Statistical Language Models
2019
Week 1 part 2

Dr. Dave Campbell
davecampbell@math.carleton.ca

https://en.wikipedia.org/wiki/Global_warming

In 2013, the Intergovernmental Panel on Climate Change (IPCC) Fifth Assessment Report concluded, "It is *extremely likely* that human influence has been the dominant cause of the observed warming since the mid-20th century."[9] The largest human influence has been the emission of greenhouse gases such as carbon dioxide, methane, and nitrous oxide. Climate model projections summarized in the report indicated that during the 21st century, the global surface temperature is likely to rise a further 0.3 to 1.7 °C (0.5 to 3.1 °F) to 2.6 to 4.8 °C (4.7 to 8.6 °F) depending on the rate of greenhouse gas emissions and on climate feedback effects.[10] These findings have been recognized by the national science academies of the major industrialized nations[11][a] and are not disputed by any scientific body of national or international standing.[13][14]

Assignment #1
See culearn.carleton.ca
for examples and data

Build a ShinyApp to explore Canadian climate change data.

Use any plots and tools from any courses to find and convey the story in the data.

ui.R and server.R are to be submitted as a zip file that will load the data files as named from the current active directory.

My directory names will not be the same as yours

Any library loading / installing must be part of your code.

P(Grade > 0 | Submission > January 20th at 5pm) = 0

Shiny Apps

Goal:

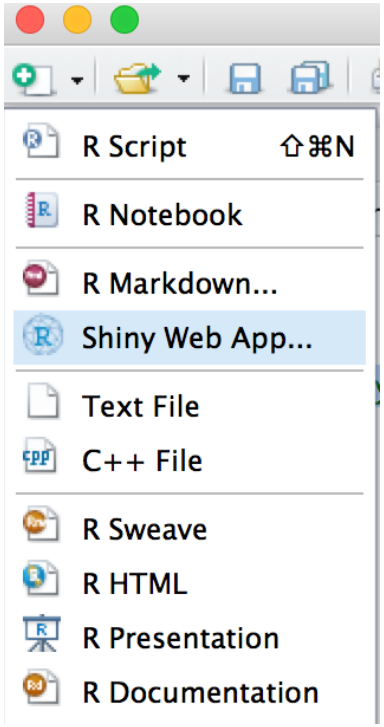
Communicate information

Allow the user to change the variables, timing, type of analysis

Run analysis in real time, output insight into a html page

Viewable on any device since R runs on the server not the user device

Getting Started

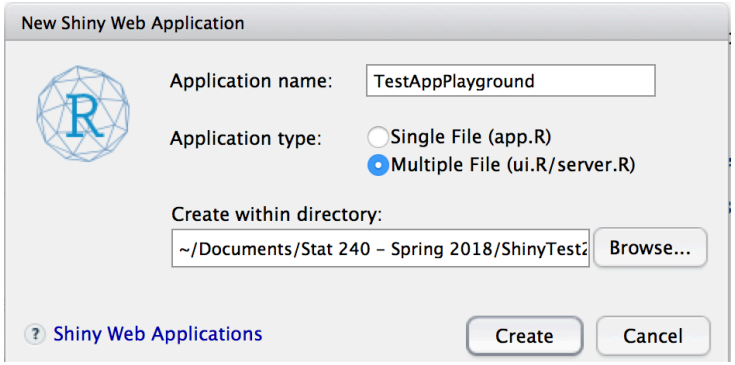


Start a new Shiny Web App

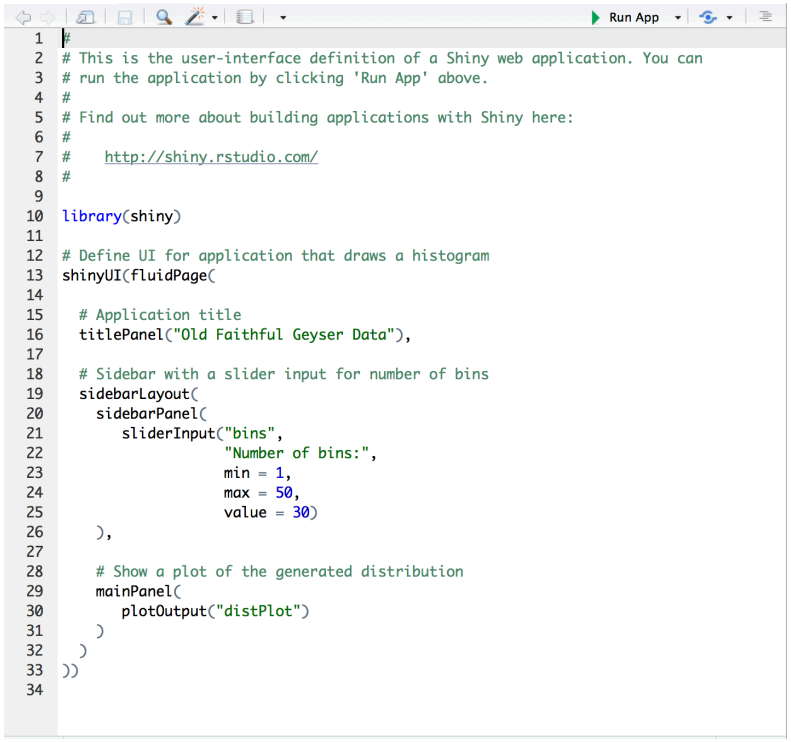
Name it

We'll use "Multiple File"

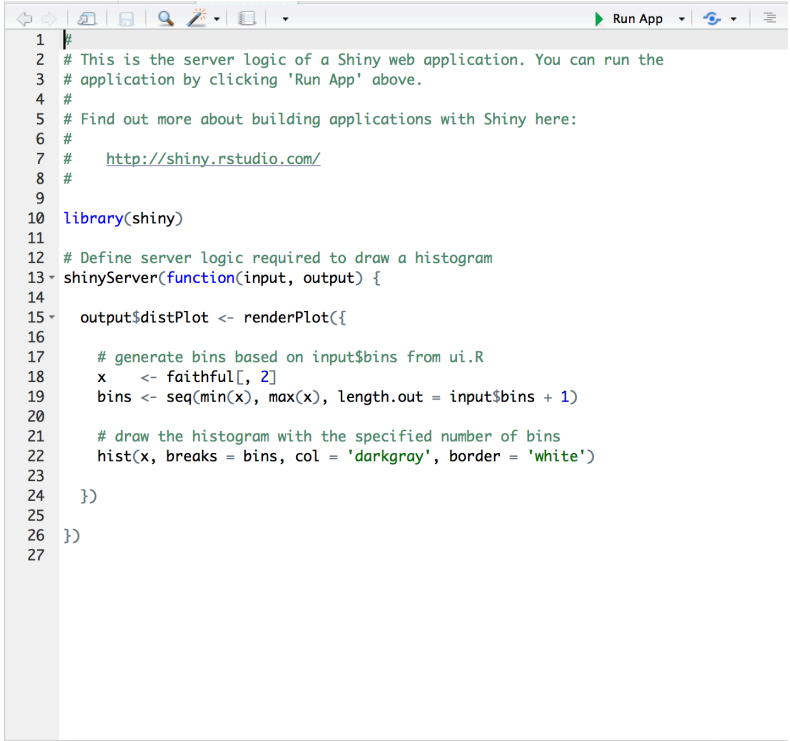
Give it a save location



RStudio will make 2 files, ui.R and server.R



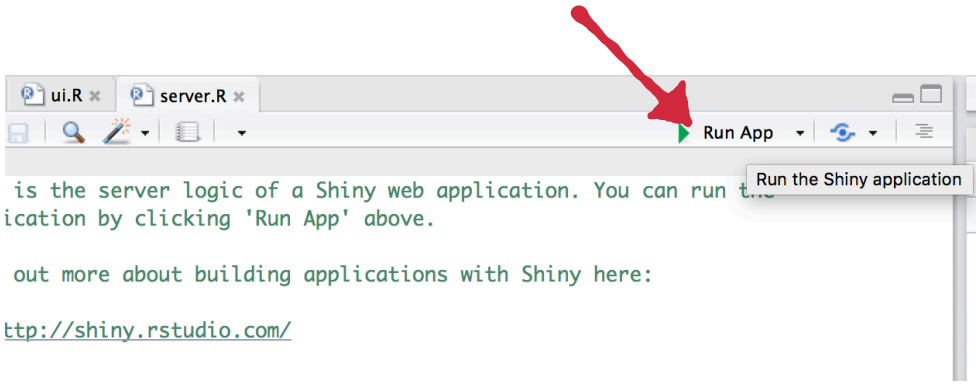
```r
#
# This is the user-interface definition of a Shiny web application. You can
# run the application by clicking 'Run App' above.
#
# Find out more about building applications with Shiny here:
#
#    http://shiny.rstudio.com/
#

library(shiny)

# Define UI for application that draws a histogram
shinyUI(fluidPage(

  # Application title
  titlePanel("Old Faithful Geyser Data"),

  # Sidebar with a slider input for number of bins
  sidebarLayout(
    sidebarPanel(
       sliderInput("bins",
                   "Number of bins:",
                   min = 1,
                   max = 50,
                   value = 30)
    ),

    # Show a plot of the generated distribution
    mainPanel(
       plotOutput("distPlot")
    )
  )
))

```



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

library(shiny)

# Define server logic required to draw a histogram
shinyServer(function(input, output) {

  output$distPlot <- renderPlot({

    # generate bins based on input$bins from ui.R
    x    <- faithful[, 2]
    bins <- seq(min(x), max(x), length.out = input$bins + 1)

    # draw the histogram with the specified number of bins
    hist(x, breaks = bins, col = 'darkgray', border = 'white')

  })

})

```

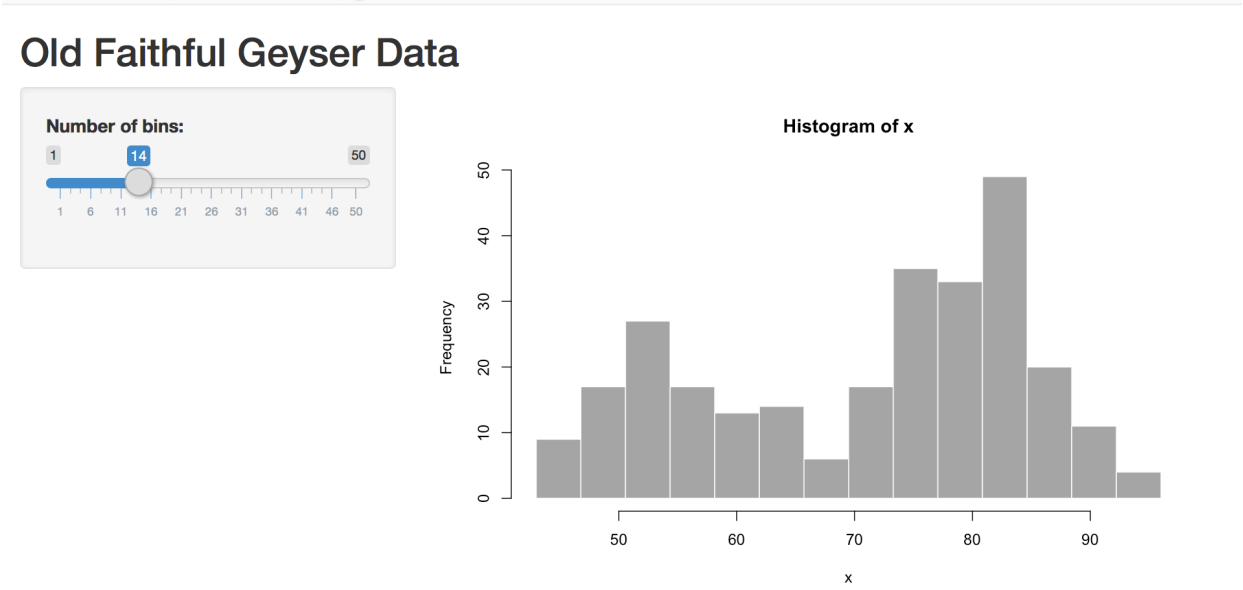
This is a working Shiny App.

Run it by clicking >Run App



A browser window will open that is running the app.

We'll modify these files to build our page.



Inputs are defined using sliders, buttons, check boxes etc… in the html, but coded within the ui.R file

Those inputs are passed to the server.R file which takes them as elements of a list.

https://shiny.rstudio.com/images/shiny-cheatsheet.pdf

Shiny Apps

Part1: The webpage (UI)

Part2: The computer running R (Server)

Part3: test it locally on your own (or lab) machine using: shinyAppDir(“directory/name/goes/here”)

Part4 (save this step for later): Upload the app to a server and it will run automatically when a user visits the page

ui.R

This file sets inputs and makes use of outputs from the html page.

Inputs are passed to the server file

Outputs are retrieved from the server file

server.R

Takes inputs from ui.R and produces outputs

Please stick with common libraries, preferably those in a tagged topic:

Library to install all packages tagged by a topic

library(ctv)

for example:

install.views(“Graphics”)

See topics here: https://cran.r-project.org/web/views/

Folder

Must contain only one ui.R and server.R, but can contain any other data or R files

install and load libraries locally into that folder <— this will simplify things when using a different machine

```
install.packages("pkgname",lib=getwd())

library(pkgname,lib.loc=getwd())
```

Trouble Shooting

Make few small changes at a time and try running your App.

Make an input list and test the server.R file running it line by line

ShinyApp examples

More trouble shooting

Easy way to get started:

Sort out the ui.R ignoring the server.R file.

Then slowly update the server.R

The app will run with a broken server, but this will isolate the problems

Clarify

When working on server.R

start with:

input=list()

then define the input list elements and walk through the server code to make sure that it'll run.