Wine Quality EDA

2025-05-01

Business Understanding

##What Makes a High-Quality Wine?

Grape Variety and Terroir Grape Variety: Different grape varieties possess unique characteristics in terms of flavor, aroma, acidity, and tannin levels. These traits, combined with the winemaker's skill, contribute to the overall style and quality of the wine.

Terroir: This French term encompasses the environmental factors that affect a crop's phenotype, including unique environment contexts, farming practices, and a crop's specific growth habitat. Terroir includes climate, soil, topography, and other factors that influence the grape's growth and, consequently, the wine's quality.

2. Viticultural Practices Canopy Management: Proper leaf and shoot management ensures optimal sunlight exposure and air circulation, reducing disease risk and promoting even ripening.

Yield Control: Limiting the number of grape clusters can lead to more concentrated flavors, enhancing wine quality.

Harvest Timing: Picking grapes at the right time ensures a balance between sugar, acidity, and phenolic maturity.

3. Winemaking Techniques Fermentation Control: Managing fermentation temperature and duration affects the development of desired flavors and aromas.

Use of Oak: Aging wine in oak barrels can impart additional flavors and tannins, contributing to complexity.

Malolactic Fermentation: This secondary fermentation can soften acidity and add buttery notes to the wine.

4. Chemical Composition Alcohol Content: Higher alcohol levels can enhance body and mouthfeel but must be balanced to avoid overpowering other flavors.

Acidity: Acidity provides freshness and structure; too little can make wine taste flat, while too much can make it sharp.

Tannins: These compounds add structure and astringency, important for red wines and aging potential.

Residual Sugar: The amount of sugar left after fermentation affects sweetness and balance.

5. Sensory Attributes Aroma and Flavor Complexity: High-quality wines exhibit a range of aromas and flavors that evolve over time.

Balance: A harmonious integration of all components—acidity, tannins, alcohol, and sweetness—is crucial.

Finish: The length of time flavors linger after swallowing; a longer finish is often associated with higher quality.

6. Aging Potential Wines with the right balance of acidity, tannins, and fruit concentration can develop more complex flavors over time, enhancing quality.

7. Absence of Faults High-quality wines are free from defects such as oxidation, cork taint, or excessive sulfur dioxide, which can negatively impact flavor and aroma.

##Here I am loading the dataset for white wine into R and seperating the commas into columns, and loading necessary libraries

```
setwd("C:/Users/19177/OneDrive/Desktop")


White_wine <- read.csv("winequality-white.csv", sep = ";")
head(White_wine)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1           7.0             0.27        0.36           20.7     0.045
## 2           6.3             0.30        0.34            1.6     0.049
## 3           8.1             0.28        0.40            6.9     0.050
## 4           7.2             0.23        0.32            8.5     0.058
## 5           7.2             0.23        0.32            8.5     0.058
## 6           8.1             0.28        0.40            6.9     0.050
##   free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 1                  45                  170  1.0010 3.00      0.45     8.8
## 2                  14                  132  0.9940 3.30      0.49     9.5
## 3                  30                   97  0.9951 3.26      0.44    10.1
## 4                  47                  186  0.9956 3.19      0.40     9.9
## 5                  47                  186  0.9956 3.19      0.40     9.9
## 6                  30                   97  0.9951 3.26      0.44    10.1
##   quality
## 1       6
## 2       6
## 3       6
## 4       6
## 5       6
## 6       6
```

```
# Install required packages if not already installed
if(!require(tidyverse)) install.packages("tidyverse")
```

```
## Loading required package: tidyverse
```

```
## Warning: package 'tidyverse' was built under R version 4.4.3
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```
## Warning: package 'tibble' was built under R version 4.4.3
```

```
## Warning: package 'tidyr' was built under R version 4.4.3
```

```
## Warning: package 'readr' was built under R version 4.4.3
```

```
## Warning: package 'purrr' was built under R version 4.4.3
```

```
## Warning: package 'dplyr' was built under R version 4.4.3
```

```
## Warning: package 'stringr' was built under R version 4.4.3
```

```
## Warning: package 'forcats' was built under R version 4.4.3
```

```
## Warning: package 'lubridate' was built under R version 4.4.3
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.4     ✓ tidyr     1.3.1
## ✓ purrr     1.0.4
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
if(!require(corrplot)) install.packages("corrplot")
```

```
## Loading required package: corrplot
```

```
## Warning: package 'corrplot' was built under R version 4.4.3
```

```
## corrplot 0.95 loaded
```

```
if(!require(ggplot2)) install.packages("ggplot2")
if(!require(dplyr)) install.packages("dplyr")
if(!require(knitr)) install.packages("knitr")
```

```
## Loading required package: knitr
```

```
## Warning: package 'knitr' was built under R version 4.4.3
```

```
if(!require(gridExtra)) install.packages("gridExtra")
```

```
## Loading required package: gridExtra
```

```
## Warning: package 'gridExtra' was built under R version 4.4.3
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
if(!require(car)) install.packages("car")
```

```
## Loading required package: car
```

```
## Warning: package 'car' was built under R version 4.4.3
```

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

```
## Warning: package 'carData' was built under R version 4.4.3
```

```
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```
if(!require(reshape2)) install.packages("reshape2")
```

```
## Loading required package: reshape2
```

```
## Warning: package 'reshape2' was built under R version 4.4.3
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
# Load the libraries
library(tidyverse)
library(corrplot)
library(ggplot2)
library(dplyr)
library(knitr)
library(gridExtra)
library(car)
library(reshape2)
```

##Loading and combining datasets

```
# Set your working directory if needed
setwd("C:/Users/19177/OneDrive/Desktop")
# Load the datasets
white_wine <- read.csv("winequality-white.csv", sep = ";")
red_wine <- read.csv("winequality-red.csv", sep = ";")

# Add a type variable to distinguish between red and white wine
white_wine$type <- "white"
red_wine$type <- "red"

# Combine both datasets
wine_data <- rbind(white_wine, red_wine)

# Examine the structure of the combined dataset
str(wine_data)
```

```
## 'data.frame':    6497 obs. of  13 variables:
##  $ fixed.acidity       : num  7 6.3 8.1 7.2 7.2 8.1 6.2 7 6.3 8.1 ...
##  $ volatile.acidity    : num  0.27 0.3 0.28 0.23 0.23 0.28 0.32 0.27 0.3 0.22 ...
##  $ citric.acid         : num  0.36 0.34 0.4 0.32 0.32 0.4 0.16 0.36 0.34 0.43 ...
##  $ residual.sugar      : num  20.7 1.6 6.9 8.5 8.5 6.9 7 20.7 1.6 1.5 ...
##  $ chlorides           : num  0.045 0.049 0.05 0.058 0.058 0.05 0.045 0.045 0.049 0.044 ...
##  $ free.sulfur.dioxide : num  45 14 30 47 47 30 30 45 14 28 ...
##  $ total.sulfur.dioxide: num  170 132 97 186 186 97 136 170 132 129 ...
##  $ density             : num  1.001 0.994 0.995 0.996 0.996 ...
##  $ pH                  : num  3 3.3 3.26 3.19 3.19 3.26 3.18 3 3.3 3.22 ...
##  $ sulphates           : num  0.45 0.49 0.44 0.4 0.4 0.44 0.47 0.45 0.49 0.45 ...
##  $ alcohol             : num  8.8 9.5 10.1 9.9 9.9 10.1 9.6 8.8 9.5 11 ...
##  $ quality             : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ type                : chr  "white" "white" "white" "white" ...
```

Data Understanding

When was the data acquired? - The UCI Wine Quality dataset does not specify exact collection dates. However, it was made publicly available through the UCI Machine Learning Repository and has been widely used since at least 2009. It is reasonable to assume that the data was collected sometime prior to that year.

Where was the data acquired? - The data was acquired from physicochemical tests conducted by the Portuguese “Vinho Verde” Wine Region. All wines originated from northern Portugal, specifically from the Minho region.

How was the data acquired? - The dataset was created using physicochemical (e.g., pH, alcohol) and sensory (quality scores) tests on red and white variants of the Portuguese “Vinho Verde” wine. The sensory quality scores were provided by at least three wine experts, and the physicochemical properties were measured using standard analytical chemistry techniques.

What are the attributes of this dataset? Description of each column in the dataset:

fixed acidity - Concentration of non-volatile acids (mainly tartaric). Helps preserve wine. volatile acidity - Amount of acetic acid. High levels give wine a vinegar taste. citric acid - A weak organic acid that adds freshness and flavor. residual sugar - Sugar left after fermentation; wines with >45g/L are considered sweet. chlorides - Salt content in wine. High values can indicate undesirable tastes. free sulfur dioxide - SO_2 in free form; acts as antioxidant and antimicrobial agent. total sulfur dioxide - Total SO_2, including both free and bound forms. Excess may affect taste. density - Mass-to-volume ratio; can relate to sugar and alcohol content. pH - Indicates acidity/basicity. Affects stability, color, and taste. sulphates - Additive for stabilization and as an antimicrobial. Higher levels may be perceived as bitter. alcohol - Alcohol percentage by volume. Often associated with wine body and warmth. quality - Quality score (0–10) based on sensory analysis by experts. This is the target variable.

What type of data do these attributes contain?

fixed acidity Numerical (continuous, ratio) volatile acidity Numerical (continuous, ratio) citric acid Numerical (continuous, ratio) residual sugar Numerical (continuous, ratio) chlorides Numerical (continuous, ratio) free sulfur dioxide Numerical (continuous, ratio) total sulfur dioxide Numerical (continuous, ratio) density Numerical (continuous, ratio) pH Numerical (continuous, interval) sulphates Numerical (continuous, ratio) alcohol Numerical (continuous, ratio) quality Ordinal (integers from 0 to 10)

#Data exploration Summary statistics and check for missing values

```
# Summary statistics
summary(wine_data)
```

```
##  fixed.acidity    volatile.acidity  citric.acid     residual.sugar
##  Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600
##  1st Qu.: 6.400   1st Qu.:0.2300   1st Qu.:0.2500   1st Qu.: 1.800
##  Median : 7.000   Median :0.2900   Median :0.3100   Median : 3.000
##  Mean   : 7.215   Mean   :0.3397   Mean   :0.3186   Mean   : 5.443
##  3rd Qu.: 7.700   3rd Qu.:0.4000   3rd Qu.:0.3900   3rd Qu.: 8.100
##  Max.   :15.900   Max.   :1.5800   Max.   :1.6600   Max.   :65.800
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.00900   Min.   :  1.00      Min.   :  6.0        Min.   :0.9871
##  1st Qu.:0.03800   1st Qu.: 17.00      1st Qu.: 77.0        1st Qu.:0.9923
##  Median :0.04700   Median : 29.00      Median :118.0        Median :0.9949
##  Mean   :0.05603   Mean   : 30.53      Mean   :115.7        Mean   :0.9947
##  3rd Qu.:0.06500   3rd Qu.: 41.00      3rd Qu.:156.0        3rd Qu.:0.9970
##  Max.   :0.61100   Max.   :289.00      Max.   :440.0        Max.   :1.0390
##        pH          sulphates         alcohol         quality
##  Min.   :2.720   Min.   :0.2200   Min.   : 8.00   Min.   :3.000
##  1st Qu.:3.110   1st Qu.:0.4300   1st Qu.: 9.50   1st Qu.:5.000
##  Median :3.210   Median :0.5100   Median :10.30   Median :6.000
##  Mean   :3.219   Mean   :0.5313   Mean   :10.49   Mean   :5.818
##  3rd Qu.:3.320   3rd Qu.:0.6000   3rd Qu.:11.30   3rd Qu.:6.000
##  Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :9.000
##      type
##  Length:6497
##  Class :character
##  Mode  :character
##
##
##
```

```
# Check for 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
##                 type
##                    0
```

```
# Get dimensions of datasets
cat("White wine dataset dimensions:", dim(white_wine), "\n")
```

```
## White wine dataset dimensions: 4898 13
```

```
cat("Red wine dataset dimensions:", dim(red_wine), "\n")
```

```
## Red wine dataset dimensions: 1599 13
```

```
cat("Combined dataset dimensions:", dim(wine_data), "\n")
```

```
## Combined dataset dimensions: 6497 13
```

```
# Check class distribution (quality) in both datasets
table(white_wine$quality)
```

```
##
##    3    4    5    6    7    8    9
##   20  163 1457 2198  880  175    5
```

```
table(red_wine$quality)
```

```
##
##   3   4   5   6   7   8
##  10  53 681 638 199  18
```

```
table(wine_data$quality)
```

```
##
##    3    4    5    6    7    8    9
##   30  216 2138 2836 1079  193    5
```

Converting wine quality into a factor for analysis

```
# Convert quality to a factor for classification purposes
wine_data$quality_category <- ifelse(wine_data$quality >= 7, "high",
                                     ifelse(wine_data$quality <= 4, "low", "medium"))
wine_data$quality_category <- factor(wine_data$quality_category,
                                          levels = c("low", "medium", "high"))

# Check the distribution of quality categories
table(wine_data$quality_category)
```

```
##
##    low medium   high
##    246   4974   1277
```

```
table(wine_data$type, wine_data$quality_category)
```

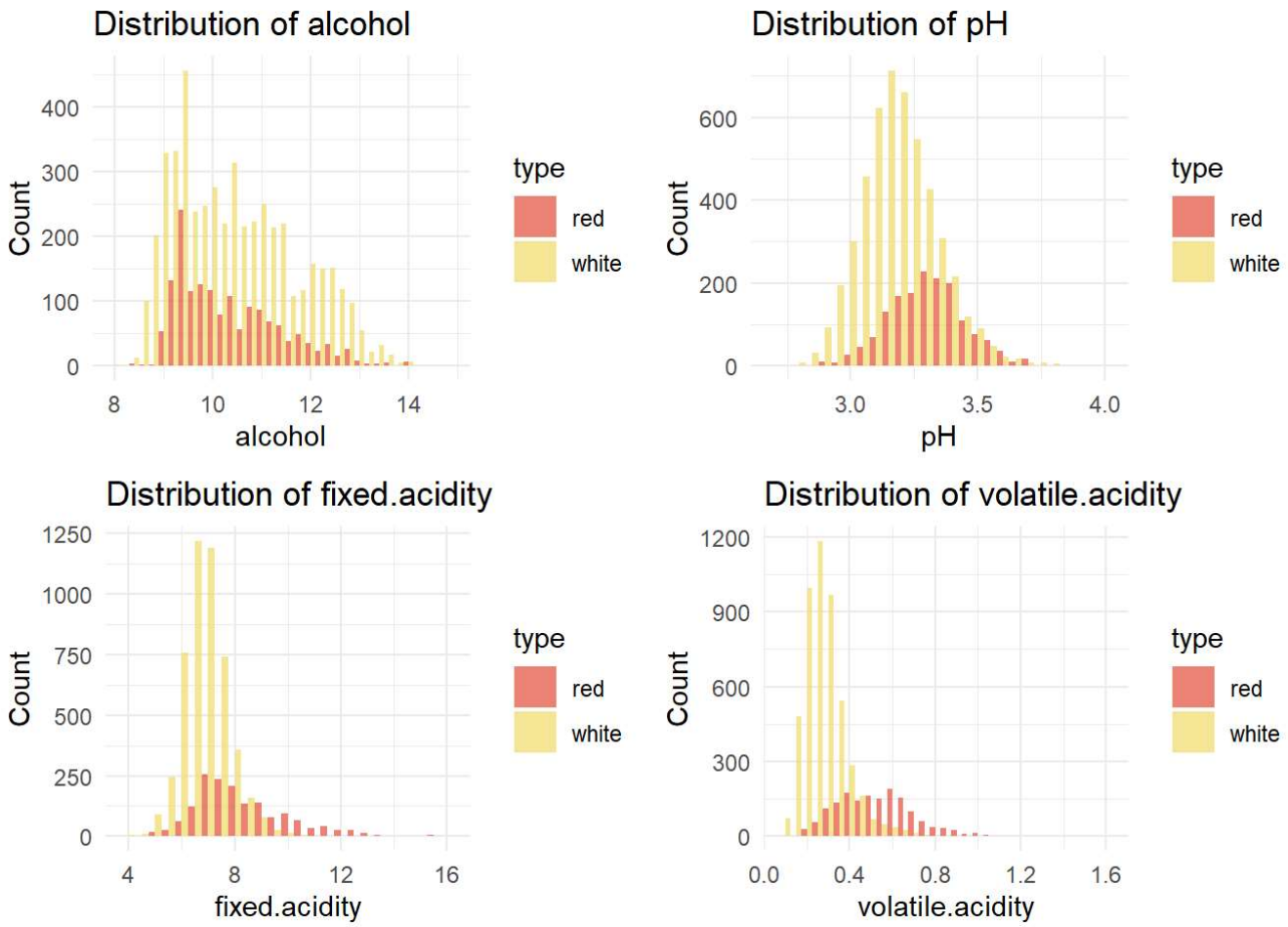
```
##
##          low medium high
##   red     63   1319  217
##   white  183   3655 1060
```

Examining distribution of values with histograms

```
# Function to create histogram for a specific variable
create_histogram <- function(data, variable, binwidth = NULL, title = NULL) {
  if(is.null(title)) title <- paste("Distribution of", variable)

  ggplot(data, aes_string(x = variable, fill = "type")) +
    geom_histogram(position = "dodge", alpha = 0.7, binwidth = binwidth) +
    labs(title = title, x = variable, y = "Count") +
    theme_minimal() +
    scale_fill_manual(values = c("red" = "#E74C3C", "white" = "#F7DC6F"))
}

# Create histograms for key variables
p1 <- create_histogram(wine_data, "alcohol", binwidth = 0.2)
```

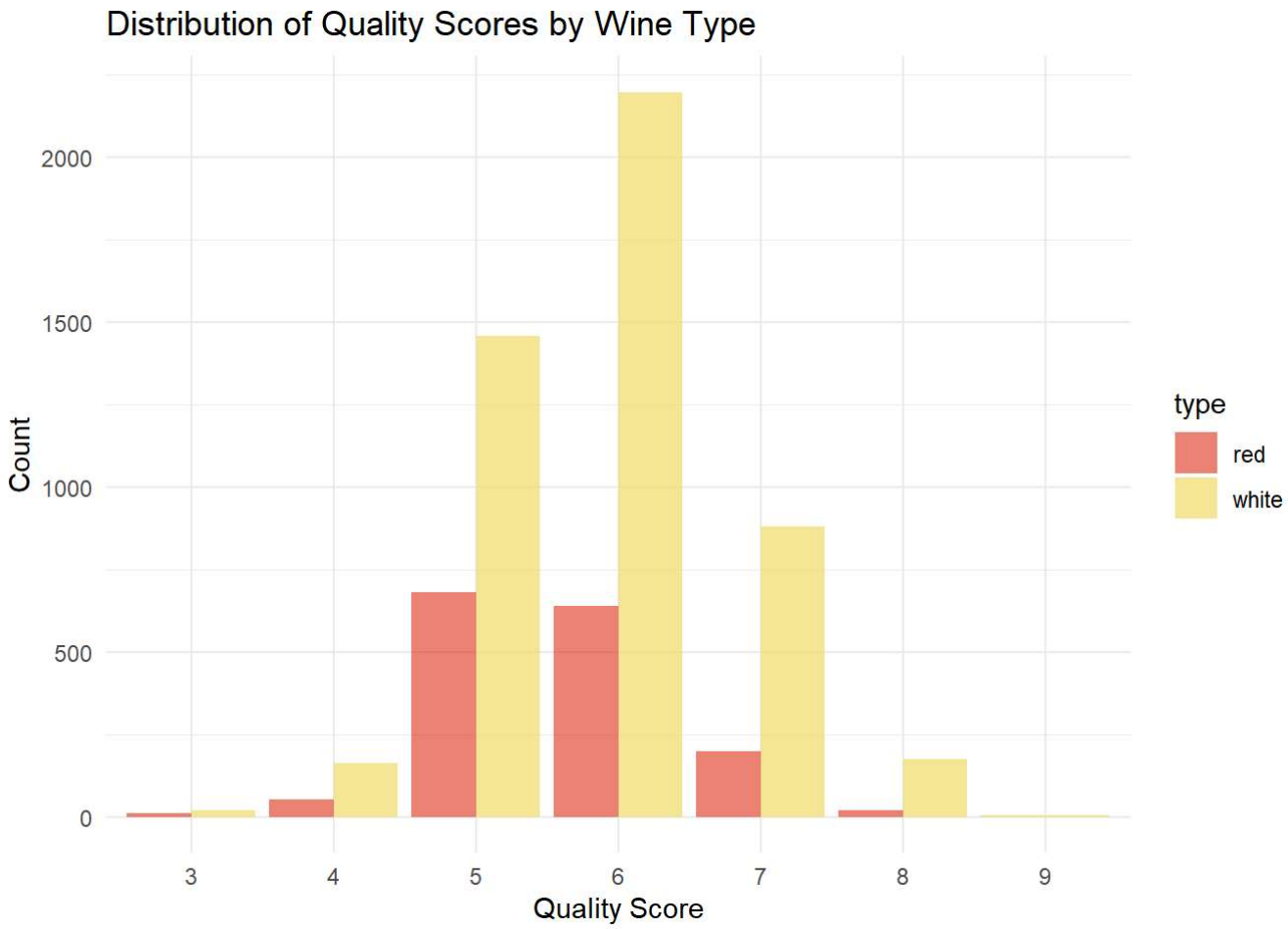
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## ℹ Please use tidy evaluation idioms with `aes()`.
## ℹ See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
p2 <- create_histogram(wine_data, "pH", binwidth = 0.05)
p3 <- create_histogram(wine_data, "fixed.acidity", binwidth = 0.5)
p4 <- create_histogram(wine_data, "volatile.acidity", binwidth = 0.05)

# Display plots in a grid
grid.arrange(p1, p2, p3, p4, ncol = 2)
```



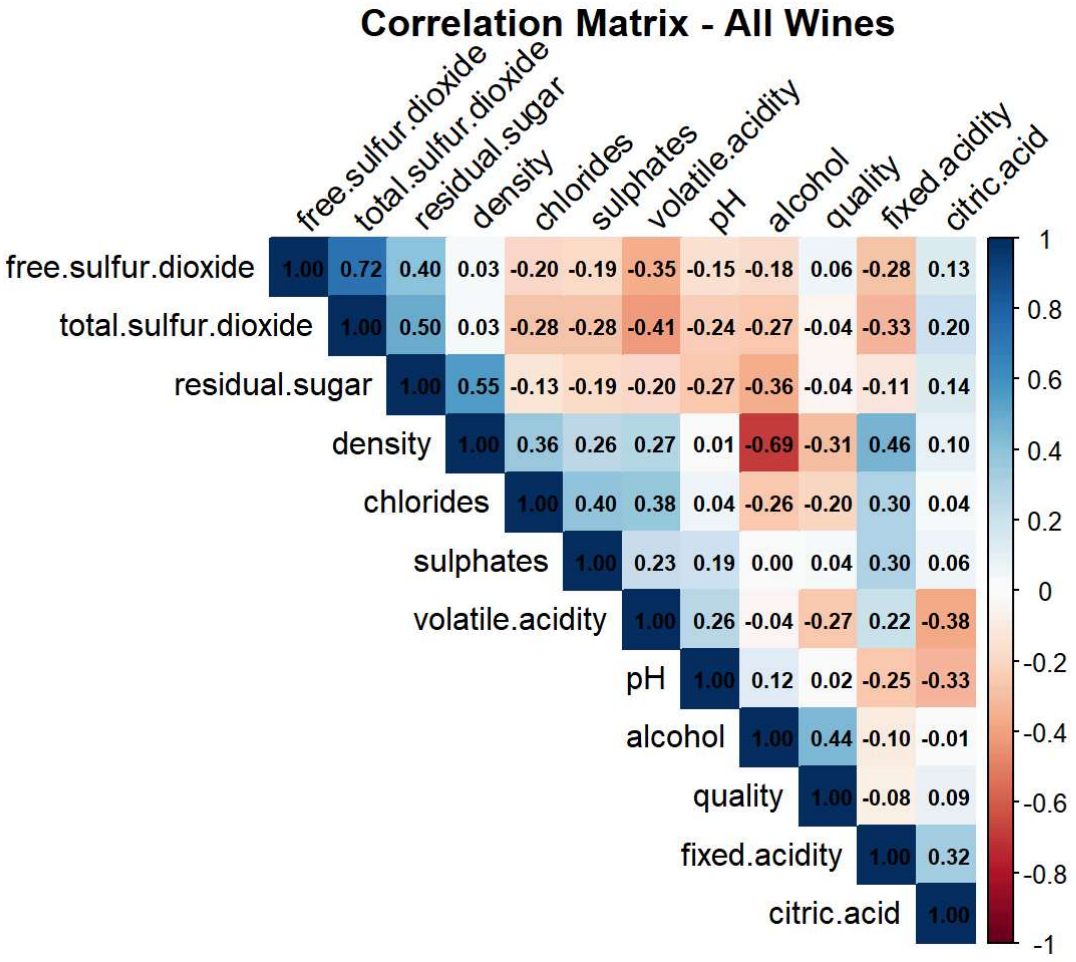
Quality distribution by wine type

```
# Visualize the distribution of quality scores by wine type
ggplot(wine_data, aes(x = factor(quality), fill = type)) +
  geom_bar(position = "dodge", alpha = 0.7) +
  labs(title = "Distribution of Quality Scores by Wine Type",
       x = "Quality Score", y = "Count") +
  theme_minimal() +
  scale_fill_manual(values = c("red" = "#E74C3C", "white" = "#F7DC6F"))
```



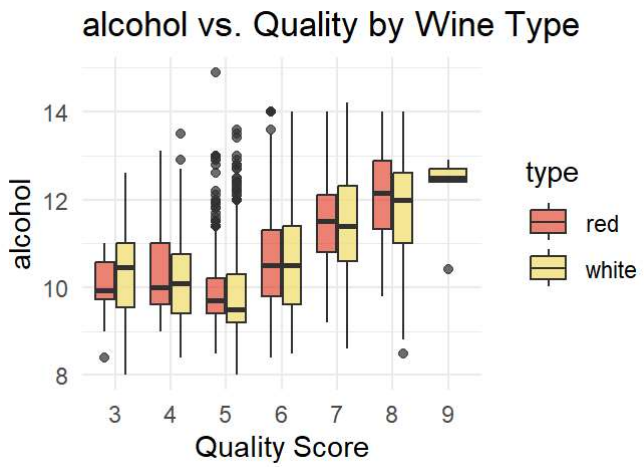
Correlation analysis

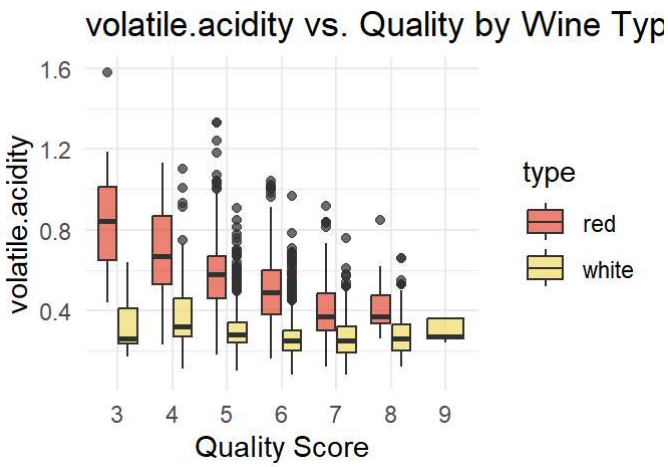
```
# Function to create correlation matrix
plot_correlation <- function(data, title) {
  # Remove non-numeric columns
  data_numeric <- data %>% select_if(is.numeric)

  # Calculate correlation matrix
  corr_matrix <- cor(data_numeric)

  # Plot correlation matrix
  corrplot(corr_matrix, method = "color", type = "upper", order = "hclust",
           tl.col = "black", tl.srt = 45, addCoef.col = "black",
           number.cex = 0.7, title = title, mar = c(0, 0, 1, 0))
}

# Plot correlation matrices
par(mfrow = c(1, 1))
plot_correlation(wine_data, "Correlation Matrix - All Wines")
```

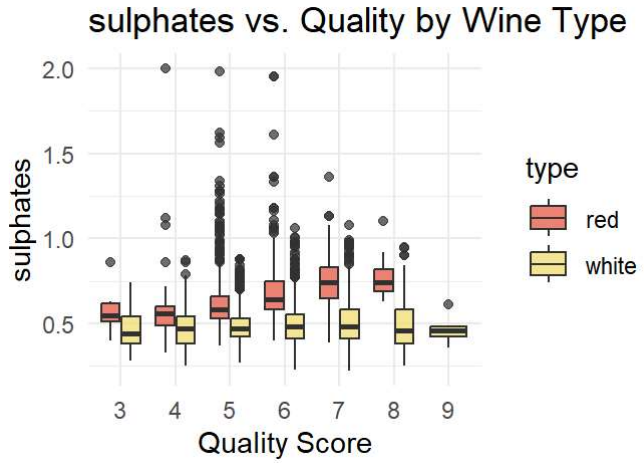


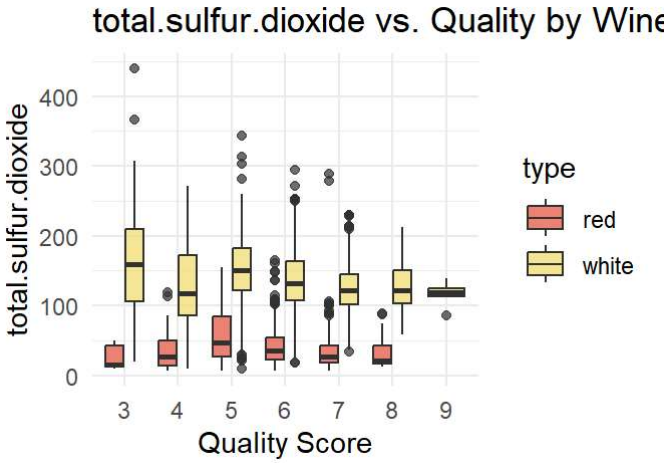
Analyzing quality according to key features

```
# Function to create boxplots
create_boxplot <- function(data, y_var, title = NULL) {
  if(is.null(title)) title <- paste(y_var, "vs. Quality by Wine Type")

  ggplot(data, aes_string(x = "factor(quality)", y = y_var, fill = "type")) +
    geom_boxplot(alpha = 0.7) +
    labs(title = title, x = "Quality Score", y = y_var) +
    theme_minimal() +
    scale_fill_manual(values = c("red" = "#E74C3C", "white" = "#F7DC6F"))
}

# Create boxplots for key variables vs quality
p5 <- create_boxplot(wine_data, "alcohol")
p6 <- create_boxplot(wine_data, "volatile.acidity")
p7 <- create_boxplot(wine_data, "sulphates")
p8 <- create_boxplot(wine_data, "total.sulfur.dioxide")

# Display boxplots in a grid
grid.arrange(p5, p6, p7, p8, ncol = 2)
```









Statistical analysis of red and white wine differences

```r
# T-tests for key variables between red and white wines
t_test_results <- data.frame(
  Variable = character(),
  t_value = numeric(),
  p_value = numeric(),
  significant = character(),
  stringsAsFactors = FALSE
)

variables_to_test <- c("fixed.acidity", "volatile.acidity", "citric.acid", "residual.sugar",
                       "chlorides", "free.sulfur.dioxide", "total.sulfur.dioxide",
                       "density", "pH", "sulphates", "alcohol", "quality")

for(var in variables_to_test) {
  t_result <- t.test(red_wine[[var]], white_wine[[var]])
  t_test_results <- rbind(t_test_results, data.frame(
    Variable = var,
    t_value = round(t_result$statistic, 3),
    p_value = round(t_result$p.value, 5),
    significant = ifelse(t_result$p.value < 0.05, "Yes", "No"),
    stringsAsFactors = FALSE
  ))
}

# Display t-test results
kable(t_test_results, caption = "T-test Results: Red vs. White Wine")
```

T-test Results: Red vs. White Wine

	Variable	t_value	p_value	significant
t	fixed.acidity	32.423	0.00000	Yes
t1	volatile.acidity	53.059	0.00000	Yes
t2	citric.acid	-12.229	0.00000	Yes
t3	residual.sugar	-47.802	0.00000	Yes
t4	chlorides	34.240	0.00000	Yes
t5	free.sulfur.dioxide	-54.428	0.00000	Yes
t6	total.sulfur.dioxide	-89.872	0.00000	Yes
t7	density	42.709	0.00000	Yes
t8	pH	27.775	0.00000	Yes
t9	sulphates	37.056	0.00000	Yes
t10	alcohol	-2.859	0.00428	Yes
t11	quality	-10.149	0.00000	Yes

Linear regression analysis

```
# setwd("path/to/your/data")
setwd("C:/Users/19177/OneDrive/Desktop")
# Load the datasets
white_wine <- read.csv("winequality-white.csv", sep = ";")
red_wine <- read.csv("winequality-red.csv", sep = ";")

# Add type labels
red_wine$type <- "red"
white_wine$type <- "white"

# Combine into one dataset
wine_data <- rbind(red_wine, white_wine)
# Fit linear regression model for quality
model_all <- lm(quality ~ . - type, data = wine_data)

model_summary <- summary(model_all)

# Display regression results
model_summary
```

```
##
## Call:
## lm(formula = quality ~ . - type, data = wine_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7569 -0.4597 -0.0412  0.4694  2.9907
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.576e+01  1.189e+01   4.688 2.81e-06 ***
## fixed.acidity         6.768e-02  1.557e-02   4.346 1.41e-05 ***
## volatile.acidity     -1.328e+00  7.737e-02 -17.162  < 2e-16 ***
## citric.acid          -1.097e-01  7.962e-02  -1.377    0.168
## residual.sugar        4.356e-02  5.156e-03   8.449  < 2e-16 ***
## chlorides            -4.837e-01  3.327e-01  -1.454    0.146
## free.sulfur.dioxide   5.970e-03  7.511e-04   7.948 2.22e-15 ***
## total.sulfur.dioxide -2.481e-03  2.767e-04  -8.969  < 2e-16 ***
## density              -5.497e+01  1.214e+01  -4.529 6.04e-06 ***
## pH                    4.393e-01  9.037e-02   4.861 1.20e-06 ***
## sulphates             7.683e-01  7.612e-02  10.092  < 2e-16 ***
## alcohol               2.670e-01  1.673e-02  15.963  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7353 on 6485 degrees of freedom
## Multiple R-squared:  0.2921, Adjusted R-squared:  0.2909
## F-statistic: 243.3 on 11 and 6485 DF,  p-value: < 2.2e-16
```

```
# Check for multicollinearity
vif_results <- vif(model_all)
vif_data <- data.frame(
  Variable = names(vif_results),
  VIF = round(vif_results, 2),
  stringsAsFactors = FALSE
)
kable(vif_data, caption = "Variance Inflation Factors")
```

Variance Inflation Factors

	Variable	VIF
fixed.acidity	fixed.acidity	4.90
volatile.acidity	volatile.acidity	1.95
citric.acid	citric.acid	1.61
residual.sugar	residual.sugar	7.23
chlorides	chlorides	1.63
free.sulfur.dioxide	free.sulfur.dioxide	2.14
total.sulfur.dioxide	total.sulfur.dioxide	2.94
density	density	15.91
pH	pH	2.54
sulphates	sulphates	1.54
alcohol	alcohol	4.78

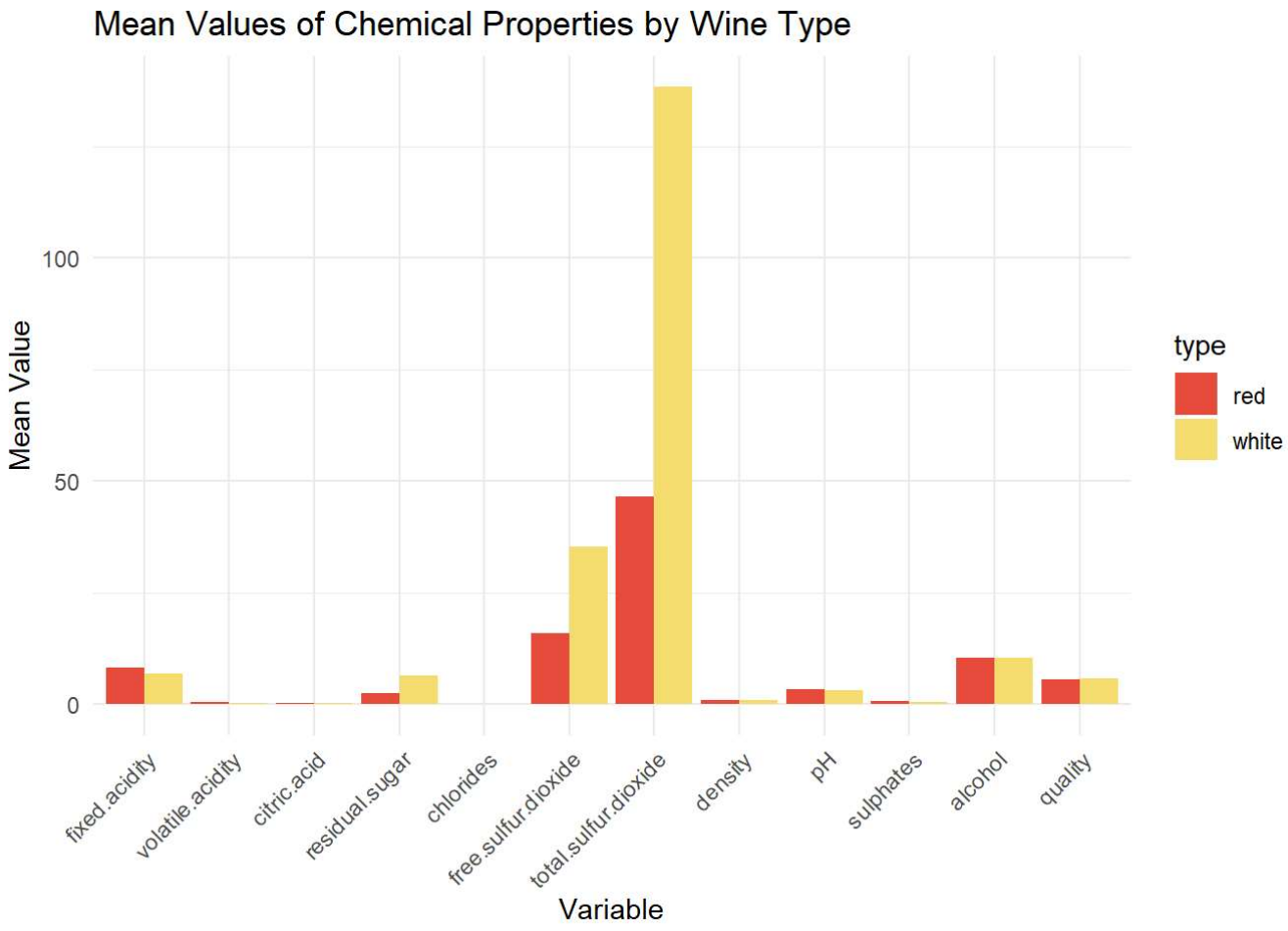
```
# Fit separate models for red and white wines


model_red <- lm(quality ~ . , data = red_wine[, !(names(red_wine) %in% c("type"))])

model_white <- lm(quality ~ . , data = white_wine[, !(names(white_wine) %in% c("type"))])


# Compare key coefficients
coef_comparison <- data.frame(
  Variable = names(coef(model_all))[2:length(coef(model_all))],
  All_Wines = round(coef(model_all)[2:length(coef(model_all))], 4),
  Red_Wine = round(coef(model_red)[2:length(coef(model_red))], 4),
  White_Wine = round(coef(model_white)[2:length(coef(model_white))], 4),
  stringsAsFactors = FALSE
)

kable(coef_comparison, caption = "Comparison of Regression Coefficients")
```

Comparison of Regression Coefficients

	Variable	All_Wines	Red_Wine	White_Wine
fixed.acidity	fixed.acidity	0.0677	0.0250	0.0655
volatile.acidity	volatile.acidity	-1.3279	-1.0836	-1.8632
citric.acid	citric.acid	-0.1097	-0.1826	0.0221
residual.sugar	residual.sugar	0.0436	0.0163	0.0815
chlorides	chlorides	-0.4837	-1.8742	-0.2473
free.sulfur.dioxide	free.sulfur.dioxide	0.0060	0.0044	0.0037
total.sulfur.dioxide	total.sulfur.dioxide	-0.0025	-0.0033	-0.0003
density	density	-54.9669	-17.8812	-150.2842
pH	pH	0.4393	-0.4137	0.6863
sulphates	sulphates	0.7683	0.9163	0.6315
alcohol	alcohol	0.2670	0.2762	0.1935

Key differences by wine type

```
# Calculate mean values for key variables by wine type
wine_means <- wine_data %>%
  group_by(type) %>%
  summarize(across(c(fixed.acidity, volatile.acidity, citric.acid, residual.sugar,
                     chlorides, free.sulfur.dioxide, total.sulfur.dioxide,
                     density, pH, sulphates, alcohol, quality), mean))

# Reshape data for visualization
wine_means_long <- melt(wine_means, id.vars = "type")

# Plot mean values by wine type
ggplot(wine_means_long, aes(x = variable, y = value, fill = type)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Mean Values of Chemical Properties by Wine Type",
       x = "Variable", y = "Mean Value") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_manual(values = c("red" = "#E74C3C", "white" = "#F7DC6F"))
```


Mean Values of Chemical Properties by Wine Type
Mean Value
100
50
0
fixed.acidity
volatile.acidity
citric.acid
residual.sugar
chlorides
free.sulfur.dioxide
total.sulfur.dioxide
density
pH
sulphates
alcohol
quality
Variable
type
red
white