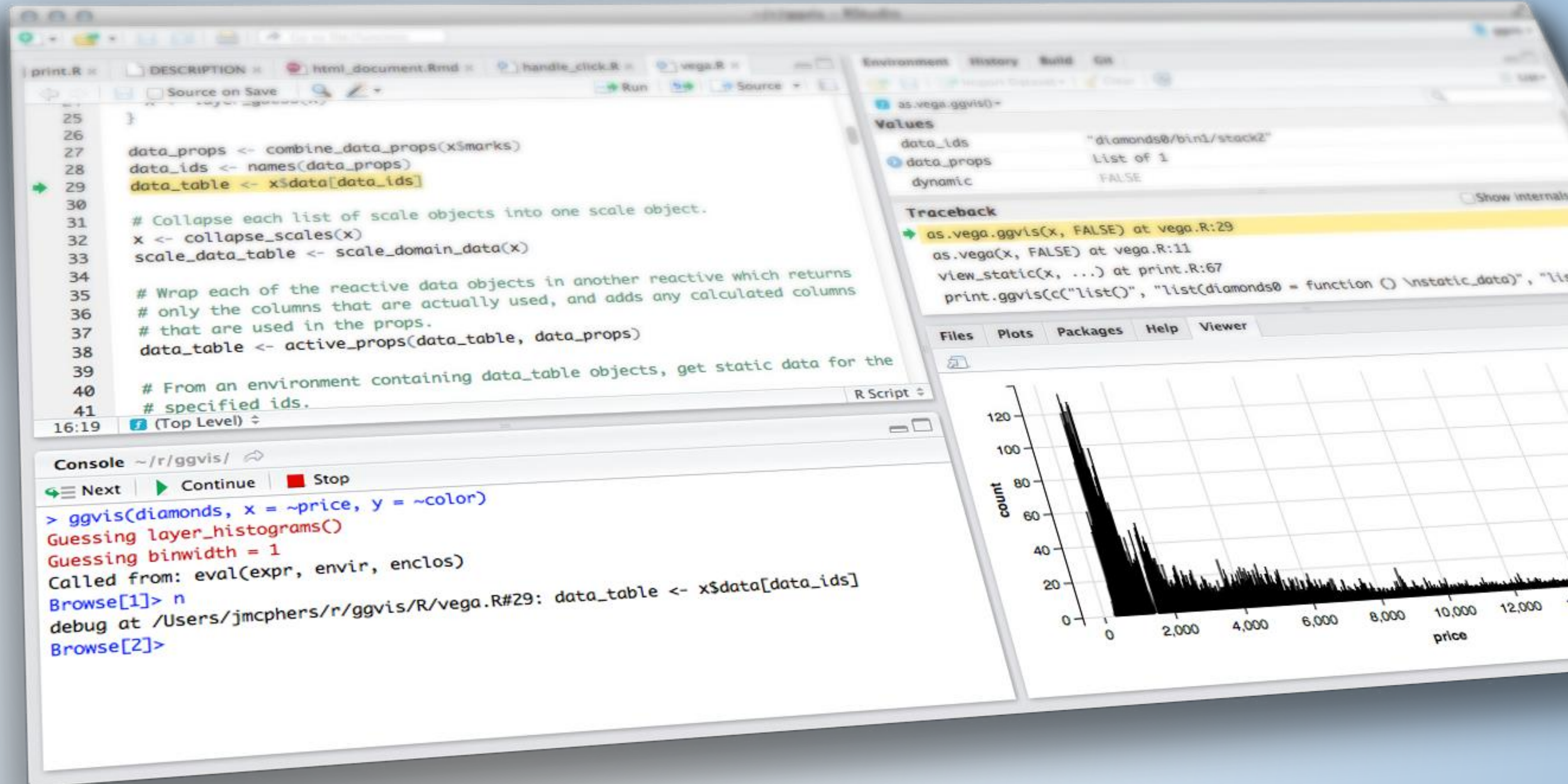


# MAPS



# OUTLINE

- ▶ Motivation
- ▶ sf
- ▶ ggmap
- ▶ Leaflet
  - ▶ Basemaps
  - ▶ Lines
  - ▶ Points
  - ▶ Shapes
  - ▶ Legends & Colors
- ▶ leaflet.extras
  - ▶ Heatmaps

# Motivation





Map of the 1854 Cholera outbreak - John Snow



*“‘I have the same problem’ is a famous  
**last** post in many forum-threads on the  
esri forum.”*

—[sebastian](#)



**NOT SURE IF ARCMAP IS  
LOADING**

**OR IF IT CRASHED**



*“R and Leaflet are free and open-source, ArcMap is very much not.”*

—Geoffrey Arnold



# PACKAGES

- ▶ Install these packages:

- ▶ rgdal:

- <https://github.com/cran/rgdal/blob/master/inst/README>

- ▶ rgeos

- ▶ sf

- ▶ leaflet

- ▶ leaflet.extras



# Spatial Data

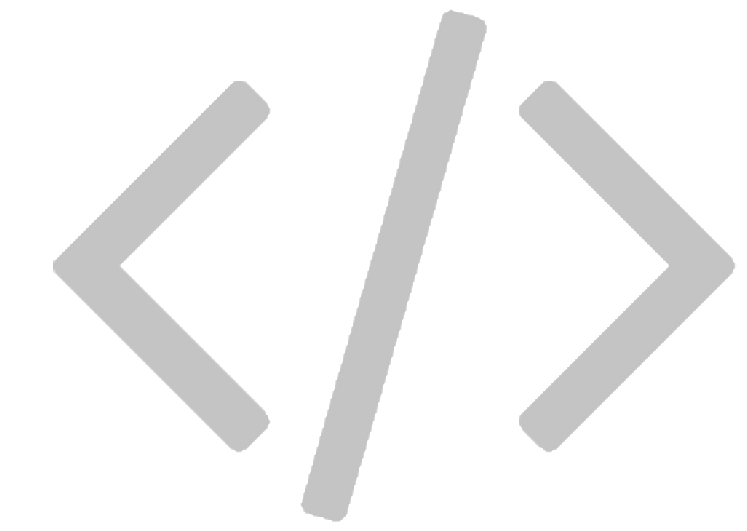
with sf



# LOADING SPATIAL DATA

- ▶ Sometimes your source data will be a data frame which you will need to join to spatial data, other times it will be included
- ▶ It may also be a CSV with coordinates, in those instances no further cleaning needs to take place





# DEMO

examples.Rmd

Go to loading code chunk



# IMPORTING DATA

```
polls <- st_read("Allegheny_County_Polling_Place_Locations_November_2016.geojson")  
  
cds.load <- st_read("./cb_2015_us_cd114_500k/cb_2015_us_cd114_500k.shp")  
plot(cds.load)
```

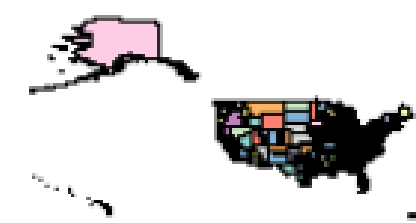


# BASIC PLOT

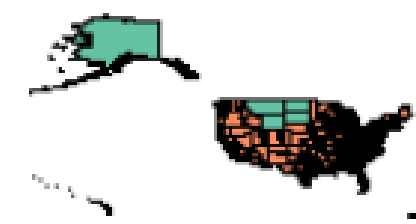
```
plot(cds.load)
```



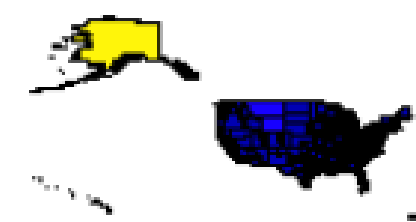
STATEFP



AFFGEOID



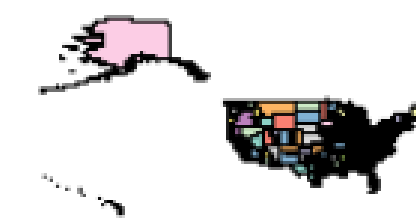
LSAD



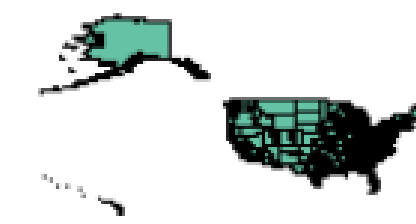
ALAND



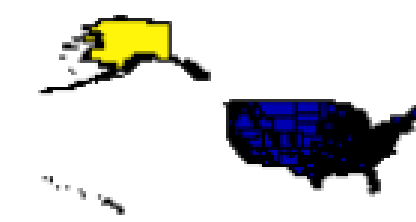
CD114FP



GEOID



CDSSESN

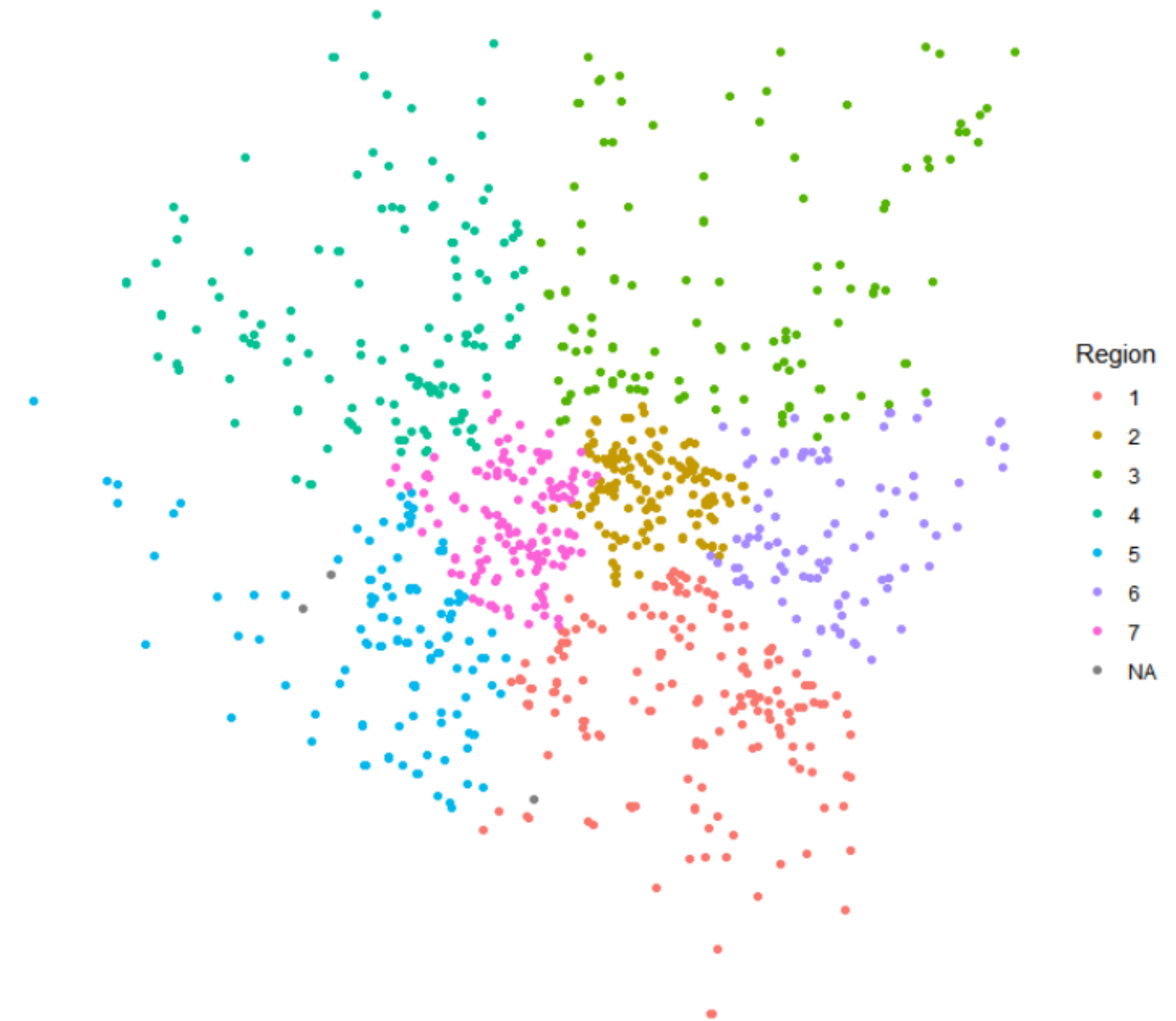


AWATER



# BASIC GGPLOT

```
ggplot(data=polls, aes(color = Region)) +  
  geom_sf() +  
  theme_void()
```





# ADDING A BASEMAP

```
# Get the boundaries of your file to get the basemap
```

```
bbox <- st_bbox(polls)
```

```
bbox_trans <- c(left = bbox[[1]], bottom = bbox[[2]], right = bbox[[3]], top = bbox[[4]])
```

```
# Get the basemap
```

```
map <- get_stamenmap(bbox_trans, maptype = "toner-lite")
```

```
map %>% ggmap() +  
  coord_sf(crs = st_crs(3857)) +  
  geom_sf(data=polls, aes(color = Region), inherit.aes = FALSE) +  
  theme_map()
```







# LEAFLET



# WHAT IS LEAFLET?

- ▶ R version of the Javascript API of Leaflet
- ▶ One of the ways to get interactive maps
  - ▶ Others include: tmap, mapview and plotly
- ▶ Documentation: <https://rstudio.github.io/leaflet/>

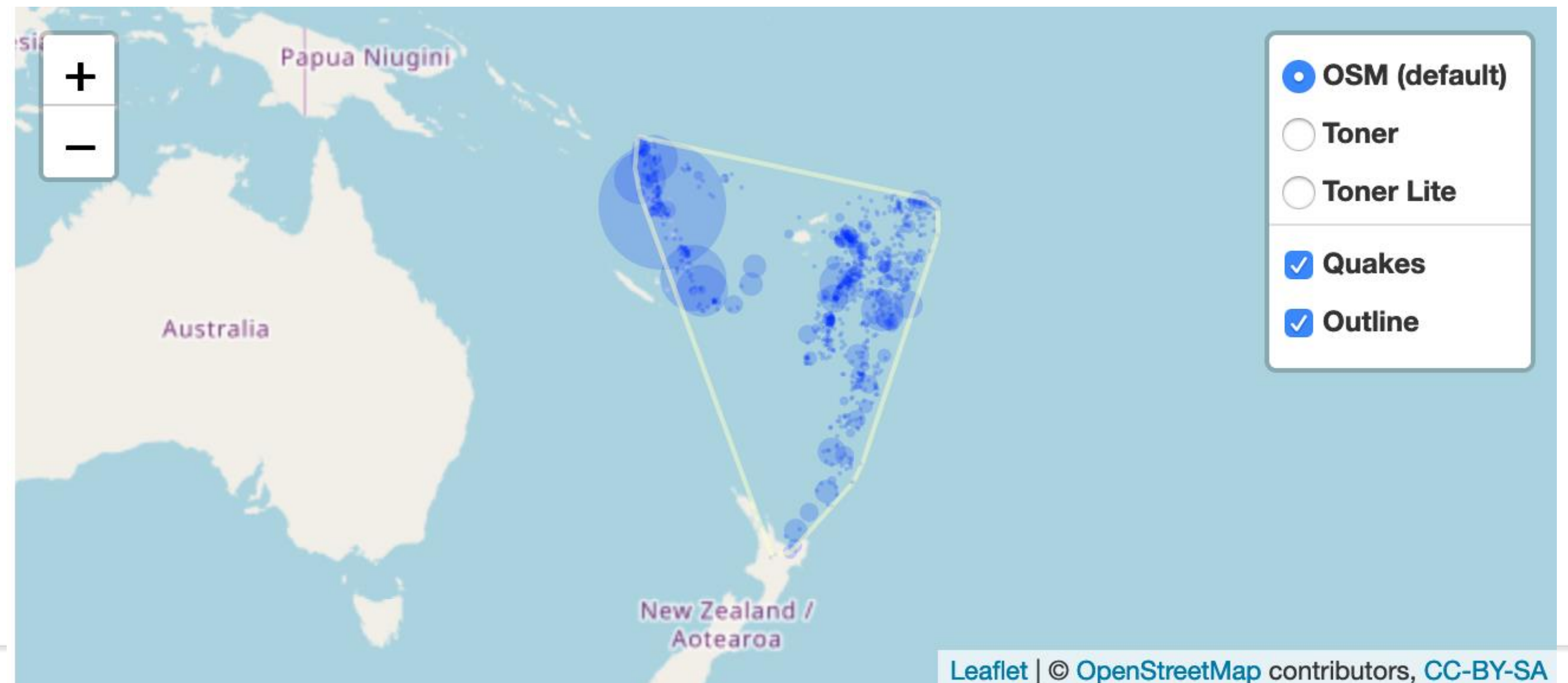


# GROUP CONTROLS

In leaflet you can add groups and give them a name.

Group names are what show up in the layer control

```
addProviderTiles(providers$Stamen.Toner, group = "Toner") %>%  
addLayersControl(  
  baseGroups = c("OSM (default)", "Toner", "Toner Lite"),  
  overlayGroups = c("Quakes", "Outline"),  
  options = layersControlOptions(collapsed = FALSE)  
)
```



# EXERCISE



- ▶ Create a map with layer controls
  - ▶ Get basemap provider names from here: <https://leaflet-extras.github.io/leaflet-providers/preview/>
  - ▶ Create 3 named basemap groups
  - ▶ Add a layer control that has the named base groups

**5<sub>m</sub> 00<sub>s</sub>**



# Shapes

# TYPICAL ARGUMENTS

- ▶ lng (if a column)
- ▶ lat (if a column)
- ▶ layerId
- ▶ group (for layerControls)
- ▶ stroke (boolean, shape outline)
- ▶ color (outline color, hex values)
- ▶ weight (outline width)
- ▶ opacity (alpha of the line)
- ▶ fill (boolean)
- ▶ fillColor (hex color)
- ▶ fillOpacity (alpha of fill)
- ▶ And more!



# EXERCISE



- ▶ Using the example cds to create a polygon layer
  - ▶ Make sure to include a basemap

**5<sub>m</sub> 00<sub>s</sub>**



# SOLUTION

```
leaflet(data = cds) %>%  
  addProviderTiles("Stamen.Toner") %>%  
  addPolygons()
```



# PALETTES

- ▶ palette (color brewer palette)
- ▶ domain (values to be colored)
- ▶ na.color
- ▶ alpha
- ▶ reverse (values of palette)
- ▶ bins
- ▶ pretty
- ▶ right (for cutting)
- ▶ n (number of quantities)
- ▶ probs
- ▶ levels
- ▶ ordered

# LEGENDS

- ▶ position (“bottomright” etc...)
- ▶ pal (palette from colorBrewer)
- ▶ values (domain from data)
- ▶ na.labels
- ▶ bins (buckets)
- ▶ colors
- ▶ labFormat (separate function  
labelFormat)
- ▶ title
- ▶ className (for custom CSS to apply)
- ▶ layerId (for input usage)
- ▶ group (for layerControls)
- ▶ digits
- ▶ big.mark
- ▶ transform (function to be applied to  
labels)



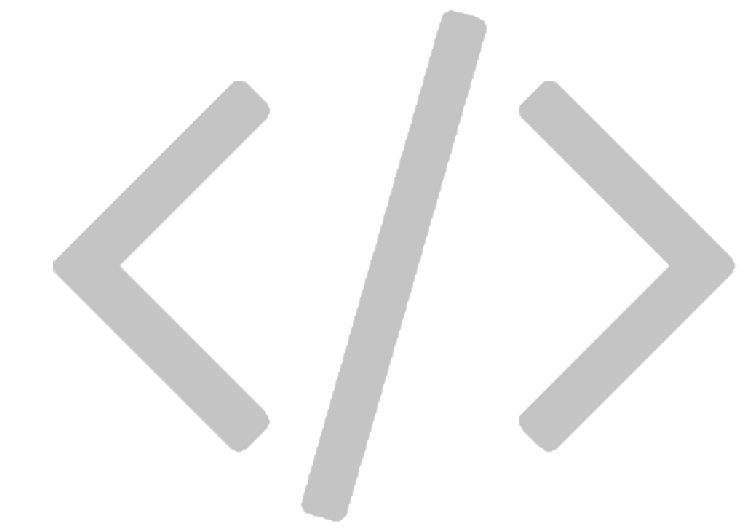
# EXERCISE



- ▶ Choropleth Polygon map with
  - ▶ Create a color palette for Life Expectancy at Birth (years) column in the merged Congressional District data.
    - ▶ addPolygons to a leaflet map and apply palette to column
      - ▶ Stretch goal: Add a legend

5<sub>m</sub> 00<sub>s</sub>

Hint: <https://rstudio.github.io/leaflet/choropleths.html>



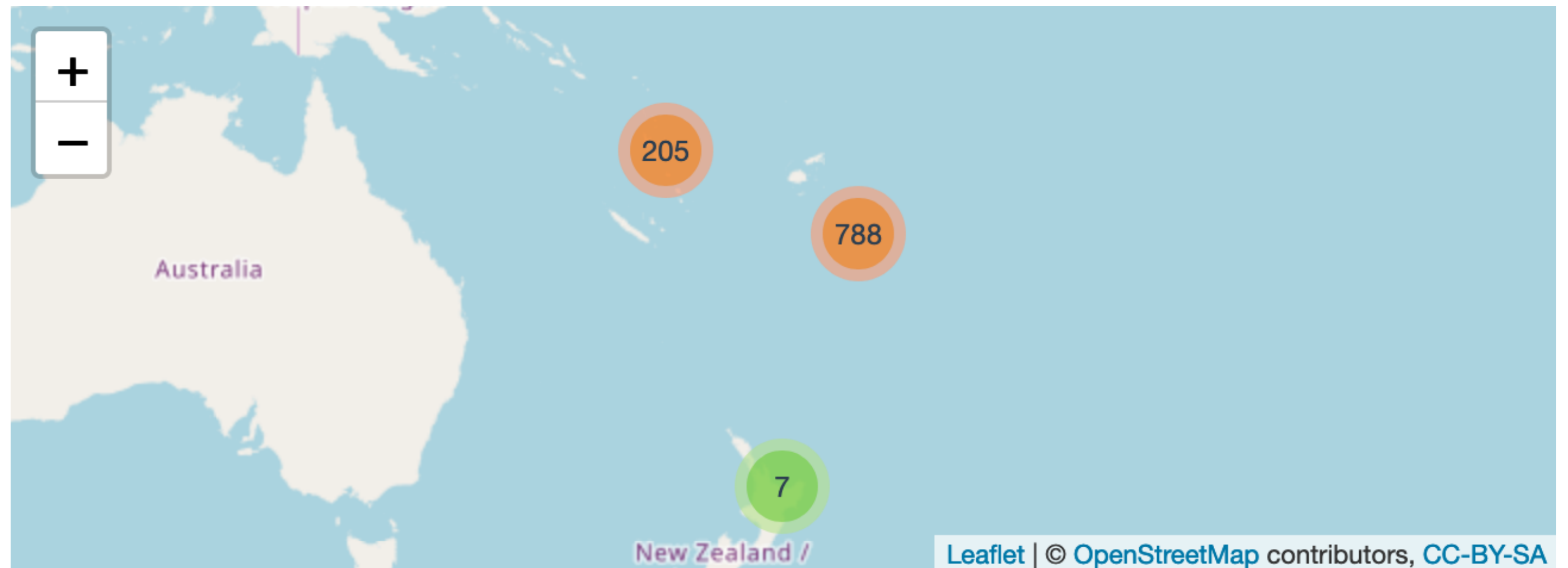
# DEMO

08-Maps-Exercises.Rmd  
Go to polygon code chunk



# CLUSTERS

```
leaflet(quakes) %>%  
  addTiles() %>%  
  addMarkers(  
    clusterOptions = markerClusterOptions()  
  )
```



# WHY CLUSTERS?

- ▶ Sometimes there's just a ton of data to show and trying to visualize them will bring any browser window it a crawl.
- ▶ Example:  
<https://pittsburghpa.shinyapps.io/TreesNAt/>



# EXERCISE



- ▶ Go to 08-Maps-Exercises.Rmd clusters code chunk
  - ▶ Use 311 data to create a cluster map
    - ▶ Create a factor palette
      - ▶ Create a legend
        - ▶ Make sure to include a basemap

**5<sub>m</sub> 00<sub>s</sub>**

Hint: <https://rstudio.github.io/leaflet/markers.html>

leaflet

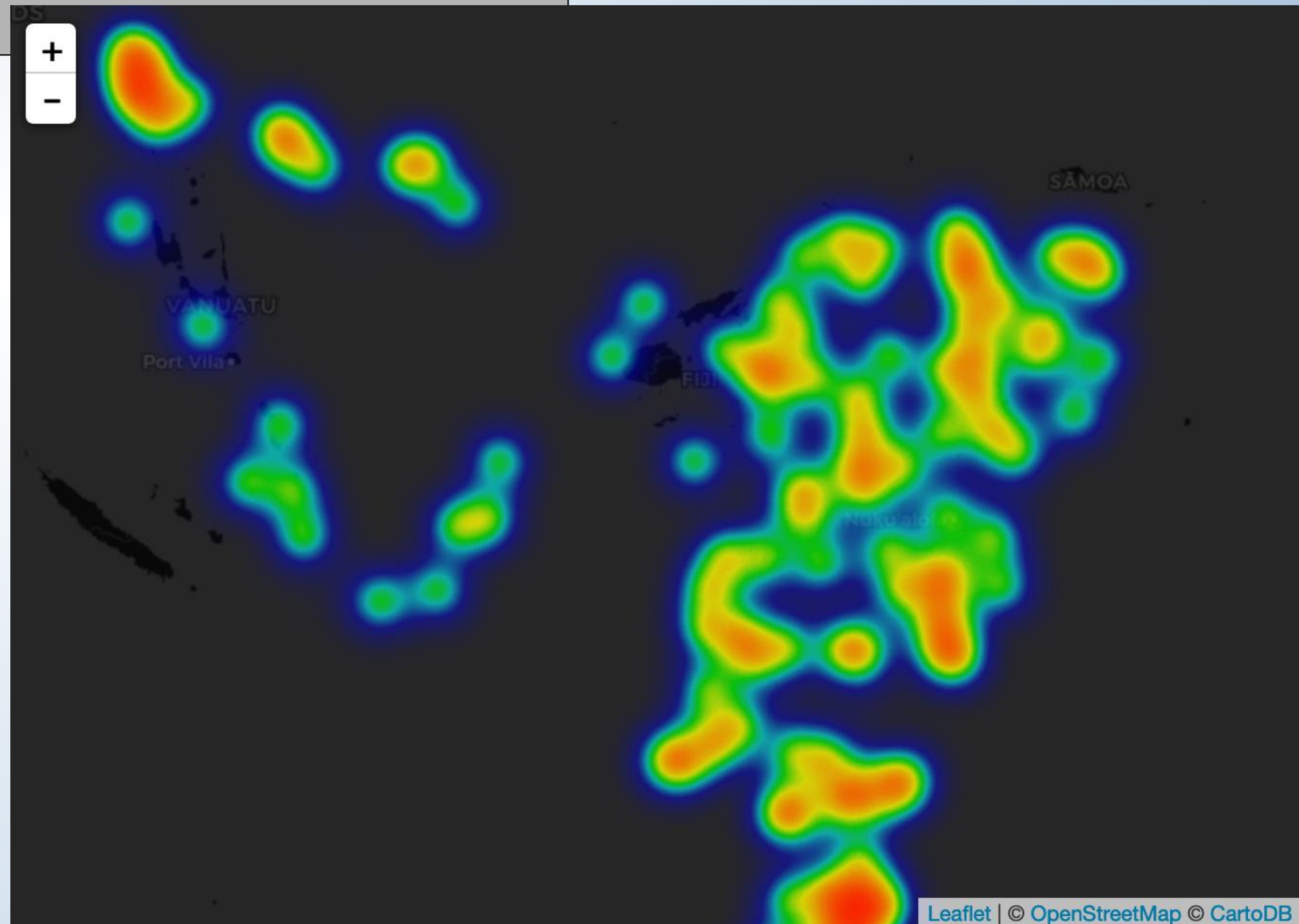
.extras



# LEAFLET.EXTRAS

- ▶ Adds some pretty nice functions to complement leaflet
  - ▶ Tile caching: <http://rpubs.com/bhaskarvk/TileLayer-Caching>
  - ▶ weather icons: <http://rpubs.com/bhaskarvk/leaflet-weather>
  - ▶ Pulse icons: <http://rpubs.com/bhaskarvk/leaflet-pulseicon>
  - ▶ and heat maps!

```
leaflet(quakes) %>%  
addProviderTiles(providers$CartoDB.DarkMatter) %>%  
  setView( 178, -20, 5 ) %>%  
  addHeatmap(lng = ~long, lat = ~lat, intensity = ~mag,  
    blur = 20, max = 0.05, radius = 15)
```



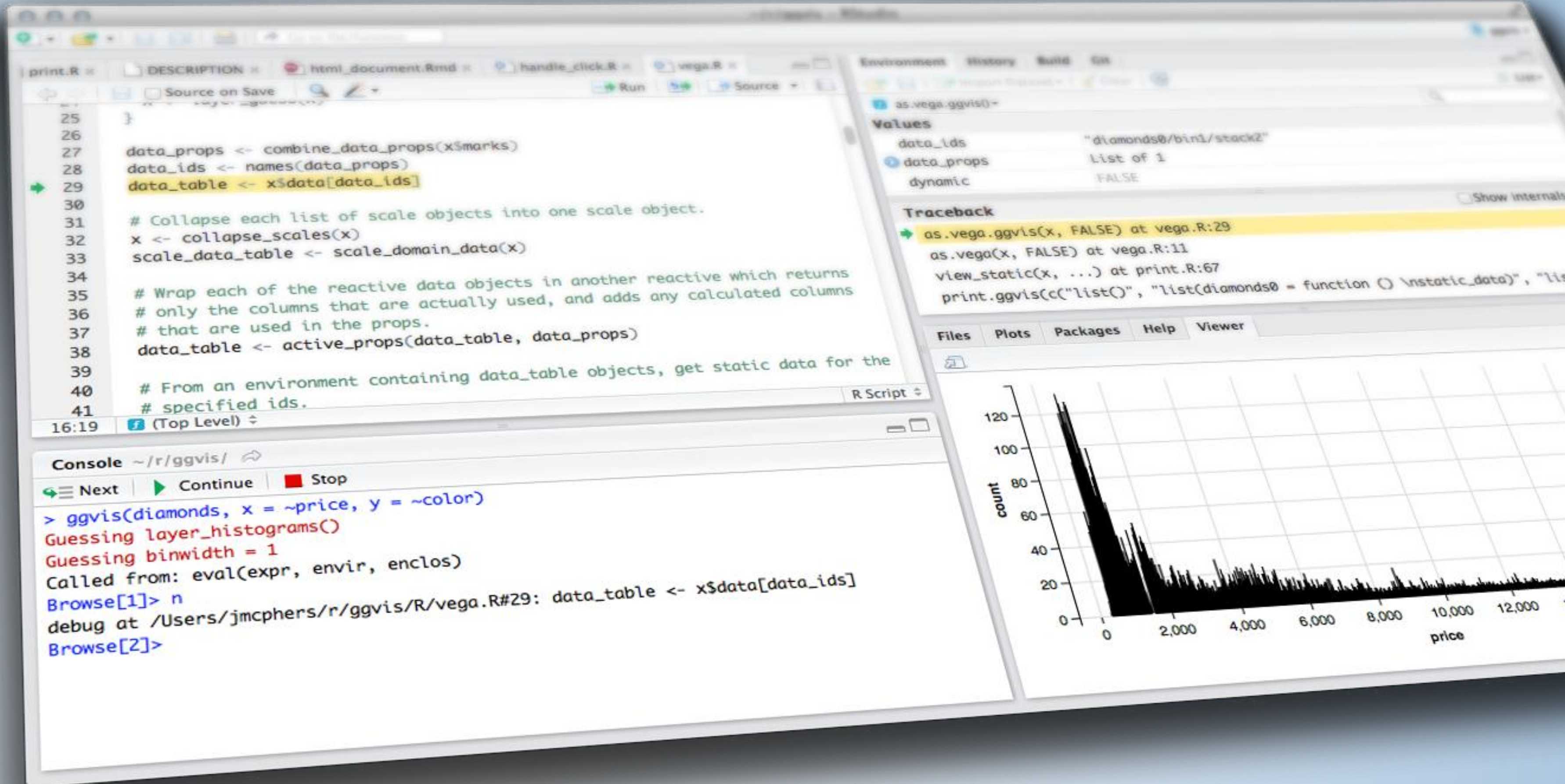


# EXERCISE



- ▶ Using the pothole data create a heat map
  - ▶ This won't be very different from your cluster map from before.
    - ▶ Play around with the arguments for heat maps and find a radius that looks good.
- ▶ Compare your outcome with your neighbor

**5<sub>m</sub> 00<sub>s</sub>**



LEAFLET