Practice Set 2

Kauila Tengan

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)

Mahathi & Caroline, problem 1.1, 1.2 SDS fellows (Maggie), 1.3, 2.1 *Rohil 2.2* Abbey, 3

I used the following sources to help complete this assignment:

Source(s) and corresponding problem(s)

- textbook chapter 4 and 5 for 1.1, 1.2

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.

```
batting2 <- Batting %>%
  group_by(playerID) %>%
  summarize(HR = sum(HR), SB = sum(SB)) %>%
  left_join(y = Master, by = "playerID") %>%
  filter(HR >= 300 & SB >= 300) %>%
  select(nameFirst, nameLast, playerID, HR, SB)

batting2
```

```
# A tibble: 8 x 5
  nameFirst nameLast  playerID     HR    SB
  <chr>     <chr>     <chr>     <int> <int>
1 Carlos    Beltran   beltrca01   435   312
2 Barry     Bonds     bondsba01   762   514
3 Bobby     Bonds     bondsbo01   332   461
4 Andre     Dawson    dawsoan01   438   314
5 Steve     Finley    finlest01   304   320
6 Willie    Mays      mayswi01    660   338
7 Alex      Rodriguez rodrial01   696   329
8 Reggie    Sanders   sandere02   305   304
```

```
# Master
```

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).

```
# create a new dataset
 pitching2 <- Pitching %>%
  group_by(playerID) %>%
  summarize(W = sum(W), SO = sum(SO)) %>% #summarize W and SO
  left_join(y = Master, by = "playerID") %>% #left join
  filter(W >= 300 & SO >= 300) %>%
  select(nameFirst, nameLast, playerID, W, SO)

pitching2
```

```
# A tibble: 24 x 5
   nameFirst nameLast  playerID      W    SO
```

```
   <chr>     <chr>     <chr>     <int> <int>
 1 Pete      Alexander alexape01   373  2198
 2 Steve     Carlton   carltst01   329  4136
 3 John      Clarkson  clarkjo01   328  1978
 4 Roger     Clemens   clemero02   354  4672
 5 Pud       Galvin    galvipu01   365  1807
 6 Tom       Glavine   glavito02   305  2607
 7 Lefty     Grove     grovele01   300  2266
 8 Randy     Johnson   johnsra05   303  4875
 9 Walter    Johnson   johnswa01   417  3509
10 Tim       Keefe     keefeti01   342  2564
# ... with 14 more rows
```

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).

Pete Alonso had the lowest batting average in year 2019.

```
batting3 <- Batting %>%
  group_by(playerID) %>%
  filter(HR >= 50) %>% #HR >= 50 filter
  mutate(BA = H/AB) %>% #create a new variable BA
  inner_join(Master, by = "playerID") %>%
  select(nameFirst, nameLast, yearID, HR, BA, playerID) %>%
  arrange(BA)

batting3
```

```
# A tibble: 45 x 6
# Groups:   playerID [30]
   nameFirst nameLast yearID    HR    BA playerID
   <chr>     <chr>     <int> <int> <dbl> <chr>
 1 Pete      Alonso     2019    53 0.260 alonspe01
 2 Jose      Bautista   2010    54 0.260 bautijo02
 3 Andruw    Jones      2005    51 0.263 jonesan01
 4 Roger     Maris      1961    61 0.269 marisro01
 5 Greg      Vaughn     1998    50 0.272 vaughgr01
 6 Cecil     Fielder    1990    51 0.277 fieldce01
 7 Mark      McGwire    1999    65 0.278 mcgwima01
 8 Giancarlo Stanton    2017    59 0.281 stantmi03
 9 Aaron     Judge      2017    52 0.284 judgeaa01
10 Ken       Griffey    1998    56 0.284 griffke02
# ... with 35 more rows
```

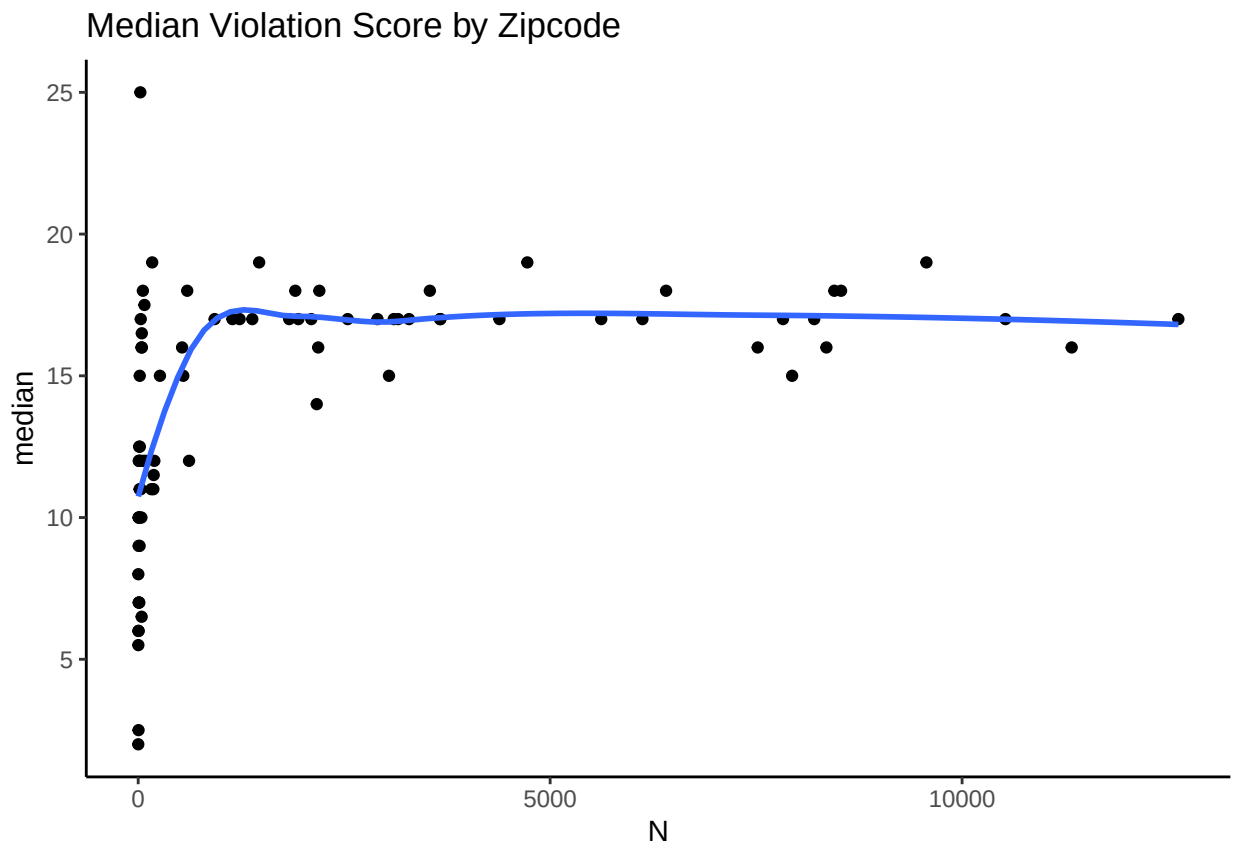
Problem 2 **MDSR 4.11 (modified)** The `Violations` data set in the **mdsr** package contains information regarding the outcome of health inspections of restaurants in New York City. Note that higher inspection scores indicate worse violations: "restaurants with an inspection score between 0 and 13 points 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.

```
viomedians <- Violations %>%
  filter(boro == "MANHATTAN") %>%
  drop_na(score) %>%
  group_by(zipcode) %>%
  summarize(median = median(score),
            N = n())

m <- ggplot(data = viomedians, aes(x = N, y = median)) +
  geom_point() +
  labs(title = "Median Violation Score by Zipcode",
       median = "Median",
       N = "Number of Inspections") +
  geom_smooth(method = "loess", se = FALSE)


m
```



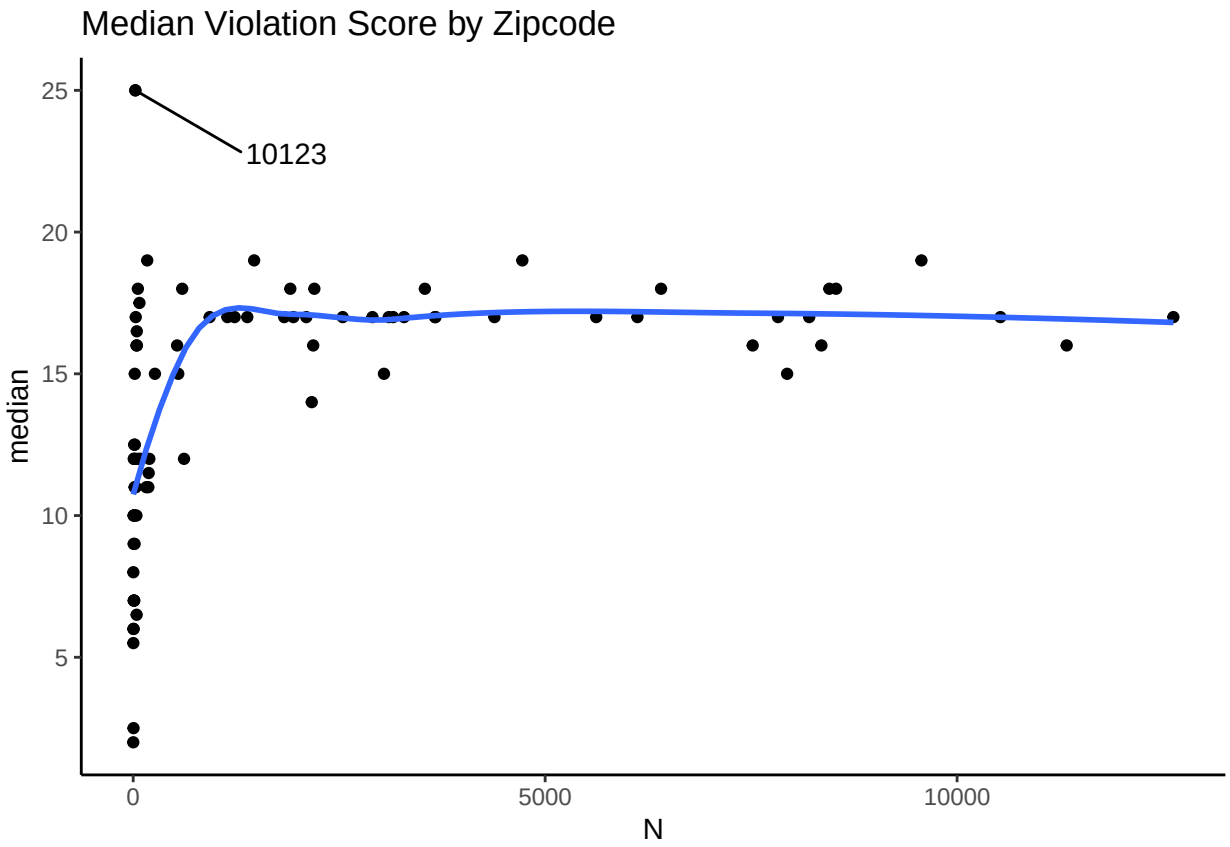
2.2 In your visualization above, there are several potential outliers but there is one zipcode in particular that does not seem to fall along the general trend. Add text to the outlier identifying what zipcode it is, and add an arrow pointing from the text to the observation. Note: first, you may want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

The asymptote is approximately 18. There is one clear outlier around 25. As the number increases, the distribution becomes tighter and it looks like there are less outliers.

```
viomedians %>%  #check for the outlier
  filter(median > 20)
```

```
# A tibble: 1 x 3
  zipcode median     N
    <int>  <dbl> <int>
1   10123     25    26
```

```
m2 <- subset(viomedians, zipcode == 10123) #create new dataset with outlier
m + geom_point(data = m2) +
  # geom_segment(aes(x = 5, y = 30, xend = 3.5, yend = 25),
  #              arrow = arrow(length = unit(0.2, "cm"),)))
  ggrepel::geom_text_repel(data = m2, label = "10123", vjust = 2)
```


Median Violation Score by Zipcode
10123
median
N
25
20
15
10
5
0
5000
10000

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

```
datatable <- 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)
)

data_long <- datatable %>%
  pivot_longer(-c("grp", "sex"), names_to = "group", values_to = "value")

data_wide <- data_long %>%
  pivot_wider(names_from = c("sex", "group"),
              values_from = value,
              values_fill = 0) %>%
  kable()

data_wide
```

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