STA302H1F/STA1001HF: Mini Project 2
Due on 7th June, 2020 11:59 PM Sharp on Quercus

The purpose of these mini projects is to deepen your understanding of linear regression properties and develop your data analysis skills (which will be useful for the final project and future courses). Also, the emphasis will be on `R` coding, which you learned during lecture5-lecture10. This is again a simulation-based study. Here we are trying to focus on correlated predictors and interaction models. Recall that:

- To submit your results, you will be required to prepare a 5 minute presentation that you will need to record (using your computer, phone, etc.). You will be required to display your T-card alongside your face at the beginning of your video to verify your identity.
- You will be required to submit the R codes that you have created in a separate file. This helps us to check the reproducibility of your codes.
- You will need to display the results of your project in a logical way using slides (e.g. PowerPoint, latex, R Markdown or other) and record yourself discussing these results, with a focus on why you chose to do certain things and interpretation of your results for non-statisticians.
- Presentations should be submitted on time (i.e. by the deadline). Late submissions will receive a 20% penalty for each day that the project is late.
- In general, extensions will <u>not</u> be given unless a valid reason is provided. In such cases, the instructor may decide to grant an extension of up to 5 days.
- **There are no make-up mini projects**. A missed mini project will be given a grade of 0.

Project Description: In multiple linear regression we have more than one predictors.In this project we are interested in two aspects of the predictor,

1. What happens when the predictors are correlated
2. How do model diagnostics work when the true model is non linear?

Task 1: In the first task you need to simulate a dataset for multiple linear regression. The dataset will consist of one outcome variable (Y) and three predictor variabls ($\mathbf{X} = (X_1, X_2, X_3)$). The $\mathbf{X}$ has to be simulated from a multivariate normal distribution. You can use the following simulation codes (Note: these are just initial codes)

```
library(MASS)

## Simulation for correlated predictors ##
set.seed(1002656486)

nsample <- 10; nsim <- 100
sig2 <- rchisq(1, df = 1) ## The true error variance
bet <- c(rnorm(3, 0, 1), 0) ## 4 values of beta that is beta0, beta1, beta2, beta3 = 0
```

```
muvec <- rnorm(3, 0, 1)
sigmat <- diag(rchisq(3, df = 4))
X <- mvrnorm(nsample, mu = muvec, Sigma = sigmat)
Xmat <- cbind(1, X)

## Simulate the response ##
bets <- matrix(NA, ncol = length(bet), nrow = nsim)
for(i in 1:nsim){
Y <- Xmat%*%bet + rnorm(nsample, 0, sig2)
model1 <- lm(Y ~ X)
bets[i,] <- coef(model1)
}
```

The provided seed is my student ID. Please change the seed to your student ID

There are few thigs to be noticed from simulations. The $\boldsymbol{\beta}$ has four values, $\beta_0, \beta_1, \beta_2$ and β_3. You can see that $\beta_3 = 0$, i.e., the third predictor is not linearly related with the response. Here `sigmat` is the variance-covariance matrix, where the diagonal elements are variances (not standard errors) and the off diagonals are covariances.

1. First assume that the correlation between the three predictors are zero, i.e., the off diagonals of `sigmat` are zero. Set the number of simulations `nsim = 100`. Generate Y for each simulation. Then run simple linear regression for each of the three variables seperately. Obtain the regression parameter estimates and their variances from the coefficients tables obtained from the `lm` function. Comment on whether the estimates are unbiased. Also, comment on their variances.

2. Now fit a multiple linear regression and obtain the regression parameter estimates along with their variances from each simulation. Again check the unbiasedness and the variances. Compare the results with step 1.

3. Now assume X_1 and X_2 are correlated. You can select a value for correlation (e.g., $r_{12} = 0.2$). Then add the following covariance terms in the `sigmat` matrix,

   ```
   ## The correlation ##
   r12 <- 0.2
   sigmat[1,2] <- sigmat[2,1] <- r12*sqrt(sigmat[1,1])*sqrt(sigmat[2,2])
   ## Simulation for Categorical Variables with Interaction ##
   set.seed(1002656486)
   X <- mvrnorm(nsample, mu = muvec, Sigma = sigmat); cor(X[,1], X[,2])
   Xmat <- cbind(1, X)
   ```

   Again run simple linear regressions on each of the predictors and also a multiple linear regression. Compare the results with step 1 and 2 and comment on the differences/similarities between the results. Start increasing the value of the correlation coefficient r_{12}, (e.g., 0.5, 0.7, 0.8 etc.) and again perform step 1 and 2. How do the estimates and standard error of β_1 and β_2 change for simple and multiple linear regressions as the correlation changes?

4. Now assume X_1 and X_2 are uncorrelated, i.e., $r_{12} = 0$ and `sigmat[1,2]` = `sigmat[2,1]` = 0. Instead X_1 and X_3 are correlated. Select a value for r_{13} arbitrarily (e.g., $r_{13} = 0.5$). Now

change the values of `sigma[1,3]` and `sigmat[3,1]` using similar codes as You can select a value for correlation (e.g., $r_{13} = 0.5$). Recall, that the true $\beta_3 = 0$. Againperform step 1 and 2. Compare the results with the results obtained from step 3 and comment on the differences/similarities. Start increasing the value of the correlation coefficient r_{13}, (e.g., 0.5, 0.7, 0.8, 0.9, 0.95 etc.). How do the estimates and standard error of β_1 and β_2 change for simple and multiple linear regression as the correlation changes?

Task 2: In the second task you need to again simulate a dataset for multiple linear regression. First perform the following steps,

1. Please set your student ID as seed. Generate 500 random values from $X_1 \sim \text{Uniform}[0, 1]$, $X_2 \sim \text{Uniform}[0, 1]$, $X_3 \sim \text{Uniform}[0, 1]$, $X_4 \sim \text{Uniform}[0, 1]$, $X_5 \sim \text{Uniform}[0, 1]$

2. Generate, $Y = 4[\sin(\pi x_1 x_2) + 8(x_3 - 0.5)^3 + 1.5x_4 - x_5 - 0.77] + \epsilon$. Here, $\pi = 3.14.....$ and $\epsilon \sim \text{N}(0, 1)$. You can use the following codes,

```
# This following function provides a data set with p+1 columns #
gendata <- function(n, p){
Xmat <- matrix(runif(n*p, 0, 1), nrow = n, ncol = p)
Y <- 4*( (sin(pi*Xmat[,1]*Xmat[,2])) + 8*(Xmat[,3] - 0.5)^3 +
1.5*Xmat[,4] - Xmat[,5] - 0.77    ) + rnorm(n, 0, 1)
dat <- cbind(Xmat, Y)
return(dat)
}

set.seed(1002656486)
dat <- gendata(500, 5)
```

Now perform the following task,

1. In real life data we don't know the true relationship between the response and predictors. The only thing we have is the dataset `dat`. Perform a multiple regression analysis on the response Y with the 5 predictors in the dataset. Check all the diagnostics (leverages, influential observations, standardized residuals etc.). Comment on your findings.The simulation here needs to be done only once.

2. Simulate further 200 observations using gendata using the following code,

```
dat.new <- gendata(200, 5)
```

   Using the results obtained from step 1 calculate the mean prediction error of the new dataset.

3. Apply appropriate transformations (whatever transformation you think will help). Perform step 1 and 2 again. Explain your findings.

Note: For both the tasks you can present the results according to your convenience. For example you can add further plots and tables which you think are going to be useful. **The presentation video needs to be between 5-6 minutes**. You are going to lose important points if your presentations exceeds 6 minutes or is under 5 minutes.