MSCS 264: Homework #13

Due Tues Nov 20 at 11:59 PM

You should submit a knitted pdf file on Moodle, but be sure to show all of your R code, in addition to your output, plots, and written responses.

Web scraping

1. Read in the table of data found at https://en.wikipedia.org/wiki/List_of_United_States_cities_by_crime_rate and create a plot showing violent crime rate (total violent crime) vs. property crime rate (total property crime). Identify outlier cities (those with "extreme" values for `VCrate` and/or `PCrate`) by feeding a data set of outliers into `geom_label_repel()`.

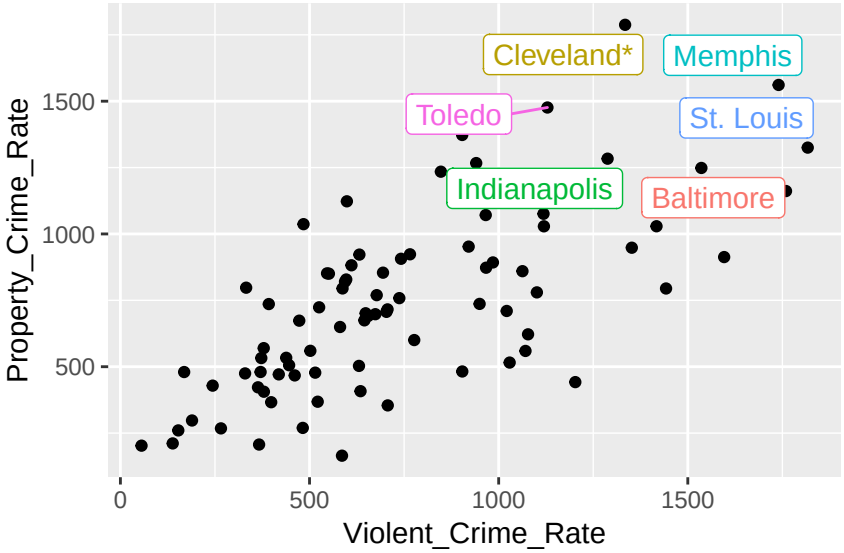
Hints:

- after reading in the table using `html_table()`, create a data frame with just the columns you want, using a command such as: `crimes3 <- as.data.frame(crimes2)[,c(LIST OF COLUMN NUMBERS)]`. Otherwise, R gets confused since it appears as if several columns all have the same column name.
- then, turn `crimes3` into a tibble with `as.tibble(crimes3)` and do necessary tidying: get rid of unneeded rows, parse columns into proper format, etc.

```
crime <- read_html("https://en.wikipedia.org/wiki/List_of_United_States_cities_by_crime_rate")
crimetable <- html_nodes(crime, css = "table")
crimedata <- html_table(crimetable, header = TRUE, fill = TRUE)[[2]]
crimedata1 <- as.data.frame(crimedata)[,c(1,2,4,10)]
crimedata2 <- as.tibble(crimedata1)

crimedatatidy <- crimedata2 %>%
  rename(`Violent_Crime` = "Violent Crime",
         `Property_Crime` = "Property Crime") %>%
  mutate(Violent_Crime_Rate = parse_double(Violent_Crime),
         Property_Crime_Rate = parse_double(Property_Crime)) %>%
  select(State, City, Violent_Crime_Rate, Property_Crime_Rate)
```

```
## Warning: 1 parsing failure.
## row # A tibble: 1 x 4 col     row   col expected actual expected   <int> <int> <chr>    <chr>  actual

## Warning: 1 parsing failure.
## row # A tibble: 1 x 4 col     row   col expected actual expected   <int> <int> <chr>    <chr>  actual
```

```
crimedatatidy1 <- crimedatatidy[-c(1), ] %>%
  arrange(State)

outliercrime <- crimedatatidy1 %>%
  filter(Violent_Crime_Rate >= 1000, Property_Crime_Rate >= 1240)

ggplot(data = crimedatatidy1,aes(x = Violent_Crime_Rate, y = Property_Crime_Rate)) +
  geom_point() +
  ggrepel::geom_label_repel(aes(label = City, colour = City), data = outliercrime, show.legend = FALSE)
```



2. As we did in class, use the `rvest` package to pull off data from imdb's top grossing films released in 2017 at https://www.imdb.com/search/title?year=2017&title_type=feature&sort=boxoffice_gross_us,desc. Create a tibble that contains the title, gross, imdbscore, and metascore for the top 50 films. Then generate a scatterplot of one of the ratings vs. gross, labelling outliers as in Question 1 with the title of the movie.

```
myapikey <- "44dc8f2e"
top50 <- read_html("https://www.imdb.com/search/title?year=2017&title_type=feature&sort=boxoffice_gross

top50table <- tibble(Title = character(), Metascore = double(),
                     imdbRating = double(), BoxOffice = double())
```

3. 5 points if you push your Rmd file with HW13 solutions along with the knitted pdf file to your MSCS264-HW13 repository in your GitHub account. So that I can check, make your repository private (good practice when doing HW), but add me (username = proback) as a collaborator under Settings > Collaborators.

Factors

Read Chapter 15 on factors and attempt the following problems:

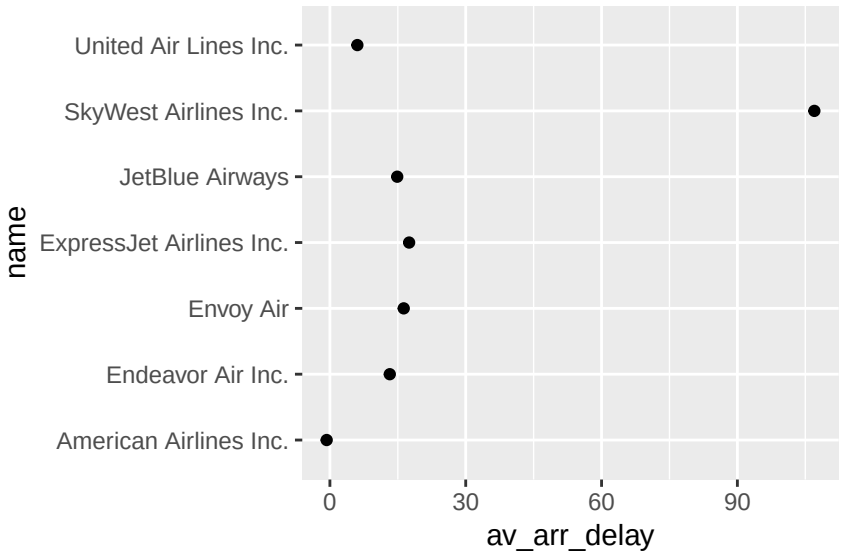
4. In the `nycflights13` data, just consider flights to O'Hare (dest=="ORD"), and summarize the mean arrival delay by carrier (actually use the entire name of the carrier after merging carrier names into `flights`). Then use `geom_point` to plot mean arrival delay vs. carrier - first without reordering carrier names, and second after reordering carrier names by mean arrival delay.

```
library(nycflights13)

oharedelay <- flights %>%
  filter(dest == "ORD") %>%
  left_join(airlines, by = "carrier") %>%
  group_by(name) %>%
  summarise(av_arr_delay = mean(arr_delay, na.rm = TRUE))

oharedelay
```

```
## # A tibble: 7 x 2
##   name                     av_arr_delay
```

```
##   <chr>                          <dbl>
## 1 American Airlines Inc.        -0.714
## 2 Endeavor Air Inc.             13.2
## 3 Envoy Air                     16.3
## 4 ExpressJet Airlines Inc.      17.5
## 5 JetBlue Airways               14.9
## 6 SkyWest Airlines Inc.        107
## 7 United Air Lines Inc.          6.07
```

```
ggplot(data = oharedelay, aes(x = av_arr_delay, y = name)) +
  geom_point()
```




```
oharedelay %>%
  mutate(name1 = fct_reorder(name, av_arr_delay)) %>%
  ggplot(aes(x = av_arr_delay, y = name1)) +
  geom_point()
```

5. Again considering only flights to O'Hare, create a new factor variable which differentiates national carriers (American and United) from regional carriers (all others which fly to O'Hare). Then create a

violin plot comparing arrival delays for all flights to O'Hare from those two groups (you might want to exclude arrival delays over a certain level).

```
flights %>%
   filter(dest == "ORD") %>%
  left_join(airlines, by = "carrier") %>%
  count(name)
```

```
## # A tibble: 7 x 2
##   name                          n
##   <chr>                     <int>
## 1 American Airlines Inc.     6059
## 2 Endeavor Air Inc.          1056
## 3 Envoy Air                  2276
## 4 ExpressJet Airlines Inc.      2
## 5 JetBlue Airways             905
## 6 SkyWest Airlines Inc.         1
## 7 United Air Lines Inc.      6984
```

```
oharedelay1 <- flights %>%
  filter(dest == "ORD") %>%
  left_join(airlines, by = "carrier") %>%
  mutate(name = fct_collapse(name,
                             nationalcarriers = c("American Airlines Inc.", "United Air Lines Inc."),
                             regionalcarriers = c("Endeavor Air Inc.", "Envoy Air", "ExpressJet Airlines
                                                  "JetBlue Airways", "SkyWest Airlines Inc."))) %>%
  select(name, arr_delay)

oharedelay2 <- oharedelay1 %>%
  filter(arr_delay <= 300)

ggplot(data = oharedelay2, aes(x = name, y = arr_delay)) +
  geom_violin()
```