Conditional Inference Trees & Cox Regression to Predict Heart Failure Survival Time

Antonio Pano

11/10/2022

Dataset found at this link (https://archive.ics.uci.edu/ml/datasets/Heart+failure+clinical+records).

- This study was focused on survival analysis of heart failure patients who were admitted to Institute of Cardiology and Allied hospital Faisalabad-Pakistan during April-December (2015). All the patients were aged 40 years or above, having left ventricular systolic dysfunction, belonging to NYHA class III and IV.
- All 299 patients had left ventricular systolic dysfunction

Goal: To find the most influential variables and to explore them.

Findings:

- There is a statistically significant difference between proportion of males that are diabetic and females that are diabetic in the larger population from this hospital. Females hold the larger proportion in being diabetic.
- At a given instance in time, someone who has `hypertension` is 0.42 times as likely to die as someone without hypertension adjusting for age.
- Probability of survival after 150 days for those younger than 70 is 77%.
- Probability of survival after 200 days for those younger than 70 is 70%.
- 24% probability of survival after t=130 days for patients older than 79, that have less than or equal to 1.8 in serum creatine, and an ejection fraction over 25.
- Suggestion: For those diabetic, plateletes seem to reduce as age increases. [Regressions may not be statistically significant].
- On average, `creatinine_phosphokinase` is higher for non-smokers.
- Men, on average, have higher `creatinine_phosphokinase` .
- Women, on average, have a higher `platelets` count.
- `age` , `ejection fraction` , the presence of `hypertension` , and a value of `serum creatinine` greater than 1.25 are the variables that contribute most to an accurate prediction of mortality.
- `age` , `creatinine_phosphokinase` , `ejection_fraction` , `serum_creatinine` , and the presence of `hypertension` are the variables which most influence the survival rate probability.
- `anemia` , `smoking` , `sex` status, and `diabetes` , and are the fields that contribute the least to survival rates, in that order (greatest contribution to least).

Initial Variables:

- age: age of the patient (years)
- anemia: presence of critically low haematocrit levels (boolean)
- high blood pressure: if the patient has hypertension (boolean)
- creatine phosphokinase (CPK): level of the CPK enzyme in the blood (mcg/L)
- diabetes: if the patient has diabetes (boolean)
- ejection fraction: percentage of blood leaving the heart at each contraction (percentage)
- platelets: platelets in the blood (kiloplatelets/mL)
- sex: woman or man (binary)
- serum creatinine: level of serum creatinine in the blood (mg/dL)
- serum sodium: level of serum sodium in the blood (mEq/L)
- smoking: if the patient smokes or not (boolean)
- time: follow-up period (days)
- [target] death event: if the patient is deceased during the follow-up period (boolean)

```
library(skimr)
library(ggplot2)
library(dplyr)
library(tidyr)
library(patchwork)
library(survival)
library(survminer)
library(partykit)
library(coin)
library(survminer)
library(flexsurv)
library(randomForestSRC)
library(broom)
library(gtsummary)
library(splines)
```

Loading the data

```
HF <- read.csv("heart_failure_clinical_records_dataset.csv")
```

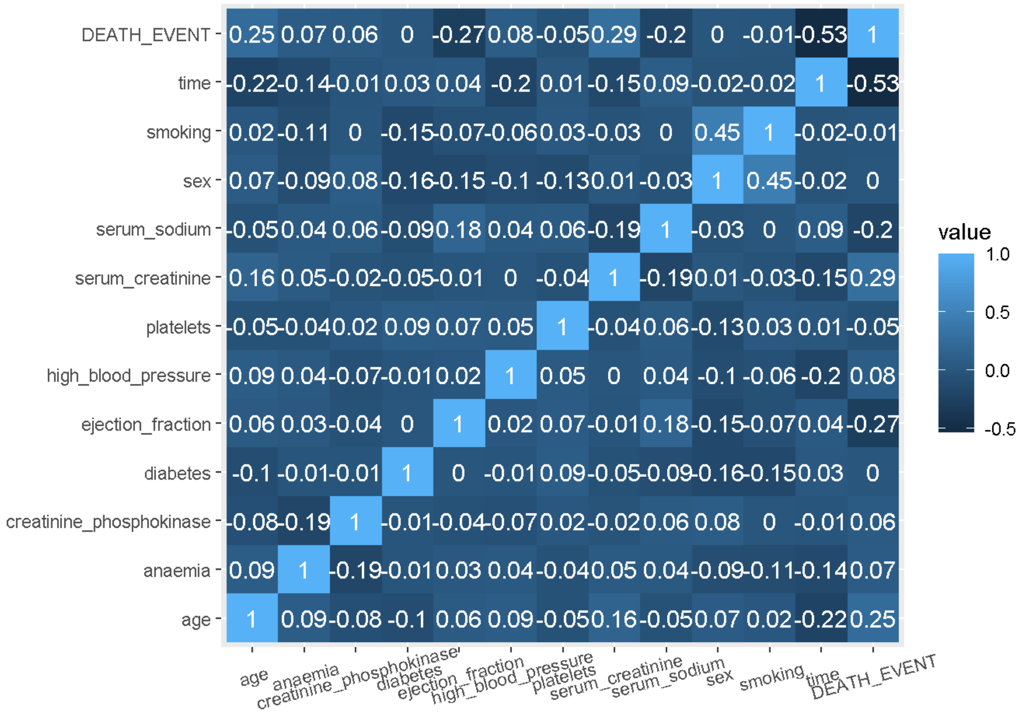
Exploratory Data Analysis

Correlation Plot

- Plotted to determine if multi-collinearity is present. If so, certain potential classification methods cannot be used.

```
cormat <- HF %>% cor() %>% round(2)
melted_cormat <- reshape2::melt(cormat)

ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() +
  geom_text(aes(Var2, Var1, label = value), color = "white", size = 4) +
   theme(axis.title.x=element_blank(),
        axis.title.y=element_blank(),
        axis.text.x = element_text(angle = 15, vjust = 0.8)
        )
```



Adjusting Variables

```
HF$anaemia = as.factor(HF$anaemia)
HF$diabetes = factor(HF$diabetes,levels=c(0,1),labels=c("Absent","Present"))
HF$hypertension = factor(HF$high_blood_pressure,levels=c(0,1),labels=c("Absent","Present"))

HF$sex = factor(HF$sex,levels=c(0,1),labels=c("Female","Male"))
HF$smoking = factor(HF$smoking,levels=c(0,1),labels=c("No","Yes"))
HF$DEATH_EVENT = as.factor(HF$DEATH_EVENT)


HF <- select(HF, -high_blood_pressure)

skim(HF)
```

Data summary

Name	HF
Number of rows	299
Number of columns	13
_______________________	
Column type frequency:	
factor	6
numeric	7
________________________	
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
anaemia	0	1	FALSE	2	0: 170, 1: 129
diabetes	0	1	FALSE	2	Abs: 174, Pre: 125
sex	0	1	FALSE	2	Mal: 194, Fem: 105
smoking	0	1	FALSE	2	No: 203, Yes: 96
DEATH_EVENT	0	1	FALSE	2	0: 203, 1: 96
hypertension	0	1	FALSE	2	Abs: 194, Pre: 105

Variable type: numeric

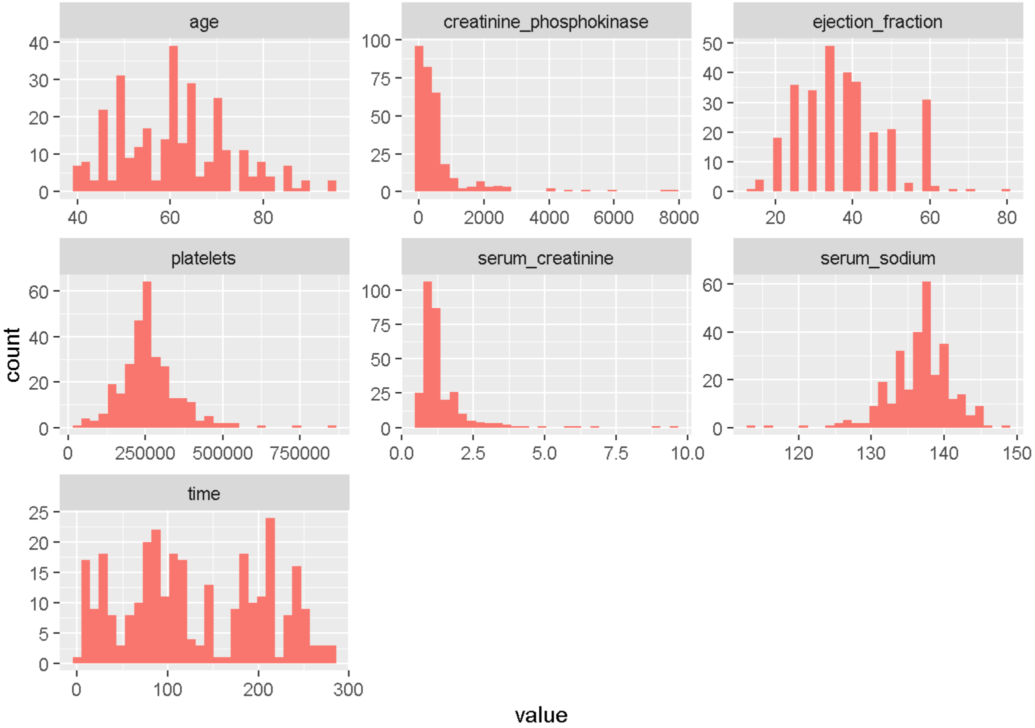
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
age	0	1	60.83	11.89	40.0	51.0	60.0	70.0	95.0	▆▇▇▃▁
creatinine_phosphokinase	0	1	581.84	970.29	23.0	116.5	250.0	582.0	7861.0	▇▁▁▁▁
ejection_fraction	0	1	38.08	11.83	14.0	30.0	38.0	45.0	80.0	▃▇▂▂▁
platelets	0	1	263358.03	97804.24	25100.0	212500.0	262000.0	303500.0	850000.0	▂▇▂▁▁
serum_creatinine	0	1	1.39	1.03	0.5	0.9	1.1	1.4	9.4	▇▁▁▁▁
serum_sodium	0	1	136.63	4.41	113.0	134.0	137.0	140.0	148.0	▁▁▃▇▁
time	0	1	130.26	77.61	4.0	73.0	115.0	203.0	285.0	▆▇▃▆▃

```
HF %>% group_by(sex, DEATH_EVENT) %>%
  summarize(count = n(), .groups="drop")
```

```
## # A tibble: 4 × 3
##   sex    DEATH_EVENT count
##   <fct>  <fct>       <int>
## 1 Female 0              71
## 2 Female 1              34
## 3 Male   0             132
## 4 Male   1              62
```

Histograms for all numeric variables

```
HF %>%
  purrr::keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_histogram(aes(fill="orange"), show.legend = FALSE)
```



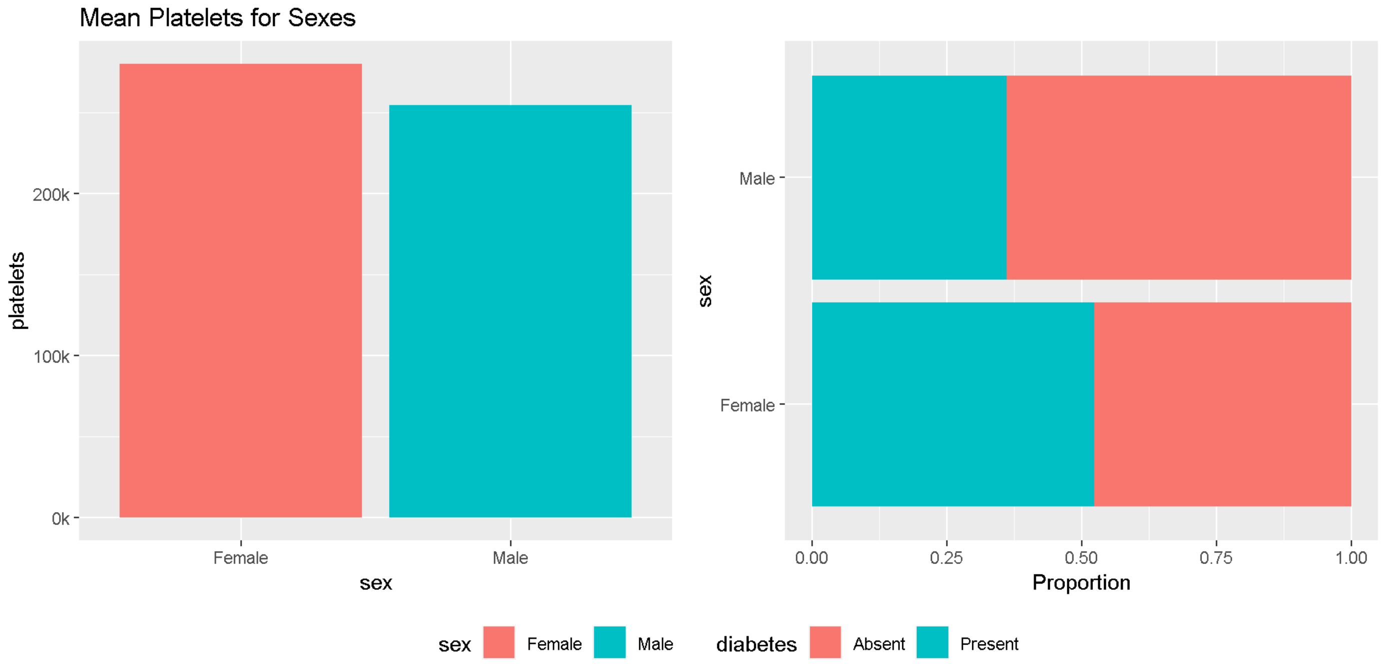
Comparing `creatinine_phosphokinase` to Men & Women– those who smoke and those who do not.

- Noticing that the average `creatinine_phosphokinase` is higher for non-smokers.
- Women, on average, have a higher `platelets` count.

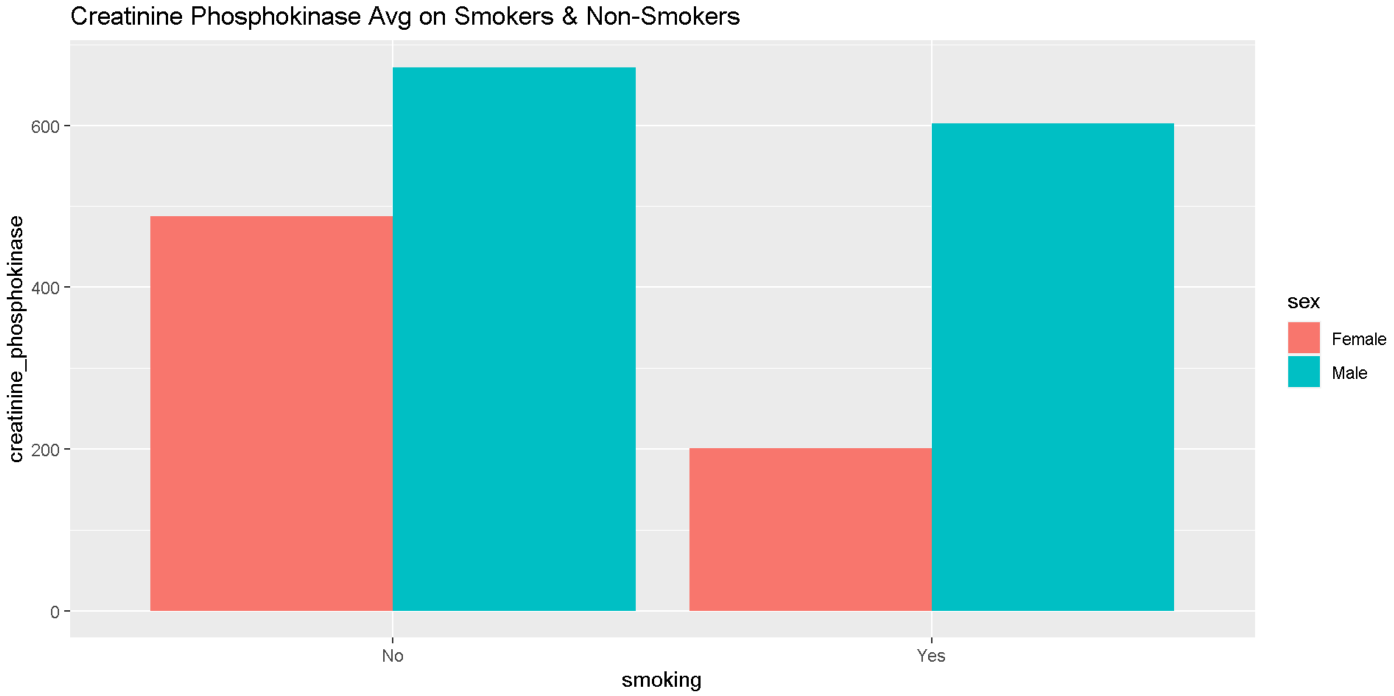
```
p1 <- ggplot(HF, aes(x=sex, y=platelets, fill=sex)) +
  geom_bar(position = "dodge", stat="summary", fun="mean") +
  scale_y_continuous(labels = scales::label_number(suffix = "k", scale = 1e-3)) +
  ggtitle("Mean Platelets for Sexes")

p2 <- ggplot(HF, aes(y=sex, fill=diabetes)) +
  geom_bar(position = "fill") + xlab("Proportion")

combined <- p1 + p2 & theme(legend.position = "bottom")
combined + plot_layout(guides = "collect")
```



```
ggplot(HF, aes(x=smoking, y=creatinine_phosphokinase, fill=sex)) +
  geom_bar(position = "dodge", stat="summary", fun="mean") +
  ggtitle("Creatinine Phosphokinase Avg on Smokers & Non-Smokers")
```



Chi-Squared Inference Testing

Is there a statistically significant correlation in the proportion of Males and Females that: have anemia, have hypertension, smoke, or are diabetic?

- No statistically significant difference on sex & anemia / sex & hypertension.
- Yes, there is a statistically significant correlation between females and diabetes. That correlation has a moderate association.
- Yes, there is a statistically significant correlation between males and smoking. Males and smoking have a very strong association.

```
HF %>% select(sex, anaemia) %>%
  table() %>% chisq.test()
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  .
## X-squared = 2.2995, df = 1, p-value = 0.1294
```

```
HF %>% select(sex, hypertension) %>%
  table() %>% chisq.test()
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  .
## X-squared = 2.8293, df = 1, p-value = 0.09256
```

```
HF %>% select(sex, smoking) %>%
  table() %>% chisq.test()
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  .
## X-squared = 57.463, df = 1, p-value = 3.444e-14
```

```
HF %>% select(sex, smoking) %>%
  table() %>% psych::Yule()
```

```
## [1] 0.9158763
```

```
HF %>% select(sex, diabetes) %>%
  table() %>% chisq.test()
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  .
## X-squared = 6.7839, df = 1, p-value = 0.009199
```

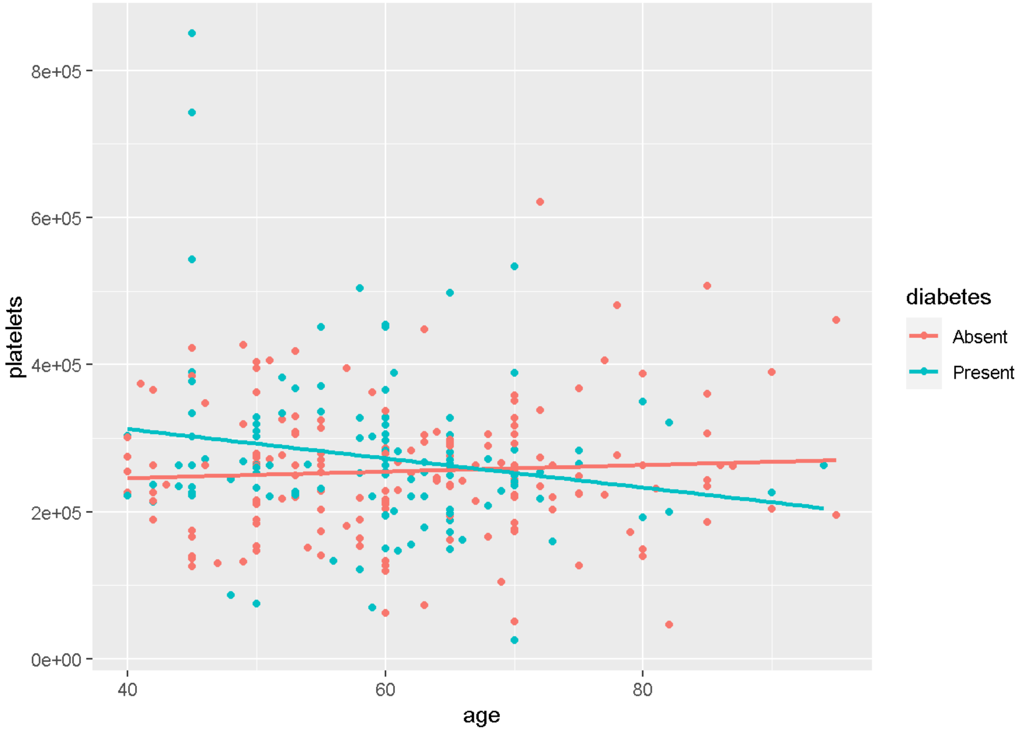
```
HF %>% select(sex, diabetes) %>%
  table() %>% psych::Yule()
```

```
## [1] -0.3217054
```

Suggestion:

- For those diabetic, plateletes reduce as age increases.
- For those who aren’t diabetic, plateletes generally stay the same and potentially, increase by a marginal amount for an unknown reason.

```
ggplot(HF, aes(x=age, y=platelets,color=diabetes)) + geom_point() +
  geom_smooth(method='lm', se = FALSE)
```



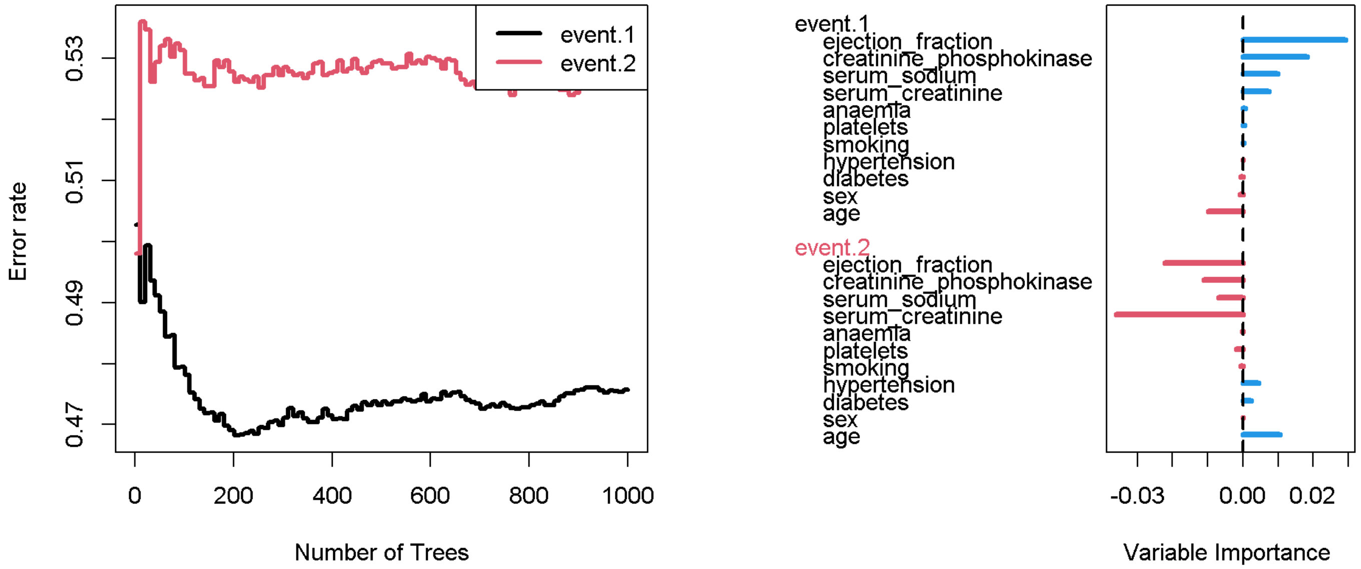
Random Forest Survival

Used to get variable importance chart.

```
set.seed(0)

# mtry means how many nodes at each split
fit <- rfsrc(Surv(time, DEATH_EVENT==1) ~ .,
             data = HF,
             ntree = 1000,
             importance = TRUE,
             nsplit = 5)

#fit
plot(fit)
```



```
## 
##                            event.1   event.2
## ejection_fraction           0.0292   -0.0222
## creatinine_phosphokinase    0.0183   -0.0112
## serum_sodium                0.0098   -0.0070
## serum_creatinine            0.0075   -0.0361
## anaemia                     0.0006   -0.0002
## platelets                   0.0004   -0.0020
## smoking                     0.0002   -0.0005
## hypertension               -0.0001    0.0044
## diabetes                   -0.0005    0.0023
## sex                        -0.0008    0.0000
## age                        -0.0098    0.0104
```

Conditional Inference Trees - Kaplan Meier Curves

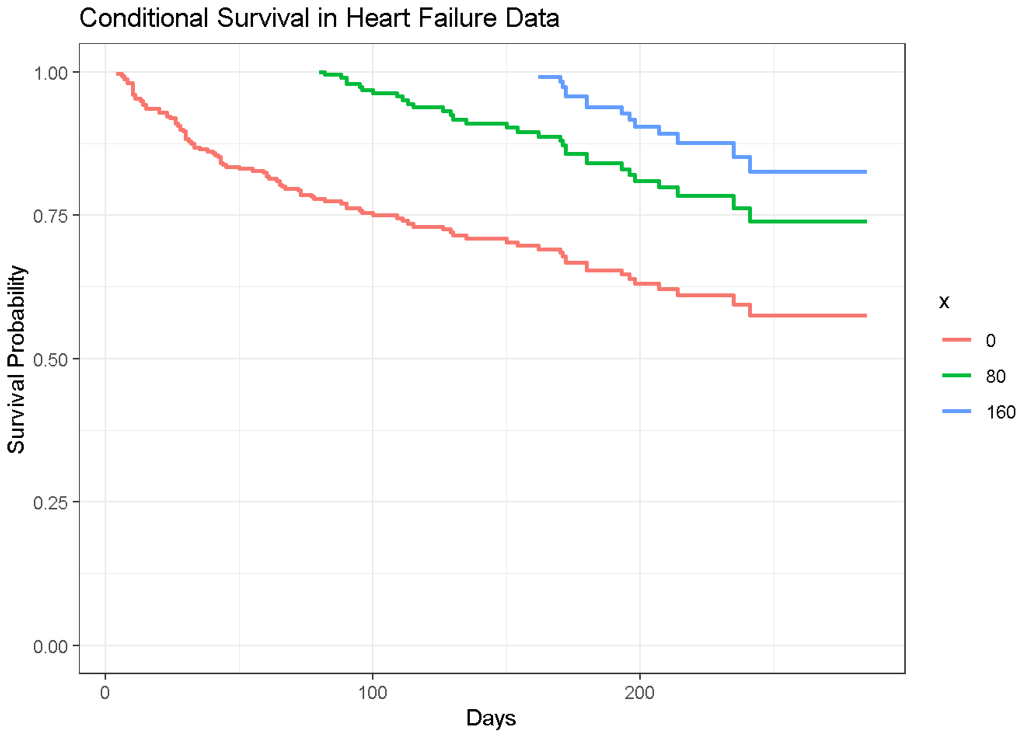
Conditional Survival: The probability of surviving further ‘t’ years, given that a patient has already survived ‘s’ years.

```
# When it comes to survfit() & surv() objects, death variable must be numeric!
HF$DEATH_EVENT = as.numeric(as.character(HF$DEATH_EVENT))


remotes::install_github("zabore/condsurv")
library(condsurv)

fit_cond <- survfit(Surv(time, DEATH_EVENT) ~ 1, data = HF)

gg_conditional_surv(
  basekm = fit_cond,
  at = seq(0, 160, 80),
  main = "Conditional Survival in Heart Failure Data",
  xlab = "Days",
  ylab = "Survival Probability"
  )
```



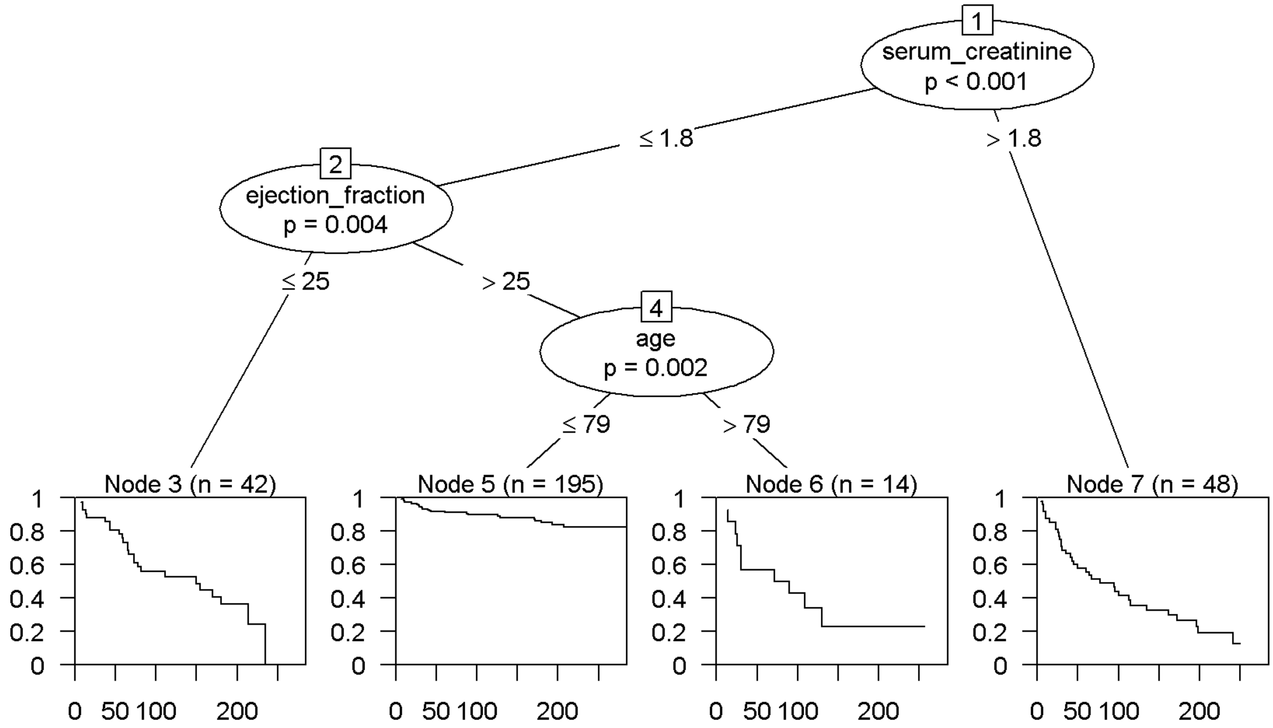
Conditional Inference Tree

We can see remaining cases in which the curves did not drop to the x-axis due to there still being patients alive by the end of the subsets.

Insight from this graph: * `Serum Creatinine` is highly significant with the showcased split at 1.8 for survival prediction.

```
# Creating a Conditional Inference Tree for descriptive analytics
CondInfTree <- ctree(Surv(time, DEATH_EVENT) ~ .,
                     data = HF,
                     control = ctree_control(alpha = 0.05))

plot(CondInfTree)
```



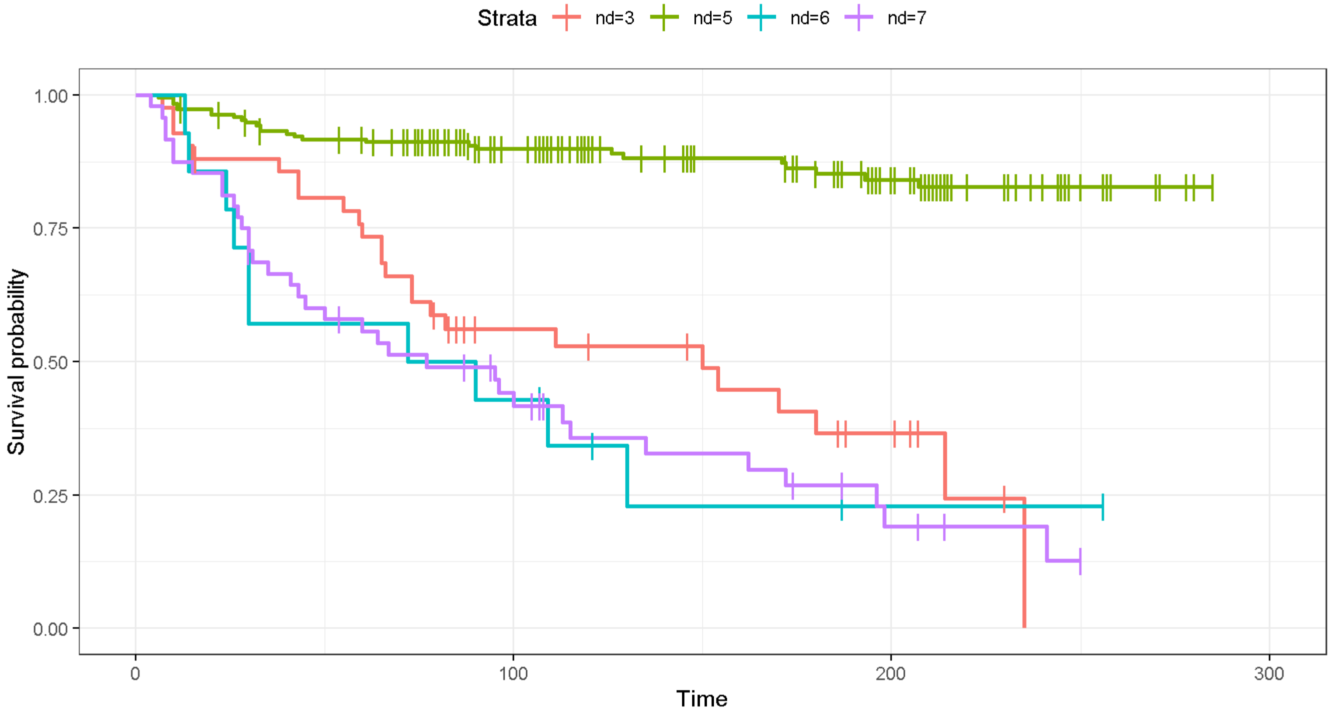
Plotting all node distributions/curves in one plot.

- Most notable is the second distribution (left to right) with a minimal survival rate of 82% at any given point in time.

```
nd <- factor(predict(CondInfTree, type = "node"))

all_nd <- survfit(Surv(time, DEATH_EVENT) ~ nd, data = HF)

ggsurvplot(all_nd, data = HF,
           censor.shape="|",
           conf.int = FALSE, #surv.median.line = "hv",
           ggtheme = theme_bw())
```



```
# Extracting survival curve for only one observation from the ctree. Perhaps an outlier.
#nd1 <- predict(CondInfTree, type = "prob")[[10]]
#summary(nd1, times=c(20, 45, 60, 80, 100, 10*(11:15)))
```

Constructing an exponential curve for previous graph’s first survival curve. * 48% probability of survival after t=150 days for patients older that have less than or equal to 1.8 in serum creatine, and an ejection fraction under 25.

```
K <- HF %>%
  filter(serum_creatinine <= 1.8, ejection_fraction <= 25)



# This one is best.
# The ~ 1 is our way ofletting R know that we aren't using any x variables. Just time and whether event occured which are both
y variabes.
pred_k_surv <- survfit(Surv(time, DEATH_EVENT) ~ 1, data = K)

summary(pred_k_surv, times=c(20, 45, 60, 80, 100, 10*(11:15)))
```

```
## Call: survfit(formula = Surv(time, DEATH_EVENT) ~ 1, data = K)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    20     36       5    0.881  0.0500        0.788        0.985
##    45     33       3    0.808  0.0612        0.696        0.937
##    60     31       3    0.734  0.0688        0.611        0.882
##    80     23       6    0.587  0.0768        0.454        0.759
##   100     17       1    0.562  0.0776        0.429        0.736
##   110     17       0    0.562  0.0776        0.429        0.736
##   120     16       1    0.529  0.0798        0.393        0.711
##   130     14       0    0.529  0.0798        0.393        0.711
##   140     14       0    0.529  0.0798        0.393        0.711
##   150     13       1    0.488  0.0834        0.349        0.682
```

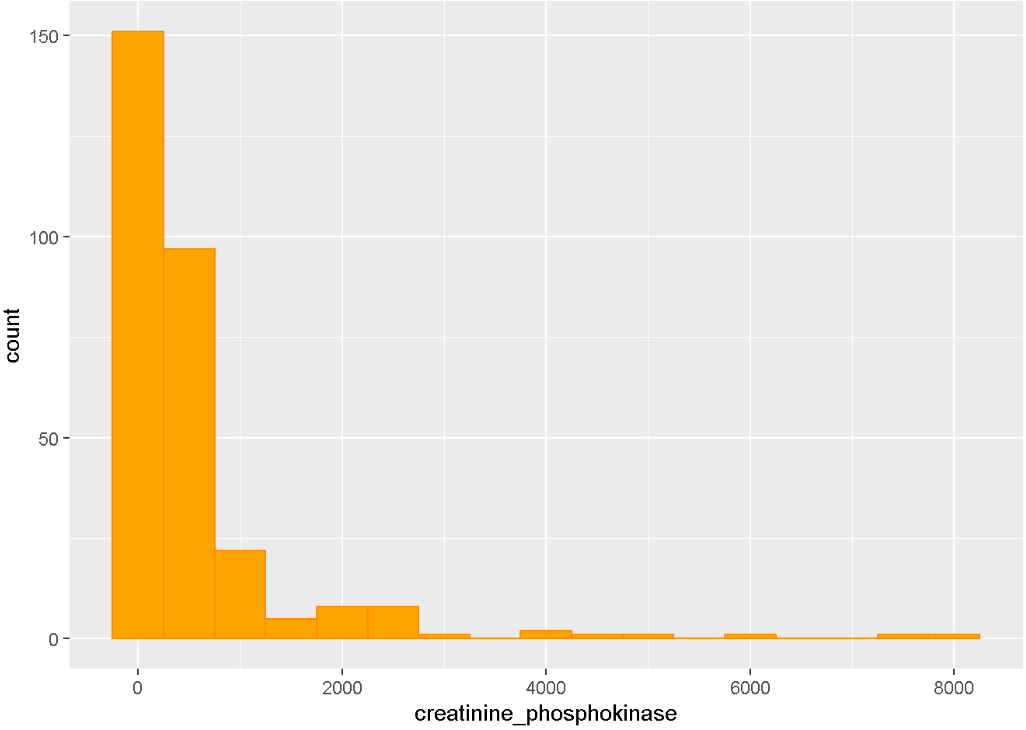
- No pruning was done since most trees found revolve around the same 3 variables.
- Probability of survival after 150 days for those younger than 70 is 77%.
- Probability of survival after 200 days for those younger than 70 is 70%.

```
survfit(Surv(time, DEATH_EVENT) ~ 1, data = HF %>% filter(age <= 70)) %>%
  tbl_survfit(
    times = c(150,200),
    label_header = "**{time} Day Survival (95% CI) For Those Younger Than 70**"
    )
```

Characteristic	**150 Day Survival (95% CI) For Those Younger Than 70**	**200 Day Survival (95% CI) For Those Younger Than 70**
Overall	77% (71%, 82%)	70% (64%, 77%)

- Creatine_Phosphokinase not being a split variable in the conditional inference tree lead me to look at in closer.

```
ggplot(HF, aes(x=creatinine_phosphokinase)) + geom_histogram(binwidth = 500, fill = "orange", color = "darkorange")
```



```
survfit(Surv(time, DEATH_EVENT) ~ 1, data = HF %>% filter(creatinine_phosphokinase <= 1000)) %>%
  tbl_survfit(
    times = c(150,200),
    label_header = "**{time} Day Survival (95% CI) For Those with less than 1000 in Creatine Phosphokinase**"
    )
```

Characteristic	150 Day Survival (95% CI) For Those with less than 1000 in Creatine Phosphokinase	200 Day Survival (95% CI) For Those with less than 1000 in Creatine Phosphokinase
Overall	69% (64%, 76%)	62% (55%, 69%)

Cox Proportional Hazards Model (for predictions)

Checking Cox Regression Assumptions for Final Model

- Checking Linearity of Model
- * Linearity of the final cox regression is sufficient. *
- `
- `Anaemia`, `platelets`, `diabetes`, `smoking`, and `sex` were the least useful variables for an optimal model in predicting the survival rate of a random future patient.

```
initialMod = coxph(Surv(time, DEATH_EVENT) ~ ., data=HF)
reducedMod <- step(initialMod, direction = "backward", trace = FALSE)
summary(reducedMod)
```

```
## Call:
## coxph(formula = Surv(time, DEATH_EVENT) ~ age + anaemia + creatinine_phosphokinase +
##     ejection_fraction + serum_creatinine + serum_sodium + hypertension,
##     data = HF)
##
##   n= 299, number of events= 96
##
##                               coef  exp(coef)   se(coef)      z Pr(>|z|)
## age                       4.357e-02  1.045e+00  8.831e-03  4.934 8.05e-07 ***
## anaemia1                  4.460e-01  1.562e+00  2.150e-01  2.074   0.0380 *
## creatinine_phosphokinase  2.101e-04  1.000e+00  9.825e-05  2.138   0.0325 *
## ejection_fraction        -4.747e-02  9.536e-01  1.027e-02 -4.621 3.82e-06 ***
## serum_creatinine          3.139e-01  1.369e+00  6.895e-02  4.552 5.31e-06 ***
## serum_sodium             -4.569e-02  9.553e-01  2.336e-02 -1.956   0.0505 .
## hypertensionPresent       4.965e-01  1.643e+00  2.137e-01  2.324   0.0201 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                          exp(coef) exp(-coef) lower .95 upper .95
## age                         1.0445     0.9574    1.0266     1.063
## anaemia1                    1.5621     0.6402    1.0249     2.381
## creatinine_phosphokinase    1.0002     0.9998    1.0000     1.000
## ejection_fraction           0.9536     1.0486    0.9346     0.973
## serum_creatinine            1.3688     0.7306    1.1957     1.567
## serum_sodium                0.9553     1.0468    0.9126     1.000
## hypertensionPresent         1.6430     0.6086    1.0808     2.498
##
## Concordance= 0.738  (se = 0.027 )
## Likelihood ratio test= 80.58  on 7 df,   p=1e-14
## Wald test            = 88.43  on 7 df,   p=3e-16
## Score (logrank) test = 87.66  on 7 df,   p=4e-16
```

```
# Comparing AICs between the reduced model & the model above since there is a chance an optimal model wasn't found due to the n
ature of backward selection
# `reducedMod` has a lower AIC, after all
extractAIC(initialMod)
```

```
## [1]  11.0000 958.4557
```

```
extractAIC(reducedMod)
```

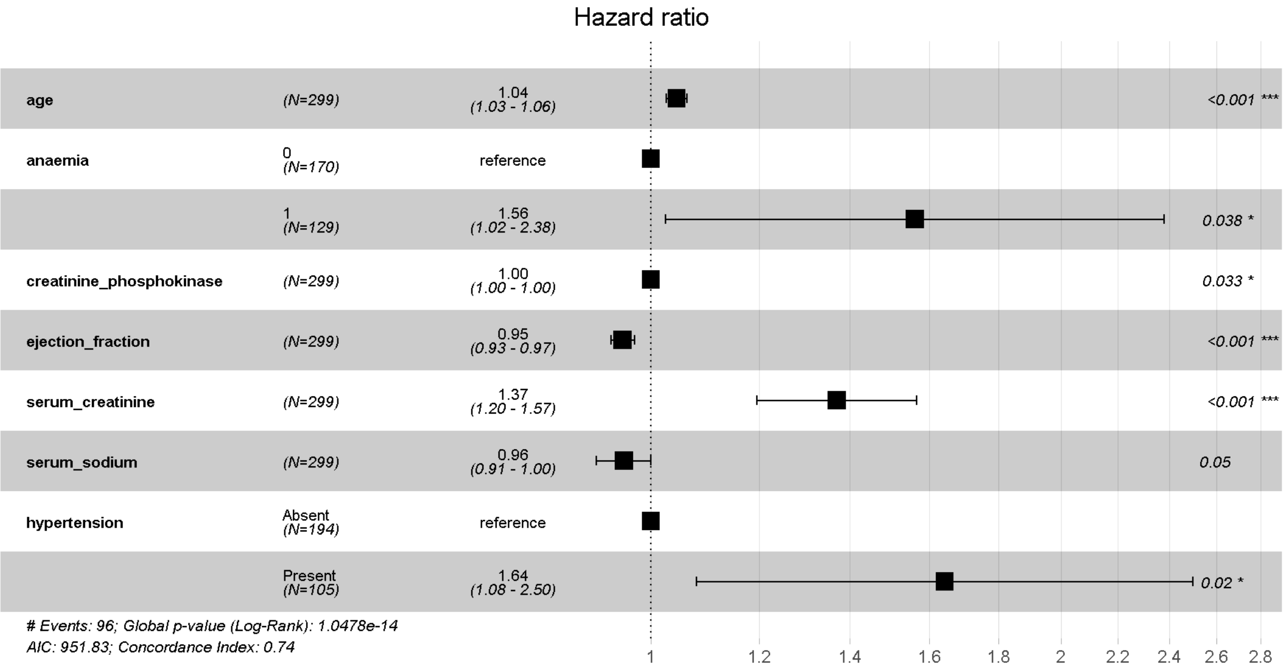
```
## [1]   7.0000 951.8277
```

```
# Likelihood ratio test
# Ho: Both models are equally as good for predictions
# Ha: Larger model is better
# We fail to reject the null hypothesis
anova(initialMod, reducedMod, test = "LRT")
```

```
## Analysis of Deviance Table
##  Cox model: response is  Surv(time, DEATH_EVENT)
##  Model 1: ~ age + anaemia + creatinine_phosphokinase + diabetes + ejection_fraction + platelets + serum_creatinine + serum_s
odium + sex + smoking + hypertension
##  Model 2: ~ age + anaemia + creatinine_phosphokinase + ejection_fraction + serum_creatinine + serum_sodium + hypertension
##    loglik  Chisq Df P(>|Chi|)
## 1 -468.23
## 2 -468.91 1.3719  4    0.8491
```

```
# Choosing the model that *does* include `Serum Sodium` since it is an easy-to-obtain predictor. Choosing to err on the side of
inclusion.

# Plotting a forest plot
ggforest(reducedMod, data = HF)
```



Checking Cox Regression assumptions for potential inference Cox model

- See r chunk for explanation on why I am categorizing `ejection_fraction` using [Mayo Clinic](https://www.mayoclinic.org/tests-procedures/ekg/expert-answers/ejection-fraction/faq-20058286# (https://www.mayoclinic.org/tests-procedures/ekg/expert-answers/ejection-fraction/faq-20058286#):~:text=A%20normal%20ejection%20fraction%20is,between%2041%25%20and%2050%25.

```
# Checking for the proportional hazards assumption using Schoenfeld test for PH
# Ho: Hazards are proportional; Ha: Hazards are NOT proportional
# Returns a test for each var and for overall model
cox.zph(reducedMod)
```

```
##                             chisq df     p
## age                       0.05920  1 0.808
## anaemia                   0.00531  1 0.942
## creatinine_phosphokinase  0.98930  1 0.320
## ejection_fraction         4.76495  1 0.029
## serum_creatinine          1.67518  1 0.196
## serum_sodium              0.09377  1 0.759
## hypertension              0.00943  1 0.923
## GLOBAL                   10.52084  7 0.161
```

```
# using spline to fix ejection fraction
splineMod <- coxph(Surv(time, DEATH_EVENT) ~ age+anaemia+creatinine_phosphokinase+ns(ejection_fraction, knots=c(15))+
                serum_sodium+serum_creatinine+hypertension, data=HF)

cox.zph(splineMod)
```

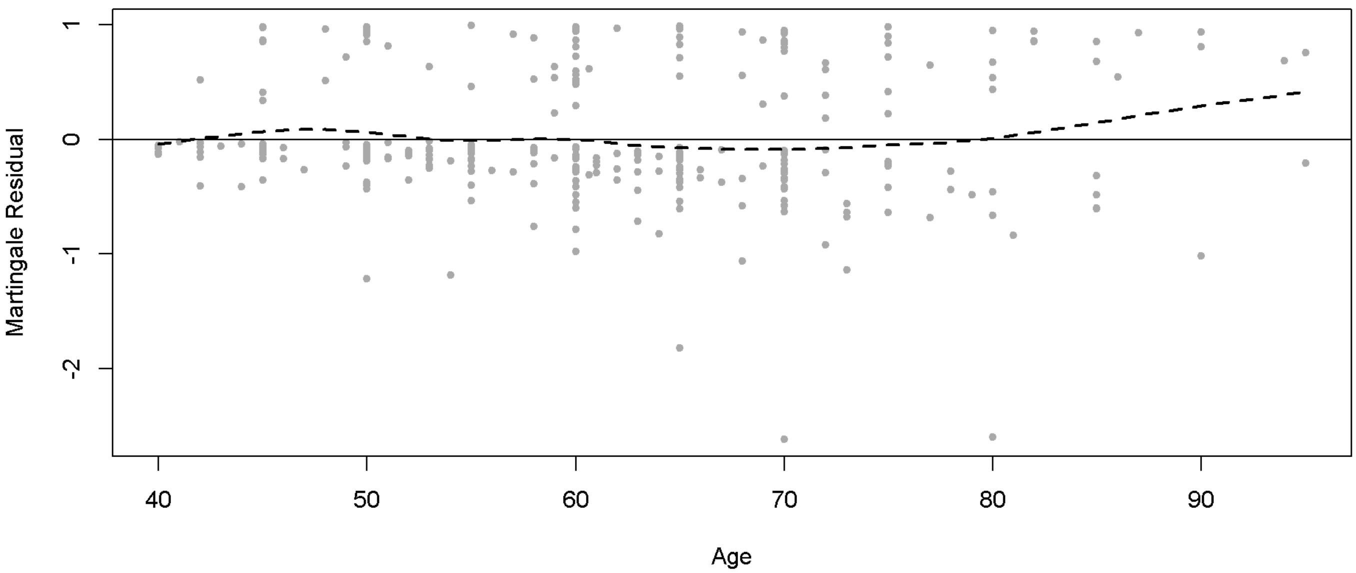
```
##                                        chisq df    p
## age                                   0.1930  1 0.66
## anaemia                               0.0405  1 0.84
## creatinine_phosphokinase              1.1392  1 0.29
## ns(ejection_fraction, knots = c(15)) 5.6171  2 0.06
## serum_sodium                          0.1133  1 0.74
## serum_creatinine                      0.4177  1 0.52
## hypertension                          0.0683  1 0.79
## GLOBAL                                9.1733  8 0.33
```

```
# summary of spline model; Final Model
# Leaving serum sodium in the model as it is an easier feature to measure in a patient. Choosing to err on the side of inclusio
n.
summary(splineMod)
```

```
## Call:
## coxph(formula = Surv(time, DEATH_EVENT) ~ age + anaemia + creatinine_phosphokinase + 
##     ns(ejection_fraction, knots = c(15)) + serum_sodium + serum_creatinine + 
##     hypertension, data = HF)
## 
##   n= 299, number of events= 96 
## 
##                                             coef  exp(coef)   se(coef)      z
## age                                    4.805e-02  1.049e+00  8.937e-03  5.377
## anaemia1                               4.466e-01  1.563e+00  2.173e-01  2.055
## creatinine_phosphokinase               2.389e-04  1.000e+00  9.772e-05  2.444
## ns(ejection_fraction, knots = c(15))1 -5.000e+00  6.739e-03  8.620e-01 -5.800
## ns(ejection_fraction, knots = c(15))2 -6.889e-01  5.021e-01  7.908e-01 -0.871
## serum_sodium                          -4.724e-02  9.539e-01  2.363e-02 -1.999
## serum_creatinine                       2.276e-01  1.256e+00  7.413e-02  3.070
## hypertensionPresent                    3.919e-01  1.480e+00  2.181e-01  1.796
##                                       Pr(>|z|)    
## age                                   7.57e-08 ***
## anaemia1                               0.03991 *  
## creatinine_phosphokinase               0.01451 *  
## ns(ejection_fraction, knots = c(15))1 6.62e-09 ***
## ns(ejection_fraction, knots = c(15))2  0.38367    
## serum_sodium                           0.04561 *  
## serum_creatinine                       0.00214 ** 
## hypertensionPresent                    0.07244 .  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
##                                       exp(coef) exp(-coef) lower .95 upper .95
## age                                    1.049227     0.9531  1.031009    1.0678
## anaemia1                               1.562948     0.6398  1.020803    2.3930
## creatinine_phosphokinase               1.000239     0.9998  1.000047    1.0004
## ns(ejection_fraction, knots = c(15))1  0.006739   148.3865  0.001244    0.0365
## ns(ejection_fraction, knots = c(15))2  0.502133     1.9915  0.106589    2.3655
## serum_sodium                           0.953854     1.0484  0.910678    0.9991
## serum_creatinine                       1.255549     0.7965  1.085756    1.4519
## hypertensionPresent                    1.479720     0.6758  0.964939    2.2691
## 
## Concordance= 0.758  (se = 0.025 )
## Likelihood ratio test= 91.2  on 8 df,   p=3e-16
## Wald test            = 99.93  on 8 df,   p=<2e-16
## Score (logrank) test = 106.8  on 8 df,   p=<2e-16
```

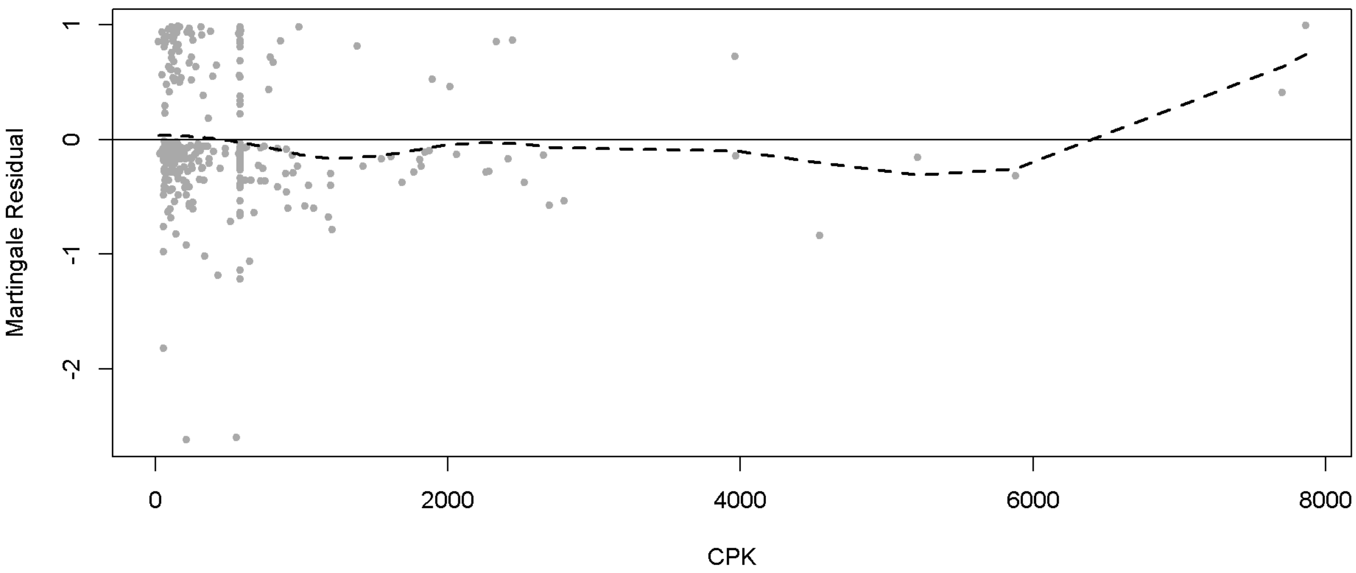
```
# including natural cubic splines raised p-values too much for my liking on other included vars. P-Values already not valid after stepwise reduction so choosing to stratify, instead, by first categorizing it; strata only works on categorical vars


# Checking that the linearity assumption is met for each variable
# age, cpk, ejection fraction, serum creatine, serum sodium
X <- HF$age
Y <- resid(splineMod, type = "martingale")
plot(X, Y, pch = 20, col = "darkgray",
     xlab = "Age", ylab = "Martingale Residual")+
abline(h = 0)+
lines(smooth.spline(X, Y, df = 7), lty = 2, lwd = 2)
```



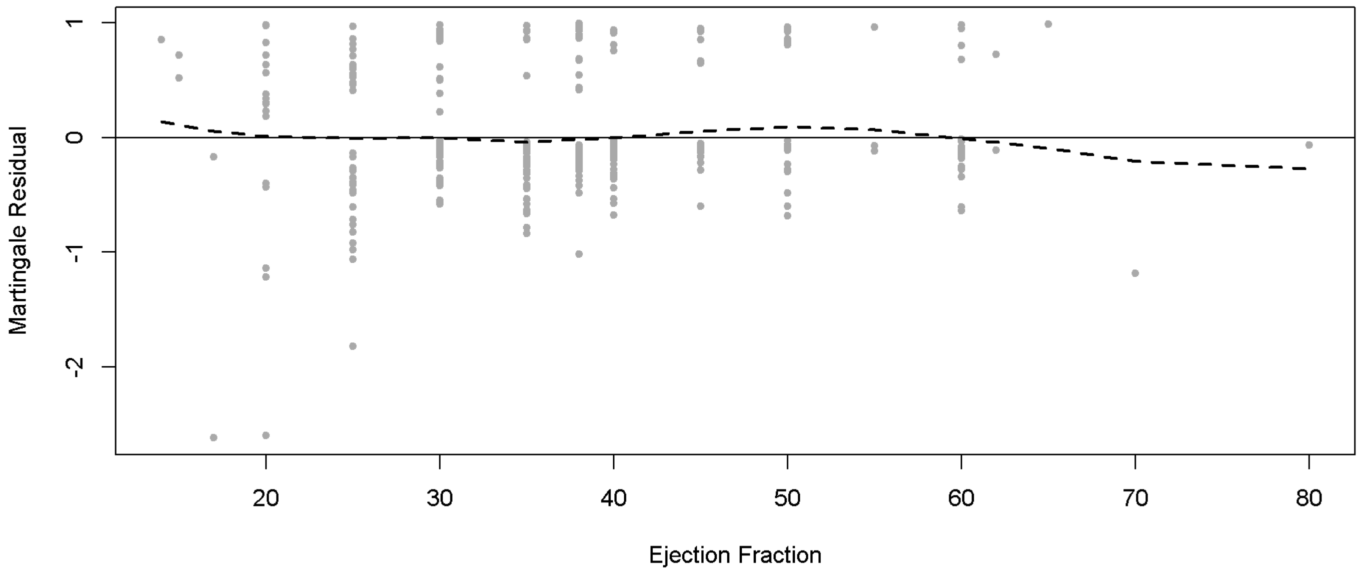
```
## integer(0)
```

```
X <- HF$creatinine_phosphokinase
Y <- resid(splineMod, type = "martingale")
plot(X, Y, pch = 20, col = "darkgray",
     xlab = "CPK", ylab = "Martingale Residual")+
abline(h = 0)+
lines(smooth.spline(X, Y, df = 7), lty = 2, lwd = 2)
```



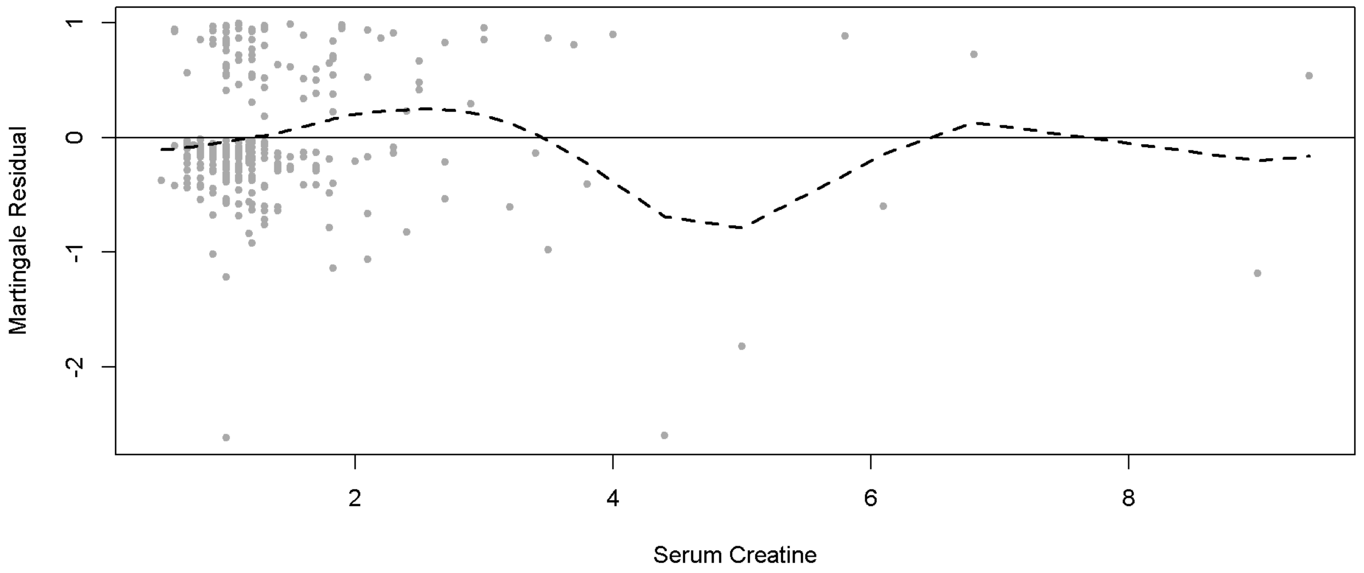
```
## integer(0)
```

```
X <- HF$ejection_fraction
Y <- resid(splineMod, type = "martingale")
plot(X, Y, pch = 20, col = "darkgray",
     xlab = "Ejection Fraction", ylab = "Martingale Residual")+
abline(h = 0)+
lines(smooth.spline(X, Y, df = 7), lty = 2, lwd = 2)
```



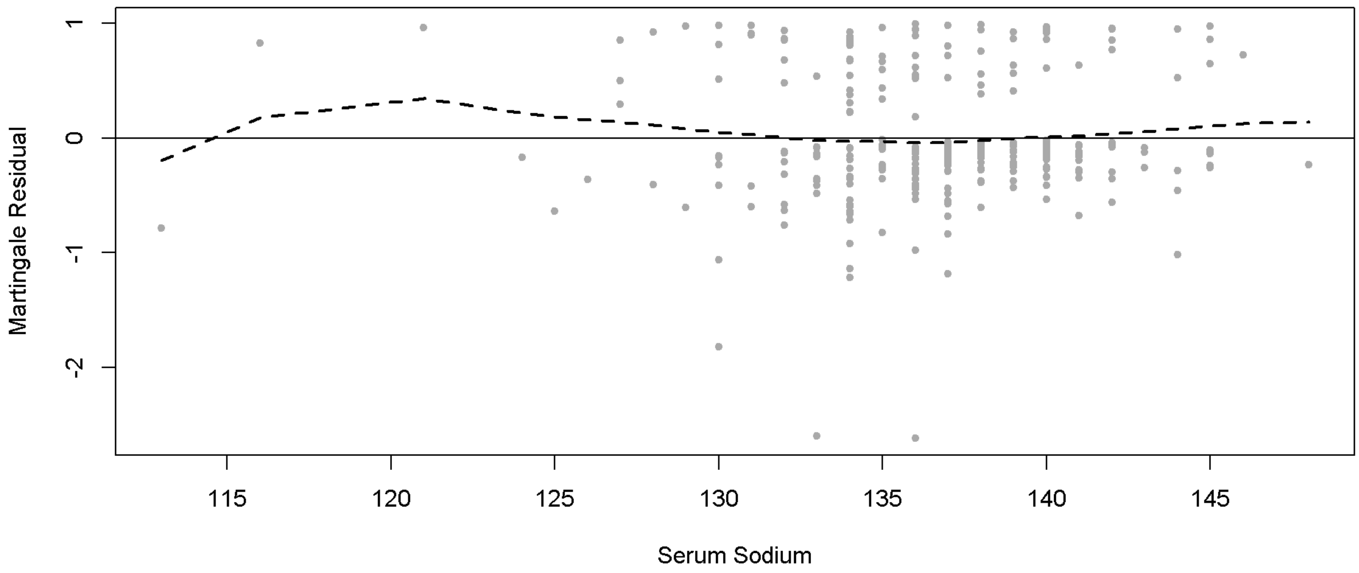
```
## integer(0)
```

```
X <- HF$serum_creatinine
Y <- resid(splineMod, type = "martingale")
plot(X, Y, pch = 20, col = "darkgray",
     xlab = "Serum Creatine", ylab = "Martingale Residual")+
abline(h = 0)+
lines(smooth.spline(X, Y, df = 7), lty = 2, lwd = 2)
```



```
## integer(0)
```

```
X <- HF$serum_sodium
Y <- resid(splineMod, type = "martingale")
plot(X, Y, pch = 20, col = "darkgray",
     xlab = "Serum Sodium", ylab = "Martingale Residual")+
abline(h = 0)+
lines(smooth.spline(X, Y, df = 7), lty = 2, lwd = 2)
```



```
## integer(0)
```

- At a given instance in time, someone who has hypertension is 52% more likely to die as someone without hypertension, adjusting for age.
- At any given instance in time, someone who does *not* have hypertension is 34% less likely (0.66) to die as someone who does, adjusting for age.
- ‘Adjusting for age’ meaning that this is true in a case where two people have the same age.
- Concordance: Goodness of fit for survival analysis.

```
# `hypertension` useful bc tree didn't output it. I paired it w/ age bc why not?
coxMod <- coxph(Surv(time, DEATH_EVENT) ~ hypertension + age, data=HF)
summary(coxMod)
```

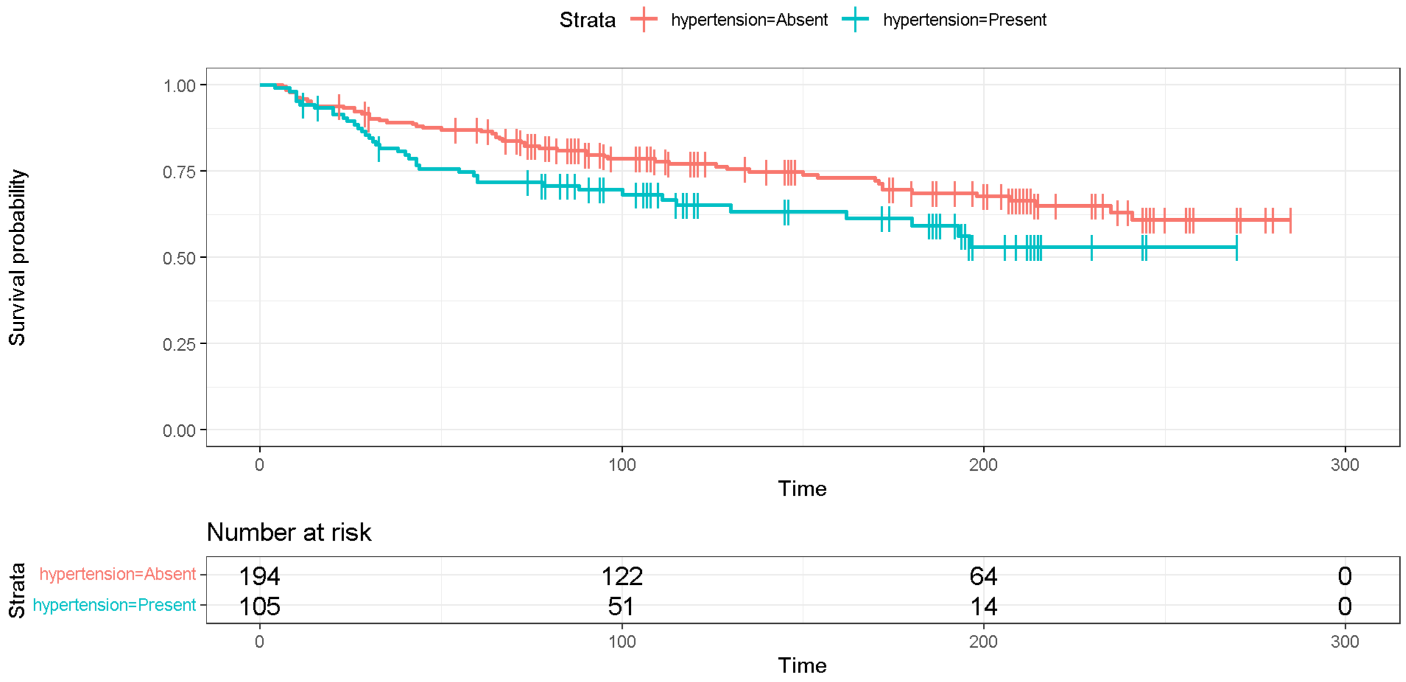
```
## Call:
## coxph(formula = Surv(time, DEATH_EVENT) ~ hypertension + age,
##     data = HF)
##
##   n= 299, number of events= 96
##
##                         coef exp(coef) se(coef)     z Pr(>|z|)
## hypertensionPresent 0.417717  1.518491 0.209708 1.992   0.0464 *
## age                 0.042424  1.043336 0.008693 4.880 1.06e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## hypertensionPresent     1.518     0.6585     1.007     2.290
## age                     1.043     0.9585     1.026     1.061
##
## Concordance= 0.638  (se = 0.031 )
## Likelihood ratio test= 27.36  on 2 df,   p=1e-06
## Wald test            = 27.52  on 2 df,   p=1e-06
## Score (logrank) test = 28.25  on 2 df,   p=7e-07
```

Performing the Log-Rank Test on select, binary variables to extract significance

Doing this to further confirm the elimination or acceptance of features seen previously. Log Rank isn't used for feature selection but if previously unhelpful variables don't have a sense of variance, I can feel assured about not including them

- Patients with `hypertension` do have wide enough difference in their survival rate from those without hypertension to keep the variable as significant.
- Patients with `diabetes` have a similar survival rate curve and thus, survival probability at any point in time, to those without diabetes–leaving the variable insignificant for influence on heart failure.
- Males and Females from the `sex` variable have similar survival curves and lead the variable to not play a significant role when predicting survival times.

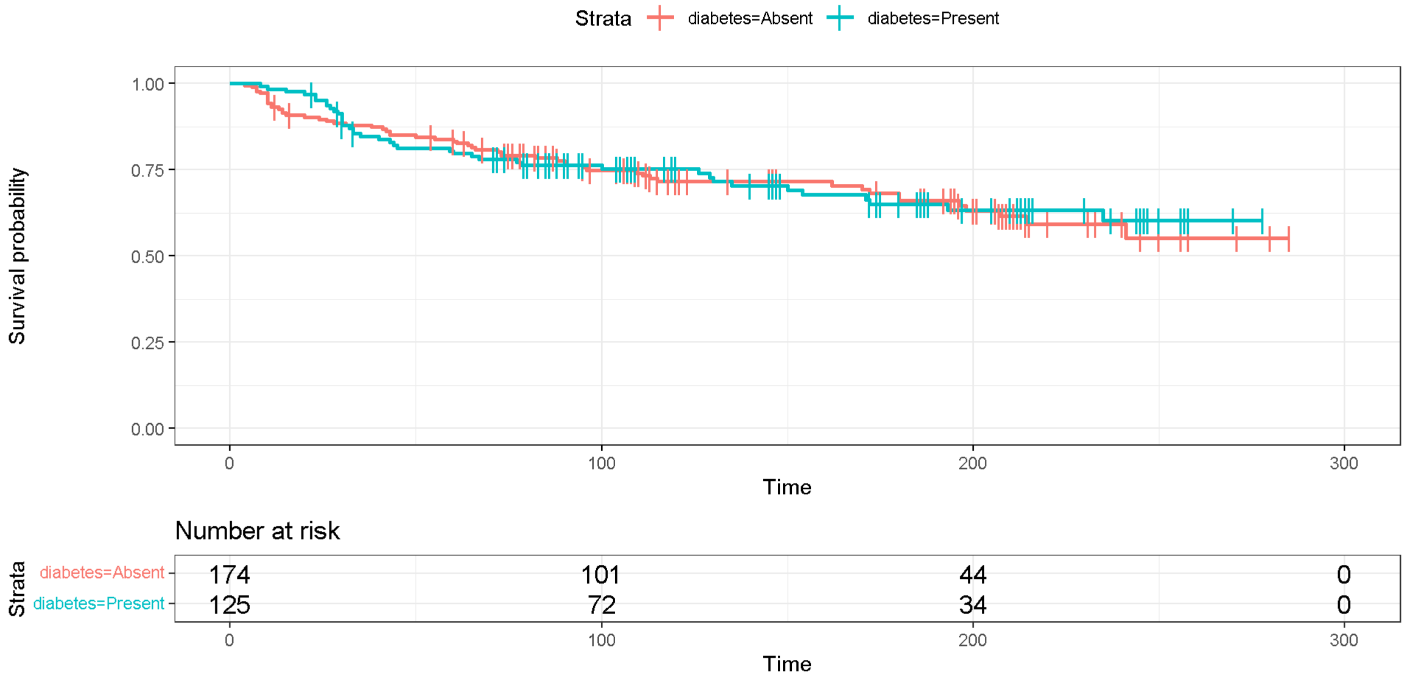
```
#Hypertension
ggsurvplot(survfit(Surv(time,DEATH_EVENT) ~ hypertension, data=HF),
           data = HF,
           censor.shape="|",
           conf.int = FALSE, #surv.median.line = "hv",
           risk.table = TRUE,
           ggtheme = theme_bw())
```



```
survdiff(Surv(time,DEATH_EVENT) ~ hypertension, data=HF)
```

```
## Call:
## survdiff(formula = Surv(time, DEATH_EVENT) ~ hypertension, data = HF)
##
##                          N Observed Expected (O-E)^2/E (O-E)^2/V
## hypertension=Absent  194       57     66.4      1.34      4.41
## hypertension=Present 105       39     29.6      3.00      4.41
##
##  Chisq= 4.4  on 1 degrees of freedom, p= 0.04
```

```
#Diabetes
ggsurvplot(survfit(Surv(time,DEATH_EVENT) ~ diabetes, data=HF),
           data = HF,
           censor.shape="|",
           conf.int = FALSE,
           risk.table = TRUE,
           ggtheme = theme_bw())
```



```
survdiff(Surv(time,DEATH_EVENT) ~ diabetes, data=HF)
```

```
## Call:
## survdiff(formula = Surv(time, DEATH_EVENT) ~ diabetes, data = HF)
##
##                      N Observed Expected (O-E)^2/E (O-E)^2/V
## diabetes=Absent  174       56       55    0.0172    0.0405
## diabetes=Present 125       40       41    0.0231    0.0405
##
##  Chisq= 0  on 1 degrees of freedom, p= 0.8
```

```
#Sex
survdiff(Surv(time,DEATH_EVENT) ~ sex, data=HF)
```

```
## Call:
## survdiff(formula = Surv(time, DEATH_EVENT) ~ sex, data = HF)
## 
##              N Observed Expected (O-E)^2/E (O-E)^2/V
## sex=Female 105       34     34.3   0.00254   0.00397
## sex=Male   194       62     61.7   0.00141   0.00397
## 
##  Chisq= 0  on 1 degrees of freedom, p= 0.9
```

```
#Smoking
survdiff(Surv(time,DEATH_EVENT) ~ smoking, data=HF)
```

```
## Call:
## survdiff(formula = Surv(time, DEATH_EVENT) ~ smoking, data = HF)
## 
##               N Observed Expected (O-E)^2/E (O-E)^2/V
## smoking=No  203       66     65.8   0.00064   0.00204
## smoking=Yes  96       30     30.2   0.00139   0.00204
## 
##  Chisq= 0  on 1 degrees of freedom, p= 1
```

```
#Anemia
survdiff(Surv(time,DEATH_EVENT) ~ anaemia, data=HF)
```

```
## Call:
## survdiff(formula = Surv(time, DEATH_EVENT) ~ anaemia, data = HF)
## 
##             N Observed Expected (O-E)^2/E (O-E)^2/V
## anaemia=0 170       50     57.9      1.07      2.73
## anaemia=1 129       46     38.1      1.63      2.73
## 
##  Chisq= 2.7  on 1 degrees of freedom, p= 0.1
```

```
#Platelets
#Dichotomizing Platelets by the median
plat <- HF %>% select(time, DEATH_EVENT, platelets) %>%
  mutate(platelets_binary = ifelse(platelets > median(platelets), "OverMedian", "UnderMedian"))

survdiff(Surv(time,DEATH_EVENT) ~ platelets_binary, data=plat)
```

```
## Call:
## survdiff(formula = Surv(time, DEATH_EVENT) ~ platelets_binary, 
##     data = plat)
## 
##                                 N Observed Expected (O-E)^2/E (O-E)^2/V
## platelets_binary=OverMedian  149       47     47.5   0.00459   0.00912
## platelets_binary=UnderMedian 150       49     48.5   0.00449   0.00912
## 
##  Chisq= 0  on 1 degrees of freedom, p= 0.9
```

Binary Logistic Regression

- Performing a classification method for a hypothetical scenario in which all patients were followed-up after the same length of time rather than varying times.

Using sources One (https://labs.selfdecode.com/blog/creatine-kinase/#:~:text=The%20low%20normal%20limit%20for,3%2C%204%2C%205%5D./) and Two (https://www.mayoclinic.org/tests-procedures/creatinine-test/about/pac-20384646#:~:text=The%20typical%20range%20for%20serum,52.2%20to%2091.9%20micromoles%2FL).

- Adjusting logistic regression probability selection; lowering it.
- The choice of a lower prob. selection is to advise even those not predicted for heart failure, to take adequate rest.

```
set.seed(0)
library(caTools)

split_log <- sample.split(HF$DEATH_EVENT, SplitRatio = 0.75)
train_log <- subset(HF, split_log == TRUE) %>% select(-time)
test_log <- subset(HF, split_log == FALSE)


# Creating a function to remove outliers
is_outlier <- function(x){
  condition <- quantile(x, 0.75, na.rm = TRUE) + 1.5 * IQR(x,na.rm = TRUE)
  output <- ifelse(x >= condition, TRUE, FALSE)
  return(output)
}

# placing 'i' in front of all values that are outliers so as to keep only non-outlier values.
train_no_outliers <- train_log %>%
  filter(!( is_outlier(serum_creatinine) | is_outlier(creatinine_phosphokinase)) )
```

- Creating category variables for `Serum Creatinine` &
- `Creatinine Phosphokinase` due to their heavy right skewness.

```
train_log <- train_log %>%
  mutate(SC_Condition = cut(train_log$serum_creatinine, breaks = c(0, 0.7, 1.25, Inf),
          labels = c("Low", "Normal", "High"), include.lowest = TRUE),
         CPK_Condition = cut(train_log$creatinine_phosphokinase, breaks = c(0, 30,200, 300, Inf),
          labels = c("Low", "Normal", "High", "Severely High"), include.lowest = TRUE)) %>%
  select(-serum_creatinine, -creatinine_phosphokinase)


train_no_outliers <- train_no_outliers %>%
  mutate(SC_Condition = cut(train_no_outliers$serum_creatinine, breaks = c(0, 0.7, 1.25, Inf),
          labels = c("Low", "Normal", "High"), include.lowest = TRUE),
         CPK_Condition = cut(train_no_outliers$creatinine_phosphokinase, breaks = c(0, 30,200, 300, Inf),
          labels = c("Low", "Normal", "High", "Severely High"), include.lowest = TRUE)) %>%
  select(-serum_creatinine, -creatinine_phosphokinase)


test_log <- test_log %>%
  mutate(SC_Condition = cut(test_log$serum_creatinine, breaks = c(0, 0.7, 1.25, Inf),
          labels = c("Low", "Normal", "High"), include.lowest = TRUE),
         CPK_Condition = cut(test_log$creatinine_phosphokinase, breaks = c(0, 30,200, 300, Inf),
          labels = c("Low", "Normal", "High", "Severely High"), include.lowest = TRUE)) %>%
  select(-serum_creatinine, -creatinine_phosphokinase)
```

- Adjusting logistic regression probability selection; lowering it.
- The choice of a lower prob. selection is to advise even those not predicted for heart failure, to take adequate rest.

```
logit1 <- glm(DEATH_EVENT~., family = binomial,data = train_log)
summary(logit1)$coefficients[,4] %>% round(digits = 5)
```

```
##                (Intercept)                        age
##                    0.54569                    0.00101
##                   anaemia1            diabetesPresent
##                    0.46862                    0.33690
##          ejection_fraction                  platelets
##                    0.00031                    0.39021
##               serum_sodium                    sexMale
##                    0.44889                    0.13125
##                 smokingYes        hypertensionPresent
##                    0.59266                    0.20803
##         SC_ConditionNormal           SC_ConditionHigh
##                    0.59445                    0.03500
##        CPK_ConditionNormal          CPK_ConditionHigh
##                    0.52806                    0.37918
## CPK_ConditionSeverely High
##                    0.42043
```

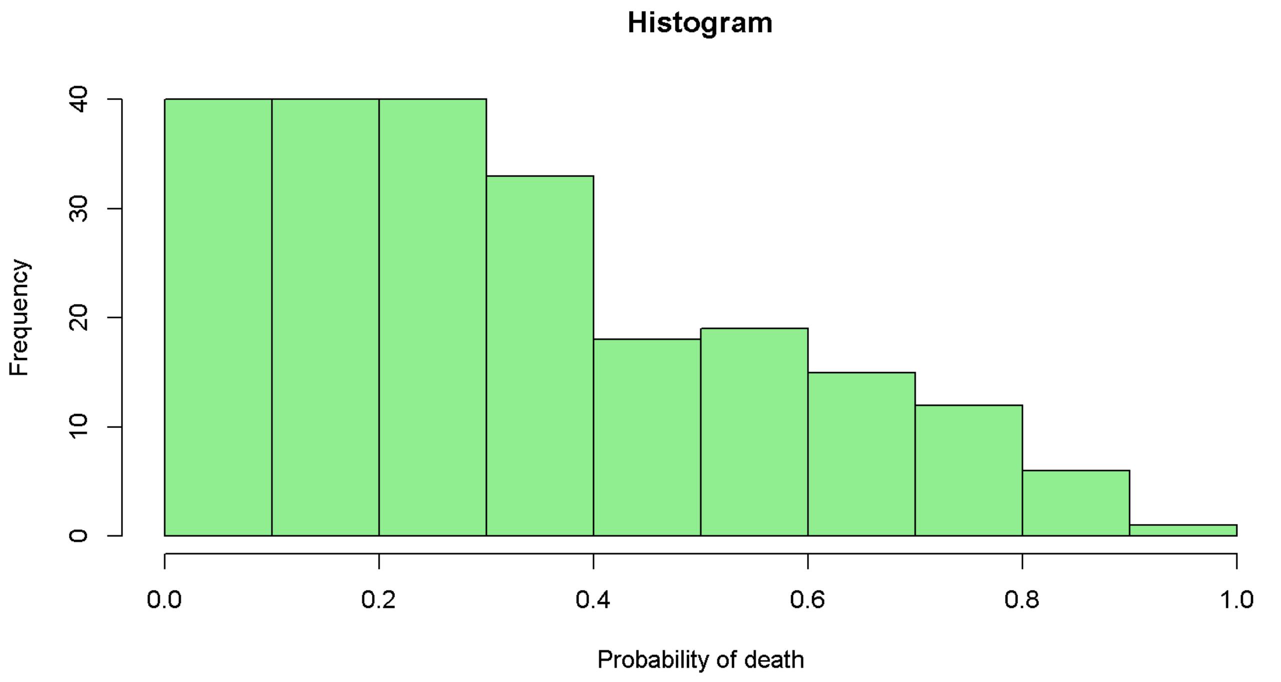
```
summary(logit1)$aic
```

```
## [1] 249.3829
```

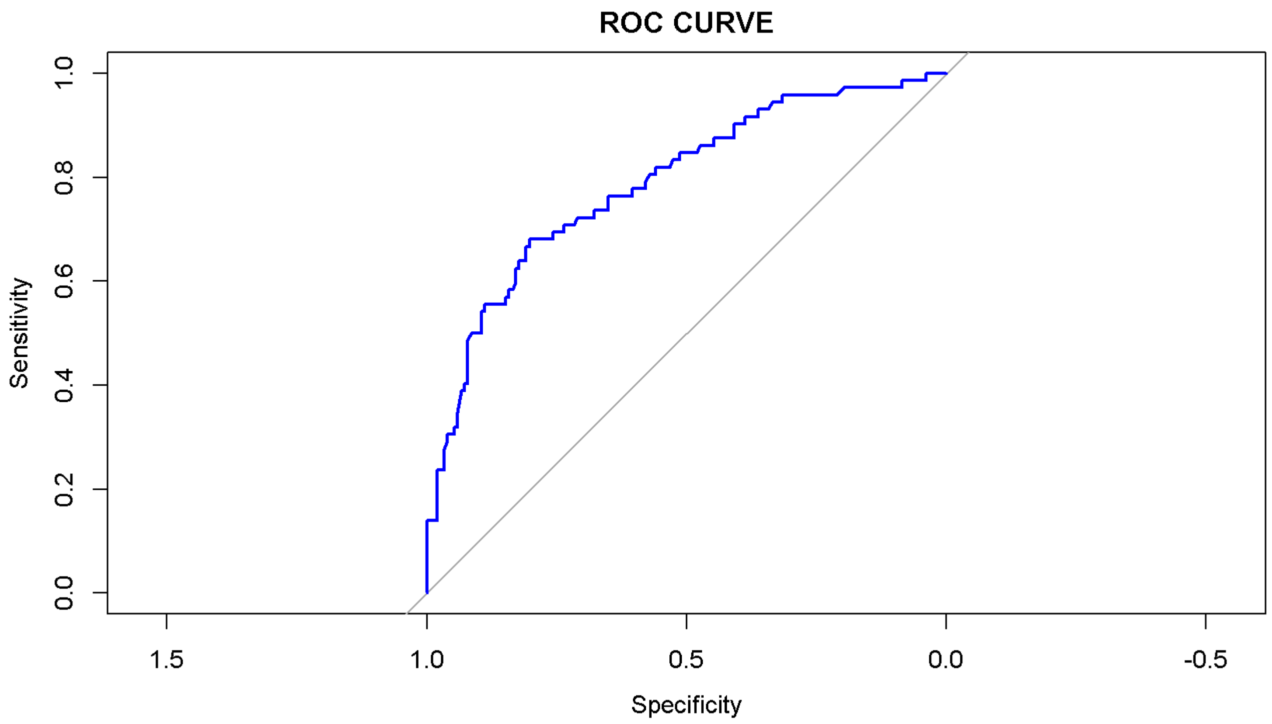
```
logit2 <- step(logit1, direction = "backward", trace = FALSE)
summary(logit2)$coefficients[,4] %>% round(digits = 5)
```

```
##         (Intercept)                 age   ejection_fraction hypertensionPresent
##             0.00884             0.00037             0.00054             0.09503
##  SC_ConditionNormal    SC_ConditionHigh
##             0.68178             0.03917
```

```
hist(logit2$fitted.values, main=" Histogram ",xlab="Probability of death", col='light green')
```



```
r <- pROC::roc(DEATH_EVENT~logit2$fitted.values, data = train_log, plot = TRUE, main = "ROC CURVE", col= "blue")
```



```
optimal_roc <- r$thresholds[which.max(r$sensitivities + r$specificities)] # 0.37


test_log <- test_log %>%
      mutate(p1=predict(logit2, newdata=test_log, type="response")) %>%
              mutate(Predict=ifelse(p1 < optimal_roc,0,1))


cm <- table(test_log$DEATH_EVENT,test_log$Predict) %>% prop.table()
rownames(cm) <- c("Obs. neg","Obs. pos")
colnames(cm) <- c("Pred. neg","Pred. pos")

ERROR.RESULTS <- tibble(
  Sensitivity=c(cm[1,1]/sum(cm[1,])),
  Specificity=c(cm[2,2]/sum(cm[2,])),
  FalsePositives=c(cm[2,1]/sum(cm[2,])),
  FalseNegatives=c(cm[1,2]/sum(cm[1,]))
)

efficiency <- sum(diag(cm))/sum(cm)

ERROR.RESULTS
```

```
## # A tibble: 1 × 4
##   Sensitivity Specificity FalsePositives FalseNegatives
##         <dbl>       <dbl>          <dbl>          <dbl>
## 1       0.863       0.458          0.542          0.137
```

```
efficiency
```

```
## [1] 0.7333333
```

Performing the binary logistic regression once more with outliers removed from the training set.

- RESULTS: The removal of outliers dramatically increased specificity

```
logit1 <- glm(DEATH_EVENT~., family = binomial,data = train_no_outliers)
summary(logit1)$coefficients[,4] %>% round(digits = 5)
```

```
##                (Intercept)                        age 
##                    0.99138                    0.00171 
##                   anaemia1            diabetesPresent 
##                    0.26175                    0.44394 
##          ejection_fraction                  platelets 
##                    0.00062                    0.22419 
##               serum_sodium                    sexMale 
##                    0.58998                    0.08291 
##                 smokingYes        hypertensionPresent 
##                    0.13634                    0.52459 
##         SC_ConditionNormal           SC_ConditionHigh 
##                    0.67849                    0.04772 
##        CPK_ConditionNormal          CPK_ConditionHigh 
##                    0.98987                    0.99084 
## CPK_ConditionSeverely High 
##                    0.99036
```

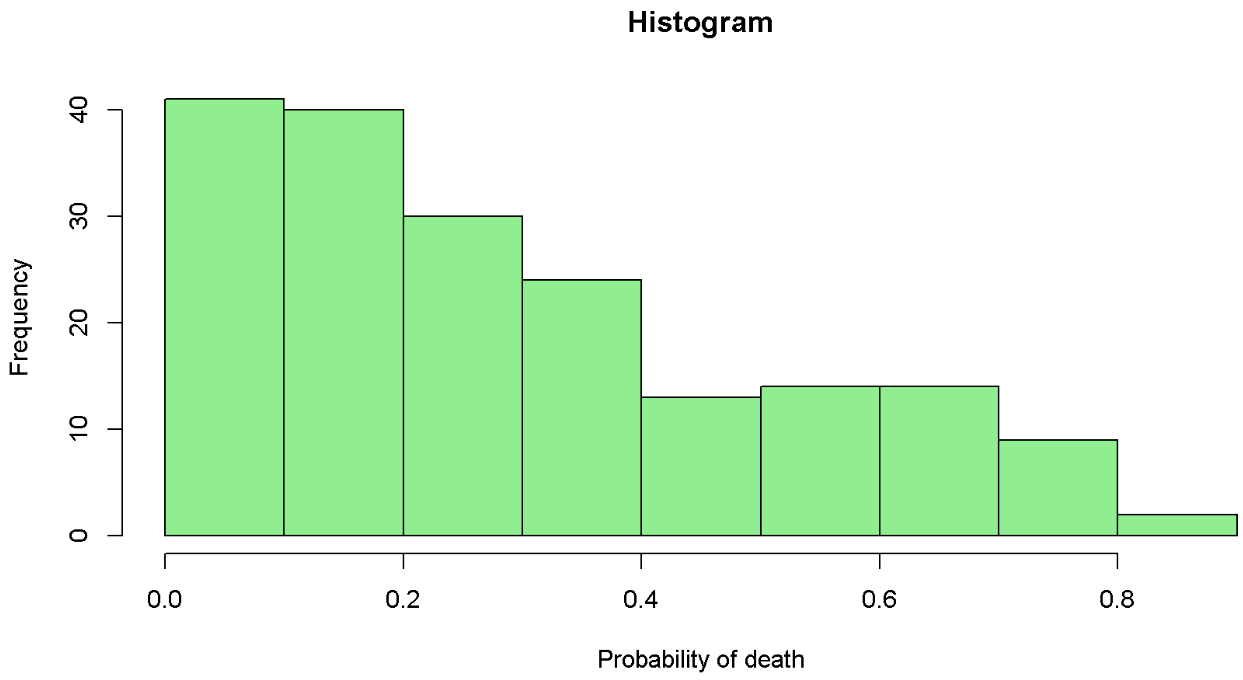
```
summary(logit1)$aic
```

```
## [1] 203.417
```

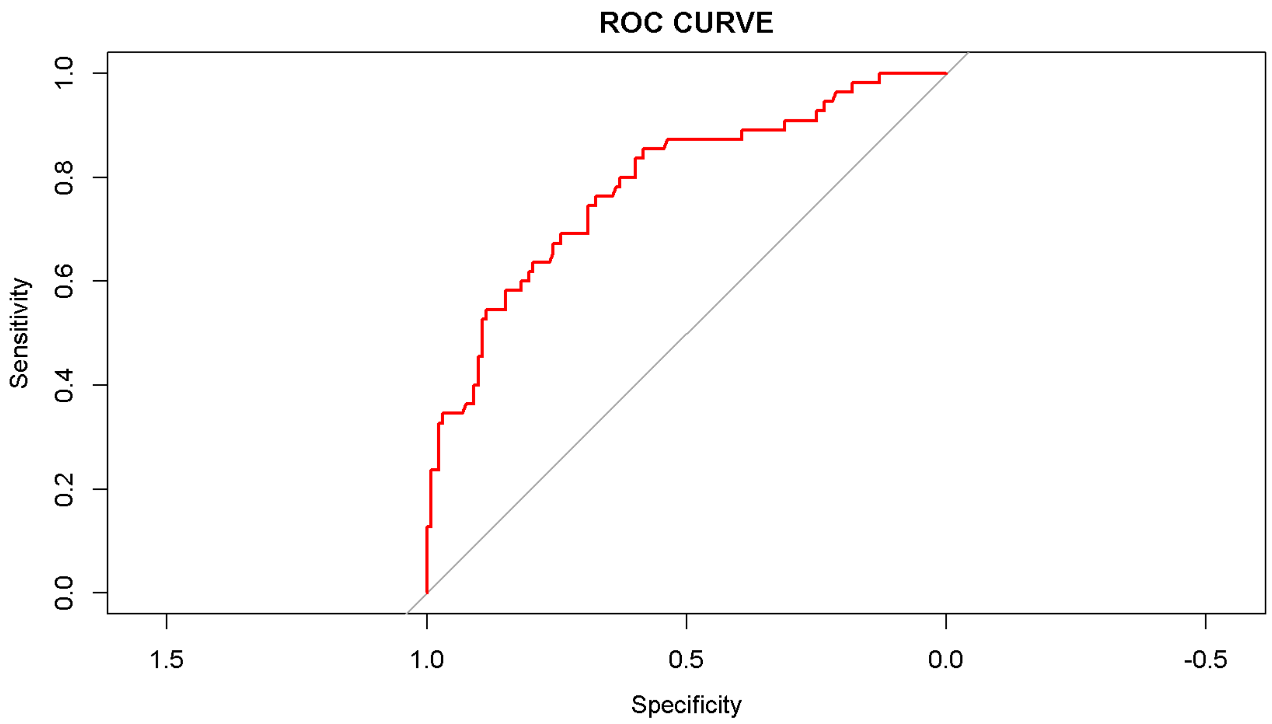
```
logit2 <- step(logit1, direction = "backward", trace = FALSE)
summary(logit2)$coefficients[,4] %>% round(digits = 5)
```

```
##        (Intercept)                age  ejection_fraction            sexMale 
##            0.06270            0.00104            0.00078            0.15303 
## SC_ConditionNormal   SC_ConditionHigh 
##            0.81008            0.07298
```

```
hist(logit2$fitted.values, main=" Histogram ",xlab="Probability of death", col='light green')
```



```
r <- pROC::roc(DEATH_EVENT~logit2$fitted.values, data = train_no_outliers, plot = TRUE, main = "ROC CURVE", col= "red")
```



```
optimal_roc <- r$thresholds[which.max(r$sensitivities + r$specificities)] # 0.21


test_log <- test_log %>%
      mutate(p2=predict(logit2, newdata=test_log, type="response")) %>%
              mutate(Predict2=ifelse(p2 < optimal_roc,0,1))


cm <- table(test_log$DEATH_EVENT,test_log$Predict2) %>% prop.table()
rownames(cm) <- c("Obs. neg","Obs. pos")
colnames(cm) <- c("Pred. neg","Pred. pos")

ERROR.RESULTS <- tibble(
  Sensitivity=c(cm[1,1]/sum(cm[1,])),
  Specificity=c(cm[2,2]/sum(cm[2,])),
  FalsePositives=c(cm[2,1]/sum(cm[2,])),
  FalseNegatives=c(cm[1,2]/sum(cm[1,]))
)

efficiency <- sum(diag(cm))/sum(cm)

ERROR.RESULTS
```

```
## # A tibble: 1 × 4
##   Sensitivity Specificity FalsePositives FalseNegatives
##         <dbl>       <dbl>          <dbl>          <dbl>
## 1       0.667       0.625          0.375          0.333
```

```
efficiency
```

```
## [1] 0.6533333
```

End