STAT 4360 (Introduction to Statistical Learning, Spring 2023)
Mini Project 2
Name: Sayema Rahman

---

1. (a) There are many things we can conclude from the observations. The correlation matrix tells us that the strongest positive correlation is between variables Quality and Flavor at approximately 0.79. As the quality of wine increases, so does the flavor, and vice versa. The second strongest correlation we have is between variables Quality and Aroma at approximately 0.70. This means that as the quality of the wine increases, so does the aroma of it. There is also a strong positive correlation between Aroma and Flavor at approximately 0.74.

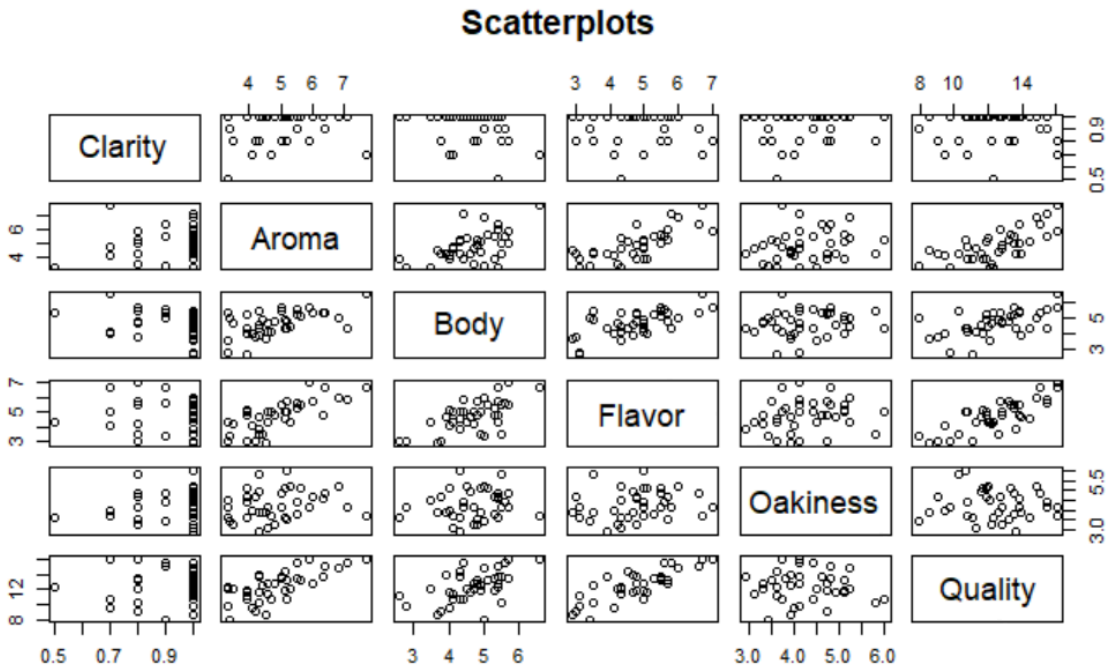


Figure 1 is a Scatterplot Matrix

	Quality	Clarity	Aroma	Body	Flavor	Oakiness
Quality	1.00000000	0.02844131	0.7073243	0.5487022	0.79004713	-0.04704047
Clarity	0.02844131	1.00000000	0.0619021	-0.3083783	-0.08515993	0.18321471
Aroma	0.70732432	0.06190210	1.0000000	0.5489102	0.73656121	0.20164445
Body	0.54870219	-0.30837826	0.5489102	1.0000000	0.64665917	0.15210591
Flavor	0.79004713	-0.08515993	0.7365612	0.6466592	1.00000000	0.17976051
Oakiness	-0.04704047	0.18321471	0.2016444	0.1521059	0.17976051	1.00000000

Figure 2 is a Correlation Matrix

Clarity	Aroma	Body	Flavor	Oakiness	Quality	Region
Min. :0.5000	Min. :3.300	Min. :2.600	Min. :2.900	Min. :2.900	Min. : 7.90	Min. :1.000
1st Qu.:0.8250	1st Qu.:4.125	1st Qu.:4.150	1st Qu.:4.225	1st Qu.:3.700	1st Qu.:11.15	1st Qu.:1.000
Median :1.0000	Median :4.650	Median :4.750	Median :4.800	Median :4.100	Median :12.45	Median :2.000
Mean :0.9237	Mean :4.847	Mean :4.684	Mean :4.768	Mean :4.255	Mean :12.44	Mean :1.868
3rd Qu.:1.0000	3rd Qu.:5.450	3rd Qu.:5.375	3rd Qu.:5.500	3rd Qu.:4.775	3rd Qu.:13.75	3rd Qu.:3.000
Max. :1.0000	Max. :7.700	Max. :6.600	Max. :7.000	Max. :6.000	Max. :16.10	Max. :3.000

Figure 3 is a summary of the data

(b) There is a significant relationship between Quality and Aroma, Quality and Body, Quality and Flavor, and Quality and Region. Our observations show that Aroma, Flavor, Body, and Region are all statistically significant variables. There is an association shown for each of these variables from the linear models created. On the linear graphs created, we can also come to the same conclusion that Aroma, Flavor, Body, and Region are all statistically significant variables.

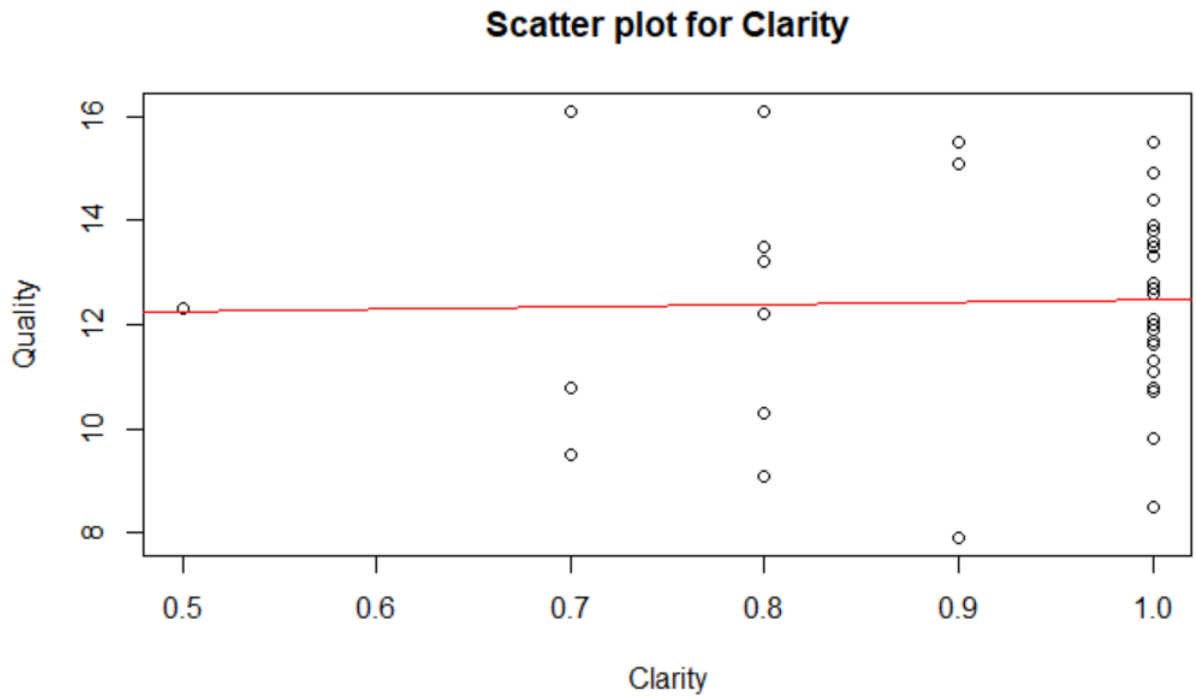


Figure 4: Scatter plot for Clarity versus Quality

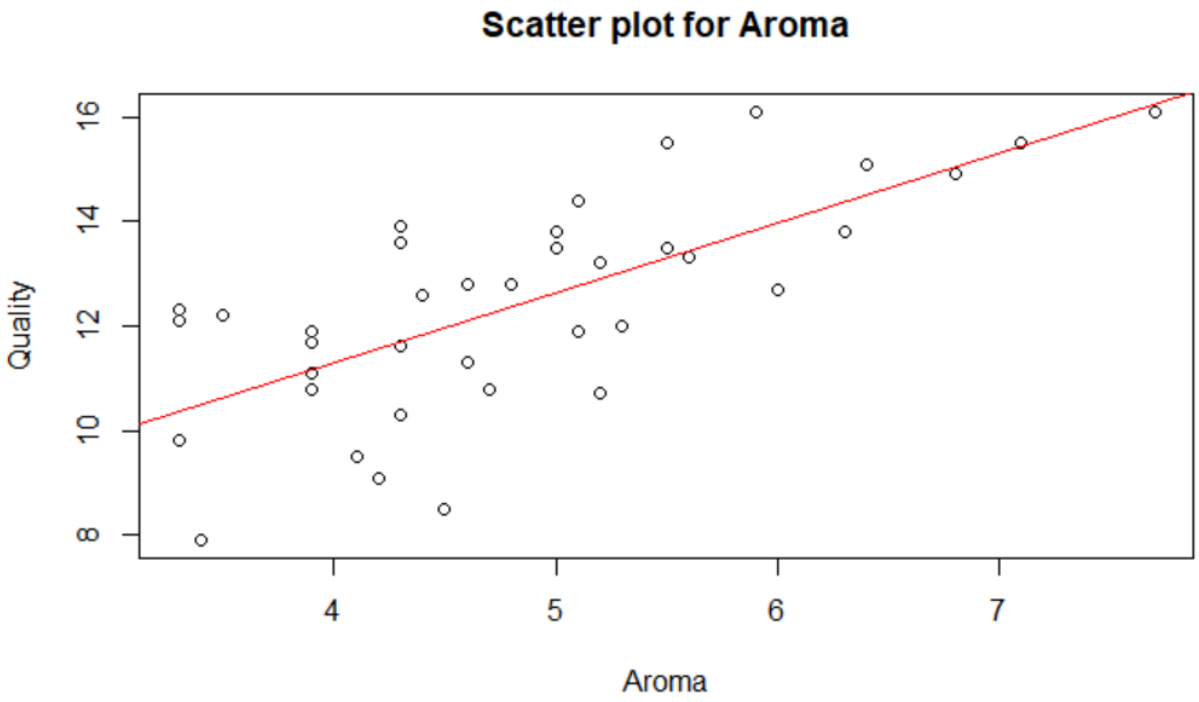


Figure 5: Scatter plot for Aroma versus Quality

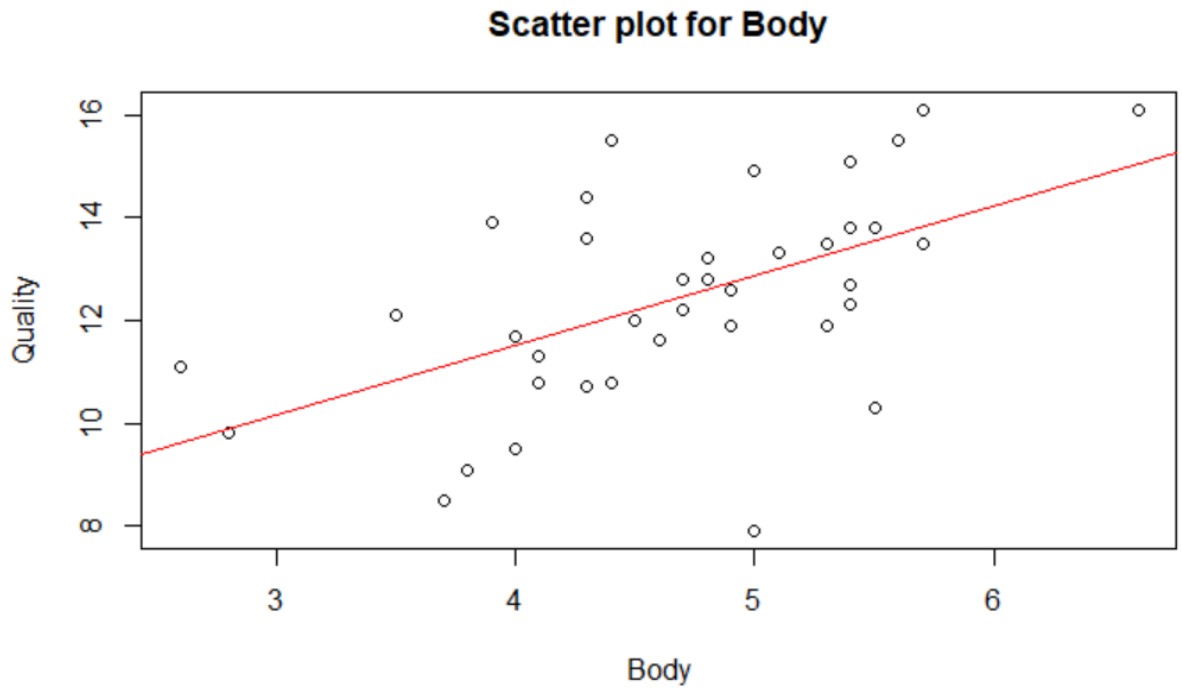


Figure 6: Scatter plot for Body versus Quality

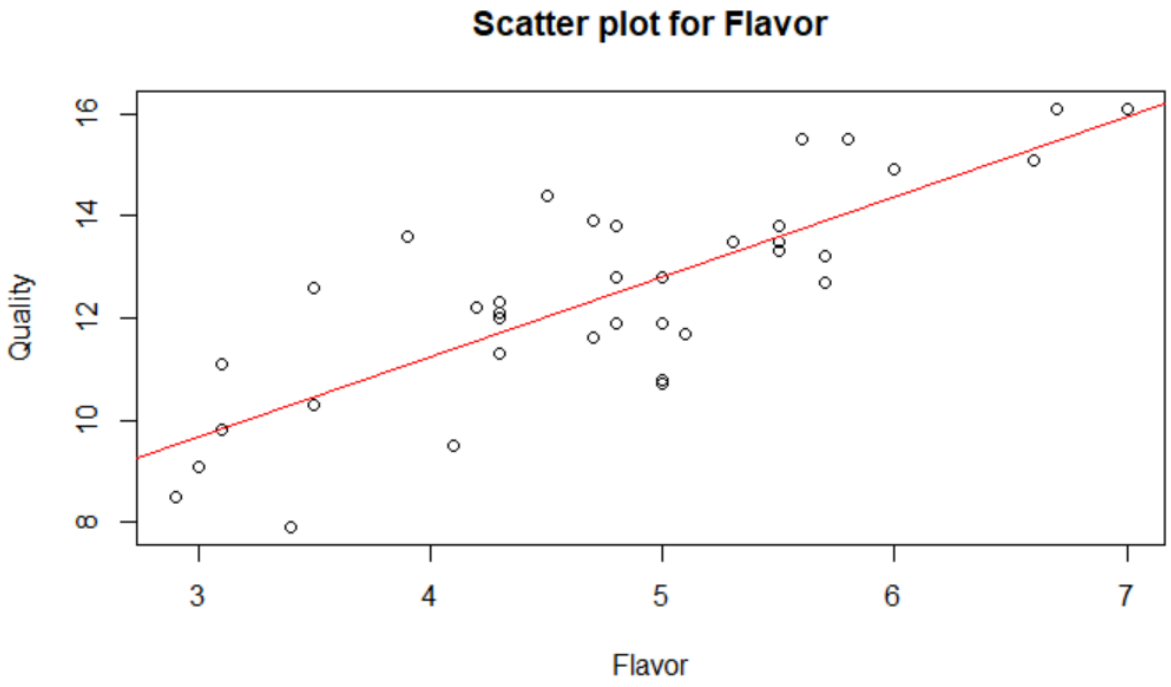


Figure 7: Scatter plot for Flavor versus Quality

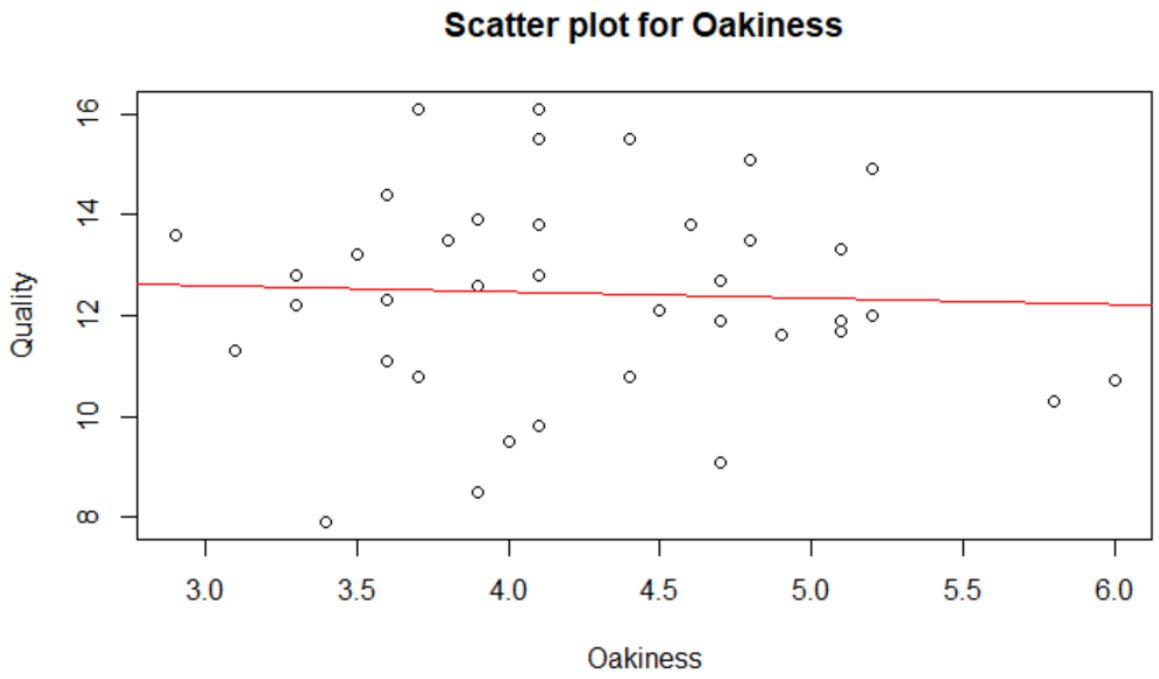


Figure 8: Scatter plot for Oakiness versus Quality

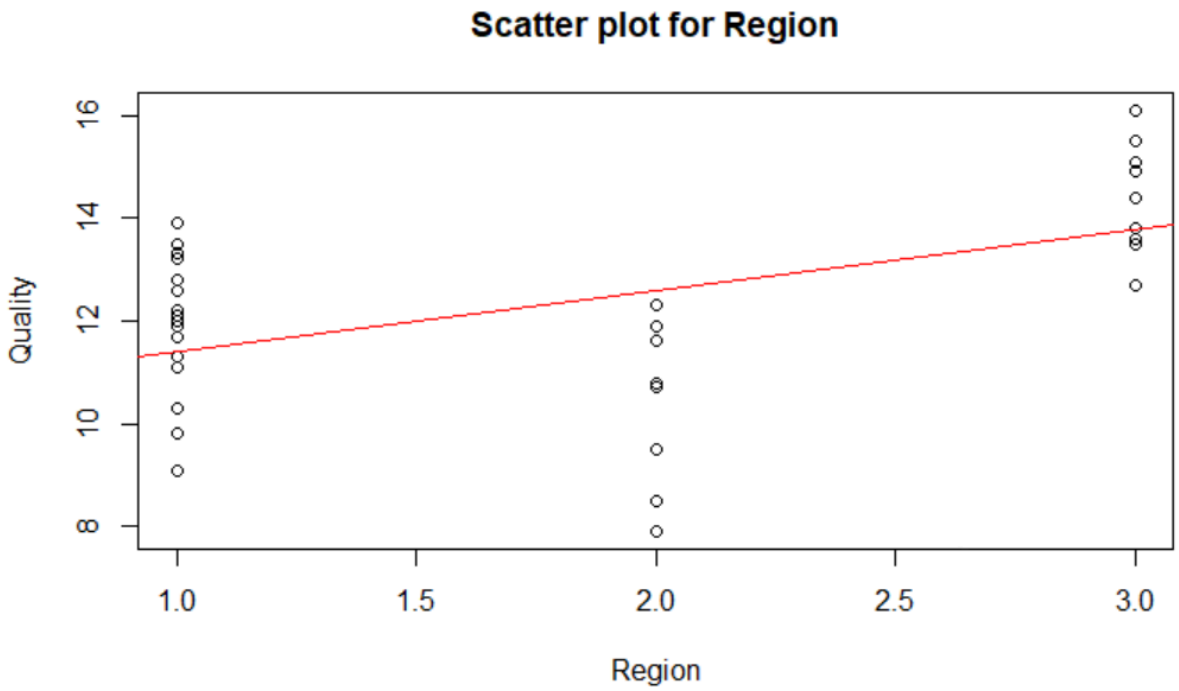


Figure 9: Scatter plot for Region versus Quality

```
Call:
lm(formula = wine$Quality ~ wine$Aroma)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4726 -0.8574 -0.0091  0.8346  2.2563

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.9583     1.1050   5.392 4.51e-06 ***
wine$Aroma    1.3365     0.2226   6.004 6.87e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.466 on 36 degrees of freedom
Multiple R-squared:  0.5003,    Adjusted R-squared:  0.4864
F-statistic: 36.04 on 1 and 36 DF,  p-value: 6.871e-07
```

Figure 10: Summary of Quality and Aroma

```
Call:
lm(formula = wine$Quality ~ wine$Body)

Residuals:
    Min      1Q  Median      3Q     Max
-4.9669 -0.8386  0.0620  1.2204  3.4502

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.0580     1.6441   3.685 0.000748 ***
wine$Body     1.3618     0.3458   3.938 0.000361 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.734 on 36 degrees of freedom
Multiple R-squared:  0.3011,    Adjusted R-squared:  0.2817
F-statistic: 15.51 on 1 and 36 DF,  p-value: 0.0003612
```

Figure 11: Summary of Quality and Body

```
Call:
lm(formula = wine$Quality ~ wine$Flavor)

Residuals:
     Min       1Q   Median       3Q      Max
-2.38583 -0.72226 -0.00756  0.62006  2.52822

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.9414     0.9911   4.986 1.57e-05 ***
wine$Flavor   1.5719     0.2033   7.732 3.68e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.271 on 36 degrees of freedom
Multiple R-squared:  0.6242,    Adjusted R-squared:  0.6137
F-statistic: 59.79 on 1 and 36 DF,  p-value: 3.683e-09
```

Figure 12: Summary of Quality and Flavor

```
Call:
lm(formula = wine$Quality ~ wine$Oakiness)

Residuals:
    Min      1Q  Median      3Q     Max
-4.6483 -1.3886 -0.0527  1.2907  3.6429

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   12.9916     1.9918   6.522  1.4e-07 ***
wine$Oakiness -0.1304     0.4614  -0.283    0.779
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.071 on 36 degrees of freedom
Multiple R-squared:  0.002213,  Adjusted R-squared:  -0.0255
F-statistic: 0.07984 on 1 and 36 DF,  p-value: 0.7791
```

Figure 13: Summary of Quality and Oakiness

```
Call:
lm(formula = wine$Quality ~ wine$Region)

Residuals:
    Min      1Q  Median      3Q     Max
-4.6928 -1.0565  0.1572  1.3747  2.4922

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  10.2228     0.6910   14.79  < 2e-16 ***
wine$Region   1.1850     0.3357    3.53  0.00116 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.787 on 36 degrees of freedom
Multiple R-squared:  0.2571,    Adjusted R-squared:  0.2365
F-statistic: 12.46 on 1 and 36 DF,  p-value: 0.001159
```

Figure 14: Summary of Quality and Region

(c) I can reject the null hypothesis for the two predictors, flavor and oakiness. When doing a multiple linear regression model the model shows that only the variables Flavor and Oakiness are significant as they are less than 0.05 in their P values.

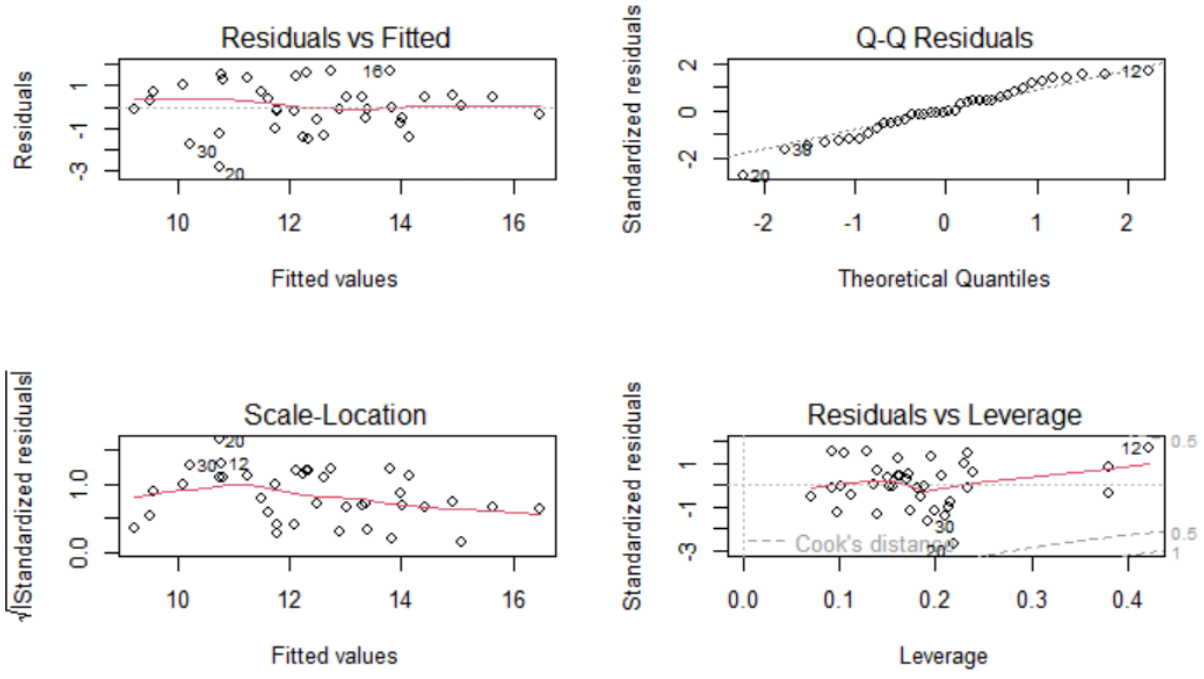
```
Call:
lm(formula = Quality ~ Clarity + Aroma + Body + Flavor + Oakiness +
    Region, data = wine)

Residuals:
     Min       1Q   Median       3Q      Max
-2.83614 -0.57561 -0.06547  0.66181  1.70485

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.98433    2.26965   1.755 0.089056 .
Clarity      2.34751    1.76362   1.331 0.192872
Aroma        0.49731    0.30536   1.629 0.113516
Body         0.27841    0.34091   0.817 0.420357
Flavor       1.16987    0.30958   3.779 0.000673 ***
Oakiness    -0.69229    0.28483  -2.431 0.021058 *
Region      -0.03381    0.29568  -0.114 0.909694
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.181 on 31 degrees of freedom
Multiple R-squared:  0.7207,    Adjusted R-squared:  0.6667
F-statistic: 13.33 on 6 and 31 DF,  p-value: 2.037e-07
```

Significant predictors in the multiple regression model: Flavor, Oakiness



(d) To build a “reasonable good” multiple regression model I used the statistically significant variables which were Aroma, Flavor, and Oakiness. According to the ANOVA table I made, this model is an accurate measurement of the data. All the values in the ANOVA table show that the model is a good fit for the data.

```
Call:
lm(formula = Quality ~ Aroma + Flavor + Oakiness)

Residuals:
    Min      1Q  Median      3Q     Max
-2.5707 -0.6256  0.1521  0.6467  1.7741

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.4672     1.3328   4.852 2.67e-05 ***
Aroma         0.5801     0.2622   2.213 0.033740 *
Flavor        1.1997     0.2749   4.364 0.000113 ***
Oakiness     -0.6023     0.2644  -2.278 0.029127 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.161 on 34 degrees of freedom
Multiple R-squared:  0.7038,    Adjusted R-squared:  0.6776
F-statistic: 26.92 on 3 and 34 DF,  p-value: 4.203e-09

                 2.5 %     97.5 %
(Intercept)  3.75864235  9.1757473
Aroma        0.04729651  1.1129440
Flavor       0.64106744  1.7583182
Oakiness    -1.13965261 -0.0649967
       fit      lwr     upr
1 9.748857 7.281012 12.2167
```

```
Analysis of Variance Table

Response: Quality
           Df Sum Sq Mean Sq F value    Pr(>F)
Aroma       1 77.442  77.442 57.4226 8.401e-09 ***
Flavor      1 24.494  24.494 18.1624  0.000152 ***
Oakiness    1  6.999   6.999  5.1896  0.029127 *
Residuals 34 45.853   1.349
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Call:
lm(formula = Quality ~ Aroma + Flavor + Oakiness)

Residuals:
    Min      1Q  Median      3Q     Max
-2.5707 -0.6256  0.1521  0.6467  1.7741

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.4672     1.3328   4.852 2.67e-05 ***
Aroma         0.5801     0.2622   2.213 0.033740 *
Flavor        1.1997     0.2749   4.364 0.000113 ***
Oakiness     -0.6023     0.2644  -2.278 0.029127 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.161 on 34 degrees of freedom
Multiple R-squared:  0.7038,    Adjusted R-squared:  0.6776
F-statistic: 26.92 on 3 and 34 DF,  p-value: 4.203e-09
```

(e) The final model, in equation form, is Quality = 6.4672 + 0.5801 * Aroma + 1.1997 * Flavor - 0.6023 * Oakiness.

(f) I first calculated the mean values of Aroma, Flavor, and Oakiness. Then put these values in the final model and predict the Quality. The mean value for Aroma is 4.358824, Flavor is 4.376471 and Oakiness is 4.276471. Then I used these values to get the predicted value quality which is 11.67049. The prediction interval values are 7.865333 to 15.475641. This accounts for the uncertainty and the variability of the predictions. The confidence interval values are 10.74760 and 12.59337. This provides good measures of the mean response estimate and a narrow range for the average quality of wine.

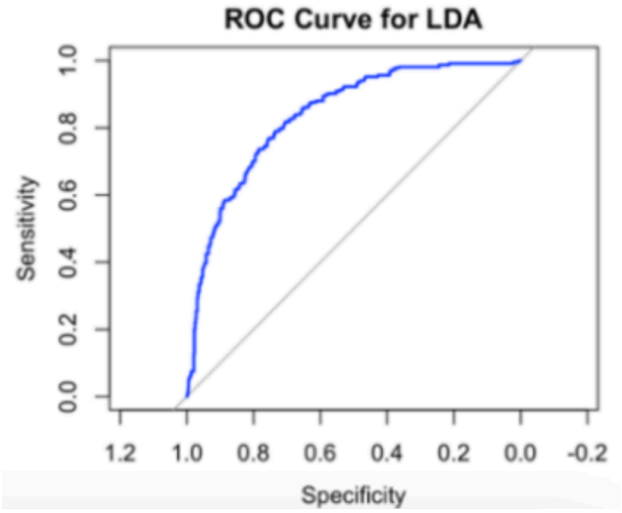
```
> # calculating mean values
> aroma_mean <- mean(wine$Aroma[wine$Region == 1], na.rm = TRUE)
> print(aroma_mean)
[1] 4.358824
> flavor_mean <- mean(wine$Flavor[wine$Region == 1], na.rm = TRUE)
> print(flavor_mean)
[1] 4.376471
> oakiness_mean <- mean(wine$Oakiness[wine$Region == 1], na.rm = TRUE)
> print(oakiness_mean)
[1] 4.276471
> # standard error
> residualStandardError = summary(reduced_model)$sigma
> # calculating mean values
> aroma_mean <- mean(wine$Aroma[wine$Region == 1], na.rm = TRUE)
> print(aroma_mean)
[1] 4.358824
> flavor_mean <- mean(wine$Flavor[wine$Region == 1], na.rm = TRUE)
> print(flavor_mean)
[1] 4.376471
> oakiness_mean <- mean(wine$Oakiness[wine$Region == 1], na.rm = TRUE)
> print(oakiness_mean)
[1] 4.276471
>
> intercept = 6.4672
> aroma_coeff = 0.5801
> flavor_coeff = 1.1997
> oakiness_coeff = -0.6023
>
> predicted_val = intercept + aroma_coeff*aroma_mean + flavor_coeff*flavor_mean + oakiness_coeff*oakiness_mean
> print(predicted_val)
[1] 11.67049
>
> # standard error
> residualStandardError = summary(reduced_model)$sigma
> SE_predict = residualStandardError * sqrt(1+sum(c((aroma_mean - mean(wine$Aroma))^2,
+                                                   (flavor_mean - mean(wine$Flavor))^2,
+                                                  (oakiness_mean -  mean(wine$Oakiness))^2)))
>
> # 95 percent prediction interval
> df_residual <- summary(reduced_model)$df[3]
> t_val = qt(0.975, df_residual)
> error_margin = t_val * SE_predict
> prediction_interval = c(predicted_val - error_margin, predicted_val + error_margin)
> print(prediction_interval)
[1]  7.865333 15.475641
>
> # 95 percent confidence interval
> mean_response = SE_predict / sqrt(length(wine$Quality[wine$Region == 1]))
> t_mean = qt(0.975, df_residual)
> error_margin_mean = t_mean * mean_response
> conf_interval_mean = c(predicted_val - error_margin_mean, predicted_val + error_margin_mean)
> print(conf_interval_mean)
[1] 10.74760 12.59337
```

2. (a) a strong positive correlation between pregnancies and age was 0.54. There is also a slight correlation between glucose and BMI at 0.227 and insulin and skin thickness at 0.448. Also, outcome and glucose have one at 0.458. Age and outcome also have one at 0.237. Age and skin thickness have a negative correlation at -0.1.

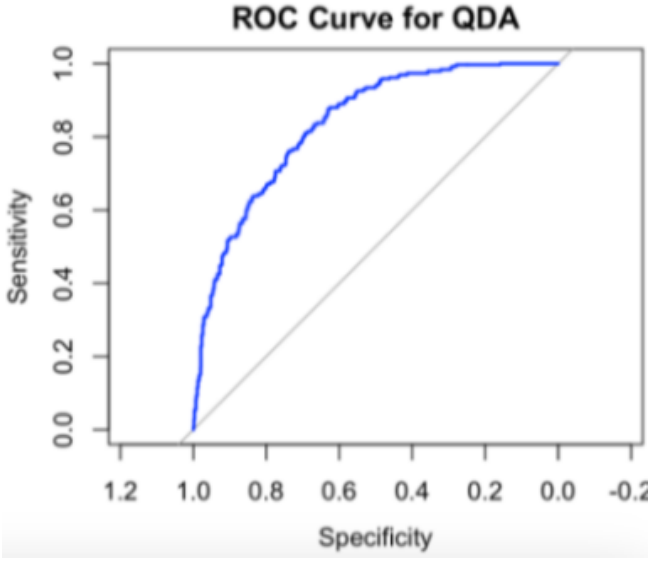
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.00000000	0.12040541	0.14967246	-0.06337462	-0.07659977	0.01947503	-0.02545316	0.53945719	0.22443699
Glucose	0.12040541	1.00000000	0.13804400	0.06236813	0.32037084	0.22686443	0.12324343	0.25449621	0.45842130
BloodPressure	0.14967246	0.13804400	1.00000000	0.19880047	0.08738405	0.28154513	0.05133095	0.23837508	0.07595808
SkinThickness	-0.06337462	0.06236813	0.19880047	1.00000000	0.44885895	0.39376029	0.17829888	-0.11103369	0.07604025
Insulin	-0.07659977	0.32037084	0.08738405	0.44885895	1.00000000	0.22301161	0.19271873	-0.08587910	0.12092362
BMI	0.01947503	0.22686443	0.28154513	0.39376029	0.22301161	1.00000000	0.12571935	0.03898737	0.27672554
DiabetesPedigreeFunction	-0.02545316	0.12324343	0.05133095	0.17829888	0.19271873	0.12571935	1.00000000	0.02656950	0.15545908
Age	0.53945719	0.25449621	0.23837508	-0.11103369	-0.08587910	0.03898737	0.02656950	1.00000000	0.23650925
Outcome	0.22443699	0.45842130	0.07595808	0.07604025	0.12092362	0.27672554	0.15545908	0.23650925	1.00000000

(b) When performing an LDA of the data you need to make a confusion matrix, sensitivity, specificity, and overall misclassification rate. A confusion matrix calculates the counts of true positives, true negatives, false positives, and false negatives. The true negative is

1174, the true positive is 386, the false positive is 142, and the false negative is 298. The sensitivity calculation was 0.564, meaning that approximately 56.4% of individuals with diabetes were correctly identified by the model. The specificity calculation came out to 0.8921 which concludes that approximately 89.21% of individuals without diabetes were correctly identified by the model. The overall misclassification rate was 0.22 which is 22%.



(c) When performing a QDA of the data you need to make a confusion matrix, sensitivity, specificity, and overall misclassification rate. A confusion matrix calculates the counts of true positives, true negatives, false positives, and false negatives. The true negative is 1135, the true positive is 394, the false positive is 181, and the false negative is 290. The sensitivity calculation was 57.6% meaning that individuals with diabetes were correctly identified by the model. The specificity calculation came out to 0.8625 which concludes that approximately 86.25% of individuals without diabetes were correctly identified by the model. The overall misclassification rate was 0.23 which is 23% of the data is misclassified by the model. The graph shows a high positive rate and a low false positive rate.



(d) When comparing the QDA and LDA models, you will get lots of differences. The QDA model does have a higher sensitivity than LDA but the specificity will be a bit lower. This means that the QDA model will perform better for identifying individuals compared to the LDA model. The misclassification rate is significantly higher in the QDA than in the LDA model. The cutoff value is used to maximize the models. And to make sure the sensitivity is maximized as well. I think that the QDA model is the best model for this data set.

Python Code (or R Code)

```
---
title: "Project 2 Stat4360"
output: pdf_document
date: "2024-02-12"
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```


```{r}
library(ggplot2)
wine <- read.table("~/wine.txt", header = T, sep = '')
#View(wine)
```

```{r}
head(wine)
attach(wine)
```
Question 1:
(a) Perform an exploratory analysis of data. Comment on findings that interest you.
```{r}
# make the full model of the data
full <- lm(Quality~Clarity + Aroma + Body + Flavor + Oakiness + Region)
summary(full)
# make a reduced model from the full model
```
```

```
reduce <- lm(Quality~Aroma + Flavor + Oakiness)
summary(reduce)
# confidence interval
confint(reduce, level = 0.95)
data <- data.frame(Aroma = 3.4, Flavor = 3.2, Oakiness = 4.2)
predict(reduce, newData = data, interval = 'predict')
# make a correlation
correlation = cor(wine[,c("Quality", "Clarity", "Aroma", "Body",
"Flavor", "Oakiness")])
print(correlation)
# make a scatter plot
pairs(wine[, -7], main = " Scatterplots")
summary(wine)
```
(b) Do part (a) of Exercise 15 in Chapter 3 for these data. For each predictor, fit a simple linear regression model to predict the response. Describe your results. In which of the models is there a statistically significant association between the predictor and the response? Create some plots to back up your assertions.
```{r}
predictors <- c("Clarity", "Aroma", "Body", "Flavor", "Oakiness",
"Region")
significant_predictors <- c()
# making the plots to find a correlation
for (predictor in predictors) {
  model <- lm(Quality ~ get(predictor), data = wine)
  summary_model <- summary(model)

  # Display results
  cat("\n\n=== Simple Linear Regression for", predictor, "===\n")
  print(summary_model)

  # Check for statistical significance (p-value < 0.05)
  if (summary_model$coefficients[2, "Pr(>|t|)"] < 0.05) {
    significant_predictors <- c(significant_predictors, predictor)
  }

  # Create scatter plots
  plot(wine[[predictor]], wine$Quality, main = paste("Scatter plot
for", predictor),
       xlab = predictor, ylab = "Quality")
  abline(model, col = "red")
}
Quality_Aroma = lm(wine$Quality ~ wine$Aroma)
summary(Quality_Aroma)
Quality_Body = lm(wine$Quality ~ wine$Body)
summary(Quality_Body)
Quality_Flavor = lm(wine$Quality ~ wine$Flavor)
summary(Quality_Flavor)
Quality_Oakiness = lm(wine$Quality ~ wine$Oakiness)
summary(Quality_Oakiness)
Quality_Region = lm(wine$Quality ~ wine$Region)
```

```
summary(Quality_Region)

```
(c) Do part (b) of Exercise 15 in Chapter 3 for these data.
Fit a multiple regression model to predict the response using
all of the predictors. Describe your results. For which predictors
can we reject the null hypothesis H0 : βj = 0?
```{r}
# Multiple Regression Model
multiple_model <- lm(Quality ~ Clarity + Aroma + Body + Flavor +
Oakiness + Region, data = wine)
summary_multiple_model <- summary(multiple_model)

# Display results for multiple regression
cat("\nMultiple Regression Model\n")
print(summary_multiple_model)

# Check for significant predictors (p-value < 0.05)
significant_predictors_multiple <-
names(which(summary_multiple_model$coefficients[, "Pr(>|t|)"] <
0.05))

# Diagnostic plots for multiple regression
par(mfrow = c(2, 2))
plot(multiple_model)

# Display significant predictors in the multiple regression model
cat("\nSignificant predictors in the multiple regression model:",
paste(significant_predictors_multiple, collapse = ", "), "\n")
# because less than 0.05 i got flavor and oakiness as the two
significant predictions both having a positive direction of
association and I can reject the null hypothesis for these two
predictors
```
(d) Based on your observation in (b) and (c), build a “reasonably
good” multiple
regression model for these data. Be sure to explore interactions of
Region with
other predictors. Carefully justify all the choices you make in
building the
model and verify the model assumptions.
```{r}
reduced_model <- lm(Quality~Aroma + Flavor + Oakiness)
summary(reduced_model)
confint(reduced_model, level = 0.95)
df = data.frame(Aroma = 3.4, Flavor = 3.2, Oakiness = 4.2)
predict(reduced_model, newdata = df, interval = 'predict')
anova(reduced_model)
summary(reduced_model)
```
(e) Write the final model in equation form, being careful to handle
the

qualitative predictors and interactions (if any) properly.
(f) Use the final model to predict the Quality of a wine from Region 1 with other predictors set
equal to their sample means. Also provide a 95% prediction interval for the response and a 95%
confidence interval for the mean response. Interpret the results.

```{r}
# calculating mean values
aroma_mean <- mean(wine$Aroma[wine$Region == 1], na.rm = TRUE)
print(aroma_mean)
flavor_mean <- mean(wine$Flavor[wine$Region == 1], na.rm = TRUE)
print(flavor_mean)
oakiness_mean <- mean(wine$Oakiness[wine$Region == 1], na.rm = TRUE)
print(oakiness_mean)

intercept = 6.4672
aroma_coeff = 0.5801
flavor_coeff = 1.1997
oakiness_coeff = -0.6023

predicted_val = intercept + aroma_coeff*aroma_mean +
flavor_coeff*flavor_mean + oakiness_coeff*oakiness_mean
print(predicted_val)

# standard error
residualStandardError = summary(reduced_model)$sigma
SE_predict = residualStandardError * sqrt(1+sum(c((aroma_mean -
mean(wine$Aroma))^2,
                                                  (flavor_mean -
mean(wine$Flavor))^2,
                                                 (oakiness_mean -
mean(wine$Oakiness))^2)))

# 95 percent prediction interval
df_residual <- summary(reduced_model)$df[3]
t_val = qt(0.975, df_residual)
error_margin = t_val * SE_predict
prediction_interval = c(predicted_val - error_margin, predicted_val +
error_margin)
print(prediction_interval)

# 95 percent confidence interval
mean_response = SE_predict / sqrt(length(wine$Quality[wine$Region ==
1]))
t_mean = qt(0.975, df_residual)
error_margin_mean = t_mean * mean_response
conf_interval_mean = c(predicted_val - error_margin_mean,
predicted_val + error_margin_mean)
print(conf_interval_mean)
```

Question 2

```{r}
```

```
diabetes <- read.csv2("C:/Users/sayem/Downloads/diabetes.csv",
sep="")
str(diabetes)
summary(diabetes)
colnames(diabetes) <- gsub("\\.\\.", "", colnames(diabetes))
names(diabetes)
correlation_matrix = cor(diabetes [, c("Pregnancies", "Glucose",
"BloodPressure",
"SkinThickness", "Insulin", "BMI", "DiabetesPedigreeFunction", "Age",
"Outcome")])
print(correlation_matrix)
pairs(diabetes[, -10], main = "Scatterplot Matrix")


```
Part B
```{r}
lda_model = lda(Outcome ~ ., data = diabetes)
lda_predictions <- predict(lda_model, diabetes)
matrix <- table(Actual = diabetes$Outcome, Predicted =
lda_predictions$class)

# Calculate sensitivity, specificity, and misclassification rate
sensitivity <- matrix[2, 2] / sum(diabetes$Outcome == 1)
specificity <- matrix[1, 1] / sum(diabetes$Outcome == 0)
misclassification_rate <- (matrix[1, 2] + matrix[2, 1]) / sum(matrix)

# Print the confusion matrix and metrics
print(matrix)
cat("Sensitivity:", sensitivity, "\n")
cat("Specificity:", specificity, "\n")
cat("Misclassification Rate:", misclassification_rate, "\n")

# Plot ROC curve
library(pROC)
rocCurve <- roc(diabetes$Outcome, lda_predictions$posterior[, 2])
par(mfrow = c(1,1))
plot(rocCurve, main = "ROC Curve for LDA", col = "blue", lwd = 2)
```
Part C
```{r}
# libraries
library(MASS)
library(pROC)

# Perform QDA
model <- qda(Outcome ~ ., data = diabetes)
# Make predictions
qda_predic <- predict(model, diabetes)
# Compute confusion matrix
conf_matrix_qda <- table(Actual = diabetes$Outcome, Predicted =
qda_predictions$class)
```

```
sensitivity_qda = conf_matrix_qda[2, 2] / sum(diabetes$Outcome == 1)
specificity_qda = conf_matrix_qda[1, 1] / sum(diabetes$Outcome == 0)
misclassification_rate_qda =
(conf_matrix_qda[1, 2] + conf_matrix_qda[2, 1]) /
sum(conf_matrix_qda)
# Print confusion matrix and metrics for QDA
print(conf_matrix_qda)
cat("Sensitivity (QDA):", sensitivity_qda, "\n")
cat("Specificity (QDA):", specificity_qda, "\n")
cat("Misclassification Rate (QDA):", misclassification_rate_qda,
"\n")
# Plot ROC curve for QDA
roc_curve <- roc(diabetes$Outcome, qda_predict$posterior[, 2])
plot(roc_curve, main = "ROC Curve for QDA", col = "blue", lwd = 2)
```