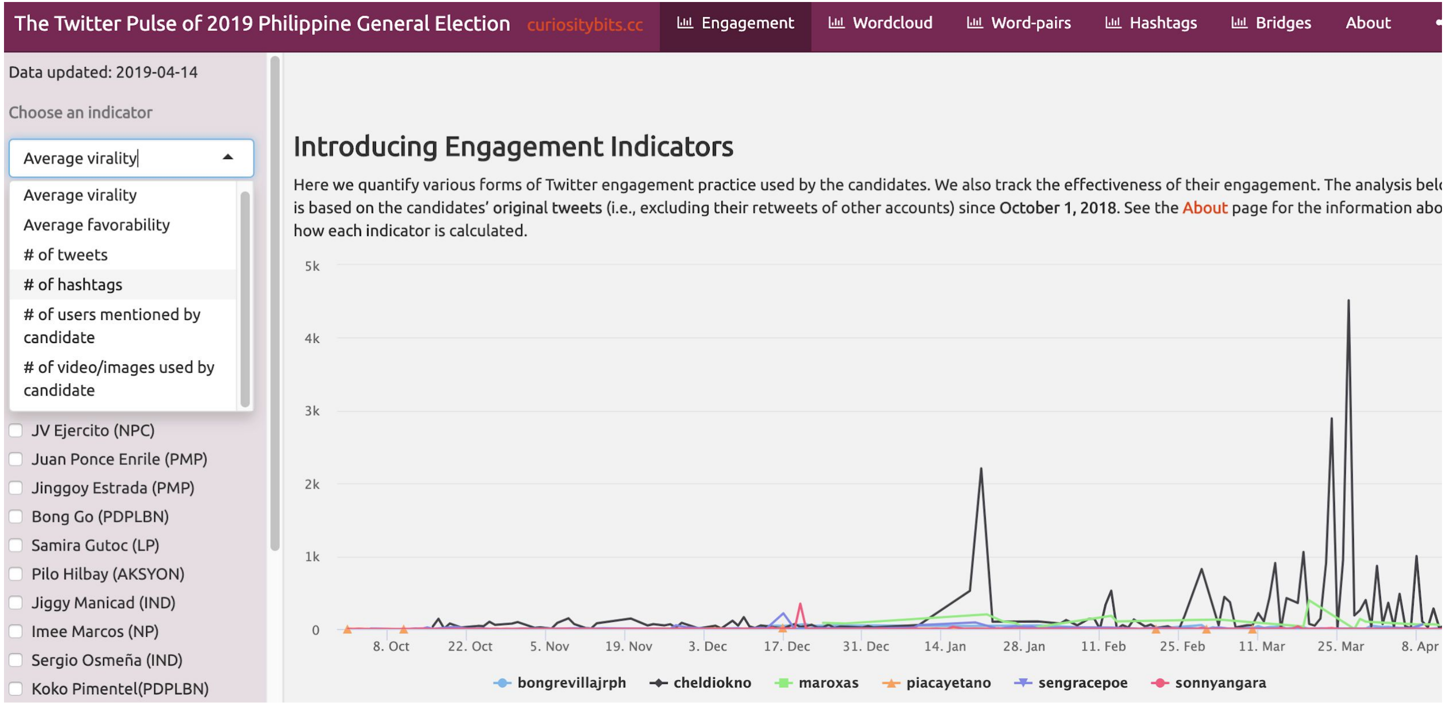
Interactive visualization:
R Shiny

Weiai Xu (Wayne), PhD
Assistant Professor
Department of Communication, UMass-Amherst
Email: weiaixu@umass.edu
curiositybits.cc

Examples of R Shiny apps



Data updated: 2019-04-14

Choose an indicator

Average virality

- Average virality
- Average favorability
- # of tweets
- # of hashtags
- # of users mentioned by candidate
- # of video/images used by candidate

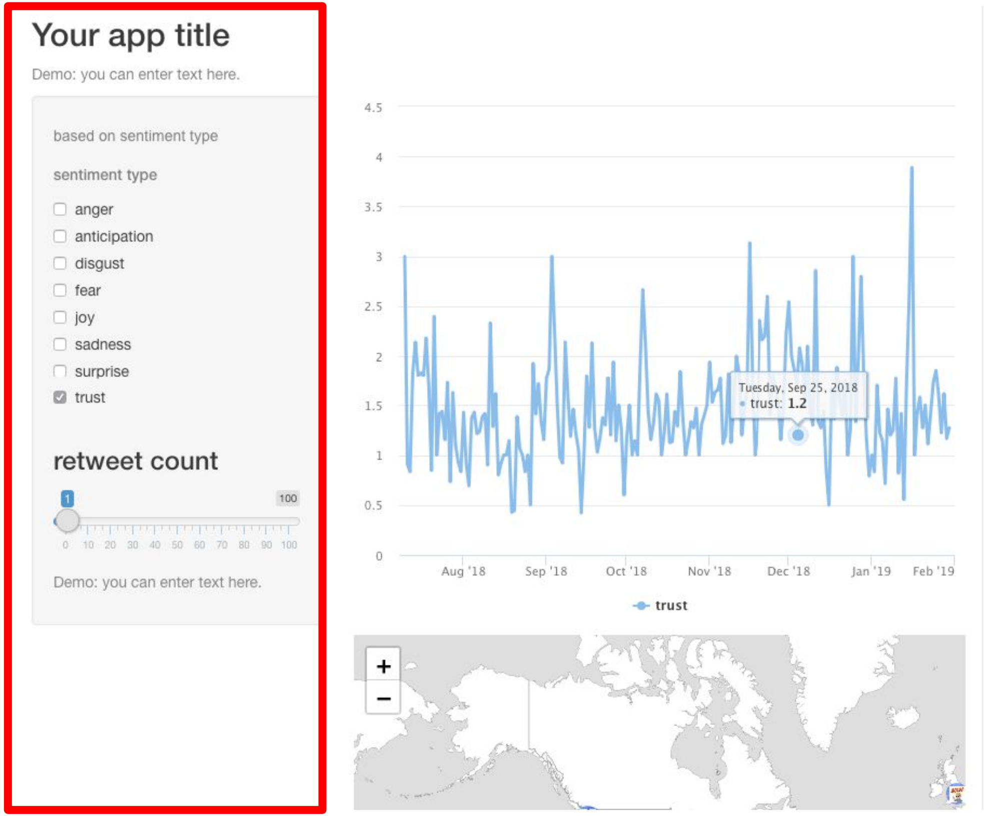
- □ JV Ejercito (NPC)
- □ Juan Ponce Enrile (PMP)
- □ Jinggoy Estrada (PMP)
- □ Bong Go (PDPLBN)
- □ Samira Gutoc (LP)
- □ Pilo Hilbay (AKSYON)
- □ Jiggy Manicad (IND)
- □ Imee Marcos (NP)
- □ Sergio Osmeña (IND)
- □ Koko Pimentel(PDPLBN)

Introducing Engagement Indicators

Here we quantify various forms of Twitter engagement practice used by the candidates. We also track the effectiveness of their engagement. The analysis belc is based on the candidates' original tweets (i.e., excluding their retweets of other accounts) since **October 1, 2018**. See the About page for the information abo how each indicator is calculated.

https://curiositybits.shinyapps.io/PH_Tracker_dashboard/

Components in a R Shiny app



Your app title

Demo: you can enter text here.

based on sentiment type

sentiment type

- □ anger
- □ anticipation
- □ disgust
- □ fear
- □ joy
- □ sadness
- □ surprise
- ☑ trust

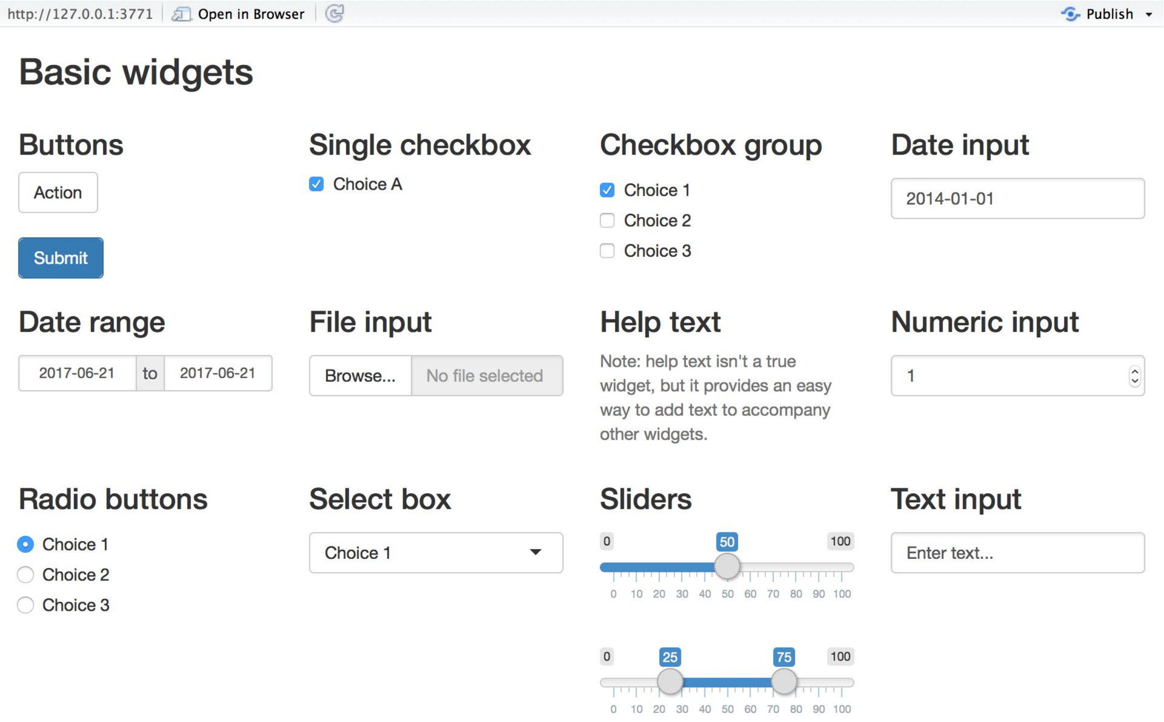
retweet count

Demo: you can enter text here.

Sidebar Panel: contains one or multiple ***control widgets***

Allow users to make selection or enter values

Components in a R Shiny app



Basic widgets

Buttons

Single checkbox

Checkbox group

Date input

2014-01-01

Date range

2017-06-21 to 2017-06-21

File input

Help text

Note: help text isn't a true widget, but it provides an easy way to add text to accompany other widgets.

Numeric input

1

Radio buttons

Select box

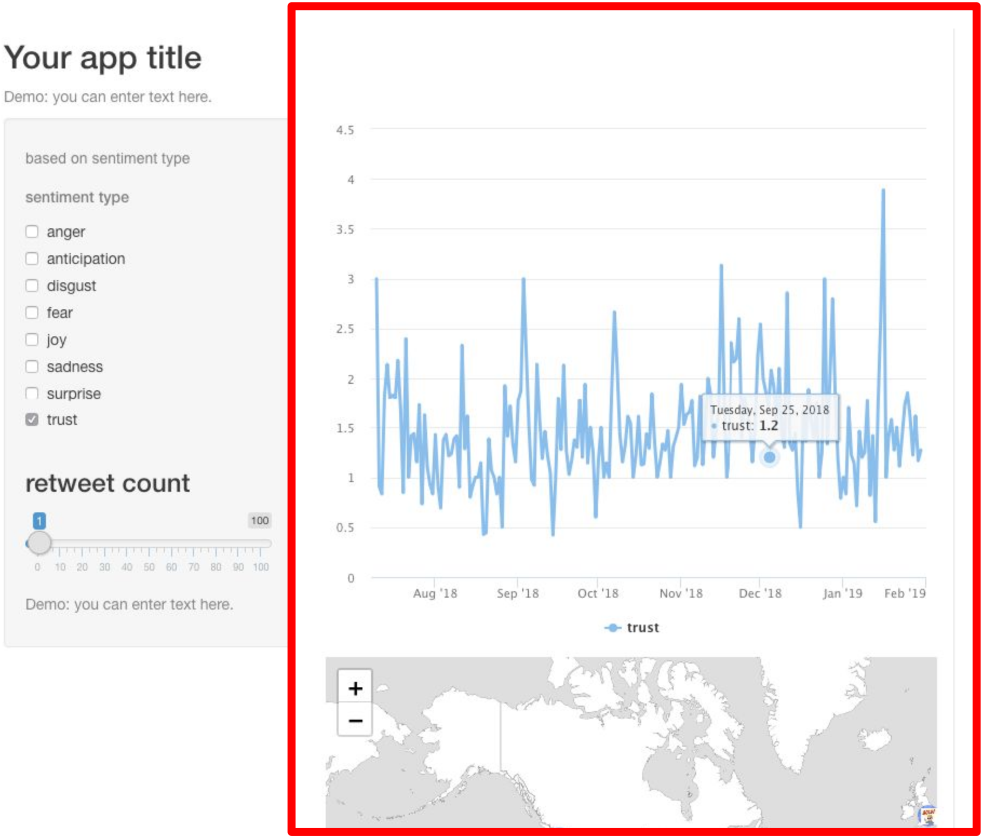
Sliders

Text input

Basic widgets available in R Shiny

https://shiny.rstudio.com/tutorial/written-tutorial/lesson3/

Components in a R Shiny app



Your app title

Demo: you can enter text here.

based on sentiment type

sentiment type

☐ anger
☐ anticipation
☐ disgust
☐ fear
☐ joy
☐ sadness
☐ surprise
☑ trust

retweet count

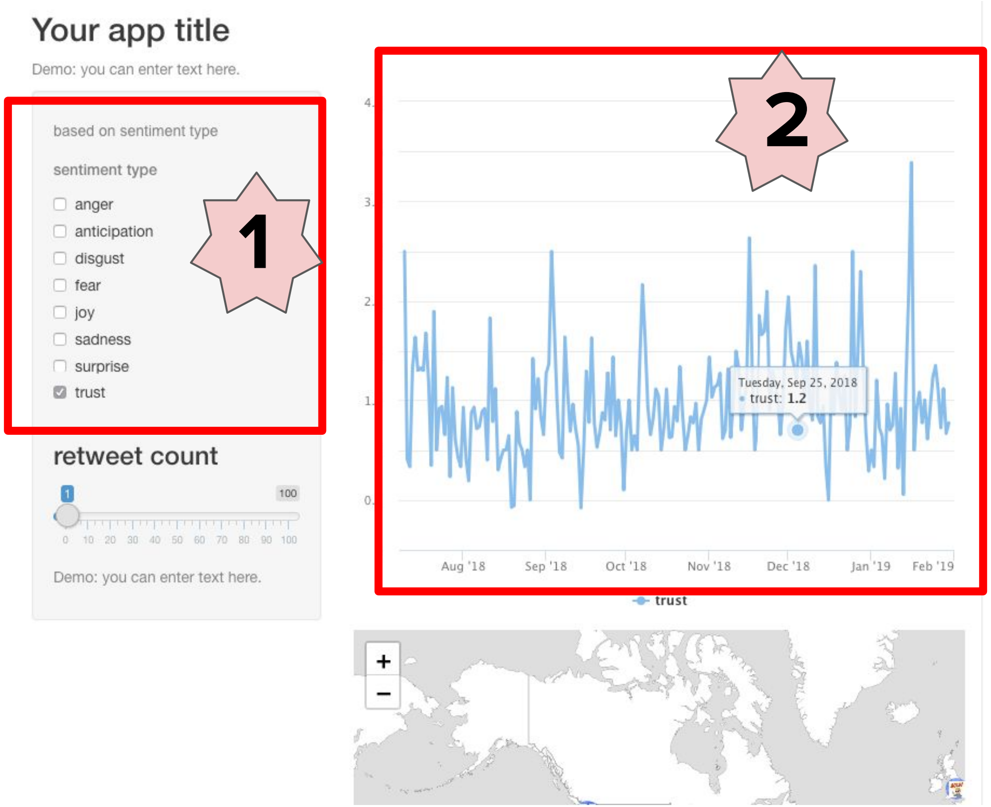
1 – 100

Demo: you can enter text here.

Main Panel: contains visual outputs

Outputs change in response to user selection

How does a R Shiny app work



Your app title

Demo: you can enter text here.

based on sentiment type

sentiment type

- □ anger
- □ anticipation
- □ disgust
- □ fear
- □ joy
- □ sadness
- □ surprise
- ☑ trust

retweet count

1 – 100

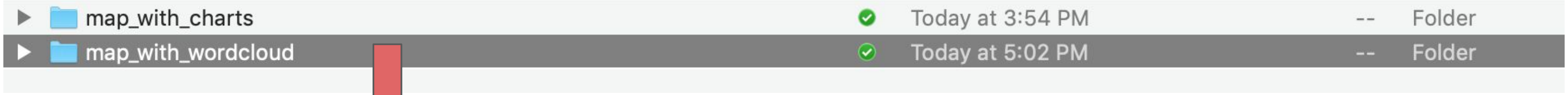
Demo: you can enter text here.

Tuesday, Sep 25, 2018
● trust: 1.2

1 When a user makes a selection in the Sidebar Panel, it creates **an input value.**

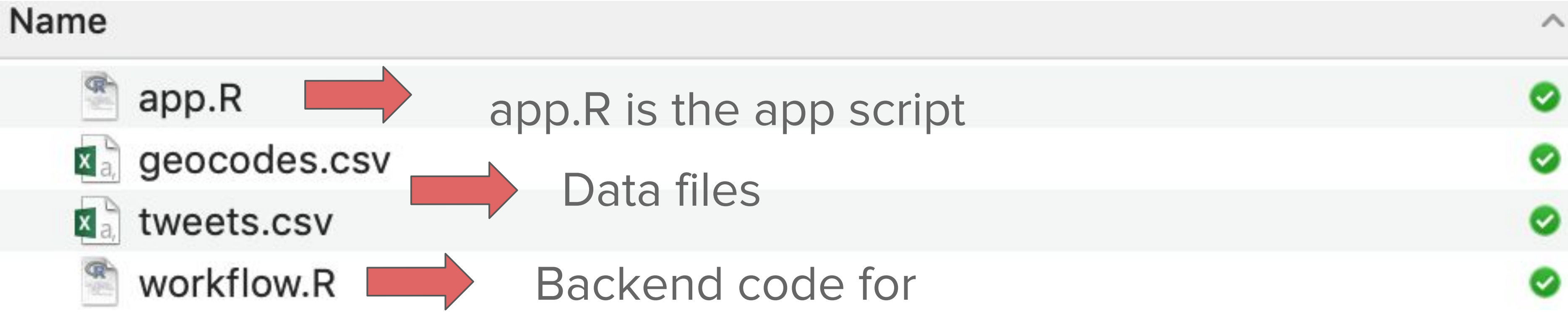
2 **The input value** is used to select cases for visualization

What files does a R Shiny app consist of



map_with_charts	Today at 3:54 PM	--	Folder
map_with_wordcloud	Today at 5:02 PM	--	Folder

Each app has a stand-alone folder



Name

app.R

geocodes.csv

tweets.csv

workflow.R

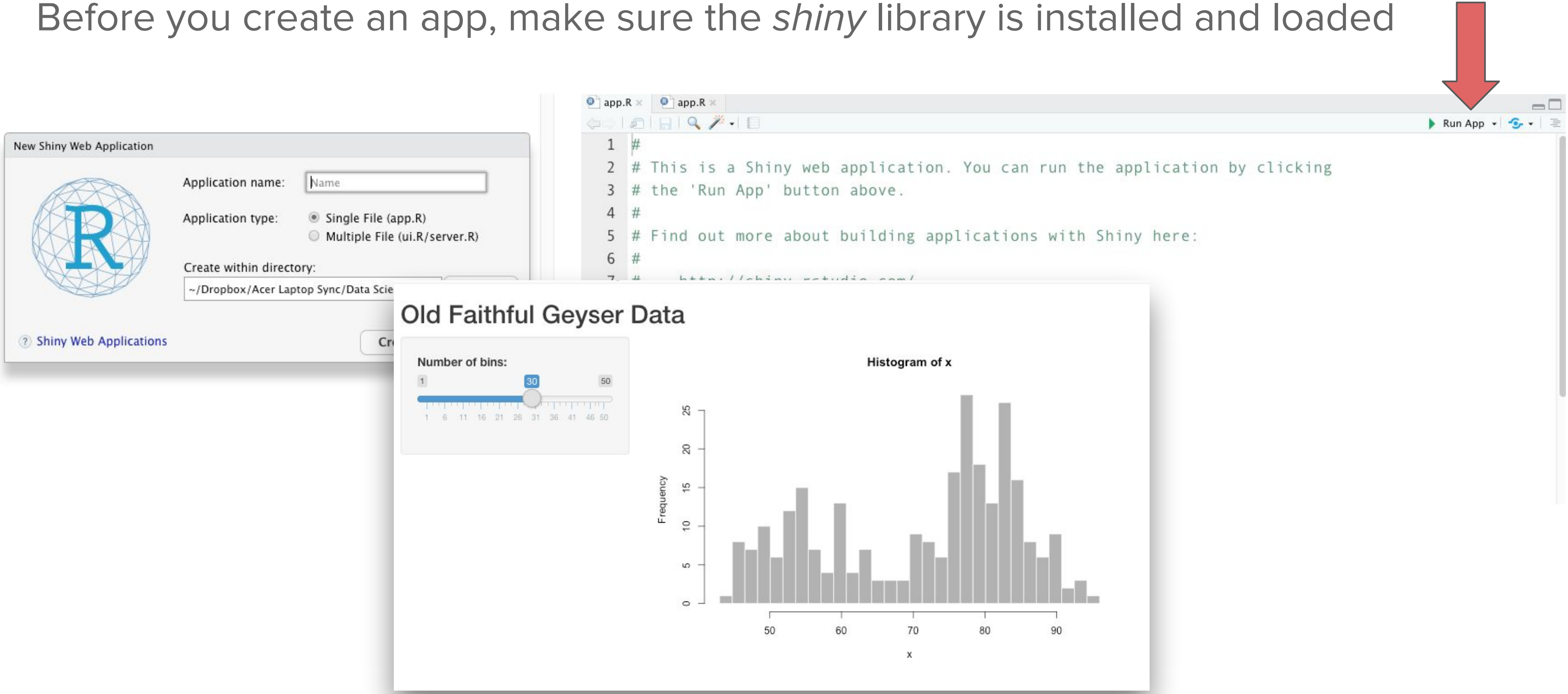
app.R is the app script

Data files

Backend code for pre-processing data

Create an example

Before you create an app, make sure the *shiny* library is installed and loaded



New Shiny Web Application

Application name: Name

Application type: ● Single File (app.R) ○ Multiple File (ui.R/server.R)

Create within directory: ~/Dropbox/Acer Laptop Sync/Data Scie

Shiny Web Applications

Cr

```
#
# This is a Shiny web application. You can run the application by clicking
# the 'Run App' button above.
#
# Find out more about building applications with Shiny here:
#
#    http://shiny.rstudio.com/
```

Old Faithful Geyser Data

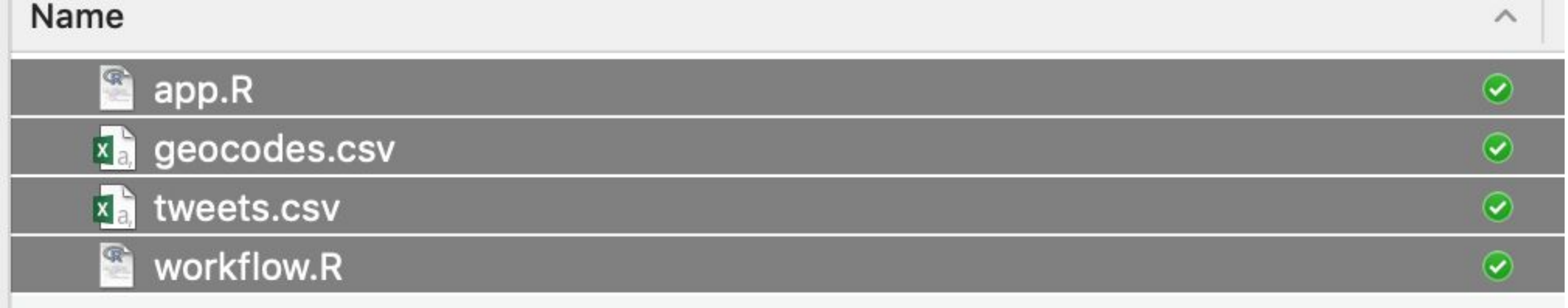
Number of bins: 1 30 50

Test example files

Example files are available on Moodle

Create a folder in your laptop and move app.R and associated data files to the folder. Then open app.R in RStudio.

Make sure the example app works on your laptop



Name

app.R ✓

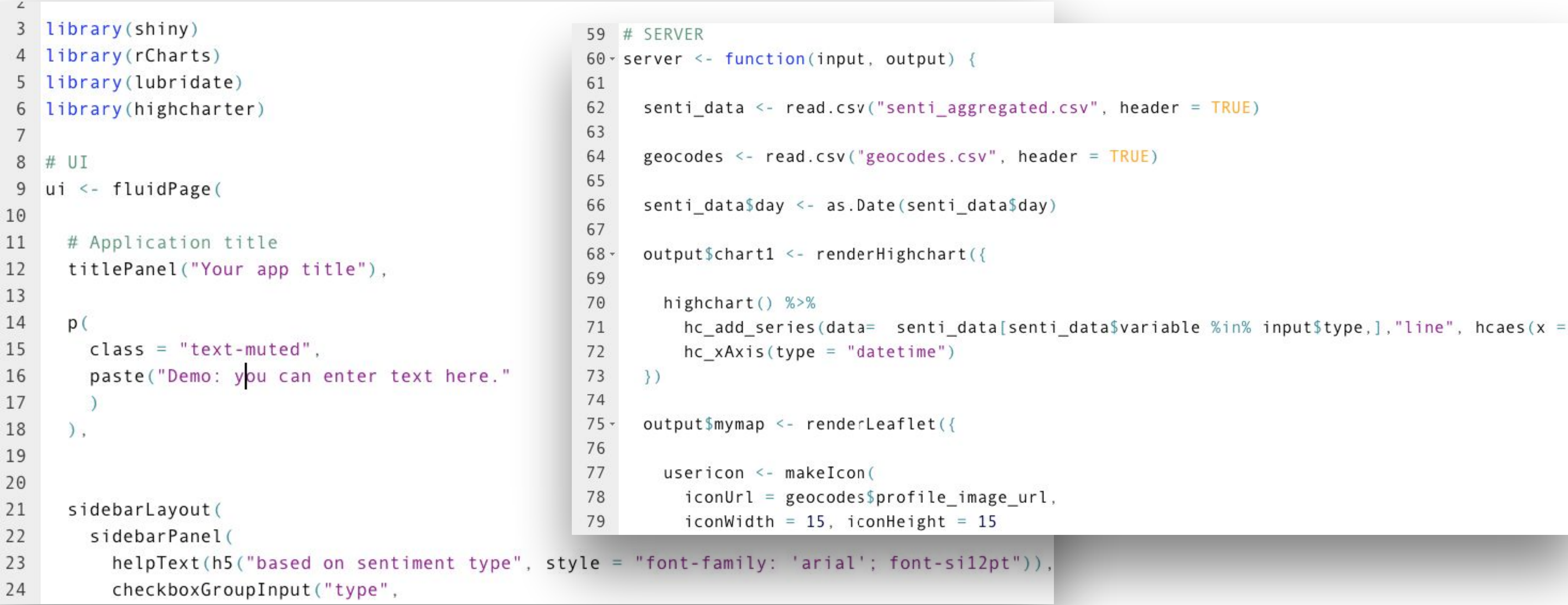
geocodes.csv ✓

tweets.csv ✓

workflow.R ✓

What’s in app.R?

The **ui** part and the **server** part

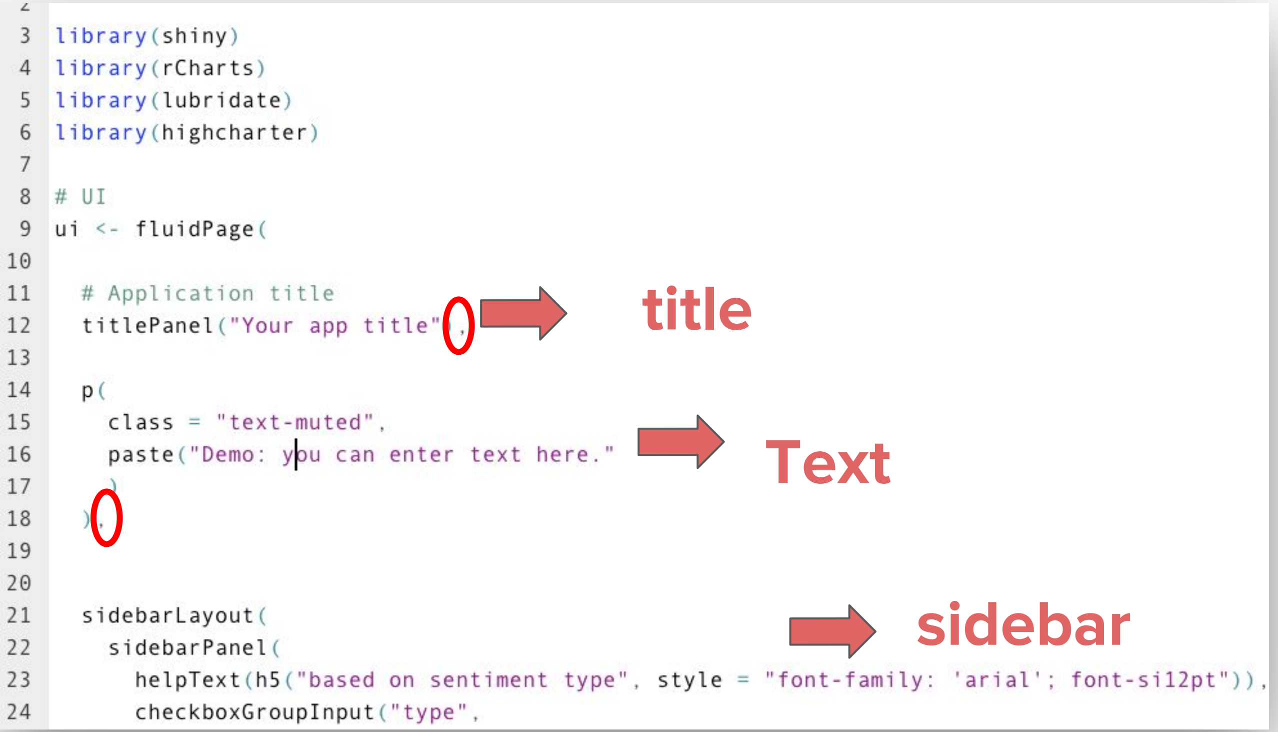


```r
library(shiny)
library(rCharts)
library(lubridate)
library(highcharter)

# UI
ui <- fluidPage(

  # Application title
  titlePanel("Your app title"),

  p(
    class = "text-muted",
    paste("Demo: you can enter text here."
    )
  ),


  sidebarLayout(
    sidebarPanel(
      helpText(h5("based on sentiment type", style = "font-family: 'arial'; font-si12pt")),
      checkboxGroupInput("type",
```

```r
# SERVER
server <- function(input, output) {

  senti_data <- read.csv("senti_aggregated.csv", header = TRUE)

  geocodes <- read.csv("geocodes.csv", header = TRUE)

  senti_data$day <- as.Date(senti_data$day)

  output$chart1 <- renderHighchart({

    highchart() %>%
      hc_add_series(data=  senti_data[senti_data$variable %in% input$type,],"line", hcaes(x =
      hc_xAxis(type = "datetime")
  })

  output$mymap <- renderLeaflet({

    usericon <- makeIcon(
      iconUrl = geocodes$profile_image_url,
      iconWidth = 15, iconHeight = 15
```

What’s in app.R?

The **ui** part

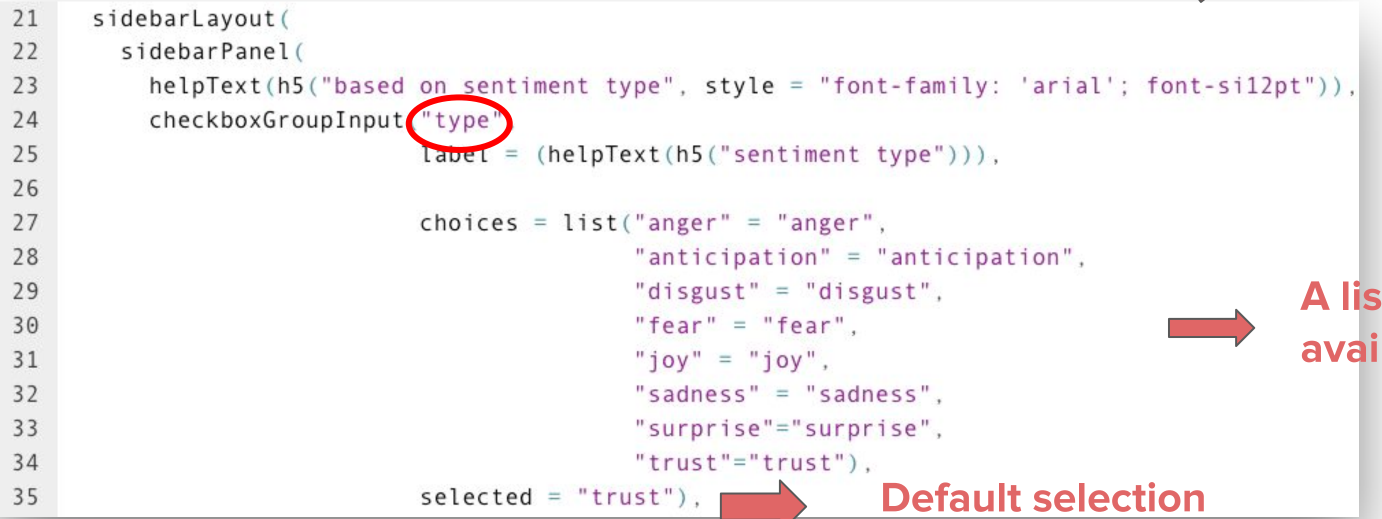


```
library(shiny)
library(rCharts)
library(lubridate)
library(highcharter)

# UI
ui <- fluidPage(

  # Application title
  titlePanel("Your app title"),

  p(
    class = "text-muted",
    paste("Demo: you can enter text here."
    )
  ),


  sidebarLayout(
    sidebarPanel(
      helpText(h5("based on sentiment type", style = "font-family: 'arial'; font-si12pt")),
      checkboxGroupInput("type",
```

Q: What should I do if I want to change the app name to *Class Demo* and add an introduction?

What’s in app.R?

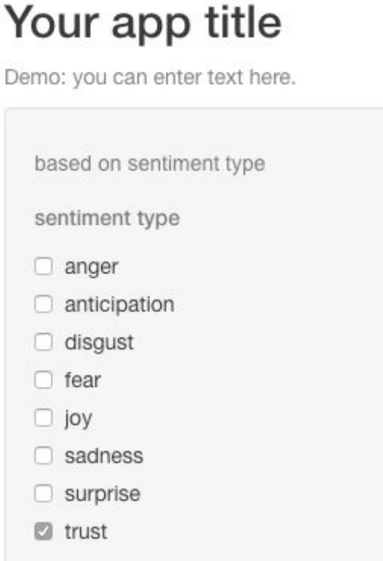
The **ui** part



```
sidebarLayout(
  sidebarPanel(
    helpText(h5("based on sentiment type", style = "font-family: 'arial'; font-si12pt")),
    checkboxGroupInput("type",
                       label = (helpText(h5("sentiment type"))),

                       choices = list("anger" = "anger",
                                      "anticipation" = "anticipation",
                                      "disgust" = "disgust",
                                      "fear" = "fear",
                                      "joy" = "joy",
                                      "sadness" = "sadness",
                                      "surprise"="surprise",
                                      "trust"="trust"),
                       selected = "trust"),
```

A list of choices available to users

Default selection



Your app title

Demo: you can enter text here.

based on sentiment type

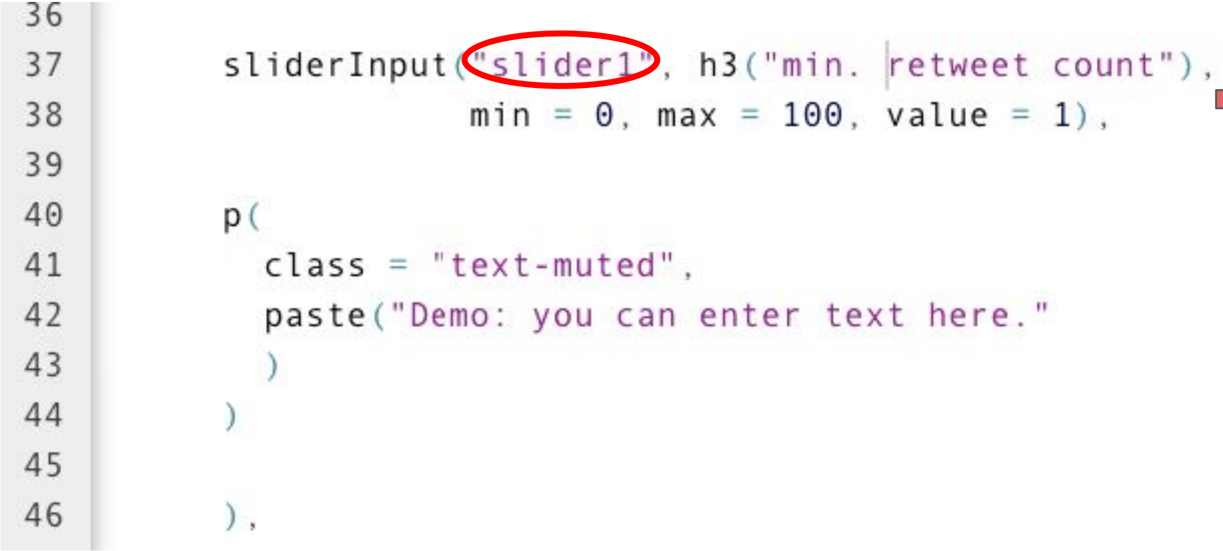
sentiment type

□ anger
□ anticipation
□ disgust
□ fear
□ joy
□ sadness
□ surprise
☑ trust

When a user selects anger, the selection creates an input value. The value is the string *anger*, the input value is stored as input$type

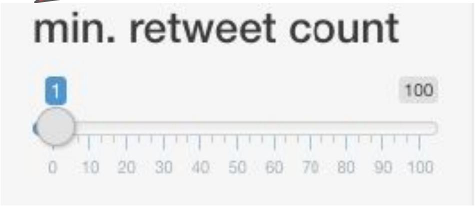
What’s in app.R?

The **ui** part



```
      sliderInput("slider1", h3("min. retweet count"),
                  min = 0, max = 100, value = 1),

      p(
        class = "text-muted",
        paste("Demo: you can enter text here."
        )
      )

      ),
```

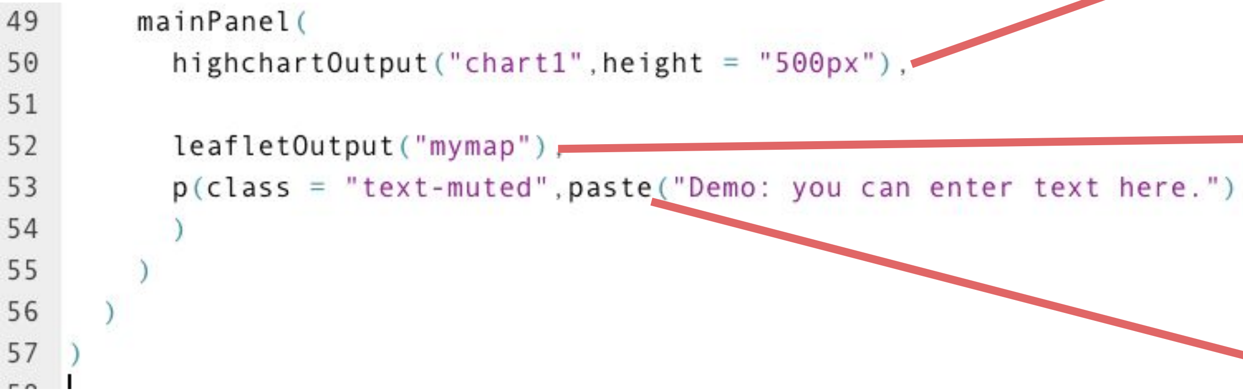
A slider bar



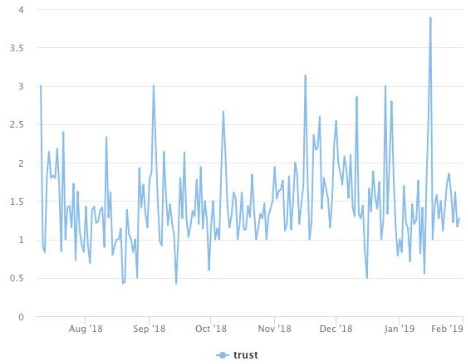
Every time a user sets on a number, it creates an input value. The input value is stored as input$slider1

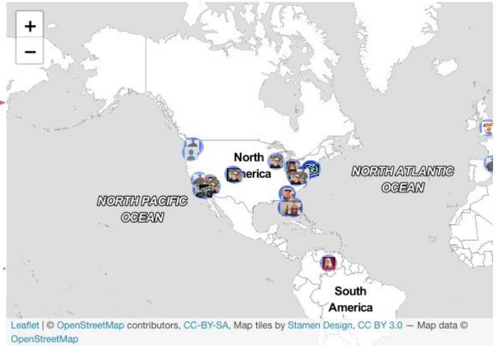
What’s in app.R?

The ui part



```
  mainPanel(
   highchartOutput("chart1",height = "500px"),

   leafletOutput("mymap"),
   p(class = "text-muted",paste("Demo: you can enter text here.")
   )
  )
 )
)
```





+
−
North America
NORTH PACIFIC OCEAN
NORTH ATLANTIC OCEAN
South America
Leaflet | © OpenStreetMap contributors, CC-BY-SA, Map tiles by Stamen Design, CC BY 3.0 — Map data © OpenStreetMap

Demo: you can enter text here.

What to visualize

What’s in app.R?

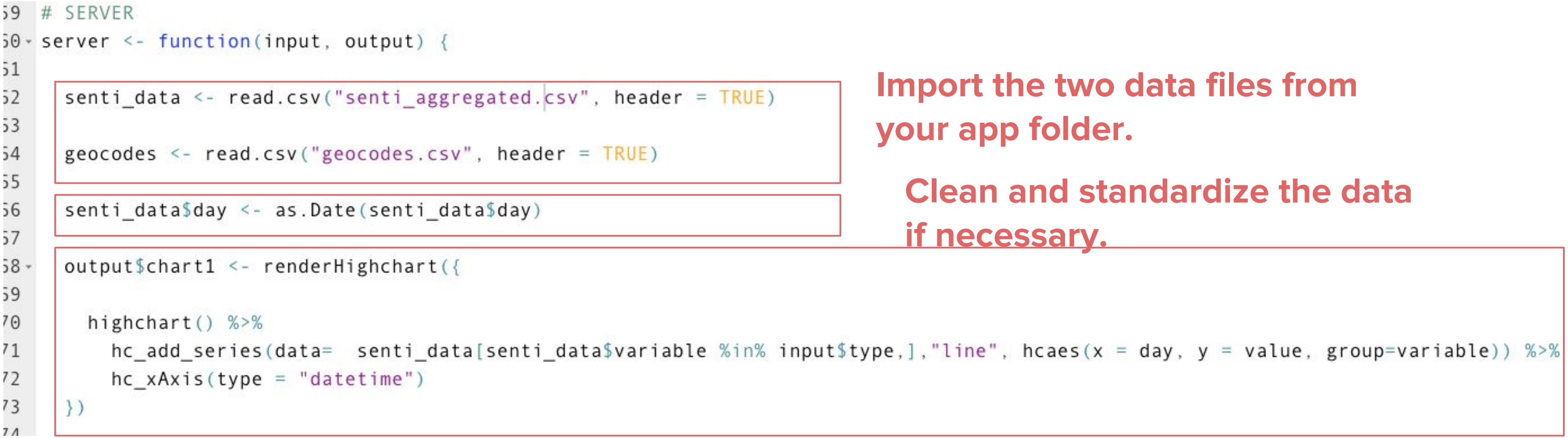
The **server** part

Do you remember the two input values created from user selection?

input$type
input$slider1

What’s in app.R?

The **server** part: import data, process data, and create visualization



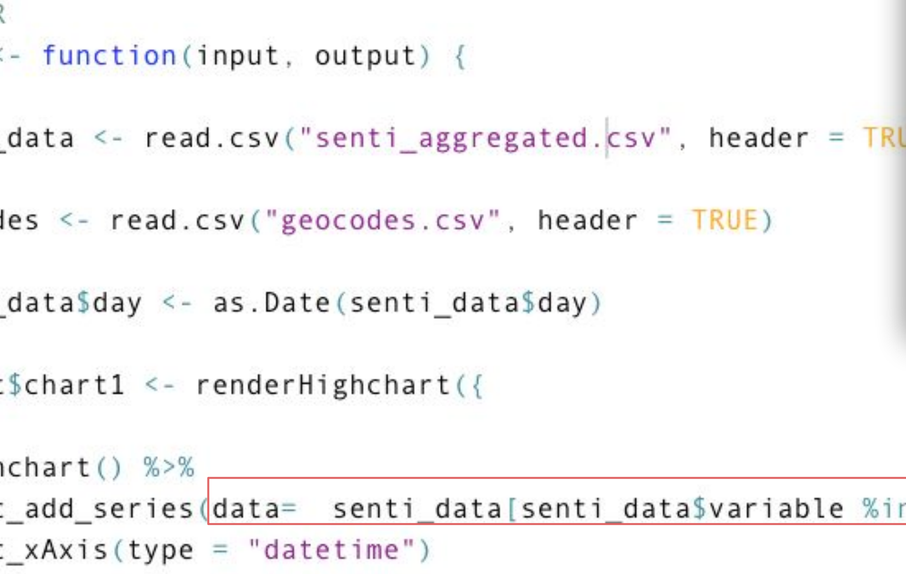
```
# SERVER
server <- function(input, output) {

  senti_data <- read.csv("senti_aggregated.csv", header = TRUE)

  geocodes <- read.csv("geocodes.csv", header = TRUE)

  senti_data$day <- as.Date(senti_data$day)

  output$chart1 <- renderHighchart({

    highchart() %>%
      hc_add_series(data=  senti_data[senti_data$variable %in% input$type,],"line", hcaes(x = day, y = value, group=variable)) %>%
      hc_xAxis(type = "datetime")
  })
```

Import the two data files from your app folder.

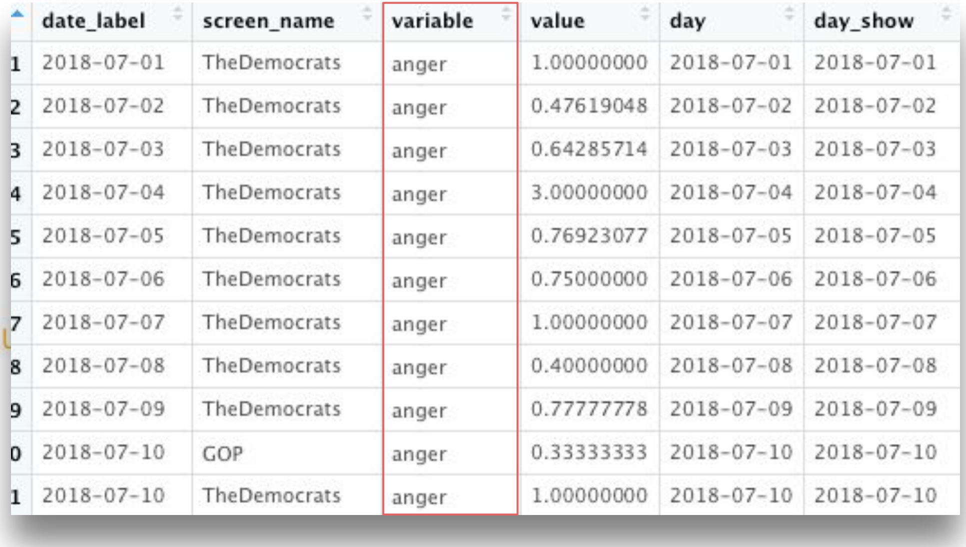
Clean and standardize the data if necessary.

The part where you create a highchart showing sentiment trends.

What’s in app.R?



```
R
<- function(input, output) {

_data <- read.csv("senti_aggregated.csv", header = TRU

des <- read.csv("geocodes.csv", header = TRUE)

_data$day <- as.Date(senti_data$day)

t$chart1 <- renderHighchart({

hchart() %>%
_add_series(data=  senti_data[senti_data$variable %in% input$type,],"line", hcaes(x = day, y = value, group=vari
_xAxis(type = "datetime")
```

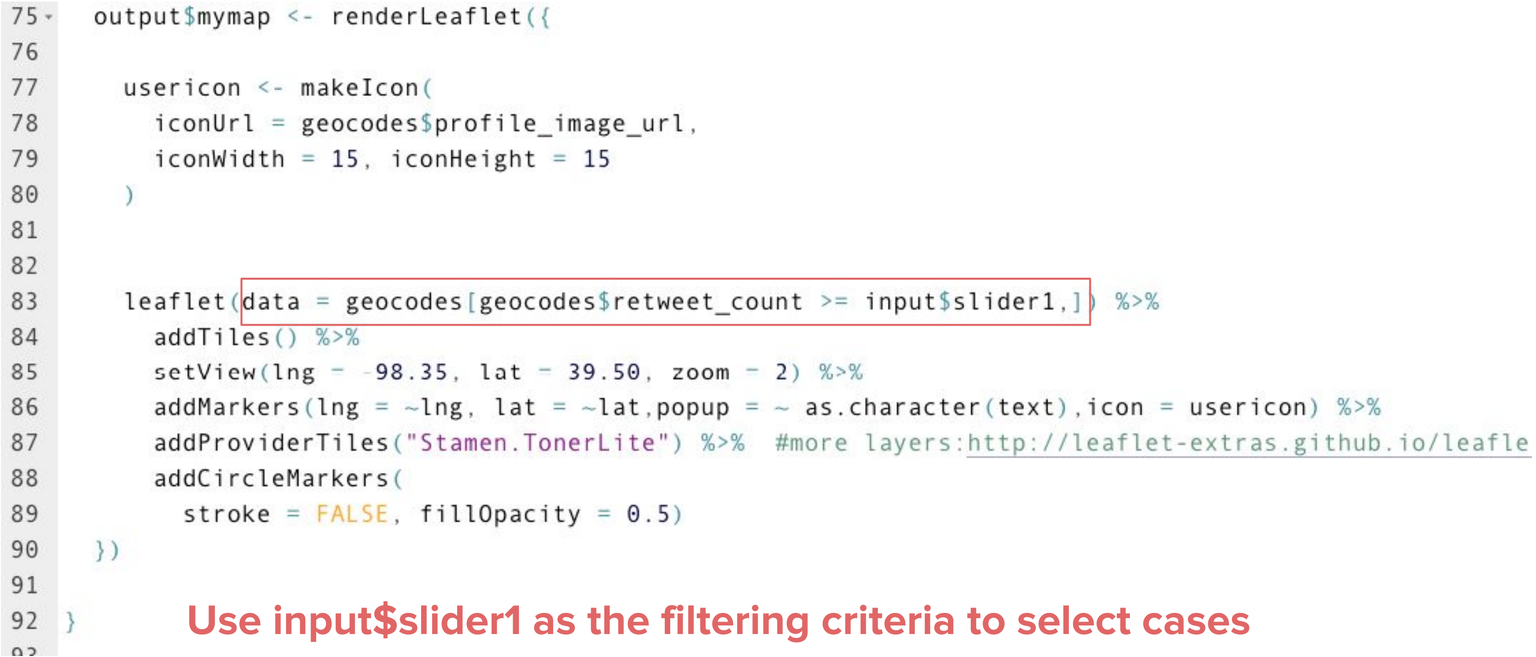


	date_label	screen_name	variable	value	day	day_show
1	2018-07-01	TheDemocrats	anger	1.00000000	2018-07-01	2018-07-01
2	2018-07-02	TheDemocrats	anger	0.47619048	2018-07-02	2018-07-02
3	2018-07-03	TheDemocrats	anger	0.64285714	2018-07-03	2018-07-03
4	2018-07-04	TheDemocrats	anger	3.00000000	2018-07-04	2018-07-04
5	2018-07-05	TheDemocrats	anger	0.76923077	2018-07-05	2018-07-05
6	2018-07-06	TheDemocrats	anger	0.75000000	2018-07-06	2018-07-06
7	2018-07-07	TheDemocrats	anger	1.00000000	2018-07-07	2018-07-07
8	2018-07-08	TheDemocrats	anger	0.40000000	2018-07-08	2018-07-08
9	2018-07-09	TheDemocrats	anger	0.77777778	2018-07-09	2018-07-09
0	2018-07-10	GOP	anger	0.33333333	2018-07-10	2018-07-10
1	2018-07-10	TheDemocrats	anger	1.00000000	2018-07-10	2018-07-10

Use input$type as the filtering criteria to select cases from senti_data

What’s in app.R?

The **server** part: import data, process data, and create visualization



```
  output$mymap <- renderLeaflet({

    usericon <- makeIcon(
      iconUrl = geocodes$profile_image_url,
      iconWidth = 15, iconHeight = 15
    )


    leaflet(data = geocodes[geocodes$retweet_count >= input$slider1,]) %>%
      addTiles() %>%
      setView(lng = -98.35, lat = 39.50, zoom = 2) %>%
      addMarkers(lng = ~lng, lat = ~lat,popup = ~ as.character(text),icon = usericon) %>%
      addProviderTiles("Stamen.TonerLite") %>%  #more layers:http://leaflet-extras.github.io/leafle
      addCircleMarkers(
        stroke = FALSE, fillOpacity = 0.5)
  })

}
```

Use input$slider1 as the filtering criteria to select cases from geocodes

In-class practice

Make sure the example app works on your laptop

See if you can add a new slider bar with the input value named “slider3” and the help text “min. Favorite count.”

Publish a R Shiny app

https://shiny.rstudio.com/articles/shinyapps.html

More resources

https://shiny.rstudio.com/articles/basics.html

Other visualization options

Using R Markdown: https://rmarkdown.rstudio.com/