- Please go to the following link to access these slides: **https://github.com/rladies-chicago/2019-04-18-shiny-workshop**

Data Viz with Shiny

You're ready to go if...

- You have the following packages installed on your local machine:

  `sf, shiny, leaflet, spData, dplyr`

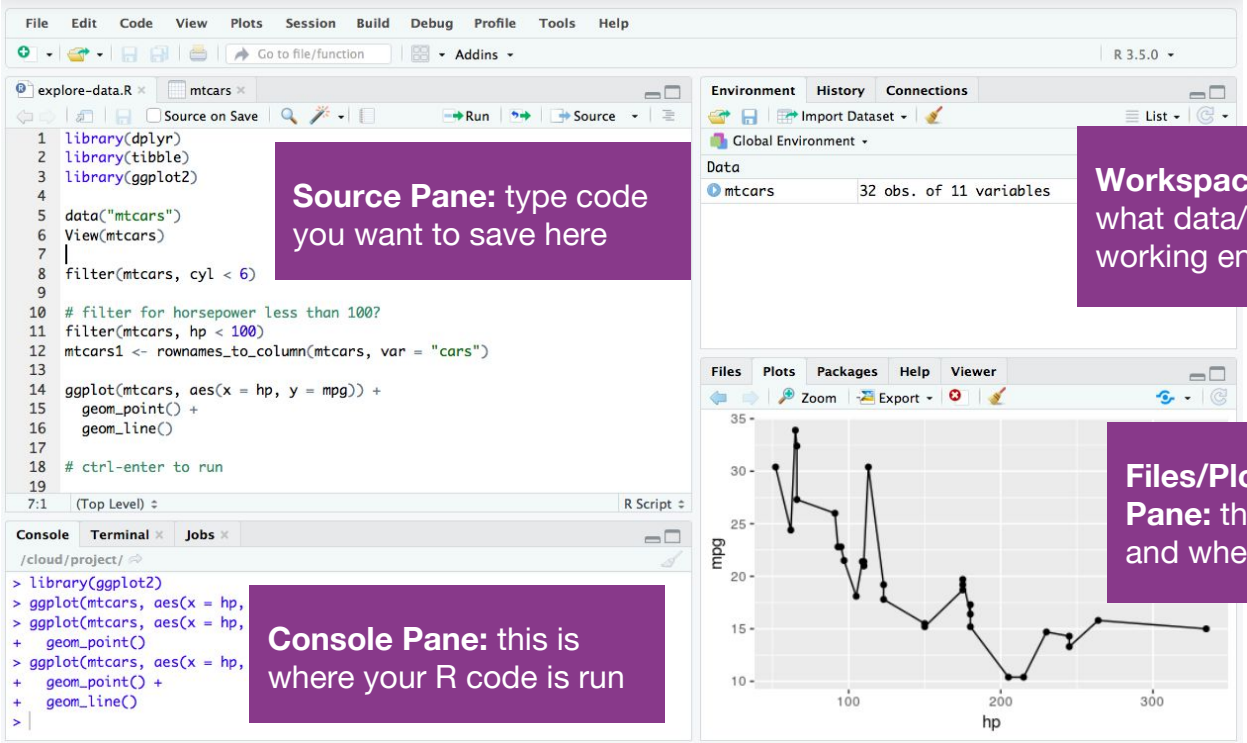
- Or if you've created a RStudio Cloud account and installed the above packages in a new project (let us know if you need help with this)

Learning goals:

- Work with spatial data structures in R
- Make an interactive map with leaflet
- Make a Shiny app
- Deploy Shiny app (if time)

Work with spatial data

A tour of RStudio



Source Pane: type code you want to save here

Workspace Pane: this tells you what data/objects are in your working environment

Console Pane: this is where your R code is run

Files/Plots/Packages/Help Pane: this is where plots display, and where you can get help

Load spatial packages

- To work with spatial data today, we need to load a few packages:

```
library(sf)
library(spData)
library(leaflet)
```

- And two others we'll need later:

```
library(dplyr)
library(shiny)
```

Load data

- We are using a dataset called `urban_agglomerations`
  - Found in the spData package!
- Load and view the data:

```
data("urban_agglomerations")
View(urban_agglomerations)
head(urban_agglomerations)
```

Load data

```
> head(urban_agglomerations)
  index year rank_order country_code          country_or_area city_code  urban_agglomeration note
1     1 1950          1          840 United States of America     23083      New York-Newark    …
2     2 1950          2          392                    Japan     21671                Tokyo    1
3     3 1950          3          826           United Kingdom     22860               London    2
4     4 1950          4          392                    Japan    206459 Kinki M.M.A. (Osaka)    3
5     5 1950          5          250                   France     20985                Paris    …
6     6 1950          6          643       Russian Federation     22299      Moskva (Moscow)    …
  population_millions            geometry
1           12.338471 -74.00366, 40.71704
2           11.274641   139.6917, 35.6895
3            8.360847  -0.12574, 51.50853
4            7.005284 135.55382, 34.67583
5            6.283018   2.34880, 48.85341
6            5.356392  37.62185, 55.75500
```

Explore data

- Try using the following commands to inspect the data:

```
dim(urban_agglomerations)
names(__)
str(__)
summary(__)
class(__)
```

Filter data by criteria

- Filter for city populations by year:

```
filter(urban_agglomerations, year == 2015)
```

```
> filter(urban_agglomerations, year == 2015)
# A tibble: 30 x 10
   index  year rank_order country_code country_or_area city_code urban_agglomera… note
   <dbl> <dbl>      <dbl>        <dbl> <chr>               <dbl> <chr>            <chr>
 1   391  2015          1          392 Japan               21671 Tokyo            1
 2   392  2015          2          356 India               21228 Delhi            17
 3   393  2015          3          156 China               20656 Shanghai         5
 4   394  2015          4           76 Brazil              20287 São Paulo        …
 5   395  2015          5          356 India               21206 Mumbai (Bombay)  …
 6   396  2015          6          484 Mexico              21853 Ciudad de Méxic… 6
 7   397  2015          7          156 China               20464 Beijing          12
 8   398  2015          8          392 Japan              206459 Kinki M.M.A. (O… 3
 9   399  2015          9          818 Egypt               22812 Al-Qahirah (Cai… 7
10   400  2015         10          840 United States …     23083 New York-Newark  …
# … with 20 more rows, and 2 more variables: population_millions <dbl>, geometry <list>
```

Save filtered data as an object

- Call it urban_2015!

```
urban_2015 <- filter(urban_agglomerations,
                     year == 2015)
```

Make an interactive map

Make a map!

- Use the leaflet package to create a simple interactive map:

```
library(leaflet)

leaflet(data = urban_agglomerations) %>%
    addTiles() %>%
    addMarkers()
```

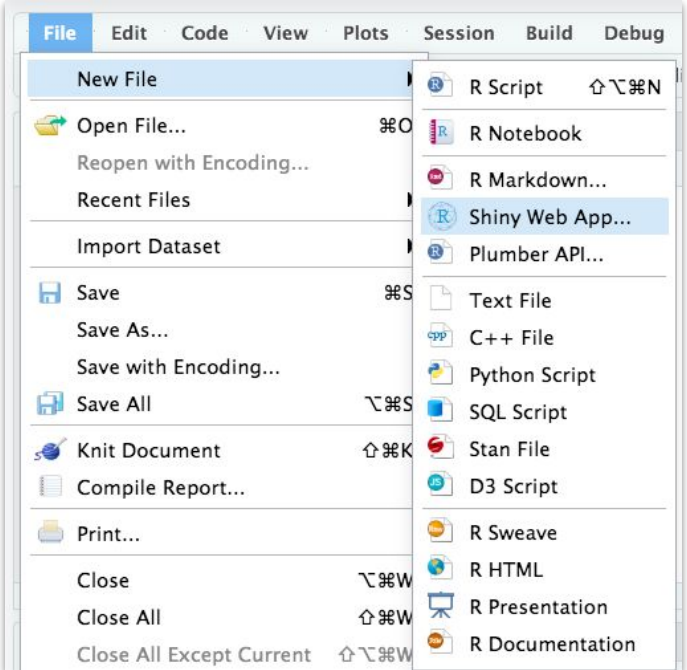
Challenge 1

1. Make an interactive map of urban_15!
2. Make an interactive map of the cities in 2010. (Call it urban_10)

Make a Shiny app

Make a Shiny app!

- Go to File > New File > Shiny Web App



Challenge

- Create a new Shiny app in RStudio. Run the app. Which parts of the Shiny UI code map to the app? How are `ui` and `server` linked (what are the features that are the same across both?)

- Change the title of the app.

Challenge

- In the UI object, add a `leafletOutput("map")` call in the `mainPanel()` function. Then, in the server object, add a `output$map <- renderLeaflet({})` call.

```
mainPanel(
   # Put one line of code here
)
```

```
output$map <- renderLeaflet({
  # Put three lines of leaflet code here




})
```

Challenge

- In the UI object, add a `sliderInput` of `"year"`. Change the step size to 5, and remove the comma for thousands (hint: do `?sliderInput` to look at the documentation, and options).

Challenge

- Create a new variable called `pop_per_year` that is a subset of city by year, depending on which year you enter (`input$year`). Use the `filter()` command in the `dplyr` package.

Challenge

- Try resizing the marker size depending on population, adding a popup, or doing more to customize your map!
- Try adding a feature in your app so that you only show cities over a certain population in millions (specified by the user), using `numericInput()` instead of `sliderInput()`

Challenge

- Add a data table element with `renderDataTable()` and `dataTableOutput()` so you can see the attributes of the points in the map.