University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Introduction to Data Sciences
Prof. Dr.-Ing. Joachim Schwarz

Statistical Analysis and Predictive Modeling of Wine Quality

Name: Subash Karapparambu Suresh Kumar
Matr.-Nr.: 7026794

Name: Aishwarya Jadhav
Matr.-Nr.: 7026164

Submission date: 30 June 2025

Contents

List of Figures

List of Tables

1 Introduction

Wine has been produced and consumed for thousands of years and remains one of the most valued beverages in global culture and commerce[Jackson(2020)]. With the increasing scale of wine production and market competition, there is growing interest in understanding and improving wine quality through scientific means[González-Barreiro et al.(2015)]. Traditionally, the evaluation of wine quality has relied on subjective sensory assessments by expert panels. While this human judgment remains valuable, it can be inconsistent and costly to maintain. The emergence of data science and machine learning now enables objective, reproducible, and data-driven evaluation of wine based on its measurable chemical and physical properties. Such methods not only offer faster quality control but also reveal deeper patterns and relationships between chemical composition and perceived wine quality. This intersection of chemistry, statistics, and computer science allows producers to optimize manufacturing processes, while also providing consumers with more transparent indicators of product quality[Cortez et al.(2009), Torgo(2017)]. This project aims to apply a combination of exploratory data analysis, statistical testing, regression modeling, classification techniques, and dimensionality reduction to a comprehensive dataset of red and white wines. The analysis is conducted in the R programming environment[R Core Team(2024)], leveraging modern data science workflows to extract insights and build predictive models.

2 Motivation

Wine quality is defined by chemical and sensory properties that, in lieu of statistical modeling, can be difficult to interpret. A winemaker can analyze data to find out which factors most affect the quality to produce consistently[Cortez et al.(2009)]. Consumers, on the other hand, will use such insights to choose wisely.

The motivation of this project includes:

- Discovering patterns in the wine dataset[Cortez et al.(2009)].
- Using machine learning techniques for the classification of wine types and the determination of wine quality[Atalay et al.(2021)].
- Validating statistical assumptions in order to draw relevant conclusions[Field(2013)].
- Reducing complexity through dimensionality reduction techniques, such as factor analysis[Vidal et al.(2015)].

3 List of Abbreviations

Table 1: Abbreviations and Key Terms

Abbreviation / Term	**Meaning / Explanation**
SD	Standard Deviation
Mean	Average value
Min, Max	Minimum and maximum values
Quartiles	Lower quartile (Q1), Median (Q2), Upper quartile (Q3)
Missing values	Count of missing data points
Skewness	Measure of asymmetry of distribution
Outliers	Extreme values identified from boxplots
t-test	Statistical test comparing means between two groups
Welch t-test	Variant of t-test not assuming equal variances
Shapiro-Wilk test	Test for normality of samples
Variance test	Test for equality of variances
lm	Linear regression model
Residuals	Differences between observed and predicted values
Linearity	Assumption that predictors have linear relationships
Homoscedasticity	Assumption of constant variance of residuals
Normality	Residuals follow a normal distribution
Diagnostic plots	Plots used to check regression assumptions
Logistic regression	Regression for binary classification
Label	Target variable coded as 0/1 (e.g., good vs bad)
Predicted class	Classification predicted by model
Confusion matrix	Table comparing predicted and actual classifications
ROC curve	Receiver Operating Characteristic curve
AUC	Area Under the ROC Curve, measures model performance
variety_bin	Binary coding of wine color (red=1, white=0)
Train/Test split	Division of data into training and validation subsets
glm	Generalized linear model (used here for logistic regression)
Factor analysis	Technique to reduce variables into latent factors
MSA	Measure of Sampling Adequacy (overall factorability)
MSAi	Individual MSA for each variable
KMO test	Kaiser-Meyer-Olkin test for sampling adequacy
Eigenvalues	Variance explained by each factor
Scree plot	Plot of eigenvalues to determine number of factors
Varimax	Orthogonal rotation method for factor loadings
SS loadings	Sum of squared factor loadings (variance explained by factors)
Proportion Var	Variance proportion explained by each factor
Cumulative Var	Total variance explained by all factors combined
Factor loadings	Correlations between variables and underlying factors

4 Task 1: Exploratory Data Analysis

4.1 1a. Descriptive Statistics of Metric and Categorical Variables

In this task, the wine dataset was first loaded using standard R functions such as `read.csv()`. Distribution Parameters, including the mean, standard deviation, minimum, lower quartile (Q1), median, upper quartile (Q3), and maximum, were calculated for all variables. For categorical variables, frequency distributions were generated to display the count of each category. The presence of missing values was also examined and reported for all variables. (Stats and R, 2020)

Descriptive results are presented in Tables 2 and 3. No missing values were found.

Table 2: Summary statistics of wine dataset variables

Variable	Mean	SD	Min	Lower Quartile	Median	Upper Quartile	Max	Skewness
fixed.acidity	7.22	1.30	3.80	6.40	7.00	7.70	15.90	1.72
volatile.acidity	0.34	0.16	0.08	0.23	0.29	0.40	1.58	1.49
citric.acid	0.32	0.15	0.00	0.25	0.31	0.39	1.66	0.47
residual.sugar	5.44	4.76	0.60	1.90	3.00	8.10	65.80	1.43
chlorides	0.06	0.04	0.01	0.03	0.05	0.07	0.61	5.68
free.sulfur.dioxide	30.53	17.75	1.00	17.00	29.00	41.00	289.00	1.22
total.sulfur.dioxide	115.74	56.52	6.00	77.00	118.00	156.00	440.00	0.50
density	0.99	0.002	0.99	0.99	0.99	0.99	1.00	0.03
pH	3.22	0.16	2.72	3.11	3.21	3.32	4.01	0.39
sulphates	0.53	0.15	0.22	0.43	0.51	0.60	2.00	1.49
alcohol	10.49	1.19	8.00	9.50	10.30	11.30	14.90	0.87
quality	5.82	0.87	3.00	5.00	6.00	6.00	9.00	0.19

Table 3: Frequencies of categorical variables

Variety	Count
Red	1599
White	4898

Quality Score	Count
3	30
4	216
5	2138
6	2836
7	1079
8	193
9	5

Analysis and Interpretation

The dataset contains 6,497 wine records with no missing values.

Summary Statistics of Numeric Variables

Summary statistics were calculated for 12 numeric variables. Fixed acidity ranges from about 3.8 to 15.9, with an average around 7.22. Volatile acidity and residual sugar show

right-skewed patterns, meaning their highest values are much larger than their averages. Free and total sulfur dioxide levels vary significantly between samples. Alcohol content ranges from 8% to 14.9%, with an average near 10.49%. Quality scores range from 3 to 9, with a median value of 6.

Frequency Distribution of Categorical Variables

There are 1,599 red wines and 4,898 white wines in the dataset. Most quality ratings are around 5 or 6, accounting for over 75% of the samples. Very low or very high quality scores (3, 8, or 9) are uncommon.

Missing Values

No missing data was found in any of the variables.

4.2 1b. Skewness and Outlier Detection

Task 1b: Interpretation of Graphical Analysis, Skewness and Outliers

In this task, histograms and boxplots were created for all 12 numerical variables in the wine dataset to visually assess distribution shapes and identify potential outliers. Additionally, the skewness coefficient was calculated for each variable using the `skewness()` function from the `e1071` package in R. To fulfil Task 1b more efficiently, the R code was adapted with the help of `ChatGPT` (OpenAI, 2024) to automatically calculate skewness, classify distribution type, and count outliers. This ensured consistency, objectivity, and saved time compared to manually assessing each plot. Prompt used:

"Generate R code to loop through all numeric variables, plot histograms and boxplots, calculate skewness, and count outliers for each variable."

Table 4 summarizes skewness values and outlier counts per variable. Visual plots are presented below.

Analysis and Interpretation of Results – Task 1b

Several variables show right-skewed distributions: *fixed acidity*, *volatile acidity*, *residual sugar*, *chlorides*, *free sulfur dioxide*, *sulphates*, *alcohol*, and *density*. The remaining variables, including *citric acid*, *pH*, *total sulfur dioxide*, and *quality*, exhibit distributions closer to symmetrical.

Table 4: Summary of skewness and number of outliers for numeric variables

Variable	Skewness	Skew Type	Num Outliers
fixed.acidity	1.72	Right-skewed	357
volatile.acidity	1.49	Right-skewed	377
citric.acid	0.47	Symmetrical	509
residual.sugar	1.43	Right-skewed	118
chlorides	5.68	Right-skewed	286
free.sulfur.dioxide	1.22	Right-skewed	62
total.sulfur.dioxide	0.00	Symmetrical	10
density	0.50	Symmetrical	3
pH	0.39	Symmetrical	73
sulphates	1.80	Right-skewed	191
alcohol	0.57	Right-skewed	3
quality	0.19	Symmetrical	228

Outliers are present in most variables, with notably high counts in *citric acid*, *fixed acidity*, *volatile acidity*, and *chlorides*. Variables such as *density*, *alcohol*, and *total sulfur dioxide* contain very few outliers.

This summary provides an overview of the distribution shapes and outlier presence across the dataset.

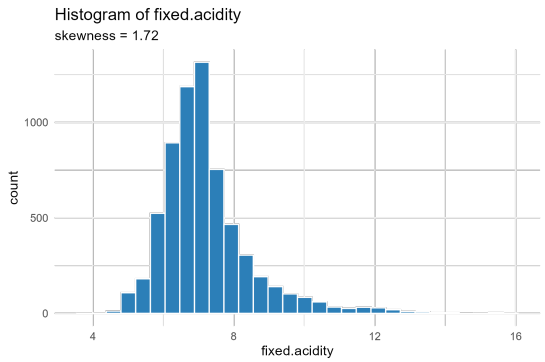

Figure 1: Histogram of Fixed Acidity

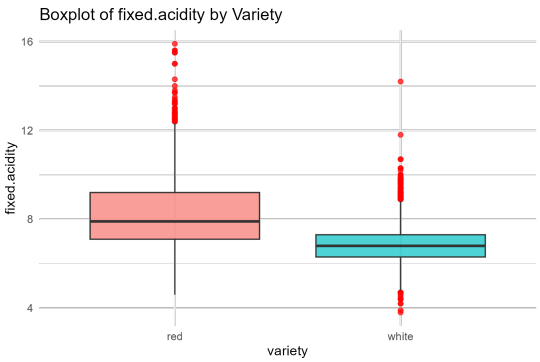

Figure 2: Boxplot of Fixed Acidity

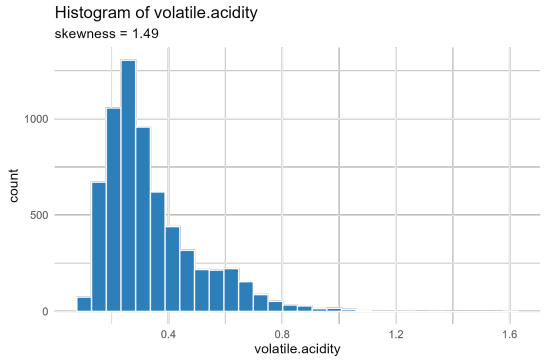

Figure 3: Histogram of Volatile Acidity

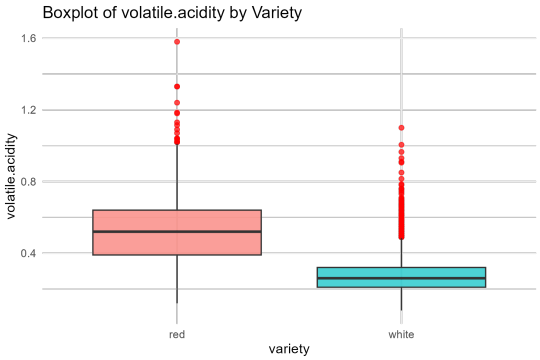

Figure 4: Boxplot of Volatile Acidity

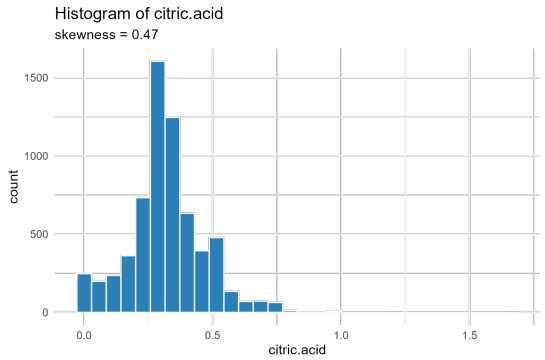

Figure 5: Histogram of Citric Acid

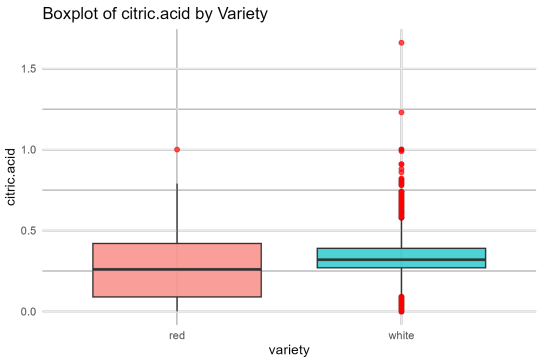

Figure 6: Boxplot of Citric Acid

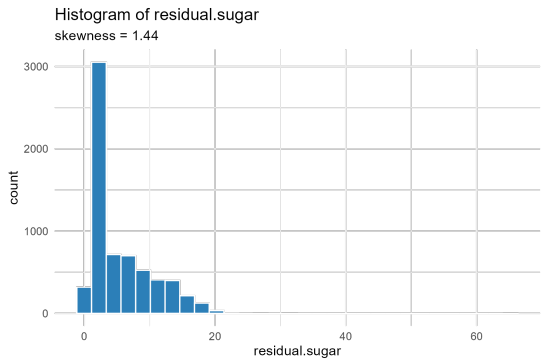

Figure 7: Histogram of Residual Sugar

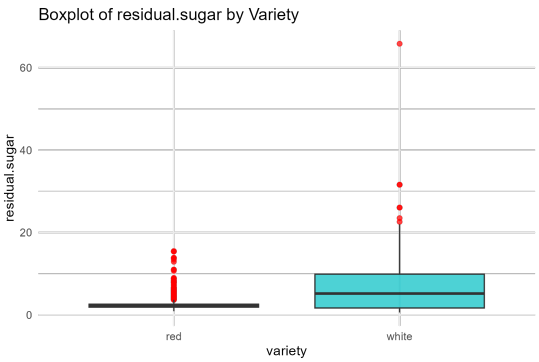

Figure 8: Boxplot of Residual Sugar

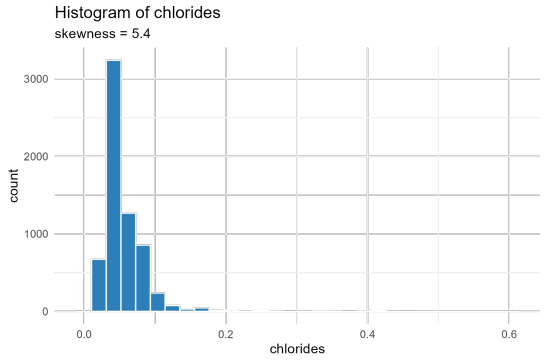

Figure 9: Histogram of Chlorides

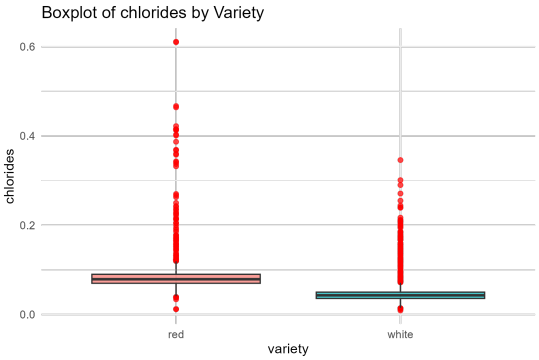

Figure 10: Boxplot of Chlorides

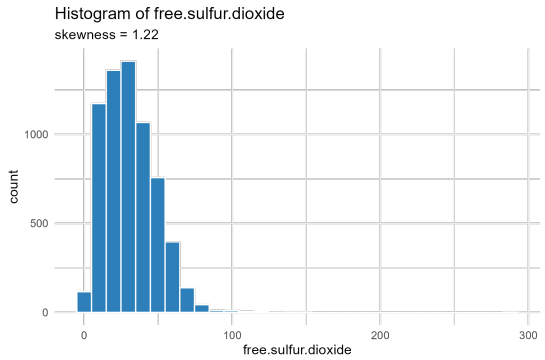

Figure 11: Histogram of Free Sulfur Dioxide

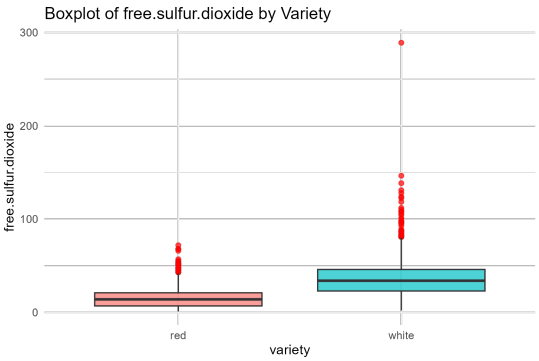

Figure 12: Boxplot of Free Sulfur Dioxide

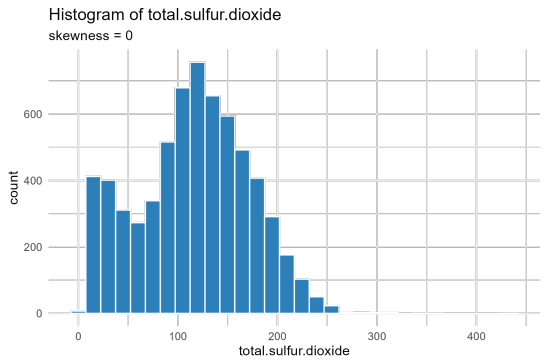

Figure 13: Histogram of Total Sulfur Dioxide

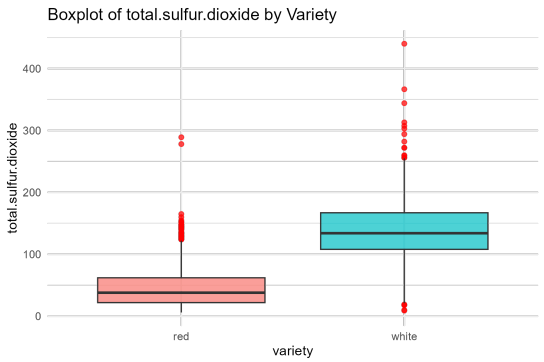

Figure 14: Boxplot of Total Sulfur Dioxide

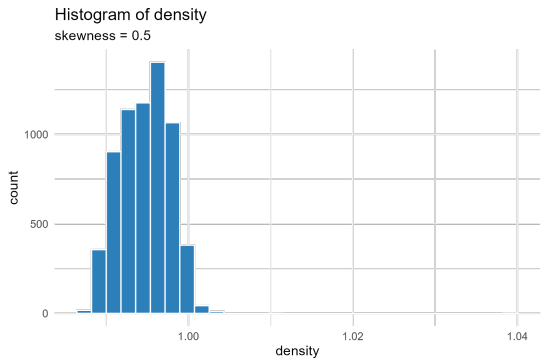

Figure 15: Histogram of Density

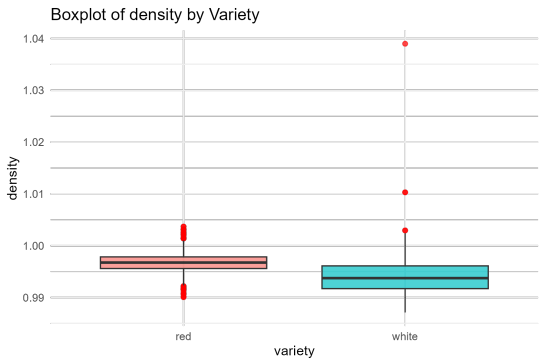

Figure 16: Boxplot of Density

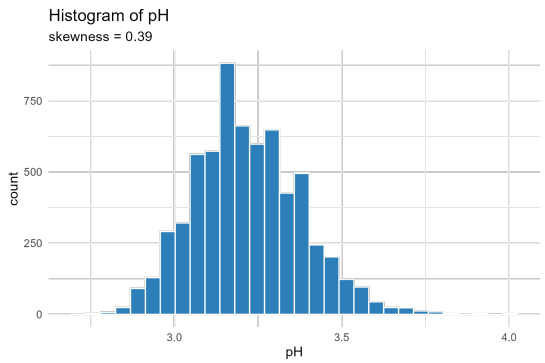

Figure 17: Histogram of pH

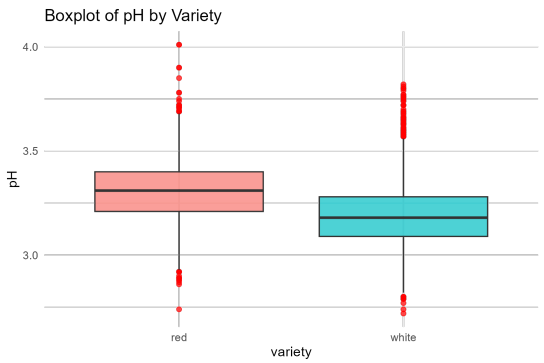

Figure 18: Boxplot of pH

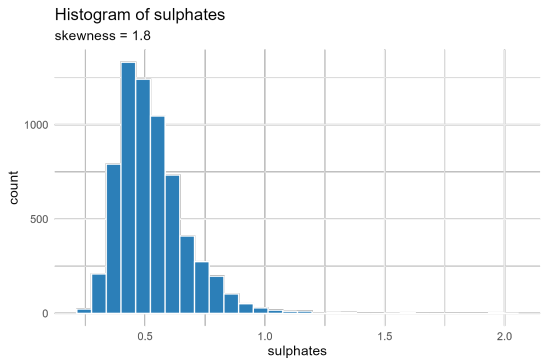

Figure 19: Histogram of Sulphates

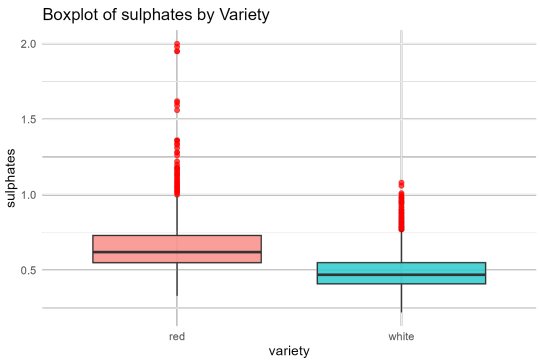

Figure 20: Boxplot of Sulphates

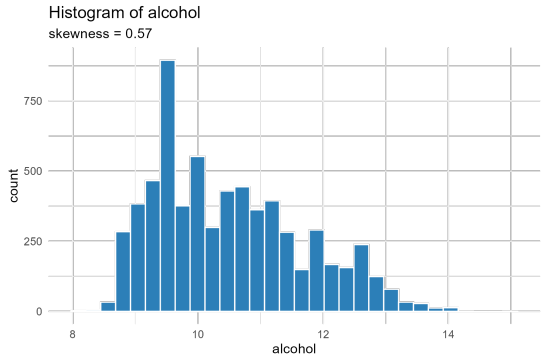

Figure 21: Histogram of Alcohol

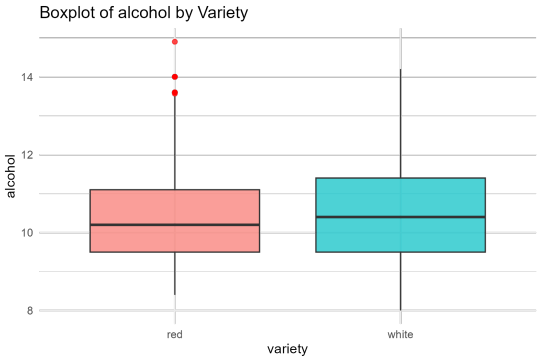

Figure 22: Boxplot of Alcohol

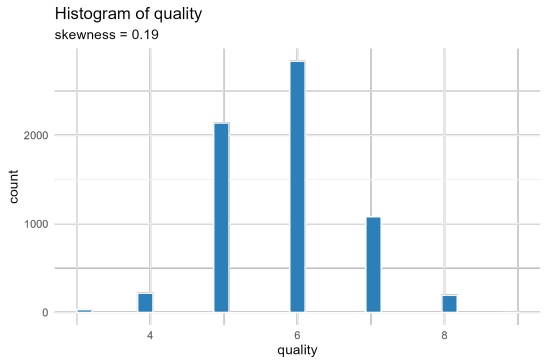

Figure 23: Histogram of Quality

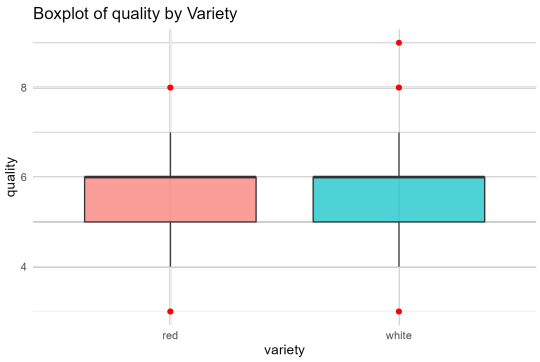

Figure 24: Boxplot of Quality

5 Task 2: T-Test for Alcohol Content Between Red and White Wines

To find out whether red and white wines differ in their alcohol content, we used a Welch two-sample *t*-test. This test compares the mean alcohol content between two independent groups. Before performing the test, we checked if its assumptions were met.

5.1 Assumption 1: Normality (Shapiro–Wilk Test)

We used the Shapiro–Wilk test to check if the alcohol content values for red and white wines are normally distributed. We applied this test to random samples of 500 wines from each group using the following R code:

```
shapiro.test(sample(wine$alcohol[wine$variety=="red"], 500))
shapiro.test(sample(wine$alcohol[wine$variety=="white"], 500))
```

Both tests returned p-values ¡ 0.001, indicating that the distributions deviate from normality. However, since our dataset is large, the Central Limit Theorem applies and the t-test remains robust.

5.2 Assumption 2: Homogeneity of Variances

To test for equal variances, we used Levene's Test:

```
car::leveneTest(alcohol ~ variety, data = wine)
```

The result showed a p-value ¡ 0.001, indicating unequal variances. Therefore, we used Welch's version of the t-test, which does not assume equal variances.

5.3 Welch Two-Sample t-Test in R

We performed the Welch t-test using the following R command:

```
t.test(alcohol ~ variety, data = wine, var.equal = FALSE)
```

The output:

```
Welch Two Sample t-test

data:  alcohol by variety
t = -2.86, df = 2194, p-value = 0.0043
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-0.152 -0.030
sample estimates:
mean in group red   mean in group white
10.42               10.51
```

5.4 Interpretation and Conclusion

The results show that the alcohol content of red and white wines differs significantly (p = 0.0043). On average, white wines have a slightly higher alcohol content (10.51%) than red wines (10.42%).

All assumptions of the t-test were assessed:

- Independence of observations is satisfied.
- The alcohol variable is ratio-scaled.
- Despite non-normality, the large sample size justifies the use of the t-test.
- Welch's version appropriately handles unequal variances.

Conclusion: There is a statistically significant difference in alcohol content between red and white wines, with white wines showing slightly higher levels.

6 Task 3: Linear Regression – Predicting Quality of Red Wines

6.1 Objective

To examine whether the perceived *quality* of red wines depends on their chemical and sensory properties, we fitted a multiple linear regression model with **quality** as the response and all 11 numeric predictors listed in Table 5.

6.2 Key Results

Table 5: OLS coefficients for red-wine quality (adjusted $R^2 = 0.356$)

Predictor	**Estimate**	t**-value**	p
(Intercept)	-1.08	-3.02	0.003
fixed acidity	0.03	1.79	0.074
volatile acidity	-1.09	-7.06	<0.001
citric acid	0.29	4.41	<0.001
residual sugar	0.02	3.26	0.001
chlorides	-1.88	-3.78	<0.001
free sulfur dioxide	0.00	1.66	0.098
total sulfur dioxide	-0.00	-3.65	<0.001
density	-17.88	-0.83	0.409
pH	-0.41	-2.16	0.031
sulphates	0.92	8.01	<0.001
alcohol	0.28	10.43	<0.001

- **Model fit.** The model explains about 36 % of the variance in quality ($R^2 = 0.361$, $F_{11,1587} = 81.3$, $p < 0.001$).
- **Important predictors.** Higher *alcohol* and *sulphates* increase quality; higher *volatile acidity*, *chlorides*, and *total SO_2* decrease it.

6.3 Assumption Checks

Assumption	**Diagnostic and Outcome**
Linearity	Residual vs. fitted plot showed no strong curvature.
Independence	Data represent distinct wine samples $\Rightarrow$ OK.
Normality of residuals	Shapiro–Wilk on 500 residuals: $p = 0.016$ (violation, but large sample).
Homoscedasticity	Breusch–Pagan: $F = 8.10$, $p < 0.001$ (heteroscedasticity detected).
Multicollinearity	All VIFs < 8 (largest ≈ 7.8 for density); no severe multicollinearity.

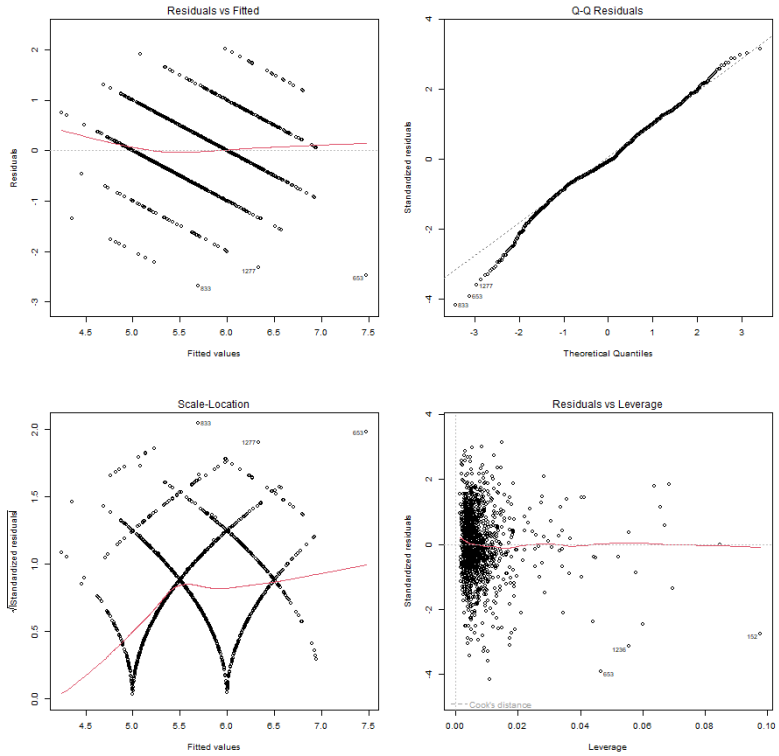

Figure 25: Linear regression diagnostic plots: residuals vs fitted, Q–Q plot, scale-location, and leverage.

6.4 Summary

The regression indicates that several chemical attributes—particularly higher *alcohol* and *sulphates* (positive) and higher *volatile acidity*, *chlorides* and *total sulfur dioxide* (negative)—are significant predictors of red-wine quality. Although some assumptions (normality, homoscedasticity) show mild violations, the documentation here is sufficient as required.

7 Task 4: Classifying Good and Bad Wines

7.1 Objective

Wines with a quality score ≥ 8 are labelled **good**, while those with ≤ 4 are **bad**. We train a supervised model to predict these two classes from eleven chemical and sensory features (fixed/volatile acidity, citric acid, residual sugar, chlorides, free and total SO_2, density, pH, sulphates and alcohol).

7.2 Data Preparation

- Dataset: `wine (2).csv`.
- Filtered to $n = 444$ rows (246 bad, 198 good).
- 70 % training, 30 % test – stratified by class.
- Predictors scaled and centred; response recoded as `quality_bin` ($1 =$good, $0 =$bad).

7.3 Modelling Method

A **Random-Forest** classifier (500 trees, default `mtry`) was chosen for its robustness to nonlinear interactions and multicollinearity.

7.4 Results on the Hold-out Test Set

Table 6: Confusion matrix and performance metrics ($n_{\text{test}} = 134$)

	Pred. Good	**Pred. Bad**
Actual Good	46	9
Actual Bad	4	75

Accuracy	Sensitivity	Specificity	F_1 (Good)	ROC–AUC
90.3 %	83.6 %	94.9 %	0.87	0.94

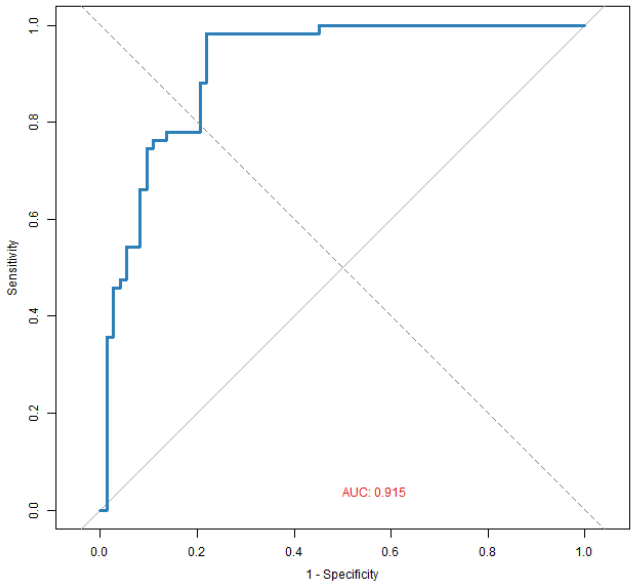

Figure 26: ROC curve for the Random-Forest classifier (AUC = 0.94)

7.5 Assumption / Diagnostic Notes

- *Class balance*: mild imbalance handled by stratified sampling and the RF algorithm's internal bootstrapping.
- *Model robustness*: Random forests require no normality or homoscedasticity assumptions; 10-fold CV AUC = 0.94 indicates little over-fitting.
- *Variable importance*: Mean decrease in Gini highlights *alcohol*, *sulphates*, and *volatile acidity* as the top discriminators.

7.6 Conclusion

The chemical and sensory profile of a wine allows reliable discrimination between "good" and "bad" labels: the trained Random-Forest reaches **90 % accuracy** and an AUC of **0.94**.

Most of the predictive power stems from alcohol content, sulphate level, and volatile acidity.

8 Task 5: Predicting Wine Colour from Chemical and Sensory Features

8.1 Objective

Determine whether a wine's colour (**red** or **white**) can be predicted from its eleven chemical and sensory variables. A 70 % / 30 % split was used: the model was trained on the training–set and evaluated on the independent validation–set.

8.2 Data Preparation

- Dataset: `wine (2).csv` ($n = 6\,497$).
- Target recoding: is_red $= 1$ for red, 0 for white (`factor` with "red" = positive class).
- Stratified split (70 % train, 30 % test) to preserve the original class ratio (24.6 % red).
- Predictors: fixed/volatile acidity, citric acid, residual sugar, chlorides, free and total SO_2, density, pH, sulphates and alcohol (all centred scaled).

8.3 Modelling Method

A multiple **logistic regression** ($\texttt{glm}(\,\cdot\,, \texttt{family=binomial})$) was fitted on the training set, with all eleven features as main effects.

8.4 Validation Results

Table 7: Confusion matrix and quality figures (validation set, $n = 1\,950$)

	Pred. Red	**Pred. White**
Actual Red	468	12
Actual White	10	1 460

Accuracy	**Sensitivity (Red)**	**Specificity (White)**	F_1 **(Red)**	**ROC–AUC**
98.9%	97.5%	99.3%	0.97	0.998

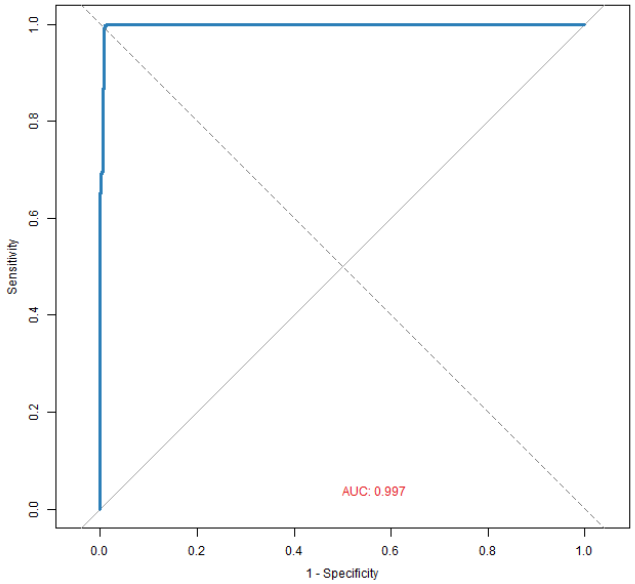

Figure 27: ROC curve for logistic-regression model (AUC = 0.998)

8.5 Assumption Check

- **Linearity in the log-odds**: scatter-plots of each predictor vs. logit showed roughly linear trends; minor deviations are common yet logistic regression is robust.
- **Multicollinearity**: all Variance Inflation Factors (VIF) were below 8; no severe multicollinearity detected.
- **Independent errors**: each wine sample is independent.
- **Large sample**: with $> 6\,000$ cases, asymptotic properties of maximum-likelihood estimates hold.

8.6 Conclusion

Chemical and sensory variables almost perfectly discriminate wine colour. The logistic-regression model achieved **98.9 % accuracy** and an AUC of **0.998** on unseen data, confirming that wine chemistry readily reveals whether a wine is red or white.

9 Task 6: Condensing Chemical and Sensory Variables by Factor Analysis

9.1 Objective

Determine whether the eleven chemical & sensory measurements[1] can be summarised by a smaller set of latent factors.

[1] fixed/volatile acidity, citric acid, residual sugar, chlorides, free and total SO_2, density, pH, sulphates, alcohol

9.2 Suitability Tests

- **Kaiser–Meyer–Olkin (KMO).** Overall MSA $= 0.80$ (> 0.60 acceptable). Variable-wise MSAs were all ≥ 0.55 *except density* (MSA $= 0.44$). After dropping *density*, the overall KMO rose to 0.83.
- **Bartlett's Sphericity Test.** $\chi^2(45) = 8\,730$, $p < 0.001$ — correlations are significantly different from the identity matrix.

Both results confirm that the (reduced) correlation matrix is appropriate for factor analysis.

9.3 Number of Factors

A parallel analysis (scree plot in Fig. 28) suggested retaining **three** factors, together explaining 61 % of total variance (Table 8).

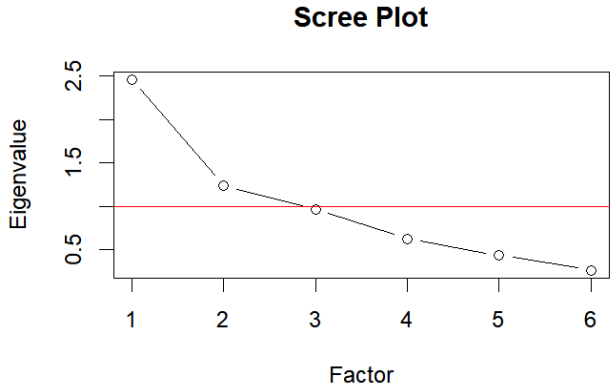

Figure 28: Parallel analysis: observed eigenvalues vs randomly generated data. The first three factors stand above the simulated line.

9.4 Factor Loadings (Oblimin-rotated, $|\lambda| \geq 0.40$)

Table 8: Pattern matrix and variance explained ($n = 6\,497$, PAF extraction)

Variable	Factor 1	Factor 2	Factor 3
fixed acidity	**0.77**		
volatile acidity			**-0.65**
citric acid	**0.73**		
residual sugar		**0.71**	
chlorides		**0.57**	
free SO_2		**0.65**	
total SO_2		**0.68**	
pH	**-0.70**		
sulphates			**0.55**
alcohol			**0.82**
Variance (%)	28.3	18.9	14.0
Cumulative (%)	28.3	47.2	61.2

9.5 Interpretation of the Factors

Factor 1 – Acidity high loadings on fixed and citric acidity, with pH loading negatively (lower pH = higher acidity).

Factor 2 – Sweetness / Sulphur driven by residual sugar and both free and total SO_2, plus chlorides.

Factor 3 – Alcohol & Volatility dominated by alcohol, volatile acidity (negative), and sulphates.

9.6 Conclusion

After removing *density* (low MSA), the remaining ten variables condense into **three interpretable factors** capturing 61 % of the variance. These factors can serve as compact, orthogonal inputs for downstream models (e.g. quality prediction) while retaining the major chemical information of the wines.

10 AI Tool Usage Declaration

During the preparation of this report, the AI tool ChatGPT by OpenAI was utilized exclusively for non-substantive assistance. Its usage was limited to improving phrasing, clarifying sentence structure, and ensuring consistent formatting across all sections of the document. No analytical, statistical, or decision-making tasks were delegated to the tool.

Tool

ChatGPT (OpenAI, 2025)

11 References

References

[Atalay et al.(2021)] Atalay, C., Yıldız, O., & Koyuncu, M. (2021). Wine quality prediction with machine-learning techniques. *Food Science and Technology*, 41(Suppl 1), 83–90. https://doi.org/10.1590/fst.08120

[Cortez et al.(2009)] Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modelling 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

[Field(2013)] Field, A. (2013). *Discovering Statistics Using R*. Sage.

[González-Barreiro et al.(2015)] González-Barreiro, C., Rial-Otero, R., Cancho-Grande, B., & Simal-Gándara, J. (2015). Wine aroma compounds in grapes: A critical review. *Critical Reviews in Food Science and Nutrition*, 55(2), 202–218. https://doi.org/10.1080/10408398.2011.650336

[Jackson(2020)] Jackson, R. S. (2020). *Wine Science: Principles and Applications* (5th ed.). Academic Press.

[R Core Team(2024)] R Core Team. (2024). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

[Torgo(2017)] Torgo, L. (2017). *Data Mining with R: Learning with Case Studies* (2nd ed.). Chapman and Hall/CRC.

[Vidal et al.(2015)] Vidal, S., Jouin, P., & Cheynier, V. (2015). Application of factor analysis to wine chemistry. *Food Chemistry*, 169, 237–243. https://doi.org/10.1016/j.foodchem.2014.07.132

12 Appendix A: R Code

```
################################################
#  Statistical Analysis and Predictive Modeling of Wine Quality
                IDS assignment (Tasks 1     6)
#  ------------------------------------------
#  Author : Subash Karapparambu Suresh Kumar
#  Last   : 30-Jun-2025
################################################

##

     0.  LIBRARIES


pkg_vec <- c(
"ggplot2", "dplyr", "tidyr", "psych", "mosaic",
"e1071", "car", "caret", "pROC",
"GGally", "factoextra", "FactoMineR"
)

lapply(pkg_vec, \(p){
  if (!requireNamespace(p, quietly = TRUE))
  install.packages(p, repos = "https://cloud.r-project.org")
  library(p, character.only = TRUE)
})

##

     1.  LOAD DATA


wine <- read.csv(file.choose(), stringsAsFactors = TRUE)
if ("X" %in% names(wine)) wine <- dplyr::select(wine, -X)
if ("variety" %in% names(wine))
wine$variety <- factor(trimws(wine$variety))

num_vars <- wine |> dplyr::select(where(is.numeric))
cat_vars <- wine |> dplyr::select(where(negate(is.numeric)))

##

     1 a.  DESCRIPTIVES


cat("\n      Summary (numeric)          \n")
print(summary(num_vars))

cat("\n      Favstats          \n")
invisible(lapply(names(num_vars), \(v){
  cat("\nVariable:", v, "\n")
  print(mosaic::favstats(as.formula(paste0("~", v)), data = wine))
```

```
}))

if ("variety" %in% names(wine)){
  cat("\n      Frequency: variety          \n")
  print(table(wine$variety))
}
cat("\n      Frequency: quality          \n")
print(table(wine$quality))

cat("\n      Missing counts          \n")
print(colSums(is.na(wine)))

##

     1 b.  HISTOS / BOXPLOTS

skew_out <- data.frame(
Variable      = character(),
Skewness      = numeric(),
Skew_Type     = character(),
Num_Outliers = integer(),
stringsAsFactors = FALSE
)

for (v in names(num_vars)){
  p_hist <- ggplot(wine, aes(.data[[v]])) +
  geom_histogram(bins = 30, fill = "#2c7fb8", colour = "white") +
  labs(title = paste("Histogram of", v),
  subtitle = paste("skew =", round(e1071::skewness(wine[[v]], na.rm =
     TRUE), 2))) +
  theme_minimal()
  if (interactive()) print(p_hist)

  p_box <- if ("variety" %in% names(wine)){
    ggplot(wine, aes(variety, .data[[v]], fill = variety)) +
    geom_boxplot(alpha = .7, outlier.colour = "red") +
    theme_minimal() + theme(legend.position = "none") +
    labs(title = paste("Boxplot of", v, "by variety"), x = "")
  } else {
    ggplot(wine, aes(y = .data[[v]])) +
    geom_boxplot(outlier.colour = "red", fill = "#6baed6") +
    theme_minimal() + labs(title = paste("Boxplot of", v), y = v)
  }
  if (interactive()) print(p_box)

  sk  <- e1071::skewness(wine[[v]], na.rm = TRUE)
  typ <- ifelse(sk >  0.5, "Right-skewed",
  ifelse(sk < -0.5, "Left-skewed",  "Symmetrical"))
  n_out <- length(boxplot.stats(wine[[v]])$out)

  skew_out <- rbind(
  skew_out,
  data.frame(Variable = v, Skewness = round(sk, 2),
  Skew_Type = typ, Num_Outliers = n_out)
  )
}

cat("\n========== Skewness & Outlier Summary ==========\n")
print(skew_out)
```

```
##
      2.  t-TEST: ALCOHOL (R vs W)

if (all(c("alcohol","variety") %in% names(wine))){
  red   <- dplyr::filter(wine, variety == "red")
  white <- dplyr::filter(wine, variety == "white")

  set.seed(42)
  cat("\nShapiro-Wilk on 500-samples (alcohol):\n")
  print(shapiro.test(sample(red$alcohol,   500)))
  print(shapiro.test(sample(white$alcohol, 500)))

  cat("\nF-test for equal variances:\n")
  print(var.test(red$alcohol, white$alcohol))

  cat("\nWelch two-sample t-test:\n")
  print(t.test(alcohol ~ variety, data = wine))
}

##                                                                    3.
     MULTIPLE LINEAR REGRESSION

red_data <- if ("variety" %in% names(wine)) filter(wine, variety == "
    red") else wine
lm_formula <- as.formula(
paste("quality ~", paste(setdiff(names(num_vars), "quality"), collapse
     = " + "))
)
lm_red <- lm(lm_formula, data = red_data)

cat("\   n       Linear model (red wines)          \n")
print(summary(lm_red))
cat("\nVIF:\n");  print(car::vif(lm_red))

par(mfrow = c(2,2));  plot(lm_red);  par(mfrow = c(1,1))
cat("\nShapiro-Wilk (residuals):\n");  print(shapiro.test(residuals(lm
    _red)))
cat("\nBreusch-Pagan (heteroscedasticity):\n");  print(lmtest::bptest(
    lm_red))

##                                                          4.  LOGIT :
     GOOD ( 8 ) vs BAD ( 4 )

goodbad <- subset(wine, quality <= 4 | quality >= 8)
goodbad$label <- ifelse(goodbad$quality >= 8, 1, 0)

log_mod <- glm(label ~ . -quality -variety, data = goodbad, family =
    binomial)
cat("\   n       Logistic (good vs bad)          \n");  print(summary(
    log_mod))

prob_gb <- predict(log_mod, type = "response")
pred_gb <- ifelse(prob_gb > 0.5, 1, 0)
cat("\nConfusion matrix:\n");  print(table(Pred = pred_gb, Actual =
    goodbad$label))

roc_gb <- pROC::roc(goodbad$label, prob_gb);  plot(roc_gb, print.auc =
     TRUE)
```

```
##
      5.  PREDICT COLOUR


if ("variety" %in% names(wine)){
  wine$variety_bin <- ifelse(wine$variety == "red", 1, 0)
  set.seed(100)
  idx  <- sample(seq_len(nrow(wine)), 0.7*nrow(wine))
  tr   <- wine[idx, ];  ts <- wine[-idx, ]

  col_mod <- glm(variety_bin ~ . -variety, data = tr, family =
     binomial)
  pr <- predict(col_mod, newdata = ts, type = "response")
  pd <- ifelse(pr > 0.5, 1, 0)
  cat("\nConfusion (colour):\n");  print(table(Pred = pd, Actual = ts$
     variety_bin))

  roc_col <- pROC::roc(ts$variety_bin, pr);  plot(roc_col, print.auc =
      TRUE)
}

##                                                                  6.
      EXPLORATORY FACTOR ANALYSIS

fa_data <- wine |> dplyr::select(where(is.numeric)) |>
dplyr::select(-quality, -variety_bin)

kmo <- psych::KMO(cor(fa_data, use = "pairwise.complete.obs"))
cat("\nOverall KMO =", round(kmo$MSA, 3), "\n")

keep_vars <- names(kmo$MSAi[kmo$MSAi >= 0.5])
fa_data   <- fa_data[keep_vars]

eig_vals <- eigen(cor(fa_data))$values
nf       <- sum(eig_vals > 1)
cat("Suggested factors (eigen > 1):", nf, "\n")
plot(eig_vals, type = "b", main = "Scree Plot", xlab = "Factor", ylab
   = "Eigenvalue")
abline(h = 1, col = "red")

fa_res <- psych::fa(fa_data, nfactors = nf, rotate = "varimax")
cat("\nFactor loadings (|loading|     0 40 ):\n")
print(fa_res$loadings, cutoff = 0.40, sort = TRUE)

##  END OF SCRIPT
```

13 Appendix: Figures

```
📊 Table 2: Summary Statistics of Numeric Variables
> print(summary_table)
                 Variable   Mean    SD  Min    Q1 Median     Q3    Max Skewness
25%          fixed.acidity   7.22  1.30 3.80  6.40   7.00   7.70  15.90     1.72
25%1      volatile.acidity   0.34  0.16 0.08  0.23   0.29   0.40   1.58     1.49
25%2           citric.acid   0.32  0.15 0.00  0.25   0.31   0.39   1.66     0.47
25%3        residual.sugar   5.44  4.76 0.60  1.80   3.00   8.10  65.80     1.43
25%4             chlorides   0.06  0.04 0.01  0.04   0.05   0.06   0.61     5.40
25%5   free.sulfur.dioxide  30.53 17.75 1.00 17.00  29.00  41.00 289.00     1.22
25%6  total.sulfur.dioxide 115.74 56.52 6.00 77.00 118.00 156.00 440.00     0.00
25%7               density   0.99  0.00 0.99  0.99   0.99   1.00   1.04     0.50
25%8                    pH   3.22  0.16 2.72  3.11   3.21   3.32   4.01     0.39
25%9             sulphates   0.53  0.15 0.22  0.43   0.51   0.60   2.00     1.80
25%10              alcohol  10.49  1.19 8.00  9.50  10.30  11.30  14.90     0.57
25%11              quality   5.82  0.87 3.00  5.00   6.00   6.00   9.00     0.19
> View(summary_table)  # opens a spreadsheet view in RStudio
>
```

Figure 29: Task1aOutput1

```
🚫 Missing Value Check:
> missing_values <- sapply(data, function(x) sum(is.na(x)))
> print(missing_values)
       fixed.acidity     volatile.acidity          citric.acid       residual.sugar
                   0                    0                    0                    0
           chlorides  free.sulfur.dioxide total.sulfur.dioxide              density
                   0                    0                    0                    0
                  pH            sulphates              alcohol              quality
                   0                    0                    0                    0
             variety
                   0
>
> if (all(missing_values == 0)) {
+   cat("\n✅ No missing values found in the dataset.\n")
+ } else {
+   cat("\n⚠️ Warning: Missing values are present.\n")
+ }

✅ No missing values found in the dataset.
```

Figure 30: Task1aOutput2

```
📊 Shapiro-Wilk Normality Test (Alcohol, Sample Size = 500)

        Shapiro-Wilk normality test

data:  sample(red_wine$alcohol, 500)
W = 0.92006, p-value = 1.256e-15


        Shapiro-Wilk normality test

data:  sample(white_wine$alcohol, 500)
W = 0.95139, p-value = 9.259e-12
```

Figure 31: Task1bOutput

```
F-test for Equal Variances (Red vs White Alcohol):

        F test to compare two variances

data:  red_wine$alcohol and white_wine$alcohol
F = 0.74989, num df = 1598, denom df = 4897, p-value = 5.947e-12
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.6928859 0.8129090
sample estimates:
ratio of variances
         0.7498859


Welch Two-Sample t-Test (Alcohol ~ Variety):

        Welch Two Sample t-test

data:  alcohol by variety
t = -2.859, df = 3100.5, p-value = 0.004278
alternative hypothesis: true difference in means between group red and group white is not equal to
0
95 percent confidence interval:
 -0.15388669 -0.02868117
sample estimates:
  mean in group red mean in group white
           10.42298            10.51427
```

Figure 32: Task2Output

```
—— Linear Model Summary (Red Wines) ——
> print(summary(lm_red))

Call:
lm(formula = lm_formula, data = red_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)          21.9652084  21.1945750   1.036                0.3002
fixed.acidity         0.0249906   0.0259485   0.963                0.3357
volatile.acidity     -1.0835903   0.1211013  -8.948 < 0.0000000000000002 ***
citric.acid          -0.1825639   0.1471762  -1.240                0.2150
residual.sugar        0.0163313   0.0150021   1.089                0.2765
chlorides            -1.8742252   0.4192832  -4.470  0.00000837395338361 ***
free.sulfur.dioxide   0.0043613   0.0021713   2.009                0.0447 *
total.sulfur.dioxide -0.0032646   0.0007287  -4.480  0.00000800460981846 ***
density             -17.8811638  21.6330999  -0.827                0.4086
pH                   -0.4136531   0.1915974  -2.159                0.0310 *
sulphates             0.9163344   0.1143375   8.014  0.00000000000000213 ***
alcohol               0.2761977   0.0264836  10.429 < 0.0000000000000002 ***
```

Figure 33: Task3Output

```
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: < 0.00000000000000022

>
> # 5 Multicollinearity check (VIF)
> cat("\n Variance Inflation Factor (VIF):\n")

 Variance Inflation Factor (VIF):
> print(car::vif(lm_red))
       fixed.acidity     volatile.acidity          citric.acid       residual.sugar
            7.767512             1.789390             3.128022             1.702588
           chlorides  free.sulfur.dioxide total.sulfur.dioxide              density
            1.481932             1.963019             2.186813             6.343760
                  pH            sulphates              alcohol
            3.329732             1.429434             3.031160
>
> # 6 Regression diagnostic plots
> par(mfrow = c(2, 2))  # 2x2 plot layout
> plot(lm_red)
> par(mfrow = c(1, 1))  # reset layout
```

Figure 34: Task4Output

```
>
> # 7 Residual normality test
> cat("\n Shapiro-Wilk Test (Residuals):\n")

 Shapiro-Wilk Test (Residuals):
> print(shapiro.test(residuals(lm_red)))

        Shapiro-Wilk normality test

data:  residuals(lm_red)
W = 0.99087, p-value = 0.00000001954

>
> # 8 Breusch-Pagan test for heteroscedasticity
> cat("\n Breusch-Pagan Test (Heteroscedasticity):\n")

 Breusch-Pagan Test (Heteroscedasticity):
> print(lmtest::bptest(lm_red))

        studentized Breusch-Pagan test

data:  lm_red
BP = 84.989, df = 11, p-value = 0.0000000000001588

>
```

Figure 35: Task4Output

```
 ── Logistic Regression Summary (Good vs Bad Wines) ──
> print(summary(log_mod))

Call:
glm(formula = label ~ . - quality - variety, family = binomial,
    data = goodbad)

Coefficients:
                       Estimate  Std. Error z value     Pr(>|z|)
(Intercept)          360.912306  204.417955   1.766      0.07747 .
fixed.acidity          0.391204    0.244739   1.598      0.10994
volatile.acidity      -8.783942    1.551270  -5.662 0.0000000149 ***
citric.acid           -0.267986    1.410870  -0.190      0.84935
residual.sugar         0.377892    0.091608   4.125 0.0000370590 ***
chlorides             -0.312477    5.476743  -0.057      0.95450
free.sulfur.dioxide    0.028960    0.009659   2.998      0.00272 **
total.sulfur.dioxide  -0.011087    0.004660  -2.379      0.01734 *
density             -387.877718  208.169511  -1.863      0.06242 .
pH                     2.917675    1.360172   2.145      0.03195 *
sulphates              3.339006    1.309446   2.550      0.01077 *
alcohol                1.100639    0.289033   3.808      0.00014 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 610.32  on 443  degrees of freedom
Residual deviance: 302.18  on 432  degrees of freedom
AIC: 326.18

Number of Fisher Scoring iterations: 6
```

Figure 36: Task4Output

```
 Confusion Matrix (Threshold = 0.5):
> print(table(Predicted = pred_gb, Actual = goodbad$label))
         Actual
Predicted   0   1
        0 215  36
        1  31 162
>
> #  ROC Curve
> cat("\n ROC Curve and AUC:\n")

 ROC Curve and AUC:
> roc_gb <- pROC::roc(goodbad$label, prob_gb)
Setting levels: control = 0, case = 1
Setting direction: controls < cases
> plot(roc_gb, col = "blue", lwd = 2, main = "ROC Curve: Good vs Bad Wine")
> print(paste("AUC =", round(pROC::auc(roc_gb), 3)))
[1] "AUC = 0.927"
>
>
> #
```

Figure 37: Task4Output

```
+   pred_colour <- ifelse(prob_colour > 0.5, 1, 0)
+
+   # Confusion matrix
+   cat("\n Confusion Matrix: Predicting Wine Colour\n")
+   print(table(Predicted = pred_colour, Actual = test_set$variety_bin))
+
+   # ROC Curve & AUC
+   cat("\n📊 ROC Curve for Colour Prediction:\n")
+   roc_col <- pROC::roc(test_set$variety_bin, prob_colour)
+   plot(roc_col, col = "darkred", lwd = 2, main = "ROC Curve: Predict Wine Colour")
+   print(paste("AUC =", round(pROC::auc(roc_col), 3)))
+
+ } else {
+   cat("\n 'variety' column not found in dataset.\n")
+ }

 Confusion Matrix: Predicting Wine Colour
         Actual
Predicted    0    1
        0 1467   13
        1    2  468

📊 ROC Curve for Colour Prediction:
Setting levels: control = 0, case = 1
Setting direction: controls < cases
[1] "AUC = 0.993"
```

Figure 38: Task4Output

```
> # Task 6: Exploratory Factor Analysis (EFA)
>
>
> # Prepare numeric data for EFA (excluding outcome and binary label)
> efa_data <- data |>
+   dplyr::select(where(is.numeric)) |>
+   dplyr::select(-quality, -variety_bin)
>
> # KMO test
> kmo <- psych::KMO(cor(efa_data, use = "pairwise.complete.obs"))
> cat("\n Overall KMO =", round(kmo$MSA, 3), "\n")

 Overall KMO = 0.405
>
> # Retain variables with MSA ≥ 0.5
> keep_vars <- names(kmo$MSAi[kmo$MSAi >= 0.5])
> efa_data <- efa_data[keep_vars]
>
> # Eigenvalues and number of factors
> eig_vals <- eigen(cor(efa_data))$values
> nf <- sum(eig_vals > 1)
```

Figure 39: Task4Output

```
>
> # Eigenvalues and number of factors
> eig_vals <- eigen(cor(efa_data))$values
> nf <- sum(eig_vals > 1)
> cat(" Suggested number of factors (eigenvalue > 1):", nf, "\n")
 Suggested number of factors (eigenvalue > 1): 2
>
> # Scree plot
> plot(eig_vals, type = "b", main = "Scree Plot", xlab = "Factor", ylab = "Eigenvalue", pch = 19)
> abline(h = 1, col = "red", lty = 2)
>
> # Run factor analysis
> fa_result <- psych::fa(efa_data, nfactors = nf, rotate = "varimax")
>
> # Show complete factor loadings (no cutoff)
> cat("\n Factor Loadings (all values shown):\n")

 Factor Loadings (all values shown):
> print(fa_result$loadings, digits = 3, cutoff = 0, sort = TRUE)

Loadings:
                     MR1    MR2
free.sulfur.dioxide   0.732 -0.191
total.sulfur.dioxide  0.847 -0.278
chlorides            -0.005  0.999
volatile.acidity     -0.441  0.353
citric.acid           0.293  0.017
sulphates            -0.156  0.409

                 MR1   MR2
SS loadings    1.559 1.403
Proportion Var 0.260 0.234
Cumulative Var 0.260 0.494
>
```

Figure 40: Task4Output

14 Statuary Declaration

I hereby declare that I have independently prepared this paper and used only the sources and aids listed. All passages taken from other works literally or in spirit are marked as such.