Generalized Linear Model

Claudio Mazzi
Department of Computer Science, University of Pisa
MeS Laboratory, Sant'Anna School for Advanced Studies, Pisa
claudio.mazzi@santannapisa.it

Applied Statistical Modelling 1
A.A. 2024-2025

Contents

1 Limitations of the Linear Regression Model

The linear regression model (LRM), while powerful for predicting and explaining relationships between a continuous outcome and predictor variables, has limitations that restrict its use in many real-world applications. Understanding these limitations is essential for appreciating why we require other statistical models, such as Generalized Linear Models (GLMs), to handle different types of data effectively. The key limitations of LRM are summarized as follows:

1. **Inappropriate for Non-Continuous Outcomes:**
   - Linear regression assumes that the response variable Y is continuous and unbounded. However, many research and practical scenarios involve outcomes that are not continuous, such as binary outcomes, counts, or categorical responses with multiple classes.
   - Attempting to fit a linear model to binary data, for example, can lead to predictions outside the feasible range of probabilities (i.e., values less than 0 or greater than 1), violating the core principles of probability and making interpretation misleading.
2. **Assumes a Linear Relationship Between Predictors and the Outcome:**
   - Linear regression is restricted by its assumption of a linear relationship between predictors X and the outcome Y. However, relationships in data are often non-linear. While transformations can sometimes help linearize relationships, they may not always be effective or interpretable.
3. **Variance of Errors is Constant (Homoscedasticity):**
   - Linear regression requires homoscedasticity, where the variance of errors (residuals) remains constant across values of the predictor variables. In practice, this assumption often fails. For example, in count data, variance frequently increases with the mean, a phenomenon known as overdispersion. When this assumption is violated, standard errors can be biased, leading to inaccurate inferences and potentially flawed conclusions.
4. **Normality of Errors:**
   - Linear regression also assumes that errors are normally distributed. While linear models can be robust to slight departures from normality, substantial deviations can impact the model's performance and the reliability of hypothesis tests.
5. **No Direct Control for Probability Structures in Outcomes:**
   - For categorical or count data, linear regression does not align with the natural probability distribution of the outcomes. For instance, binary outcomes fit naturally within the binomial distribution and count with the Poisson distribution. Linear regression lacks mechanisms to account for these structures, making it unsuitable for categorical or discrete data.

By addressing these limitations directly, GLMs allow us to model a broader range of outcomes, thereby overcoming the restrictive assumptions of linear regression.

2 Generalized Linear Models: A Broader Framework

GLMs expand the traditional linear regression model by allowing for a more adaptable structure that can handle various types of response variables and distributions. Unlike linear regression, which assumes a continuous and unbounded response, GLMs are designed to accommodate binary, categorical, and count data, making them versatile across diverse fields of research. The GLM framework consists of three core components that work together to provide this flexibility. The first component, known as the *random component*, specifies the probability distribution of the response variable. For example, binary data are typically modeled with a binomial distribution, while count data are often modeled using a Poisson distribution. Choosing an appropriate distribution is essential, as it ensures that the model aligns with the inherent nature of the data. The second component of a GLM is the *systematic component*, which consists of the linear predictor. This predictor is a linear combination of the explanatory variables and is often represented as $X\beta$, where X denotes the matrix of predictor variables and β represents the vector of coefficients to be estimated. The linear predictor forms the foundation for the model's systematic variation, providing a basis for estimating the relationship between the predictors and the response variable. The final component of the GLM framework is the *link function*, a critical feature that distinguishes GLMs from traditional linear models. The link function transforms the expected value of the response variable to a scale that can be related linearly to the predictor variables. This transformation is crucial because it ensures that predictions remain within the plausible range for the type of response data being modeled. For instance, in logistic regression for binary outcomes, the logit link function maps probabilities, which range from 0 to 1, to the entire real line, thus aligning the probability scale with the linear predictor. Similarly, the log link function in Poisson regression allows counting data predictions

to remain non-negative. By providing a flexible means of connecting the predictors with the response variable, the link function allows GLMs to model a wide variety of data structures more accurately.

In summary, the GLM framework adapts the linear model to fit different types of data by combining an appropriate probability distribution for the response, a linear predictor for systematic effects, and a link function that ensures the model predictions remain in a feasible range. This flexibility makes GLMs a powerful tool in statistical modeling, enabling analysts to overcome the limitations of linear regression and address the unique characteristics of different data types effectively.

2.1 Specific GLM Models

Within the framework of Generalized Linear Models, several specific models are tailored to address various types of response data, each offering a solution for situations where linear regression would be insufficient or inappropriate. Below, we discuss these models in detail, following an ordered progression from binary outcomes to categorical and count data, illustrating how each model is designed to capture different characteristics of response variables.

Logistic Regression (Logit Model) for Binary Responses

Logistic regression, or the **logit model**, is one of the most widely used models within the GLMs family, particularly suited for binary response variables (e.g., success/failure, disease/no disease). This model applies the logit link function to map the response probability onto the real line, thereby ensuring that predictions remain within the $[0, 1]$ interval, consistent with probability theory. Mathematically, logistic regression is expressed as:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = X\beta$$

where p is the probability of the response being 1 (e.g., success). By modeling the log odds of the outcome as a linear function of the predictors, logistic regression handles binary data effectively and provides interpretable coefficients that indicate the change in log odds for a unit increase in each predictor. Furthermore, Maximum Likelihood Estimation (MLE) is used to estimate model parameters, yielding asymptotically unbiased estimates for large samples.

Probit Model for Binary Responses

The probit model is designed for binary outcomes, and it is closely related to logistic regression but differs in the link function used. Instead of the logit link, the probit model uses the Cumulative Distribution Function (CDF) of the standard normal distribution. This choice of link assumes that an underlying latent variable, normally distributed, drives the binary outcome, making the probit model appropriate in cases where data suggest a normally distributed response at a latent level. The model is expressed as:

$$\Phi^1(p) = X\beta$$

where Φ^{-1} represents the inverse of the normal CDF. Although logistic and probit models often yield similar results, the choice between them can depend on theoretical considerations or the distributional assumptions underlying the data.

Ordinal Logistic Regression (Ordinal Logit) and Ordinal Probit

When the response variable is ordinal (i.e., has ordered categories such as "low", "medium" and "high"), ordinal logistic regression (ordinal logit) is appropriate. Unlike binary models, ordinal models account for the ordered nature of the categories by modeling cumulative probabilities for each level of the response variable. The ordinal logit model, commonly specified using a cumulative logit link, is expressed as:

$$\text{logit}(P(Y \leq j)) = \alpha_j - X\beta$$

for each category j, where α_j represents category-specific intercepts. This model assumes proportional odds across levels, meaning that the relationship between each predictor and the log odds of being in a higher versus a lower category remains constant across categories. This is known as the proportional odds assumption, a key feature of ordinal logit models.

Similarly, the ordinal probit model uses the probit link function instead of the logit, modeling the ordinal outcome as arising from a latent normal distribution. Both ordinal logit and probit models provide a structured approach to analyze ordinal outcomes, capturing the ordinal structure in ways that neither binary nor multinomial models can.

Multinomial Logistic Regression (Multinomial Logit) for Nominal Data

For response variables with multiple, unordered categories (e.g., types of transport: car, bike, bus), multinomial logistic regression, or the multinomial logit model, is used. Unlike ordinal models, multinomial models do not assume an order among categories, treating each outcome as a distinct class. The multinomial logit model extends the binary logit model by modeling the probability of each category k relative to a chosen baseline category, often expressed as:

$$\log\left(\frac{P(Y=k)}{P(Y=\text{baseline}}\right) = X\beta_k$$

where β_k is the coefficient vector for category k relative to the baseline. By estimating separate equations for each category compared to the baseline, the multinomial model allows for the modeling of complex, categorical outcomes, where each predictor's effect on the probability of choosing each category can differ. Multinomial logit models are widely used in fields such as marketing and social sciences to model choices among a set of discrete, nominal options.

Poisson Regression for Count Data

For count data—where the response variable represents the number of occurrences of an event—Poisson regression is a natural choice. This model assumes that the response follows a Poisson distribution, with the mean count determined by the linear predictor. Poisson regression employs a log link function, as follows:

$$\log(\mu) = X\beta$$

where μ is the expected count. This model is particularly useful for event counts (e.g., number of hospital visits), as it ensures non-negative predictions. Poisson regression also assumes that the mean and variance of the response variable are equal, a property known as equidispersion. However, this assumption is often violated in real-world data, which may exhibit overdispersion.

Overdispersed Poisson Model (Quasi-Poisson)

In cases where count data exhibit overdispersion—meaning that the variance exceeds the mean—the quasi-Poisson model provides a more flexible alternative. While retaining the Poisson model's log link, the quasi-Poisson model introduces a dispersion parameter, allowing for variance inflation. This approach addresses overdispersion by relaxing the assumption of equidispersion, making it more suitable for real-world applications where the Poisson model might otherwise struggle. The variance of the quasi-Poisson model can be expressed as:

$$\text{Var}(Y) = \phi \cdot \mu$$

where ϕ is the dispersion parameter. Quasi-Poisson models are commonly applied in fields where count data are prone to high variability, offering robust parameter estimates even when standard Poisson assumptions are not met.

3 Introduction: beyond linear regression

In this exercise, we will perform a comprehensive analysis of generalized linear models (GLMs) using the R programming language. We will start by exploring logit and probit models for binary outcomes, then proceed to ordinal logit for ordered categories, and finally, we will illustrate the multinomial logist regression for non-ordered categories. This progression will illustrate how GLMs, and in particular logit and probit models, adapt to various data structures, providing a flexible framework for accurately modeling relationships in diverse data types.

We start with a synthetic database to show how the Linear Regression fails in modeling binary outcomes.

Listing 1: Linear Regression on a Binary Outcome

```
# Step 1: Install and Load Required Packages
library(ggplot2)

# Step 2: Create a simulated dataset
set.seed(123)
```

```r
X <- rnorm(100, mean = 5, sd = 2)

# Outcome as a function of X
prob <- 1 / (1 + exp(-0.5 * (X - 5)))
Outcome <- rbinom(100, 1, prob = prob)
data_binary <- data.frame(Outcome, X)

# Display the first few rows of the dataset
head(data_binary)

# Step 3: Linear Regression
# Linear regression model
linear_model <- lm(as.numeric(Outcome) ~ X, data = data_binary)
summary(linear_model)

# Plot the linear regression fit
ggplot(data_binary, aes(x = X, y = as.numeric(Outcome))) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(y = "Outcome")
```

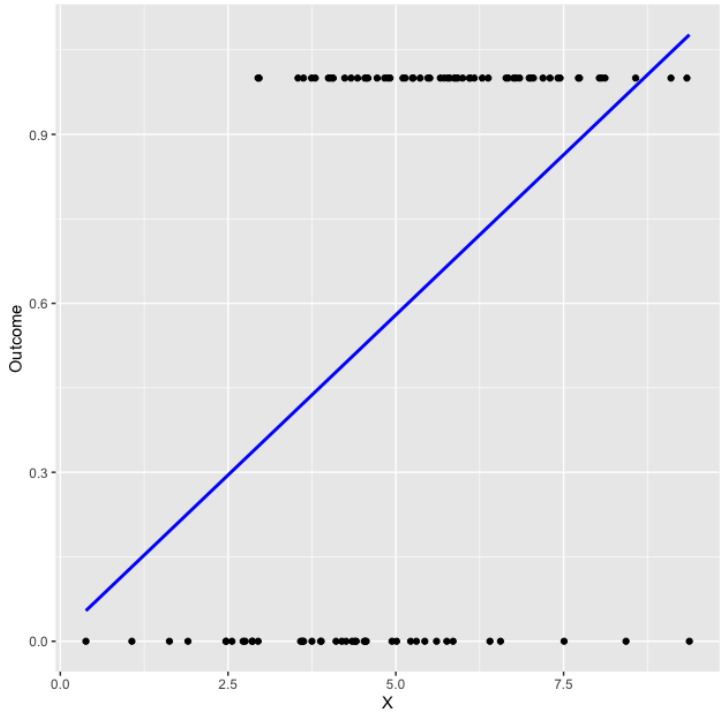

Figure 1: The plot shows how a linear regression is inappropriate to model a binary outcome. In general this is true for non-continuous and encoded outcomes.

The regression plot in Figure 1 displays the result of the linear regression between variables X and the binary response variable *Outcome*. It is easy to see how the LRM fails in model appropriately the relation between variables, as clearly shown by the regression stats in Figure 2. Indeed, the linear regression model poorly fits the data, as indicated by a low R-squared value (0.006), suggesting that the predictor X explains less than 1% of the variance in the outcome. Additionally, the coefficient for X is not statistically significant ($p = 0.434$), highlighting its limited predictive value. The high residual standard error and non-significant F-statistic further confirm that X does not meaningfully contribute to predicting the outcome in this model.

Now, we test logistic, and probit regression on the simulated database "data_binary".

```
Call:
lm(formula = as.numeric(Outcome) ~ X, data = data_binary)

Residuals:
    Min      1Q  Median      3Q     Max
-1.0769 -0.4267  0.1525  0.3695  0.6540

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.01084    0.13562   0.080    0.936
X            0.11372    0.02470   4.603 1.25e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4487 on 98 degrees of freedom
Multiple R-squared:  0.1778,    Adjusted R-squared:  0.1694
F-statistic: 21.19 on 1 and 98 DF,  p-value: 1.248e-05
```

Figure 2: Linear Regression Model statistics for the relation between X and the binary *Outcome*.

3.1 GLM Syntax and Key Arguments on R

The basic syntax of the `glm` function is similar to `lm`, but with additional flexibility provided by the `family` argument:

Listing 2: The glm R-function

```r
glm(formula, family = ..., data = ...)
```

- `formula`: Specifies the model in the format `response` $\sim$ `predictors`.
- `family`: Defines the error distribution and link function for the model (e.g., `binomial(link = "logit")` for logistic regression or `poisson(link = "log")` for count data).
- `data`: The dataset containing the variables.

The `family` argument in `glm` allows users to specify the appropriate error distribution and link function for their data type. Some common families include:

- **Binomial**: Suitable for binary outcomes, typically used with a `logit` or `probit` link function for logistic and probit regression models.
- **Poisson**: Designed for count data, often paired with a `log` link function.
- **Gaussian**: Equivalent to ordinary linear regression, with an `identity` link function by default.
- **Quasi**: Provides flexibility for overdispersion in the data, allowing for `quasi-binomial` or `quasi-poisson` families.

3.2 Beyond linear regression: logit regression

To correctly model a binary outcome, we apply logistic regression, which uses a specific transformation known as the logit function. Logistic regression is specifically designed for binary data and is based on the principle of modeling the log-odds of the outcome rather than the outcome itself.

The logit function transforms probabilities, which are limited to the range $[0, 1]$, into log-odds, which can take any real value. This transformation allows us to apply a linear model to the log-odds while keeping the predicted probabilities within a valid range. The logistic regression model is then fit to estimate the parameters of the log-odds relationship, making it an appropriate model for binary data, as shown in Figures 3-4.

Listing 3: Logistic Regression

```r
# Logistic regression model (logit)
logistic_model <- glm(Outcome ~ X, data = data_binary, family = binomial(link =
    "logit"))
summary(logistic_model)
```

```r

# Plotting logistic regression
X_seq <- seq(min(data_binary$X), max(data_binary$X), length.out = 100)
predicted_probs <- predict(logistic_model, newdata = data.frame(X = X_seq), type
    = "response")

plot(data_binary$X, data_binary$Outcome, pch = 16, col = "blue",
     xlab = "X", ylab = "Probability of Outcome = 1")
lines(X_seq, predicted_probs, col = "red", lwd = 2)
legend("right", legend = c("Observed Data", "Predicted Probability"),
       col = c("blue", "red"), pch = c(16, NA), lty = c(NA, 1), lwd = c(NA, 2))
```

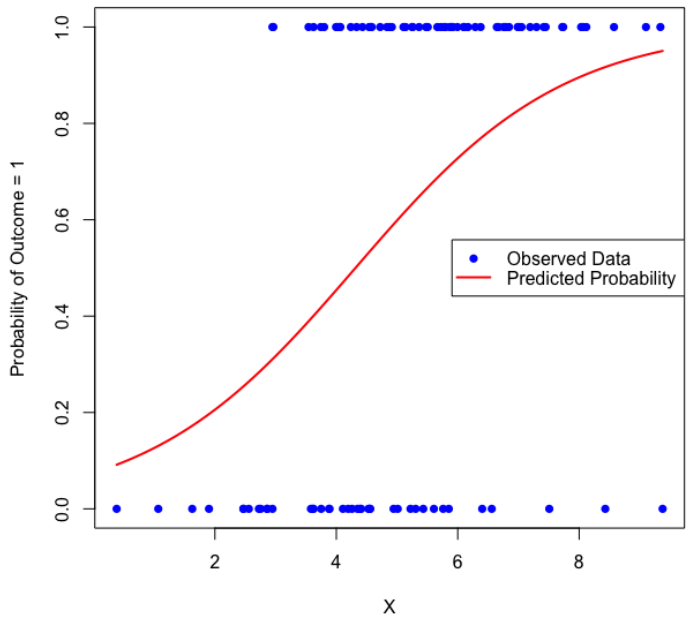

Figure 3: Logistic Regression Model outlining the relation between X and *Outcome* in *data_binary*.

```
Call:
glm(formula = Outcome ~ X, family = binomial(link = "logit"),
    data = data_binary)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.5179     0.7751  -3.248 0.001161 **
X             0.5835     0.1521   3.836 0.000125 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 134.60  on 99  degrees of freedom
Residual deviance: 114.94  on 98  degrees of freedom
AIC: 118.94

Number of Fisher Scoring iterations: 4
```

Figure 4: Summary for the Logistic Regression Model between X and *Outcome*.

The logit link function overcomes the issues encountered with LRM, predicting probabilities constrained between 0 and 1. In this model, the coefficients are interpretable as changes in the log-odds of

Outcome = 1 for each unit increase in the predictor variable X.

With dichotomic variables, we can also use probit regression, which differs from the logit in the link function it employs. The probit model uses the CDF of the standard normal distribution to link the probability of the outcome to the linear predictors. This difference in link functions results in slightly different scaling of predicted probabilities, but both approaches typically yield similar results in practice. In the next section, we will perform both probit and logit regression on a real dataset.

4 Logit and Probit Regression

In this analysis, we examine the dataset "heart.csv"[1] to investigate the factors associated with heart disease. The dataset includes a binary outcome variable, *HeartDisease*, which indicates the presence (1) or absence (0) of heart disease, as well as several predictor variables related to patient characteristics and health metrics. Our goal is to build and compare logit and probit regression models to understand the relationships between these predictors and the probability of heart disease.

4.1 Data Preprocessing

Loaded the dataset, we have to convert categorical variables *Sex* and *ExerciseAngina* into binary. Otherwise, the other categorical variables, such as *ChestPainType*, *RestingECG*, and *ST_Slope*, are one-hot encoded to create dummy variables suitable for regression analysis.

Listing 4: Preprocessing dataset

```r
# Load necessary libraries
library(stats)
library(pscl)  # For pseudo-R2 values if needed

# Load the dataset -> Insert the correct path to the dataset
data <- read.csv("path/to/heart.csv")

# Preprocess the data
# Encode 'Sex' and 'ExerciseAngina' as binary variables
data$Sex <- ifelse(data$Sex == "M", 1, 0)
data$ExerciseAngina <- ifelse(data$ExerciseAngina == "Y", 1, 0)

# One-hot encode categorical variables for 'ChestPainType', 'RestingECG', and 'ST_Slope'
data <- cbind(data, model.matrix(~ ChestPainType + RestingECG + ST_Slope - 1, data = data))
# Drop original factor columns
data$ChestPainType <- NULL
data$RestingECG <- NULL
data$ST_Slope <- NULL
```

The figures 5 and 6 show the header of the database before and after the preprocessing, with the encoding of the categorical features.

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
1	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
2	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
3	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
4	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
5	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0
6	39	M	NAP	120	339	0	Normal	170	N	0.0	Up	0

Figure 5: Original header of the heart database

[1] You can download the dataset and the corresponding documentation at the following link: https://github.com/EMbeDS-education/ComputingDataAnalysisModeling20242025/tree/main/datasets

```
  Age Sex RestingBP Cholesterol FastingBS MaxHR ExerciseAngina Oldpeak HeartDisease ChestPainTypeASY ChestPainTypeATA ChestPainTypeNAP
1  40   1       140         289         0   172              0     0.0           0                0                1                0
2  49   0       160         180         0   156              0     1.0           1                0                0                1
3  37   1       130         283         0    98              0     0.0           0                0                1                0
4  48   0       138         214         0   108              1     1.5           1                1                0                0
5  54   1       150         195         0   122              0     0.0           0                0                0                1
6  39   1       120         339         0   170              0     0.0           0                0                0                1
  ChestPainTypeTA RestingECGNormal RestingECGST ST_SlopeFlat ST_SlopeUp
1               0                1            0            0          1
2               0                1            0            1          0
3               0                0            1            0          1
4               0                1            0            1          0
5               0                1            0            0          1
6               0                1            0            0          1
```

Figure 6: Header of the preprocessed database with the encoding and the binary transformation of the response variable.

4.2 Logistic Regression (Logit) Model

Using logit regression we will interpret the coefficients in terms of changes in the log-odds of heart disease. We include in the model all the predictors in the dataset both the quantitative and the categorical ones we encoded.

Listing 5: Logit regression model

```
logit_model <- glm(HeartDisease ~ Age + Sex + RestingBP + Cholesterol +
    FastingBS +
                       MaxHR + ExerciseAngina + Oldpeak + ChestPainTypeATA +
                       ChestPainTypeNAP + ChestPainTypeTA + RestingECGST +
                       RestingECGNormal + ST_SlopeFlat + ST_SlopeUp,
                       family = binomial(link = "logit"), data = data)

# Model summary
summary(logit_model)
```

Figure 7 is shown the summary of the logit regression. It provides coefficients' estimation, standard errors, z-values, and p-values for each predictor. The significant, which are associated to a p-value < 0.05, are: *Sex*, *Cholesterol*, *FastingBS*, *ExerciseAngina*, the encoded features from *ChestPainType*, and *ST_Slope*. Note that, the positive coefficient for *Sex* suggests that being "male" increases the probability in occurring in heart disease.

4.3 Probit Regression Model

Now, we perform a probit regression model. The probit link assumes a cumulative normal distribution and models the probability of heart disease based on the same predictors used in the logit model.

Listing 6: Probit regression model

```
probit_model <- glm(HeartDisease ~ Age + Sex + RestingBP + Cholesterol +
    FastingBS +
                        MaxHR + ExerciseAngina + Oldpeak + ChestPainTypeATA +
                        ChestPainTypeNAP + ChestPainTypeTA + RestingECGST +
                        RestingECGNormal + ST_SlopeFlat + ST_SlopeUp,
                        family = binomial(link = "probit"), data = data)

# Model summary
summary(probit_model)
```

The probit regression output, shown in Figure 8, provides results that are (generally) comparable to those of the logit model in Figure 7. The direction and significance of predictors are consistent, though the magnitude of coefficients differs due to the different scaling of the probit link.

4.4 Model Comparison and Interpretation

To assess model fit, we calculate McFadden's pseudo-R^2 for each model, which approximates the proportion of variability explained by the model. A higher pseudo-R^2 indicates a better fit, though typical

```
Call:
glm(formula = HeartDisease ~ Age + Sex + RestingBP + Cholesterol +
    FastingBS + MaxHR + ExerciseAngina + Oldpeak + ChestPainTypeATA +
    ChestPainTypeNAP + ChestPainTypeTA + RestingECGST + RestingECGNormal +
    ST_SlopeFlat + ST_SlopeUp, family = binomial(link = "logit"),
    data = data_heart)

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -1.163656   1.416003  -0.822 0.411197
Age               0.016550   0.013197   1.254 0.209803
Sex               1.466477   0.279834   5.241 1.60e-07 ***
RestingBP         0.004194   0.006010   0.698 0.485296
Cholesterol      -0.004115   0.001087  -3.785 0.000154 ***
FastingBS         1.136482   0.274999   4.133 3.59e-05 ***
MaxHR            -0.004288   0.005023  -0.854 0.393249
ExerciseAngina    0.900292   0.244513   3.682 0.000231 ***
Oldpeak           0.380643   0.118466   3.213 0.001313 **
ChestPainTypeATA -1.830289   0.326293  -5.609 2.03e-08 ***
ChestPainTypeNAP -1.685682   0.266001  -6.337 2.34e-10 ***
ChestPainTypeTA  -1.488392   0.432572  -3.441 0.000580 ***
RestingECGST     -0.268546   0.350020  -0.767 0.442945
RestingECGNormal -0.177033   0.271925  -0.651 0.515022
ST_SlopeFlat      1.453902   0.429086   3.388 0.000703 ***
ST_SlopeUp       -0.994101   0.450196  -2.208 0.027234 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1262.14  on 917  degrees of freedom
Residual deviance:  594.19  on 902  degrees of freedom
AIC: 626.19

Number of Fisher Scoring iterations: 6
```

Figure 7: Summary for the logit regression.

```
Call:
glm(formula = HeartDisease ~ Age + Sex + RestingBP + Cholesterol +
    FastingBS + MaxHR + ExerciseAngina + Oldpeak + ChestPainTypeATA +
    ChestPainTypeNAP + ChestPainTypeTA + RestingECGST + RestingECGNormal +
    ST_SlopeFlat + ST_SlopeUp, family = binomial(link = "probit"),
    data = data_heart)

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)      -0.6941357  0.7879881  -0.881 0.378374
Age               0.0104074  0.0072708   1.431 0.152316
Sex               0.7999381  0.1529120   5.231 1.68e-07 ***
RestingBP         0.0032463  0.0033062   0.982 0.326161
Cholesterol      -0.0023617  0.0005978  -3.951 7.79e-05 ***
FastingBS         0.6515480  0.1500700   4.342 1.41e-05 ***
MaxHR            -0.0025532  0.0027769  -0.919 0.357872
ExerciseAngina    0.4909395  0.1365360   3.596 0.000324 ***
Oldpeak           0.2003742  0.0651587   3.075 0.002104 **
ChestPainTypeATA -1.0446444  0.1771400  -5.897 3.70e-09 ***
ChestPainTypeNAP -0.9755862  0.1453651  -6.711 1.93e-11 ***
ChestPainTypeTA  -0.8581897  0.2454271  -3.497 0.000471 ***
RestingECGST     -0.1983007  0.1927096  -1.029 0.303473
RestingECGNormal -0.1293297  0.1510546  -0.856 0.391899
ST_SlopeFlat      0.7853100  0.2340044   3.356 0.000791 ***
ST_SlopeUp       -0.6108959  0.2489349  -2.454 0.014126 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1262.14  on 917  degrees of freedom
Residual deviance:  593.63  on 902  degrees of freedom
AIC: 625.63

Number of Fisher Scoring iterations: 6
```

Figure 8: Summary for the probit regression.

values for logistic and probit models are lower than those in linear regression. Together with psuedo-R^2, we studied also the Confusion Matrix, ROC curve, and the corresponding AUC for each model.

Listing 7: Assessing Model Fit

```
# Calculating pseudo-R2 values for both models
# Pseudo-R2 for logit model
logit_pseudo_r2 <- pR2(logit_model)["McFadden"]
print(logit_pseudo_r2)

# Pseudo-R2 for probit model
probit_pseudo_r2 <- pR2(probit_model)["McFadden"]
print(probit_pseudo_r2)

# Predictions and confusion matrix
pred_logit <- ifelse(predict(logit_model, type = "response") > 0.5, 1, 0)
pred_probit <- ifelse(predict(probit_model, type = "response") > 0.5, 1, 0)

cm_logit <- table(Predicted = pred_logit, Actual = data_heart$HeartDisease)
cm_logit
cm_probit <- table(Predicted = pred_probit, Actual = data_heart$HeartDisease)
cm_probit

# ROC curve and AUC
roc_logit <- roc(data_heart$HeartDisease, predict(logit_model, type = "response"
    ))
plot(roc_logit)
auc(roc_logit)

roc_probit <- roc(data_heart$HeartDisease, predict(probit_model, type = "
    response"))
plot(roc_probit)
auc(roc_probit)
```

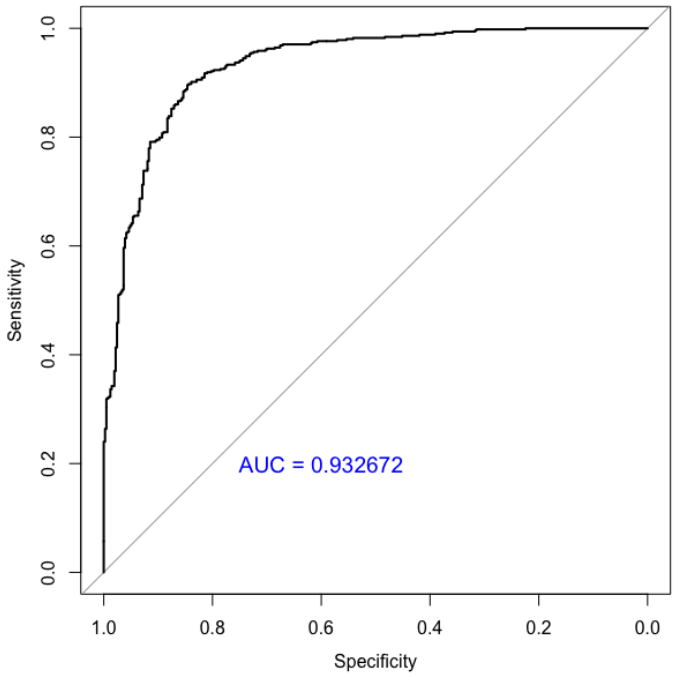

Figure 9: ROC curve for the multiple logistic regression for the occurrence of heart disease.

Both models yield similar pseudo-R^2 values:

$$\text{Logit} : R^2_{McF} = 0.5292228 \qquad \text{Probit} : R^2_{McF} = 0.5296606$$

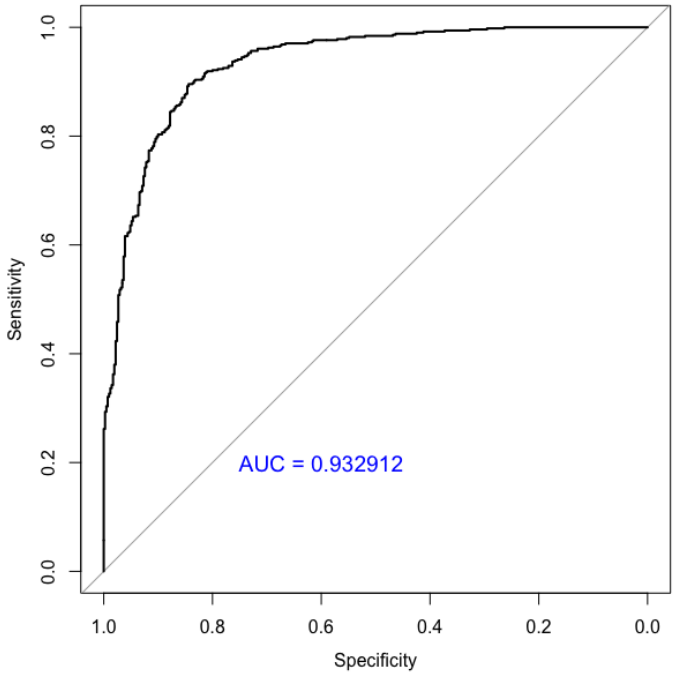

Figure 10: ROC curve for the multiple probit regression for the occurrence of heart disease.

as well as the ROC curve and AUC, shown in Figure 9 and 10, and the confusion matrix depicted in Table 1.

Logit Model			**Probit Model**		
	Actual			**Actual**	
Predicted	0	1	**Predicted**	0	1
0	341	50	0	341	49
1	69	458	1	69	459

Table 1: Confusion Matrices for Logit and Probit Models

These results suggest a comparable fit. The choice between the two depends on the nature of the problem and the assumptions about the distribution of the latent variable. In general, the logistic model is often preferred in applied research due to its interpretability and ease of computation, whereas the probit model may be favored in contexts like Bayesian statistics or when working with ordinal models.

5 Ordinal Logit/Probit Regression

The Ordinal Logit[2] Model is used when the response variable is categorical and ordered. This model is particularly suitable for cases where the response variable represents categories with a natural order (e.g., "low," "medium," "high") but without known distances between these categories. The primary goal of this model is to estimate the cumulative probability of belonging to each level or lower levels of the response variable.

5.1 Model Structure

The Ordinal Logit Model is based on the cumulative logit link function, modeling the cumulative probability of each level j as follows:

$$\text{logit}(P(Y \leq j)) = \alpha_j - X\beta$$

where:

[2]From now on, we will deal with logit regression, but all the discussion might be applied to the probit framework.

- α_j represents the intercepts specific to each level j of the response variable,
- $X\beta$ is the linear combination of the predictor variables.

This model relies on the *proportional odds assumption*, meaning that the effects of each predictor on the odds remain constant across all categories of the response variable.

5.2 Application on real data: Wine Quality dataset

To illustrate Ordinal Logistic Regression, we consider a dataset evaluating the wine quality on an ordinal scale (from 1 to 3) based on chemical characteristics such as acidity, sugar content, pH, and alcohol content. Figure 11 represents the structure of the complete database "Wine Quality", available within its documentation at the following link: https://tinyurl.com/bp52yt52.

```
> str(df_wne)
'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             : num  2 2 2 2 2 2 2 1 1 2 ...
```

Figure 11: R-Studio console showing `str(df_wne)`. This R-function shows the global structure of the database's features with the first ten values each.

5.2.1 Data Preparation and Exploratory Data Analysis

The data were preprocessed to verify discrepancies, missing values, asses the correct order of the response variable, and the standardization of the numerical features to ensure the efficiency of the model. For our purpose, the feature *quality* represents the response and it is an ordered categorical variable from 1, lowest quality, to 3, highest quality.

Before starting with the implementation of the model, as usual, we perform an Exploratory Data Analysis (EDA) to show up relations and behavior of the database. At first, we compute the correlation[3] matrix, as shown in Figure 12. The correlation matrix in Figure 12 provides an overview of the linear relationships between the different chemical properties in the wine quality dataset. For example, we can note that *alcohol* is negatively correlated with *density* (-0.33), highlighting that wines with higher alcohol content tend to have lower density. These insights are useful for understanding which chemical properties might jointly influence wine quality, guiding the variable selection in the upcoming ordinal logistic regression model.

We are also interested in the distribution of the predictors paired with the ordered response variable. This step is fundamental to verify the balances between different classes of the outcome *quality*. Strong unbalances in the database between response classes might affect the result of the regression, and they have to be cured with balance algorithms (for example SMOTE). Categories' distributions are shown in the pairplot in Figure 13 and in the histogram in 14.

5.2.2 Model Implementation in R

The ordinal regression is performed using the `polr` function (from the `MASS` package) for Ordinal Logistic Regression. The R implementation is the following:

[3]Kendall's correlation is preferred over Pearson's correlation when the outcome variable is ordinal and categorical because Pearson's method assumes a linear relationship and equal intervals between values, which may not hold for ordinal data. Kendall's correlation is non-parametric and evaluates the strength and direction of association based on rank concordance, making it more suitable for ordinal data where the exact distance between levels is not defined or meaningful.

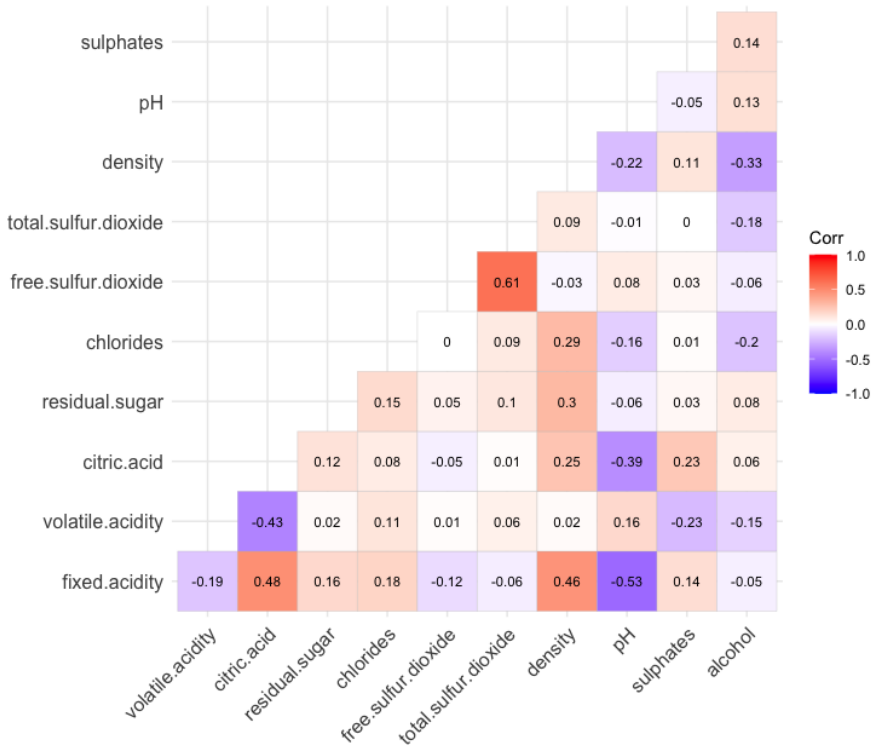

Figure 12: Correlation matrix for the features into wine quality database.

Listing 8: Ordinal Logit R-function

```r
df_wne$quality <- factor(df_wne$quality, levels = c(1, 2, 3), ordered = TRUE)
ordinal_model <- polr(as.factor(quality) ~ ., data = df_wne, Hess = TRUE)
# Display model summary
summary(ordinal_model)
```

5.3 Interpretation of Coefficients

The results of the estimated model are shown in Figure 15.

We observe that some variables, such as *volatile.acidity*, *chlorides*, *density*, *pH*, and *sulphates*, are statistically significant in determining the quality level of wine. For instance, an increase in volatile acidity is associated with an increase in the likelihood of receiving a higher quality rating.

5.4 Model Evaluation

Several evaluation tests were conducted to assess the model's goodness-of-fit, in particular, we computed the AIC, the BIC, the ANOVA test on the Log-Likelihood function, the Accuracy, and the Confusion Matrix between the ordinal categories. At first, we illustrate the implementation in R, then we discuss the single results.

Listing 9: Goodness of fit - ordinal logit

```r
# 1. Log-Likelihood and Likelihood Ratio Test
null_model <- polr(as.factor(quality) ~ 1, data = df_wne, Hess = TRUE)
likelihood_ratio_test <- anova(null_model, ordinal_model)
cat("Likelihood Ratio Test:")
print(likelihood_ratio_test)

# 2. BI
bic_value <- BIC(ordinal_model)
cat("BIC:", bic_value, "")

# 4. Compute Accuracy, deviance and the confusion matrix
```

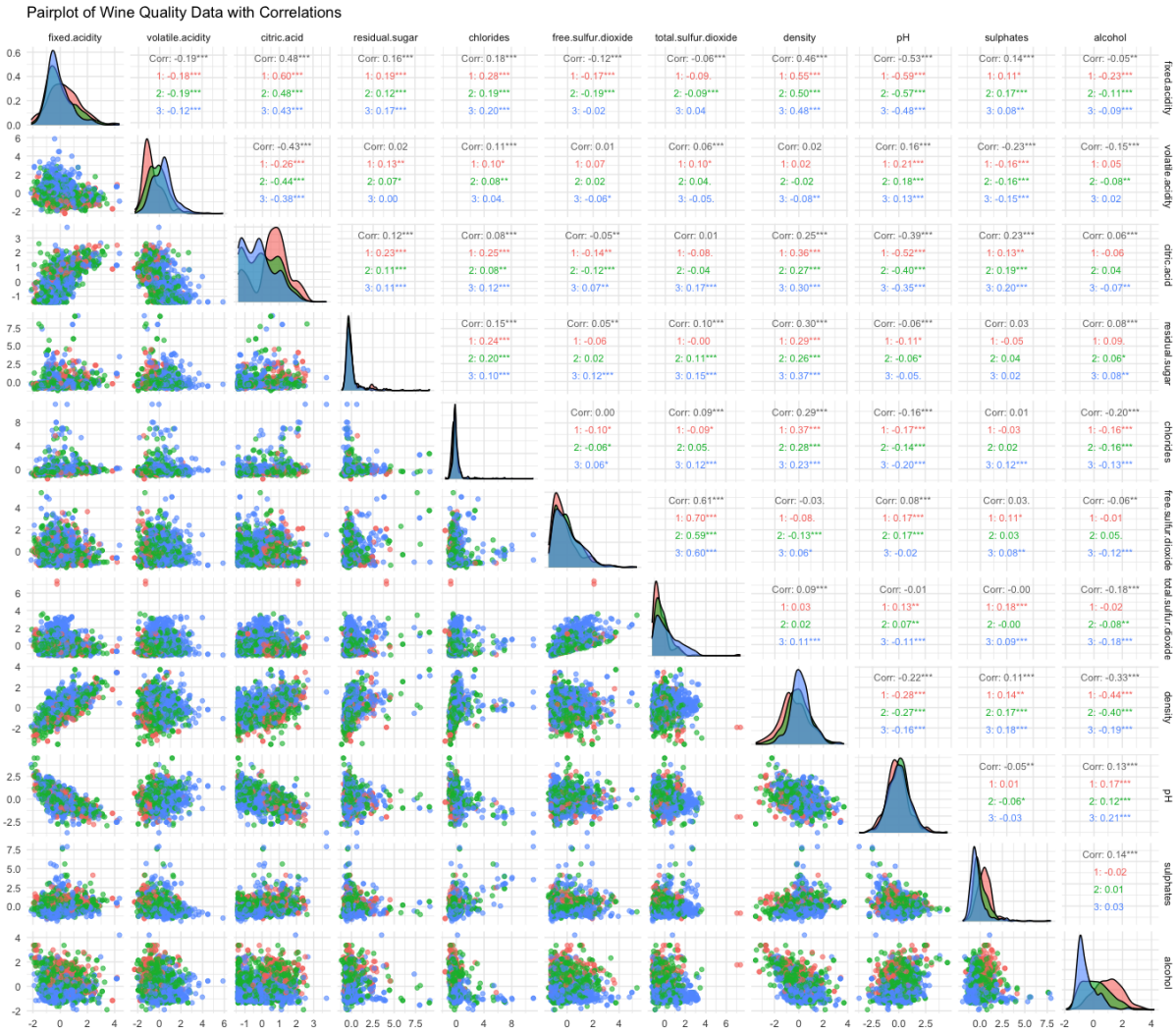

Figure 13: This plot illustrates the pairwise relationships and distributions of various chemical properties of wine. The scatter plots in the off-diagonal cells provide insights into potential linear or non-linear associations between pairs of variables, while, the diagonal shows the distributions of individual variables. Correlation coefficients are presented for each variable pair, indicating the strength and direction of their relationship. Significant correlations (p-values) are marked with asterisks to denote levels of significance: $*(p < 0.05), **$ $(p < 0.01)$, and $***$ $(p < 0.001)$. This visualization aids in understanding the interplay between wine characteristics and quality, offering insights for further analysis and model development.

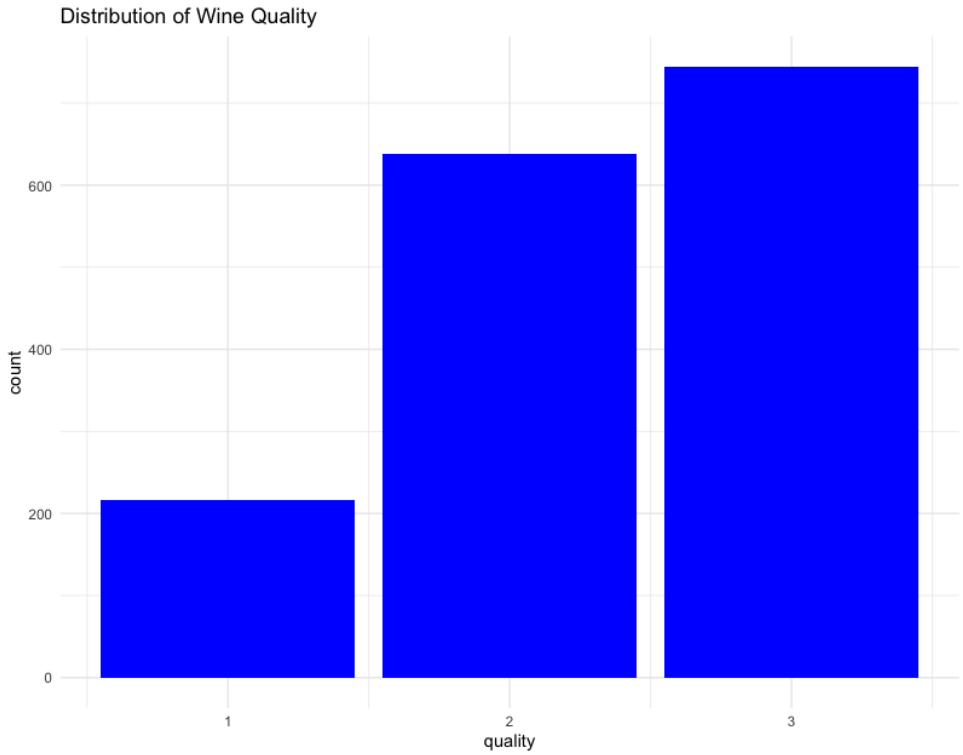

Figure 14: Histogram of the outcome categories. The database is not perfectly balanced, having a low number of items in the first class, concerning "low" quality. This might affect the final results.

```
Call:
polr(formula = as.factor(quality) ~ ., data = df_wne, Hess = TRUE)

Coefficients:
                       Value Std. Error t value
fixed.acidity        -0.32848    0.14687 -2.2365
volatile.acidity      0.55358    0.07822  7.0773
citric.acid           0.15989    0.09644  1.6579
residual.sugar       -0.17556    0.06932 -2.5325
chlorides             0.24074    0.06839  3.5200
free.sulfur.dioxide  -0.18164    0.07699 -2.3593
total.sulfur.dioxide  0.56071    0.09149  6.1284
density               0.22024    0.13337  1.6514
pH                    0.03573    0.09597  0.3723
sulphates            -0.53846    0.06392 -8.4239
alcohol              -0.87226    0.09296 -9.3829

Intercepts:
    Value    Std. Error t value
1|2  -2.6226   0.0980   -26.7633
2|3   0.2779   0.0615     4.5186

Residual Deviance: 2448.005
AIC: 2474.005
```

Figure 15: Summary if the result for the ordinal logit regression on `wine_quality` database.

```r
# Predicts the categories with the highest probability
pred_classes <- predict(ordinal_model, type = "class")
confusion_table <- table(pred_classes, df_wne$quality)
accuracy <- sum(diag(confusion_table)) / sum(confusion_table)

cat("Accuracy (Misclassification Rate):", accuracy, "")
cat("Confusion Matrix:")
print(confusion_table)
```

1. **Likelihood Ratio Test**: We compare through an ANOVA test the null model (which includes no predictors) with the full model. We obtain a likelihood ratio statistic LR = 729.6366 with a very low p-value ($p < 0,0001$), indicating that the model with predictors explains significantly more variance than the null model.

2. **AIC and BIC**: The AIC value is 2474.005, and the BIC value is 2543.908, indicating good model parsimony. Parsimony refers to the model being both simple (fewer parameters) and effective in explaining the variability in the data. The relatively close values imply consistency and further support the model's appropriateness.

3. **Accuracy and Confusion Matrix**: The model achieves an accuracy of 63.85%, as shown by the confusion matrix in Table 2. In general, an accuracy $\leq 70\%$ may not be acceptable. In this specific case, this mid-low value of the accuracy is due to the imbalance between classes. Possible solutions concern taking into account different metrics (F1-score, Recall etc.) or directly reducing imbalance with specific techniques.

	Real Class		
Predicted Class	**1**	**2**	**3**
1	73	39	2
2	133	383	177
3	11	216	565

Table 2: Confusion Matrix for the Ordinal Logistic Regression model.

Despite the low accuracy of the model, due to the imbalance in the categories, we saw that the ordinal logit is a powerful tool for analyzing ordered categorical data. The results obtained indicate that chemical properties such as acidity and alcohol content significantly influence wine quality, allowing for effective product classification.

6 Multonomial Logistic Regression

The Multinomial Logit Model (MLM) is applied when the response variable is categorical with more than two unordered classes. This model is useful when the outcome categories are nominal (e.g., "continent" with values such as "America", "Asia", etc.) and do not have a natural ordering. The primary aim is to estimate the probabilities of each class given a set of predictors and to analyze how predictor variables impact the likelihood of belonging to each category.

6.1 Model Structure

The MLM extends logistic regression by estimating the probability of each category k relative to a baseline category (often the first category) for the response variable. For each level k of the outcome, the log odds are modeled as follows:

$$\log\left(\frac{P(Y=k)}{P(Y=\text{baseline})}\right) = \alpha_k + X\beta_k$$

where:

- α_k represents the intercept for category k,
- $X\beta_k$ is the linear combination of the predictor variables specific to category k.

In this model, each predictor variable has a different effect for each category relative to the baseline.

6.2 Empirical Analysis: Predicting Continent

To illustrate multinomial logistic Regression, we consider a dataset concerning society characteristics, such as *life expectation*, *health expenditure*, *gdp per capita*, and others, among the different continents, which will be considered as the multinominal categories of our regression model. The aim is to test the capability of the multinomial logit regression of R in classifying correctly the continents ("America", "Asia", "Europe", "Africa", and "Oceania") by the values of the predictors.

6.2.1 Data Preparation

Loaded the dataset (available, within documentation at https://tinyurl.com/bp52yt52), the respective header is shown in Figure 16.

Listing 10: Loading and overviewing of the dataset

```r
# Load dataset
df_multi <- read.csv("/Users/claudiomazzi/Documents/PhD/Course_AppSTAT/2_GLM/db/
    multinomial_logit_nation.csv")

# Convert 'continent' to factor (usually unnecessary)
df_multi$continent <- as.factor(df_multi$continent)

# Overview of the dataset
head(df_multi)
```

```
> head(df_multi)
  continent life_expectancy gdp_per_capita health_expenditure education_index
1   America        70.87047       20995.30          11.153114       0.5889424
2    Europe        67.00993       16998.92          10.622500       0.8752270
3      Asia        70.91761       20349.01          16.157762       0.7935678
4   America        50.12431       18073.43          12.239150       0.8271555
5   America        67.80328       20567.59           9.744165       0.7721672
6    Europe        73.57113       23310.65           8.088919       0.5870948
```

Figure 16: Printed console for `head(df_multi)`, to show the first six rows of the database *multinomial_logit_nation*.

6.2.2 Model Implementation in R

The multinomial logistic regression model was implemented using the `multinom` function from the `nnet` package in R. The code for fitting the model is described in the following:

Listing 11: Multinomial logit model with `multinom`

```r
# Fit multinomial logit model
library(nnet)
model_multi <- multinom(continent ~ life_expectancy + gdp_per_capita +
                    health_expenditure + education_index, data = df_multi)
# Summary of results
summary(model_multi)
```

We perform a complete regression, using all the dataset's features as predictors to assess the influence of each variable in the model. Note that the multinomial regression may be compared with a port unsupervised learning model, capable of clustering observations into different categories. The result are shown in Figure 17.

```
> model_multi <- multinom(continent ~ life_expectancy + gdp_per_capita +
+                     health_expenditure + education_index, data = df_multi)
# weights:  24 (15 variable)
initial  value 277.258872
iter  10 value 274.192066
iter  20 value 268.064366
final  value 268.062889
converged
> # Summary of results
> summary(model_multi)
Call:
multinom(formula = continent ~ life_expectancy + gdp_per_capita +
    health_expenditure + education_index, data = df_multi)

Coefficients:
        (Intercept) life_expectancy gdp_per_capita health_expenditure education_index
Asia     -0.3396313     0.045459215  -3.111715e-05         -0.0827550       -2.198424
Europe    2.4040005     0.020143089  -2.342385e-05         -0.1448784       -2.638844
Oceania   5.9682019    -0.009515308  -5.930584e-05         -0.1377276       -4.102676

Std. Errors:
         (Intercept) life_expectancy gdp_per_capita health_expenditure education_index
Asia    0.0001082069      0.01087313   3.786066e-05        0.001113073    8.119172e-05
Europe  0.0001062446      0.01043649   3.540864e-05        0.001012950    7.856013e-05
Oceania 0.0001178169      0.01093045   3.707611e-05        0.001072316    8.323859e-05

Residual Deviance: 536.1258
AIC: 566.1258
```

Figure 17: Results of the multinomial logit regression performed on the *multinomial_logit_nation*.

The results of the multinomial logistic regression model provide insights into the relationship between predictors and the response variable *continent*. The model was successfully fitted using the `multinom`, and it converged after 20 iterations. The improvement in the model's log-likelihood over iterations, starting from an initial value of 277.258872 and reaching a final value of 268.062889, indicates that the model was able to find parameter estimates that maximize the likelihood of the data under the specified framework. The residual deviance of 536.1258 and the Akaike Information Criterion (AIC) of 566.1258 provide measures of model fit and parsimony, with lower values indicating better performance relative to alternative models.

6.3 Interpretation of Coefficients

The estimated coefficients, as shown in Figure 17, represent the effects of the predictors on the log-odds of belonging to a specific continent relative to the baseline category *America* (following alphabetic order, if not specified). The intercepts provide the baseline log-odds for each continent when all predictors are set to zero. Oceania has a highly positive intercept (5.9682019), suggesting substantially higher baseline log-odds relative to the reference category, while Asia and Europe exhibit lower intercepts (-0.3396313 and 2.4400005, respectively). The predictor *life_expectancy* has a positive coefficient for Asia (0.045459215), indicating that an increase in life expectancy is associated with higher log-odds of being in Asia relative to the baseline. Similarly, the coefficient for Europe (0.020143089) suggests a modest positive relationship. Conversely, Oceania's coefficient for life expectancy is slightly negative (-0.009515308), indicating a small reduction in the log-odds of belonging to this continent as life expectancy increases. The predictor *gdp_per_capita*, with coefficients close to zero across all continents, shows minimal influence on the log-odds, suggesting that it does not strongly differentiate between the categories. The variable *health_expenditure* has negative coefficients for all continents (-0.0827550 for Asia, -0.1448784 for Europe, and -0.1377276 for Oceania), implying that higher health expenditure is associated with lower log-odds of being in any of these continents relative to the baseline. This indicates a consistent inverse relationship between health expenditure and the likelihood of belonging to the modeled categories. Finally, the predictor *education_index* demonstrates a substantial negative impact across all continents, with coefficients of -2.198424 for Asia, -2.638844 for Europe, and -4.102676 for Oceania. These values suggest that higher education index levels significantly decrease the log-odds of belonging to these continents relative to the baseline. The magnitude of these coefficients underscores the importance of this variable in distinguishing between categories.

The standard errors associated with the coefficients are relatively small, indicating precise estimates and high confidence in the results. For example, the standard error for the intercept in Asia is 0.0001082069, which is considerably smaller than the corresponding coefficient, highlighting the stability of the estimate, as double-checked by the p-values computed as follows, and shown in Figure.

Listing 12: p-value for multinomial logit

```r
# Coefficients and standard errors
coefficients <- summary(model_multi)$coefficients
std_errors <- summary(model_multi)$standard.errors
# Z-values
z_values <- coefficients / std_errors
# p-value
p_values <- 2 * (1 - pnorm(abs(z_values)))
# Print p-value
print(p_values)
```

	(Intercept)	life_expectancy	gdp_per_capita	health_expenditure	education_index
Asia	0	2.903878e-05	0.4111418	0	0
Europe	0	5.359901e-02	0.5082729	0	0
Oceania	0	3.840097e-01	0.1096940	0	0

Figure 18: Console print of the p-value for each predictor concerning the baseline "America".

Finally, we can compute the odds ratio for each predictor in this way:

Listing 13: odds ratio

```r
    # Analysis of coefficients - computing odds ratio
exp(coef(model_multi))
```

The odds ratios provide insight into the direction and strength of association with each continent category. For instance, a higher *education_index* is associated with a decreased likelihood of belonging to "Oceania" compared to the baseline category, while an increase in *life_expectancy* increases the odds of being in "Asia" relative to the baseline.

Studying the coefficients offers interpretability, indeed, the analysis suggests that life expectancy and education index are likely key predictors of continent classification, while GDP per capita appears to have minimal influence in this context. The overall performance of the model should be further evaluated. In the next section, we compute predictions and compare them with the observed frequencies for each continent. Moreover, it would be useful to provide also a complete confusion matrix, the overall accuracy of the model, the AUC, and the LR test, as performed in the previous analysis on wine quality.

6.4 Efficiency Evaluation

To evaluate the model, we compute the predicted probabilities for each observation and classify each into the continent with the highest predicted probability.

Listing 14: Predicted probabilities Vs Observed frequencies

```r
# Predicted probabilities and class predictions
predicted_probs <- predict(model_multi, type = "probs")
head(predicted_probs)

# Adding probabilities to database
df_multi$predicted_continent <- colnames(predicted_probs)[apply(predicted_probs,
    1, which.max)]

# Observed frequencies
observed_counts <- table(df_multi$continent)
observed_df <- as.data.frame(observed_counts)
colnames(observed_df) <- c("continent", "observed_count")

# Predicted frequencies
predicted_counts <- table(df_multi$predicted_continent)
```

```r
predicted_df <- as.data.frame(predicted_counts)
colnames(predicted_df) <- c("continent", "predicted_count")
```

We visualize the comparison between observed and predicted counts using a bar chart, as shown in Figure 19.

```r
library(ggplot2)
combined_df <- merge(as.data.frame(observed_counts), as.data.frame(predicted_
    counts), by.x = "Var1", by.y = "Var1")
colnames(combined_df) <- c("continent", "observed_count", "predicted_count")
combined_long <- tidyr::pivot_longer(combined_df,
                                     cols = c("observed_count", "predicted_count
                                         "),
                                     names_to = "Type",
                                     values_to = "Count")
combined_long$Type <- recode(combined_long$Type, "observed_count" = "Observed",
    "predicted_count" = "Predicted")

# Plot observed vs predicted counts
ggplot(combined_long, aes(x = continent, y = Count, fill = Type)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.9), width =
      0.8) +
  labs(x = "Continent", y = "Count", title = "Observed vs Predicted Counts by
      Continent") +
  scale_fill_manual(name = "Legend", values = c("Observed" = "blue", "Predicted"
       = "red")) +
  theme_minimal()
```

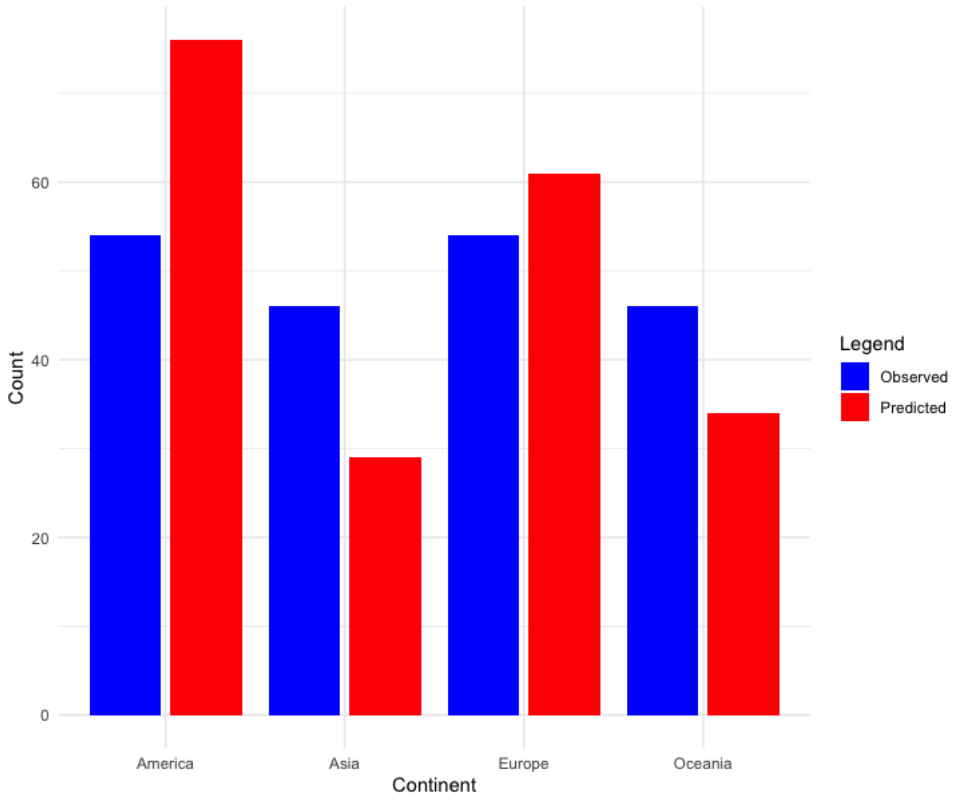

Figure 19: Histogram comparing observed against predicted frequencies for each continent in the multinomial logit.

As expected the model shows an acceptable capability in selecting the correct nominal predictor, with only a notable imbalance in the first two categories “America”, and “Asia”, showing respectively over-, and under-estimation.

7 Conclusion

In this exercise, we demonstrated how to perform different regressions in the framework of Generalized Linear Models, such as logit, probit, ordered logit/probit, and multinomial logit. We applied regression models on different datasets showing the peculiarity, pros, and cons of each model. In the next exercise, we will focus on GLM with Poisson and over-dispersed Poisson distribution function.