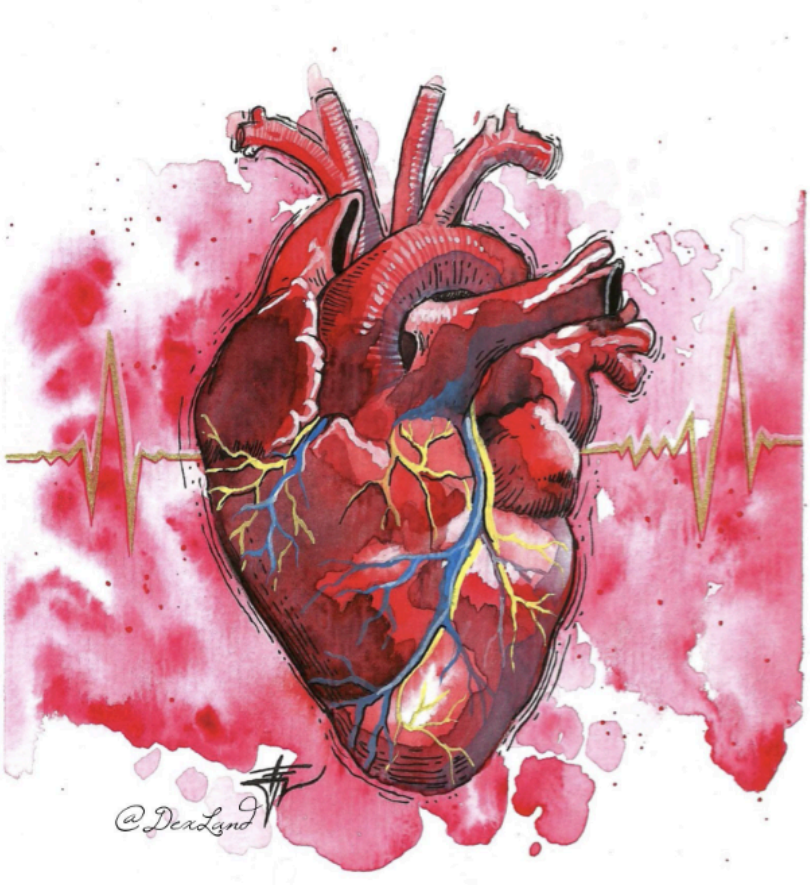
HEART DISEASE

FINAL COURSE PROJECT

STATISTICAL MODELS AND STOCHASTIC PROCESSES

Ainhoa López Carrasco



1.Introduction

What is heart disease? Also known as CVD, this term refers to several type of heart conditions with examples such as high blood pressure, stroke or vascular dementia, all of which have negative effects on heart health and blood circulation.[1]

CVD remains as the leading cause of global mortality, representing 31% of all deaths worldwide, with approximately 18 million deaths per year. The main cause of CVD is the obstruction caused by the accumulation of fatty deposits on the inner walls of blood vessels, preventing regular blood flow to the heart or brain. We have to be aware of this issue because heart problems can affect everyone around the world.[3]

We need pathological and clinical information in order to perform reliable diagnoses, and that's why our goal for this project is to forecast the significant predictors to construct the most effective model.

1.2. Techniques

Logistic Regression analysis and Maximum Likelihood Estimation (MLE)

We're using Logistic Regression and Maximum Likelihood Estimation to study our dataset. The patients are labeled with 1 for dying due to a condition of heart disease and 0 for surviving. Logistic regression will help us understand how the variables are connected and predict if someone has died due to that predictor. Maximum Likelihood Estimation estimates the model parameters by maximizing the likelihood function, and we will use it to contrast the efficiency of the logistic regression analysis. To make sure we are using the best possible models, we will also require likelihood ratio tests, such as ANOVA tests.

2. Dataset

This dataset was obtained from the UC Irvine Machine Learning Repository.[2]

- It contains 13 features which its data is obtained from different observations on 299 patients:

Feature	Type	Description	Units/Values
1. Age	Integer	Age of the patient	Years
2. Anemia	Binary	Decrease of red blood cells or hemoglobin	0 = False 1 = True
3. Creatinine Phosphokinase	Integer	Level of the CPK enzyme in the blood	mcg/L
4. Diabetes	Binary	If the patient has diabetes	0 = False 1 = True
5. Ejection fraction	Integer	Percentage of blood leaving the heart at each contraction	%
6. High blood pressure	Binary	If the patient has hypertension	0 = False 1 = True
7. Platelets	Continuous	Platelets in the blood	kilo platelets/mL
8. Serum Creatinine	Continuous	Level of serum creatinine in the blood	mg/dL
9. Serum Sodium	Integer	Level of serum sodium in the blood	mEq/L
10. Sex	Binary	Woman or man	0 = Woman 1 = Man
11. Smoking	Binary	If the patient smokes or not	0 = False 1 = True
12. Time	Integer	Follow-up period	Days
13. Death Event	Binary	If the patient died during the follow-up period	0 = False 1 = True

3. Data Analysis

We will check each predictor individually. Then, our goal will be to construct a model that uses many predictors, aiming to predict the probability of experiencing death due to heart disease.

Age

When we think of heart diseases, what comes to mind first is an image of elderly people. But is this common thought true? We were determined to prove it statistically:

What this histogram is showing is a bar for each age group. The part covered in red means that they survived, and the part covered in blue means that they perished. The size of the bar is linked to the count for each age group, meaning that the tallest bars are the ones with the highest number of people. By only looking at the plot we can observe that as older the people get, the lower the probability of survival is and by recognizing this pattern we expect a relationship between the variables "age" and "DEATH_EVENT". *FIGURE 1*

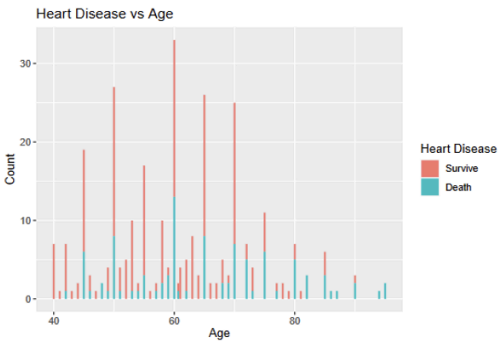


Figure 1 - Barplot Heart Disease vs Age

That's why we performed a logistic regression test, to confirm the relationship we expect.

```
Call:
glm(formula = formula_age, family = "binomial", data = heart_data)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.65433    0.70662  -5.172 2.32e-07 ***
age          0.04695    0.01107   4.241 2.23e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 375.35  on 298  degrees of freedom
Residual deviance: 355.99  on 297  degrees of freedom

                2.5 %      97.5 %
(Intercept) -5.082954 -2.30468335
age          0.025692  0.06923512
```

```
MLE Parameters: -3.654989 0.04695763
```

Figure 2 - R result: Age

Since the p-value is statistically significant for α = 0.05, we have a clear indication that the predictor "age" is significant. We also computed the confidence interval for 95% confidence to double-check the significance and we can confirm that since the CI does not include 0, the significance is not due to random chance and does in fact have a relationship with the response variable. We also compared the estimated coefficients of the logistic regression by computing the MLE parameters in R and we obtained the same coefficients thus meaning the approach we took is correct and ratifies the relationship between the variables. *FIGURE 2*

To visualize the relationship, we've computed a plot with a logistic regression curve, plotted some data points and created a decision boundary striped line to determine if the predictor is significant. As we can clearly see, the huge steeps the curve takes confirms the significance of the predictor "age". *FIGURE 3*

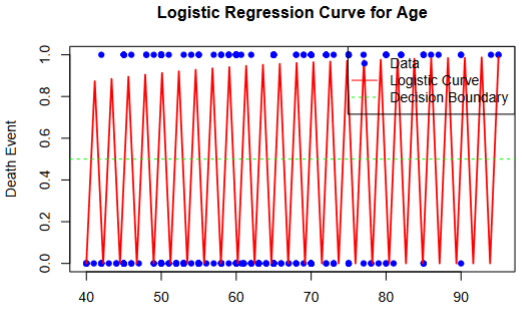


Figure 3 – Logistic regression curve: Age

Anemia

Anemia is a condition in which the body does not produce the necessary amount of healthy red blood cells. This lack of oxygen might make you feel weaker or tired[4]. We wanted to check if having anemia was a significant predictor by statistical analysis:

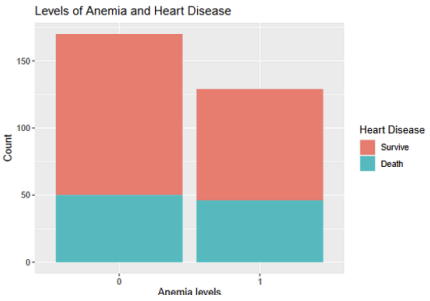


Figure 4 - Boxplot Levels of Anemia vs heart disease

The data in this plot is in binary format, value 0 meaning that they do not have the condition and value 1 meaning that they do. The color scheme applies the same as the last plot, red being survival and blue being death. We can observe that the red color is prominent when compared to the blue one, thus we can predict that anemia will not be a significant predictor of heart disease since there is no clear relationship between the predictor "anemia" and the response variable "DEATH_EVENT". *FIGURE 4*

When we performed logistic regression analysis on the "anemia" predictor, we found that the p-value is not statistically significant for α = 0.05, so we can not reject the null hypothesis and conclude that the predictor does not have a significant relationship with the response variable "DEATH_EVENT". Moreover, we can clearly see that in the 95% confidence interval we computed that it includes the value 0, so keeping in mind the obtained p-value, there is no true effect in the population and any observed variation in the sample could be explained by random chance. When computing the MLE parameters we obtain, yet again, the same estimated coefficients as in logistic regression *FIGURE 5*.

```
Call:
glm(formula = formula_anemia, family = "binomial", data = heart_data)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.8755     0.1683  -5.201 1.98e-07 ***
anaemia       0.2853     0.2492   1.145    0.252
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 375.35  on 298  degrees of freedom
Residual deviance: 374.04  on 297  degrees of freedom

                2.5 %     97.5 %
(Intercept) -1.2136721 -0.5522640
anaemia     -0.2040502  0.7746349

MLE Parameters: -0.8755349 0.2853027
```

Figure 5 - R result: Anemia

In the following plot, we can see that the peaks in the logistic regression curve tend to lower values of the death event, thus confirming the non-significance of the predictor. This can also be interpreted as a gradually changing logistic regression curve, which endorses that the predictor does not significantly influence the probability of the positive outcome, in this case, having a relationship with "DEATH_EVENT". The peaks we observe indicate points where the probability is relatively higher, this being caused by a wide confidence interval and creating uncertainty about the impact. *FIGURE 6*

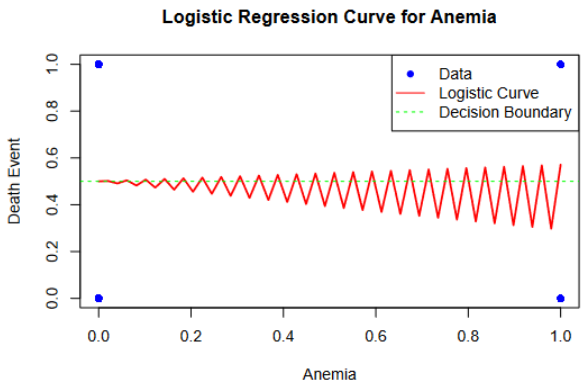


Figure 6 – Logistic regression curve: Anemia

Creatine Phosphokinase

Creatine Phosphokinase (also known as CPK) is an enzyme in the body found mainly in the heart, brain and skeletal muscle. When the total CPK level is extremely high, it often means that there's been an injury or stress on muscle tissue, the heart or the brain. When a muscular organ such as the heart is damaged, CPK leaks into the bloodstream.[5]

In our data set, the variable was called "creatinine_phosphokinase", so we will be referring to CPK as such on any occasion that CPK appears on the R file. In the following plot we observe that when having lower concentrations of CPK the DEATH_EVENT is unlikely when compared to the other extrema, this being high concentrations of CPK which leads to no survival cases. *FIGURE 7*

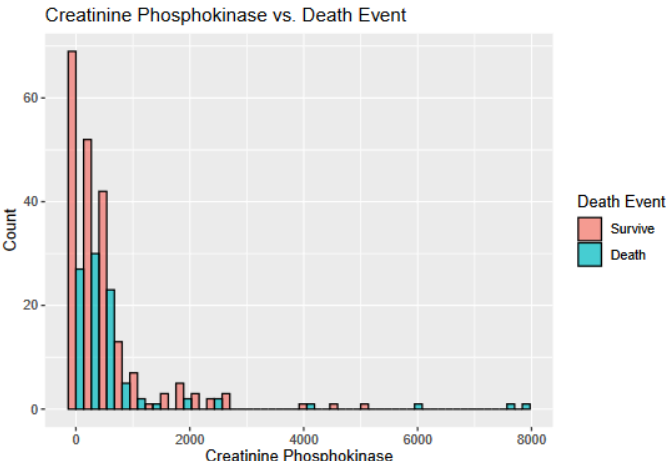


Figure 7 – Histogram Creatine phosphokinase vs heart disease

This indicates a possible relationship between the variables, but to confirm it we will perform a logistic regression analysis. This analysis shows us that although the relationship might seem clear by only looking at the plot, the reality is far from what we expected since the p-value is not significant for α = 0.05. Moreover, when we compute the confidence interval for 95% confidence we can clearly see that the value 0 is inside the interval. This suggests that there's a chance that the effect on the population is random and not due to the significance of the predictor. We confirm that our model has been built correctly because the MLE parameters coincide with the estimated coefficients in the logistic regression. *FIGURE 8*

```
Call:
glm(formula = formula_creatinine, family = "binomial", data = heart_data)

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)              -0.8265731  0.1447064  -5.712 1.12e-08 ***
creatinine_phosphokinase  0.0001297  0.0001218   1.065    0.287
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 375.35  on 298  degrees of freedom
Residual deviance: 374.23  on 297  degrees of freedom

                                2.5 %        97.5 %
(Intercept)              -1.1149446147 -0.5466385477
creatinine_phosphokinase -0.0001155848  0.0003757555

MLE Parameters: -0.8268835 0.0001300591
```

Figure 8 - R result Creatine Phosphokinase:

What we can determine from the plot below is that although the DEATH_EVENT is unlikely to occur, it is not impossible due to the increase of the peaks in the logistic regression curve. We can see that the curve starts with its highest peak just at the decision boundary striped green line, but when increasing the CPK concentration, the death event is slowly increasing. This might be due to random chance, but also due to correlation between predictors. *FIGURE 9*

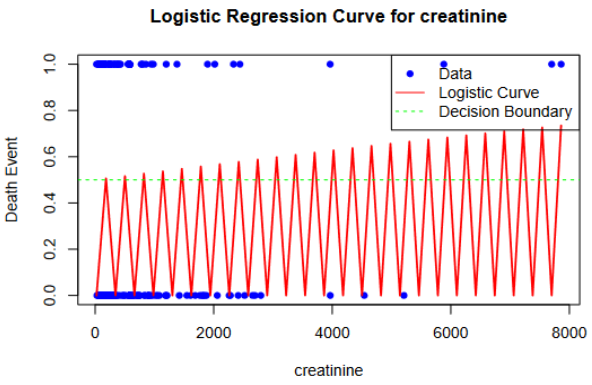


Figure 9 – Logistic regression curve: creatinine phosphokinase

Diabetes

Diabetes is a chronic metabolic disease which is characterized by the high levels of blood glucose (also known as blood sugar) which if not treated inevitably leads to detrimental damage to the heart, blood vessels, eyes, kidneys and nerves. There are 2 types of diabetes, but we will not cover the distinction between the two and just contemplate if the patient has or does not have diabetes, as shown on the plot below.[6]

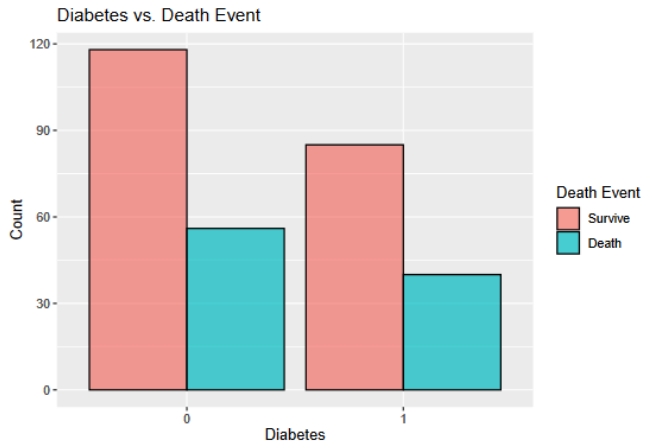


Figure 10 – Boxplot Diabetes vs heart disease

It seems clear that there is no relationship between having diabetes and the DEATH_EVENT, since people having or not having diabetes have died and also there are more survivors than deaths. It is a similar plot like the anemia plot because they share the same binary values for the variables, and also it is similar to the other plots in the sense that they share the same colors for each possible outcome. *FIGURE 10*

To test the significance of this predictor we will perform a logistic regression test. This test shows that the p-value is extremely high, being close to p-value = 1. This is an evidence for the non-significance of this predictor in predicting the outcome for the response variable, not just at α = 0.05 but at no significance level practically. Furthermore, we confirm that the value 0 is right in the middle of our confidence interval at 95% which is another indication that there is a non-existent relationship between "diabetes" and "DEATH_EVENT". *FIGURE 11*

```
Call:
glm(formula = formula_diabetes, family = "binomial", data = heart_data)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.745333   0.162270  -4.593 4.37e-06 ***
diabetes    -0.008439   0.251190  -0.034    0.973
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 375.35  on 298  degrees of freedom
Residual deviance: 375.35  on 297  degrees of freedom



                2.5 %     97.5 %
(Intercept) -1.0701093 -0.4326179
diabetes    -0.5041359  0.4823413

MLE Parameters: -0.7451029 -0.008418086
```

Figure 11 - R result: Diabetes

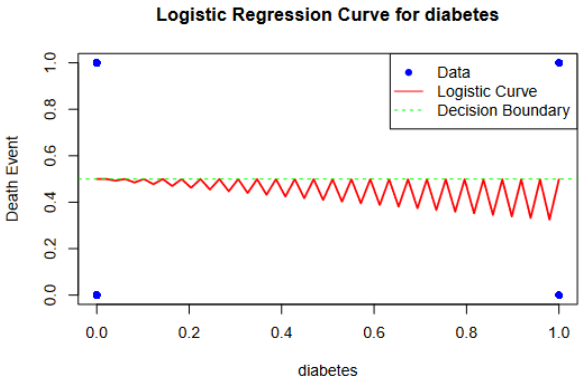


Figure 12 – Logistic regression curve: diabetes

To check if our model is right, we compare it with the MLE parameters we have also computed, and we see that they are the same coefficients as the estimated ones. When visualizing the logistic regression curve we have plotted beneath, we see that the pattern is similar to the anemia plot, a rather flattened curve that down the line has more spikes, although prominently below the decision boundary line, which indicates that the impact is not clear since the predictor is not significant and the confidence interval is very large. FIGURE 12

Ejection Fraction

The ejection fraction is the percentage of blood that leaves a ventricle when the heart beats and it measures the heart's ability to pump oxygen-enriched blood to the body. A normal ejection fraction ranges between 52 and 72% for males and between 54 and 74% for females, although these percentages vary within age groups. The older we get, the less ejection fraction we produce. [7]

The following boxplot shows us the percentage of ejection fraction and the death event outcome. At first glance we can appreciate that lower ejection fraction represents more death events, which is an unmistakable suggestion of a relationship between the variables. We performed a boxplot since the data was more visible this way. The light blue represents the death event while the dark blue represents the survival and we can see that the light blue box is larger than the other box, thus meaning there have been more departures than survivors. *FIGURE 13*

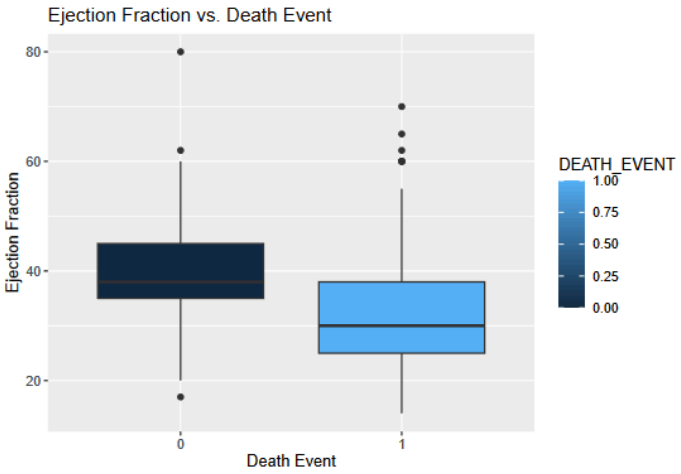


Figure 13 – Boxplot Ejection Fraction vs heart disease

```
Call:
glm(formula = formula_ejection_fraction, family = "binomial",
    data = heart_data)

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)       1.31169    0.46278   2.834  0.00459 **
ejection_fraction -0.05620    0.01258  -4.468 7.88e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 375.35  on 298  degrees of freedom

                          2.5 %      97.5 %
(Intercept)         0.42312398  2.24281588
ejection_fraction -0.08187052 -0.03242234

MLE Parameters: 1.311724 -0.05620493
```

Figure 14 - R result: Ejection fraction

To reassure our initial statement, we created a model which was used to perform a logistic regression analysis. We see that the p-value is infimum which corroborates our hypothesis. To double-check we created the confidence interval for 95% confidence and the value 0 was not inside the confidence interval, which implies that the observed effect is statistically significant and there is enough evidence to suggest a real distinction in the population and a relationship between the two variables. The MLE parameters were exactly the same as the estimated coefficients in the logistic regression analysis so we can confirm that our model was built correctly. *FIGURE 14*

As we can examine from the logistic regression curve plot, the peaks and slopes are at an all-time high in this analysis, with astounding changes throughout the curve. The steeper the curve, the more influential the predictor is in predicting the outcome, this being that the ejection fraction is a significant predictor and causes a huge impact on the response variable. *FIGURE 15*

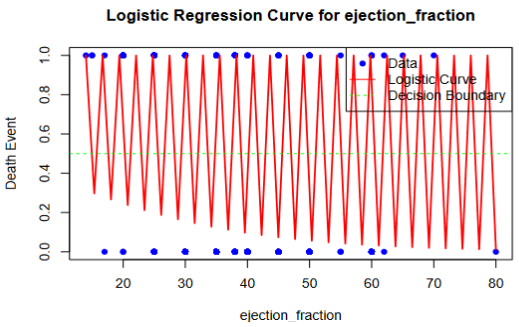


Figure 15: logistic regression curve: ejection fraction

High blood pressure

High blood pressure (also known as hypertension) is when the pressure in the blood vessels reaches a high enough level to cause harm. [8] Hypertension often develops over time and it can happen due to unhealthy lifestyle choices, such as not exercising regularly. Certain health conditions such as diabetes and obesity can also increase the risk of hypertension. [9]

Figure 16 – Boxplot High blood pressure vs heart disease

Since diabetes is a condition that can favor high blood pressure and knowing that diabetes is not a significant predictor for the response variable, we can assume that high blood pressures will not be a significant predictor either. When looking at the following barplot we see that there are much more survivors in both occasions (having or not having high blood pressure). It follows the same palette of colors as the diabetes plot and to confirm our initial guess we will carry out a logistic regression test. *FIGURE 16*

When searching for the p-value we can notice that it is not statistically significant for α = 0.05, thus confirming that our guess was well-founded. The other noticeable trait from this model is that the confidence interval for 95% confidence contains the value 0, to which we have seen in previous examples that it indicates non-significance and that there is a non-existent relationship between both variables. We also computed the MLE parameters to check if they differed from the coefficient estimates in the logistic regression model but it was not the case as we can witness. *FIGURE 17*

```
Call:
glm(formula = formula_hbp, family = "binomial", data = heart_data)

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)         -0.8769     0.1576  -5.564 2.64e-08 ***
high_blood_pressure  0.3508     0.2562   1.369    0.171
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 375.35  on 298  degrees of freedom
Residual deviance: 373.49  on 297  degrees of freedom

                       2.5 %     97.5 %
(Intercept)        -1.193088 -0.5739381
high_blood_pressure -0.153990  0.8522151

MLE Parameters: -0.8771205 0.3506951
```

Figure 17 - R result: High blood pressure

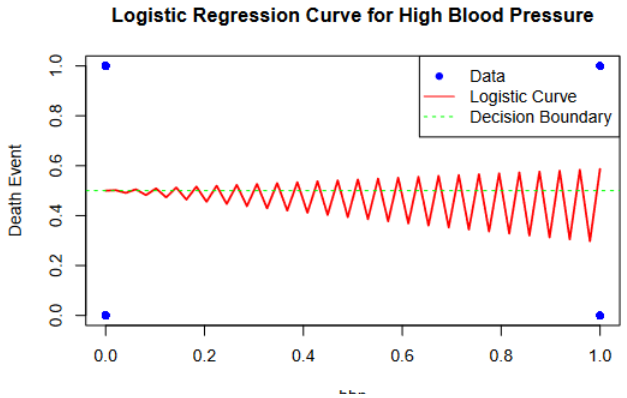


Figure 18 – Logistic regression curve: high blood pressure

When plotting the logistic regression curve, we start with a flat line right at the decision boundary line located at 0.5 in the death event axis (just like all the other decision boundary lines). The spikes begin to increase but not drastically as we have seen on the previous plot. The peaks tend to the lower values of the death event axis, thus denying any chance for high blood pressure to become a significant predictor, not only but also approving our initial guess once more. *FIGURE 18*

Platelets

Platelets (also known as thrombocytes) are colorless cell fragments that, despite its tiny size, form clots and stop bleeding. When the level of platelets is low, it prevents blood cells from clotting and can lead to vast amounts of blood loss.[10] If the platelet count is too high then blood clots can form in blood vessels, which can block blood flow through the body. The donation of platelets help cancer, trauma, transplant and burn patients to recover faster and hospitals are in need of donors. [11] [12]

This plot is pretty similar to the ejection fraction plot above, since both of them are boxplots. Although the death event box seems larger than the survival, we can confidently say that since they are on the same level, that platelets are not a significant predictor because on the same quantity of platelets there are more death events than survivors. If they were not on the same level, then we could guess that maybe it is a significant predictor.To test our hypothesis we will create a model and test it using a logistic regression test. *FIGURE 19*

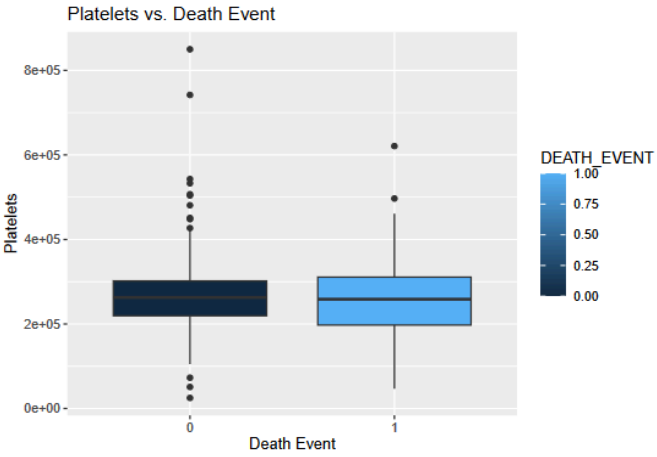


Figure 19 – Boxplot Platelets vs heart disease

```
Call:
glm(formula = formula_platelets, family = "binomial", data = heart_data)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.573e-01  3.632e-01  -1.259    0.208
platelets   -1.115e-06  1.316e-06  -0.847    0.397

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 375.35  on 298  degrees of freedom
Residual deviance: 374.61  on 297  degrees of freedom
AIC: 378.61

                    2.5 %        97.5 %
(Intercept) -1.163506e+00 2.659704e-01
platelets   -3.790860e-06 1.396279e-06

MLE Parameters: -0.4569875 -1.116187e-06
```

Figure 20 - R result: Platelets

As we can perceive from the p-value, it is not statistically significant for α = 0.05. To double-check, we compute the 95% confidence interval in order to search for the value 0 in the interval. What this information indicates is that there is no difference in the population, and any variation in the sample can be explained by random chance. Nonetheless, we computed the MLE parameters to see if the estimated coefficients matched and they did, as expected. *FIGURE 20*

The results from the logistic regression curve plot do not shock us since the peaks are situated where we expected, right under the decision boundary line. The highest death event value is situated at 0.5 which is the first one, and from there the curve decreases. We could interpret a negative correlation, since the plot shows us that having more platelets decreases the response variable, but we can discard this interpretation since the p-value is not statistically significant and the confidence interval has confirmed that this can be explained by random chance. Keeping all of this information in mind, we can conclude that there exists no relationship between the quantity of platelets and the response variable. *FIGURE 21*

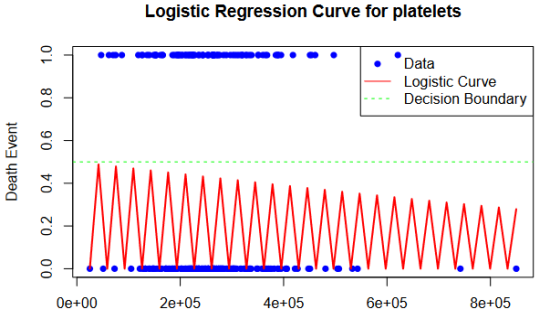


Figure 21 – Logistic regression curve: platelets

Serum creatinine

Creatinine is a waste product of the blood that comes from the muscles. Healthy kidneys filter out the creatinine of the blood through urine.[13] The serum creatinine level is based on a blood test that measures the amount of creatinine in the blood, while also providing insight into how the kidneys are currently working. High creatinine levels that reach 5.0 in adults might indicate severe kidney impairment, which can lead to chronic kidney disease.[14] The risk for cardiovascular disease is increased in all stages of the impairment of renal function, and serum creatinine is used as a marker of diabetes and coronary artery disease.[15]

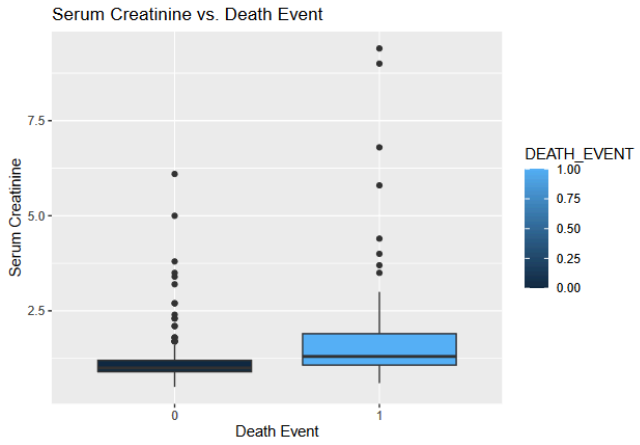


Figure 22 – Boxplot Creatinine vs heart disease

Based on previous knowledge of serum creatinine we wanted to check if it was a significant predictor of death event, which taking into account the consequences of high creatinine levels we can assume that it will. By looking at the boxplot we created we can detect that there are more death events with higher levels of serum creatinine, but also that there are more death events in general. This is a notable marker for our hypothesis, and to confirm it we built a model and performed a logistic regression analysis. *FIGURE 22*

Our assumption was right, since the p-value is very much statistically significant for α = 0.05. What we did in order to check that the significance was not due to random chance was to compute the 95% confidence interval for 95% confidence and see if the value 0 lied between the interval, which it does not. The absence of this value serves as a critical indicator for assessing the statistical significance of the results, in this case reassuring the relationship between high levels of serum creatinine and death events. The MLE parameters were proposed to validate the veracity of the model we generated and as we can see, it concurs with the estimated coefficients in our model. *FIGURE 23*

```
Call:
glm(formula = formula_serum_creatinine, family = "binomial",
    data = heart_data)

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)       -1.8917     0.2939  -6.438 1.21e-10 ***
serum_creatinine   0.8242     0.1972   4.180 2.91e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 375.35  on 298  degrees of freedom

                    2.5 %     97.5 %
(Intercept)      -2.4983867 -1.346526
serum_creatinine  0.4689544  1.239956

MLE Parameters: -1.890959 0.8238106
```

Figure 23 - R result: Serum creatinine

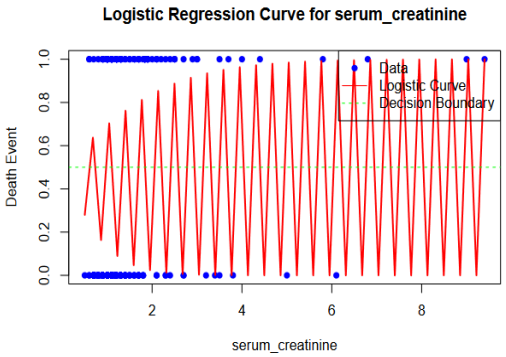


Figure 24: logistic regression curve: serum creatinine

The strong association between serum creatinine and the response variable makes the logistic regression curve plot have drastic steep changes, with the direction of the curve sloping upward and downward. This is suggestive of higher values being linked to an increased probability of the death event, thus confirming that the predictor "serum creatinine" is significant and has an effect on the response variable. *FIGURE 24*

Serum sodium

The sodium blood test measures the concentration of sodium in the blood, but it can also be measured using a urine test. This test is performed by drawing blood from a vein, usually from the inside of the elbow.[16] Hypernatremia is the condition caused by high serum sodium levels, and involves dehydration which can lead to muscle twitches and seizures. This can be caused by diarrhea, not drinking enough fluids or kidney dysfunction, among others. On the other hand, hyponatremia is the condition in which an individual has low levels of serum sodium, and symptoms may include nausea and vomiting, confusion or loss of energy. If not treated it can cause seizures, coma and even death.[17] [18]

The following boxplot represents the possible relationship between levels of serum sodium and death events. It seems plausible to assume that the relationship does in fact exist, not only due to the death event box being larger than the survivor box but also the location, since it is situated on lower levels of serum sodium. *FIGURE 25*

To clear up doubts, we conducted a logistic regression analysis to attest our assumption.

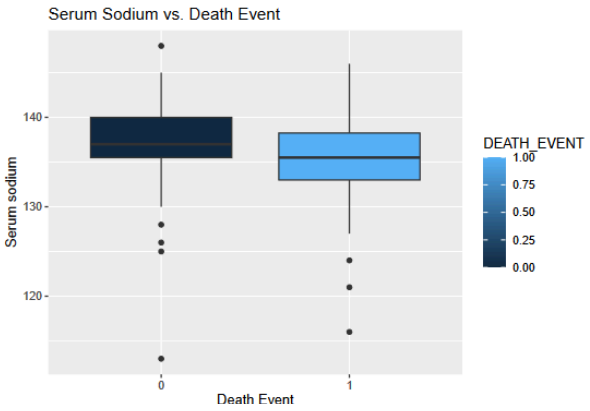


Figure 25 – Boxplot Serum sodium vs heart disease

```
Call:
glm(formula = formula_serum_sodium, family = "binomial", data = heart_data)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  12.39442    4.07264   3.043  0.00234 **
serum_sodium -0.09639    0.02989  -3.224  0.00126 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 375.35  on 298  degrees of freedom
Residual deviance: 364.02  on 297  degrees of freedom

                   2.5 %      97.5 %
(Intercept)    4.642457 20.68373443
serum_sodium -0.157266 -0.03952612

MLE Parameters: 12.39461 -0.09638622
```

Figure 26 - R result: Serum sodium

As we can detect, the p-value is statistically significant for α = 0.05 and to double-check we will calculate the 95% confidence interval for 95% confidence and validate our initial hypothesis. The interval does not include the value 0, which if it did would signify that the predictor is not significant to the changes in the response variable. When comparing the estimated coefficients to the MLE parameters we can also corroborate that they are the same coefficients, which means that our model has been conducted accordingly. *FIGURE 26*

What the logistic regression curve plot displays is a series of steep slopes, which is an obvious sign of an ongoing relationship between the two variables. A trait of this plot is starting the curve at 1.0 value of the death event (which is uncommon for this type of graphs) but it can be explained since it has a negative correlation, which means that the lower the values of serum sodium, the greater the chance of the death event occurring. *FIGURE 27*

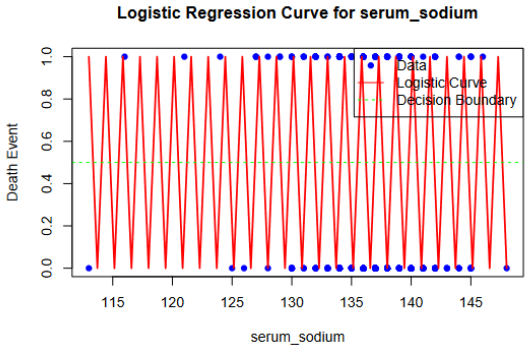


Figure 27: logistic regression curve: serum sodium

Figure 27: logistic regression curve: serum sodium

Sex

Cardiovascular disease is the leading cause of death worldwide, yet some distinctions are key between males and females. Men generally develop CVD at younger ages than women, and have a higher coronary heart disease risk. In contrast, women are at a higher risk of stroke, occurring more often at older ages.[19] Among other differences, men typically develop a plaque buildup in the large arteries that supply blood to the heart and women are more likely to develop this buildup in the heart's smallest vessels, commonly known as the microvasculature. Heart disease in both sexes is partly related to the accumulation of cholesterol and overall the presence of coronary artery disease is lower in women.[20]

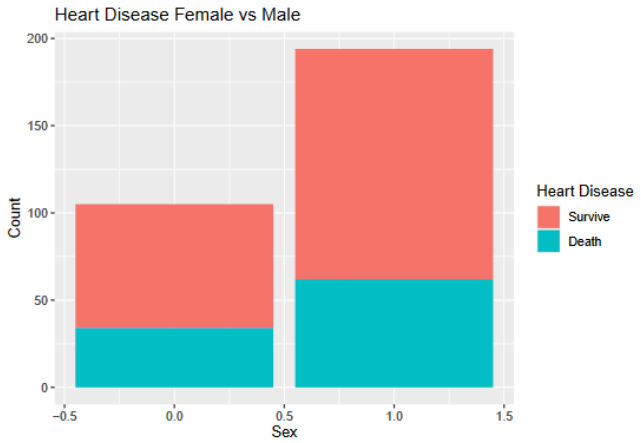


Figure 28 – Boxplot Sex vs heart disease

The barplot beside includes 2 columns, the left one being the count for women and the right one the count for men. As we observe the plot, we can clearly appreciate the difference between men and women, since there are more men than women in this dataset and also more death events in men than in women*FIGURE 28*. We can predict that it will be a significant predictor since more men than women have triggered the response variable and we will try to prove it statistically by performing a logistic regression test.

To our surprise, the p-value for the predictor variable was not statistically significant for α = 0.05, nowhere near close to being significant. This value brings us to the conclusion that the value 0 must be in the 95% confidence interval, so we will create it. With further inspection we see that inevitably the parameter 0 lies right in the middle of the confidence interval, which confirms that our hypothesis is not right and that the predictor sex is not significant for the changes in the response variable. Nevertheless, we evaluate the MLE parameters and approve the correctness of the model, since the estimated coefficients match.*FIGURE 29*

```
Call:
glm(formula = formula_sex, family = "binomial", data = heart_data)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.73632    0.20856  -3.531 0.000415 ***
sex         -0.01935    0.25923  -0.075 0.940504
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 375.35  on 298  degrees of freedom
Residual deviance: 375.34  on 297  degrees of freedom

                2.5 %     97.5 %
(Intercept) -1.1564376 -0.3359325
sex         -0.5240747  0.4943253

MLE Parameters: -0.7361556 -0.01939857
```

Figure 29 - R result: Sex

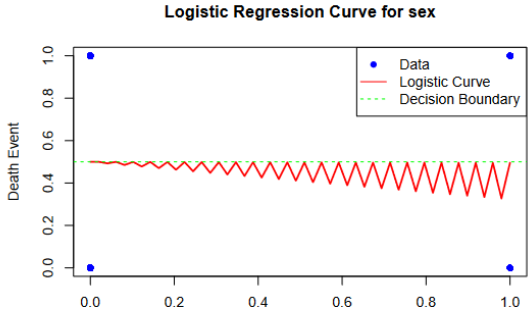


Figure 30: logistic regression curve: sex

In addition, the logistic regression curve plot helps us visualize the non-significance of the predictor, since the steeps are tending to lower values of the death event. We can also conclude that a possible impact of the predictor on the response variable can be due to random chance, since we have seen and proven by multiple methods the lack of significance of the predictor.*FIGURE 30*

Smoking

Smoking is a major cause of cardiovascular disease, and causes approximately one of every four deaths from CVD. CVD is the single largest cause of death around the world, killing more than 800,000 people a year only in the United States.[21] This activity is a major risk factor for heart disease since the chemicals inhaled when smoking cause damage to the heart and blood vessels, making you more likely to develop plaque buildup in the arteries. Cigarette smokers are twice to four times more likely to develop some type of CVD than non-smokers, and it doubles a person's risk for stroke.[22] [23]

The plot clearly illustrates that people that smoke survive far more than they die, so we can suppose that there will not be a relationship between the two variables. The non-smokers are situated on the left side of the plot, since value 0 indicates that they do not smoke, while the right side indicates that they do. Despite the previous knowledge, in this dataset it seems pretty clear that the predictor "smoking" will not be statistically significant *FIGURE 31*.
To test this hypothesis we will assemble a logistic regression model.

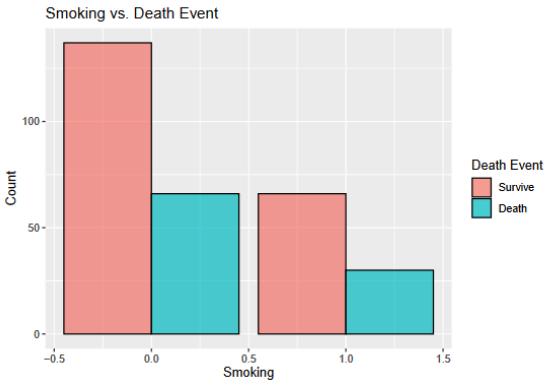


Figure 31 – Boxplot Smoking vs heart disease

```
Call:
glm(formula = formula_smoking, family = "binomial", data = heart_data)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.73033    0.14984  -4.874 1.09e-06 ***
smoking     -0.05813    0.26634  -0.218    0.827
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 375.35  on 298  degrees of freedom
Residual deviance: 375.30  on 297  degrees of freedom
```

```
                2.5 %     97.5 %
(Intercept) -1.0295959 -0.4411474
smoking     -0.5879133  0.4587681
```

```
MLE Parameters: -0.730246 -0.05828572
```

Figure 32 - R result: Smoking

As we imagined, the p-value is not statistically significant for α = 0.05. This also points out that the value 0 will lie between the CI which is not bold to predict since the predictor has no significance whatsoever on the response variable. We do in fact perceive that the value 0 falls within the range of both percentiles. Let's not forget that we have to authenticate the reliability of the model, so we are going to figure out what the MLE parameters will be. Now we can finally confirm that the model is reliable, since the MLE parameters are the spittin image of the estimated coefficients in our logistic regression model.*FIGURE 32*

Besides, the logistic regression curve plot is crucial to substantiate our hypothesis, considering that the curve lies beneath the decision boundary line, and the tendency to lower values of death events is noticeable. What makes this plot interesting is that it shows that smoking has more deaths than non-smoking, but this could be due to random chance or due to causes of death not related to heart diseases.*FIGURE 33*

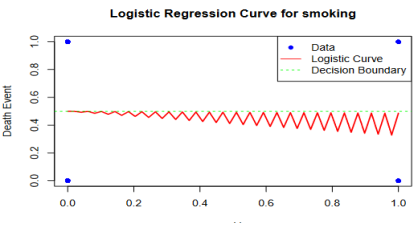


Figure 33: logistic regression curve: smoking

Final model

First of all, we created a model that included all of the predictors, and we were surprised to see that CPK had turned out to be statistically significant, and serum sodium was no longer statistically significant.

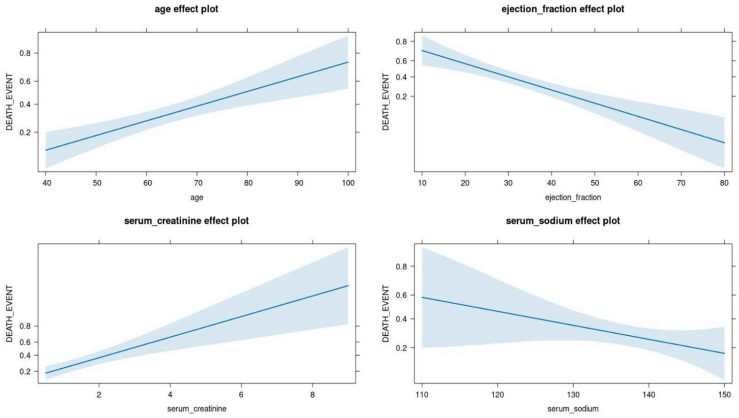


Figure 34: effect plot of the final model

We then plotted the effects of each variable on the response variable and observed that the significant predictors had a diagonal plot. The approach we took for the next model was simple, we only took into account the significant predictors and we found out that serum sodium was still not statistically significant. *FIGURE 34*

We also checked for multicollinearity which had a negative result in our data set since there is no type of correlation between significant predictor variables While iterating through all of the predictors and taking out the one that had the highest p-value without losing information, we checked if there were changes on the other p-values, which there were since there were predictors somewhat correlated to each other.*FIGURE 35*

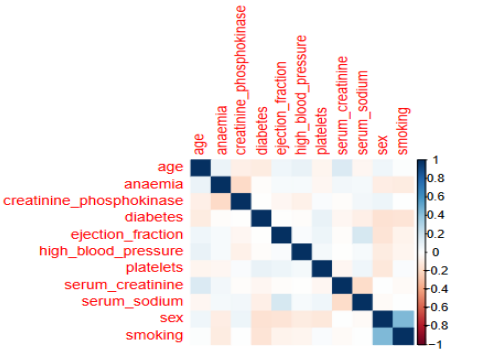


Figure 35: multicollinearity matrix of the final model

```
Call:
glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
    family = "binomial", data = heart_data)

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)       -2.35306    0.83954  -2.803  0.00507 **
age                0.05173    0.01231   4.202 2.65e-05 ***
ejection_fraction -0.07000    0.01423  -4.918 8.72e-07 ***
serum_creatinine   0.66592    0.15916   4.184 2.86e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 375.35  on 298  degrees of freedom
Residual deviance: 305.28  on 295  degrees of freedom
AIC: 313.28
```

Figure 36: R result of the final model

The final model had the following parameters which computed this prediction table and as we have proven in the R code below, it is the exact same table, with a 77% chance of correctly predicting the outcome on the response variable.*FIGURE 36*

4. Conclusions

3 out of 11 possible predictors were significant (age, ejection fraction and serum creatinine), so those were the only ones needed to predict the probability of having a heart disease. The model built forecasts 77% of the cases. By adding more predictors to the data set we could possibly increase the chance of prognosis and by doing so the efficiency and accuracy of the model would rise.

5. Bibliography

[1] How the Heart Works | Congenital Heart Defects | NCBDDD | CDC. (2018, 26 septiembre). Centers for Disease Control and Prevention. https://www.cdc.gov/ncbddd/spanish/heartdefects/howtheheartworks.html

[2] UCI Machine Learning Repository. (s. f.). https://archive.ics.uci.edu/dataset/519/heart+failure+clinical+records

[3] Cardiovascular heart disease. (s. f.). British Heart Foundation. https://www.bhf.org.uk/informationsupport/conditions/cardiovascular-heart-disease

[4] What is anemia? | NHLBI, NIH. (2022, 24 marzo). NHLBI, NIH. https://www.nhlbi.nih.gov/health/anemia

[5] Creatine phosphokinase test. (s. f.). Mount Sinai Health System. https://www.mountsinai.org/health-library/tests/creatine-phosphokinase-test

[6] World Health Organization: WHO. (2019, 13 mayo). Diabetes. https://www.who.int/health-topics/diabetes

[7] Carroll, M., PhD. (2023, 25 junio). Everything you need to know about Ejection fraction. Healthline. https://www.healthline.com/health/ejection-fraction

[8] World Health Organization: WHO & World Health Organization: WHO. (2023, 16 marzo). Hypertension. https://www.who.int/news-room/fact-sheets/detail/hypertension

[9] High blood pressure symptoms, causes, and problems | Cdc.gov. (2023, 29 agosto). Centers for Disease Control and Prevention. https://www.cdc.gov/bloodpressure/about.htm

[10] What are platelets in blood. (s. f.). https://www.redcrossblood.org/donate-blood/dlp/platelet-information.html

[11] What causes a low platelet count? (s. f.). https://www.oneblood.org/media/blog/platelets/what-causes-a-low-platelet-count.stml

[12] Thrombocythemia and thrombocytosis | NHLBI, NIH. (2022, 24 marzo). NHLBI, NIH. https://www.nhlbi.nih.gov/health/thrombocythemia-thrombocytosis

[13] American Kidney Fund. (2023, 10 noviembre). Serum creatinine test. https://www.kidneyfund.org/all-about-kidneys/tests/serum-creatinine-test

[14] High creatinine levels can indicate chronic kidney disease. (s. f.). UCLA Health. https://www.uclahealth.org/news/high-creatinine-levels-can-indicate-chronic-kidney-disease

[15] Bagheri, B., Radmard, N., Faghani-Makrani, A., & Rasouli, M. (2019). Serum creatinine and occurrence and severity of coronary artery disease. Medicinski arhiv, 73(3), 154. https://doi.org/10.5455/medarh.2019.73.154-156

[16] Sodium blood test. (s. f.). Mount Sinai Health System. https://www.mountsinai.org/health-library/tests/sodium-blood-test

[17] Lewis, J. L., III. (2023, 12 noviembre). Hypernatremia (High level of sodium in the blood). MSD Manual Consumer Version. https://www.msdmanuals.com/en-gb/home/hormonal-and-metabolic-disorders/electrolyte-balance/hypernatremia-high-level-of-sodium-in-the-blood

[18] Low blood sodium in older adults: a concern? (2023, 16 mayo). Mayo Clinic. https://www.mayoclinic.org/diseases-conditions/hyponatremia/expert-answers/low-blood-sodium/faq-20058465

[19] Bots, S. H., Peters, S. A., & Woodward, M. (2017). Sex Differences in Coronary heart Disease and Stroke Mortality: A Global Assessment of the Effect of ageing between 1980 and 2010. BMJ Global Health, 2(2), e000298. https://doi.org/10.1136/bmjgh-2017-000298

[20] Bwhgive. (2023, 5 octubre). 1908AGEMNL – AG Newsletter – 7 Differences between Men and women. Brigham and Women's Hospital Giving. https://give.brighamandwomens.org/7-differences-between-men-and-women/

[21] SMOKING AND CARDIOVASCULAR DISEASE. (s/f). Cdc.gov, https://www.cdc.gov/tobacco/sgr/50th-anniversary/pdfs/fs_smoking_cvd_508.pdf

[22] How smoking affects the heart and blood vessels | NHLBI, NIH. (2022, 24 marzo). NHLBI, NIH. https://www.nhlbi.nih.gov/health/heart/smoking

[23] Smoking and cardiovascular disease. (2020, 20 julio). Johns Hopkins Medicine. https://www.hopkinsmedicine.org/health/conditions-and-diseases/smoking-and-cardiovascular-disease

6. Appendix

```
# Load necessary libraries
library(ggplot2)
library(car)
library(corrplot)
library(lme4)
library(broom)
library(effects)

# Read the data
heart_data <- read.csv("heart_failure_clinical_records_dataset.csv")
sapply(heart_data, class)

table(heart_data$DEATH_EVENT)

round(prop.table(table(heart_data$DEATH_EVENT))*100, digits = 2)

#################
### Death Event ###
#################
ggplot(heart_data, aes(x = as.factor(DEATH_EVENT), fill = as.factor(DEATH_EVENT))) +
  geom_bar() +
  xlab("Heart Disease") +
  ylab("Count") +
  ggtitle("Death events") +
  scale_fill_discrete(name = "Heart Disease", labels = c("Survive", "Death"))


##########
### Age ###
##########
ggplot(heart_data, aes(x = age, fill = as.factor(DEATH_EVENT))) +
  geom_bar() +
  xlab("Age") +
  ylab("Count") +
  ggtitle("Heart Disease vs Age") +
  scale_fill_discrete(name = "Heart Disease", labels = c("Survive", "Death"))


# Age model

age <- "age"

formula_age <- as.formula(paste("DEATH_EVENT ~", age))
```

```
age_model <- glm(formula_age, data = heart_data, family = "binomial")

# Check significance
age_summary <- summary(age_model)
age_summary
# Extract p-value
age_p_value <- age_summary$coefficients[, "Pr(>|z|)"][2]
age_p_value

# Double-check if age is a significant predictor by computing CI
age_coefficient <- coef(age_model)
age_coefficient
# Get confidence intervals for coefficients
age_conf_intervals <- confint(age_model)
age_conf_intervals

# Compute MLE parameters
likelihood_age <- function(parameters, data) {
  log_odds_age <- parameters[1] + parameters[2] * data$age
  probabilities <- 1 / (1 + exp(-log_odds_age))
  log_likelihood_age <- sum(data$DEATH_EVENT * log(probabilities) +
                            (1 - data$DEATH_EVENT) * log(1 - probabilities))
  return(-log_likelihood_age)  # We want to maximize, so we minimize the negative log
likelihood
}

initial_parameters <- c(0, 0)

result <- optim(par = initial_parameters, fn = likelihood_age, data = heart_data)

mle_age_parameters <- result$par
cat("MLE Parameters:", mle_age_parameters, "\n")

# Plot to visualize the significance of the predictor
age_values <- seq(min(heart_data$age), max(heart_data$age), length.out = 50)
log_odds <- age_coefficient * age_values
probability <- 1 / (1 + exp(-log_odds))

# Plotting
plot(heart_data$age, heart_data$DEATH_EVENT, pch = 16, col = "blue",
     xlab = "Age", ylab = "Death Event",
     main = "Logistic Regression Curve for Age")
lines(age_values, probability, col = "red", lwd = 2)

# Adding a horizontal line at y = 0.5 (decision boundary)
abline(h = 0.5, col = "green", lty = 2)
```

```
# Adding legend
legend("topright", legend = c("Data", "Logistic Curve", "Decision Boundary"),
       col = c("blue", "red", "green"), lty = c(NA, 1, 2), pch = c(16, NA, NA))

##############
### Anemia  ###
##############
ggplot(heart_data, aes(x = as.factor(anaemia), fill = as.factor(DEATH_EVENT))) +
  geom_bar() +
  xlab("Anemia levels") +
  ylab("Count") +
  ggtitle("Levels of Anemia and Heart Disease") +
  scale_fill_discrete(name = "Heart Disease", labels = c("Survive", "Death"))

# Anemia model
anemia <- "anaemia"

formula_anemia <- as.formula(paste("DEATH_EVENT ~", anemia))

anemia_model <- glm(formula_anemia, data = heart_data, family = "binomial")

# Check significance
anemia_summary <- summary(anemia_model)
anemia_summary

# Extract p-value
anemia_p_value <- anemia_summary$coefficients[, "Pr(>|z|)"][2]
anemia_p_value

# Double-check if anemia is a significant predictor by computing CI
anemia_coefficient <- coef(anemia_model)
anemia_coefficient
# Get confidence intervals for coefficients
anemia_conf_intervals <- confint(anemia_model)
anemia_conf_intervals

# Compute MLE parameters
likelihood_anemia <- function(parameters, data) {
  log_odds_anemia <- parameters[1] + parameters[2] * data$anaemia
  probabilities <- 1 / (1 + exp(-log_odds_anemia))
  log_likelihood_anemia <- sum(data$DEATH_EVENT * log(probabilities) +
                               (1 - data$DEATH_EVENT) * log(1 - probabilities))
  return(-log_likelihood_anemia)  # We want to maximize, so we minimize the negative
log likelihood
}
```

```
initial_parameters <- c(0, 0)

result <- optim(par = initial_parameters, fn = likelihood_anemia, data = heart_data)

mle_anemia_parameters <- result$par
cat("MLE Parameters:", mle_anemia_parameters, "\n")

# Plot to visualize the significance of the predictor
anemia_values <- seq(min(heart_data$anaemia), max(heart_data$anaemia),
length.out = 50)
log_odds <- anemia_coefficient * anemia_values

probability <- 1 / (1 + exp(-log_odds))

# Plotting
plot(heart_data$anaemia, heart_data$DEATH_EVENT, pch = 16,
        col = "blue", xlab = "Anemia", ylab = "Death Event",
        main = "Logistic Regression Curve for Anemia")
lines(anemia_values, probability, col = "red", lwd = 2)

# Adding a horizontal line at y = 0.5 (decision boundary)
abline(h = 0.5, col = "green", lty = 2)

# Adding legend
legend("topright", legend = c("Data", "Logistic Curve", "Decision Boundary"),
        col = c("blue", "red", "green"), lty = c(NA, 1, 2), pch = c(16, NA, NA))

############################
### Creatinine Phosphokinase ###
############################
ggplot(heart_data, aes(x =creatinine_phosphokinase, fill = as.factor(DEATH_EVENT))) +
  geom_histogram(position = "dodge", bins = 30, color = "black", alpha = 0.7) +
  xlab("Creatinine Phosphokinase") +
  ylab("Count") +
  ggtitle("Creatinine Phosphokinase vs. Death Event") +
  scale_fill_discrete(name = "Death Event", labels = c("Survive", "Death"))


# Creatinine Phosphokinase model
creatinine_phosphokinase <- "creatinine_phosphokinase"

formula_creatinine <- as.formula(paste("DEATH_EVENT ~", creatinine_phosphokinase))

creatinine_model <- glm(formula_creatinine, data = heart_data, family = "binomial")
```

```
# Check significance
creatinine_summary <- summary(creatinine_model)
creatinine_summary

# Extract p-value
creatinine_p_value <- creatinine_summary$coefficients[, "Pr(>|z|)"][2]
creatinine_p_value

# Double-check if creatinine phosphokinase is a significant predictor by computing CI
creatinine_coefficient <- coef(creatinine_model)
creatinine_coefficient
# Get confidence intervals for coefficients
creatinine_conf_intervals <- confint(creatinine_model)
creatinine_conf_intervals

# Compute MLE parameters
likelihood_cp <- function(parameters, data) {
  log_odds_cp <- parameters[1] + parameters[2] * data$creatinine_phosphokinase
  probabilities <- 1 / (1 + exp(-log_odds_cp))
  log_likelihood_cp <- sum(data$DEATH_EVENT * log(probabilities) +
                           (1 - data$DEATH_EVENT) * log(1 - probabilities))
  return(-log_likelihood_cp)  # We want to maximize, so we minimize the negative log
likelihood
}

initial_parameters <- c(0, 0)

result <- optim(par = initial_parameters, fn = likelihood_cp, data = heart_data)

mle_cp_parameters <- result$par
cat("MLE Parameters:", mle_cp_parameters, "\n")


# Plot to visualize the significance of the predictor
creatinine_values <- seq(min(heart_data$creatinine), max(heart_data$creatinine),
length.out = 50)
log_odds <- creatinine_coefficient * creatinine_values
probability <- 1 / (1 + exp(-log_odds))

# Plotting
plot(heart_data$creatinine, heart_data$DEATH_EVENT, pch = 16, col = "blue",
        xlab = "CPK", ylab = "Death Event", main = "Logistic Regression Curve for CPK")
lines(creatinine_values, probability, col = "red", lwd = 2)

# Adding a horizontal line at y = 0.5 (decision boundary)
abline(h = 0.5, col = "green", lty = 2)
```

```
# Adding legend
legend("topright", legend = c("Data", "Logistic Curve", "Decision Boundary"),
       col = c("blue", "red", "green"), lty = c(NA, 1, 2), pch = c(16, NA, NA))


###############
### Diabetes ###
###############
ggplot(heart_data, aes(x = as.factor(diabetes), fill = as.factor(DEATH_EVENT))) +
  geom_bar(position = "dodge", color = "black", alpha = 0.7) +
  xlab("Diabetes") +
  ylab("Count") +
  ggtitle("Diabetes vs. Death Event") +
  scale_fill_discrete(name = "Death Event", labels = c("Survive", "Death"))



diabetes <- "diabetes"

formula_diabetes <- as.formula(paste("DEATH_EVENT ~", diabetes))

# Diabetes model
diabetes_model <- glm(formula_diabetes, data = heart_data, family = "binomial")

# Check significance
diabetes_summary <- summary(diabetes_model)
diabetes_summary

# Extract p-value
diabetes_p_value <- diabetes_summary$coefficients[, "Pr(>|z|)"][2]
diabetes_p_value

# Double-check if diabetes is a significant predictor by computing CI
diabetes_coefficient <- coef(diabetes_model)
diabetes_coefficient
# Get confidence intervals for coefficients
diabetes_conf_intervals <- confint(diabetes_model)
diabetes_conf_intervals

# Compute MLE parameters
likelihood_diabetes <- function(parameters, data) {
  log_odds_diabetes <- parameters[1] + parameters[2] * data$diabetes
  probabilities <- 1 / (1 + exp(-log_odds_diabetes))
  log_likelihood_diabetes <- sum(data$DEATH_EVENT * log(probabilities) +
                                 (1 - data$DEATH_EVENT) * log(1 - probabilities))
```

```
  return(-log_likelihood_diabetes)  # We want to maximize, so we minimize the negative
log likelihood
}

initial_parameters <- c(0, 0)

result <- optim(par = initial_parameters, fn = likelihood_diabetes, data = heart_data)

mle_diabetes_parameters <- result$par
cat("MLE Parameters:", mle_diabetes_parameters, "\n")

# Plot to visualize the significance of the predictor
diabetes_values <- seq(min(heart_data$diabetes), max(heart_data$diabetes),
length.out = 50)
log_odds <- diabetes_coefficient * diabetes_values
probability <- 1 / (1 + exp(-log_odds))

# Plotting
plot(heart_data$diabetes, heart_data$DEATH_EVENT, pch = 16, col = "blue",
        xlab = "Diabetes", ylab = "Death Event", main = "Logistic Regression Curve for
Diabetes")
lines(diabetes_values, probability, col = "red", lwd = 2)

# Adding a horizontal line at y = 0.5 (decision boundary)
abline(h = 0.5, col = "green", lty = 2)

# Adding legend
legend("topright", legend = c("Data", "Logistic Curve", "Decision Boundary"),
        col = c("blue", "red", "green"), lty = c(NA, 1, 2), pch = c(16, NA, NA))

####################
### Ejection fraction ###
####################
ggplot(heart_data, aes(x = as.factor(DEATH_EVENT), y = ejection_fraction, fill =
DEATH_EVENT)) +
  geom_boxplot() +
  xlab("Death Event") +
  ylab("Ejection Fraction") +
  ggtitle("Ejection Fraction vs. Death Event")

# Ejection fraction model
ejection_fraction <- "ejection_fraction"

formula_ejection_fraction <- as.formula(paste("DEATH_EVENT ~", ejection_fraction))
```

```
ejection_fraction_model <- glm(formula_ejection_fraction, data = heart_data, family =
"binomial")

# Check significance
ejection_fraction_summary <- summary(ejection_fraction_model)
ejection_fraction_summary
# Extract p-value
ejection_fraction_p_value <- ejection_fraction_summary$coefficients[, "Pr(>|z|)"][2]
ejection_fraction_p_value

# Double-check if ejection fraction is a significant predictor by computing CI
ejection_fraction_coefficient <- coef(ejection_fraction_model)
ejection_fraction_coefficient
# Get confidence intervals for coefficients
ejection_fraction_conf_intervals <- confint(ejection_fraction_model)
ejection_fraction_conf_intervals

# Compute MLE parameters
likelihood_ej <- function(parameters, data) {
  log_odds_ej <- parameters[1] + parameters[2] * data$ejection_fraction
  probabilities <- 1 / (1 + exp(-log_odds_ej))
  log_likelihood_ej <- sum(data$DEATH_EVENT * log(probabilities) +
                           (1 - data$DEATH_EVENT) * log(1 - probabilities))
  return(-log_likelihood_ej)  # We want to maximize, so we minimize the negative log
likelihood
}

initial_parameters <- c(0, 0)

result <- optim(par = initial_parameters, fn = likelihood_ej, data = heart_data)

mle_ej_parameters <- result$par
cat("MLE Parameters:", mle_ej_parameters, "\n")

# Plot to visualize the significance of the predictor
ejection_fraction_values <- seq(min(heart_data$ejection_fraction),
max(heart_data$ejection_fraction), length.out = 50)
log_odds <- ejection_fraction_coefficient * ejection_fraction_values
probability <- 1 / (1 + exp(-log_odds))

# Plotting
plot(heart_data$ejection_fraction, heart_data$DEATH_EVENT, pch = 16,
        col = "blue", xlab = "ejection_fraction", ylab = "Death Event",
        main = "Logistic Regression Curve for Ejection fraction")
lines(ejection_fraction_values, probability, col = "red", lwd = 2)
```

```
# Adding a horizontal line at y = 0.5 (decision boundary)
abline(h = 0.5, col = "green", lty = 2)

# Adding legend
legend("topright", legend = c("Data", "Logistic Curve", "Decision Boundary"),
       col = c("blue", "red", "green"), lty = c(NA, 1, 2), pch = c(16, NA, NA))

######################
### High blood pressure ###
######################
ggplot(heart_data, aes(x = as.factor(high_blood_pressure), fill =
as.factor(DEATH_EVENT))) +
  geom_bar(position = "dodge", color = "black", alpha = 0.7) +
  xlab("High Blood Pressure") +
  ylab("Count") +
  ggtitle("High Blood Pressure vs. Death Event") +
  scale_fill_discrete(name = "Death Event", labels = c("Survive", "Death"))


# High blood pressure model
hbp <- "high_blood_pressure"

formula_hbp <- as.formula(paste("DEATH_EVENT ~", hbp))

hbp_model <- glm(formula_hbp, data = heart_data, family = "binomial")

# Check significance
hbp_summary <- summary(hbp_model)
hbp_summary
# Extract p-value
hbp_p_value <- hbp_summary$coefficients[, "Pr(>|z|)"][2]
hbp_p_value

# Double-check if high blood pressure is a significant predictor by computing CI
hbp_coefficient <- coef(hbp_model)
hbp_coefficient
# Get confidence intervals for coefficients
hbp_conf_intervals <- confint(hbp_model)
hbp_conf_intervals

# Compute MLE parameters
likelihood_hbp <- function(parameters, data) {
  log_odds_hbp <- parameters[1] + parameters[2] * data$high_blood_pressure
  probabilities <- 1 / (1 + exp(-log_odds_hbp))
  log_likelihood_hbp <- sum(data$DEATH_EVENT * log(probabilities) +
                            (1 - data$DEATH_EVENT) * log(1 - probabilities))
```

```
  return(-log_likelihood_hbp)  # We want to maximize, so we minimize the negative log
likelihood
}

initial_parameters <- c(0, 0)

result <- optim(par = initial_parameters, fn = likelihood_hbp, data = heart_data)

mle_hbp_parameters <- result$par
cat("MLE Parameters:", mle_hbp_parameters, "\n")

# Plot to visualize the significance of the predictor
hbp_values <- seq(min(heart_data$high_blood_pressure),
max(heart_data$high_blood_pressure), length.out = 50)
log_odds <- hbp_coefficient * hbp_values
probability <- 1 / (1 + exp(-log_odds))

# Plotting
plot(heart_data$high_blood_pressure, heart_data$DEATH_EVENT, pch = 16,
        col = "blue", xlab = "hbp", ylab = "Death Event",
        main = "Logistic Regression Curve for High Blood Pressure")
lines(hbp_values, probability, col = "red", lwd = 2)

# Adding a horizontal line at y = 0.5 (decision boundary)
abline(h = 0.5, col = "green", lty = 2)

# Adding legend
legend("topright", legend = c("Data", "Logistic Curve", "Decision Boundary"),
        col = c("blue", "red", "green"), lty = c(NA, 1, 2), pch = c(16, NA, NA))

##############
### Platelets ###
##############
ggplot(heart_data, aes(x = as.factor(DEATH_EVENT), y = platelets, fill =
DEATH_EVENT)) +
  geom_boxplot() +
  xlab("Death Event") +
  ylab("Platelets") +
  ggtitle("Platelets vs. Death Event")


# Platelets model
platelets <- "platelets"

formula_platelets <- as.formula(paste("DEATH_EVENT ~", platelets))
```

```
platelets_model <- glm(formula_platelets, data = heart_data, family = "binomial")

# Check significance
platelets_summary <- summary(platelets_model)
platelets_summary
# Extract p-value
platelets_p_value <- platelets_summary$coefficients[, "Pr(>|z|)"][2]
platelets_p_value

# Double-check if platelets is a significant predictor by computing CI
platelets_coefficient <- coef(platelets_model)
platelets_coefficient
# Get confidence intervals for coefficients
platelets_conf_intervals <- confint(platelets_model)
platelets_conf_intervals

# Compute MLE parameters
likelihood_platelets <- function(parameters, data) {
  log_odds_platelets <- parameters[1] + parameters[2] * data$platelets
  probabilities <- 1 / (1 + exp(-log_odds_platelets))
  log_likelihood_platelets <- sum(data$DEATH_EVENT * log(probabilities) +
                                  (1 - data$DEATH_EVENT) * log(1 - probabilities))
  return(-log_likelihood_platelets)  # We want to maximize, so we minimize the negative
log likelihood
}

initial_parameters <- c(0, 0)

result <- optim(par = initial_parameters, fn = likelihood_platelets, data = heart_data)

mle_platelets_parameters <- result$par
cat("MLE Parameters:", mle_platelets_parameters, "\n")

# Plot to visualize the significance of the predictor
platelets_values <- seq(min(heart_data$platelets), max(heart_data$platelets),
length.out = 50)
log_odds <- platelets_coefficient * platelets_values
probability <- 1 / (1 + exp(-log_odds))

# Plotting
plot(heart_data$platelets, heart_data$DEATH_EVENT, pch = 16, col = "blue",
          xlab = "Platelets", ylab = "Death Event",
          main = "Logistic Regression Curve for Platelets")
lines(platelets_values, probability, col = "red", lwd = 2)

# Adding a horizontal line at y = 0.5 (decision boundary)
```

```
abline(h = 0.5, col = "green", lty = 2)

# Adding legend
legend("topright", legend = c("Data", "Logistic Curve", "Decision Boundary"),
       col = c("blue", "red", "green"), lty = c(NA, 1, 2), pch = c(16, NA, NA))

####################
### Serum creatinine ###
####################
ggplot(heart_data, aes(x = as.factor(DEATH_EVENT), y = serum_creatinine, fill =
DEATH_EVENT)) +
  geom_boxplot() +
  xlab("Death Event") +
  ylab("Serum Creatinine") +
  ggtitle("Serum Creatinine vs. Death Event")


# Serum creatinine model
serum_creatinine <- "serum_creatinine"

formula_serum_creatinine <- as.formula(paste("DEATH_EVENT ~", serum_creatinine))

serum_creatinine_model <- glm(formula_serum_creatinine, data = heart_data, family =
"binomial")

# Check significance
serum_creatinine_summary <- summary(serum_creatinine_model)
serum_creatinine_summary
# Extract p-value
serum_creatinine_p_value <- serum_creatinine_summary$coefficients[, "Pr(>|z|)"][2]
serum_creatinine_p_value

# Double-check if serum creatinine is a significant predictor by computing CI
serum_creatinine_coefficient <- coef(serum_creatinine_model)
serum_creatinine_coefficient
# Get confidence intervals for coefficients
serum_creatinine_conf_intervals <- confint(serum_creatinine_model)
serum_creatinine_conf_intervals

# Compute MLE parameters
likelihood_sc <- function(parameters, data) {
  log_odds_sc <- parameters[1] + parameters[2] * data$serum_creatinine
  probabilities <- 1 / (1 + exp(-log_odds_sc))
  log_likelihood_sc <- sum(data$DEATH_EVENT * log(probabilities) +
                           (1 - data$DEATH_EVENT) * log(1 - probabilities))
```

```
  return(-log_likelihood_sc)  # We want to maximize, so we minimize the negative log
likelihood
}

initial_parameters <- c(0, 0)

result <- optim(par = initial_parameters, fn = likelihood_sc, data = heart_data)

mle_sc_parameters <- result$par
cat("MLE Parameters:", mle_sc_parameters, "\n")

# Plot to visualize the significance of the predictor
serum_creatinine_values <- seq(min(heart_data$serum_creatinine),
max(heart_data$serum_creatinine), length.out = 50)
log_odds <- serum_creatinine_coefficient * serum_creatinine_values
probability <- 1 / (1 + exp(-log_odds))

# Plotting
plot(heart_data$serum_creatinine, heart_data$DEATH_EVENT, pch = 16, col = "blue",
        xlab = "Serum Creatinine", ylab = "Death Event",
        main = "Logistic Regression Curve for Serum Creatinine")
lines(serum_creatinine_values, probability, col = "red", lwd = 2)

# Adding a horizontal line at y = 0.5 (decision boundary)
abline(h = 0.5, col = "green", lty = 2)

# Adding legend
legend("topright", legend = c("Data", "Logistic Curve", "Decision Boundary"),
        col = c("blue", "red", "green"), lty = c(NA, 1, 2), pch = c(16, NA, NA))

##################
### Serum sodium ###
##################
ggplot(heart_data, aes(x = as.factor(DEATH_EVENT), y = serum_sodium, fill =
DEATH_EVENT)) +
  geom_boxplot() +
  xlab("Death Event") +
  ylab("Serum sodium") +
  ggtitle("Serum Sodium vs. Death Event")


# Serum sodium model
serum_sodium <- "serum_sodium"

formula_serum_sodium <- as.formula(paste("DEATH_EVENT ~", serum_sodium))
```

```
serum_sodium_model <- glm(formula_serum_sodium, data = heart_data, family =
"binomial")

# Check significance
serum_sodium_summary <- summary(serum_sodium_model)
serum_sodium_summary
# Extract p-value
serum_sodium_p_value <- serum_sodium_summary$coefficients[, "Pr(>|z|)"][2]
serum_sodium_p_value

# Double-check if serum sodium is a significant predictor by computing CI
serum_sodium_coefficient <- coef(serum_sodium_model)
serum_sodium_coefficient
# Get confidence intervals for coefficients
serum_sodium_conf_intervals <- confint(serum_sodium_model)
serum_sodium_conf_intervals

# Compute MLE parameters
likelihood_ss <- function(parameters, data) {
  log_odds_ss <- parameters[1] + parameters[2] * data$serum_sodium
  probabilities <- 1 / (1 + exp(-log_odds_ss))
  log_likelihood_ss <- sum(data$DEATH_EVENT * log(probabilities) +
                           (1 - data$DEATH_EVENT) * log(1 - probabilities))
  return(-log_likelihood_ss)  # We want to maximize, so we minimize the negative log
likelihood
}

initial_parameters <- c(0, 0)

result <- optim(par = initial_parameters, fn = likelihood_ss, data = heart_data)

mle_ss_parameters <- result$par
cat("MLE Parameters:", mle_ss_parameters, "\n")

# Plot to visualize the significance of the predictor
serum_sodium_values <- seq(min(heart_data$serum_sodium),
max(heart_data$serum_sodium), length.out = 50)
log_odds <- serum_sodium_coefficient * serum_sodium_values
probability <- 1 / (1 + exp(-log_odds))

# Plotting
plot(heart_data$serum_sodium, heart_data$DEATH_EVENT, pch = 16, col = "blue",
        xlab = "Serum Sodium", ylab = "Death Event",
        main = "Logistic Regression Curve for Serum Sodium")
lines(serum_sodium_values, probability, col = "red", lwd = 2)
```

```
# Adding a horizontal line at y = 0.5 (decision boundary)
abline(h = 0.5, col = "green", lty = 2)

# Adding legend
legend("topright", legend = c("Data", "Logistic Curve", "Decision Boundary"),
       col = c("blue", "red", "green"), lty = c(NA, 1, 2), pch = c(16, NA, NA))

##########
### Sex ###
##########
ggplot(heart_data, aes(x = sex, fill = as.factor(DEATH_EVENT))) +
  geom_bar() +
  xlab("Sex") +
  ylab("Count") +
  ggtitle("Heart Disease Female vs Male") +
  scale_fill_discrete(name = "Heart Disease", labels = c("Survive", "Death"))


# Sex model
sex <- "sex"

formula_sex <- as.formula(paste("DEATH_EVENT ~", sex))

sex_model <- glm(formula_sex, data = heart_data, family = "binomial")

# Check significance
sex_summary <- summary(sex_model)
sex_summary
# Extract p-value
sex_p_value <- sex_summary$coefficients[, "Pr(>|z|)"][2]
sex_p_value

# Double-check if sex is a significant predictor by computing CI
sex_coefficient <- coef(sex_model)
sex_coefficient
# Get confidence intervals for coefficients
sex_conf_intervals <- confint(sex_model)
sex_conf_intervals

# Compute MLE parameters
likelihood_sex <- function(parameters, data) {
  log_odds_sex <- parameters[1] + parameters[2] * data$sex
  probabilities <- 1 / (1 + exp(-log_odds_sex))
  log_likelihood_sex <- sum(data$DEATH_EVENT * log(probabilities) +
                            (1 - data$DEATH_EVENT) * log(1 - probabilities))
```

```
  return(-log_likelihood_sex)  # We want to maximize, so we minimize the negative log likelihood
}

initial_parameters <- c(0, 0)

result <- optim(par = initial_parameters, fn = likelihood_sex, data = heart_data)

mle_sex_parameters <- result$par
cat("MLE Parameters:", mle_sex_parameters, "\n")

# Plot to visualize the significance of the predictor
sex_values <- seq(min(heart_data$sex), max(heart_data$sex), length.out = 50)
log_odds <- sex_coefficient * sex_values
probability <- 1 / (1 + exp(-log_odds))

# Plotting
plot(heart_data$sex, heart_data$DEATH_EVENT, pch = 16, col = "blue",
        xlab = "Sex", ylab = "Death Event", main = "Logistic Regression Curve for Sex")
lines(sex_values, probability, col = "red", lwd = 2)

# Adding a horizontal line at y = 0.5 (decision boundary)
abline(h = 0.5, col = "green", lty = 2)

# Adding legend
legend("topright", legend = c("Data", "Logistic Curve", "Decision Boundary"),
        col = c("blue", "red", "green"), lty = c(NA, 1, 2), pch = c(16, NA, NA))

##############
### Smoking ###
##############
ggplot(heart_data, aes(x = smoking, fill = as.factor(DEATH_EVENT))) +
  geom_bar(position = "dodge", color = "black", alpha = 0.7) +
  xlab("Smoking") +
  ylab("Count") +
  ggtitle("Smoking vs. Death Event") +
  scale_fill_discrete(name = "Death Event", labels = c("Survive", "Death"))


# Smoking model
smoking <- "smoking"

formula_smoking <- as.formula(paste("DEATH_EVENT ~", smoking))

smoking_model <- glm(formula_smoking, data = heart_data, family = "binomial")
```

```
# Check significance
smoking_summary <- summary(smoking_model)
smoking_summary
# Extract p-value
smoking_p_value <- smoking_summary$coefficients[, "Pr(>|z|)"][2]
smoking_p_value

# Double-check if smoking is a significant predictor by computing CI
smoking_coefficient <- coef(smoking_model)
smoking_coefficient
# Get confidence intervals for coefficients
smoking_conf_intervals <- confint(smoking_model)
smoking_conf_intervals

# Compute MLE parameters
likelihood_smoking <- function(parameters, data) {
  log_odds_smoking <- parameters[1] + parameters[2] * data$smoking
  probabilities <- 1 / (1 + exp(-log_odds_smoking))
  log_likelihood_smoking <- sum(data$DEATH_EVENT * log(probabilities) +
                                (1 - data$DEATH_EVENT) * log(1 - probabilities))
  return(-log_likelihood_smoking)  # We want to maximize, so we minimize the negative
log likelihood
}

initial_parameters <- c(0, 0)

result <- optim(par = initial_parameters, fn = likelihood_smoking, data = heart_data)

mle_smoking_parameters <- result$par
cat("MLE Parameters:", mle_smoking_parameters, "\n")

# Plot to visualize the significance of the predictor
smoking_values <- seq(min(heart_data$smoking), max(heart_data$smoking),
length.out = 50)
log_odds <- smoking_coefficient * smoking_values
probability <- 1 / (1 + exp(-log_odds))

# Plotting
plot(heart_data$smoking, heart_data$DEATH_EVENT, pch = 16, col = "blue",
          xlab = "Smoking", ylab = "Death Event", main = "Logistic Regression Curve for
Smoking")
lines(smoking_values, probability, col = "red", lwd = 2)

# Adding a horizontal line at y = 0.5 (decision boundary)
abline(h = 0.5, col = "green", lty = 2)
```

```
# Adding legend
legend("topright", legend = c("Data", "Logistic Curve", "Decision Boundary"),
       col = c("blue", "red", "green"), lty = c(NA, 1, 2), pch = c(16, NA, NA))

############################
### Classification of predictors ###
############################
alpha <- 0.05

predictors <- c(age, anemia, creatinine_phosphokinase, diabetes,
       ejection_fraction, hbp, platelets, serum_creatinine,
       serum_sodium, sex, smoking)

p_values <- c(age_p_value, anemia_p_value, creatinine_p_value, diabetes_p_value,
       ejection_fraction_p_value, hbp_p_value, platelets_p_value,
       serum_creatinine_p_value, serum_sodium_p_value, sex_p_value,
       smoking_p_value)


if (any(p_values < alpha)) {
  significant_predictors <- predictors[p_values < alpha]

  cat("Significant predictors:", paste(significant_predictors, collapse = ", "), "\n")
  cat("Corresponding p-values:", paste(p_values[p_values < alpha], collapse = ", "),
"\n\n")

}

if (any(p_values > alpha)){
       non_significant_predictors <- predictors[p_values > alpha]

       cat("Non-significant predictors:", paste(non_significant_predictors, collapse = ",
"), "\n")
       cat("Corresponding p-values:", paste(p_values[p_values > alpha], collapse = ", "),
"\n")
}


###############
### All model ###
###############
all_model_formula <- as.formula(paste("DEATH_EVENT ~", paste(predictors, collapse =
" + ")))

all_model <- glm(all_model_formula, data = heart_data, family = "binomial")
```

```
# Check significance
all_summary <- summary(all_model)
all_summary

# Check if there are significant predictors by computing CI
coefficients <- coef(all_model)

# Get confidence intervals for coefficients
conf_intervals <- confint(all_model)

# Display the results
results <- data.frame(coefficients, conf_intervals)
print(results)

# Compute MLE parameters
likelihood <- function(parameters, data) {
  log_odds <- parameters[1] + sum(parameters[2:length(parameters)] * data[,
2:ncol(data)])
  probabilities <- 1 / (1 + exp(-log_odds))
  log_likelihood <- sum(data$DEATH_EVENT * log(probabilities) +
                        (1 - data$DEATH_EVENT) * log(1 - probabilities))
  return(-log_likelihood)  # We want to maximize, so we minimize the negative log
likelihood
}

initial_parameters <- rep(0, ncol(heart_data)-2)

result <- optim(par = initial_parameters, fn = likelihood, data = heart_data)
mle_parameters <- result$par
cat("MLE Parameters:", mle_parameters, "\n")


# Create effects plot
effect_all <- allEffects(all_model)
plot(effect_all, col = "red", lines = TRUE, rug = FALSE)

######################
### Significant model ###
######################

significant_model_formula <- as.formula(paste("DEATH_EVENT ~",
paste(significant_predictors, collapse = " + ")))

significant_model <- glm(significant_model_formula, data = heart_data, family =
"binomial")
```

```
# Check significance
significant_summary <- summary(significant_model)
significant_summary

# Double-check if the significant predictors are significant by computing CI
significant_coefficient <- coef(significant_model)
significant_coefficient
# Get confidence intervals for coefficients
significant_conf_intervals <- confint(significant_model)
significant_conf_intervals


# Create effects plot
effect_sign <- allEffects(significant_model)
plot(effect_sign, col = "red", lines = TRUE, rug = FALSE)

# Correlation matrix between predictors to check for multicollinearity
corr_matrix <- cor(heart_data[predictors])
corr_matrix
corrplot(corr_matrix, method = "color")


# Alternative way to check for multicollinearity
vif_values <- car::vif(all_model)
vif_values

# NO values between 5-10, no collinearity


# Create a model with all predictors except smoking since it has the highest p-value
not_smoking_model <- glm(formula = DEATH_EVENT ~ age + anaemia +
creatinine_phosphokinase
                + diabetes + ejection_fraction + high_blood_pressure + platelets +
serum_creatinine
                + serum_sodium + sex, data = heart_data, family = "binomial")

# Check significance
not_smoking_summary <- summary(not_smoking_model)
not_smoking_summary

# Little increase in deviance, anova test not to lose info
not_smoking_anova <- anova(all_model, not_smoking_model, test = "Chisq")
not_smoking_anova


# We have not lost info, p-value non-significant
```

```
# Create a model with all predictors except platelets since it has the highest p-value
not_platelets_model <- glm(formula = DEATH_EVENT ~ age + anaemia +
creatinine_phosphokinase
                + diabetes + ejection_fraction + high_blood_pressure + serum_creatinine
                + serum_sodium + sex, data = heart_data, family = "binomial")

# Check significance
not_platelets_summary <- summary(not_platelets_model)
not_platelets_summary

# Little increase in deviance, anova test not to lose info
not_platelets_anova <- anova(all_model, not_platelets_model, test = "Chisq")
not_platelets_anova


# We have not lost info, p-value non-significant
# Create a model with all predictors except diabetes since it has the highest p-value
not_diabetes_model <- glm(formula = DEATH_EVENT ~ age + anaemia +
creatinine_phosphokinase
                + ejection_fraction + high_blood_pressure + serum_creatinine
                + serum_sodium + sex, data = heart_data, family = "binomial")

# Check significance
not_diabetes_summary <- summary(not_diabetes_model)
not_diabetes_summary

# Little increase in deviance, anova test not to lose info
not_diabetes_anova <- anova(all_model, not_diabetes_model, test = "Chisq")
not_diabetes_anova



# We have not lost info, p-value non-significant
# Create a model with all predictors except sex since it has the highest p-value
not_sex_model <- glm(formula = DEATH_EVENT ~ age + anaemia +
creatinine_phosphokinase
                + ejection_fraction + high_blood_pressure + serum_creatinine
                + serum_sodium, data = heart_data, family = "binomial")

# Check significance
not_sex_summary <- summary(not_sex_model)
not_sex_summary

# Little increase in deviance, anova test not to lose info
not_sex_anova <- anova(all_model, not_sex_model, test = "Chisq")
not_sex_anova
```

```
# We have not lost info, p-value non-significant
# Create a model with all predictors except anemia since it has the highest p-value
not_anemia_model <- glm(formula = DEATH_EVENT ~ age + creatinine_phosphokinase
                        + ejection_fraction + high_blood_pressure + serum_creatinine
                        + serum_sodium, data = heart_data, family = "binomial")

# Check significance
not_anemia_summary <- summary(not_anemia_model)
not_anemia_summary

# Little increase in deviance, anova test not to lose info
not_anemia_anova <- anova(all_model, not_anemia_model, test = "Chisq")
not_anemia_anova



# We have not lost info, p-value non-significant
# Create a model with all predictors except serum_sodium since it has the highest
p-value
not_serum_sodium_model <- glm(formula = DEATH_EVENT ~ age +
creatinine_phosphokinase
                        + ejection_fraction + high_blood_pressure + serum_creatinine,
                        data = heart_data, family = "binomial")

# Check significance
not_serum_sodium_summary <- summary(not_serum_sodium_model)
not_serum_sodium_summary

# Little increase in deviance, anova test not to lose info
not_serum_sodium_anova <- anova(all_model, not_serum_sodium_model, test =
"Chisq")
not_serum_sodium_anova



# We have not lost info, p-value non-significant
# Create a model with all predictors except creatinine since it has the highest p-value
not_creatinine_model <- glm(formula = DEATH_EVENT ~ age + ejection_fraction
                        + high_blood_pressure + serum_creatinine,
                        data = heart_data, family = "binomial")
# Check significance
not_creatinine_summary <- summary(not_creatinine_model)
not_creatinine_summary
```

```
# Little increase in deviance, anova test not to lose info
not_creatinine_anova <- anova(all_model, not_creatinine_model, test = "Chisq")
not_creatinine_anova



# We have not lost info, p-value non-significant
# Create a model with all predictors except hbp since it has the highest p-value
not_hbp_model <- glm(formula = DEATH_EVENT ~ age + ejection_fraction +
serum_creatinine,
                     data = heart_data, family = "binomial")
# Check significance
not_hbp_summary <- summary(not_hbp_model)
not_hbp_summary

# Little increase in deviance, anova test not to lose info
not_hbp_anova <- anova(all_model, not_hbp_model, test = "Chisq")
not_hbp_anova

# Compute probabilities for the final model
final_model <- not_hbp_model
summary(final_model)

heart_data$prob_pred <- predict(final_model, newdata = heart_data, type =
"response")

heart_data$prediction <- cut(heart_data$prob_pred, breaks = c(0, 0.5, 1), labels = c(0,
1))
attach(heart_data)
table(heart_data$DEATH_EVENT, heart_data$prediction)



# Compare probabilities with the all model
heart_data$prob_pred <- predict(all_model, newdata = heart_data, type = "response")

heart_data$prediction <- cut(heart_data$prob_pred, breaks = c(0, 0.5, 1), labels = c(0,
1))
attach(heart_data)
table(heart_data$DEATH_EVENT, heart_data$prediction)

# Exactly the same numbers, the final model is reliable with 77% accuracy
```