Code

Heart Failure Analysis and Prediction

About this dataset

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 worlwide.Heart failure is a common event caused by CVDs and this dataset contains 12 features that can be used to predict mortality by heart failure.Most cardiovascular diseases can be prevented by addressing behavioural risk factors such as tobacco use, unhealthy diet and obesity, physical inactivity and harmful use of alcohol using population-wide strategies.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, hyperlipidaemia or already established disease) need early detection and management wherein a machine learning model can be of great help.

Importing libraries

Hide

```
library(ggplot2)
library(dplyr);
library(rpart)
library(rpart.plot)
library(caret);
library(reshape2)
library(DataExplorer)
library(gridExtra)
```

Impontring the data

We will import `heart_failure_clinical_records_dataset.csv`

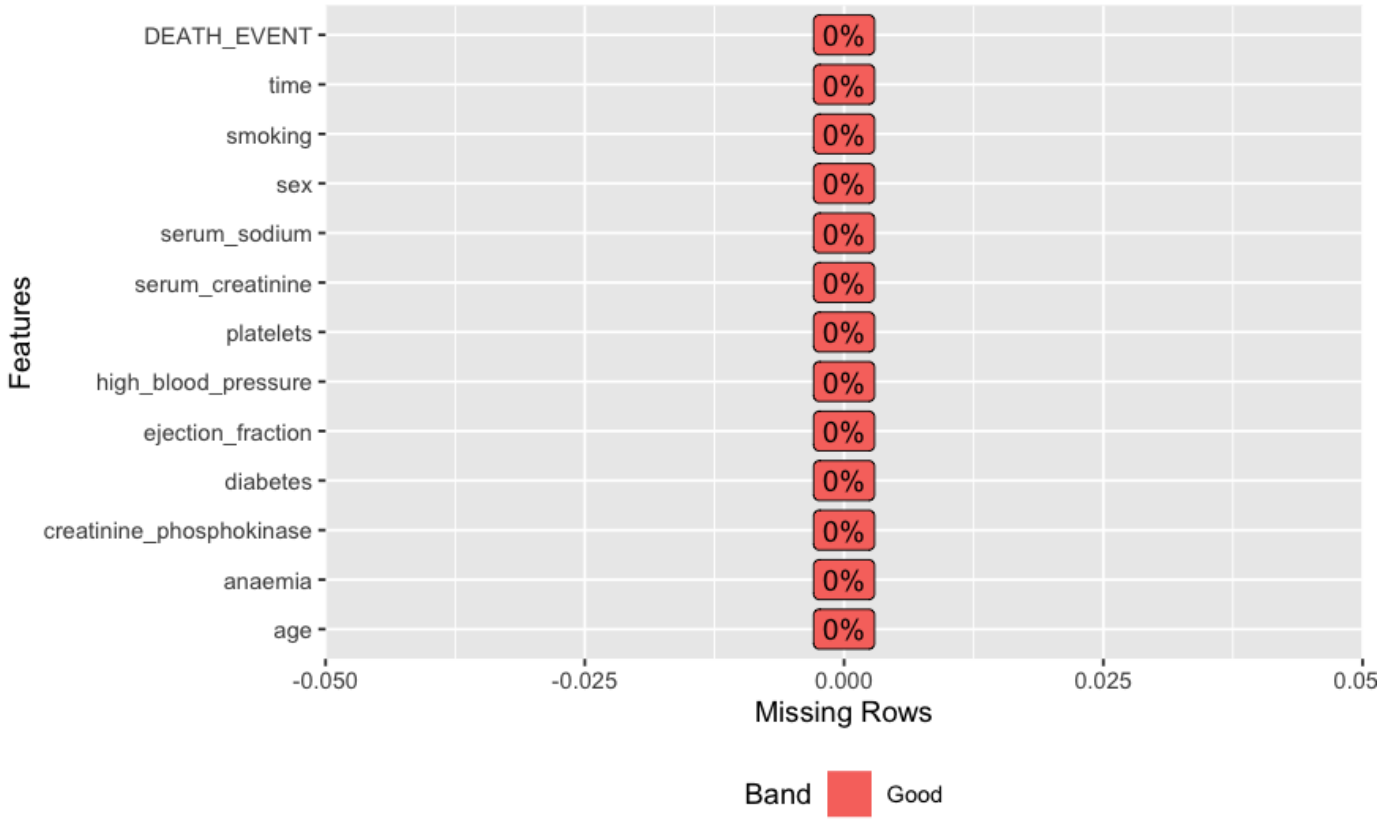
Hide

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

First, let's check nulls in the dataset

Hide

```
plot_missing(data)
```



Get a look at the dataset

Hide

```
head(data, 10)
```

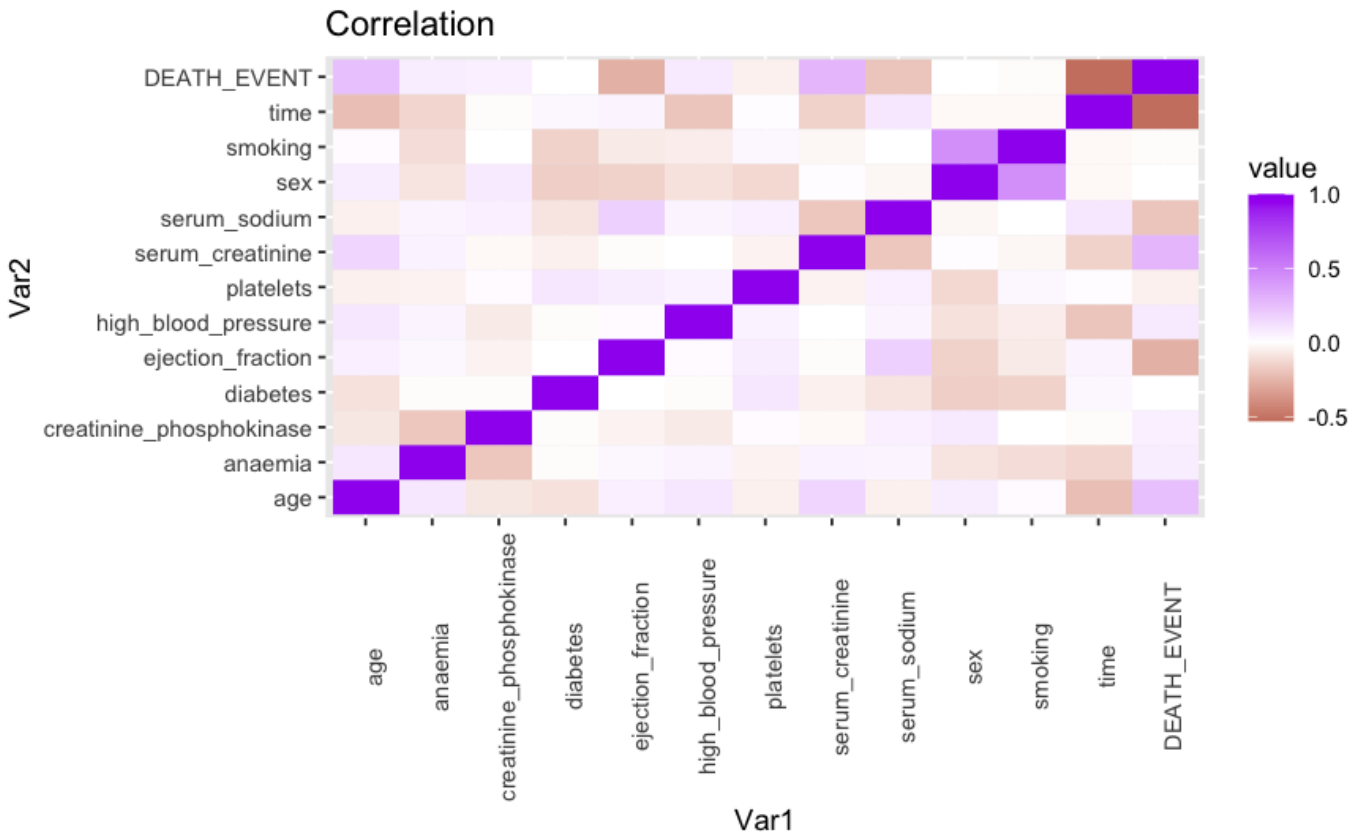
	...	anae...	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pres
	<dbl>	<int>	<int>	<int>	<int>	
1	75	0	582	0	20	
2	55	0	7861	0	38	
3	65	0	146	0	20	
4	50	1	111	0	20	
5	65	1	160	1	20	
6	90	1	47	0	40	
7	75	1	246	0	15	
8	60	1	315	1	60	
9	65	0	157	0	65	
10	80	1	123	0	35	

1-10 of 10 rows | 1-8 of 13 columns

Fining correlations between features

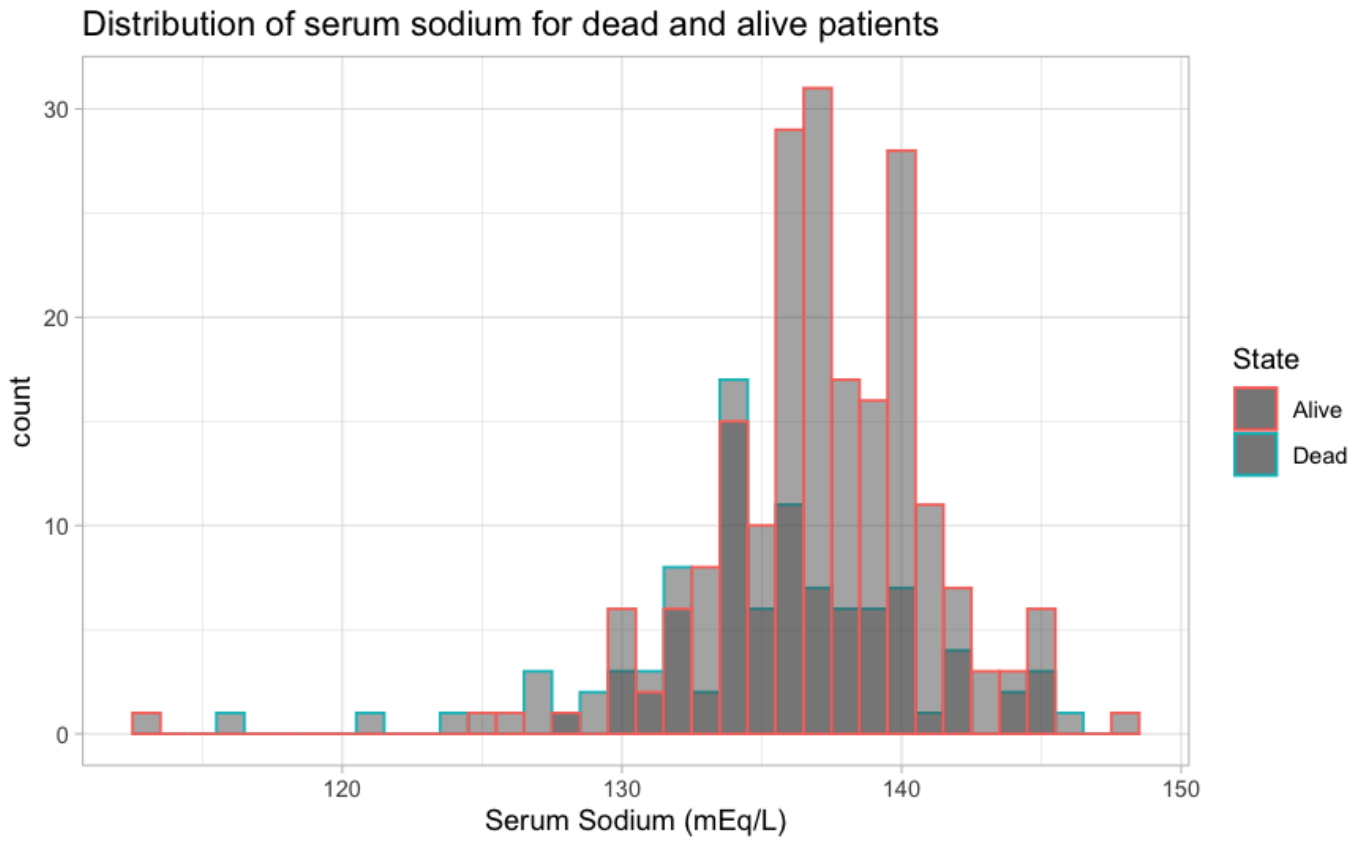
Hide

```
# calculating correlations and rounding to nearest 2 decimal points
cormap <- round( cor(data), 2)

#convert the matrix to a dataframe
cormap_melted<-melt(cormap)


#creating heatmap
ggplot(data = cormap_melted, aes(x=Var1, y=Var2, fill=value) ) +
  geom_tile() + scale_fill_gradient2(low="darkred",high="purple",mid="white") + theme
(axis.text.x = element_text(angle = 90)) + labs(title = 'Correlation')
```



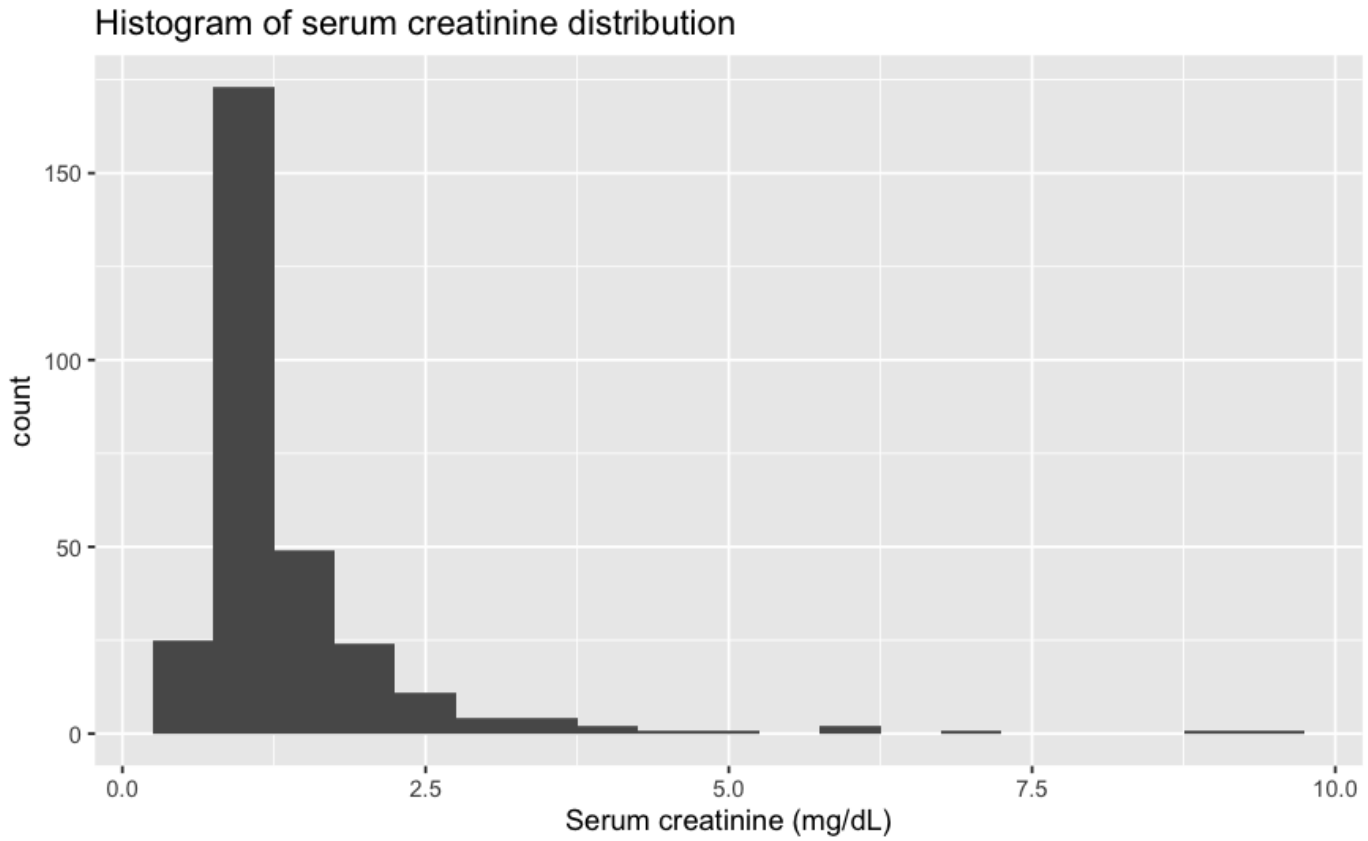
Hide

```
dead_sodium <- filter(data, DEATH_EVENT==1) %>% select(serum_sodium)
alive_sodium <- filter(data, DEATH_EVENT==0) %>% select(serum_sodium)

ggplot() +
  geom_histogram(data = dead_sodium, aes(serum_sodium, color='Dead'), alpha=0.5, binw
idth = 1) +
  geom_histogram(data = alive_sodium,aes(serum_sodium, color='Alive'), alpha = 0.5, b
inwidth = 1) +
  theme_light() + labs(color='State', title='Distribution of serum sodium for dead an
d alive patients', x='Serum Sodium (mEq/L)')
```



Hide

```
ggplot(data, aes(x = serum_creatinine)) + geom_histogram(binwidth = 0.5) + labs(title ='Histogram of serum creatinine distribution', x='Serum creatinine (mg/dL)')
```



The distribution of `serum_sodium` is left skewed, while `serum_creatinine` is right skewed.

Hide

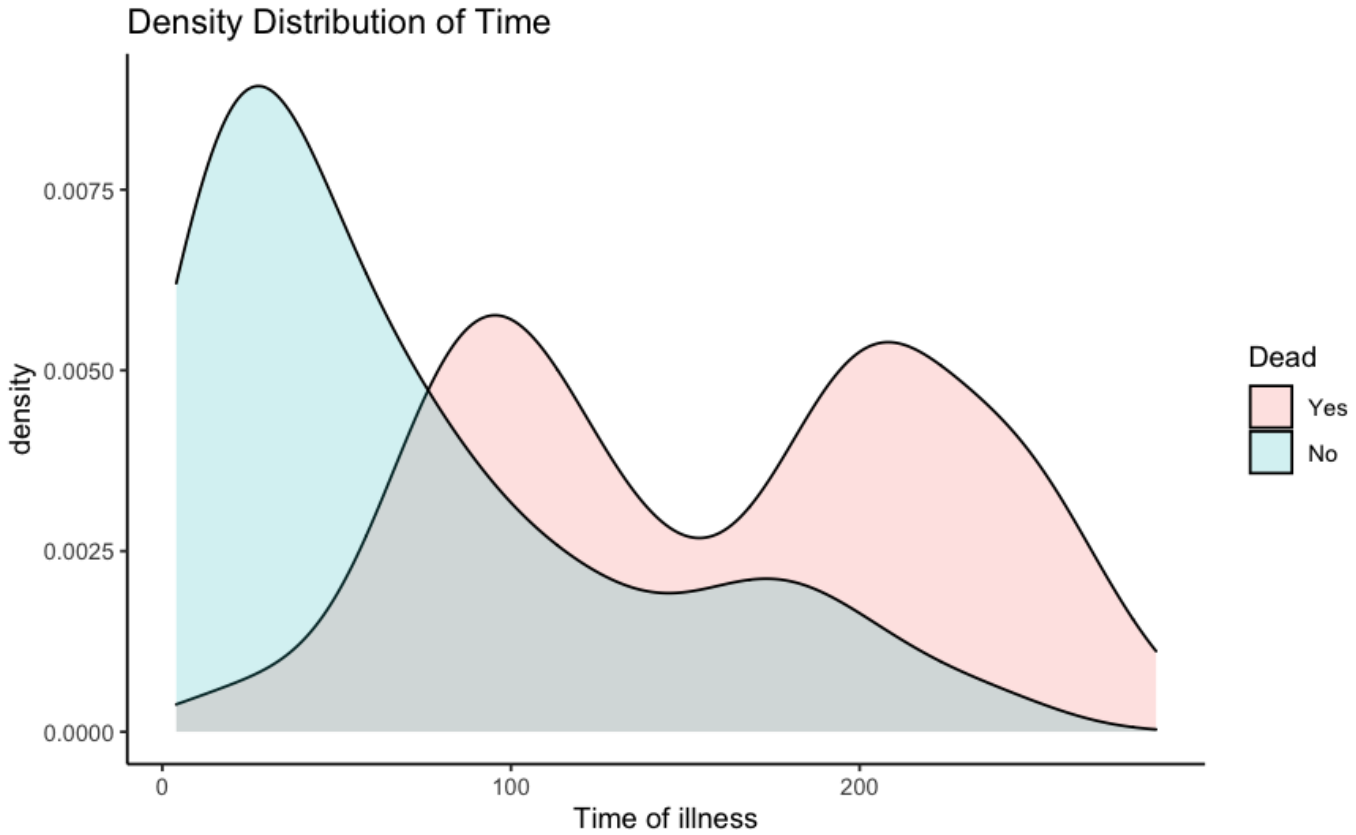
```
ggplot(data, aes(x=factor(sex), y=age, color=factor(sex))) + geom_boxplot(outlier.col
or = 'black', outlier.shape = 8) + geom_jitter(shape=16, position=position_jitter(0.2
)) + labs(x='sex', title='Distribution of gender and their ages', color='sex') + scal
e_color_discrete(labels=c('Female', 'Male'))
```

Next, lets investigate some relations

The duration of illness of dead patient and recovered patient

Hide

```
#Visualize the density distribution function of duration of illness and death event
 ggplot(data, aes(x = time, fill = as.factor(DEATH_EVENT) )) +
        geom_density(alpha = 0.2) + theme_classic() +
        labs(title = "Density Distribution of Time", fill="Dead", x='Time of illness'
) + scale_fill_discrete(labels=c("Yes", "No"))
```

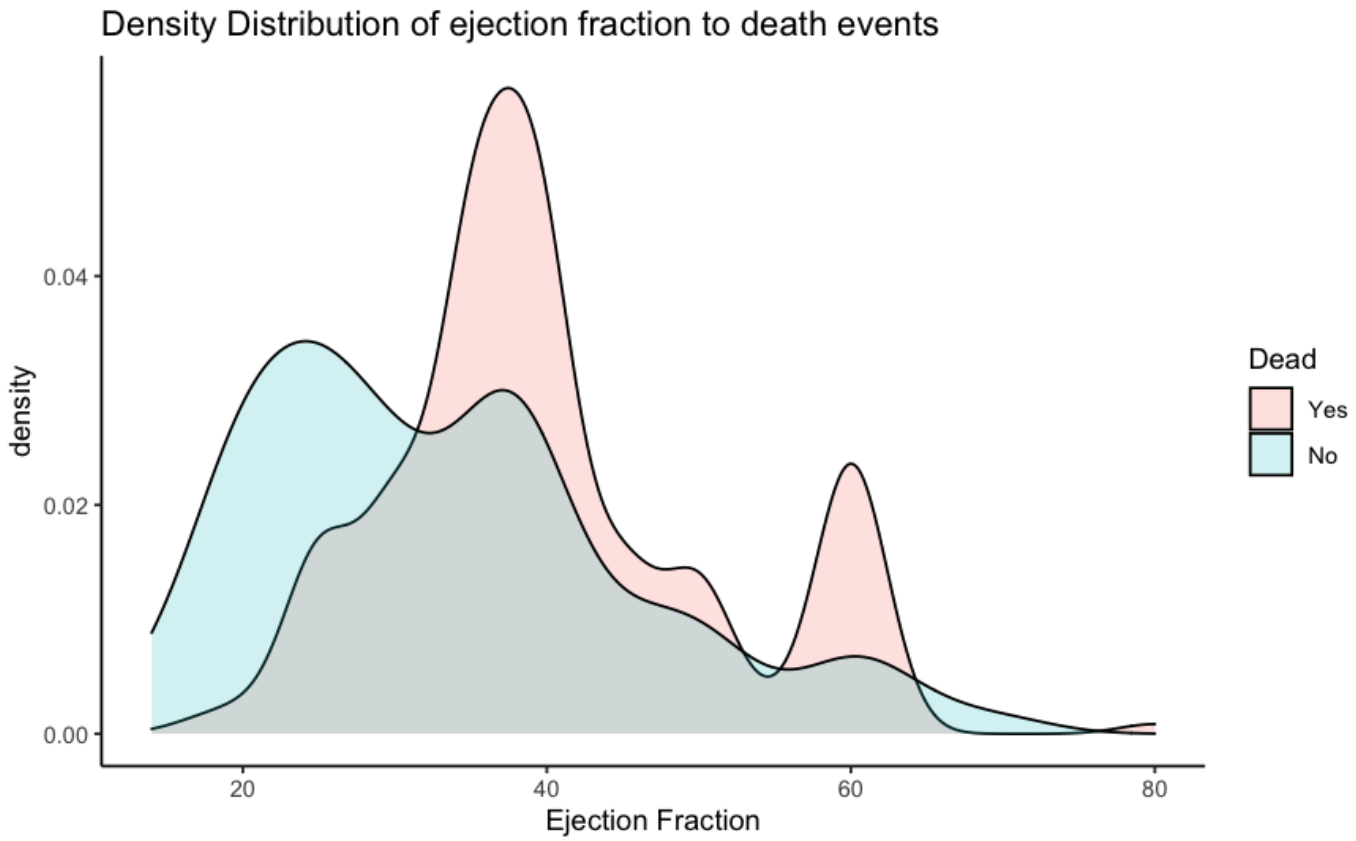


At first, patient have higher probablility of recovery, while passing 100+ days incease the probablility of death.

Let's see ejection fraction affects the occurance of heart failure.

Hide

```
#Visualize the density distribution function of smoking and the death event
 ggplot(data,aes(x = ejection_fraction, fill = as.factor(DEATH_EVENT))) +
        geom_density(alpha = 0.2) + theme_classic() +
        labs(title = "Density Distribution of ejection fraction to death events", fil
l='Dead', x='Ejection Fraction')+ scale_fill_discrete(labels=c("Yes", "No"))
```



Dead patients have higher probablility of having ejection fraction.

Predict using Logistic Regression

Hide

```
set.seed(123) #setting the seed to make sure that the split function give us the same
results

taining.samples <- data$DEATH_EVENT %>%
  createDataPartition(p=0.5, list=FALSE) # splitting the data 50-50, to avoid over fi
tting

# store the test data and train data
train.data <- data[taining.samples, ] #first partition of sample
test.data <- data[- taining.samples, ] #last partition of sample

# run the logistic regression model using all of the features we have using binomial
 classification
model <- glm(DEATH_EVENT~., data = train.data, family=binomial())

# inspect the coefficients resulted from the logistic regression
summary(model)$coef
```

```
                            Estimate   Std. Error    z value     Pr(>|z|)
(Intercept)                4.997730e+00 9.422564e+00  0.5304002 5.958345e-01
age                        4.585887e-02 2.462071e-02  1.8626135 6.251667e-02
anaemia                    3.921844e-01 5.633341e-01  0.6961843 4.863134e-01
creatinine_phosphokinase -4.944054e-05 3.384572e-04 -0.1460762 8.838612e-01
diabetes                   7.300236e-01 5.704696e-01  1.2796889 2.006546e-01
ejection_fraction        -9.656250e-02 2.568859e-02 -3.7589649 1.706178e-04
high_blood_pressure        3.043644e-01 5.365284e-01  0.5672849 5.705207e-01
platelets                -2.082016e-06 2.489475e-06 -0.8363275 4.029707e-01
serum_creatinine           9.266276e-01 2.658751e-01  3.4851988 4.917715e-04
serum_sodium             -2.615251e-02 6.628936e-02 -0.3945205 6.931968e-01
sex                        1.069458e-01 6.458398e-01  0.1655919 8.684781e-01
smoking                  -1.866607e-01 6.101210e-01 -0.3059405 7.596499e-01
time                     -2.710215e-02 5.310282e-03 -5.1037120 3.330550e-07
```

Hide

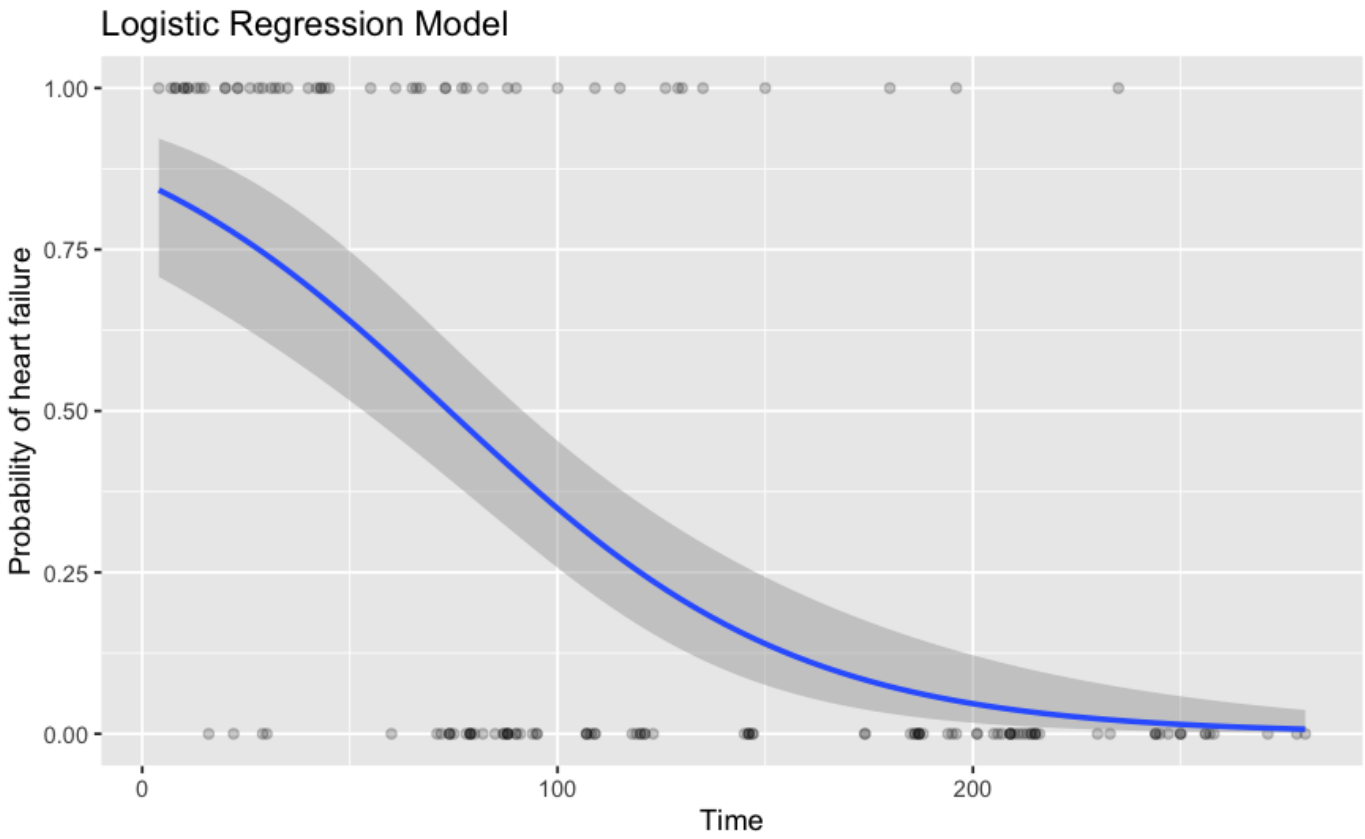
```
# testing the model
probabilities <- model %>% predict(test.data, type = "response")

# thresholding results; results from the logistic function >= 0.5 lies in the 1 clas
s, while < 0.5 lies in the opposite one
predicted.classes <- ifelse(probabilities >= 0.5, "1", "0")

# Model accuracy
mean(predicted.classes == test.data$DEATH_EVENT) * 100
```

```
[1] 81.20805
```

plotting logistic function with `time` parameter

Hide

```
train.data %>%
  mutate(prob = ifelse(DEATH_EVENT == "1", 1, 0)) %>%
  ggplot(aes(time, prob)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "glm", method.args = list(family = "binomial")) +
  labs(
    title = "Logistic Regression Model",
    x = "Time",
    y = "Probability of heart failure")
```



Creating the descision tree

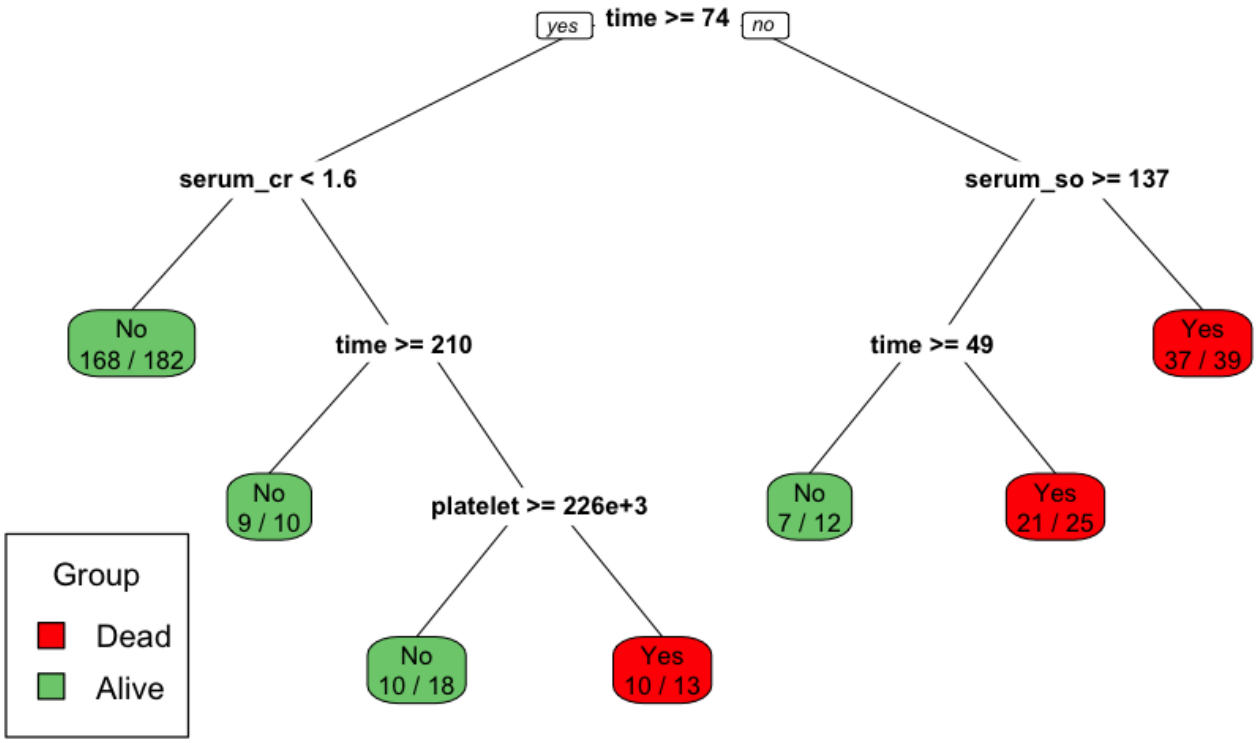
Hide

```
set.seed(123)
data$Dead<-ifelse(data$DEATH_EVENT!=1,"No","Yes")

#This variable is no needed for constructing a classification tree
data$DEATH_EVENT<-NULL

#creating tree, I will use a small critical point
heartTree<-rpart(Dead~.,data=data,control=rpart.control(cp=0.00001))


#Creating a matrix to check the accuracy of decision tree
conf.matrix <- table(data$Dead, predict(heartTree, type="class"))

rownames(conf.matrix) <- paste("Actual", rownames(conf.matrix), sep = ":")

colnames(conf.matrix) <- paste("Pred", colnames(conf.matrix), sep = ":")

print(conf.matrix)
```

```
            Pred:No Pred:Yes
  Actual:No      194         9
  Actual:Yes      28        68
```

Hide

```
boxcols <- c("palegreen3","red")[heartTree$frame$yval]

par(xpd=TRUE)
prp(heartTree, faclen = 0, cex = 0.8, box.col = boxcols,extra=2)
legend("bottomleft", legend = c("Dead","Alive"), fill = c("red", "palegreen3"),
       title = "Group")
```



Hide

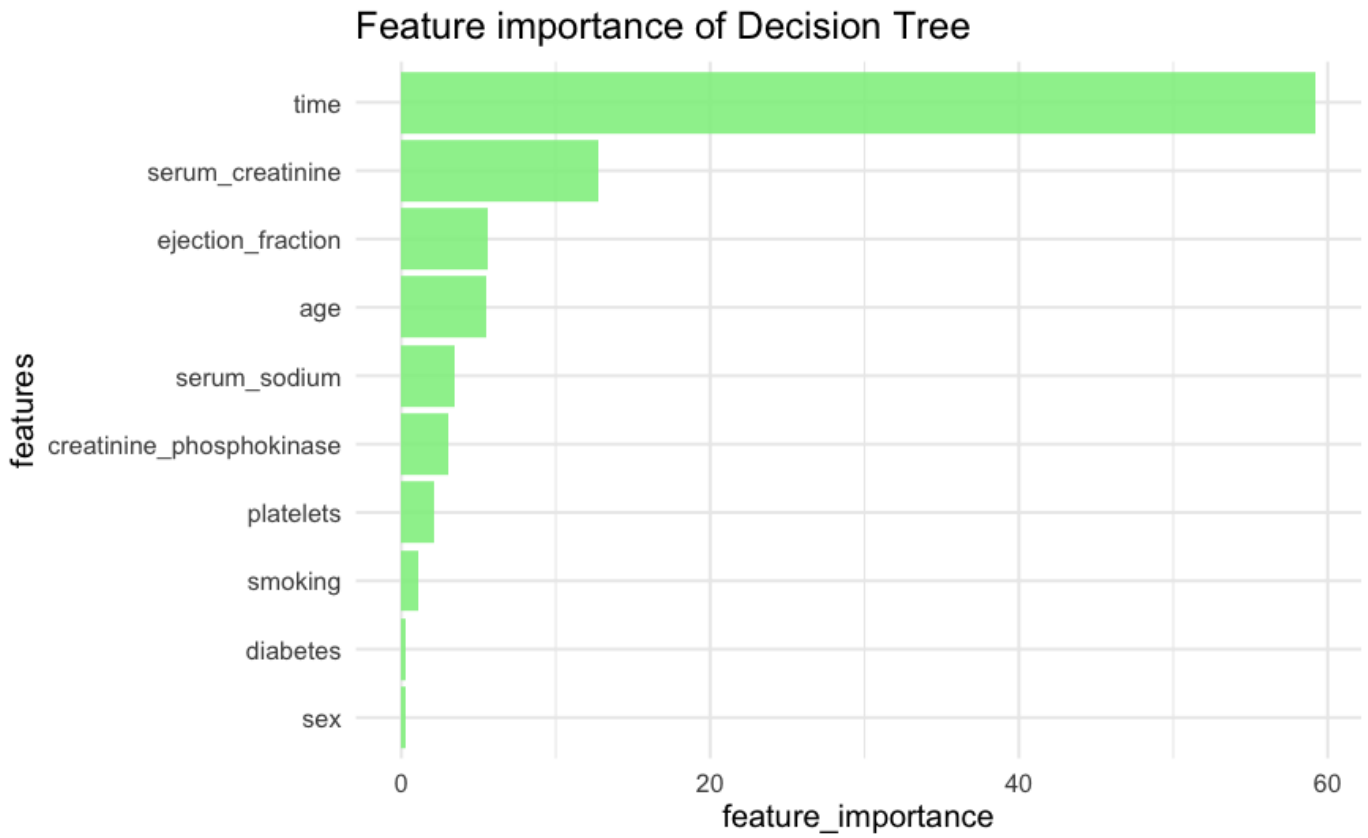
```
Accuracy<-(conf.matrix[1,1] + conf.matrix[2,2])/sum(conf.matrix)*100

Accuracy
```

```
[1] 87.62542
```

Finding the real impact of features

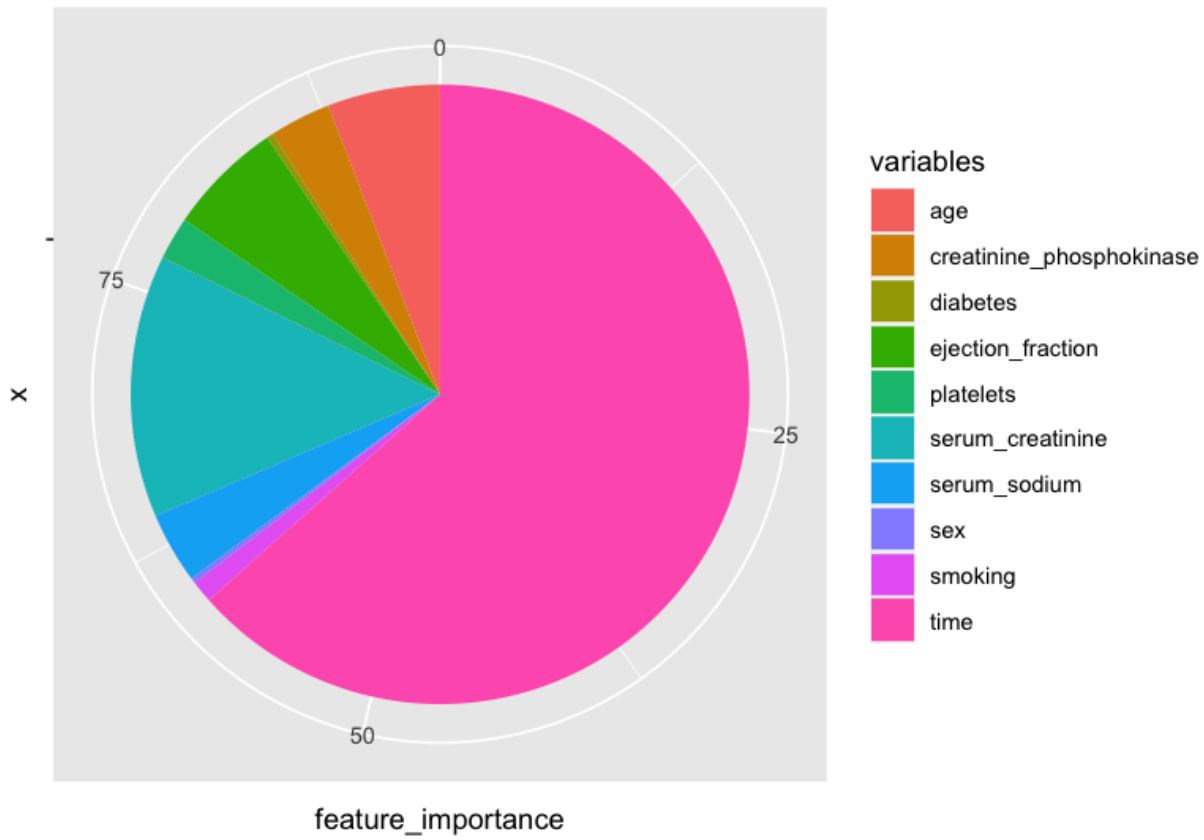
Hide

```
# We will create a bar chart to visualize the feature importance descendingly
importance <- data.frame(variables = names(heartTree$variable.importance), feature_im
portance = heartTree$variable.importance)

  ggplot(data = importance, aes(x=feature_importance, y=reorder(variables, X= feature
_importance))) + geom_bar(stat = "identity",
            fill = 'lightgreen',
             alpha=0.9) +
    labs(y = "features", title = "Feature importance of Decision Tree") +
    theme_minimal(base_size = 12)
```



Hide

```
#Visualize the feature importance using pie chart
importance <- data.frame(variables=names(heartTree$variable.importance), feature_impo
rtance=heartTree$variable.importance)
ggplot(data=importance, aes(x="", y=feature_importance, fill=variables)) + geom_bar(s
tat="identity", width=1) +
  coord_polar("y", start=0)
```



* `time` is the most effective of all, while `diabetes` is the least effective
* `sex` is not as effective as time