STA141b HW5

Kwasie Agbemadon

12/7/2020

Packages used: XML, httr, and RJSONIO

This is my interactive COVID data visualization, where I will make an interactive plot using ggplot and plotly. To start I will load my packages, grab the website and the JSON file.

```
library(XML)

## Warning: package 'XML' was built under R version 4.0.3

library(httr)

## Warning: package 'httr' was built under R version 4.0.3

library(RJSONIO)

## Warning: package 'RJSONIO' was built under R version 4.0.3

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.0.3

library(plotly)

## Warning: package 'plotly' was built under R version 4.0.3

##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:httr':
##
##     config

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout

library(htmlwidgets)
```

```
## Warning: package 'htmlwidgets' was built under R version 4.0.3
```

```
library(htmltools)

covidny =
htmlParse(GET('https://www.nytimes.com/interactive/2020/us/coronavirus-us-
cases.html?action=click&module=Top%20Stories&pgtype=Homepage'))
node = xpathSApply(covidny, "//*[contains(./text(), 'USA.json')]", xmlValue)
link = gsub('.*"(https://[^"]+USA.json)".*', "\\1", node)
link = unique(link)
us = fromJSON(link)
```

Now that we have obtained the COVID data from the NYT site, let's build our dataset. I did this by making a vector of each date using the ranges in the 'us' list. Then I made empty vectors for the important values. I put the counties, cases, and deaths in a for loop so the values from each county and each state from the list are added onto the vectors. Then I made new vectors to replicate everything. Those vectors were thrown into a dataset.

Additionally, I made vectors for the cases and deaths that appeared on each day and the capita for each cumulative case/death, whihc are also on the dataset as well.

```
dates = seq(as.Date(us$range[1]), as.Date(us$range[2]), by = 'day')
counties = c()
cases = c()
deaths = c()
population = c()

for (i in 1:length(us$data)) {
  g = match('county', us$data[[i]]$region_type) && 'USA-06' %in%
us$data[[i]]$hierarchy
  counties = append(counties, us$data[[i]]$display_name[g])
  population = append(population, us$data[[i]]$population[g])
  cases = append(cases, unlist(us$data[[i]]$cases[g]))
  deaths = append(deaths, unlist(us$data[[i]]$deaths[g]))
}

cases_per_day = c(0, diff(cases))
deaths_per_day = c(0, diff(deaths))

for (i in 1:length(cases_per_day)) {
  if (cases_per_day[i] < 0)
    cases_per_day[i] = 0
  if (deaths_per_day[i] < 0)
    deaths_per_day[i] = 0
}

counties = counties[-which(counties=='Unknown')]
county = c()
date = c()
date = rep(dates, length(counties))
```

```
county = rep(counties, each = length(dates))
pop = rep(population, each = length(dates))
cases_per_capita = formatC(c(cases/pop), digits = 4, format = 'f')
deaths_per_capita = formatC(c(deaths/pop), digits = 4, format = 'f')

covid = data.frame(date, county, cases, cases_per_day, deaths,
deaths_per_day, cases_per_capita, deaths_per_capita)
```

We can now get to plotting. I concatenated a string of vital information to use in my tooltip. It will contain everything that is in the dataset. I then used ggplot to plug everything in, and renamed the axes to make it boujee. I used ggplotly to plug in the plot and add in the tooltip, then saved it as a widget.

```
info = paste0('Date: ', date, '\n County: ', county, '\n Total Cases: ',
cases, '\n Total Deaths: ', deaths, '\n New Cases: ', cases_per_day, '\n New
Deaths: ', deaths_per_day, '\n Cases per Capita: ', cases_per_capita, '\n
Deaths per Capita: ', deaths_per_capita)

p = covid %>% highlight_key(~county) %>% ggplot(aes(date, cases, group =
county, colour = county, info = info)) + geom_line() + geom_point() +
ggtitle('Plot of COVID Cases for each County') + xlab('Date') + ylab('No. of
Cases') + labs(color = 'County')
plot = ggplotly(p, width = 700, height = 500, tooltip = 'info') %>%
highlight(on = 'plotly_hover', off = 'plotly_doubleclick')

## Warning: `group_by_()` is deprecated as of dplyr 0.7.0.
## Please use `group_by()` instead.
## See vignette('programming') for more help
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.

plot

saveWidget(plot, 'plot.html', selfcontained = F)
```

Here, I am using the htmltools package to create the HTML file of the plot. It contains text, a button called 'Hide Lines' which turns off all of the plots for each county (using HideLines.js), and the plot that was saved as a widget.

We create a new widget which combiness all of these aspects into a full HTML page.

```
h1 = HTML('<h1>HW5 COVID Visualization for STA 141B</h1>
<p>This is the data visualization for Homerowk 5 for STA 141B.</p>

<p>It contains a plot of all of the cases and deaths that occured on each
county in California, as well as a visual map of the cases overtime, from
3/1/20 to 12/15/20.</p>

<p>This plot shows the culumative amount of cases that occured on each day.
Hover over the lines to see the number of cases, deaths, new cases, new
deaths, and capita information</p>
<p>Click on a county name to select/deselect the plot for that county. To
disable all of the counties, wither double click on any county or click on
the "Hide Lines" button</p>

<p><button onclick="hideLines()">Hide Lines</button></p>')
hl = tags$script(src = 'HideLines.js')
p = prependContent(plot, hl, h1)

saveWidget(p, 'visualization.html', selfcontained = F)
```

We can now talk about the results. It turns out that LA county has the most cases and deaths out of all of the counties due to its population size. The second county with the highest cases/deaths is San Bernardino with 129k cases and 1.2k deaths. Every other county has low numbers. Some other counties had extremely little cases, like Alpine and Amador County. This is information going up to December 15.

You can play with the plot in the html file as well. Open visualization.html.