STAT 231: Problem Set 2B

Kim Zhou

due by 5 PM on Friday, March 5

Series B homework assignments are designed to help you further ingest and practice the material covered in class over the past week(s). You are encouraged to work with other students, but all code must be written by you and you must indicate below who you discussed the assignment with (if anyone).

Steps to proceed:

1. In RStudio, go to File > Open Project, navigate to the folder with the course-content repo, select the course-content project (course-content.Rproj), and click "Open"
2. Pull the course-content repo (e.g. using the blue-ish down arrow in the Git tab in upper right window)
3. Copy ps2B.Rmd from the course repo to your repo (see page 6 of the GitHub Classroom Guide for Stat231 if needed)
4. Close the course-content repo project in RStudio
5. Open YOUR repo project in RStudio
6. In the ps2B.Rmd file in YOUR repo, replace "YOUR NAME HERE" with your name
7. Add in your responses, committing and pushing to YOUR repo in appropriate places along the way
8. Run "Knit PDF"
9. Upload the pdf to Gradescope. Don't forget to select which of your pages are associated with each problem. *You will not get credit for work on unassigned pages (e.g., if you only selected the first page but your solution spans two pages, you would lose points for any part on the second page that the grader can't see).*

If you discussed this assignment with any of your peers, please list who here:

ANSWER: Jamie Dailey

MDSR Exercise 4.14 (modified)

Use the `Pitching` data frame from the `Lahman` package to identify every pitcher in baseball history who has accumulated at least 300 wins (`W`) and at least 3,000 strikeouts (`SO`).

a. How many pitchers meet this criteria?

ANSWER: There are 10 pitchers that have had more than 300 wins and 3000 strike outs over their career.

```
library(Lahman)

pitching_clean <- Pitching %>%
  group_by(playerID) %>%
  summarise(num_wins = sum(W), num_so = sum(SO)) %>%
  filter(num_wins > 300, num_so > 3000)

pitching_clean
```

```
## # A tibble: 10 x 3
##    playerID  num_wins num_so
##    <chr>        <int>  <int>
##  1 carltst01      329   4136
##  2 clemero02      354   4672
##  3 johnsra05      303   4875
##  4 johnswa01      417   3509
##  5 maddugr01      355   3371
##  6 niekrph01      318   3342
##  7 perryga01      314   3534
##  8 ryanno01       324   5714
##  9 seaveto01      311   3640
## 10 suttodo01      324   3574
```

b. Which of these pitchers had the most accumulated strikeouts? How many strikeouts had he accumulated? What is the most strikeouts he had in one season?

ANSWER: The player with the most accumulated strikeouts was Nolan Ryan with a total of 5714 strikeouts over his career. The most strikeouts he got in a season was 383 in 1973.

```
pitching_clean <- pitching_clean %>%
  arrange(desc(num_so))

pitching_clean
```

```
## # A tibble: 10 x 3
##    playerID  num_wins num_so
##    <chr>        <int>  <int>
##  1 ryanno01       324   5714
##  2 johnsra05      303   4875
##  3 clemero02      354   4672
##  4 carltst01      329   4136
```

```
##  5 seaveto01      311   3640
##  6 suttodo01      324   3574
##  7 perryga01      314   3534
##  8 johnswa01      417   3509
##  9 maddugr01      355   3371
## 10 niekrph01      318   3342
```

```
pitching_so <- Pitching %>%
  filter(playerID == "ryanno01") %>%
  select(playerID, yearID, SO) %>%
  arrange(desc(SO)) %>%
  left_join(Master, by = c("playerID" = "playerID")) %>%
  select(nameFirst, nameLast, yearID, SO)

pitching_so
```

```
##    nameFirst nameLast yearID  SO
## 1      Nolan     Ryan   1973 383
## 2      Nolan     Ryan   1974 367
## 3      Nolan     Ryan   1977 341
## 4      Nolan     Ryan   1972 329
## 5      Nolan     Ryan   1976 327
## 6      Nolan     Ryan   1989 301
## 7      Nolan     Ryan   1987 270
## 8      Nolan     Ryan   1978 260
## 9      Nolan     Ryan   1982 245
## 10     Nolan     Ryan   1990 232
## 11     Nolan     Ryan   1988 228
## 12     Nolan     Ryan   1979 223
## 13     Nolan     Ryan   1985 209
## 14     Nolan     Ryan   1991 203
## 15     Nolan     Ryan   1980 200
## 16     Nolan     Ryan   1984 197
## 17     Nolan     Ryan   1986 194
## 18     Nolan     Ryan   1975 186
## 19     Nolan     Ryan   1983 183
## 20     Nolan     Ryan   1992 157
## 21     Nolan     Ryan   1981 140
## 22     Nolan     Ryan   1971 137
## 23     Nolan     Ryan   1968 133
## 24     Nolan     Ryan   1970 125
## 25     Nolan     Ryan   1969  92
## 26     Nolan     Ryan   1993  46
## 27     Nolan     Ryan   1966   6
```

MDSR Exercise 4.17 (modified)

a. The Violations data set in the `mdsr` package contains information regarding the outcome of health inspections in New York City. Use these data to calculate the median violation score by zipcode and dba for zipcodes 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.

ANSWER: There does not appear to be a patter between the number of inspections and the median score. The majority of the data falls under 75 visits and under a median score of 45.

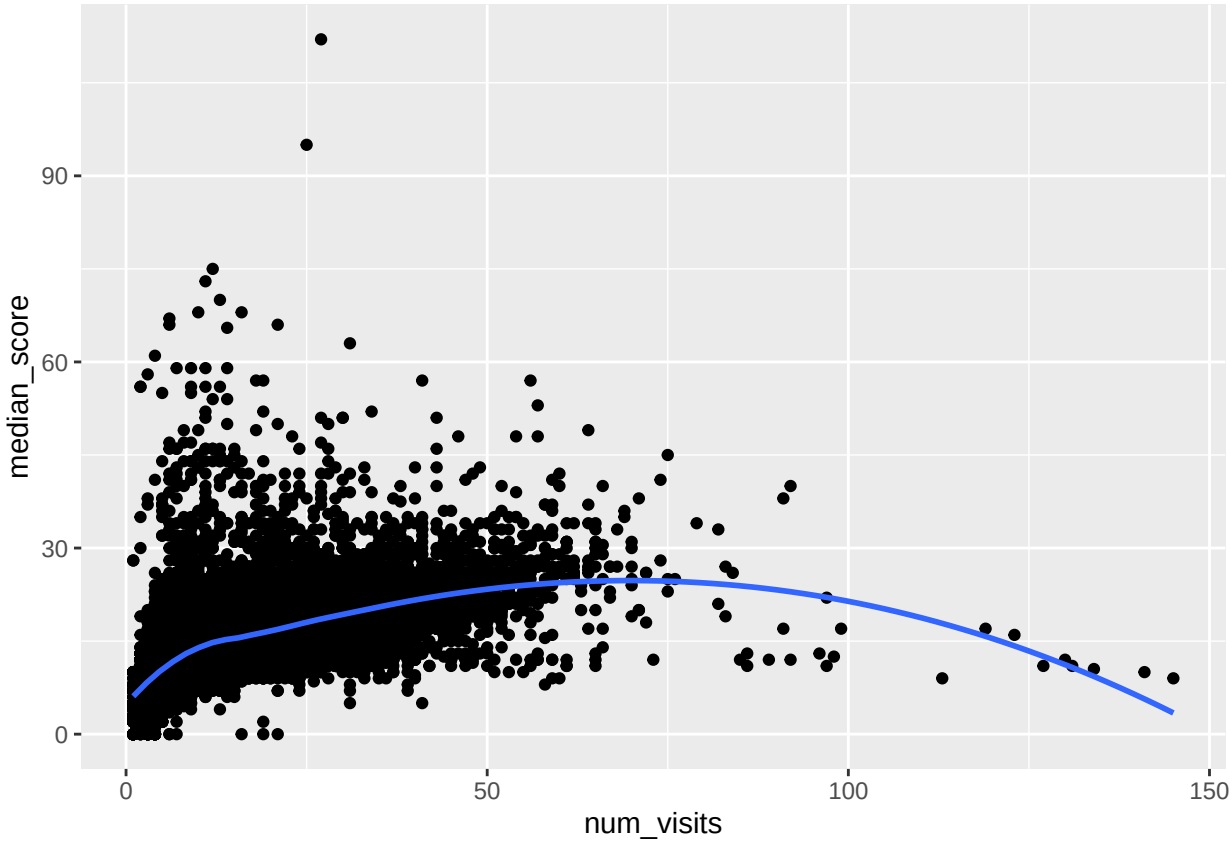
```
library(mdsr)
# calculate median score for each unique dba within a unique zipcode for all
# zipcodes in manhattan
nyc_violations <- Violations %>%
  filter(boro == "MANHATTAN") %>%
  select(dba, zipcode, score, inspection_date)

nyc_violations <- nyc_violations %>%
  filter(is.na(score) == FALSE) %>%
  group_by(dba, zipcode) %>%
  summarise(median_score = median(score), num_visits = n())
```

```
## `summarise()` has grouped output by 'dba'. You can override using the `.groups` argument.
```

```
# plot number of inspections vs median score
ggplot(data = nyc_violations, aes(x = num_visits, y = median_score)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
nyc_violations
```

```
## # A tibble: 9,359 x 4
## # Groups:   dba [8,106]
##    dba                                              zipcode median_score num_visits
##    <chr>                                              <int>        <dbl>      <int>
##  1 ''W'' CAFE                                         10018         22           23
##  2 (PUBLIC FARE) 81st street and central park w~      10019         19           19
##  3 @NINE                                              10036         14           50
##  4 / L'ECOLE                                          10013         19           15
##  5 $1 PIZZA $2 BEER                                   10012         17           40
##  6 1 2 3 BURGER SHOT BEER                             10019         20           18
##  7 1 DARBAR                                           10017         13           24
##  8 1 EAST 66TH STREET KITCHEN                         10065          3.5          4
##  9 1 OAK                                              10011         10           13
## 10 1 STOP PATTY SHOP                                  10031         11           30
## # ... with 9,349 more rows
```

b. In your visualization in part (a), there should be at least a few points that stand out as outliers. For *one of the outliers*, add text to the outlier identifying what business it is and an arrow pointing from the text to the observation. First, you may want to `filter` to identify the name of the business (so you know what text to add to the plot).

(Can't remember how to create a curved arrow in `ggplot`? The answers to this question on Stack Exchange may help. Can't remember how to add text to the plot in `ggplot`? Check out the text examples with `annotate` here, or answers to this question that use `geom_text`.)

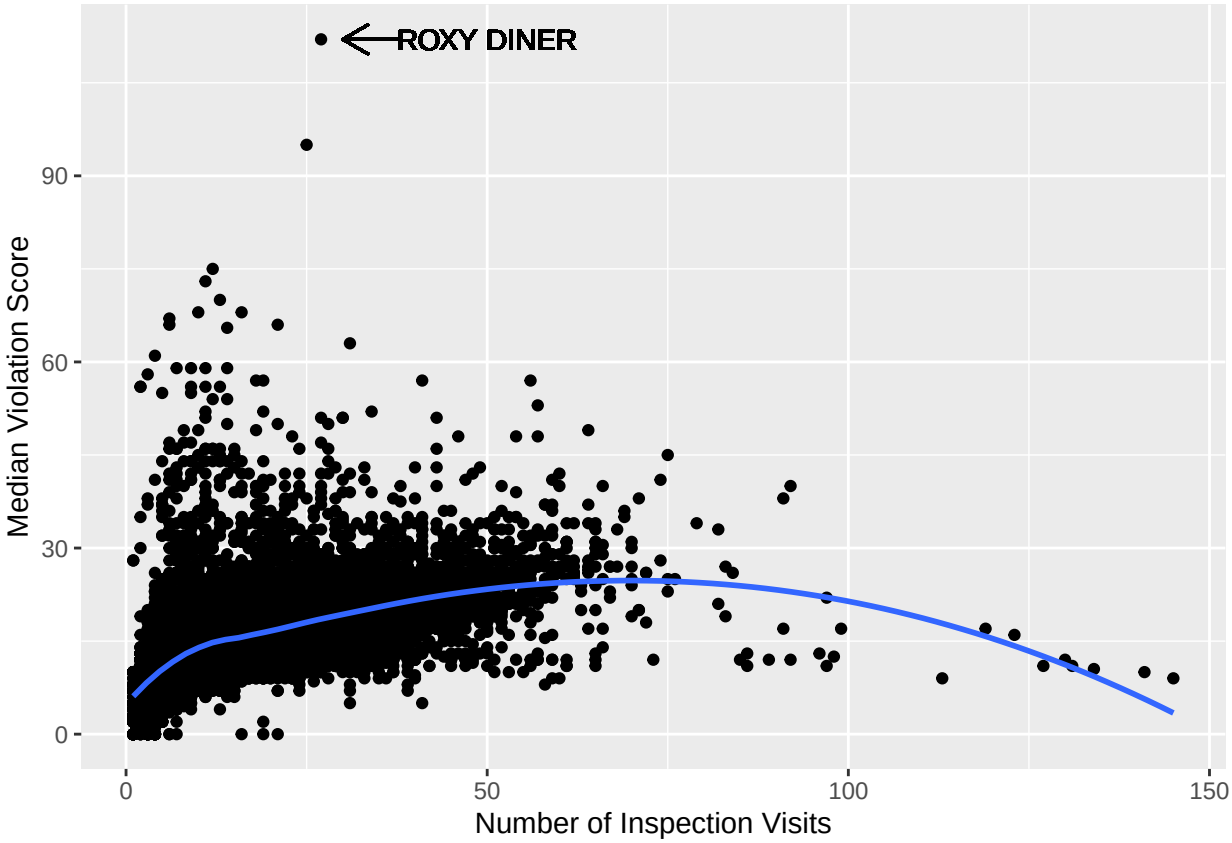
```r
# highest median score is an outlier
nyc_violations <- nyc_violations %>%
  arrange(desc(median_score))

nyc_violations
```

```
## # A tibble: 9,359 x 4
## # Groups:   dba [8,106]
##    dba                      zipcode median_score num_visits
##    <chr>                      <int>        <dbl>      <int>
##  1 ROXY DINER                 10036          112         27
##  2 BONJOUR CREPES & WINE      10128           95         25
##  3 SUSHI DOJO EXPRESS         10014           75         12
##  4 BY CHLOE                   10012           73         11
##  5 BAO BAO CAFE               10010           70         13
##  6 ORTIZ RESTAURANT           10032           68         10
##  7 VILLAGE CROWN              10280           68         16
##  8 FANTASTIC TEA SHOP         10003           67          6
##  9 BTH RESTAURANT & LOUNGE    10027           66         21
## 10 PUEBLA MEXICAN FOOD        10002           66          6
## # ... with 9,349 more rows
```

```r
ggplot(data = nyc_violations, aes(x = num_visits, y = median_score)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE) +
  geom_text(x = 50, y = 112, label = "ROXY DINER") +
  geom_segment(aes(x = 38, y = 112, xend = 30, yend = 112),
                   arrow = arrow(length = unit(0.4, "cm"))) +
  labs(
    x = "Number of Inspection Visits",
    y = "Median Violation Score"
  )
```

```
## `geom_smooth()` using formula 'y ~ x'
```


ROXY DINER
Median Violation Score
90
60
30
0
0
50
100
150
Number of Inspection Visits

MDSR Exercise 5.7

Generate the code to convert the data frame shown with this problem in the textbook (on page 130, and shown below) to wide format (i.e., the result table). Hint: use `gather()` in conjunction with `spread()`; OR `pivot_longer()` in conjunction with `pivot_wider()`.

```
FakeDataLong <- data.frame(grp = c("A","A","B", "B")
                           , sex = c("F", "M", "F", "M")
                           , meanL = c(0.22, 0.47, 0.33, 0.55)
                           , sdL = c(0.11, 0.33, 0.11, 0.31)
                           , meanR = c(0.34, 0.57, 0.40, 0.65)
                           , sdR = c(0.08, 0.33, 0.07, 0.27))

FakeDataNarrow <- FakeDataLong %>%
  gather(key = method, value = values, meanL, meanR, sdL, sdR) %>%
  unite(sex_method, sex, method, sep = ".") %>%
  spread(key = sex_method, value = values)


FakeDataNarrow
```

```
##   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
```

PUG Brainstorming

What topics or questions are you interested in exploring related to your PUG theme? Dream big here. Don't worry about whether there is data out there that's available and accessible that you could use to address your questions/topics. Just brainstorm some ideas that get you excited. Then, email your PUG team with your ideas. Title the email "PS2B Brainstorming: PUG [#] [Topic]" and CC me (kcorreia@amherst.edu) on the email. If another PUG member already initiated the email, reply all to their email.

If you don't remember your PUG # and Topic, please see the file "PUGs" on the Moodle page under this week.

If you don't know your PUG members email address, go to the class's Google group conversations (e.g., by clicking the link "Link to Google group conversations" at the top of our Moodle course page). Then, on the navigation panel (left hand side), select "Members".

> ANSWER: Do not write anything here. Email your ideas to your PUG team and me in a message titled "PS2B Brainstorming: PUG [#] [Topic]".