STAT 425 - Final Case Study

2024-12-04

Part 1:

```
df <- read.csv("~/Desktop/uiuc/stat425/Case Study 2/wines.csv")
head(df)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1           7.4             0.70        0.00            1.9     0.076
## 2           7.8             0.88        0.00            2.6     0.098
## 3           7.8             0.76        0.04            2.3     0.092
## 4          11.2             0.28        0.56            1.9     0.075
## 5           7.4             0.70        0.00            1.9     0.076
## 6           7.4             0.66        0.00            1.8     0.075
##   free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 1                  11                   34  0.9978 3.51      0.56     9.4
## 2                  25                   67  0.9968 3.20      0.68     9.8
## 3                  15                   54  0.9970 3.26      0.65     9.8
## 4                  17                   60  0.9980 3.16      0.58     9.8
## 5                  11                   34  0.9978 3.51      0.56     9.4
## 6                  13                   40  0.9978 3.51      0.56     9.4
##      type
## 1 redwine
## 2 redwine
## 3 redwine
## 4 redwine
## 5 redwine
## 6 redwine
```

Summary Statistics:

We will first be setting the white wine variable into binary and then checking if there are any missing values in our data frame.

```
df$type <- ifelse(df$type == "whitewine", 1, 0)
print(sum(is.na(df)))
```

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

Now we will start the explaratory data analysis, choosing summary statistics and plots to describe the data. To start, we have a correlation matrix:

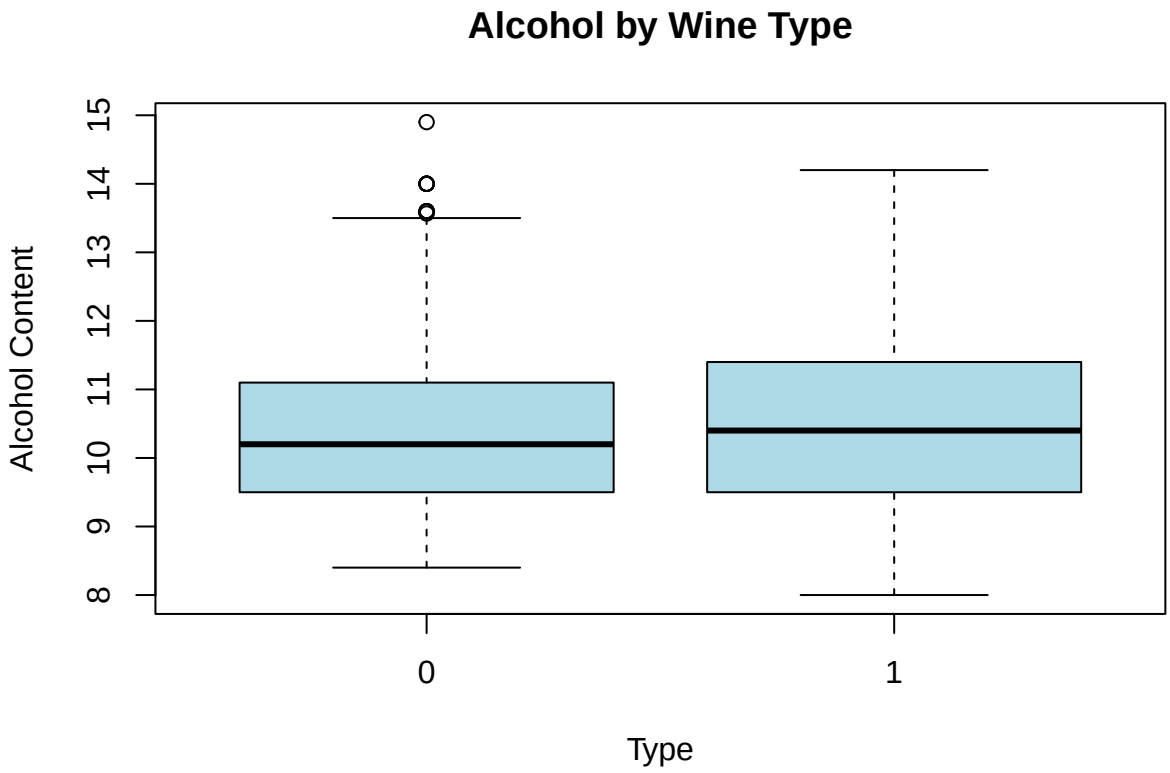
```
cor(df[, sapply(df, is.numeric)], use = "complete.obs")
```

```
##                       fixed.acidity volatile.acidity citric.acid residual.sugar
## fixed.acidity           1.00000000       0.21900826  0.32443573     -0.1119813
## volatile.acidity        0.21900826       1.00000000 -0.37798132     -0.1960112
## citric.acid             0.32443573      -0.37798132  1.00000000      0.1424512
## residual.sugar         -0.11198128      -0.19601117  0.14245123      1.0000000
## chlorides               0.29819477       0.37712428  0.03899801     -0.1289405
## free.sulfur.dioxide    -0.28273543      -0.35255731  0.13312581      0.4028706
## total.sulfur.dioxide   -0.32905390      -0.41447619  0.19524198      0.4954816
## density                 0.45890998       0.27129565  0.09615393      0.5525170
## pH                     -0.25270047       0.26145440 -0.32980819     -0.2673198
## sulphates               0.29956774       0.22598368  0.05619730     -0.1859274
## alcohol                -0.09545152      -0.03764039 -0.01049349     -0.3594148
## type                   -0.48673983      -0.65303559  0.18739650      0.3488210
##                        chlorides free.sulfur.dioxide total.sulfur.dioxide
## fixed.acidity         0.29819477         -0.28273543          -0.32905390
## volatile.acidity      0.37712428         -0.35255731          -0.41447619
## citric.acid           0.03899801          0.13312581           0.19524198
## residual.sugar       -0.12894050          0.40287064           0.49548159
## chlorides             1.00000000         -0.19504479          -0.27963045
## free.sulfur.dioxide  -0.19504479          1.00000000           0.72093408
## total.sulfur.dioxide -0.27963045          0.72093408           1.00000000
## density               0.36261466          0.02571684           0.03239451
## pH                    0.04470798         -0.14585390          -0.23841310
## sulphates             0.39559331         -0.18845725          -0.27572682
## alcohol              -0.25691558         -0.17983843          -0.26573964
## type                 -0.51267825          0.47164366           0.70035716
##                           density          pH    sulphates      alcohol
## fixed.acidity         0.45890998 -0.25270047  0.299567744 -0.095451523
## volatile.acidity      0.27129565  0.26145440  0.225983680 -0.037640386
## citric.acid           0.09615393 -0.32980819  0.056197300 -0.010493492
## residual.sugar        0.55251695 -0.26731984 -0.185927405 -0.359414771
## chlorides             0.36261466  0.04470798  0.395593307 -0.256915580
## free.sulfur.dioxide   0.02571684 -0.14585390 -0.188457249 -0.179838435
## total.sulfur.dioxide  0.03239451 -0.23841310 -0.275726820 -0.265739639
## density               1.00000000  0.01168608  0.259478495 -0.686745422
## pH                    0.01168608  1.00000000  0.192123407  0.121248467
## sulphates             0.25947850  0.19212341  1.000000000 -0.003029195
## alcohol              -0.68674542  0.12124847 -0.003029195  1.000000000
## type                 -0.39064532 -0.32912865 -0.487217970  0.032969551
##                             type
## fixed.acidity        -0.48673983
## volatile.acidity     -0.65303559
## citric.acid           0.18739650
## residual.sugar        0.34882101
## chlorides            -0.51267825
## free.sulfur.dioxide   0.47164366
## total.sulfur.dioxide  0.70035716
## density              -0.39064532
## pH                   -0.32912865
## sulphates            -0.48721797
## alcohol               0.03296955
## type                  1.00000000
```

This correlation matrix shows low evidence of multicollinearity; however, there are a few instances where

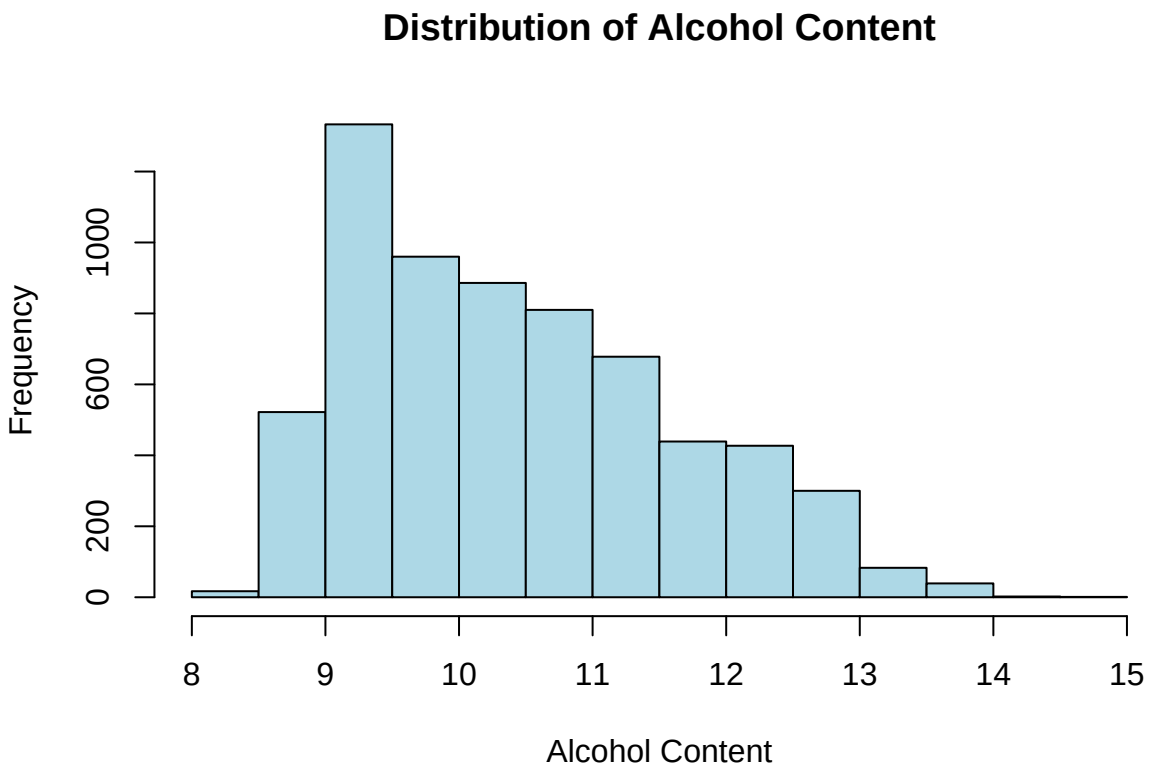
multicollinearity is present such as free.sulfur.dioxide and total.sulfur.dioxide.

```
boxplot(alcohol ~ type, data = df, main = "Alcohol by Wine Type",
        xlab = "Type", ylab = "Alcohol Content", col = "lightblue")
```

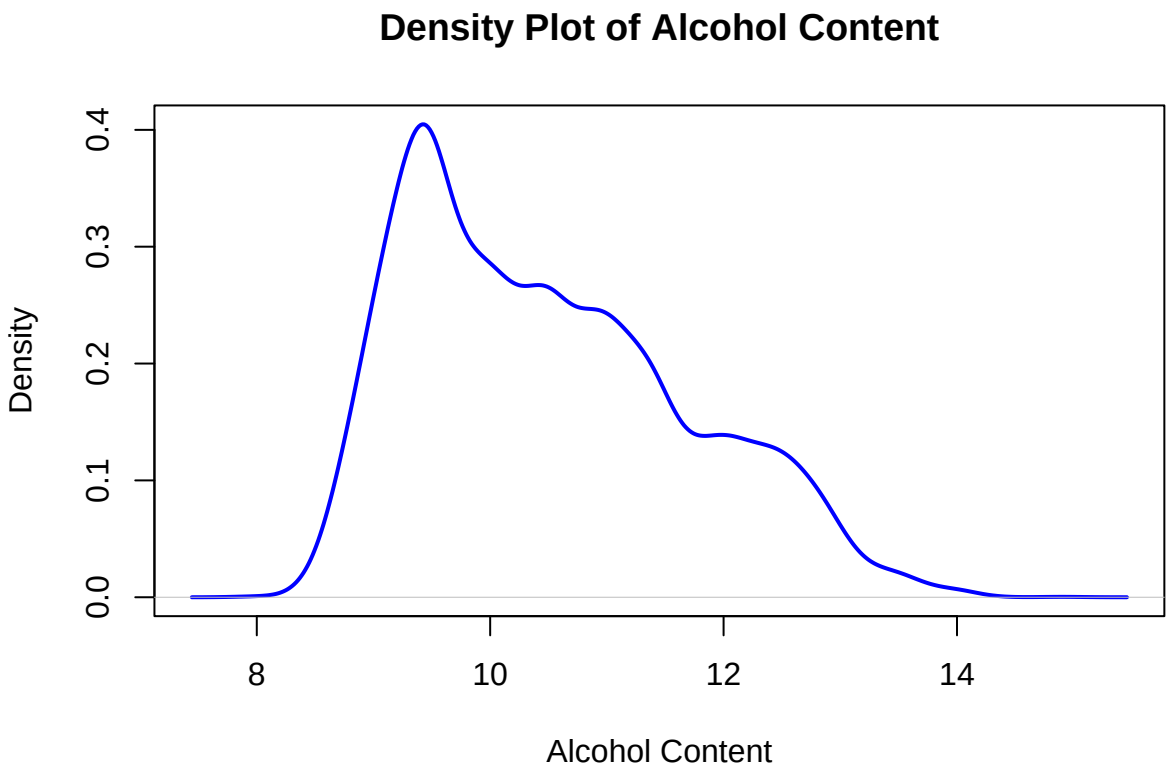


These box plots show that whitewine and redwine have similar distributions of alcohol content; however, redwine has upper outliters while whitewine does not have any outliers. Having this plot of the alcohol content by wine type can help us understand how important this binary variable will be when predicting alcohol percent.

```
hist(df$alcohol, breaks = 20, main = "Distribution of Alcohol Content",
     xlab = "Alcohol Content", col = "lightblue", border = "black")
```




```r
plot(density(df$alcohol, na.rm = TRUE), main = "Density Plot of Alcohol Content",
     xlab = "Alcohol Content", col = "blue", lwd = 2)
```



This histogram and density plot of alcohol content provide a clear visualization of its distribution across the wine samples. The plot highlights the central tendency and variability, showing the most common alcohol content levels and how they are distributed. These insights help us understand the overall characteristics of the dataset and identify any patterns or outliers in the alcohol content.

Model A - Part 2:

We will now start the process of creating of Model A, starting with breaking up the training and testing sets.

```
l <- nrow(df)

train_data <- df[1:round(l * 0.8, 0), ]
test_data <- df[(round(l * 0.8, 0) + 1):l, ]

dim(train_data)
```

```
## [1] 5198   12
```

```
dim(test_data)
```

```
## [1] 1299   12
```

Part A)

Here we will begin to construct the full model.

```r
full_model = lm(alcohol ~., data=df)
summary(full_model)
```

```
##
## Call:
## lm(formula = alcohol ~ ., data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5559 -0.2892 -0.0361  0.2549 15.6752
##
## Coefficients:
##                         Estimate Std. Error  t value Pr(>|t|)
## (Intercept)           6.757e+02  4.890e+00  138.179  < 2e-16 ***
## fixed.acidity         5.432e-01  8.474e-03   64.109  < 2e-16 ***
## volatile.acidity      6.502e-01  5.531e-02   11.756  < 2e-16 ***
## citric.acid           5.320e-01  5.437e-02    9.784  < 2e-16 ***
## residual.sugar        2.404e-01  2.776e-03   86.606  < 2e-16 ***
## chlorides            -1.013e+00  2.294e-01   -4.415 1.03e-05 ***
## free.sulfur.dioxide  -2.954e-03  5.252e-04   -5.625 1.93e-08 ***
## total.sulfur.dioxide -2.499e-04  2.224e-04   -1.124    0.261
## density              -6.827e+02  5.014e+00 -136.159  < 2e-16 ***
## pH                    2.721e+00  5.226e-02   52.058  < 2e-16 ***
## sulphates             1.095e+00  5.059e-02   21.645  < 2e-16 ***
## type                 -1.210e+00  3.598e-02  -33.645  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5037 on 6485 degrees of freedom
## Multiple R-squared:  0.822,  Adjusted R-squared:  0.8217
## F-statistic:  2722 on 11 and 6485 DF,  p-value: < 2.2e-16
```

From this model summary, we have one predictor with a p-value of more than 0.05; moreover, total.sulfur.dioxide should be removed from this model because it is not statistically significant.

```r
step_selected_aic <- step(full_model, direction="backward")
```

```
## Start:  AIC=-8899.42
## alcohol ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + type
##
##                        Df Sum of Sq    RSS     AIC
## - total.sulfur.dioxide  1       0.3 1645.5 -8900.2
## <none>                              1645.2 -8899.4
## - chlorides             1       4.9 1650.2 -8881.9
## - free.sulfur.dioxide   1       8.0 1653.2 -8869.8
## - citric.acid           1      24.3 1669.5 -8806.2
## - volatile.acidity      1      35.1 1680.3 -8764.4
## - sulphates             1     118.9 1764.1 -8448.2
## - type                  1     287.2 1932.4 -7856.1
## - pH                    1     687.5 2332.8 -6632.8
```

```
## - fixed.acidity         1    1042.7 2687.9 -5712.1
## - residual.sugar        1    1902.9 3548.1 -3908.2
## - density               1    4703.3 6348.5  -128.2
##
## Step:  AIC=-8900.15
## alcohol ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     chlorides + free.sulfur.dioxide + density + pH + sulphates +
##     type
##
##                       Df Sum of Sq    RSS     AIC
## <none>                             1645.5 -8900.2
## - chlorides            1       4.9 1650.4 -8882.9
## - free.sulfur.dioxide  1      15.3 1660.9 -8841.9
## - citric.acid          1      24.0 1669.5 -8808.2
## - volatile.acidity     1      34.9 1680.5 -8765.7
## - sulphates            1     118.6 1764.1 -8450.1
## - type                 1     444.2 2089.7 -7349.7
## - pH                   1     690.8 2336.3 -6624.9
## - fixed.acidity        1    1066.3 2711.8 -5656.5
## - residual.sugar       1    1944.5 3590.0 -3833.9
## - density              1    5277.2 6922.8   432.4
```

The step selection with AIC also resulted in us eliminating total.sulfur.dioxide because it resulted in a lower model AIC.

```
aic_full <- AIC(full_model)
aic_selected <- AIC(step_selected_aic)

cat("AIC of the full model:", aic_full, "\n")
```

```
## AIC of the full model: 9540.27
```

```
cat("AIC of the selected model:", aic_selected, "\n")
```

```
## AIC of the selected model: 9539.536
```

As we see, the AIC did not decrease significantly, leading us to believe we should run even more tests to confirm that total.sulfur.dioxide is truly statistically insignificant.

```
n <- nrow(df)
step_selected_bic <- step(full_model, direction="both", k=log(n))
```

```
## Start:  AIC=-8818.07
## alcohol ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + type
##
##                        Df Sum of Sq    RSS     AIC
## - total.sulfur.dioxide  1       0.3 1645.5 -8825.6
## <none>                              1645.2 -8818.1
## - chlorides             1       4.9 1650.2 -8807.4
```

```
## - free.sulfur.dioxide   1       8.0 1653.2 -8795.2
## - citric.acid           1      24.3 1669.5 -8731.6
## - volatile.acidity      1      35.1 1680.3 -8689.9
## - sulphates             1     118.9 1764.1 -8373.6
## - type                  1     287.2 1932.4 -7781.6
## - pH                    1     687.5 2332.8 -6558.3
## - fixed.acidity         1    1042.7 2687.9 -5637.5
## - residual.sugar        1    1902.9 3548.1 -3833.6
## - density               1    4703.3 6348.5   -53.6
##
## Step:  AIC=-8825.58
## alcohol ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     chlorides + free.sulfur.dioxide + density + pH + sulphates +
##     type
##
##                        Df Sum of Sq    RSS     AIC
## <none>                              1645.5 -8825.6
## + total.sulfur.dioxide  1       0.3 1645.2 -8818.1
## - chlorides             1       4.9 1650.4 -8815.2
## - free.sulfur.dioxide   1      15.3 1660.9 -8774.1
## - citric.acid           1      24.0 1669.5 -8740.4
## - volatile.acidity      1      34.9 1680.5 -8697.9
## - sulphates             1     118.6 1764.1 -8382.3
## - type                  1     444.2 2089.7 -7281.9
## - pH                    1     690.8 2336.3 -6557.2
## - fixed.acidity         1    1066.3 2711.8 -5588.7
## - residual.sugar        1    1944.5 3590.0 -3766.1
## - density               1    5277.2 6922.8   500.2
```

```
bic_full <- BIC(full_model)

bic_selected <- BIC(step_selected_bic)

cat("BIC of the full model:", bic_full, "\n")
```

```
## BIC of the full model: 9628.399
```

```
cat("BIC of the selected model:", bic_selected, "\n")
```

```
## BIC of the selected model: 9620.885
```

The step selection with bic also tells us to remove total.sulfur.dioxide as it improves the model and lowers the AIC.

Part B:

We begin the leaps and bounds algorithm by outputting the summary of the regsubsets selection.

```
regsubsets_selection=regsubsets(alcohol~. -total.sulfur.dioxide, data = df)
rs = summary(regsubsets_selection)
rs$which
```

```
##   (Intercept) fixed.acidity volatile.acidity citric.acid residual.sugar
```

```
## 1        TRUE         FALSE            FALSE       FALSE          FALSE
## 2        TRUE         FALSE            FALSE       FALSE          FALSE
## 3        TRUE         FALSE            FALSE       FALSE           TRUE
## 4        TRUE          TRUE            FALSE       FALSE           TRUE
## 5        TRUE          TRUE            FALSE       FALSE           TRUE
## 6        TRUE          TRUE            FALSE       FALSE           TRUE
## 7        TRUE          TRUE             TRUE       FALSE           TRUE
## 8        TRUE          TRUE             TRUE        TRUE           TRUE
##   chlorides free.sulfur.dioxide density    pH sulphates  type
## 1     FALSE               FALSE    TRUE FALSE     FALSE FALSE
## 2     FALSE               FALSE    TRUE FALSE     FALSE  TRUE
## 3     FALSE               FALSE    TRUE FALSE     FALSE  TRUE
## 4     FALSE               FALSE    TRUE FALSE     FALSE  TRUE
## 5     FALSE               FALSE    TRUE  TRUE     FALSE  TRUE
## 6     FALSE               FALSE    TRUE  TRUE      TRUE  TRUE
## 7     FALSE               FALSE    TRUE  TRUE      TRUE  TRUE
## 8     FALSE               FALSE    TRUE  TRUE      TRUE  TRUE
```

```
rs$adjr2
```

```
## [1] 0.4715379 0.5368158 0.6349620 0.7186763 0.8026839 0.8148444 0.8174193
## [8] 0.8194936
```

The best model is the 8th model.

```
rs$cp
```

```
## [1] 12752.65560 10374.75367  6800.94714  3753.57903   696.47990   254.85001
## [7]   162.12481    87.64078
```

The best model is the 8th model.

```
rs$bic
```

```
## [1]  -4127.126  -4975.952  -6515.261  -8199.923 -10496.622 -10902.121 -10985.329
## [8] -11051.786
```

The best model is the 8th model, making all three of the criterion agree that the best model is the 8th model. This model includes the predictors fixed.acidity, volatile.acidity, citric.acid, residual.sugar, density, pH, sulphates, and typewhitewine.

Part C:

```
summary(full_model)$r.sq
```

```
## [1] 0.8219645
```

This R-squared value represents the percentage of variation in the dependent variable, alcohol, that is explained by the predictors in the model. A value of 82.196% indicates that the model has a strong fit and is highly effective at explaining the variability in alcohol. This suggests that the predictors included in the model are capturing most of the important factors influencing alcohol content.

```r
pc_model <- lm(alcohol ~ ., data = train_data)
```

```r
rmse<-function(x,y) sqrt(mean((x-y)^2))
rmse(fitted(pc_model), train_data$alcohol)
```

```
## [1] 0.5306046
```

```r
rmse(predict(pc_model, test_data), test_data$alcohol)
```

```
## [1] 0.3824433
```

These values represent the average magnitide of error for predictions. Because our test RMSE is lower than the RMSE for the training data, we can conclude that the model generalizes well to unseen data and that it does not overfit the training data.

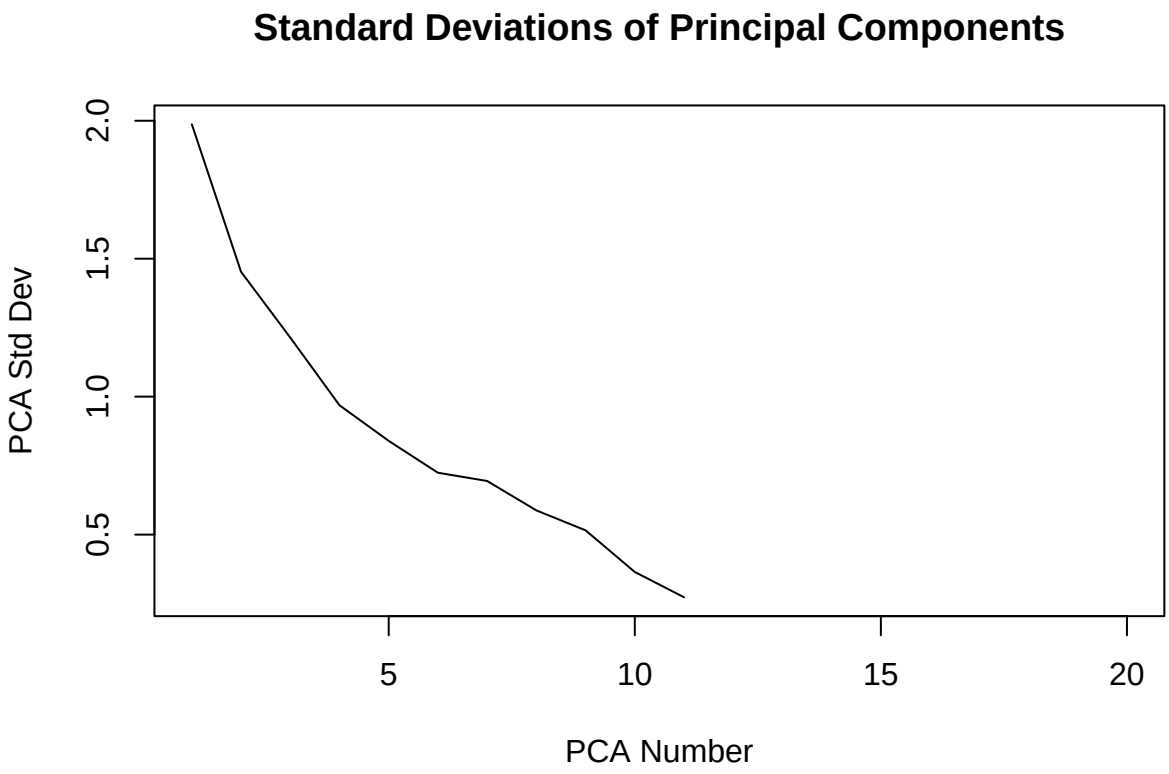
```r
num_predictors <- ncol(train_data) - 1
n <- min(8, num_predictors)
alcohol_pcr <-pcr(alcohol ~ ., scale=TRUE, data=train_data, ncomp=n)
summary(alcohol_pcr)
```

```
## Data:    X dimension: 5198 11
##  Y dimension: 5198 1
## Fit method: svdpc
## Number of components considered: 8
## TRAINING: % variance explained
##          1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X         35.888    55.07    68.46    77.00    83.41    88.17    92.56    95.70
## alcohol    0.186    19.35    34.87    35.45    36.24    36.96    36.96    39.56
```

This summary provides information about the Principal Component Regression (PCR) model trained on the dataset. The percent of variance explained for the predictors increases as the number of components increases, indicating that the first few components capture a large proportion of the variability in the predictors. For the target variable alcohol, the explained variance increases significantly after two components but plateaus after five, suggesting that additional components contribute little to improving the model's explanatory power for alcohol.

```r
alcohol_pca <- prcomp(train_data[ , -11], scale. = TRUE)

plot(alcohol_pca$sdev[1:20],
     ylab = "PCA Std Dev",
     xlab = "PCA Number",
     type = "l",
     main = "Standard Deviations of Principal Components")
```



This graph shows the standard deviations of the first 20 principal components obtained through PCA, representing the amount of variance explained by each component. The graph demonstrates a steep decline in standard deviations for the first 2-3 components, indicating that they capture most of the variance in the data. After component 3, the decline becomes more gradual, suggesting that additional components contribute progressively less to explaining the variance. This pattern implies that the first few components are the most significant for summarizing the data.

```
set.seed(135)
pcr.mse<-RMSEP(alcohol_pcr, newdata=test_data)
```

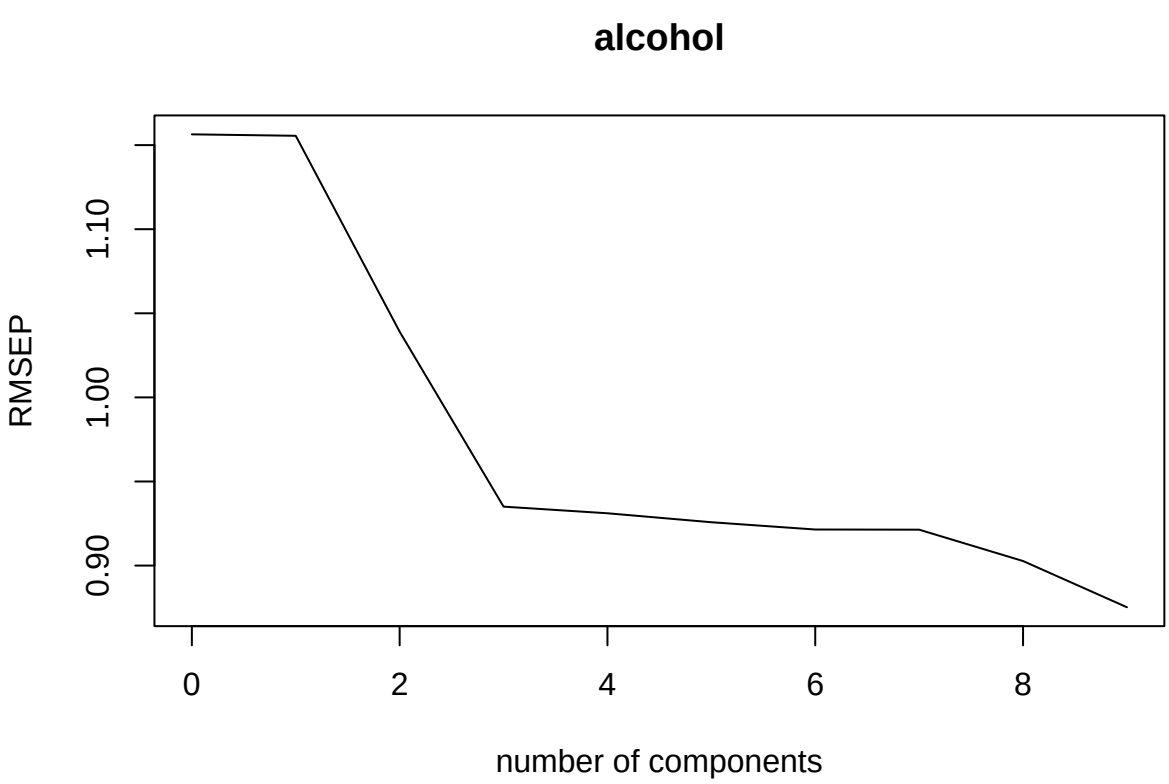
```
optimal_ncomp <- which.min(pcr.mse$val)
pcr.mse$val[optimal_ncomp]
```

```
## [1] 0.9142908
```

This value represents the minimum mean squared error achieved during the Principal Component Regression model evaluation, corresponding to the optimal number of components. This value indicates how well the model performs at its best configuration when the number of components is chosen to minimize prediction error, allowing us to better understand how useful this model is.

```
set.seed(234)
modpcrcv<-pcr(alcohol~., scale=TRUE, data=train_data, validation="CV", ncomp=optimal_ncomp)

pcrCV<-RMSEP(modpcrcv, estimate="CV")
plot(pcrCV)
```



This plot demonstrates the relationship between the number of principal components and the Root Mean Square Error of Prediction for the alcohol content model. The sharp decline in RMSEP up to approximately 3 components indicates that these components capture the majority of the variance in the data. Beyond this point, additional components yield minimal improvement, suggesting that around 3 components provide an optimal balance between model accuracy and complexity; however, the minimal improvement could be utilized within the model, leading us to believe further analysis is necessary.

```
alc_pred <- predict(modpcrcv, test_data, ncomp = optimal_ncomp)
rmse(alc_pred,test_data$alcohol)
```

```
## [1] 0.8670076
```

This RMSE value helps us to understand the average magnitude of error for predictions of the Principal Components Model on the test data. A value of 0.8670 shows us that the model does not over fit on the training data and explains the variation of the data well.

Part D:

```
ridge_model <- lm.ridge(alcohol ~ ., data = train_data, lambda = seq(0, 5e-8, length.out = 21))
```

```
best_lambda_index <- which.min(ridge_model$GCV)
best_lambda <- ridge_model$lambda[best_lambda_index]

cat("Best lambda:", best_lambda, "\n")
```

```
## Best lambda: 5e-08
```

```
cat("Minimum GCV:", min(ridge_model$GCV), "\n")
```

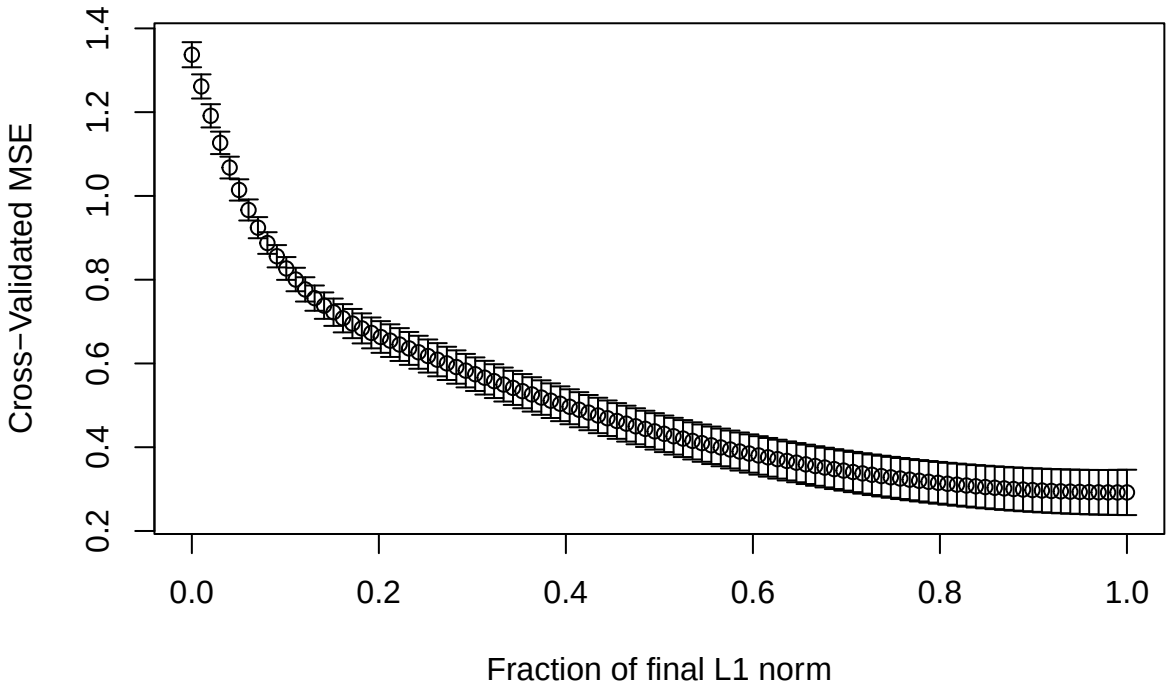
```
## Minimum GCV: 5.439335e-05
```

This Ridge Regression model determines the best value for the regularization parameter, lambda, by minimizing the Generalized Cross-Validation error. The results show that the optimal lambda is 5e-08, which balances model complexity and prediction accuracy by penalizing large coefficients. The corresponding minimum GCV value of 5.439335e-05 indicates a low prediction error, suggesting that the model generalizes well to unseen data.

```
train.y<-train_data$alcohol
train.x<-as.matrix(train_data[,-11])
```

```
alc_lasso<-lars(train.x,train.y)
```

```
set.seed(123)
cv.ml<-cv.lars(train.x,train.y)
```



This graph visualizes the cross-validated Mean Squared Error against the fraction of the L1-norm for a Lasso regression model. As the fraction of the L1-norm increases, the cross-validated MSE decreases, indicating an improvement in prediction accuracy. However, the rate of improvement slows down significantly after a certain point, suggesting diminishing returns from including additional predictors. The error bars represent variability in the cross-validation process, and the model with the minimum MSE is the optimal choice, balancing prediction accuracy and complexity. This plot highlights the trade-off between model sparsity and predictive performance.

```r
which.min(cv.ml$cv)
```

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

```r
svm<-cv.ml$index[which.min(cv.ml$cv)]
svm
```

```
## [1] 0.989899
```

```r
testx<-as.matrix(test_data[,-11])

predlasso<-predict(alc_lasso, testx, s=svm, mode="fraction")
rmse(test_data$alcohol, predlasso$fit)
```

```
## [1] 0.3841553
```

These values show that the optimal fraction of the L1-norm for the Lasso model, as determined by cross-validation, is approximately 0.989899, corresponding to the 99th index. This fraction represents the level of penalization that minimizes the cross-validated Mean Squared Error (MSE). Using this optimal model, the Root Mean Squared Error (RMSE) on the test data is 0.3841553, indicating good predictive performance and low average error for unseen data. This demonstrates that the Lasso model effectively balances sparsity and prediction accuracy.

Model B - Part 3:

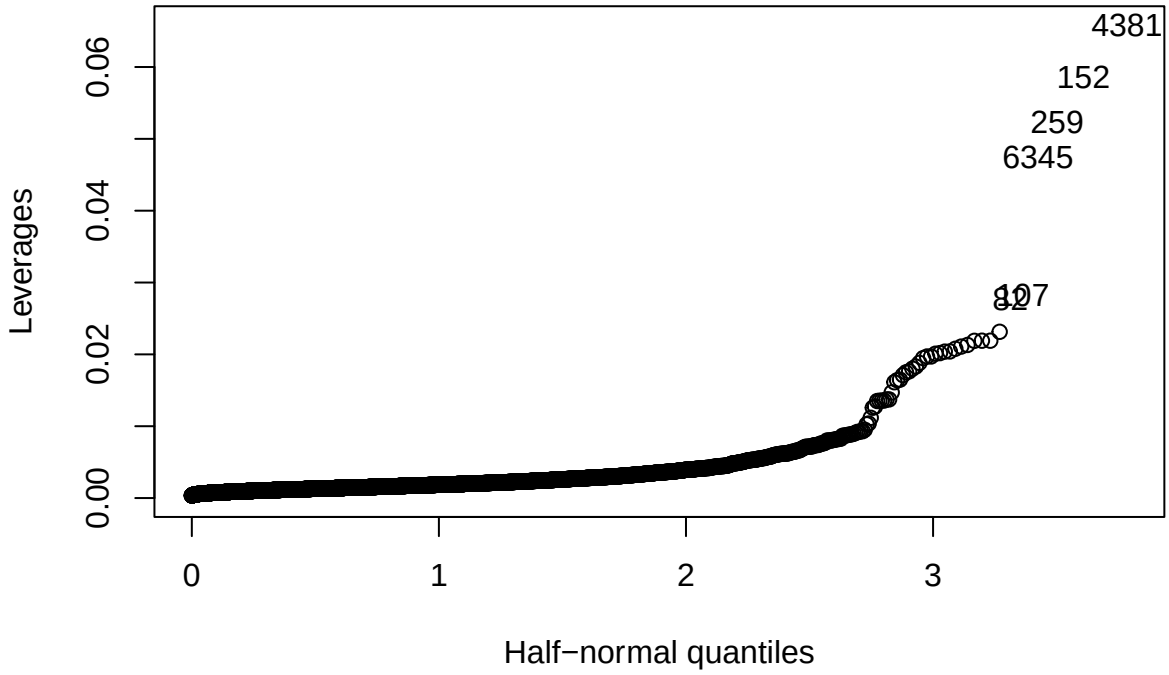
Part A - Checking for Unusual Observations

```r
df2 <- df[, c(-1, -11)]

alcohol_leverages <- lm.influence(full_model)$hat
head(alcohol_leverages)
```

```
##           1           2           3           4           5           6
## 0.001714065 0.002496074 0.001505824 0.002369147 0.001714065 0.001639561
```

```
halfnorm(alcohol_leverages, nlab=6, labs=as.character(1:length(alcohol_leverages)), ylab="Leverages")
```



This half-normal plot of leverages highlights a few high-leverage observations, such as 4381, 152, 259, and 6345, which deviate significantly from the general trend. The majority of points remain close to the lower leverage values, indicating that most observations have minimal influence on the model. However, these extreme leverage points warrant further investigation to assess their potential impact on the model's performance. Additional diagnostics, like Cook's distance, can confirm whether these points are influential and whether remedial measures are necessary.

```
n = dim(df2)[1];
n
```

```
## [1] 6497
```

```
p = length(variable.names(full_model));
p
```

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

```
alcohol_leverages_high = alcohol_leverages[alcohol_leverages>2*p/n]
(num_high_leverage <- length(alcohol_leverages_high))
```

```
## [1] 337
```

```r
(proportion_high_leverage <- num_high_leverage / n)
```

```
## [1] 0.05187009
```

```r
IQR_y <- IQR(df$alcohol)

Q1_y <- quantile(df$alcohol, 0.25)
Q3_y <- quantile(df$alcohol, 0.75)

lower_lim_y <- Q1_y - IQR_y
upper_lim_y <- Q3_y + IQR_y

(vector_lim_y <- c(lower_lim_y, upper_lim_y))
```

```
##  25%  75%
##  7.7 13.1
```

Here we are able to analyze the IQR which suggests that 50% of the alcohol percent distribution is captured between 7.7, Q1, and 13.1, Q3.

```r
high_leverage_threshold <- 2 * p / n
df_highlev <- df[alcohol_leverages > high_leverage_threshold, ]


df_highlev_lower <- df_highlev[df_highlev$alcohol < vector_lim_y[1], ]
df_highlev_upper <- df_highlev[df_highlev$alcohol > vector_lim_y[2], ]


df_highlev_bad <- rbind(df_highlev_lower, df_highlev_upper)


cat("Number of high leverage points:", nrow(df_highlev), "\n")
```

```
## Number of high leverage points: 337
```

```r
cat("Number of 'bad' high leverage points:", nrow(df_highlev_bad), "\n")
```

```
## Number of 'bad' high leverage points: 21
```

```r
df_highlev_bad
```

```
##      fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 143            5.2            0.340        0.00            1.8     0.050
## 145            5.2            0.340        0.00            1.8     0.050
## 379           11.4            0.625        0.66            6.2     0.088
## 653           15.9            0.360        0.65            7.5     0.096
## 822            4.9            0.420        0.00            2.1     0.048
## 1115           5.0            0.400        0.50            4.3     0.046
## 1229           5.1            0.420        0.00            1.8     0.044
## 1270           5.5            0.490        0.03            1.8     0.044
```

```
## 1271           5.0            0.380        0.01            1.6     0.048
## 1476           5.3            0.470        0.11            2.2     0.048
## 1478           5.3            0.470        0.11            2.2     0.048
## 2986           5.6            0.490        0.13            4.5     0.039
## 2994           5.6            0.490        0.13            4.5     0.039
## 4194           5.4            0.500        0.13            5.0     0.028
## 5310           4.7            0.670        0.09            1.0     0.020
## 5335           6.1            0.220        0.46            1.8     0.160
## 5373           5.0            0.610        0.12            1.3     0.009
## 5501           4.8            0.650        0.12            1.1     0.013
## 6103           5.8            0.610        0.01            8.4     0.041
## 6392           4.7            0.785        0.00            3.4     0.036
## 6415           6.2            0.760        0.01            3.2     0.041
##      free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates  alcohol
## 143                   27                   63 0.99160 3.68      0.79 14.00000
## 145                   27                   63 0.99160 3.68      0.79 14.00000
## 379                    6                   24 0.99880 3.11      0.99 13.30000
## 653                   22                   71 0.99760 2.98      0.84 14.90000
## 822                   16                   42 0.99154 3.71      0.74 14.00000
## 1115                  29                   80 0.99020 3.49      0.66 13.60000
## 1229                  18                   88 0.99157 3.68      0.73 13.60000
## 1270                  28                   87 0.99080 3.50      0.82 14.00000
## 1271                  26                   60 0.99084 3.70      0.75 14.00000
## 1476                  16                   89 0.99182 3.54      0.88 13.56667
## 1478                  16                   89 0.99182 3.54      0.88 13.60000
## 2986                  17                  116 0.99070 3.42      0.90 13.70000
## 2994                  17                  116 0.99070 3.42      0.90 13.70000
## 4194                  12                  107 0.99079 3.48      0.88 13.50000
## 5310                   5                    9 0.98722 3.30      0.34 13.60000
## 5335                  34                   74 0.98840 3.19      0.33 13.40000
## 5373                  65                  100 0.98740 3.26      0.37 13.50000
## 5501                   4                   10 0.99246 3.32      0.36 13.50000
## 6103                  31                  104 0.99090 3.26      0.72 14.05000
## 6392                  23                  134 0.98981 3.53      0.92 13.80000
## 6415                  18                  120 0.99026 3.20      0.94 13.70000
##      type
## 143     0
## 145     0
## 379     0
## 653     0
## 822     0
## 1115    0
## 1229    0
## 1270    0
## 1271    0
## 1476    0
## 1478    0
## 2986    1
## 2994    1
## 4194    1
## 5310    1
## 5335    1
## 5373    1
## 5501    1
```

```
## 6103    1
## 6392    1
## 6415    1
```

All of the observations in this data frame are bad high-leverage points.

To identify outliers, we use the Bonferroni test.

```
alcohol_resid = rstudent(full_model);

bonferroni_cv = qt(.05/(2*n), n-p-1)
bonferroni_cv
```

```
## [1] -4.477102
```

```
alcohol_resid_sorted <- sort(abs(alcohol_resid), decreasing = TRUE)[1:10]

print(alcohol_resid_sorted)
```

```
##      4381      5501      3126      3253      3263       560       565       396
## 35.126770  7.603225  7.135483  5.997047  5.997047  5.707284  5.707284  5.534902
##       354       494
##  5.361092  5.117044
```

```
alcohol_outliers = alcohol_resid_sorted[abs(alcohol_resid_sorted) > abs(bonferroni_cv)]
print(alcohol_outliers)
```

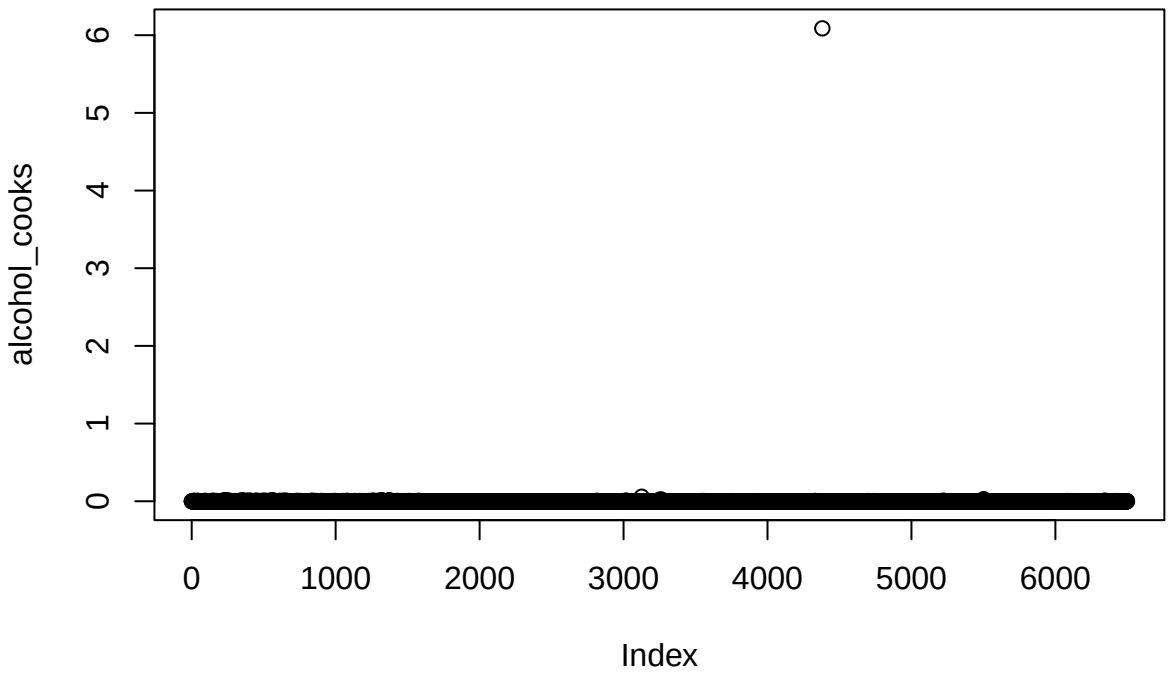
```
##      4381      5501      3126      3253      3263       560       565       396
## 35.126770  7.603225  7.135483  5.997047  5.997047  5.707284  5.707284  5.534902
##       354       494
##  5.361092  5.117044
```

To check for high influential points, we will use Cook's distance with the cooks.distance R function:

```
alcohol_cooks = cooks.distance(full_model)
sort(alcohol_cooks, decreasing = TRUE)[1:10]
```

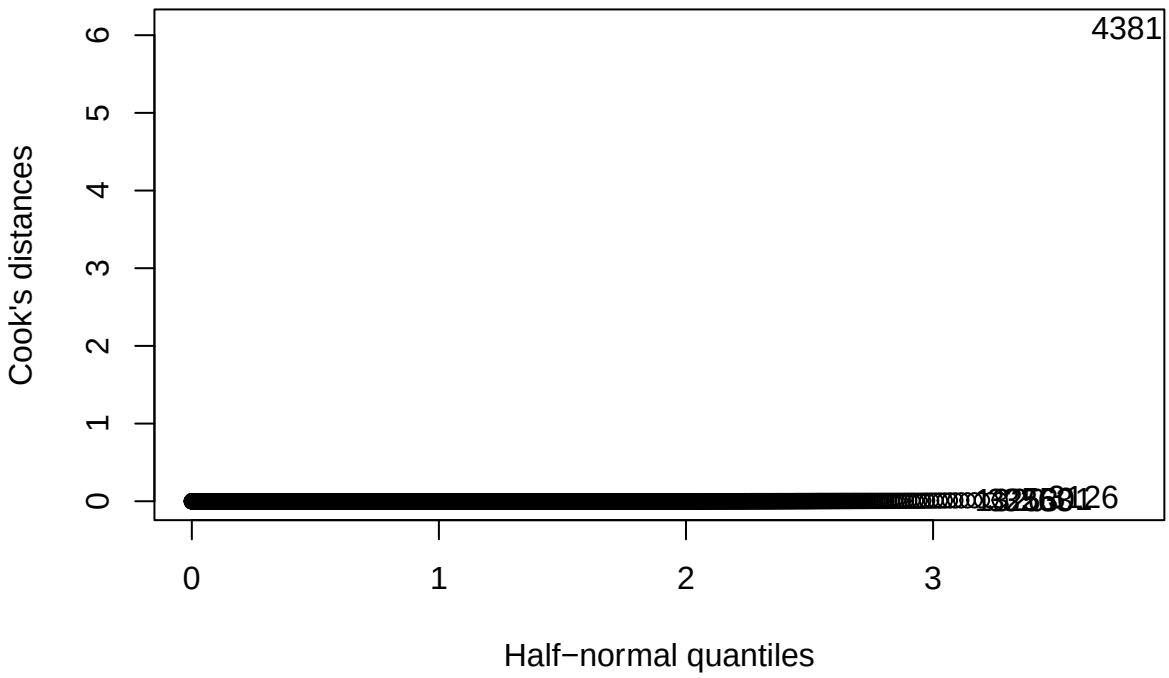
```
##       4381       3126       5501       3253       3263       1320        227
## 6.08814490 0.05794370 0.02652596 0.02141984 0.02141984 0.01439669 0.01437845
##        354       1371       1373
## 0.01243658 0.01195241 0.01195241
```

```
plot(alcohol_cooks)
```



Based on the rule-of-thumb, Cook's distance more than or equal to 1, observation 4381 is identified as a high-influential point with a Cook's distance of approximately 6.08. The remaining observations have much smaller Cook's distances, indicating they do not significantly influence the model's fit. An index plot and half-normal plot confirm that observation 4381 stands out compared to others. Further investigation of this observation is recommended to determine its validity and its impact on the model.

```
halfnorm(alcohol_cooks, 6, labs=as.character(1:length(alcohol_cooks)), ylab="Cook's distances")
```



Due to the lack in concision of the plots, we have chosen to not utilize these results within the conclusion of our case study.

Part B - checking diagnostics and model assumptions

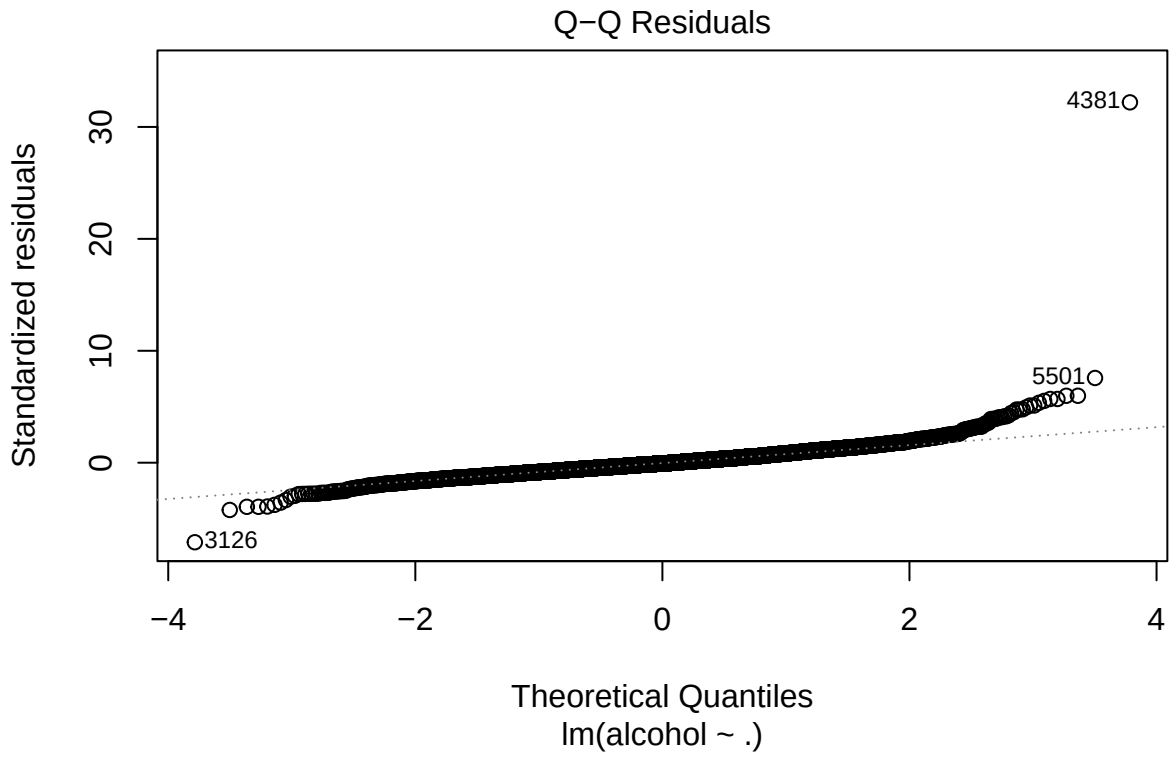
We will first check for homoscedasticity by running a studentized Breusch-Pagan test.

```
bptest(full_model)
```

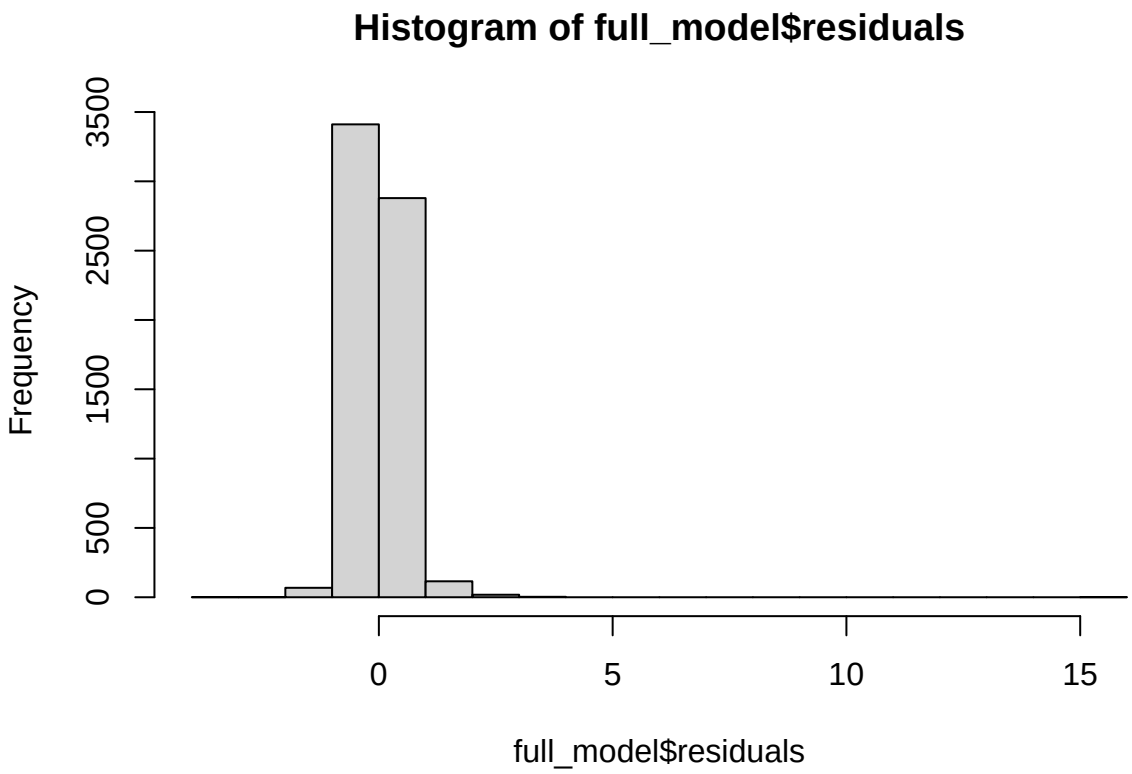
```
##
##  studentized Breusch-Pagan test
##
## data:  full_model
## BP = 475.47, df = 11, p-value < 2.2e-16
```

The p-value of 2.2e-16 is lower than the significance level (alpha = 0.05). Therefore, we choose to reject the null hypotheses of homoscedasticity and conclude that the constant variance assumption is not satisfied. This result suggests the presence of heteroscedasticity, and further diagnostic tests or corrective measures should be considered to address this issue.

```
plot(full_model, which=2)
```




```r
hist(full_model$residuals)
```



We will complete a Kolmogorov-Smirnov test for the normality assumption.

```
ks.test(full_model$residuals, "pnorm", mean = mean(full_model$residuals), sd = sd(full_model$residuals))
```

```
## Warning in ks.test.default(full_model$residuals, "pnorm", mean =
## mean(full_model$residuals), : ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  full_model$residuals
## D = 0.061401, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

The p-value of 2.2e-16 is lower than the significance level (alpha = 0.05). Therefore, we choose to reject the null hypotheses of normality. This indicates that the residuals are not normally distributed.

We will now check the linearity assumption using residuals.

```
summary(full_model)
```

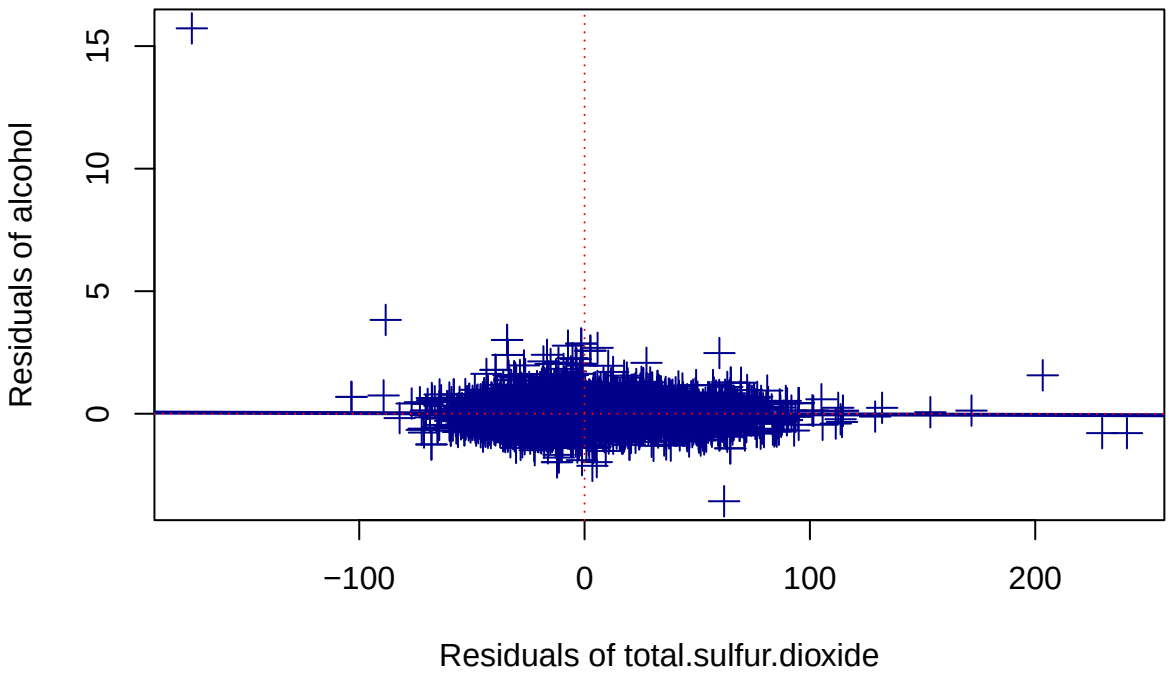
```
##
## Call:
## lm(formula = alcohol ~ ., data = df)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5559 -0.2892 -0.0361  0.2549 15.6752
##
## Coefficients:
##                        Estimate Std. Error  t value Pr(>|t|)
## (Intercept)           6.757e+02  4.890e+00  138.179  < 2e-16 ***
## fixed.acidity         5.432e-01  8.474e-03   64.109  < 2e-16 ***
## volatile.acidity      6.502e-01  5.531e-02   11.756  < 2e-16 ***
## citric.acid           5.320e-01  5.437e-02    9.784  < 2e-16 ***
## residual.sugar        2.404e-01  2.776e-03   86.606  < 2e-16 ***
## chlorides            -1.013e+00  2.294e-01   -4.415 1.03e-05 ***
## free.sulfur.dioxide  -2.954e-03  5.252e-04   -5.625 1.93e-08 ***
## total.sulfur.dioxide -2.499e-04  2.224e-04   -1.124    0.261
## density              -6.827e+02  5.014e+00 -136.159  < 2e-16 ***
## pH                    2.721e+00  5.226e-02   52.058  < 2e-16 ***
## sulphates             1.095e+00  5.059e-02   21.645  < 2e-16 ***
## type                 -1.210e+00  3.598e-02  -33.645  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5037 on 6485 degrees of freedom
## Multiple R-squared:  0.822,  Adjusted R-squared:  0.8217
## F-statistic:  2722 on 11 and 6485 DF,  p-value: < 2.2e-16
```

The only insignificant predictor seems to be total.sulfur.dioxide. We will now analyze the reduced model accordingly.

```
model_no_tsd <- lm(alcohol ~ . - total.sulfur.dioxide, data = df)
y.TSD <- residuals(model_no_tsd)
x.TSD = lm(total.sulfur.dioxide ~ ., data = df[,-c(2,11)])$residuals
```

```
plot(x.TSD, y.TSD,
     xlab = "Residuals of total.sulfur.dioxide",
     ylab = "Residuals of alcohol",
     col = "Darkblue", pch = 3, cex = 1.5)
abline(lm(y.TSD ~ x.TSD), col = "Darkblue", lwd = 2)
abline(v = 0, col = "red", lty = 3)
abline(h = 0, col = "red", lty = 3)
```



The residual plot shows a clear pattern, with residuals forming a funnel-like shape and spreading unevenly around the horizontal line. This suggests that the linearity assumption is not fully met, and there may also be heteroscedasticity present, as stated earlier.

Part C - Remedial Measures

```
residual_sd <- abs(residuals(full_model))
weights <- 1 / (residual_sd^2)
wls_weights <- lm(alcohol ~ . - total.sulfur.dioxide,
                  data = df,
                  weights = weights)
summary(wls_weights)
```

```
##
## Call:
## lm(formula = alcohol ~ . - total.sulfur.dioxide, data = df, weights = weights)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -22.3722  -0.9955  -0.8514   0.9988  14.1403
##
## Coefficients:
##                       Estimate Std. Error  t value Pr(>|t|)
## (Intercept)          6.780e+02  2.244e-01  3021.56   <2e-16 ***
## fixed.acidity        5.440e-01  3.479e-04  1563.82   <2e-16 ***
## volatile.acidity     6.041e-01  2.336e-03   258.65   <2e-16 ***
## citric.acid          5.110e-01  3.254e-03   157.03   <2e-16 ***
```

```
## residual.sugar       2.401e-01  1.040e-04  2309.79   <2e-16 ***
## chlorides           -1.008e+00  1.391e-02   -72.47   <2e-16 ***
## free.sulfur.dioxide -3.478e-03  1.664e-05  -208.94   <2e-16 ***
## density             -6.850e+02  2.306e-01 -2970.81   <2e-16 ***
## pH                   2.726e+00  1.501e-03  1816.29   <2e-16 ***
## sulphates            1.098e+00  1.642e-03   668.39   <2e-16 ***
## type                -1.243e+00  9.502e-04 -1308.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.101 on 6486 degrees of freedom
## Multiple R-squared:  0.9998, Adjusted R-squared:  0.9998
## F-statistic: 3.991e+06 on 10 and 6486 DF,  p-value: < 2.2e-16
```

The first observation from the Weighted Least Squares (WLS) model is its exceptionally high r-squared value, indicating that 99.98% of the variation in the data is explained by the model. This suggests an excellent fit, with the model accounting for nearly all variability in the response variable.

Let us observe what the results would be if we run a standard regression with no weights:

```
wls_model <- lm(alcohol ~ . - total.sulfur.dioxide, data = df)
summary(wls_model)
```

```
##
## Call:
## lm(formula = alcohol ~ . - total.sulfur.dioxide, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5706 -0.2897 -0.0371  0.2535 15.7248
##
## Coefficients:
##                       Estimate Std. Error  t value Pr(>|t|)
## (Intercept)          6.775e+02  4.623e+00  146.563  < 2e-16 ***
## fixed.acidity        5.445e-01  8.399e-03   64.829  < 2e-16 ***
## volatile.acidity     6.396e-01  5.451e-02   11.735  < 2e-16 ***
## citric.acid          5.263e-01  5.413e-02    9.722  < 2e-16 ***
## residual.sugar       2.409e-01  2.751e-03   87.546  < 2e-16 ***
## chlorides           -1.005e+00  2.293e-01   -4.382 1.19e-05 ***
## free.sulfur.dioxide -3.302e-03  4.246e-04   -7.776 8.67e-15 ***
## density             -6.845e+02  4.746e+00 -144.224  < 2e-16 ***
## pH                   2.724e+00  5.220e-02   52.179  < 2e-16 ***
## sulphates            1.091e+00  5.046e-02   21.620  < 2e-16 ***
## type                -1.234e+00  2.948e-02  -41.842  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5037 on 6486 degrees of freedom
## Multiple R-squared:  0.8219, Adjusted R-squared:  0.8217
## F-statistic:  2994 on 10 and 6486 DF,  p-value: < 2.2e-16
```

As we can observe, there is a significant drop in the r-squared value, indicating that the weighted model is a much better model for this data.

In addition to the WLS numerical summaries, we can also perform a lack-of-fit test.

```r
df_chisq <- df.residual(wls_weights)
sigma2_wls <- summary(wls_model)$sigma^2
1 - pchisq(sigma2_wls * df_chisq, df_chisq)
```

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

As you can see, the p-value is far outside of the rejection region of 0.05, which means that we fail to reject the null and conclude that there is not a lack of fit.

Because the normality assumption is questionable, we will use a permutation test to check that our predictors are statistically significant.

```r
n.iter <- 200
fstats <- numeric(n.iter)

observed_fstat <- summary(wls_weights)$fstatistic[1]

for (i in 1:n.iter) {
  newdata <- df

  newdata$alcohol <- sample(df$alcohol)

  wls_perm <- lm(alcohol ~ fixed.acidity + volatile.acidity + citric.acid +
                           residual.sugar + chlorides + free.sulfur.dioxide + density + pH + sulphates
                 data = newdata,
                 weights = weights)

  fstats[i] <- summary(wls_perm)$fstatistic[1]
}

p_value <- sum(fstats >= observed_fstat) / n.iter

cat("P-value from permutation test:", p_value, "\n")
```

```
## P-value from permutation test: 0
```

When running the permutation test with the statistically significant predictors, the p-value is approximately 0, which means we reject the null hypothesis that the predictors have no effect. This result provides strong evidence that the predictors in the model explain a significant portion of the variability in the response variable. Consequently, we choose to work with the wls model as it captures the relationships in the data more effectively compared to the full model.

In the case study, we began with the initial subsets selection model. This model included predictors that were selected based on statistical significance, using criteria such as adjusted r-squared, Mallows' CP, and BIC. However, we identified shortcomings, such as residual diagnostics issues and potential violations of model assumptions, particularly linearity, homoscedasticity, and normality. To address these issues, we explored alternative models.

We then moved to the Weighted Least Squares model, which uses weights to account for heteroscedasticity. The weights were computed as the inverse of the squared residuals from the reduced model, giving less influence to observations with higher variance. This approach significantly improved the model fit, as evidenced by the extremely high r-squared value, 99.98%, which shows that the WLS model explains almost all the variation in the response variable.

To ensure the correctness and robustness of the WLS model, we conducted a series of diagnostic tests throughout the analysis: - Permutation tests confirmed the statistical significance of the predictors. - Cook's distance analysis checked for influential points. - Residual diagnostics ensured that assumptions of linearity and homoscedasticity were addressed. - Chi-squared tests validated the lack of fit, supporting the choice of the WLS model.

In summary, the final WLS model addressed the shortcomings of the initial subsets model by correcting for heteroscedasticity, improving explanatory power, and passing key diagnostic tests. Through this process, we arrived at a robust model that accurately explains the relationship between the predictors and the response variable, alcohol content.

Part D - ANOVA Test

Let us now use an ANOVA test to analyze the results of all three models conducted in this case study, those being the full model, subsets selected model (reduced_model), and the WLS model.

```
reduced_model <- lm(alcohol ~ fixed.acidity + volatile.acidity + citric.acid +
                              residual.sugar +
                              density + pH + sulphates + type, data = df)
summary(full_model)
```

```
##
## Call:
## lm(formula = alcohol ~ ., data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5559 -0.2892 -0.0361  0.2549 15.6752
##
## Coefficients:
##                         Estimate Std. Error  t value Pr(>|t|)
## (Intercept)           6.757e+02  4.890e+00  138.179  < 2e-16 ***
## fixed.acidity         5.432e-01  8.474e-03   64.109  < 2e-16 ***
## volatile.acidity      6.502e-01  5.531e-02   11.756  < 2e-16 ***
## citric.acid           5.320e-01  5.437e-02    9.784  < 2e-16 ***
## residual.sugar        2.404e-01  2.776e-03   86.606  < 2e-16 ***
## chlorides            -1.013e+00  2.294e-01   -4.415 1.03e-05 ***
## free.sulfur.dioxide  -2.954e-03  5.252e-04   -5.625 1.93e-08 ***
## total.sulfur.dioxide -2.499e-04  2.224e-04   -1.124    0.261
## density              -6.827e+02  5.014e+00 -136.159  < 2e-16 ***
## pH                    2.721e+00  5.226e-02   52.058  < 2e-16 ***
## sulphates             1.095e+00  5.059e-02   21.645  < 2e-16 ***
## type                 -1.210e+00  3.598e-02  -33.645  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5037 on 6485 degrees of freedom
## Multiple R-squared:  0.822,  Adjusted R-squared:  0.8217
## F-statistic:  2722 on 11 and 6485 DF,  p-value: < 2.2e-16
```

```
summary(reduced_model)
```

```
##
## Call:
```

```
## lm(formula = alcohol ~ fixed.acidity + volatile.acidity + citric.acid +
##     residual.sugar + density + pH + sulphates + type, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6799 -0.2916 -0.0356  0.2441 16.1511
##
## Coefficients:
##                    Estimate Std. Error  t value Pr(>|t|)
## (Intercept)       6.870e+02  4.425e+00  155.252   <2e-16 ***
## fixed.acidity     5.617e-01  8.155e-03   68.885   <2e-16 ***
## volatile.acidity  6.425e-01  5.421e-02   11.852   <2e-16 ***
## citric.acid       4.668e-01  5.370e-02    8.693   <2e-16 ***
## residual.sugar    2.422e-01  2.674e-03   90.554   <2e-16 ***
## density          -6.945e+02  4.532e+00 -153.249   <2e-16 ***
## pH                2.782e+00  5.105e-02   54.508   <2e-16 ***
## sulphates         1.040e+00  5.002e-02   20.794   <2e-16 ***
## type             -1.260e+00  2.863e-02  -43.992   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5067 on 6488 degrees of freedom
## Multiple R-squared:  0.8197, Adjusted R-squared:  0.8195
## F-statistic:  3687 on 8 and 6488 DF,  p-value: < 2.2e-16
```

```
summary(wls_weights)
```

```
##
## Call:
## lm(formula = alcohol ~ . - total.sulfur.dioxide, data = df, weights = weights)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -22.3722  -0.9955  -0.8514   0.9988  14.1403
##
## Coefficients:
##                        Estimate Std. Error  t value Pr(>|t|)
## (Intercept)           6.780e+02  2.244e-01  3021.56   <2e-16 ***
## fixed.acidity         5.440e-01  3.479e-04  1563.82   <2e-16 ***
## volatile.acidity      6.041e-01  2.336e-03   258.65   <2e-16 ***
## citric.acid           5.110e-01  3.254e-03   157.03   <2e-16 ***
## residual.sugar        2.401e-01  1.040e-04  2309.79   <2e-16 ***
## chlorides            -1.008e+00  1.391e-02   -72.47   <2e-16 ***
## free.sulfur.dioxide -3.478e-03  1.664e-05  -208.94   <2e-16 ***
## density              -6.850e+02  2.306e-01 -2970.81   <2e-16 ***
## pH                    2.726e+00  1.501e-03  1816.29   <2e-16 ***
## sulphates             1.098e+00  1.642e-03   668.39   <2e-16 ***
## type                 -1.243e+00  9.502e-04 -1308.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.101 on 6486 degrees of freedom
## Multiple R-squared:  0.9998, Adjusted R-squared:  0.9998
## F-statistic: 3.991e+06 on 10 and 6486 DF,  p-value: < 2.2e-16
```

```r
anova(reduced_model, wls_model, full_model)
```

```
## Analysis of Variance Table
##
## Model 1: alcohol ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     density + pH + sulphates + type
## Model 2: alcohol ~ (fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + type) - total.sulfur.dioxide
## Model 3: alcohol ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + type
##   Res.Df    RSS Df Sum of Sq       F Pr(>F)
## 1   6488 1666.0
## 2   6486 1645.5  2   20.4591 40.3220 <2e-16 ***
## 3   6485 1645.2  1    0.3205  1.2634 0.2611
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
cat("AIC of subsets selection model:", AIC(reduced_model), "\n")
```

```
## AIC of subsets selection model: 9615.816
```

```r
cat("AIC of WLS model:", AIC(wls_weights), "\n")
```

```
## AIC of WLS model: 117.8112
```

```r
cat("AIC of full model:", AIC(full_model), "\n")
```

```
## AIC of full model: 9540.27
```

Model A, the subsets selection model, focuses on selecting a subset of predictors based on statistical criteria such as adjusted r-squared, BIC, and Mallows' CP, aiming for a balance between simplicity and explanatory power. It retains only the most significant predictors, making the model more interpretable but potentially prone to issues like heteroscedasticity.

Model B, the Weighted Least Squares model, begins with the full model, assessing predictors based on their p-values and statistical significance. Unlike Model A, which uses a selection process to identify a subset of predictors, Model B retains all predictors initially and focuses on addressing potential issues with model assumptions. Diagnostics such as residual plots, normality tests, and homoscedasticity checks are performed to identify departures from assumptions, such as non-constant variance.

The key difference lies in the treatment of residual variance: Model A assumes constant variance, whereas Model B explicitly accounts for heteroscedasticity through weights. Despite the added complexity, Model B provides a more robust fit by addressing diagnostic issues identified in Model