Final Project Analysis

Julie Hwang

2025-05-15

Instructions

You're almost done with the semester! Take a second to congratulate yourself on getting here. As a reminder, this final project is simply an (imperfect) way of measuring what you have learned throughout the semester. So take a deep breath and do your best, but also remember that it doesn't determine your value as a human being.

The exam is split into 4 sections: Module 1, 2 and 3 (6 questions), Modules 4 and 5 (3 questions), Module 6 (2 questions) and the final project. Most of the questions on this exam are short answers. You don't need to write out an overly long response (a sentence or so for each part of the question should be fine), but you should be specific in explaining your response. For example, if there is a question about whether the assumptions are reasonable. You shouldn't just say "from the plot we can see that the linearity assumption is (or is not) reasonable," but instead you should explain specifically why the plot leads you to believe the linearity assumption is (or is not) reasonable.

The exam is open notes so you **can** use any of the material or any of the notes you have taken throughout the class. You **cannot** discuss the exam (while it is in progress) with anyone else. You also **cannot** use any generative AI tools. Submissions will be sent by e-mail to **nbb45@cornell.edu** before **May 14th 11:59pm**.

Final Project (30 pts)

```
#install.packages("lmtest")
```

Dataset Selection and Exploratory Data Analysis

```
wine_data <- read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequali
#summary statistics
summary(wine_data)
```

```
##  fixed.acidity   volatile.acidity  citric.acid    residual.sugar
##  Min.   : 4.60   Min.   :0.1200   Min.   :0.000   Min.   : 0.900
##  1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090   1st Qu.: 1.900
##  Median : 7.90   Median :0.5200   Median :0.260   Median : 2.200
##  Mean   : 8.32   Mean   :0.5278   Mean   :0.271   Mean   : 2.539
##  3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420   3rd Qu.: 2.600
##  Max.   :15.90   Max.   :1.5800   Max.   :1.000   Max.   :15.500
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.01200   Min.   : 1.00       Min.   :  6.00       Min.   :0.9901
##  1st Qu.:0.07000   1st Qu.: 7.00       1st Qu.: 22.00       1st Qu.:0.9956
##  Median :0.07900   Median :14.00       Median : 38.00       Median :0.9968
##  Mean   :0.08747   Mean   :15.87       Mean   : 46.47       Mean   :0.9967
##  3rd Qu.:0.09000   3rd Qu.:21.00       3rd Qu.: 62.00       3rd Qu.:0.9978
##  Max.   :0.61100   Max.   :72.00       Max.   :289.00       Max.   :1.0037
##        pH          sulphates         alcohol         quality
##  Min.   :2.740   Min.   :0.3300   Min.   : 8.40   Min.   :3.000
##  1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50   1st Qu.:5.000
##  Median :3.310   Median :0.6200   Median :10.20   Median :6.000
##  Mean   :3.311   Mean   :0.6581   Mean   :10.42   Mean   :5.636
##  3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10   3rd Qu.:6.000
##  Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :8.000
```

```
#checking data structure
str(wine_data)
```

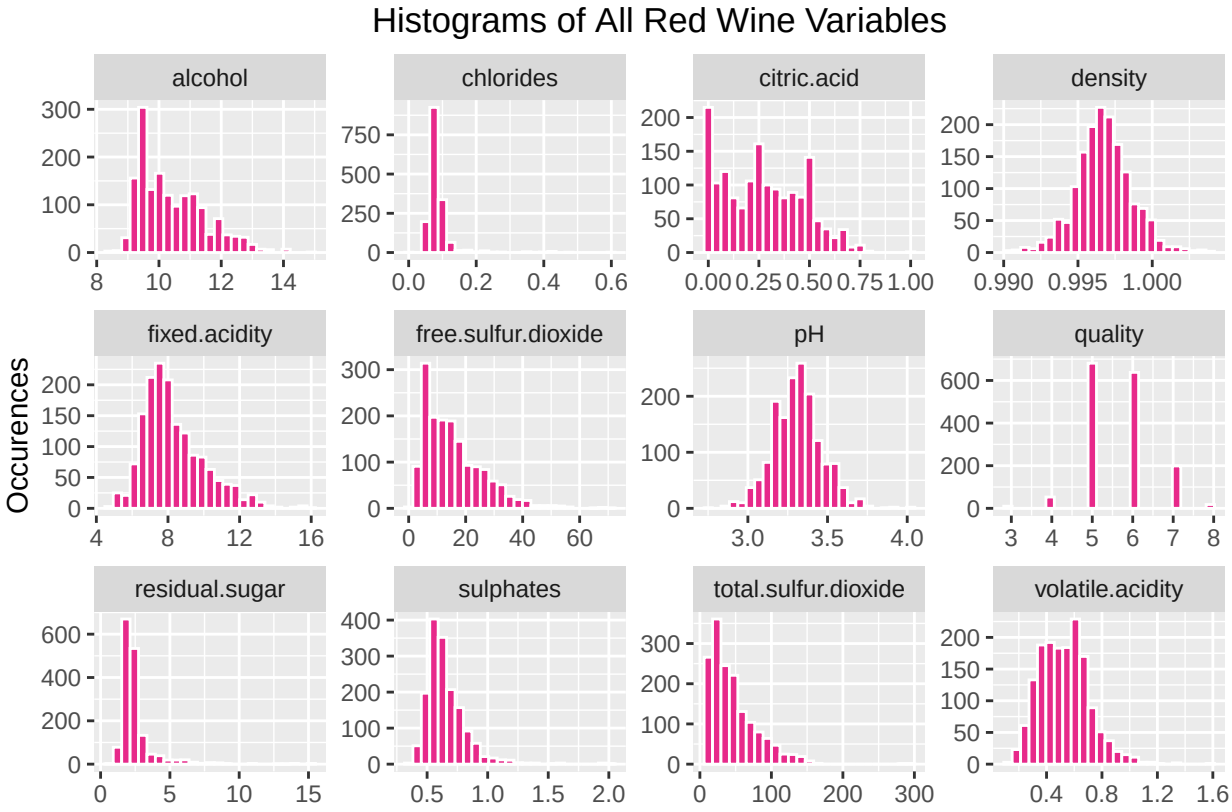
```
## 'data.frame':    1599 obs. of  12 variables:
##  $ fixed.acidity       : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
##  $ volatile.acidity    : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
##  $ citric.acid         : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
##  $ residual.sugar      : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
##  $ chlorides           : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...
##  $ free.sulfur.dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
##  $ total.sulfur.dioxide: num  34 67 54 60 34 40 59 21 18 102 ...
##  $ density             : num  0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
##  $ sulphates           : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
##  $ alcohol             : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
##  $ quality             : int  5 5 5 6 5 5 5 7 7 5 ...
```

```
#checking missing values
colSums(is.na(wine_data))
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##                    0                    0                    0
##       residual.sugar            chlorides  free.sulfur.dioxide
```

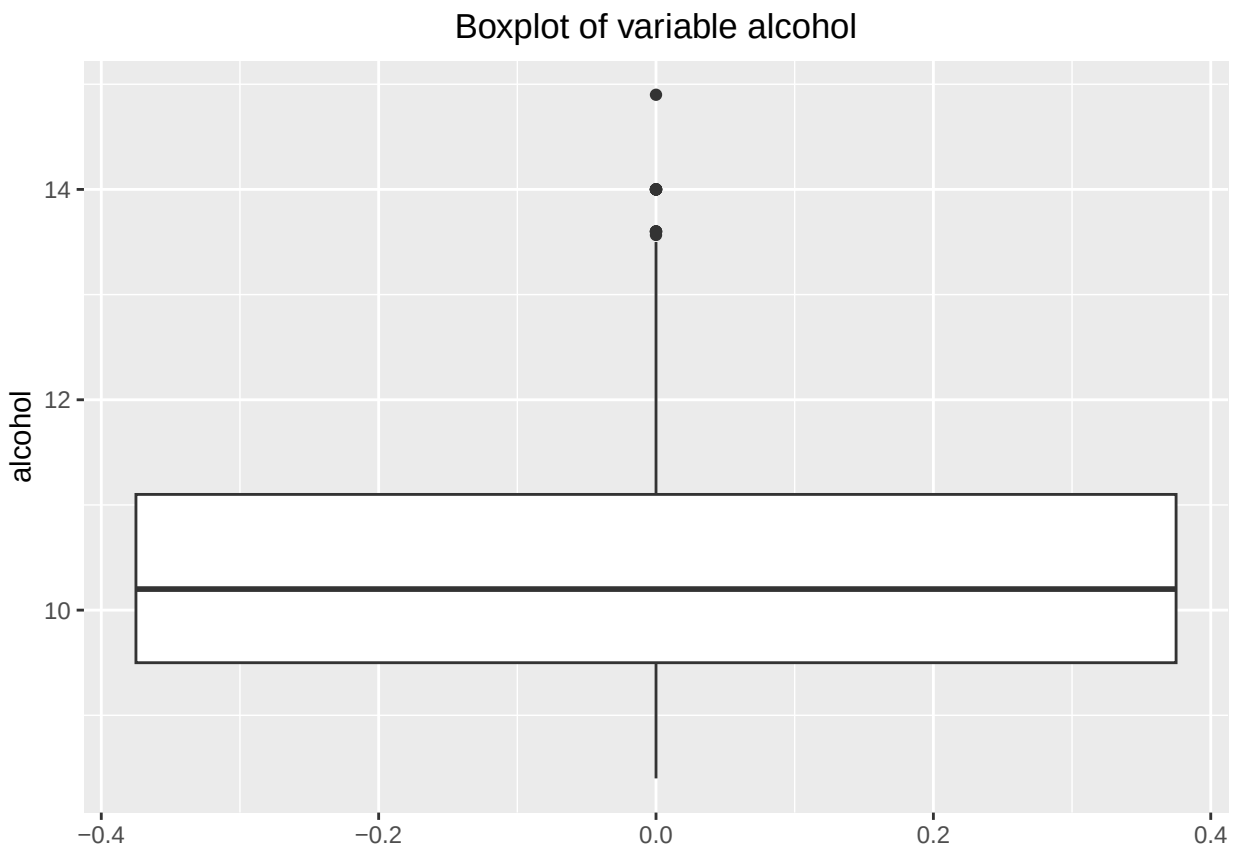
```
##                    0                    0                    0
## total.sulfur.dioxide              density                   pH
##                    0                    0                    0
##            sulphates              alcohol              quality
##                    0                    0                    0
```

```
#visualization of distributions and relationships
#Histograms
library(tidyr)
library(ggplot2)

# reshaping data format
re_wine <- pivot_longer(wine_data, everything(), names_to = "variables", values_to = "values")

# all histogram
ggplot(re_wine, aes(x = values)) +
  geom_histogram(bins = 25, fill = "#E7298A", color = "white") +
  facet_wrap(~ variables, scales = "free", ncol = 4) +
  labs(title = "Histograms of All Red Wine Variables", x = "", y = "Occurences")+
  theme(plot.title = element_text(hjust = 0.5))
```



alcohol, chlorides, citric acid, fixed acidity, free sulfur dioxide, residual sugar, sulphates, total sulfur dioxide, and volatile acid are right skewed. The density and pH is normally distributed. Quality is discrete and clusterd around 5 and 6, resembling a bell shape but not truly normal.

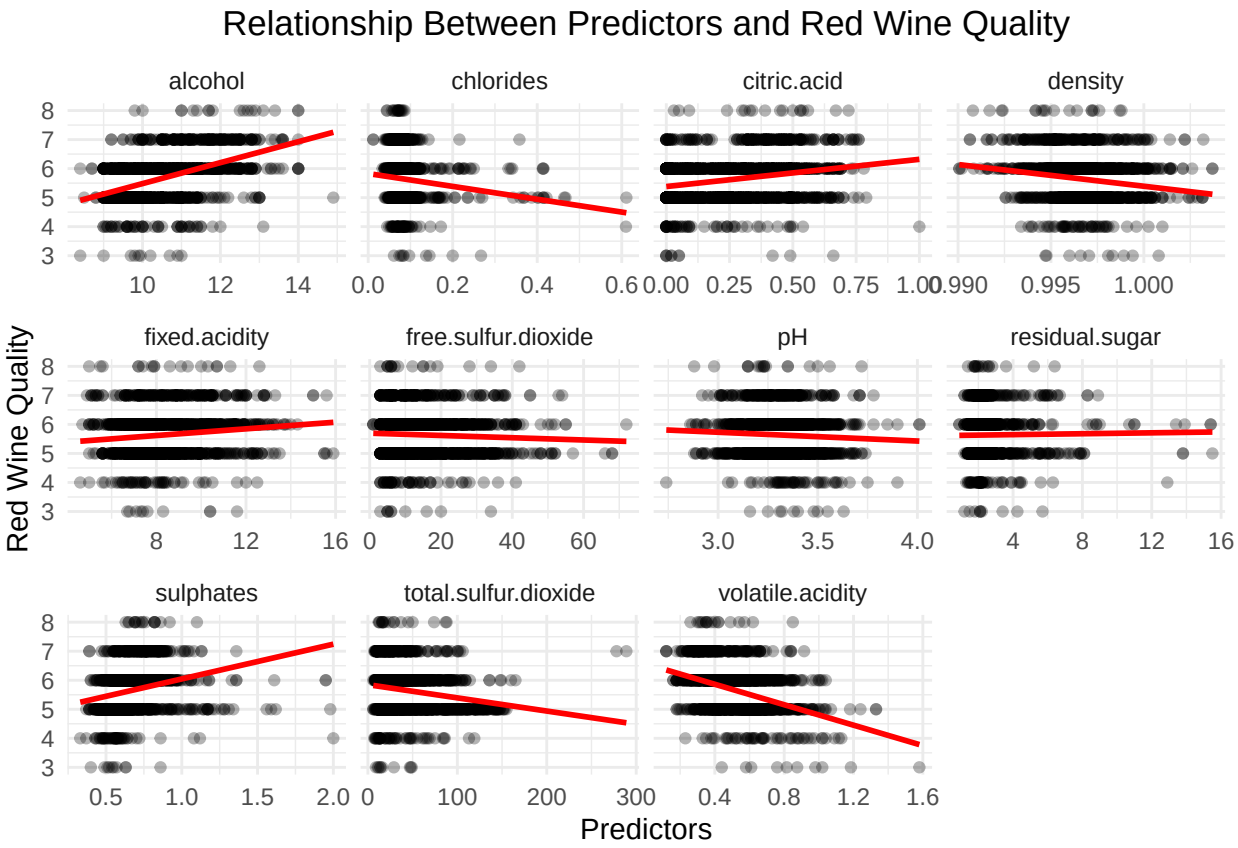
```
# Boxplot for alcohol
ggplot(wine_data, aes(y = `alcohol`)) +
  geom_boxplot() +
  labs(title = "Boxplot of variable alcohol")+
  theme(plot.title = element_text(hjust = 0.5))
```



two outliers present for variable alcohol.

```
# Scatterplots
# Reshape the format except quality
re_wine_without_quality <- pivot_longer(wine_data,
                          cols = -quality,
                          names_to = "predictors",
                          values_to = "values")

ggplot(re_wine_without_quality, aes(x = values, y = quality)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  facet_wrap(~ predictors, scales = "free_x", ncol = 4) +
  labs(title = "Relationship Between Predictors and Red Wine Quality",
       x = "Predictors", y = "Red Wine Quality") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Positive association: alcohol, citric acid, fixed acidity, sulphates

Negative association: chlorides, density, total sulfur dioxide, and volatile acidity

Little to no association: free sulfur dioxide, pH, and residual sugar

Data cleaning and preprocessing steps

The dataset required minimal cleaning. There were no missing values or incorrect data types. Some high-end outliers were present (e.g., in alcohol), but they appeared plausible and were retained. No transformations or standardizations were applied at this stage, as the modeling will use variables in their original units.

```
#fitting full model
redwine_fullmodel<-lm(quality ~ ., data = wine_data)
summary(redwine_fullmodel)
```

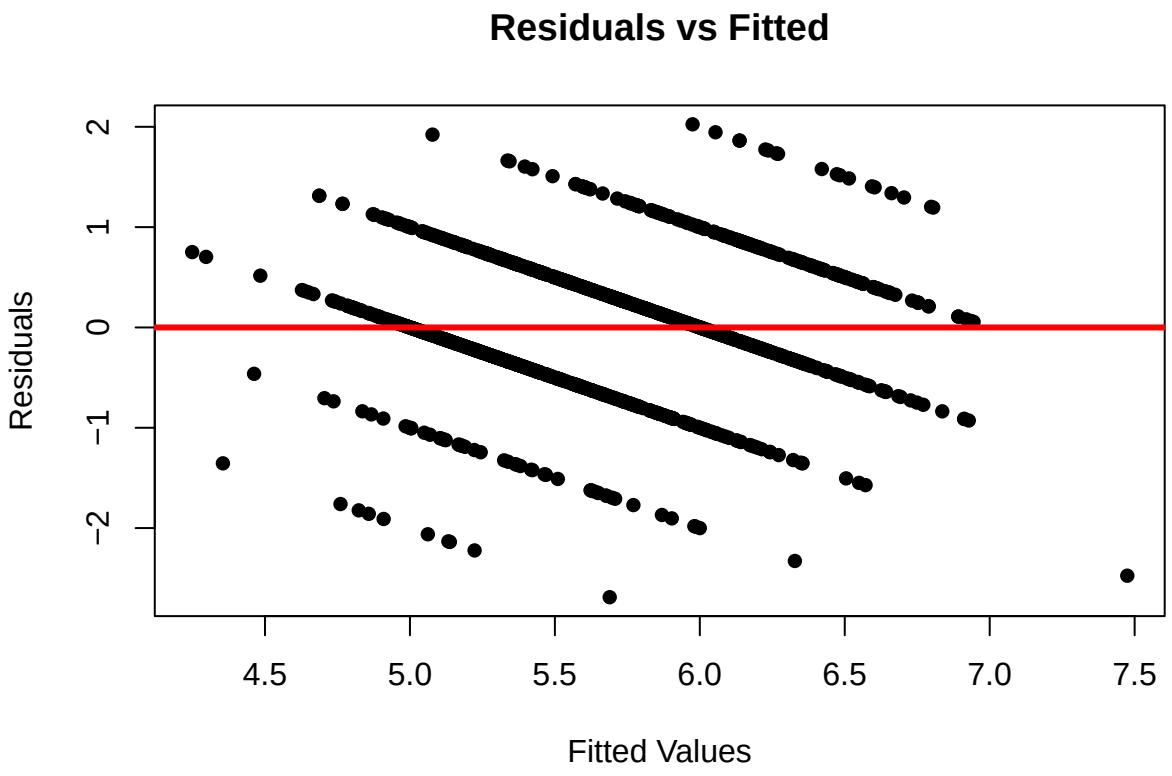
Regression Assumptions Verification

```
##
## Call:
## lm(formula = quality ~ ., data = wine_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.68911 -0.36652 -0.04699  0.45202  2.02498
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)           2.197e+01  2.119e+01   1.036   0.3002
## fixed.acidity         2.499e-02  2.595e-02   0.963   0.3357
## volatile.acidity     -1.084e+00  1.211e-01  -8.948  < 2e-16 ***
## citric.acid          -1.826e-01  1.472e-01  -1.240   0.2150
## residual.sugar        1.633e-02  1.500e-02   1.089   0.2765
## chlorides            -1.874e+00  4.193e-01  -4.470 8.37e-06 ***
## free.sulfur.dioxide   4.361e-03  2.171e-03   2.009   0.0447 *
## total.sulfur.dioxide -3.265e-03  7.287e-04  -4.480 8.00e-06 ***
## density              -1.788e+01  2.163e+01  -0.827   0.4086
## pH                   -4.137e-01  1.916e-01  -2.159   0.0310 *
## sulphates             9.163e-01  1.143e-01   8.014 2.13e-15 ***
## alcohol               2.762e-01  2.648e-02  10.429  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.648 on 1587 degrees of freedom
## Multiple R-squared:  0.3606, Adjusted R-squared:  0.3561
## F-statistic: 81.35 on 11 and 1587 DF,  p-value: < 2.2e-16
```

- Linearity assessment

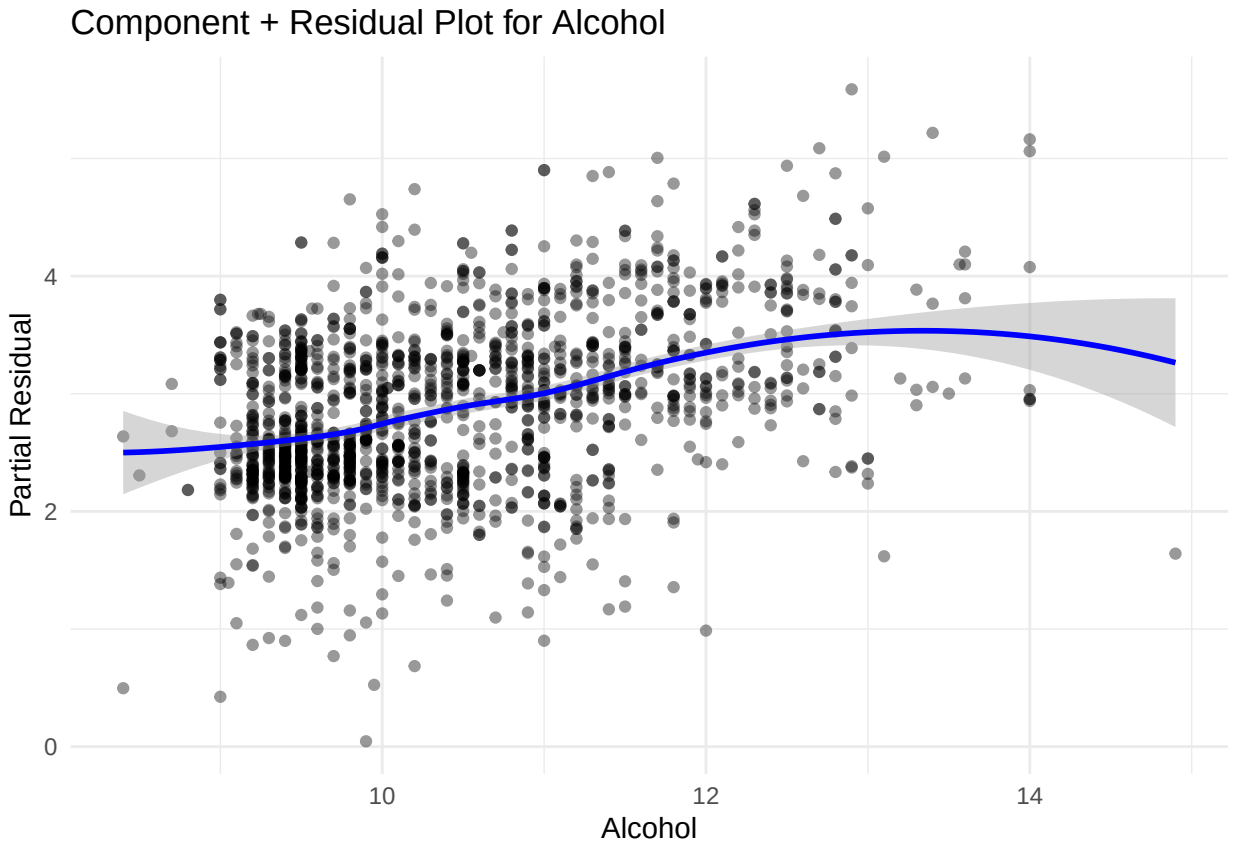
```r
# Residual vs. Fitted plot
plot(redwine_fullmodel$fitted.values, redwine_fullmodel$residuals,
     pch = 16, col = "black", xlab = "Fitted Values", ylab = "Residuals",
     main = "Residuals vs Fitted")
abline(h = 0, col = "red", lwd = 3)
```




```r
#linearity of alcohol
ggplot(wine_data, aes(x = alcohol, y = residuals(redwine_fullmodel) + coef(redwine_fullmodel)["alcohol"]
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", color = "blue") +
```

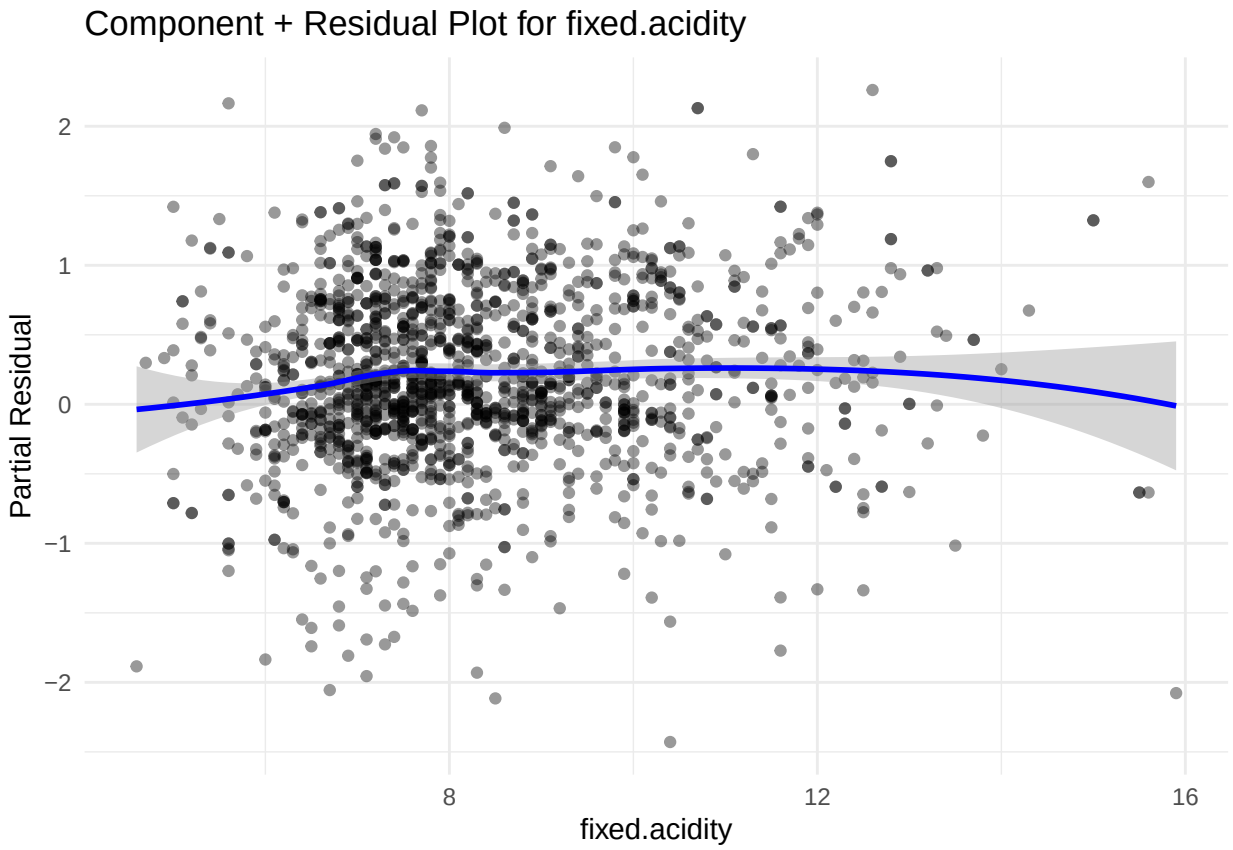
```
  labs(title = "Component + Residual Plot for Alcohol",
       x = "Alcohol", y = "Partial Residual") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



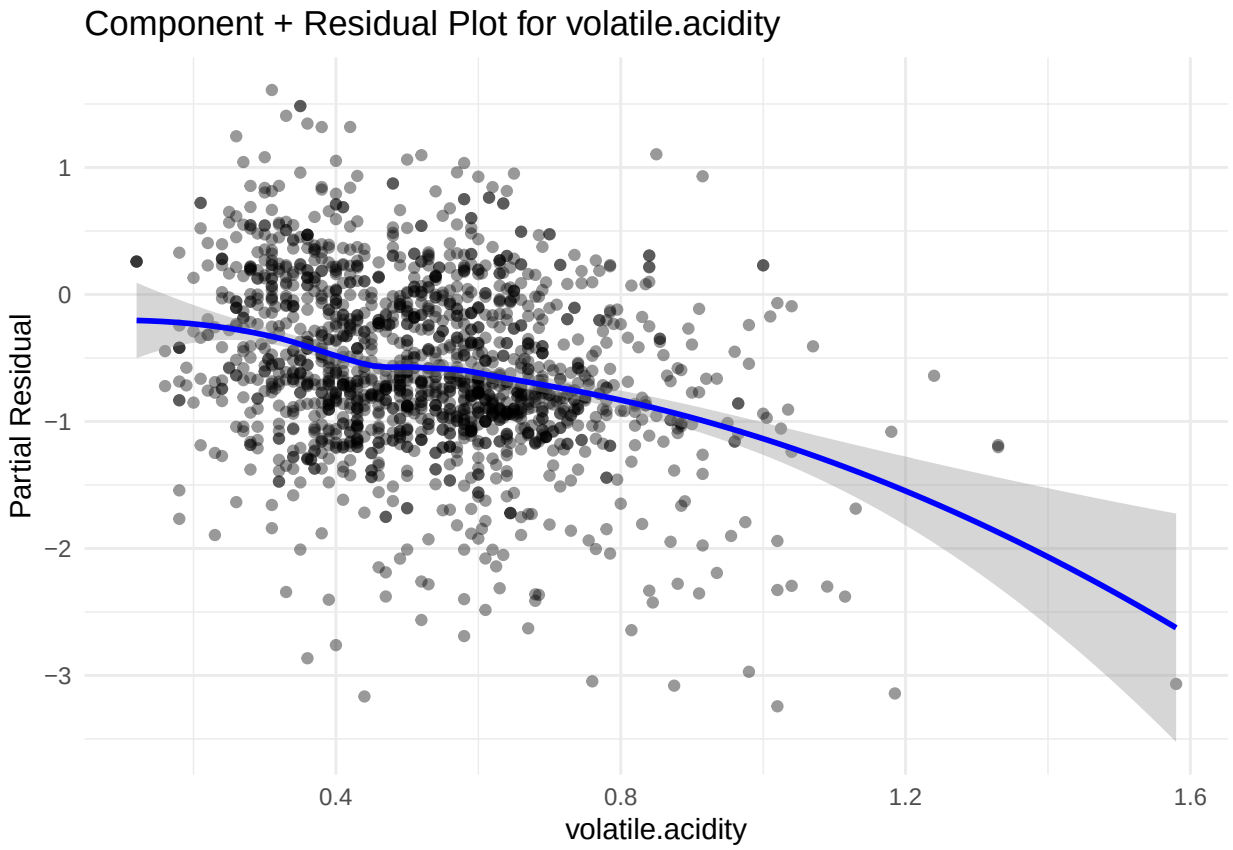

```
# fixed acidity linearity
ggplot(wine_data, aes(x = fixed.acidity, y = residuals(redwine_fullmodel) + coef(redwine_fullmodel)["fi
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", color = "blue") +
  labs(title = "Component + Residual Plot for fixed.acidity",
       x = "fixed.acidity", y = "Partial Residual") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



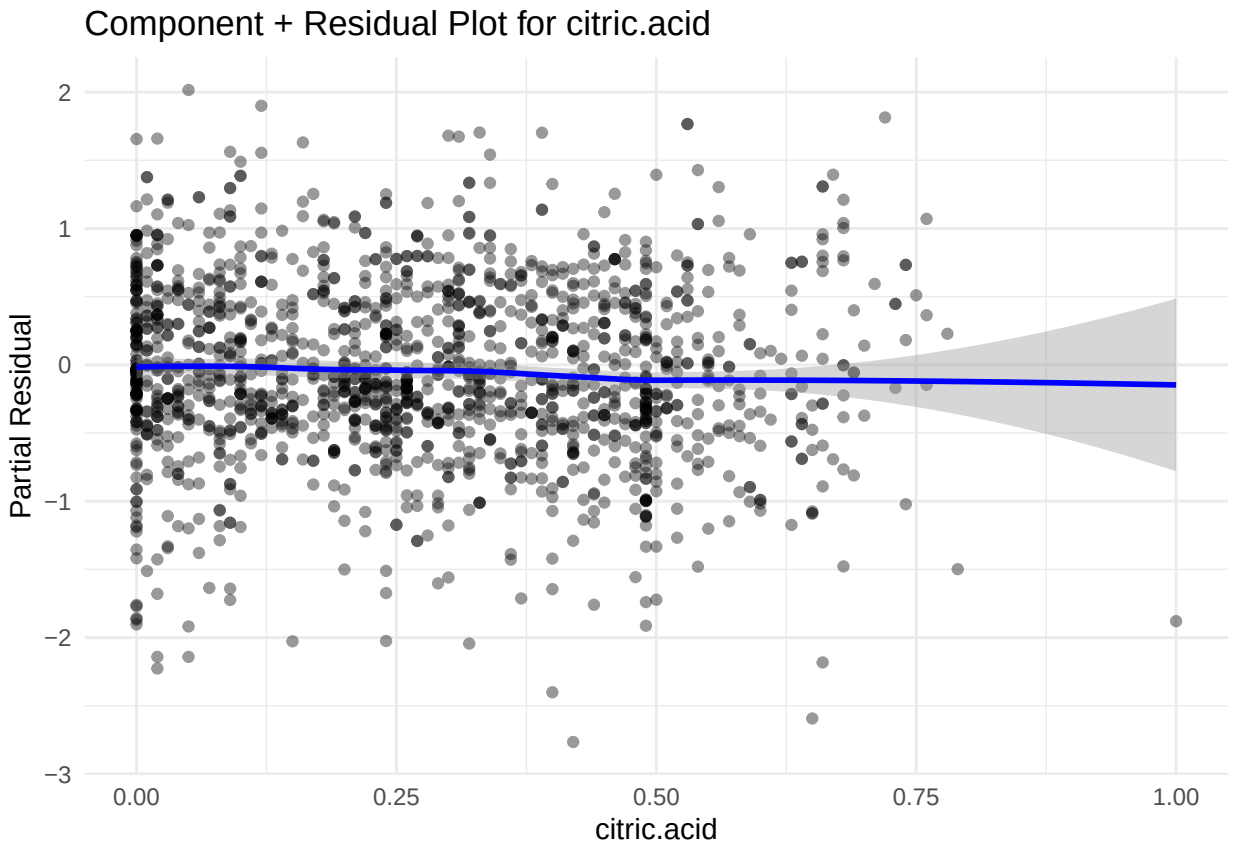

```
#volatile.acidity linearity
ggplot(wine_data, aes(x = volatile.acidity, y = residuals(redwine_fullmodel) + coef(redwine_fullmodel)[
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", color = "blue") +
  labs(title = "Component + Residual Plot for volatile.acidity",
       x = "volatile.acidity", y = "Partial Residual") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



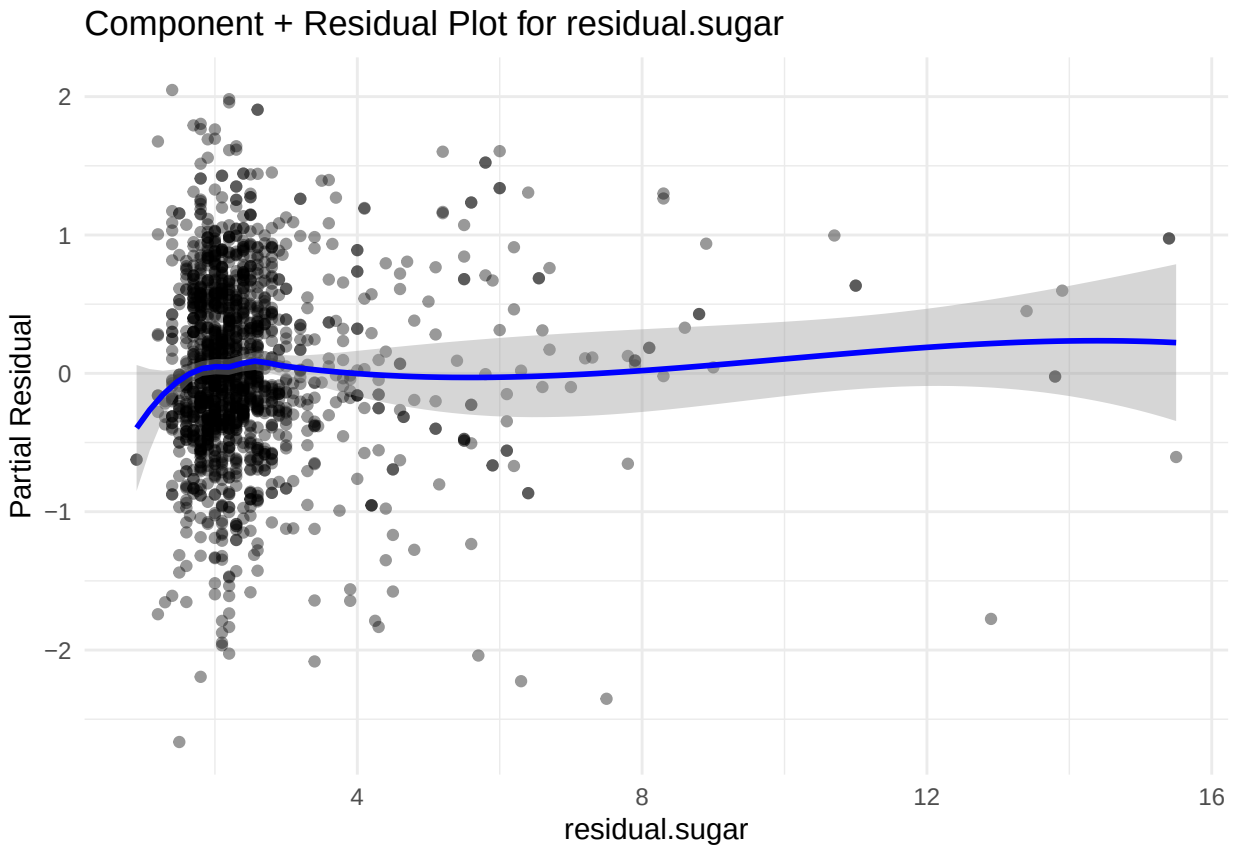

```
# citric.acid
ggplot(wine_data, aes(x = citric.acid, y = residuals(redwine_fullmodel) + coef(redwine_fullmodel)["citri
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", color = "blue") +
  labs(title = "Component + Residual Plot for citric.acid",
       x = "citric.acid", y = "Partial Residual") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



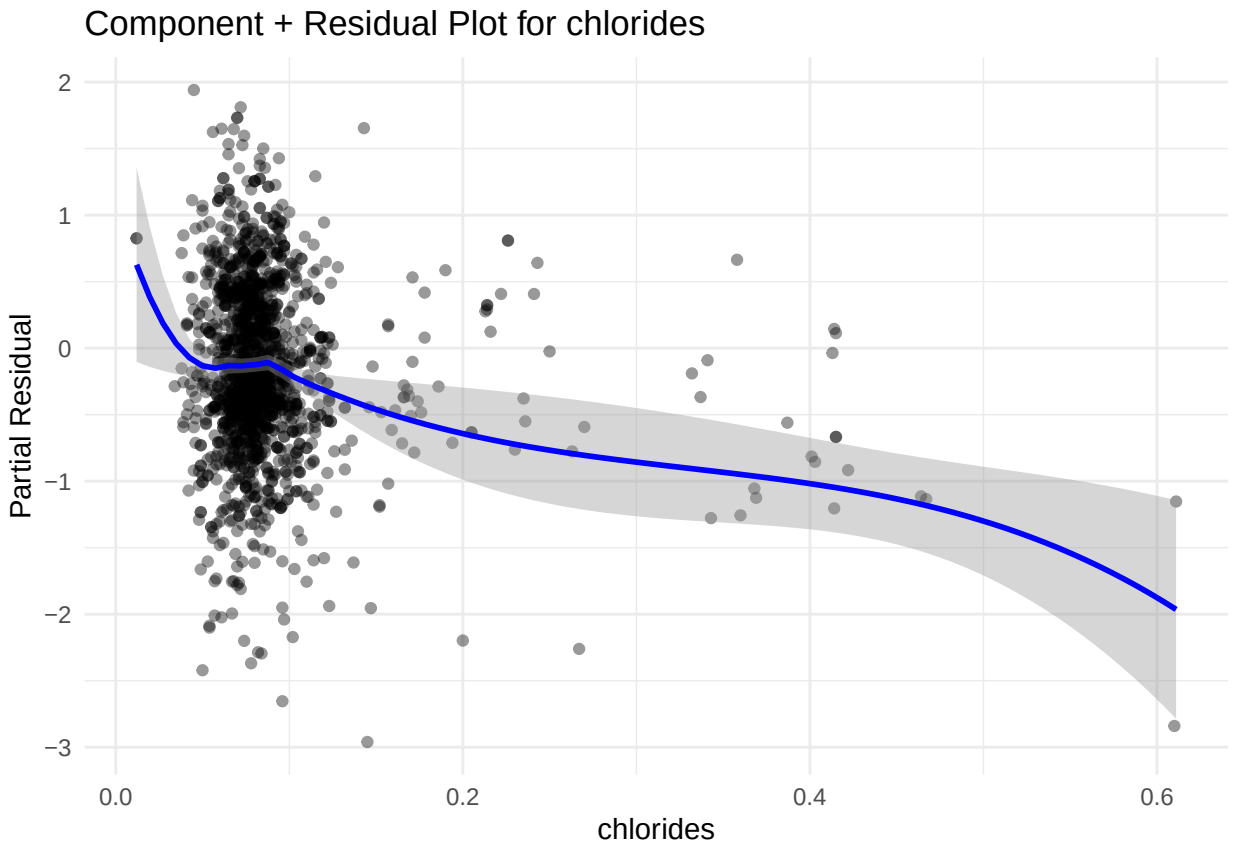

```
# residual.sugar
ggplot(wine_data, aes(x = residual.sugar, y = residuals(redwine_fullmodel) + coef(redwine_fullmodel)["r
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", color = "blue") +
  labs(title = "Component + Residual Plot for residual.sugar",
       x = "residual.sugar", y = "Partial Residual") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



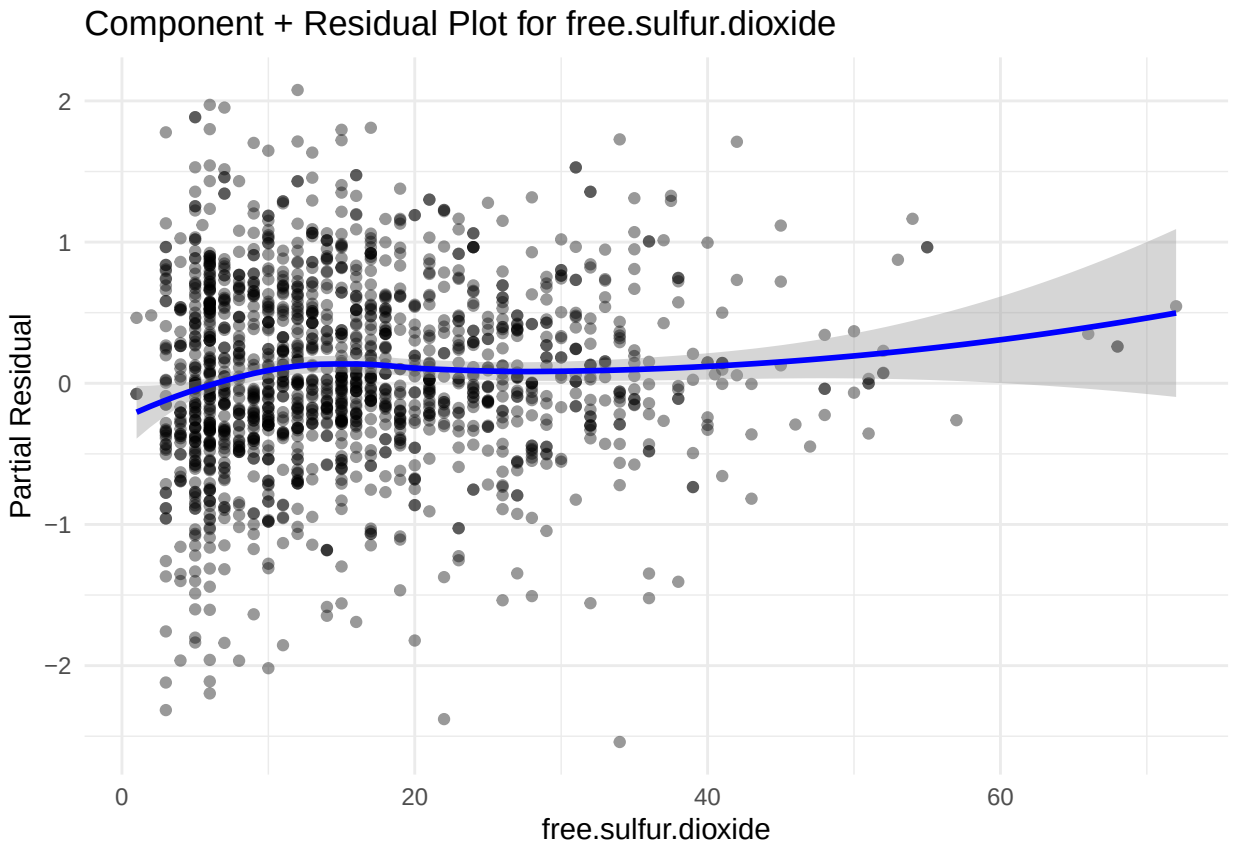

```
# chlorides
ggplot(wine_data, aes(x = chlorides, y = residuals(redwine_fullmodel) + coef(redwine_fullmodel)["chlorid
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", color = "blue") +
  labs(title = "Component + Residual Plot for chlorides",
       x = "chlorides", y = "Partial Residual") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



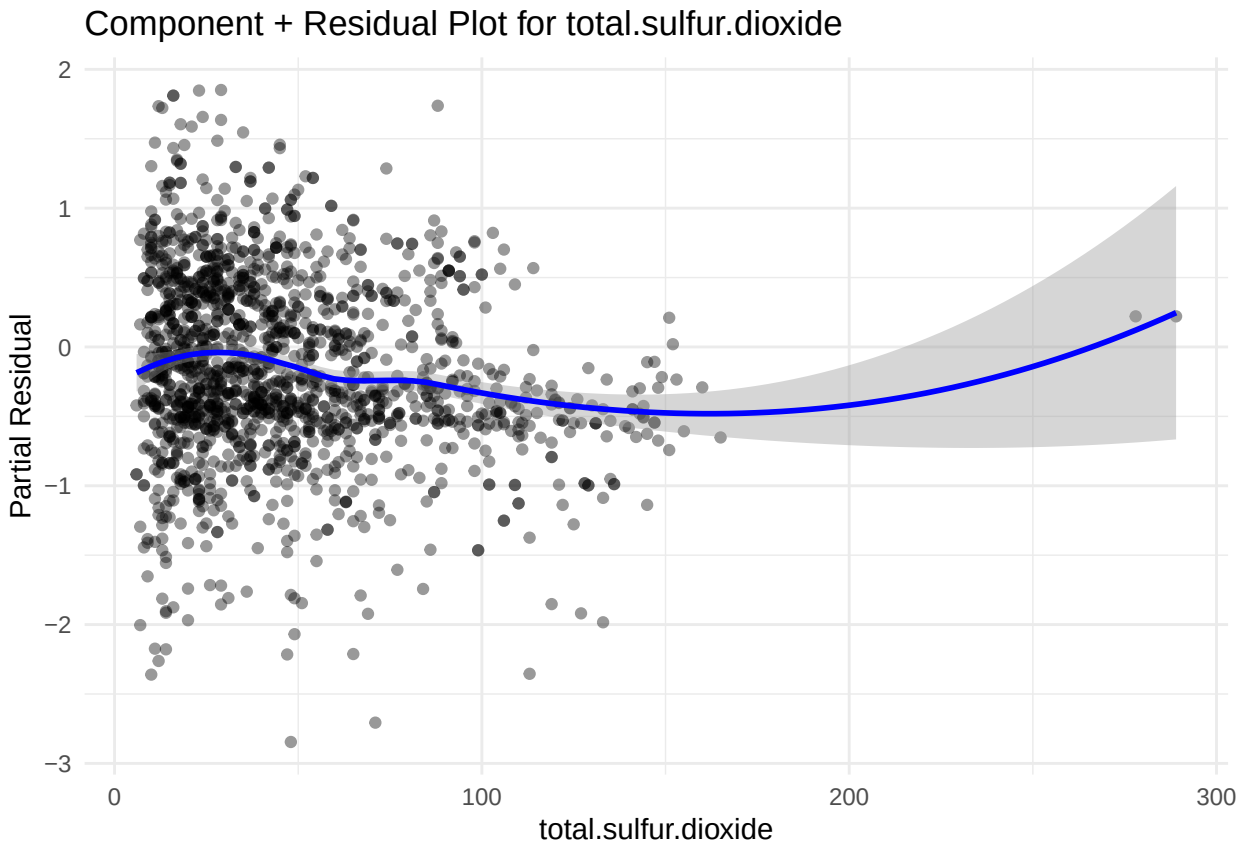

```
#free.sulfur.dioxide
ggplot(wine_data, aes(x = free.sulfur.dioxide, y = residuals(redwine_fullmodel) + coef(redwine_fullmodel
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", color = "blue") +
  labs(title = "Component + Residual Plot for free.sulfur.dioxide",
       x = "free.sulfur.dioxide", y = "Partial Residual") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



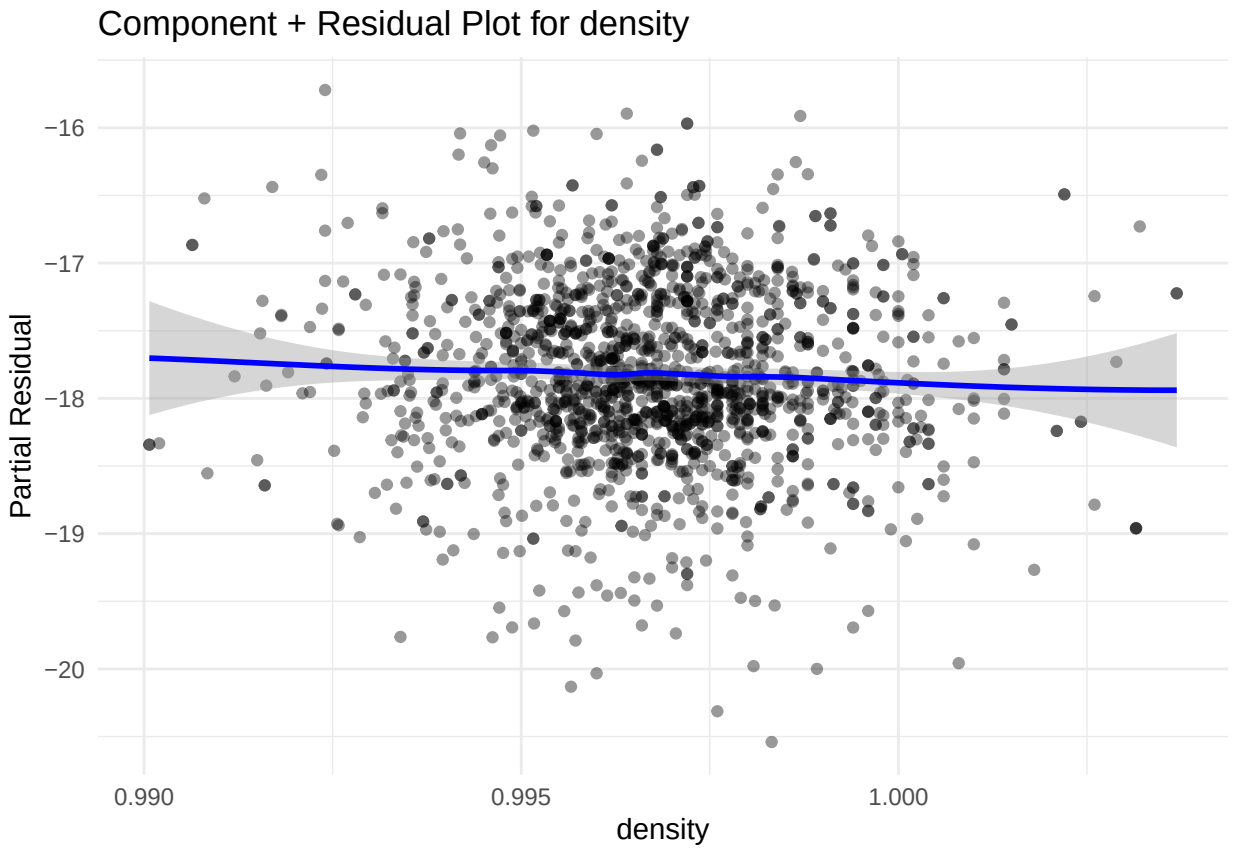

```
# total.sulfur.dioxide
ggplot(wine_data, aes(x = total.sulfur.dioxide, y = residuals(redwine_fullmodel) + coef(redwine_fullmod
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", color = "blue") +
  labs(title = "Component + Residual Plot for total.sulfur.dioxide",
       x = "total.sulfur.dioxide", y = "Partial Residual") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



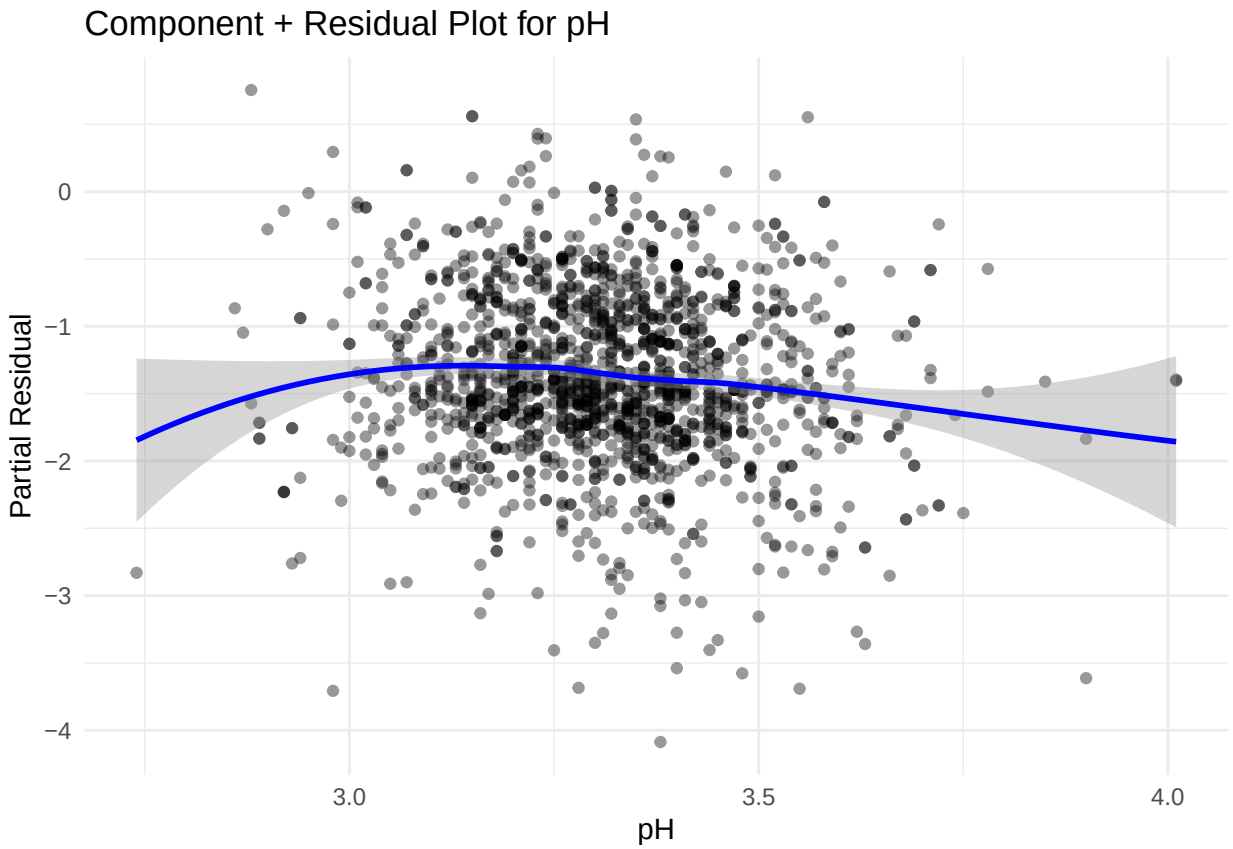

```
# density
ggplot(wine_data, aes(x = density, y = residuals(redwine_fullmodel) + coef(redwine_fullmodel)["density"]
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", color = "blue") +
  labs(title = "Component + Residual Plot for density",
       x = "density", y = "Partial Residual") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



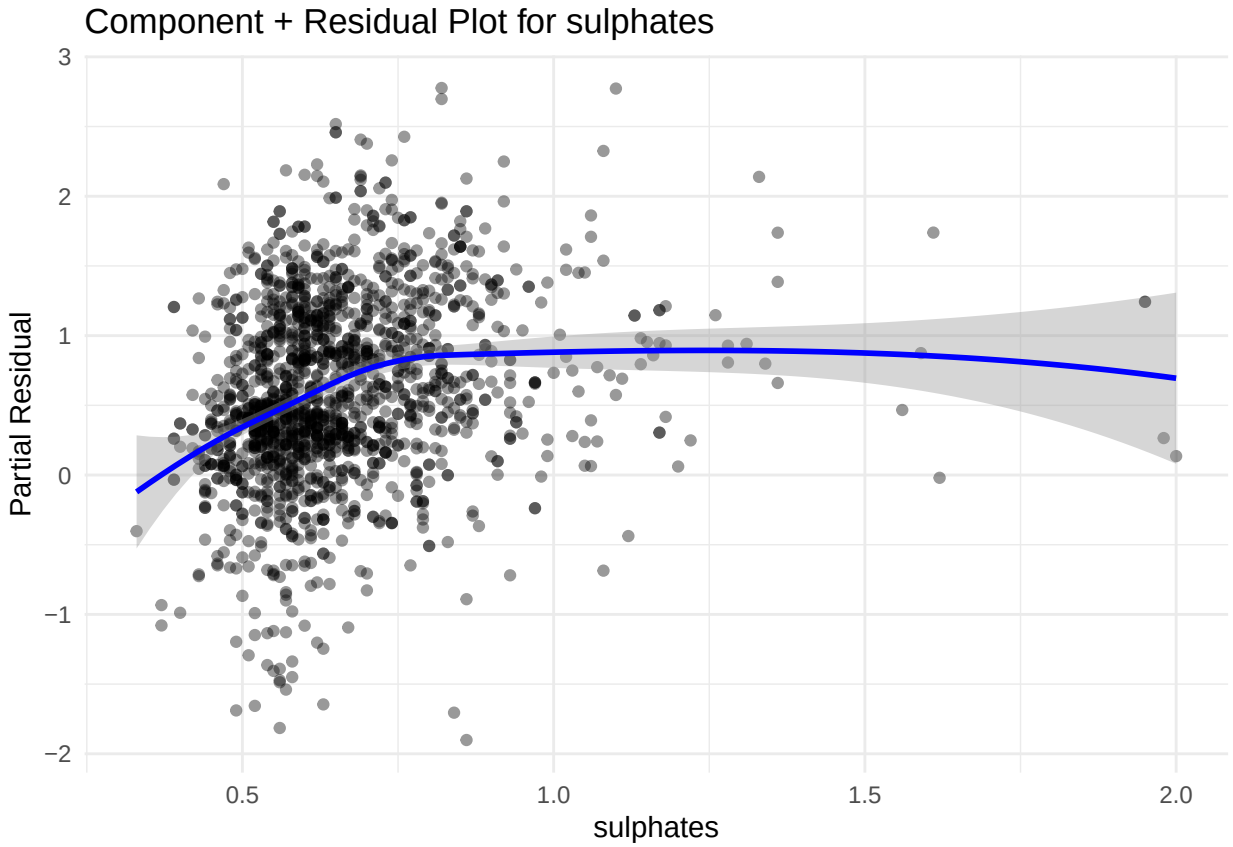

```
# pH
ggplot(wine_data, aes(x = pH, y = residuals(redwine_fullmodel) + coef(redwine_fullmodel)["pH"] * pH)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", color = "blue") +
  labs(title = "Component + Residual Plot for pH",
       x = "pH", y = "Partial Residual") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# sulphates
ggplot(wine_data, aes(x = sulphates, y = residuals(redwine_fullmodel) + coef(redwine_fullmodel)["sulpha
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", color = "blue") +
  labs(title = "Component + Residual Plot for sulphates",
       x = "sulphates", y = "Partial Residual") +
  theme_minimal()
```

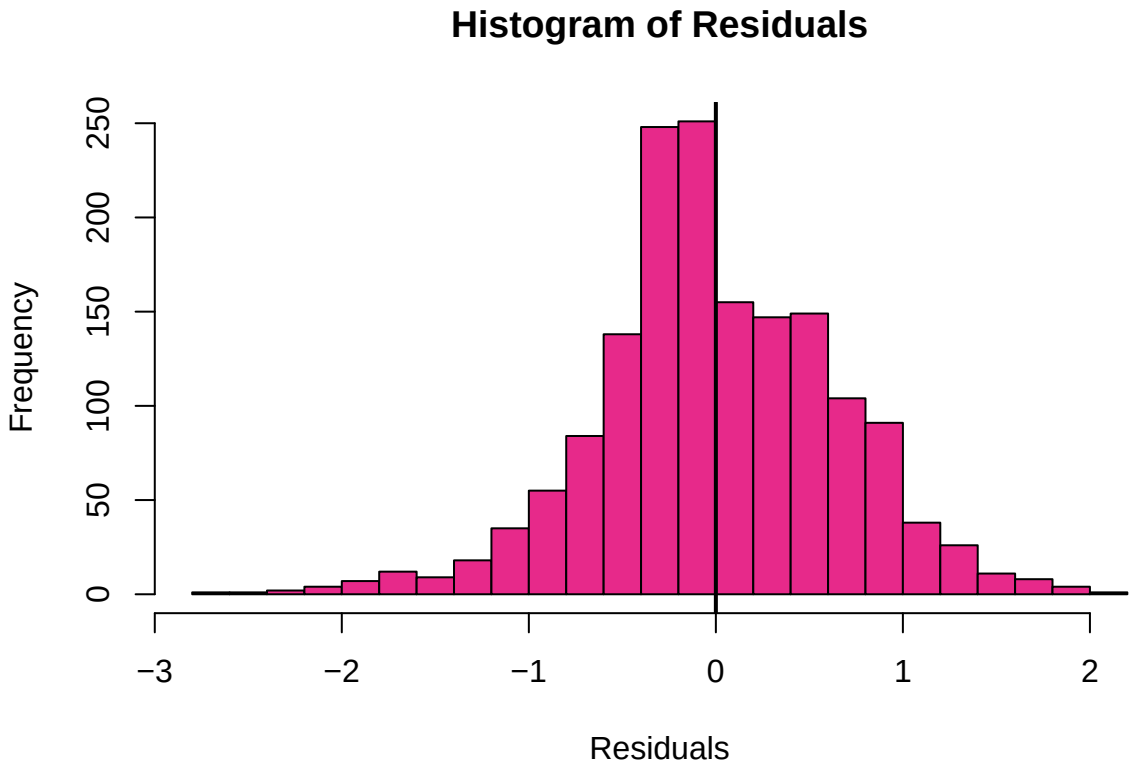
```
## `geom_smooth()` using formula = 'y ~ x'
```



Although some component + residual plots show mild curvature, such as for total sulfur dioxide or free sulfur dioxide, the patterns are not strongly nonlinear. Therefore, I conclude that the linearity assumption is reasonably satisfied for the predictors in this model.

- Normality of residuals

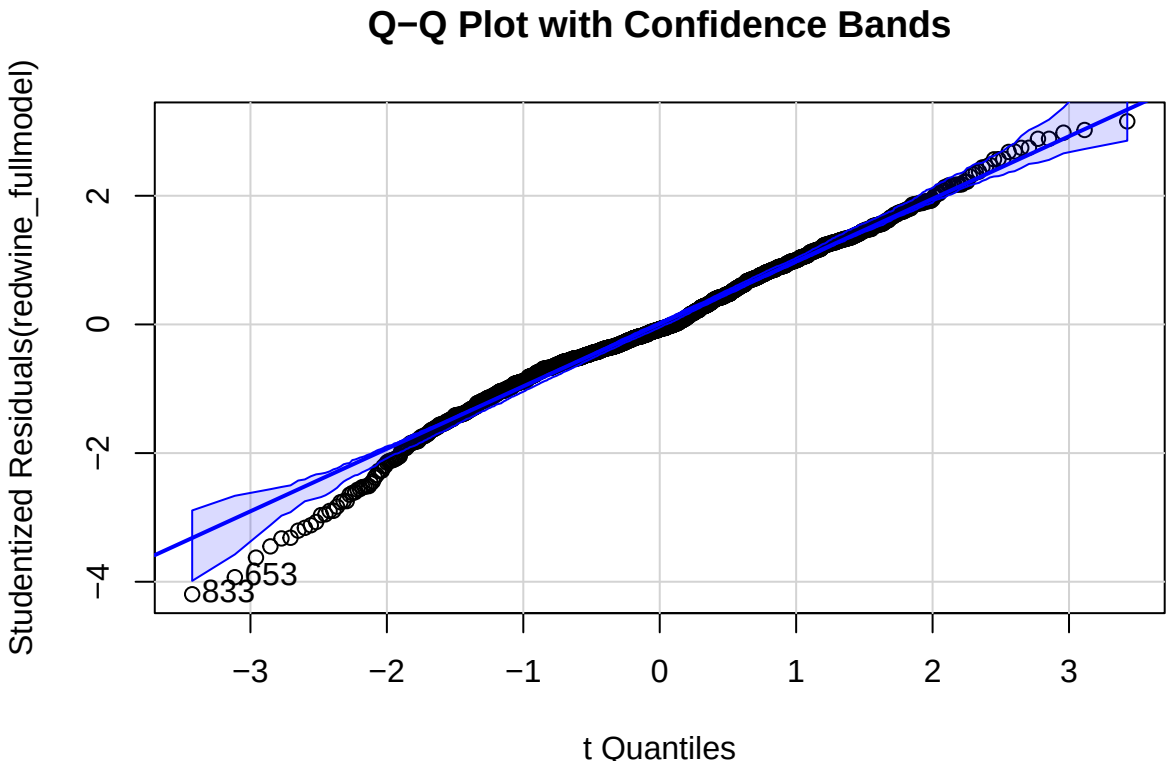
```
# Histogram of residuals with vertical line at 0
hist(residuals(redwine_fullmodel),
     breaks = 30,
     col = "#E7298A",
     main = "Histogram of Residuals",
     xlab = "Residuals")

abline(v = 0, col = "black", lwd = 2)
```




```
# Q-Q plot
library(car)
```

```
## Loading required package: carData
```

```
qqPlot(redwine_fullmodel,
       main = "Q-Q Plot with Confidence Bands")
```

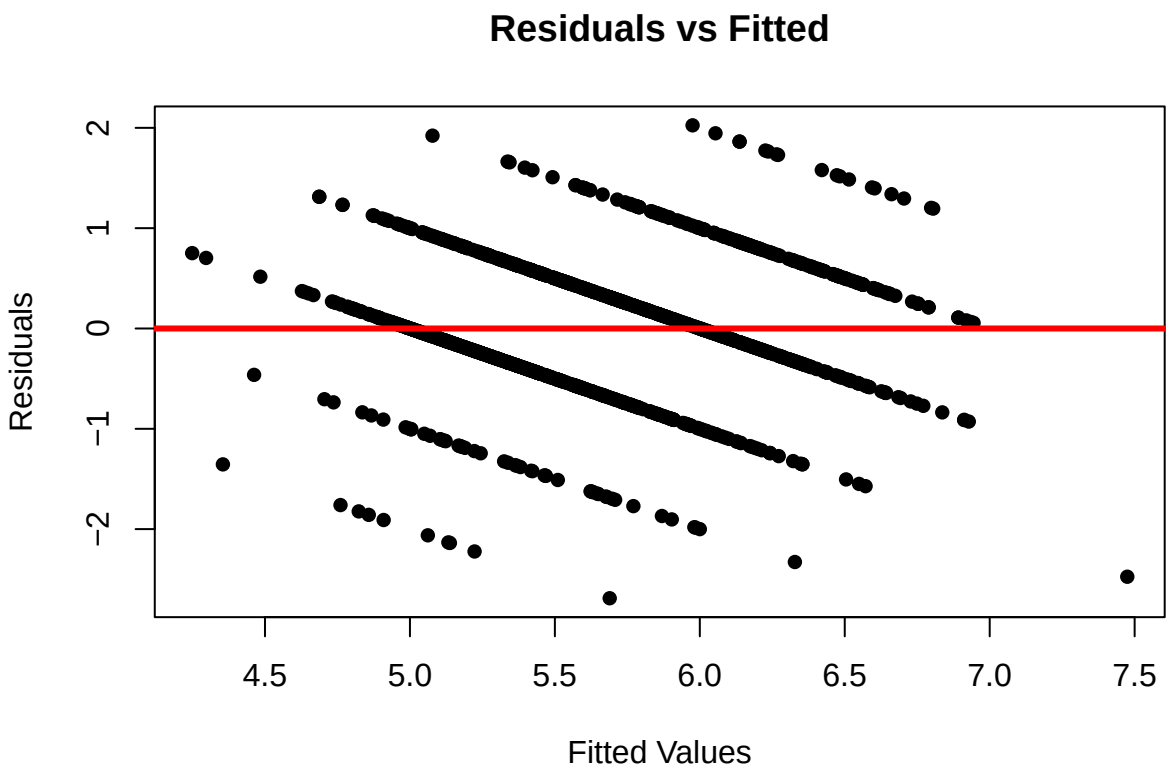



```
## [1] 653 833
```

The histogram of residuals shows a roughly symmetric, bell-shaped distribution centered at 0. In the Q-Q plot with confidence bands, most residuals fall within the band, but several points in the lower left tail fall outside the band. This indicates a mild violation of the normality assumption.

- Homoscedasticity (constant variance of residuals)

```
# Residual vs. Fitted plot
plot(redwine_fullmodel$fitted.values, redwine_fullmodel$residuals,
     pch = 16, col = "black", xlab = "Fitted Values", ylab = "Residuals",
     main = "Residuals vs Fitted")
abline(h = 0, col = "red", lwd = 3)
```




```
# bp tesst
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
bptest(redwine_fullmodel)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  redwine_fullmodel
## BP = 84.989, df = 11, p-value = 1.588e-13
```

H_0: The residuals are homoscedastic.

H_a: The residuals are heteroscedastic.

The p-value is very small (p= 1.588e-13), so we have sufficient evidence to reject the null. Thus, the residuals are heteroscedastic. This means the constant variance assumption does not hold.

- Independence of observations

According to the publication of this dataset (Cortez et al., 2009), each row represents a distinct wine sample, with physico-chemical and sensory tests recorded for each. The database was preprocessed to ensure one row per sample, and sensory scores from multiple assessors were aggregated using the median. There is no indication of repeated measures or clustering. Thus, the assumption of independent observations holds.

- Multicollinearity assessment # Variance inflation factor (VIF)

```r
vif(redwine_fullmodel)
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##             7.767512             1.789390             3.128022
##       residual.sugar            chlorides  free.sulfur.dioxide
##             1.702588             1.481932             1.963019
## total.sulfur.dioxide              density                   pH
##             2.186813             6.343760             3.329732
##            sulphates              alcohol
##             1.429434             3.031160
```

The Variance Inflation Factor (VIF) is used to assess multicollinearity among predictors. A rule of thumb is that when VIF values above 5, it suggesst moderate multicollinearity. When VIF values are above 10, it suggests severe multicollinearity. In this model, fixed.acidity (VIF = 7.77) and density (VIF = 6.34) both exceed the threshold of 5, indicating the presence of moderate multicollinearity. The remaining predictors all have VIFs below 5, suggesting no strong multicollinearity is present.

Assumption Violation Handling In assessing the assumptions of linear regression for the red wine quality model, the linearity assumption appeared reasonably satisfied based on partial residual plots, which showed generally linear relationships between predictors and the outcome. The normality of residuals showed mild violation; although the histogram was roughly bell-shaped and centered at zero, the Q-Q plot indicated deviations in the lower tail. The Breusch–Pagan test returned a p-value of 1.588e-13, providing strong evidence against the homoscedasticity assumption, indicating that residual variance is not constant. Independence of observations is likely satisfied, as each entry in the dataset represents a distinct wine sample collected and evaluated independently. Lastly, variance inflation factor (VIF) analysis showed moderate multicollinearity for 'fixed.acidity' and 'density', with VIF values above 5, suggesting some redundancy among predictors.

- Handing homoscedasticity violation

log transformation

```r
redwine_logged <- wine_data

#checking if zeroes are present
any(wine_data$residual.sugar == 0)
```

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

```r
any(wine_data$chlorides == 0)
```

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

```r
any(wine_data$free.sulfur.dioxide == 0)
```

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

```r
any(wine_data$total.sulfur.dioxide == 0)
```

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

```r
any(wine_data$sulphates == 0)
```

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

```r
any(wine_data$alcohol == 0)
```

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

```r
any(wine_data$citric.acid == 0)
```

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

```
any(wine_data$fixed.acidity == 0)
```

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

```
any(wine_data$volatile.acidity == 0)
```

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

```
#log transformation to right-skewed predictors
redwine_logged$residual.sugar <- log(redwine_logged$residual.sugar)
redwine_logged$chlorides <- log(redwine_logged$chlorides)
redwine_logged$free.sulfur.dioxide <- log(redwine_logged$free.sulfur.dioxide)
redwine_logged$total.sulfur.dioxide <- log(redwine_logged$total.sulfur.dioxide)
redwine_logged$sulphates <- log(redwine_logged$sulphates)
redwine_logged$alcohol <- log(redwine_logged$alcohol)
redwine_logged$citric.acid <- log(redwine_logged$citric.acid+1)
redwine_logged$fixed.acidity <- log(redwine_logged$fixed.acidity)
redwine_logged$volatile.acidity <- log(redwine_logged$volatile.acidity)
```

```
#fitting log transformed model
redwine_logged_fit <- lm(quality ~ ., data = redwine_logged)
summary(redwine_logged_fit)
```

```
##
## Call:
## lm(formula = quality ~ ., data = redwine_logged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.67690 -0.35684 -0.04525  0.44379  1.94058
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           29.13264   22.98357   1.268 0.205147
## fixed.acidity          0.42056    0.22944   1.833 0.066998 .
## volatile.acidity      -0.51209    0.06291  -8.140 7.88e-16 ***
## citric.acid           -0.38684    0.18378  -2.105 0.035460 *
## residual.sugar         0.09064    0.06305   1.438 0.150764
## chlorides             -0.23758    0.05875  -4.044 5.50e-05 ***
## free.sulfur.dioxide    0.09848    0.03999   2.463 0.013899 *
## total.sulfur.dioxide  -0.14364    0.04087  -3.515 0.000453 ***
## density              -30.05645   23.31059  -1.289 0.197450
## pH                    -0.34440    0.19621  -1.755 0.079409 .
## sulphates              0.81712    0.08554   9.552  < 2e-16 ***
## alcohol                2.73983    0.29369   9.329  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6461 on 1587 degrees of freedom
## Multiple R-squared:  0.3642, Adjusted R-squared:  0.3598
## F-statistic: 82.66 on 11 and 1587 DF,  p-value: < 2.2e-16
```

```
#compare full model vs. log-transformed model
summary(redwine_fullmodel)
```

```
##
## Call:
## lm(formula = quality ~ ., data = wine_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.68911 -0.36652 -0.04699  0.45202  2.02498
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           2.197e+01  2.119e+01   1.036   0.3002
## fixed.acidity         2.499e-02  2.595e-02   0.963   0.3357
## volatile.acidity     -1.084e+00  1.211e-01  -8.948  < 2e-16 ***
## citric.acid          -1.826e-01  1.472e-01  -1.240   0.2150
## residual.sugar        1.633e-02  1.500e-02   1.089   0.2765
## chlorides            -1.874e+00  4.193e-01  -4.470 8.37e-06 ***
## free.sulfur.dioxide   4.361e-03  2.171e-03   2.009   0.0447 *
## total.sulfur.dioxide -3.265e-03  7.287e-04  -4.480 8.00e-06 ***
## density              -1.788e+01  2.163e+01  -0.827   0.4086
## pH                   -4.137e-01  1.916e-01  -2.159   0.0310 *
## sulphates             9.163e-01  1.143e-01   8.014 2.13e-15 ***
## alcohol               2.762e-01  2.648e-02  10.429  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.648 on 1587 degrees of freedom
## Multiple R-squared:  0.3606, Adjusted R-squared:  0.3561
## F-statistic: 81.35 on 11 and 1587 DF,  p-value: < 2.2e-16
```

```
summary(redwine_logged_fit)
```

```
##
## Call:
## lm(formula = quality ~ ., data = redwine_logged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.67690 -0.35684 -0.04525  0.44379  1.94058
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           29.13264   22.98357   1.268 0.205147
## fixed.acidity          0.42056    0.22944   1.833 0.066998 .
## volatile.acidity      -0.51209    0.06291  -8.140 7.88e-16 ***
## citric.acid           -0.38684    0.18378  -2.105 0.035460 *
## residual.sugar         0.09064    0.06305   1.438 0.150764
## chlorides             -0.23758    0.05875  -4.044 5.50e-05 ***
## free.sulfur.dioxide    0.09848    0.03999   2.463 0.013899 *
## total.sulfur.dioxide  -0.14364    0.04087  -3.515 0.000453 ***
## density              -30.05645   23.31059  -1.289 0.197450
## pH                    -0.34440    0.19621  -1.755 0.079409 .
## sulphates              0.81712    0.08554   9.552  < 2e-16 ***
## alcohol                2.73983    0.29369   9.329  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.6461 on 1587 degrees of freedom
## Multiple R-squared:  0.3642, Adjusted R-squared:  0.3598
## F-statistic: 82.66 on 11 and 1587 DF,  p-value: < 2.2e-16
```

Log-transformed model has higher R^2 than the original model.

```
AIC(redwine_fullmodel, redwine_logged_fit)
```

```
##                     df      AIC
## redwine_fullmodel   13 3164.277
## redwine_logged_fit  13 3155.032
```

```
BIC(redwine_fullmodel, redwine_logged_fit)
```

```
##                     df      BIC
## redwine_fullmodel   13 3234.179
## redwine_logged_fit  13 3224.934
```

To address the violation of homoscedasticity, log transformation was applied to the right-skewed predictors. After the transformation, model performance showed slight improvement: the adjusted R^2 increased from 0.3561 to 0.3598, AIC decreased from 3164.28 to 3155.03, and BIC decreased from 3234.18 to 3224.93. These changes suggest that the transformed model provides a slightly better fit while maintaining similar complexity.

```
# checking if log-transformation handled the violation successfully
bptest(redwine_logged_fit)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  redwine_logged_fit
## BP = 70.418, df = 11, p-value = 1.018e-10
```

Still the homoscedasticity assumption does not hold after the log transformation.

```
# using robust standard error for valid inference
library(sandwich)
library(lmtest)
redwine_robust <- vcovHC(redwine_logged_fit, type = "HC3")
coeftest(redwine_logged_fit, vcov = redwine_robust)
```

```
##
## t test of coefficients:
##
##                         Estimate Std. Error t value  Pr(>|t|)
## (Intercept)           29.132640  25.837471  1.1275 0.2596870
## fixed.acidity          0.420559   0.270208  1.5564 0.1198050
## volatile.acidity      -0.512094   0.066120 -7.7449 1.693e-14 ***
## citric.acid           -0.386841   0.179662 -2.1532 0.0314567 *
## residual.sugar         0.090636   0.075597  1.1989 0.2307339
## chlorides             -0.237580   0.067482 -3.5206 0.0004427 ***
## free.sulfur.dioxide    0.098484   0.041760  2.3583 0.0184779 *
## total.sulfur.dioxide  -0.143642   0.042279 -3.3975 0.0006970 ***
## density              -30.056449  26.296256 -1.1430 0.2532137
## pH                    -0.344402   0.225266 -1.5289 0.1264952
## sulphates              0.817123   0.088661  9.2163 < 2.2e-16 ***
## alcohol                2.739826   0.325650  8.4134 < 2.2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Heteroscedasticity was addressed by applying log transformations to right-skewed predictors and using robust standard errors to obtain valid inference. Under this correction, the predictors that remain statistically significant include: volatile acidity, citric acid, chlorides, free sulfur dioxide, total sulfur dioxide, sulphates, and alcohol. In contrast, predictors such as fixed acidity and density do not show strong evidence of association with wine quality.

- Handing normality violation

The normality assumption appeared mildly violated based on the Q-Q plot, where several residuals in the lower left tail fell outside the confidence band. However, the histogram showed a roughly symmetric distribution centered around zero, and the majority of residuals in the Q-Q plot remained within the band. Given the mild nature of the violation, the use of robust standard errors already addresses concerns related to inference. Since linear regression is generally robust to mild departures from normality, no further corrective action was taken.

- Handling multicollinearity

Multicollinearity is problematic because it inflates standard errors and introduces redundancy in the model when predictors are highly correlated with one another. This can make coefficient estimates unstable and reduce the reliability of statistical inference. However, since the primary goal of this project is to build a predictive model of wine quality using physicochemical variables, I chose not to remove predictors that showed moderate multicollinearity based on VIF values. These variables may still contribute useful information to the prediction task. In later steps, model reduction or regularization methods may be considered to retain only the most informative predictors for generalizing to new data.

Variable Selection & Hypothesis Testing

- Implement at least two different variable selection techniques

```r
# backward selection

library(MASS)

redwine_back <- stepAIC(redwine_logged_fit, direction = "backward", trace = FALSE)
summary(redwine_back)
```

```
##
## Call:
## lm(formula = quality ~ fixed.acidity + volatile.acidity + citric.acid +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     pH + sulphates + alcohol, data = redwine_logged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.67779 -0.35087 -0.04455  0.44129  1.93469
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -0.57296    0.81576  -0.702 0.482557
## fixed.acidity         0.19709    0.13527   1.457 0.145290
## volatile.acidity     -0.51955    0.06176  -8.412  < 2e-16 ***
## citric.acid          -0.37933    0.18313  -2.071 0.038492 *
## chlorides            -0.24369    0.05777  -4.219  2.6e-05 ***
## free.sulfur.dioxide   0.10361    0.03975   2.607 0.009231 **
## total.sulfur.dioxide -0.14512    0.04025  -3.605 0.000322 ***
## pH                   -0.48920    0.15624  -3.131 0.001774 **
```

```
## sulphates             0.78034    0.08160   9.563  < 2e-16 ***
## alcohol               3.05019    0.19197  15.889  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6462 on 1589 degrees of freedom
## Multiple R-squared:  0.3633, Adjusted R-squared:  0.3597
## F-statistic: 100.8 on 9 and 1589 DF,  p-value: < 2.2e-16
```

```
#LASSO regression
library(glmnet)
```

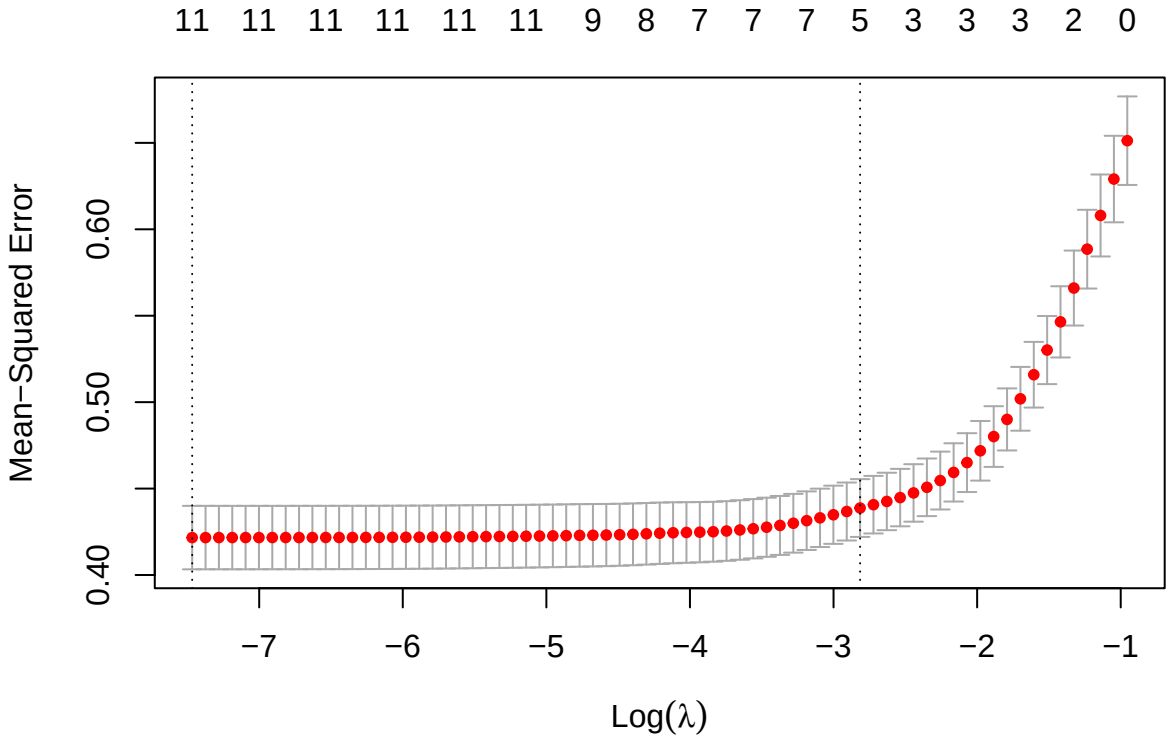
```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-8
```

```
redwine_predictors <- model.matrix(quality ~ ., data = redwine_logged)[, -1]
redwine_response <- redwine_logged$quality

set.seed(6020)
redwine_lasso<-cv.glmnet(redwine_predictors,redwine_response, alpha=1, standardize=TRUE)

#cross-validation curve
plot(redwine_lasso)
```




```
redwine_lambda<-redwine_lasso$lambda.min
```

```
redwine_lambda
```

```
## [1] 0.000571824
```

```
#lasso model with best lamda
redwine_lasso_best<-glmnet(redwine_predictors,redwine_response, alpha=1, lambda=redwine_lambda)

coef(redwine_lasso_best)
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                               s0
## (Intercept)           27.49470746
## fixed.acidity          0.40300196
## volatile.acidity      -0.50970616
## citric.acid           -0.36815041
## residual.sugar         0.08533159
## chlorides             -0.23623229
## free.sulfur.dioxide    0.09517200
## total.sulfur.dioxide  -0.14061771
## density              -28.41167189
## pH                    -0.34249823
## sulphates              0.81158177
## alcohol                2.75239859
```

- Perform hypothesis tests on coefficients

```
summary(redwine_back)
```

```
##
## Call:
## lm(formula = quality ~ fixed.acidity + volatile.acidity + citric.acid +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     pH + sulphates + alcohol, data = redwine_logged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.67779 -0.35087 -0.04455  0.44129  1.93469
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -0.57296    0.81576  -0.702 0.482557
## fixed.acidity          0.19709    0.13527   1.457 0.145290
## volatile.acidity      -0.51955    0.06176  -8.412  < 2e-16 ***
## citric.acid           -0.37933    0.18313  -2.071 0.038492 *
## chlorides             -0.24369    0.05777  -4.219  2.6e-05 ***
## free.sulfur.dioxide    0.10361    0.03975   2.607 0.009231 **
## total.sulfur.dioxide -0.14512    0.04025  -3.605 0.000322 ***
## pH                    -0.48920    0.15624  -3.131 0.001774 **
## sulphates              0.78034    0.08160   9.563  < 2e-16 ***
## alcohol                3.05019    0.19197  15.889  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6462 on 1589 degrees of freedom
## Multiple R-squared:  0.3633, Adjusted R-squared:  0.3597
## F-statistic: 100.8 on 9 and 1589 DF,  p-value: < 2.2e-16
```

Null hypothesis(H_0): The coefficient of the predictor is equal to zero, meaning that predictor has no linear effect on wine quality after adjusting for other predictors

Alternative hypothesis(H_a): The coefficient of the predictor is not equal to zero, meaning that the predictor does have a linear effect on wine quality.

Based on the summary, residual sugar and density were removed during the backward selection process. Among the remaining predictors, all except fixed.acidity showed statistically significant p-values (less than 0.05), providing evidence to reject the null hypothesis. This indicates that all predictors, except fixed.acidity, have a significant linear effect on wine quality. For fixed.acidity, the p-value was greater than or equal to 0.05, indicating insufficient evidence to reject the null hypothesis. Therefore, fixed.acidity does not appear to have a significant linear effect on wine quality after adjusting for the other predictors.

- Assess model performance with metrics (R^2, adjusted R^2, RMSE, etc.)

```
#R^2
summary(redwine_back)$r.squared
```

```
## [1] 0.3633211
```

```
#adjusted R^2
summary(redwine_back)$adj.r.squared
```

```
## [1] 0.359715
```

```
#RMSE
sqrt(mean(residuals(redwine_back)^2))
```

```
## [1] 0.6441756
```

Based on the model summaries, the R^2 value indicates the proportion of variability in wine quality that is explained by the predictors in the log-transformed and backward-selected model. A higher R^2 suggests that the model explains more of the outcome variability. In this case, the R^2 is 0.363, meaning approximately 36.3% of the variation in wine quality is explained by the model. The adjusted R^2 accounts for the number of predictors in the model and penalizes the inclusion of unnecessary variables. With an adjusted R^2 of 0.3597 — very close to the R^2 — this suggests that the model does not include many uninformative predictors. RMSE measures the typical prediction error of the model in the units of the outcome. A lower RMSE indicates better predictive accuracy. Here, the RMSE is 0.644, meaning the model's predictions deviate from the observed wine quality by approximately 0.64 units, on average.

- Validate your model using appropriate cross-validation techniques

```
#k-fold cross-validation
# Refit using glm
library(boot)
```

```
##
## Attaching package: 'boot'
```

```
## The following object is masked from 'package:car':
##
##     logit
```

```
redwine_back_glm <- glm(quality ~ fixed.acidity + volatile.acidity + citric.acid +
                          chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
                          pH + sulphates + alcohol,
                        data = redwine_logged)

# 10-fold cross-validation
set.seed(6020)
redwine_log_cross_result <- cv.glm(data = redwine_logged, glmfit = redwine_back_glm, K = 10)
```

```
#MSE and RMSE
redwine_log_cross_result$delta
```

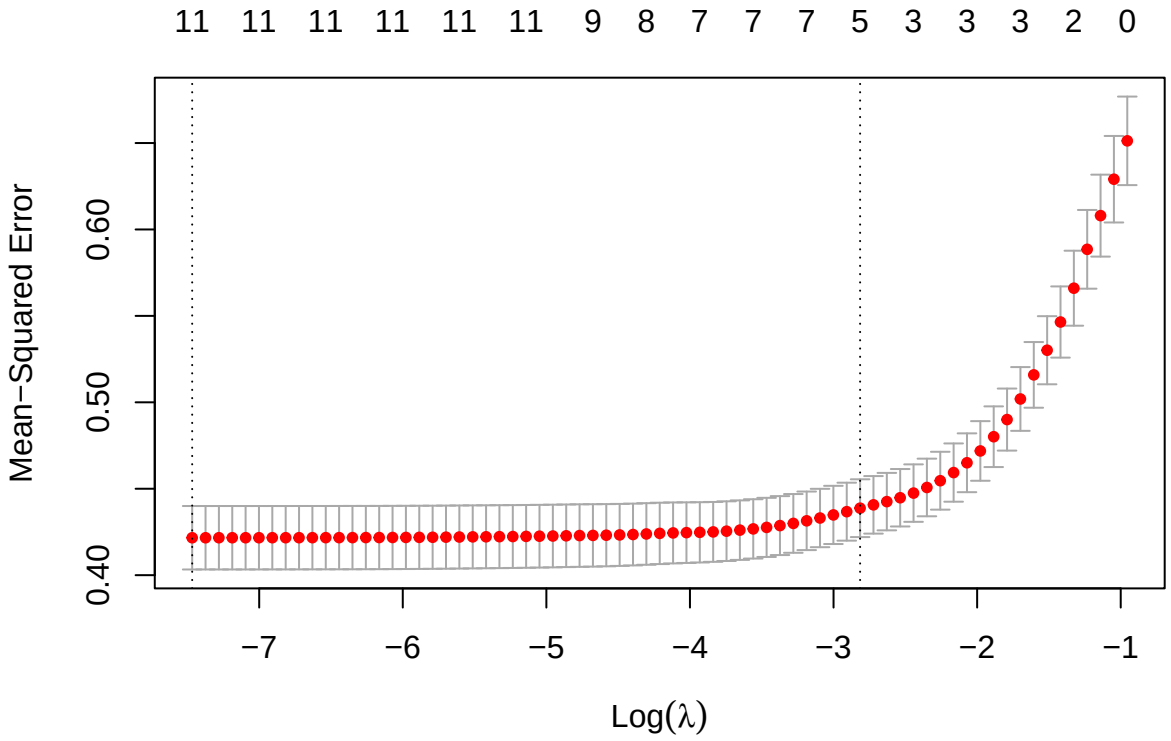
```
## [1] 0.4225564 0.4221550
```

```
sqrt(redwine_log_cross_result$delta)
```

```
## [1] 0.6500434 0.6497345
```

Based on the 10-fold cross-validation, the model's RMSE is approximately 0.65. This means the predicted wine quality is, on average, off by 0.65 units when applied to new data. Given that the outcome variable "quality" is an integer score ranging from 0 to 10 and reflects subjective sensory evaluation, a deviation of less than 1 point is relatively small. Therefore, the model demonstrates reasonably good predictive performance for this context.

```
#for Lasso Regression

#cross-validation curve
plot(redwine_lasso)
```




```
coef(redwine_lasso_best)
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                               s0
## (Intercept)           27.49470746
## fixed.acidity          0.40300196
## volatile.acidity      -0.50970616
## citric.acid           -0.36815041
## residual.sugar         0.08533159
## chlorides             -0.23623229
## free.sulfur.dioxide    0.09517200
## total.sulfur.dioxide  -0.14061771
```

```
## density              -28.41167189
## pH                    -0.34249823
## sulphates              0.81158177
## alcohol                2.75239859
```

```
redwine_lasso$cvm[which(redwine_lasso$lambda == redwine_lambda)]
```

```
## [1] 0.4216549
```

```
sqrt(redwine_lasso$cvm[which(redwine_lasso$lambda == redwine_lambda)])
```

```
## [1] 0.6493496
```

The cross-validation curve showed a relatively flat region around the optimal penalty value, lambda.min = 0.00057, indicating model stability — small changes in lambda around this value do not substantially affect the prediction performance. The corresponding root mean squared error (RMSE) is approximately 0.65, suggesting that, on average, the model's predictions deviate from the actual wine quality scores by about 0.65 units.

Both the backward selection model and the Lasso regression yielded the same cross-validated RMSE of approximately 0.65, suggesting comparable predictive performance. Given that wine quality is a subjective rating between 0 and 10, assessed by a panel of three, this deviation is relatively small. In this specific case, backward selection led to a slightly more parsimonious model by removing residual.sugar and density, whereas Lasso retained all predictors, likely due to the small optimal penalty value (lambda = 0.00057). This shows that although Lasso typically favors simpler models, the actual outcome depends on the chosen penalty strength.

Feature Impact Analysis

- Quantify and interpret the impact of each feature on the target

```
redwine_back
```

```
##
## Call:
## lm(formula = quality ~ fixed.acidity + volatile.acidity + citric.acid +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     pH + sulphates + alcohol, data = redwine_logged)
##
## Coefficients:
##          (Intercept)         fixed.acidity      volatile.acidity
##              -0.5730                0.1971               -0.5195
##          citric.acid             chlorides   free.sulfur.dioxide
##              -0.3793               -0.2437                0.1036
## total.sulfur.dioxide                   pH             sulphates
##              -0.1451               -0.4892                0.7803
##              alcohol
##               3.0502
```

```
fixed.acidity_interp=0.1971*0.01
volatile.acidity_interp= -0.5195*.01
citric.acid_interp=-0.3793*.01
chlorides_interp=-0.2437*.01
free.sulfur.dioxide_interp= 0.1036*.01
total.sulfur.dioxide_interp= -0.1451 *0.01
pH_interp= -0.4892*.01
sulphates_interp=  0.7803*.01
alcohol_interp=3.0502*.01
```

```
fixed.acidity_interp
```

```
## [1] 0.001971
```

```
volatile.acidity_interp
```

```
## [1] -0.005195
```

```
citric.acid_interp
```

```
## [1] -0.003793
```

```
chlorides_interp
```

```
## [1] -0.002437
```

```
free.sulfur.dioxide_interp
```

```
## [1] 0.001036
```

```
total.sulfur.dioxide_interp
```

```
## [1] -0.001451
```

```
pH_interp
```

```
## [1] -0.004892
```

```
sulphates_interp
```

```
## [1] 0.007803
```

```
alcohol_interp
```

```
## [1] 0.030502
```

fixed.acidity: A 1% increase in fixed.acidity is associated with an approximately 0.002 unit increase in average in predicted red wine quality, holding other variables constant.

volatile.acidity: A 1% increase in volatile.acidity is associated with an approximately 0.005 unit decrease in average in predicted red wine quality, holding other variables constant.

citric.acid: A 1% increase in citric.acid is associated with an approximately 0.004 unit decrease in average in predicted red wine quality, holding other variables constant.

chlorides: A 1% increase in chlorides is associated with an approximately 0.002 unit decrease in average in predicted red wine quality, holding other variables constant.

free.sulfur.dioxide: A 1% increase in free.sulfur.dioxide is associated with an approximately 0.001 unit increase in average in predicted red wine quality, holding other variables constant.

total.sulfur.dioxide: A 1% increase in total.sulfur.dioxide is associated with an approximately 0.001 unit decrease in average in predicted red wine quality, holding other variables constant.

pH: A 1% increase in pH is associated with an approximately 0.005 unit decrease in average in predicted red wine quality, holding other variables constant.

sulphates: A 1% increase in sulphates is associated with an approximately 0.008 unit increase in average in predicted red wine quality, holding other variables constant.

alcohol: A 1% increase alcohol is associated with an approximately 0.031 unit increase in average in predicted red wine quality, holding other variables constant.

- Provide confidence intervals for significant coefficients

```r
confint(redwine_back)
```

```
##                            2.5 %      97.5 %
## (Intercept)          -2.17302865  1.02711772
## fixed.acidity        -0.06822445  0.46241413
## volatile.acidity     -0.64069693 -0.39839995
## citric.acid          -0.73853564 -0.02011486
## chlorides            -0.35700032 -0.13038591
## free.sulfur.dioxide   0.02564429  0.18158351
## total.sulfur.dioxide -0.22407232 -0.06615848
## pH                   -0.79566082 -0.18273594
## sulphates             0.62028067  0.94040650
## alcohol               2.67364604  3.42672416
```

The confidence intervals for significant coefficients are provided above.

- Explain the practical significance of your findings in the context of the dataset

Final Report (PDF) containing:

- Introduction: dataset description and problem statement
- Methodology: techniques used and justification
- Results: findings from your analysis
- Discussion: interpretation of results and limitations
- Conclusion: summary and potential future work
- References: cite all sources used

*Introduction

Wine quality assessment plays a critical role in the certification process, which not only ensures product integrity by preventing illegal adulteration but also guarantees the quality of wines exported from Portugal. Portugal is among the top ten wine-exporting countries, and exports of its vinho verde wine have increased substantially in recent years (Cortez et al., 2009; Food and Agriculture Organization of the United Nations, 2024). In the current certification process, wine quality is evaluated using both physicochemical tests (e.g., pH, alcohol content, and sugar levels) and sensory analysis by trained human panels (Teranishi et al., 2012). While physicochemical properties are expected to influence sensory characteristics such as taste and aroma, the relationship between these measurable inputs and human perception remains complex and not fully understood. Sensory evaluation, in particular, is inherently subjective, and taste is considered one of the least understood human senses (Smith & Margolskee, 2001). This makes modeling wine quality based on chemical properties a challenging but important task, with potential applications in improving wine production, supporting certification decisions, and even informing marketing strategies (Turban, 2008). The dataset used in this project was sourced from Kaggle and originally made available by Cortez et al. (2009). Due to privacy and logistical constraints, only physicochemical variables (inputs) and the sensory-based quality score (output) are included. No information on grape variety, producer, or market pricing is available. The dataset contains 1599 observations of red wine samples, with 11 continuous numeric predictors including fixed acidity, alcohol, residual sugar, and others. The target variable is wine quality, rated on an integer scale from 0 (very poor) to 10 (excellent), and represents the median of scores given by at least three trained panelists during blind sensory evaluations. The objective of this project is to develop a linear regression model to predict wine quality using the available physicochemical attributes. Through this analysis, the goal is to better understand which features most significantly influence quality and to evaluate the predictive performance of the model under various statistical approaches.

*Methodology

This project utilized a publicly available dataset on Portuguese red wine samples originally provided by Cortez et al. (2009) and accessed via Kaggle. The dataset contains 1,599 observations and 11 continuous

physicochemical variables, such as pH, alcohol, and residual sugar, along with an integer outcome variable representing wine quality (rated on a scale from 0 to 10).

Exploratory Data Analysis (EDA)

Exploratory data analysis (EDA) was performed to understand the structure and distribution of the data. Histograms were used to examine the distribution and detect skewness in each variable, while boxplots helped identify potential outliers. Additionally, scatterplots with fitted linear trend lines were generated to visualize the relationships between each predictor and the wine quality score.

Regression Assumption Verification

A multiple linear regression model was initially fitted using all predictors. Key regression assumptions—linearity, normality of residuals, homoscedasticity, multicollinearity, and independence of observations—were systematically evaluated. Linearity was assessed using a residuals versus fitted values plot to check for randomness around the horizontal line at zero, along with component-plus-residual (partial residual) plots for each predictor to examine individual linear relationships. To evaluate normality of residuals, a histogram was used to assess symmetry and central tendency around zero, and a Q-Q plot with confidence bands was examined to detect deviations from normality, particularly in the tails. Homoscedasticity was tested using the Breusch-Pagan test. Multicollinearity was assessed using variance inflation factors (VIF), where values exceeding 5 indicated moderate multicollinearity. For the independence of observations, data collection procedures described in Cortez et al. (2009) were reviewed. Each row in the dataset corresponds to a unique wine sample, with no evidence of repeated measures or clustering. Sensory ratings were aggregated across assessors using the median, and the preprocessing ensured that each observation represented a distinct sample. Therefore, the assumption of independent observations was considered to hold.

Transformation and Model Comparison

Several predictors exhibited right-skewed distributions, contributing to violations of the normality and homoscedasticity assumptions. To address this, log transformations were applied to the skewed predictors, and a new regression model was fitted using the transformed variables. Model performance was compared between the original and transformed models using R^2, adjusted R^2, Akaike Information Criterion (AIC), and Bayesian Information Criterion (BIC). Although the transformation improved model diagnostics, the Breusch-Pagan test continued to indicate the presence of heteroscedasticity. Consequently, heteroscedasticity-consistent (robust) standard errors were applied to ensure valid statistical inference. While moderate multicollinearity was detected via variance inflation factors (VIF), no predictors were removed, as the primary objective was to develop a predictive model of wine quality using all available physicochemical variables. Even if correlated, these variables may retain predictive value, and excluding them could reduce model performance.

Variable Selection and Model Performance

For variable selection, two techniques were applied: backward stepwise selection using AIC and Lasso regression with 10-fold cross-validation to determine the optimal penalty term (λ). Hypothesis tests were conducted on the coefficients of the selected model to identify statistically significant predictors, using a significance level of $\alpha = 0.05$. Model performance was evaluated using R^2, adjusted R^2, RMSE, and 10-fold cross-validation.

Feature Impact

Feature importance was assessed by interpreting the magnitude and direction of the regression coefficients and by providing confidence intervals for statistically significant predictors. Since the model included log-transformed predictors, the coefficients were multiplied by 0.01 to approximate the change in wine quality associated with a 1% change in each predictor.

*Results

Exploratory Data Analysis (EDA)

Histograms of all red wine variables showed that alcohol, chlorides, citric acid, fixed acidity, free sulfur dioxide, residual sugar, sulphates, total sulfur dioxide, and volatile acidity were right-skewed. In contrast, density and

pH appeared approximately normally distributed. The outcome variable, quality, is discrete and clustered around scores of 5 and 6, resembling a bell-shaped distribution but not truly normal. Boxplots revealed outliers in alcohol, with two observations exceeding the upper whisker (defined as 1.5 times the interquartile range above the third quartile). Scatterplots of predictors versus wine quality indicated that alcohol, citric acid, fixed acidity, and sulphates were positively associated with wine quality. In contrast, chlorides, density, total sulfur dioxide, and volatile acidity showed negative associations. Free sulfur dioxide, pH, and residual sugar demonstrated little to no clear relationship with quality.

Regression Assumption Verification

Linearity

Linearity was assessed using the residuals versus fitted values plot, which displayed a striped pattern, and component-plus-residual plots for individual predictors. While mild curvature was observed for variables such as total sulfur dioxide and free sulfur dioxide, the overall patterns did not suggest strong nonlinearity. Therefore, the linearity assumption was considered reasonably satisfied.

Normality

Normality of residuals was evaluated using a histogram and a Q-Q plot with confidence bands. The histogram showed a roughly symmetric, bell-shaped distribution centered at zero. Although several residuals in the lower left tail fell outside the confidence band in the Q-Q plot, the majority of points were within the band, indicating only a mild violation of the normality assumption. Given this minor departure and the use of robust standard errors for inference, no further corrective action was taken.

Homoscedasticity

To assess homoscedasticity, the Breusch-Pagan test was conducted. The test returned a highly significant p-value ($p = 1.588e\text{-}13$), providing strong evidence that the residuals exhibited heteroscedasticity. As a result, the constant variance assumption was violated, and robust standard errors were used to ensure valid inference.

Multicollinearity

Multicollinearity was evaluated using the Variance Inflation Factor (VIF). Fixed acidity and density showed VIF values greater than 5, indicating moderate multicollinearity, while all other predictors were within acceptable ranges. Although multicollinearity can inflate standard errors and reduce the reliability of coefficient estimates, the decision was made to retain these predictors in the model because the primary objective of the project was prediction rather than inference. Variables with moderate collinearity may still hold valuable information for forecasting wine quality.

Independence

The independence assumption was examined using contextual information provided in the dataset documentation. Each observation corresponds to a unique wine sample, with no indication of repeated measures or clustering. The sensory scores were aggregated across assessors using the median, and data collection protocols ensured that each row represented an independently evaluated sample. Therefore, the assumption of independence was satisfied.

Transformation and Model Comparison

To address the violation of the homoscedasticity assumption, log transformations were applied to right-skewed predictors. Following this transformation, model performance showed a slight improvement: the adjusted R^2 increased from 0.3561 to 0.3598, the AIC decreased from 3164.28 to 3155.03, and the BIC decreased from 3234.18 to 3224.93. These changes suggest that the transformed model offers a marginally better fit while maintaining comparable model complexity. After the log transformation, the Breusch-Pagan (BP) test was conducted again, yielding a p-value of 1.018e-10. This result indicates that the homoscedasticity assumption still does not hold. Therefore, heteroscedasticity was addressed by applying both the log transformation and heteroscedasticity-consistent (robust) standard errors to ensure valid statistical inference. Under this correction, the predictors that remained statistically significant included volatile acidity, citric acid, chlorides,

free sulfur dioxide, total sulfur dioxide, sulphates, and alcohol. In contrast, predictors such as fixed acidity and density did not show strong evidence of association with wine quality.

Variable Selection and Model Performance

Backward stepwise selection removed two predictors—residual sugar and density—from the model. Among the remaining variables, all predictors except fixed acidity had p-values less than 0.05, providing sufficient evidence to reject the null hypothesis. This indicates that all remaining predictors, aside from fixed acidity, have a statistically significant linear association with wine quality. The p-value for fixed acidity was greater than or equal to 0.05, suggesting insufficient evidence to conclude that it has a meaningful effect after adjusting for other variables. The model's performance was evaluated using multiple metrics. The R^2 value of 0.363 indicates that approximately 36.3% of the variability in wine quality is explained by the selected predictors. The adjusted R^2 of 0.3597, which accounts for model complexity, is very close to the R^2, suggesting the model does not include unnecessary predictors. The root mean squared error (RMSE) of 0.644 indicates that the typical prediction deviates from the actual wine quality score by about 0.64 units. Using 10-fold cross-validation, the RMSE was approximately 0.65, further validating the model's generalization performance. Given that wine quality is a subjective rating on a 0–10 scale, typically evaluated by three panelists, this level of prediction error (less than 1 point) is considered reasonably accurate for practical purposes. For comparison, Lasso regression was also applied with 10-fold cross-validation. The optimal penalty value (lambda.min = 0.00057) was located in a relatively flat region of the cross-validation curve, indicating model stability—small changes in lambda do not significantly affect predictive performance. The Lasso model achieved the same cross-validated RMSE of approximately 0.65. While Lasso often leads to simpler models by shrinking coefficients of less important variables toward zero, in this case it retained all predictors. This outcome is likely due to the small optimal lambda value, which applied only minimal regularization. In contrast, backward selection produced a slightly more parsimonious model by excluding residual sugar and density. Both approaches yielded similar predictive accuracy, highlighting the robustness of the selected feature set.

Feature Impact

Among the predictors, volatile acidity had a statistically significant negative effect, where a 1% increase was associated with an approximate 0.005 unit decrease in predicted wine quality. This effect was supported by a 95% confidence interval of [–0.641, –0.398]. Similarly, citric acid showed a significant negative association, with a 1% increase linked to an approximate 0.004 unit decrease in wine quality (95% CI: [–0.739, –0.020]). Chlorides also demonstrated a statistically significant negative effect, where a 1% increase corresponded to an approximate 0.002 unit decrease in quality, with a confidence interval of [–0.357, –0.130]. On the other hand, several variables showed positive effects. A 1% increase in free sulfur dioxide was associated with an approximate 0.001 unit increase in predicted wine quality, supported by a 95% confidence interval of [0.026, 0.182]. Similarly, sulphates had a strong positive impact: a 1% increase was associated with an approximate 0.008 unit increase in predicted quality (95% CI: [0.620, 0.940]). Alcohol exhibited the strongest positive relationship with wine quality; a 1% increase in alcohol content was associated with an approximate 0.031 unit increase in quality, with a confidence interval of [2.674, 3.427]. Total sulfur dioxide and pH both showed statistically significant negative associations with wine quality. A 1% increase in total sulfur dioxide was associated with a 0.001 unit decrease (95% CI: [–0.224, –0.066]), while a 1% increase in pH led to a 0.005 unit decrease (95% CI: [–0.796, –0.183]). Finally, although fixed acidity showed a positive coefficient—indicating that a 1% increase was associated with an approximate 0.002 unit increase in predicted wine quality—the 95% confidence interval [–0.068, 0.462] included zero, suggesting that this effect was not statistically significant after adjusting for other predictors.

- Discussion

Model Performance and Feature Impact

In Cortez et al. (2009), three modeling approaches were tested to predict wine quality: Multiple Regression (MR), Neural Networks (NN), and Support Vector Machines (SVM). Among them, the MR model is the most comparable to our linear regression approach. The MR model achieved a Mean Absolute Deviation (MAD) of 0.50, indicating an average prediction error of 0.50 quality points. In our analysis, we applied a

log-transformed multiple linear regression model with backward selection, which resulted in a Root Mean Squared Error (RMSE) of 0.65 based on 10-fold cross-validation. Although RMSE and MAD are not directly equivalent, both are commonly used to measure average prediction error. RMSE penalizes larger errors more heavily than MAD (Chai & Draxler, 2014), but our result is roughly similar to the accuracy of the model reported in the 2009 study. In terms of feature importance, both analyses identified similar influential variables. Cortez et al. highlighted alcohol (4th), volatile acidity (6 th), and sulphates (1st) as top predictors. Our model similarly found alcohol (a 1% increase is associated with an average 0.031-unit increase in predicted quality), volatile acidity (0.005-unit decrease), and sulphates (0.008-unit increase) to be among the most impactful predictors, based on the log-transformed regression coefficients.

- Conclusion

The objective of this project was to develop a predictive model for Portuguese red wine quality based on physicochemical attributes. To do this, a multiple linear regression framework was employed, followed by log transformations of skewed predictors, diagnostic assessment of model assumptions, variable selection, and evaluation of predictive performance. Several predictors exhibited right-skewed distributions, prompting log transformations to address violations of normality and homoscedasticity assumptions. Although the Breusch-Pagan test continued to indicate heteroscedasticity, robust standard errors were used to ensure valid inference. Backward stepwise selection produced a more parsimonious model in this project, and both the backward-selected and Lasso regression models demonstrated comparable predictive performance, with RMSE values of approximately 0.65 under 10-fold cross-validation. Key predictors associated with wine quality included alcohol, volatile acidity, sulphates, and total sulfur dioxide—consistent with findings from Cortez et al. (2009). While the model demonstrated reasonable performance, there is still limitations. The assumption of linearity may restrict the model's flexibility. Future research could investigate non-linear approaches such as neural networks or support vector machines to capture more complex patterns in wine quality assessment.

- References

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems, 47(4), 547–553. https://doi.org/10.1016/j.dss.2009.05.016

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. Geoscientific Model Development, 7(3), 1247–1250. https://doi.org/10.5194/gmd-7-1247-2014

Food and Agriculture Organization of the United Nations. (2024). FAOSTAT. Www.fao.org. https://www.fao.org/faostat/en/#home

Smith, D. V., & Margolskee, R. F. (2001). Making Sense of Taste. Scientific American, 284(3), 32–39. https://www.jstor.org/stable/26059127

Teranishi, R., Wick, E. L., & Hornstein, I. (2012). Flavor Chemistry. Springer Science & Business Media.

Turban, E., Sharda, R., Aronson, J. E., & King, D. (2008). Business intelligence: A managerial approach (pp. 58-59). Upper Saddle River, NJ: Pearson Prentice Hall.