heart_failure_project

William Kubin

11/12/2021

Analysis of Heart Failure Data

This is an R Markdown document illustrating the prediction of Heart Failure in people. The data has 299 observations and 13 variables namely age, anemia, creatinine phosphokinase, diabetes, ejection fraction, high blood pressure, platelets, serum creatinine, serum sodium, sex, smoking, time and the target variable DEATH EVENT.

1. age : Age of individual.
2. anemia : Reduction of red blood cells or hemoglobin (1: True, 0: False).
3. creatinine_phosphokinase : Level of CPK enzyme in blood.
4. diabetes : Whether individual has diabetes (1: True, 0: False).
5. ejection_fraction : Percentage of blood leaving the heart at each contraction.
6. high_blood_pressure : Whether individual has hypertension (1: True, 0: False).
7. platelets : Platelets in the blood.
8. serum_creatinine : Amount of serum creatinine in blood.
9. serum_sodium : Amount of serum sodium in blood.
10. sex : Whether male or female (1: Man, 0: Woman).
11. smoking : Whether individual smokes or not (1: True, 0: False).
12. time : Follow-up days.
13. DEATH_EVENT : Whether individual died during follow-up period (1: True, 0: False).

Codes below indicate the importation of the data in R and a few rows of the data given all the variables.

```
setwd("/Users/paa.willie/myStuff/GitHub_Projects/Heart_failure_project")

heart_failure_data = read.csv("heart_failure_clinical_records_dataset.csv", header = TRUE)
head(heart_failure_data)
```

```
##   age anaemia creatinine_phosphokinase diabetes ejection_fraction
## 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
##   high_blood_pressure platelets serum_creatinine serum_sodium sex smoking time
## 1                   1    265000             1.9          130   1       0    4
## 2                   0    263358             1.1          136   1       0    6
## 3                   0    162000             1.3          129   1       1    7
## 4                   0    210000             1.9          137   1       0    7
## 5                   0    327000             2.7          116   0       0    8
## 6                   1    204000             2.1          132   1       1    8
##   DEATH_EVENT
## 1           1
```

```
## 2           1
## 3           1
## 4           1
## 5           1
## 6           1
```

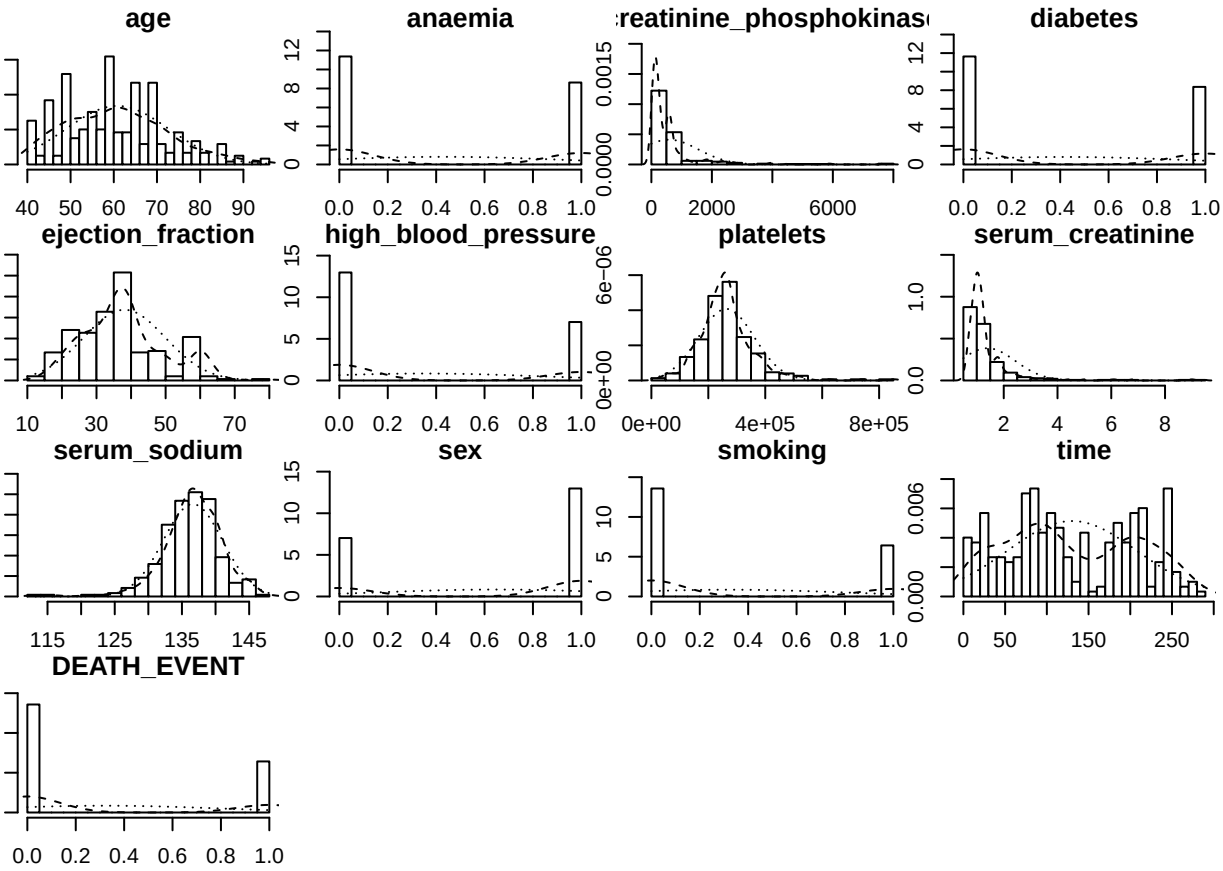
Let's now take a look at some descriptive statistics to get an overview of the variables in the data.

```
summary(heart_failure_data)
```

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

Histogram plots of the variables

```
library(psych)
multi.hist(heart_failure_data)
```




```
dataM = data.matrix(heart_failure_data)
#boxplot.matrix(dataM, use.cols = T)
#boxplot(heart_failure_data$age, heart_failure_data$DEATH_EVENT)
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
features <- c("anaemia", "diabetes", "high_blood_pressure", "sex", "smoking", "DEATH_EVENT")
HF_data <- heart_failure_data %>% mutate_at(features, as.factor)
```

Distribution of Numeric Features against target variable (DEATH EVENT)

1. AGE vs DEATH_EVENT

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following objects are masked from 'package:psych':
```

```
##
##     %+%, alpha
```

```
library(ggthemes)
library(patchwork)
library(stringr)

p <- ggplot(HF_data, aes(x = age)) +
  geom_histogram(binwidth = 5, colour = "white", alpha = 0.5) +
  geom_density(eval(bquote(aes(y = ..count.. * 5))), alpha = 0.25) +
  scale_x_continuous(breaks = seq(40, 100, 10)) +
  geom_vline(xintercept = median(HF_data$age), linetype="longdash") +
  annotate(geom = "text",
           x = max(HF_data$age)-5, y = 50,
           label = str_c("Min.     : ", min(HF_data$age),
                         "\nMedian : ", median(HF_data$age),
                         "\nMean    : ", round(mean(HF_data$age), 1),
                         "\nMax.    : ", max(HF_data$age))) +
  labs(title = "Age distribution") +
  theme_minimal(base_size = 12)
# binwidth can be calculated from "diff(range(df$age))/20"

q <- ggplot(HF_data, aes(x = age, fill = DEATH_EVENT)) +
  geom_density(alpha = 0.64) +
  scale_fill_manual(values = c("red", "blue"),
                    name = "DEATH_EVENT",
                    labels = c("False", "True")) +
  scale_x_continuous(breaks = seq(40, 100, 10)) +

  geom_vline(xintercept = median(filter(HF_data, DEATH_EVENT == 0)$age), linetype="longdash") +
  geom_vline(xintercept = median(filter(HF_data, DEATH_EVENT == 1)$age), linetype="longdash") +
  annotate(geom = "text",
           x = max(HF_data$age)-10, y = 0.03,
           label = str_c("Survived median: ", median(filter(HF_data, DEATH_EVENT == 0)$age),
                         "\nDead median: ", median(filter(HF_data, DEATH_EVENT == 1)$age))) +

  labs(title = "Relationship: Age vs DEATH_EVENT") +
  theme_minimal(base_size = 12) +
  theme(legend.position = "bottom", legend.direction = "horizontal")

library(data.table)
```

```
##
## Attaching package: 'data.table'
```

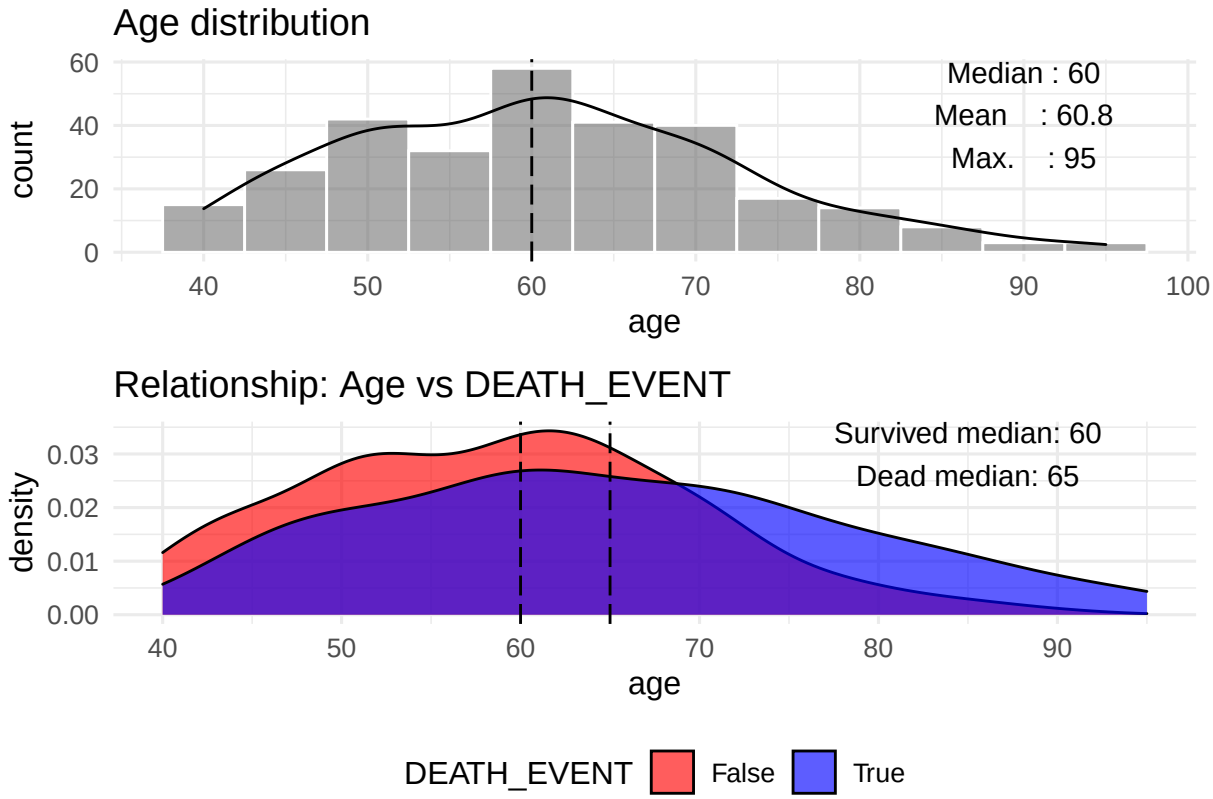
```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
breaks.in.Age <- c(40,45,50,55,60,65,70,75,80,85,90,200)
labels.Age <- c("Below 45","45-49","50-54","55-59","60-64","65-69",
                "70-74","75-79","80-84","85-89","Above 89")

mm <- heart_failure_data
setDT(mm)[ , groups.Age := cut(age,
                               breaks = breaks.in.Age,
```

```
                                right = FALSE,
                                labels = labels.Age)]

Ages = mm %>% group_by(groups.Age) %>% count() %>% pull(n)
Ages.percent = round((Ages/sum(Ages))*100, 1)

death.Ages = mm %>% filter(DEATH_EVENT==1) %>% group_by(groups.Age) %>% count() %>% pull(n)
death.Age.percent = round((death.Ages/sum(death.Ages))*100, 1)

perc.death = round((death.Ages/Ages)*100, 1)

table.Ages.stats = cbind(labels.Age, Ages, death.Ages, perc.death)
colnames(table.Ages.stats) = c("Age Groups (Years)", "Number of Patients",
                         "Number of Deaths",
                         "Percentage of Deaths (%)")
table.Ages.stats
```

```
##       Age Groups (Years) Number of Patients Number of Deaths
##  [1,] "Below 45"         "18"               "1"
##  [2,] "45-49"            "29"               "10"
##  [3,] "50-54"            "48"               "11"
##  [4,] "55-59"            "34"               "9"
##  [5,] "60-64"            "55"               "15"
##  [6,] "65-69"            "38"               "12"
##  [7,] "70-74"            "36"               "13"
##  [8,] "75-79"            "16"               "7"
##  [9,] "80-84"            "11"               "8"
## [10,] "85-89"            "8"                "5"
## [11,] "Above 89"         "6"                "5"
##       Percentage of Deaths (%)
##  [1,] "5.6"
##  [2,] "34.5"
##  [3,] "22.9"
##  [4,] "26.5"
##  [5,] "27.3"
##  [6,] "31.6"
##  [7,] "36.1"
##  [8,] "43.8"
##  [9,] "72.7"
## [10,] "62.5"
## [11,] "83.3"
```

```
p / q
```

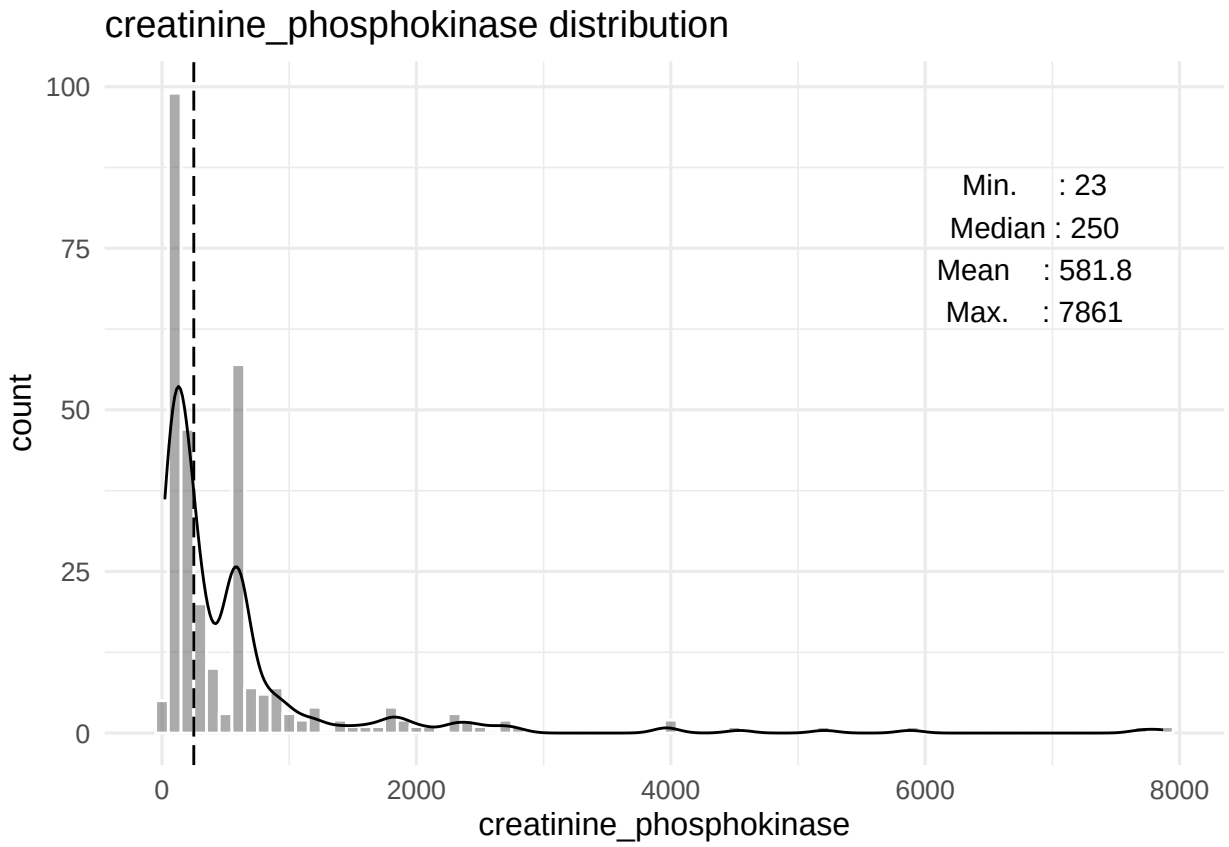


Observation: The modal age of patients is around 60 years old. Also, the younger your age (before 68years old), the more difficulty to die. After a patient grows beyond about 68 years, the probability for them to die of heart failure increases rapidly. These results are indicated in the table showing percentages of death within different age groups.

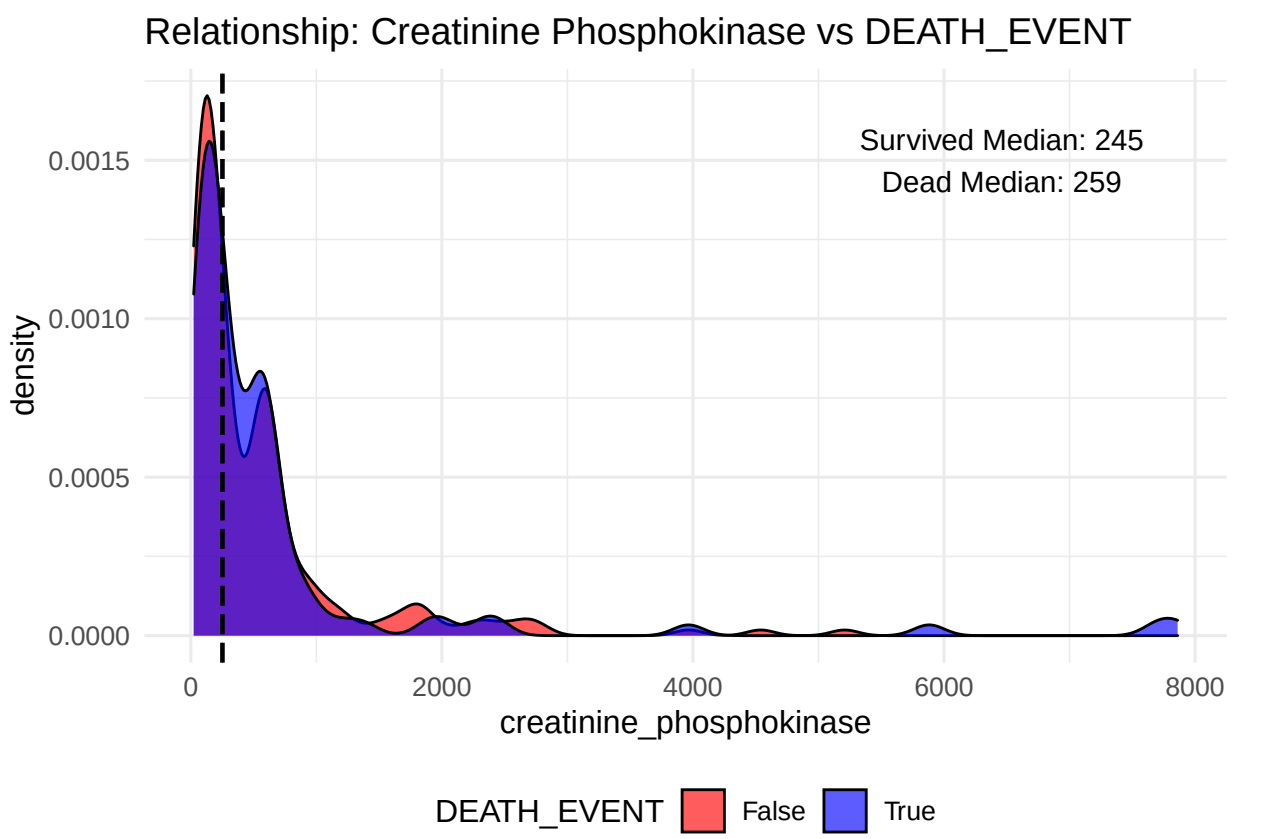
2. CREATININE PHOSPHOKINASE vs DEATH_EVENT

```
V <- ggplot(HF_data, aes(x = creatinine_phosphokinase)) +
  geom_histogram(binwidth = 100, colour = "white", alpha = 0.5) +
  geom_density(eval(bquote(aes(y = ..count.. * 100))), alpha = 0.25) +
  geom_vline(xintercept = median(HF_data$creatinine_phosphokinase), linetype="longdash") +
  annotate(geom = "text",
           x = max(HF_data$creatinine_phosphokinase)-1000, y = 75,
           label = str_c("Min.     : ", min(HF_data$creatinine_phosphokinase),
                         "\nMedian : ", median(HF_data$creatinine_phosphokinase),
                         "\nMean    : ", round(mean(HF_data$creatinine_phosphokinase), 1),
                         "\nMax.    : ", max(HF_data$creatinine_phosphokinase))) +
  labs(title = "creatinine_phosphokinase distribution") +
  theme_minimal(base_size = 12)

W <- ggplot(HF_data, aes(x = creatinine_phosphokinase, fill = DEATH_EVENT)) +
  geom_density(alpha = 0.64) +
  scale_fill_manual(values = c("red", "blue"),
                    name = "DEATH_EVENT",
                    labels = c("False", "True")) +
  geom_vline(xintercept = median(filter(HF_data, DEATH_EVENT == 0)$creatinine_phosphokinase), linetype=
  geom_vline(xintercept = median(filter(HF_data, DEATH_EVENT == 1)$creatinine_phosphokinase), linetype=
  annotate(geom = "text",
           x = max(HF_data$creatinine_phosphokinase)-1400, y = 0.0015,
```

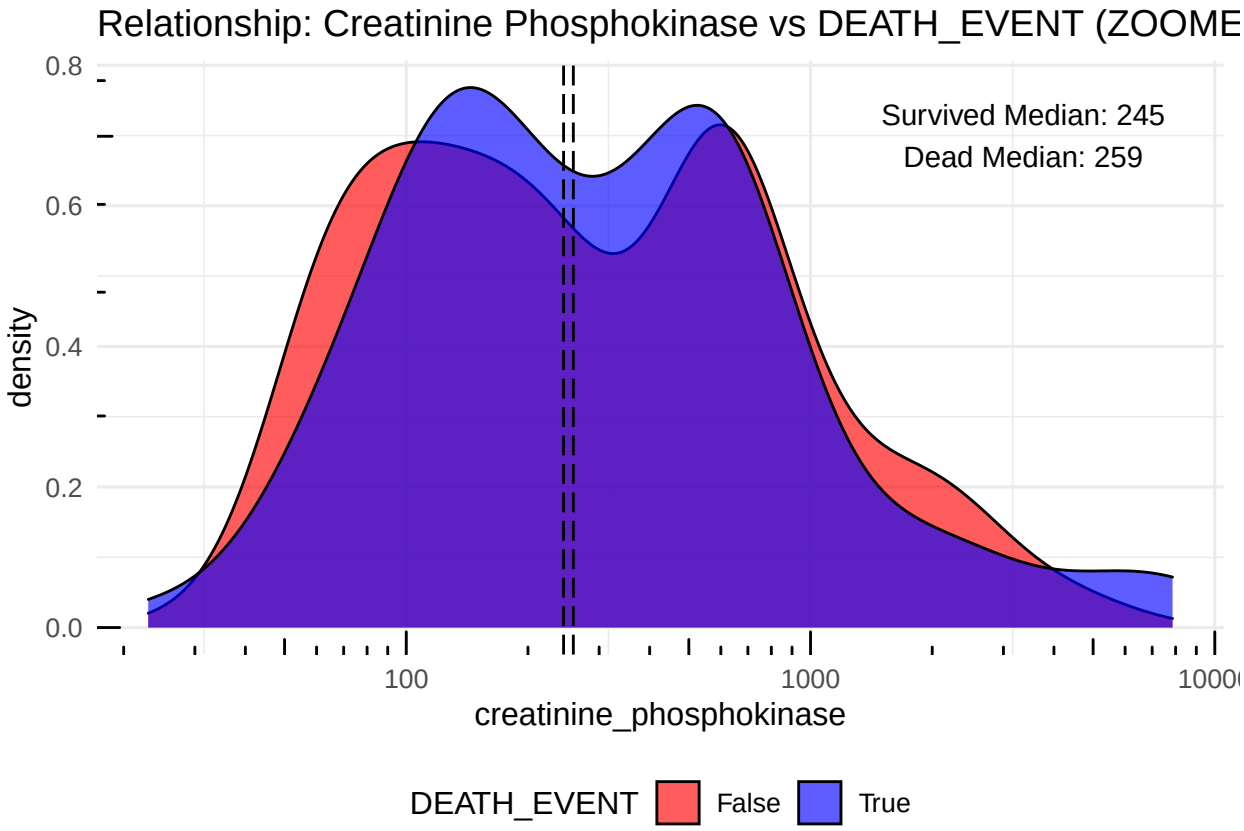
```
          label = str_c("Survived Median: ", median(filter(HF_data, DEATH_EVENT == 0)$creatinine_phospl
                        "\nDead Median: ", median(filter(HF_data, DEATH_EVENT == 1)$creatinine_phosphol

  labs(title = "Relationship: Creatinine Phosphokinase vs DEATH_EVENT") +
  theme_minimal(base_size = 12) +
  theme(legend.position = "bottom", legend.direction = "horizontal")

Z <- ggplot(HF_data, aes(x = creatinine_phosphokinase, fill = DEATH_EVENT)) +
  geom_density(alpha = 0.64) +
  scale_fill_manual(values = c("red", "blue"),
                    name = "DEATH_EVENT",
                    labels = c("False", "True")) +
  geom_vline(xintercept = median(filter(HF_data, DEATH_EVENT == 0)$creatinine_phosphokinase), linetype=
  geom_vline(xintercept = median(filter(HF_data, DEATH_EVENT == 1)$creatinine_phosphokinase), linetype=
  annotate(geom = "text",
           x = max(HF_data$creatinine_phosphokinase)-4500, y = 0.7,
           label = str_c("Survived Median: ", median(filter(HF_data, DEATH_EVENT == 0)$creatinine_phospl
                         "\nDead Median: ", median(filter(HF_data, DEATH_EVENT == 1)$creatinine_phosphol

  labs(title = "Relationship: Creatinine Phosphokinase vs DEATH_EVENT (ZOOMED)") +
  theme_minimal(base_size = 12) +
  theme(legend.position = "bottom", legend.direction = "horizontal") +
  scale_x_log10() +
  annotation_logticks()

V
```



```
W
```


Relationship: Creatinine Phosphokinase vs DEATH_EVENT
Survived Median: 245
Dead Median: 259
density
0.0015
0.0010
0.0005
0.0000
0
2000
4000
6000
8000
creatinine_phosphokinase
DEATH_EVENT
False
True

```
Z
```

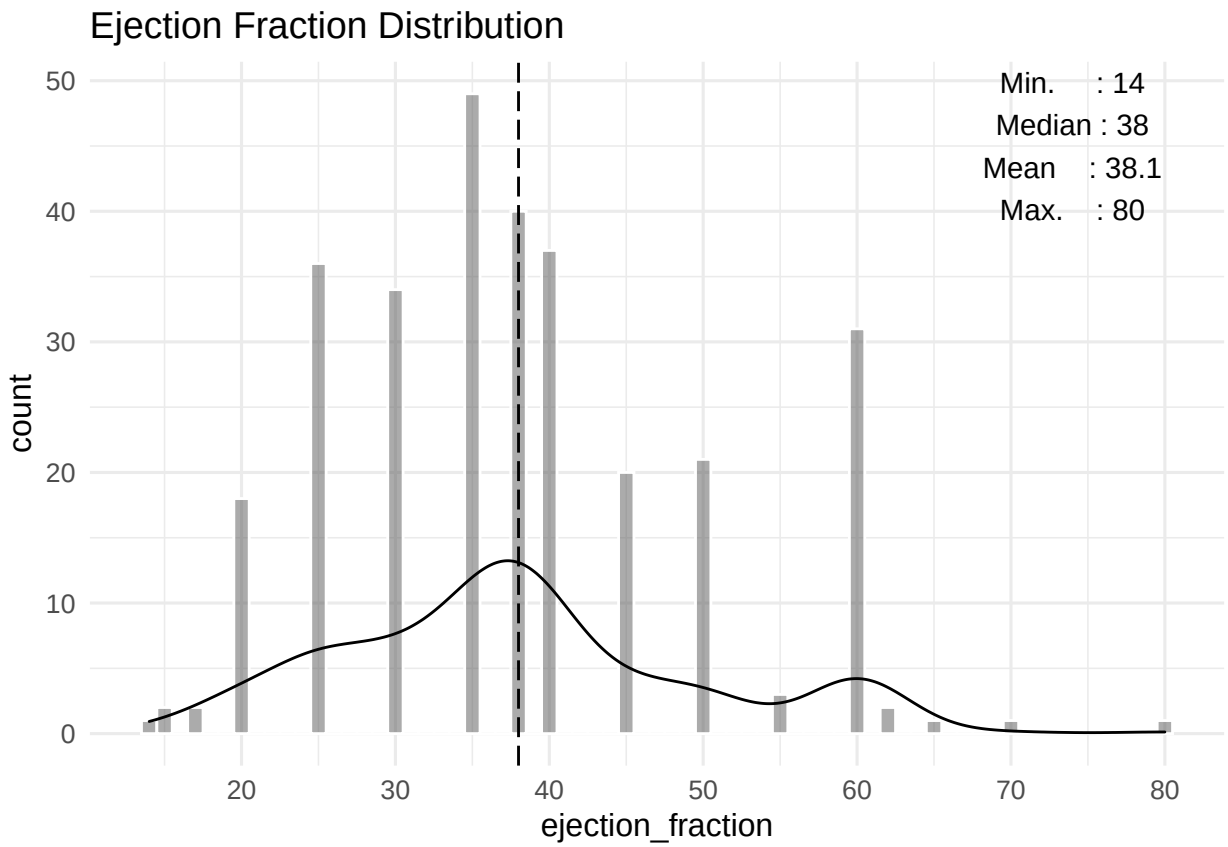


Observation: The median Creatinine Phosphokinase of patients is around 250 mcg/L and the mean is 581.8 mcg/L. Its distribution is skewed on the right. The minimum observation is 23 mcg/L whereas the maximum observation is 7861 mcg/L which is about 13 times the average of Creatinine Phosphokinase. We observe a little difference in the median.

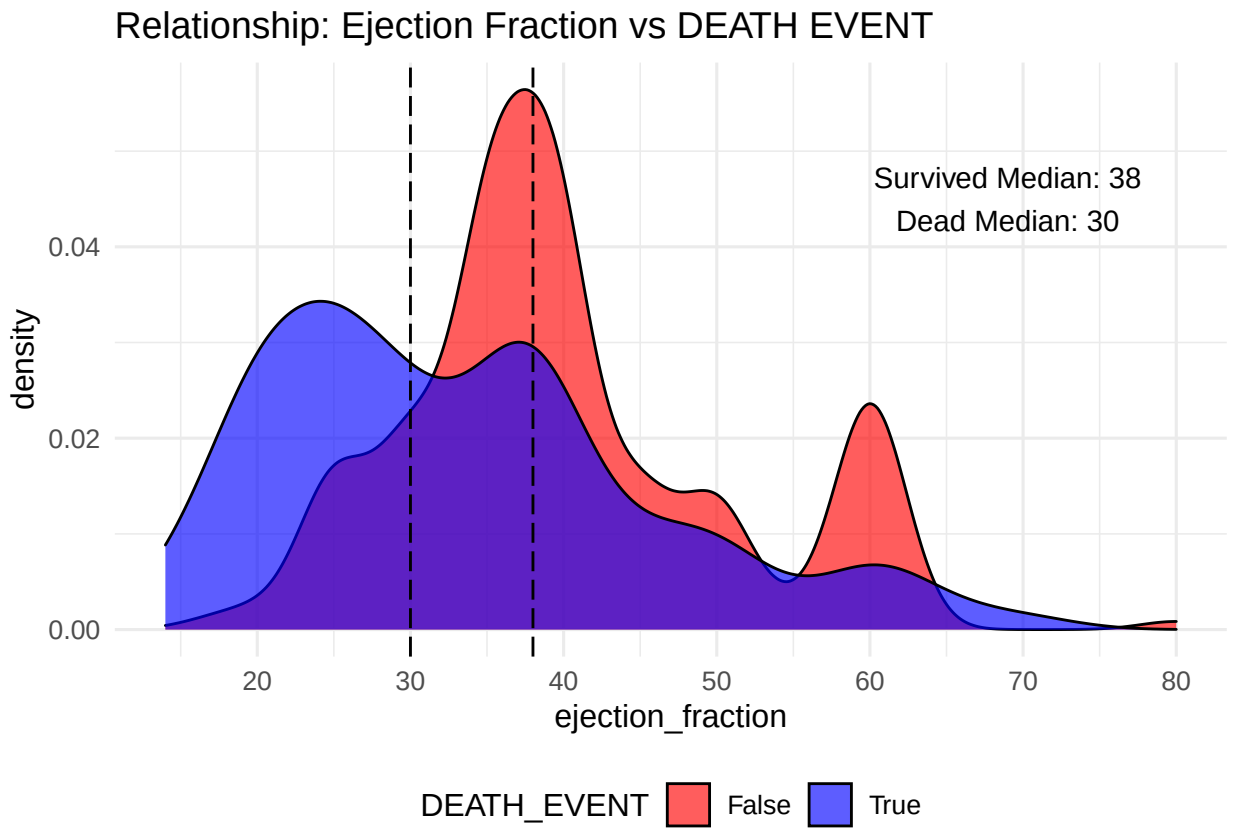
2. EJECTION FRACTION vs DEATH EVENT

```
p <- ggplot(HF_data, aes(x = ejection_fraction)) +
  geom_histogram(binwidth = 1, colour = "white", alpha = 0.5) +
  geom_density(eval(bquote(aes(y = ..count.. * 1))), alpha = 0.25) +
  scale_x_continuous(breaks = seq(10, 80, 10)) +
  geom_vline(xintercept = median(HF_data$ejection_fraction), linetype="longdash") +
  annotate(geom = "text",
           x = max(HF_data$ejection_fraction)-6, y = 45,
           label = str_c("Min.      : ", min(HF_data$ejection_fraction),
                         "\nMedian : ", median(HF_data$ejection_fraction),
                         "\nMean    : ", round(mean(HF_data$ejection_fraction), 1),
                         "\nMax.    : ", max(HF_data$ejection_fraction))) +
  labs(title = "Ejection Fraction Distribution") +
  theme_minimal(base_size = 12)

q <- ggplot(HF_data, aes(x = ejection_fraction, fill = DEATH_EVENT)) +
  geom_density(alpha = 0.64) +
  scale_fill_manual(values = c("red", "blue"),
                    name = "DEATH_EVENT",
                    labels = c("False", "True")) +
  scale_x_continuous(breaks = seq(10, 80, 10)) +
```

```
  geom_vline(xintercept = median(filter(HF_data, DEATH_EVENT == 0)$ejection_fraction), linetype="longdas
  geom_vline(xintercept = median(filter(HF_data, DEATH_EVENT == 1)$ejection_fraction), linetype="longdas
  annotate(geom = "text",
           x = max(HF_data$age)-26, y = 0.045,
           label = str_c("Survived Median: ", median(filter(HF_data, DEATH_EVENT == 0)$ejection_fractio
                         "\nDead Median: ", median(filter(HF_data, DEATH_EVENT == 1)$ejection_fraction)

  labs(title = "Relationship: Ejection Fraction vs DEATH EVENT") +
  theme_minimal(base_size = 12) +
  theme(legend.position = "bottom", legend.direction = "horizontal")

p
```




```
q
```



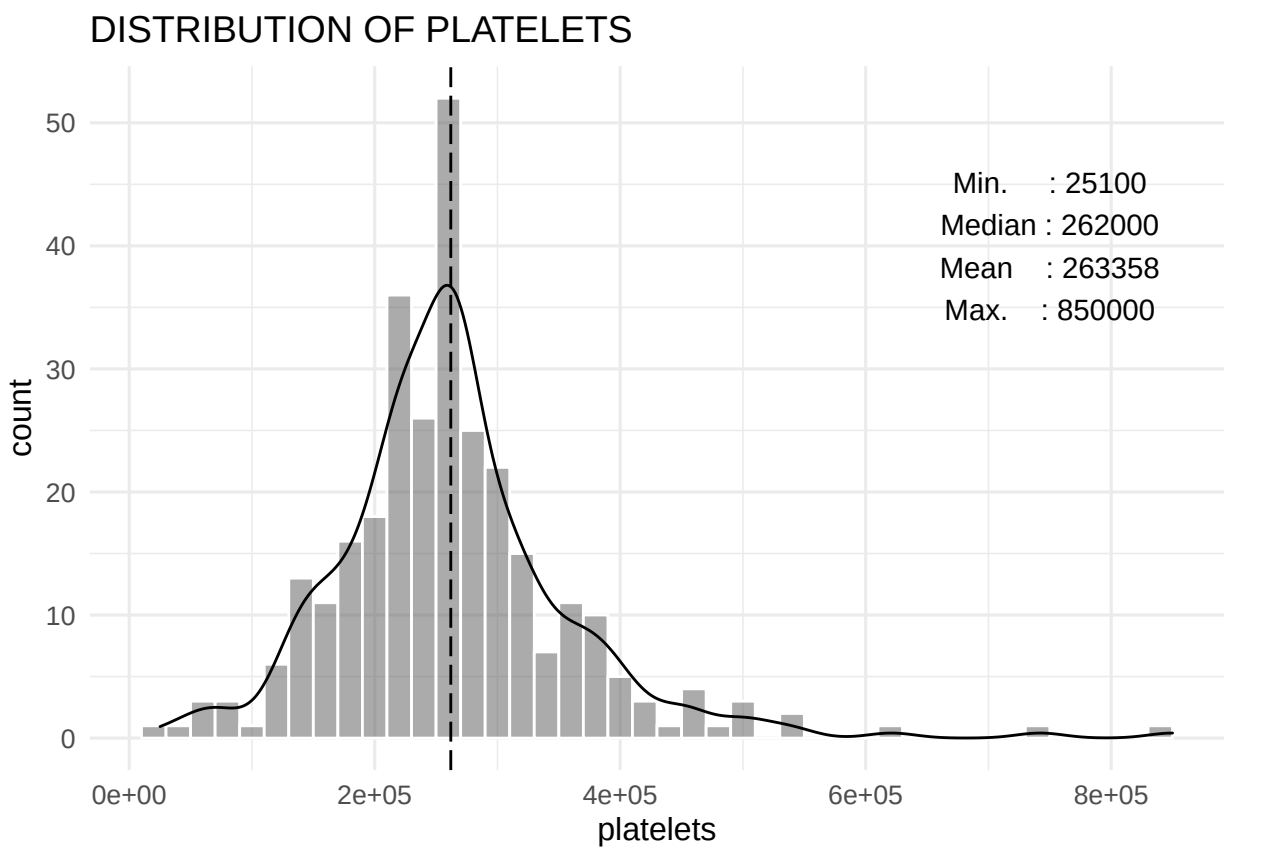
Observation: The median and mean Ejection Fraction of patients is approximately 38%.The distribution of Ejection Fraction looks discrete, not continuous. We observe some difference between median of survival and death. The values for death is highly distributed around 30% and then diminishes slowly.

3. PLATELETS vs DEATH EVENT

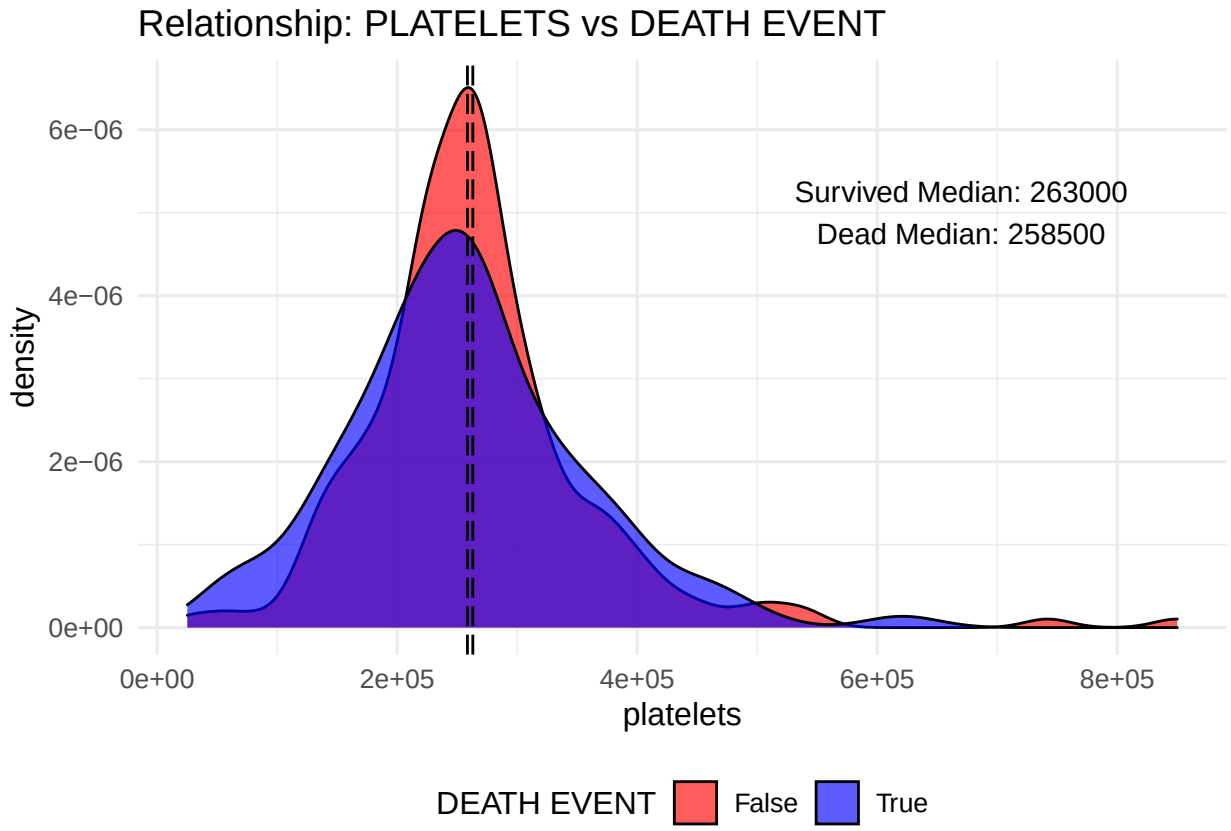
```
a <- ggplot(HF_data, aes(x = platelets)) +
  geom_histogram(binwidth = 20000, colour = "white", alpha = 0.5) +
  geom_density(eval(bquote(aes(y = ..count.. * 20000))), alpha = 0.25) +
  geom_vline(xintercept = median(HF_data$platelets), linetype="longdash") +
  annotate(geom = "text",
           x = max(HF_data$platelets)-100000, y = 40,
           label = str_c("Min.      : ", min(HF_data$platelets),
                         "\nMedian : ", median(HF_data$platelets),
                         "\nMean    : ", round(mean(HF_data$platelets), 1),
                         "\nMax.     : ", max(HF_data$platelets))) +
  labs(title = "DISTRIBUTION OF PLATELETS") +
  theme_minimal(base_size = 12)

b <- ggplot(HF_data, aes(x = platelets, fill = DEATH_EVENT)) +
  geom_density(alpha = 0.64) +
  scale_fill_manual(values = c("red", "blue"),
                    name = "DEATH EVENT",
                    labels = c("False", "True")) +

  geom_vline(xintercept = median(filter(HF_data, DEATH_EVENT == 0)$platelets), linetype="longdash") +
  geom_vline(xintercept = median(filter(HF_data, DEATH_EVENT == 1)$platelets), linetype="longdash") +
  annotate(geom = "text",
```

```
          x = max(HF_data$platelets)-180000, y = 0.000005,
          label = str_c("Survived Median: ", median(filter(HF_data, DEATH_EVENT == 0)$platelets),
                        "\nDead Median: ", median(filter(HF_data, DEATH_EVENT == 1)$platelets))) +

  labs(title = "Relationship: PLATELETS vs DEATH EVENT") +
  theme_minimal(base_size = 12) +
  theme(legend.position = "bottom", legend.direction = "horizontal")
```

a



b

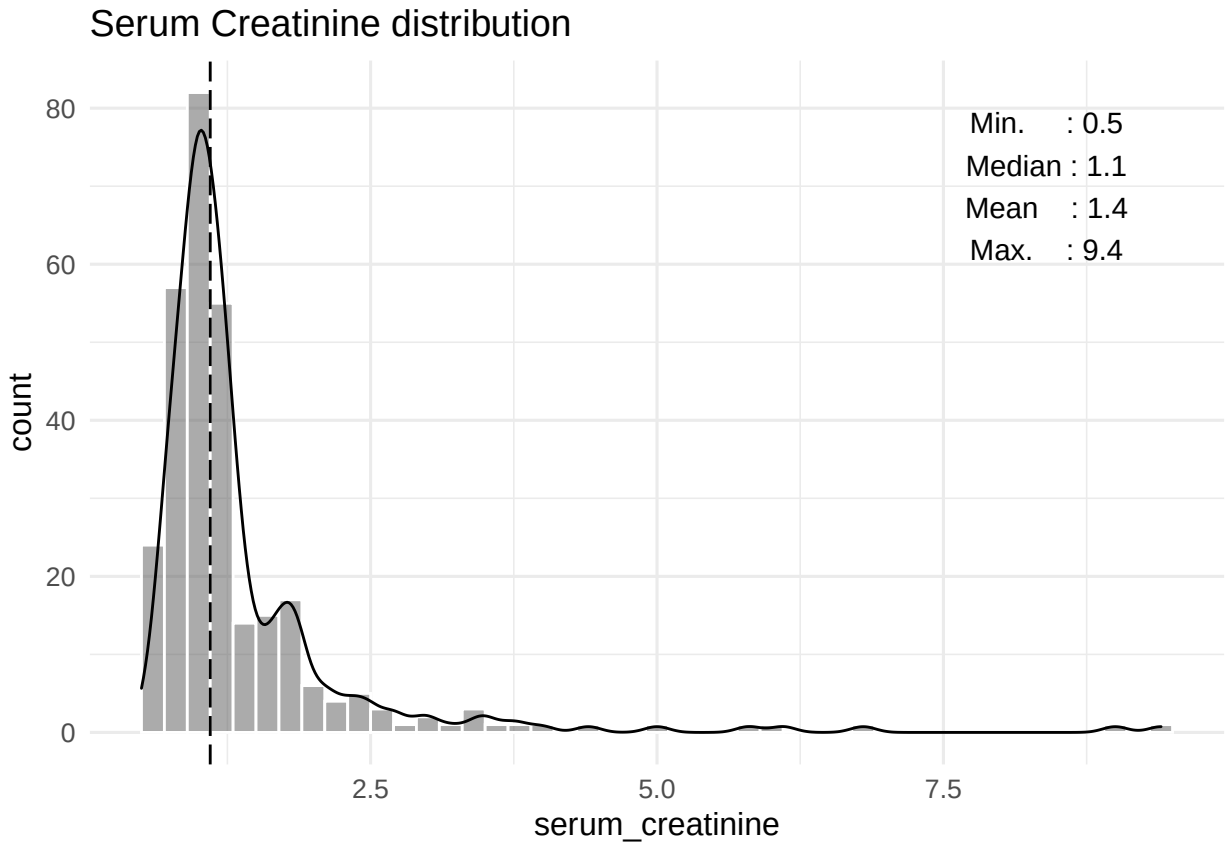


Observation: The distribution of platelets looks symmetric (close to bell-shape). Meanwhile, survivors have slightly higher platelet counts than those with high propensity to die.

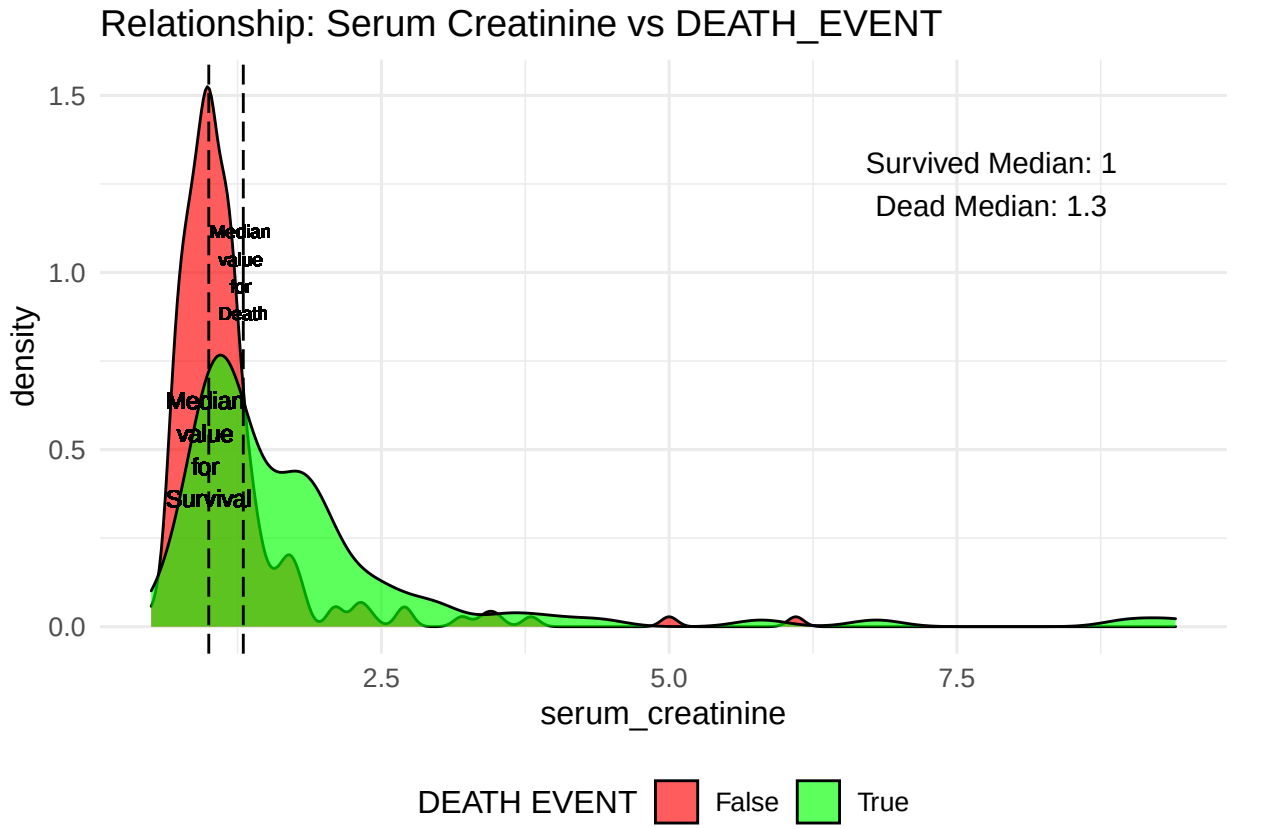
4. SERUM CREATININE vs DEATH EVENT

```
a <- ggplot(HF_data, aes(x = serum_creatinine)) +
  geom_histogram(binwidth = 0.2, colour = "white", alpha = 0.5) +
  geom_density(eval(bquote(aes(y = ..count.. * 0.2))), alpha = 0.25) +
  geom_vline(xintercept = median(HF_data$serum_creatinine), linetype="longdash") +
  annotate(geom = "text",
           x = max(HF_data$serum_creatinine)-1, y = 70,
           label = str_c("Min.     : ", min(HF_data$serum_creatinine),
                         "\nMedian : ", median(HF_data$serum_creatinine),
                         "\nMean   : ", round(mean(HF_data$serum_creatinine), 1),
                         "\nMax.    : ", max(HF_data$serum_creatinine))) +
  labs(title = "Serum Creatinine distribution") +
  theme_minimal(base_size = 12)

b <- ggplot(HF_data, aes(x = serum_creatinine, fill = DEATH_EVENT)) +
  geom_density(alpha = 0.64) +
  scale_fill_manual(values = c("red", "green"),
                    name = "DEATH EVENT",
                    labels = c("False", "True")) +

  geom_vline(xintercept = median(filter(HF_data, DEATH_EVENT == 0)$serum_creatinine), linetype="longdash
  geom_vline(xintercept = median(filter(HF_data, DEATH_EVENT == 1)$serum_creatinine), linetype="longdash
    geom_text(label = "Median \nvalue \nfor \nSurvival", x = 1, y = 0.5, size = 3) +
  geom_text(label = "Median \nvalue \nfor \nDeath", x = 1.3, y = 1, size = 2.5) +
```

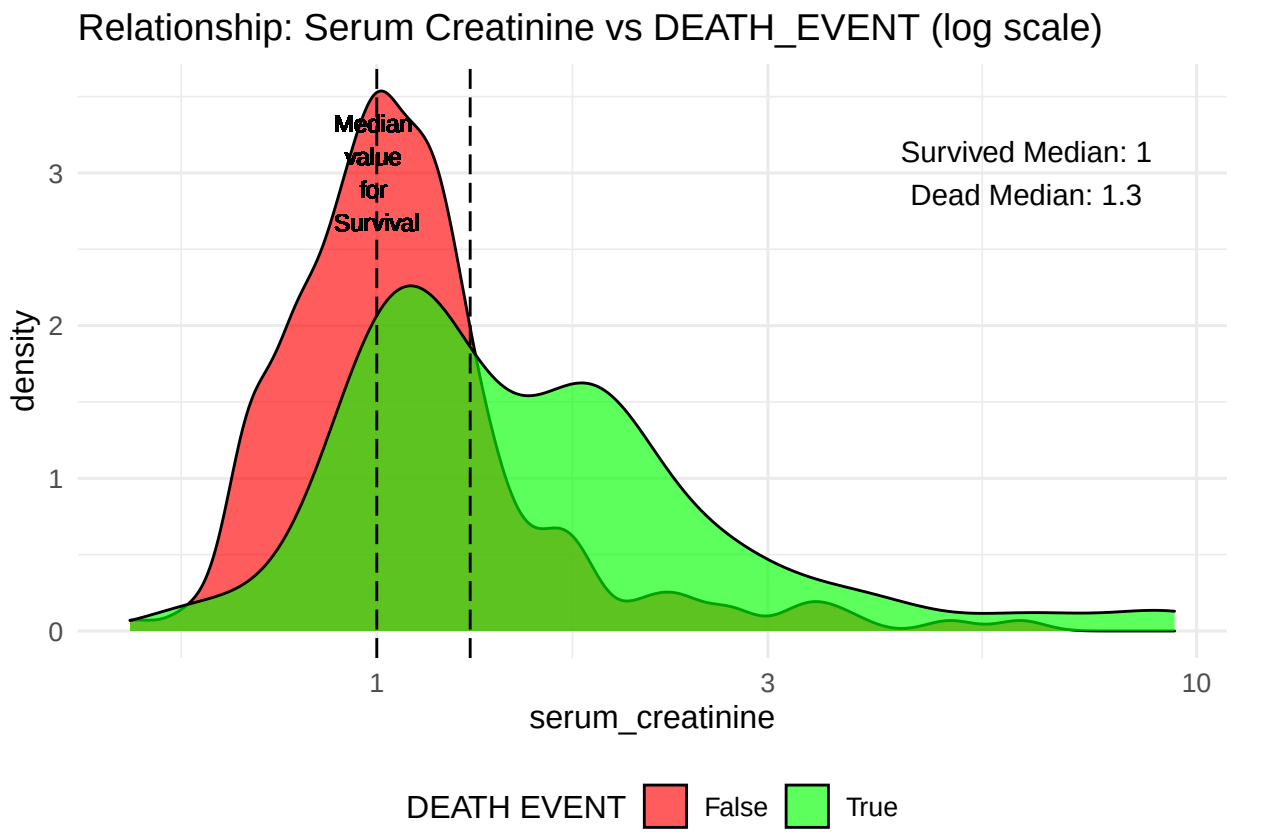
```
  annotate(geom = "text",
           x = max(HF_data$serum_creatinine)-1.6, y = 1.25,
           label = str_c("Survived Median: ", median(filter(HF_data, DEATH_EVENT == 0)$serum_creatinine)
                         "\nDead Median: ", median(filter(HF_data, DEATH_EVENT == 1)$serum_creatinine)))

  labs(title = "Relationship: Serum Creatinine vs DEATH_EVENT") +
  theme_minimal(base_size = 12) +
  theme(legend.position = "bottom", legend.direction = "horizontal")

c <- ggplot(HF_data, aes(x = serum_creatinine, fill = factor(DEATH_EVENT))) +
  geom_density(alpha = 0.64) +
  scale_fill_manual(values = c("red", "green"),
                    name = "DEATH EVENT",
                    labels = c("False", "True")) +

  geom_vline(xintercept = median(filter(HF_data, DEATH_EVENT == 0)$serum_creatinine), linetype="longdash
  geom_vline(xintercept = median(filter(HF_data, DEATH_EVENT == 1)$serum_creatinine), linetype="longdash
  geom_text(label = "Median \nvalue \nfor \nSurvival", x = log(1), y = 3, size = 3) +
  geom_text(label = "Median \nvalue \nfor \nDeath", x = 1.3, y = 1, size = 2.5) +
  annotate(geom = "text",
           x = max(HF_data$serum_creatinine)-3.2, y = 3,
           label = str_c("Survived Median: ", median(filter(HF_data, DEATH_EVENT == 0)$serum_creatinine)
                         "\nDead Median: ", median(filter(HF_data, DEATH_EVENT == 1)$serum_creatinine)))

  labs(title = "Relationship: Serum Creatinine vs DEATH_EVENT (log scale)") +
  theme_minimal(base_size = 12) +
  theme(legend.position = "bottom", legend.direction = "horizontal") +
  scale_x_log10()

a
```


Serum Creatinine distribution
Min. : 0.5
Median : 1.1
Mean : 1.4
Max. : 9.4
count
serum_creatinine

b


Relationship: Serum Creatinine vs DEATH_EVENT
Survived Median: 1
Dead Median: 1.3
Median value for Death
Median value for Survival
density
serum_creatinine
DEATH EVENT
False
True

c



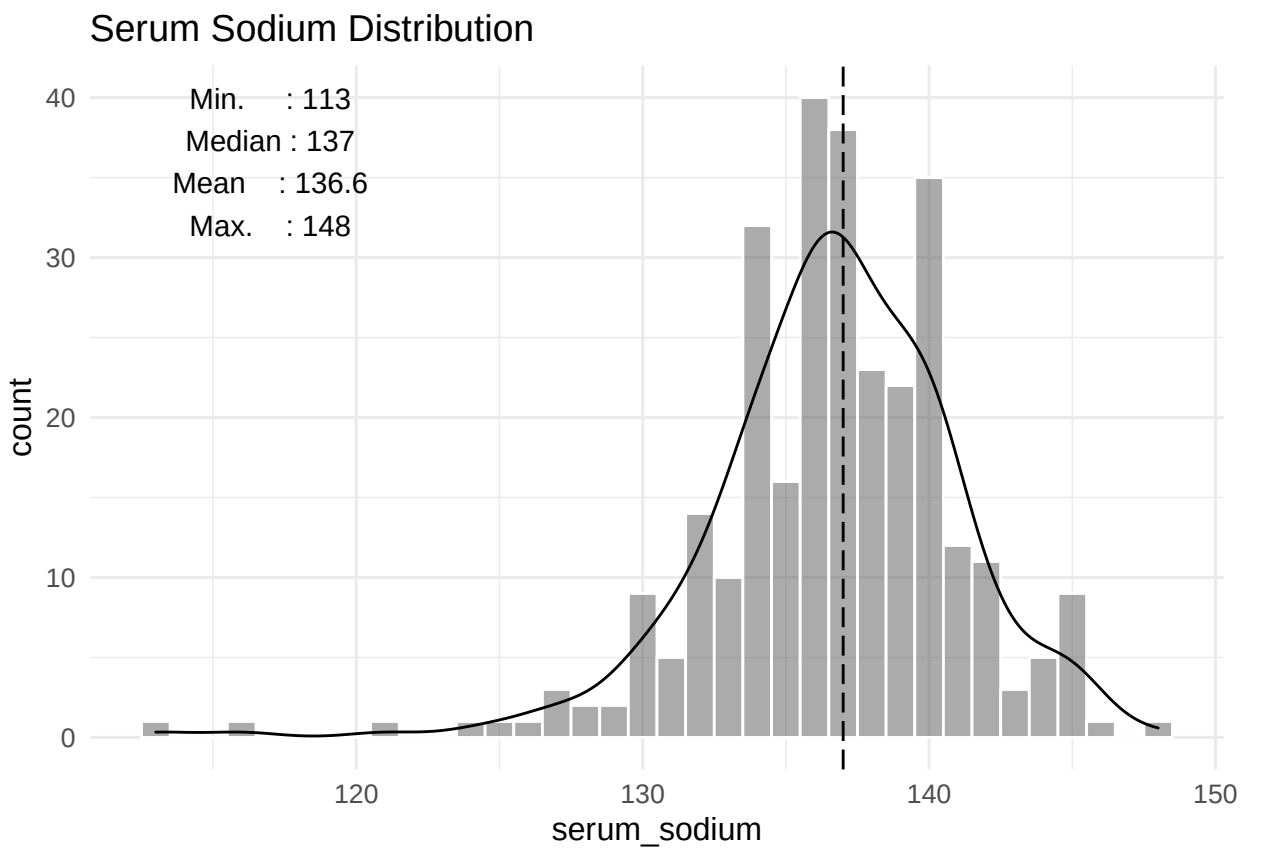
Observation: Distribution of serum creatinine is skewed to the left. The values of the survivors are clustered around the median. This is not so for the values of death where there are a lot of cases that exceed its median (i.e. 1.3).

5. SERUM SODIUM vs DEATH EVENT

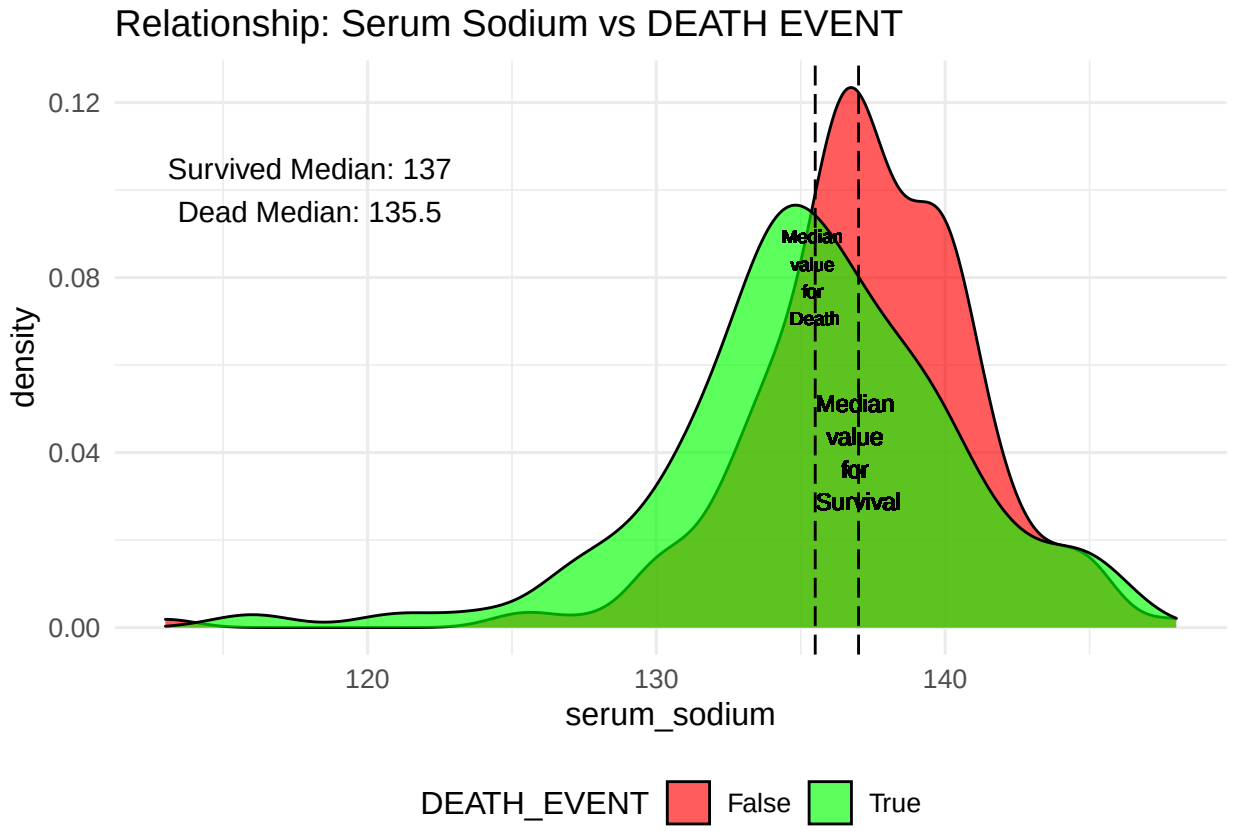
```
X <- ggplot(HF_data, aes(x = serum_sodium)) +
  geom_histogram(binwidth = 1, colour = "white", alpha = 0.5) +
  geom_density(eval(bquote(aes(y = ..count.. * 1))), alpha = 0.25) +
  scale_x_continuous(breaks = seq(110, 150, 10)) +
  geom_vline(xintercept = median(HF_data$serum_sodium), linetype="longdash") +
  annotate(geom = "text",
           x = min(HF_data$serum_sodium)+4, y = 36,
           label = str_c("Min.      : ", min(HF_data$serum_sodium),
                         "\nMedian : ", median(HF_data$serum_sodium),
                         "\nMean    : ", round(mean(HF_data$serum_sodium), 1),
                         "\nMax.     : ", max(HF_data$serum_sodium))) +
  labs(title = "Serum Sodium Distribution") +
  theme_minimal(base_size = 12)

Y <- ggplot(HF_data, aes(x = serum_sodium, fill = DEATH_EVENT)) +
  geom_density(alpha = 0.64) +
  scale_fill_manual(values = c("red", "green"),
                    name = "DEATH_EVENT",
                    labels = c("False", "True")) +
  scale_x_continuous(breaks = seq(110, 150, 10)) +
```

```
  geom_vline(xintercept = median(filter(HF_data, DEATH_EVENT == 0)$serum_sodium), linetype="longdash") +
  geom_vline(xintercept = median(filter(HF_data, DEATH_EVENT == 1)$serum_sodium), linetype="longdash") +
  geom_text(label = "Median \nvalue \nfor \nSurvival", x = 137, y = 0.04, size = 3) +
  geom_text(label = "Median \nvalue \nfor \nDeath", x = 135.5, y = 0.08, size = 2.5) +
  annotate(geom = "text",
           x = min(HF_data$serum_sodium)+5, y = 0.1,
           label = str_c("Survived Median: ", median(filter(HF_data, DEATH_EVENT == 0)$serum_sodium),
                         "\nDead Median: ", median(filter(HF_data, DEATH_EVENT == 1)$serum_sodium))) +

  labs(title = "Relationship: Serum Sodium vs DEATH EVENT") +
  theme_minimal(base_size = 12) +
  theme(legend.position = "bottom", legend.direction = "horizontal")
```

```
X
```




```
Y
```



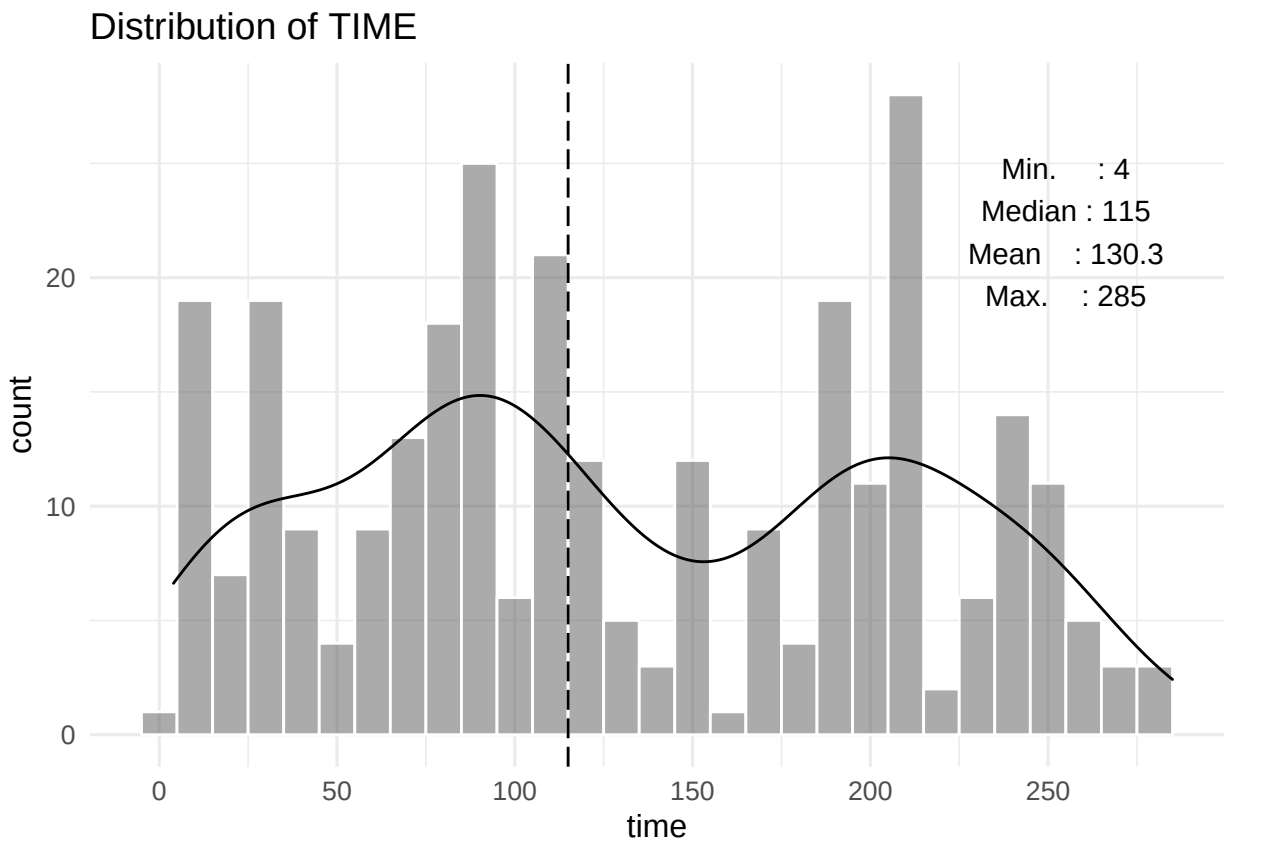
Observation: The distribution of values of serum sodium is close to symmetric (bell-shape) even though there are some low values. There is some difference between the median values of dead and survived patients.

6. TIME (FOLLOW-UP PERIOD) vs DEATH EVENT

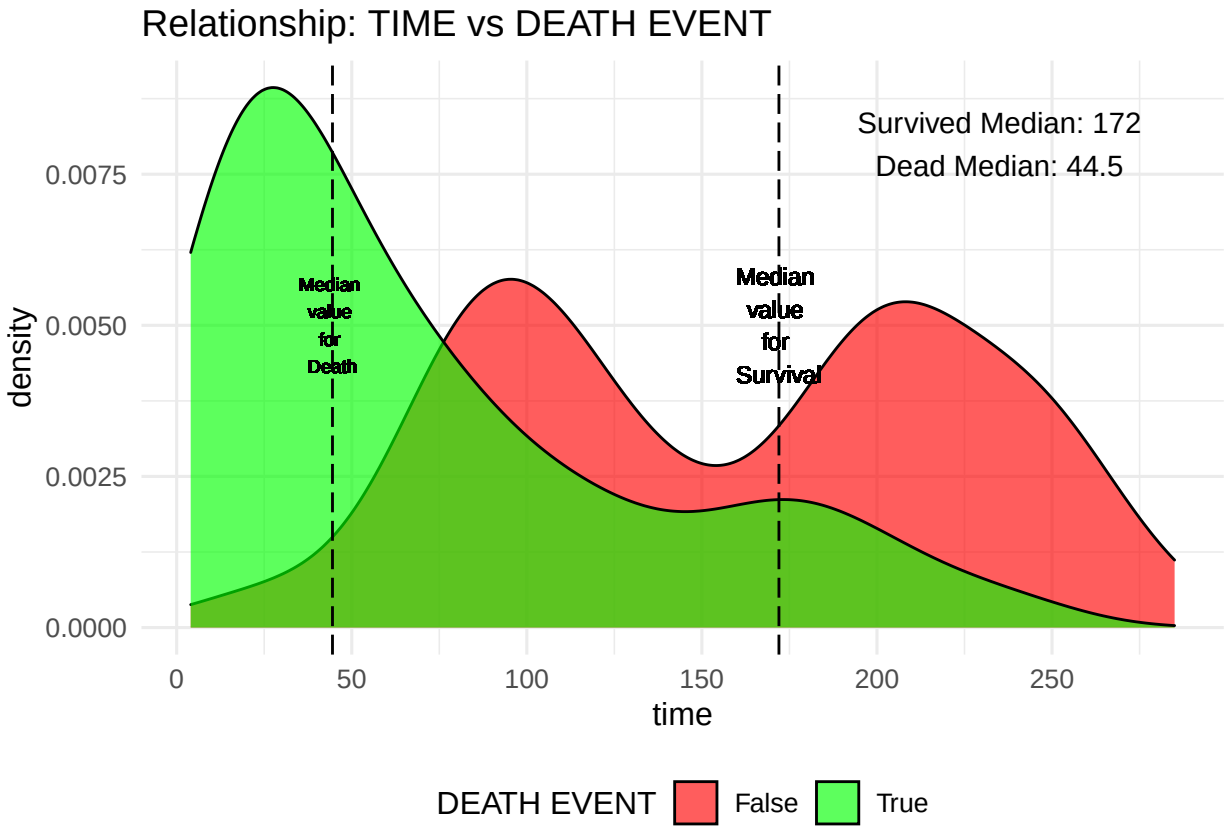
```
t <- ggplot(HF_data, aes(x = time)) +
  geom_histogram(binwidth = 10, colour = "white", alpha = 0.5) +
  geom_density(eval(bquote(aes(y = ..count.. * 10))), alpha = 0.25) +
  scale_x_continuous(breaks = seq(0, 300, 50)) +
  geom_vline(xintercept = median(HF_data$time), linetype="longdash") +
  annotate(geom = "text",
           x = max(HF_data$time)-30, y = 22,
           label = str_c("Min.      : ", min(HF_data$time),
                         "\nMedian : ", median(HF_data$time),
                         "\nMean    : ", round(mean(HF_data$time), 1),
                         "\nMax.     : ", max(HF_data$time))) +
  labs(title = "Distribution of TIME") +
  theme_minimal(base_size = 12)

s <- ggplot(HF_data, aes(x = time, fill = DEATH_EVENT)) +
  geom_density(alpha = 0.64) +
  scale_fill_manual(values = c("red", "green"),
                    name = "DEATH EVENT",
                    labels = c("False", "True")) +
  scale_x_continuous(breaks = seq(0, 300, 50)) +

  geom_vline(xintercept = median(filter(HF_data, DEATH_EVENT == 0)$time), linetype="longdash") +
  geom_vline(xintercept = median(filter(HF_data, DEATH_EVENT == 1)$time), linetype="longdash") +
```

```
  geom_text(label = "Median \nvalue \nfor \nSurvival", x = 172, y = 0.005, size = 3) +
  geom_text(label = "Median \nvalue \nfor \nDeath", x = 44.5, y = 0.005, size = 2.5) +
  annotate(geom = "text",
           x = max(HF_data$time)-50, y = 0.008,
           label = str_c("Survived Median: ", median(filter(HF_data, DEATH_EVENT == 0)$time),
                         "\nDead Median: ", median(filter(HF_data, DEATH_EVENT == 1)$time))) +

  labs(title = "Relationship: TIME vs DEATH EVENT") +
  theme_minimal(base_size = 12) +
  theme(legend.position = "bottom", legend.direction = "horizontal")
```

```
t
```




```
s
```

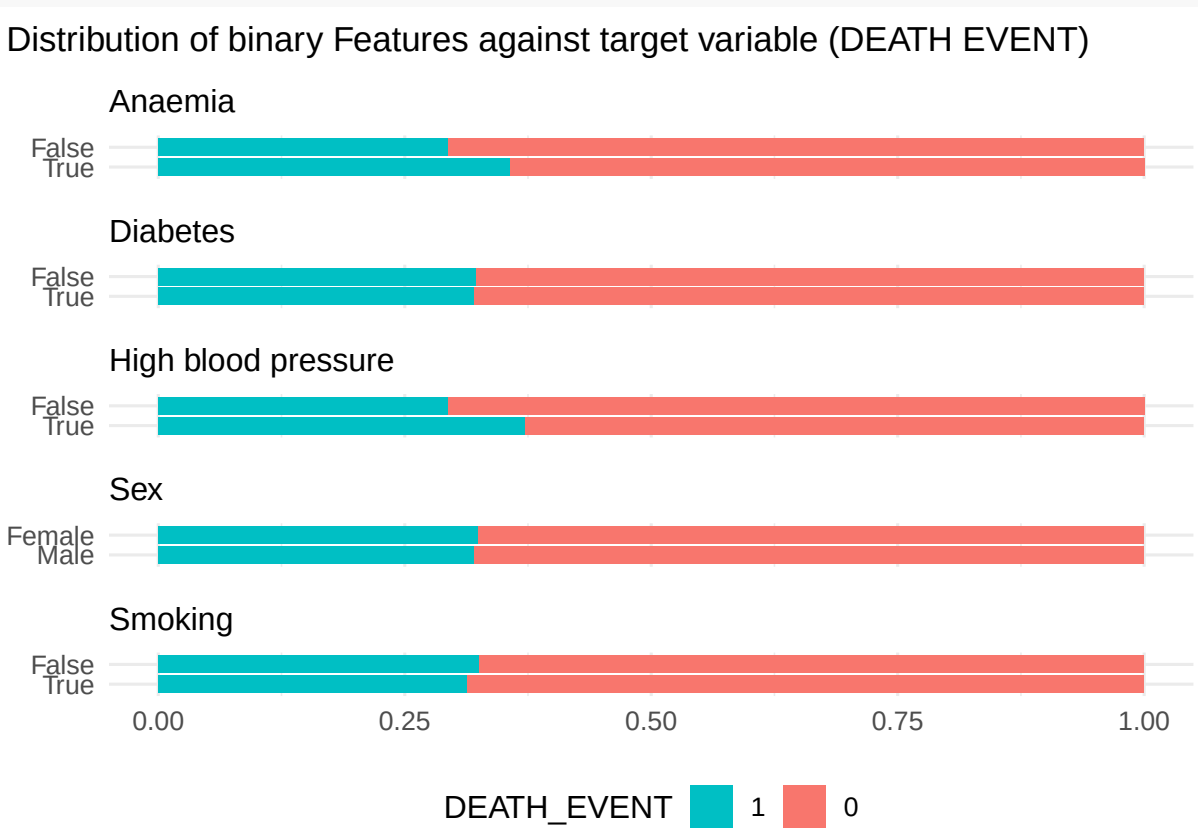


Observation: The distribution of follow-up periods is spread out. Peaks are not as loud as compared to some of the other variables. We also observe differences in the median 172 and 44.5 respectively. Patients that survive have long but gradual follow-up periods as compared to that of dead patients where follow-up periods are short.

Distribution of binary Features against target variable (DEATH EVENT)

```
library(ggplot2)
library(ggthemes)
library(patchwork)
one <- ggplot(HF_data, aes(y = reorder(anaemia, as.numeric(anaemia) * -1), fill = DEATH_EVENT)) +
  geom_bar(position = "fill", show.legend = FALSE) +
  scale_y_discrete(labels  = c("True", "False"))+
  labs(subtitle = "Anaemia") +
  theme_minimal(base_size = 12) +
  theme(axis.title = element_blank(), axis.text.x = element_blank())

two <- ggplot(HF_data, aes(y = reorder(diabetes, as.numeric(diabetes) * -1), fill = DEATH_EVENT)) +
  geom_bar(position = "fill", show.legend = FALSE) +
  scale_y_discrete(labels  = c("True", "False")) +
  labs(subtitle = "Diabetes") +
  theme_minimal(base_size = 12) +
  theme(axis.title = element_blank(), axis.text.x = element_blank())

three <- ggplot(HF_data, aes(y = reorder(high_blood_pressure, as.numeric(high_blood_pressure) * -1), fi
  geom_bar(position = "fill", show.legend = FALSE) +
  scale_y_discrete(labels  = c("True", "False")) +
```

```
  labs(subtitle = "High blood pressure") +
  theme_minimal(base_size = 12) +
  theme(axis.title = element_blank(), axis.text.x = element_blank())

four <- ggplot(HF_data, aes(y = reorder(sex, as.numeric(sex) * -1), fill = DEATH_EVENT)) +
  geom_bar(position = "fill", show.legend = FALSE) +
  scale_y_discrete(labels  = c("Male", "Female")) +
  labs(subtitle = "Sex") +
  theme_minimal(base_size = 12) +
  theme(axis.title = element_blank(), axis.text.x = element_blank())

five <- ggplot(HF_data, aes(y = reorder(smoking, as.numeric(smoking) * -1), fill = DEATH_EVENT)) +
  geom_bar(position = "fill", show.legend = TRUE) +
  scale_y_discrete(labels  = c("True", "False")) +
  labs(subtitle = "Smoking") +
  theme_minimal(base_size = 12) +
  theme(axis.title = element_blank(), legend.position = "bottom", legend.direction = "horizontal") +
  guides(fill = guide_legend(reverse = TRUE))

(one + two + three + four + five + plot_layout(ncol = 1)) +
  plot_annotation(title = "Distribution of binary Features against target variable (DEATH EVENT)")
```



We observe from the plot of distribution of binary features versus DEATH_EVENT that the difference between diabetes, sex and smoking is very small with respect to the target variable DEATH_EVENT. Meanwhile, that's not the case for anemia and high blood pressure. We observe some difference in their distribution with respect to the target variable DEATH_EVENT. As to whether the difference is significant or not, we are gonna find out.

Correlation Matrix

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
HF.binary.features <- c("anaemia", "diabetes", "high_blood_pressure", "sex", "smoking", "DEATH_EVENT")
heart_failure_data <- heart_failure_data %>% mutate_at(HF.binary.features, as.factor)

# corrplot(cor(heart_failure_data), type = "upper", method="shade", order = "original", addCoef.col = T
```

DATA SPLITTING INTO TRAINING AND TESTING DATA SETS

```
library(rsample)
library(readr)
set.seed(555)

# dataH <- read_csv(str_c("heart_failure_clinical_records_dataset.csv"))
heart_failure_data.split <- initial_split(heart_failure_data, prop = 0.8, strata = DEATH_EVENT)
train.heart_failure_data <- training(heart_failure_data.split)
test.heart_failure_data <- testing(heart_failure_data.split)
head(train.heart_failure_data)
```

```
##    age anaemia creatinine_phosphokinase diabetes ejection_fraction
## 1:  49       1                      80        0                30
## 2:  65       1                      52        0                25
## 3:  53       0                      63        1                60
## 4:  50       1                     159        1                30
## 5:  60       0                    2656        1                30
## 6:  72       0                     127        1                50
##    high_blood_pressure platelets serum_creatinine serum_sodium sex smoking time
## 1:                   1    427000              1.0          138   0       0   12
## 2:                   1    276000              1.3          137   0       0   16
## 3:                   0    368000              0.8          135   1       0   22
## 4:                   0    302000              1.2          138   0       0   29
## 5:                   0    305000              2.3          137   1       0   30
## 6:                   1    218000              1.0          134   1       0   33
##    DEATH_EVENT
## 1:           0
## 2:           0
## 3:           0
## 4:           0
## 5:           0
## 6:           0
```

```
head(test.heart_failure_data)
```

```
##    age anaemia creatinine_phosphokinase diabetes ejection_fraction
## 1:  75       0                     582        0                20
## 2:  65       0                     146        0                20
## 3:  60       1                     315        1                60
## 4:  80       1                     123        0                35
## 5:  50       1                     168        0                38
## 6:  95       1                     112        0                40
##    high_blood_pressure platelets serum_creatinine serum_sodium sex smoking time
## 1:                   1    265000              1.9          130   1       0    4
```

```
## 2:                   0    162000              1.3          129   1       1    7
## 3:                   0    454000              1.1          131   1       1   10
## 4:                   1    388000              9.4          133   1       1   10
## 5:                   1    276000              1.1          137   1       0   11
## 6:                   1    196000              1.0          138   0       0   24
##    DEATH_EVENT
## 1:           1
## 2:           1
## 3:           1
## 4:           1
## 5:           1
## 6:           1
```

```
HF_data.split <- initial_split(HF_data, prop = 0.8, strata = DEATH_EVENT)
train.HF_data <- training(HF_data.split)
test.HF_data <- testing(HF_data.split)
head(train.HF_data)
```

```
##    age anaemia creatinine_phosphokinase diabetes ejection_fraction
## 15  49       1                      80        0                30
## 21  65       1                      52        0                25
## 24  53       0                      63        1                60
## 34  50       1                     159        1                30
## 44  72       0                     127        1                50
## 58  60       1                     607        0                40
##    high_blood_pressure platelets serum_creatinine serum_sodium sex smoking time
## 15                   1    427000              1.0          138   0       0   12
## 21                   1    276000              1.3          137   0       0   16
## 24                   0    368000              0.8          135   1       0   22
## 34                   0    302000              1.2          138   0       0   29
## 44                   1    218000              1.0          134   1       0   33
## 58                   0    216000              0.6          138   1       1   54
##    DEATH_EVENT
## 15           0
## 21           0
## 24           0
## 34           0
## 44           0
## 58           0
```

```
head(test.HF_data)
```

```
##    age anaemia creatinine_phosphokinase diabetes ejection_fraction
## 9   65       0                     157        0                65
## 14  50       1                     168        0                38
## 17  87       1                     149        0                38
## 30  82       0                      70        1                30
## 31  94       0                     582        1                38
## 39  60       0                    2656        1                30
##    high_blood_pressure platelets serum_creatinine serum_sodium sex smoking time
## 9                    0    263358             1.50          138   0       0   10
## 14                   1    276000             1.10          137   1       0   11
## 17                   0    262000             0.90          140   1       0   14
## 30                   0    200000             1.20          132   1       1   26
## 31                   1    263358             1.83          134   1       0   27
```

```
## 39                   0    305000             2.30          137    1        0    30
##    DEATH_EVENT
## 9            1
## 14           1
## 17           1
## 30           1
## 31           1
## 39           0
```

RANDOM FOREST CLASSIFICATION

```
set.seed(555)
library(ranger) # Random Forest library
library(caret)  # Create Confusion Matrix
```

```
## Loading required package: lattice
```

```
RandF <- ranger(DEATH_EVENT ~ age + serum_creatinine + ejection_fraction,
             data = train.heart_failure_data,
             mtry = 3, num.trees = 400,
             write.forest = T, importance = "permutation")

pred.RandF <- predict(RandF, data = test.heart_failure_data)$predictions
confusionMatrix(pred.RandF, factor(test.heart_failure_data$DEATH_EVENT), positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 34  7
##          1  7 13
##
##                Accuracy : 0.7705
##                  95% CI : (0.645, 0.8685)
##     No Information Rate : 0.6721
##     P-Value [Acc > NIR] : 0.06369
##
##                   Kappa : 0.4793
##
##  Mcnemar's Test P-Value : 1.00000
##
##             Sensitivity : 0.6500
##             Specificity : 0.8293
##          Pos Pred Value : 0.6500
##          Neg Pred Value : 0.8293
##              Prevalence : 0.3279
##          Detection Rate : 0.2131
##    Detection Prevalence : 0.3279
##       Balanced Accuracy : 0.7396
##
##        'Positive' Class : 1
##
```

```
# Binary variables
RandF.binary <- ranger(DEATH_EVENT ~.,
             data = train.heart_failure_data,
```

```
            mtry = 2, num.trees = 400,
            write.forest = T, importance = "permutation")

pred.RandF.binary <- predict(RandF.binary, data = test.heart_failure_data)$predictions
confusionMatrix(pred.RandF.binary, factor(test.heart_failure_data$DEATH_EVENT), positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 40  6
##          1  1 14
##
##                Accuracy : 0.8852
##                  95% CI : (0.7778, 0.9526)
##     No Information Rate : 0.6721
##     P-Value [Acc > NIR] : 0.0001138
##
##                   Kappa : 0.7218
##
##  Mcnemar's Test P-Value : 0.1305700
##
##             Sensitivity : 0.7000
##             Specificity : 0.9756
##          Pos Pred Value : 0.9333
##          Neg Pred Value : 0.8696
##              Prevalence : 0.3279
##          Detection Rate : 0.2295
##    Detection Prevalence : 0.2459
##       Balanced Accuracy : 0.8378
##
##        'Positive' Class : 1
##
```

DECISION TREE

```
set.seed(444)
library(rpart)      # for recursive partitioning and regression trees
library(rpart.plot) # generates plots for recursive partitioning and regression trees
# fit a rpart model
DTree.fit <- rpart(DEATH_EVENT ~ .,
             data = train.heart_failure_data, method = "class",
             control=rpart.control(minsplit=10, minbucket=5, maxdepth=10, cp=0.03))
```

LOGISTIC REGRESSION

```
# logit.fit = glm(DEATH_EVENT ~ ., data = train, family = "binomial")
#
# logit.predict = predict(logit.fit, newdata = test.x, type = "response")
#
# logit.predict = ifelse(logit.predict > 0.5,1,0)
# # confusionMatrix(as.factor(logit.predict), test.y$DEATH_EVENT)
#
# table(test.y$DEATH_EVENT,logit.predict)
# print(paste0("Accuracy of Logistic Regression is"," ", round(confusionMatrix(as.factor(logit.predict)
```

SUPPORT VECTOR MACHINE (SVM)

DECISION TREE

EXTREME GRADIENT BOOST

SURVIVAL ANALYSIS

K-NEAREST NEIGHBORS (KNN)