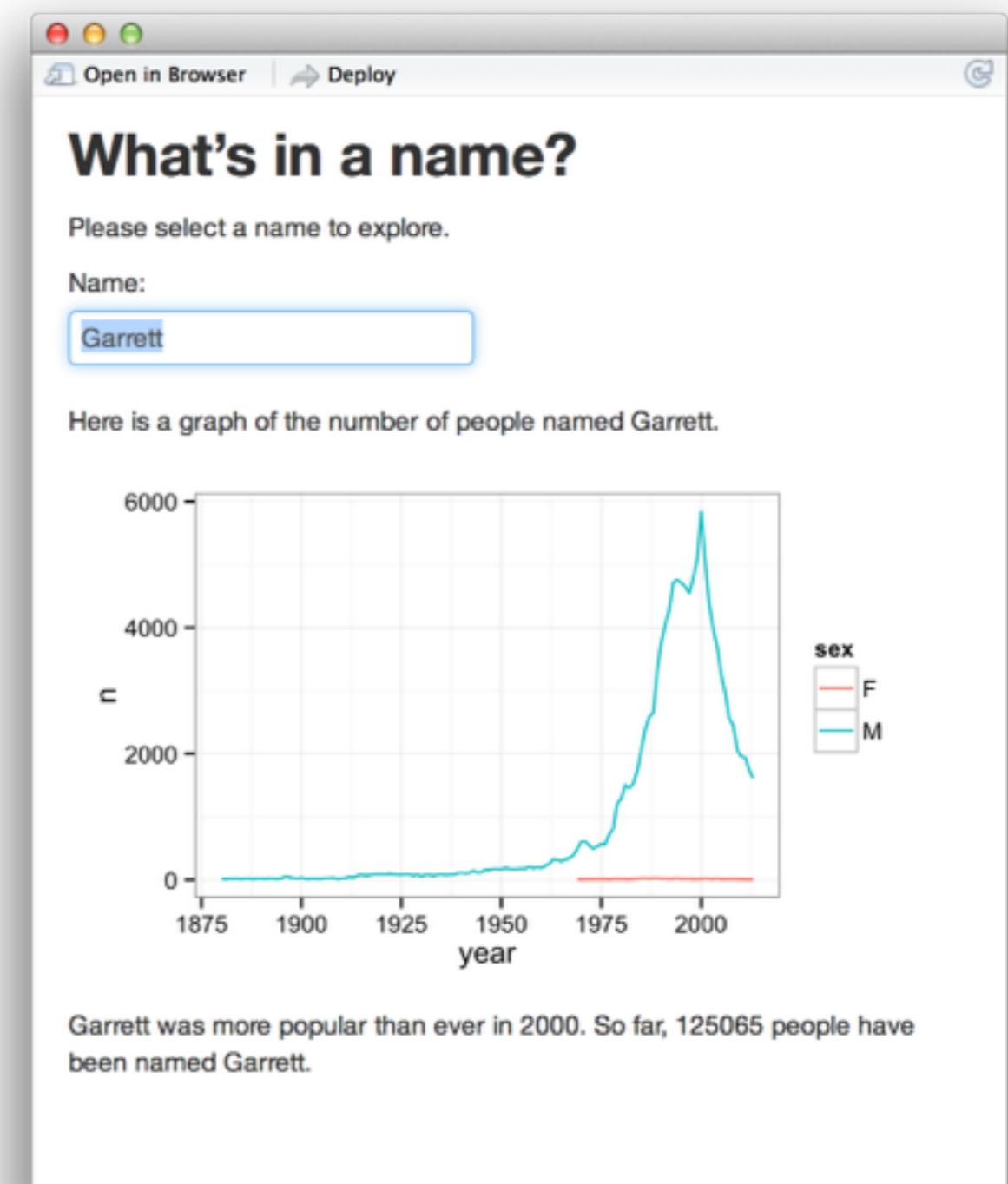
Open an interactive document:

1. File > New File > R Markdown...
2. Select Shiny, then Shiny Document (default)



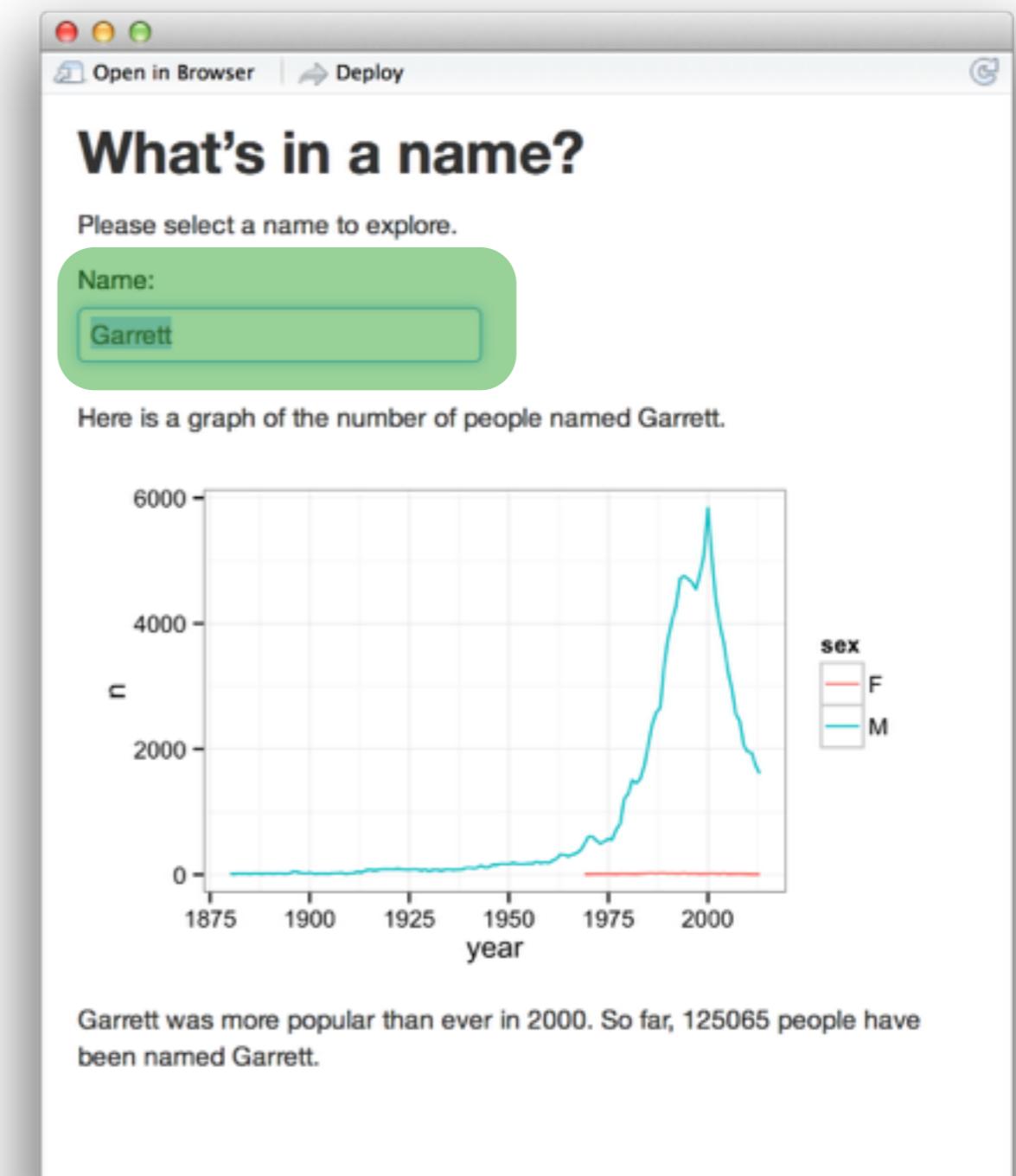
3. Delete everything under the second — — —

# Types of reactive R objects



# Types of reactive R objects

**Widgets** - let users set a value by typing, clicking, etc.

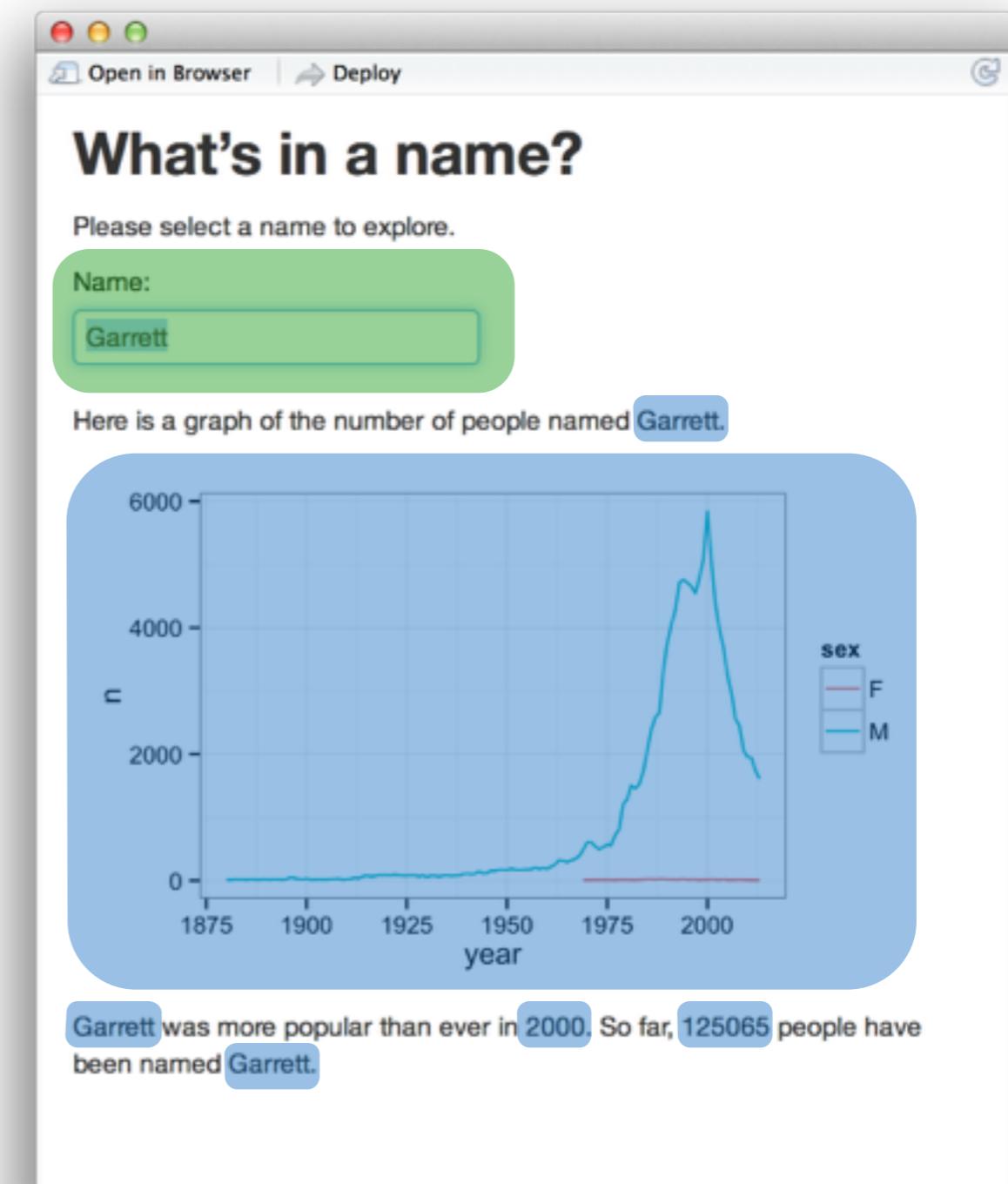
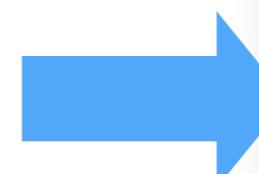


# Types of reactive R objects

**Widgets** - let users set a value by typing, clicking, etc.



**Rendered** - respond whenever a widget value changes



[Open in Browser](#) [Deploy](#)

## What's in a name?

Please select a name to explore.

Name:

Here is a graph of the number of people named Garrett.

The graph shows the number of people named Garrett over time, categorized by sex (F for Female, M for Male). The x-axis represents the year from 1875 to 2013, and the y-axis represents the count of people, ranging from 0 to 6000. The male population (blue line) shows a significant peak around 2000, reaching approximately 5800. The female population (red line) remains very low, near zero throughout the period.

Garrett was more popular than ever in 2000. So far, 125065 people have been named Garrett.

[Open in Browser](#) [Deploy](#)

## What's in a name?

Please select a name to explore.

Name:

Here is a graph of the number of people named Hadley.

The graph shows the number of people named Hadley over time, categorized by sex (F for Female, M for Male). The x-axis represents the year from 1925 to 2013, and the y-axis represents the count of people, ranging from 0 to 2000. The female population (red line) shows a sharp increase starting around 2000, peaking at approximately 2500 in 2013. The male population (blue line) remains very low, near zero throughout the period.

Hadley was more popular than ever in 2013. So far, 16866 people have been named Hadley.

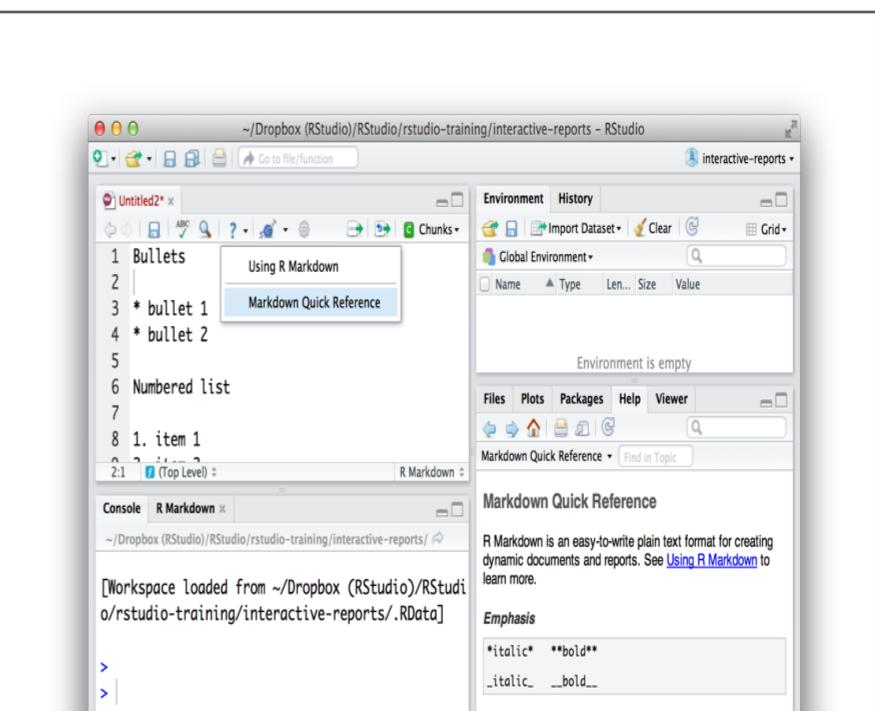
# Widgets

# Widgets

Create a widget with a widget function. Since the function is R code, you'll need to put it in a code chunk.

```
A select box widget
```{r echo=FALSE}
selectInput(name = "choice",
            label = "Pick one, Any one",
            choices = c("Choice 1",
                      "Choice 2",
                      "Choice 3"))
```

```



| function        | widget                                     |
|-----------------|--------------------------------------------|
| actionButton    | Action button                              |
| checkboxGroupIn | Group of checkboxes                        |
| checkboxInput   | Single checkbox                            |
| dateInput       | Calendar to aid date selection             |
| dateRangeInput  | Pair of calendars for selecting date range |
| fileInput       | File upload control wizard                 |
| helpText        | Help text to accompany other widgets       |
| numericInput    | Field to enter numbers                     |
| radioButtons    | Set of radio buttons                       |
| selectInput     | Box with choices to select from            |
| sliderInput     | Slider bar                                 |
| submitButton    | Submit button                              |
| textInput       | Field to enter text                        |

[shiny.rstudio.com/gallery/widgets](http://shiny.rstudio.com/gallery/widgets)

The screenshot shows a web browser window displaying the Shiny Widgets Gallery. The title bar reads "~/Documents/widgets - Shiny" and the address bar shows "http://127.0.0.1:4587". The main content area features a blue header with the text "Shiny Widgets Gallery". Below the header, a descriptive text states: "For each widget below, the Current Value(s) window displays the value that the widget provides to shinyServer. Notice that the values change as you interact with the widgets." The page is divided into six sections, each demonstrating a different type of Shiny input widget:

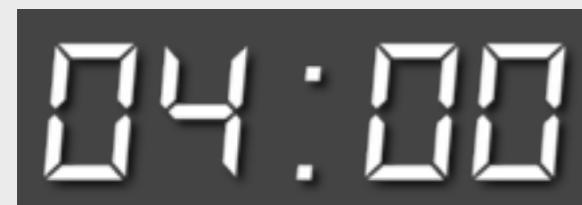
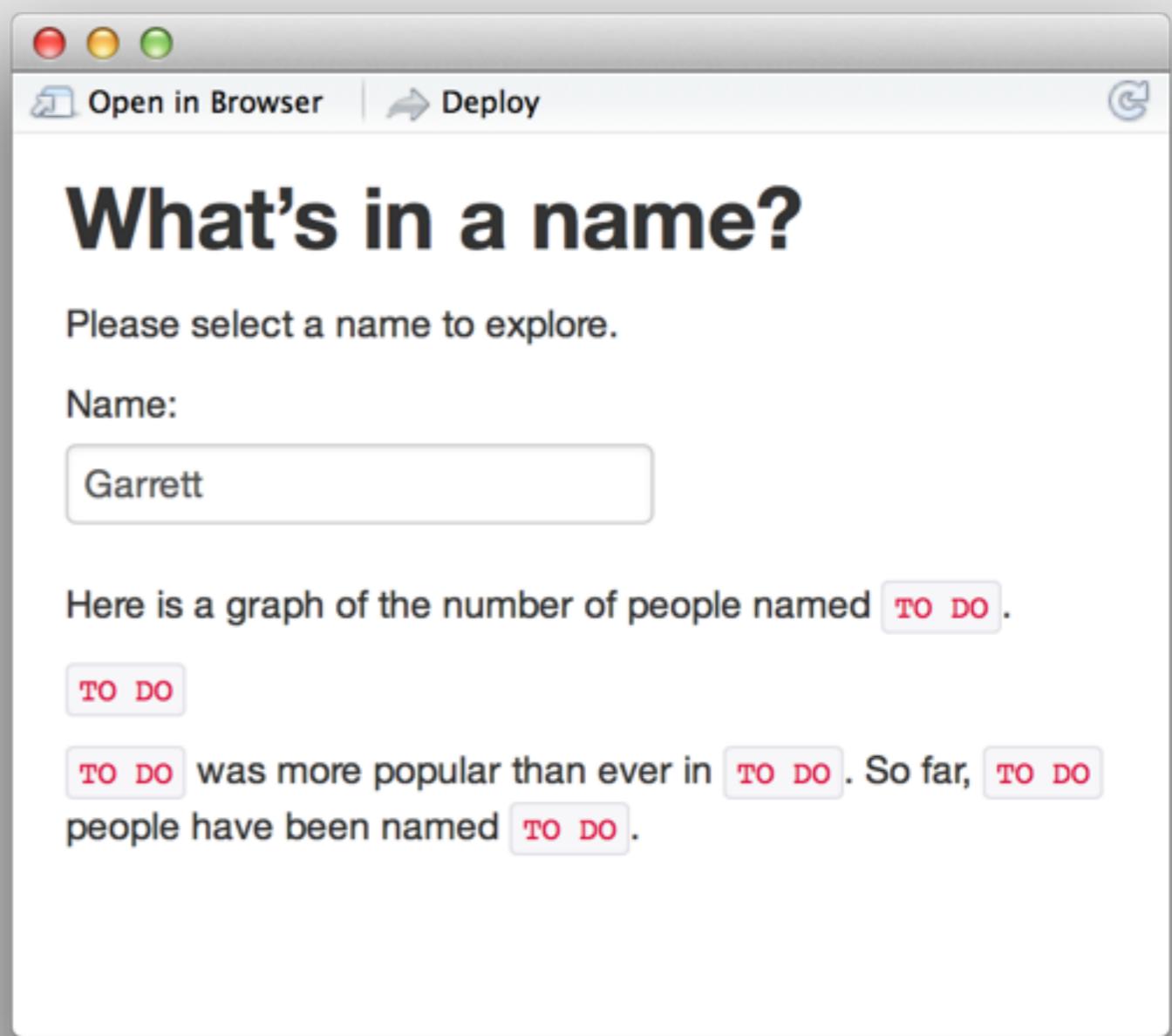
- Action button**: Contains a button labeled "Action". Below it, a "Current Value:" section shows the R code: [1] 0 attr(,"class") [1] "numeric" "shinyActionButtonValue". A "See Code" button is at the bottom.
- Single checkbox**: Shows a checked checkbox labeled "Choice A". Below it, a "Current Value:" section shows the R code: [1] TRUE. A "See Code" button is at the bottom.
- Checkbox group**: Shows three checkboxes: "Choice 1" (checked), "Choice 2" (unchecked), and "Choice 3" (unchecked). Below it, a "Current Values:" section shows the R code: [1] "1". A "See Code" button is at the bottom.
- Date input**: Shows a single date input field containing the value 2014-01-01.
- Date range**: Shows two date input fields: one for the start date (2014-06-20) and one for the end date (2014-06-20).
- File input**: Shows a file input field with the placeholder text "no file selected" and a "Choose File" button.

# Your turn

Write the text of your document and add a widget like the one below (use your own name).

*Look up the widget's help page or gallery example to find its arguments.*

*I added the place holders with `TO DO`.*



---

```
runtime: shiny
output: html_document
```

---

```
What's in a name?
```

Please select a name to explore.

```
```{r echo=FALSE}  
textInput("name", "Name:", value = "Garrett")  
```
```

Here is a graph of the number of people named `TO DO`.

`TO DO`

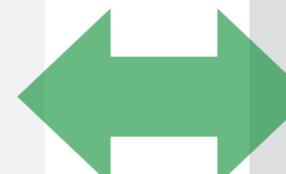
`TO DO` was more popular than ever in `TO DO`. So far, `TO DO` people have been named `TO DO`.

# Reactivity 101

Your widget saves a value in R that you can call with `input$name`.  
For example, you would call the value of this widget with this call.

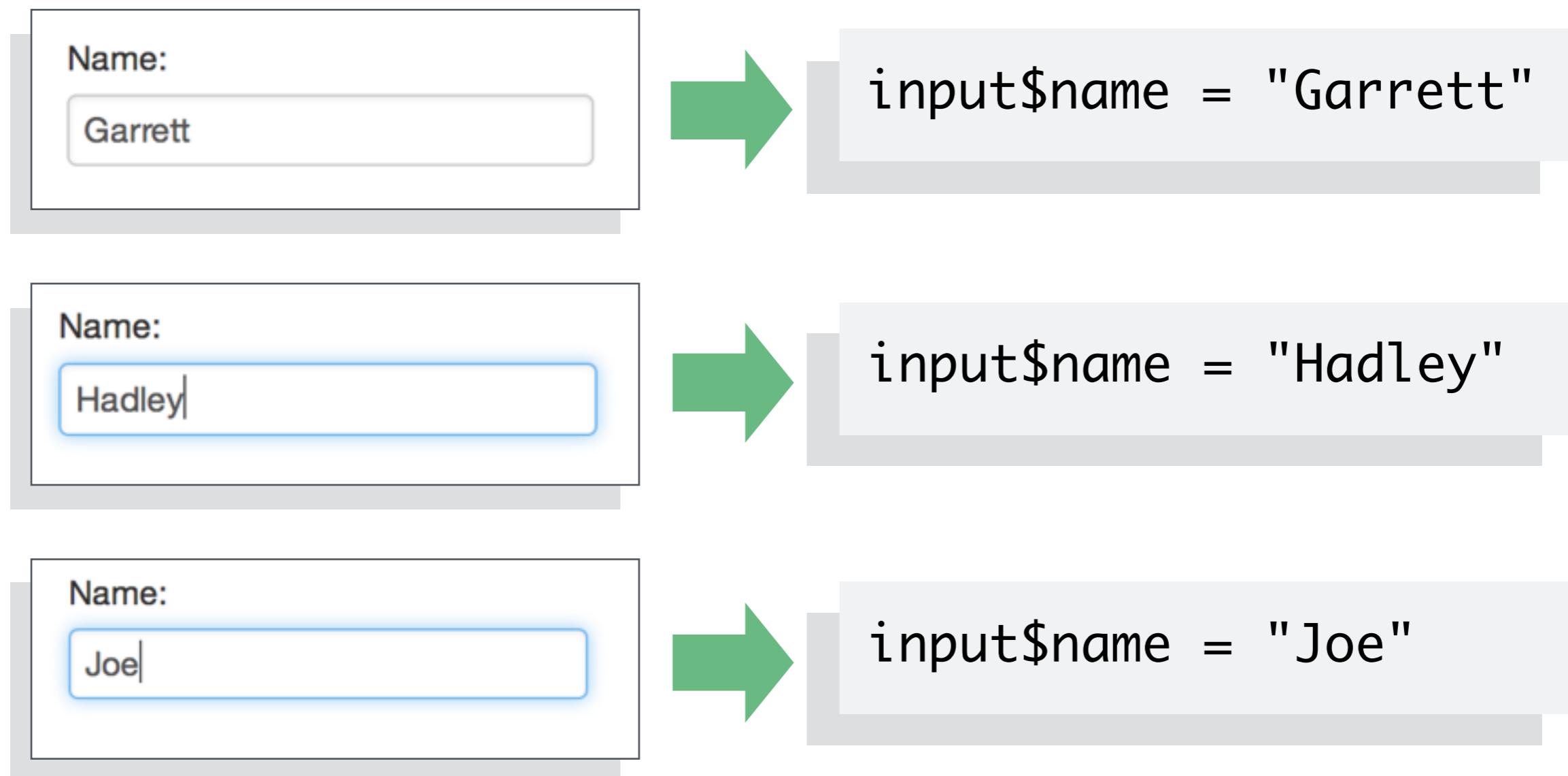
```
textInput(
 inputId = "name",
 label = "Name:",
 value = "Garrett")
```

`input$name`



# Reactivity 101

The widget value changes whenever a user changes the widget.  
*(Any Shiny output that uses the widget value will also change).*



# Reactivity 101

You **CANNOT** call a widget value with a normal R function. (Shiny won't let you).



```
nchar(inputcname)
```

# Reactivity 101

You **CAN** call a widget value when you wrap an R expression with one of *render\**, *reactive*, *isolate*, or *observe*.

```
renderText({nchar(input$name)})
```



```
reactive({nchar(input$name)})
```



```
isolate({nchar(input$name)})
```



```
observe({nchar(input$name)})
```



# Reactivity 101

You **CAN** call a widget value when you wrap an R expression with one of *render\**, *reactive*, *isolate*, or *observe*.

```
renderText({nchar(input$name)})
```



```
reactive({nchar(input$name)})
```



```
isolate({nchar(input$name)})
```



```
observe({nchar(input$name)})
```



# **Rendered outputs**

# render\*

A family of functions that make R objects that use widget values and **update** whenever a widget value changes.

```
renderText({
 names <- subset(babynames, name == input$name)
 sum(names$n)
})
```

Use the function that creates the type of object you wish to make.

function

creates

renderDataTable An interactive table  
(from a data frame, matrix, or other table-like structure)

renderImage

An image (saved as a link to a source file)

renderPlot

A plot

renderPrint

A code block of printed output

renderTable

A table  
(from a data frame, matrix, or other table-like structure)

renderText

A character string

renderUI

a Shiny UI element

# render\*

Each render function takes one argument: the code needed to build the final object. (*Use as many lines as you like*)

```
renderText({
 names <- subset(babynames, name == input$name)
 sum(names$n)
})
```



# Putting it together

Once you make a widget, you can refer to it in later code blocks.

```
```{r echo=FALSE}
textInput("name", "Name:", value =
"Garrett")
```

There were this many people with that name:
```{r echo=FALSE}
library(babynames)
renderText({
  names <- subset(babynames,
    name == input$name)
  sum(names$n)
})
```
```

```

Name:
Garrett

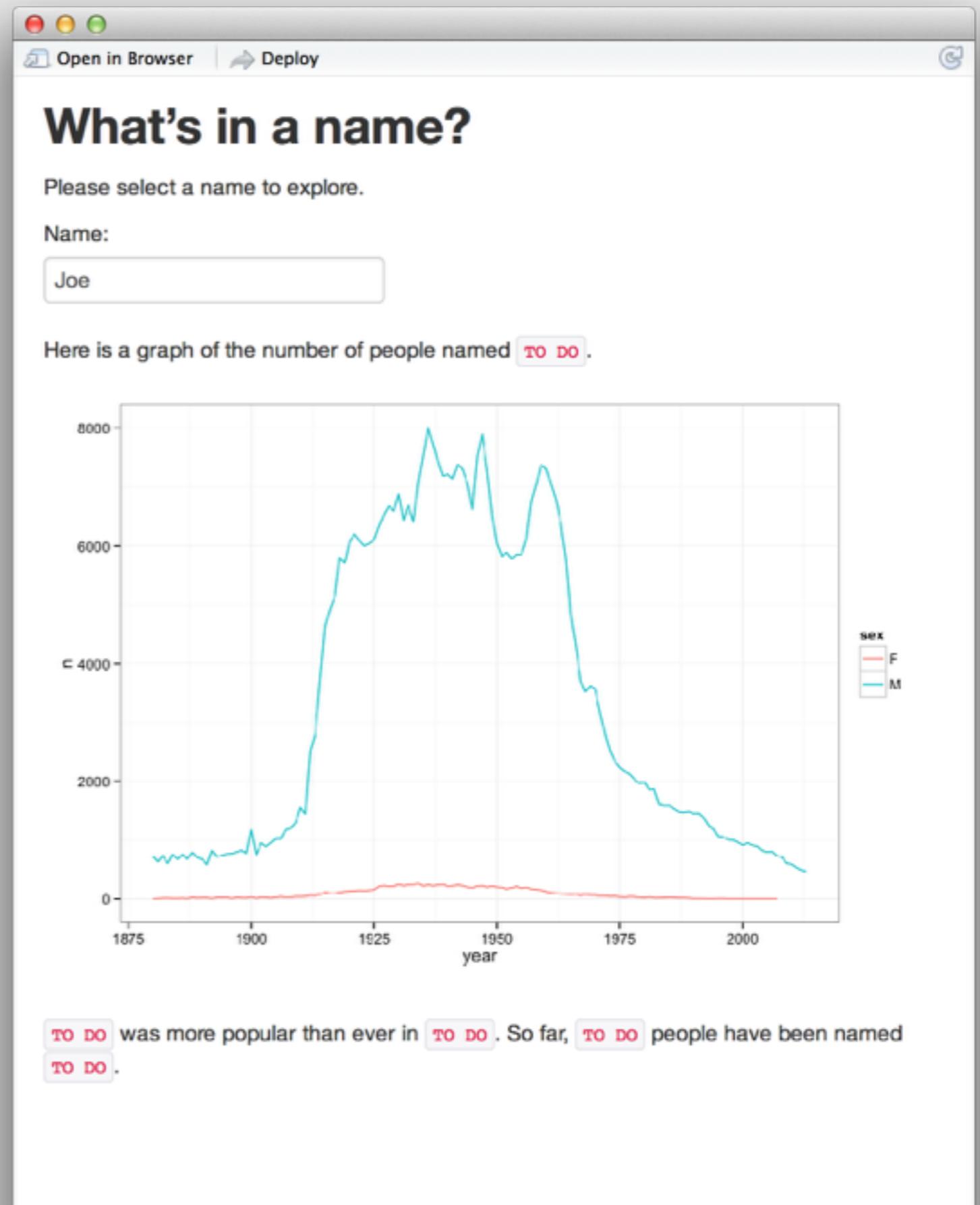
There were this many people with
that name:
125065



Your turn

Add the plot to your document. Make sure it reacts when a user types a new name.

02 : 30



```
---
```

```
runtime: shiny
```

```
output: html_document
```

```
--
```

```
## What's in a name?
```

Please select a name to explore.

```
```{r echo=FALSE}
```

```
textInput("name", "Name:", value = "Garrett")
```

```
```
```

Here is a graph of the number of people named `TO DO`.

```
```{r echo=FALSE, fig.height=3, fig.width=5}
```

```
library(babynames)
```

```
library(ggplot2)
```

```
renderPlot({
```

```
 names <- subset(babynames, name == input$name)
```

```
 qplot(year, n, data = names, geom ="line", color = sex) +
```

```
 theme_bw()
```

```
)
```

```
```
```

`TO DO` was more popular than ever in `TO DO`. So far, `TO DO` people have been named `TO DO`.

Please select a name to explore.

```
```{r echo=FALSE}
textInput("name", "Name:", value = "Garrett")
```
```

```

Here is a graph of the number of people named `TO DO`.

```
```{r echo=FALSE, fig.height=3, fig.width=5}
library(babynames)
library(ggplot2)

renderPlot({
  names <- subset(babynames, name == input$name)
  qplot(year, n, data = names, geom ="line", color = sex) +
    theme_bw()
})
```
```

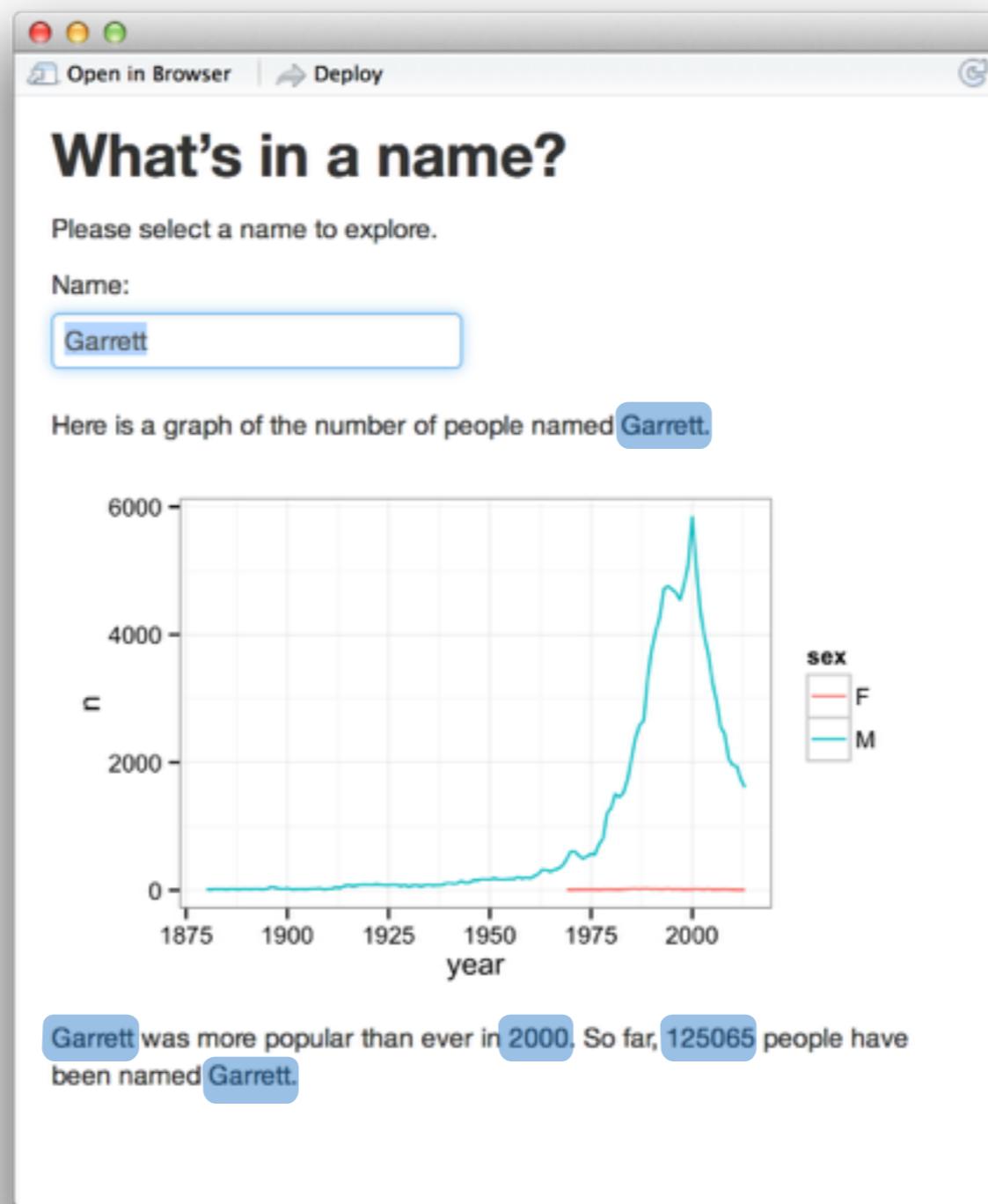
```

`TO DO` was more popular than ever in `TO DO`. So far, `TO DO`

Reactive expressions

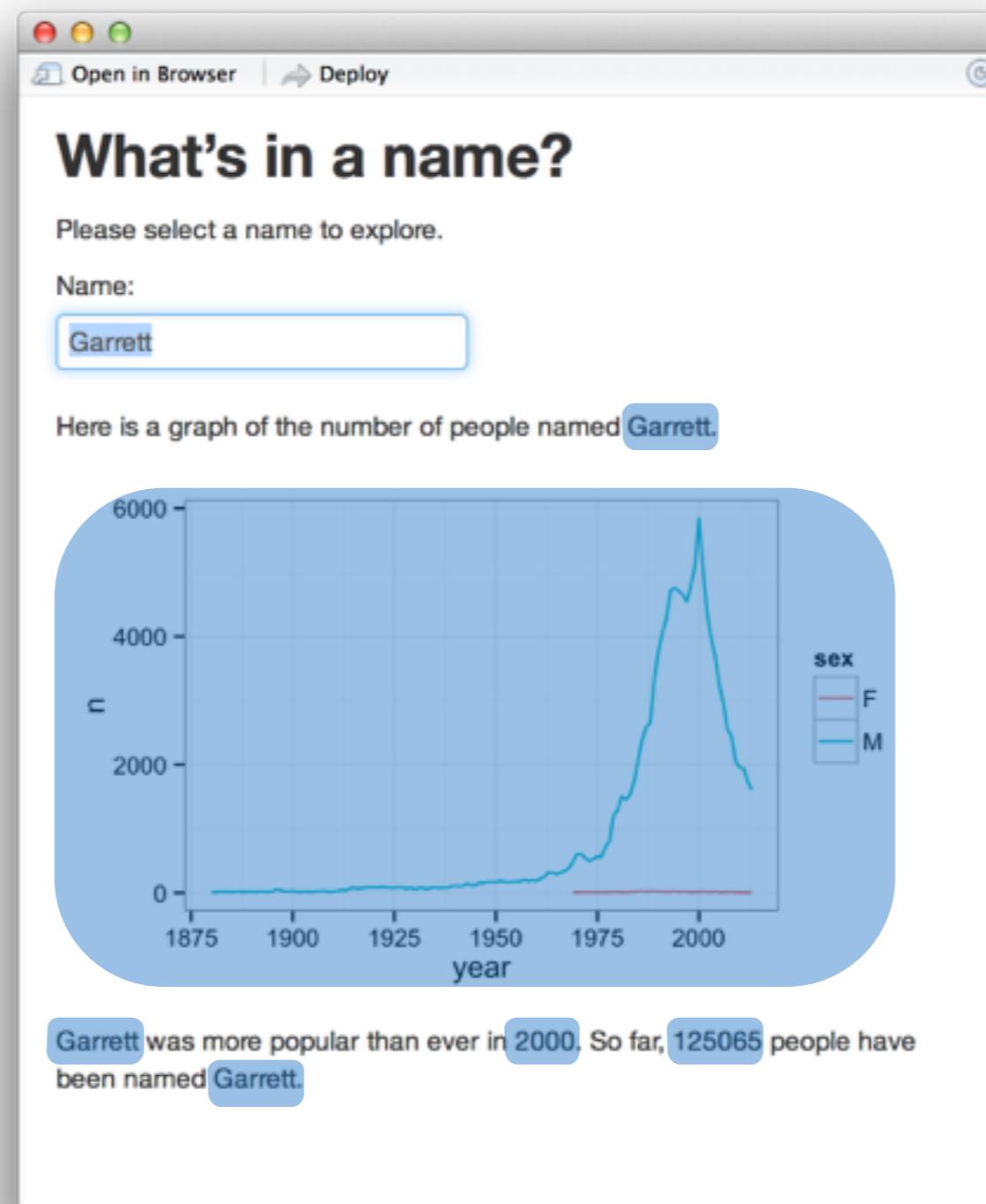
We've built the reactive plot.





Now, let's build the inline reactive text.

Each of these depends on `input$name` in the same way.



One reactive object used in multiple places.

```
names <- subset(babynames, name == input$name)
```

Garrett.

```
names$name[1]
```

Garrett

```
names$name[1]
```

Garrett.

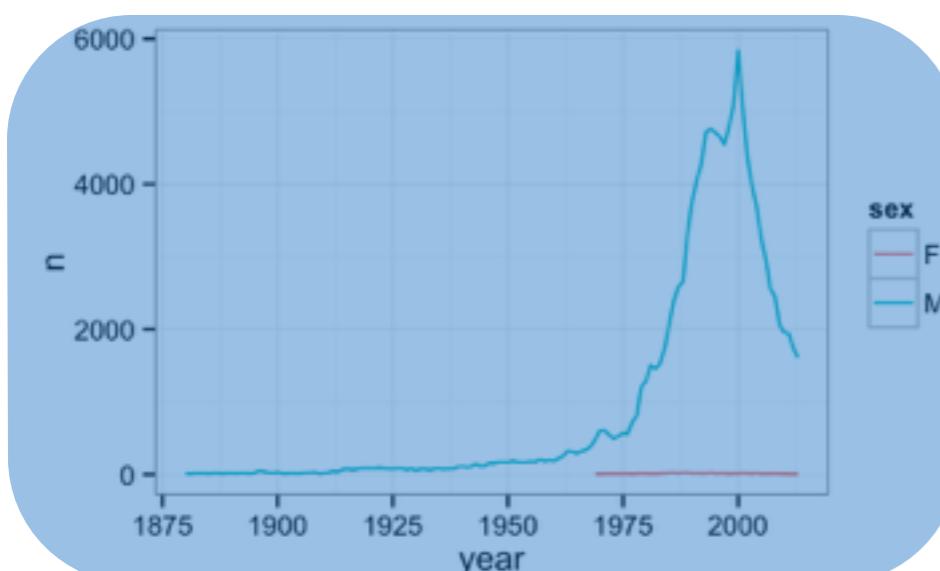
```
names$name[1]
```

2000.

```
names$year[which.max(names$n)]
```

125065

```
sum(names$n)
```



```
qplot(year, n, data = names,  
geom = "line", color = sex) +  
theme_bw()
```

One reactive object used in multiple places.

```
names <- subset(babynames, name == input$name)
```

Garrett.

```
names$name[1]
```

Garrett

```
names$name[1]
```

Garrett.

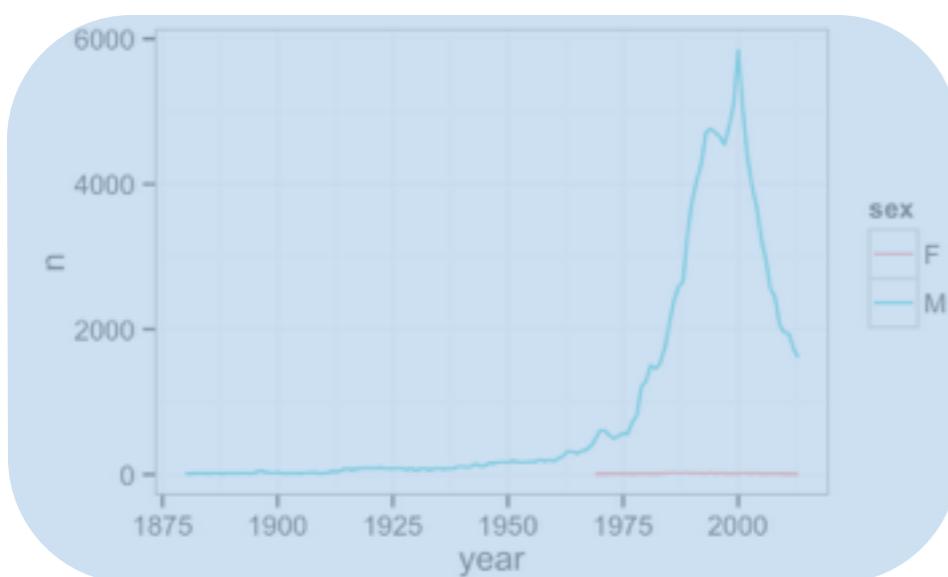
```
names$name[1]
```

2000.

```
names$year[which.max(names$n)]
```

125065

```
sum(names$n)
```



```
qplot(year, n, data = names,  
geom = "line", color = sex) +  
theme_bw()
```

Please select a name to explore.

```
```{r echo=FALSE}
textInput("name", "Name:", value = "Ganesh")
````
```

Here is a graph of the number of people

```
```{r echo=FALSE, fig.height=3, fig.width=6}
library(babynames)
library(ggplot2)
```

```
renderPlot({
 names <- subset(babynames, name == input$name)
 qplot(year, n, data = names, geom ="line", color = sex) +
 theme_bw()
})
```

`TO DO` was more popular than ever in `TO DO`.

Objects created *inside* a render\* function are only available *inside* that render function

Inefficient to repeat this  
in every render\* function

# Reactivity 101

You **CAN** call a widget value when you wrap an R expression with one of *render\**, *reactive*, *isolate*, or *observe*.

```
renderText({nchar(input$name)})
```



```
reactive({nchar(input$name)})
```



```
isolate({nchar(input$name)})
```



```
observe({nchar(input$name)})
```



# Reactive expressions

Take widget values and make a new **reactive** value.

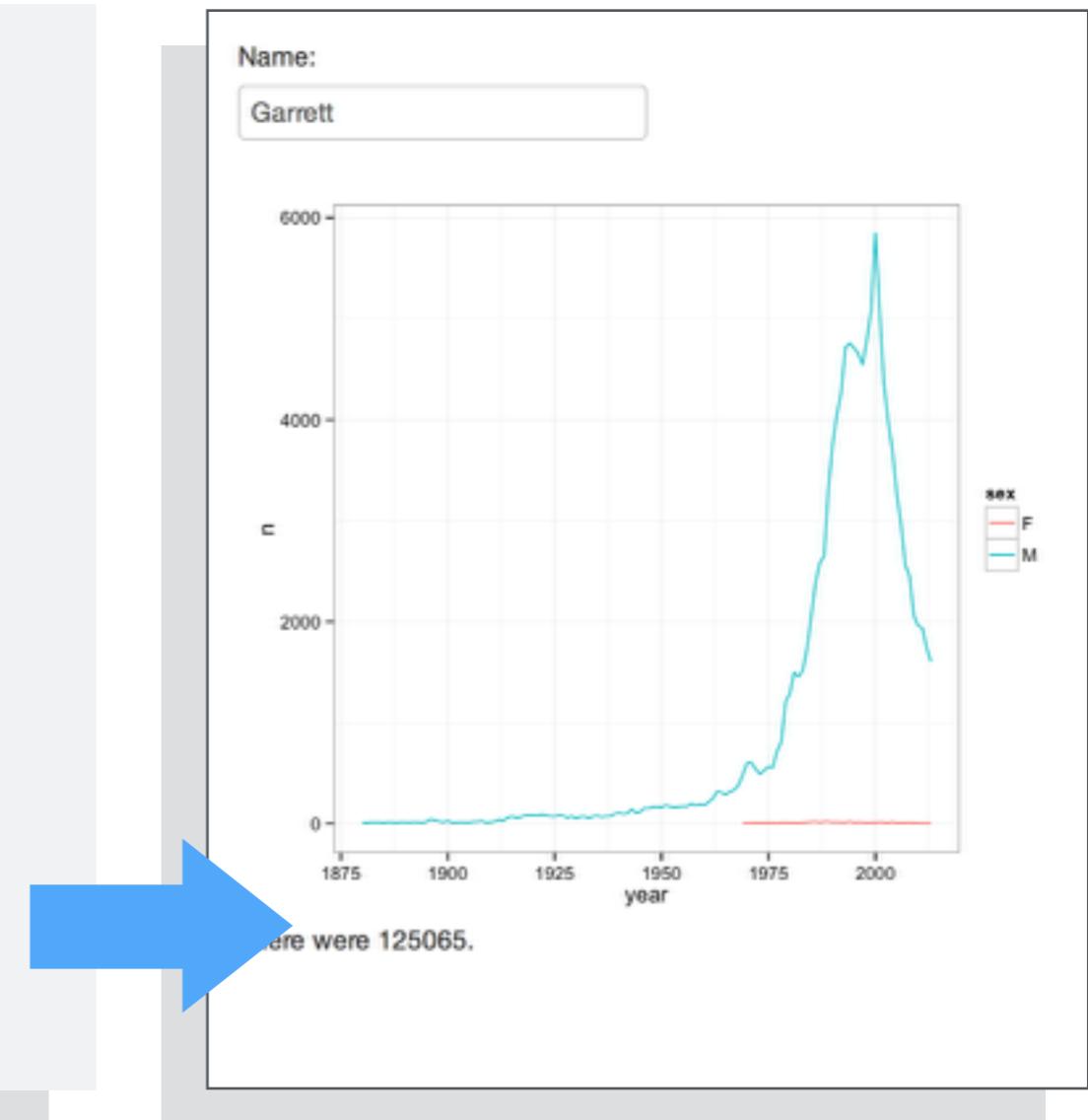
```
names <- reactive({
 subset(babynames, name == input$name)
})
```

The value will change whenever the widget changes

# Reactive expressions

You can use a reactive value in multiple render\* functions.  
Call a reactive values, as if it were a function.

```
```{r echo=FALSE}
textInput("name", "Name:", value = "Garrett")
library(babynames)
names <- reactive({
  subset(babynames, name == input$name)
})
```
```
```{r echo=FALSE}
library(ggplot2)
renderPlot({
 qplot(year, n, data = names(),
 color = sex) + theme_bw()
})
```
```
There were `r renderText(sum(names()$n)`.
```



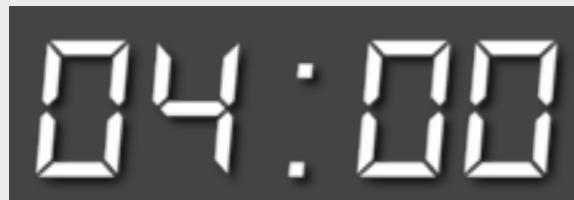
# Reactive expressions are reactive

So you must call reactive values from  
within *render\**, *reactive*, *isolate*, or *observe*.

# Your turn

Finish your reactive document.

1. Create a reactive expression that returns the data set for the selected name.
2. Replace each `TO DO` with an inline rendered output (that uses the reactive expression from step 1).



```

```

```
runtime: shiny
output: html_document

```

```
What's in a name?
```

Please select a name to explore.

```
```{r echo=FALSE}
library(babynames)
textInput("name", "Name:", value = "Garrett")
names <- reactive(subset(babynames, name == input$name))
```

```

Here is a graph of the number of people named `r renderText(input\$name)`.

```
```{r echo=FALSE, fig.height=3, fig.width=5}
library(ggplot2)
renderPlot{
  qplot(year, n, data = names(), geom ="line", color = sex) +
    theme_bw()
}
```

```

`r renderText(input\$name)` was more popular than ever in `r renderText(names()\$year[which.max(names()\$n)])`. So far, `r renderText(sum(names()\$n))` people have been named `r renderText(input\$name)`.

**isolate &  
observe**

# Reactivity 101

You **CAN** call a widget value when you wrap an R expression with one of *render\**, *reactive*, *isolate*, or *observe*.

```
renderText({nchar(input$name)})
```



```
reactive({nchar(input$name)})
```



```
isolate({nchar(input$name)})
```



```
observe({nchar(input$name)})
```



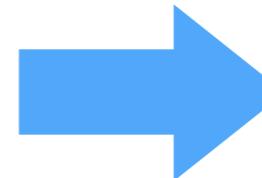
# isolate

Isolate isolates an expression from a widget. The object will **NOT** react when the isolated widget changes.

```
`r renderText(isolate(input$name))` was more popular than ever...
```

Name:

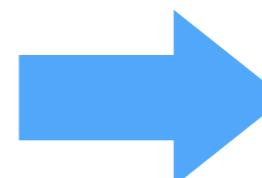
Garrett



Garrett was more popular  
than ever...

Name:

Joe



Garrett was more popular  
than ever...

# observe

`observe` will rerun the R expression each time a widget changes, but `observe` will not return a result.

Use `observe` for the side effects of the R expression.

```
```{r echo=FALSE}
ActionButton("button", "Run Script!")
observe({
  if(input$button != 0)
    source("secret-script.R")
})
```
```

Run Script!

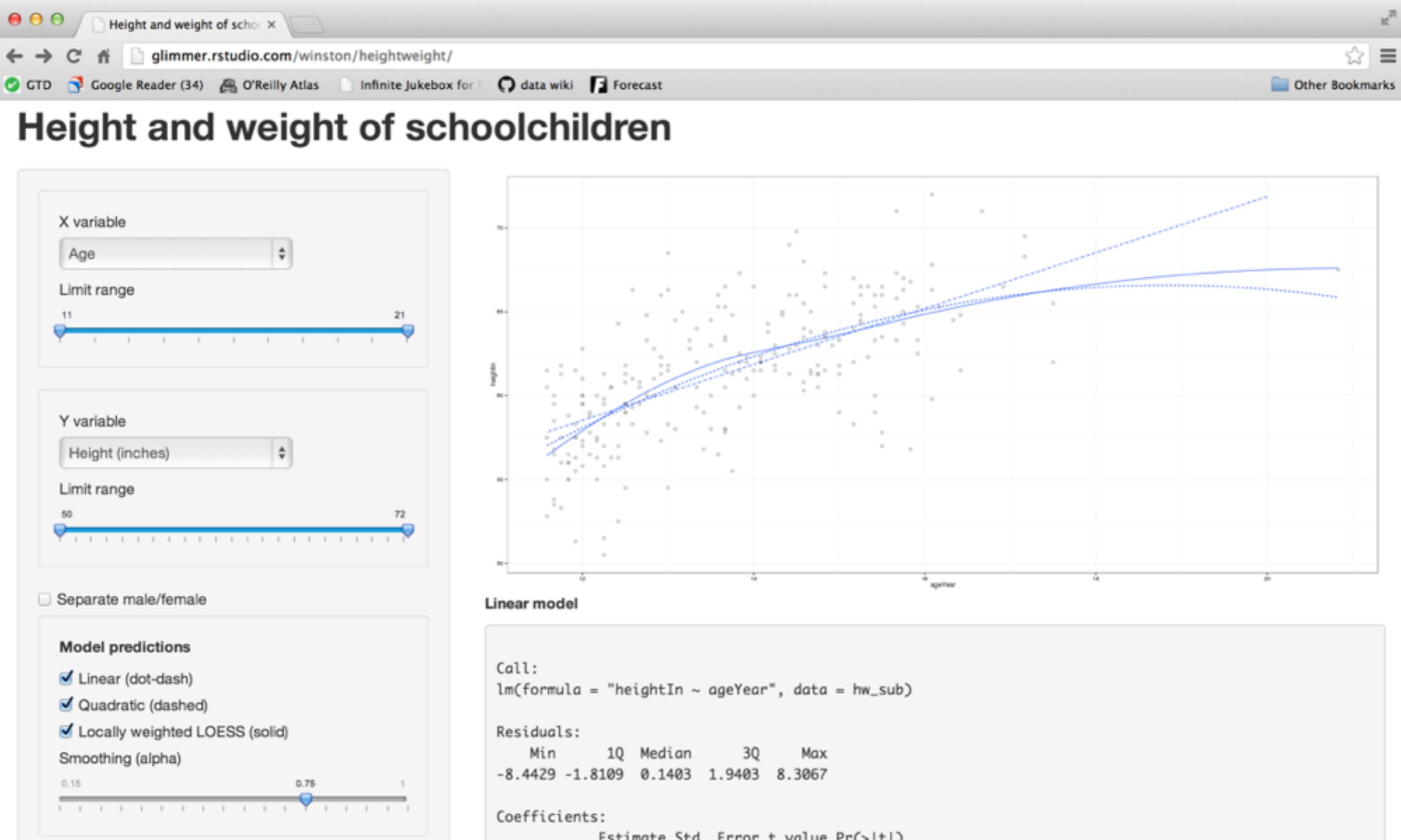


# Shiny

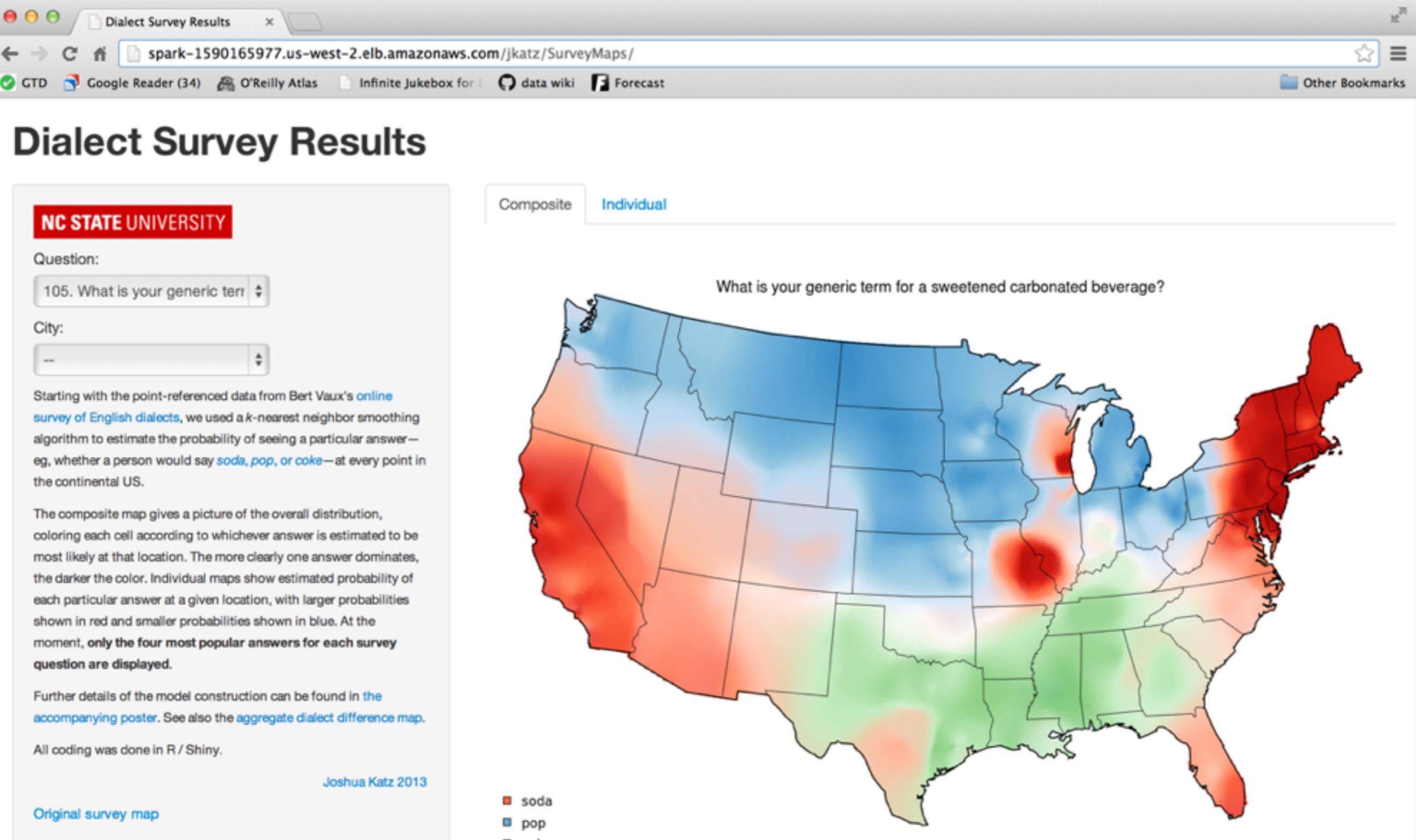
An R package that does two things

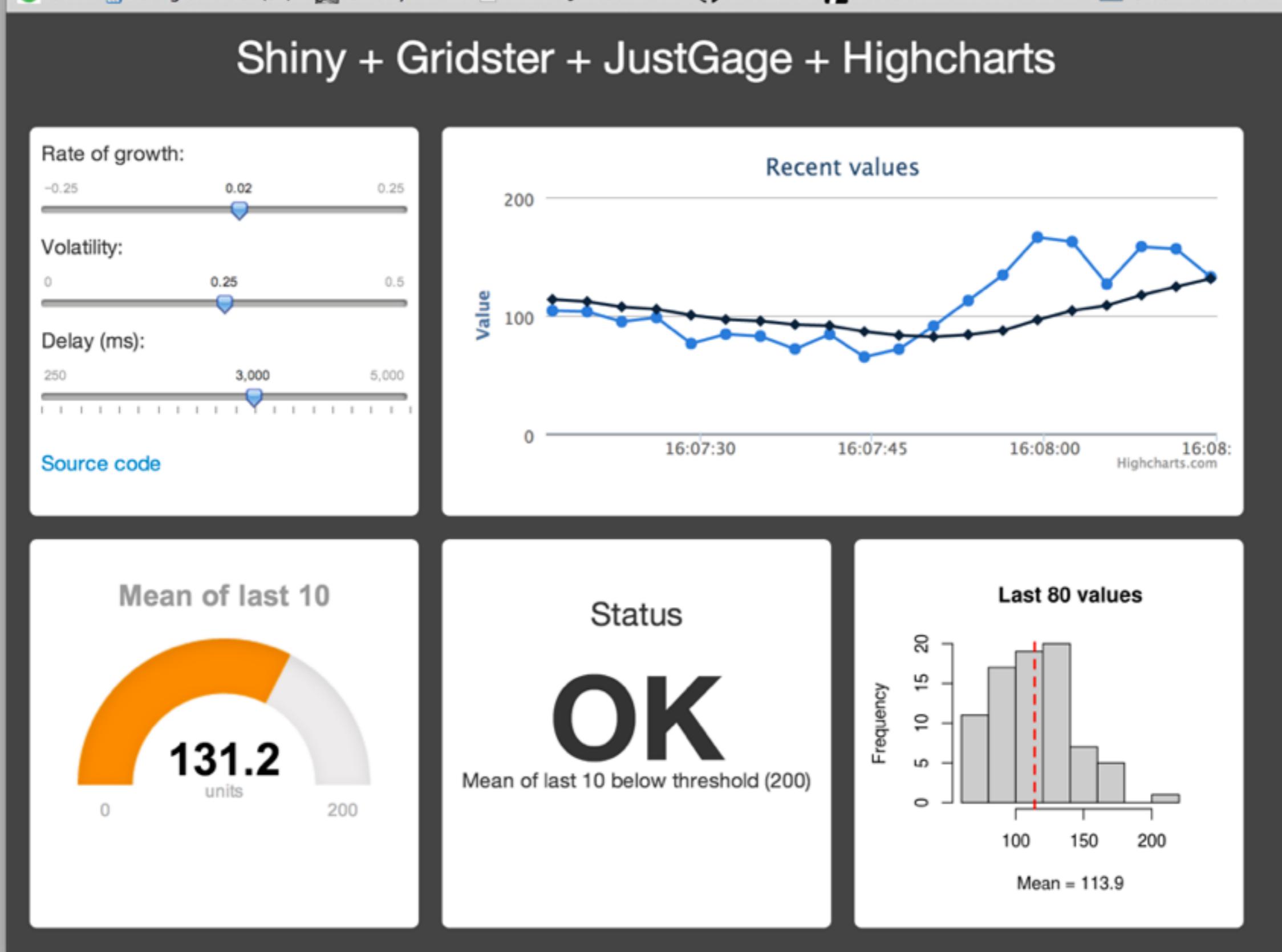
1. Creates reactive R objects
- 2. Builds HTML web pages**

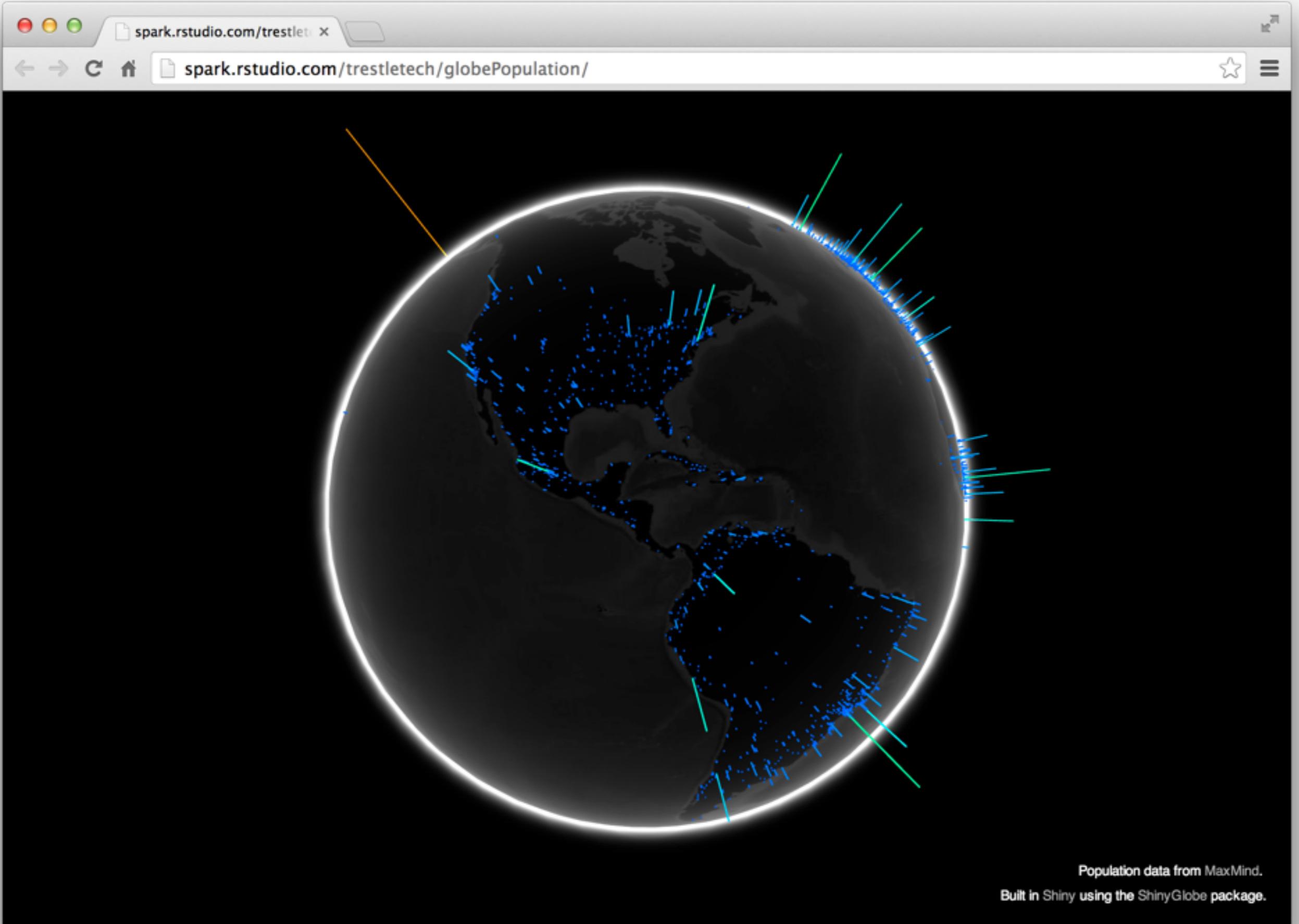
# web Apps



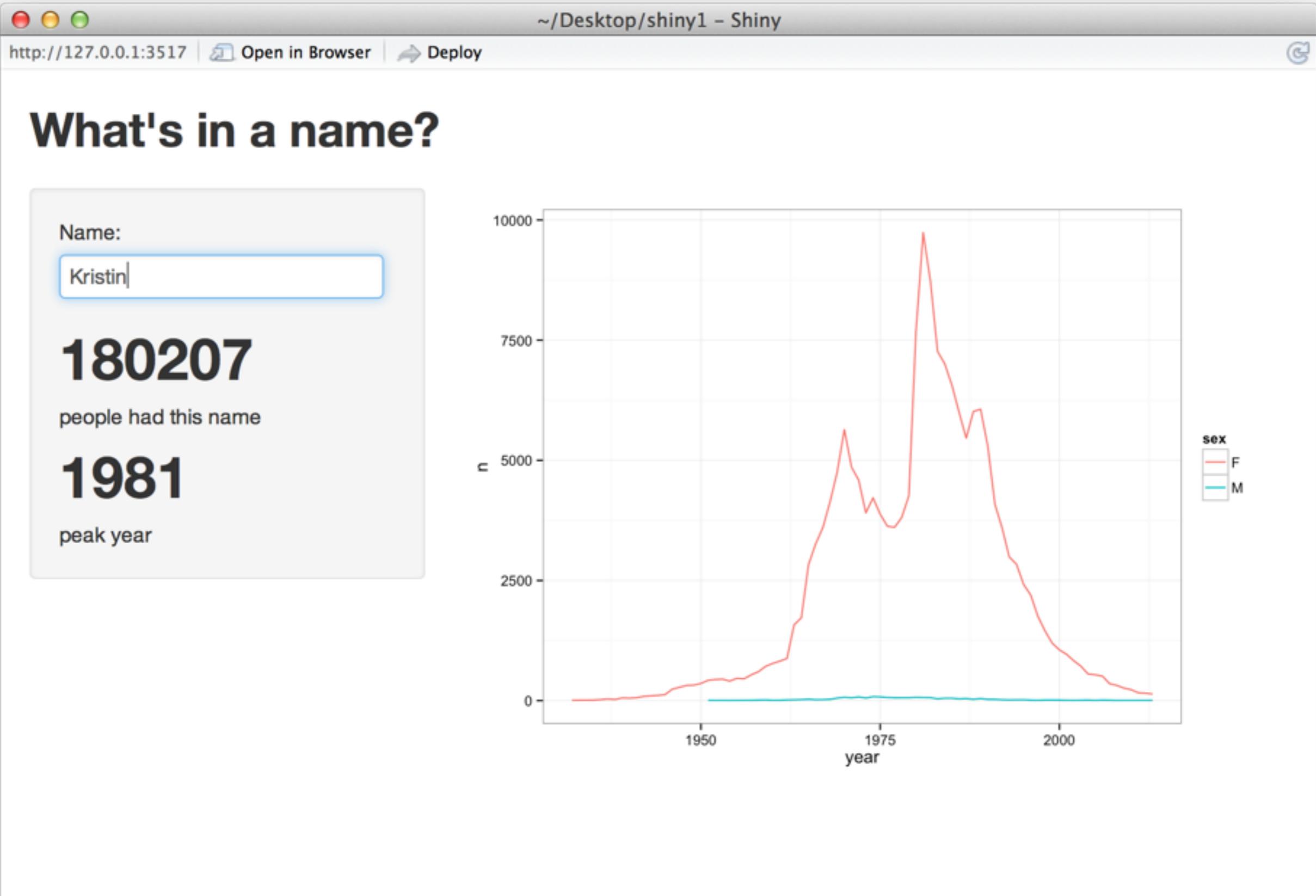
<http://glimmer.rstudio.com/winston/heightweight/>







<http://spark.rstudio.com/trestletech/globePopulation/>

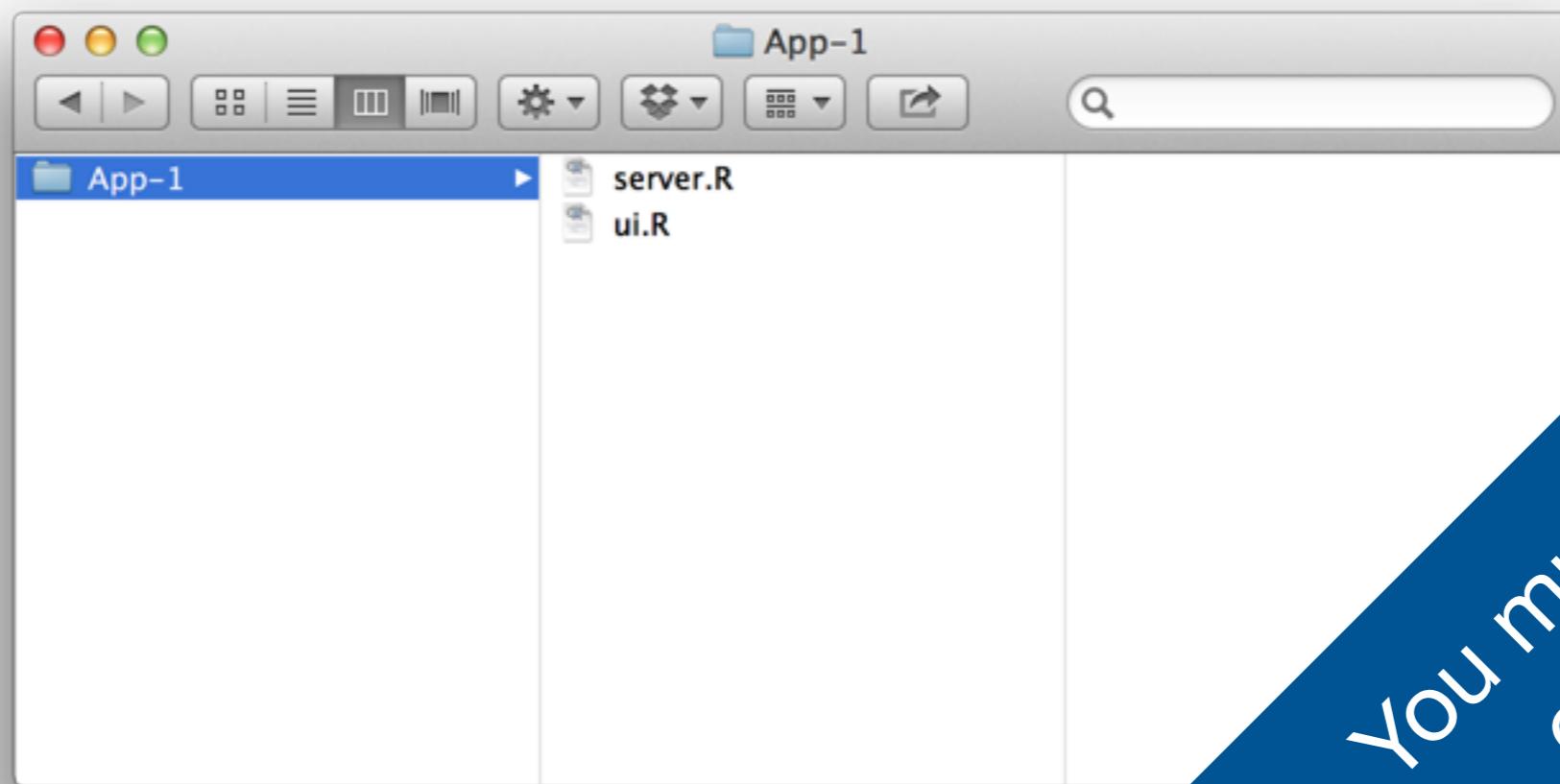


# Structure

# Structure of a Shiny web app

One directory with two files:

- server.R
- ui.R



You must use these  
exact names

# runApp

You launch the app with `runApp` (your computer will build a local web site that hosts the app).

```
runApp("~/Documents/App-1")
```

File path to app directory.  
R will append the file path to the working directory, if path does not begin at the home directory

# Warm up

Open a new RStudio project.

New Project > New Directory > Shiny Web App

RStudio will create a directory that has a template server.R and ui.R file in it. (Rstudio will set your working directory to this directory).

Run the template app.

**server.R**

server.R contains all of the instructions (R code) your app needs to build its rendered outputs.

server.R **always** includes the following code.

```
shinyServer(function(input, output) {
})
```

Place all of the R code from your document between the braces (except for the widget functions).

```
shinyServer(function(input, output) {
 library(babynames)
 names <- reactive(subset(babynames, name == input$name))
 renderText(input$name)

 library(ggplot2)
 renderPlot({
 qplot(year, n, data = names(), geom ="line", color = sex) +
 theme_bw()
 })
 renderText(input$name)
 renderText(names()$year[which.max(names()$n)])
 renderText(sum(names()$n))
 renderText(input$name)
})
```

Move to the top anything that only needs to be run once per server (e.g., *loading packages, data sets, or scripts*)

```
library(babynames)
library(ggplot2)

shinyServer(function(input, output) {
 names <- reactive(subset(babynames, name == input$name))
 renderText(input$name)
 renderPlot({
 qplot(year, n, data = names(), geom ="line", color = sex) +
 theme_bw()
 })
 renderText(input$name)
 renderText(names()$year[which.max(names()$n)])
 renderText(sum(names()$n))
 renderText(input$name)
})
```

# Save **every** rendered object to an element of output

```
library(babynames)
library(ggplot2)

shinyServer(function(input, output) {
 names <- reactive(subset(babynames, name == input$name))
 output$name <- renderText(input$name)
 output$trend <- renderPlot({
 qplot(year, n, data = names(), geom ="line", color = sex) +
 theme_bw()
 })
 output$peak <- renderText(names()$year[which.max(names()$n)])
 output$total <- renderText(sum(names()$n))
})
```

# Save **every** rendered object to an element of output

```
library(babynames)
library(ggplot2)

shinyServer(function(input, output) {
 names <- reactive(subset(babynames, name == input$name))
 output$name <- renderText(input$name)
 output$trend <- renderPlot({
 qplot(year, n, data = names(), geom ="line", color = sex) +
 theme_bw()
 })
 output$peak <- renderText(names()$year[which.max(names()$n)])
 output$total <- renderText(sum(names()$n))
})
```

# Warm up 2

Copy the code on the following slide into your server.R script. What rendered objects does it create?

```
library(babynames)
library(ggplot2)

shinyServer(function(input, output) {
 names <- reactive({
 subset(babynames, name == input$name)
 })

 output$trend <- renderPlot({
 qplot(year, n, data = names(), geom = "line", color = sex) +
 theme_bw()
 })

 output$total <- renderText({
 sum(names()$n)
 })

 output$peak <- renderText({
 names()$year[which.max(names()$n)]
 })

})
```

**ui.R**

ui.R builds the web page that displays your widgets and rendered output. In this file, you

- create the layout
- place widgets and output
- add html elements (optional)

# Layout

# The shortest viable ui.R file:

```
shinyUI(fluidPage())
```

Place layout functions inside fluidPage to layout the app

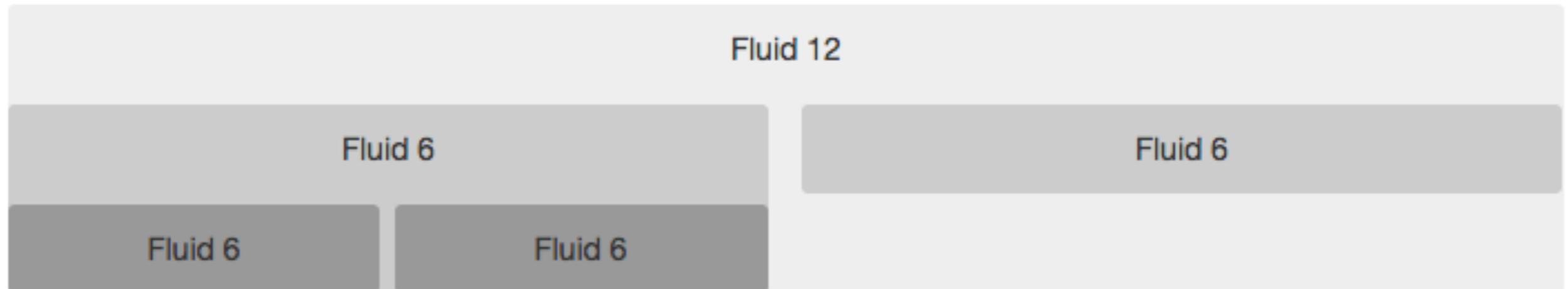
```
shinyUI(fluidPage(
 titlePanel(""),
 sidebarLayout(
 sidebarPanel(),
 mainPanel()
)
)
```



# Application Layout Guide

<http://shiny.rstudio.com/articles/layout-guide.html>

sidebarLayout is an easy, pre-packaged layout, but you can create any layout you like with Shiny's fluid grid system.



# Adding text

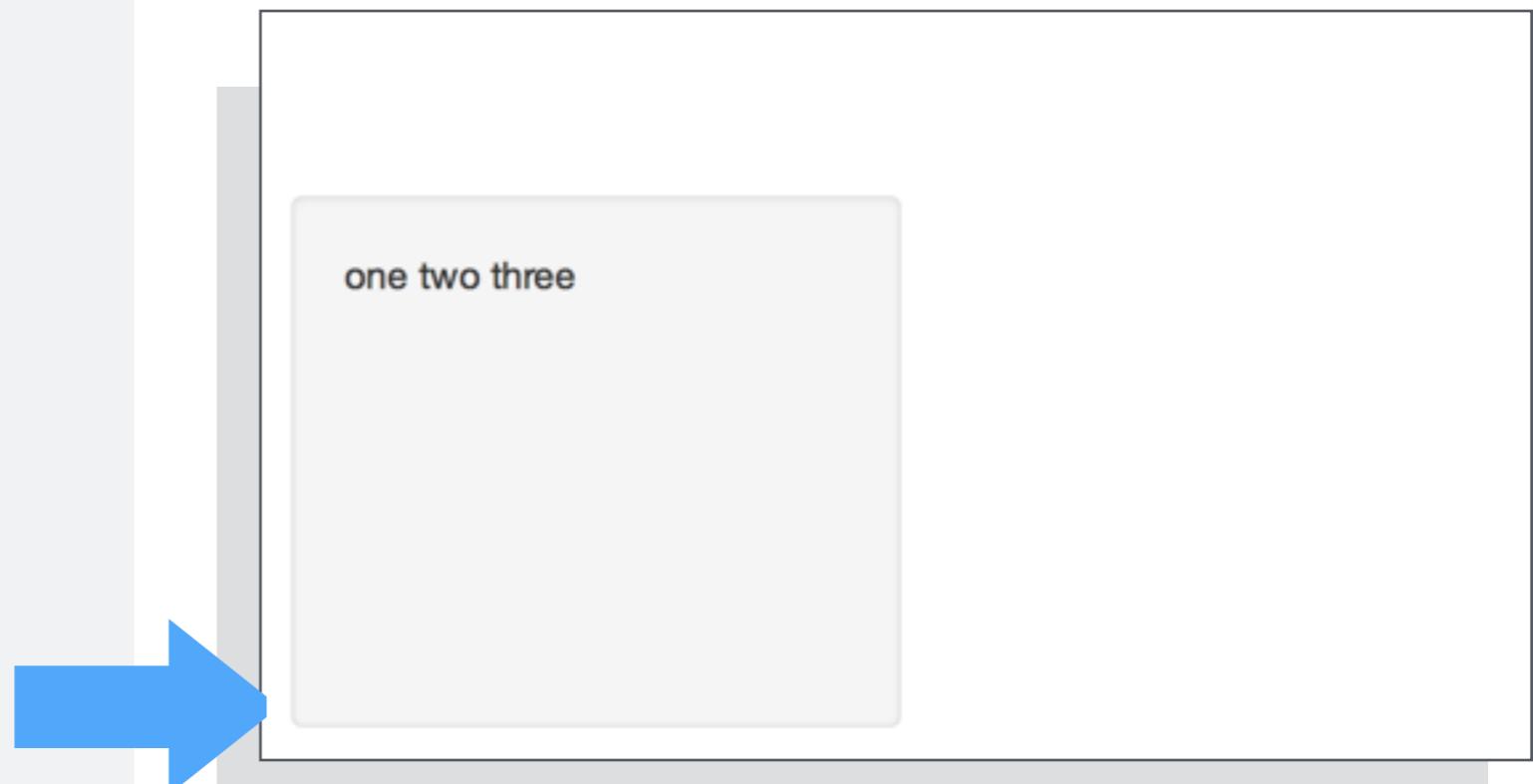
# Place raw text into any panel you like.

```
shinyUI(fluidPage(
 titlePanel("one"),
 sidebarLayout(
 sidebarPanel("two"),
 mainPanel("three"))
)
)
```



# Place raw text into any panel you like.

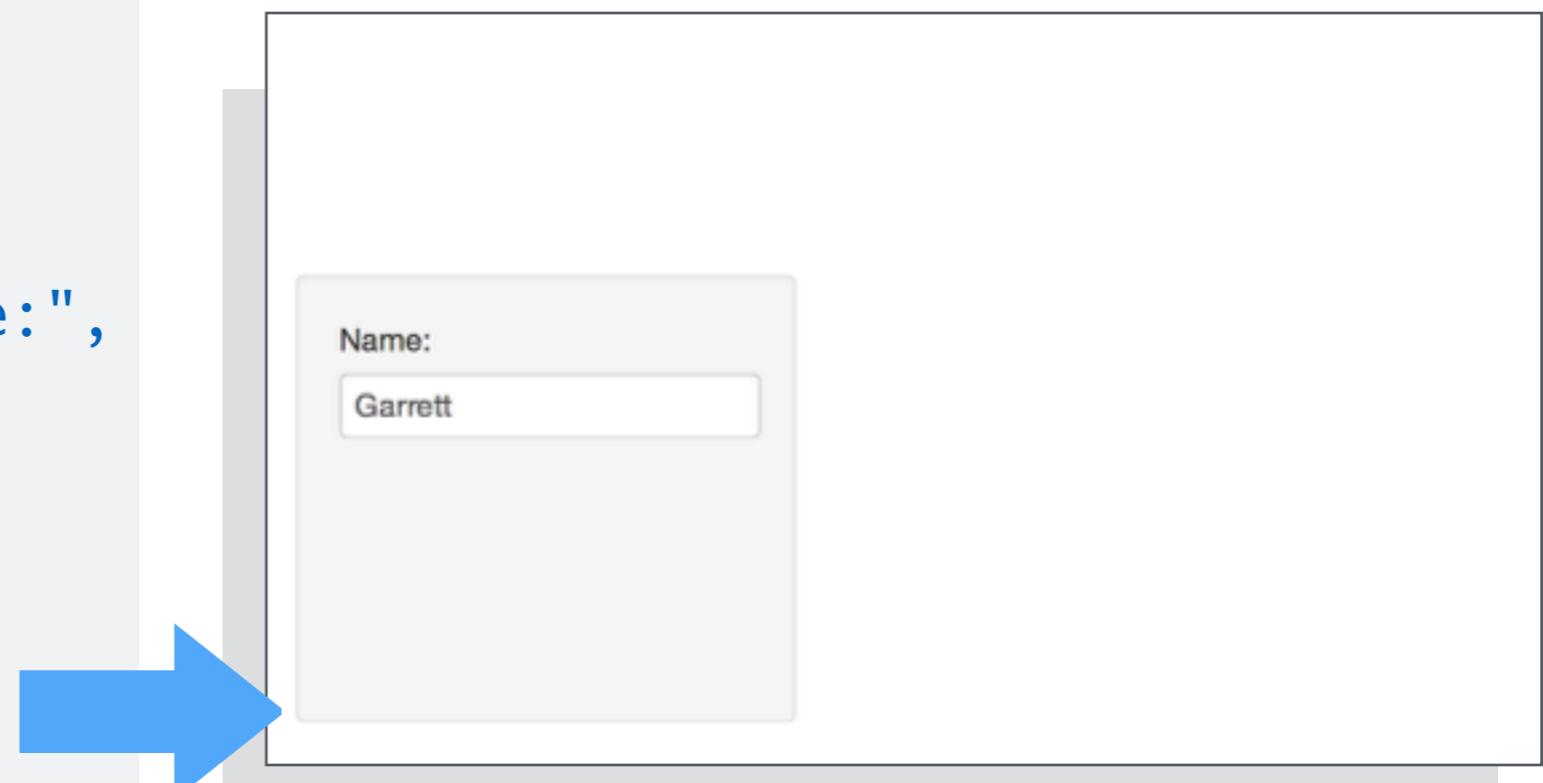
```
shinyUI(fluidPage(
 titlePanel(""),
 sidebarLayout(
 sidebarPanel(
 "one",
 "two",
 "three"),
 mainPanel()
)
)
```



# Adding widgets

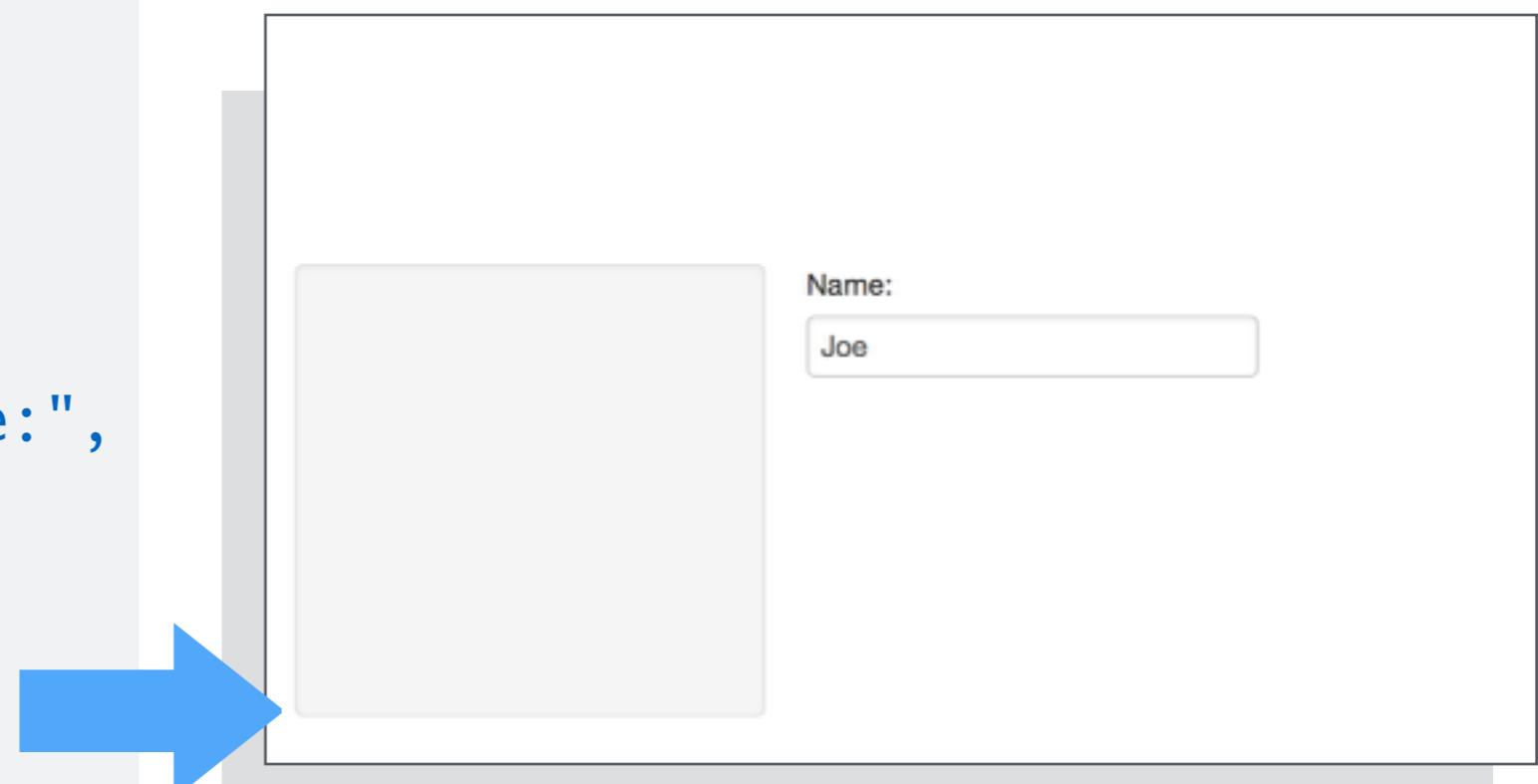
Place your widget functions where you want the widgets to appear. (*titlePanel* is not an option)

```
shinyUI(fluidPage(
 titlePanel(""),
 sidebarLayout(
 sidebarPanel(
 textInput("name", "Name:",
 value = "Joe")
),
 mainPanel()
)
)
```



Place your widget functions where you want the widgets to appear. (*titlePanel* is not an option)

```
shinyUI(fluidPage(
 titlePanel(""),
 sidebarLayout(
 sidebarPanel(),
 mainPanel(
 textInput("name", "Name:",
 value = "Joe")
)
)
)
```



# Adding output

# ui.R

Use an \*Output function to place each element of output in a panel.

`plotOutput("trend")`

a \*Output function that matches the type of object

name of element (set in server.R)

## Function

## Inserts

dataTableOutput an interactive table

htmlOutput rawHTML

imageOutput image

plotOutput plot

tableOutput table

textOutput text

uiOutput a Shiny UI element

verbatimText text

# server.R

```
counties <- readRDS("data/
counties.rds")
source("helpers.R")

shinyServer(function(input, output) {
 output$table <- renderDataTable({
 counties
 })

 output$map <- renderPlot({
 percent_map(var = counties$asian,
 color = "magenta",
 legend.title = "Percent Asian",
 min = 0, max = 10)
 })
})
```

# ui.R

```
shinyUI(fluidPage(
 titlePanel("censusVis"),
 sidebarLayout(
 sidebarPanel(
 helpText("Create demographic
maps with information from the
2010 US Census."),
 selectInput("var", label =
 "Choose a variable to display",
 choices = c("Percent White",
 "Percent Black",
 "Percent Hispanic",
 "Percent Asian",
 "Percent Other"),
 selected = "Percent White"),
 sliderInput("range",
 label = "Range of interest:",
 min=0,max=100,value =c(0,100)),
 img(src = "bigorb.png", height =
 50, width = 50), © 2014 RStudio, Inc. All rights reserved.
```

# server.R

```
counties <- readRDS("data/
counties.rds")
source("helpers.R")

shinyServer(function(input, output) {
 output$table <- renderDataTable({
 counties
 })

 output$map <- renderPlot({
 percent_map(var = counties$asian,
 color = "magenta",
 legend.title = "Percent Asian",
 min = 0, max = 10)
 })
})
```

```
choose a variable to display",
choices = c("Percent White",
 "Percent Black",
 "Percent Hispanic",
 "Percent Asian",
 "Percent Other"),
selected = "Percent White"),
sliderInput("range",
 label = "Range of interest:",
 min=0,max=100,value =c(0,100)),
img(src = "bigorb.png", height =
 50, width = 50),
"This was made with",
a("Shiny", href =
 "shiny.rstudio.com"), ".")
,
mainPanel(
 plotOutput("map"),
 hr(),
 dataTableOutput("table"))
)
)
```

# server.R

```
counties <- readRDS("data/
counties.rds")
source("helpers.R")

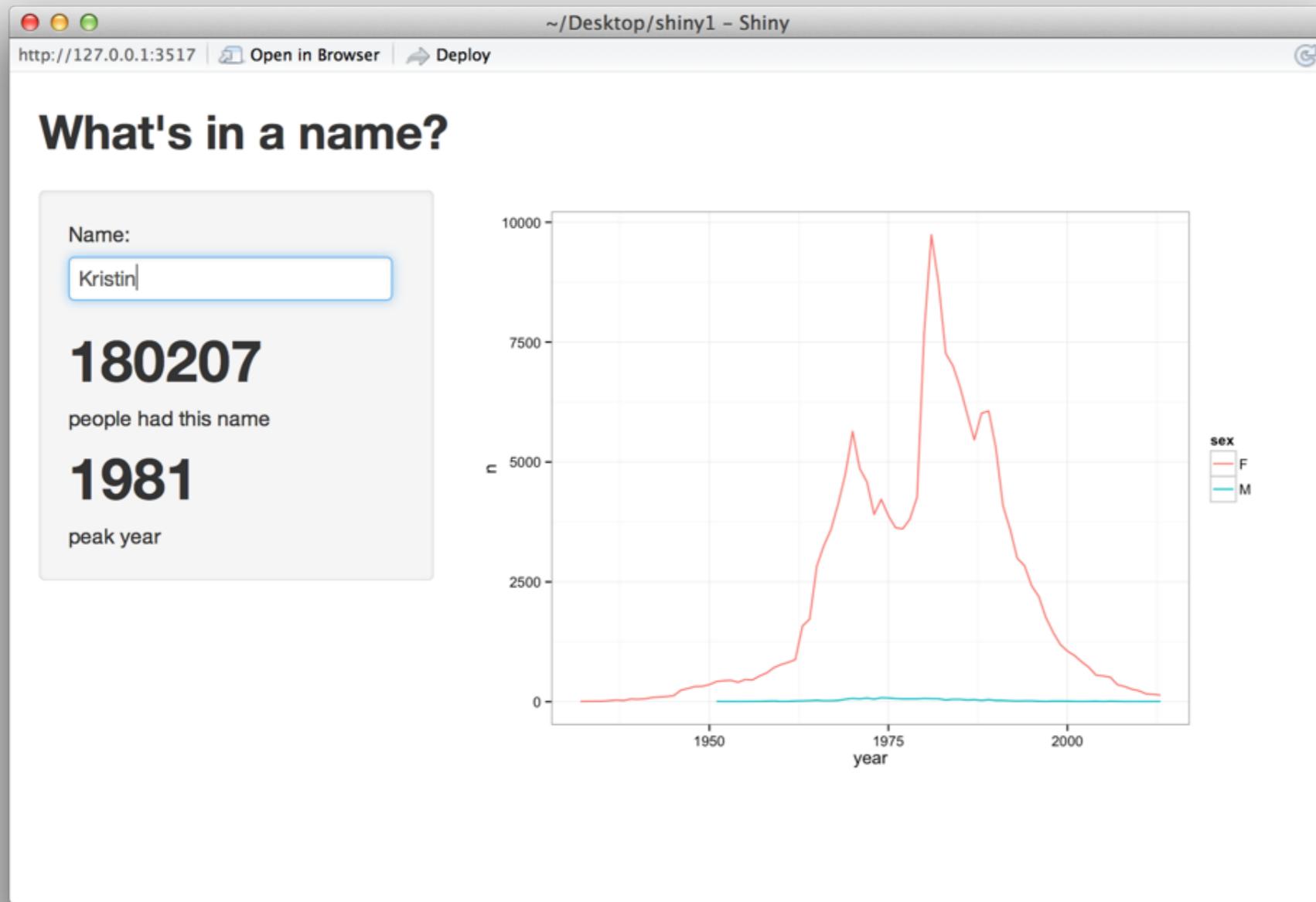
shinyServer(function(input, output) {
 output$table <- renderDataTable({
 counties
 })

 output$map <- renderPlot({
 percent_map(var = counties$asian,
 color = "magenta",
 legend.title = "Percent Asian",
 min = 0, max = 10)
 })
})
```

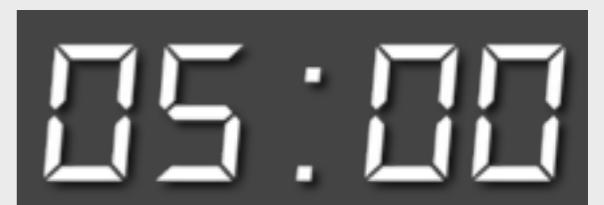
```
choose a variable to display ,
choices = c("Percent White",
 "Percent Black",
 "Percent Hispanic",
 "Percent Asian",
 "Percent Other"),
selected = "Percent White",
sliderInput("range",
 label = "Range of interest:",
 min=0,max=100,value =c(0,100)),
img(src = "bigorb.png", height =
 50, width = 50),
"This was made with",
a("Shiny", href =
 "shiny.rstudio.com"), ".."
,
mainPanel(
 plotOutput("map"),
 hr(),
 dataTableOutput("table"))
)
)
```

# Your turn

Modify your ui.R file to recreate the web app.



*Tip: You can increase the size of text (or `textOutput(...)`) by surrounding it with `h1()`, e.g. `h1("Header One size")`*



```
shinyUI(fluidPage(

 titlePanel("What's in a name?"),

 sidebarLayout(
 sidebarPanel(
 textInput("name", "Name:", value = "Garrett"),
 h1(textOutput("total")),
 "people had this name",
 h1(textOutput("peak")),
 "peak year"
),
 mainPanel(
 plotOutput("trend")
)
)
)
```

```
shinyUI(fluidPage(

 titlePanel("What's in a name?"),

 sidebarLayout(
 sidebarPanel(
 textInput("name", "Name:", value = "Garrett"),
 h1(textOutput("total")),
 "people had this name",
 h1(textOutput("peak")),
 "peak year"
),
 mainPanel(
 plotOutput("trend")
)
)
)
```

# **HTML** elements

# Shiny HTML tag functions

`h1()` ~ `<h1></h1>`  
`a` ~ `<a></a>`

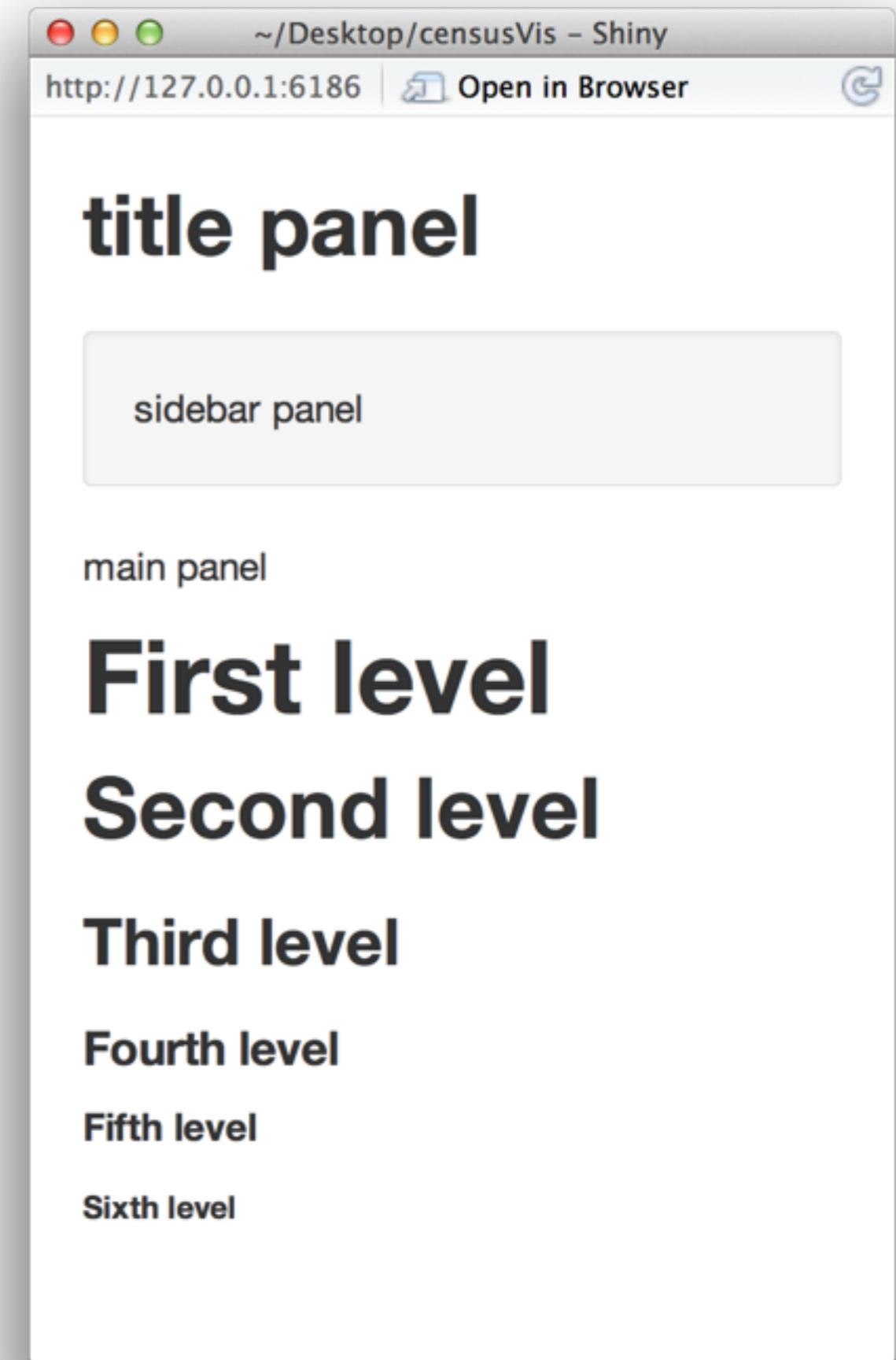
<http://shiny.rstudio.com/articles/html-tags.html>

Shiny provides R functions that recreate common HTML tags. You can find a list of these functions and how to use them to enhance your app at the link above.

# h1 - h6

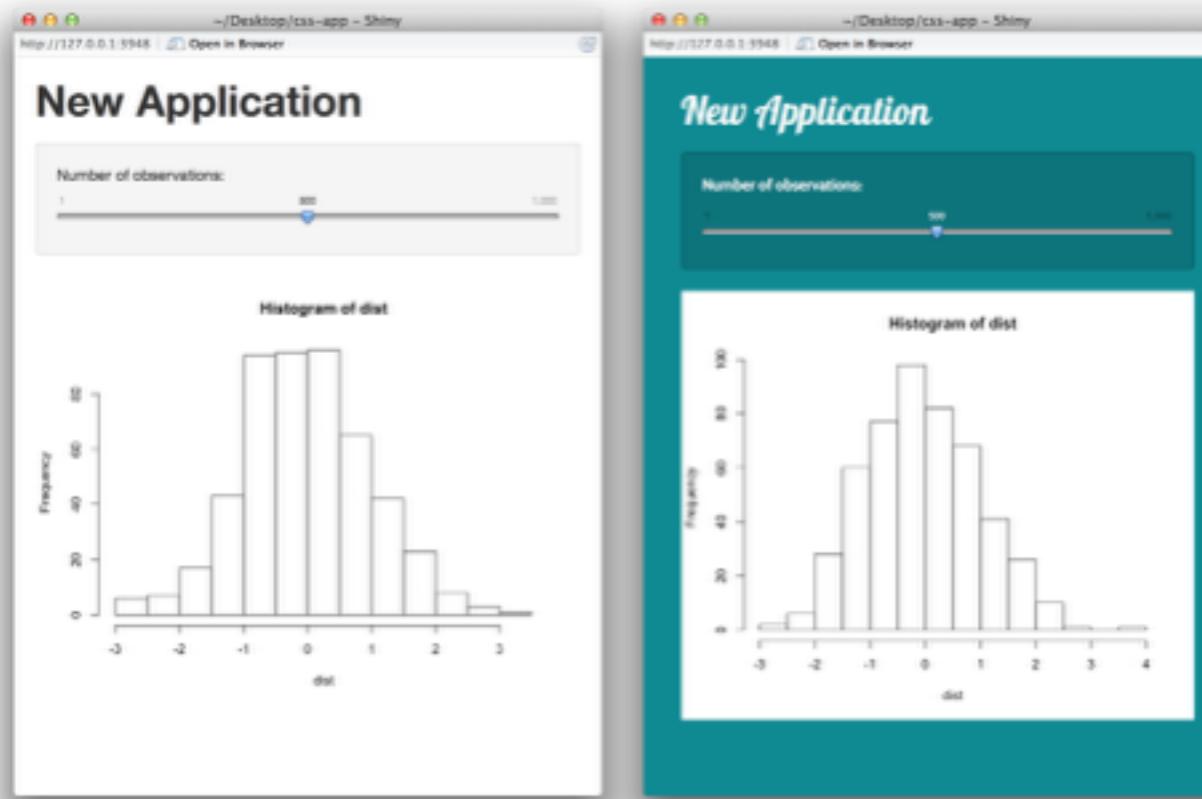
## Headers

```
shinyUI(fluidPage(
 titlePanel("title panel"),
 sidebarLayout(
 sidebarPanel("sidebar panel"),
 mainPanel("main panel",
 h1("First level"),
 h2("Second level"),
 h3("Third level"),
 h4("Fourth level"),
 h5("Fifth level"),
 h6("Sixth level"))
))
```



# Customize your apps with HTML, CSS, and Javascript

<http://shiny.rstudio.com/articles/css.html>



You can pair any app with whatever web technologies you wish. The above guide explains how to style your app with CSS.

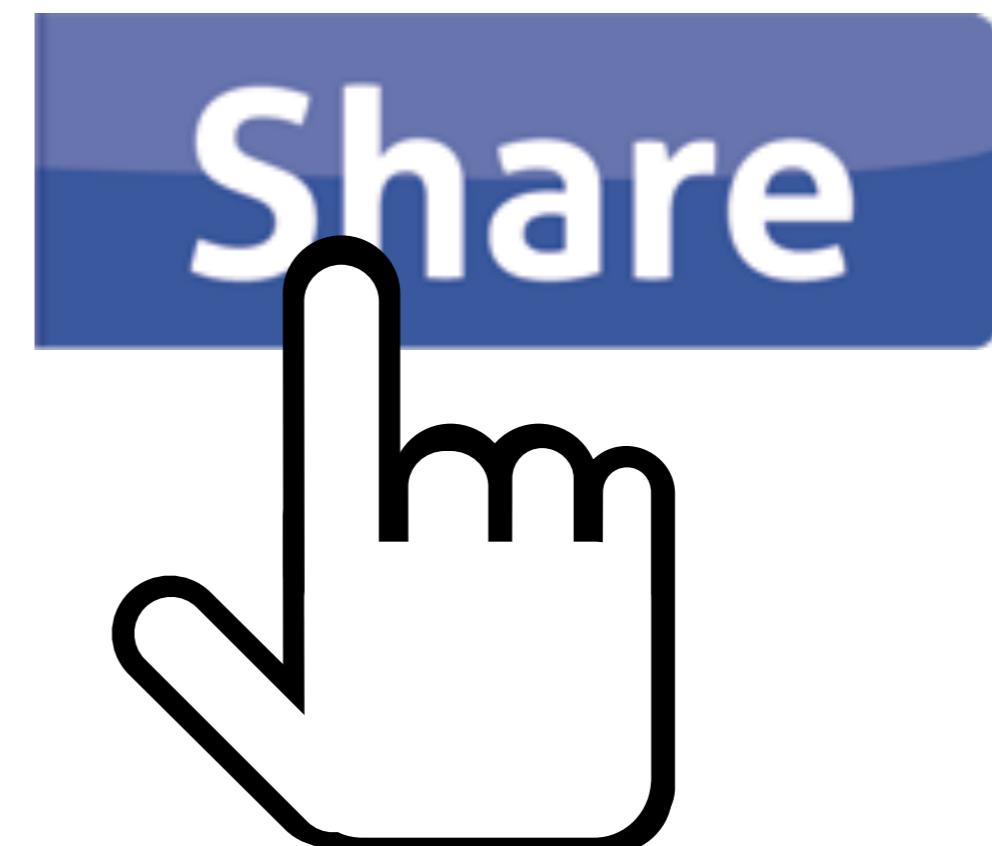
# Sharing

These functions can help users run your server.R and ui.R files on their own computer.

`runApp()`

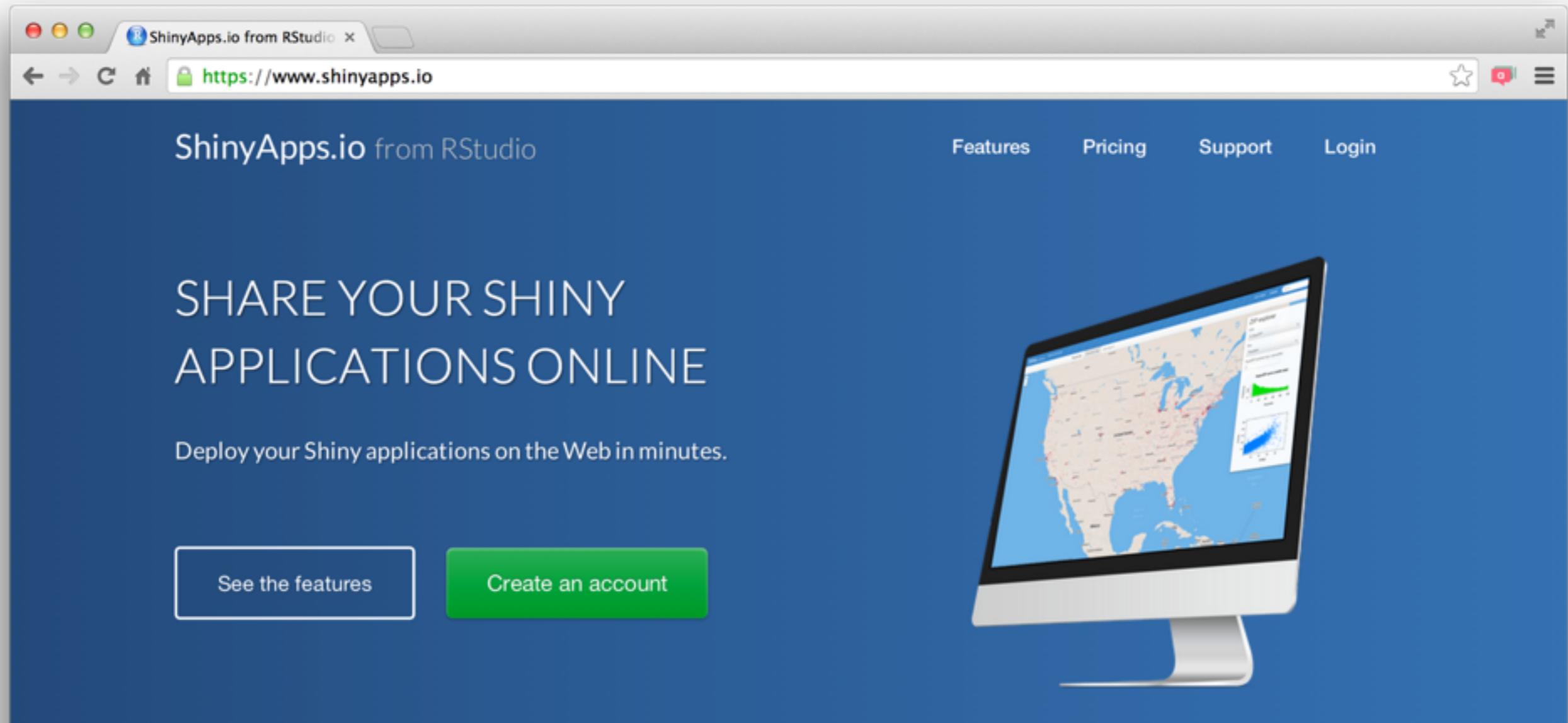
`runGitHub()`

`runGist()`



**ShinyApps.io**

# Hassle-free cloud hosting for Shiny



The screenshot shows a web browser window displaying the ShinyApps.io homepage. The address bar shows the URL <https://www.shinyapps.io>. The page has a blue header with the text "ShinyApps.io from RStudio". In the center, there's a large white text area with the heading "SHARE YOUR SHINY APPLICATIONS ONLINE" and a subtext "Deploy your Shiny applications on the Web in minutes.". Below this are two buttons: a white button with "See the features" and a green button with "Create an account". To the right, there's a large image of a computer monitor displaying a Shiny application with a map of the United States. The top navigation bar includes links for "Features", "Pricing", "Support", and "Login". The overall theme is clean and professional.

Looking for an easy way to deploy Shiny?

ShinyApps.io hosts your Shiny applications & documents.

# Hassle-free cloud hosting for Shiny

The screenshot shows the ShinyApps.io dashboard interface. On the left, a sidebar menu includes 'Dashboard' (selected), 'Applications' (with a dropdown arrow), and 'Accounts'. The main content area features a large blue box displaying '60 Apps online' with a 'View your applications' button and a cloud icon. Below this is a table titled 'Latest applications' with the following data:

| ID    | Name       | Status  |
|-------|------------|---------|
| 11941 | front-page | Running |
| 11214 | twitter    | Running |
| 11199 | submit     | Running |
| 11195 | text       | Running |
| 11194 | slider     | Running |
| 11193 | select     | Running |

At the bottom right, a copyright notice reads '© 2014 RStudio, Inc. All rights reserved.'

# ShinyApps.io

- Each app is hosted at its own URL on server run by RStudio
- Administrative tools, etc.
- Designed to host Shiny apps

# Get Started

**Guide at:**

<http://shiny.rstudio.com/articles/shinyapps.html>

1. install shinyapps package
2. create account at www.shinyapps.io
3. Run setAccountInfo() at command line

deployApp() or click button to host app online

**Shiny  
Server(Pro)**

# Shiny Server

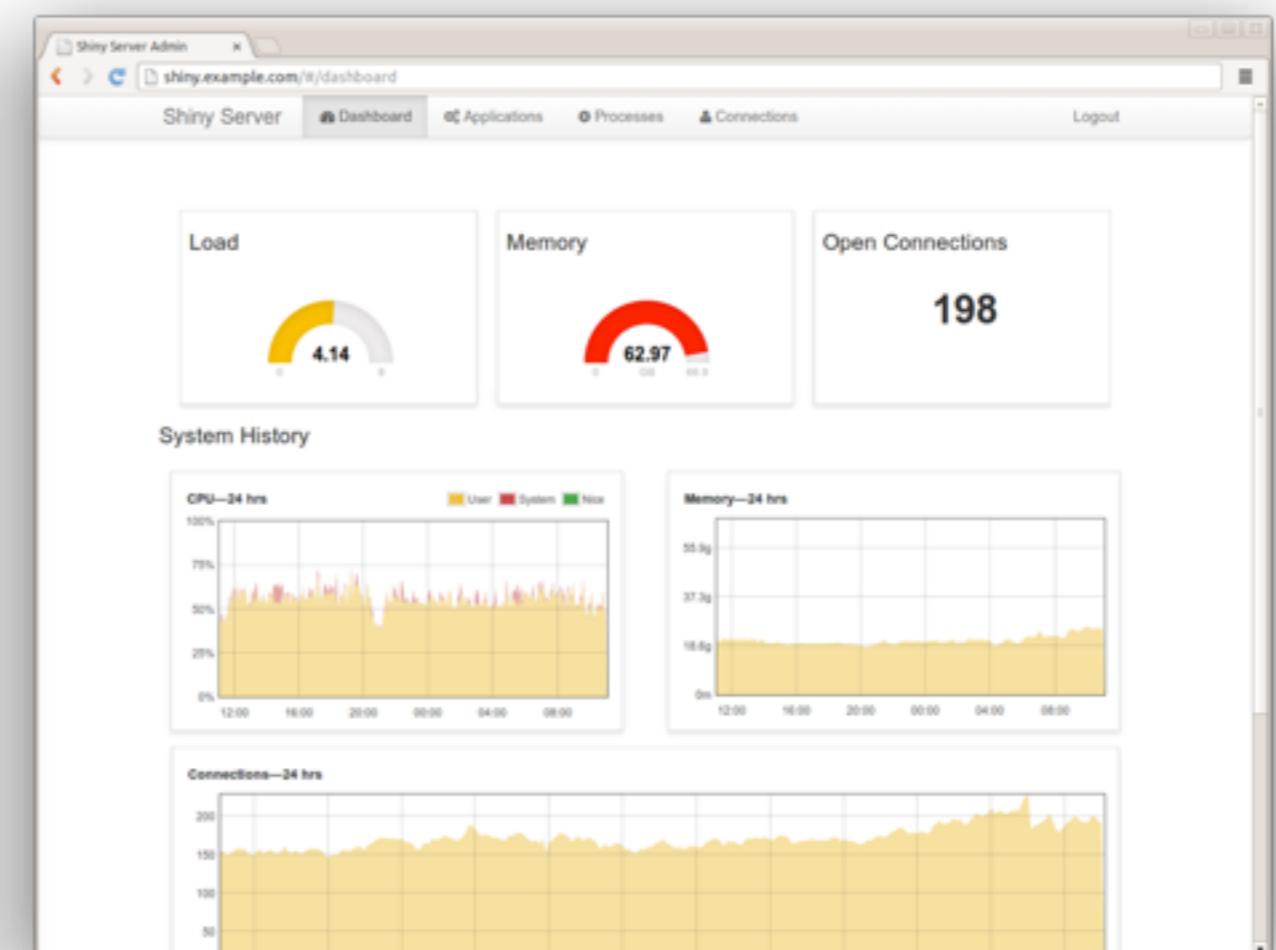
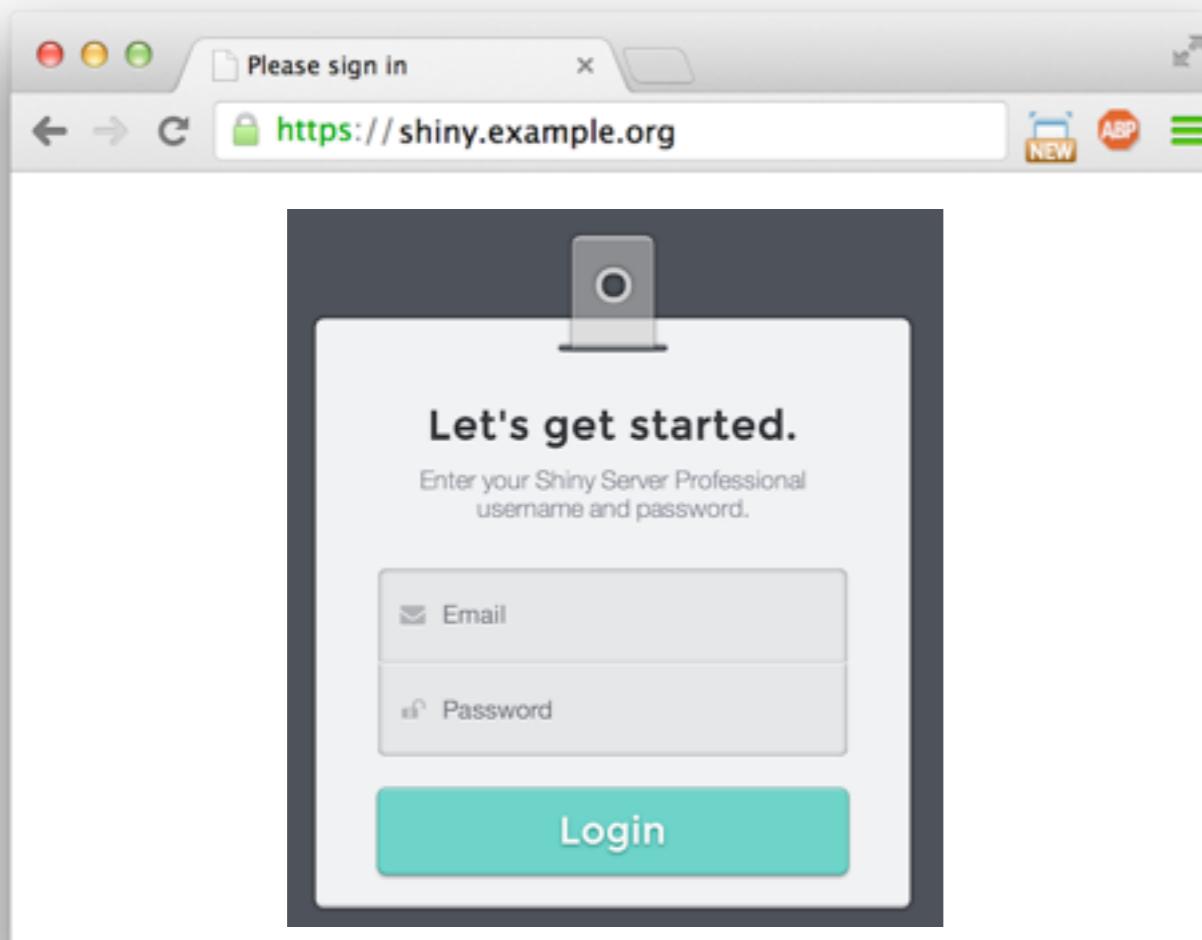
A back end program that builds a web server specifically designed to host Shiny apps

- Each app is hosted at its own URL
- Can deploy to internet, or within a controlled environment
- Starts app when user visits, closes app when user leaves
- Runs on a linux server

<http://shiny.rstudio.com/articles/shiny-server.html>

# Shiny Server Pro

- Password protect apps, SSL
- Administrative tools, priority support



# <http://rstudio.com/shiny/server>

|                                    |                                                                                                                                        | Open Source<br>Edition | Professional<br>Edition |
|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|------------------------|-------------------------|
| <b>General</b>                     | Deploy Shiny applications to the Internet                                                                                              | •                      | •                       |
|                                    | Move computation close to the data                                                                                                     | •                      | •                       |
|                                    | Host multiple applications on a single server                                                                                          | •                      | •                       |
|                                    | Deploy Shiny applications behind firewalls                                                                                             | •                      | •                       |
|                                    | Custom page templates                                                                                                                  | •                      | •                       |
| <b>Security and authentication</b> | Password file authentication                                                                                                           | •                      |                         |
|                                    | LDAP and Active Directory authentication                                                                                               |                        | •                       |
|                                    | Google authentication (OAuth2)                                                                                                         |                        | •                       |
|                                    | PAM authentication & sessions                                                                                                          |                        | •                       |
|                                    | Group based authorization                                                                                                              |                        | •                       |
|                                    | SSL support                                                                                                                            |                        | •                       |
| <b>Tuning and scaling</b>          | Scale applications across multiple processes                                                                                           |                        | •                       |
|                                    | View and manage active sessions                                                                                                        |                        | •                       |
|                                    | Allocate resources on a per application basis                                                                                          |                        | •                       |
|                                    | Define application concurrency limits                                                                                                  |                        | •                       |
| <b>Server monitoring</b>           | System performance and resource metrics           |                        | •                       |
|                                    | Per application performance and resource metrics  |                        | •                       |
|                                    | Application usage metrics                                                                                                              |                        | •                       |
|                                    | Customizable health check end point                                                                                                    |                        | •                       |

\* For volume discounts, OEMs, or additional capacity for larger audiences please email us at [sales@rstudio.com](mailto:sales@rstudio.com).

<http://www.rstudio.com/shiny/server/pro>

## Shiny Server Pro Edition

Security tools (authentication, encryption)

Management tools (monitoring, auditing, allocation)

Scalability (multiple processes per application)

Pricing per year: \$9,995 - 20 concurrent sessions

\$24,995 - 200 concurrent sessions



Academic and  
Small Business  
discounts available

# **How to learn more**

# Shiny

by RStudio

A web application framework for R

Turn your analyses into interactive web applications

No HTML, CSS, or JavaScript knowledge required

TUTORIAL

ARTICLES

GALLERY

REFERENCE

DEPLOY

HELP



Get inspired  
(gallery)



Get started  
(tutorial)



Go deeper  
(articles)

Centralized  
learning site

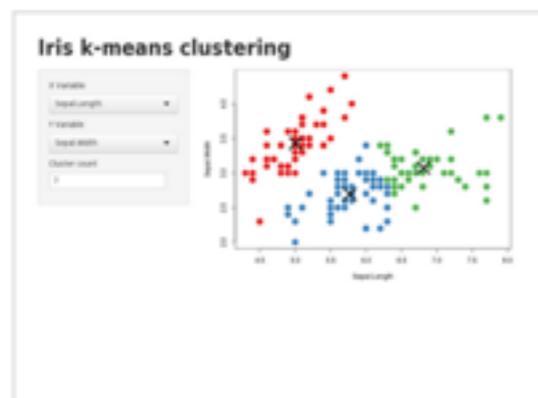
<http://shiny.rstudio.com>

All rights reserved.

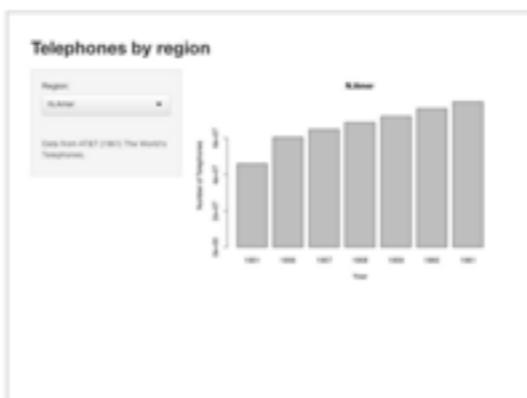
# Gallery

## Start simple

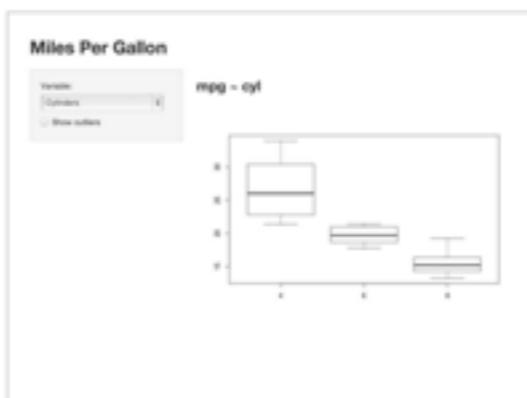
If you're new to Shiny, these simple but complete applications are designed for you to study.



Kmeans example



Telephones by region



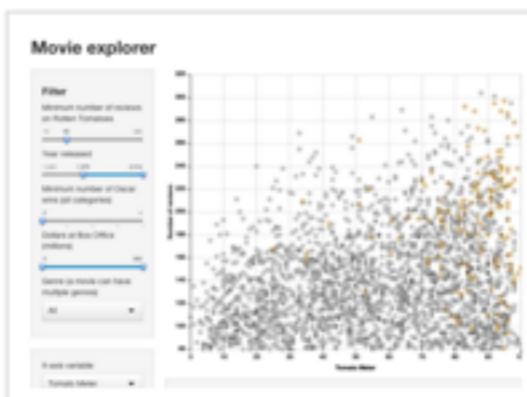
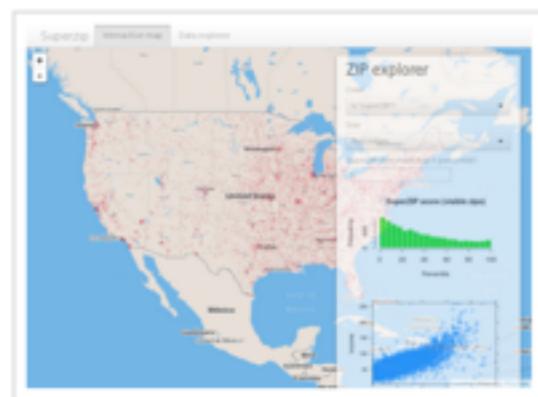
Miles per gallon



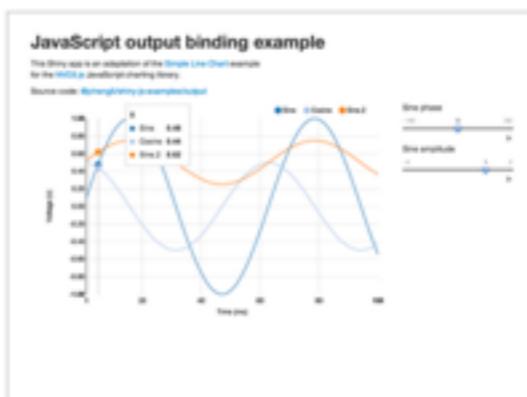
Word cloud

## Interactive visualizations

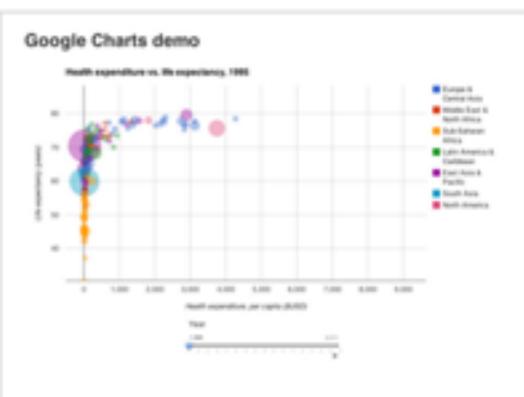
Shiny is designed for fully interactive visualization, using JavaScript libraries like [d3](#), [Leaflet](#), and [Google Charts](#).



Movie explorer



NVD3 line chart output



Google Charts

## Widgets

Get to know many of the input and output widgets that are available in Shiny with these examples.

[OVERVIEW](#)[TUTORIAL](#)[ARTICLES](#)[GALLERY](#)[REFERENCE](#)[DEPLOY](#)[HELP](#)

# Teach yourself Shiny

## Who should take the tutorial?

You will get the most out of this tutorial if you already know how to program in R, but not Shiny.

If R is new to you, you may want to check out the learning resources at [www.rstudio.com/training](http://www.rstudio.com/training) before taking this tutorial. If you are not sure whether you are ready for Shiny, try our [quiz](#).

If you use Shiny on a regular basis, you may want to skip this tutorial and visit the articles section of the Development Center. In the articles section, we cover individual Shiny topics at an advanced level.

## Get started with Shiny

With this seven lesson tutorial, you will move up from R programmer to Shiny developer. Each lesson takes about 15 minutes and teaches one new Shiny skill. By the end of the lessons, you will know how to build and deploy a Shiny app.

Each lesson includes an exercise. Don't skip the exercises, even if you are tempted to get to the next lesson. The learning occurs in the exercises. How do we know? Because we designed the tutorial and organized the material around the exercises.

Click the Lesson 1 button to get started and say hello to Shiny!

- [Lesson 1 - Welcome to Shiny](#)
- [Lesson 2 - Layout the user interface](#)
- [Lesson 3 - Add control widgets](#)
- [Lesson 4 - Display reactive output](#)
- [Lesson 5 - Use R scripts and data](#)
- [Lesson 6 - Use reactive expressions](#)
- [Lesson 7 - Share your apps](#)

[Continue to lesson 1](#)

[OVERVIEW](#)[TUTORIAL](#)[ARTICLES](#)[GALLERY](#)[REFERENCE](#)[DEPLOY](#)[HELP](#)

# Articles

## The basics

If you've been through the [tutorial](#) and need a refresher, these articles are a good place to start. They describe the lay of the land.

[The basic parts of a Shiny app](#)[How to build a Shiny app](#)[How to launch a Shiny app](#)[How to get help](#)[The Shiny Cheat sheet](#)

## Layouts and UI

These articles explain how to control the layout, user-interface, and general appearance of your Shiny apps.

[Application layout guide](#)[Display modes](#)[Tabssets](#)[Customize your UI with HTML](#)[Build your entire UI with HTML](#)[Build a dynamic UI that reacts to user input](#)[Shiny HTML Tags Glossary](#)

## Widgets

These articles describe Shiny's pre-built widgets and provide ideas on how to use them. (See also [Lesson 3](#) in the tutorial, and the [Widgets](#) section in the [gallery](#).)

[Using sliders](#)[Help users download data from your app](#)[Using selectize input](#)

## Outputs

These articles show you how to create and use different output objects, the parts of your app that display results and react to user input.

[Render images in a Shiny app](#)[How to use DataTables in a Shiny App](#)

## Deploying apps

These articles describe the different ways to share your Shiny apps with users.

[Getting started with ShinyApps.io](#)[Introduction to Shiny Server](#)[Deploying Shiny apps over the web](#)[Sharing apps to run locally](#)

## Reactive programming

These articles describe reactivity from a conceptual level. Understanding reactivity will help you build apps that are more efficient, robust, and correct.

[Reactivity: An overview](#)[Stop reactions with isolate\(\)](#)[Execution scheduling](#)

# Shiny Discussion group

<https://groups.google.com/forum/#!forum/shiny-discuss>

The screenshot shows a web browser window with the title bar '(99+) Shiny - Web Framework'. The address bar contains the URL <https://groups.google.com/forum/#!forum/shiny-discuss>. The page itself is a Google Groups discussion forum for the 'shiny-discuss' group. The left sidebar has links for 'My groups', 'Home', 'Starred', and 'Favorites' (with a note to click stars to add to favorites). The main area shows a topic titled 'Shiny - Web Framework for R' with 30 of 2207 topics (99+ unread). It includes a 'Join group to post' button and a member count of 8+. Below the topic summary is a descriptive text about Shiny, followed by a call to action to post comments or show a gist. A list of recent posts follows:

- Browser differences? (7)**  
By Barb Banbury - 7 posts - 22 views  
3:46 PM
- fileInput, reload and defaults (1)**  
By Andrew Booker - 1 post - 3 views  
2:36 PM
- sudden problems deploying shiny app from spark.studio (2)**  
By Veronica Morales - 2 posts - 7 views  
11:24 AM
- shiny, ggplot2, and X11 on R 3.0.2 (3)**  
By Daniel Bowen - 3 posts - 51 views  
6:19 AM

At the bottom of the page, the URL <https://groups.google.com/forum/#!topic/shiny-discuss/rmAMf1vL6uY> is visible.

# RStudio training

[www.rstudio.com/resources/training/online-learning.html](http://www.rstudio.com/resources/training/online-learning.html)

Links to recommended tutorials, articles, examples  
to help you learn R and its extensions

The screenshot shows a web browser window with the title bar "Online Learning - RStudio". The address bar contains the URL "www.rstudio.com/resources/training/online-learning/". The page itself has a header with the RStudio logo and navigation links for Products, Resources, Pricing, About Us, Blog, and a search icon. Below the header, there's a section titled "Online Learning" with a breadcrumb trail "Home / Resources / Training / Online Learning". The main content area is titled "Online learning" and contains a paragraph about the availability of tutorials, articles, and examples. It features three cards with icons and titles: "R Programming" (heart icon), "Shiny" (star icon), and "R Markdown" (document icon). Each card has a "Read More >" link at the bottom. A large blue diagonal banner on the right side of the page contains the text "To learn R".

Online Learning

Home / Resources / Training / Online Learning

Online learning

A wealth of tutorials, articles, and examples exist to help you learn R and its extensions. Scroll down or click a link below for a curated guide to learning R and its extensions.

R Programming

Shiny

R Markdown

Read More >

Read More >

Read More >

To learn R

# Codecademy

<http://www.codecademy.com>

Free tutorials that teach HTML, CSS, and Javascript with instant feedback. I recommend this for learning the basics of these languages.

The screenshot shows a web browser window titled "Learn | Codecademy". The URL in the address bar is "www.codecademy.com/learn". The page content displays three cards for popular programming languages:

- HTML & CSS**: Features a teal icon of a document with code. Description: "Learn the building blocks of web development with HTML and CSS, and create your own website by the end of the course."
- jQuery**: Features a blue icon of three curved arrows. Description: "jQuery uses JavaScript to easily build interactive websites. Learn animation, events and DOM manipulation."
- JavaScript**: Features a white icon with "JS" in a blue box. Description: "Learn how to make your websites interactive and build browser based games."

A large blue diagonal banner across the bottom right of the image contains the text "To learn HTML, CSS, or JavaScript".