Homework 3

Ashutosh Ekade

Question 1: Variance-Covariance Matrix of the OLS estimator vector

Suppose $X = [1, X_1, X_2]'$. Let $X_1 \sim N(3,4)$ and $X_2 \sim N(2,6)$ where the notation $N(\mu, \sigma^2)$ indicates a univariate normal distribution with mean μ and variance σ^2 (ie, the two variances are 4 and 6). Further suppose X_1 and X_2 are independent. Let $Y = X'\beta + e$ where $\beta = [5, 0.4, 0.2]$ and $e \sim N(0, 10)$. Assume $\mathbb{E}[e|X] = 0$.

1.a) Calculate the 3 by 3 matrix $Q_{XX} = \mathbb{E}[X_i X_i']$ in LaTeX. You may need to use the fact that var$(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2$.

$$\mathbf{XX}' = \begin{bmatrix} 1 \\ X_1 \\ X_2 \end{bmatrix} \begin{bmatrix} 1 & X_1 & X_2 \end{bmatrix} = \begin{bmatrix} 1 & X_1 & X_2 \\ X_1 & X_1^2 & X_1X_2 \\ X_2 & X_1X_2 & X_2^2 \end{bmatrix}$$

$\mathbf{E[X_1]} = 3$ $\mathbf{E[X_2]} = 2$ $\mathbf{E[X_1^2]} = \mathbf{Var(X_1)} + [\mathbf{E[X_1]}]^2 = 4 + 3^2 = 13$ $\mathbf{E[X_2^2]} = \mathbf{Var(X_2)} + [\mathbf{E[X_2]}]^2 = 6 + 2^2 = 10$ $\mathbf{E[X_1X_2]} = \mathbf{E[X_1]E[X_2]} = 3 * 2 = 6$ $\mathbf{Q_{XX}} = \begin{bmatrix} 1 & 3 & 2 \\ 3 & 13 & 6 \\ 2 & 6 & 10 \end{bmatrix}$

1.b) Set the seed to 1234. The use the `simdat()` function to simulate a dataset with $n = 100$ observations. Calculate $\hat{Q}_{XX} = \frac{1}{n} X'X$. Round the values in the resulting matrix to two decimal places. Is $\hat{Q}_{XX}$ close to Q_{XX}?

```
simdat <- function(mu, sd, rho, n, beta, sigma) {
    cv <- rho*sd[1]*sd[2]
    Sigma <- matrix(c(sd[1]^2, cv, cv, sd[2]^2), 2, 2)
    x <- MASS::mvrnorm(n, mu, Sigma)
    y <- cbind(1, x) %*% beta + rnorm(n, 0, sigma)
    return(data.frame(y, x1=x[,1], x2=x[,2]))
}
```

```
mu_1 <- c(3,2)
sd_1 <- c(2,sqrt(6))
rho_1 <- 0
beta_1 <- c(5,0.4,0.2)
sigma_1 <- sqrt(10)
set.seed(1234)
data_1b <- simdat(mu_1,sd_1,rho_1,100,beta_1,sigma_1)
x_1b <- cbind(1,data_1b['x1'],data_1b['x2'])
Q__xx_b <- t(x_1b)%*%as.matrix(x_1b)/100
Q__xx_b
```

```
          1        x1       x2
1  1.000000  2.917514 1.616014
x1 2.917514 12.730912 4.842371
x2 1.616014  4.842371 8.603951
```

$\hat{Q}_{XX}$ is close to Q_{XX}

1.c) Set the seed to 2345. The use the `simdat()` function to simulate a dataset with $n = 1,000$ observations. Calculate $\hat{Q}_{XX} = \frac{1}{n}X'X$. Round the values in the resulting matrix to two decimal places. Is $\hat{Q}_{XX}$ close to Q_{XX}?

```
set.seed(2345)
data_1c <- simdat(mu_1,sd_1,rho_1,1000,beta_1,sigma_1)
x_1c <- cbind(1,data_1c['x1'],data_1c['x2'])
Q__xx_c <- t(x_1c)%*%as.matrix(x_1c)/1000
Q__xx_c
```

```
          1        x1       x2
1  1.000000  3.003920 1.903861
x1 3.003920 12.733160 5.730567
x2 1.903861  5.730567 9.474682
```

$\hat{Q}_{XX}$ even more closer Q_{XX}.

1.d) Input the matrix Q_{XX} from 1(a) above as an object in R. Then calculate in R $V_{\hat{\beta}} = \sigma^2 Q_{XX}^{-1}$. Round the values in the resulting matrix to two decimal places.

```r
Q_xx <- matrix(c(1, 3, 2, 3, 13, 6, 2, 6, 10), nrow=3)
V_beta <- round(10 * solve(Q_xx), 2)
V_beta
```

```
      [,1] [,2]  [,3]
[1,] 39.17 -7.5 -3.33
[2,] -7.50  2.5  0.00
[3,] -3.33  0.0  1.67
```

1.e) Use the data from 1(c) above to calculate $\hat{V}_{\hat{\beta}} = s^2(X'X)^{-1}$. Round the values in the resulting matrix to four decimal places.

```r
model <- lm(y~x1+x2,data=data_1c)
s_sq <- sum(residuals(model)^2)/(1000-3)
V_b_h <- round(solve(t(x_1c)%*%as.matrix(x_1c))*s_sq,4)
V_b_h
```

```
         1      x1      x2
1   0.0402 -0.0080 -0.0032
x1 -0.0080  0.0027  0.0000
x2 -0.0032  0.0000  0.0017
```

1.f) Mulitply n times $\hat{V}_{\hat{\beta}}$ from 1(e) above to approximate V_{β}? Round the values in the resulting matrix to two decimal places. Is this close to $V_{\hat{\beta}}$ in 1(d) above?

```r
V_b_h * 1000
```

```
      1   x1   x2
1  40.2 -8.0 -3.2
x1 -8.0  2.7  0.0
x2 -3.2  0.0  1.7
```

Yes this is much closer to $V_{\hat{beta}}$ in 1.d.

Question 2: Confidence Intervals

Use the function `simdat()` to simulate a dataset with `mu=c(1,4)`, `sd=c(1,2)`, `rho=0.3`, `n=100`, `beta=c(0.5, 1.5, 3.0)`, and `sigma=2`. Denote the 3 elements of the β vector as $\beta = [\beta_0, \beta_1, \beta_2]'$. Regress y on $x1$ and $x2$. Assuming homoskedasticity. Choose $\alpha = 0.20$. Calculate:

1. **the 80% confidence interval for β_1 (the slope coefficient on x_1), and**
2. **the test statistic $T(1.5) = (\hat{\beta}_1 - 1.5)/s(\hat{\beta}_1)$**

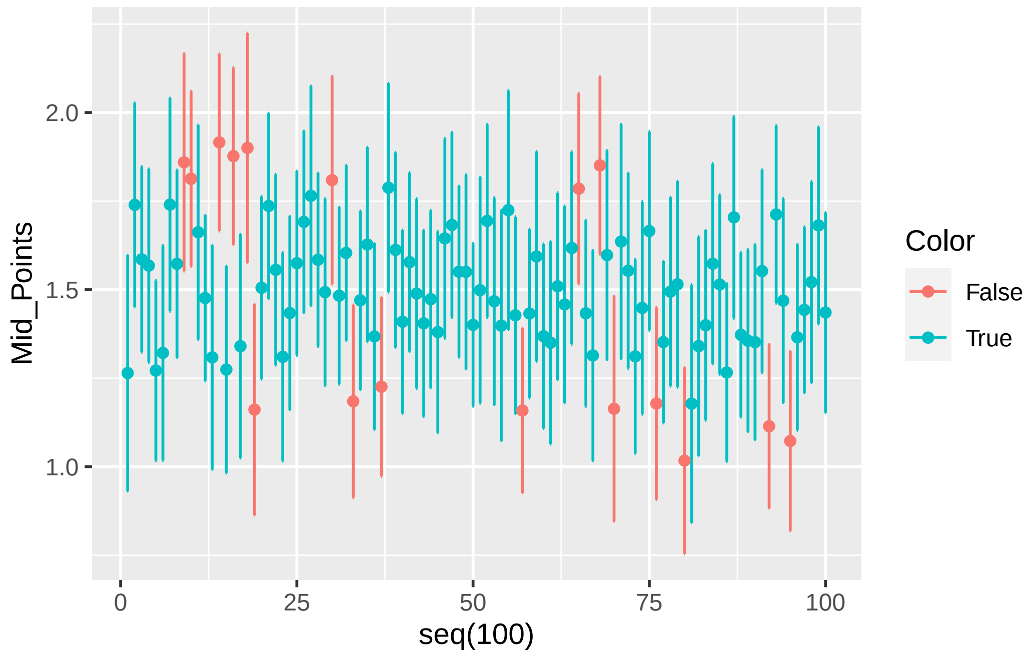
Repeat this process 100 times, calculating $CI^{(1)}, \dots, CI^{(100)}$ and $T^{(1)}(1.5), \dots, T^{(100)}(1.5)$.

```r
mu_2 <- c(1,4)
sd_2 <- c(1,2)
rho_2 <- 0.3
beta_2 <- c(0.5, 1.5, 3.0)
sigma_2 <- 2
lower <- c()
upper <- c()
t_vals <- c()
for (i in seq(100)){
  data_2 <- simdat(mu_2,sd_2,rho_2,100,beta_2,sigma_2)
  model_2 <- lm(y~x1+x2,data=data_2)
  beta1_hat <- coefficients(model_2)[2]
  sd_beta1_hat <- summary(model_2)$coefficients[2,2]
  alpha <- 0.2
  t_critical <- qt(1 - alpha/2, 100-3)
  lower <- c(lower,beta1_hat-sd_beta1_hat*t_critical)
  upper <- c(upper,beta1_hat+sd_beta1_hat*t_critical)
  t_val <- (beta1_hat-1.5)/sd_beta1_hat
  t_vals[i] <- t_val
}
cis <- data.frame(lower=lower,upper=upper)
```

2.a) Plot the 100 confidence intervals. You can use slide 27 from class 4 as an example. You may find `geom_errorbar()` to be helpful.

```r
library(ggplot2)
cis$Color <- ifelse(cis$lower <= 1.5 & cis$upper >= 1.5, "True", "False")
for (i in seq(100)){
  cis$Mid_Points[i]<-(cis$lower[i]+cis$upper[i])/2 }
```

```r
ggplot(cis,aes(x=seq(100),col=Color))+
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.2)+
  geom_point(aes(y=Mid_Points))
```



2.b) What proportion of the confidence intervals contain the true value of $\beta_1 = 1.5$?

About 80% of the confidence intervals contain the true value of $\beta_1 = 1.5$.

2.c) What proportion of the 100 test statistics lead you to reject the Null Hypothesis that $\beta_1 = 1.5$ at the $\alpha = 0.20$ level?

All (100%) of the test stats lead us to reject the Null Hypothesis that $\beta_1 = 1.5$ at the $\alpha = 0.20$ level.

Question 3: Hypothesis Tests

This question picks up where Question 4 left off from HW2.

Load the `Hitters` dataset from the `ISLR` package. Drop any rows where Salary is `NA`. Assume the model is $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + e$ where Y denotes `Salary`, X_1 denotes `Hits`, and X_2 denotes `Years`. See `?ISLR::Hitters` for definitions of these variables.

```
data(Hitters, package="ISLR")
Hitters <- Hitters[!is.na(Hitters$Salary), ]
```

3.a) By hand, calculate the t-statistic and associated p-value for the Hypothesis the $\hat{\beta}_0 = 0$ under the assumption that the errors are normally distributed.

```
x_3 <- cbind(1, Hitters$Hits, Hitters$Years)
y_3 <- as.matrix(Hitters$Salary, ncol=1)
df <- nrow(y_3) - ncol(x_3)
xxi_3 <- solve(t(x_3)%*%x_3)
betahat_3 <- xxi_3%*%crossprod(x_3, y_3)
y_3_fit <- x_3%*%betahat_3
e_3_hat <- y_3-y_3_fit
s2_3 <- sum(e_3_hat^2)/df
v_3 <- s2_3*xxi_3
sbeta_3 <- sqrt(diag(v_3))
t_3 <- as.vector(betahat_3)/sbeta_3
p_3 <- 2 * (1 - pt(t_3, df=df))
cat("t-statistics:", t_3,'\n')
```

```
t-statistics: -2.953224 8.603116 7.830537
```

```
cat("p-values:", p_3,'\n')
```

```
p-values: 1.996567 8.881784e-16 1.234568e-13
```

3.b) By hand, calculate the z-statistic and associated p-value for the Hypothesis the $\hat{\beta}_0 = 0$ stemming from the asymptotic Normal distribution of the coefficient estimates (ie, when we do not assume the errors are normally distributed). Assume a homoskedastic linear CEF model.

```r
z_3 <- betahat_3/sbeta_3
p_3b <- 2*(1-pnorm(z_3))
cat("z-statistics:", z_3,'\n')
```

```
z-statistics: -2.953224 8.603116 7.830537
```

```r
cat("p-values:", p_3b,'\n')
```

```
p-values: 1.996855 0 4.884981e-15
```

3.c) By hand, calculate the z-statistic and associated p-value for the Hypothesis the $\hat{\beta}_0 = 0$ stemming from the asymptotic Normal distribution of the coefficient estimates (ie, when we do not assume the errors are normally distributed). Assume a heteroskedastic linear CEF model. Use $V_{\hat{\beta}}^{\textbf{HC1}}$ as your estimator of $V_{\hat{\beta}}$.

```r
u <- x_3*(e_3_hat%*%matrix(1,ncol=3))
v_3_hc1 <- xxi_3%*%(t(u)%*%u)%*%xxi_3*(nrow(y_3)/df)
sbeta_3_het <- sqrt(diag(v_3_hc1))
z_3_het <- betahat_3/sbeta_3_het
p_asy_het <- 2*(1-pnorm(z_3_het))
cat("z-statistics:", z_3_het,'\n')
```

```
z-statistics: -2.060441 5.712599 7.537414
```

3.d) By hand, calculate the F-statistic and associated p-value for the Hypothesis that both slope coefficients equal zero.

```r
sst <- sum((y_3-mean(y_3))^2)/nrow(y_3)
sse <- sum(e_3_hat^2)/nrow(y_3)
rsq <- 1 - sse/sst
f_3 <- (rsq/(ncol(x_3)))/((1-rsq)/df)
p_f <- 1-pf(f_3,ncol(x_3),df)
cat("f-statistics:", f_3,'\n')
```

```
f-statistics: 45.96147
```

```
cat("p-values:", p_f,'\n')
```

```
p-values: 0
```

3.e) By hand, test the linear hypothesis that $6\beta_1 = \beta_2$ at the $\alpha = 0.05$ confidence level. Assume errors are normally distributed.

```
se_5e <- sqrt(36*v_3[2,2]+v_3[3,3]-12*v_3[2,3])
t_3e <- (6*betahat_3[2]-betahat_3[3])/se_5e
tc_3e <- qt(1-0.05/2,df)
abs(t_3e)>tc_3e
```

```
[1] FALSE
```

Fail to reject under two-tailed test.

3.f) By hand, test the joint set of linear hypotheses that $\beta_1 = 5$ & $\beta_2 = 30$ at the $\alpha = 0.05$ confidence level. Assume errors are normally distributed.

```
c_3 <- matrix(c(0,0,1,0,0,1),nrow=2)
value_3 <- c(5,30)
sse_r <- sse + t(c_3%*%betahat_3-value_3)%*%(c_3%*%xxi_3%*%t(c_3))%*%(c_3%*%betahat_3-valu
f_3f <- ((sse_r-sse)/2)/(sse/df)
abs(f_3f) > qf(1-0.05/2,2,df)
```

```
      [,1]
[1,] FALSE
```

Failed to reject under f-test.

3.g) Calculate the leverage values for each observation in the dataset. Which data point has the highest leverage value?

```
h_3 <- x_3%*%xxi_3%*%t(x_3)
index <- which(h_3==max(h_3),arr.ind=TRUE)
cat('Data point with the highest leverage value: ', x_3[index[1],2:3], '\n')
```

```
Data point with the highest leverage value:  52 24
```

Question 4: Categorical Variables

Load the `diamonds` dataset from the `ggplot2` package. Ensure clarity is a factor (not an ordered factor) variable. Clarity ranges from Included through (Very) (Very) Slightly Included to Internally Flawless. You can learn more about diamond clarity here.

```
library(tidyverse)
data(diamonds, package="ggplot2")
diamonds <- diamonds |> mutate(clarity=factor(clarity, ordered=FALSE))
```

4.a) Regress log(price) on log(carat) and clarity. How do you interpret the coefficient estimate on the row labeled `claritySI1`?

```
model <- lm(log(price) ~ log(carat) + clarity, data=diamonds)
summary(model)
```

```
Call:
lm(formula = log(price) ~ log(carat) + clarity, data = diamonds)

Residuals:
     Min       1Q   Median       3Q      Max
-0.97521 -0.12085  0.01048  0.12561  1.85854

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 7.768115   0.006940 1119.25   <2e-16 ***
log(carat)  1.806424   0.001514 1193.23   <2e-16 ***
claritySI2  0.479658   0.007217   66.46   <2e-16 ***
claritySI1  0.624558   0.007163   87.19   <2e-16 ***
clarityVS2  0.775248   0.007197  107.72   <2e-16 ***
clarityVS1  0.820461   0.007306  112.30   <2e-16 ***
clarityVVS2 0.979221   0.007529  130.05   <2e-16 ***
clarityVVS1 1.028298   0.007745  132.77   <2e-16 ***
clarityIF   1.114625   0.008376  133.07   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1888 on 53931 degrees of freedom
Multiple R-squared:  0.9654,    Adjusted R-squared:  0.9654
```

```
F-statistic: 1.879e+05 on 8 and 53931 DF,  p-value: < 2.2e-16
```

The coefficient estimate for the row labelled `claritySI1` is around 0.625 which means it has a positive effect on the price on the diamond and is similar to the effect of the all the variants of the similar group.

4.b) What is the average price of a diamond with clarity I1 and strictly (ie, $>$ not $\geq$) between 1.25 and 1.5 carats? What is the average price of a diamond with clarity SI1 and size strictly between 1.25 and 1.5 carats? What is the ratio of these two averages? How does this ratio compare to coefficient estimate described in 4(a) above?

```
diamonds_I1 <- diamonds %>%
filter(clarity == "I1" & carat > 1.25 & carat < 1.5)
price_I1 <- mean(diamonds_I1$price)
diamonds_SI1 <- diamonds %>%
filter(clarity == "SI1" & carat > 1.25 & carat < 1.5)
price_SI1 <- mean(diamonds_SI1$price)
cat("The average price of a diamond with clarity I1 and strictly between 1.25 and 1.5 cara
```

```
The average price of a diamond with clarity I1 and strictly between 1.25 and 1.5 carats: 446
```

```
cat("The average price of a diamond with clarity SI1 and size strictly between 1.25 and 1.
```

```
The average price of a diamond with clarity SI1 and size strictly between 1.25 and 1.5 carat
```

```
cat("Ratio of these prices: ", price_I1 / price_SI1, '\n')
```

```
Ratio of these prices:  0.6411698
```

This ratio is very close to the coefficient of ‘claritySI1’ category mentioned in 4.a.

4.c) According to the fitted linear regression model, what is the expected price of a 1.5 carat diamond with VS1 clarity? Use the `predict()` function in R to answer this question.

```
df_vs1 <- data.frame(carat=1.5,clarity='VS1')
exp(predict(model,df_vs1))
```

```
       1
11170.34
```

This is the expected price of a diamond of clarity VS1 according to the fitted LR model.

4.d) What is the average price of a diamond in the dataset with VS1 clarity and size strictly between 1.4 and 1.6 carats? How many diamond's prices are included in this average?

```
diamonds_VS1 <- diamonds %>%
filter(clarity == "VS1" & carat > 1.4 & carat < 1.6)
price_VS1 <- mean(diamonds_VS1$price)
nums <- nrow(diamonds_VS1)
cat("The average price of a diamond in the dataset with VS1 clarity and size strictly betw
```

```
The average price of a diamond in the dataset with VS1 clarity and size strictly between 1.4
```

```
cat("Number of dimaond's prices which were included in this average: ", nums)
```

```
Number of dimaond's prices which were included in this average:  467
```

Question 5: Omitted Variable Bias

Let Y denote the price of a used car (in dollars),let X_1 denote the mileage of the car (ie, the total number of miles the car has ever been driven), and let X_2 denote the age of the car (in years). Suppose we are interested in estimating the effect of mileage on price. We have data on mileage and price of 1,000 recently sold used cars; we do not have data on the age of the cars.

The model we would like to estimate is $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + e$.

The model we can feasibly estimate is $Y = \gamma_0 + \gamma_1 X_1 + u$.

5.a) What relationship (positive, negative, none, or unknown) do you expect between X_2 and X_1? Why?

If X_1 represents the total distance driven so far then I expect it to have a positive relationship with X_2 because longer the age, more distance it is expected to have been driven for.

5.b) What relationship (positive, negative, none, or unknown) do you expect between X_2 and Y? Why?

A negative relationship is expected between Price and Age, because older cars are expected to be sold for lesser price than newer ones with more features and latest tech.

5.c) How does the relationship between X_1 and Y that you are able to estimate (γ_1) compare to the relationship you wish you could estimate β_1)? Why?

Because X_2 is positively correlated with X_1 its effect will be included when we account for just X_1.