Practice Set 2

Dhyey Mavani

Due by 10pm ET on Friday

Practice Set Information

During the week, you will get further practice with the material by working through the Practice Set, a set of problems designed to give you practice beyond the examples produced in the text.

You may work through these problems with peers, but all work must be completed by you (see the Honor Code in the syllabus) and you must indicate who you worked with below.

Even then, the best approach here is to try the problems on your own before discussing them with peers, and then write your final solutions yourself.

GitHub Workflow

1. Before editing this file, verify you are working on the copy saved in *your* repo for the course (check the filepath and the project name in the top right corner).
2. Before editing this file, make an initial commit of the file to your repo to add your copy of the problem set.
3. Change your name at the top of the file and get started!
4. You should *save, knit, and commit* the .Rmd file each time you've finished a question, if not more often. You should also *push* your commits back onto GitHub occasionally (you can do this after each commit).
5. When you think you are done with the assignment, save the pdf as "*Name_thisfilename_date*.pdf" before committing and pushing (this is generally good practice but also helps me in those times where I need to download all student homework files).

Gradescope Upload

For each question (e.g., 3.1), allocate all pages associated with the specific question. If your work for a question runs onto a page that you did not select, you may not get credit for the work. If you do not allocate *any* pages when you upload your pdf, you may get a zero for the assignment.

You can resubmit your work as many times as you want before the deadline, so you should not wait until the last minute to submit some version of your work. Unexpected delays/crises that occur on the day the assignment is due do not warrant extensions (please submit whatever you have done to receive partial credit).

Practicing Academic Integrity

If you worked with others or used resources outside of provided course material (notes, textbook, etc) to complete this assignment, please acknowledge them below using a bulleted list.

I acknowledge the following individuals with whom I worked on this assignment:

Name(s) and corresponding problem(s)

- SDS Fellows
- Kayla Ko

I used the following sources to help complete this assignment:

Source(s) and corresponding problem(s)

- N/A

Problem 1 **MDSR 5.2** Use the `Batting`, `Pitching`, and `Master` tables in the **Lahman** package to answer the following questions.

1.1 List the name of every player in baseball history who has accumulated at least 300 home runs (HR) and at least 300 stolen bases (SB). You can find the first and last name of the player in the `Master` data frame. Join this to your result along with the total home runs and total bases stolen for each of these elite players.

```
library(Lahman)
JoinedData<-full_join(Batting, Master)
BattingGrouped<-Batting%>%
  group_by(playerID) %>%
  summarize(
    totalHR = sum(HR),
    totalSB = sum(SB)
  )
JoinedData<-full_join(Master,BattingGrouped)
JoinedData%>%
  select(totalHR, totalSB, nameFirst, nameLast) %>%
  filter(totalHR >300 & totalSB > 300)
```

```
  totalHR totalSB nameFirst  nameLast
1     435     312    Carlos   Beltran
2     762     514     Barry     Bonds
3     332     461     Bobby     Bonds
4     438     314     Andre    Dawson
5     304     320     Steve    Finley
6     660     338    Willie      Mays
7     696     329      Alex Rodriguez
8     305     304    Reggie   Sanders
```

1.2 Similarly, list the names every pitcher in baseball history who has accumulated at least 300 wins (W) and at least 3,000 strikeouts (SO).

```
PitchingGrouped <- Pitching %>%
  group_by(playerID) %>%
  summarize(
    totalW = sum(W),
    totalSO = sum(SO)
  )
PitchingGrouped<-full_join(Master,PitchingGrouped)
PitchingGrouped%>%
  select(totalW, totalSO, nameFirst, nameLast) %>%
  filter(totalW >300 & totalSO > 3000)
```

```
  totalW totalSO nameFirst nameLast
```

```
1     329    4136     Steve  Carlton
2     354    4672     Roger  Clemens
3     303    4875     Randy  Johnson
4     417    3509    Walter  Johnson
5     355    3371      Greg   Maddux
6     318    3342      Phil   Niekro
7     314    3534   Gaylord    Perry
8     324    5714     Nolan     Ryan
9     311    3640       Tom   Seaver
10    324    3574       Don   Sutton
```

1.3 Finally, list the name and year of every player who has hit at least 50 home runs in a single season. Which player had the lowest batting average in that season? Note: Batting average is calculated as the number of hits (H) divided by the number of at bats (AB).

```
ModBatting <- full_join(Master, Batting) %>%
  mutate(BattingAverage = H/AB) %>%
  select(yearID, HR, BattingAverage, nameFirst, nameLast) %>%
  filter(HR >50)

ModBatting
```

```
   yearID HR BattingAverage nameFirst  nameLast
1    2019 53      0.2596315      Pete    Alonso
2    2010 54      0.2601054      Jose  Bautista
3    2001 73      0.3277311     Barry     Bonds
4    2013 53      0.2859589     Chris     Davis
5    1990 51      0.2774869     Cecil   Fielder
6    1977 52      0.3203252    George    Foster
7    1932 58      0.3641026    Jimmie      Foxx
8    2001 57      0.3251232      Luis  Gonzalez
9    1938 58      0.3147482      Hank Greenberg
10   1997 56      0.3042763       Ken   Griffey
11   1998 56      0.2843602       Ken   Griffey
12   2006 58      0.3132530      Ryan    Howard
13   2005 51      0.2627986    Andruw     Jones
14   2017 52      0.2841328     Aaron     Judge
15   1947 51      0.3132743     Ralph     Kiner
16   1949 54      0.3096539     Ralph     Kiner
17   1956 52      0.3527205    Mickey    Mantle
18   1961 54      0.3171206    Mickey    Mantle
19   1961 61      0.2694915     Roger     Maris
20   1955 51      0.3189655    Willie      Mays
21   1965 52      0.3172043    Willie      Mays
22   1996 52      0.3120567      Mark   McGwire
23   1998 70      0.2986248      Mark   McGwire
24   1999 65      0.2783109      Mark   McGwire
25   1947 51      0.3020478    Johnny      Mize
26   2006 54      0.2867384     David     Ortiz
27   2001 52      0.3180380      Alex Rodriguez
```

```
28   2002 57       0.2996795       Alex Rodriguez
29   2007 54       0.3138937       Alex Rodriguez
30   1920 54       0.3763676       Babe       Ruth
31   1921 59       0.3777778       Babe       Ruth
32   1927 60       0.3555556       Babe       Ruth
33   1928 54       0.3227612       Babe       Ruth
34   1998 66       0.3079316      Sammy       Sosa
35   1999 63       0.2880000      Sammy       Sosa
36   2001 64       0.3275563      Sammy       Sosa
37   2017 59       0.2814070 Giancarlo   Stanton
38   2002 52       0.3041667        Jim     Thome
39   1930 56       0.3555556       Hack    Wilson
```

```
min(ModBatting$BattingAverage)
```

```
[1] 0.2596315
```

```
GetMin <- ModBatting %>%
  filter(BattingAverage == min(ModBatting$BattingAverage))

GetMin
```

```
  yearID HR BattingAverage nameFirst nameLast
1   2019 53      0.2596315      Pete   Alonso
```

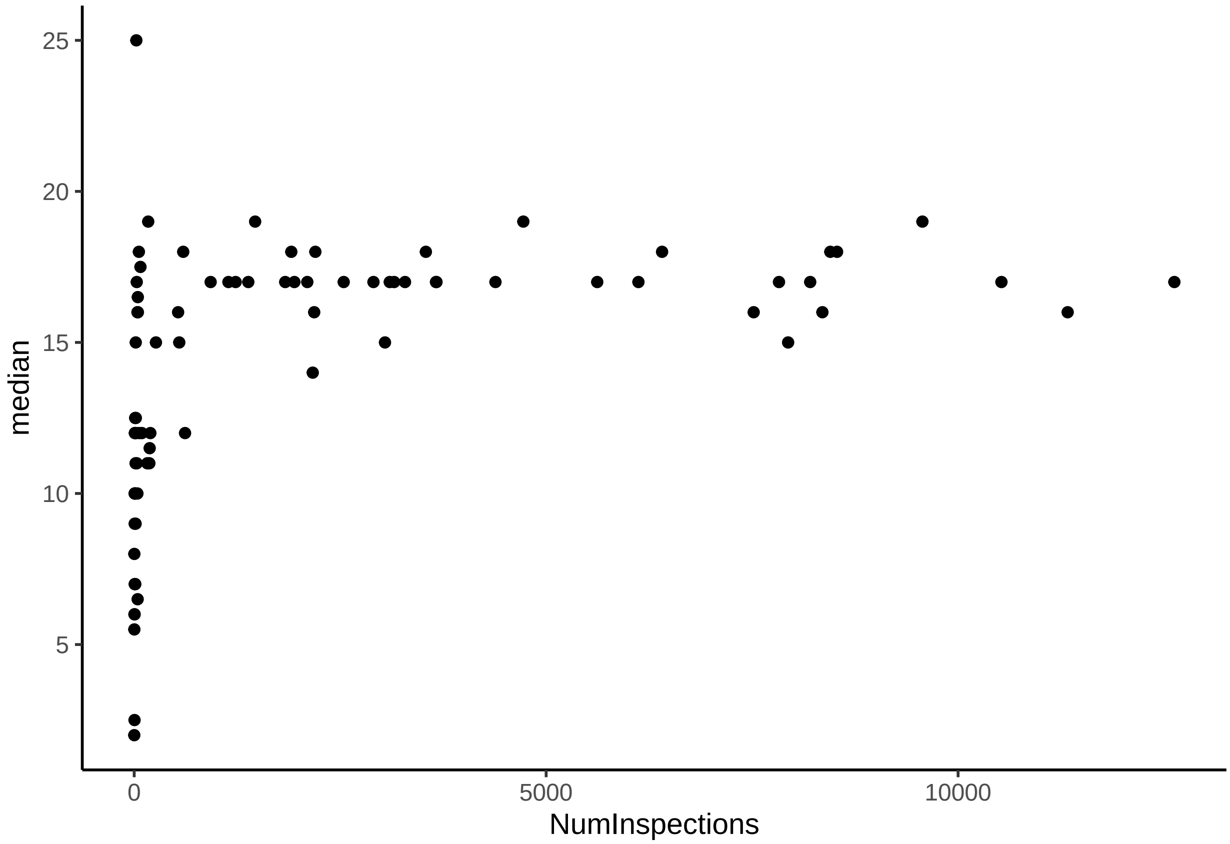
Problem 2 **MDSR 4.11 (modified)** The `Violations` data set in the **mdsr** package contains information regarding the outcome of health inspections of restaurants in New York City. Note that higher inspection scores indicate worse violations: "restaurants with an inspection score between 0 and 13 points earn an A, those with 14 to 27 points receive a B and those with 28 or more a C" (nyc.gov).

2.1 Use these data to calculate the median violation score by zip code for zip codes in Manhattan. What pattern, if any, do you see between the number of inspections and the median score? Generate a visualization to support your response.

```
Violations_median <- Violations %>%
  select(boro, zipcode, score) %>%
  na.omit() %>%
  filter(boro == "MANHATTAN") %>%
  group_by(zipcode)%>%
  summarise(
    median=median(score),
    NumInspections=n()
  )
Violations_median
```

```
## # A tibble: 81 x 3
##    zipcode median NumInspections
##      <int>  <dbl>          <int>
##  1   10001     15           7937
##  2   10002     18           8449
##  3   10003     17          12625
##  4   10004     14           2167
##  5   10005     17           1144
##  6   10006     17            928
##  7   10007     16           2185
##  8   10009     17           5620
##  9   10010     17           4385
## 10   10011     17           8205
## # ... with 71 more rows
```

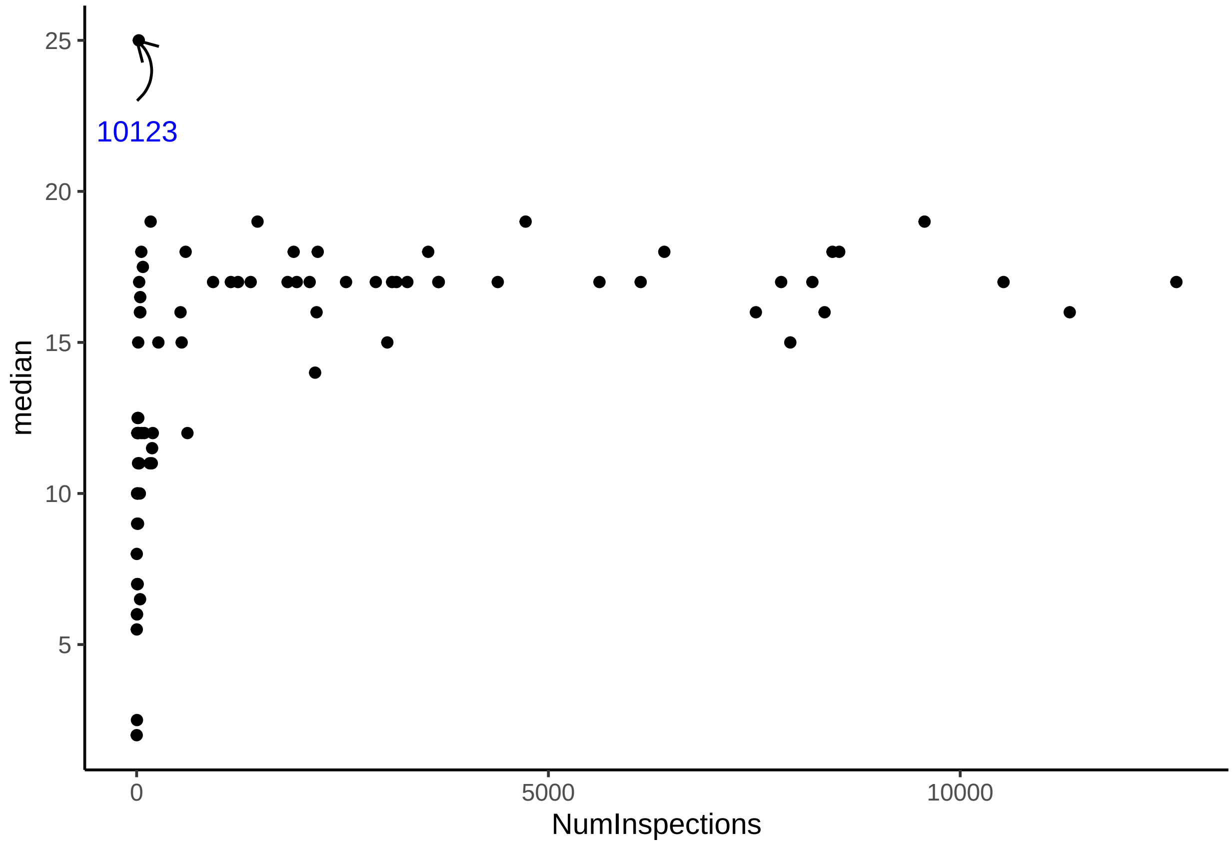
```
b <- ggplot(data = Violations_median)+
  geom_point(mapping = aes(x = NumInspections,
                           y = median))
b
```



2.2 In your visualization above, there are several potential outliers but there is one zipcode in particular that does not seem to fall along the general trend. Add text to the outlier identifying what zipcode it is, and add an arrow pointing from the text to the observation. Note: first, you may want to `filter()` to identify the zipcode (so you know what text to add to the plot).

```
#zip= 10123 filter(Violations_median, median > 20)
df <- data.frame(x1 = 5.57, x2 = 2.62, y1 = 23.0, y2 = 25.0)

b + geom_curve(
  aes(x = x1, y = y1, xend = x2, yend = y2),
  data = df,
  arrow = arrow(length = unit(0.03, "npc")))+
  annotate("text", x = 5.56, y = 22.0, color = "blue",
           label = "10123")
```


10123
25
20
15
10
5
median
0
5000
10000
NumInspections

Problem 3 **MDSR 6.5** Generate the code to convert the data frame from the starting point (Figure 1) to the results (Figure 2). Hint: use `pivot_longer()` in conjunction with `pivot_wider()`.

grp	sex	meanL	sdL	meanR	sdR
A	F	0.225	0.106	0.340	0.085
A	M	0.470	0.325	0.570	0.325
B	F	0.325	0.106	0.400	0.071
B	M	0.547	0.308	0.647	0.274

Figure 1: Starting point

	grp	F.meanL	F.meanR	F.sdL	F.sdR	M.meanL	M.meanR	M.sdL	M.sdR
1	A	0.22	0.34	0.11	0.08	0.47	0.57	0.33	0.33
2	B	0.33	0.40	0.11	0.07	0.55	0.65	0.31	0.27

Figure 2: Results

```
g<-data.frame(grp = c("A", "A", "B", "B"),
           sex = c("F", "M", "F", "M"),
           meanL = c(0.225, 0.470, 0.325, 0.547),
           sdL = c(0.106,0.325,0.106,0.308),
           meanR = c(0.340, 0.570, 0.400, 0.647),
           sdR=c(0.085, 0.325,0.071,0.274))
g
```

```
  grp sex meanL   sdL meanR   sdR
1   A   F 0.225 0.106 0.340 0.085
2   A   M 0.470 0.325 0.570 0.325
3   B   F 0.325 0.106 0.400 0.071
4   B   M 0.547 0.308 0.647 0.274
```

```
longtable <- g %>%
  pivot_longer(
    cols = meanL:sdR,
    names_to = "measurement",
    values_to = "value"
  )
widetable <- longtable %>%
  pivot_wider(
    names_from = c(sex, measurement),
    values_from = value,
    names_sep = "."
  )
widetable %>% kable()
```

grp	F.meanL	F.sdL	F.meanR	F.sdR	M.meanL	M.sdL	M.meanR	M.sdR
A	0.225	0.106	0.34	0.085	0.470	0.325	0.570	0.325
B	0.325	0.106	0.40	0.071	0.547	0.308	0.647	0.274