STAT 4360, Fall 2024
Benny Frisella

Mini Project II

Section 1

1)

a) In the *Pinot Noir* dataset, we have 38 observations of 7 variables and no missing fields. Summary statistics are presented in Figure 1.

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 1

Since the response variable, Quality, is quantitative, we will employ Linear Regression instead of Logistic Regression. Additionally, notice the ranges of each ranked from highest to least:

Quality(8.2), Flavor(4.1), Body(4.0), Aroma(3.7), Oakiness(3.1), Clarity(0.5), Region(Discrete)

Since the variables are measured on different scales but none exhibit extreme values or significant discrepancies in range, normalization of the predictors will not be necessary. Linear regression can handle predictors of varying scales if they are consistently scaled. *Note: Clarity may not be sufficiently scaled, and Regions are disproportionately distributed at*

- *17 from Region 1*
- *9 from Region 2*
- *12 from Region 3*

b) Individual analyses of each predictor against the response variable revealed that **Aroma, Body, Flavor, and Region were all statistically significant**. This suggests they each reliably effect Pinot Noir Quality. Clarity and Oakiness exhibited high p-values, indicating that we cannot reject the null hypothesis that they have no significant impact on determining Quality.

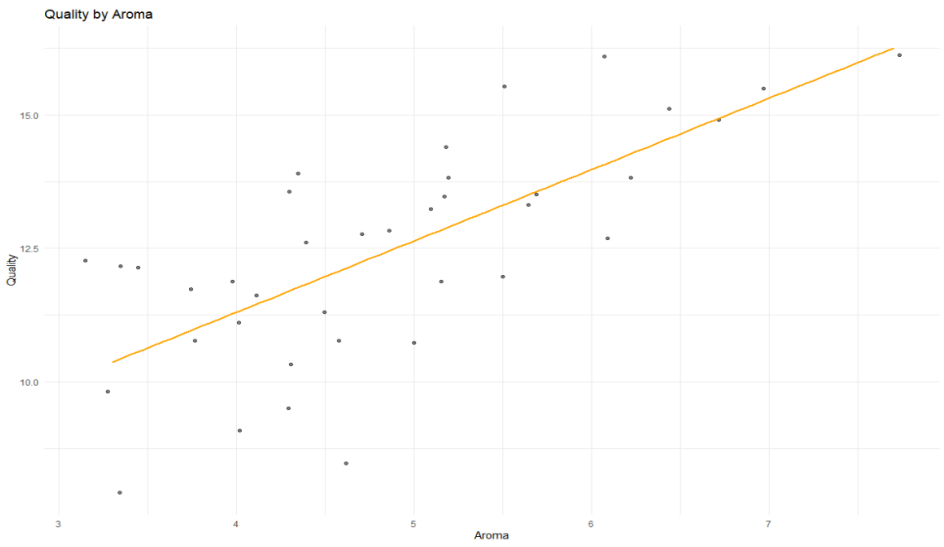

Figure 2

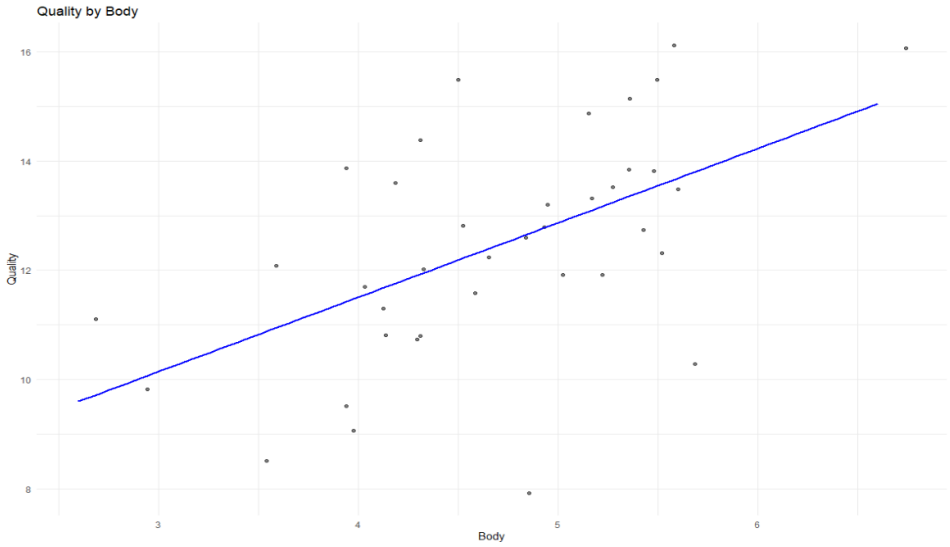

Figure 3

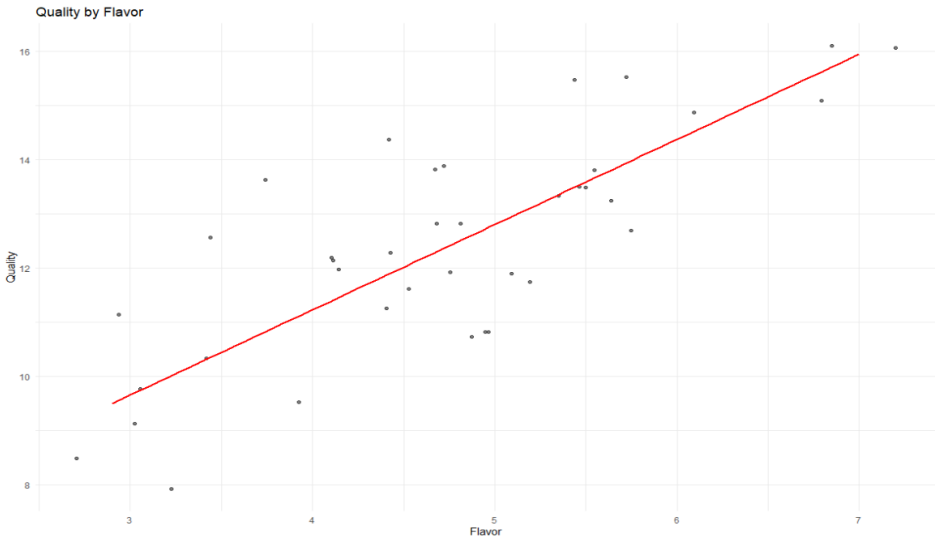

Figure 4

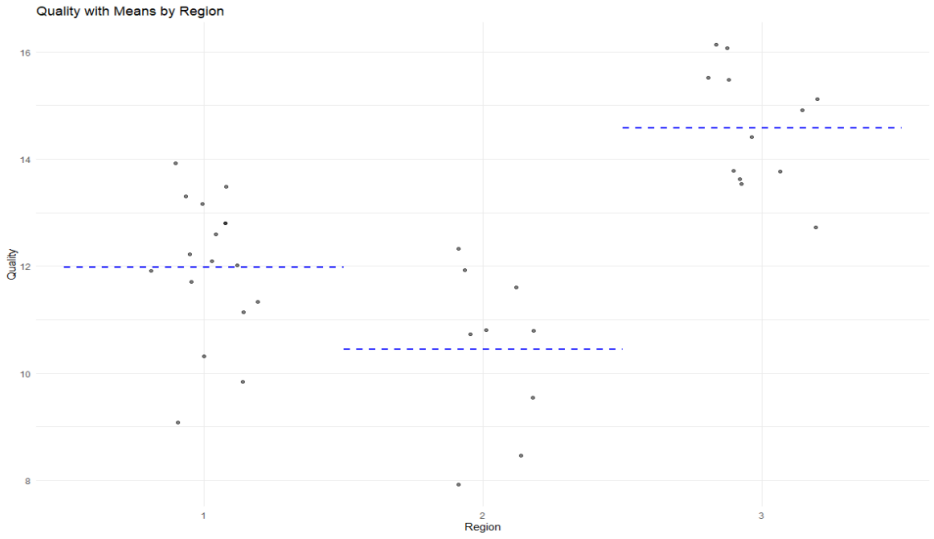

Figure 5

Probabilities

P (> F)

Aroma: 6.871e-07

Body: 3.612e-04

Flavor: 3.683e-09

Region: 6.587e-08

With probabilities extremely close to zero, **these attributes were all significant.**

Figures 2 – 4 illustrate how the linear models of each predictor could accurately predict results in Pinot Noir Quality to a certain degree.

Figure 5 illustrates the Quality of Pinot Noir in each of the three regions. The dotted lines represent the means in each region. In this data, on average, the highest quality Regions for Pinot Noir are Region 3, Region 1, then Region 2.

c) ANOVA results from using **all predictors together** show that Aroma, Body, Flavor, Oakiness, and Region are all significant predictors of Quality; Again, Aroma, Flavor, and Region were particularly strong**. All variables besides Clarity showed a significant relationship with Quality.** *Reference Table 1*.

```
Analysis of Variance Table

Response: Quality
                 Df Sum Sq Mean Sq F value    Pr(>F)
Clarity           1  0.125   0.125  0.1494 0.7018243
Aroma             1 77.353  77.353 92.3064 1.152e-10 ***
Body              1  6.414   6.414  7.6544 0.0096032 **
Flavor            1 19.050  19.050 22.7324 4.484e-05 ***
Oakiness          1  8.598   8.598 10.2598 0.0032129 **
as.factor(Region)  2 18.108   9.054 10.8042 0.0002924 ***
Residuals         30 25.140   0.838
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Table 1

d) To find a better linear model, I tested the **interactions** between Aroma, Body, Flavor, and Region in every way possible and **found nothing of significance**. Then, after seeing *Table 1,* I removed Clarity as it was insignificant but kept the rest of the predictors. *Reference Table 2 for the latest model*.

```
Analysis of Variance Table

Response: Quality
                 Df Sum Sq Mean Sq F value    Pr(>F)
Aroma             1 77.442  77.442 95.4921 5.543e-11 ***
Body              1  5.703   5.703  7.0322  0.012502 *
Flavor            1 18.878  18.878 23.2786 3.547e-05 ***
Oakiness          1  7.060   7.060  8.7049  0.005994 **
as.factor(Region)  2 20.565  10.283 12.6795 9.466e-05 ***
Residuals         31 25.140   0.811
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Table 2

Clearly, the removal of Clarity improved the model as the Residual Mean Square dropped by 0.027.
The F statical probabilities of **Aroma, Flavor and Region were also reduced** with only slight increases to Body and Oakiness.

Model assumptions are verified in R Code section 1d).

e) The Final Model:

$$\text{Quality} = \beta_0 + \beta_1 \times \text{Aroma} + \beta_2 \times \text{Body} + \beta_3 \times \text{Flavor} + \beta_4 \times \text{Oakiness} + \beta_5 \times \text{Region2} + \beta_6 \times \text{Region3} + \epsilon$$

Where:

- β_0 is the intercept, representing the expected **Quality** when **Aroma**, **Body**, **Flavor**, **Oakiness**, and **Region2/Region3** are zero, and **Region1** is the baseline.
- $\beta_1 \times \text{Aroma}$, $\beta_2 \times \text{Body}$, $\beta_3 \times \text{Flavor}$, and $\beta_4 \times \text{Oakiness}$ represent the effect of these continuous variables on **Quality**.
- $\beta_5 \times \text{Region2}$ and $\beta_6 \times \text{Region3}$ represent the difference in **Quality** between **Region2** and **Region3**, respectively, compared to **Region1**.

In this model:

- **Region1** is absorbed into the intercept β_0, so it's not explicitly represented in the equation.
- **Region2** and **Region3** are the dummy variables, and their coefficients indicate how much the **Quality** score changes compared to **Region1**.

f) **Use Case:** Using the model to predict a Pinot Noir from Region 1 with other predictors set to their sample means. A 95% prediction interval for the response and a 95% confidence interval for the mean response are also provided.
Region 1 is the base case and thus inherited by the intercept, so set the other regions to zero. The resulting equation:

Quality = 7.8324 + 0.0892 * Aroma + 0.0784 * Body + 1.1171 * Flavor - 0.3456 * Oakiness - 1.514 * Region2 + 0.9737 * Region3

Interpretation:

- **Intercept (7.8324)**: This is the expected **Quality** of the wine when all predictors are set to zero, and the wine is from **Region 1** (baseline).

- **Aroma (0.0892)**: For each unit increase in **Aroma**, the **Quality** is expected to increase by **0.0892**, all other variables constant.

- **Body (0.0784)**: For each unit increase in **Body**, the **Quality** is expected to increase by **0.0784**, all other variables constant.

- **Flavor (1.1171)**: For each unit increase in **Flavor**, the **Quality** is expected to increase significantly by **1.1171**, all other variables constant.

- **Oakiness (-0.3456)**: For each unit increase in **Oakiness**, the **Quality** is expected to decrease by **0.3456**, all other variables constant. ***This indicates that higher Oakiness negatively impacts the perceived quality of Pinot Noir.***

- **Region2 (-1.514)**: Being in **Region 2** is associated with a decrease in **Quality** by **1.514** compared to Region 1, all other factors constant. This can be confirmed from *Figure 5*.

- **Region3 (0.9737)**: Being in **Region 3** is associated with an increase in **Quality** by **0.9737** compared to Region 1, all other factors constant. Again, this is confirmed by *Figure 5*.

Lastly, Table 3 shows a 95% confidence interval a 95% prediction interval using the latest model

fit	lwr	upr
12.48795	11.98974	12.98616
fit	lwr	upr
12.48795	10.58491	14.39099

Table 3 top line shows a 95% confidence interval while the bottom a 95% prediction interval

- From the confidence interval, for a new wine from **Region 1** with average levels of **Aroma**, **Body**, **Flavor**, and **Oakiness**, one can expect its **Quality** score to fall between approximately **11.99** and **12.99** with **95% confidence**.
- From the prediction interval, it can be inferred that for wines from **Region 1**, the average **Quality** is predicted to be between **10.58** and **14.39** with a mean estimate of **12.49** when other factors are held constant at their means.

2)

a) In the *Diabetes* dataset of 2000 observations of 7 variables out of a hospital in Frankfurt, Germany I noticed a couple of issues:

- Missing values
  - Insulin: 956
  - Skin Thickness: 573
  - Blood Pressure: 90
  - BMI: 28
  - Glucose: 13
- Outliers?
  - Pregnancies: 3 records had 17 pregnancies. This is the maximum value of this attribute.

Luckily, there are no missing values in Pedigree Function nor in Age and all missing values were filled with zeroes.

b) **After performing LDA**, Table 4 below shows the resulting Confusion Matrix.

	Actual	
Predicted	0	1
0	1174	298
1	142	386

Table 4

The current LDA model performs better at correctly identifying patients without Diabetes than with Diabetes.

From Table 4, let

TP = True Positive Rate = 386
TN = True Negative Rate = 1174
FP = False Positive Rate = 142
FN = False Negative Rate = 298

The model accuracy,

$$\frac{TP + TN}{TP + TN + FP + FN} = \frac{1174 + 386}{1174 + 298 + 142 + 386} = \frac{1560}{2000} = 0.78\,(78\%)$$

indicates the misclassification rate was 22% (100% – 78%)

The Specificity (True Negative Rate),

$$\frac{TN}{TN + FP} = \frac{1174}{1174 + 298} = \frac{1174}{1472} \approx 0.7973$$

indicates the model correctly identified patients without Diabetes at 79.73%

The Sensitivity (True Positive Rate),

$$\frac{TP}{TP + FN} = \frac{386}{386 + 142} = \frac{386}{528} \approx 0.73\,(73\%)$$

indicates the model correctly identified patients with Diabetes at 73%

The ROC curve for LDA can be seen below in Figure 6,

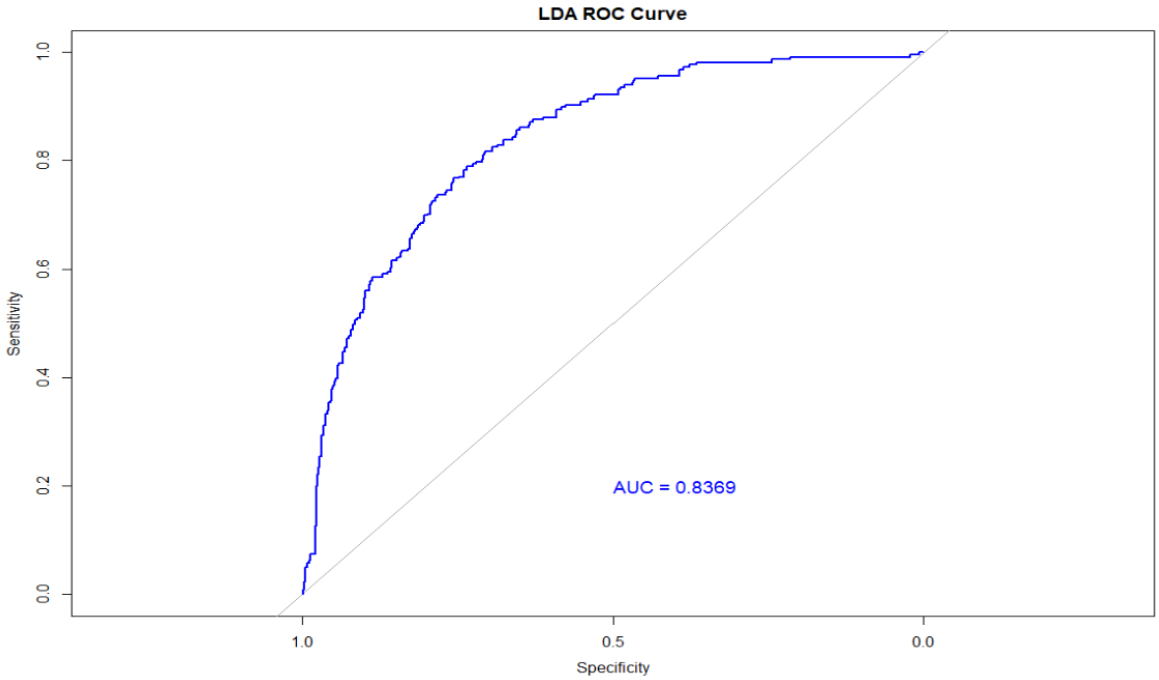

Figure 6

with a shape that's indicative of the model accurately predicting true positives while keeping false positives low – although there is room for improvement, the **Area Under the Curve of 0.84** demonstrates that the model does a good job of distinguishing between the diabetic outcomes of patients.

c) After performing QDA, Table 5 below shows the resulting Confusion Matrix.

	Actual	
Predicted	0	1
0	1135	290
1	181	394

Table 5

From Table 5, let

TP = True Positive Rate = 394
TN = True Negative Rate = 1135
FP = False Positive Rate = 181
FN = False Negative Rate = 290

Following the same formulas as the previous Confusion Matrix, **QDA gives a model accuracy of 76%, which leaves the misclassification rate at 24% for QDA**

The Specificity indicates the model correctly identified patients without Diabetes at 79.47%

The Sensitivity indicates the model correctly identified patients with Diabetes at 68.52%

The ROC curve for QDA can be seen below in Figure 7,

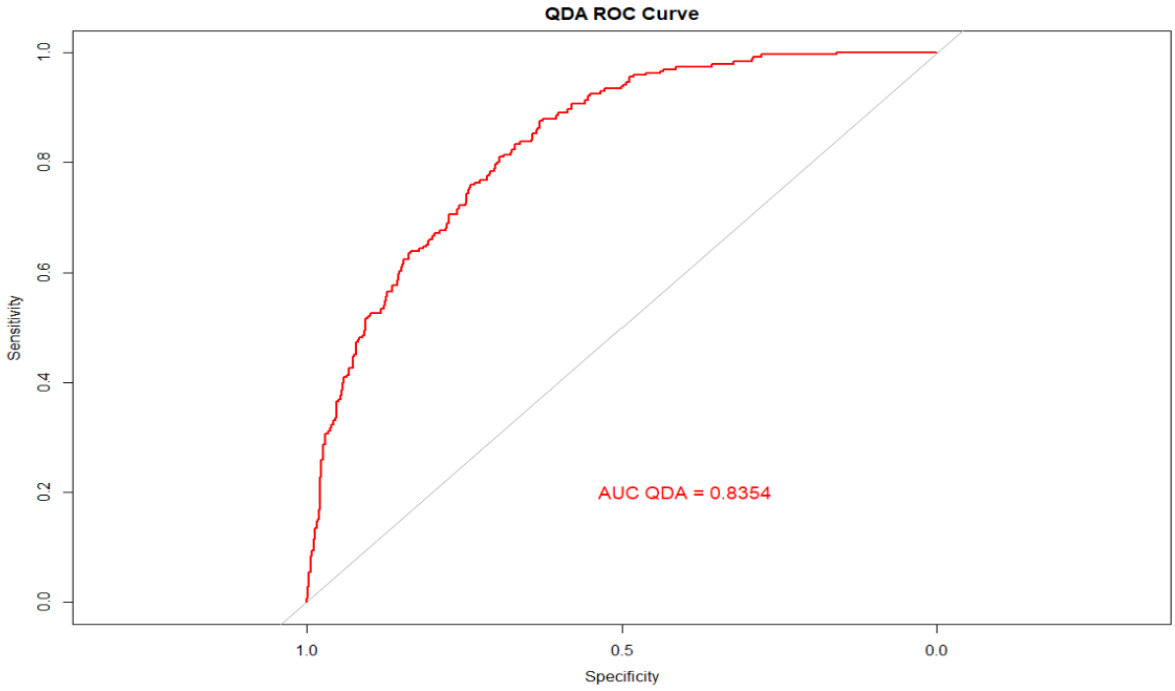

Figure 7

Clearly, the QDA ROC is shaped in the matter we want and the **Area Under the Curve of 0.84** demonstrates that the model does a good job of distinguishing between the diabetic outcomes of patients.

d) Although the models were close in terms of performance, **LDA faired 2% better than QDA in misclassification rate and 5% better in terms of Sensitivity, with all else equal, therefore with the posterior probability cutoff at 50%, I would recommend LDA** as it will perform at least as well or better than QDA.

<u>However, with the classes of diabetics/non-diabetics out of balance, I would recommend lowering the posterior probability cutoffs.</u>

With a **posterior probability cutoff of 33%**, the **LDA** model maintains a good balance between precision and recall while having the highest accuracy among all LDA sets (from **PPC < 0.5**).

```
[1] "Metrics for LDA:"
   Accuracy   Precision      Recall Specificity    F1_Score
  0.7640000   0.7368421   0.6331658   0.8504983   0.6810811
```

With a **posterior probability cutoff of 25%**, the **QDA** model offers a good balance between precision and recall, with the highest specificity among all QDA sets (from **PPC < 0.5**).

```
[1] "Metrics for QDA:"
   Accuracy   Precision      Recall Specificity    F1_Score
  0.7415000   0.7631579   0.5952109   0.8557435   0.6688020
```

R code

Section 2

#Question 1#
#(a)

```r
#Exploratory Data Analysis
wineData <- read.csv("wineData.txt", header = TRUE, sep = '\t')
head(wineData)

#check out main characteristics
summary(wineData)

#check regions
regions <- table(wineData$Region)  #17 from region 1, 9 from 2, 12 from 3
```

```r
#fit linear models of each predictor against response

qualClarity <- lm(Quality ~ Clarity, data = wineData)
qualAroma <- lm(Quality ~ Aroma, data = wineData)
qualBody <- lm(Quality ~ Body, data = wineData)
qualFlavor <- lm(Quality ~ Flavor, data = wineData)
qualOakiness <- lm(Quality ~ Oakiness, data = wineData)
qualRegion <- lm(Quality ~ as.factor(Region), data = wineData)

```

```r
#evaluate F statistic to test model significance
anova(qualClarity)
print("")
anova(qualAroma)
print("")
anova(qualBody)
print("")
anova(qualFlavor)
print("")
anova(qualOakiness)
print("")
anova(qualRegion)
#Aroma, Body, Flavor, Region are all significant ***
```

```r
```

```r
#Region's interactions with the other signifcant predictors
# note: * captures all effects, : captures interaction only

regionAr <- lm(Quality ~ Aroma * Region, data = wineData)
anova(regionAr)
print("")
regionBod <- lm(Quality ~ Body * Region, data = wineData)
anova(regionBod)
print("")
regionFlav <- lm(Quality ~ Flavor * Region, data = wineData)
anova(regionFlav)

#Interaction Effects: The interaction terms were not significant,
#indicating that the influence of each predictor does not change based on the Region.
```

```r
#Flavor's interactions with Aroma and Body
flavorAr <- lm(Quality ~ Flavor * Aroma, data = wineData)
anova(flavorAr)
flavorBod <- lm(Quality ~ Body * Flavor, data = wineData)
anova(flavorBod)
```

```r
#lastly, check Aroma with Body
bodyAr <- lm(Quality ~ Body * Aroma, data = wineData)
anova(bodyAr)
```

#(b)

```r
library(ggplot2)

#scatter plot for Aroma/Quality
ggplot(wineData, aes(x = Aroma, y = Quality)) +
  geom_jitter(width = 0.2, alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE, color = "orange") +
  labs(title = "Quality by Aroma", x = "Aroma", y = "Quality") +
  theme_minimal()

#scatter plot for Body/Quality
ggplot(wineData, aes(x = Body, y = Quality)) +
  geom_jitter(width = 0.2, alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Quality by Body", x = "Body", y = "Quality") +
  theme_minimal()
```

```
#scatter plot for Flavor/Quality
ggplot(wineData, aes(x = Flavor, y = Quality)) +
  geom_jitter(width = 0.2, alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Quality by Flavor", x = "Flavor", y = "Quality") +
  theme_minimal()

library(dplyr)

#calculate mean Quality per Region
meanQuality <- wineData %>%
  group_by(Region) %>%
  summarize(meanQuality = mean(Quality, na.rm = TRUE))

#show organized mean lines
mean_lines <- data.frame(
  Region = c(1, 2, 3),  # Assuming there are 3 regions
  mean_quality = mean_quality$mean_quality,
  x_start = c(0.5, 1.5, 2.5),  #start of the line for each region
  x_end = c(1.5, 2.5, 3.5)  #end of the line for each region
)

#scatter plot for Region/Quality with means
ggplot(wineData, aes(x = as.factor(Region), y = Quality)) +
  geom_jitter(width = 0.2, alpha = 0.5) +  # Add jitter to points for visibility
  geom_smooth(method = "lm", se = FALSE, color = "green") +  # Add linear model fit
  geom_segment(data = mean_lines,
               aes(x = x_start, xend = x_end, y = mean_quality, yend = mean_quality),
               linetype = "dashed", color = "blue", size = 1) +  # Add horizontal lines for mean
  labs(title = "Quality by Region", x = "Region", y = "Quality") +
  theme_minimal()

```
#(c)
```{r}
#fit a lm using all predictors

allPredictorModel <- lm(Quality ~ Clarity + Aroma + Body + Flavor + Oakiness + as.factor(Region), data =
wineData)
anova(allPredictorModel)

```
#(d)

```{r}
#remove Clarity
latestModel <- lm(Quality ~ Aroma + Body + Flavor + Oakiness + as.factor(Region), data = wineData)
anova(allPredictorModel)
```

```
```
```{r}
#extract coefficients
coefficients <- coef(latestModel)

#set coefficients
intercept <- coefficients[1]
betaAroma <- coefficients["Aroma"]
betaBody <- coefficients["Body"]
betaFlavor <- coefficients["Flavor"]
betaOakiness <- coefficients["Oakiness"]
betaRegion2 <- coefficients["as.factor(Region)2"]
betaRegion3 <- coefficients["as.factor(Region)3"]

#format as string
equation <- paste0("Quality = ", round(intercept, 4), " + ",
                   round(beta_aroma, 4), " * Aroma + ",
                   round(beta_body, 4), " * Body + ",
                   round(beta_flavor, 4), " * Flavor + ",
                   round(beta_oakiness, 4), " * Oakiness + ",
                   round(beta_region2, 4), " * Region2 + ",
                   round(beta_region3, 4), " * Region3")

print(equation)

```
```

#(f)

```{r}
#mean values of predictors
meanAroma <- mean(wineData$Aroma)    # Replace with actual mean value
meanBody <- mean(wineData$Body)      # Replace with actual mean value
meanFlavor <- mean(wineData$Flavor)   # Replace with actual mean value
meanOakiness <- mean(wineData$Oakiness) # Replace with actual mean value

#create new data frame for prediction
newData <- data.frame(
  Aroma = meanAroma,
  Body = meanBody,
  Flavor = meanFlavor,
  Oakiness = meanOakiness,
  Region = factor("1", levels = c("1", "2", "3"))  #region 1 as the baseline
)

# Make predictions with confidence interval
confidence <- predict(latestModel, newdata = newData, interval = "confidence", level = 0.95)
confidence
```
```

```
# Make predictions with prediction interval
predictions <- predict(latestModel, newdata = newData, interval = "prediction", level = 0.95)
predictions


```

```{r}
#                         1d) ASSUMPTION VALIDATIONs FOR LATEST LINEAR MODEL
fittedLatestModel <- fitted(latestModel)
residuals <- resid(latestModel)

 #residual plot
 plot(fittedLatestModel, residuals)
 abline(h = 0)

 #QQ plot
 qqnorm(residuals)

 #time series plot of residuals
 plot(residuals, type="l")
 abline(h=0)

 print("Residual Plot: Make sure there isn't any patterns

QQ Plot: Points should lie along the reference line. Deviations from this line, especially at the ends, indicate departures from normality.

Time Series Plot of Residuals: Residuals should hover around 0 without significant trends or patterns. Patterns may indicate problems with the model specification.")

```

#Question 2

#(a)

```{r}
#                               QUESTION 2
diabetes <- read.csv("diabetes.csv")
summary(diabetes)
```

```{r}
#find missing values (they were set to zero) but it's impossible for these to be zero
sum(diabetes$Glucose == 0)       # 13
sum(diabetes$BloodPressure == 0)  # 90
sum(diabetes$SkinThickness == 0)  # 573
sum(diabetes$BMI == 0)       # 28
sum(diabetes$Age == 0)    # 0
sum(diabetes$Insulin == 0)     # 956
```

```
sum(diabetes$DiabetesPedigreeFunction == 0)   # 0

which(diabetes$Pregnancies == max(diabetes$Pregnancies))    # 3 results
```

(b)

```{r}
#perform LDA, using MASS to save space
library(MASS)

#fit LDA model using all variables to predict
LDA <- lda(Outcome ~ ., data = diabetes)

#predict
predictions <- predict(LDA, diabetes)

#check
#predictions$class

#create confusion matrix (shorthand)
table(Predicted = predictions$class, Actual = diabetes$Outcome)

#classify based on a 0.5 cutoff
predictedClasses <- ifelse(predictions$posterior >= 0.5, 1, 0)

```
```{r}
library(pROC)

LDA.ROC <- roc(diabetes$Outcome, predictions$posterior[, 2])

#plot Reciever Operating Characteristic
plot(LDA.ROC, col = "blue", main = "LDA ROC Curve")

#print Area Under Curve
AUC.LDA <- auc(LDA.ROC)
AUC.LDA

text(0.4, 0.2, paste("AUC =", round(AUC.LDA, 4)), col = "blue", cex = 1.2)

```

#(c)

```{r}
#fit QDA model using all variables to predict
QDA <- qda(Outcome ~ ., data = diabetes)

#predict
predictions2 <- predict(QDA, diabetes)
```

```
#check
#predictions2$class

#create confusion matrix (shorthand)
table(Predicted = predictions2$class, Actual = diabetes$Outcome)

#classify based on a 0.5 cutoff
predictedClasses2 <- ifelse(predictions2$posterior >= 0.5, 1, 0)
```
```{r}
QDA.ROC <- roc(diabetes$Outcome, predictions2$posterior[, 2])

#plot Receiver Operating Characteristic
plot(QDA.ROC, col = "red", main = "QDA ROC Curve")

#print Area Under Curve
AUC.QDA <- auc(QDA.ROC)
AUC.QDA

text(0.4, 0.2, paste("AUC QDA =", round(AUC.QDA, 4)), col = "red", cex = 1.2)

```
#(d)

```{r}
#test other posterior probabilities for QDA and LDA
predictedClasses <- ifelse(predictions$posterior[, 2] >= 0.33, 1, 0)

# Classify based on a 0.5 cutoff (or other cutoff like 0.33)
predictedClasses2 <- ifelse(predictions2$posterior[, 2] >= 0.25, 1, 0)

#create confusion matrix (shorthand)
confusionQDA <- table(Predicted = predictedClasses2, Actual = diabetes$Outcome)

#create confusion matrix (shorthand)
confusionLDA <- table(Predicted = predictedClasses, Actual = diabetes$Outcome)

```
```{r}
#calculate metrics
confusionMetrics <- function(confusionMatrix)
{
  TP <- confusionMatrix[2, 2]  #true Positives
  TN <- confusionMatrix[1, 1]  #true Negatives
  FP <- confusionMatrix[1, 2]  #false Positives
  FN <- confusionMmatrix[2, 1]  #false Negatives

  accuracy <- (TP + TN) / sum(confusionMatrix)
```

```
  precision <- TP / (TP + FP)
  recall <- TP / (TP + FN)
  specificity <- TN / (TN + FP)
  f1_score <- 2 * (precision * recall) / (precision + recall)

  return(c(Accuracy = accuracy, Precision = precision, Recall = recall, Specificity = specificity, F1_Score =
f1_score))
}

#QDA
metrics_QDA <- calculate_metrics(confusionQDA)
print("Metrics for QDA:")
print(metrics_QDA)

#LDA
metrics_LDA <- calculate_metrics(confusionLDA)
print("Metrics for LDA:")
print(metrics_LDA)

```