STAT 231: Problem Set 9B

Jack Dove

due by 5 PM on Friday, November 13

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. 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.

> ANSWER: THe first cluster has very few home runs and RBIs (representative of power at the plate), but a medium amount of hits and a lot of stolen bases, meaning cluster 1 is filled with speed-focused players. THe second group is a group of extremely strong hitters with low on base percentages: they hit for a lot of homeruns and RBIs, but not a lot of stolen bases or hits. The third group would be considered the "best" statistically, as they on average stole lots of bases, and hit the most RBIs, HRs, and hits of any group.

```r
##### 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)

vars <- c("tH", "tHR", "tRBI", "tSB")

cluster_analysis <- kmeans(hof[,vars], centers=3, nstart=20)
```

2. MDSR Exercise 10.6

Equal variance assumption: What is the impact of the violation of the equal variance assumption for linear regression models? Repeatedly generate data from a "true" model given by the following code. (Note that 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.)

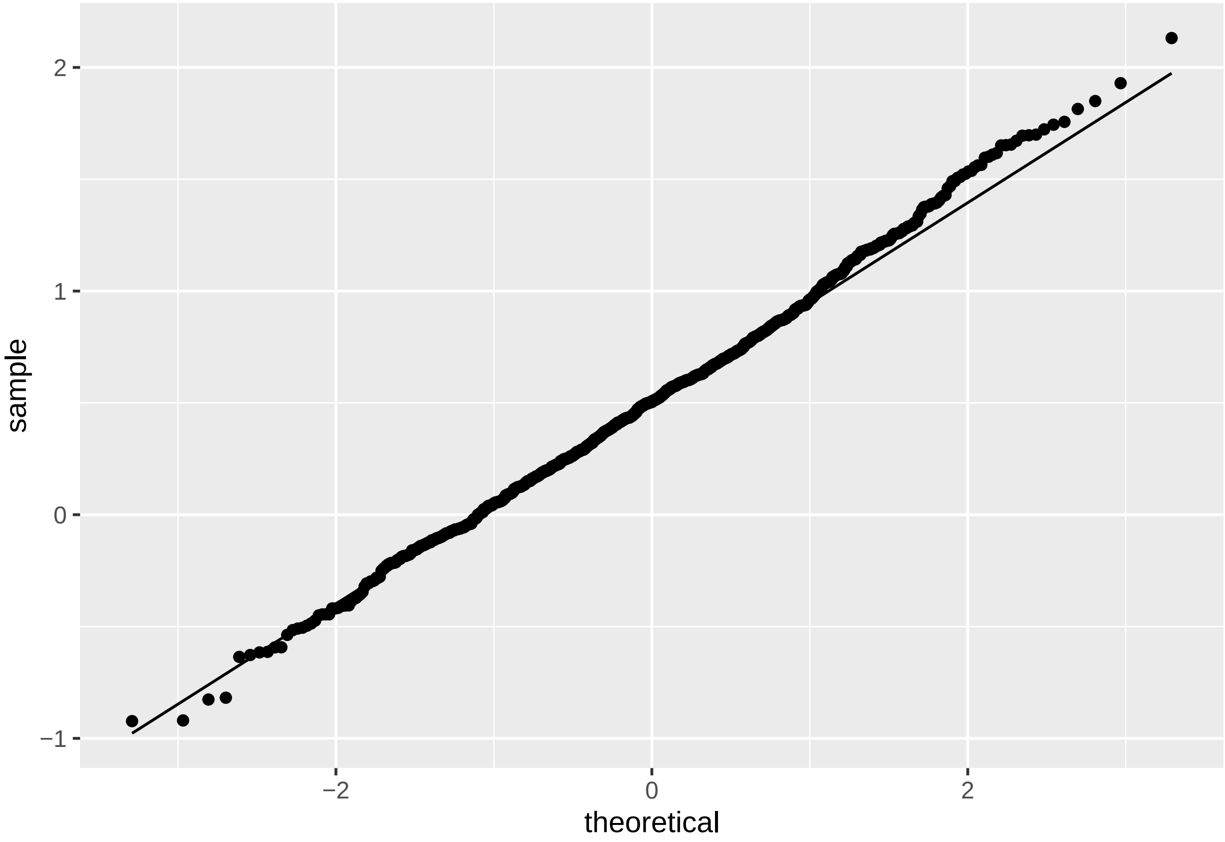
For each simulation, fit the linear regression model and display the distribution of 1,000 estimates of the β_1 parameter. Does the distribution of the estimates follow a normal distribution?

> ANSWER: Yes: the histogram and qq plot both display an incredibly normal distribution. Therefore, the effect on the beta coefficient wasn't seen (the distribution was still centered on 0.5).

```r
# number of observations in each sample
n_obs <- 250

# parameters held constant
rmse <- 1
beta0 <- -1
beta1 <- 0.5
beta2 <- 1.5

# how to generate data
x1 <- rep(c(0,1), each=n_obs/2)
x2 <- runif(n_obs, min=0, max=5)
y <- beta0 + beta1*x1 + beta2*x2 + rnorm(n=n_obs, mean=0, sd=rmse + x2)

# fit model
mod <- lm(y ~ x1 + x2)

# extract beta1 coefficient
summary(mod)$coeff["x1","Estimate"]
```

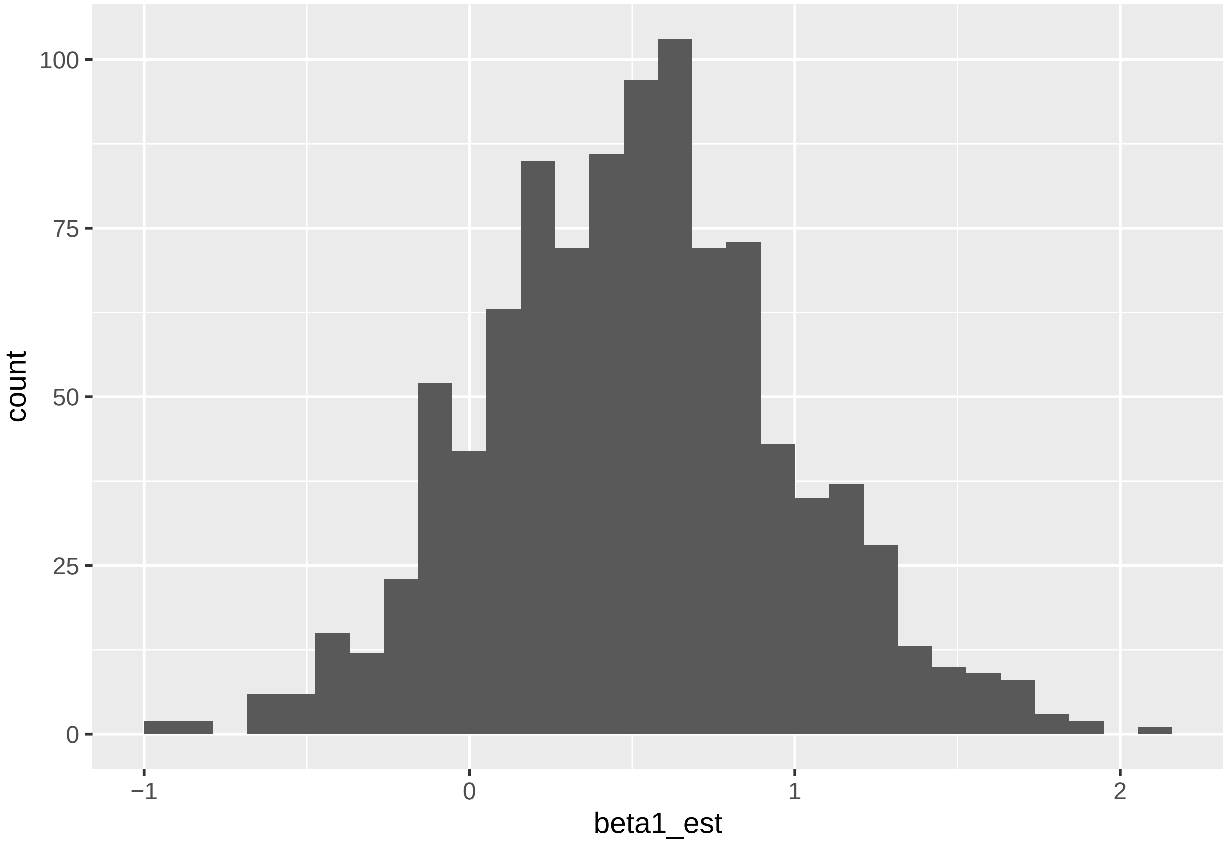
```
## [1] 0.02016556
```

```r
# now, write code to repeatedly generate data, fit the model, and extract the beta coefficient (1,000 t

# number of simulations
n_sim <- 1000

#Simulation of typical model
beta1_est <- array(NA, dim=c(n_sim,1))
for (i in 1:n_sim) {
rmse <- 1
beta0 <- -1
beta1 <- 0.5
beta2 <- 1.5
x1 <- rep(c(0,1), each=n_obs/2)
x2 <- runif(n_obs, min=0, max=5)
y <- beta0 + beta1*x1 + beta2*x2 + rnorm(n=n_obs, mean=0, sd=rmse + x2)
mod <- lm(y ~ x1 + x2)
# extract beta1 coefficient
beta1_est[i]  <- summary(mod)$coeff["x1","Estimate"]
}

# QQ Plot of typical
ggplot(data = as.data.frame(beta1_est), aes(sample = beta1_est)) +
  stat_qq() +
  stat_qq_line()
```



```r
# Histogram of typical
ggplot(data = as.data.frame(beta1_est), aes(sample = beta1_est, x=beta1_est)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


count
100
75
50
25
0
−1
0
1
2
beta1_est

3. MDSR Exercise 10.7

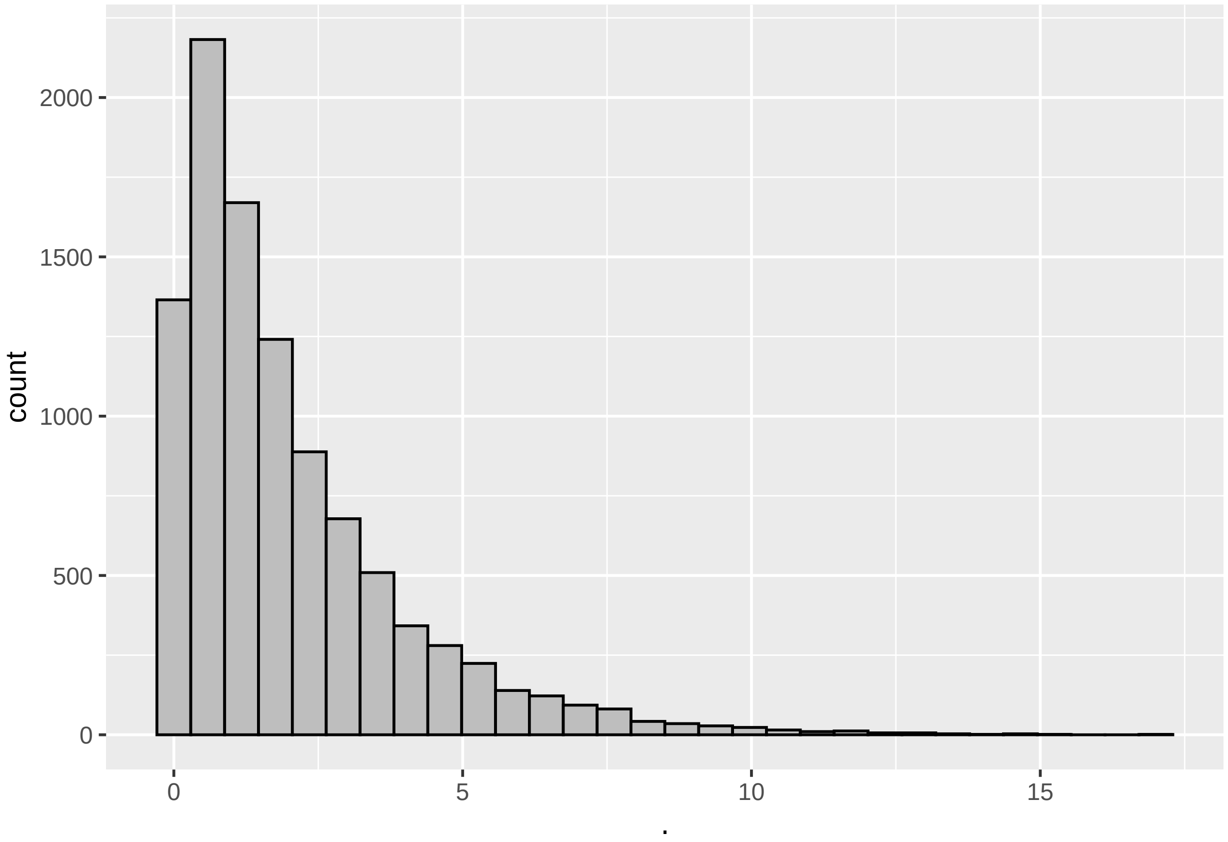
Skewed residuals: What is the impact if the residuals from a linear regression model are skewed (and not from a normal distribution)? Repeatedly generate data from a "true" model given by the parameters below.

For each simulation, fit the linear regression model and display the distribution of 1,000 estimates of the β_1 parameter.

> ANSWER: The histogram and qq plot both display an incredibly normal distribution. Therefore, the effect on the beta coefficient wasn't seen (the distribution was still centered on 0.5).

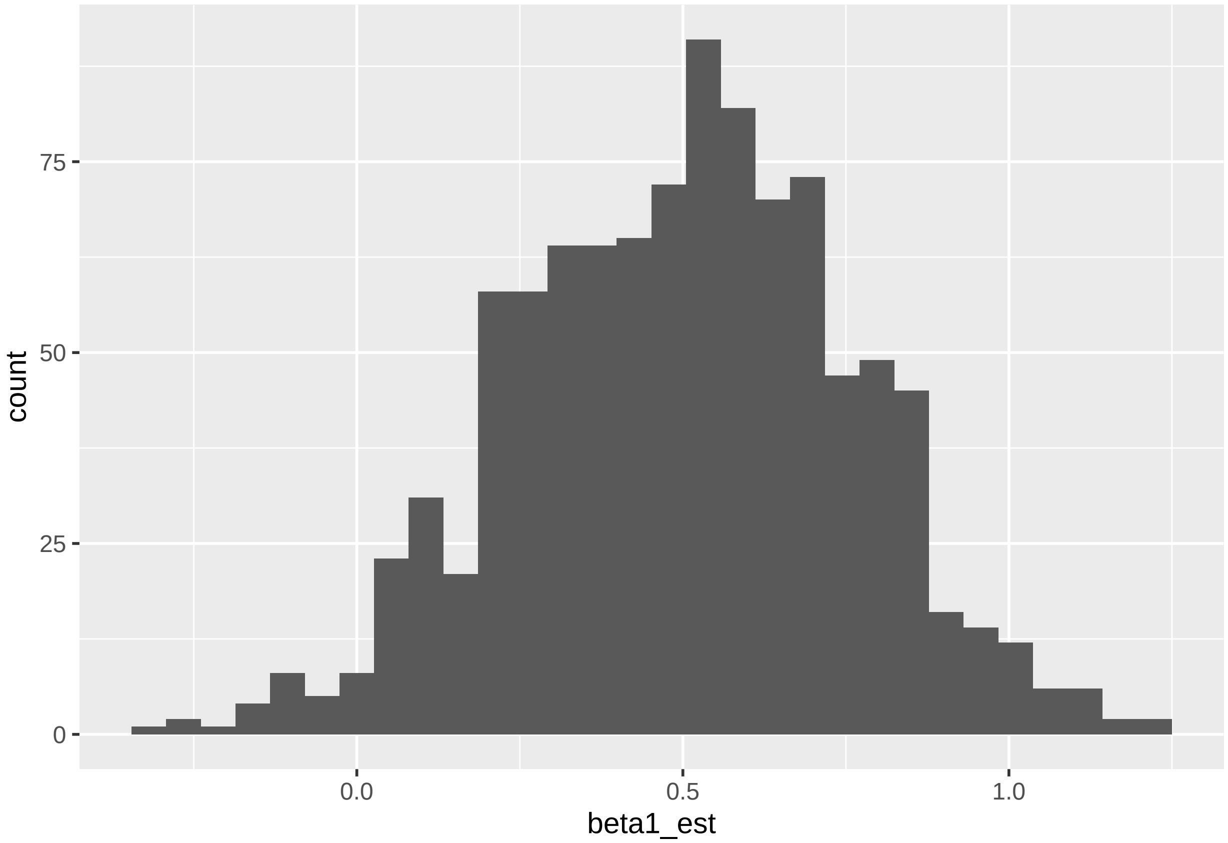
```
# number of observations in each sample
n_obs <- 250

# parameters held constant
rmse <- 1
beta0 <- -1
beta1 <- 0.5
beta2 <- 1.5

# how to generate data
x1 <- rep(c(0,1), each=n_obs/2)
x2 <- runif(n_obs, min=0, max=5)
y <- beta0 + beta1*x1 + beta2*x2 + rexp(n=n_obs, rate=1/2)

beta1_est <- array(NA, dim=c(n_sim,1))
for (i in 1:n_sim) {
rmse <- 1
beta0 <- -1
beta1 <- 0.5
beta2 <- 1.5
x1 <- rep(c(0,1), each=n_obs/2)
x2 <- runif(n_obs, min=0, max=5)
y <- beta0 + beta1*x1 + beta2*x2 + rexp(n=n_obs, rate=1/2)
mod <- lm(y ~ x1 + x2)
# extract beta1 coefficient
beta1_est[i]  <- summary(mod)$coeff["x1","Estimate"]
}

# what does an exponential dist'n with rate = 1/2 look like?
# very skewed!
rexp(n=10000, rate=1/2) %>%
  as.data.frame() %>%
  ggplot(aes(x=`.`)) +
    geom_histogram(color="black", fill="grey")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

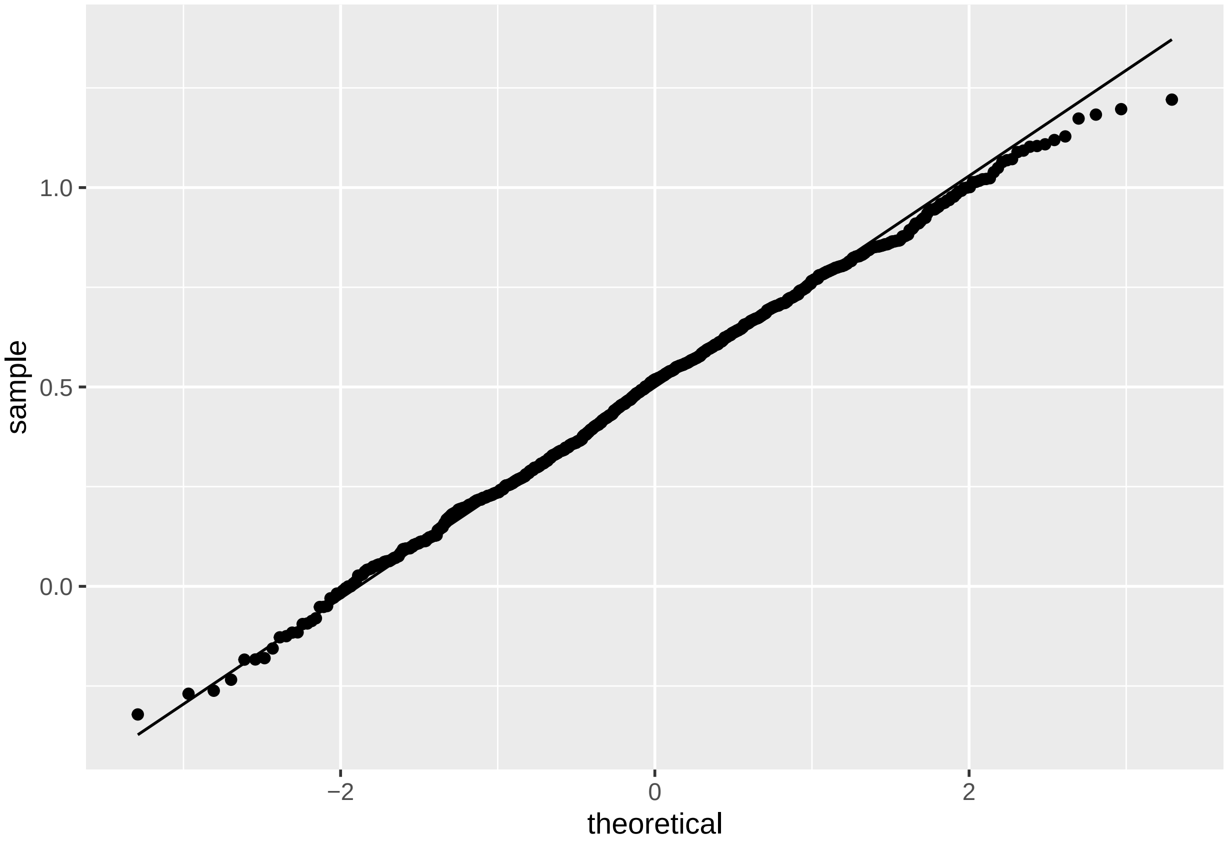


```r
#My distribution display
ggplot(data = as.data.frame(beta1_est), aes(sample = beta1_est, x=beta1_est)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
# QQ Plot
ggplot(data = as.data.frame(beta1_est), aes(sample = beta1_est)) +
  stat_qq() +
  stat_qq_line()
```


sample
1.0
0.5
0.0
−2
0
2
theoretical