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.

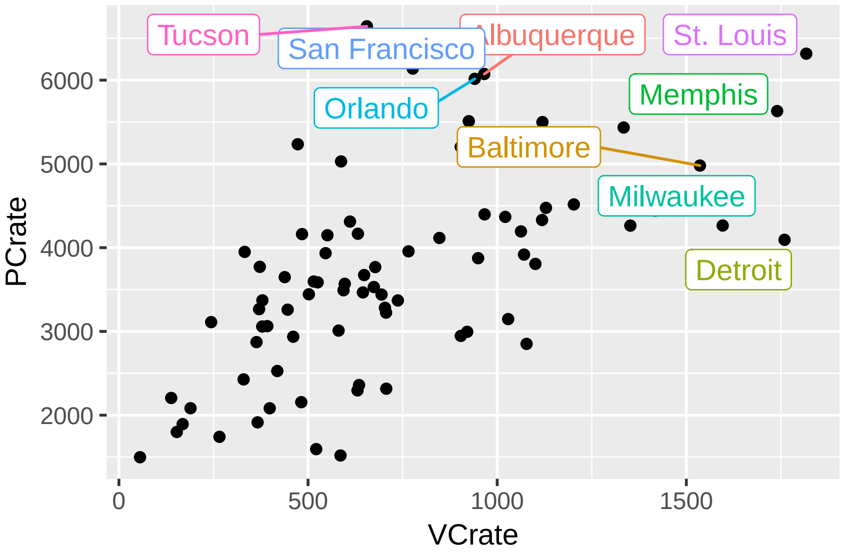
Web scraping

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.

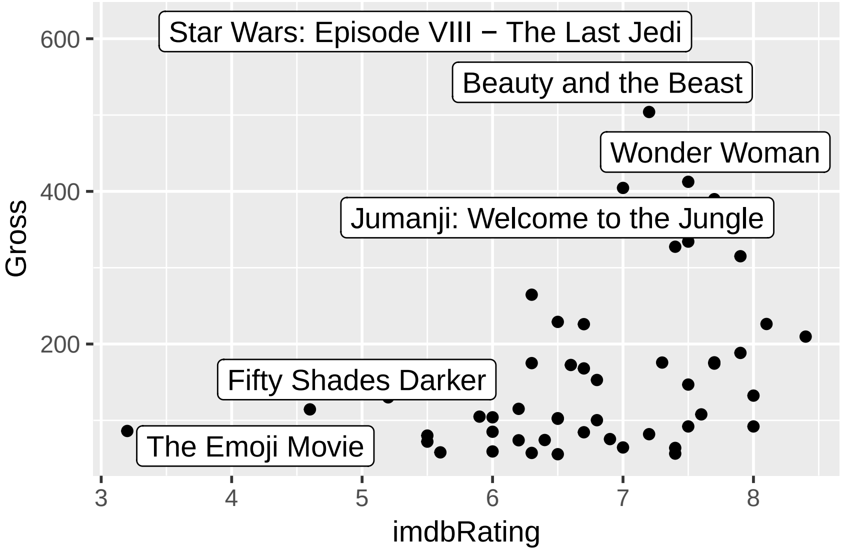
```r
wiki <- read_html("https://en.wikipedia.org/wiki/List_of_United_States_cities_by_crime_rate")
wiki1 <- html_nodes(wiki, css = "table")


wiki2 <- html_table(wiki1, header = TRUE, fill = TRUE)[[2]]

crimes <- as.data.frame(wiki2)[-c(1),c(1, 2, 4, 9)]

crimes1 <- as.tibble(crimes) %>%
  mutate("VCrate" = as.double(`Violent Crime`),
  "PCrate" = as.double(`Property Crime`))
  outliers <- crimes1 %>%
  filter(PCrate >= 6000 | VCrate >= 1500)

crimes1 %>%
  ggplot(aes(x=VCrate, y=PCrate))+
  geom_point()+
  ggrepel::geom_label_repel(aes(label = City, colour = City), data = outliers,
  show.legend = FALSE)
```



Test line for class

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.

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

imdbscore0 <- html_nodes(top50, ".ratings-imdb-rating strong")
imdbscore <- html_text(imdbscore0)

imdbtitle0 <- html_nodes(top50, ".lister-item-header a")
imdbtitle <- html_text(imdbtitle0)

imdbmetascore0 <- html_nodes(top50,".favorable")
imdbmetascore <- html_text(imdbmetascore0)

imdbgross0 <- html_nodes(top50, ".ghost~ .text-muted+ span")
imdbgross <- html_text(imdbgross0)

imdb <- tibble(Title = character(), Metascore = double(), imdbRating = double(), Gross = double())
for(i in 1:50){
imdb[i,1] <- imdbtitle[i]
imdb[i,2] <- parse_number(imdbmetascore)[i]
imdb[i,3] <- parse_number(imdbscore)[i]
imdb[i,4] <- parse_number(imdbgross)[i]
}

outliers <- imdb %>%
filter(imdbRating<5 | Gross >= 400)

ggplot(imdb, aes(imdbRating,Gross))+
  geom_point()+
  ggrepel::geom_label_repel(aes(label = Title), data = outliers,show.legend = FALSE)
```



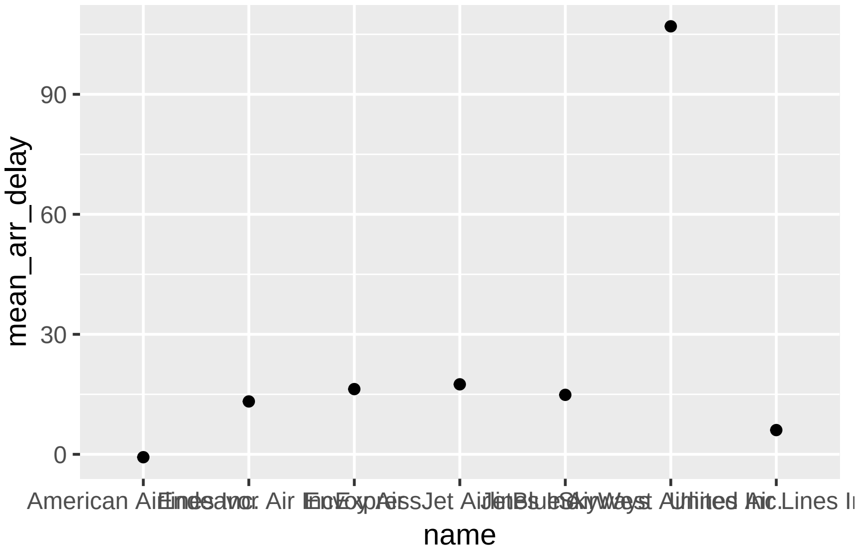
3. 5 points if you push your Rmd file with HW13 solutions along with the knitted pdf file to your MSCS264-HW13 repository in your GitHub account. So that I can check, make your repository private (good practice when doing HW), but add me (username = proback) as a collaborator under Settings > Collaborators.

Factors

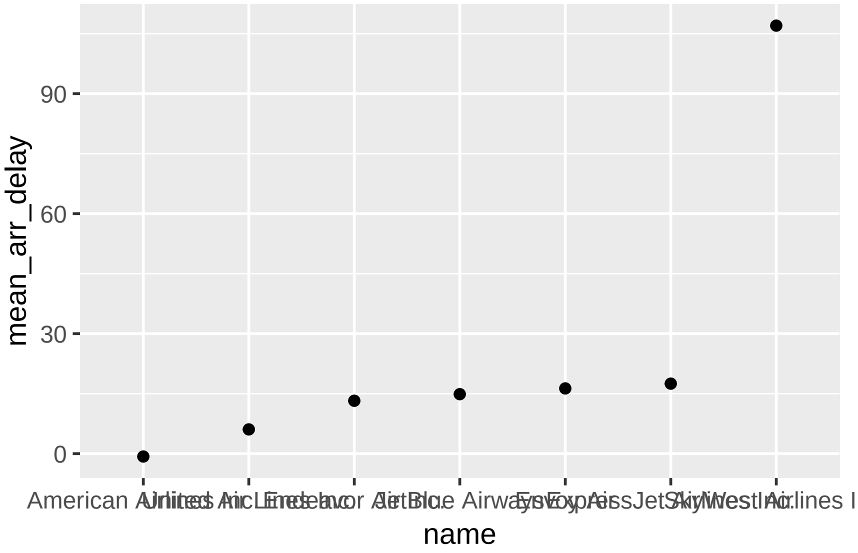
Read Chapter 15 on factors and attempt the following problems:

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.

```r
flights1 <- flights %>%
  full_join(airlines, by = "carrier")

ord <- filter(flights1, dest == "ORD")

ord %>%
  group_by(name)%>%
  summarise("mean_arr_delay" = mean(arr_delay, na.rm = TRUE)) %>%
  ggplot(aes(x=name, y=mean_arr_delay))+
    geom_point()
```

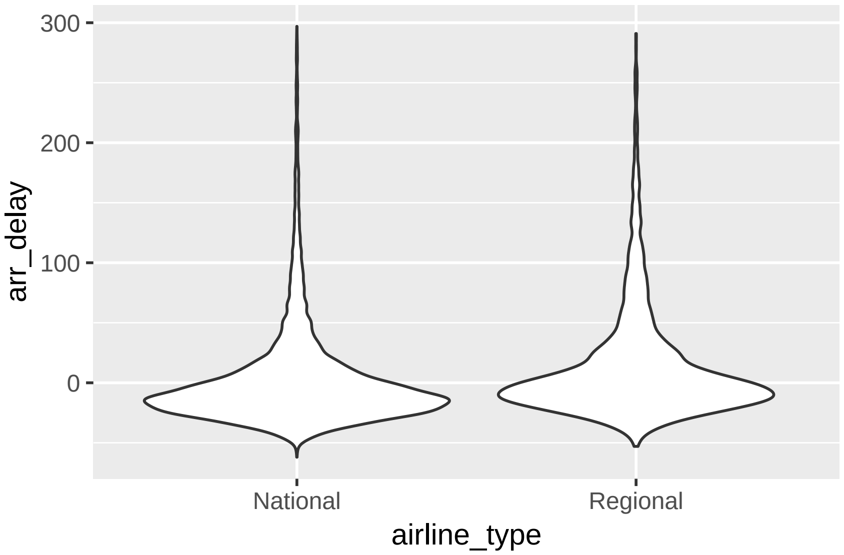


```
ord %>%
  group_by(name)%>%
  summarise("mean_arr_delay" = mean(arr_delay, na.rm = TRUE)) %>%
  mutate(name = fct_reorder(name, mean_arr_delay))%>%
  ggplot(aes(x=name, y=mean_arr_delay))+
    geom_point()
```



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).

```
flights1 %>%
filter(arr_delay <= 300, dest == "ORD") %>%
mutate(airline_type = ifelse(carrier == "UA" | carrier == "AA", "National", "Regional")) %>%
ggplot(aes(x = airline_type, y= arr_delay))+
geom_violin()
```


300
200
100
0
arr_delay
National
Regional
airline_type