STAT 231: Problem Set 9B

Evan Daisy

due by 10 PM on Friday, May 14

This homework assignment is designed to help you further ingest, practice, and expand upon the material covered in class over the past week(s). You are encouraged to work with other students, but all code and text 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 ps9B.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 ps9B.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:

1. Ethics follow-up

(a) Thinking about the discussion you had with the first group you were with during class on Tuesday 5/4 (focused on either "Predicting Policing & Recidivism" or "Predicting Financial Risk"), did your perspective on, or understanding of, any of the questions shift? If so, please describe. If not, was there anything you found surprising in the resources or your first group discussion?

ANSWER: During our discussion, I began to consider the perspective that one of my groupmates brought up that even ignoring algorithmic bias, there are some decisions that simply should not be made by algorithms. We talked about how algorithms can only ever approximate human decisions. Part of this might be that they feel no human emotions, and this can be both a blessing and a curse. Theoretically this helps eliminate bias, but as we learned in the resources this often is not the case, and without the ability to feel compassion for the people who are impacted by a decision, an algorithm may end up making a poorer choice than a human who understands human emotions.

An aspect of the resources that I had never considered before is how quickly algorithms travel and how difficult it is to correct algorithmic bias once a biased piece of software is released. The obvious lesson from these resources is that we need to be extremely careful in making software to ensure that it doesn't incorporate human bias, but we also need to consider how to eliminate the bias that has already been incorporated into software, which is an even more difficult undertaking.

(b) Thinking about the discussion you had with the second group you were with during class on Tuesday 5/4 (focused on considering the use of algorithms in the college admissions process), did your perspective on, or understanding of, the use of algorithms in these contexts shift? If not, was there anything you found surprising in the resources or your second group discussion?

ANSWER: We discussed the ways in which we had heard of algorithms being used in these contexts, such as filtering out applications using GPA cutoffs or things like that, and wondered how the use of algorithms might expand in the future. We considered text analysis in the reading of college essays by computers, but came to the conclusion that there was no way for an algorithm to holisticly evaluate a college application. This makes sense, but I had never thought about it before. We did suggest the use of an algorithm as a supplementary tool, to provide a score based on numerical aspects of the application which could be the last thing an admissions person looks at before making a decision.

It surprised me that schools have been using data on how often prospective students visit the college website to determine likelihood of acceptance. It seems like a good idea, but to do so without the knowledge of the students (whose data it is) is necessary for it to work and seems a bit unethical.

CHOOSE ONE OF 2 (Clustering), 3 (Simulations) or 4 (SQL) to COMPLETE

2. Clustering

MDSR Exercise 9.5

Baseball players are voted into the Hall of Fame by the members of the Baseball Writers of America Association. Quantitative criteria are used by the voters, but they are also allowed wide discretion. The following code identifies the position players who have been elected to the Hall of Fame and tabulates a few basic statistics, include their number of career hits (`tH`), home runs (`tHR`), runs batted in (`tRBI`), and stolen bases (`tSB`). Use the `kmeans()` function to perform a cluster analysis on these players. Describe the properties that seem common to each cluster.

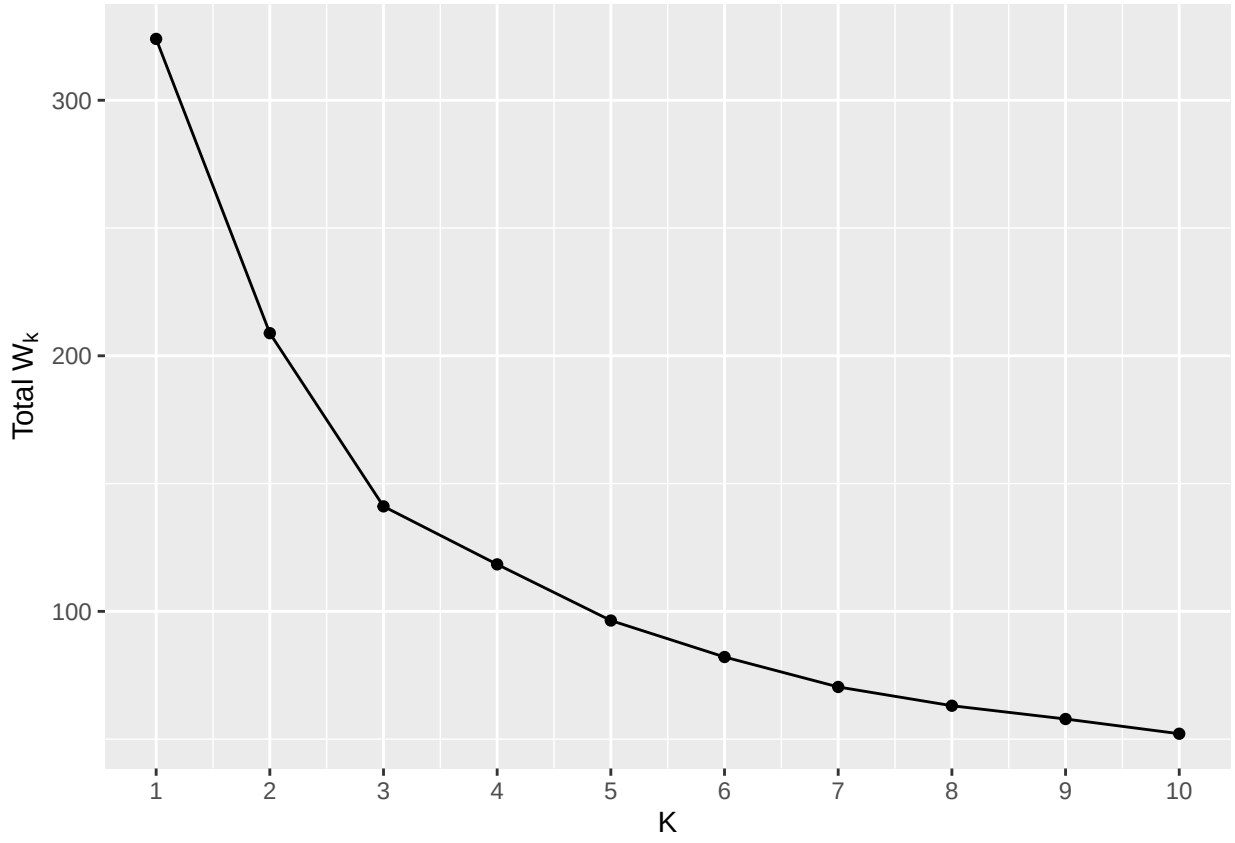
Don't forget to standardize the variables before clustering, if applicable.

> ANSWER: Because I see a significant decrease in within-cluster variation between 2 and 3 clusters, I will be using 3 clusters for this analysis. Cluster 1 is characterized by few Home Runs and many Stolen Bases. Cluster 2 is characterized by many Home Runs and many Runs Batted In. Cluster 3 is characterized by few Stolen Bases and relatively few Career Hits.

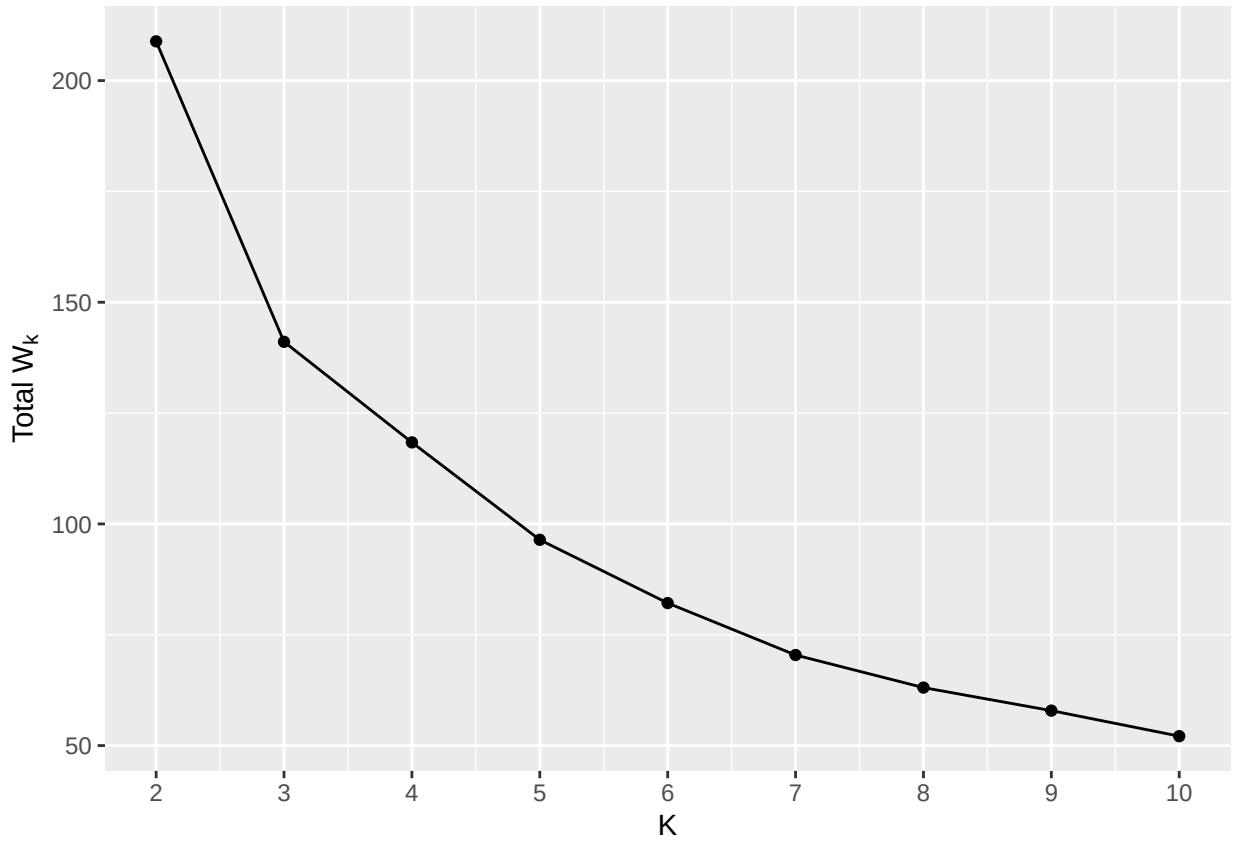
```
library(tidyverse)
library(mdsr)
library(Lahman)
library(GGally)

##### PLEASE DO NOT CHANGE THIS SEED NUMBER
##### keep set.seed(75)
set.seed(75)

hof <- Batting %>%
  group_by(playerID) %>%
  inner_join(HallOfFame, by = "playerID") %>%
  filter(inducted == "Y" & votedBy == "BBWAA") %>%
  summarize(tH = sum(H), tHR = sum(HR), tRBI = sum(RBI), tSB = sum(SB)) %>%
  filter(tH > 1000)
# Start by Standardizing
hof_std <- hof %>%
  mutate_if(is.numeric, funs(`std`=scale(.) %>% as.vector()))
```

```
## Warning: ‘funs()‘ was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with ‘tibble::lst()‘:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
```

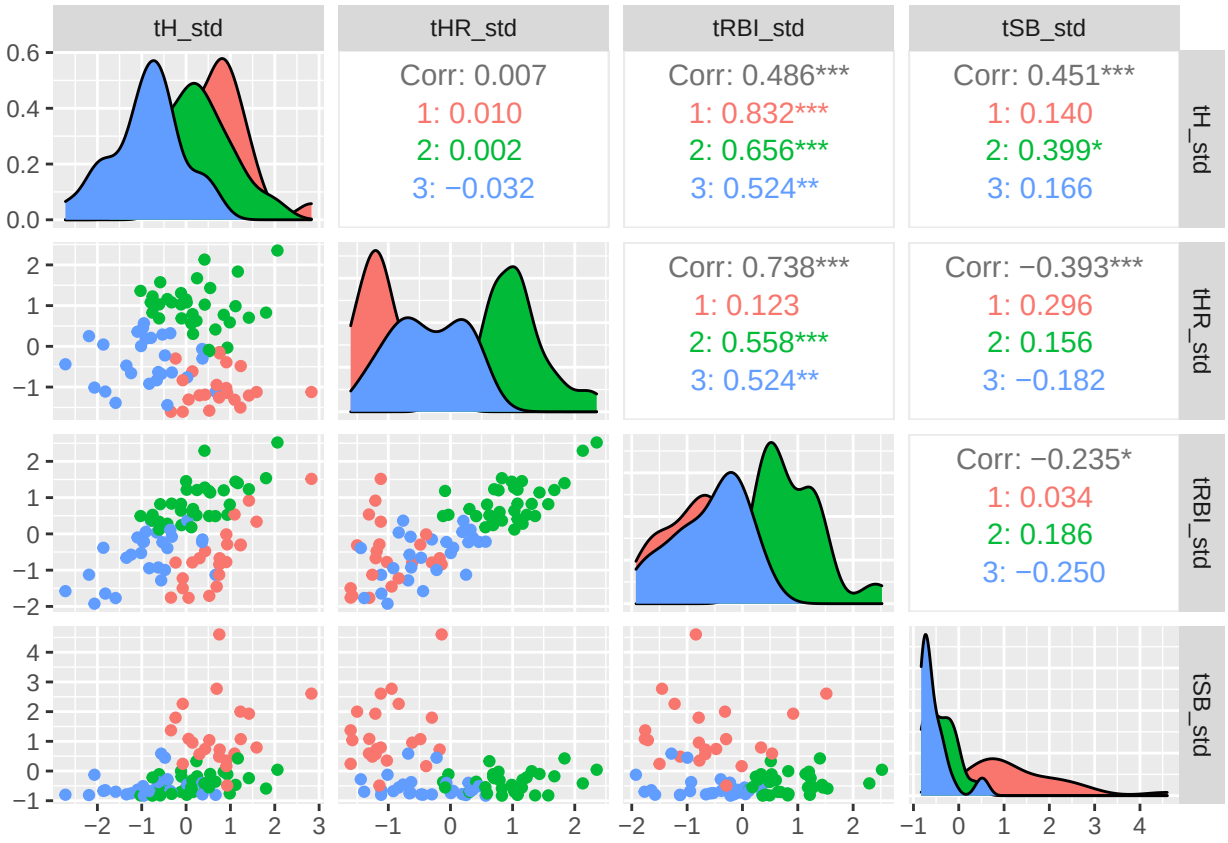
```r
# Then Make an Elbow Plot
hof_fig <- matrix(NA, nrow=10, ncol=2)

set.seed(75)
for (i in 1:10){
  hof_fig[i,1] <- i
  hof_fig[i,2] <- kmeans(hof_std[,c(6 : 9)]
                         , centers=i
                         , nstart=20)$tot.withinss
}
ggplot(data = as.data.frame(hof_fig), aes(x = V1, y = V2)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(breaks=c(1:10)) +
  labs(x = "K", y = expression("Total W"[k]))
```




```r
ggplot(data = as.data.frame(hof_fig[2:10,]), aes(x = V1, y = V2)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(breaks=c(1:10)) +
  labs(x = "K", y = expression("Total W"[k]))
```




```
hof_out <- kmeans(hof_std[, c(6 : 9)], centers = 3, nstart = 20)
hof_clusts <- hof_std %>%
  mutate(cluster = as.character(hof_out$cluster))
vars <- c("tH_std", "tHR_std", "tRBI_std", "tSB_std")
ggpairs(data = hof_clusts
        , aes(color = cluster)
        , columns = vars)
```


tH_std
tHR_std
tRBI_std
tSB_std
Corr: 0.007
1: 0.010
2: 0.002
3: −0.032
Corr: 0.486***
1: 0.832***
2: 0.656***
3: 0.524**
Corr: 0.451***
1: 0.140
2: 0.399*
3: 0.166
Corr: 0.738***
1: 0.123
2: 0.558***
3: 0.524**
Corr: −0.393***
1: 0.296
2: 0.156
3: −0.182
Corr: −0.235*
1: 0.034
2: 0.186
3: −0.250

CHOOSE ONE OF 2 (Clustering), 3 (Simulations) or 4 (SQL) to COMPLETE

3. Simulation

MDSR Exercise 10.6 (modified)

Equal variance assumption: What is the impact of the violation of the equal variance assumption for linear regression models? Repeatedly generate data from two "true" models:

(1) where the equal variance assumption is met: $y_i \sim N(\mu_i, \sigma)$
(2) where the equal variance assumption is violated: $y_i \sim N(\mu_i, \sigma_i)$

, where $\mu_i = -1 + 0.5 * X_{1i} + 1.5 * X_{2i}$, σ=1 in (1), $\sigma_i = 1 + X_{2i}$ in (2), and X_1 is a binary predicator and X_2 is Uniform(0,5).

Code to get you started is given below. (Note that in (2) the standard deviation is dependent upon x2, which is random; i.e., the equal variance assumption is violated. The Ys are *not* generated from a distribution with the same variance in (2).)

For each simulation, fit the linear regression model and display the distribution of 1,000 estimates of the β_1 parameter. Does the distribution of the parameter follow a normal distribution in both cases? Is it centered around β_1? How does the variability in the distributions compare (variance in $\hat{\beta}_1$ when the equal variance assumption is met vs. when it is violated)?

ANSWER:

```
library(tidyverse)

# number of observations in each sample
n_obs <- 250

rmse <- 1
x1 <- rep(c(0,1), each=n_obs/2)
x2 <- runif(n_obs, min=0, max=5)
beta0 <- -1
beta1 <- 0.5
beta2 <- 1.5

# for scenario 1, where equal var assumption is met (sd is constant value, rmse)
y1 <- beta0 + beta1*x1 + beta2*x2 + rnorm(n=n_obs, mean=0, sd=rmse)
# for scenario 2, where equal var assumption is violated (sd depends on x2)
y2 <- beta0 + beta1*x1 + beta2*x2 + rnorm(n=n_obs, mean=0, sd=rmse + x2)

# for scenario 1
mod1 <- lm(y1 ~ x1 + x2)
# for scenario 2
mod2 <- lm(y2 ~ x1 + x2)

summary(mod1)$coeff["x1","Estimate"]
```

```
## [1] 0.691144
```

```r
# repeatedly generate data, fit the model, and extra the beta1 coefficient (1,000 times)
# number of simulations
n_sim <- 1000

# target visualization: sampling distribution of \hat{beta}_1
#                 (histogram or density plot of \beta_1 estimates), by scenario
# target summary numbers: mean and sd/variance of beta_1 estimates, by scenario

# loop through iterations

# create target visualization

# create target summaries
```

CHOOSE ONE OF 2 (Clustering), 3 (Simulations) or 4 (SQL) to COMPLETE

4. SQL

Airline flights

4a. Identify what years of data are available in the `flights` table of the airlines database using SQL code.

ANSWER:

```
library(RMySQL)
```

```
## Loading required package: DBI
```

```
library(mdsr)
# dbConnect_scidb is accesible from the mdsr package
aircon <- dbConnect_scidb("airlines")

# can use SHOW and EXPLAIN commands to explore what tables are available
# through this connection, and what variables/fields are in each table
dbGetQuery(aircon, "SHOW TABLES")
```

```
##   Tables_in_airlines
## 1           airports
## 2           carriers
## 3            flights
## 4             planes
```

```
dbGetQuery(aircon, "EXPLAIN airports")
```

```
##     Field          Type Null Key Default Extra
## 1     faa    varchar(3)   NO PRI
## 2    name  varchar(255)  YES        <NA>
## 3     lat decimal(10,7)  YES        <NA>
## 4     lon decimal(10,7)  YES        <NA>
## 5     alt       int(11)  YES        <NA>
## 6      tz   smallint(4)  YES        <NA>
## 7     dst       char(1)  YES        <NA>
## 8    city  varchar(255)  YES        <NA>
## 9 country  varchar(255)  YES        <NA>
```

```
# can view first few obs of a table to see what the fields look like
dbGetQuery(aircon, "SELECT *
                    FROM airports
                    LIMIT 0,5")
```

```
## Warning in .local(conn, statement, ...): Decimal MySQL column 2 imported as
## numeric
```

```
## Warning in .local(conn, statement, ...): Decimal MySQL column 3 imported as
## numeric
```

```
##   faa                          name      lat       lon  alt tz dst
## 1 04G             Lansdowne Airport 41.13047 -80.61958 1044 -5   A
## 2 06A Moton Field Municipal Airport 32.46057 -85.68003  264 -6   A
## 3 06C           Schaumburg Regional 41.98934 -88.10124  801 -6   A
## 4 06N               Randall Airport 41.43191 -74.39156  523 -5   A
## 5 09J         Jekyll Island Airport 31.07447 -81.42778   11 -5   A
##            city       country
## 1    Youngstown United States
## 2      Tuskegee United States
## 3    Schaumburg United States
## 4    Middletown United States
## 5 Jekyll Island United States
```

4b. How many domestic flights flew into Dallas-Fort Worth (DFW) on May 14, 2010? Use SQL to compute this number. (You can use R code to check it, if you wish.)

ANSWER:

4c. *Among the flights that flew into Dallas-Fort Worth (DFW) on May 14, 2010*, compute (using SQL) the number of flights and the average arrival delay time for each airline carrier. Among these flights, how many carriers had an average arrival delay of 60 minutes or longer?

ANSWER: