Group coursework

Computational Statistics

Undergraduate degree in Mathematics, 2024-25

University of Granada

Pablo Gálvez Ortigosa

Mario Megías Mateo

José Luis Mera Cardoso

Eduardo Rodríguez Cao

June 3, 2025

Contents

1 Introduction **3**

2 The dataset **3**

3 The model **5**

4 Statistical Analysis **6**

4.1 Model fitting 6

4.2 Interpretation of coefficients 8

4.3 Goodness of the fit 8

4.4 Basic inference 9

4.5 Model diagnostics 9

4.6 Possible model simplification 14

5 Conclusions **16**

1 Introduction

This project focuses on using logistic regression to analyze the UCI Heart Disease dataset. Our goal is to understand which patient factors are linked to the presence of heart disease and to build a model that can help predict it. Since the response variable is binary (disease or no disease), linear regression can not be applied, so logistic regression is a suitable choice in this case.

The topic connects directly to what we have learned in our Computational Statistics course, especially in the areas of regression modeling and prediction. Also we refer to the work of [3], *Notes for Predictive Modeling*, to guide our model building and model diagnostics.

The dataset is interesting because it has real-world relevance and allows us to apply statistical tools to a health-related problem. It also gives us the chance to explore key steps in data analysis, such as variable selection, checking model assumptions, and interpreting results clearly.

2 The dataset

As mentioned in the introduction, we are working with the UCI Heart Disease dataset. It consists of 14 variables measured on 297 patients. Although the original dataset contains 303 instances, the version available in the *kmed* R package excludes 6 patients due to missing values. The full dataset can be accessed in [1]. The most important variable is `class`, which indicates whether a patient has heart disease. Originally, this variable has five levels (from 0 to 4), but it can be simplified into a binary outcome by recoding values greater than 0 as 1.

This dataset is particularly interesting, as it allows us to apply many of the concepts learned throughout the course in a context that is both meaningful and practical: detecting heart disease. As evidence of its relevance, the dataset has received a total of 64 citations and over 775000 views on [1]. It has been used in various studies employing a wide range of models and algorithms, such as Agglomerative Clustering [2] and privacy-preserving clinical decision-making with cloud support [4].

After introducing the dataset, we load the `kmed` library, which contains the data, and verify that it matches the description previously provided:

```
> library(kmed)
> str(heart)

'data.frame': 297 obs. of  14 variables:
 $ age     : num  63 67 67 37 41 56 62 57 63 53 ...
 $ sex     : logi  TRUE TRUE TRUE TRUE FALSE TRUE ...
 $ cp      : Factor w/ 4 levels "1","2","3","4": 1 4 4 3 2 2 4 4 4 4 ...
 $ trestbps: num  145 160 120 130 130 120 140 120 130 140 ...
 $ chol    : num  233 286 229 250 204 236 268 354 254 203 ...
 $ fbs     : logi  TRUE FALSE FALSE FALSE FALSE FALSE ...
 $ restecg : Factor w/ 3 levels "0","1","2": 3 3 3 1 3 1 3 1 3 3 ...
 $ thalach : num  150 108 129 187 172 178 160 163 147 155 ...
 $ exang   : logi  FALSE TRUE TRUE FALSE FALSE FALSE ...
 $ oldpeak : num  2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...
 $ slope   : Factor w/ 3 levels "1","2","3": 3 2 2 3 1 1 3 1 2 3 ...
 $ ca      : num  0 3 2 0 0 0 2 0 1 0 ...
 $ thal    : Factor w/ 3 levels "3","6","7": 2 1 3 1 1 1 1 1 3 3 ...
 $ class   : int  0 2 1 0 0 0 3 0 2 1 ...
 - attr(*, "na.action")= 'omit' Named int [1:6] 88 167 193 267 288 303
  ..- attr(*, "names")= chr [1:6] "88" "167" "193" "267" ...
```

Now, we apply the transformation mentioned earlier, converting the original four-class response variable into a binary one:

```
> heart$class <- ifelse(heart$class > 0, 1, 0)
> unique(heart$class)

[1] 0 1
```

Even though the dataset is expected to exclude missing values, we perform a check to confirm that no missing data remains:

```
> colSums(is.na(heart))

     age      sex       cp trestbps     chol      fbs  restecg
       0        0        0        0        0        0        0
 thalach    exang  oldpeak    slope       ca     thal    class
       0        0        0        0        0        0        0
```

3 The model

The model we are implementing is the logistic regression. At first, we are going to provide the mathematical formulation of the model and, lately, the explanation of the assumptions made for it.

Suppose Y is a Bernoulli variable and $X_1, ..., X_p$ are predictors associated to Y. Our goal is to model the conditional expectation:

$$\mathbb{E}[Y \mid X_1 = x_1, \ldots, X_p = x_p] = \mathbb{P}[Y = 1 \mid X_1 = x_1, \ldots, X_p = x_p]$$

which corresponds to the probability that the outcome is 1 given the values of the predictors.

Seeing what we want to model, the first natural idea might be to consider a linear regression model:

$$\mathbb{E}[Y \mid X_1 = x_1, \ldots, X_p = x_p] = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p =: \eta$$

However, this model is not suitable, as it can produce predicted values outside the $[0, 1]$ interval, which is not valid for probabilities. To address this, we consider applying a function that maps η to the interval $[0, 1]$. Specifically, we use the *logistic function*:

$$\mathbb{P}[Y = 1 \mid X_1 = x_1, \ldots, X_p = x_p] = \frac{1}{1 + e^{-\eta}}.$$

This value represents the model's estimate of the probability that the outcome is $Y = 1$ given the observed inputs. To classify the observation, we compare this probability to a threshold, typically 0.5. The decision rule is:

- If $\mathbb{P}[Y = 1 \mid X_1, \ldots, X_p] > 0.5$, the model predicts $Y = 1$.
- If $\mathbb{P}[Y = 1 \mid X_1, \ldots, X_p] \leq 0.5$, the model predicts $Y = 0$.

This threshold-based classification approach allows us to turn the continuous output of the model into a binary decision.

Apart from the model itself, it is also important to understand how the *odds* behave in logistic regression, since the coefficients are directly related to them:

- The quantity e^{β_0} represents the odds of the outcome being 1 when all the predictor variables are equal to zero.

- For any predictor X_j, the value e^{β_j} tells us how much the odds multiply when X_j increases by one unit, assuming all other variables remain constant. More generally, if X_j increases by r units, the odds are multiplied by $(\mathrm{e}^{\beta_j})^r$.

In summary, if $\beta_j > 0$, then $\mathrm{e}^{\beta_j} > 1$, which means increasing X_j leads to higher odds (and thus a higher probability) of the outcome being 1. If $\beta_j < 0$, then $\mathrm{e}^{\beta_j} < 1$, so increasing X_j decreases the odds and the probability of $Y = 1$.

Once all the mathematical foundations behind logistic regression are clear, it is important to explain the assumptions required for using the model, as stated in [3]. There are a total of three assumptions:

- Linearity in the transformed expectation:

$$\mathbb{E}[Y \mid X_1 = x_1, \ldots, X_p = x_p] = \frac{1}{1 + \mathrm{e}^{-\eta}} = \frac{1}{1 + \mathrm{e}^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)}}$$

- $Y_1, \ldots, Y_n$ are independent, conditionally on $X_1, \ldots, X_n$.

- The response distribution must be:

$$Y \mid (X_1 = x_1, \ldots, X_p = x_p) \sim Ber\left(\frac{1}{1 + \mathrm{e}^{-\eta}}\right)$$

4 Statistical Analysis

4.1 Model fitting

As previously explained, we will fit a logistic regression model using the variable `class` as the response variable, and all the remaining variables as predictors. We show the summary of the fitted model:

```
> modelo_log <- glm(class ~ ., data = heart, family = binomial)
> summary(modelo_log)



Call:
glm(formula = class ~ ., family = binomial, data = heart)

Coefficients:
```

```
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.029468   2.891768  -2.085  0.03707 *
age         -0.013763   0.024745  -0.556  0.57808
sexTRUE      1.546014   0.529995   2.917  0.00353 **
cp2          1.239566   0.770874   1.608  0.10784
cp3          0.245959   0.663312   0.371  0.71078
cp4          2.086480   0.666547   3.130  0.00175 **
trestbps     0.024364   0.011269   2.162  0.03062 *
chol         0.004448   0.003993   1.114  0.26526
fbsTRUE     -0.596246   0.607848  -0.981  0.32664
restecg1     0.810202   2.435102   0.333  0.73935
restecg2     0.473895   0.383518   1.236  0.21659
thalach     -0.017723   0.011109  -1.595  0.11065
exangTRUE    0.709456   0.440018   1.612  0.10689
oldpeak      0.357875   0.230070   1.556  0.11983
slope2       1.155286   0.473794   2.438  0.01475 *
slope3       0.525147   0.919661   0.571  0.56798
ca           1.311510   0.279276   4.696 2.65e-06 ***
thal6       -0.010974   0.790210  -0.014  0.98892
thal7        1.392715   0.425194   3.275  0.00105 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 409.95  on 296  degrees of freedom
Residual deviance: 191.64  on 278  degrees of freedom
AIC: 229.64

Number of Fisher Scoring iterations: 6
```

The summary output shows that the first step of the logistic regression model is to compute the linear predictor:

$$
\begin{aligned}
z = &-6.03 - 0.014 \cdot \text{age} + 1.55 \cdot \text{sex} + 1.24 \cdot \text{cp}_2 + 0.25 \cdot \text{cp}_3 + 2.09 \cdot \text{cp}_4 + 0.024 \cdot \text{trestbps} \\
&+ 0.004 \cdot \text{chol} - 0.60 \cdot \text{fbs} + 0.81 \cdot \text{restecg}_1 + 0.47 \cdot \text{restecg}_2 - 0.018 \cdot \text{thalach} + 0.71 \cdot \text{exang} \\
&+ 0.36 \cdot \text{oldpeak} + 1.16 \cdot \text{slope}_2 + 0.53 \cdot \text{slope}_3 + 1.31 \cdot \text{ca} - 0.011 \cdot \text{thal}_6 + 1.39 \cdot \text{thal}_7,
\end{aligned}
$$

Once this linear combination z is calculated, it is transformed using the logistic function, which gives the estimated probability $\hat{p}$ that `class = 1`. If $\hat{p} > 0.5$, the model predicts `class = 1`. Otherwise, it predicts `class=0`.

In the previous expression, categorical variables with multiple categories—such as `cp`, `restecg`, `slope`, and `thal`—are represented by separate binary variables that indicate the presence of

each category (except the baseline one). For example, $\text{cp}_2 = 1$ if cp equals 2, and 0 otherwise. Similarly, $\text{cp}_3 = 1$ if cp equals 3, and so on. The category not shown (in this case, $\text{cp} = 1$) serves as the reference category.

Logical variables like `sex`, `fbs`, and `exang` are coded as 1 when `TRUE`, and 0 when `FALSE`.

4.2 Interpretation of coefficients

As we explained before, a positive coefficient means that the variable increases the probability that `class = 1` (i.e., the person has heart disease), while a negative coefficient means it decreases that probability. For example, higher values of `ca` increase the probability of having heart disease, while higher values of `thalach` decrease it.

For categorical variables, the coefficient shows the effect of each category compared to a reference category. A positive value means a person in that category is more likely to have heart disease than someone in the reference category, and a negative value means they are less likely to have heart disease.

4.3 Goodness of the fit

The `summary` function provides some indicators of the model's fit, such as:

- Null deviance: Measures the fit of a model with only the intercept.
- Residual deviance: Measures the fit of the current model with predictors.
- AIC (Akaike Information Criterion): Balances model fit and complexity; lower is better.

Although the residual deviance is much lower than the null deviance — suggesting that the model fits the data better than a model with no predictors — these values are mainly useful for comparing models. They do not directly measure predictive performance. Therefore, we also compute metrics such as `accuracy` and the `confusion matrix` to evaluate how well the model classifies observations.

```
> probabilidades <- predict(modelo_log, type = "response")
> pred_clase <- ifelse(probabilidades >= 0.5, 1, 0)
> real <- heart$class
```

```
>
> accuracy <- mean(pred_clase == real)
> print(paste("Accuracy:", round(accuracy, 4)))

[1] "Accuracy: 0.8721"

> confusion_matrix <- table(Predicted = pred_clase, Actual = heart$class)
> print(confusion_matrix)

         Actual
Predicted   0   1
        0 146  24
        1  14 113
```

The confusion matrix shows that the model correctly classified 146 individuals without heart disease and 113 individuals with heart disease. It misclassified 14 cases as having the disease when they did not (false positives), and 24 cases as not having the disease when they actually did (false negatives). With an overall accuracy of 87.21%, the model demonstrates good predictive performance.

4.4 Basic inference

We assess the statistical significance of each predictor using the $\Pr(>|z|)$ values from the model summary. At a significance level of 0.001, the only significant variable is `ca`, indicating a strong association with the response variable.

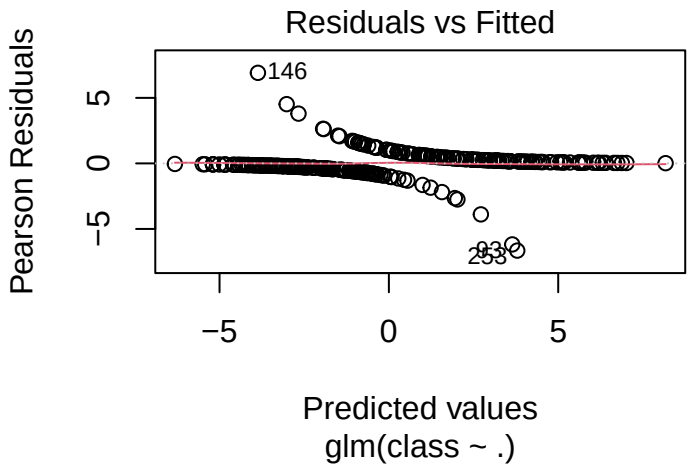
If we increase the significance level to 0.01, additional variables become significant: `sex`, `cp4`, and `thal7`, suggesting moderate evidence of their effect on the probability of heart disease.

At a 0.05 level, even more variables show significance: the intercept, `trestbps`, and `slope2`, meaning that these predictors also contribute meaningfully to the model.

4.5 Model diagnostics

Now, we perform model diagnostics to determine if the assumptions associated with the logistic regression are met. We begin by looking at the residual plot.

```
> plot(modelo_log, which=1)
```

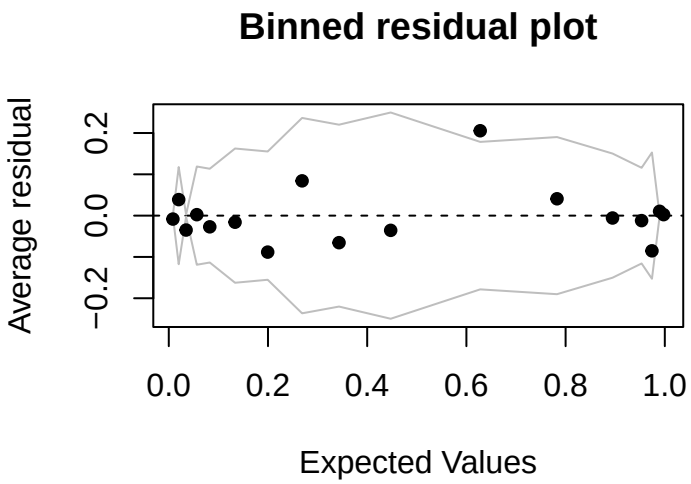


As we can see, it is not very helpful when used with the logistic regression. The following is an excerpt from the documentation of the binned residual plot from the `arm` package.

> In logistic regression, as with linear regression, the residuals can be defined as observed minus expected values. The data are discrete and so are the residuals. As a result, plots of raw residuals from logistic regression are generally not useful. The binned residuals plot instead, after dividing the data into categories (bins) based on their fitted values, plots the average residual versus the average fitted value for each bin.

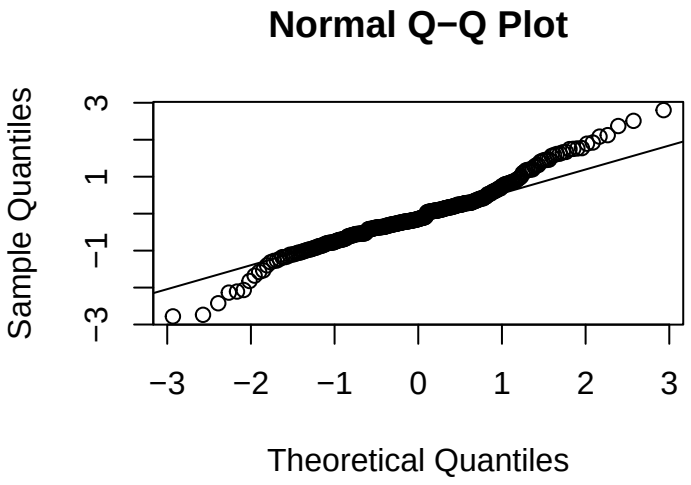
```
library(arm)

binnedplot(fitted(modelo_log), residuals(modelo_log, type = "response"),
           nclass = NULL, xlab = "Expected Values", ylab = "Average residual",
           main = "Binned residual plot", cex.pts = 0.8, col.pts = 1,col.int = "gray")
```



The grey lines indicate ± 2 standard-error bounds. If the model is true, then it is expected that 95% of the data should be contained inside. As such, the binned plot above suggests that our model is reasonable enough. Next, we verify the normality of the residuals with `qqnorm` and `qqline`.

```
> qqnorm(rstandard(modelo_log))
> qqline(rstandard(modelo_log))
```



The plot suggests there is no normality. We can confirm this observation with a Kolmogorov-Smirnov test.

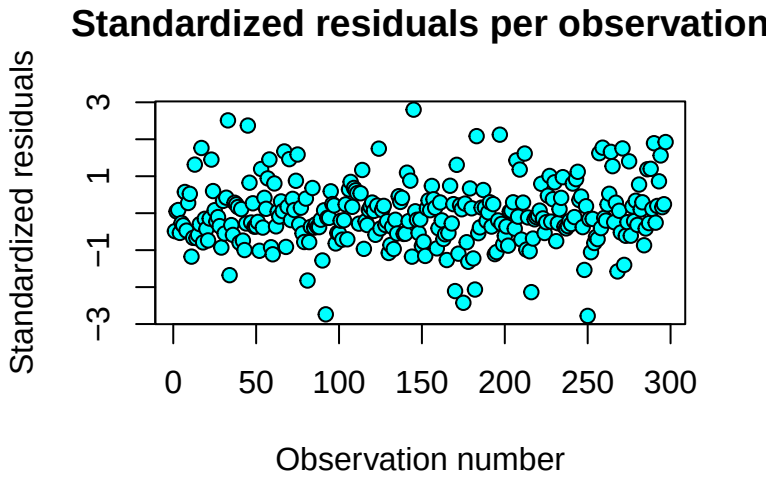
```
> ks.test(rstandard(modelo_log), "pnorm")


 Asymptotic one-sample Kolmogorov-Smirnov test

data:  rstandard(modelo_log)
D = 0.122, p-value = 0.0002891
alternative hypothesis: two-sided
```

The resulting p-value is small enough to reject the null hypothesis of normality (which is possible even if the model is perfectly correct [3]). Now, we proceed with the identification and deletion of potential outliers. We first take a look at the standardised residuals plot.

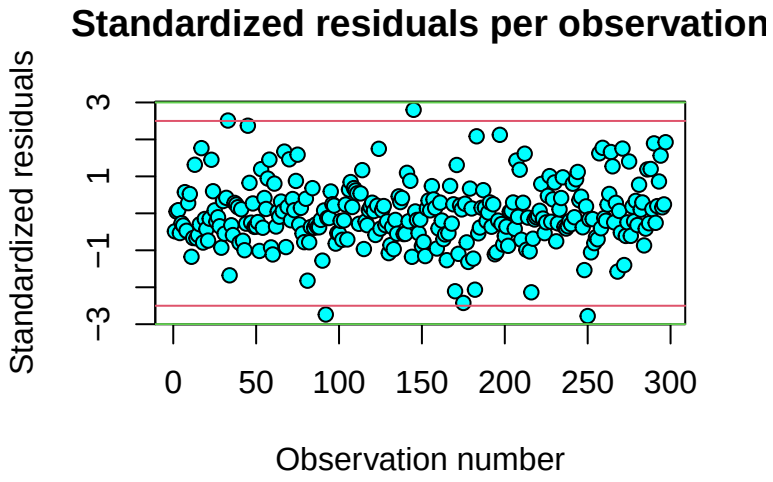
```
> plot(rstandard(modelo_log), main='Standardized residuals per observation',
+      xlab='Observation number', ylab='Standardized residuals', pch=21, bg='cyan')
```



We observe a random pattern, which suggests that the residuals may be independent. We can also identify quite a few outliers here. With a threshold of 2.5, we visually separate the outliers with the greatest residuals.

```
> plot(rstandard(modelo_log), main='Standardized residuals per observation',
+      xlab='Observation number', ylab='Standardized residuals', pch=21, bg='cyan')
> abline(h=-2.5, col=2)
> abline(h=-3, col=3)
```

```
> abline(h=2.5, col=2)
> abline(h=3, col=3)
```



Next, we calculate their total number.

```
> outliers <- which(abs(rstandard(modelo_log)) > 2.5)
> length(outliers)

[1] 4
```

Now, we create another dataset without outliers and fit the model once more.

```
> heart2 <- heart[-outliers, ]
> # Modelo con todas las variables como predictores
> modelo_log <- glm(class ~ ., data = heart2, family = binomial)
> # Resumen del modelo
> summary(modelo_log)



Call:
glm(formula = class ~ ., family = binomial, data = heart2)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.124666   3.261685  -2.184 0.028936 *
age         -0.019132   0.027231  -0.703 0.482315
sexTRUE      1.706733   0.605621   2.818 0.004830 **
cp2          1.858059   0.853377   2.177 0.029458 *
```

```
cp3          0.218809   0.723359   0.302 0.762279
cp4          2.569979   0.754441   3.406 0.000658 ***
trestbps     0.030693   0.012555   2.445 0.014497 *
chol         0.003308   0.004506   0.734 0.462852
fbsTRUE     -0.772546   0.700616  -1.103 0.270171
restecg1     0.692926   3.208680   0.216 0.829024
restecg2     0.532623   0.426734   1.248 0.211980
thalach     -0.020721   0.012641  -1.639 0.101170
exangTRUE    0.823848   0.485662   1.696 0.089822 .
oldpeak      0.409745   0.258657   1.584 0.113165
slope2       1.790728   0.544765   3.287 0.001012 **
slope3       0.864547   1.032137   0.838 0.402240
ca           1.856938   0.353333   5.255 1.48e-07 ***
thal6       -0.120377   0.848831  -0.142 0.887226
thal7        1.879570   0.487547   3.855 0.000116 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 404.38  on 292  degrees of freedom
Residual deviance: 158.08  on 274  degrees of freedom
AIC: 196.08

Number of Fisher Scoring iterations: 7
```

The summary shows that the AIC has dropped from 230 to 196, which is a considerable improvement. To measure it's predictive performance, we calculate its accuracy.

```
> probabilidades <- predict(modelo_log, type = "response")
> pred_clase <- ifelse(probabilidades >= 0.5, 1, 0)
> real <- heart2$class
>
> accuracy <- mean(pred_clase == real)
> print(paste("Accuracy:", round(accuracy, 4)))

[1] "Accuracy: 0.8805"
```

There is a small improvement in accuracy, it has gone up from 0.87 to 0.88.

4.6 Possible model simplification

Finally, we attempt to to simplify the model using a stepwise algorithm.

```
> step_model <- step(modelo_log, direction='both')
```

```
> summary(step_model)


Call:
glm(formula = class ~ sex + cp + trestbps + thalach + exang +
    oldpeak + slope + ca + thal, family = binomial, data = heart2)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.24772    2.62675  -2.759 0.005794 **
sexTRUE      1.65292    0.57250   2.887 0.003887 **
cp2          1.90244    0.83834   2.269 0.023251 *
cp3          0.14586    0.70460   0.207 0.836005
cp4          2.67637    0.74358   3.599 0.000319 ***
trestbps     0.02762    0.01142   2.419 0.015556 *
thalach     -0.01726    0.01148  -1.504 0.132615
exangTRUE    0.77833    0.47724   1.631 0.102912
oldpeak      0.44447    0.24948   1.782 0.074816 .
slope2       1.82084    0.53395   3.410 0.000649 ***
slope3       0.80102    1.01436   0.790 0.429714
ca           1.74519    0.32449   5.378 7.52e-08 ***
thal6       -0.38496    0.81719  -0.471 0.637586
thal7        1.83114    0.47135   3.885 0.000102 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 404.38  on 292  degrees of freedom
Residual deviance: 162.06  on 279  degrees of freedom
AIC: 190.06

Number of Fisher Scoring iterations: 7
```

We obtain a simpler model with 9 predictors (instead of the previous 14), most of which are statistically significant. The summary also shows a slight improvement in model fit, as the AIC has decreased to 190.

5 Conclusions

In this project, we worked with the UCI Heart Disease dataset, chosen for its clinical relevance and availability in the kmed R package. After converting the original response variable (0–4) into a binary one, we applied a logistic regression model, a method not previously studied in our coursework.

We explored the theory behind logistic regression and its assumptions, fitted the model, and interpreted the results. In particular, we discussed the meaning of some coefficients in terms of their impact on the probability of having heart disease. For example, males are about 4.7 times more likely to have heart disease than females (since $\mathrm{e}^{1.55} \approx 4.7$). Similarly, individuals with chest pain type 4 (`cp = 4`) are about 8.1 times more likely to have heart disease compared to those with chest pain type 1 (since $\mathrm{e}^{2.09} \approx 8.1$). We also identified 7 variables (including the intercept) as statistically significant at conventional significance levels.

To evaluate the goodness of fit, we computed the model's accuracy and confusion matrix. The model correctly classified 146 out of 170 healthy individuals and 113 out of 137 with heart disease. Misclassifications were relatively low, with 24 false negatives and 14 false positives, which resulted in a solid overall accuracy of 87.21%.

We then carried out model diagnostics to check whether the assumptions of logistic regression were reasonably met. This mainly included inspecting the residuals of the model. Overall, the diagnostic checks suggested that the model was generally appropriate for the data.

Finally, we identified the individuals with the largest residuals, treating them as potential outliers. After removing these outliers, we proceeded to simplify the model, reducing it from 14 predictors to 9, most of which were statistically significant.

While the logistic regression model provided interpretable results and solid performance, one limitation is that we evaluated it on the same data used for training. In future work, we could split the dataset into training and test sets, or use *cross-validation*, to better assess the model's predictive ability on unseen data. Additionally, more advanced machine learning models, such as *random forests* or *support vector machines*, could be applied to potentially improve accuracy. Finally, a more detailed analysis of the outliers, rather than simply removing them, could provide additional insights into the data or model limitations.

References

[1] Robert Detrano. Uci heart disease dataset, 1989. URL `https://archive.ics.uci.edu/dataset/45/heart+disease`.

[2] Ni Ding. A submodularity-based agglomerative clustering algorithm for the privacy funnel. 2019.

[3] Eduardo García Portugués. Notes for predictive modeling. 2025.

[4] Jui Ma. Ppcd: Privacy-preserving clinical decision with cloud support. 2019.