MATH 535 EXAM 1

Gustavo Esparza

03/09/2019

THEORETICAL PORTION

1.) Someone tells you that the leverage of the ith point in a regression model is .20. In non-technical language, what specifically does leverage measure?

Leverage is the measure of how much the observed value, y_i, impacts the estimated value, $\hat{y}_i$. Then if we have a fitted regression model, an observed value that has a significant distance from the corresponding fitted value of the model is said to have high leverage. In other words, a leverage point is a value from the observed data that is either significantly larger or significantly smaller than what the regression model fit from the data estimates that value to be. These leverage points influence the overall fitted model by moving the model towards them, thus making the residuals of leverage points appear to be small.

Is high leverage always bad?

By definition, a value that has high leverage will dampen the residuals of the model and make the model shift towards the value. Then, if a high leverage point does not follow the general trend of our data, we will have a model that accommodates the leverage point and thus poorly fits the rest of the data. However, if a high leverage point does follow the general trend of the data, then we can end up with a stronger model despite having to acommodate for the leverage point.

2.) In a random sample of 643 males, 210 of them have been assigned to receive a testosterone supplement designed to help men lose weight. Both groups were then assigned the same regular exercise routine and diets over an 8-week observation period. After 8 weeks, the 210 males receiving the supplement had lost on average 11.2 lbs with standard deviation equal to 6.1 lbs, while the 433 individuals not receiving the supplement lost on average 6.6 lbs with standard deviation equal to 4.1 lbs.

Develop a statistical test to determine whether or not the supplement will help males lose weight then address this hypothesis using your test.

We are considering two assigned groups that are independent of each other, with $n_1 = 210$, $\bar{x}_1 = 11.2$, $\bar{s}_1 = 6.1$ and $n_2 = 433$, $\bar{x}_2 = 6.6$, $\bar{s}_2 = 4.1$.

Given this information regarding the sample, we have the following Hypothesis Statement:

$$H_0 : \mu_1 = \mu_2 \text{ VS. } H_1 : \mu_1 > \mu_2$$

We are conducting a test for the difference in means of two samples with differing sizes and sample variances. Then, the two sample t-test statistic with non pooled variance is defined as follows :

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

Thus, using the data provided in this sample, we arrive at the following statistics:

$$t = \frac{11.2 - 6.6}{\sqrt{\frac{6.1^2}{210} + \frac{4.1^2}{433}}} = 9.897334$$

Our t-statistics follows a t-dstribution with degrees of freedom equivalent to the minimum of our sample sizes, 209. Thus, our p-value is defined as $\text{PR}(\text{t}(209) > 9.897334) \approx 0 < \alpha$, where α is a desired significance level (.01,.05, etc.). Therefore, we reject the null-hypothesis and conclude that the supplement will help males lose weight.

APPLIED PORTION

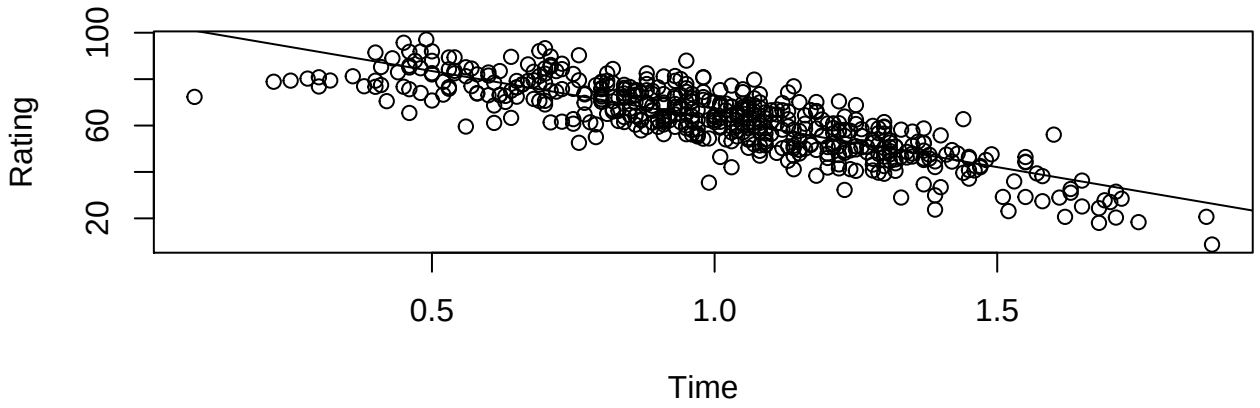
3.)

```
library(readr)
CableData = read_csv("/Users/gustavo/Desktop/535 Exam 1/cable.csv")
Time = CableData$calltime
Rating = CableData$satisfaction
model1=lm(Rating~Time)
```

a.) Do you think a linear model is appropriate for describing and/or inferring customer satisfaction based on time spent waiting in the queue?

We will begin by creating a linear model for the data and checking its' validity, both visually and statistically. First, let's observe how our linear model fits over our plotted data.

```
plot(Time,Rating)
abline(model1)
```



We observe that the linear model passes through a good amount of our data, rather than missing the mark completely. This is a good start. Now, we will look into the statistical relevance of our model, beginning with a quick summary of the model.

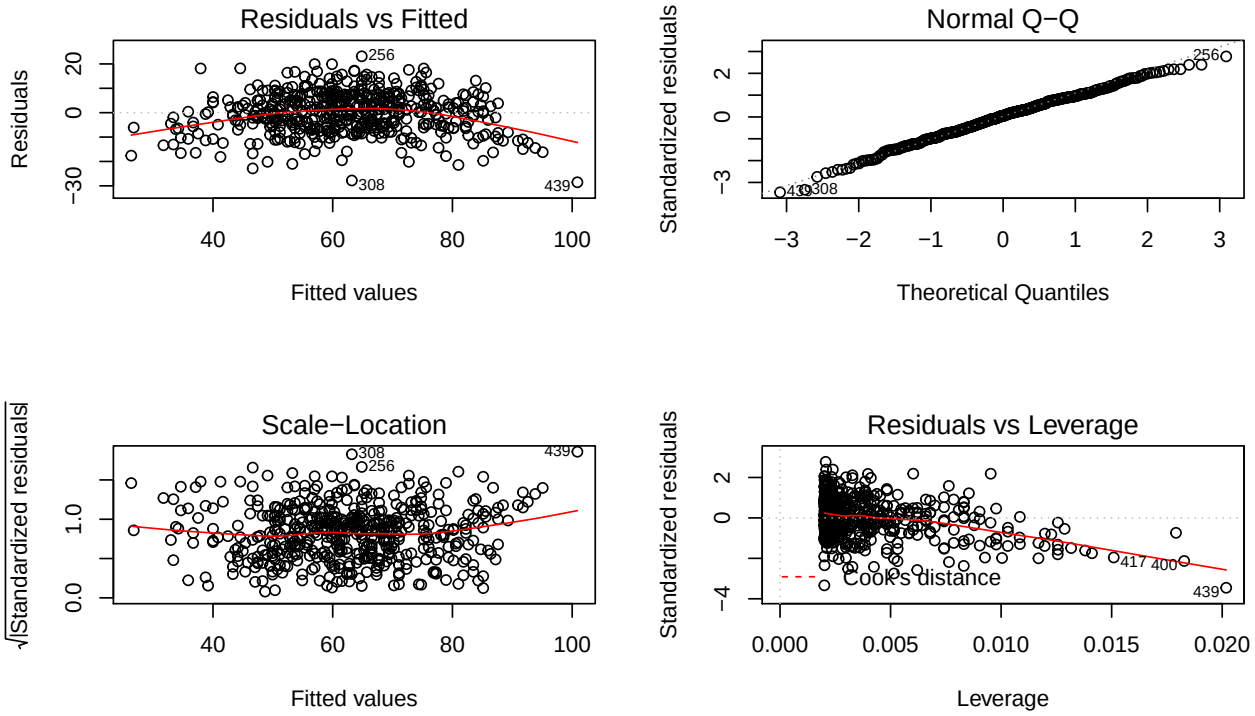
```
summary(model1)
```

```
##
## Call:
## lm(formula = Rating ~ Time)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -28.5323  -5.6875   0.3991   6.2116  23.1199
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  104.185      1.278   81.52   <2e-16 ***
## Time         -41.405      1.216  -34.04   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.343 on 498 degrees of freedom
## Multiple R-squared:  0.6994, Adjusted R-squared:  0.6988
## F-statistic:  1159 on 1 and 498 DF,  p-value: < 2.2e-16
```

From a quick review of our model summary, we can see that our model and predictor appear to be statistically significant, according to their respective p-values. We can also observe an R^2 value of .6994. This essentially means that our model accounts for 69% of the variability in our data, the higher this value the better our model fits the data.

Now, we will take a look at the residual plots of our model to check our assumptions for a linear regression model.

```
par(mfrow=c(2,2))
plot(model1)
```



The Residuals vs Fitted vales plot has our data points spread out along the x-axis, which is what we want to see in order to validly assume Linearity. Although the data points are reasonably spread in a random fashion, we clearly notice a curve-like trend (emphasized by the red curvature over the plot) which implies that our predictor and response do not have a linear relationship.

The Normal-QQ plot has a majority of the points lying on quantile line, which is implies that our residuals are normally distributed. There is some slight deviation from the line on the tail ends, but it is not a gratuitous departure.

The Scale-Location plot should have the data points randomly spread across the x line, similar to the Residuals vs Fitted values, in order to validate constant variance of our model. We do notice random scattering across the line, so we are able to assume constant variance. There does seem to be a slight bend in the residuals, but this is ultimately due to the non-linearity that we observed in the Residuals vs Fitted values plot.

From our prior research, we can conclude that the linear model used to fit the data is not quite a valid model for this data. We did see proof of relevance to the model and the none of the residual plots provided showed an extreme violation. However, we still noted a few trends (Mainly the curve in Residuals vs Fitted values) that leads us to believe that a direct linear model is ineffecient and can be improved.

b.) Provide a model that is more appropriate for statistical inference. Explain how you arrived at your

model. Support your argument with figures and results if possible.

Based on the results from part **A**, we have a model that has a relevant predictor and well-behaving residuals. However, we have an unwanted non-linear relationship between our predictor, Waiting Time, and the response of Satisfaction rating (again, this is evident from the curve seen in the Residuals vs Fitted values plot). For these reasons, we will consider a transformation of the predictor variable. If we also had a severe issue with non-constant variance, we would also need to consider a transformation of the response variable. Moving forward with our decided variable to transform, we shall consider which transformation seems appropriate and effecient.

A beneficial method for finding our desired power value is shown as follows:

```
y = Rating
x = Time

x.power.trans = function(lambda){
  x.trans = (x^lambda - 1)/(lambda)
  RSS.lambda = sum(lm(y~x.trans)$residuals^2)
  RSS.lambda
}
optim= optim(1,x.power.trans)
exponent = optim(1,x.power.trans)$par
```

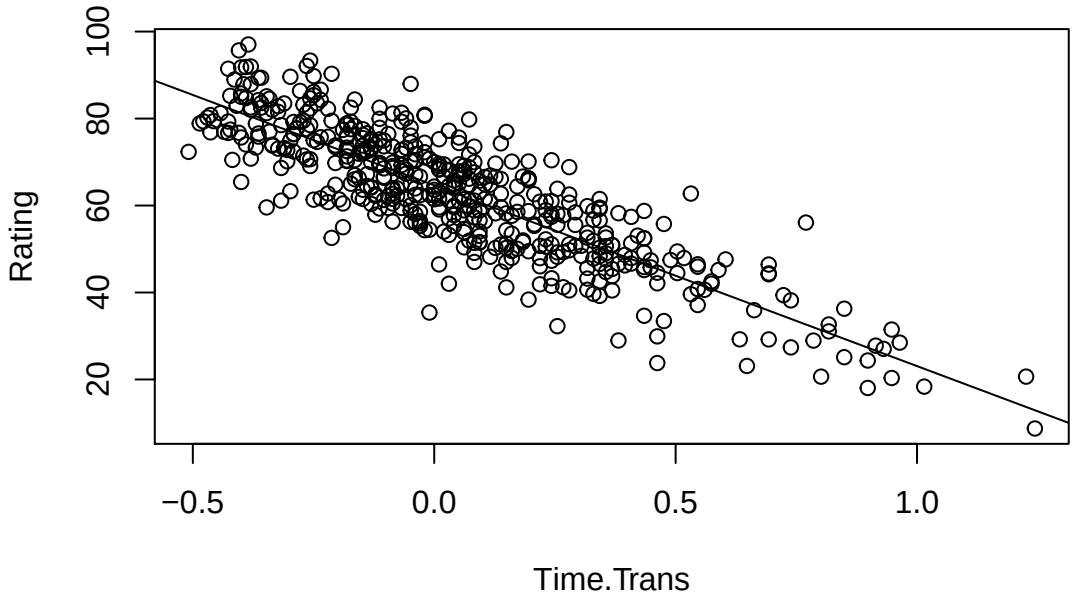
```
exponent
```

```
## [1] 1.951563
```

Thus, our optimal power value for our predictor is 1.951563. This makes sense when considering the shape of the residual trend, which appeared to closely model a parabola of some sort. This also makes a great deal of sense in the context of our dataset. We wouldn't expect the satisfaction rating to behave linearly when considering a wait time of 30 minutes versus a 1 hour waiting time. It is much more likely that disatisfaction will proceed to grow as the wait time increases. So, we have a new exponent value that makes sense both mathematically and conceptually. We will transform our predictor and use it to create a new model.

```
Time.Trans=(Time^exponent-1)/(exponent)
```

```
model2=lm(Rating~Time.Trans)
plot(Time.Trans,Rating)
abline(model2)
```



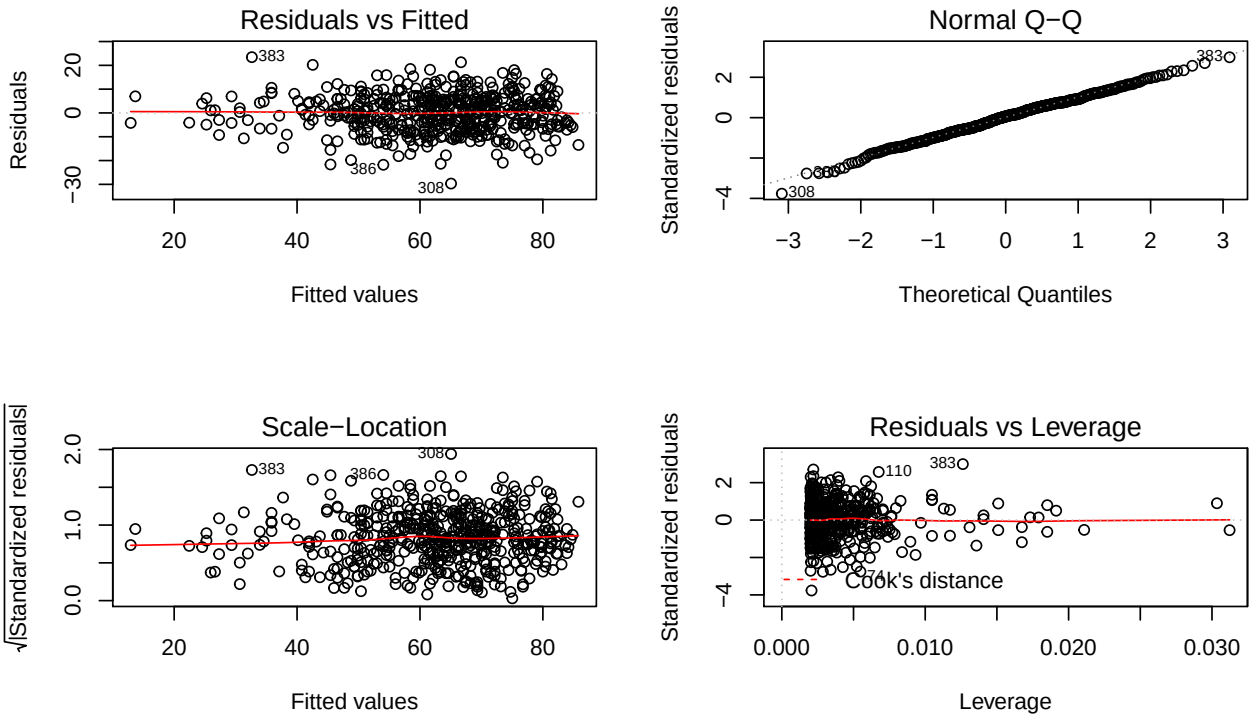
Once again, we see our new fitted line going through a majority of the data. This is what we hope to see after transforming the predictor.

```
summary(model2)
```

```
##
## Call:
## lm(formula = Rating ~ Time.Trans)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -29.6679  -5.1572   0.4341   5.4757  23.4362
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  64.6442     0.3571  181.02   <2e-16 ***
## Time.Trans  -41.5679     1.1288  -36.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.886 on 498 degrees of freedom
## Multiple R-squared:  0.7314, Adjusted R-squared:  0.7309
## F-statistic:  1356 on 1 and 498 DF,  p-value: < 2.2e-16
```

Analyzing our p-values, we see that our model and predictor are stil statistically significant. The R^2 value increased by three percent. Although this is not a huge jump, we still gained model strength without adding any predictors, which is always a sign of a better fitting model.

```
par(mfrow=c(2,2))
plot(model2)
```



The Residuals vs Fitted vales plot has our data points spread out along the x-axis, which is what we want to see in order to validly assume Linearity. We no longer see any curvature within the residuals, so our non-linearity issue has been resolved.

The Normal-QQ plot still looks very good, and the tails appear to be more inline with our quantile line. Normality is still assumed.

The Scale-Location plot should have the data points randomly spread across the x line, similar to the Residuals vs Fitted values, in order to validate constant variance of our model. We still have random scattering across the x-line, and the curvature is once again absent from this plot.

The transformation overall increased the strength of our model by a slim margin, but it did make the model itself muchhealthier in terms of the assumptions/residual plots. Thus, we have found a model that is more effecient for statistical inference than our initial linear model.

c.) Using your response to part b.) Can you come up with a confidence interval for the average satisfaction index score of individuals who have to wait 2 hours in the queue.

```
Conf.value = (2^exponent-1)/(exponent) #Transforming our value

predict(model2,newdata=data.frame(Time.Trans=Conf.value),interval="confidence")
```

```
##        fit      lwr      upr
## 1 3.557775 0.333508 6.782042
```

A 95% confidence interval for the average satisfaction index score of individuals who have to wait 2 hours in the queue is [.333508,6.782042]. Thus, we can say that we are 95% confident that the true satisfaction index score for an individual that waits 2 hours in the queue will be between 0 and 7 (rounding to whole numbers)

d.) Do you have any reservations about using your confidence interval from part c.)? Please explain if you do.

For the specific confidence interval that we created, the call length of 2 hours is being considered for the interval. Looking at the dataset provided, it appears that our highest wait time is 1.88 hours. Thus, our value used for the confidence interval is outside the scope of our dataset. In addition, we can see that the transformed predictor range also does not include the transformed value of 2 (1.469557). We would not want to make any inference for a value that exceeds the scope of our data because, while we are computationally correct, we do not know how the data behaves outside of our given range and it is very possible that the appropriate model will differ as a result. For these reasons, I have reservations about creating a confidence interval around 2 hours waiting time.

```
summary(CableData)
```

```
##     calltime       satisfaction
##  Min.   :0.0800   Min.   : 8.73
##  1st Qu.:0.8175   1st Qu.:52.40
##  Median :1.0100   Median :63.75
##  Mean   :1.0048   Mean   :62.58
##  3rd Qu.:1.2200   3rd Qu.:73.61
##  Max.   :1.8800   Max.   :97.04
```

```
range(Time.Trans)
```

```
## [1] -0.5087037  1.2441126
```

4.)

```
library(readr)
AlbuminData = read_csv("/Users/gustavo/Desktop/535 Exam 1/Albumin.csv")

Albumin = AlbuminData$Albumin
OpLength = AlbuminData$Operation.Length
Diabetic = AlbuminData$Diabetes
BMI = AlbuminData$BMI
```

a.) Determine whether or not body mass index (BMI) is a statistically significant predictor of albumin in the context of the other covariates?

We are attempting to determine if BMI is statistically significant predictor of Albumin in the context of Operation Length and Diabetes status, so we should begin by comparing two seperate models. The first should be a model with all of our predictors included, while the second should exclude BMI to observe how its' absense alters the model.

For our first model, including all three predictor variables, we have the following results:

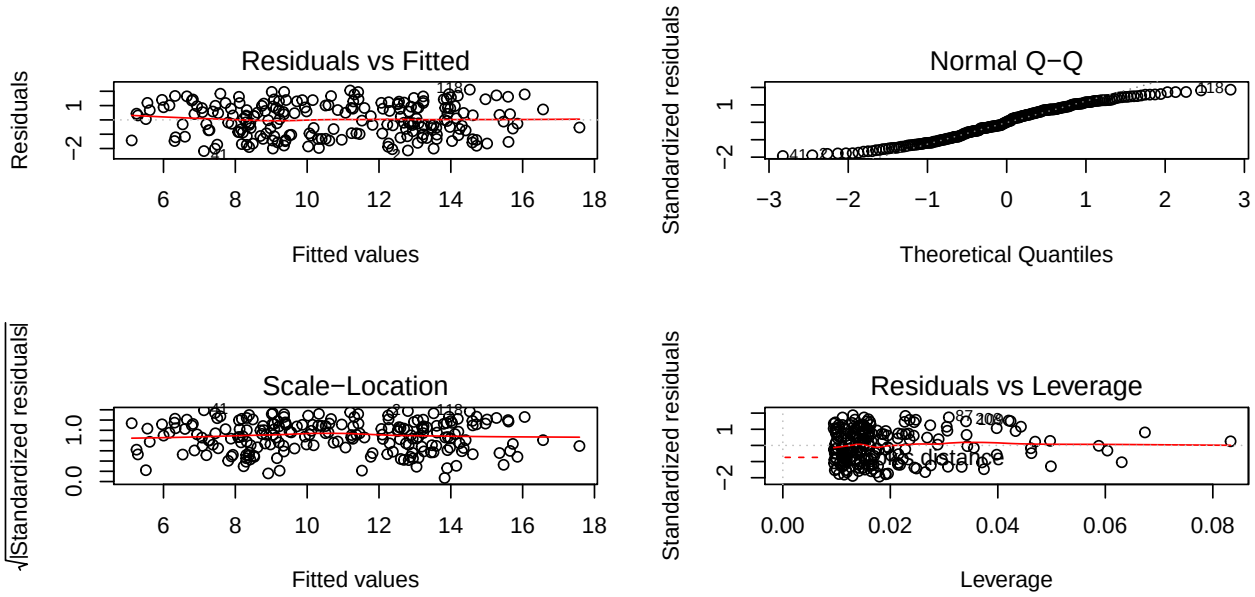
```
model3= lm(Albumin~OpLength + Diabetic + BMI)
summary(model3)
```

```
##
## Call:
## lm(formula = Albumin ~ OpLength + Diabetic + BMI)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.17850 -0.95855  0.02268  0.92349  2.11312
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23.05098    0.52873   43.60   <2e-16 ***
## OpLength    -0.66924    0.04714  -14.20   <2e-16 ***
## Diabetic    -4.88531    0.15681  -31.16   <2e-16 ***
## BMI         -0.27755    0.01645  -16.87   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.137 on 208 degrees of freedom
## Multiple R-squared:  0.8628, Adjusted R-squared:  0.8608
## F-statistic:   436 on 3 and 208 DF,  p-value: < 2.2e-16
```

From our summary of the model, we can immediatly note that the model itself is statistically valid according to the F-statistic and corresponding p-value. In addition, we can see that each of the predictor variables are also statistically significant. This tells us that the predictor and response variables are valid options for modeling the Albumin data provided. Once again, it's not the entire story but it's a quick snapshot of immediate results. We can also note the R^2 value of .8628, which is quite strong.

Now, we will dig deeper into the validity of our respective model by investigating the residual plots.

```
par(mfrow=c(2,2))
plot(model3)
```



The Residuals vs Fitted vales plot has our data points spread out along the x-axis, which is what we want to see in order to validly assume Linearity. No distinct pattern to really make out from the residuals, so we are happy with this plot.

The Normal-QQ plot has a great deal of deviation from the quantile line. We can observe an S shape, thus the assumption of Normality is clearly violated.

The Scale-Location plot should have the data points randomly spread across the x line, similar to the Residuals vs Fitted values, in order to validate constant variance of our model. In our plot, we do see the desired random scattering across the x line, so constant variance can be assumed.

For the model including all three of our predictors, including BMI, we see a statistically valid model with a high R^2 value (indicating a strong model). From our residual plots, we have the assumptions of linearity and constant variance validated. However, we have a strong violation of Normality present. We will move on to a model that excludes BMI as a predictor.

```
model4 = lm(Albumin~OpLength + Diabetic)
summary(model4)
```

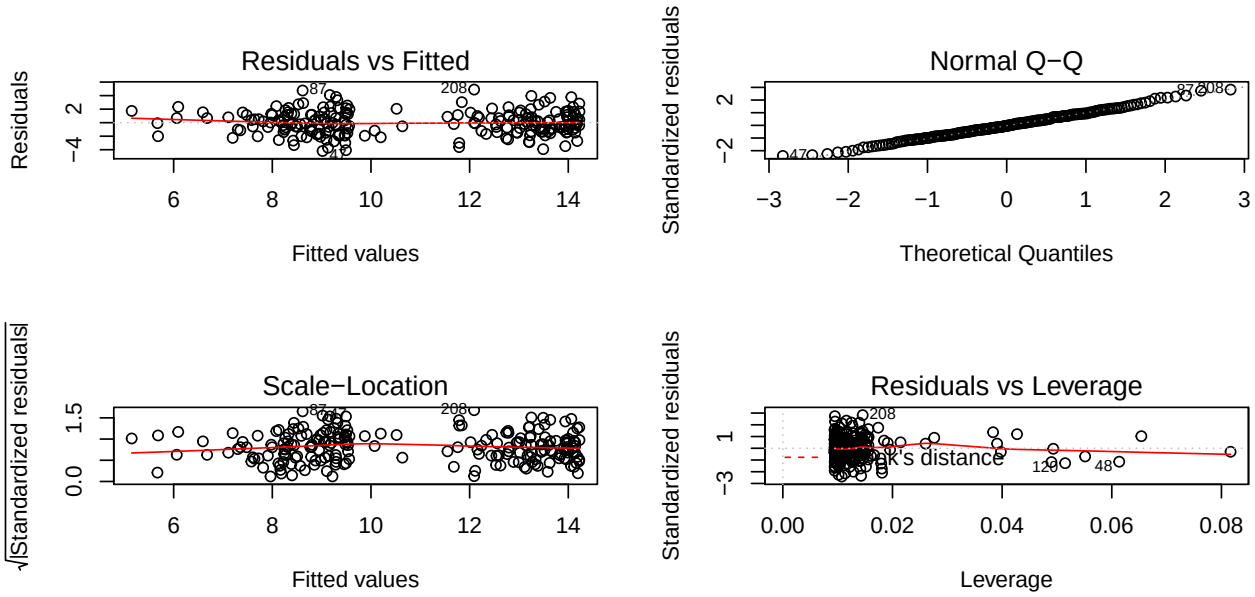
```
##
## Call:
## lm(formula = Albumin ~ OpLength + Diabetic)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1896 -1.2386 -0.0555  1.1555  4.8924
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14.54661    0.24518  59.330  < 2e-16 ***
## OpLength    -0.58479    0.07197  -8.126 3.85e-14 ***
## Diabetic    -4.68991    0.24010 -19.533  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.746 on 209 degrees of freedom
## Multiple R-squared:  0.675,  Adjusted R-squared:  0.6719
## F-statistic: 217.1 on 2 and 209 DF,  p-value: < 2.2e-16
```

From our summary of the model, we also see a statistically valid model according the F-statistic and p-values. We can also note that our R^2 value has dropped significantly by over 20%. Although adding any predictor will automatically increase the R^2 value, such a dramatic drop from one predictor being excluded is not something that can be brushed away so easily.

Now, we will dig deeper into the residuals of our respective model.

```
par(mfrow=c(2,2))
plot(model4)
```



Much like the previous model, we can see that the Residuals vs Fitted values and Scale-location plots have random scattering across their respective x-lines. Thus, linearity and constant variance can once again be assumed. In addition, we can see that the issue of Non-normality has been resolved. The QQ-plot for this model has the data points spread about the line in a much more consistent manner.

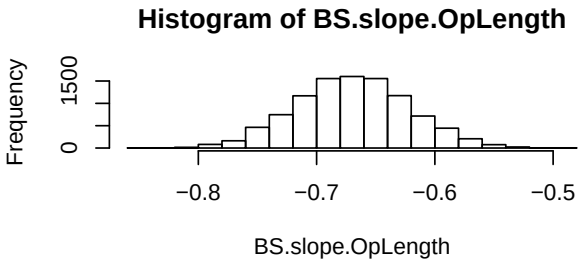
Although we have gained normality by dropping BMI as a predictor, we have lost a great deal of accuracy, according to the drop in R^2. We would ultimately like to keep the strongest model as an option, and we can do this by proving normality in a different manner than the qq-plots. By bootstrapping, we can show that the estimates and standard errors of the slopes, $\hat{b}$ found in the model including BMI are still able to use the benefits of the Normality assumption (ie any inference based on our model). This will done in the following manner:

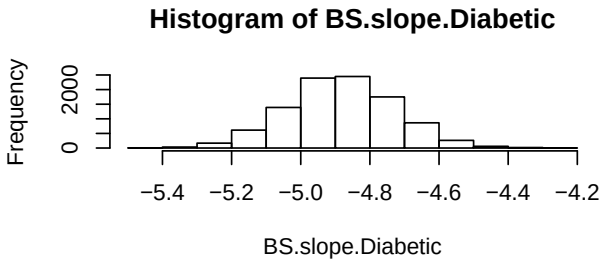
```
X=data.frame(OpLength,Diabetic,BMI)
n=length(Albumin)
residuals= as.numeric(model3$res)

BS.slope.OpLength = rep(0,10000)
BS.slope.Diabetic = rep(0,10000)
BS.slope.BMI = rep(0,10000)

for(i in 1:10000){
```

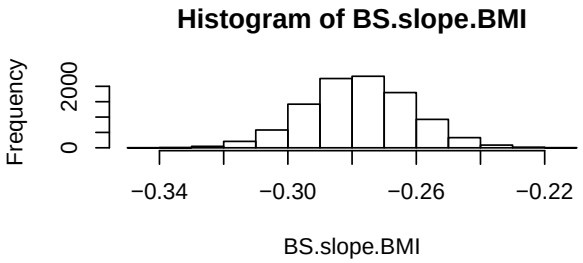
```
  new.x = X[sample(1:n,n,replace=T),]
  fit.y = predict(model3,new.x)

  new.y = fit.y + sample(residuals,n,replace=T)

  BS.slope.OpLength[i] = lm(new.y~new.x[,1]+new.x[,2]+new.x[,3])$coef[2]
  BS.slope.Diabetic[i] = lm(new.y~new.x[,1]+new.x[,2]+new.x[,3])$coef[3]
  BS.slope.BMI[i]      = lm(new.y~new.x[,1]+new.x[,2]+new.x[,3])$coef[4]
}
```

Here are histograms showing the distributions of each of the predictor slopes:

```
par(mfrow=c(2,2))
hist(BS.slope.OpLength)
hist(BS.slope.Diabetic)
hist(BS.slope.BMI)
```

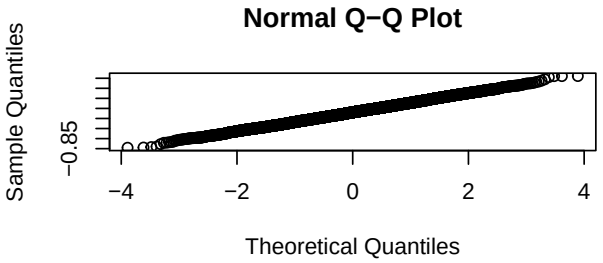


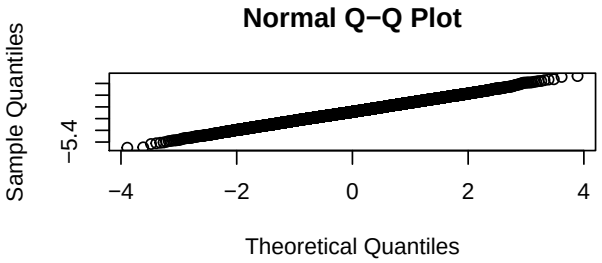


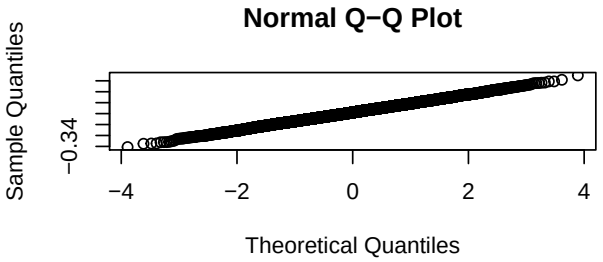


The slopes look normally distributed, our bootstrapping seems to work nicely. Now, let's see how the qq plots appear for each of our predictors.

```
par(mfrow=c(2,2))
qqnorm(BS.slope.OpLength)
qqnorm(BS.slope.Diabetic)
qqnorm(BS.slope.BMI)
```







Similar to the histograms, we can clearly see the assumption of normality is valid. Finally, let us compare the summary results from our model and our bootstrapping.

```
summary(model3)

Coefficients:
            Estimate        SE      t     Pr(>|t|)
(Intercept) 23.05098    0.52873   43.60   <2e-16 ***
OpLength    -0.66924    0.04714  -14.20   <2e-16 ***
Diabetic    -4.88531    0.15681  -31.16   <2e-16 ***
BMI         -0.27755    0.01645  -16.87   <2e-16 ***
```

```
mean(BS.slope.OpLength)
```

```
## [1] -0.6697287
```

```
sd(BS.slope.OpLength)
```

```
## [1] 0.04779861
```

```
mean(BS.slope.Diabetic)
```

```
## [1] -4.882601
```

```
sd(BS.slope.Diabetic)
```

```
## [1] 0.1555167
```

```
mean(BS.slope.BMI)
```

```
## [1] -0.2778295
```

```
sd(BS.slope.BMI)
```

```
## [1] 0.0163477
```

We can observe that the slopes and standard errors derived from our bootstrapping process are nearly identitcal to those provided in the model summary including BMI. Thus, we have shown that the assumption of Normality is valid and have preserved a stronger model by retaining a valuable predictor in the process. We can surely conclude that BMI is a statistically significant predictor in the context of the other variables.