STAT 302: Project 2

Jaiden Atterbury, Ty Mellish, Qinyi Zhong

Due on 06-08-2023

Problem 1:

Using the glass data set on Canvas - determined the regression equation for predicting the response (Y) being variable (Si).

Data description: https://archive.ics.uci.edu/ml/datasets/glass+identification

a) Select any three or four variables of your choice and use one visualization plot to tell a more complete and compelling story of the data set making use of all the variables selected. Consider using at least **color, faceting, theme** among many others in ggplot2. Write at least two paragraphs for this. One explaining why you choose those variables and the other what you see from the graph.

Answer:
For this problem we will select three or four variables and use one visualization to tell a more compelling story of the data. The plot we decided to make was a scatter plot with the response variable being Silicon (`Si`) and the independent variable being Sodium (`Ca`). Furthermore, the points in the scatter plot are faceted by glass type (`Type`) which is renamed for readability below, and colored by Refractive Index (`RI`). In this case the Refractive Index (`RI`) was converted into a categorical variable by being split into the categories low and high based on if the `RI` value was greater or less than the mean Refractive Index score.

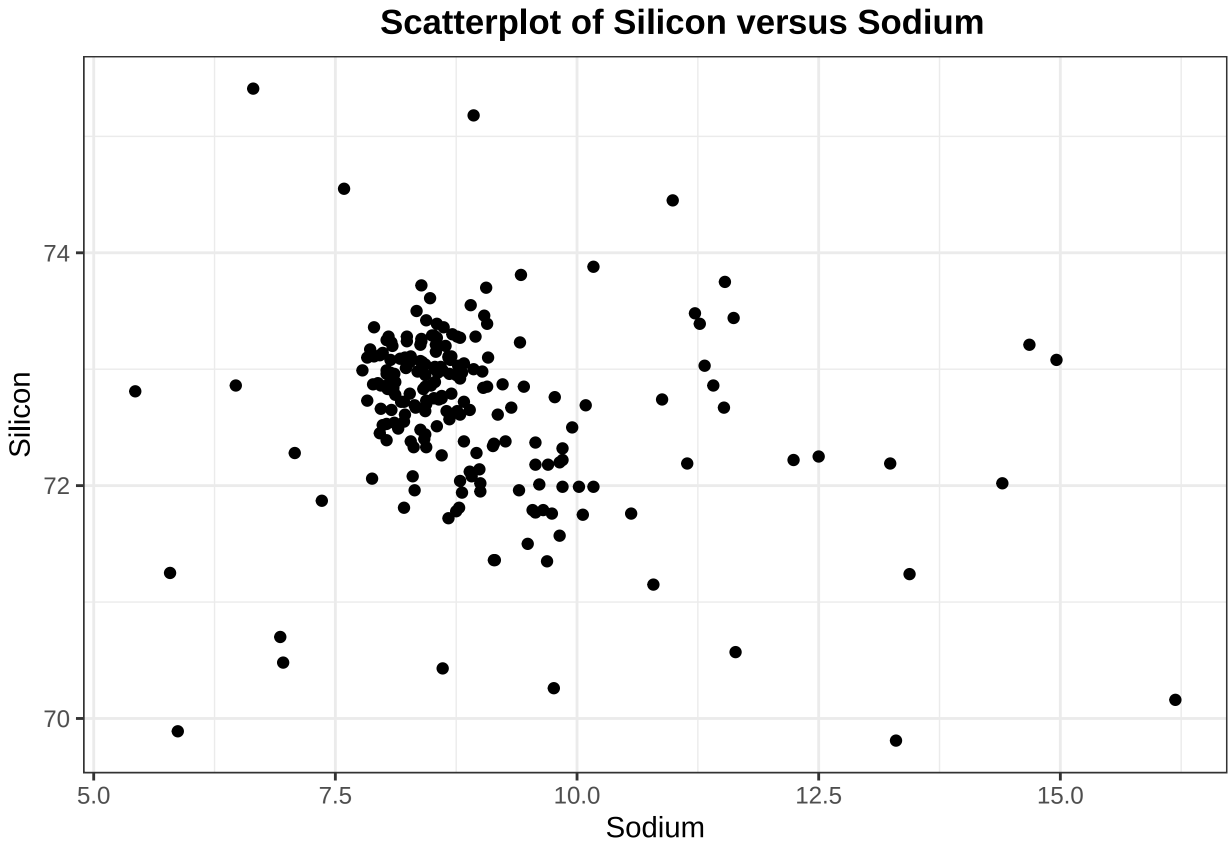
```r
# Create a new data frame to manipulate the columns:
Glass_cat <- Glass

# Re categorize the Refractive Index variable:
Glass_cat$Refractive_Index[Glass_cat$RI < 0.3654] <- "Low"
Glass_cat$Refractive_Index[Glass_cat$RI >= 0.3654] <- "High"

# Rename the glass type variable
Glass_cat$type_new[Glass_cat$type == "Con"] <- "Containers"
Glass_cat$type_new[Glass_cat$type == "Head"] <- "Headlamps"
Glass_cat$type_new[Glass_cat$type == "Tabl"] <- "Tableware"
Glass_cat$type_new[Glass_cat$type == "Veh"] <- "Vehicles"
Glass_cat$type_new[Glass_cat$type == "WinF"] <- "Windows Float"
Glass_cat$type_new[Glass_cat$type == "WinNF"] <- "Windows Non-Float"
```

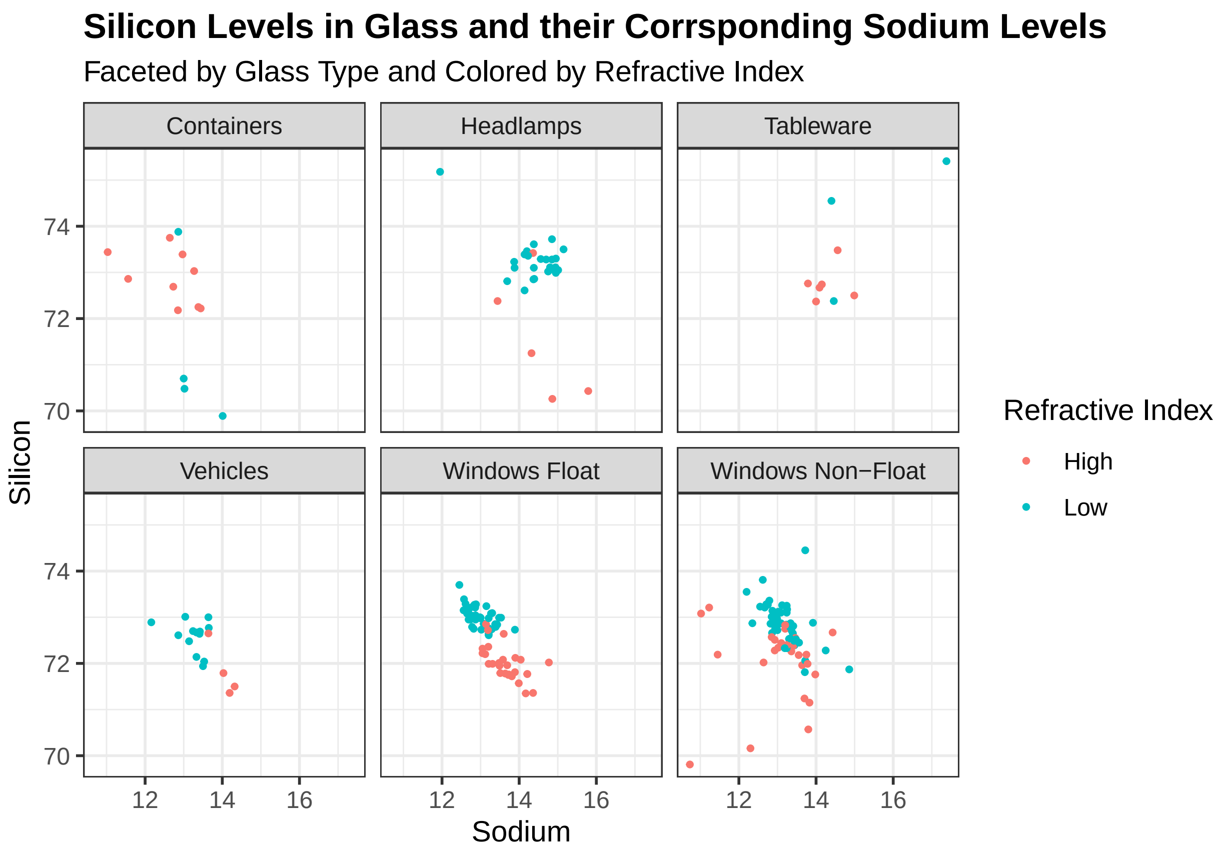
Before we create this complex visualization we will plot Silicon (`Si`) versus Sodium (`Ca`) by themselves to see if there are any patterns we can see without any of the faceting and coloring described above.

```r
# Create a scatter plot of Silicon against Sodium:
ggplot(data=Glass) +
  geom_point(mapping=aes(x=Ca, y=Si)) +
  labs(title="Scatterplot of Silicon versus Sodium",
       x="Sodium",
       y="Silicon") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.55, face = "bold", size = 13))
```



As can be seen by this introductory scatter plot there seems to be no linear relationship between the two variables, we are hoping by faceting and coloring the points that we can learn more about the relationship between these two variables. Below we will create a plot of these two variables with the given faceting and coloring described above.

```
ggplot(data = Glass_cat,
       aes(x = Na, y = Si, group = Refractive_Index, color = Refractive_Index)) +
  geom_point(size = 0.7) +
  labs(title = "Silicon Levels in Glass and their Corrsponding Sodium Levels",
       subtitle = "Faceted by Glass Type and Colored by Refractive Index",
       x = "Sodium",
       y = "Silicon",
       color = "Refractive Index") +
  facet_wrap(~type_new) +
  theme_bw() +
  theme(plot.title = element_text(face = "bold"))
```



We chose these particular variables for various reasons which we will describe now. For the Refractive Index, we felt that perhaps the level of refraction in the glass displayed could be connected to the makeup/type of the glass. Furthermore, we thought that maybe the RI could be used to predict the Silicon levels. For the type of glass, this would obviously be helpful in deciding whether the type of glass affects the Silicon levels. As for Sodium, we wanted some other kind of chemical component of the glass to see how other chemicals in the glass would affect the concentration of Silicon.

From the graph, we can make several observations. First of all, it seems that in most cases, an increase in Sodium (Na) leads to a decrease in Silicon (Si). This is shown in most of the best fit lines' slopes being negative. Notably, there are several cases where this trend is denied, with the Tableware type of glass having a positive relationship between Na and Si, and the High Refraction glass on Non-Float Buildings having no visible relationship, but this may be due to some outliers. Speaking of outliers, there are several extreme values in the Headlamps, Non-Float Buildings, and Tableware Categories. For Tableware, it could be that there are just too few values, but in the non-float building and headlamps, there exist some extreme values that may impact/influence the best fit lines. As for the Refractive Index scores, It seems that generally, a low Refractive Index leads to a higher value of Silicon in the glass, the only exception in this trend being the Containers type of glass, which may just be due to a lack of data. For the types of glass, it seems that the type of glass does not greatly effect the silicon levels, as each type stayed mostly concentrated within the range of 71-74, and no visible patterns in the Silicon levels between glass types.

b) Fit **at least two different models** and select the best competing model for statistical analyses. Argue carefully on your choice of the best model. Perform optimization in R to compute the regression coefficients of your best model.

Answer:
In this problem we will use `Si` as our dependent variable and we will fit two different models and select the

best competing model for statistical analysis. Once our best model is chosen we will perform optimization in R to compute the regression coefficients of our best model.

Since we know we will have to use linear algebra when using optimization packages to compute our regression coefficients, we will instantly remove the `type` variable from our analysis because it is categorical. Thus to determine which variables we will use as independent variables in the model, we will compute the correlation matrix of the data set to see which variables have the strongest linear relationship with `Si`. Once this is done we will select the top 3 and 4 variables with the highest correlation to be our two competing models.

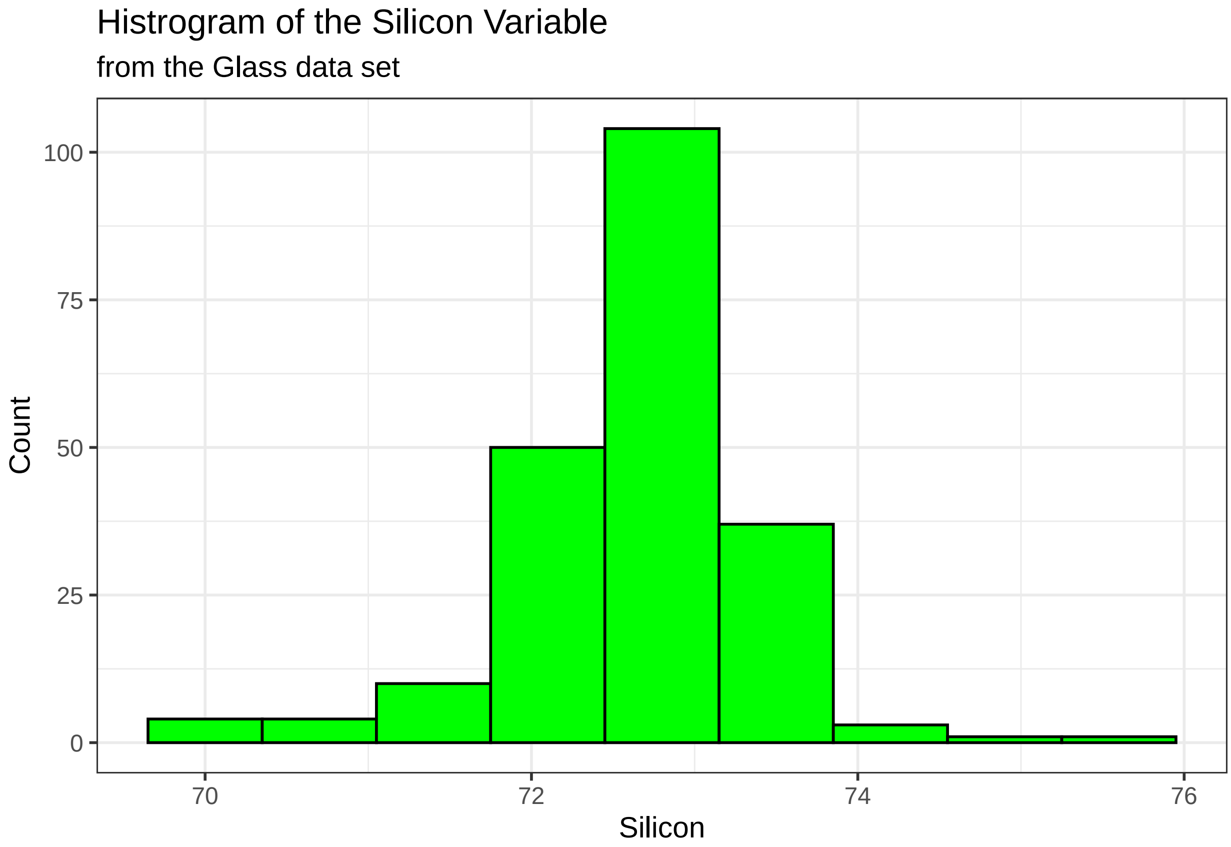
```r
# Remove the Type variable:
Glass_new <- Glass[ ,-c(10)]

# Compute a correlation matrix:
cor(Glass_new)
```

```
##               RI          Na           Mg          Al          Si            K
## RI  1.0000000000 -0.19188538 -0.122274039 -0.40732603 -0.54205220 -0.289832711
## Na -0.1918853790  1.00000000 -0.273731961  0.15679367 -0.06980881 -0.266086504
## Mg -0.1222740393 -0.27373196  1.000000000 -0.48179851 -0.16592672  0.005395667
## Al -0.4073260341  0.15679367 -0.481798509  1.00000000 -0.00552372  0.325958446
## Si -0.5420521997 -0.06980881 -0.165926723 -0.00552372  1.00000000 -0.193330854
## K  -0.2898327111 -0.26608650  0.005395667  0.32595845 -0.19333085  1.000000000
## Ca  0.8104026963 -0.27544249 -0.443750026 -0.25959201 -0.20873215 -0.317836155
## Ba -0.0003860189  0.32660288 -0.492262118  0.47940390 -0.10215131 -0.042618059
## Fe  0.1430096093 -0.24134641  0.083059529 -0.07440215 -0.09420073 -0.007719049
##            Ca            Ba           Fe
## RI  0.8104027 -0.0003860189  0.143009609
## Na -0.2754425  0.3266028795 -0.241346411
## Mg -0.4437500 -0.4922621178  0.083059529
## Al -0.2595920  0.4794039017 -0.074402151
## Si -0.2087322 -0.1021513105 -0.094200731
## K  -0.3178362 -0.0426180594 -0.007719049
## Ca  1.0000000 -0.1128409671  0.124968219
## Ba -0.1128410  1.0000000000 -0.058691755
## Fe  0.1249682 -0.0586917554  1.000000000
```

As shown in the correlation matrix, the variables with the highest correlation to `Si` are: `RI`, `Ca`, `K`, and `Mg` (in that order). With correlation coefficients of -0.54, -0.26, -0.19, and -0.17 respectively. Thus, our first model will contain `RI`, `Ca`, and `K` as independent variables, and lastly, our second model will contain `RI`, `Ca`, `K`, and `Mg` as independent variables. Before we fit any models, we will need to check the distribution of the dependent variable, we will do this through the use of a histogram.

```r
# Create a histogram of the dependent variable Si, use Sturge's rule to find the
# number of bins to use:
ggplot(data=Glass_new) +
  geom_histogram(mapping = aes(x=Si),
                 bins=9,
                 color="black",
                 fill="green") +
  labs(title="Histrogram of the Silicon Variable",
       subtitle="from the Glass data set",
       x="Silicon",
       y="Count") +
  theme_bw()
```



As can be seen from the above histogram, the distribution of the `Si` variable seems approximately normally distributed, thus we can continue on with the linear regression without making any data transformations. We will now create the two competing models below.

```
# Create model 1:
model_1 <- lm(Si ~ RI + Ca + K, data=Glass_new)

# Create model 2:
model_2 <- lm(Si ~ RI + Ca + K + Mg, data=Glass_new)
```

In order to see which model is our best model we will assess the output we obtain from using the `compareLM` function from the `rcompanion` package.

```
# Compare the three models:
compareLM(model_1, model_2)
```

```
## $Models
##   Formula
## 1 "Si ~ RI + Ca + K"
## 2 "Si ~ RI + Ca + K + Mg"
##
## $Fit.criteria
##   Rank Df.res   AIC  AICc   BIC R.squared Adj.R.sq   p.value Shapiro.W
## 1    4    210 338.2 338.4 355.0    0.5456   0.5391 9.203e-36    0.9430
## 2    5    209 339.1 339.5 359.3    0.5479   0.5393 5.426e-35    0.9479
##   Shapiro.p
## 1 1.915e-07
## 2 5.575e-07
```

As can be seen from the above output, the AIC and BIC of model 1 is slightly lower than the AIC and BIC of model 2 while only having only a slightly smaller adjusted R squared than model 2. Furthermore, model 1 is the simpler model. With all of that said, we will choose model 1 as our best model and use it for statistical analysis. The reason for the use of these criteria in model comparison is that both AIC and BIC are techniques used to find the balance between a good model fit and over complexity in a model. The reason why smaller models are preferred is that even though adding more variables increases the R squared of a model, the model also grows in complexity and risks over fitting the model to the sample data. Over fitting is not desirable as it limits the predicting power of the model, which is the main reason why the model was made in the first place. We will now show a quick summary of the model in order to see what the coefficients of the model should be before we optimize using R.

```
# Get a model summary of the best model
summary(model_1)
```

```
##
## Call:
## lm(formula = Si ~ RI + Ca + K, data = Glass_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2231 -0.2907 -0.0216  0.2679  2.8753
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 70.06254    0.39259 178.460  < 2e-16 ***
## RI          -0.28496    0.02028 -14.049  < 2e-16 ***
## Ca           0.32229    0.04369   7.376 3.71e-12 ***
## K           -0.39066    0.05836  -6.694 1.95e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5258 on 210 degrees of freedom
## Multiple R-squared:  0.5456, Adjusted R-squared:  0.5391
## F-statistic: 84.06 on 3 and 210 DF,  p-value: < 2.2e-16
```

Now that we see what the coefficients should be, we will now use the `solve.QP` from the `quadprog` package along with the linear algebra equations for multiple linear regression to use linear algebra and optimization to find the regression coefficients for our best model.

```
# Number of observations:
n <- nrow(Glass)

# Independent variables:
x1 <- Glass$RI
x2 <- Glass$Ca
x3 <- Glass$K

# Create the design matrix:
X <- cbind(rep(1, times = n), x1, x2, x3)
```

```r
# Dependent variable
y <- Glass$Si

# Use the linear algebra formula and optimization to find the coefficients:
s <- solve.QP(t(X) %*% X, t(y) %*% X, matrix(nr=4,nc=0), numeric(), 0)

# Output these coefficients:
s$solution
```

```
## [1] 70.0625372 -0.2849618  0.3222863 -0.3906555
```

As can be seen from above, the slope is 70.06, the slope estimate for `RI` is -0.28, the slope estimate for `Ca` is 0.32, and lastly the slope estimate for `K` is -0.39. These match the coefficients we obtained when using the `lm` function in R.

c) Use the results from your best model to answer the following questions:

- What is the coefficient of determination and what does it mean?

Since R squared increases every time you add an independent variable to the model, the R squared value always increases. Thus, since we are running a multiple regression model our coefficient of determination is the adjusted R squared. In our model, the adjusted R squared was 0.5391. Our adjusted R squared value means, adjusting for the number of independent variables and their corresponding significance, around 53.91% of the variability in Silicon (`Si`) can be explained by Refractive Index (`RI`), Calcium (`Ca`), and Potassium (`K`).

- Find the least-squares estimates for the regression line.

We will now share the least-squares estimates for the regression line along with their corresponding 90% confidence intervals.

```r
# Calculate the confidence intervals for the coefficients:
confint(model_1, level=0.9)
```

```
##                    5 %       95 %
## (Intercept) 69.4139150 70.7111595
## RI          -0.3184737 -0.2514498
## Ca           0.2501015  0.3944710
## K           -0.4870780 -0.2942330
```

As calculated through the use of two different R packages and the above confidence interval output, we can see that the intercept is estimated to be 70.06254 with a 90% confidence interval of [69.4139150, 70.7111595]. The Refractive Index (`RI`) variable has an estimated slope of -0.28496 with a 90% confidence interval of [-0.3184737, -0.2514498]. The Calcium (`Ca`) variable has an estimated slope of 0.32229 with a 90% confidence interval of [0.250101, 0.3944710]. Lastly, the Potassium (`K`) variable has an estimated slope of -0.39066 with a 90% confidence interval of [-0.4870780, -0.2942330]. With that said, our multiple linear regression model takes the form:

$$\widehat{\text{Si}} = 70.06 - 0.28 \cdot \text{RI} + 0.32 \cdot \text{Ca} - 0.39 \cdot \text{K}$$

- Interpret the value of the slopes and intercept in the context of the problem. In addition, state which variables are significant predictors at the 10% level of significance? Explain.

As computed above, the regression slope for Refractive Index (`RI`) was -0.28496 with standard error 0.02028 and with a 90% confidence interval of [-0.3184737, -0.2514498]. This means, on average, for every 1 unit increase in the Refractive Index of the glass, the mean Silicon content of the glass is estimated to decrease by about 0.28496 units over the sampled range of Silicon content values, holding all other variables in the model fixed. Furthermore, the p-value for this estimate was < 2e-16 which means this estimate is highly significant at the 10% level of significance.

As computed above, the regression slope for Calcium (`Ca`) was 0.32229 with a standard error of 0.04369 and with a 90% confidence interval of [0.250101, 0.3944710]. This means that on average, for every 1 unit increase in the Calcium content of the glass, the mean Silicon content of the glass is estimated to increase by about 0.32229 units over the sampled range of of Silicon values, holding all other variables in the model fixed. Furthermore, the p-value for this estimate was 3.71e-12 which means this estimate is highly significant at the 10% level of significance.

As computed above, the regression slope for Potassium was -0.39066 with a standard error of 0.05836 and with a 90% confidence interval of [-0.4870780, -0.2942330]. This means that on average, for every 1 unit increase in the Potassium content of the glass, the mean Silicon content of the glass is estimated to decrease by about 0.39066 units over the sampled range of of Silicon values, holding all other variables in the model fixed. Furthermore, the p-value for this estimate was 1.95e-10 which means this estimate is highly significant at the 10% level of significance.

As computed above, the regression y-intercept was 70.06254 with a standard error of 0.39259 with a 90% confidence interval of [69.4139150, 70.7111595]. This means that the estimated mean Silicon content in the glass is equal to about 70.06254 when all of the predictor variables are zero. Furthermore, the p-value for this estimate was < 2e-16 which means this estimate is practically zero and thus highly significant, especially at the 10% level of significance.

- Perform a residual analysis to decide whether considering the assumptions for regression inferences met by the variables in the data set appears reasonable.

We will now perform a residual analysis to decide whether considering the assumptions for regression inferences met by the variables in the data set appears reasonable. We will start off by checking the linearity assumption. Furthermore, all tests will be conducted at the 10% significance level to stay consistent with the previous parts of the analysis.

Checking for Linearity:

```r
# Run a Rainbow test to check the linearity of the variables:
raintest(model_1)
```

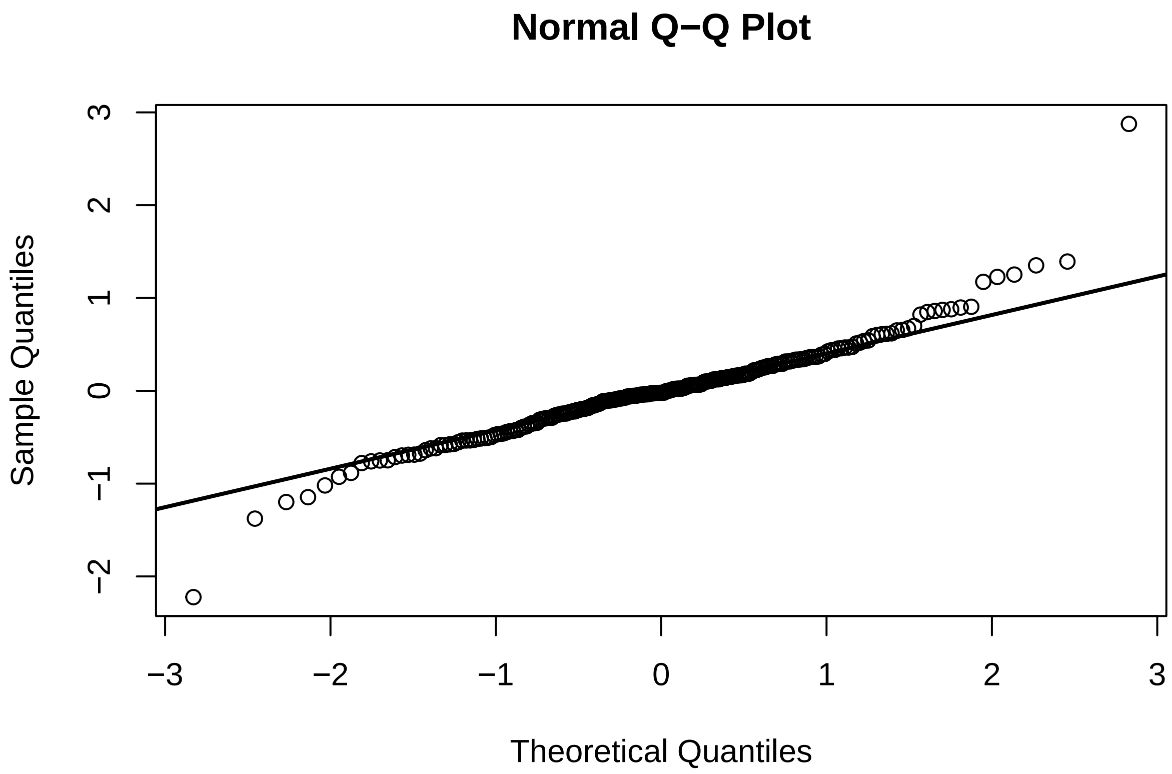
```
##
##  Rainbow test
##
## data:  model_1
## Rain = 1.8561, df1 = 107, df2 = 103, p-value = 0.0008773
```

As can be seen from the above Rainbow test, since the p-value is 0.0008773 which is less than 0.1, hence we reject the null hypothesis and we assume that the relationship between the independent and dependent variables is non-linear at the 10% level of significance.

Checking for Normality of the Residuals:

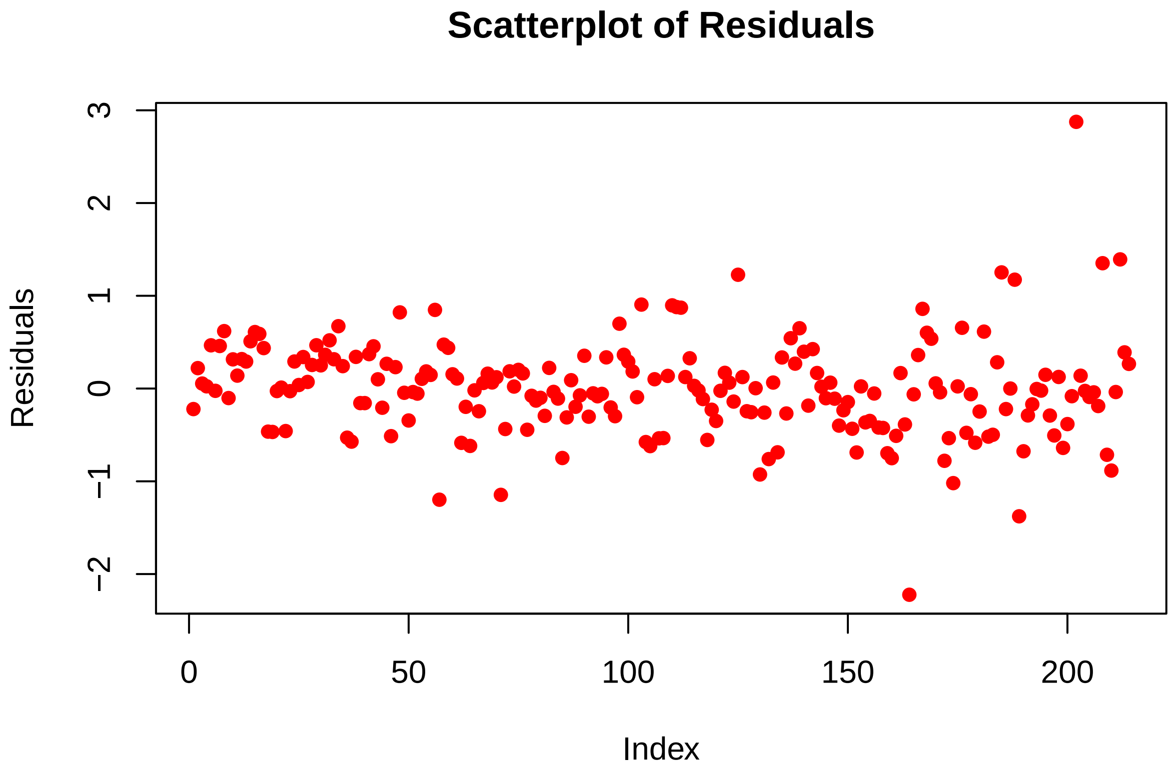
```r
# Obtain the residuals of the best model:
resid <- residuals(model_1)
```

```
# QQ-plot of the residuals:
qqnorm(resid)
qqline(resid,  lwd = 2)
```



As can be seen from the above QQ-plot of the residuals, most of the quantiles fall on the 45 degree line with small departures near the end and only two significant departures. We should be safe to assume the normality of the residuals but we will also create a scatter plot of the residuals to check this assumption as well.

```
# Scatter plot of the residuals:
plot(resid, pch = 16, col = "red", main="Scatterplot of Residuals",
     xlab="Index", ylab="Residuals")
```



Since there is no obvious pattern apparent in the scatter plot of the residuals, paired with the decent QQ-plot, we will assume the residuals are normally distributed.

Checking for equal variance:

```
# Run the Breusch Pagan Test for Heteroskedasticity to test for equal variance:
ols_test_breusch_pagan(model_1)
```

```
##
##  Breusch Pagan Test for Heteroskedasticity
##  -----------------------------------------
##  Ho: the variance is constant
##  Ha: the variance is not constant
##
##              Data
##  ------------------------------
##  Response : Si
##  Variables: fitted values of Si
##
##         Test Summary
##  -------------------------------
##  DF            =    1
##  Chi2          =    11.15609
##  Prob > Chi2   =    0.0008375605
```

As can be seen from the above Breusch Pagan Test for Heteroskedasticity, since the p-value is 0.0008375605 which is less than 0.1, we reject the null hypothesis, thus we have evidence that the equal/constant variance assumption is violated at the 10% level of significance.

Checking for Autocorrelation:

```
# Durbin-Watson test for autocorrelation:
dwtest(model_1)
```

```
##
##  Durbin-Watson test
##
## data:  model_1
## DW = 1.6814, p-value = 0.007196
## alternative hypothesis: true autocorrelation is greater than 0
```

As can be seen from the above Durbin-Watson test, since the p-value is 0.007196 which is less than 0.1, we reject the null hypothesis and we see that there is evidence that the residuals are auto-correlated at the 10% level of significance.

Checking for Multicollinearity:

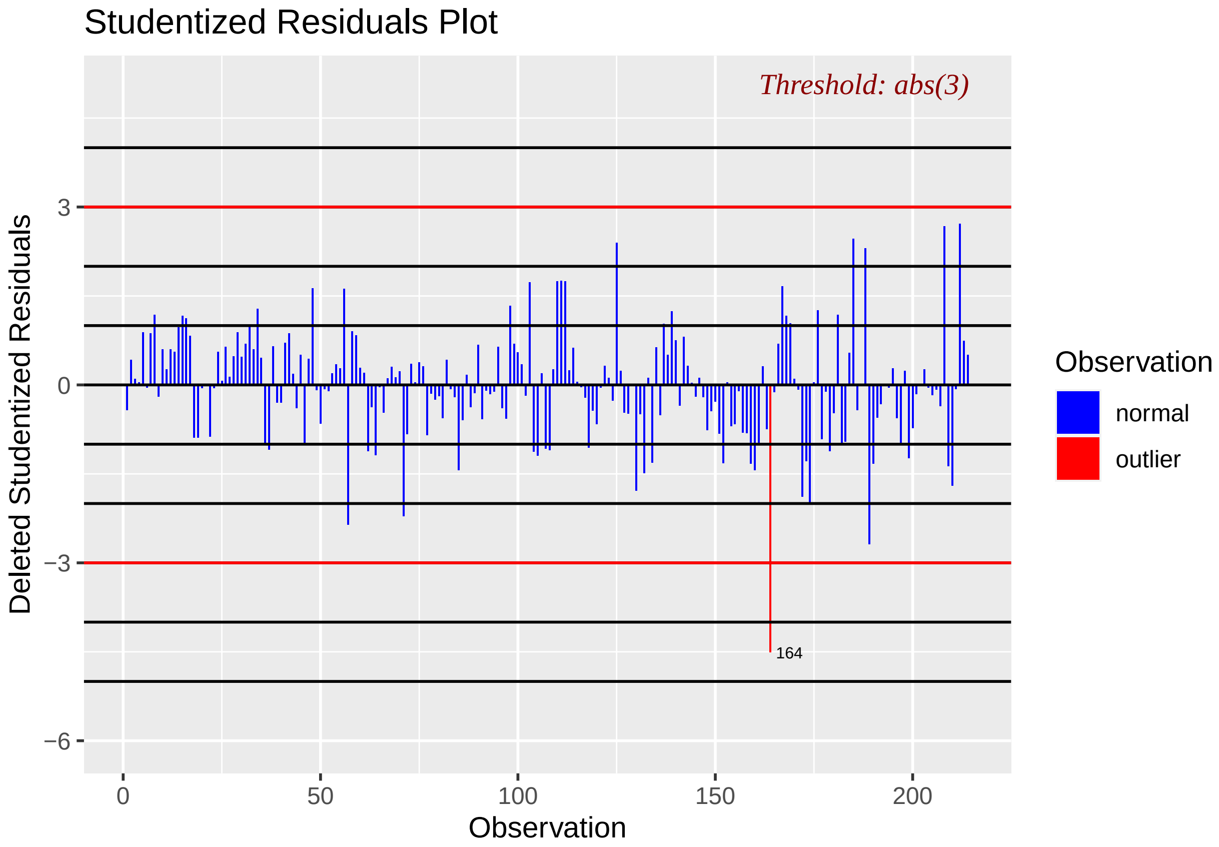
```
# Checking the Variance Inflation Factors of the model variables:
vif(model_1)
```

```
##       RI       Ca        K
## 2.923207 2.978541 1.116135
```

As can be seen from the above vif tests, the multicollinearity for each variable is below 5, hence we can assume that there is little to no multicollinearity between the variables, and hence we don't violate the assumption.

Checking for no outliers:

```
# Create a Studentized Residual Plot to check for outliers:
ols_plot_resid_stud(model_1)
```



As can be seen from the above studentized residual plot, there is only one outlier/influential point present in the data. Given the sheer size of the data set this shouldn't be too much of an issue but technically the assumption is still violated.

Due to the fact that we violated the linearity, autocorrelation, equal variance, and the no outliers assumptions, it is safe to assume that the assumptions for regression inference are violated and we would have to do some extra work to fit this model/use it for prediction.

d) Separate observations in two groups: `group 1: type=WinF` and `group 2: type=WinNF`. Based on variable `Al`, are the distributions of the two groups significantly different under significance level $\alpha = 0.10$?

Answer:
To check if the distributions of the two groups are significantly different at the significance level $\alpha = 0.10$ we will compare the centers of the two distribution, which in this case we will take to be the mean. Specifically we will use two methods; a two sample t-test, and if there is any doubt with the normality of the two groups in the t-test, we will run a permutation test.

In this part of the problem we will use the t.test function to test the null hypothesis that the mean Aluminum content of Group 1: `WinF` is equal to the mean Aluminium of Group 2: `WinNF`. The competing alternative hypothesis will be a two-sided hypothesis. Before we run this test we will check our model assumptions.

First off since the dependent variable is the Aluminium (`Al`) content of the pieces of glass it is measured on an interval/ratio scale. Secondly, the independent variable has two discrete levels; glass type `WinF` (float processed) and `WinNF` (non-float processed). Next, since these two types of glass are manufactured in different ways and appeared in different places we can safely assume the two groups being compared are independent of each other. We can assume independence because the Silicon content of one piece of glass doesn't impact the Silicon content of another piece of glass, given that they weren't from the same original piece of glass. We will now reconstruct the data for the tests.

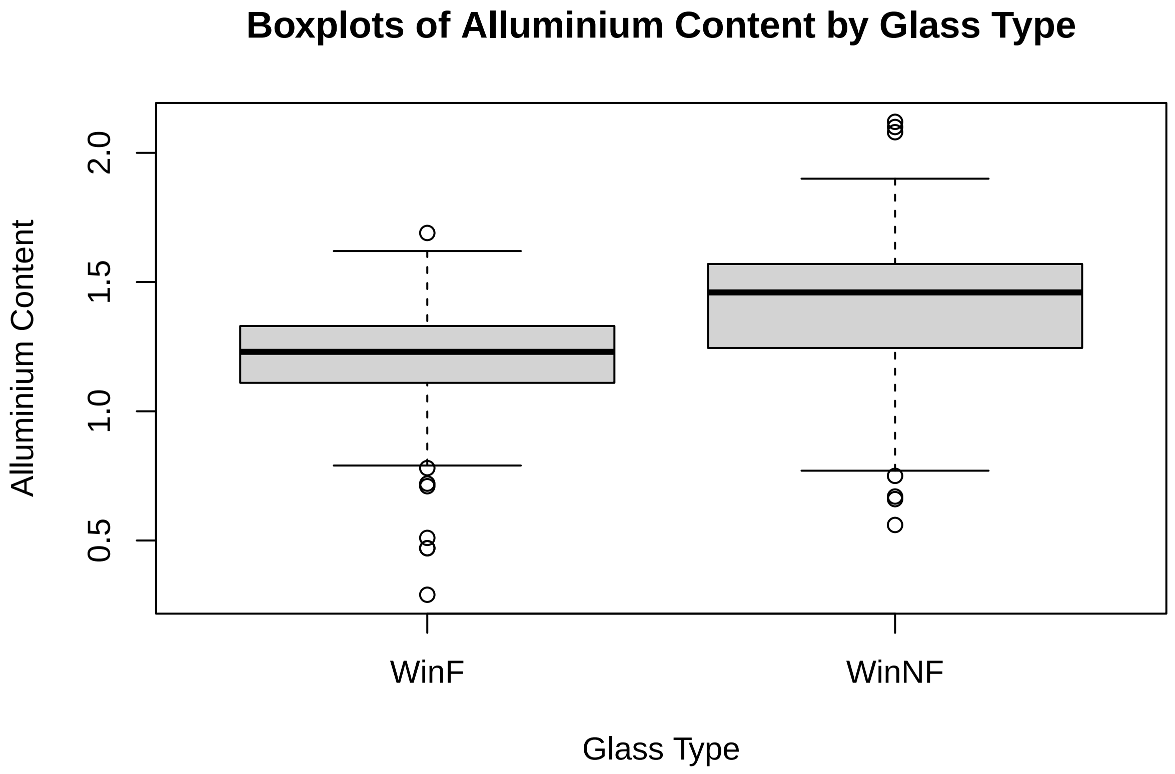
Reconstruct the data for the tests:

```r
# Filter the Glass data set to only take the data points in which the type is
# either WinF or WinNF. Also only select the variables of interest; AI and type.
Glass_new <- Glass %>%
  filter(type=="WinF" | type=="WinNF") %>%
  select(Al, type)

# Separate the two types of glass:
group_1 <- Glass_new$Al[Glass_new$type == "WinF"]
group_2 <- Glass_new$Al[Glass_new$type == "WinNF"]
```

All of the tests for this problem will be conducted at the 10% level of significance. First off, we will test the normality of the different glass types.

```r
boxplot(Glass_new$Al ~ Glass_new$type,
        main="Boxplots of Alluminium Content by Glass Type",
        xlab="Glass Type",
        ylab="Alluminium Content")
```



As can be seen from the above box plots the response variable is approximately normally distributed across both groups if we ignore the few outliers on both tails. However, it could be argued that the use of a t-distribution isn't justified due to these outliers. However, for the sake of the problem we will continue with the assumption that the dependent variable is normally distributed in order to run a two sample t-test. After the t-test we will run a permutation test to be safe/validate our results from the t-test. We will now test the equal variance assumption of the two groups.

Equal variance assumption:

```
# Run a F test for equal variance assumption:
var.test(group_1, group_2)
```

```
##
##  F test to compare two variances
##
## data:  group_1 and group_2
## F = 0.73628, num df = 69, denom df = 75, p-value = 0.1989
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.4630531 1.1761064
## sample estimates:
## ratio of variances
##          0.7362831
```

As can be seen from the above F test to compare two variances, the p-value is 0.1989 which is greater than 0.1, thus we fail to reject the null and assume that the two groups have similar variances. We will now run the t-test using the `t.test` function in R using the parameter `var.equal=TRUE` since we are assuming equal variance.

Run the two sample t-test:

```
# Run the two sample t-test to compare the means/distribution:
t.test(Al ~ type, data=Glass_new, var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  Al by type
## t = -4.9562, df = 144, p-value = 1.992e-06
## alternative hypothesis: true difference in means between group WinF and group WinNF is not equal to
## 95 percent confidence interval:
##  -0.3417303 -0.1468712
## sample estimates:
##  mean in group WinF mean in group WinNF
##            1.163857            1.408158
```

As can be seen from the above two sample t-test, the p-value was 1.992e-06 with a t-statistic value of 0.37032 on 144 degrees of freedom. This means that at the 10% level of significance we reject the null hypothesis. Hence we have evidence that the means of the two groups are different, and thus assume that their distributions are not the same.

A permutation test is a non-parametric method for testing the null hypothesis that the population distributions of two independent samples are identical without specifying their shape. Since the normality assumption across the levels of the dependent variable couldn't be justified with certainty, we will run a permutation test to answer this question and try to validate the conclusion made using the t-test.

Below we will write code to run a permutation test to test the same hypothesis as above. Namely we're testing $H_0 : \mu_1 - \mu_2 = 0$ versus $H_A : \mu_1 - \mu_2 \neq 0$, where μ_1 is the mean aluminium content from Group 1, and μ_2 is the mean aluminium content from Group 2.

Run the permutation test:

```r
# Set the seed for reproducibility:
set.seed(123)

# Initialize empty difference vector:
permuted_diff <- numeric()

# Run a permutation test:
for (i in 1:10000) {
  # For each iteration, scramble the label variable:
  permuted_data <- Glass_new %>% mutate(type = sample(type))

  # Compute the difference in means:
  diff <- mean(permuted_data$Al[permuted_data$type == "WinF"]) -
          mean(permuted_data$Al[permuted_data$type == "WinNF"])

  # Store the difference in means:
  permuted_diff[i] <- diff
}
```

Now that we have run the permutation test we will calculate the permuted p-value below.

Compute the permuted p-value:

```r
# Compute the observed difference in means from the data:
obs_diff <- mean(Glass_new$Al[Glass_new$type == "WinF"]) -
            mean(Glass_new$Al[Glass_new$type == "WinNF"])

# Find the permuted p-value:
mean(abs(permuted_diff) > abs(obs_diff))
```

```
## [1] 0
```

As computed above, there were no times in which the difference between the means from the generated samples exceeded the difference between the means that we observed. Hence out empirical p-value is thus 0.

However since we observed a p-value of 0, there is one work around to this that will allow us to obtain a non-zero p-value. Since we already know that we have one difference between means that is at least as big as the difference we observed, which is just the difference we observed, we can think of our simulated p-value as the number of simulated values that are greater than our observed value plus 1 over the number of simulated values plus 1. Thus we can say our p-value is actually $\frac{1}{10001} = 0.0001$.

This validates our conclusion from the t-test and hence this means that at the 10% level of significance we reject the null hypothesis. Hence we have evidence that the means of the two groups are different, and thus assume that their distributions are not the same.

e) Perform a simulation analyses to mimic your best model. Explain your choice of values (it's completely up to you but should be reasonable values). Calculate the intercept and slopes for the least-squares regression line for the simulated data. Examine the residuals from the real data and the simulated data. Make sure to comment.

Answer:
In this problem we will perform a simulation analyses to mimic our best model. In particular we will calculate the intercept and slopes for the least-squares regression line of the simulated data and examine the residuals from the real data and the simulated data.

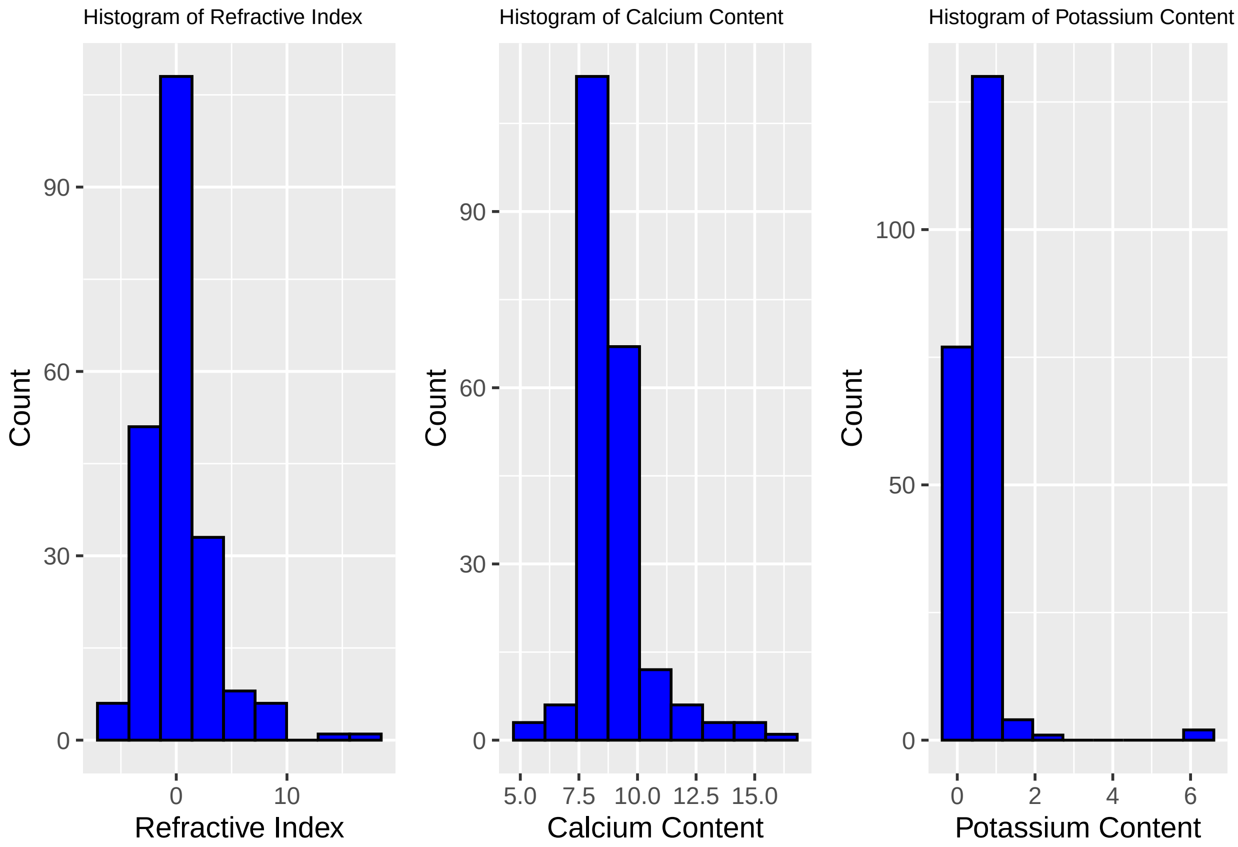
To mimic the results of our best model using simulation, we will follow the approach that we did in Lab 6 in which we assume all of our variables are normally distributed with the corresponding means and standard deviation of the normal distribution being the mean and standard deviation of the given variable from the data. Furthermore, we will use the coefficients we found in our best model, but in addition we will add error terms that will be simulated from a normal distribution with mean 0 and standard deviation equal to the standard deviation of the residuals of our best model. Although we already know the constant variance assumption was violated, we do know the residuals were normally distributed, thus this simulating the residuals this way has some justification. To assess the reasonableness of the assumption that all of the data is normally distributed we will plot histograms of all of our independent variables in order to check their distributions.

```r
# Histogram of Refractive Index:
p1 <- ggplot(data=Glass) +
  geom_histogram(mapping = aes(x=RI), bins=9, color="black",
                 fill="blue") +
  labs(title="Histogram of Refractive Index",
       x="Refractive Index",
       y="Count") +
  theme(plot.title=element_text(size=8))

# Histogram of Calcium content:
p2 <- ggplot(data=Glass) +
  geom_histogram(mapping = aes(x=Ca), bins=9, color="black",
                 fill="blue") +
  labs(title="Histogram of Calcium Content",
       x="Calcium Content",
       y="Count") +
  theme(plot.title=element_text(size=8))

# Histogram of Potassium content:
p3 <- ggplot(data=Glass) +
  geom_histogram(mapping = aes(x=K), bins=9, color="black",
                 fill="blue") +
  labs(title="Histogram of Potassium Content",
       x="Potassium Content",
       y="Count") +
  theme(plot.title=element_text(size=7.5))

# Create plotting grid:
grid.arrange(p1, p2, p3, nrow=1, ncol=3)
```



As can be seen by the above histograms, we can safely assume that the Refractive Index (RI) and Calcium (Ca) variables are approximately normally distributed. However, it is harder to justify that the Potassium (K) variable is approximately normally distributed. Notice that because of the big outlier in the Potassium variable it is hard to see the natural/true shape of the distribution. However, we will still assume that it is approximately normal due to the fact that we have no insights on its actual distribution. Below we will simulate these variables and error terms as well as run a linear model on these simulated terms.

```
# Set the seed for reproducibility:
set.seed(123)

# Simulate the data set assuming all of the data is normally distributed:
RI <- rnorm(214,  mean(Glass$RI), sd(Glass$RI))
Ca <- rnorm(214, mean(Glass$Ca), sd(Glass$Ca))
K <- rnorm(214,  mean(Glass$K), sd(Glass$K))

# Simulate the error terms:
e <- rnorm(214, 0, sd(resid))

# Create the dependent variable:
Si <- 70.062 - 0.285*RI + 0.322*Ca - 0.391*K + e

# Create the simulated model:
model_sim <- lm(Si ~ RI + Ca + K)

# Get the summary of the simulated model:
summary(model_sim)
```

```
##
## Call:
## lm(formula = Si ~ RI + Ca + K)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.33510 -0.34280  0.03491  0.33562  1.36290
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 69.96864    0.23697  295.26  < 2e-16 ***
## RI          -0.28389    0.01277  -22.23  < 2e-16 ***
## Ca           0.33655    0.02586   13.01  < 2e-16 ***
## K           -0.43208    0.05670   -7.62 8.55e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5336 on 210 degrees of freedom
## Multiple R-squared:  0.7823, Adjusted R-squared:  0.7792
## F-statistic: 251.6 on 3 and 210 DF,  p-value: < 2.2e-16
```

As can be seen from the above model output the intercept is 69.96864, the estimated `RI` slope is -0.28389, the estimated `Ca` slope is 0.33655, and the estimated `K` slope is -0.43208. All of these estimates are extremely significant at the 10% level of significance. Thus our simulated model equation is:

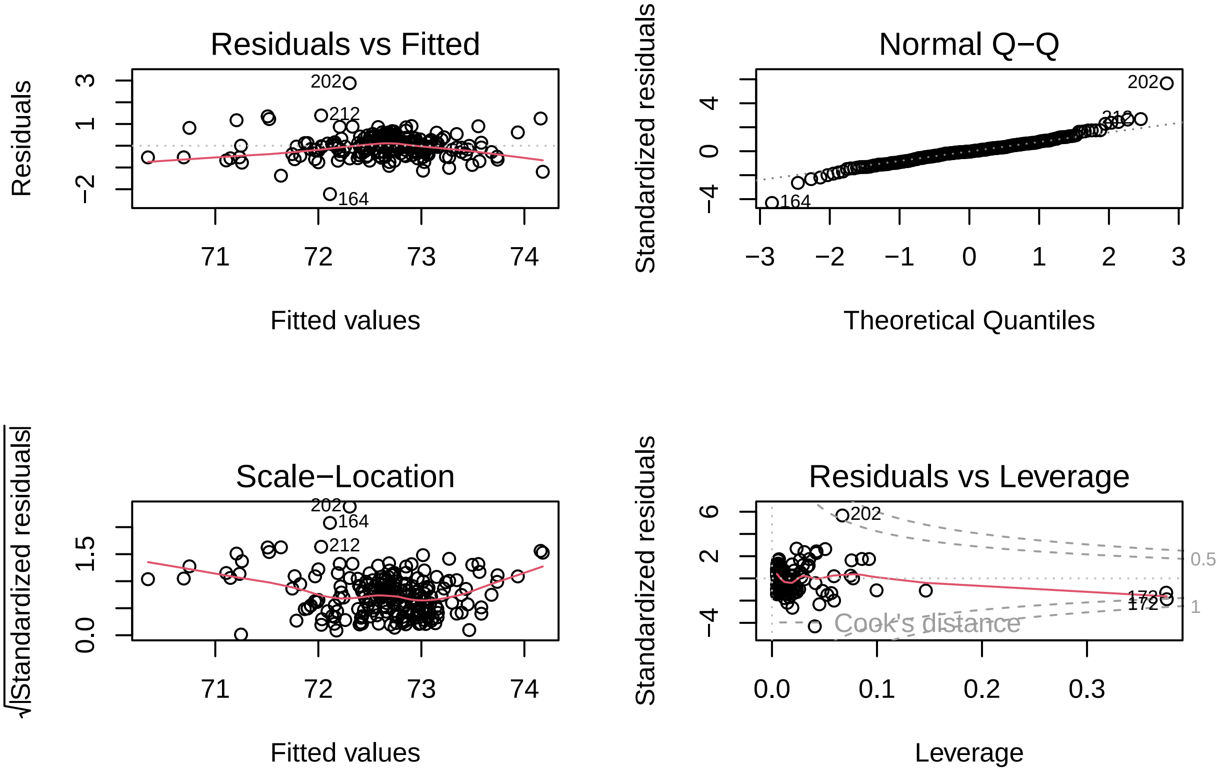
$$\widehat{\text{Si}} = 69.97 - 0.28 \cdot \text{RI} + 0.34 \cdot \text{Ca} - 0.43 \cdot \text{K}$$

Notice that this is very similar to the equation/coefficient estimates that we obtained from our actual best model. This gives us some assurance that the assumptions we made were justifiable. Lastly, we will compare the residuals from the real data and the simulated data.

Residuals from the real data:

```r
# Set up the frame:
par(mfrow = c(2, 2))

# Plot the residual assumptions for model 1:
plot(model_1)
```



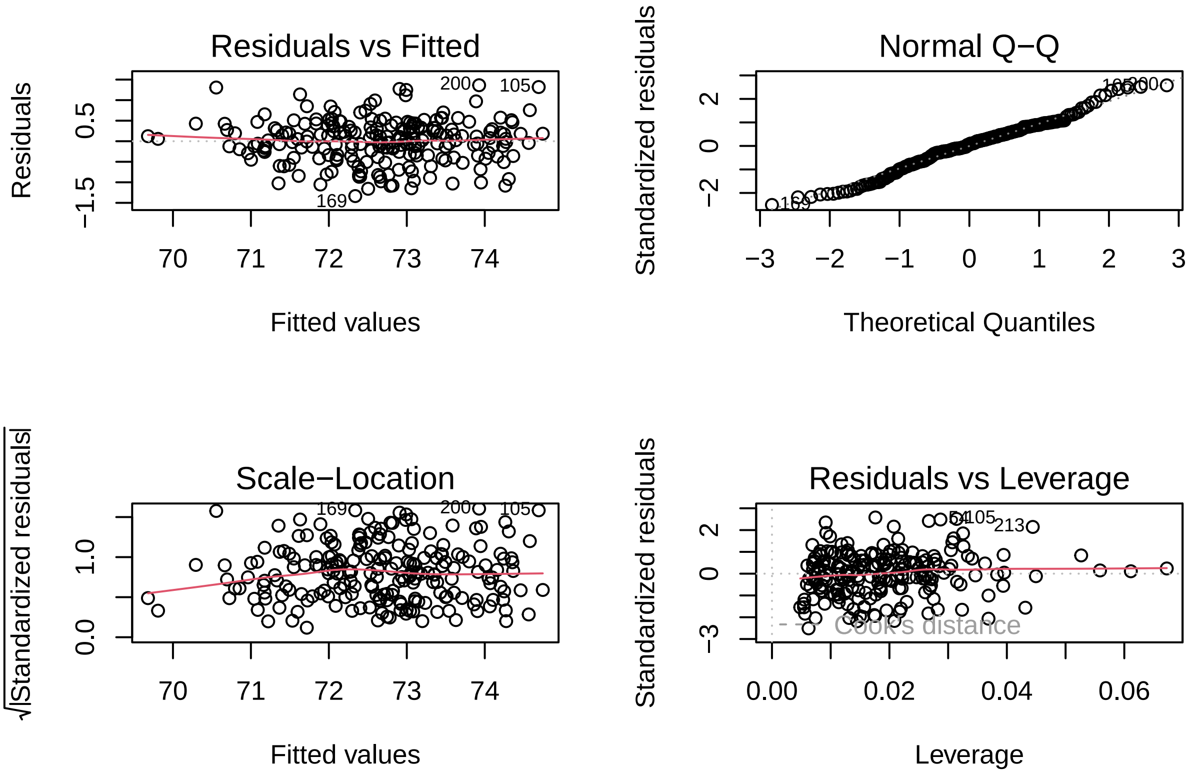
As can be seen from the above plots, due to the fact that the residuals vs fitted plot shows no patterns and seems to be evenly spaced around the red line, we assume that the linearity assumption holds. Next, since we see that the standardized residuals line up pretty well with the 45 degree line, we assume that the normality of the residuals assumption is not violated. Due to the fact that the scale-location plot shows a somewhat curvilinear pattern and seems to be not evenly spaced around the red line, we assume that the constant variance assumption is violated. Lastly, since a few of the standardized residuals fall past cook's distance we assume that the extreme outliers assumption is also violated. These model diagnostics all agree with the manual residual assumption tests we did above.

Below we will assess the model assumptions for the simulated model.

Residuals from the simulated data:

```r
# Set up the frame:
par(mfrow = c(2, 2))

# Plot the residual assumptions for model 2:
plot(model_sim)
```



As can be seen from the above plots, due to the fact that the residuals vs fitted plot shows no patterns and seems to be evenly spaced around the red line, we assume that the linearity assumption holds. Next, since we see that the standardized residuals line up perfectly with the 45 degree line, we assume that the normality of the residuals assumption holds. Also, due to the fact that the scale-location plot shows no patterns and seem to be evenly spaced around the red line, we assume that the constant variance assumption holds. Lastly, since none of the standardized residuals fall past cook's distance we assume that the no extreme outliers assumption holds. All of these residual assumptions are as we'd expect since we purposely made all of the model assumptions hold for our simulated model.

As can be seen above, both models may have had similar regression equations, however the real model violates some of the regression assumptions while our simulated model doesn't violate any of these assumptions.

Problem 2:

a) Using the **Handout 1 data set** on Canvas. Estimate the the prediction error and the prediction error rate of the model using either (i) Training and Testing or (ii) 5-fold cross-validation. The description of the data set can be found on Canvas and select your own independent variables to use.

Answer:
In this problem we will use the Handout 1 data set from Canvas and estimate the prediction error and prediction error rate of a model using a training and testing set. The model we will use will be selected in a particular way, this method is explained below.

Since we know there is no missing data and quite a few data points/the sample size is large we will keep all of our categorical variables as independent variables. However, for our continuous variables we will make a correlation matrix to assess which variables we will keep.

```
# Remove the categorical variables:
cat_vars_removed <- Handout_1[, -c(3, 4, 5)]

# Create a correlation matrix with all of the continuous variables:
cor(cat_vars_removed)
```

```
##                 COMMIT         AGE      SALARY   CLASSSIZE   RESOURCES   AUTONOMY
## COMMIT     1.0000000  0.18094973  0.38070161 -0.36595143  0.32994994 0.30979764
## AGE        0.1809497  1.00000000  0.10563458  0.04912606  0.17689868 0.08721293
## SALARY     0.3807016  0.10563458  1.00000000 -0.52143944  0.39674186 0.01661954
## CLASSSIZE -0.3659514  0.04912606 -0.52143944  1.00000000 -0.21293023 0.25116551
## RESOURCES  0.3299499  0.17689868  0.39674186 -0.21293023  1.00000000 0.03983696
## AUTONOMY   0.3097976  0.08721293  0.01661954  0.25116551  0.03983696 1.00000000
## CLIMATE    0.4271400 -0.12804799  0.23932435 -0.24608641  0.22124426 0.31358756
## SUPPORT    0.2424249  0.02112233  0.32526239 -0.30817162  0.01128116 0.17733753
##              CLIMATE     SUPPORT
## COMMIT     0.4271400  0.24242491
## AGE       -0.1280480  0.02112233
## SALARY     0.2393244  0.32526239
## CLASSSIZE -0.2460864 -0.30817162
## RESOURCES  0.2212443  0.01128116
## AUTONOMY   0.3135876  0.17733753
## CLIMATE    1.0000000  0.28470265
## SUPPORT    0.2847027  1.00000000
```

As can be seen from the above correlation matrix, there are no variables that are strongly correlated with the commitment score of a teacher, however SALARY, CLASSSIZE, and CLIMATE all have correlation coefficients that are higher than 0.35, which is better than any of the other variables in the data set. Thus we will also add those to our list of predictor variables. In the following code chunk we will create our new data set with only our variables of interest.

```
# Select our variables of interest:
Handout_1_new <- Handout_1 %>%
  select(COMMIT, SEX, SCHTYPE, SCHLEVEL, SALARY, CLASSSIZE, CLIMATE)
```

We will now test the normality of our dependent variable to see if we need to make a transformation or simply proceed with the regression. We will run all the tests for this problem at the 5% level of significance.

```
# Run a Shapiro test to assess the normality of the dependent variable:
shapiro.test(Handout_1_new$COMMIT)
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  Handout_1_new$COMMIT
## W = 0.98438, p-value = 0.08778
```

As seen from the above Shapiro-Wilk normality test, the p-value is 0.08778, thus we fail to reject the null hypothesis at the 5% level of significance. Hence we assume that the dependent variable is approximately normally distributed and we can continue on with our regression normally.

Now we will split our new data set into a training and testing set in order to evaluate our model accuracy.

```
# Set seed for reproducibility:
set.seed(1)

# Split the data set into 80% training and 20% testing:
index <- createDataPartition(Handout_1_new$SEX, p=.8, list=FALSE)
```

Below we will officially split our data set into the training and testing set. Furthermore, we will check the number of observations in each data set to see if splitting worked correctly.

```
# Create the training set:
train_data  <- Handout_1_new[index, ]

# Create the testing set:
test_data <- Handout_1_new[-index, ]

# Test and see if the split worked correctly:
nrow(train_data)
```

```
## [1] 120
```

```
nrow(test_data)
```

```
## [1] 30
```

As we can see from the above output, we had 120 observations in our training set and 30 observations in our testing set which is exactly what we would expect from an 80/20 split of 150 observations.

We will now construct our model using our training data that we created above.

```
# Create the model on the training set:
model_1 <- lm(COMMIT ~ factor(SEX) + factor(SCHTYPE) + factor(SCHLEVEL) + SALARY
              + CLASSSIZE + CLIMATE, data = train_data)

# Get a model summary:
summary(model_1)
```

```
##
## Call:
## lm(formula = COMMIT ~ factor(SEX) + factor(SCHTYPE) + factor(SCHLEVEL) +
##     SALARY + CLASSSIZE + CLIMATE, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.2519  -9.8949   0.8926   8.7568  27.6590
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        18.7636    25.6721   0.731   0.4664
## factor(SEX)2        1.8785     2.9539   0.636   0.5261
## factor(SCHTYPE)2    5.2398     3.2549   1.610   0.1103
## factor(SCHTYPE)3    6.8303     4.7615   1.434   0.1543
## factor(SCHLEVEL)2  -5.4967     2.9703  -1.851   0.0669 .
## factor(SCHLEVEL)3  -6.5230     3.2574  -2.003   0.0477 *
## SALARY              0.3837     0.5563   0.690   0.4918
## CLASSSIZE          -0.4742     0.3316  -1.430   0.1555
## CLIMATE             2.0679     0.4920   4.203 5.35e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.71 on 111 degrees of freedom
## Multiple R-squared:  0.346,  Adjusted R-squared:  0.2989
## F-statistic: 7.342 on 8 and 111 DF,  p-value: 8.425e-08
```

As can be seen from the above model output ignoring the significance of variables and model assumptions since that isn't the aim of this problem the regression equation is $\widehat{\text{COMMIT}} = 18.8 + 1.9 \cdot \text{Female} + 5.2 \cdot \text{Private} + 6.8 \cdot \text{Charter} - 5.5 \cdot \text{Middle} - 6.5 \cdot \text{High} + 0.4 \cdot \text{SALARY} - 0.5 \cdot \text{CLASSSIZE} + 2.1 \cdot \text{CLIMATE}$

Where Female refers to SEX, Private and Charter refer to SCHTYPE, and Middle and High refer to SCHLEVEL.

Lastly we will analyze the estimated prediction error and prediction error rate of the model.

```r
# Create the prediction vector:
predictions_1 <- model_1 %>% predict(test_data)

# Print a data frame of estimated prediction error:
error_df <- data.frame(R2 = R2(predictions_1, test_data$COMMIT),
                       RMSE = RMSE(predictions_1, test_data$COMMIT),
                       MAE = MAE(predictions_1, test_data$COMMIT))

# Show the estimated prediction error:
error_df
```

```
##          R2     RMSE      MAE
## 1 0.2900786 12.03883 10.24386
```

As can be seen the root mean squared error (RMSE) is 12.03883, the mean absolute error (MAE) is 10.24386 and the R squared value is 0.29. Based on the size of the sample these aren't the greatest error values, but they don't have too much meaning unless compared with another model. Below we will compute the estimated prediction error rate which is calculated as the RMSE of the data divided by the mean of the dependent variable.

```r
# Calculate the estimated prediction error rate:
error_df$RMSE / mean(test_data$COMMIT)
```

```
## [1] 0.2527396
```

Since the error rate is 0.2527396, this means that the model will make an error 25.27% of the time. This isn't too high given that our model wasn't rigorously tested, but for most model applications we would want an error rate that is much lower than this in order to be assured that our model is working accurately.

b) Simulation is often used as a tool to teach statistics. There are many theoretical results that are easier to explain to students through a simulation. Choose a statistical concept, theoretical result, or problem from a homework or exam from a previous statistics class. **Design a simulation that would teach this topic or answer this question.**

Some example topics include: simple random samples vs convenience samples, the probability of a certain event, the sampling distribution of the sample mean, confidence intervals, the central limit theorem, the difference between median and mode for data from a symmetric distribution vs a skewed distribution, etc. However, feel free to choose your own statistical topic! **Please do not use Normal Distribution.**

You should write this section as if you are teaching another student. Start with identifying the statistical topic and giving a brief overview. Then, run your simulation. Show your results in a table or plot. Explain your results to the student. Conclude with what you want them to take away from this simulation.

Answer:
In this problem we will teach you the idea of a large sample confidence interval for the mean. Before we dive into the the simulation let's start with a little review on the statistical topics needed to fully understand the idea of a large sample confidence interval for the mean. First off, when estimating μ_0, the mean of a population, the natural estimator for this is the sample mean $\bar{X}$. The reason why this is the most common estimator of μ_0 takes a little bit of work to get to. If we suppose $X_1, X_2, \ldots X_n$ are independent realizations from some probability distribution with mean μ_0 and variance σ_0^2, then $E[\bar{X}] = \mu_0$ and $Var[\bar{X}] = \frac{\sigma_0^2}{n}$. In other words, $\bar{X}$ is both unbiased and consistent. Being unbiased means that on average, the estimator equals the true value of the parameter, and being consistent means that as the sample size increases the variance of the estimator goes to 0. This means that the sampling distribution of the estimator becomes more centralized around the true value of the parameter as the sample size increases. Below we will show why these results hold. We will start off with the expected value calculation:

$$\begin{aligned}
E[\bar{X}] &= E\left[\frac{1}{n}(X_1 + X_2 + \cdots + X_n\right] \quad \text{(Definition of } \bar{X}\text{)}\\
&= E\left[\frac{1}{n}X_1 + \frac{1}{n}X_2 + \cdots + \frac{1}{n}X_n\right]\\
&= \frac{1}{n}E[X_1] + \frac{1}{n}E[X_2] + \cdots + \frac{1}{n}E[X_n] \quad \text{(Linearity of Expectation)}\\
&= \frac{1}{n}\mu_0 + \frac{1}{n}\mu_0 + \cdots + \frac{1}{n}\mu_0 \quad \text{(Since } E[X_i] = \mu_0\text{)}\\
&= \mu_0
\end{aligned}$$

We will now show the variance calculation:

$$\begin{aligned}
Var[\bar{X}] &= Var\left[\frac{1}{n}(X_1 + X_2 + \cdots + X_n\right] \quad \text{(Definition of } \bar{X}\text{)}\\
&= Var\left[\frac{1}{n}X_1 + \frac{1}{n}X_2 + \cdots + \frac{1}{n}X_n\right]\\
&= \frac{1}{n^2}Var[X_1] + \frac{1}{n^2}Var[X_2] + \cdots + \frac{1}{n^2}Var[X_n] \quad \text{(Independence)}\\
&= \frac{1}{n^2}\sigma_0^2 + \frac{1}{n^2}\sigma_0^2 + \cdots + \frac{1}{n^2}\sigma_0^2 \quad \text{(Since } Var[X_i] = \sigma_0^2\text{)}\\
&= \frac{n}{n^2}\sigma_0^2\\
&= \frac{\sigma_0^2}{n}
\end{aligned}$$

The next piece of statistical machinery that you must recall in order to understand the idea of a confidence interval is the central limit theorem. The central limit theorem states that linear combinations/sums of a large number of independent random variables follow a normal distribution regardless of the original probability distribution from which the random variables are drawn from. Hence for a large number n of independent random variables from a probability distribution with mean μ_0 and standard deviation σ_0, which is usually taken to be sample sizes n of size 30 our larger, $S = X_1 + X_2 + X_3 + \cdots + X_n \approx Norm(n\mu_0, \sigma_0\sqrt{n})$.

Now that we have these facts in our toolbox we can move on to the main topic of interest, the large sample confidence interval for μ_0. In general, a confidence interval is a range of values within which a parameter is predicted to fall into with a certain probability. This probability is referred to as the confidence level of the interval and is usually chosen to be close to 1. For example, common confidence levels include 0.99, 0.95, and 0.9.

We often find the lower and upper end points of this interval using the sampling distribution of a statistic that is a good estimator of the parameter. In the case of μ_0, which is the parameter of interest for this lesson, we can make use of the central limit theorem as described above. In particular, we will make use of the fact that $\bar{X} = \frac{S}{n} \approx Norm\left(\mu_0, \frac{\sigma_0}{\sqrt{n}}\right)$. This fact comes from using the result on normal random variables which states if $Y \sim Norm(\mu, \sigma)$, then $aY \sim Norm(a\mu, |a|\sigma)$. In our case, since $S \sim Norm(n\mu_0, \sigma_0\sqrt{n})$, then $\bar{X} = \frac{1}{n}S \sim Norm\left(\mu_0, \frac{\sigma_0}{\sqrt{n}}\right)$. For large n, we can standardize $\bar{X}$ and see that $\frac{\sqrt{n}(\bar{X}-\mu_0)}{\sigma_0} \approx Norm(0,1)$. Assuming n is large is the biggest assumption in the construction of a confidence interval for μ_0.

Recalling that for the standard normal PDF, the area within ±2.575829 is 99%, we have

$$P\left(-2.575829 < \frac{\sqrt{n}(\bar{X}-\mu_0)}{\sigma_0} < 2.575829\right) = 0.99$$

we can invert each side of the inequality inside the probability statement as follows:

$$-2.575829 < \frac{\sqrt{n}(\bar{X}-\mu_0)}{\sigma_0} \implies -2.575829\frac{\sigma_0}{\sqrt{n}} < (\bar{X}-\mu_0) \implies \mu_0 < \bar{X} + 2.575829\frac{\sigma_0}{\sqrt{n}}$$

Similarly, doing the same to the other side of the probability statement we see that:

$$2.575829 > \frac{\sqrt{n}(\bar{X}-\mu_0)}{\sigma_0} \implies 2.575829\frac{\sigma_0}{\sqrt{n}} > (\bar{X}-\mu_0) \implies \mu_0 > \bar{X} - 2.575829\frac{\sigma_0}{\sqrt{n}}$$

Therefore the original probability statement is identical to stating:

$$P\left(\bar{X} - 2.575829\frac{\sigma_0}{\sqrt{n}} < \mu_0 < \bar{X} + 2.575829\frac{\sigma_0}{\sqrt{n}}\right) = 0.99$$

Hence, the interval

$$\left[\bar{X} - 2.575829\frac{\sigma_0}{\sqrt{n}}, \bar{X} + 2.575829\frac{\sigma_0}{\sqrt{n}}\right]$$

has a 99% chance of containing the true mean μ_0 as an interior point. The preceding interval is thus our 99% confidence interval for μ_0. However it is important to note that this interval assumes we know the true value of σ_0 which isn't usually going to be the case, in this case we must use an estimate for the standard deviation, such as the sample standard deviation: $s = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n-1}}$. In the case of estimating our standard deviation, this in turn means that our confidence level isn't actually 99% instead it will depend on how accurate our estimate is, and in the case of a small sample size n, the statistic $\frac{\sqrt{n}(\bar{X}-\mu_0)}{S}$ no longer follows a normal distribution, instead it follows a t distribution.

Furthermore, it is important to understand the the preceding interval is a **random interval** meaning no observed data is taken into account. Thus, when we calculate a 99% confidence interval for a given sample, it is tempting to think that the probability that the calculated interval contains the true value of μ_0 is 0.99. However, this is not correct because the interval either contains μ_0 or it does not. Instead a confidence interval calculated from a sample is known as an **interval estimate**, we need to think of this interval as a random interval which will vary from sample to sample, but over many hypothetical replications, 99 of the intervals constructed in this fashion will cover the true parameter value.

The interpretation of confidence interval estimates is one of the most confusing topics for students to grasp. In order for you to better understand this, we will run a simulation to show that this interpretation holds true for randomly simulated data, and thus it holds in general. In this simulation we will draw 100 samples from an exponential distribution with rate parameter 2, once this sample is created we will repeatedly draw

100 re-samples of size 30 from this data set and calculate the mean and standard deviation of each re-sample. We will take re-samples of size 30 because that is the standard rule of thumb used in order to assume the central limit theorem applies. The steps to do this are shown below using the `lapply` function in R.

```
# Set random number seed for reproducibility:
set.seed(123)

# Number of distinct samples:
nsamp <- 100

# Size of each sample:
sampsize <- 30

# Mock data set:
data <- rexp(100, rate=2)

# Use lapply to repeatedly draw a sample from the data. From each sample
# calculate the mean and the standard deviation of the data and return a list
# where each element is a data frame with the sample mean and sample sd for each
# sample:
x <- lapply(X=1:nsamp , FUN=function(X){
            new_sample <- sample(data, size = sampsize, replace=F)
            xbar <- mean(new_sample)
            s2 <- sd(new_sample)
            data.frame(sample_mean = xbar, sample_sd = s2)})

# Bind the rows of the individual data frames contained in the list to create
# one data frame:
resamples <- do.call(rbind,x)
```

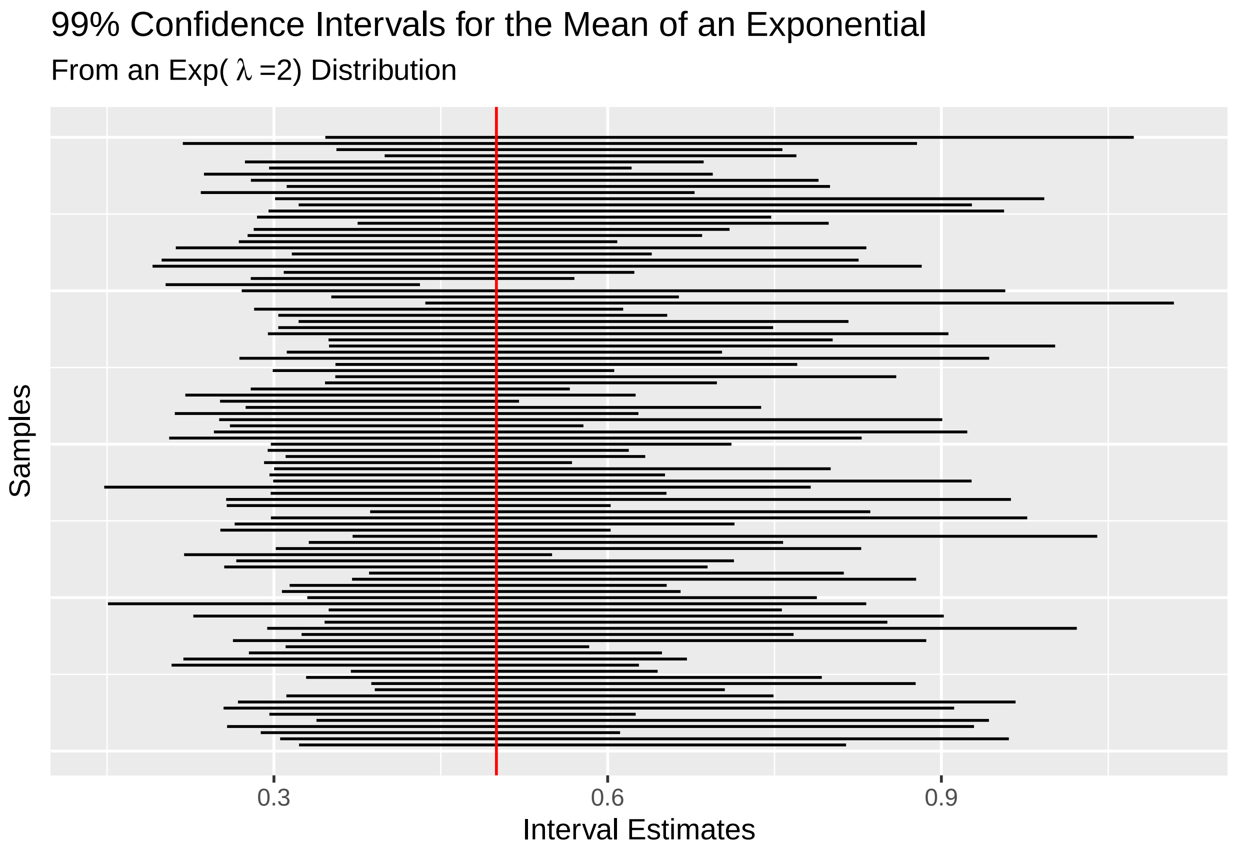
For each of these 100 simulated re-samples we will create a 99% confidence interval estimate as shown below.

```
# Find the 99.5 percentile of the standard normal:
z_crit <- qnorm(p = 0.995)

# Create 99% confidence intervals for each re-sample:
resamples$lower <- resamples$sample_mean - z_crit * resamples$sample_sd / sqrt(sampsize)
resamples$upper <- resamples$sample_mean + z_crit * resamples$sample_sd / sqrt(sampsize)
```

Recall from previous statistics classes that the mean of an exponential distribution is $\frac{1}{\lambda}$, hence in this case the mean is $\frac{1}{2}$. We will now make a plot of all of these confidence interval estimates and plot a red vertical line representing this true mean value. The code for this is shown below using ggplot.

```
# Use ggplot to plot these intervals as well as a red vertical line at the true
# mean value of 0.5:
ggplot(data=resamples) +
  geom_segment(mapping = aes(x = lower,
                             xend = upper,
                             y = 1:nsamp,
                             yend = 1:nsamp)) +
  geom_vline(xintercept = 0.5, color = "red") +
  labs(x = "Interval Estimates",
       y = "Samples",
       title = "99% Confidence Intervals for the Mean of an Exponential",
       subtitle = expression("From an Exp(" ~ lambda ~ "=2) Distribution")) +
  theme(axis.text.y=element_blank(),
        axis.ticks.y=element_blank())
```



Now that we have visualized these intervals we will calculate the number of intervals that contain this true value of the mean.

```
# Calculate and report the number of intervals that contain the true value of
# the mean:
num_intervals <- length(which(resamples$lower <= 0.5 & 0.5 <= resamples$upper))
num_intervals
```

```
## [1] 99
```

As can be seen, over hypothetical replications of taking re-samples from an exponential distribution with rate parameter 2, 99% of the intervals contained the true value of the mean, as we expected from the theoretical explanation above.

The main thing that we want you guys to take away from this simulation is that given a confidence interval calculated from a sample, the interpretation of the interval is not that the probability that the calculated interval contains the true value of μ_0 is 0.99. Instead we need to think of this interval as a random interval which will vary from one random sample to the next, but over many hypothetical replications, 99 of the intervals constructed in this fashion will cover the true parameter value. This same theory holds true for other confidence levels such as 0.95, 0.9, etc. However, it is important to note that if you were to do this simulation yourself that depending on the random seed you set you won't get *exactly* 99% of the intervals containing the true value. For example we will repeat this simulation but instead with 1000 re-samples to prove this point.

```r
# Set random number seed:
set.seed(123)

# Number of distinct samples:
nsamp_2 <- 1000

# Size of each sample:
sampsize_2 <- 30

# Mock data set:
data_2 <- rexp(100, rate=2)

# Use lapply to repeatedly draw a sample from the data. From each sample
# calculate the mean and the standard deviation of the data and return a list
# where each element is a data frame with the sample mean and sample sd for each
# sample:
x_2 <- lapply(X=1:nsamp_2 , FUN=function(X){
            new_sample <- sample(data_2, size = sampsize_2, replace=F)
            xbar <- mean(new_sample)
            s2 <- sd(new_sample)
            data.frame(sample_mean = xbar, sample_sd = s2)})

# Bind the rows of the individual data frames contained in the list to create
# one data frame:
resamples_2 <- do.call(rbind, x_2)

# Create 99% confidence intervals for each re-sample:
resamples_2$lower <- resamples_2$sample_mean - z_crit * resamples_2$sample_sd /
                       sqrt(sampsize_2)
resamples_2$upper <- resamples_2$sample_mean + z_crit * resamples_2$sample_sd /
                       sqrt(sampsize_2)
```

Again, we will see how many of these simulated intervals contain the true value of μ_0.

```r
# Calculate and report the number of intervals that contain the true value of
# the mean:
num_intervals <- length(which(resamples_2$lower <= 0.5 & 0.5 <= resamples_2$upper))
num_intervals
```

```
## [1] 985
```

As can be seen above 985 of these 1000 intervals contain the true value of μ_0 with a percentage of 98.5%. The reason for this difference is due to the random chance of simulation, it is apparent that this value is very close to 99%, and as we do more and more simulations, this value will get closer and closer to 99%.

c) Write a function to compute the mean, median, mode, variance, and skewness of Rayleigh distribution for (i) $\sigma = 6$; and (ii) $\sigma^2 = 10$). In addition, find the maximum likelihood estimator of σ and σ^2 from the Rayleigh distribution. A Rayleigh distribution with the correct formulas can be found here: https://en.wikipedia.org/wiki/Rayleigh_distribution.

Answer:
The first part of this problem requires us to write a function to compute the mean, median, mode, variance, and skewness of the Rayleigh distribution. After this function is created, we will test the function for (i) $\sigma = 6$; and (ii) $\sigma^2 = 10$.

As can be seen from the above Wikipedia page, the PDF of a Rayleigh distribution is $f(x) = \frac{x}{\sigma^2} e^{-x^2/(2\sigma^2)}, x \geq 0$. Thus we can see that this distribution is a one parameter distribution (at least for this specific parametrization). Furthermore, the mean of a Rayleigh distribution is $\sigma\sqrt{\frac{\pi}{2}}$, the median is $\sigma\sqrt{\pi \ln(2)}$, the mode is σ, the variance is $\frac{4-\pi}{2}\sigma^2$, and lastly the skewness is $\frac{2\sqrt{\pi}(\pi-3)}{(4-\pi)^{3/2}}$. Below we will write a function computing all of these formulas for any given Rayleigh distribution with parameter σ.

```r
# Defining the function to compute the statistics for a Rayleigh distribution:
rayleigh_stats <- function(sigma) {
  # Compute the mean:
  mean <- sigma * sqrt(pi/2)

  # Compute the median:
  median <- sigma * sqrt(2*log(2))

  # Compute the mode:
  mode <- sigma

  # Compute the variance:
  variance <- ((4-pi)/2)*(sigma)^2

  # Compute the skewness:
  skewness <- (2*sqrt(pi)*(pi-3))/(4-pi)^(3/2)

  # Create a list containing the above statistics:
  return_list <- list(mean, median, mode, variance, skewness)

  # Add descriptive names to each list element:
  names(return_list) <- c("Mean", "Median", "Mode", "Variance", "skewness")

  # Return the list:
  return(return_list)
}
```

Now that we have defined the function, we will first test it using a Rayleigh distribution with the parameter $\sigma = 6$.

```r
# Test case 1, sigma = 6:
rayleigh_stats(6)
```

```
## $Mean
## [1] 7.519885
##
## $Median
## [1] 7.06446
##
## $Mode
## [1] 6
##
```

```
## $Variance
## [1] 15.45133
##
## $skewness
## [1] 0.6311107
```

As can be seen above, for the Rayleigh distribution with $\sigma = 6$ as its parameter; the mean equals 7.519885, the median equals 7.06446, the mode equals 6, the variance equals 15.45133, and the skewness equals 0.6311107. These outputted values match what is expected for a Rayleigh distribution with $\sigma = 6$ as its parameter.

We will also test the above function using a Rayleigh distribution with the parameter $\sigma^2 = 10 \implies \sigma = \sqrt{10}$.

```r
# Test case 2, sigma = sqrt(10):
rayleigh_stats(sqrt(10))
```

```
## $Mean
## [1] 3.963327
##
## $Median
## [1] 3.723297
##
## $Mode
## [1] 3.162278
##
## $Variance
## [1] 4.292037
##
## $skewness
## [1] 0.6311107
```

As can be seen above, for the Rayleigh distribution with $\sigma^2 = 10$ as its parameter; the mean equals 3.963327, the median equals 3.723297, the mode equals 3.162278, the variance equals 4.292037, and the skewness equals 0.6311107. These outputted values match what is expected for a Rayleigh distribution with $\sigma^2 = 10$ as its parameter.

The next part of this problem requires us to find the maximum likelihood estimator (MLE) of σ and σ^2 from the Rayleigh distribution. To do this we will start by finding the MLE of σ^2, then use the invariance property of the MLE to find σ. The invariance property states that if $\widehat{\theta}_0^{mle}$ is the MLE of θ_0, then for any function g, the MLE of $g(\theta_0)$ is $g(\widehat{\theta}_0^{mle})$.

If we suppose that $X_1, X_2, \dots X_n \overset{i.i.d.}{\sim} Rayleigh(\sigma)$, then the PDF of the X_i's can be written as $f(x_i) = \frac{x_i}{\sigma^2} e^{-x_i^2/(2\sigma^2)}$, $x_i \geq 0$. Where X_i is the i^{th} random variable out of the sample of n independent values. With that being said, to find the maximum likelihood estimator of σ, we must find the likelihood function, take the natural log of the likelihood function, find the critical point of the log-likelihood function, and lastly run the second derivative test to show the critical point is a maximum. First off, we will compute the likelihood function as shown below.

$$\begin{aligned}
L(\sigma) &= f(x_1) \times f(x_2) \times \cdots \times f(x_n) \\
&= \frac{x_1}{\sigma^2} e^{-x_1^2/(2\sigma^2)} \times \frac{x_2}{\sigma^2} e^{-x_2^2/(2\sigma^2)} \times \cdots \times \frac{x_n}{\sigma^2} e^{-x_n^2/(2\sigma^2)} \\
&= \frac{1}{\sigma^{2n}} e^{\frac{-1}{2\sigma^2} \sum_{i=1}^{n} x_i^2} \prod_{i=1}^{n} x_i
\end{aligned}$$

Thus, as computed above, the likelihood function of σ is $L(\sigma) = \frac{1}{\sigma^{2n}} e^{\frac{-1}{\sigma^2}\sum_{i=1}^n x_i^2} \prod\limits_{i=1}^n x_i$. However, notice that taking the derivative of this function will not be easy, thus we will take the natural log of the likelihood function to turn multiplication into addition. This process is shown below.

$$\begin{aligned}
\ell(\sigma) &= \ln(L(\sigma)) \\
&= \ln\left(\frac{1}{\sigma^{2n}} e^{\frac{-1}{2\sigma^2}\sum_{i=1}^n x_i^2} \prod_{i=1}^n x_i\right) \\
&= \ln(1) - \ln(\sigma^{2n}) + \ln\left(\prod_{i=1}^n x_i\right) + \ln\left(e^{\frac{-1}{2\sigma^2}\sum_{i=1}^n x_i^2}\right) \\
&= -2n\ln(\sigma) + \sum_{i=1}^n \ln(x_i) - \frac{1}{2\sigma^2}\sum_{i=1}^n x_i^2
\end{aligned}$$

As computed above, the log-likelihood function is $\ell(\sigma) = -2n\ln(\sigma) + \sum\limits_{i=1}^n \ln(x_i) - \frac{1}{\sigma^2}\sum_{i=1}^n x_i^2$. Now we must find the critical points of this function in order to find the candidates for the maximum likelihood estimator of σ. This derivative calculation is shown below.

$$\begin{aligned}
\frac{d}{d\sigma}\ell(\sigma) &= \frac{d}{d\sigma}\left(-2n\ln(\sigma) + \sum_{i=1}^n \ln(x_i) - \frac{1}{2\sigma^2}\sum_{i=1}^n x_i^2\right) \\
&= \frac{-2n}{\sigma} + \frac{\sum_{i=1}^n x_i^2}{\sigma^3}
\end{aligned}$$

In order to find the critical points, we must find the values of σ such that this derivative is equal to zero.

$$\begin{aligned}
0 &= \frac{-2n}{\sigma} + \frac{\sum_{i=1}^n x_i^2}{\sigma^3} \\
\frac{2n}{\sigma} &= \frac{\sum_{i=1}^n x_i^2}{\sigma^3} \\
\sigma^2 &= \frac{\sum_{i=1}^n x_i^2}{2n}
\end{aligned}$$

As computed above, the critical point is $\sigma^2 = \frac{\sum_{i=1}^n x_i^2}{2n}$, hence pending the second derivative test, the MLE of σ^2 in the Rayleigh distribution is $\widehat{\sigma^2}^{mle} = \frac{\sum_{i=1}^n x_i^2}{2n}$. Furthermore, using the invariance property of the MLE explained above we can easily see that the MLE of σ in the Rayleigh distribution is $\widehat{\sigma}^{mle} = \sqrt{\frac{\sum_{i=1}^n x_i^2}{2n}}$. These MLE's match the ones found in the above Wikipedia page, however to ensure that the above MLEs are in fact maximum we will run the second derivative test for $\widehat{\sigma^2}^{mle}$. Passing this test for $\widehat{\sigma^2}^{mle}$ will be enough to prove that both MLEs are maximum since one is a transformation of the other, and the square-root function is a monotonic increasing function and thus preserves order.

The second derivative of the log likelihood is

$$\begin{aligned}
\frac{d^2}{d\sigma^2}\ell(\sigma) &= \frac{d^2}{d\sigma^2}\left(\frac{-2n}{\sigma} + \frac{\sum_{i=1}^n x_i^2}{\sigma^3}\right) \\
&= \frac{4n}{\sigma^2} - \frac{3\sum_{i=1}^n x_i^2}{\sigma^4}
\end{aligned}$$

Plugging in our value of $\widehat{\sigma^2}^{mle}$ into the second derivative of the log-likelihood function we obtain

$$\begin{aligned}
\ell''(\widehat{\sigma^2}^{mle}) &= \frac{4n}{\frac{\sum_{i=1}^n x_i^2}{2n}} - \frac{3\sum_{i=1}^n x_i^2}{\left(\frac{\sum_{i=1}^n x_i^2}{2n}\right)^2} \\
&= \frac{8n^2}{\sum_{i=1}^n x_i^2} - \frac{3\sum_{i=1}^n x_i^2}{\frac{(\sum_{i=1}^n x_i^2)^2}{4n^2}} \\
&= \frac{8n^2}{\sum_{i=1}^n x_i^2} - \frac{12n^2}{\sum_{i=1}^n x_i^2} \\
&= \frac{-4n^2}{\sum_{i=1}^n x_i^2}
\end{aligned}$$

Since $-4 < 0$, $n^2 > 0$, and $\sum_{i=1}^n x_i^2 > 0$, it follows that $\frac{-4n^2}{\sum_{i=1}^n x_i^2} < 0$ and hence $\widehat{\sigma^2}^{mle} = \frac{\sum_{i=1}^n x_i^2}{2n}$ and $\widehat{\sigma}^{mle} = \sqrt{\frac{\sum_{i=1}^n x_i^2}{2n}}$ are local maximums. However, since there was only one critical point it turns out that these values are in fact global maximums and thus are the MLEs of σ and σ^2 for the Rayleigh distribution.