DSBA 5122: Visual Analytics

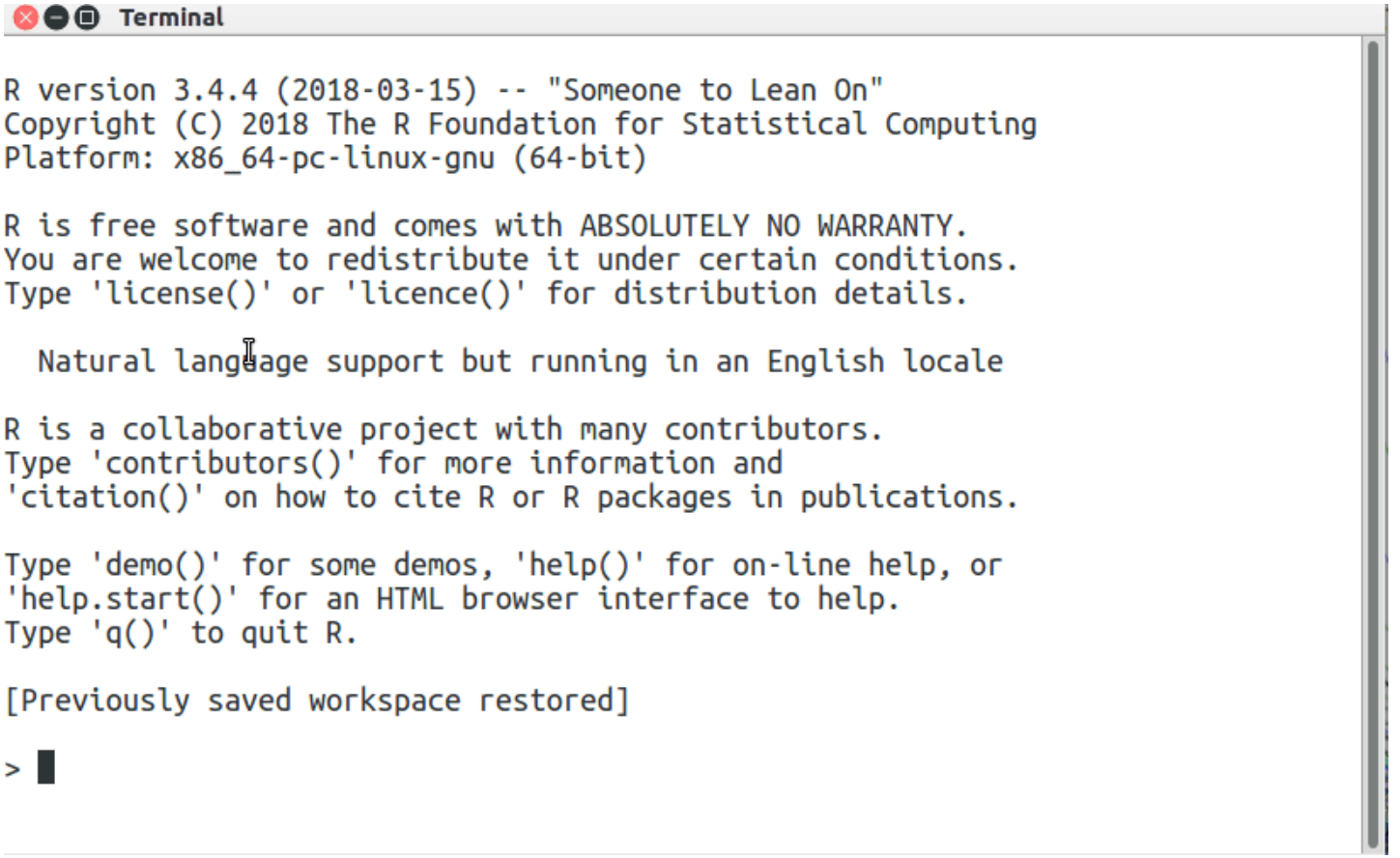
Why R (and RStudio)?

Ryan Wesslen

September 9, 2019

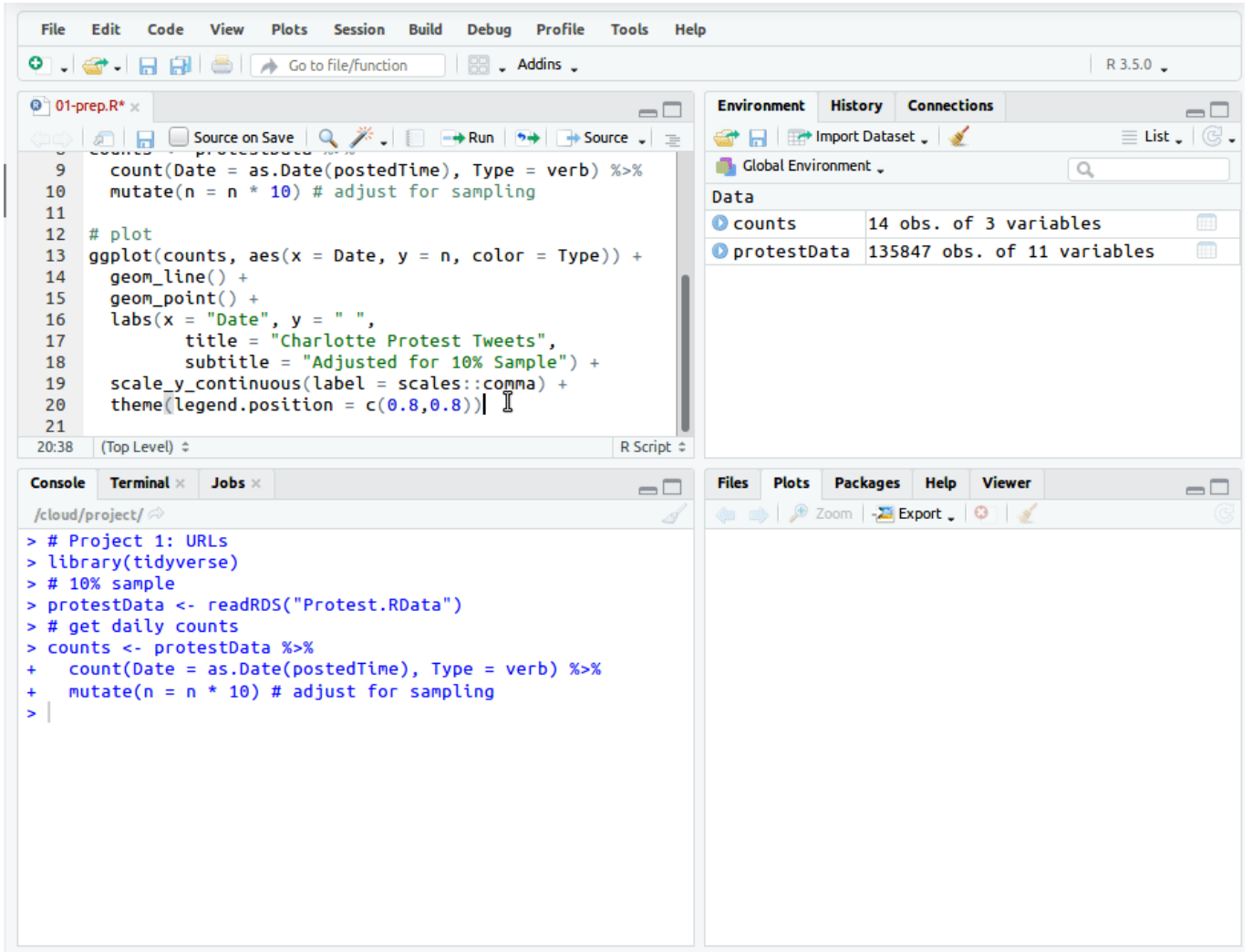
bit.ly/dsbaRintro

What is R?



bit.ly/dsbaRintro

What is RStudio?



bit.ly/dsbaRintro

1. Why R / RStudio?

It's free, as in ... free beer?

2. Hanging with the big kids.

Miles McBain
@MilesMcBain

Well then. Another take on programming language popularity just published by IEEE Spectrum has #rstats at 5, edging out JavaScript. spectrum.ieee.org/computing/soft… an incredible result for a DSL.

The Top Programming Languages 2019

Python remains the big kahuna, but specialist languages hold their…

spectrum.ieee.org

57 11:17 PM - Sep 7, 2019

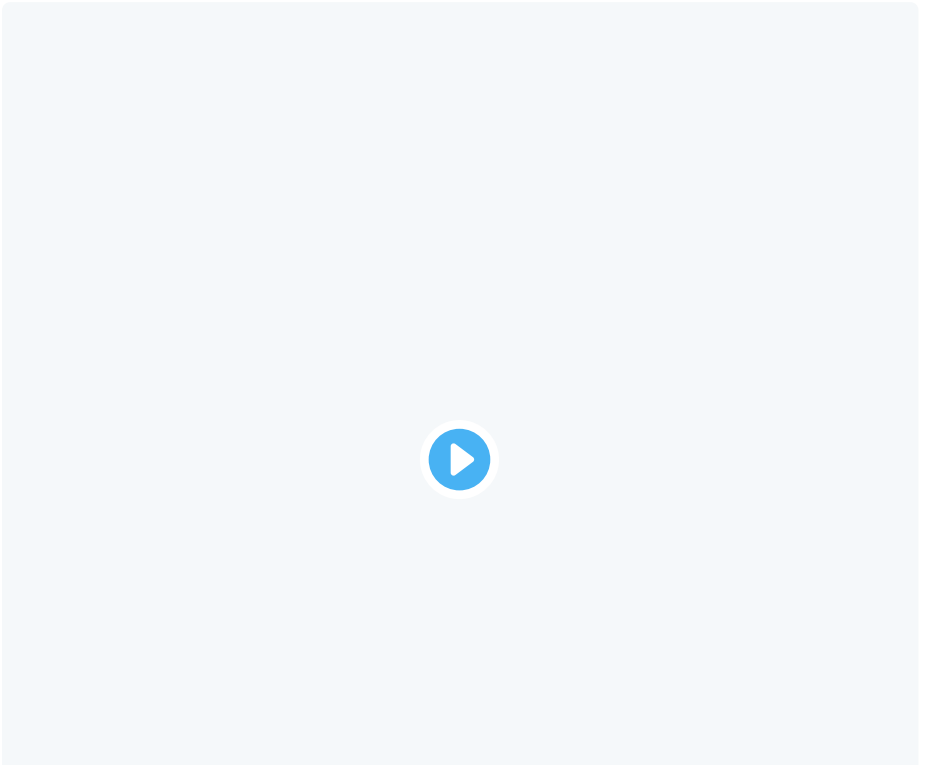
20 people are talking about this

bit.ly/dsbaRintro

3. R open source community makes it fun!

Tyler Morgan-Wall
@tylermorganwall

Did you know #rayshader comes with a function that can detect bodies of water directly from the elevation data? `detect_water()` looks for large contiguous flat regions of a user-defined minimum size and "flatness." Use `add_water()` to layer it on the map. #rstats



bit.ly/dsbaRintro

3. R open source community makes it fun!

Tyler Morgan-Wall
@tylermorganwall

Replying to @tylermorganwall

And here's a gist showing how to make the above "jumping River Derwent" visualization:gist.github.com/tylermorganwal…

Creating the River Derwent, jumping out of the map

Creating the River Derwent, jumping out of the map - jumping_wat…

gist.github.com

21 9:32 AM - Aug 27, 2019

See Tyler Morgan-Wall's other Tweets

bit.ly/dsbaRintro

4. Why R / RStudio?

JD Long
@CMastication

well this R=Batman, Python=Superman apparently showed up in a vendor pitch to one of my colleagues. That must make it official. #rstats #python

Analysis Tool	Similar Superhero	Super Powers in Common
R	Batman	• Detective Work • Intelligence • Cunning • Usage of Tools • More Brain than Muscles
Python	Superman	• Muscle Power • Super Strength • Elegance • Wide Range • More Muscles than Brain

♡ 416 7:44 PM - Sep 5, 2018

172 people are talking about this

bit.ly/dsbaRintro

bit.ly/dsbaRintro

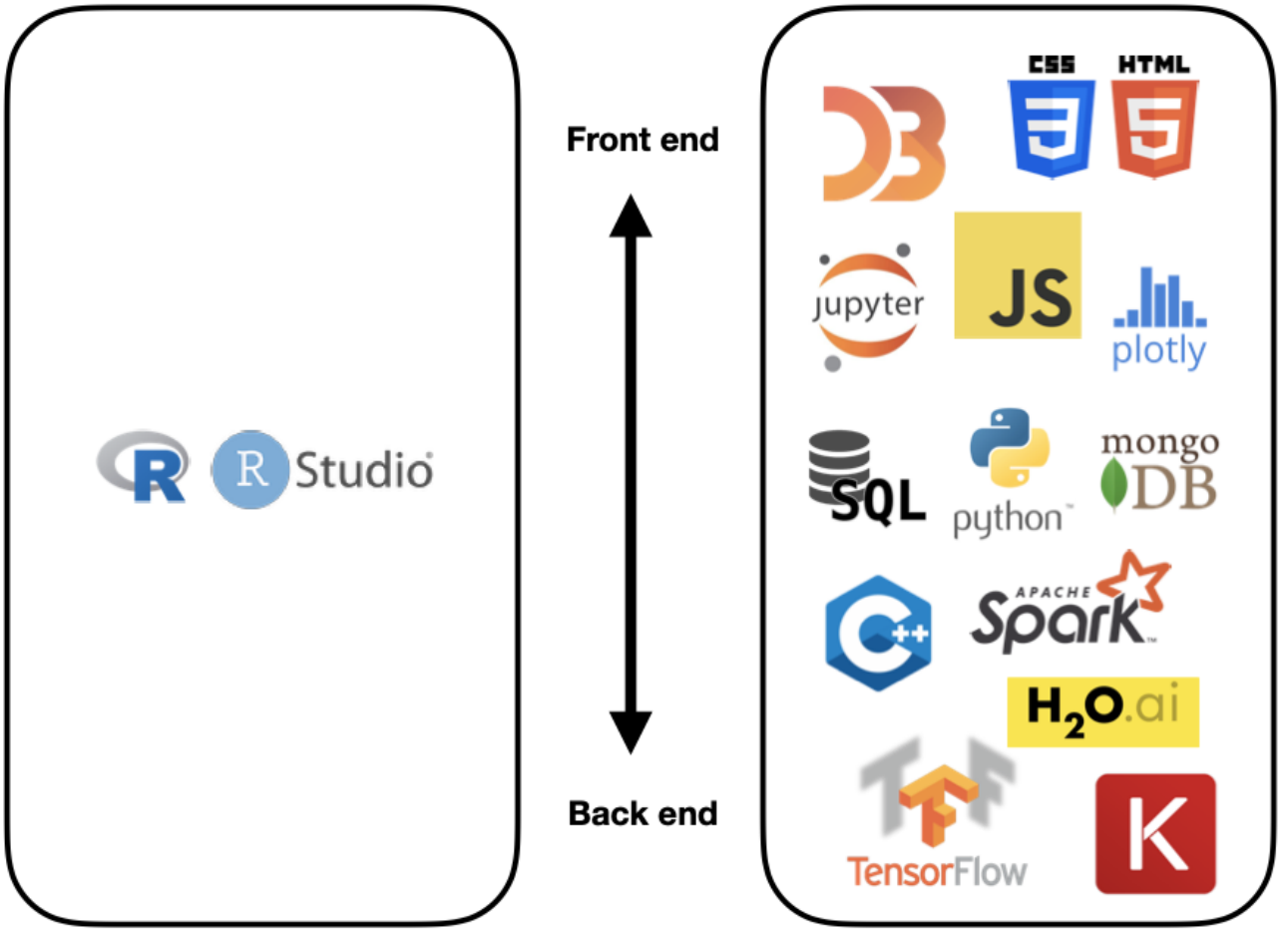
Front end

bit.ly/dsbaRintro

Front end

Back end

bit.ly/dsbaRintro



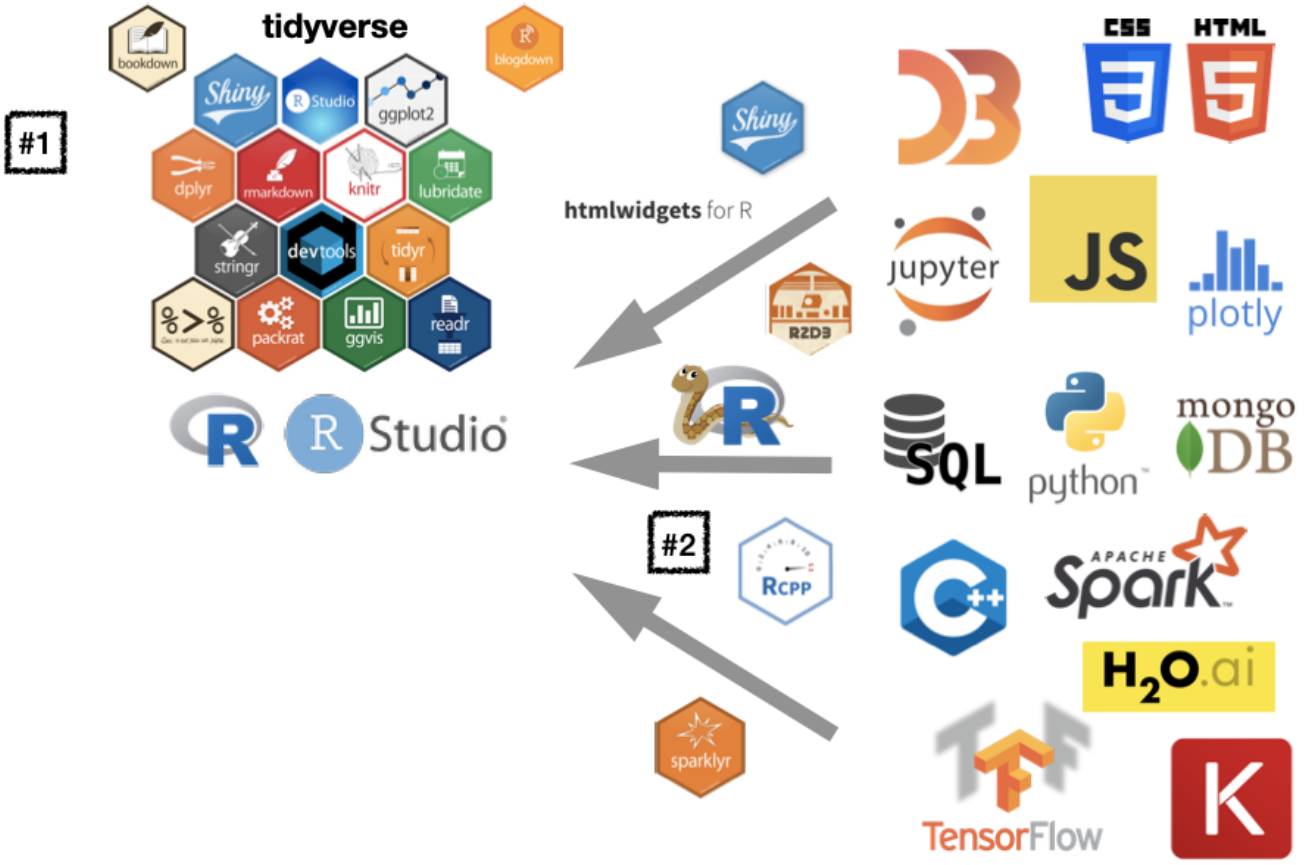
bit.ly/dsbaRintro

#1

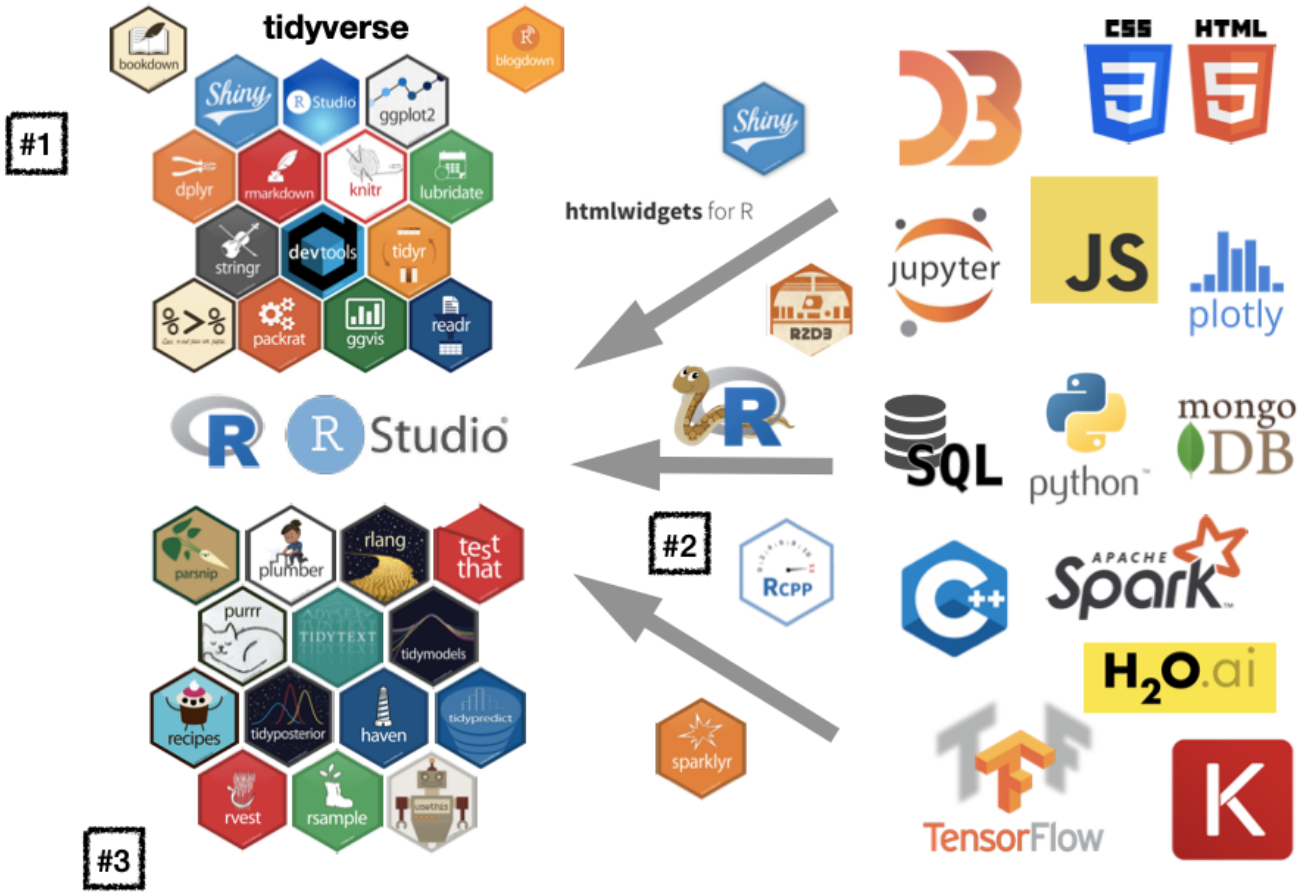
Front end

Back end

bit.ly/dsbaRintro



bit.ly/dsbaRintro



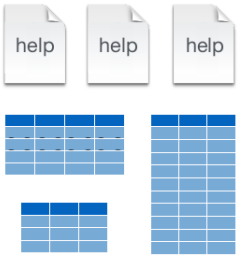
bit.ly/dsbaRintro

R / RStudio to the rescue

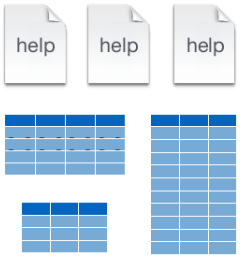
Artwork by @allison_horst

bit.ly/dsbaRintro

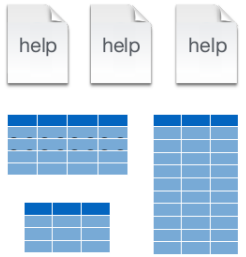
What are R packages?



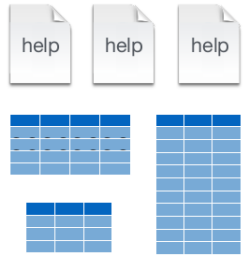

```
function1()
function2()
function3()
function4()
```




```
function5()
function6()
function7()
function8()
```

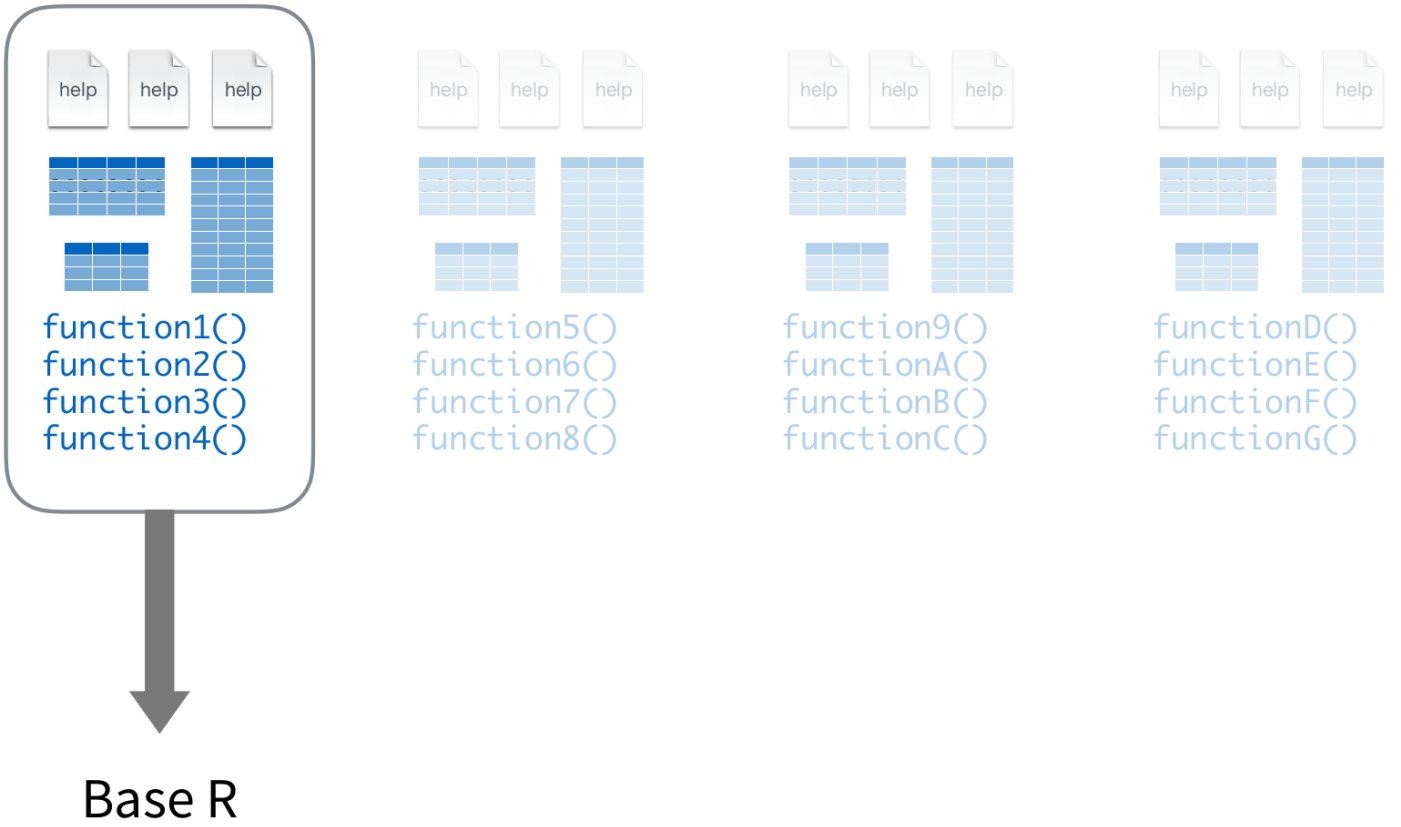



```
function9()
functionA()
functionB()
functionC()
```

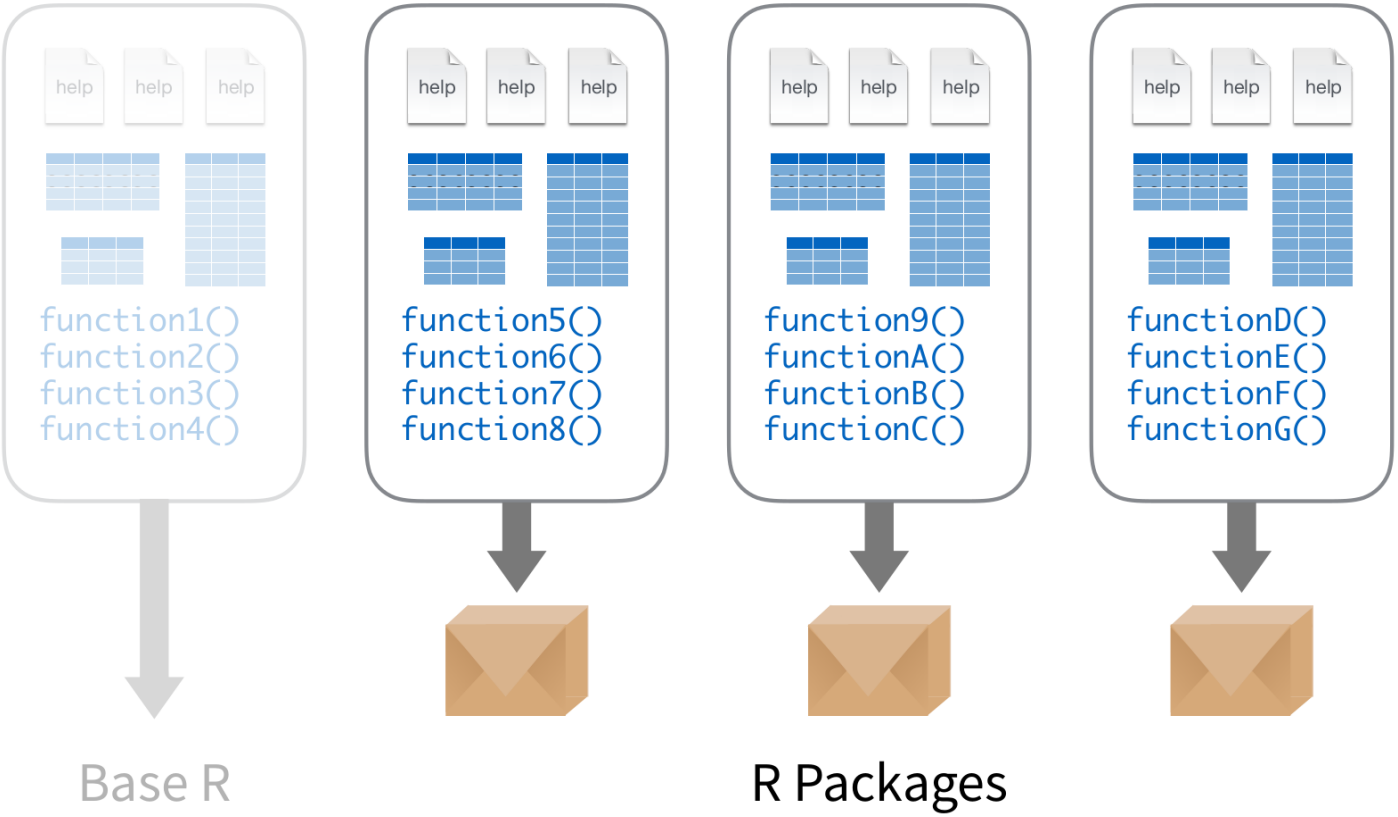



```
functionD()
functionE()
functionF()
functionG()
```

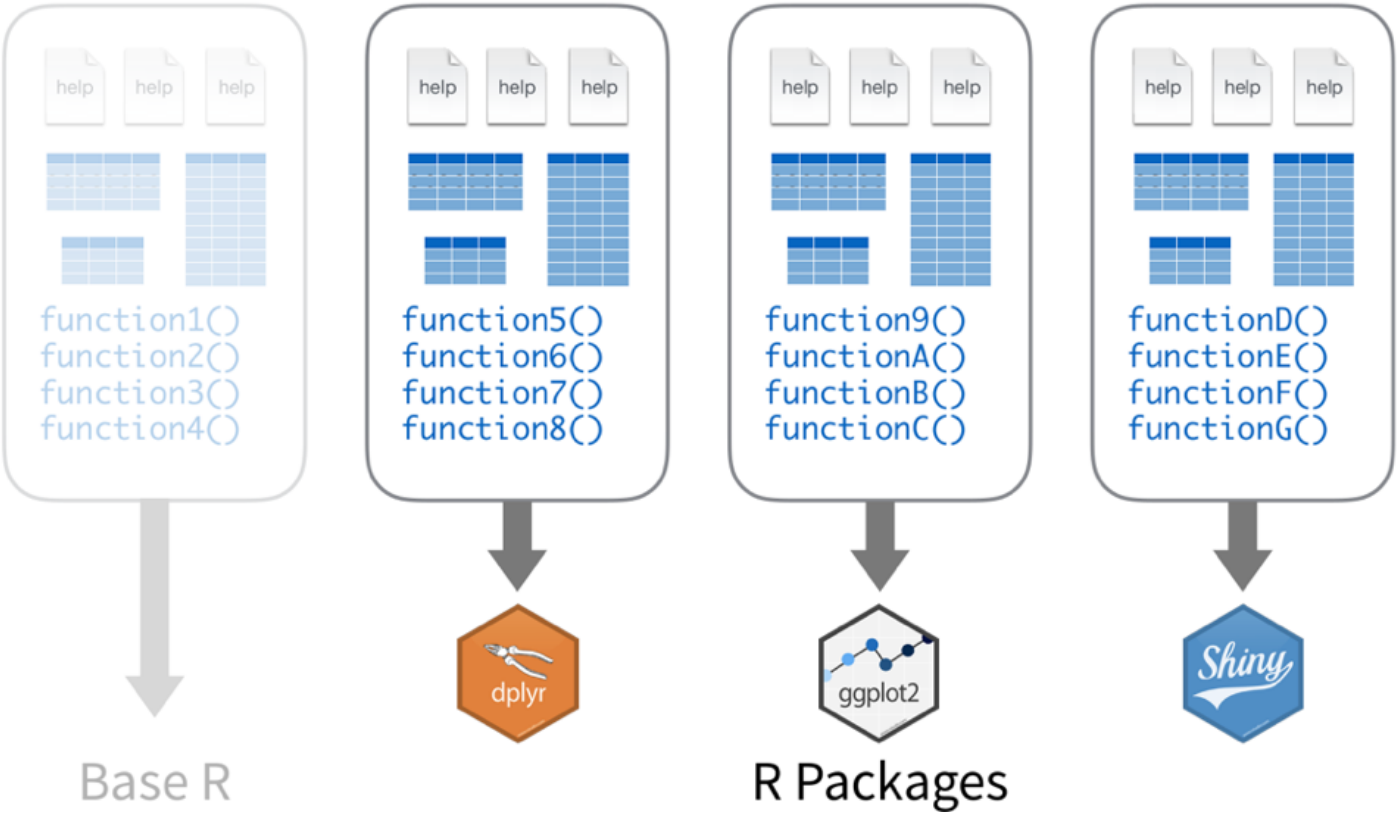
What are R packages?



What are R packages?



What are R packages?



How to install and run packages:

1

```
install.packages("foo")
```

Downloads files to computer

1 x per computer

How to install and run packages:

1

```
install.packages("foo")
```

Downloads files to computer

1 x per computer

2

```
library("foo")
```

Loads package

1 x per R Session

How to install and run packages:

1

```
install.packages("foo")
```

Downloads files to computer

1 x per computer

2

```
library("foo")
```

Loads package

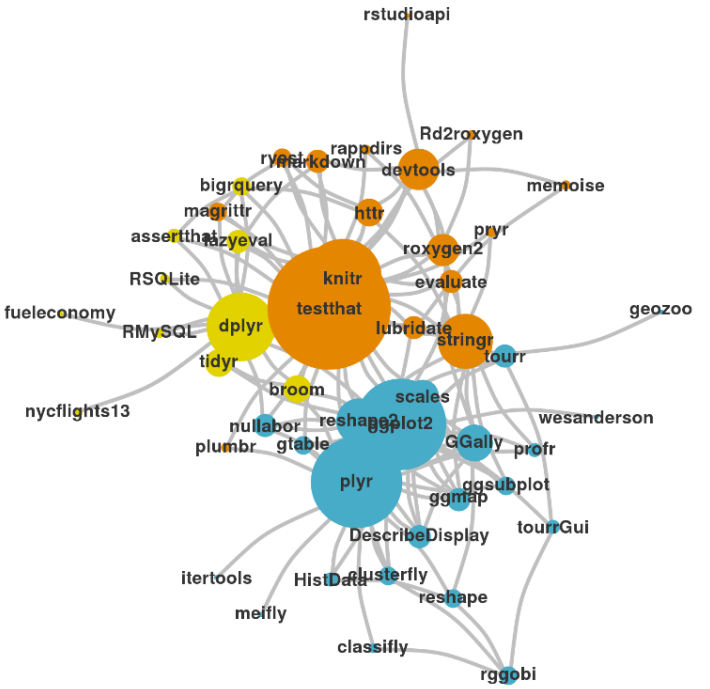
1 x per R Session

How to install and run packages:

```
install.packages("tidyverse")
```

does the equivalent of

```
install.packages("ggplot2")
install.packages("dplyr")
install.packages("tidyr")
install.packages("readr")
install.packages("purrr")
install.packages("tibble")
install.packages("hms")
install.packages("stringr")
install.packages("lubridate")
install.packages("forcats")
install.packages("DBI")
install.packages("haven")
install.packages("httr")
install.packages("jsonlite")
install.packages("readxl")
install.packages("rvest")
install.packages("xml2")
install.packages("modelr")
install.packages("broom")
```



bit.ly/dsbaRintro

How to install and run packages:

```
install.packages("tidyverse")
```

does the equivalent of

```
install.packages("ggplot2")
install.packages("dplyr")
install.packages("tidyr")
install.packages("readr")
install.packages("purrr")
install.packages("tibble")
install.packages("hms")
install.packages("stringr")
install.packages("lubridate")
install.packages("forcats")
install.packages("DBI")
install.packages("haven")
install.packages("httr")
install.packages("jsonlite")
install.packages("readxl")
install.packages("rvest")
install.packages("xml2")
install.packages("modelr")
install.packages("broom")
```

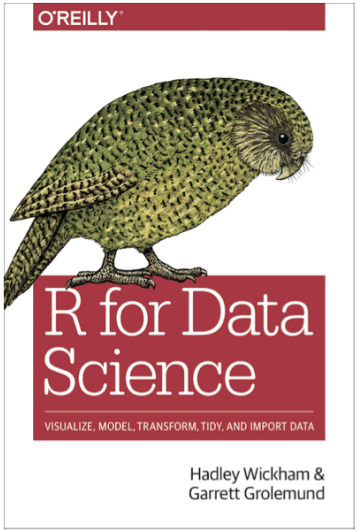
```
library("tidyverse")
```

does the equivalent of

```
library("ggplot2")
library("dplyr")
library("tidyr")
library("readr")
library("purrr")
library("tibble")
```

RStudio Visualization Ecosystem

`tidyverse`



tidyverse.org

RStudio Visualization Ecosystem

`tidyverse`

`rmarkdown`

Analyze. Share. Reproduce.

rmarkdown.rstudio.com

bit.ly/dsbaRintro

RStudio Visualization Ecosystem

`tidyverse`

`rmarkdown`

`htmlwidgets`

JavaScript libraries as R packages!
htmlwidgets.org

RStudio Visualization Ecosystem

`tidyverse`

`rmarkdown`

`htmlwidgets`

`shiny`

shiny.rstudio.com/

bit.ly/dsbaRintro

RStudio Visualization Ecosystem

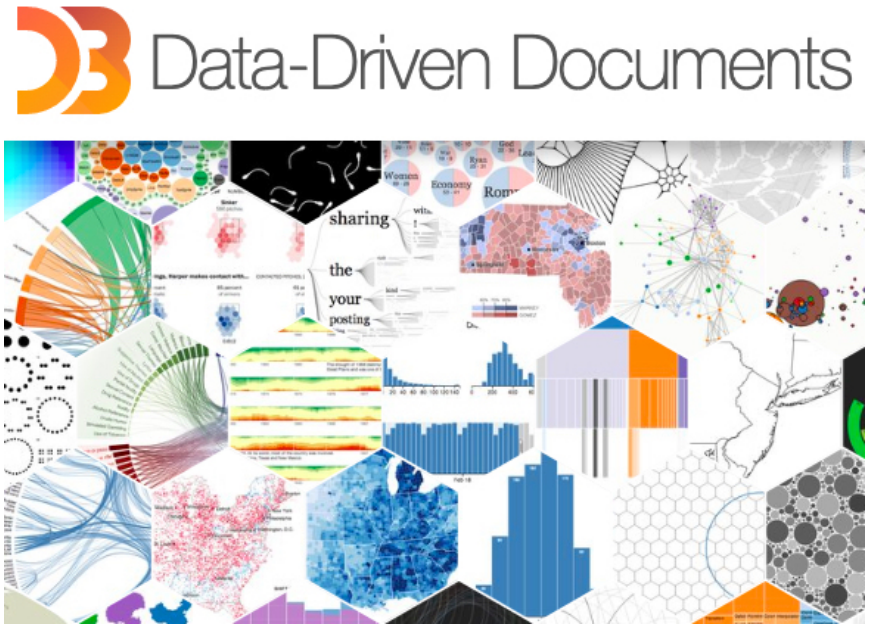
`tidyverse`

`rmarkdown`

`htmlwidgets`

`shiny`

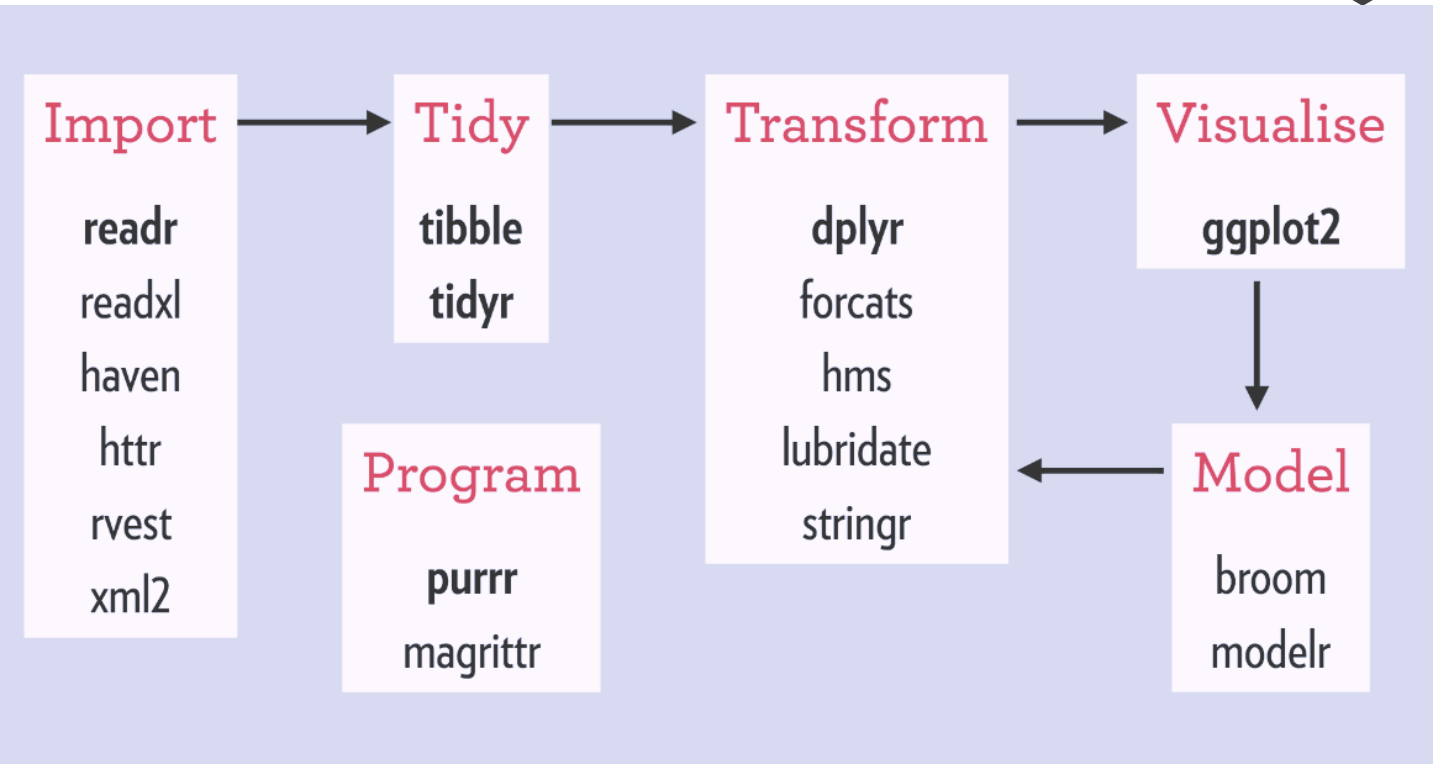
`r2d3`



d3js.org and r2d3 package

bit.ly/dsbaRintro

tidyverse: Data science process



tidyverse: dplyr + piping

US Unemployment rate

```
library(tidyverse)

# read in using readr's read_csv function
df <- read_csv("dsba-rintro_files/us-unemployment.csv",
    col_types = cols(
      date = col_date(),             # read in as date
      unemployment = col_double()    # read in as double
    )
  )

# view the data via dplyr's glimpse
glimpse(df)
```

```
## Observations: 853
## Variables: 2
## $ date         <date> 1948-01-01, 1948-02-01, 1948-03-01, 1948-04-01, 19.
## $ unemployment <dbl> 3.4, 3.8, 4.0, 3.9, 3.5, 3.6, 3.6, 3.9, 3.8, 3.7, 3.
```

tidyverse: dplyr + piping

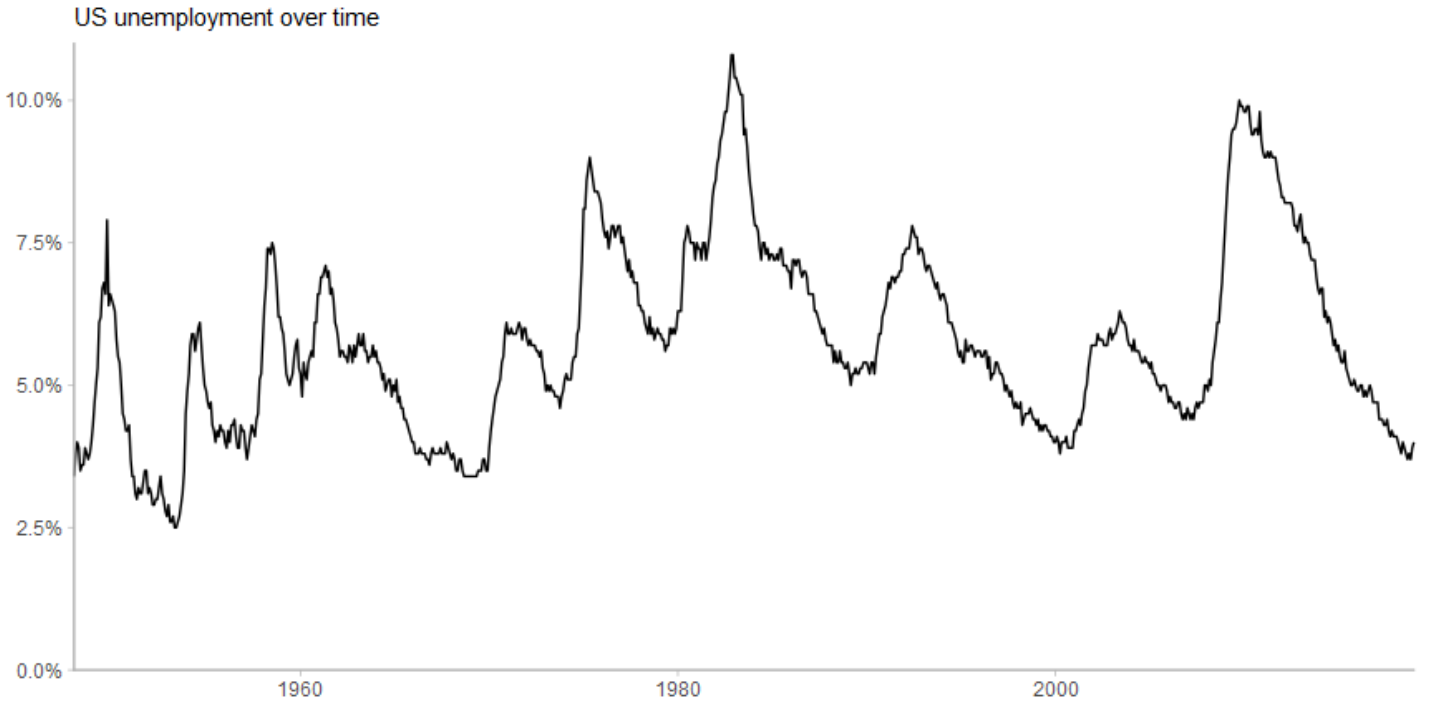
What are the years with the hightest unemployment rate in December?

```
library(lubridate) # non-standard tidyverse for dates/times

df %>%
  mutate(month = month(date)) %>% # create new column for month
  filter(month == 12) %>%         # filter for only December value
  arrange(desc(unemployment)) %>% # order by unemployment rate
  head(n = 6)                     # show top five
```

```
## # A tibble: 6 x 3
##   date       unemployment month
##   <date>            <dbl> <dbl>
## 1 1982-12-01         10.8    12
## 2 2009-12-01          9.9    12
## 3 2010-12-01          9.3    12
## 4 1981-12-01          8.5    12
## 5 2011-12-01          8.5    12
## 6 1983-12-01          8.3    12
```

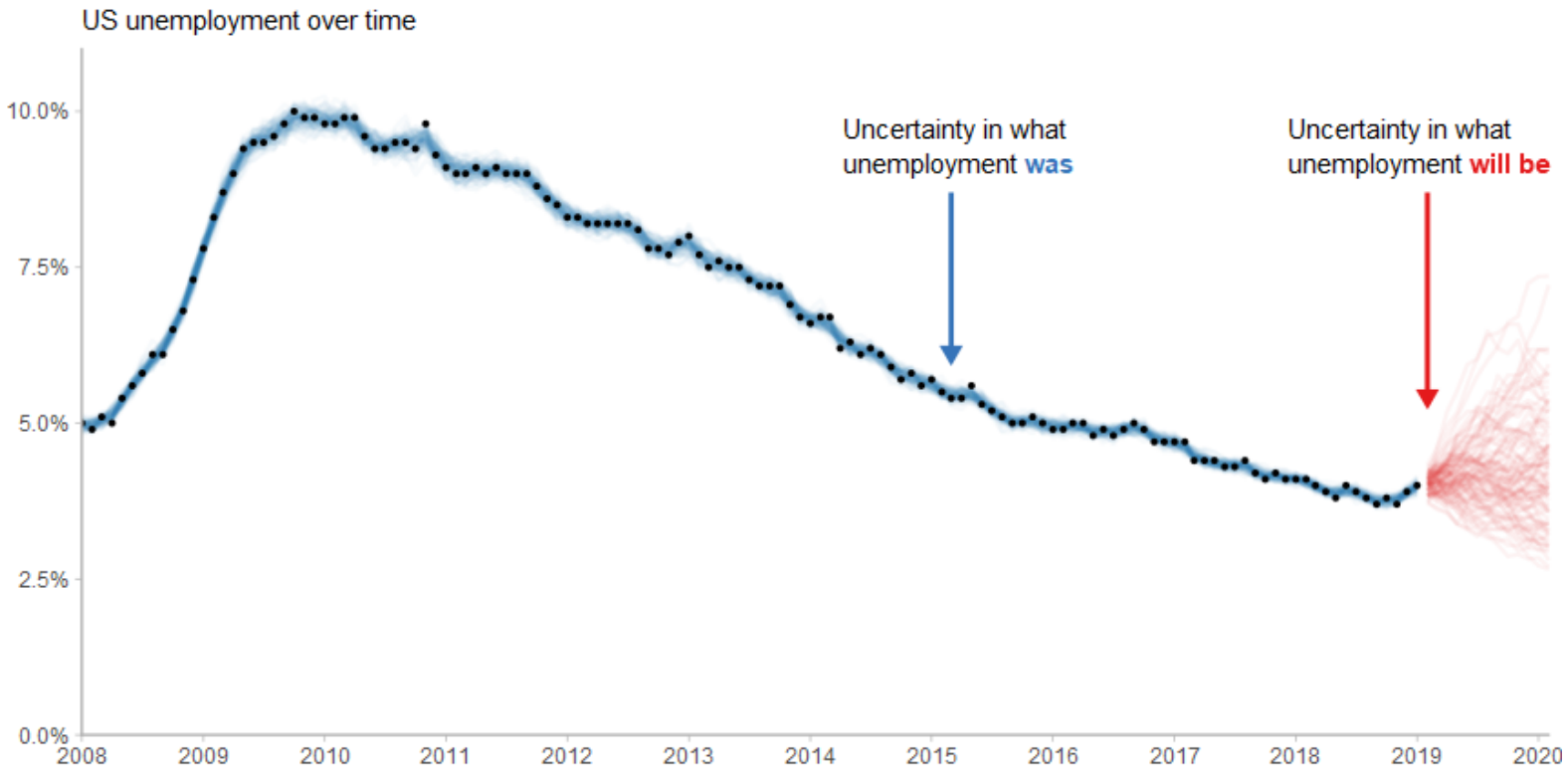
tidyverse: dplyr + piping + ggplot2


```
df %>%
  ggplot(aes(x = date, y = unemployment)) +
  geom_line() +
  coord_cartesian(ylim = c(0, .11), expand = FALSE),
  scale_y_continuous(labels = scales::percent) +
  labs(x = NULL, y = NULL, subtitle = "US unemployment over time")
```



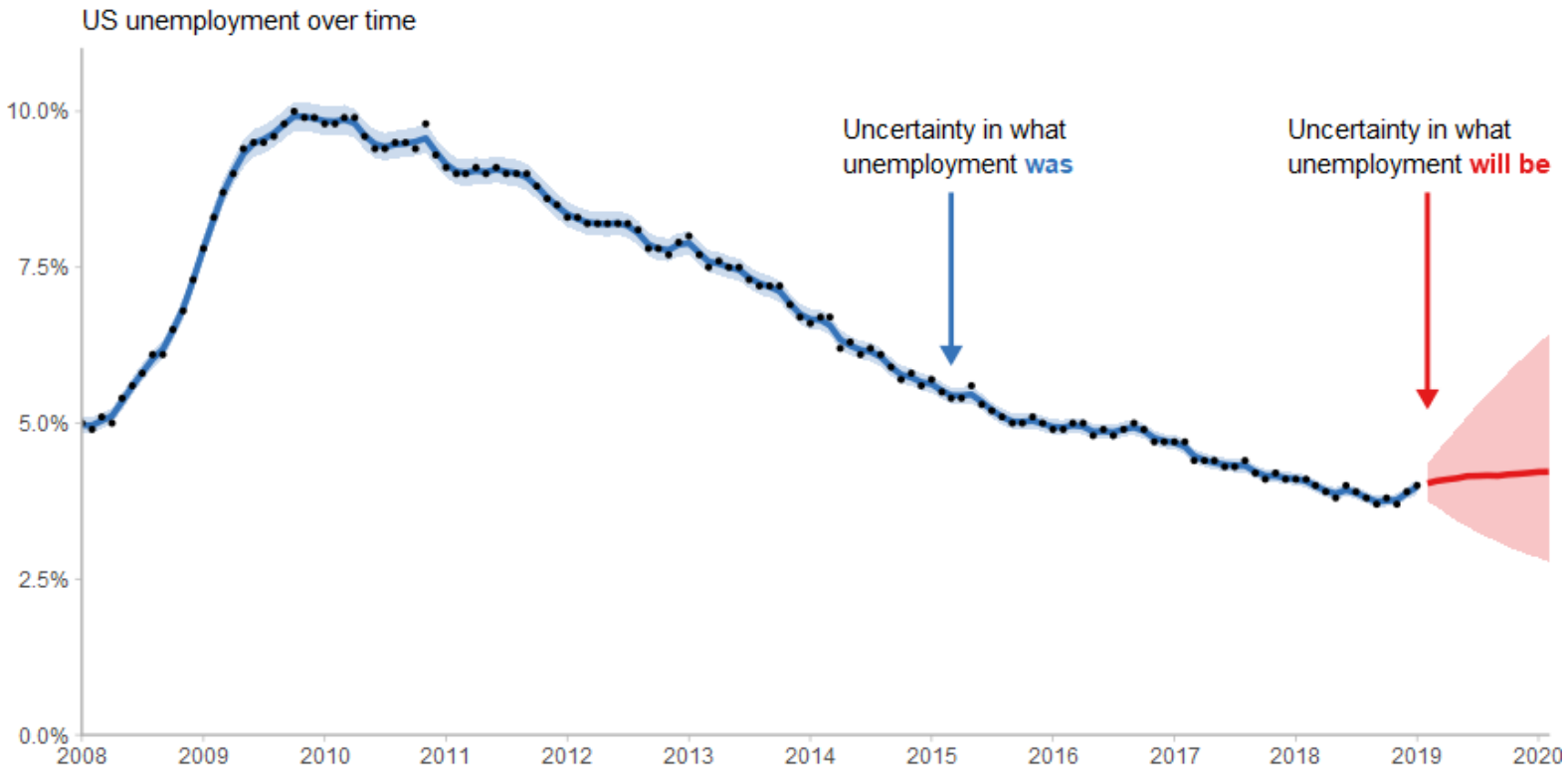
bit.ly/dsbaRintro

ggplot2 extensions



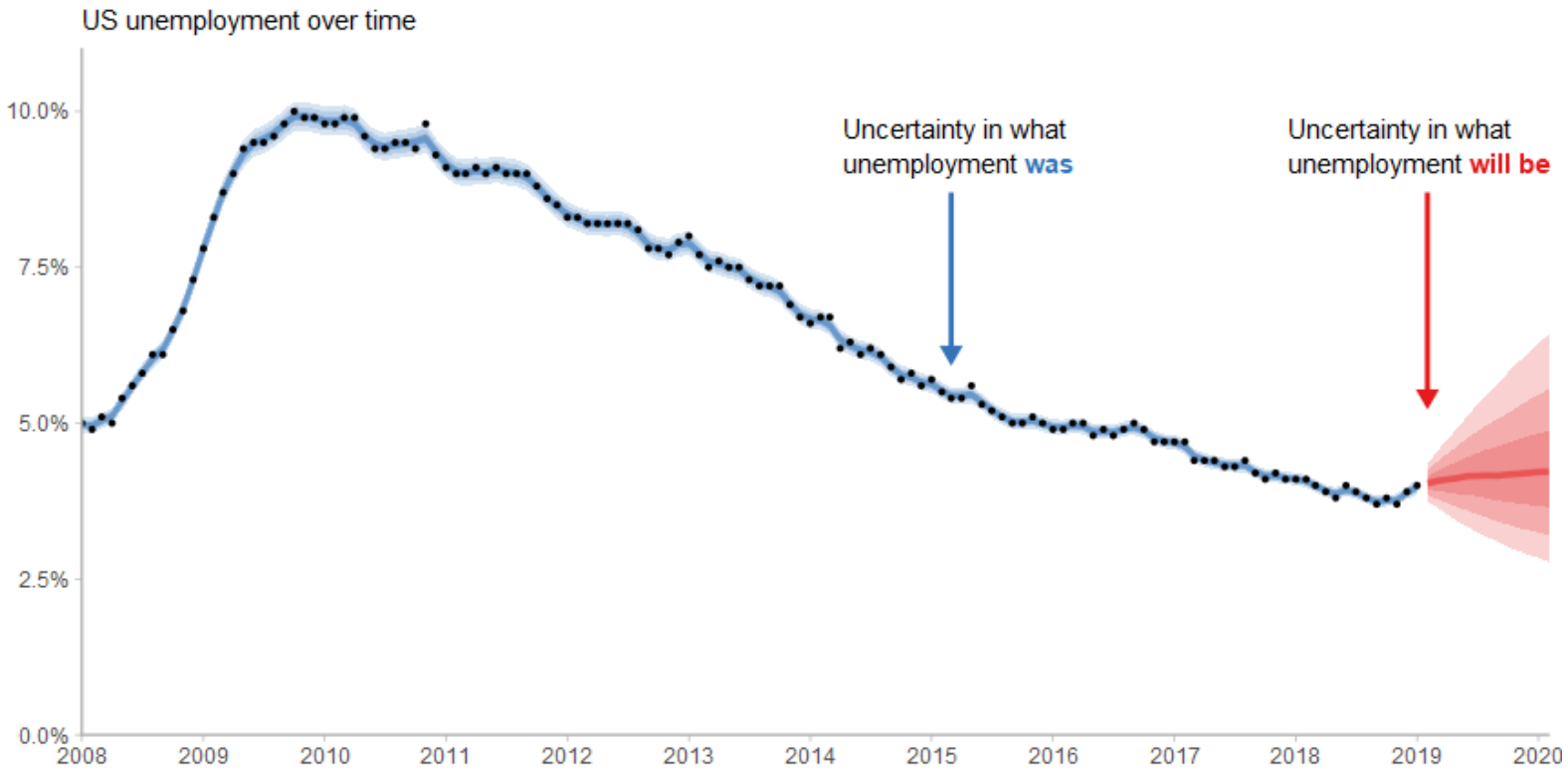
Source: Matthew Kay

ggplot2 extensions



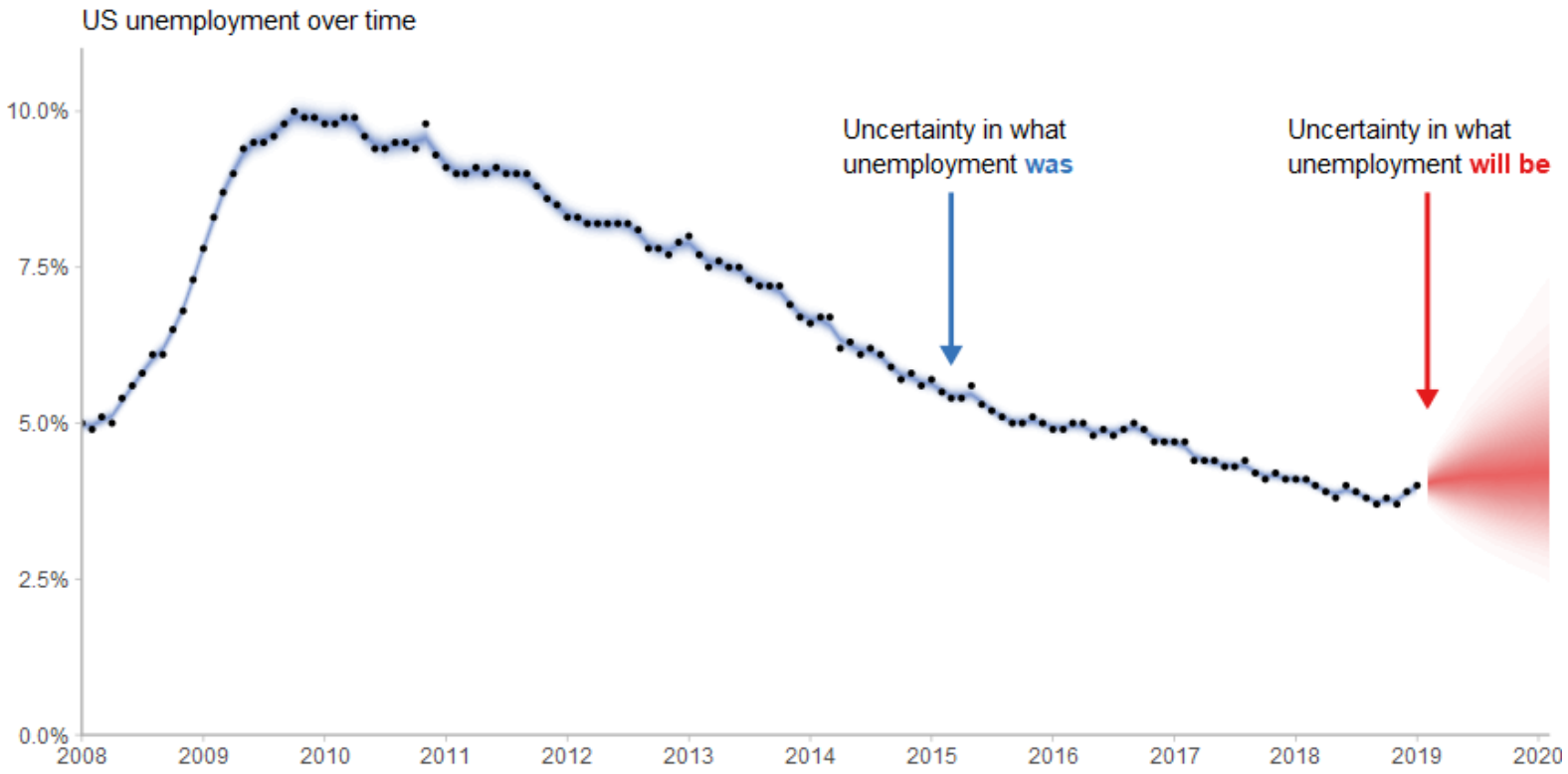
Source: Matthew Kay

ggplot2 extensions



Source: Matthew Kay

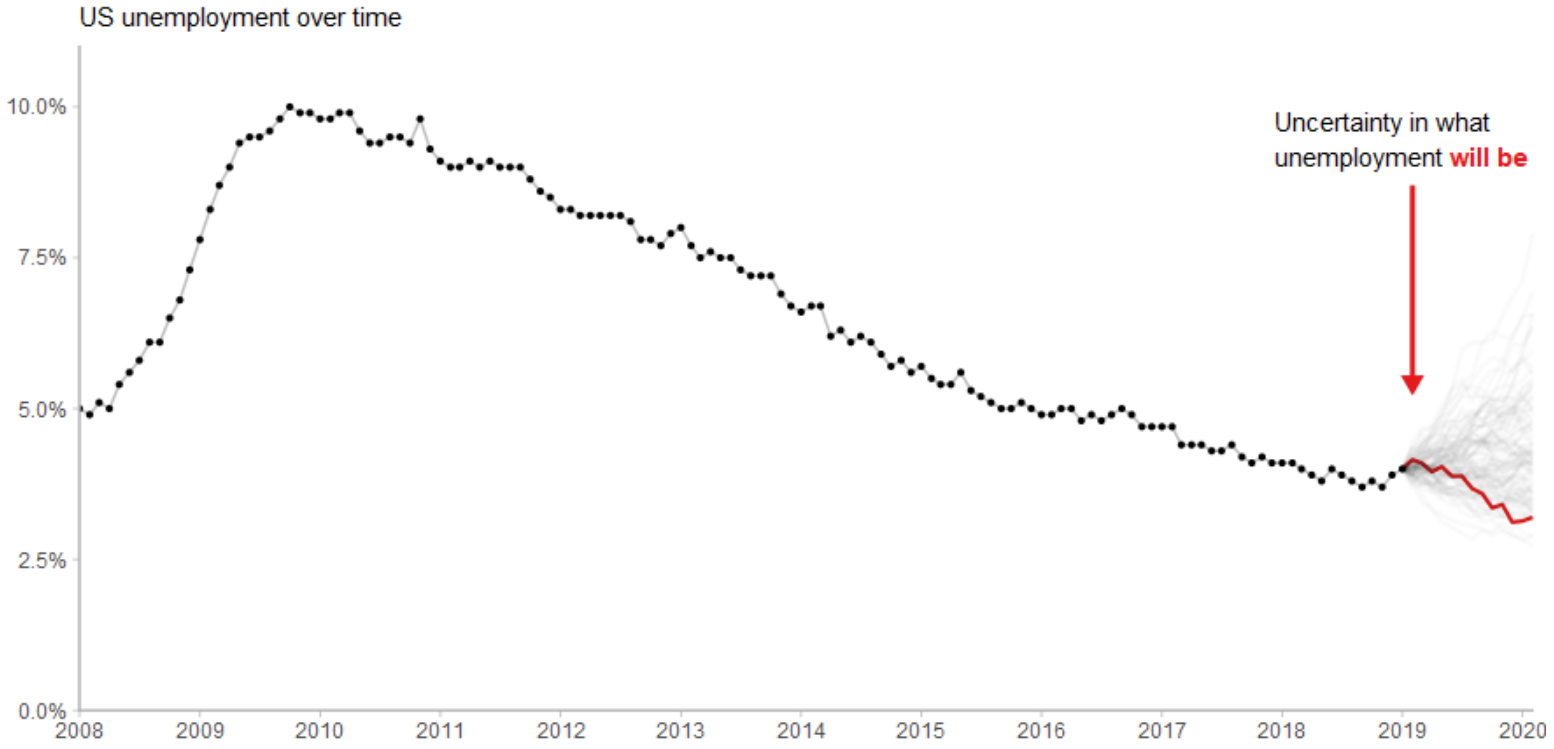
ggplot2 extensions



Source: Matthew Kay

bit.ly/dsbaRintro

ggplot2 extensions: gganimate

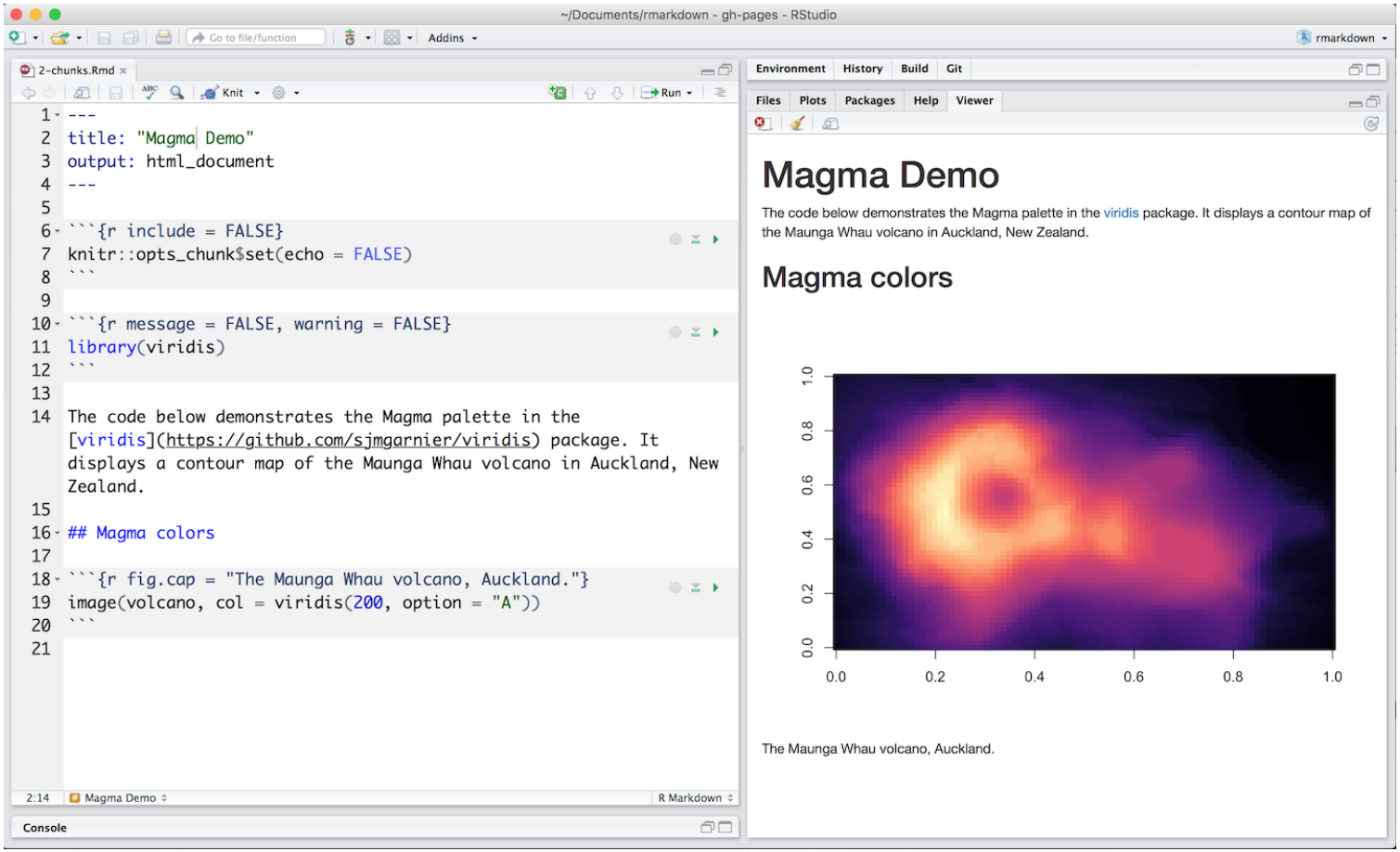


Hypothetical Outcome Plots (HOP): `tidybayes` & `gganimate`

bit.ly/dsbaRintro

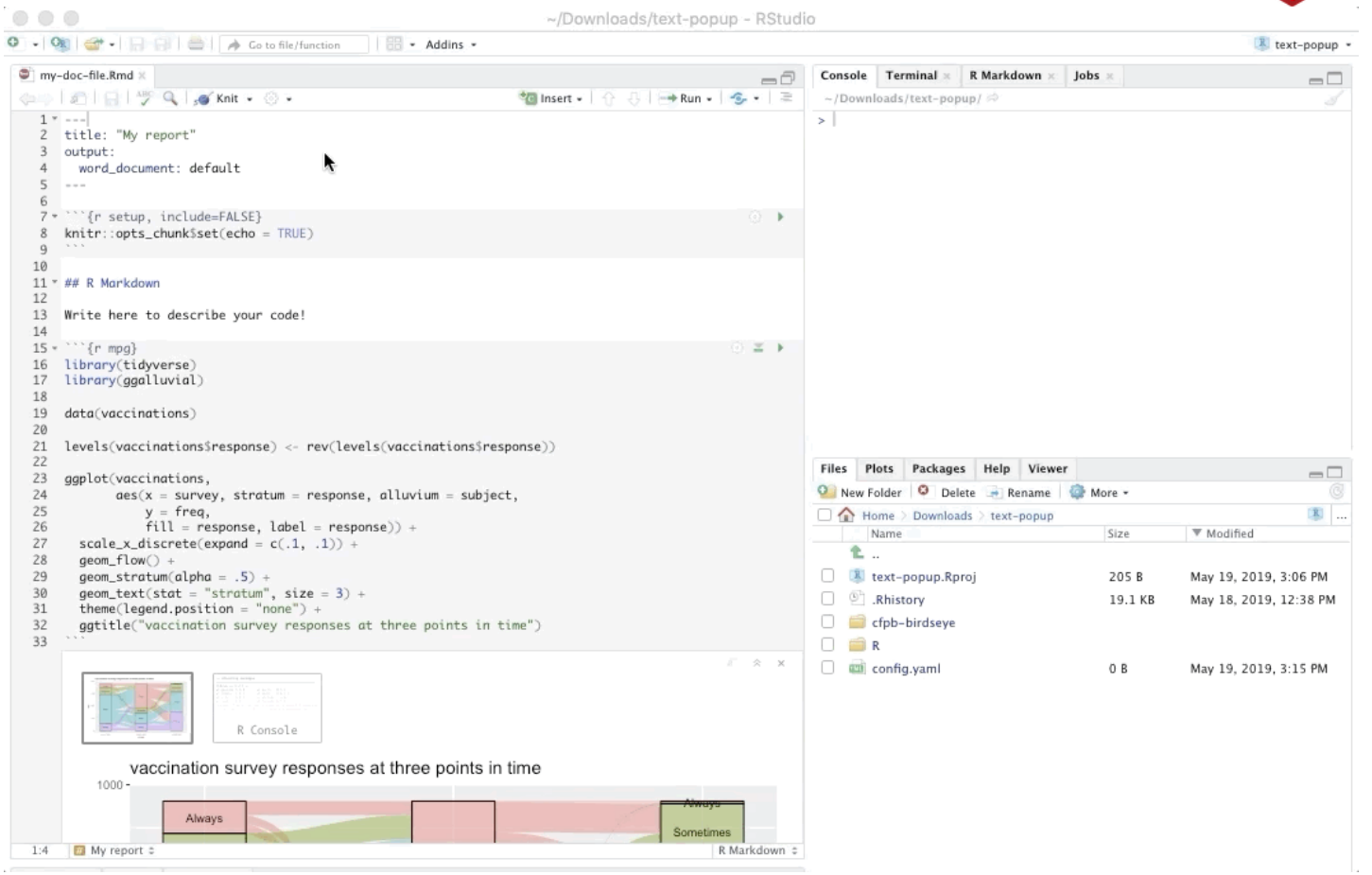
Rmarkdown: reproducibility

RStudio RMarkdown documentation



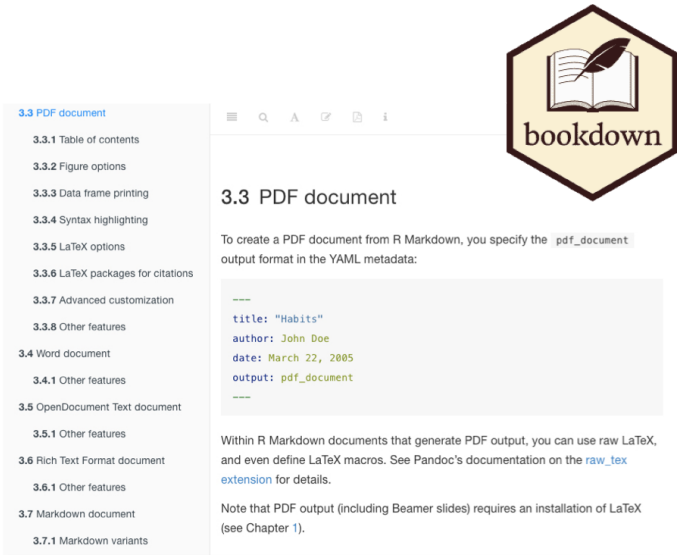
bit.ly/dsbaRintro

rmarkdown: word doc's



bit.ly/dsbaRintro

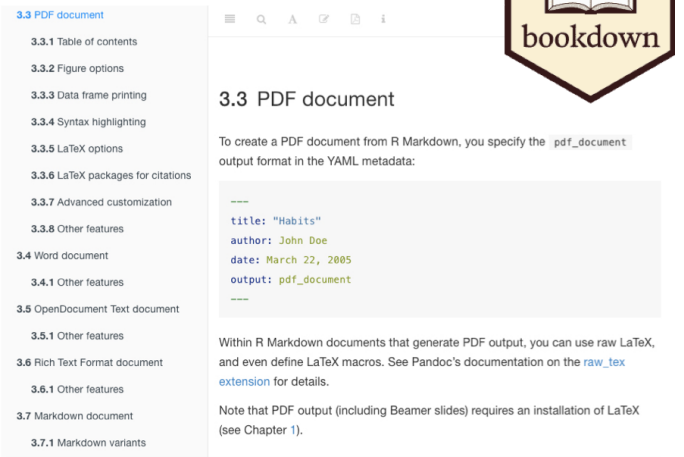
Even more of rmarkdown



- "Data Visualization: A Practical Introduction by Kieran Healy
- "Fundamentals of Data Visualization" by Claus Wilke
- "R for Data Science" by Garrett Grolemund and Hadley Wickham

bit.ly/dsbaRintro

Even more of rmarkdown



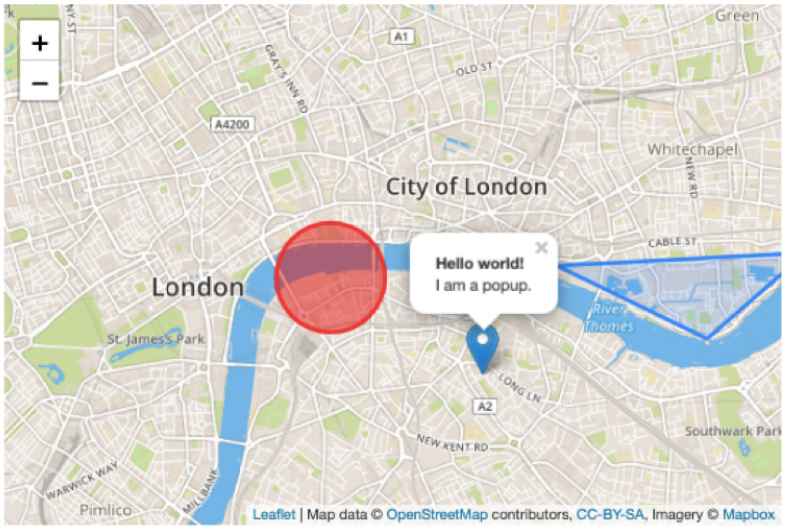
- "Data Visualization: A Practical Introduction by Kieran Healy
- "Fundamentals of Data Visualization" by Claus Wilke
- "R for Data Science" by Garrett Grolemund and Hadley Wickham

My blogdown website (all materials public) for my DSBA 5122: Visual Analytics course: dsba5122.com

bit.ly/dsbaRintro

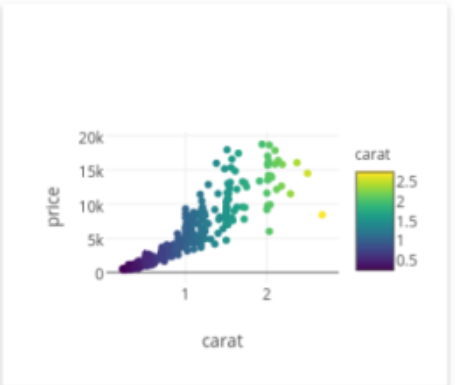
What if you need JavaScript libraries?

```
<!doctype html>
...<html> == $0
  ▶<head>…</head>
  ▼<body class>
    ▶<div id="mapid" style="width: 600px; height: 400px; position: relative;" class="leaflet-
    container leaflet-touch leaflet-fade-anim leaflet-grab leaflet-touch-drag leaflet-touch-zoom"
    tabindex="0">…</div>
    ▼<script>

          var mymap = L.map('mapid').setView([51.505, -0.09], 13);

          L.tileLayer('https://api.tiles.mapbox.com/v4/{id}/{z}/{x}/{y}.png?
      access_token=pk.eyJ1IjoibWFwYm94IiwiYSI6ImNpejY4NXVycTA2emYycXBndHRqcmZ3N3gifQ.rJcFIG214Ari
      ISLbB6B5aw', {
              maxZoom: 18,
              attribution: 'Map data &copy; <a
      href="https://www.openstreetmap.org/">OpenStreetMap</a> contributors, ' +
                  '<a href="https://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>, ' +
                  'Imagery © <a href="https://www.mapbox.com/">Mapbox</a>',
              id: 'mapbox.streets'
          }).addTo(mymap);

          L.marker([51.5, -0.09]
              .bindPopup("<b>Hel

          L.circle([51.508, -0.1
              color: 'red',
              fillColor: '#f03',
              fillOpacity: 0.5
          }).addTo(mymap).bindPo

          L.polygon([
              [51.509, -0.08],
              [51.503, -0.06],
              [51.51, -0.047]
          ]).addTo(mymap).bindPo


          var popup = L.popup();
```



bit.ly/dsbaRintro

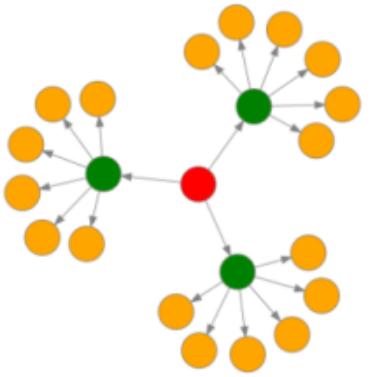
htmlwidgets: JavaScript Packages in R



plotly Star 861

Create interactive web graphics via Plotly's JavaScript graphing library.

- author: cpsievert
- tags: d3, webgl
- js libraries: plotly.js



DiagrammeR Star 680

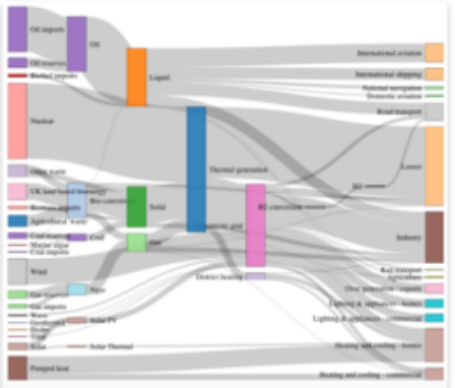
Easily create graph diagrams using R.

- author: rich-iannone
- tags: visualization, diagram, networks
- js libraries: d3,viz,mermaid

leaflet Star 306

Leaflet is an open-source JavaScript library for interactive maps. This package makes it easy to create Leaflet maps from R.

- author: rstudio
- tags: visualization, maps
- js libraries: leaflet

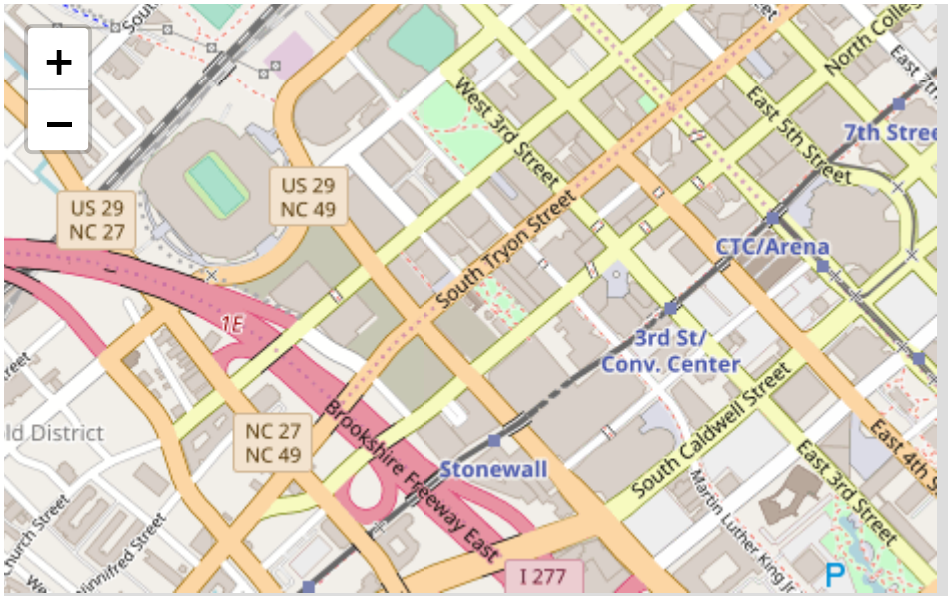


bit.ly/dsbaRintro

htmlwidgets: leaflet + rtweet

```
library(leaflet); library(rtweet) # see rtweet.info

tweets <- search_tweets(
   q = "beer",                    # query term
   geocode = "35.23,-80.84,10mi",# geolocation: charlotte + 10mi
  n = 3200) %>%                  # max tweets via public api
    lat_lng() %>%                # convert lng/lat to dataframe
    filter(!is.na(lng))          # remove records missing lng/lat

tweets %>%                     # tweets data frame
  leaflet() %>%                # pass to leaflet
  addTiles() %>%               # add tiles
  addCircleMarkers(
    lng=tweets$lng,            # latitude
    lat=tweets$lat,            # longitude
    popup = tweets$text, # add tweet text
    clusterOptions = markerClusterOptions()
  )
```

htmlwidgets: leaflet



htmlwidgets: JavaScript's DT (DataTable)

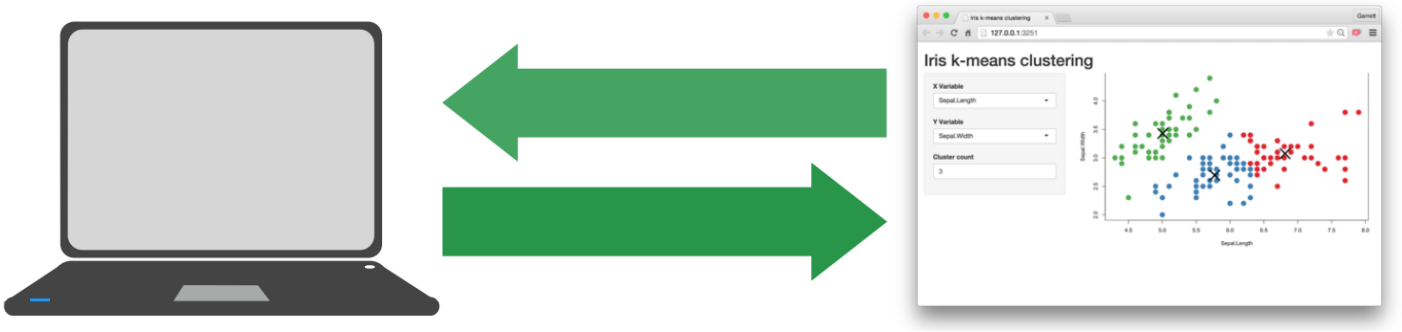
source	text
Untappd	Drinking a Just Too Likeable Lemongrass Miso Gose by @OozlefinchBeers at @SaludNODA — https://t.co/cXWDL7tpdc
Untappd	Drinking a Psychedelic Flow by @SycamoreBrewing @ Tru by Hilton Charlotte Ayrsley — https://t.co/8P7TStfCl1
Instagram	Exactly how I'm going to sleep tonight after the day I've had. Dead Sleep S'mores Stout Brewed and conditioned with graham, marshmallow, and chocolate 8% @dg_beer @ Charlotte, North Carolina https://t.co/tXCjsM4jc5

Showing 4 to 6 of 15 entries

Previous 1 2 3 4 5 Next

Shiny

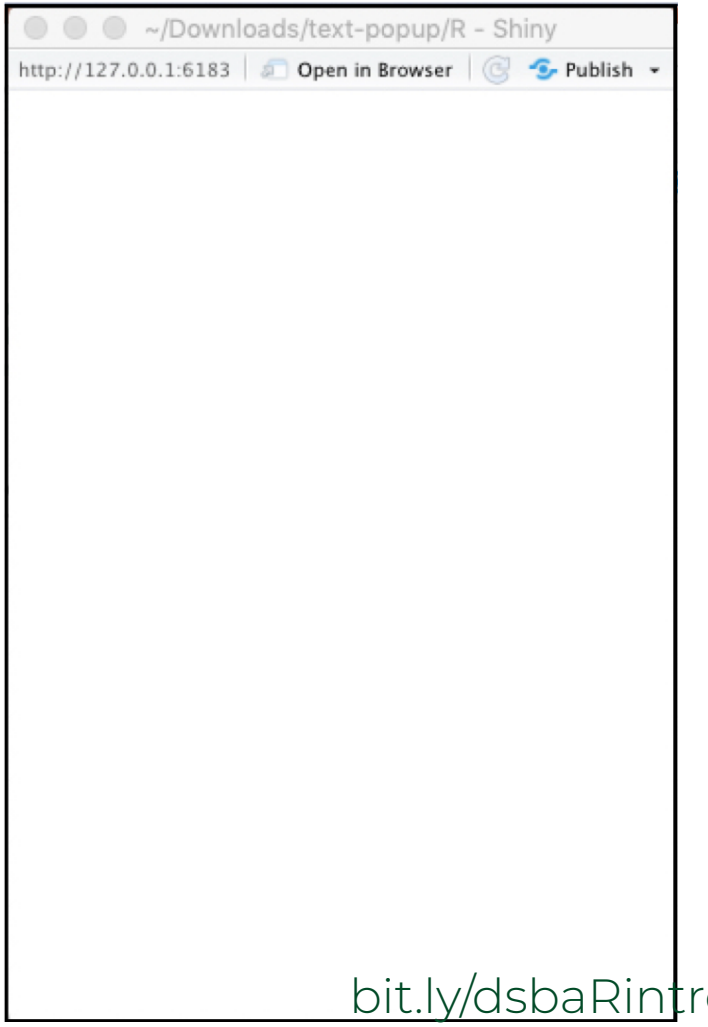
A Shiny app is a web page (UI) connected to a computer running a live R session (Server).



Users can manipulate the UI, which will cause the server to update the UI's displays (by running R code).

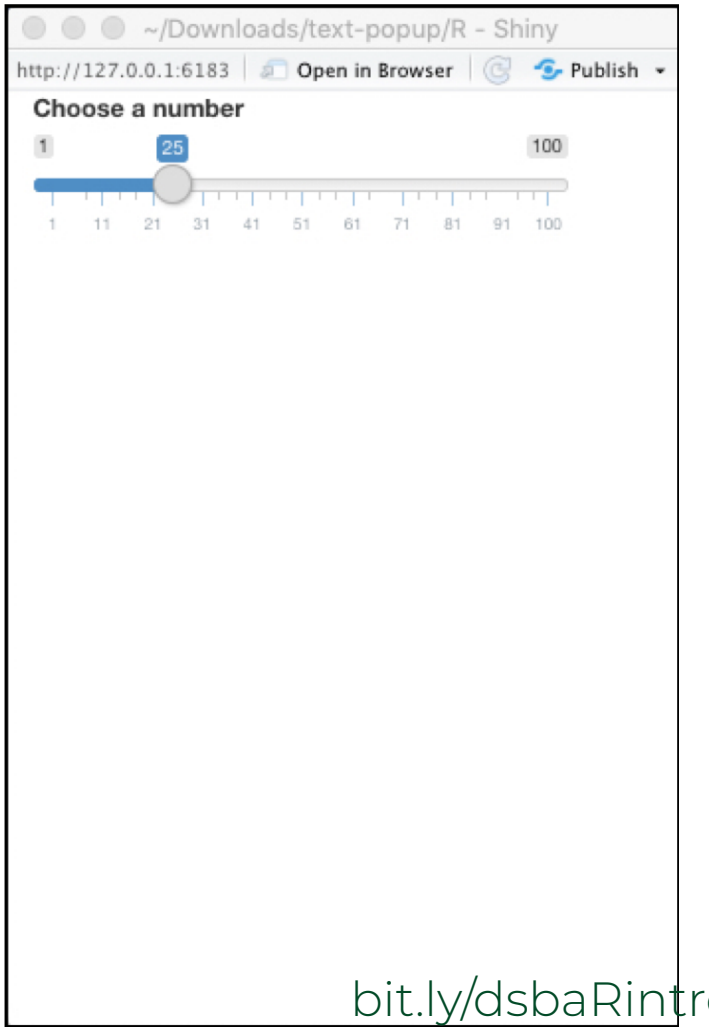
Shiny


```
library(shiny)

ui <- fluidPage()

server <- function(input, output) {}

shinyApp(ui = ui, server = server)
```



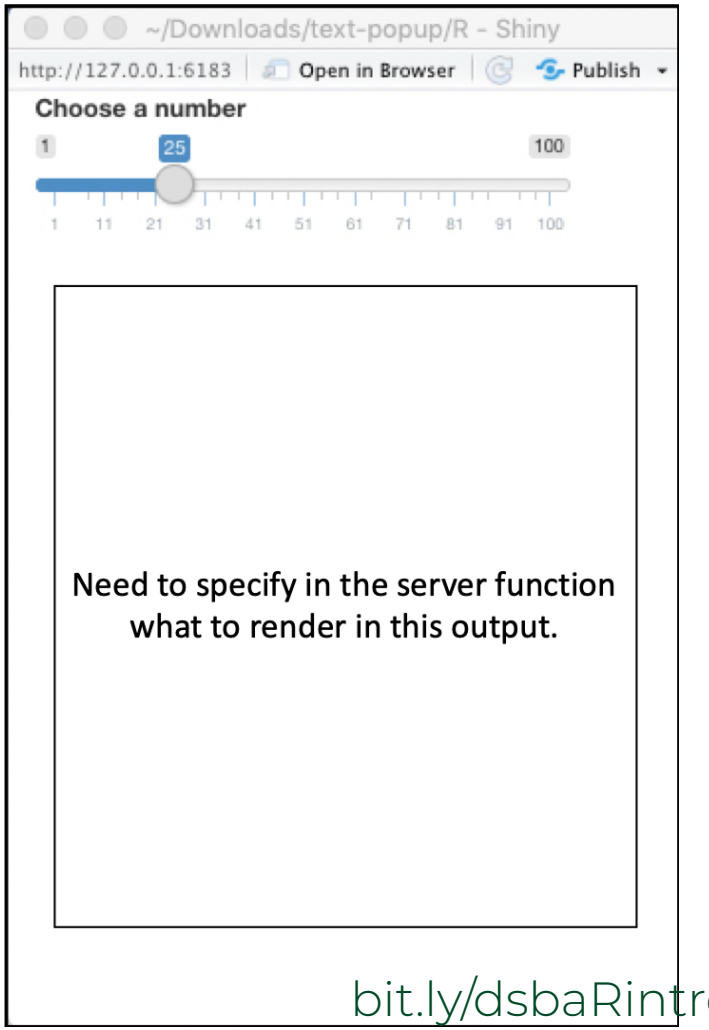
Shiny

```
library(shiny)

ui <- fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100)
)

server <- function(input, output) {

}

shinyApp(ui = ui, server = server)
```



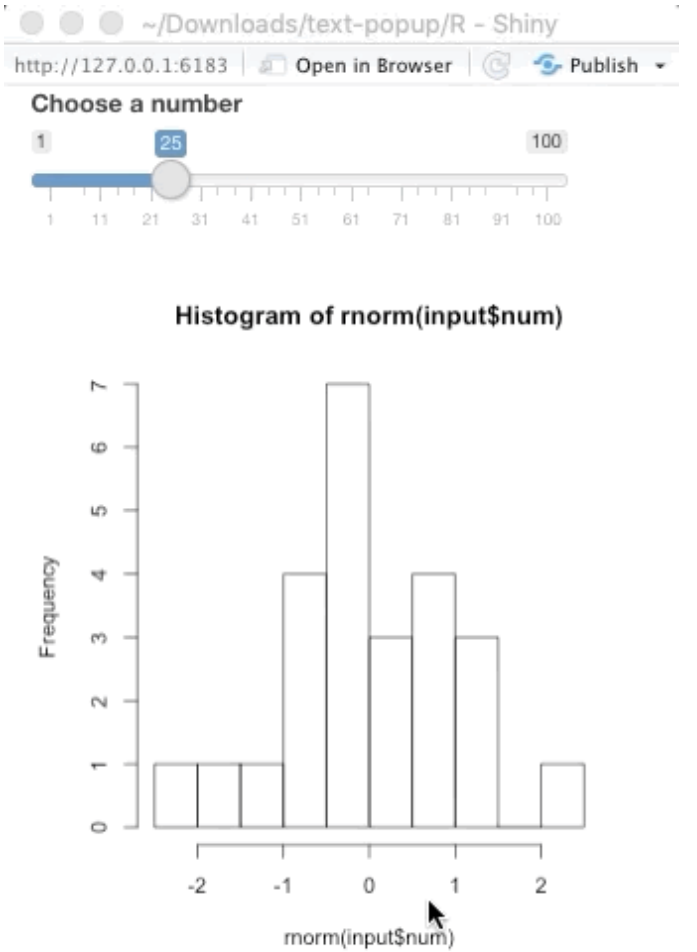
Shiny

```
library(shiny)

ui <- fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100),
  plotOutput("hist")
)

server <- function(input, output) {

}

shinyApp(ui = ui, server = server)
```



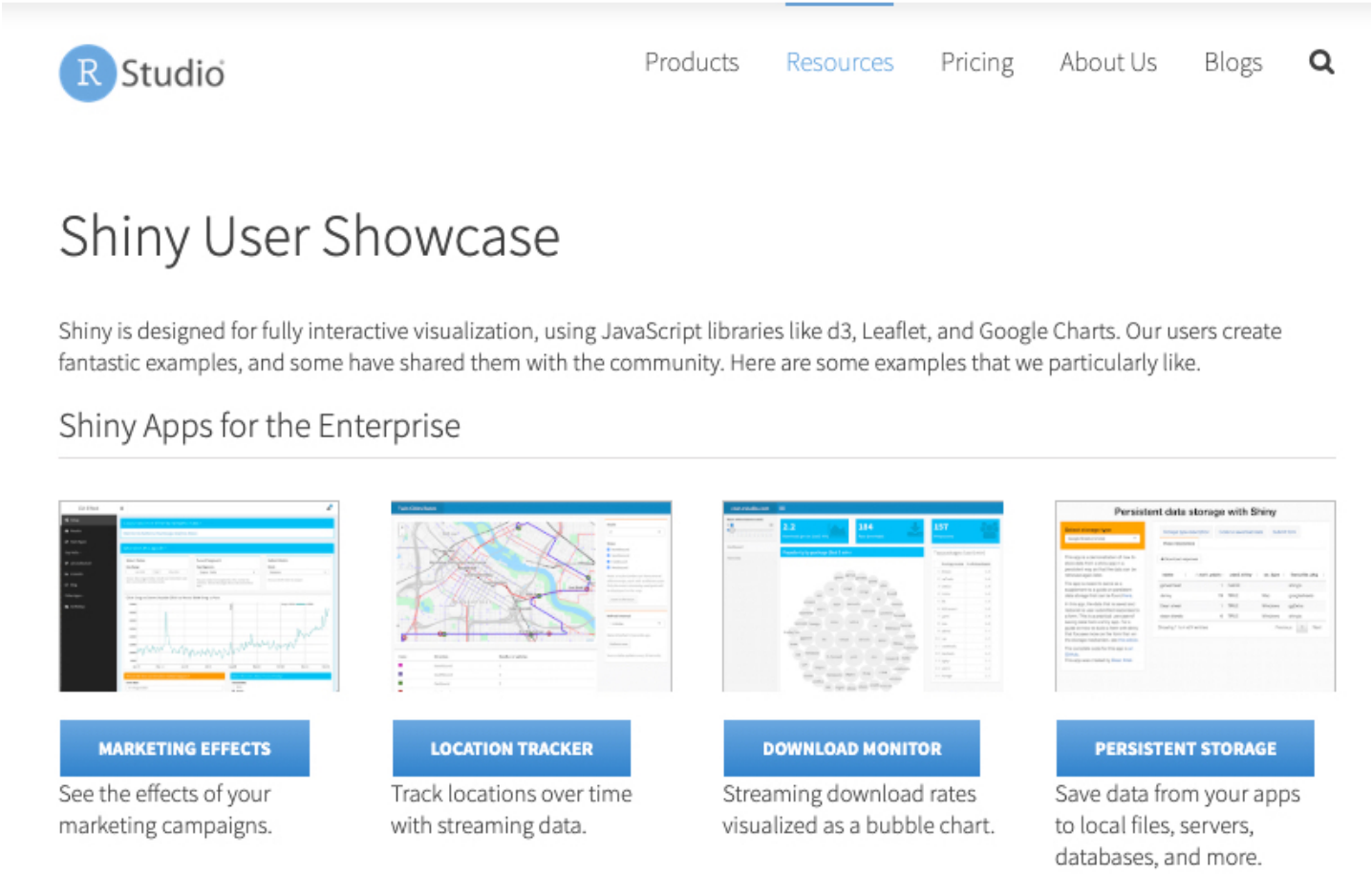
Shiny


```
library(shiny)

ui <- fluidPage(
   sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100),
   plotOutput("hist")
)

server <- function(input, output) {
  output$hist <- renderPlot({
     hist(rnorm(input$num))
  })
}

shinyApp(ui = ui, server = server)
```



bit.ly/dsbaRintro

RStudio's Shiny Gallery



https://shiny.rstudio.com/gallery/

bit.ly/dsbaRintro

Can we do more?

More
flexibility?

Interactivity?

Best in
class?

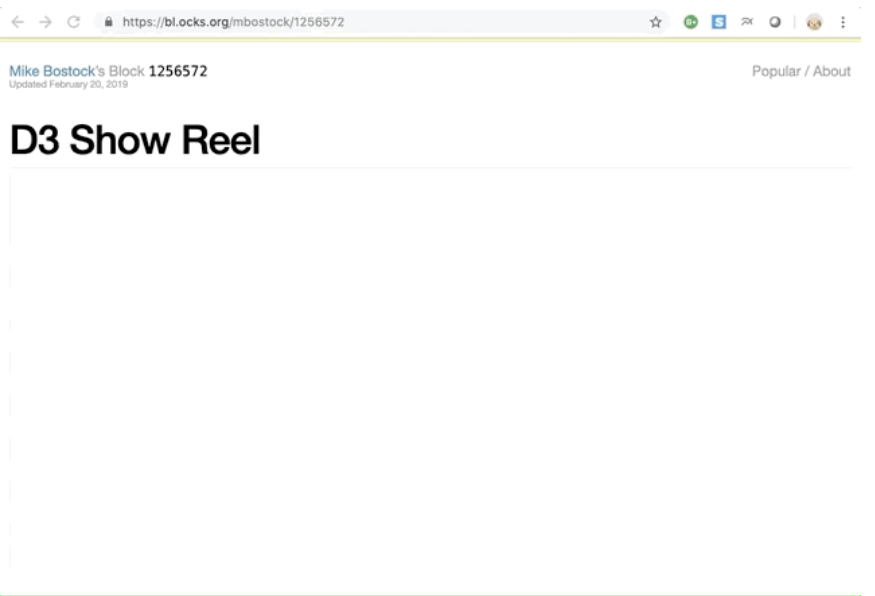
Can we do more?

More flexibility?

Interactivity?

Best in class?

Use D3 in R: r2d3.



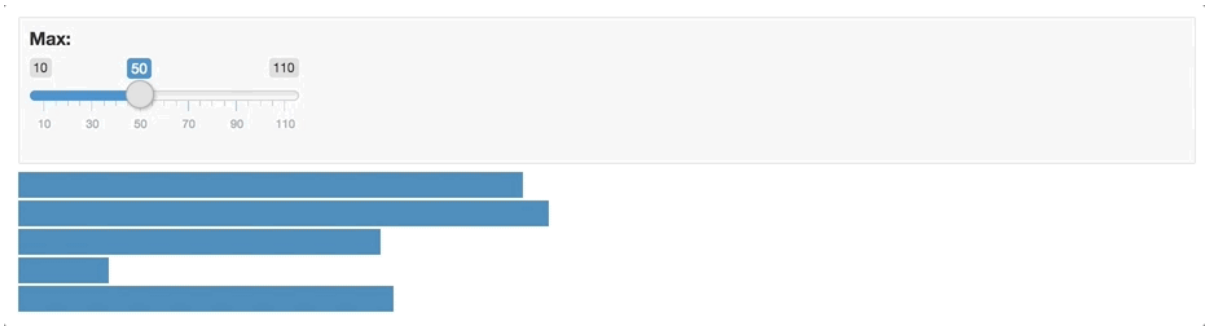
https://bl.ocks.org/mbostock/1256572

bit.ly/dsbaRintro

r2d3


```
// baranims.js saved locally
var barHeight = Math.floor(height / data.length);

var bars = r2d3.svg.selectAll('rect')
    .data(r2d3.data);

bars.enter()
    .append('rect')
      .attr('width', function(d) { return d * width; })
      .attr('height', barHeight)
      .attr('y', function(d, i) { return i * barHeight; })
      .attr('fill', 'steelblue');

bars.exit().remove();

bars.transition()
  .duration(100)
  .attr("width", function(d) { return d * width; });
```

```
# shiny code with r2d3
library(shiny)
library(r2d3)

ui <- fluidPage(
  inputPanel(
    sliderInput(
      "bar_max", label = "Max:",
      min = 0.1, max = 1.0, value = 0.2, step = 0.1)
  ),
  d3Output("d3")
)

server <- function(input, output) {
  output$d3 <- renderD3({
    r2d3(
      runif(5, 0, input$bar_max),
      script = "baranims.js" # see left
    )
  })
}

shinyApp(ui = ui, server = server)
```



bit.ly/dsbaRintro

Takeaways

R/RStudio is not your old base R tools.

- Tidyverse, shiny, RMarkdown, etc.

Takeaways

R/RStudio is not your old base R tools.

- Tidyverse, shiny, RMarkdown, etc.

Choose the best tool for the job, maybe Excel or Tableau.

- But if its a data science pipeline + data visualizations: R is hard to compete with.

Takeaways

R/RStudio is not your old base R tools.

- Tidyverse, shiny, RMarkdown, etc.

Choose the best tool for the job, maybe Excel or Tableau.

- But if its a data science pipeline + data visualizations: R is hard to compete with.

A lot of times, don't need to choose R vs. Python, C++, JS.

- Take the best of each and make it into an R package.

Takeaways

R/RStudio is not your old base R tools.

- Tidyverse, shiny, RMarkdown, etc.

Choose the best tool for the job, maybe Excel or Tableau.

- But if its a data science pipeline + data visualizations: R is hard to compete with.

A lot of times, don't need to choose R vs. Python, C++, JS.

- Take the best of each and make it into an R package.

Data visualization is part art and science (and design!).

- The best way to learn: do it, get feedback, iterate.

Happy R programming!

Artwork by @allison_horst

bit.ly/dsbaRintro