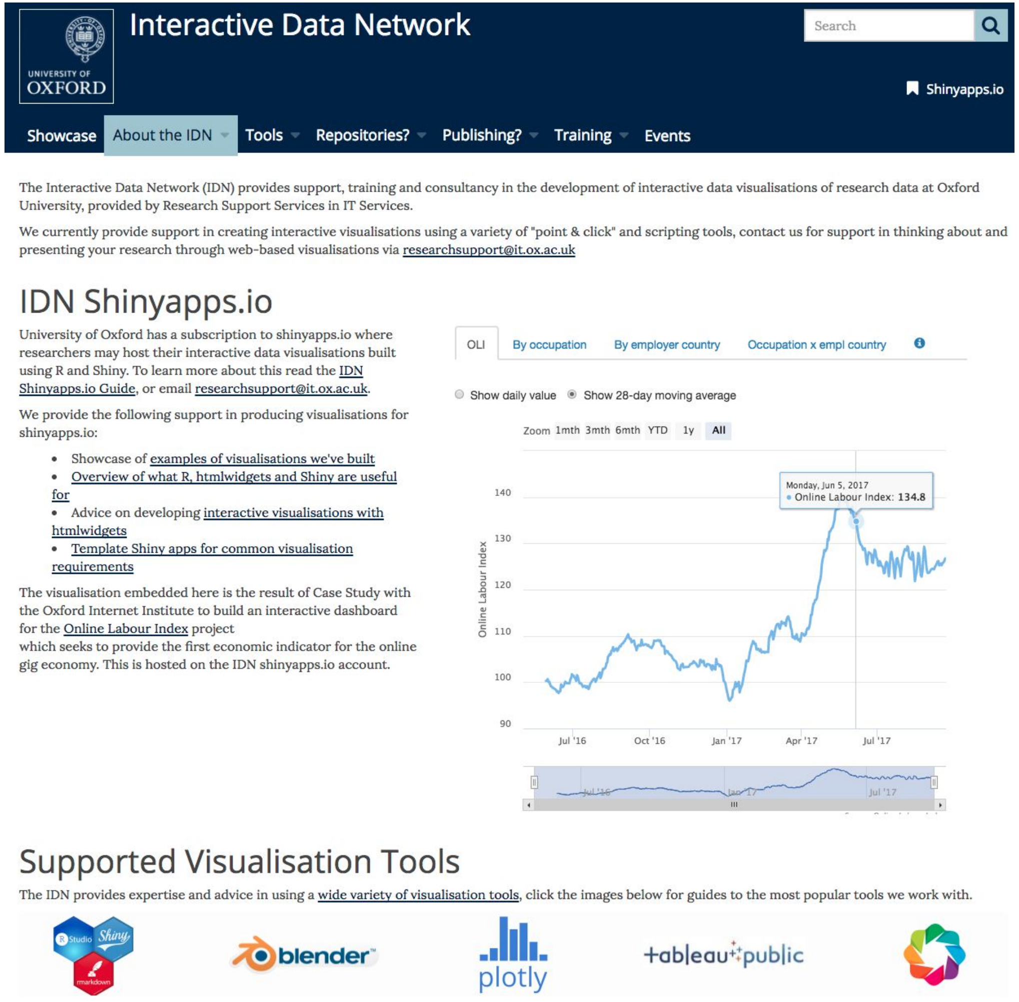
Interactive Visualisation with R (and just R)

Hands on Workshop @ #ODSC 2017 (13:30 - 15:00)

Martin John Hadley
@martinjhnhadley



Interactive Data Network

Search

Shinyapps.io

Showcase | About the IDN | Tools | Repositories? | Publishing? | Training | Events

The Interactive Data Network (IDN) provides support, training and consultancy in the development of interactive data visualisations of research data at Oxford University, provided by Research Support Services in IT Services.

We currently provide support in creating interactive visualisations using a variety of "point & click" and scripting tools, contact us for support in thinking about and presenting your research through web-based visualisations via researchsupport@it.ox.ac.uk

IDN Shinyapps.io

University of Oxford has a subscription to shinyapps.io where researchers may host their interactive data visualisations built using R and Shiny. To learn more about this read the IDN Shinyapps.io Guide, or email researchsupport@it.ox.ac.uk.

We provide the following support in producing visualisations for shinyapps.io:

- Showcase of examples of visualisations we've built
- Overview of what R, htmlwidgets and Shiny are useful for
- Advice on developing interactive visualisations with htmlwidgets
- Template Shiny apps for common visualisation requirements

The visualisation embedded here is the result of Case Study with the Oxford Internet Institute to build an interactive dashboard for the Online Labour Index project which seeks to provide the first economic indicator for the online gig economy. This is hosted on the IDN shinyapps.io account.

Supported Visualisation Tools

The IDN provides expertise and advice in using a wide variety of visualisation tools, click the images below for guides to the most popular tools we work with.

The IDN supports researchers in creating and hosting interactive visualisations of research data.

Read more @ idn.it.ox.ac.uk

lynda.com

LEARNING

Less about this. More about R.

Workshop Attendees: BEFORE ATTENDING PLEASE DO THE FOLLOWING

Setting up R, RStudio and all the packages

You must follow all of these steps BEFORE arriving to the workshop:

- An installation of R https://cran.r-project.org/
- An installation of RStudio https://www.rstudio.com/products/rstudio/download/

Before arriving at the workshop, please run the following code in RStudio:

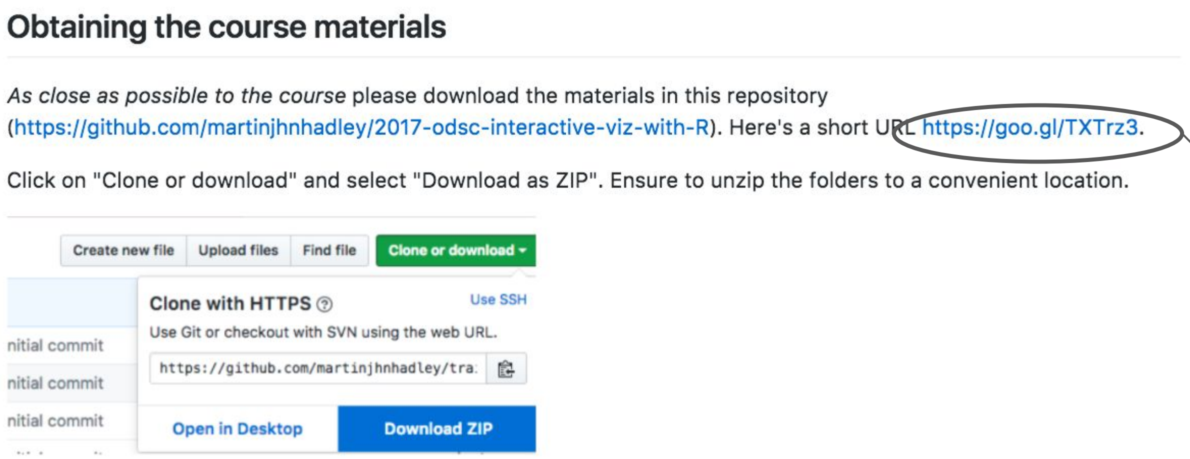
```
install.packages("tidyverse")
install.packages("gapminder")
install.packages("shiny")
install.packages("leaflet")
install.packages("highcharter")
install.packages("plotly")
install.packages("visNetwork")
```

The following package also needs to be installed, if you have ANY problems when installaing the package please refer to the instructions https://github.com/r-spatial/sf:

```
install.packages("sf")
```

Provided the above package installs correctly please also run the following code:

```
install.packages("statesRcontiguous")
```



Obtaining the course materials

As close as possible to the course please download the materials in this repository (https://github.com/martinjhnhadley/2017-odsc-interactive-viz-with-R). Here's a short URL https://goo.gl/TXTrz3.

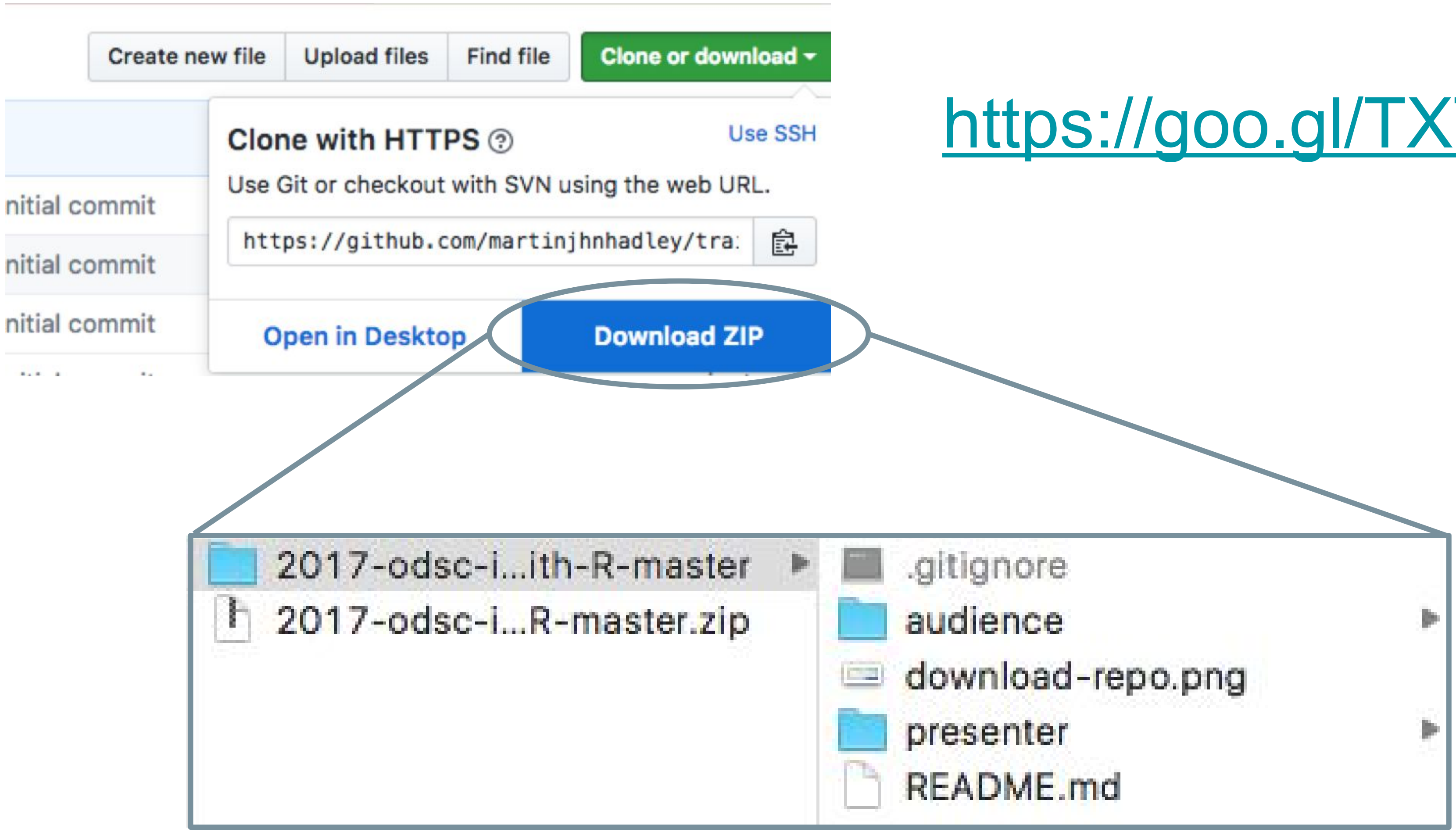
Click on "Clone or download" and select "Download as ZIP". Ensure to unzip the folders to a convenient location.

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

```
https://github.com/martinjhnhadley/tra
```

https://goo.gl/TXTrz3

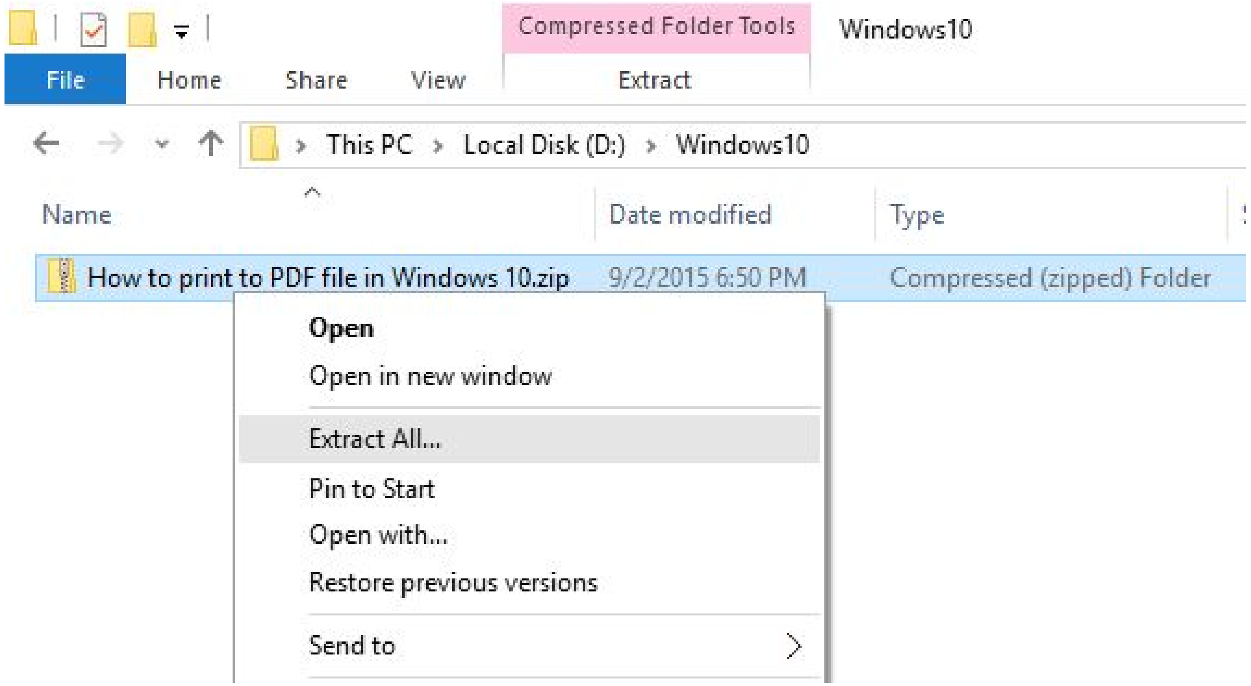

Create new file
Upload files
Find file
Clone or download
nitial commit
nitial commit
nitial commit
Clone with HTTPS
Use SSH
Use Git or checkout with SVN using the web URL.

```
https://github.com/martinjhnhadley/tra:
```

Open in Desktop
Download ZIP

```
2017-odsc-i...ith-R-master
2017-odsc-i...R-master.zip
.gitignore
audience
download-repo.png
presenter
README.md
```

https://goo.gl/TXTrz3


Compressed Folder Tools
Windows10
File
Home
Share
View
Extract
This PC › Local Disk (D:) › Windows10
Name
Date modified
Type
How to print to PDF file in Windows 10.zip
9/2/2015 6:50 PM
Compressed (zipped) Folder
Open
Open in new window
Extract All...
Pin to Start
Open with...
Restore previous versions
Send to

RStudio

tidyverse

%>%

Ceci n'est pas un pipe.

htmlwidgets

%>%

Ceci n'est pas un pipe.

Shiny

RStudio Connect

rmarkdown

R is the programming (or scripting) language we're using to make interactive viz!

If you need a quick intro:
datacamp.com/courses/intro-to-statistics-with-r-introduction

RStudio is *the* IDE for data science (and package development) for R.

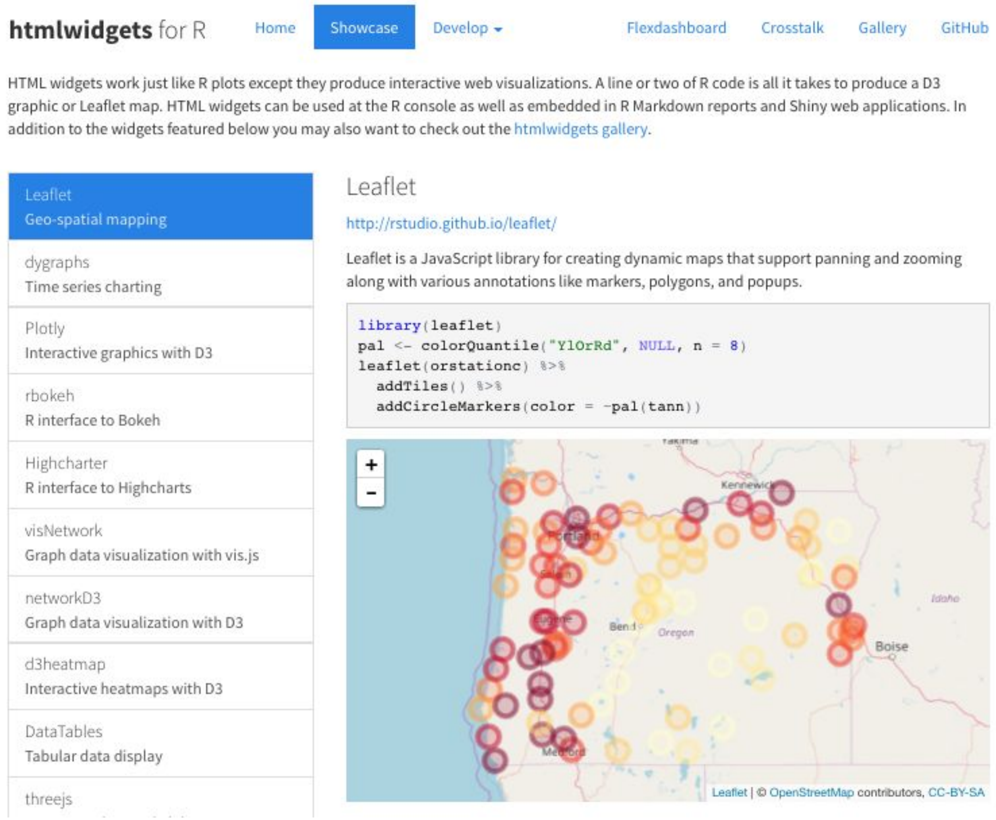
The company behind it - RStudio - build tools which work seamlessly together to make your lives easier (and to make R more powerful).

htmlwidgets is a framework for building bindings between JavaScript libraries and R.

htmlwidgets is a framework for building bindings between JavaScript libraries and R.

htmlwidgets allows develops to build their own R packages that provide end-users with access to these high-level JavaScript libraries with R code.

htmlwidgets.org



htmlwidgets for R

HTML widgets work just like R plots except they produce interactive web visualizations. A line or two of R code is all it takes to produce a D3 graphic or Leaflet map. HTML widgets can be used at the R console as well as embedded in R Markdown reports and Shiny web applications. In addition to the widgets featured below you may also want to check out the htmlwidgets gallery.

Leaflet
Geo-spatial mapping

dygraphs
Time series charting

Plotly
Interactive graphics with D3

rbokeh
R interface to Bokeh

Highcharter
R interface to Highcharts

visNetwork
Graph data visualization with vis.js

networkD3
Graph data visualization with D3

d3heatmap
Interactive heatmaps with D3

DataTables
Tabular data display

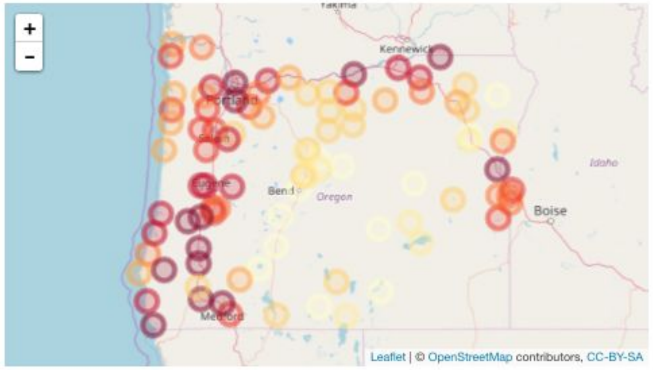
threejs

Leaflet

http://rstudio.github.io/leaflet/

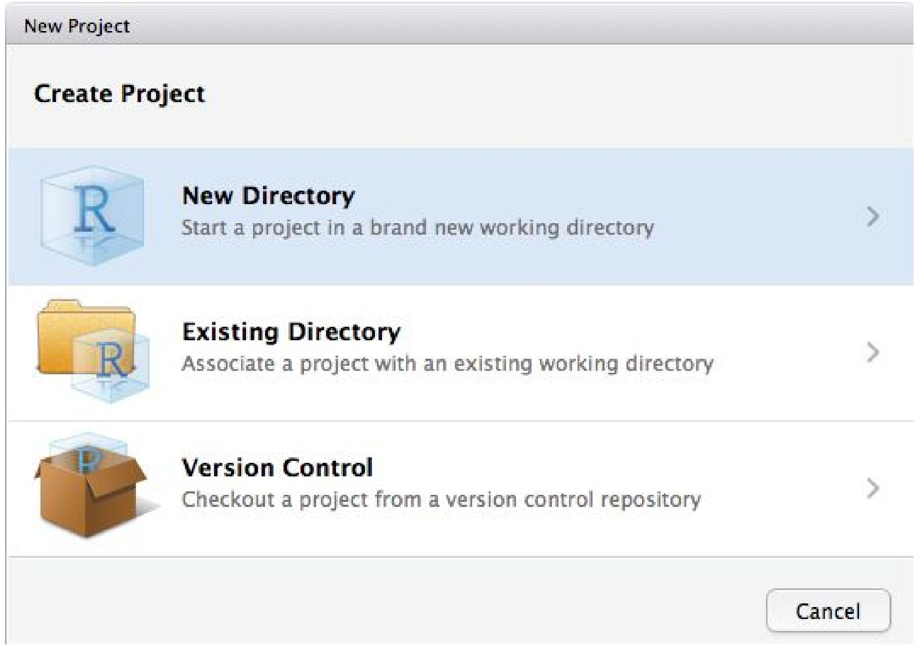
Leaflet is a JavaScript library for creating dynamic maps that support panning and zooming along with various annotations like markers, polygons, and popups.

```
library(leaflet)
pal <- colorQuantile("YlOrRd", NULL, n = 8)
leaflet(orstationc) %>%
  addTiles() %>%
  addCircleMarkers(color = ~pal(tann))
```



htmlwidgets

Leaflet


New Project

Create Project

New Directory
Start a project in a brand new working directory

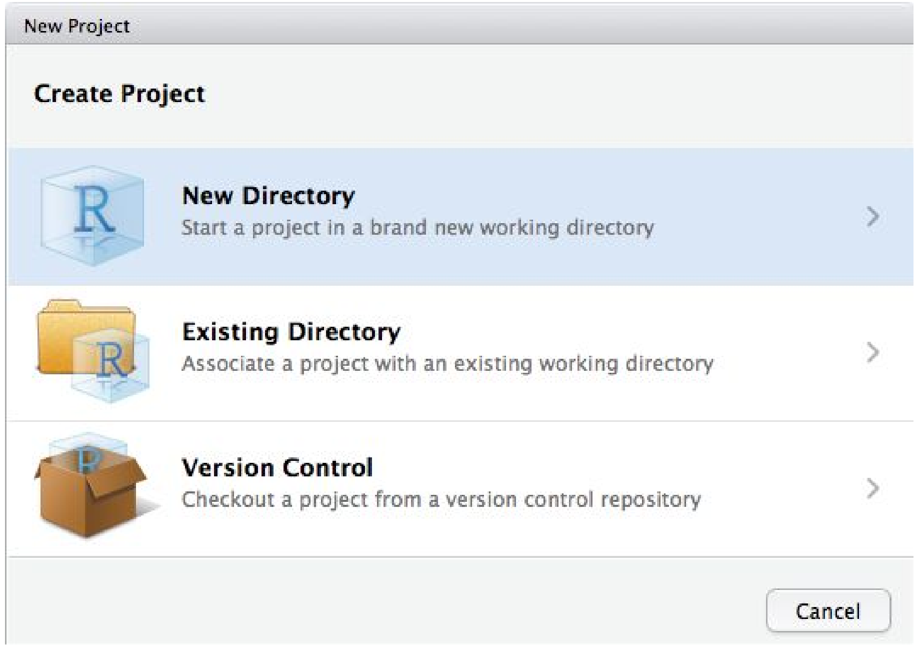
Existing Directory
Associate a project with an existing working directory

Version Control
Checkout a project from a version control repository

Cancel

Make your life easy by making your work reproducible.

Start as you mean to go on.


New Project

Create Project

New Directory
Start a project in a brand new working directory

Existing Directory
Associate a project with an existing working directory

Version Control
Checkout a project from a version control repository

Cancel

Projects make file paths easy.

Projects give you structure.

Projects make code portable.

Leaflet

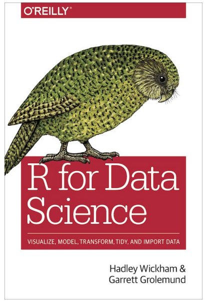
```
> install.packages("leaflet")
```

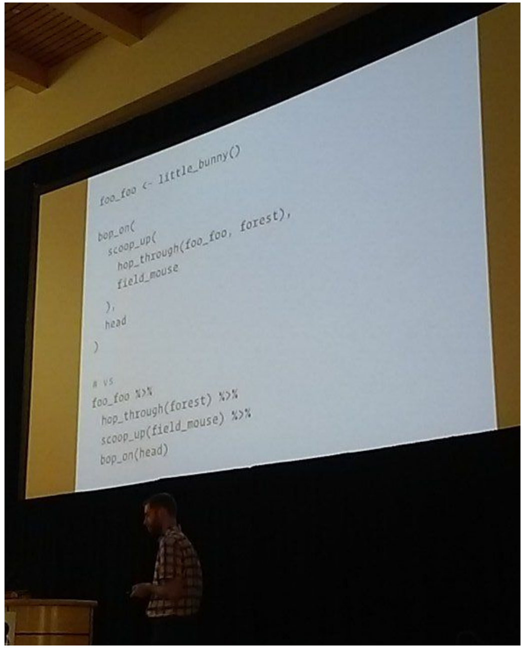
```
library("leaflet")
```

```
> head(quakes)
     lat   long depth mag stations
1 -20.42 181.62   562 4.8       41
2 -20.62 181.03   650 4.2       15
3 -26.00 184.10    42 5.4       43
4 -17.97 181.66   626 4.1       19
5 -20.42 181.96   649 4.0       11
6 -19.68 184.31   195 4.0       12
```

```r
quakes %>%
  leaflet() %>%
  addTiles() %>%
  addCircleMarkers()
```

I think the best motivating example for %>% comes from Hadley Wickham’s talk in useR2016.


O'REILLY
R for Data Science
VISUALIZE, MODEL, TRANSFORM, TIDY, AND IMPORT DATA
Hadley Wickham & Garrett Grolemund

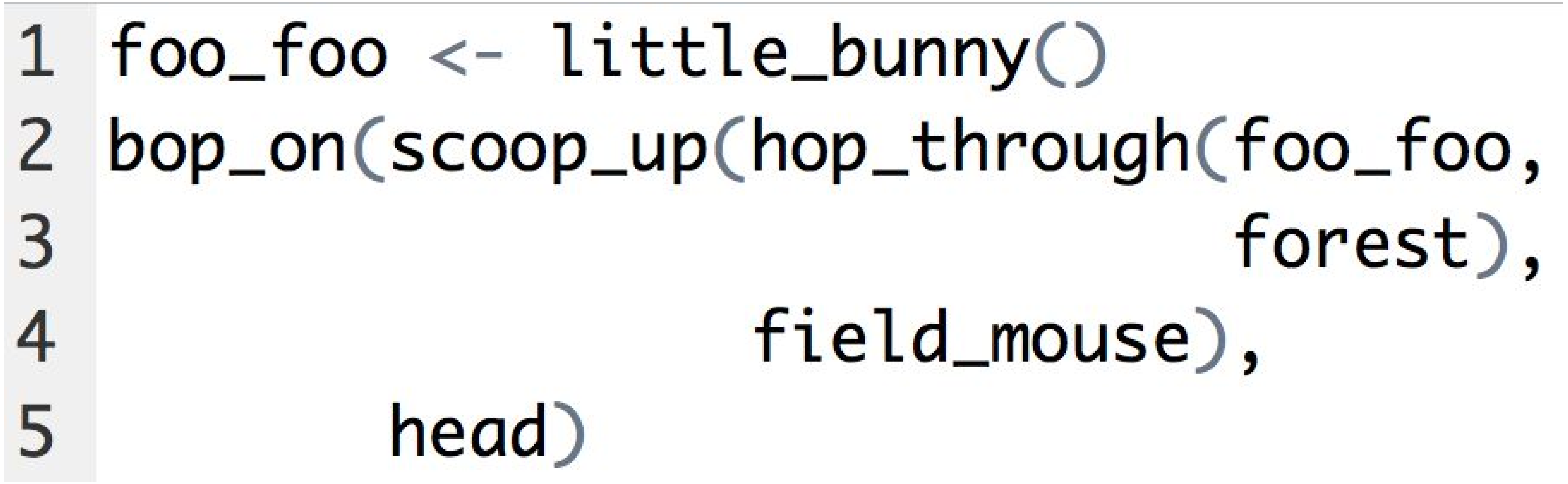


```r
foo_foo <- little_bunny()

bop_on(
  scoop_up(
    hop_through(foo_foo, forest),
    field_mouse
  ),
  head
)

# vs
foo_foo %>%
  hop_through(forest) %>%
  scoop_up(field_mouse) %>%
  bop_on(head)
```

Citation:
twitter.com/AmeliaMN/status/748193609401327616

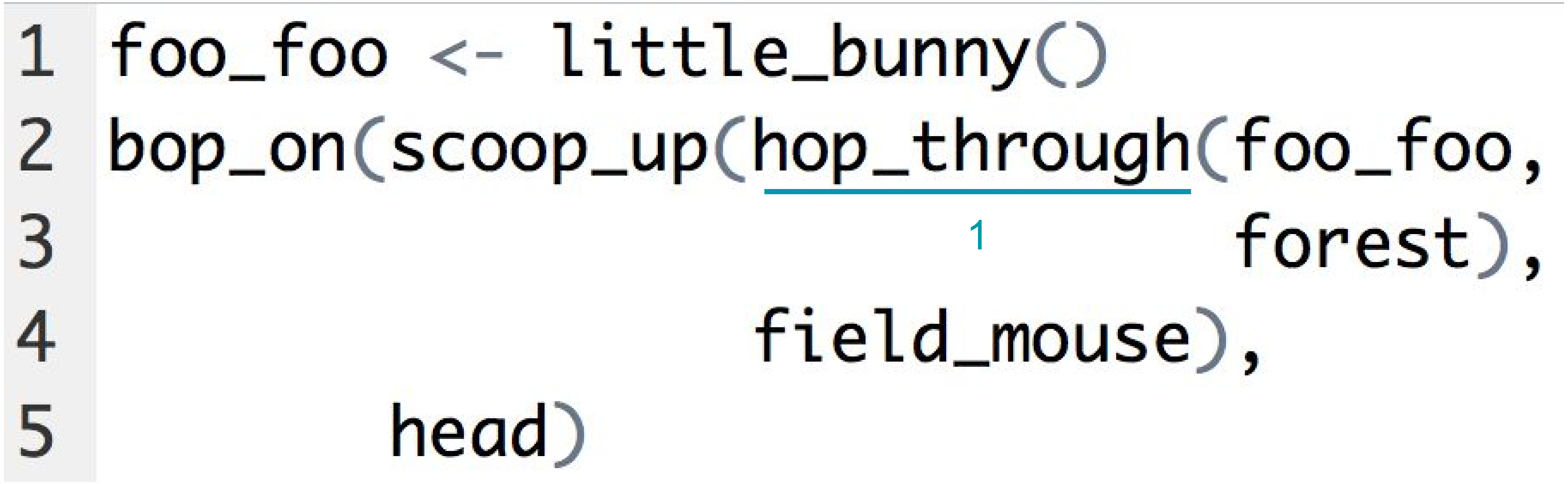
Little bunny Foo Foo
Went hopping through the forest
Scooping up the field mice
And bopping them on the head

Little bunny Foo Foo
Went hopping through the forest
Scooping up the field mice
And bopping them on the head



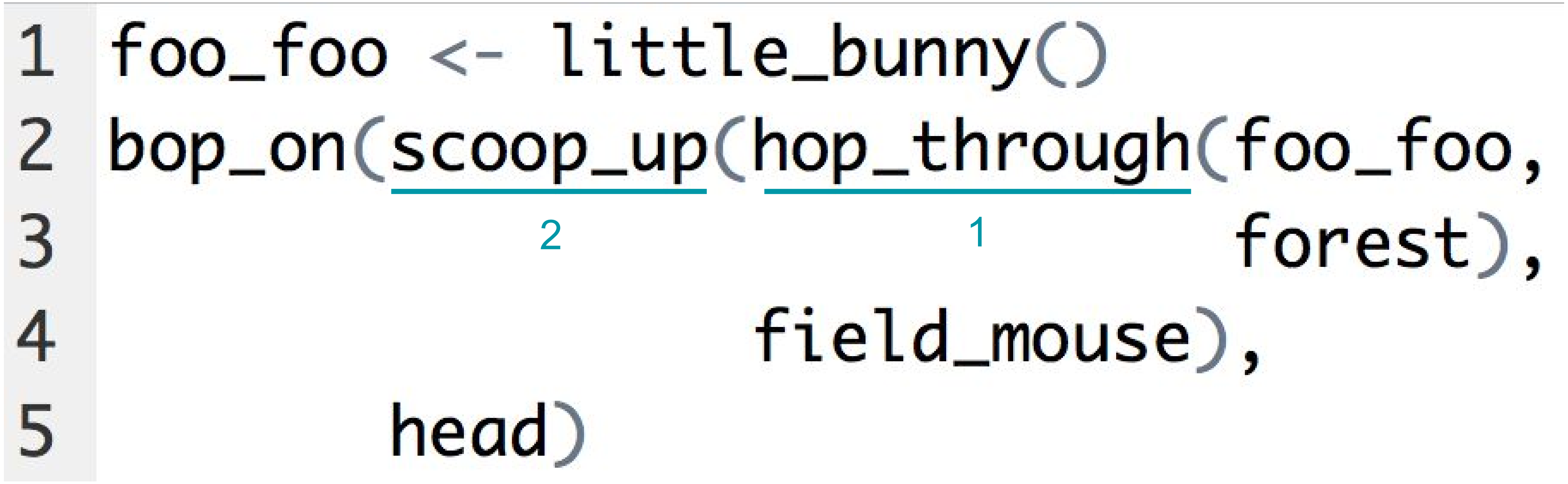
```
foo_foo <- little_bunny()
bop_on(scoop_up(hop_through(foo_foo,
                            forest),
                field_mouse),
       head)
```

Little bunny Foo Foo
Went hopping through the forest
Scooping up the field mice
And bopping them on the head



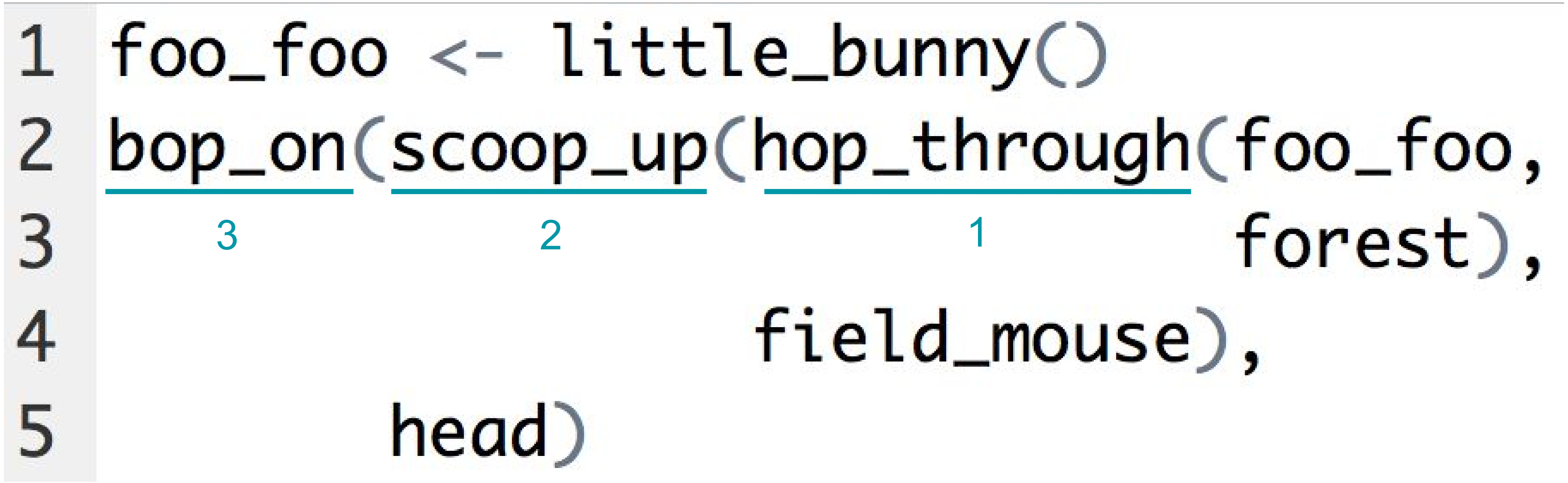
```
foo_foo <- little_bunny()
bop_on(scoop_up(hop_through(foo_foo,
                            forest),
                field_mouse),
       head)
```

Little bunny Foo Foo
Went hopping through the forest
Scooping up the field mice
And bopping them on the head



```
foo_foo <- little_bunny()
bop_on(scoop_up(hop_through(foo_foo,
                            forest),
                field_mouse),
       head)
```

Little bunny Foo Foo
Went hopping through the forest
Scooping up the field mice
And bopping them on the head



```
foo_foo <- little_bunny()
bop_on(scoop_up(hop_through(foo_foo,
                            forest),
                field_mouse),
       head)
```

```r
foo_foo <- little_bunny()
bop_on(scoop_up(hop_through(foo_foo,
    3       2           1       forest),
                      field_mouse),
          head)
```

```r
foo_foo <- little_bunny()
foo_foo %>%
  hop_through(forest) %>%
  scoop_up(field_mouse) %>%
  bop_on(head)
```

```r
foo_foo <- little_bunny()
bop_on(scoop_up(hop_through(foo_foo,
   3         2            1          forest),
                          field_mouse),
       head)
```

```r
foo_foo %>%
  hop_through(forest)
```

```r
hop_through(foo_foo, forest)
```

```r
quakes %>%
  leaflet() %>%
  addTiles() %>%
  addCircleMarkers()
```

leaflet provides access to values in columns via ~

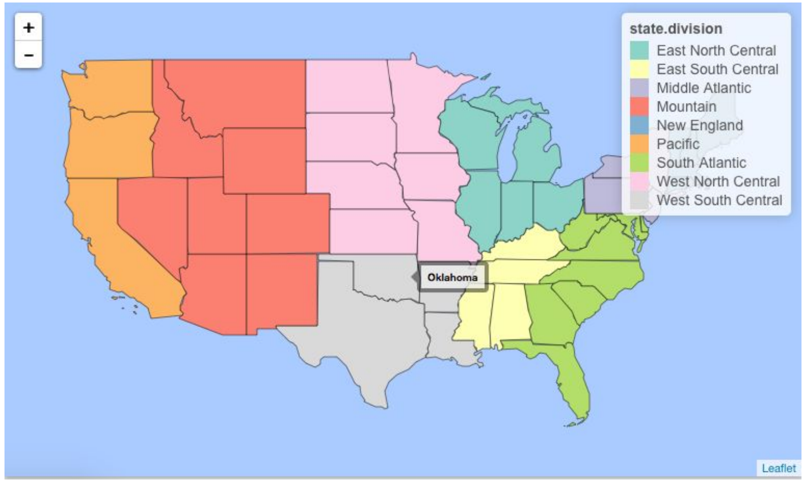
```
quakes %>%
  leaflet() %>%
  addTiles() %>%
  addCircleMarkers(label = ~mag)
```

Any guesses why this doesn't quite work as you'd expect?

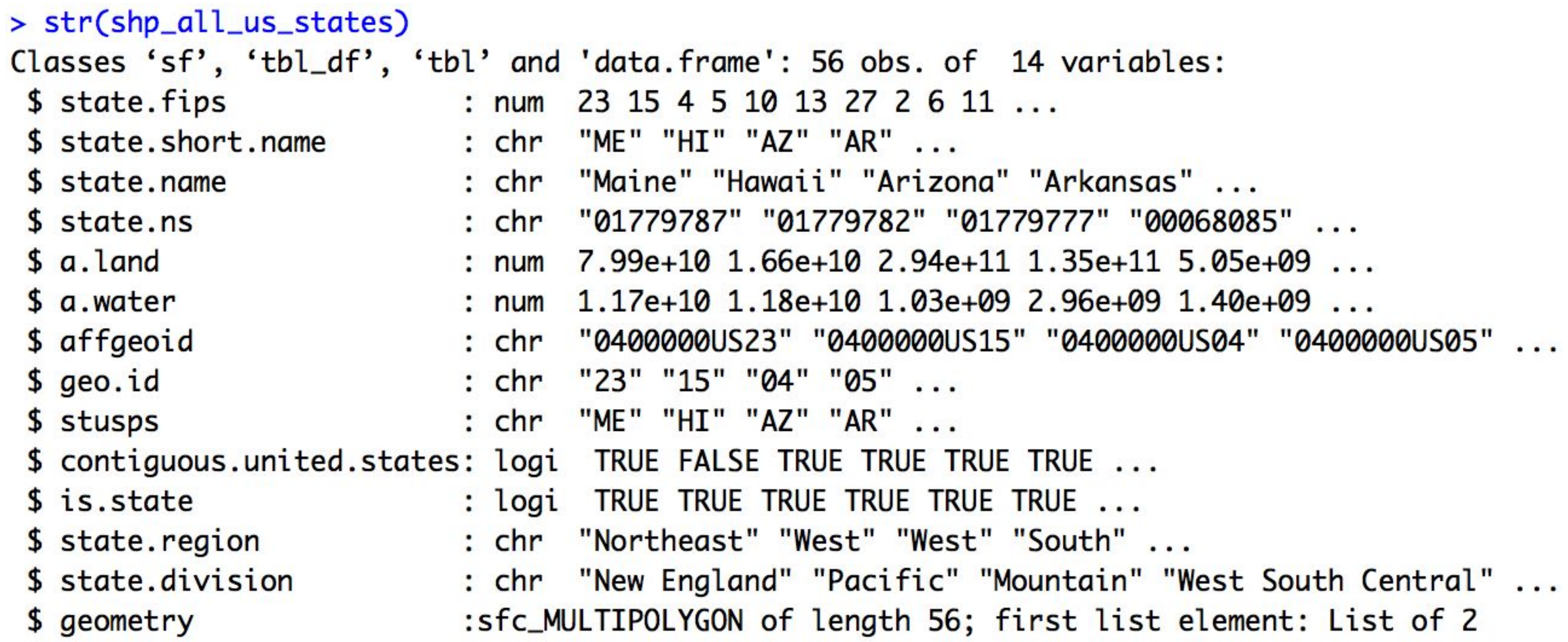
As htmlwidgets are generating HTML and JavaScript, be careful to provide [valid] strings as labels etc

```
quakes %>%
  leaflet() %>%
  addTiles() %>%
  addCircleMarkers(label = ~as.character(mag))
```

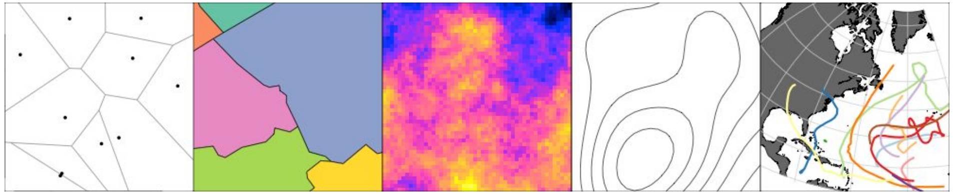
Choropleth


+
-
state.division
🟩 East North Central
🟨 East South Central
🟪 Middle Atlantic
🟥 Mountain
🟦 New England
🟧 Pacific
🟩 South Atlantic
🟪 West North Central
⬜ West South Central
Oklahoma
Leaflet

library(statesRcontiguous)



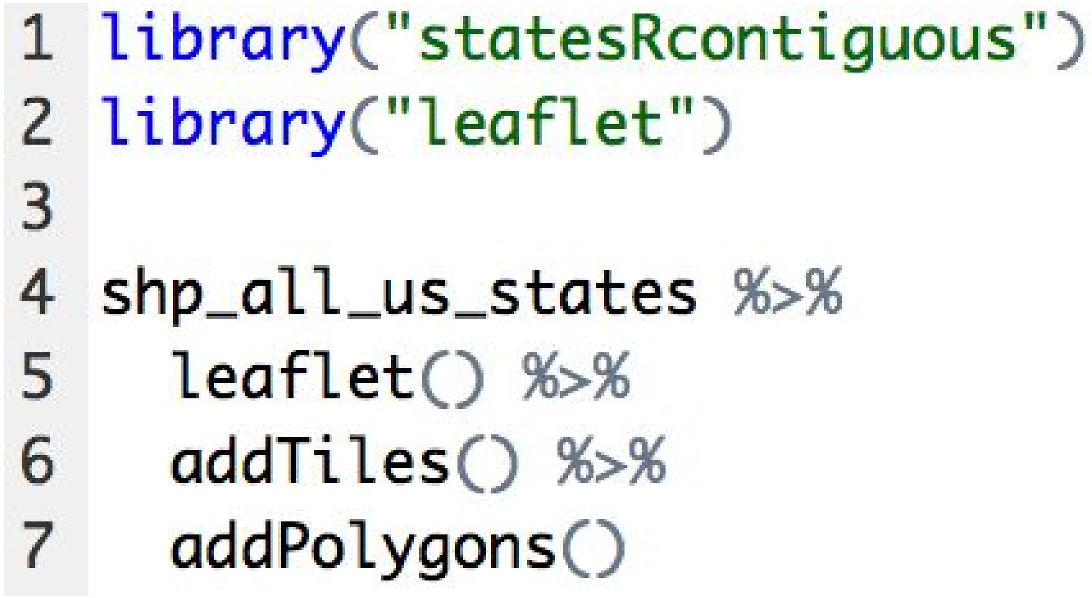
```
> str(shp_all_us_states)
Classes ‘sf’, ‘tbl_df’, ‘tbl’ and 'data.frame': 56 obs. of  14 variables:
 $ state.fips             : num  23 15 4 5 10 13 27 2 6 11 ...
 $ state.short.name       : chr  "ME" "HI" "AZ" "AR" ...
 $ state.name             : chr  "Maine" "Hawaii" "Arizona" "Arkansas" ...
 $ state.ns               : chr  "01779787" "01779782" "01779777" "00068085" ...
 $ a.land                 : num  7.99e+10 1.66e+10 2.94e+11 1.35e+11 5.05e+09 ...
 $ a.water                : num  1.17e+10 1.18e+10 1.03e+09 2.96e+09 1.40e+09 ...
 $ affgeoid               : chr  "0400000US23" "0400000US15" "0400000US04" "0400000US05" ...
 $ geo.id                 : chr  "23" "15" "04" "05" ...
 $ stusps                 : chr  "ME" "HI" "AZ" "AR" ...
 $ contiguous.united.states: logi  TRUE FALSE TRUE TRUE TRUE TRUE ...
 $ is.state               : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
 $ state.region           : chr  "Northeast" "West" "West" "South" ...
 $ state.division         : chr  "New England" "Pacific" "Mountain" "West South Central" ...
 $ geometry               :sfc_MULTIPOLYGON of length 56; first list element: List of 2
```



The sf library provides a consistent (and extremely powerful) workflow for GIS with R - I highly recommend it.

Read more here: https://r-spatial.github.io/sf/

Leaflet



```r
library("statesRcontiguous")
library("leaflet")

shp_all_us_states %>%
  leaflet() %>%
  addTiles() %>%
  addPolygons()
```

tidyverse

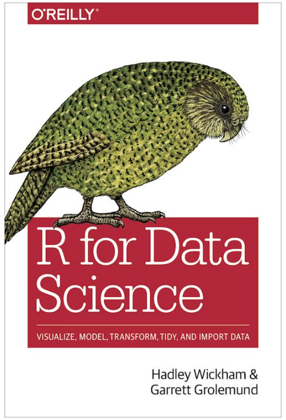
The tidyverse provides a consistent and elegant approach to doing data science with R.

The tidyverse is an ecosystem of packages maintained and developed by the folks at RStudio.

Development of the tidyverse packages started in late 2014 but only became formalised as the “tidyverse” in late 2016.

We’re going to jump in and out of the tidyverse.

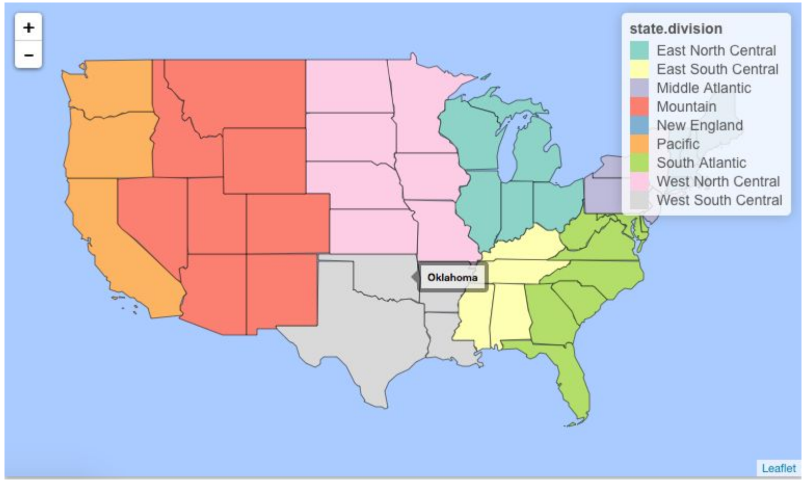
The R for Data Science book is a deep-dive into the tidyverse and available freely here: r4ds.had.co.nz

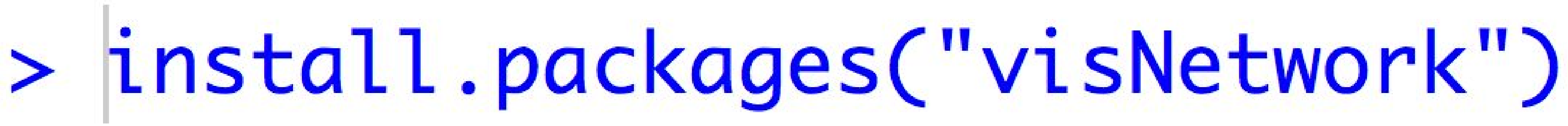

O'REILLY
R for Data Science
VISUALIZE, MODEL, TRANSFORM, TIDY, AND IMPORT DATA
Hadley Wickham & Garrett Grolemund

```r
library("tidyverse")
contiguous_states <- shp_all_us_states %>%
  filter(contiguous.united.states == TRUE)
```

```r
library("RColorBrewer")
palette_state_divisions <-
  colorFactor(brewer.pal(9, "Set3"),
              contiguous_states$state.division)
```

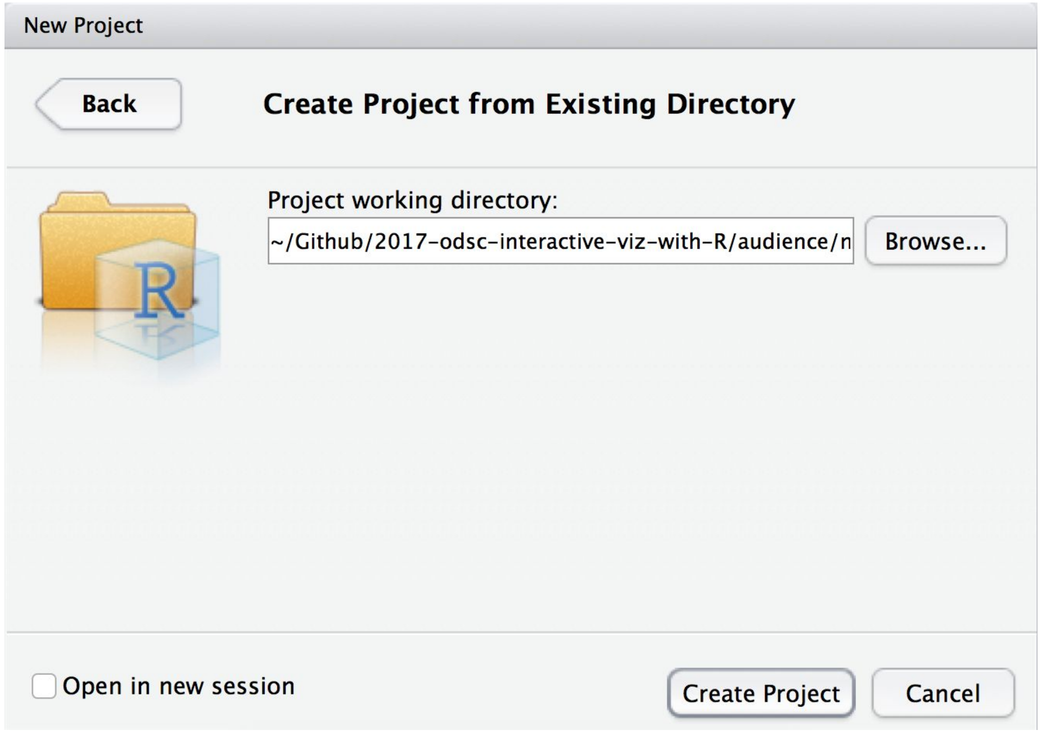
```
contiguous_states %>%
  leaflet() %>%
  addPolygons(fillColor = ~ palette_state_divisions(state.division)) %>%
  addLegend(pal = palette_state_divisions,
            values = state.division)
```





```
> install.packages("visNetwork")
```

vis.js


New Project

Back

Create Project from Existing Directory

Project working directory:

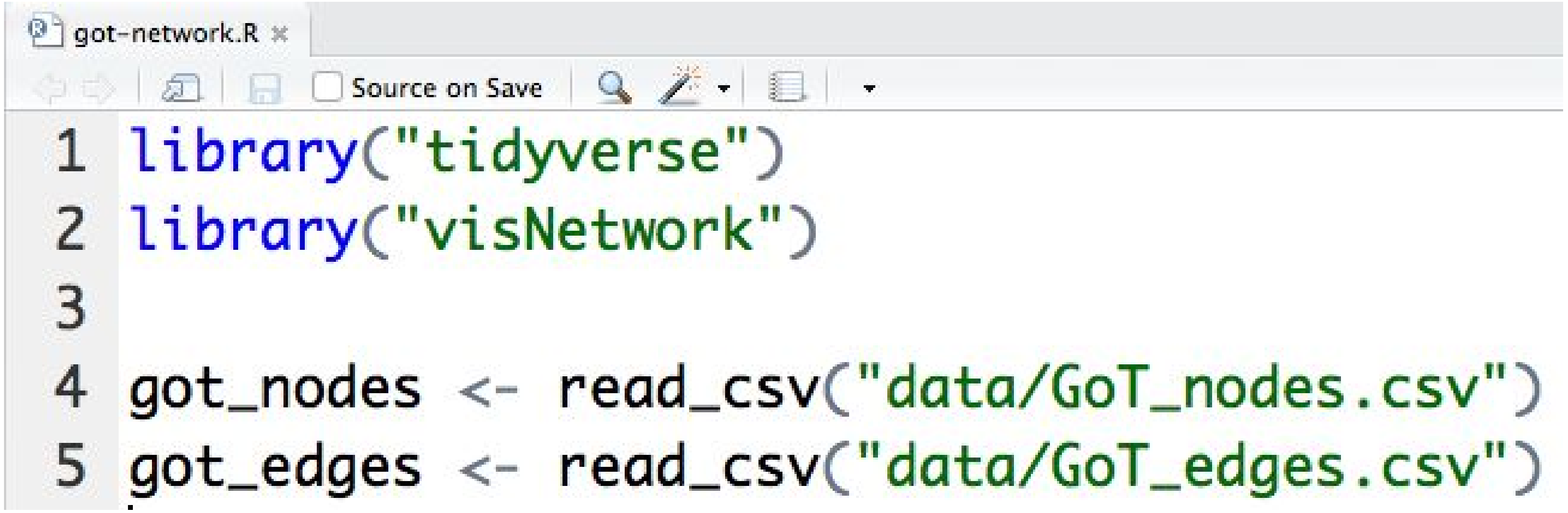
~/Github/2017-odsc-interactive-viz-with-R/audience/n

Browse...

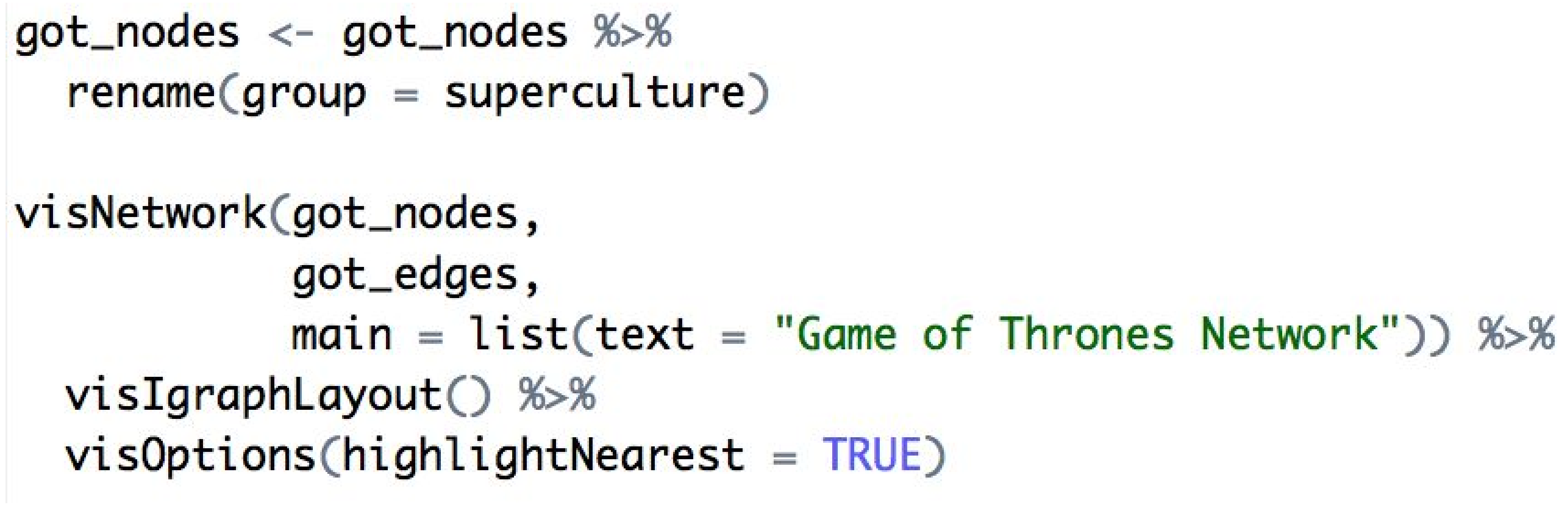
☐ Open in new session

Create Project

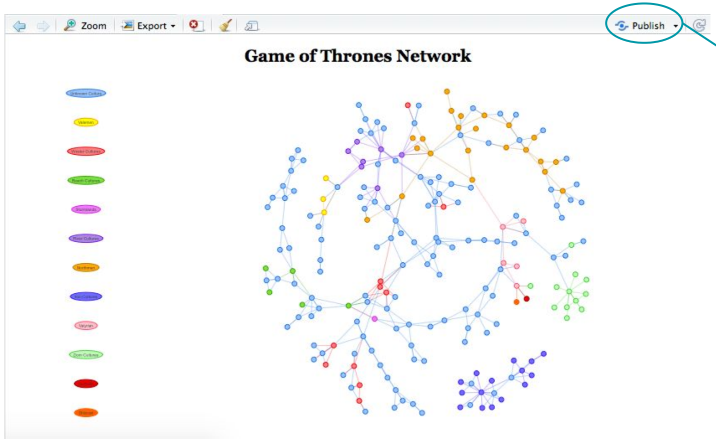
Cancel

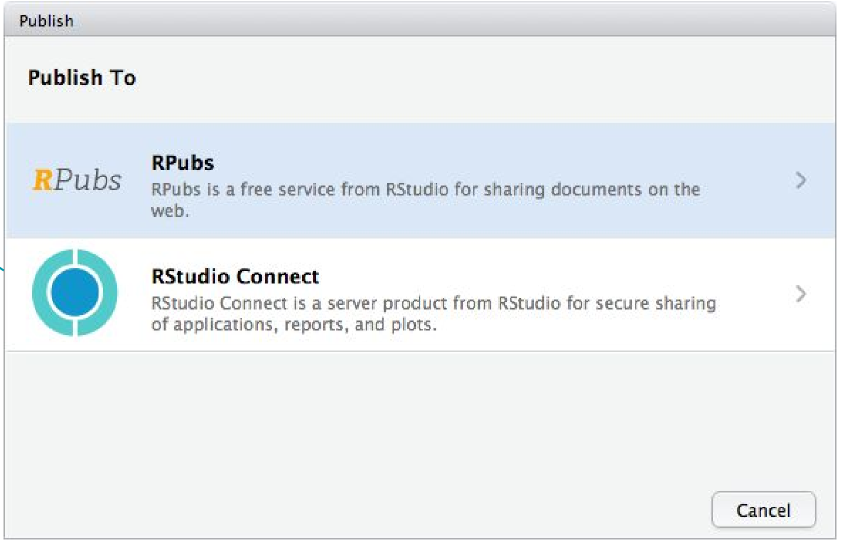


```r
library("tidyverse")
library("visNetwork")

got_nodes <- read_csv("data/GoT_nodes.csv")
got_edges <- read_csv("data/GoT_edges.csv")
got_edges <- got_edges %>%
  rename(from = source,
         to = target)

visNetwork(got_nodes,
           got_edges) %>%
  visIgraphLayout()
```



```r
got_nodes <- got_nodes %>%
  rename(group = superculture)

visNetwork(got_nodes,
           got_edges,
           main = list(text = "Game of Thrones Network")) %>%
  visIgraphLayout() %>%
  visOptions(highlightNearest = TRUE)
```




Publish

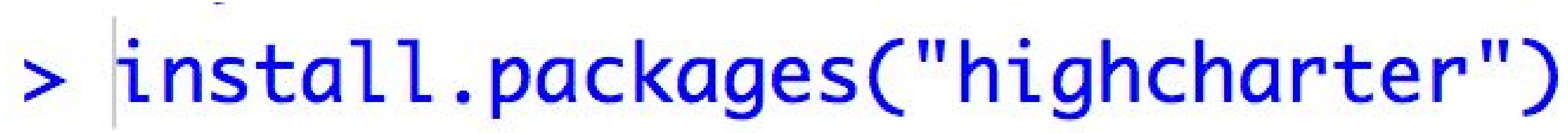
Publish To

RPubs
RPubs is a free service from RStudio for sharing documents on the web.

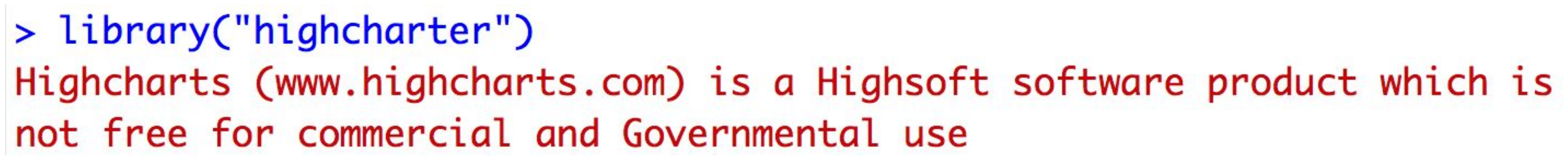
RStudio Connect
RStudio Connect is a server product from RStudio for secure sharing of applications, reports, and plots.

Cancel

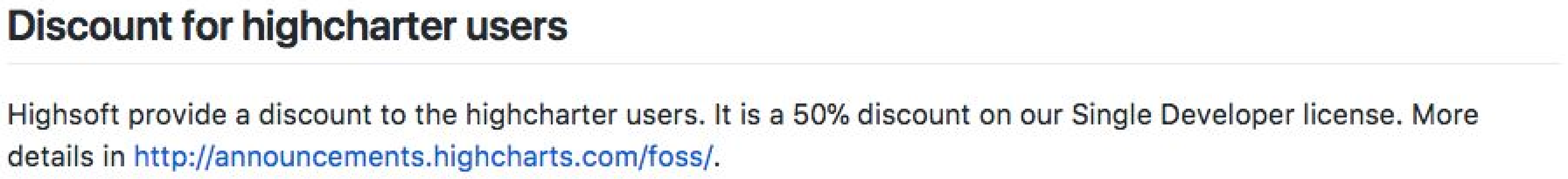
HIGHCHARTS



```
> install.packages("highcharter")
```



```
> library("highcharter")
Highcharts (www.highcharts.com) is a Highsoft software product which is
not free for commercial and Governmental use
```

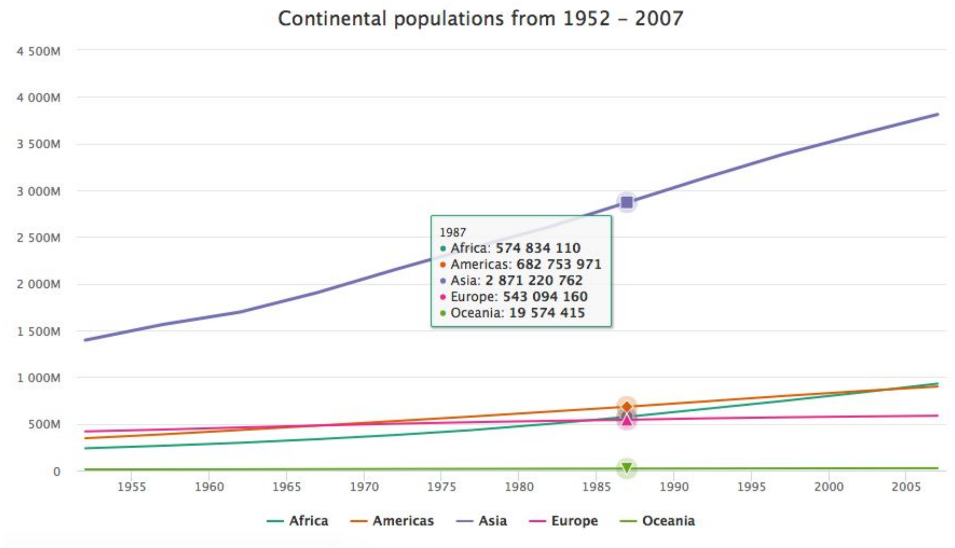


Discount for highcharter users

Highsoft provide a discount to the highcharter users. It is a 50% discount on our Single Developer license. More details in http://announcements.highcharts.com/foss/.

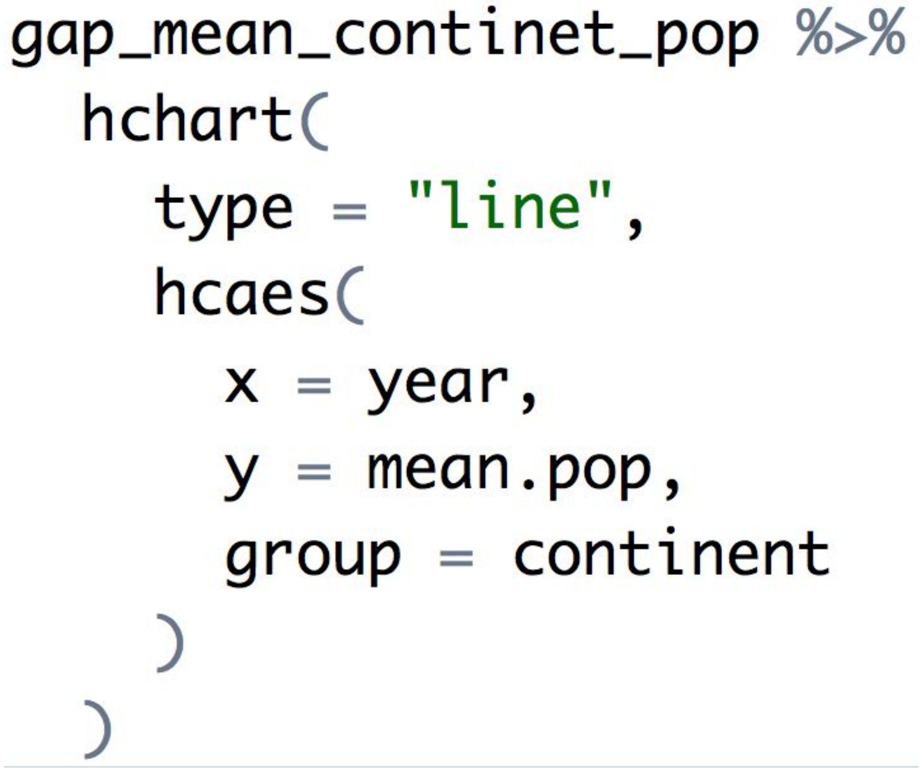
HIGHCHARTS

gapminder



HIGHCHARTS

```
gapminder %>%
   group_by(year, continent) %>%
   summarise(mean.pop = mean(pop))
```



```r
gap_mean_continet_pop %>%
  hchart(
    type = "line",
    hcaes(
      x = year,
      y = mean.pop,
      group = continent
    )
  )
```

highcharter uses a similar approach to ggplot2 for accessing data from data.frames

HIGHCHARTS

api.highcharts.com

```r
my_chart %>%
  hc_xAxis(title = "Year") %>%
  hc_yAxis(title = "") %>%
  hc_tooltip(
    shared = TRUE,
    valueDecimals = 0
  )
```

Can we make a custom style for highcharter objects?

api.highcharts.com

```
hc_style_gapminder <- function(hc){
  hc %>%
    hc_xAxis(title = "Year")
}
```

htmlwidgets allow us to create a vast range of different types of interactive visualisation with R code!

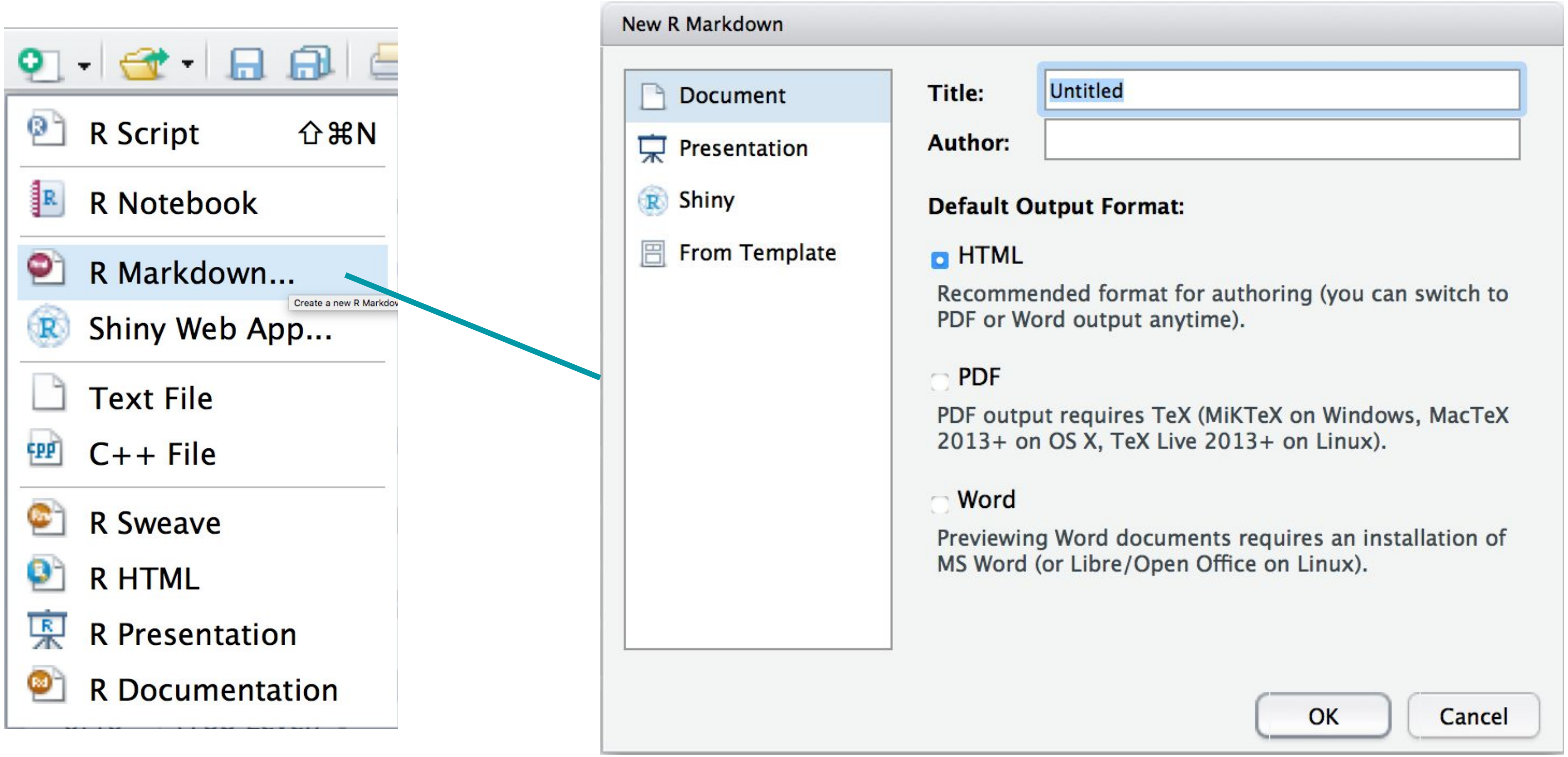
Each htmlwidgets library is a beast in its own right, you'll need to read the documentation (htmlwidgets.org)

Individual htmlwidgets may be deployed to RPubs.com!

RMarkdown is a simple way to build reports and presentations that include R code.

RMarkdown can output html reports and presentations that include htmlwidgets.

RMarkdown reports can easily and freely be published to RPubs.com



Title: Untitled

Author:

Default Output Format:

[✓ HTML]

Recommended format for authoring (you can switch to PDF or Word output anytime).

PDF

PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

Word

Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).

Use caution creating RMarkdown presentations that include htmlwidgets.

You may need to use the widgetframe library to make htmlwidgets play nicely.

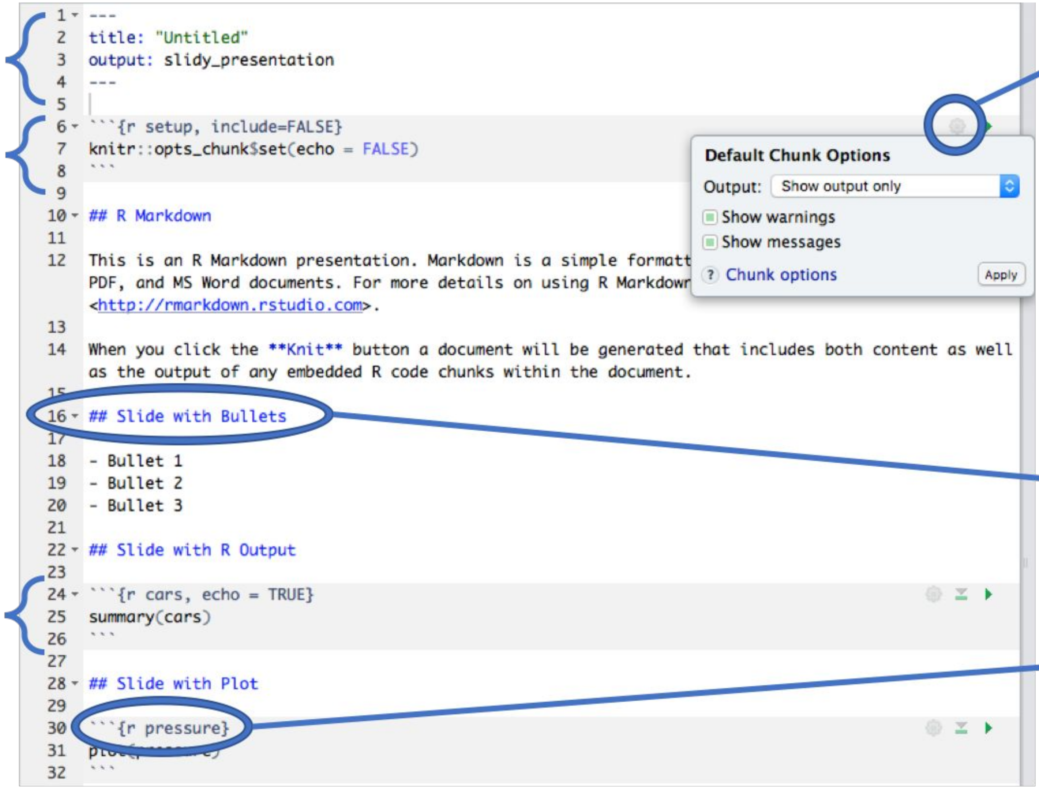
RPubs.com does not support documents using the widgetframe library.

If you need to use widgetframe in RMarkdown documents, consider using GitHub Pages to host your files.

preamble (or header) which tells RStudio what to output (and how), written in YAML

special code chunk that sets "global" code chunk options

code chunk: both output and the code will be displayed because of the option echo = TRUE



~~~
---
title: "Untitled"
output: slidy_presentation
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE)
~~~

R Markdown

This is an R Markdown presentation. Markdown is a simple formatt
PDF, and MS Word documents. For more details on using R Markdowr
<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.

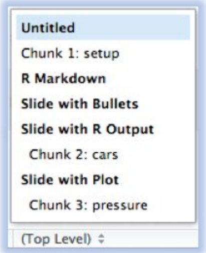
Slide with Bullets

- Bullet 1
- Bullet 2
- Bullet 3

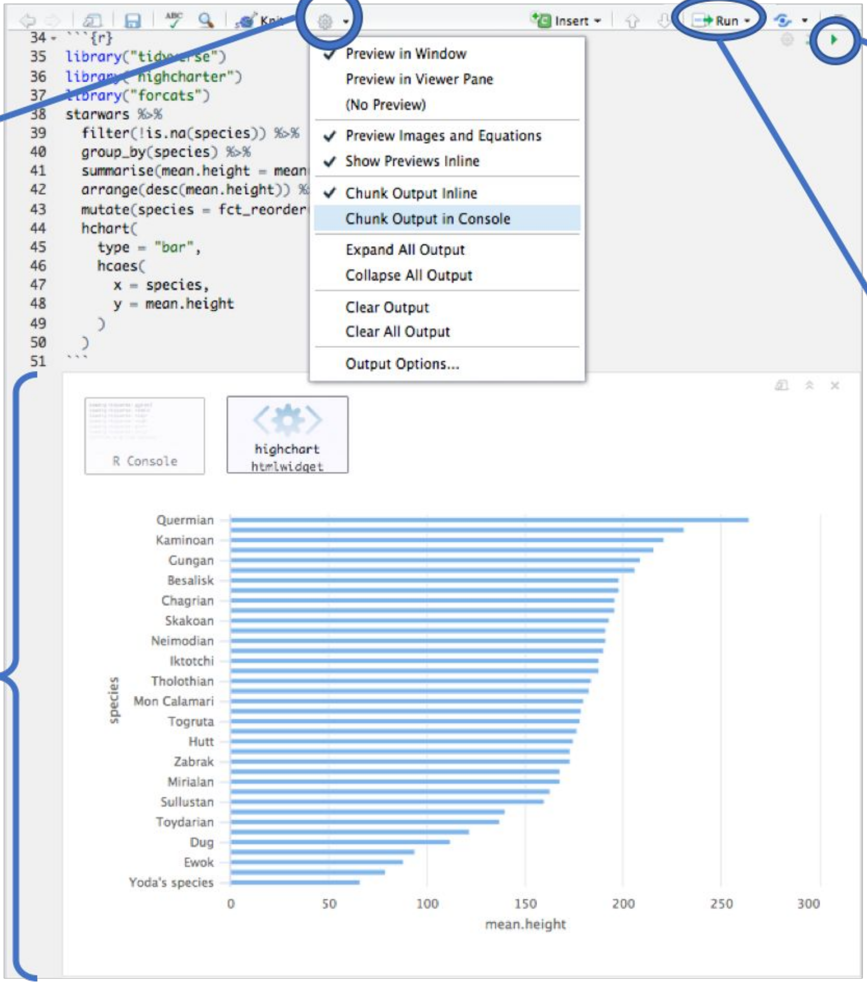
Slide with R Output

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

Slide with Plot

```{r pressure}
plot(pressure)
```
```

Default Chunk Options

Output: Show output only

☐ Show warnings

☐ Show messages

clicking the cog provides an interface for changing code chunk options (the global options in **this** case)

represent subheadings in the case of output: X_presentation documents these deliminate slides

naming code chunks allows you to easily navigate through your document and to find errors later


```

click this cog to access the options for the RMarkdown document, including “Chunk Output in Console”



~~~
```{r}
library("tid   rse")
library("highcharter")
library("forcats")
starwars %>%
  filter(!is.na(species)) %>%
  group_by(species) %>%
  summarise(mean.height = mean
  arrange(desc(mean.height)) %
  mutate(species = fct_reorder
  hchart(
    type = "bar",
    hcaes(
      x = species,
      y = mean.height
    )
  )
```
~~~

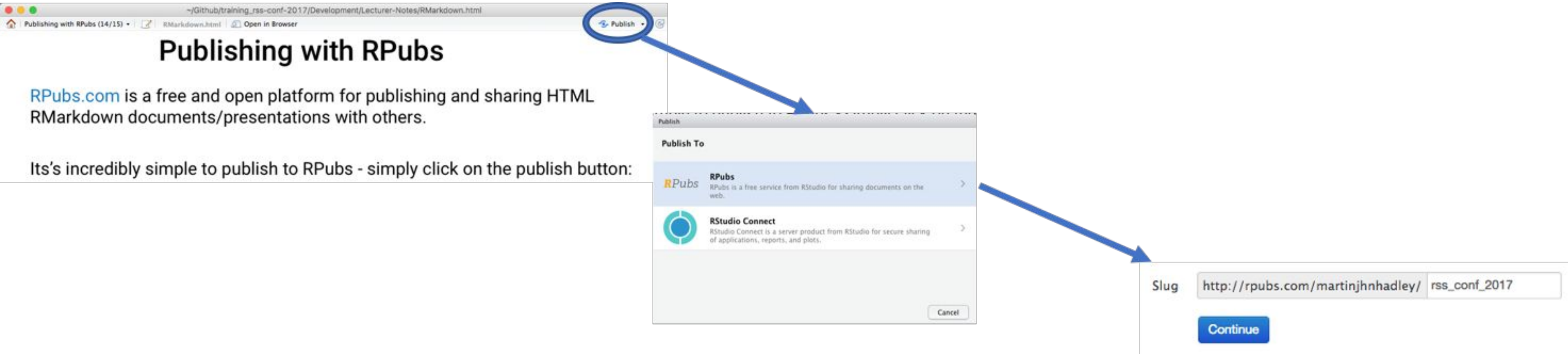
clicking the ”play” button on a code chunk will run *just* that code chunk

click here for both the commands and keyboard shortcuts to evaluate:

- all code chunks
- all codes chunks above/below current chunk
- evaluate current code chunk

output appears directly below code chunks, rather than in the console or viewer panel. This output has two elements:

1. messages that would be printed to the console
2. the htmlwidget output itself



~/Github/training_rss-conf-2017/Development/Lecturer-Notes/RMarkdown.html

Publishing with RPubs

RPubs.com is a free and open platform for publishing and sharing HTML RMarkdown documents/presentations with others.

Its's incredibly simple to publish to RPubs - simply click on the publish button:

Publish

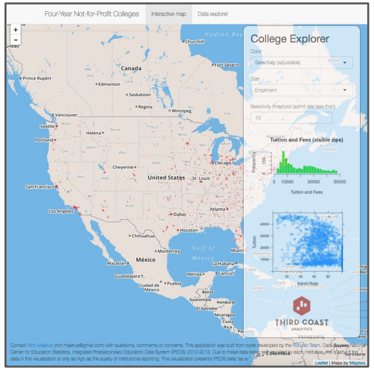
Publish To

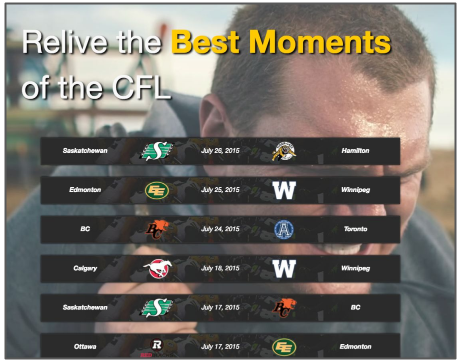
RPubs
RPubs is a free service from RStudio for sharing documents on the web.

RStudio Connect
RStudio Connect is a server product from RStudio for secure sharing of applications, reports, and plots.

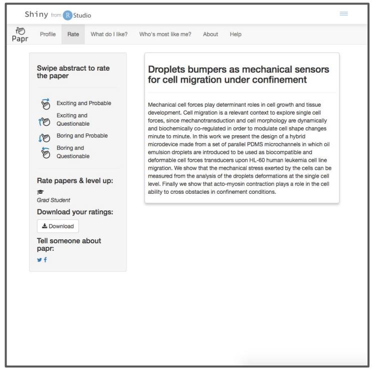
Slug http://rpubs.com/martinjhnhadley/ rss_conf_2017

Shiny is a framework for creating interactive web applications using R.





Saskatchewan	July 26, 2015	Hamilton
Edmonton	July 25, 2015	Winnipeg
BC	July 24, 2015	Toronto
Calgary	July 18, 2015	Winnipeg
Saskatchewan	July 17, 2015	BC
Ottawa	July 17, 2015	Edmonton



Swipe abstract to rate the paper

- Exciting and Probable
- Exciting and Questionable
- Boring and Probable
- Boring and Questionable

Rate papers & level up:

Grad Student

Download your ratings:

Tell someone about papr:

Droplets bumpers as mechanical sensors for cell migration under confinement

Mechanical cell forces play determinant roles in cell growth and tissue development. Cell migration is a relevant context to explore single cell forces, since mechanotransduction and cell morphology are dynamically and biochemically co-regulated in order to modulate cell shape changes minute to minute. In this work we present the design of a hybrid microdevice made from a set of parallel PDMS microchannels in which oil emulsion droplets are introduced to be used as biocompatible and deformable cell forces transducers upon HL-60 human leukemia cell line migration. We show that the mechanical stress exerted by the cells can be measured from the analysis of the droplets deformations at the single cell level. Finally we show that acto-myosin contraction plays a role in the cell ability to cross obstacles in confinement conditions.

Shiny on your local machine

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

Shiny on a server (for others to use)

shinyapps.io

Fully hosted solution for Shiny apps

(Includes a free tier!)

Shiny Server

RStudio Connect

shinyapps.io

Fully hosted solution for Shiny apps

(Includes a free tier!)

Shiny Server

Shiny Server Open Source

(Non-commercial use)

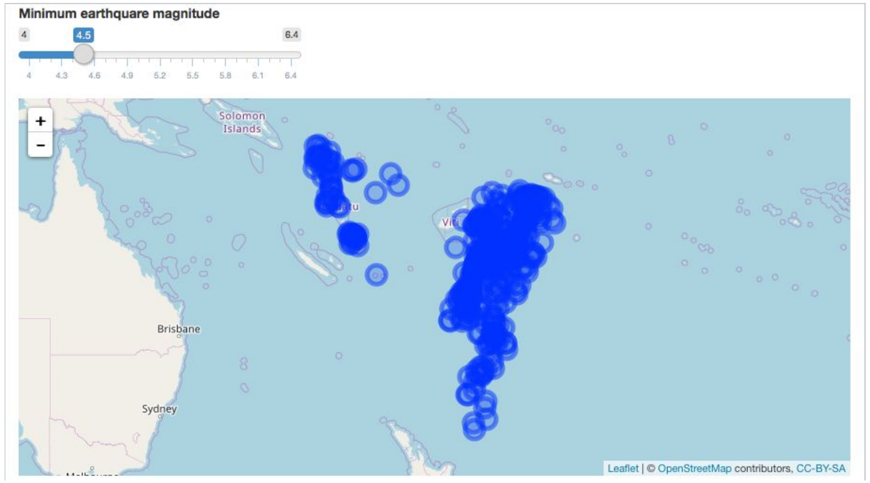
Shiny Server Pro

(Commercial use)

RStudio Connect

shinyapps.io	Fully hosted solution for Shiny apps (Includes a free tier!)
Shiny Server	Shiny Server Open Source (Non-commercial use) | Shiny Server Pro (Commercial use)
RStudio Connect	RStudio Connect (Commercial use, 45 day evaluation license available)

Basic Shiny App


Minimum earthquake magnitude
4
4.5
6.4
4 4.3 4.6 4.9 5.2 5.5 5.8 6.1 6.4
Solomon Islands
Brisbane
Sydney
Leaflet | © OpenStreetMap contributors, CC-BY-SA

server.R

```r
function(input, output){

  output$leaflet_map <- renderLeaflet({

    quakes %>%
      filter(mag <= input$min_magnitude) %>%
      leaflet() %>%
      addTiles() %>%
      addCircleMarkers()

  })

}
```

ui.R

```r
fluidPage(
  sliderInput("min_magnitude",
              "Minimum earthquare magnitude",
              min = 4,
              max = 6.4,
              value = 4.5,
              step = 0.1
              ),
  leafletOutput("leaflet_map")
)
```

server.R

```r
function(input, output){

  output$leaflet_map <- renderLeaflet({

    quakes %>%
      filter(mag <= input$min_magnitude) %>%
      leaflet() %>%
      addTiles() %>%
      addCircleMarkers()

  })

}
```

ui.R

```r
fluidPage(
  sliderInput("min_magnitude",
              "Minimum earthquare magnitude",
              min = 4,
              max = 6.4,
              value = 4.5,
              step = 0.1
              ),
  leafletOutput("leaflet_map")
)
```

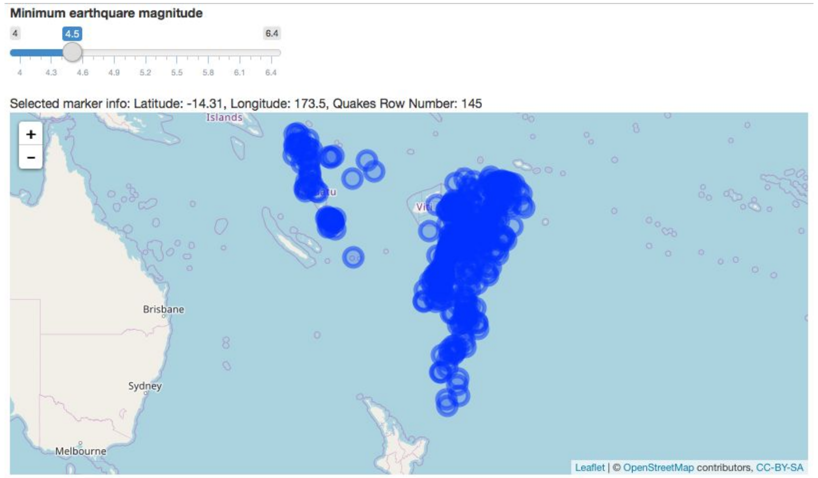
server.R

```r
function(input, output){

  output$leaflet_map <- renderLeaflet({

    quakes %>%
      filter(mag <= input$min_magnitude) %>%
      leaflet() %>%
      addTiles() %>%
      addCircleMarkers()

  })

}
```

ui.R

```r
fluidPage(
  sliderInput("min_magnitude",
              "Minimum earthquare magnitude",
              min = 4,
              max = 6.4,
              value = 4.5,
              step = 0.1
              ),
  leafletOutput("leaflet_map")
)
```

Advanced Shiny App


Minimum earthquare magnitude
4
4.5
6.4
4
4.3
4.6
4.9
5.2
5.5
5.8
6.1
6.4
Islands
+
−
Brisbane
Sydney
Melbourne
Leaflet | © OpenStreetMap contributors, CC-BY-SA

Selected marker info: Latitude: -14.31, Longitude: 173.5, Quakes Row Number: 145

server.R

```r
## Rather than re-loading the map tiles, instead clear and add shapes
## whenever input$min_magnitude changes
observeEvent(input$min_magnitude,
             {
               leafletProxy("leaflet_map") %>%
                 clearShapes() %>%
                 addCircleMarkers(
                   data = quakes_sf %>%
                     filter(mag <= input$min_magnitude),
                   layerId = ~ quake.id ## Add quake.id to the leaflet_map_marker_click object
                 )
             })
```

```r
output$magnitude_slider_UI <- renderUI({
  sliderInput(
    "min_magnitude",
    "Minimum earthquare magnitude",
    min = min(quakes$mag),
    max = max(quakes$mag),
    value = 4.5,
    step = 0.1
  )
})
```

```r
output$leaflet_map <- renderLeaflet({
  if (is.null(input$min_magnitude)) {
    return()
  }

  bbox <- quakes_sf %>%
    st_bbox() %>%
    as.list()

  leaflet() %>%
    addTiles() %>%
    fitBounds(bbox$xmin, bbox$ymin, bbox$xmax, bbox$ymax)

})
```

ui.R

```r
fluidPage(
  uiOutput("magnitude_slider_UI"),
  uiOutput("selected_point_UI"),
  leafletOutput("leaflet_map")
)
```

server.R

```r
## Rather than re-loading the map tiles, instead clear and add shapes
## whenever input$min_magnitude changes
observeEvent(input$min_magnitude,
             {
               leafletProxy("leaflet_map") %>%
                 clearShapes() %>%
                 addCircleMarkers(
                   data = quakes_sf %>%
                     filter(mag <= input$min_magnitude),
                   layerId = ~ quake.id ## Add quake.id to the leaflet_map_marker_click object
                 )
             })
```

```r
output$magnitude_slider_UI <- renderUI({
  sliderInput(
    "min_magnitude",
    "Minimum earthquare magnitude",
    min = min(quakes$mag),
    max = max(quakes$mag),
    value = 4.5,
    step = 0.1
  )
})
```

```r
output$leaflet_map <- renderLeaflet({
  if (is.null(input$min_magnitude)) {
    return()
  }

  bbox <- quakes_sf %>%
    st_bbox() %>%
    as.list()

  leaflet() %>%
    addTiles() %>%
    fitBounds(bbox$xmin, bbox$ymin, bbox$xmax, bbox$ymax)

})
```

ui.R

```r
fluidPage(
  uiOutput("magnitude_slider_UI"),
  uiOutput("selected_point_UI"),
  leafletOutput("leaflet_map")
)
```

server.R

```r
## Rather than re-loading the map tiles, instead clear and add shapes
## whenever input$min_magnitude changes
observeEvent(input$min_magnitude,
             {
               leafletProxy("leaflet_map") %>%
                 clearShapes() %>%
                 addCircleMarkers(
                   data = quakes_sf %>%
                     filter(mag <= input$min_magnitude),
                   layerId = ~ quake.id ## Add quake.id to the leaflet_map_marker_click object
                 )
             })
```

```r
output$magnitude_slider_UI <- renderUI({
  sliderInput(
    "min_magnitude",
    "Minimum earthquare magnitude",
    min = min(quakes$mag),
    max = max(quakes$mag),
    value = 4.5,
    step = 0.1
  )
})
```

```r
output$leaflet_map <- renderLeaflet({
  if (is.null(input$min_magnitude)) {
    return()
  }

  bbox <- quakes_sf %>%
    st_bbox() %>%
    as.list()

  leaflet() %>%
    addTiles() %>%
    fitBounds(bbox$xmin, bbox$ymin, bbox$xmax, bbox$ymax)

})
```

ui.R

```r
fluidPage(
  uiOutput("magnitude_slider_UI"),
  uiOutput("selected_point_UI"),
  leafletOutput("leaflet_map")
)
```

server.R

```r
## Rather than re-loading the map tiles, instead clear and add shapes
## whenever input$min_magnitude changes
observeEvent(input$min_magnitude,
             {
  leafletProxy("leaflet_map") %>%
    clearShapes() %>%
    addCircleMarkers(
      data = quakes_sf %>%
        filter(mag <= input$min_magnitude),
      layerId = ~ quake.id ## Add quake.id to the leaflet_map_marker_click object
    )
             })
```

```r
output$magnitude_slider_UI <- renderUI({
  sliderInput(
    "min_magnitude",
    "Minimum earthquare magnitude",
    min = min(quakes$mag),
    max = max(quakes$mag),
    value = 4.5,
    step = 0.1
  )
})
```

```r
output$leaflet_map <- renderLeaflet({
  if (is.null(input$min_magnitude)) {
    return()
  }

  bbox <- quakes_sf %>%
    st_bbox() %>%
    as.list()

  leaflet() %>%
    addTiles() %>%
    fitBounds(bbox$xmin, bbox$ymin, bbox$xmax, bbox$ymax)

})
```

ui.R

```r
fluidPage(
  uiOutput("magnitude_slider_UI"),
  uiOutput("selected_point_UI"),
  leafletOutput("leaflet_map")
)
```

server.R

```r
## Rather than re-loading the map tiles, instead clear and add shapes
## whenever input$min_magnitude changes
observeEvent(input$min_magnitude,
             {
  leafletProxy("leaflet_map") %>%
    clearShapes() %>%
    addCircleMarkers(
      data = quakes_sf %>%
        filter(mag <= input$min_magnitude),
      layerId = ~ quake.id ## Add quake.id to the leaflet_map_marker_click object
    )
})
```

```r
output$magnitude_slider_UI <- renderUI({
  sliderInput(
    "min_magnitude",
    "Minimum earthquare magnitude",
    min = min(quakes$mag),
    max = max(quakes$mag),
    value = 4.5,
    step = 0.1
  )
})
```

```r
output$leaflet_map <- r          t({
  if (is.null(input     n_magnitude))
    return()
  }

  bbox <- qu   es_sf %>%
    st_bbox(   %>%
    as.list(

  leaflet() %
    addTiles()   >%
    fitBounds(bb   $xmin, bbox$ymin, bb   xmax, bbox$ymax)

})
```

ui.R

```r
fluidPage(
  uiOutput("magnitude_slider_UI"),
  uiOutput("selected_point_UI"),
  leafletOutput("leaflet_map")
)
```

RStudio Connect is a content management system for htmlwidgets, RMarkdown documents and Shiny apps.

RStudio Connect allows parameterised RMarkdown documents to be scheduled.

RStudio

Shiny

tidyverse

%>%

Ceci n'est pas un pipe.

htmlwidgets

%>%

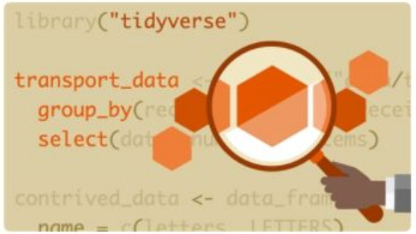
Ceci n'est pas un pipe.

RStudio Connect

rmarkdown

lynda.com

in LEARNING



Learning the R Tidyverse

Learn to integrate the tidyverse into your R workflow and get new tools for importing, filtering, visualizing, and modeling research and statistical data.

3h 44m Intermediate Views: 2,826 1 week ago

R: Interactive Visualizations with htmlwidgets

Learn how to rapidly create rich, interactive data visualizations with R and htmlwidgets—packages that connect R to popular JavaScript libraries like Plotly, Leaflet, and DT.

5h 25m Intermediate Views: 3,490 Oct 4, 2017

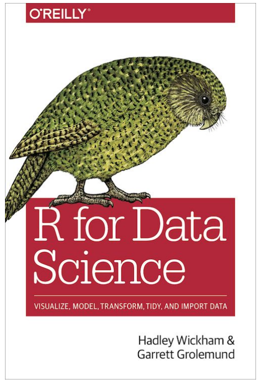
Creating Interactive Presentations with Shiny and R

Make the results of big data analysis more compelling and clear. Learn how to create interactive presentations and dashboards with RStudio and Shiny.

1h 53m Intermediate Views: 58,734 Apr 27, 2016

DataCamp

http://r4ds.had.co.nz/

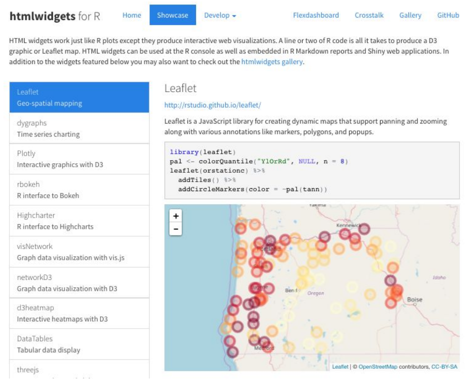

O'REILLY

R for Data Science

VISUALIZE, MODEL, TRANSFORM, TIDY, AND IMPORT DATA

Hadley Wickham &
Garrett Grolemund

htmlwidgets.org



HTML widgets work just like R plots except they produce interactive web visualizations. A line or two of R code is all it takes to produce a D3 graphic or Leaflet map. HTML widgets can be used at the R console as well as embedded in R Markdown reports and Shiny web applications. In addition to the widgets featured below you may also want to check out the htmlwidgets gallery.

Leaflet	Geo-spatial mapping
dygraphs	Time series charting
Plotly	Interactive graphics with D3
rbokeh	R interface to Bokeh
Highcharter	R interface to Highcharts
visNetwork	Graph data visualization with vis.js
networkD3	Graph data visualization with D3
d3heatmap	Interactive heatmaps with D3
DataTables	Tabular data display
threejs	

Leaflet

http://rstudio.github.io/leaflet/

Leaflet is a JavaScript library for creating dynamic maps that support panning and zooming along with various annotations like markers, polygons, and popups.

```
library(leaflet)
pal <- colorQuantile("YlOrRd", NULL, n = 8)
leaflet(orstationc) %>%
  addTiles() %>%
  addCircleMarkers(color = ~pal(tann))
```