Heart Failure Prediction

Amisha Chandrakant Patel

Data Science, [redacted] University

[redacted]: Data Mining – Final Project

[redacted]

February 21, 2024

Table of Contents

1. Problem Description and Objectives

Cardiovascular diseases (CVDs) are the number 1 cause of death globally, taking an estimated 17.9 million lives each year, which accounts for 31% of all deaths worldwide. Four out of 5CVD deaths are due to heart attacks and strokes, and one-third of these deaths occur prematurely in people under 70 years of age. Heart failure is a common event caused by CVDs and this dataset contains 12 features that can be used to predict a possible heart disease.

Heart failure occurs when the heart's ability to pump blood diminishes, leading to inadequate circulation. Common causes include coronary artery disease, hypertension, and previous heart attacks. Globally, heart failure affects millions, with an estimated 26 million people diagnosed. Its prevalence is rising due to aging populations and lifestyle factors. Timely intervention, lifestyle changes, and medical management are crucial for mitigating its impact.

People with cardiovascular disease or who are at high cardiovascular risk (due to the presence of one or more risk factors such as hypertension, diabetes, hyperlipidemia or already established disease) need early detection and management wherein a machine learning model can be of great help.

Machine learning applied to medical records, in particular, can be an effective tool both to predict the survival of each patient having heart failure symptoms, and to detect the most important clinical features (or risk factors) that may lead to heart failure. Scientists can take advantage of machine learning not only for clinical prediction, but also for feature ranking.

2. Data Description

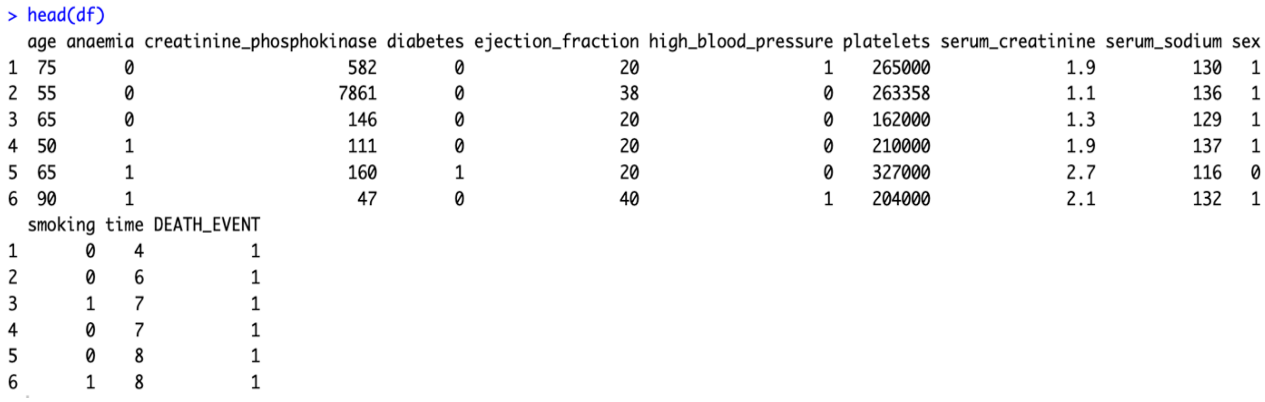
Data Source: We analyzed a dataset containing the medical records of 299 heart failure patients collected at the Faisalabad Institute of Cardiology and at the Allied Hospital in Faisalabad (Punjab, Pakistan), during April–December 2015. The patients consisted of 105 women and 194 men, and their ages range between 40 and 95 years old.

Data Contents: The dataset contains 13 features, which report clinical, body, and lifestyle information, that we briefly describe here:

- DEATH_EVENT: target label [Boolean]
- Age: age of the patient [number]
- Anaemia: Decrease of red blood cells or hemoglobin [categorical - M = male; F = female]
- Creatinine Phosphokinase: level of the CPK enzyme in the blood (mcg/L) [number]
- Diabetes: the patient has diabetes or not [Boolean]
- Ejection Fraction: percentage of blood leaving the heart at each contraction (percentage) [number]
- High Blood Pressure: the patient has hypertension or not [Boolean]
- Platelets: platelets in the blood (kilo platelets/mL) [number]
- Serum Creatinine: Level of serum creatinine in the blood (mg/dL) [number]
- Serum Sodium: Level of serum sodium in the blood (mEq/L) [number]
- Sex: gender [categorical]
- Smoking: the patient smokes or not [Boolean]
- Time: Follow-up period (days) [number]

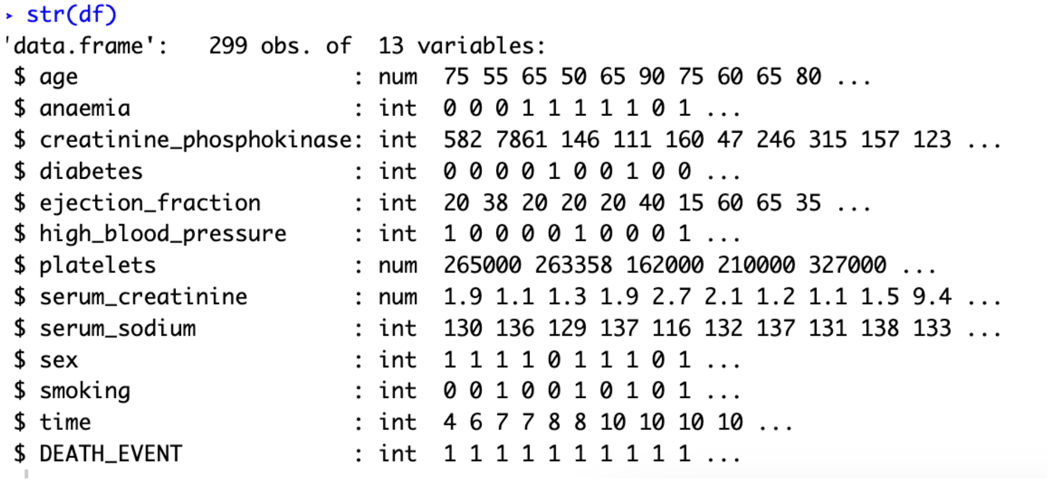
3. Data Processing

The initial data exploration involved examining the first few records of the dataset using head(data) function.



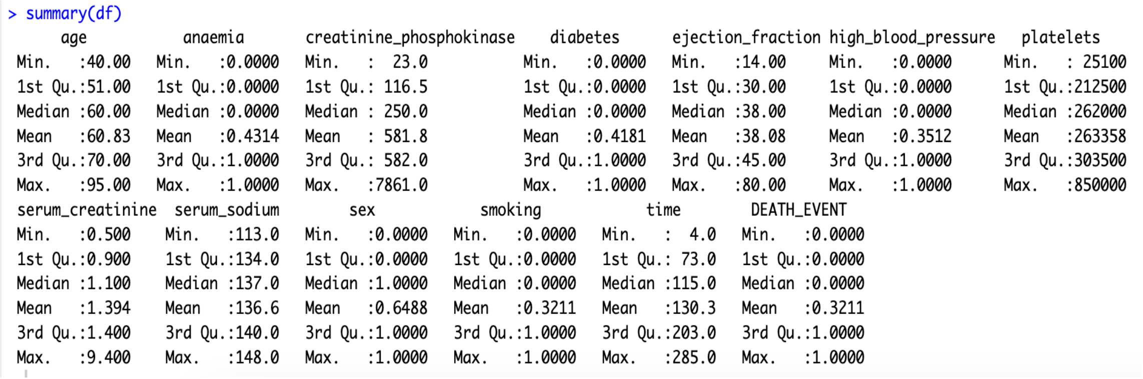
```
> head(df)
  age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure platelets serum_creatinine serum_sodium sex
1  75       0                      582        0                20                   1    265000              1.9          130   1
2  55       0                     7861        0                38                   0    263358              1.1          136   1
3  65       0                      146        0                20                   0    162000              1.3          129   1
4  50       1                      111        0                20                   0    210000              1.9          137   1
5  65       1                      160        1                20                   0    327000              2.7          116   0
6  90       1                       47        0                40                   1    204000              2.1          132   1
  smoking time DEATH_EVENT
1       0    4           1
2       0    6           1
3       1    7           1
4       0    7           1
5       0    8           1
6       1    8           1
```

To understand the structure of the dataset, we used str(data). This provided insights into the data types of each column and potential discrepancies.



```
> str(df)
'data.frame':	299 obs. of  13 variables:
 $ age                     : num  75 55 65 50 65 90 75 60 65 80 ...
 $ anaemia                 : int  0 0 0 1 1 1 1 1 0 1 ...
 $ creatinine_phosphokinase: int  582 7861 146 111 160 47 246 315 157 123 ...
 $ diabetes                : int  0 0 0 0 1 0 0 1 0 0 ...
 $ ejection_fraction       : int  20 38 20 20 20 40 15 60 65 35 ...
 $ high_blood_pressure     : int  1 0 0 0 0 1 0 0 0 1 ...
 $ platelets               : num  265000 263358 162000 210000 327000 ...
 $ serum_creatinine        : num  1.9 1.1 1.3 1.9 2.7 2.1 1.2 1.1 1.5 9.4 ...
 $ serum_sodium            : int  130 136 129 137 116 132 137 131 138 133 ...
 $ sex                     : int  1 1 1 1 0 1 1 1 0 1 ...
 $ smoking                 : int  0 0 1 0 0 1 0 1 0 1 ...
 $ time                    : int  4 6 7 7 8 8 10 10 10 10 ...
 $ DEATH_EVENT             : int  1 1 1 1 1 1 1 1 1 1 ...
```

The summary(data) function was employed to obtain summary statistics for each column, aiding in identifying key features and potential areas for data cleaning.



```
> summary(df)
      age           anaemia       creatinine_phosphokinase    diabetes      ejection_fraction high_blood_pressure    platelets     
 Min.   :40.00   Min.   :0.0000   Min.   :  23.0           Min.   :0.0000   Min.   :14.00     Min.   :0.0000      Min.   : 25100  
 1st Qu.:51.00   1st Qu.:0.0000   1st Qu.: 116.5           1st Qu.:0.0000   1st Qu.:30.00     1st Qu.:0.0000      1st Qu.:212500  
 Median :60.00   Median :0.0000   Median : 250.0           Median :0.0000   Median :38.00     Median :0.0000      Median :262000  
 Mean   :60.83   Mean   :0.4314   Mean   : 581.8           Mean   :0.4181   Mean   :38.08     Mean   :0.3512      Mean   :263358  
 3rd Qu.:70.00   3rd Qu.:1.0000   3rd Qu.: 582.0           3rd Qu.:1.0000   3rd Qu.:45.00     3rd Qu.:1.0000      3rd Qu.:303500  
 Max.   :95.00   Max.   :1.0000   Max.   :7861.0           Max.   :1.0000   Max.   :80.00     Max.   :1.0000      Max.   :850000  
 serum_creatinine  serum_sodium        sex            smoking            time         DEATH_EVENT    
 Min.   :0.500    Min.   :113.0   Min.   :0.0000   Min.   :0.0000   Min.   :  4.0   Min.   :0.0000  
 1st Qu.:0.900    1st Qu.:134.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 73.0   1st Qu.:0.0000  
 Median :1.100    Median :137.0   Median :1.0000   Median :0.0000   Median :115.0   Median :0.0000  
 Mean   :1.394    Mean   :136.6   Mean   :0.6488   Mean   :0.3211   Mean   :130.3   Mean   :0.3211  
 3rd Qu.:1.400    3rd Qu.:140.0   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:203.0   3rd Qu.:1.0000  
 Max.   :9.400    Max.   :148.0   Max.   :1.0000   Max.   :1.0000   Max.   :285.0   Max.   :1.0000  
```

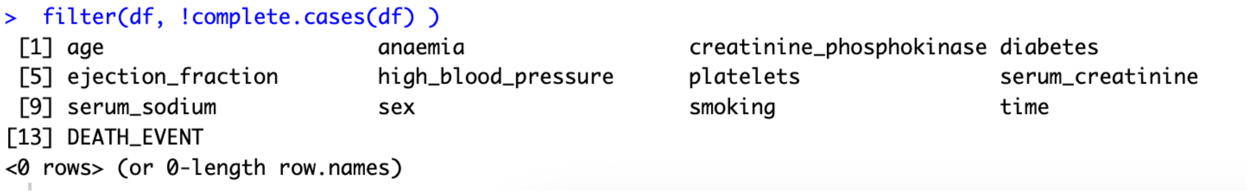
We checked for duplicate rows using data[duplicated(data),] and found no duplicate entries. The dataset for analysis originally contained binary features with integer data types representing the presence/absence of certain characteristics (encoded as 0/1 values). To enable more appropriate analysis, these binary integer variables were converted to factor data types using the mutate_at function from the dplyr R package.

The structured data processing provides a clean, analysis-ready dataset. Issues that could lead to incorrect insights were tackled upfront through rigorous validation checks.

4. Unknown Values

A thorough check for missing data was conducted on the dataset. The functions colSums(is.na(data)) and complete.cases() verified that there were no NULL or NA values present across any of the columns.

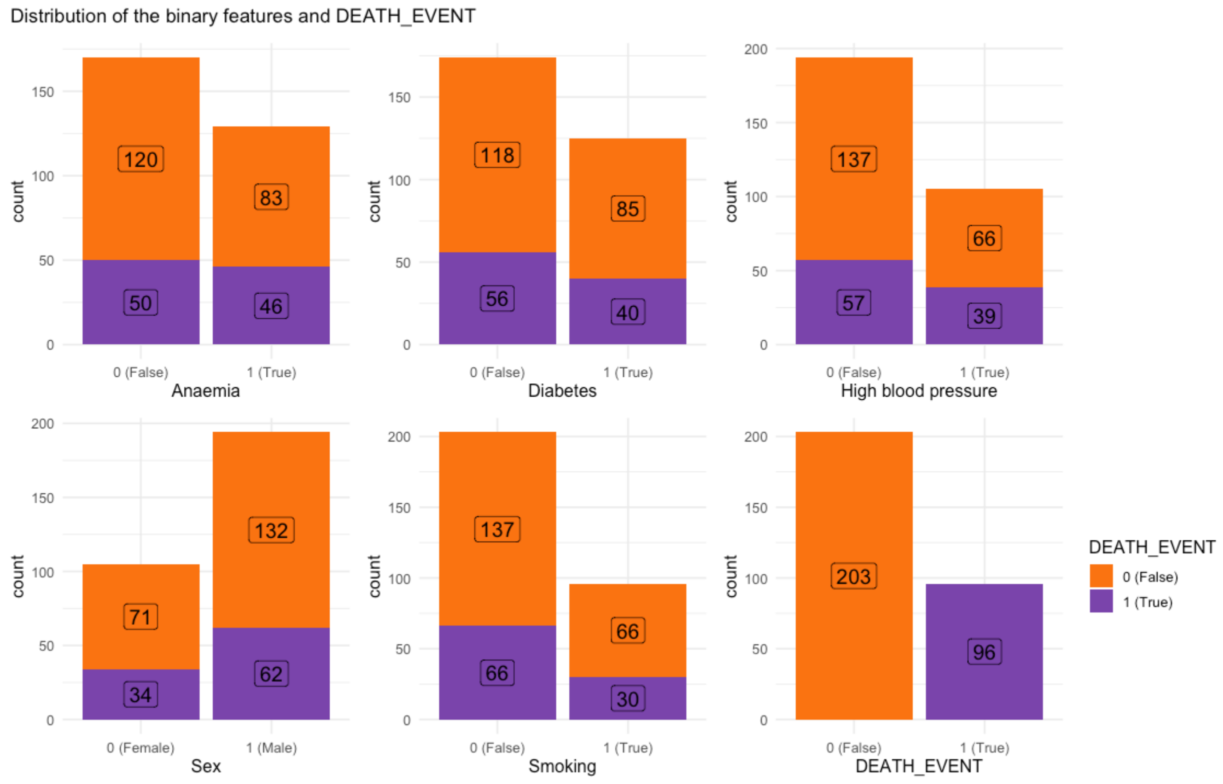
Despite not finding any missing values with these checks, as a precautionary measure na.omit() was still applied to the data before modeling. This removes any rows containing NAs, should there be any arising during data pre-processing stages.



```
>  filter(df, !complete.cases(df) )
 [1] age                      anaemia                  creatinine_phosphokinase diabetes                
 [5] ejection_fraction        high_blood_pressure      platelets                serum_creatinine        
 [9] serum_sodium             sex                      smoking                  time                    
[13] DEATH_EVENT             
<0 rows> (or 0-length row.names)
```

5. Data Visualization

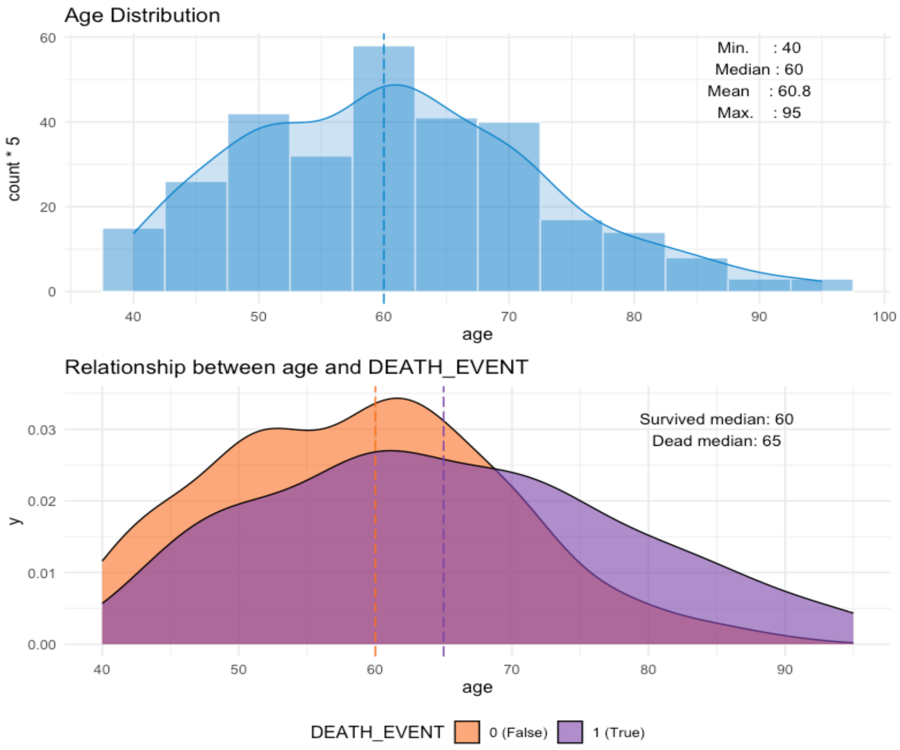
5.1. Distribution of the binary feature



Key Observations:

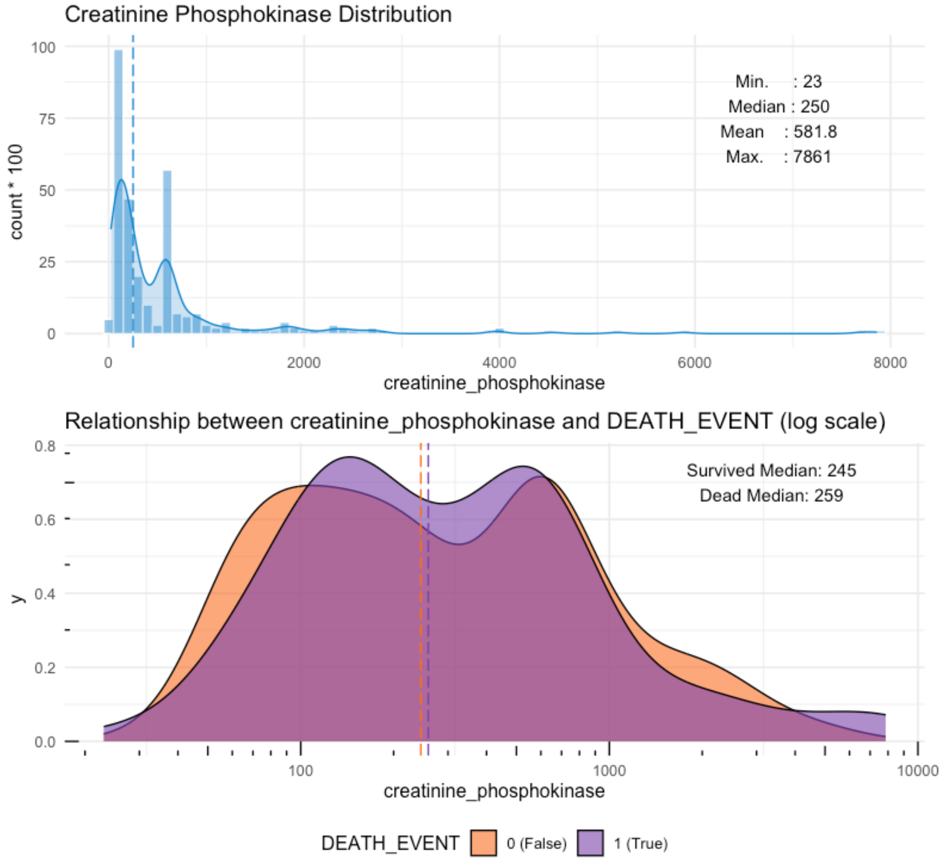
- Anaemia when present (true) showed comparable distributions, suggesting minor correlation with mortality.
- Diabetes and High BP when present (true) displayed noticeably higher death proportions, indicating strong associations with mortality likelihood.
- For Sex, the male category showed substantially higher death proportions versus females, indicating higher mortality susceptibility.
- Smoking displayed an unexpected reverse trend from anticipated correlations, with higher survival proportions among smokers.

5.2. Age Distribution and Relationship with Death Event



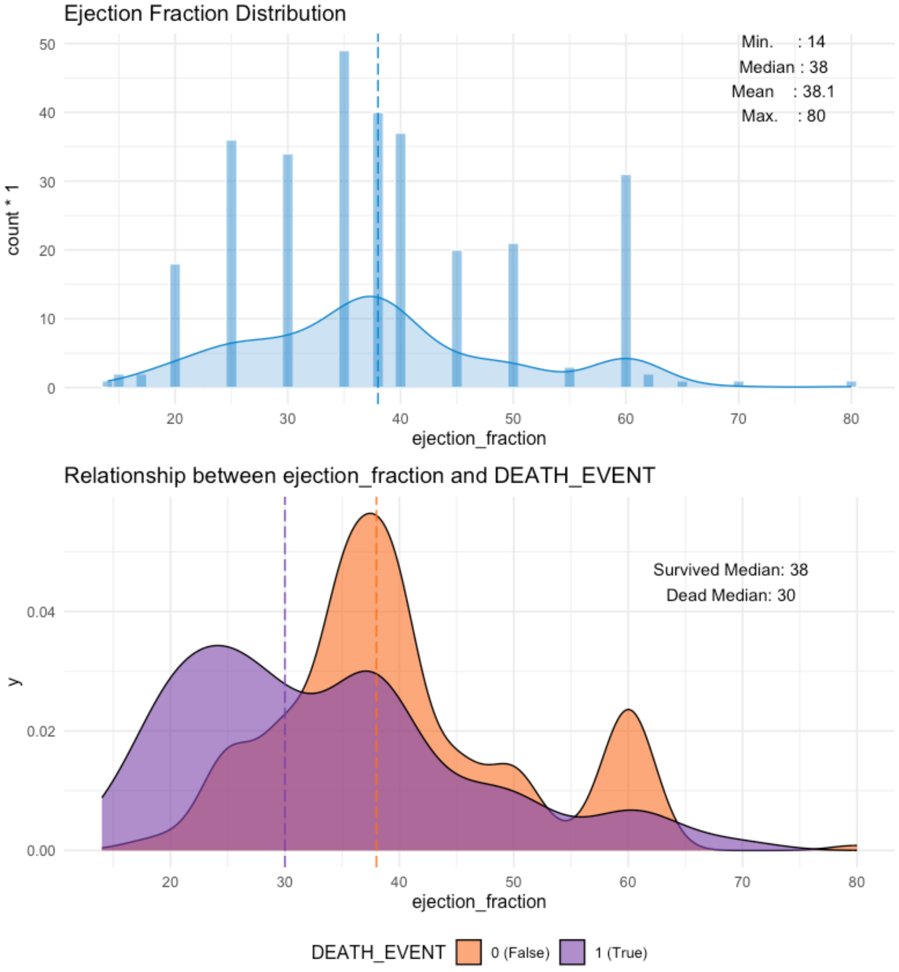
- The age of patients was highest around 60 years old, and the number of patients decreased in a bell-shaped pattern around that age.
- There is a difference in the distribution of each objective variable, with the younger the age, the more difficult it is to die; the probability density reverses after the age of just under 70.

5.3. Creatinine Phosphokinase Distribution and Relationship with Death Event



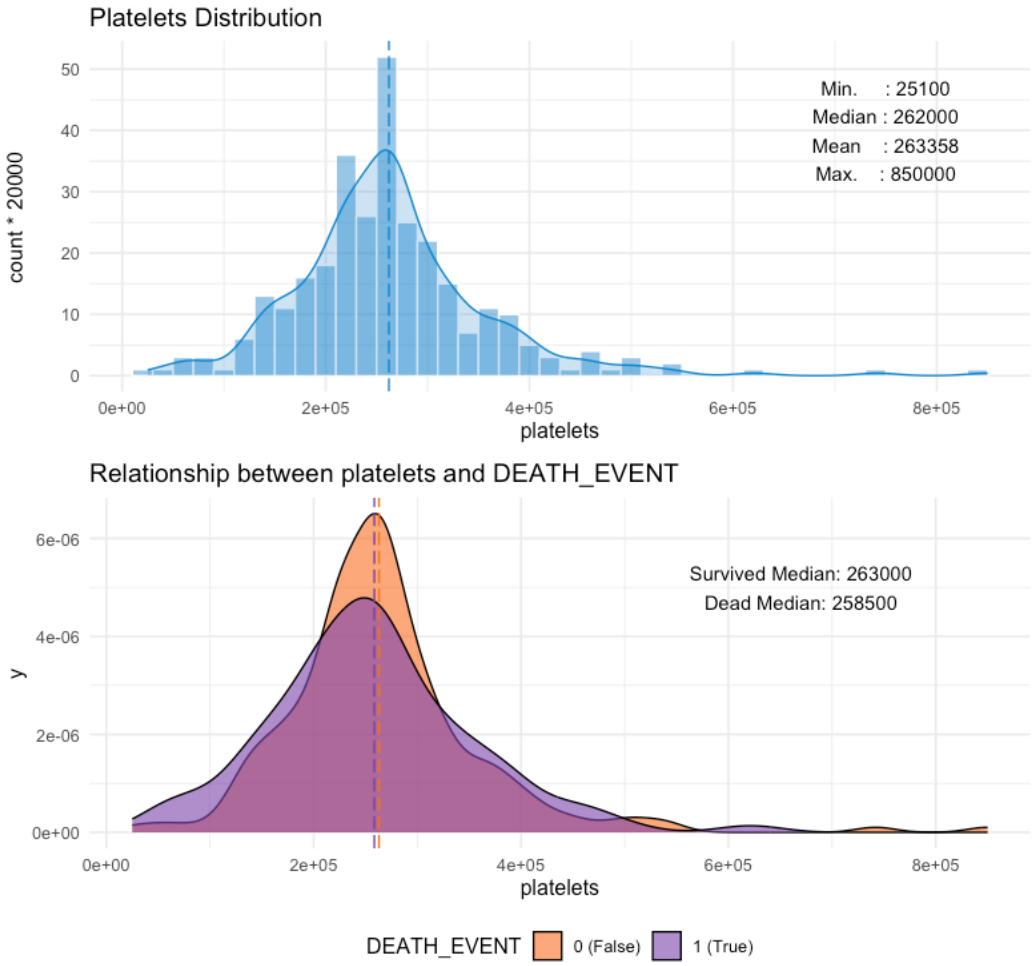
- The distribution is heavily skewed to one side, with the highest value more than 30 times the median.
- By objective variable, there is little difference in the median, although there are some differences in the distribution.

5.4. Ejection Fraction Distribution and Relationship with Death Event



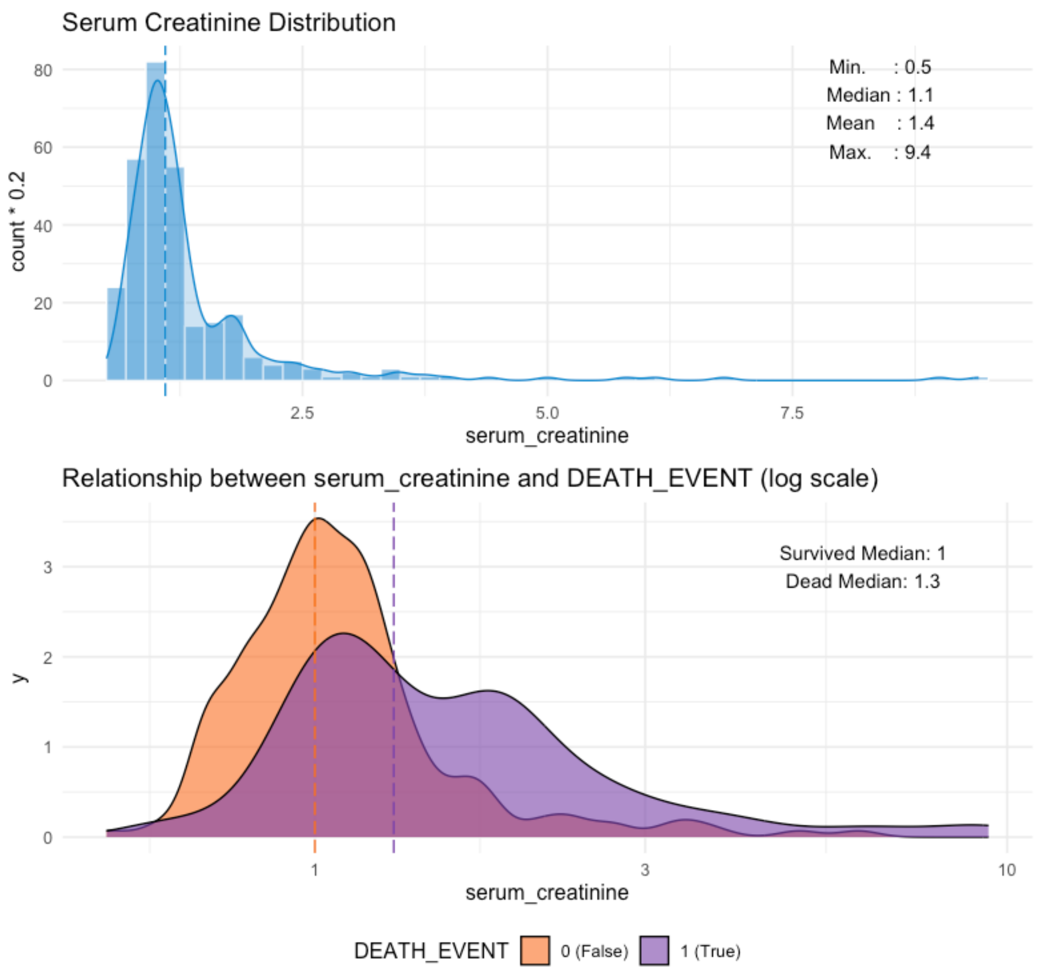
- The distribution is discrete, not continuous, with the first peak near 38 and the second peak near 60.
- By objective variable, there are considerable differences in the shape of the distribution and in the median. Survivors are mostly located near the first and second mountains. The values of the dead are mostly around 30 and decrease slowly from there.

5.5. Platelets Distribution and Relationship with Death Event



- The distribution is roughly symmetrical and almost bell-shaped.
- By objective variable, there is little difference in the median. Survivors have slightly higher platelet counts, and the values are clustered around the median.

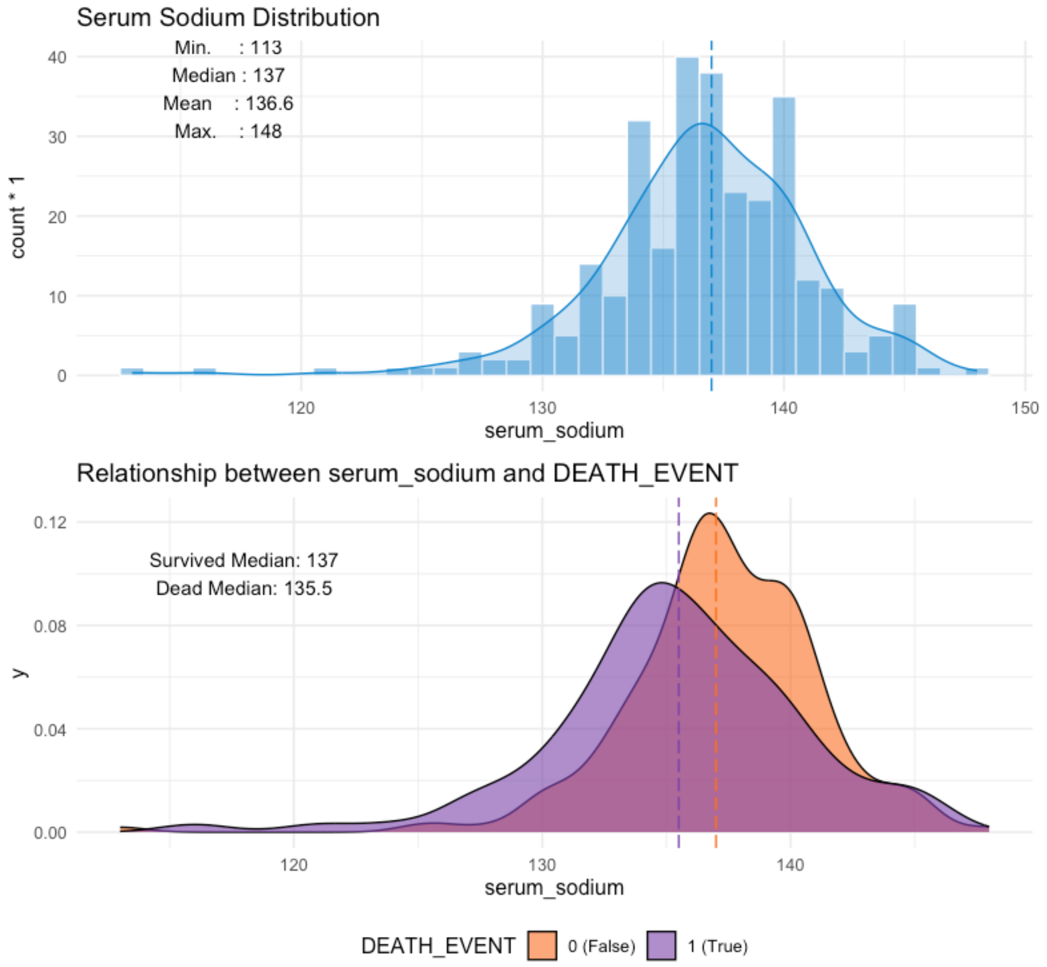
5.6. Serum Creatinine Distribution and Relationship with Death Event


Serum Creatinine Distribution

Relationship between serum_creatinine and DEATH_EVENT (log scale)

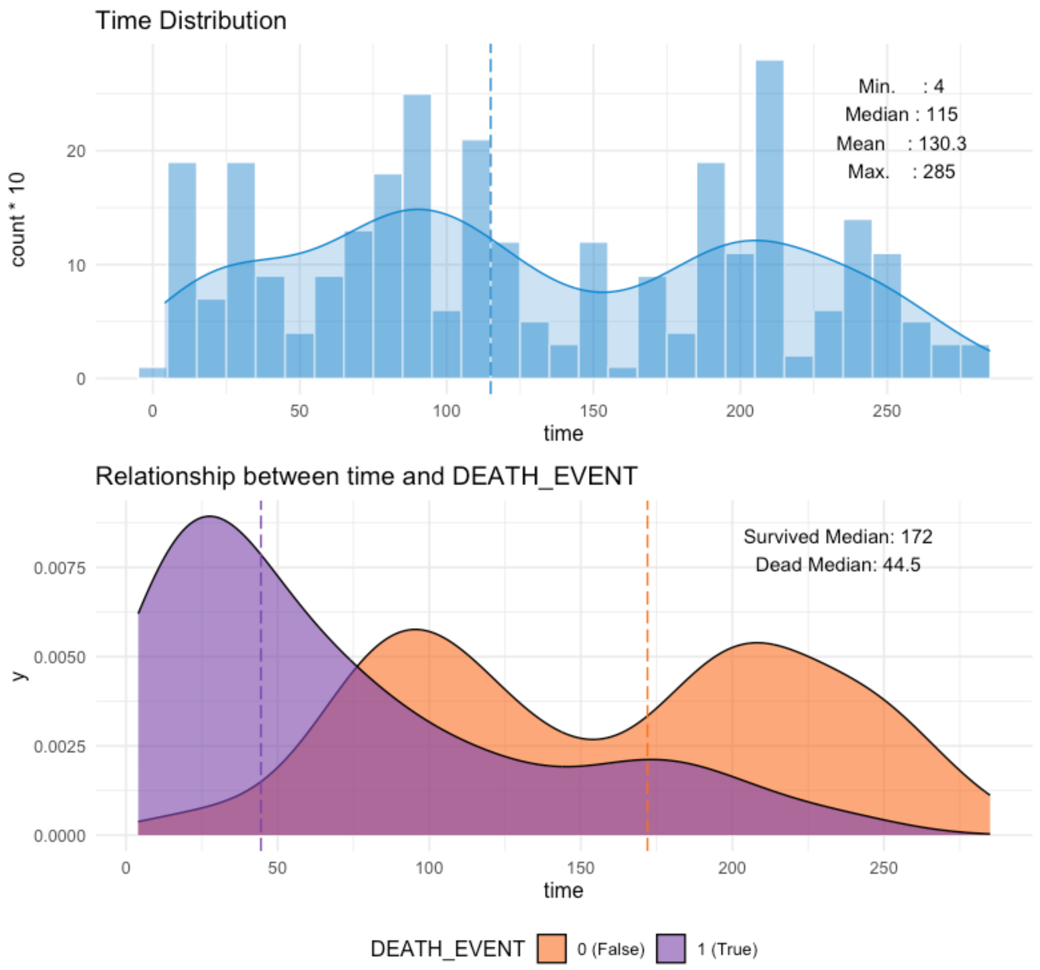
- The distribution is heavily skewed to one side, with rare cases having values more than four times the median.
- By objective variable, there are considerable differences in the shape of the distribution. For survivors, the values are clustered around the median, but for the dead, there are often cases where the values exceed 1.5.

5.7. Serum Sodium Distribution and Relationship with Death Event



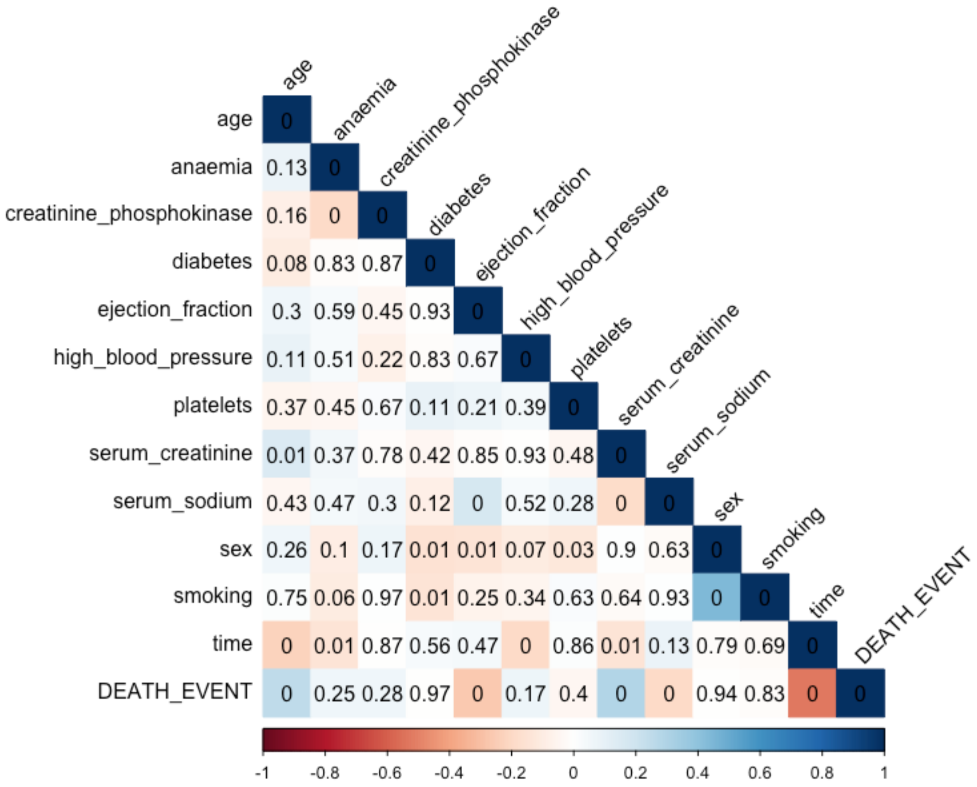
- The distribution is roughly symmetrical and almost bell-shaped, with no value exceeding 148, but there are rare cases below 125.
- By objective variable, there is some difference in the median and in the distribution. The values of survivors are clustered around the median, while the values of deaths are lower and tend to be more dispersed.

5.8. Time Distribution and Relationship with Death Event



- The distribution of the follow-up period is spread out with no large peaks, and there are small peaks around 90 and 200.
- By objective variable, there are clear differences in the medians and distributions. Survivors have a long follow-up period and two peaks in the distribution, while the dead tend to have a short follow-up period, with a gradual decrease from a large peak around 30 days.

5.9. Correlation Matrix



	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
age	0												
anaemia	0.13	0											
creatinine_phosphokinase	0.16	0	0										
diabetes	0.08	0.83	0.87	0									
ejection_fraction	0.3	0.59	0.45	0.93	0								
high_blood_pressure	0.11	0.51	0.22	0.83	0.67	0							
platelets	0.37	0.45	0.67	0.11	0.21	0.39	0						
serum_creatinine	0.01	0.37	0.78	0.42	0.85	0.93	0.48	0					
serum_sodium	0.43	0.47	0.3	0.12	0	0.52	0.28	0	0				
sex	0.26	0.1	0.17	0.01	0.01	0.07	0.03	0.9	0.63	0			
smoking	0.75	0.06	0.97	0.01	0.25	0.34	0.63	0.64	0.93	0	0		
time	0	0.01	0.87	0.56	0.47	0	0.86	0.01	0.13	0.79	0.69	0	
DEATH_EVENT	0	0.25	0.28	0.97	0	0.17	0.4	0	0	0.94	0.83	0	0

- The explanatory variables that can be said to be significantly correlated with the objective variable are, in order of increasing correlation, time, serum_creatinine, ejection_fraction, age, and serum_creatinine.
- The correlation between explanatory variables is not very high.

6. Prediction Models

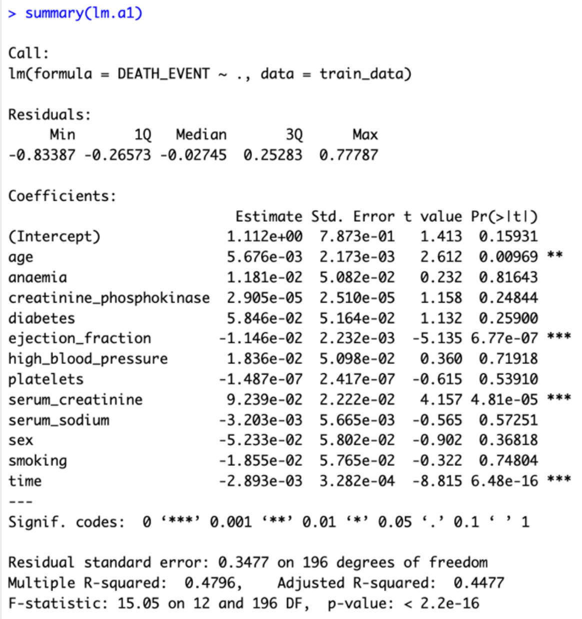
To evaluate the performance of a Multiple Linear Regression model, the original dataset containing 299 observations was divided into two sets - a training set and a test set.

For splitting the data, a random 70% sample of the total observations was taken to construct the training dataset using the sample() function in R. This resulted in 209 observations in training data out of the 299 total. The remaining 90 observations were assigned to test data using negative index subsets from the original data frame.

This 70-30 split into training (209 obs) and test (90 obs) sets will enable fitting a regression model on the train set and then assessing its performance on the unseen test data for evaluation before applying predictions to new data. The test error provides an unbiased evaluation of model fit.

6.1. Multiple Linear Regression

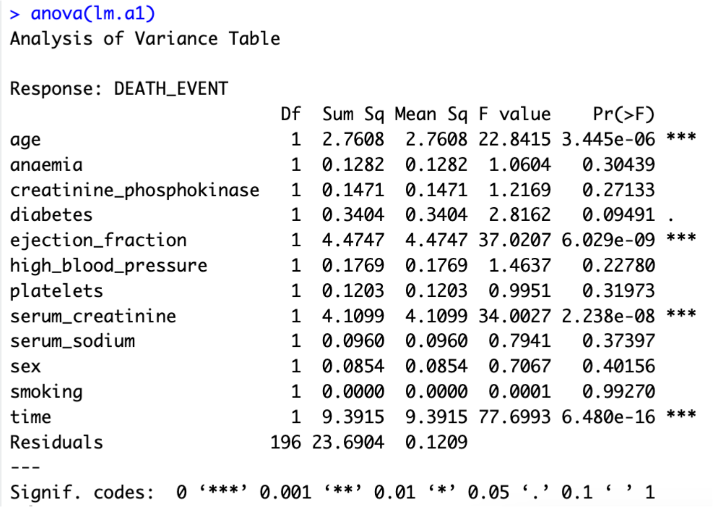
A multiple linear regression model is constructed to predict the response variable DEATH_EVENT using all available predictor variables in the dataset. Here is the summary of model.



```
> summary(lm.a1)

Call:
lm(formula = DEATH_EVENT ~ ., data = train_data)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.83387 -0.26573 -0.02745  0.25283  0.77787 

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)    
(Intercept)               1.112e+00  7.873e-01   1.413  0.15931    
age                       5.676e-03  2.173e-03   2.612  0.00969 ** 
anaemia                   1.181e-02  5.082e-02   0.232  0.81643    
creatinine_phosphokinase  2.905e-05  2.510e-05   1.158  0.24844    
diabetes                  5.846e-02  5.164e-02   1.132  0.25900    
ejection_fraction        -1.146e-02  2.232e-03  -5.135 6.77e-07 ***
high_blood_pressure       1.836e-02  5.098e-02   0.360  0.71918    
platelets                -1.487e-07  2.417e-07  -0.615  0.53910    
serum_creatinine          9.239e-02  2.222e-02   4.157 4.81e-05 ***
serum_sodium             -3.203e-03  5.665e-03  -0.565  0.57251    
sex                      -5.233e-02  5.802e-02  -0.902  0.36818    
smoking                  -1.855e-02  5.765e-02  -0.322  0.74804    
time                     -2.893e-03  3.282e-04  -8.815 6.48e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3477 on 196 degrees of freedom
Multiple R-squared:  0.4796,	Adjusted R-squared:  0.4477 
F-statistic: 15.05 on 12 and 196 DF,  p-value: < 2.2e-16
```

The model summary displays key coefficients - age, ejection_fraction, serum_creatinine and time are statistically significant predictors with p-values <0.05. The model has an R-squared of 0.4796 indicating close to 48% variability explained.

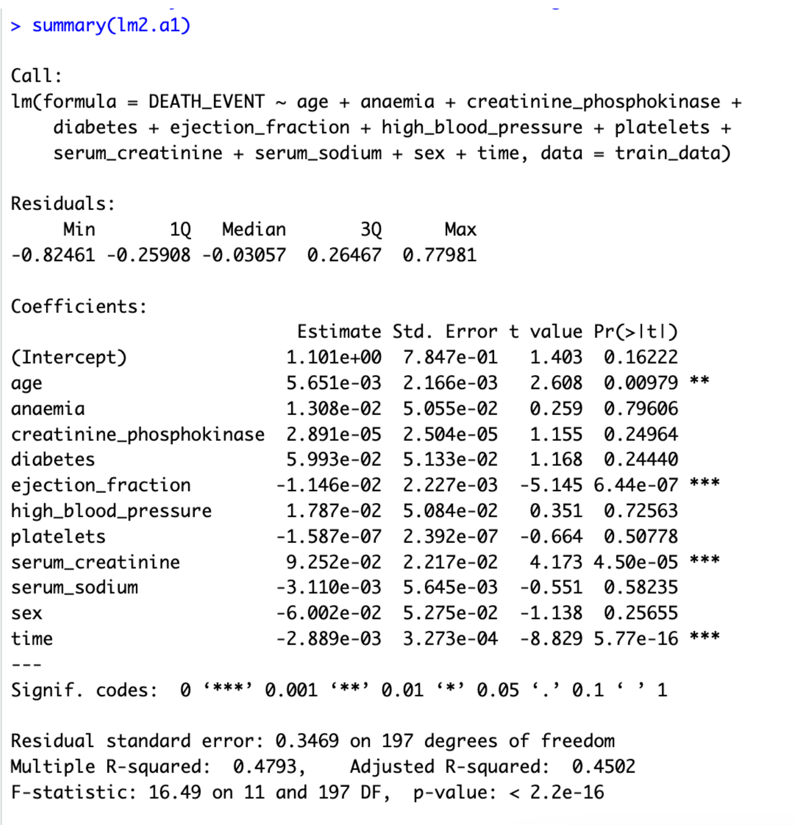
Now performs an analysis of variance on the linear regression model. Here is the result.



```
> anova(lm.a1)
Analysis of Variance Table

Response: DEATH_EVENT
                          Df  Sum Sq Mean Sq F value    Pr(>F)    
age                        1  2.7608  2.7608 22.8415 3.445e-06 ***
anaemia                    1  0.1282  0.1282  1.0604   0.30439    
creatinine_phosphokinase   1  0.1471  0.1471  1.2169   0.27133    
diabetes                   1  0.3404  0.3404  2.8162   0.09491 .  
ejection_fraction          1  4.4747  4.4747 37.0207 6.029e-09 ***
high_blood_pressure        1  0.1769  0.1769  1.4637   0.22780    
platelets                  1  0.1203  0.1203  0.9951   0.31973    
serum_creatinine           1  4.1099  4.1099 34.0027 2.238e-08 ***
serum_sodium               1  0.0960  0.0960  0.7941   0.37397    
sex                        1  0.0854  0.0854  0.7067   0.40156    
smoking                    1  0.0000  0.0000  0.0001   0.99270    
time                       1  9.3915  9.3915 77.6993 6.480e-16 ***
Residuals                196 23.6904  0.1209                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- The ANOVA results further show the individual predictive power of each variable. The large F-statistics and low p-values for variables like time, ejection_fraction and serum_creatinine indicate they explain a substantial portion of the variability. Their associations are statistically significant.
- In contrast, smoking has a tiny F-statistic of 0.0001 and high p-value of 0.9927, indicating almost zero variability explained, suggesting it likely is not a meaningful predictor in the model.
- The Residual line summarizes the unexplained variance left, which is reasonably small at 23.69 showing good model fit.

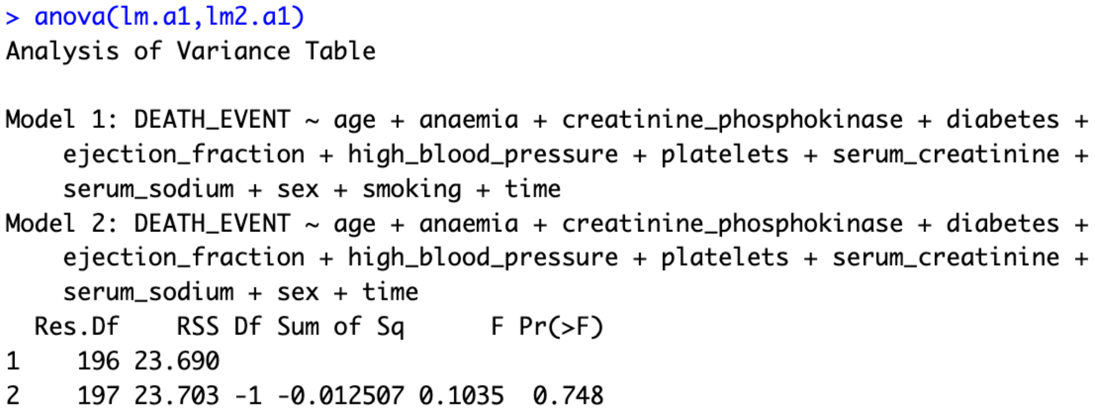
Therefore, smoking is removed from the model using update() function. So, the summary information for this new model is given below:



```
> summary(lm2.a1)

Call:
lm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
    diabetes + ejection_fraction + high_blood_pressure + platelets +
    serum_creatinine + serum_sodium + sex + time, data = train_data)

Residuals:
     Min       1Q   Median       3Q      Max
-0.82461 -0.25908 -0.03057  0.26467  0.77981

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)               1.101e+00  7.847e-01   1.403  0.16222
age                       5.651e-03  2.166e-03   2.608  0.00979 **
anaemia                   1.308e-02  5.055e-02   0.259  0.79606
creatinine_phosphokinase  2.891e-05  2.504e-05   1.155  0.24964
diabetes                  5.993e-02  5.133e-02   1.168  0.24440
ejection_fraction        -1.146e-02  2.227e-03  -5.145 6.44e-07 ***
high_blood_pressure       1.787e-02  5.084e-02   0.351  0.72563
platelets                -1.587e-07  2.392e-07  -0.664  0.50778
serum_creatinine          9.252e-02  2.217e-02   4.173 4.50e-05 ***
serum_sodium             -3.110e-03  5.645e-03  -0.551  0.58235
sex                      -6.002e-02  5.275e-02  -1.138  0.25655
time                     -2.889e-03  3.273e-04  -8.829 5.77e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3469 on 197 degrees of freedom
Multiple R-squared:  0.4793,	Adjusted R-squared:  0.4502
F-statistic: 16.49 on 11 and 197 DF,  p-value: < 2.2e-16
```

The updated model statistics reveal negligible changes - residual errors, R-squared etc remain similar even after dropping smoking variable. Therefore, to summarize, the exclusion of smoking from the multiple linear regression has not resulted in any measurable incremental percentage change in model fit or accuracy.

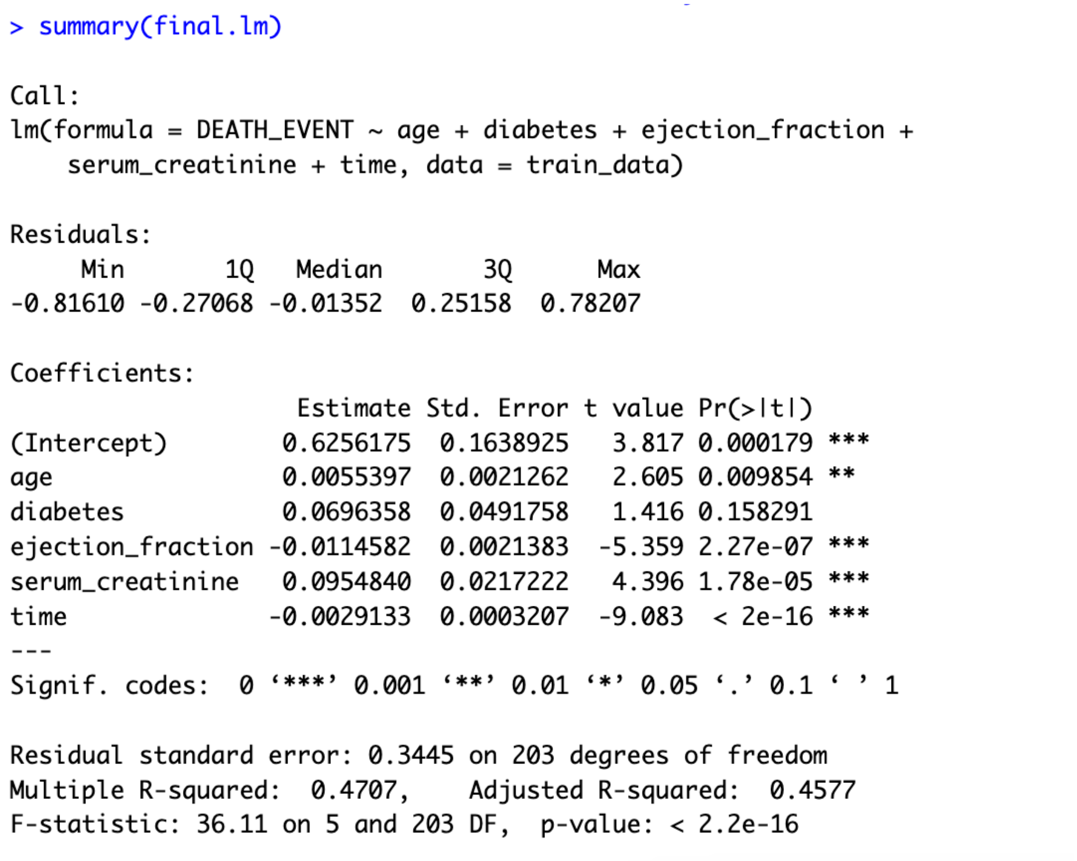
We can carry out a more formal comparison between the two models by again using the anova() function, but this time with both models.



```
> anova(lm.a1,lm2.a1)
Analysis of Variance Table

Model 1: DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + diabetes +
    ejection_fraction + high_blood_pressure + platelets + serum_creatinine +
    serum_sodium + sex + smoking + time
Model 2: DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + diabetes +
    ejection_fraction + high_blood_pressure + platelets + serum_creatinine +
    serum_sodium + sex + time
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    196 23.690
2    197 23.703 -1 -0.012507 0.1035  0.748
```

The anova() function compares model lm.a1 (original) and lm2.a1 (without smoking) in terms of residuals and fit. The high p-value > 0.05 shows excluding smoking did not improve model significantly.

Next, backwards stepwise elimination is applied on lm.a1 by using step() function which removes predictors sequentially if they do not contribute significantly.



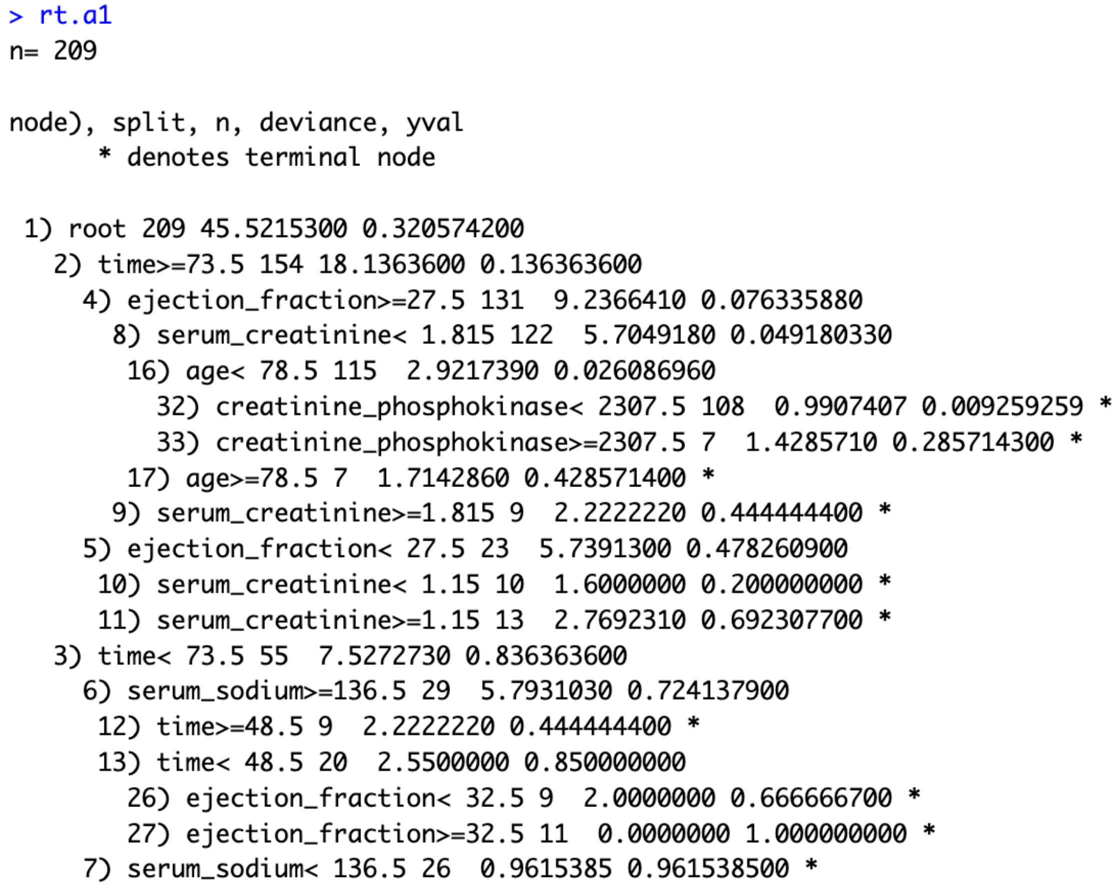
```
> summary(final.lm)

Call:
lm(formula = DEATH_EVENT ~ age + diabetes + ejection_fraction + 
    serum_creatinine + time, data = train_data)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.81610 -0.27068 -0.01352  0.25158  0.78207 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)        0.6256175  0.1638925   3.817 0.000179 ***
age                0.0055397  0.0021262   2.605 0.009854 ** 
diabetes           0.0696358  0.0491758   1.416 0.158291    
ejection_fraction -0.0114582  0.0021383  -5.359 2.27e-07 ***
serum_creatinine   0.0954840  0.0217222   4.396 1.78e-05 ***
time              -0.0029133  0.0003207  -9.083  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3445 on 203 degrees of freedom
Multiple R-squared:  0.4707,	Adjusted R-squared:  0.4577 
F-statistic: 36.11 on 5 and 203 DF,  p-value: < 2.2e-16
```

- The final model retains age, diabetes, ejection_fraction, serum_creatinine and time variables based on their statistical significance.
- The multiple R-squared has reduced from 0.4796 of lm.a1 to 0.4707 for the final model, indicating around 2% drop in variability explained. But elimination of redundant variables boosts model interpretability and efficiency focused only on significant factors.

6.2. Regression Trees

A regression tree is to predict a continuous outcome variable based on the values of predictor variables. The rpart function constructs a binary recursive partitioning model with DEATH_EVENT as response and all other variables as predictors in train data.

The content of the object is the following:



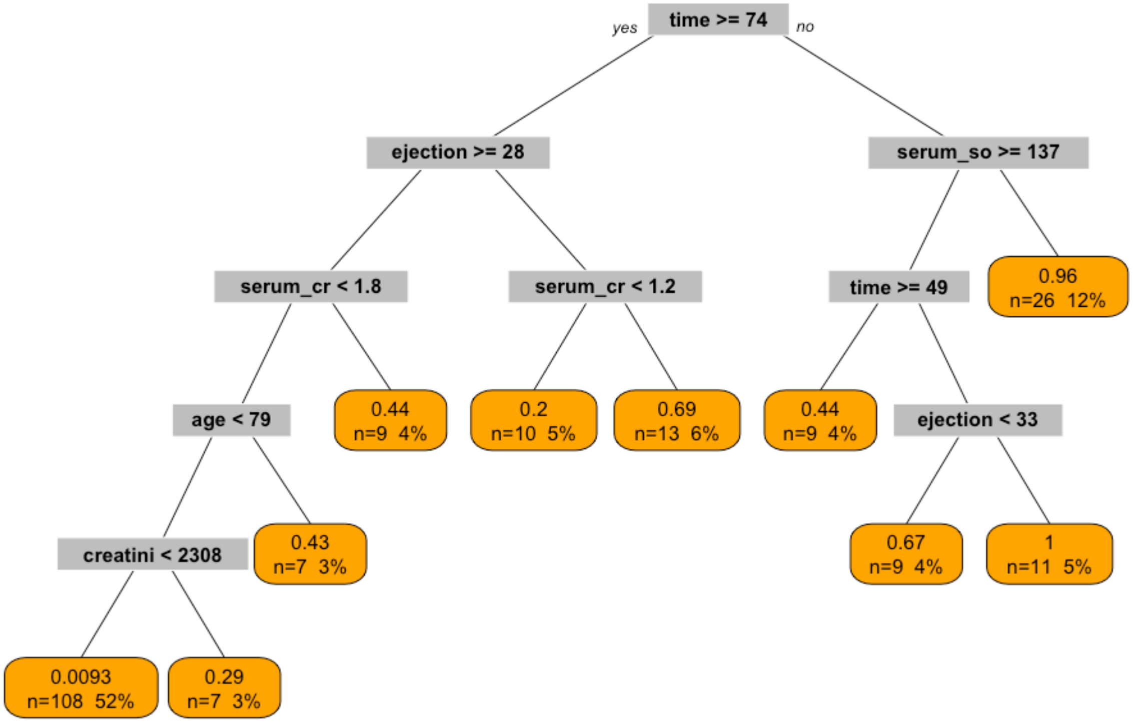
```
> rt.a1
n= 209

node), split, n, deviance, yval
      * denotes terminal node

 1) root 209 45.5215300 0.320574200
   2) time>=73.5 154 18.1363600 0.136363600
     4) ejection_fraction>=27.5 131  9.2366410 0.076335880
       8) serum_creatinine< 1.815 122  5.7049180 0.049180330
        16) age< 78.5 115  2.9217390 0.026086960
          32) creatinine_phosphokinase< 2307.5 108  0.9907407 0.009259259 *
          33) creatinine_phosphokinase>=2307.5 7  1.4285710 0.285714300 *
        17) age>=78.5 7  1.7142860 0.428571400 *
       9) serum_creatinine>=1.815 9  2.2222220 0.444444400 *
     5) ejection_fraction< 27.5 23  5.7391300 0.478260900
      10) serum_creatinine< 1.15 10  1.6000000 0.200000000 *
      11) serum_creatinine>=1.15 13  2.7692310 0.692307700 *
   3) time< 73.5 55  7.5272730 0.836363600
     6) serum_sodium>=136.5 29  5.7931030 0.724137900
      12) time>=48.5 9  2.2222220 0.444444400 *
      13) time< 48.5 20  2.5500000 0.850000000
        26) ejection_fraction< 32.5 9  2.0000000 0.666666700 *
        27) ejection_fraction>=32.5 11  0.0000000 1.000000000 *
     7) serum_sodium< 136.5 26  0.9615385 0.961538500 *
```

Key observations:

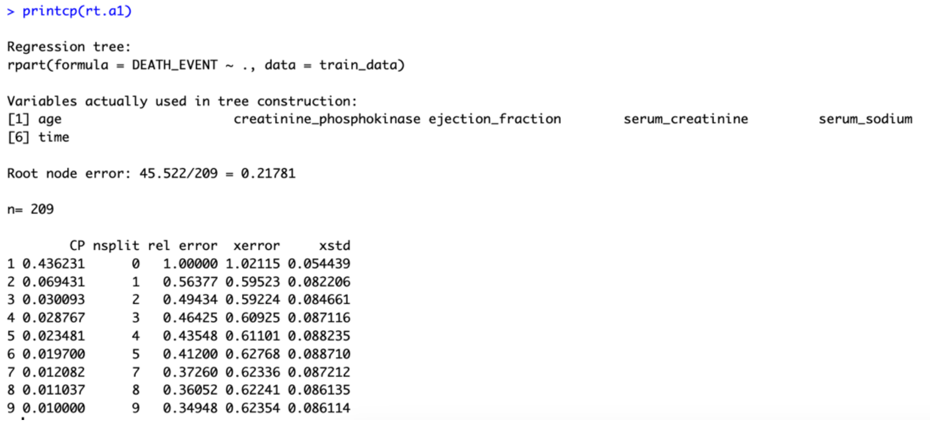
- The root node indicates that the initial split, with 209 observations and deviance is 45.52, and the predicted value for the outcome variable (DEATH_EVENT) is 0.32.
- The tree splits on the predictor variable "time" at a threshold of 73.5. When time is greater than or equal to 73.5, the predicted value for DEATH_EVENT decreases to 0.14.

- Among the observations with time >= 73.5, further splits occur based on "ejection_fraction" and "serum_creatinine" variables.
- Within the subgroup where "ejection_fraction" is greater than or equal to 27.5, additional splits occur based on "age" and "creatinine_phosphokinase."
- The tree continues to split based on different predictor variables and their thresholds, creating terminal nodes where predictions are made.
- The final tree provides a hierarchical structure for predicting the DEATH_EVENT variable based on various clinical features.

We can also obtain a graphical representation of the tree.

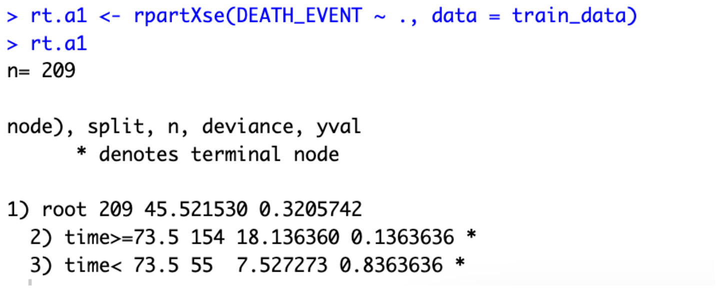


The printcp table displays complexity parameter (CP), number of splits (nsplit), relative error rates on training data (rel error) and cross-validation error rates (xerror) for a sequence of subtrees.



```
> printcp(rt.a1)

Regression tree:
rpart(formula = DEATH_EVENT ~ ., data = train_data)

Variables actually used in tree construction:
[1] age                      creatinine_phosphokinase ejection_fraction        serum_creatinine         serum_sodium
[6] time

Root node error: 45.522/209 = 0.21781

n= 209

        CP nsplit rel error  xerror     xstd
1 0.436231      0   1.00000 1.02115 0.054439
2 0.069431      1   0.56377 0.59523 0.082206
3 0.030093      2   0.49434 0.59224 0.084661
4 0.028767      3   0.46425 0.60925 0.087116
5 0.023481      4   0.43548 0.61101 0.088235
6 0.019700      5   0.41200 0.62768 0.088710
7 0.012082      7   0.37260 0.62336 0.087212
8 0.011037      8   0.36052 0.62241 0.086135
9 0.010000      9   0.34948 0.62354 0.086114
```

- The minimum test error rate occurs at CP value 0.03 corresponding to just 2 splits, representing the right-sized tree with highest predictive generalization capability.
- We can observe that we would theoretically be better off with the tree number 3, which has a lower estimated relative error (0.59224).



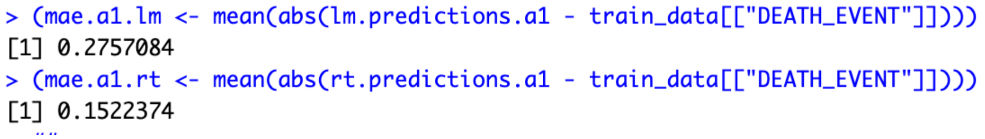
```
> rt.a1 <- rpartXse(DEATH_EVENT ~ ., data = train_data)
> rt.a1
n= 209

node), split, n, deviance, yval
      * denotes terminal node

1) root 209 45.521530 0.3205742
  2) time>=73.5 154 18.136360 0.1363636 *
  3) time< 73.5 55  7.527273 0.8363636 *
```

- It prints the final pruned tree, showing key splits on factors like time which partition the root node into child nodes more homogenous in response value.
- Here the first split identifies time >= 73.5 as the top predictor differentiating between low (0.14) and high (0.83) mortality risk subgroups. No further splits are done.
- Unlike rpart, rpartXse automatically tunes the tree size/complexity behind the scenes via 1 SE rule before returning final model. So only the optimal pruned subtree is displayed rather than entire sequence.
- In effect, it embeds safeguard against overfitting and finds simplest tree structure within 1 SE of minimum cross validation error.

7. Model Evaluation and Selection

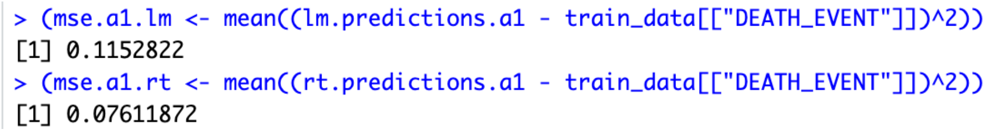
The predictive performance of regression models is obtained by comparing the predictions of the models with the real values of the target variables and calculating some average error measure from this comparison. One such measure is the mean absolute error (MAE). Let us see how to obtain this measure for our two models.

The first step is to obtain the model predictions for the set of cases where we want to evaluate it. To obtain the predictions of any model in R, one uses the function predict(). This general function receives a model and a test dataset and retrieves the corresponding model predictions. Having the predictions of the models, we can calculate their mean absolute error as follows:



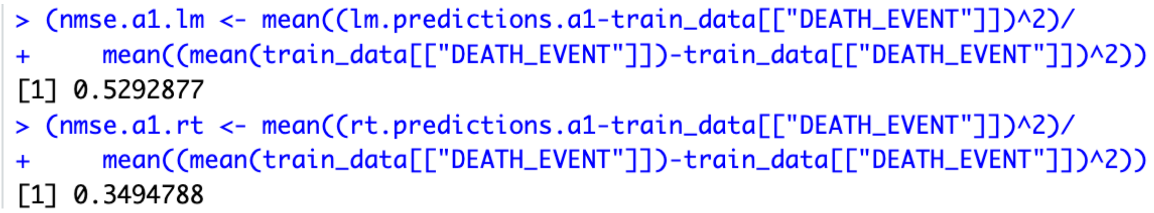
```
> (mae.a1.lm <- mean(abs(lm.predictions.a1 - train_data[["DEATH_EVENT"]])))
[1] 0.2757084
> (mae.a1.rt <- mean(abs(rt.predictions.a1 - train_data[["DEATH_EVENT"]])))
[1] 0.1522374
```

Another popular error measure is the mean squared error (MSE). This measure can be obtained as follows:



```
> (mse.a1.lm <- mean((lm.predictions.a1 - train_data[["DEATH_EVENT"]])^2))
[1] 0.1152822
> (mse.a1.rt <- mean((rt.predictions.a1 - train_data[["DEATH_EVENT"]])^2))
[1] 0.07611872
```

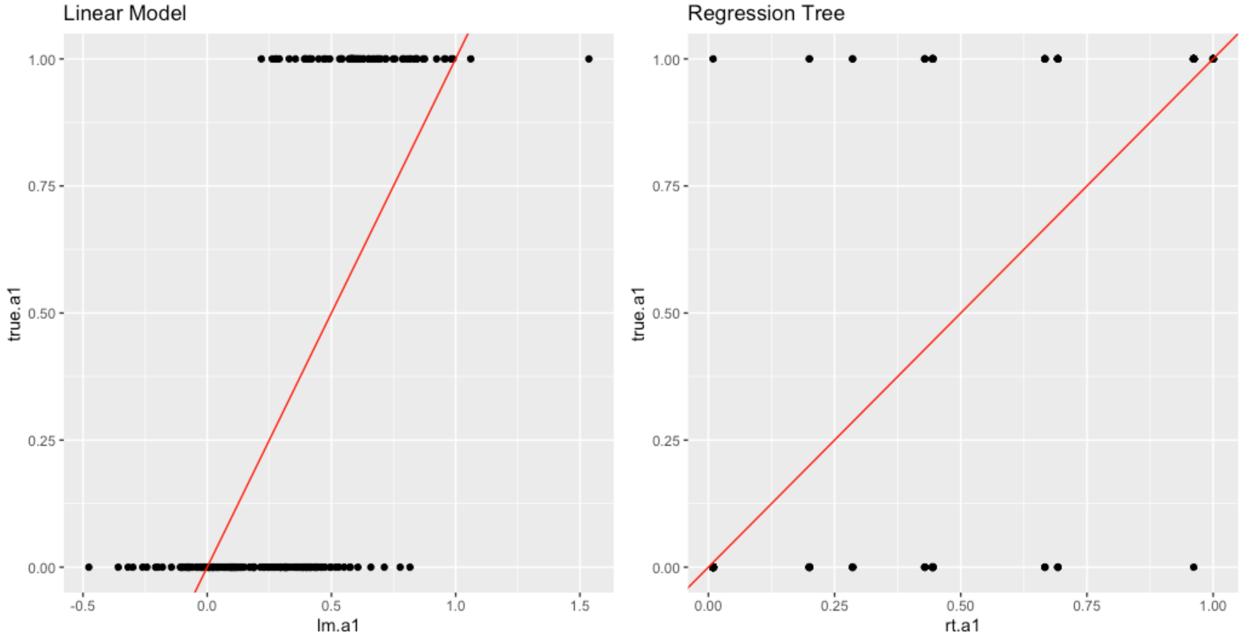
This statistic has the disadvantage of not being measured in the same units as the target variable. So Normalizes Mean Squared Error (NMSE) calculates a ratio between the performance of our models and that of a baseline predictor, usually taken as the mean value of the target variable:



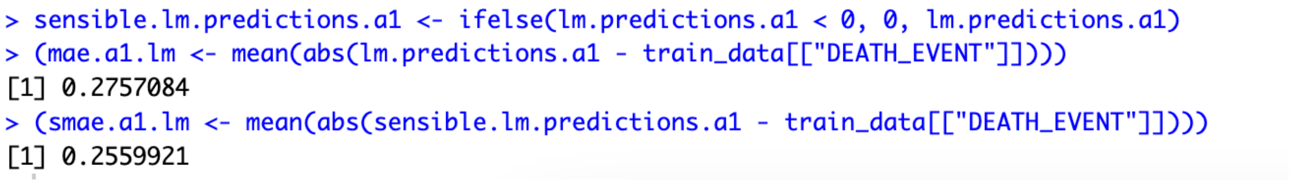
```
> (nmse.a1.lm <- mean((lm.predictions.a1-train_data[["DEATH_EVENT"]])^2)/
+     mean((mean(train_data[["DEATH_EVENT"]])-train_data[["DEATH_EVENT"]])^2))
[1] 0.5292877
> (nmse.a1.rt <- mean((rt.predictions.a1-train_data[["DEATH_EVENT"]])^2)/
+     mean((mean(train_data[["DEATH_EVENT"]])-train_data[["DEATH_EVENT"]])^2))
[1] 0.3494788
```

In summary, evaluation using both absolute error (MAE) and squared error metrics (MSE, NMSE) shows the regression tree to be the better predictive model for this problem. The non-linear patterns it captures reduces errors substantially over the linear model. Its 44% lower MAE on the training data, 34% lower MSE and 34% lower NMSE underscores the rt model's predictive advantage over the linear model.

A scatter plot of errors can be utilized for visual inspection of model predictions.



We can see that this model predicts negative DEATH EVENT frequencies for some cases. In this application domain, it makes no sense to say that the occurrence of DEATH EVENT in a sample is negative (at most, it can be zero).

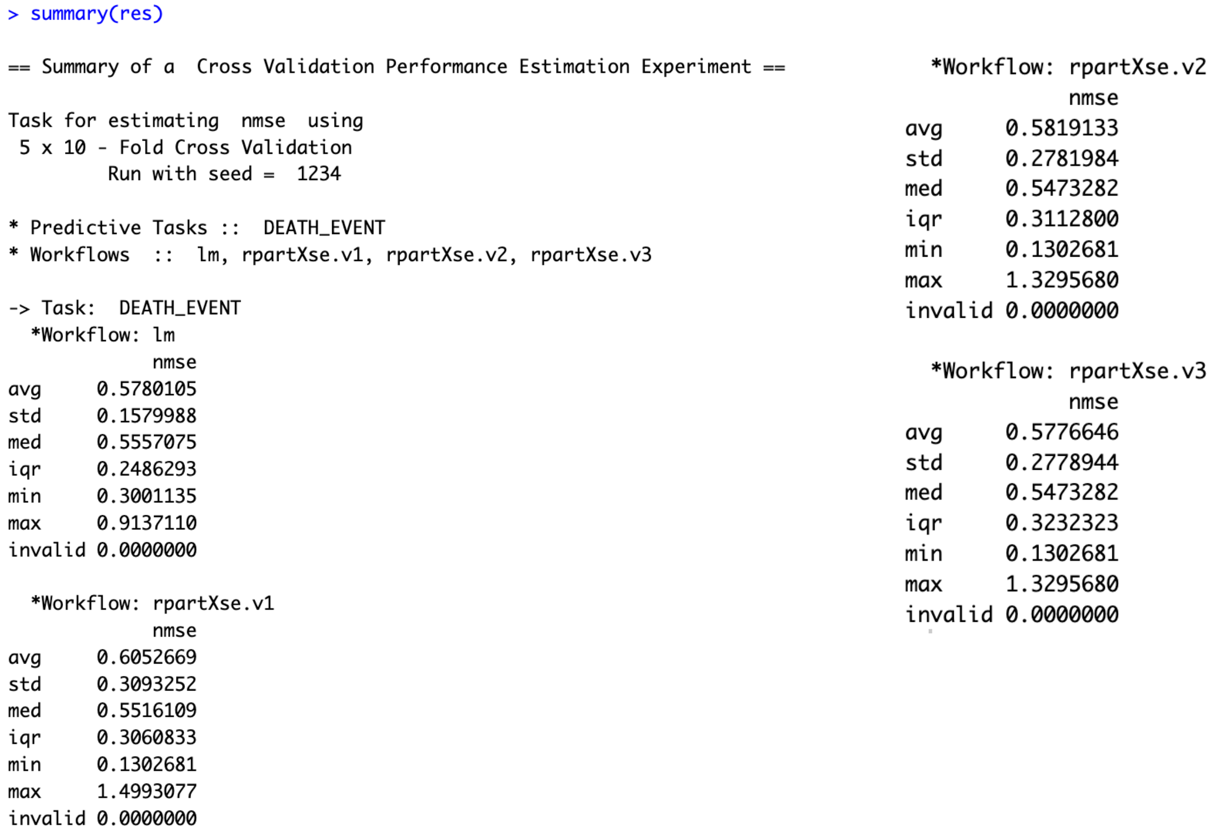


```
> sensible.lm.predictions.a1 <- ifelse(lm.predictions.a1 < 0, 0, lm.predictions.a1)
> (mae.a1.lm <- mean(abs(lm.predictions.a1 - train_data[["DEATH_EVENT"]])))
[1] 0.2757084
> (smae.a1.lm <- mean(abs(sensible.lm.predictions.a1 - train_data[["DEATH_EVENT"]])))
[1] 0.2559921
```

K-fold cross-validation

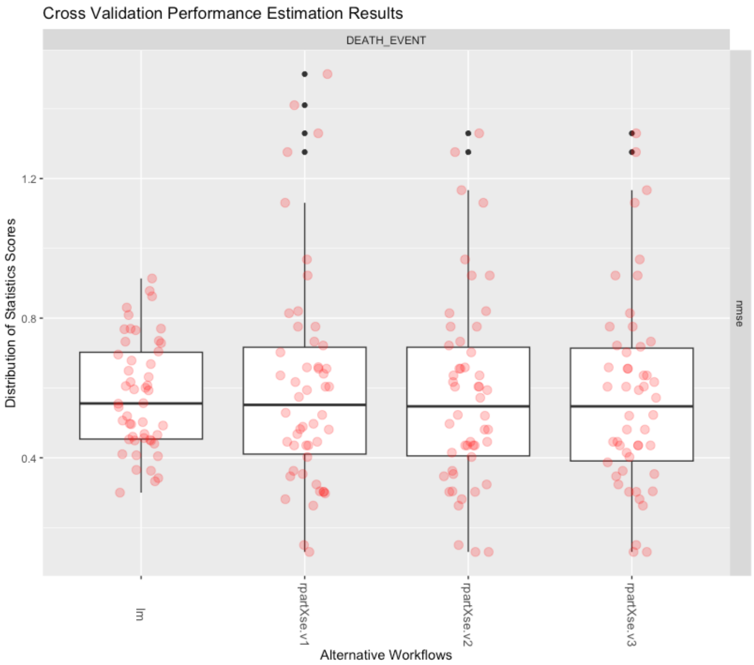
According to the performance measures calculated previously, one should prefer the regression tree to obtain the predictions for the 140 test samples as it obtained a lower NMSE. However, there is a trap in this reasoning. Calculating the performance metrics using the training data (as we did before) is unreliable because the obtained estimates are biased.

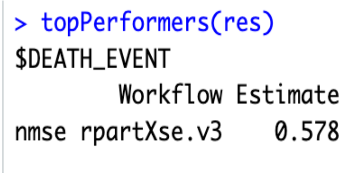
Thus, k-fold cross-validation (k-fold CV) is among the most frequently used methods for obtaining these reliable estimates for small datasets like our case study. We will use 5 repetitions of a 10-fold cross validation process to estimate the scores of this statistic of the different approaches. This estimation process is for 4 models: a linear regression model and 3 variants (different pruning levels) of a regression tree.



```
> summary(res)

== Summary of a  Cross Validation Performance Estimation Experiment ==

Task for estimating  nmse  using
 5 x 10 - Fold Cross Validation
         Run with seed =  1234

* Predictive Tasks ::  DEATH_EVENT
* Workflows  ::  lm, rpartXse.v1, rpartXse.v2, rpartXse.v3

-> Task:  DEATH_EVENT
  *Workflow: lm 
             nmse
avg     0.5780105
std     0.1579988
med     0.5557075
iqr     0.2486293
min     0.3001135
max     0.9137110
invalid 0.0000000

  *Workflow: rpartXse.v1 
             nmse
avg     0.6052669
std     0.3093252
med     0.5516109
iqr     0.3060833
min     0.1302681
max     1.4993077
invalid 0.0000000

  *Workflow: rpartXse.v2 
             nmse
avg     0.5819133
std     0.2781984
med     0.5473282
iqr     0.3112800
min     0.1302681
max     1.3295680
invalid 0.0000000

  *Workflow: rpartXse.v3 
             nmse
avg     0.5776646
std     0.2778944
med     0.5473282
iqr     0.3232323
min     0.1302681
max     1.3295680
invalid 0.0000000
```

We can also obtain a visualization of these results as follows. And Using topPerformers() function, we can check which is best model for each problem.



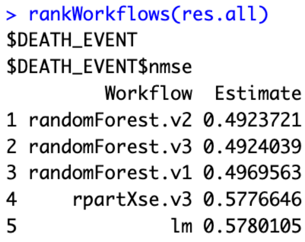


```
> topPerformers(res)
$DEATH_EVENT
         Workflow Estimate
nmse rpartXse.v3    0.578
```

Based on the analysis of the cross-validation performance estimation experiment, the best-performing workflow for predicting the DEATH_EVENT variable is rpartXse.v3, with an average normalized mean squared error (NMSE) of approximately 0.578.

Random Forest

Ensembles are model construction methods that basically try to overcome some limitations of individual models by generating a large set of alternative models and then combining their predictions. Random forests are regarded as one of the more competitive examples of ensembles.

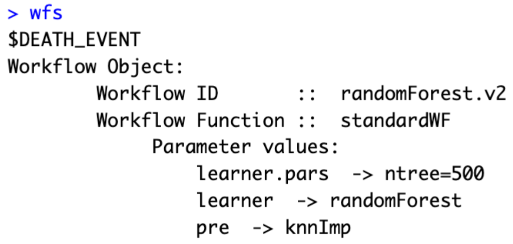


```
> rankWorkflows(res.all)
$DEATH_EVENT
$DEATH_EVENT$nmse
         Workflow  Estimate
1 randomForest.v2 0.4923721
2 randomForest.v3 0.4924039
3 randomForest.v1 0.4969563
4     rpartXse.v3 0.5776646
5              lm 0.5780105
```

The top choice for modeling this dataset is randomForest, based on cross-validated nmse comparison across multiple modeling workflows.

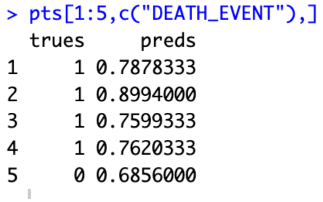
8. Predictions

We will obtain the predictions for the DEATH_EVENT on the 90 test samples. Let us start by obtaining best model using all the available training data so that we can apply them to the test set.



```
> wfs
$DEATH_EVENT
Workflow Object:
        Workflow ID       ::  randomForest.v2
        Workflow Function ::  standardWF
             Parameter values:
                 learner.pars  -> ntree=500
                 learner  -> randomForest
                 pre  -> knnImp
```

We are now obtaining the matrix with the predictions of the best workflows for the entire test set. Here is the prediction and true values for DEATH EVENT on the first 5 test cases.



```
> pts[1:5,c("DEATH_EVENT"),]
  trues     preds
1     1 0.7878333
2     1 0.8994000
3     1 0.7599333
4     1 0.7620333
5     0 0.6856000
```

Using the information, we can compare the predictions with the real values to obtain some feedback on the quality of our approach to this prediction problem.

The Normalized Mean Squared Error (NMSE) calculated on the test predictions for DEATH_EVENT is 0.577625.

This NMSE value gives insight into accuracy:

- Lower NMSE indicates smaller variance of errors between predictions and actuals.
- So NMSE of 0.577 for DEATH_EVENT suggests reasonable model performance but some scope for improvement in prediction capability for this species.

9. Summary

Initial data exploration and processing aimed to eliminate discrepancies and ensure a clean dataset, laying a solid foundation for subsequent modeling efforts. Visualizations provided insights into the distribution of features and their relationship with mortality risk, aiding in feature selection and model development.

Multiple Linear Regression and Regression Trees were constructed, with careful consideration given to feature importance and predictive capability. Model evaluation, including measures such as mean absolute error (MAE), mean squared error (MSE), and normalized mean squared error (NMSE), provided quantitative assessments of model accuracy.

The Regression Trees model emerged as the superior performer, demonstrating lower error rates and higher predictive accuracy compared to the Linear Regression model. K-fold cross-validation further validated the reliability of the models, providing robust estimates of performance across different folds of the data.

In conclusion, accuracy remains a central consideration in predictive modeling for cardiovascular diseases. By striving for accuracy and rigor in model development and evaluation, we can enhance the reliability of predictive models and ultimately improve patient outcomes in the realm of cardiac care. Continued efforts to refine and validate models will be crucial for advancing predictive analytics in the field of cardiovascular medicine.

References

Chicco, D., & Jurman, G. (2020, February 3). *Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone - BMC Medical Informatics and decision making*. BioMed Central.

Torgo, L. (2017). *Data mining with R: Learning with case studies*. CRC Press Taylor & Francis Group.