MSSC 5780 Homework 4

Due Date: November 4, 2022 11:59 PM Henrique Medeiros Dos Reis

1 Homework Instruction and Requirement

- Homework 4 covers course materials of Week 1 to 9.
- Please submit your work in **one PDF** file including all parts to **D2L > Assessments > Dropbox**. *Multiple files or a file that is not in pdf format are not allowed.*
- In your homework, please number and answer questions **in order**.
- Your answers may be handwritten on the Mathematical Derivation and Reasoning part. However, you need to scan your paper and make it a PDF file.
- Your entire work on Statistical Computing and Data Analysis should be completed by any word processing software (Microsoft Word, Google Docs, (R)Markdown, LaTex, etc) and your preferred programming language. Your document should be a PDF file.
- Questions starting with **(MSSC)** are for MSSC 5780 students.
- It is your responsibility to let me understand what you try to show. If you type your answers, make sure there are no typos. I grade your work based on *what you show, not what you want to show.* If you handwrite your answers, write them neatly. If I can't read your sloppy handwriting, your answer is judged as wrong.

2 Mathematical Derivation and Reasoning

The simple linear regression and multiple linear regression models and notations are the same as defined in our course slides and textbook.

1. **(Optional)** Suppose that we have fit the least-squares regression model $\hat{y} = b_0 + b_1 x_1$ but the response is affected by a second variable x_2 such that the true regression function is

   $$E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

   The coefficient b_1 in the original simple linear regression model is no longer unbiased. Show the bias in b_1, i.e., $E(b_1) - \beta_1$.

Let the hat matrix $\mathbf{H}_{n\times n} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ from a multiple linear regression model.

2. **(Optional)** Let h_{ii} be the i-th diagonal element of $\mathbf{H}$. Show that for the multiple linear regression model $\frac{1}{n} \leq h_{ii} \leq 1$.
3. **(Optional)** Show that the weighted least squares estimator minimizes

   $$(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'\mathbf{W}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

   where $\mathbf{W}$ is the diagonal matrix whose diagonal elements are weights $w_1, w_2, \ldots, x_n$.

3 Statistical Computing and Data Analysis

Please perform a data analysis using `R` or your preferred language. **Any results should be generated by computer outputs, and your work should be done entirely by a computer. Handwriting is not allowed. Relevant code should be attached.**

3.1 Diagnostics on Gasoline Mileage Data

We use the same data set `mpg.csv` for data analysis. Consider the multiple regression model $y = \beta_0 + \beta_1 x_1 + \beta_6 x_6 + \epsilon$ fit to the gasoline mileage data in your Homework 3.

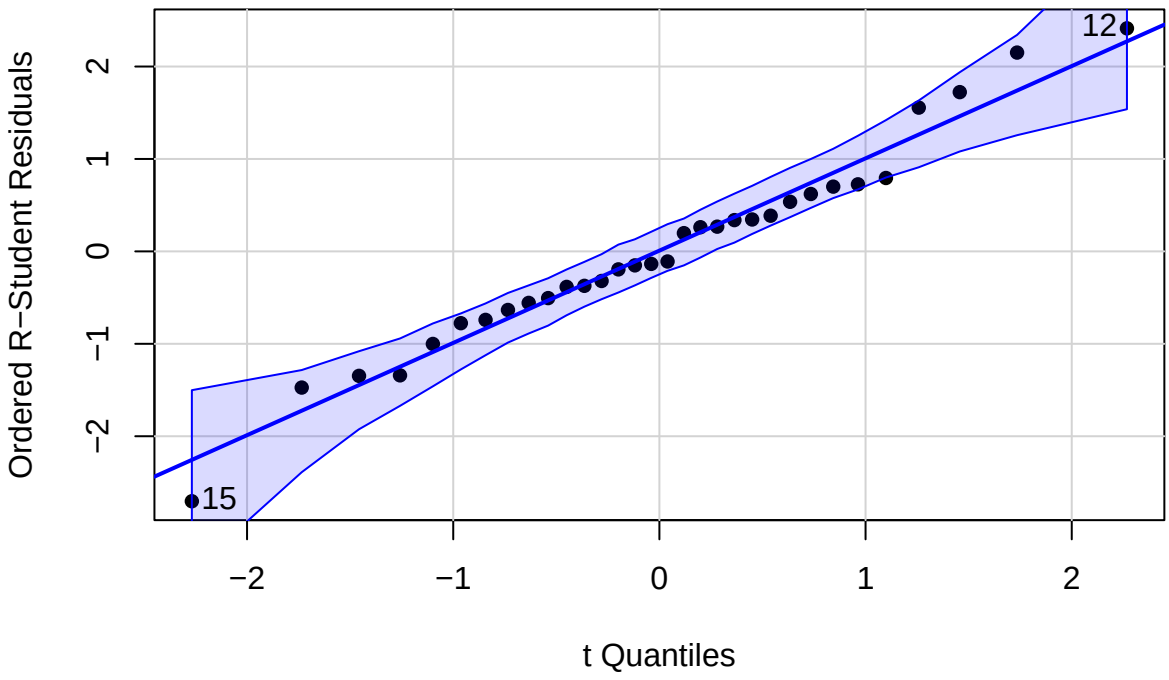
```
mpgData<-mpgData <- read.csv("mpg.csv")
mpg_engDisp_carBar_lm <- lm(y~x1+x6, data = mpgData)
mpg_engDisp_carBar_lm
```

```
##
## Call:
## lm(formula = y ~ x1 + x6, data = mpgData)
##
## Coefficients:
## (Intercept)           x1           x6
##    32.88455     -0.05315      0.95922
```

1. Compare R-student residuals t_i with Student-t t_{n-p-1} using a qqplot. Generate the histogram and density plot of t_i as well. Does there seem to be any problem with the normality assumption?

   Solution:

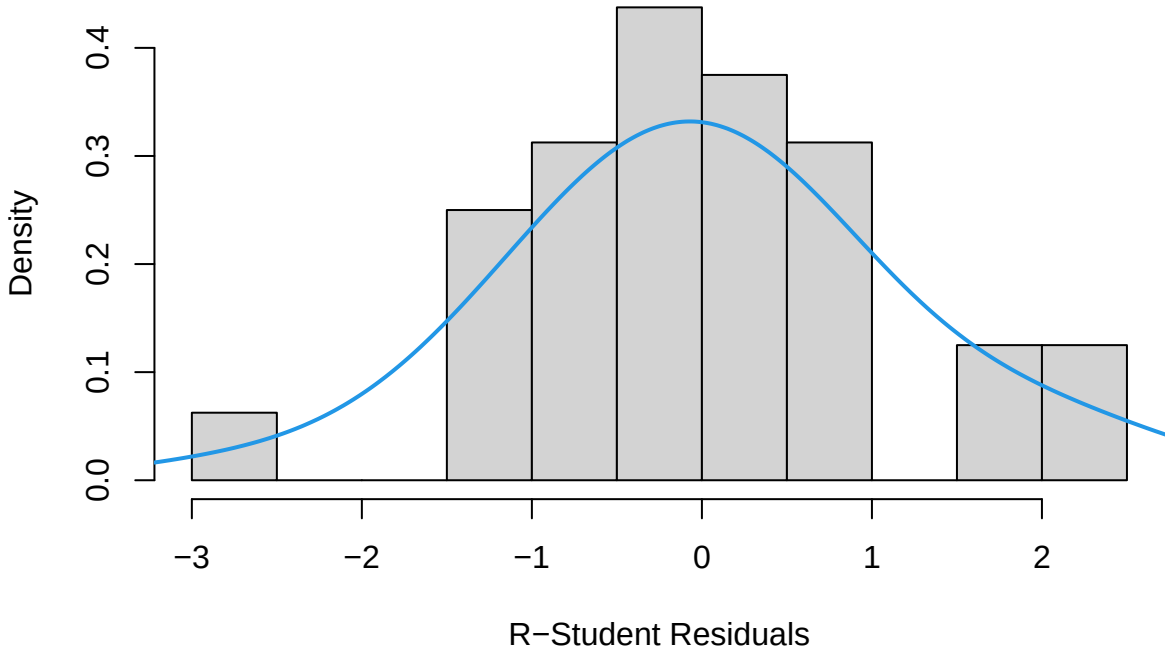
```
car::qqPlot(mpg_engDisp_carBar_lm, id = TRUE, col.lines = "blue",
            reps = 1000, ylab = "Ordered R-Student Residuals", pch = 16)
```




```
## [1] 12 15
```

```
r_stud <- rstudent(mpg_engDisp_carBar_lm)

hist(r_stud, prob = TRUE, breaks = 10, xlab = "R-Student Residuals", main = "")
lines(density(r_stud, adjust = 2), col = 4, lwd = 2)
```



We can see that the qq-plot seems fine, but the histogram seems to have heavier tails, which indicates that this is not exactly normal. And because this has heavier tails, it probably is a t-dist.

2. Perform the Box-Cox method and discuss the necessity of any transformation on y.

Solution:

```
carPT<-car::powerTransform(mpg_engDisp_carBar_lm, family = "bcPower")
summary(carPT)
```

```
## bcPower Transformation to Normality
##    Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## Y1   -0.1483           0      -0.8779       0.5812
##
## Likelihood ratio test that transformation parameter is equal to 0
##  (log transformation)
##                               LRT df    pval
## LR test, lambda = (0) 0.1598941  1 0.68925
##
## Likelihood ratio test that no transformation is needed
##                              LRT df      pval
## LR test, lambda = (1) 9.355126  1 0.0022236
```

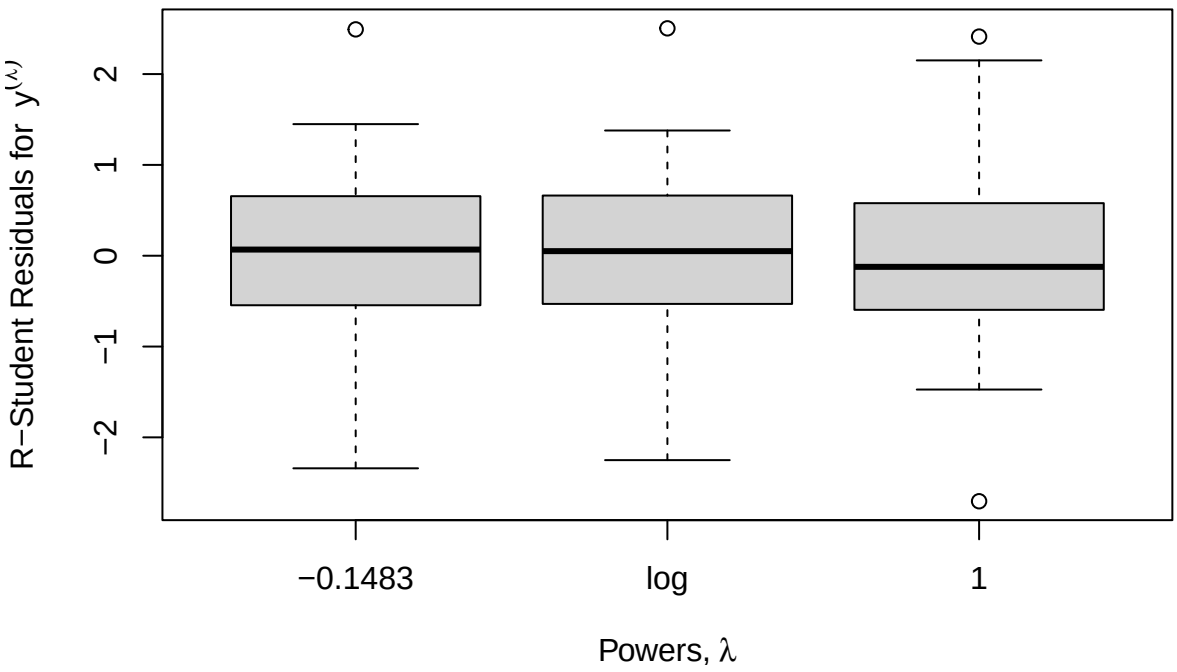
Looking at the summary enough evidence that the transformation at λ = -0.1483 should be done.

3. Use $\lambda = 0$ (log transformation) and the λ selected by the Box-Cox method to refit the model to the transformed data. Compare their R-student residuals with the R-student residuals from the non-transformed data using a boxplot.

   Solution:

```
log_mpg_engD_carB <- lm(log(y) ~ x1 + x6, data = mpgData)
lambda_mpg_engD_carB <-lm(y^(-0.1483) ~ x1+x6, data = mpgData)

mat <- matrix(r_stud)
for (lam in c(-0.1483, 0)) {
    refit <- update(
        mpg_engDisp_carBar_lm, car::bcPower(y, lam) ~ .
        )
    mat <- cbind(rstudent(refit), mat)
}
colnames(mat) <- c(-0.1483, "log", 1)
boxplot(
    mat, id = FALSE,
    xlab = expression("Powers," ~ lambda),
    ylab = expression(
        "R-Student Residuals for "
        ~ y ^ (lambda))
    )
```

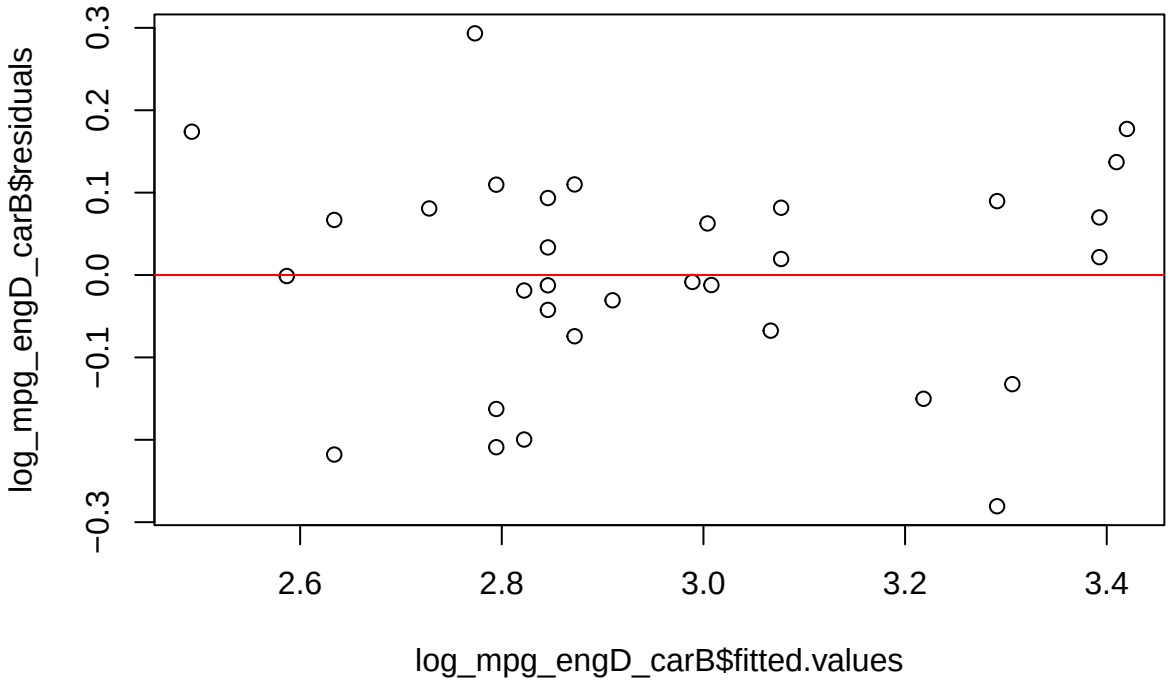


We can see that the transformations for log and -0.14 box plot are the same, while the non-transformed one is not symmetric.

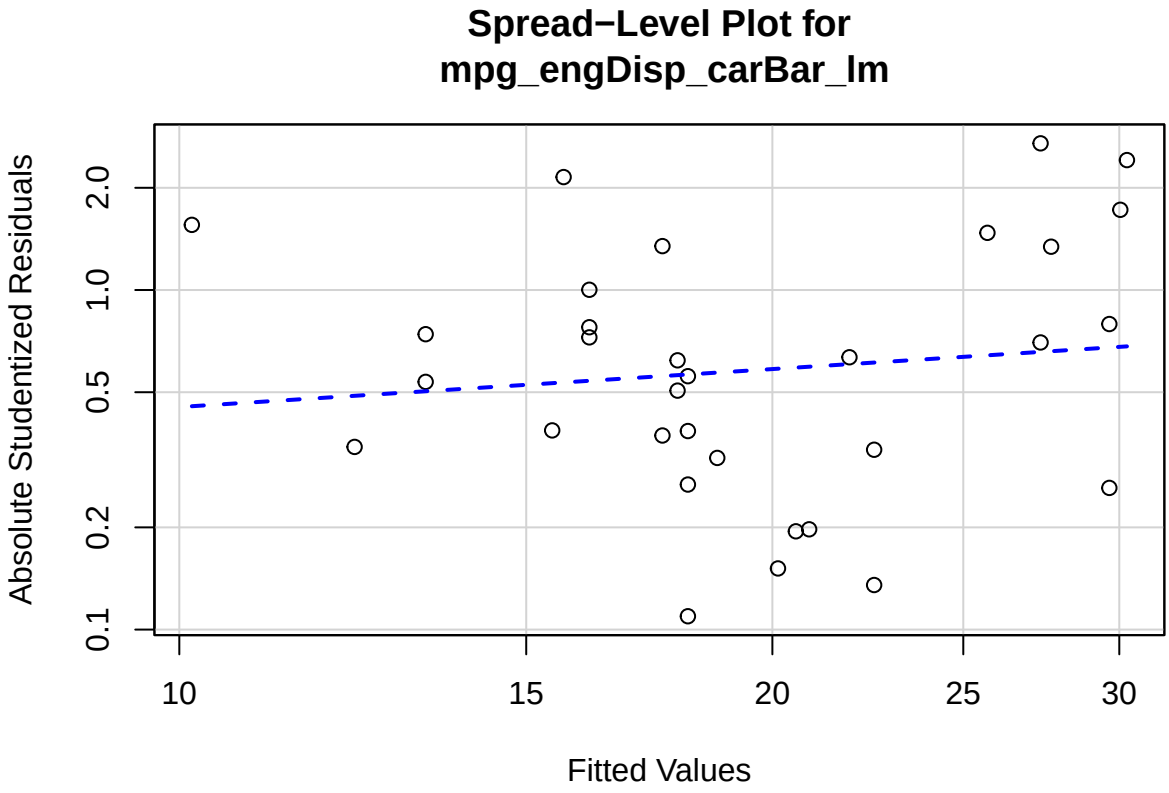
4. Construct a plot of the R-student residuals t_i versus the fitted responses and the Tukey's spread-level plot. Any sign of violation of constant variance?

Solution:

```
plot(log_mpg_engD_carB$fitted.values, log_mpg_engD_carB$residuals)
abline(0,0, col="red")
```




```
car::spreadLevelPlot(mpg_engDisp_carBar_lm, smooth = FALSE)
```




```
##
## Suggested power transformation:  0.6287351
```

It is possible to see that there definitely is some sign of constant variance. Since the line, in the plot with the blue line, has a positive slope, the variance is not constant.

5. In fact, we can perform some formal hypothesis testing on constant variance like H_0 : Constant variance vs. H_1 : Variance changes with $E(y \mid x)$. Use `car::ncvTest()` to perform the test. Explain the testing result.

Solution:

```
car::ncvTest(mpg_engDisp_carBar_lm)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 3.953199, Df = 1, p = 0.046782
```

Since the p-value is 0.046782 which is smaller than $\alpha = 0.05$. Which leads to reject null hypothesis. Therefore, not constant variance.

3.2 Influence Diagnostics on Squid Data

An experiment was conducted to study the size of squid eaten by sharks and tuna. The regressors are characteristics of the beak or month of the squid. The `squid.csv` data contain the variables

- x_1: Rostral length in inches
- x_2: Wing length in inches
- x_3: Rostral to notch length
- x_4: Notch to wing length
- x_5: Width in inches
- y: Weights in pounds

```
squidData <- read.csv("squid.csv")
squidReg <- lm(y ~ x1+x2+x3+x4+x5, data = squidData)
```

Perform a thorough leverage and influence diagnostics of the squid data.

1. Compute R-studentized residuals, hat values, Cook's distance, DFFITS, DFBETAS, and COVRATIO measures. Describe how you detect leverage and influential points. Discuss the effect of data points on coefficients, fitted values, and precision of coefficients. (The `influence.measures()` function provides all influence measures.)

   Solution:

```
summary(influence.measures(squidReg))
```

```
## Potentially influential observations of
##   lm(formula = y ~ x1 + x2 + x3 + x4 + x5, data = squidData) :
##
##    dfb.1_ dfb.x1 dfb.x2  dfb.x3  dfb.x4  dfb.x5 dffit   cov.r   cook.d hat
## 2   0.19  -0.04  -2.50_*  1.28_*  1.28_*  0.16  -2.66_*  0.52    0.92   0.56
## 18 -0.45   0.43   1.02_* -0.17   -0.73   -0.89  -1.65    0.27    0.34   0.31
## 19  0.00   0.01   0.00   -0.01    0.01   -0.01  -0.03    3.79_*  0.00   0.61
```

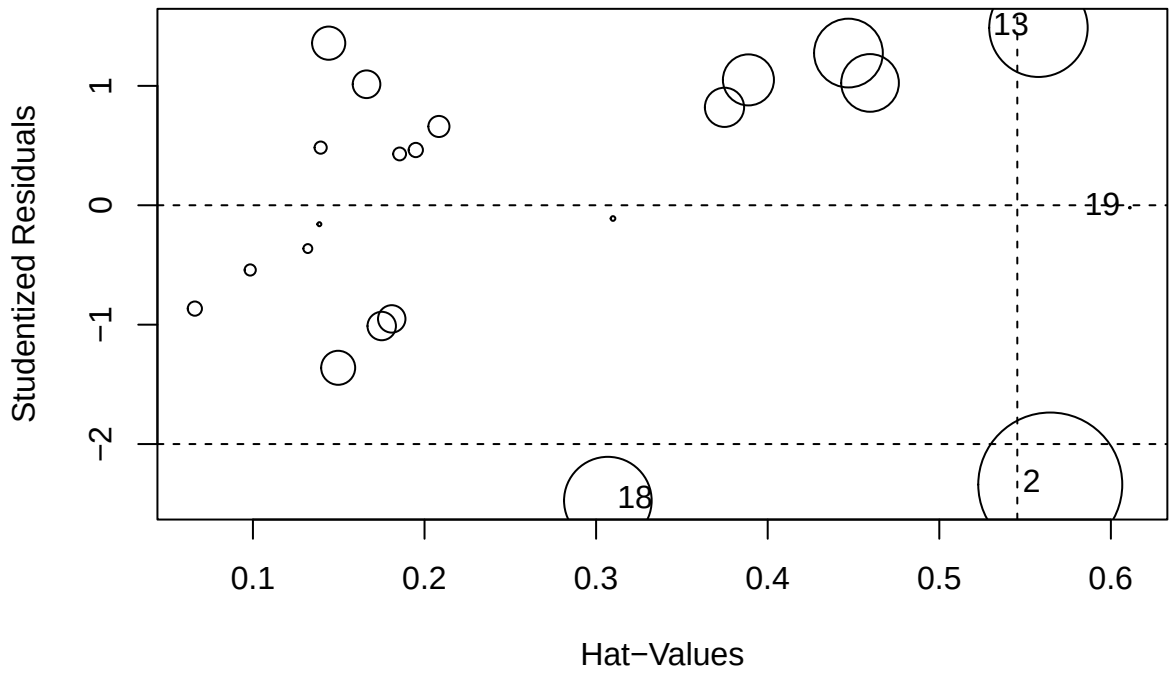
Looking at this, we can see that the points 2, 18, and 19 got flagged as influential points. The reason for this is because in at least one of the measures, this points got a value that is being influential to that specific measurement. For example, looking at df betas 2, 2 and 18 are influential points. And looking at cov.r, point 19 is influential.

2. Let's use some visualization tools.

   - Create the bubble plot.
   - Create the influence index plot (`car::influenceIndexPlot()`)

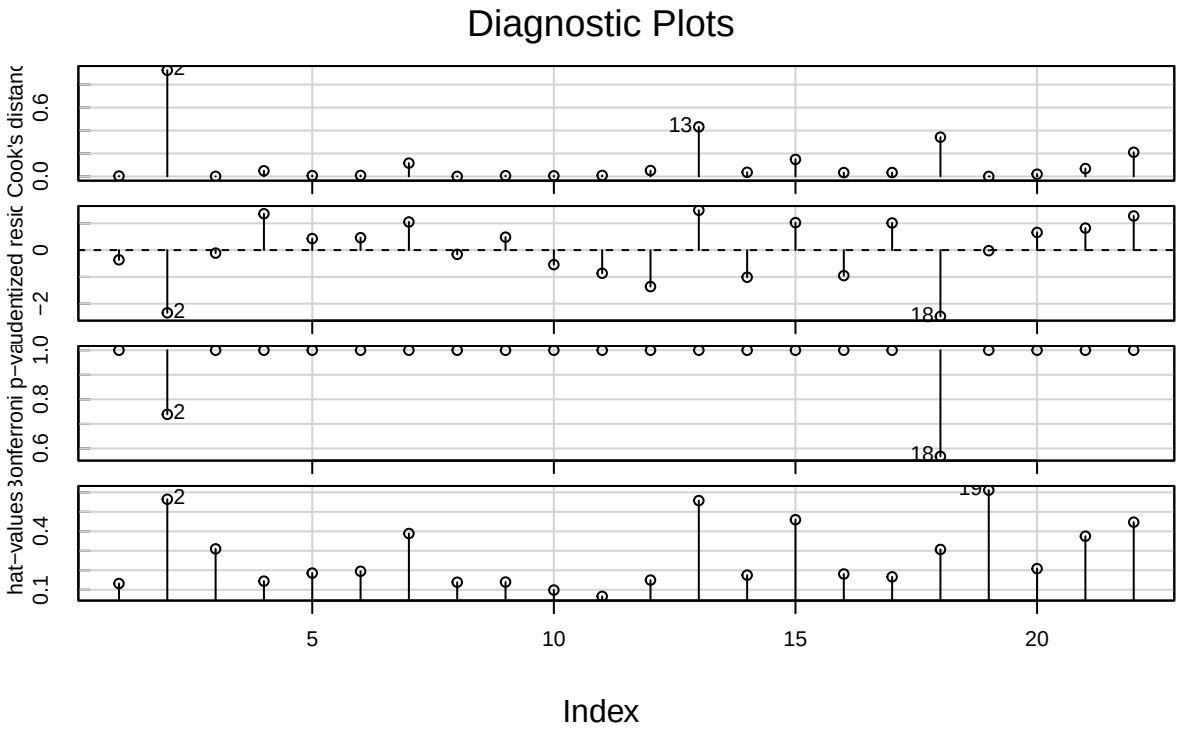
   Solution:

```
car::influencePlot(squidReg)
```




```
##        StudRes       Hat        CookD
## 2  -2.33904094 0.5646687 0.9244357141
## 13  1.48676396 0.5577881 0.4320150660
## 18 -2.47377065 0.3068443 0.3420506512
## 19 -0.02003576 0.6111113 0.0001121431
```

```
car::influenceIndexPlot(squidReg)
```

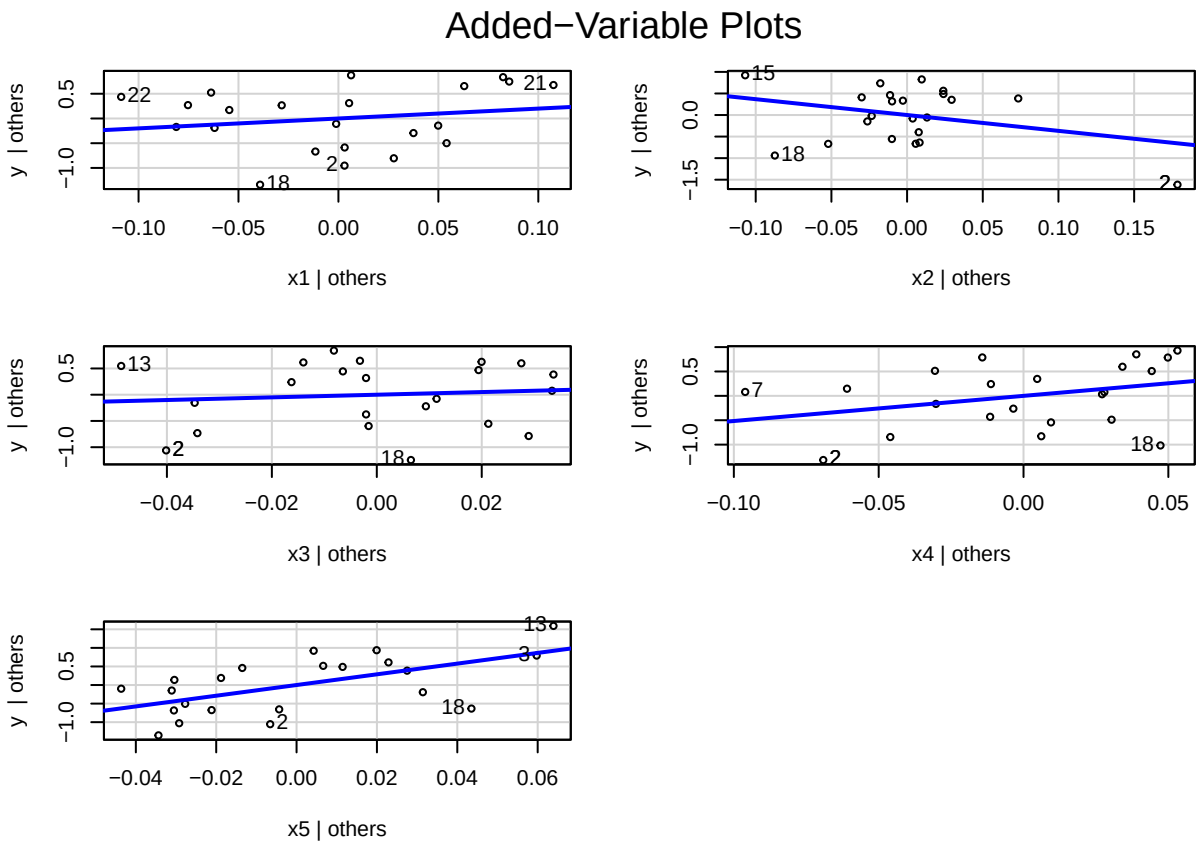


And now looking at the graphs, it agrees with the explanation in number 2.

3. Produce the added-valued plot (`car::avPlots()`) for each regressor $x_i, i = 1, \ldots, 5$. Are there any joint influence of data points on the regression coefficients?

Solution:

```
car::avPlots(squidReg)
```



Looking at these, we can see that points 2 and 18 are jointly influential, almost in all of the the coefficients. they "put down" the line.

4. **(MSSC)** Numerically verify the following properties of the added-valued plot.
   (a) The slope of the least squares *simple* regression line of $e(y \mid x_{(1)})$ on $e(x_1 \mid x_{(1)})$ is the same as the least-squares slope b_1 for x_1 in the full *multiple* regression.
   (b) The residuals from the simple regression $e(y \mid x_{(1)})$ vs. $e(x_1 \mid x_{(1)})$ are the same as the residuals e_i from the full multiple regression.
   (c) The standard error of b_1 is $s/\sqrt{\sum_{i=1}^{n} e_i^2(x_1 \mid x_{(1)})}$, where $s = \sqrt{MS_{res}}$ from the full multiple regression.

   Solution:

   a-)

```
(slope <- summary(lm(e_i_yx~e_i_xx))$coefficients[2,1])
```

```
## [1] 1.999413
```

```
my_coef[2,1]
```

```
## [1] 1.999413
```

We can see that they are the same

```
#b-)
round(squidReg$residuals) == round(lm(e_i_yx~e_i_xx)$residuals)
```

```
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##   17   18   19   20   21   22
## TRUE TRUE TRUE TRUE TRUE TRUE
```

As we can see, they are the same

```
s<- sqrt((sum(squidReg$residuals^2))/(22-5-1))
(b1<- s/sqrt(sum(e_i_xx^2)))
```

```
## [1] 2.573338
```

```
sum_squid<-summary(squidReg)
sum_squid$coefficients[ 2, 2]
```

```
## [1] 2.573338
```

Once again, they are the same.

3.3 Simulation

Consider a simple linear regression model $y_i = \beta_0 + \beta_1 x_i + \epsilon_i, i = 1, 2, \ldots, n$ with $\epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2)$. Set $x_i = i$, $\beta_0 = 2$, $\beta_1 = -1.5$, $\sigma = 1.2$, $n = 15$.

1. **(MSSC)** Generate a sample $\{y_i\}_{i=1}^n$ from this model, and compute the regression coefficients b_0, b_1 and variance estimate s^2. Repeat 10,000 times.

   Solution:

```
#Residual standard error: 1.1 == s^2
beta_0 <- 2
beta_1 <- -1.5
n <- 15
sigm <- 1.2

x<- seq(1,15)

b_0<-c()
b_1<-c()
```

```
sigmas<-c()
for(i in 1:10000)
  {
    epi <- rnorm(15, 0, sigm)
    y <- beta_0+beta_1*x + epi
    my_lm<-lm(y~x)
    coeff <- summary(my_lm)$coefficients
    b_0<-c(b_0, coeff[1,1])
    b_1<-c(b_1, coeff[2,1])
    sigmas<-c(sigmas, (summary(my_lm))$sigma^2)
}
```

2. **(MSSC)** Compute the sample mean of b_0, b_1, and s^2. Compare them with their true expected value. What is your conclusion?

   Solution:

```
mean(b_0)
```

```
## [1] 1.99626
```

```
mean(b_1)
```

```
## [1] -1.499283
```

```
sqrt(mean(sigmas)) ###sigma squared
```

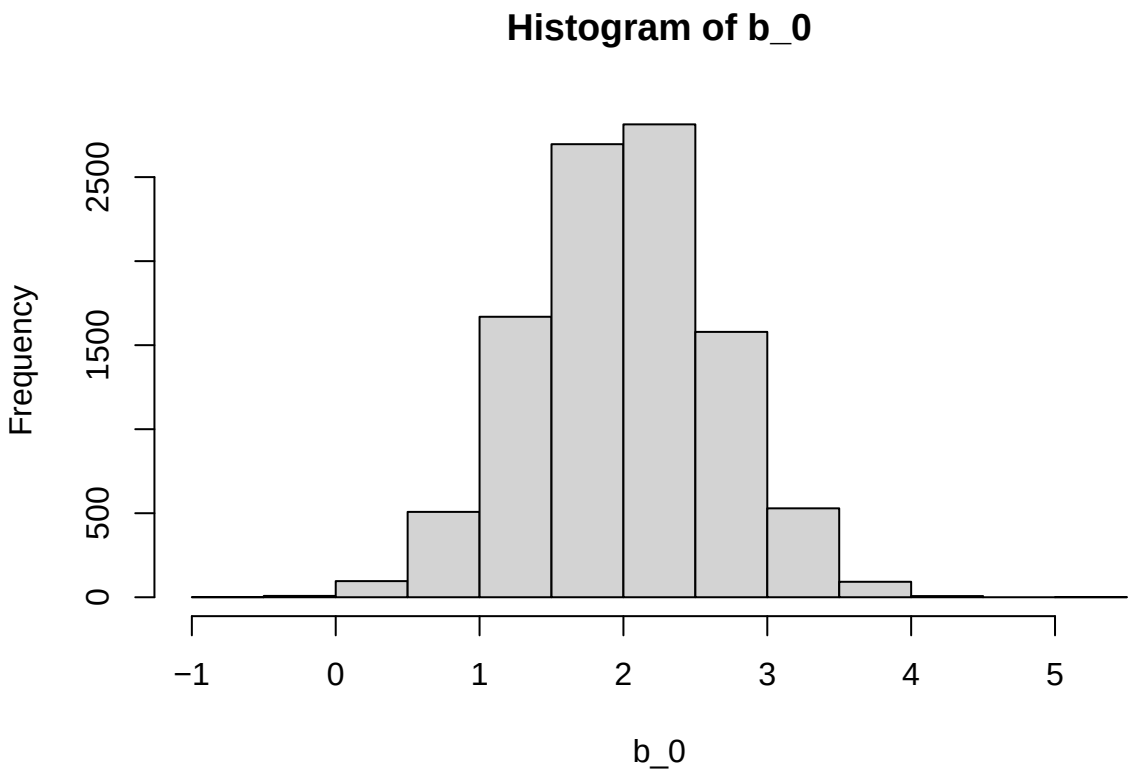
```
## [1] 1.198144
```

It is possible to see that the mean of the values for $b_0, b_1, and \sigma$ are almost equal to the values we set up at the beggining of the experiment, and are not exatly the same just because we added some noise to it.

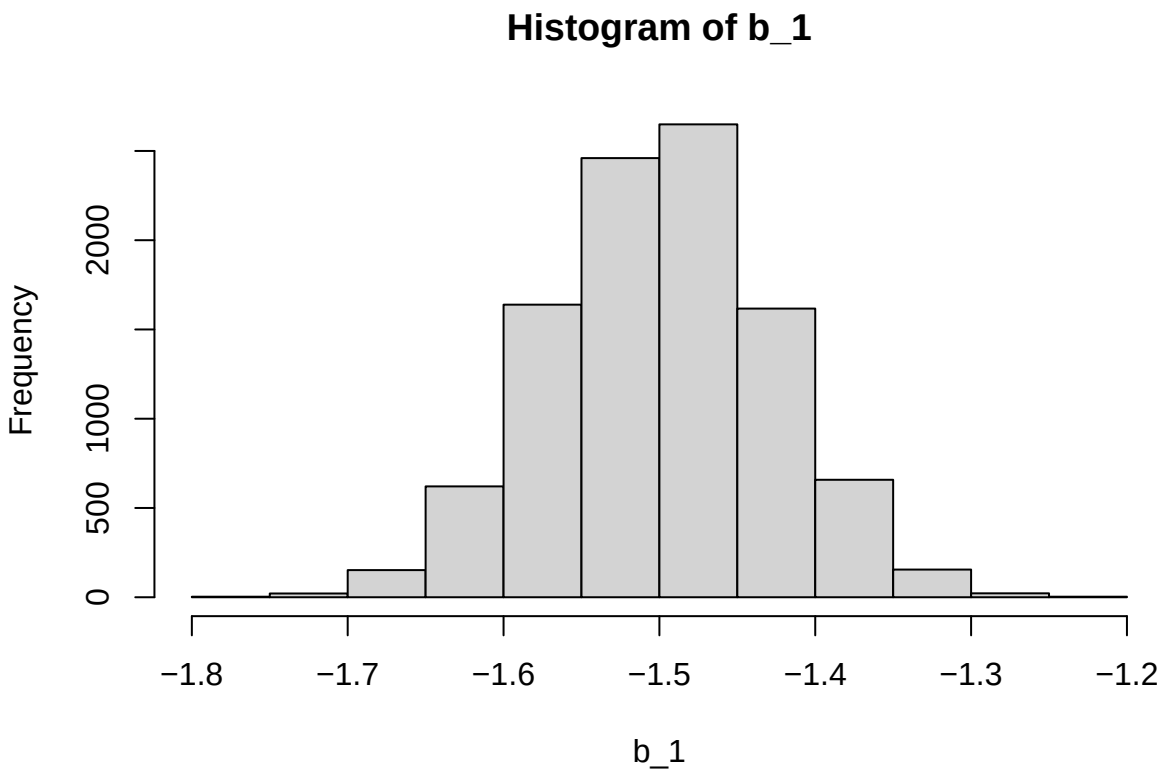
3. **(MSSC)** Plot the histograms of b_0, b_1, and s^2. Compare the histograms with their sampling distribution.

   Solution:

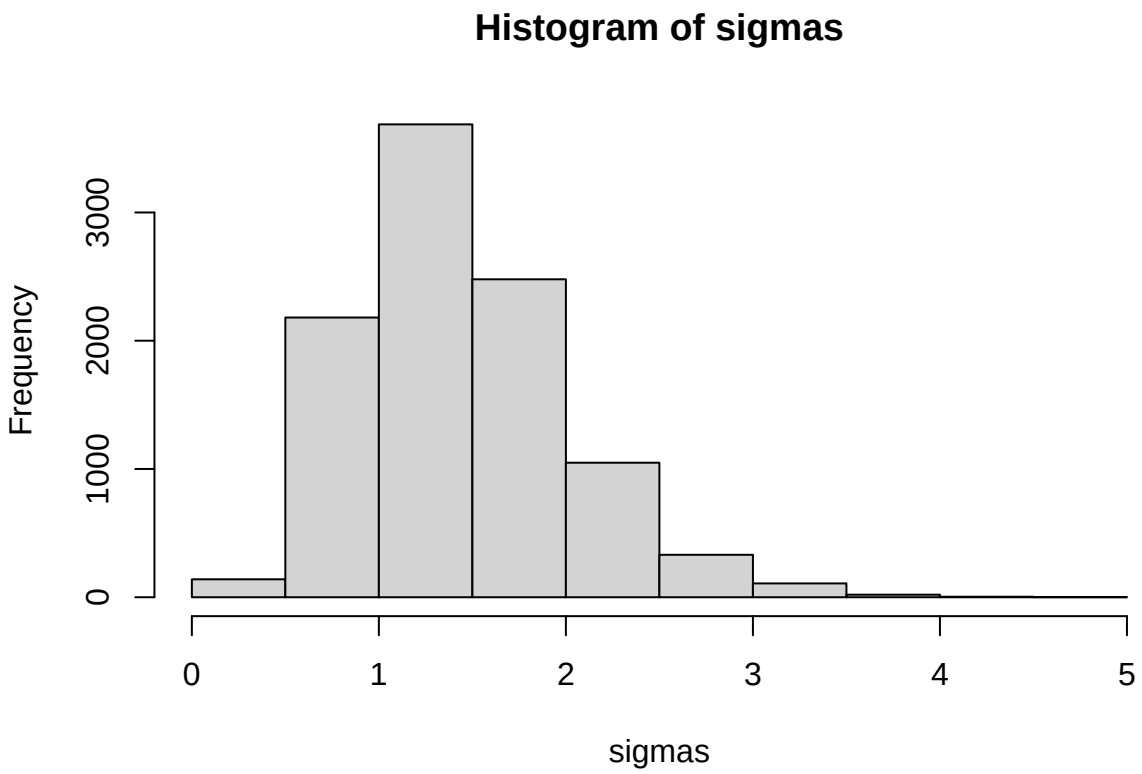
```
hist(b_0)
```




```
hist(b_1)
```


Histogram of b_1
Frequency
0
500
1000
2000
−1.8
−1.7
−1.6
−1.5
−1.4
−1.3
−1.2
b_1

```
hist(sigmas)
```



The histograms for b_0 and b_1 seem to be normal distributed. But the σ histogram is skewed to the right. Which seems more like a Gamma distribution.

4. **(MSSC)** Compute the covariance matrix of b_0 and b_1 and the true covariance matrix. Compare the two.

Solution:

```
new_x <- cbind(1, x)
(cov_matrix<- cov(data.frame( b_0, b_1)))
```

```
##             b_0          b_1
## b_0  0.42658596 -0.041094530
## b_1 -0.04109453  0.005121664
```

```
sigm^2*solve(t(new_x)%*%new_x)
```

```
##                          x
##    0.42514286 -0.041142857
## x -0.04114286  0.005142857
```

Once again, the values are almost the same, which only happens because we added some noise to the data.

5. **(MSSC)** Find the variance of s^2. Use the fact that $\frac{(n-2)s^2}{\sigma^2} \sim \chi^2_{n-2}$ and $\text{Var}(\chi^2_k) = 2k$ to get the true variance of s^2. Compare the two.

Solution:

Assuming normality, to get the the true variance of s^2, we find $Var(s^2)$. Since $Var(\frac{(n-2)s^2}{\sigma^2}) = Var(\chi^2) = 2(n-2)$, then, we have $Var(s^2) = \frac{(2\sigma^4)}{(n-2)}$

```
#sample
var(sigmas)
```

```
## [1] 0.3180418
```

```
#true
2*(sigm^4)/(n-2)
```

```
## [1] 0.3190154
```

The values are almost the same.

6. **(MSSC)** Suppose now $\beta_1 = 0.0001$. Generate the simulated data $\{x_i, y_i\}_{i=1}^n$ with $n = 10, 100, 1000, 10000$. Show that as n increases, for the test $H_0 : \beta_1 = 0$, its t_{test} gets larger and its p-value gets smaller, and therefore H_0 is rejected for a sufficiently large n, even though β_1 is practically zero.

```
#Residual standard error: 1.1 == s^2
beta_0 <- 2
beta_1 <- 0.0001
n_1 <- c(10,100,1000,10000)
p_vals<-c()
t_test<-c()
for(i in n_1)
{
  epi <- rnorm(i, 0, sigm)
  x<-seq(1,i)
  y<-beta_0+beta_1*x+epi

  p_vals<-c(p_vals,(t.test(beta_0+beta_1*x+epi, data= data.frame(x,y))$p.value))
  t_test<-c(t_test,(t.test(beta_0+beta_1*x+epi, data= data.frame(x,y))$statistic))
}
p_vals
```

```
## [1]  2.332853e-04  1.244250e-27 8.866863e-281  0.000000e+00
```

```
as.numeric(t_test)
```

```
## [1]   5.885306  15.199311  51.070696 204.595310
```

It is possible to see that the vector containing the p-values is definitely decreasing, and when $n = 10000$, the p-value is just 0. And basically the same thing happens with T-statistic, except the opposite, it increases from 4.25 when $n = 10$ to 201.87 when $n = 10000$