CHAPTER 1: INTRODUCTION

Exploratory data analysis (EDA) is a very important step which takes place after feature engineering and acquiring data and it should be done before any modeling. This is because it is very important for a data scientist to be able to understand the nature of the data without making assumptions.

The purpose of EDA is to use descriptive statistics and visualizations to better understand data, and find clues about the tendencies of the data, its quality and to formulate assumptions and the hypothesis of our analysis. EDA is NOT about making fancy visualizations or even aesthetically pleasing ones; the goal is to try and answer questions with data. My goal is that I should be able to create a figure which someone can look at in a couple of seconds and understand what is going on. If not, the visualization is too complicated (or fancy) and something similar should be used.

EDA is also very iterative since we first make assumptions based on our first exploratory visualizations, then build some models. We then make visualizations of the model results and tune our models.

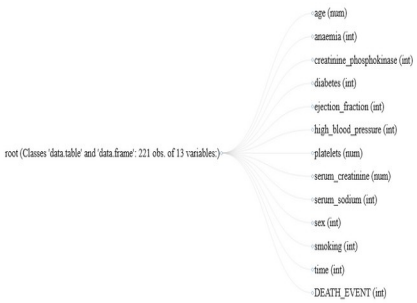
Objective and Aim:

The Aim of the project is to explore a dataset of my choosing and get a deep insight into the dataset. The dataset used for this project is "**heart_failure_patients**". The primary objective of my project is to uncover the underlying structure in the dataset. From the structure, we should be able to determine the trends, patterns and relationships among the attributes of the dataset using descriptive statistics and visualization. After performing the exploration of the dataset and uncovering the underlying structure and relationships in the dataset, I will be supposing a general form for the relationships (which is going to be my regression model). The regression model will be used to predict a dependent variable based on a set of explanatory variables "independent variables" related to the dependent variable. At the end of the project, I hope to have the answer to a burning question; "Why does the dependent variable take different values for different sample/population of data?".

CHAPTER 2: UNDERSTANDING THE DATA SET – DATASET DESCRIPTION:

Earlier, I mentioned the dataset used for this project is the "**heart_failure_patients**" dataset. This dataset was gotten from the UCI Machine learning repository. This dataset contains the medical records of heart failure patients. It has 299 instances/samples and 13 variables/attributes (3 continuous attributes, 9 discrete(integer) attributes and 1 target variable). Below is a brief description of the 13 variables in the data set:

1. age: age of the patient (years)
2. anaemia: decrease of red blood cells or hemoglobin (boolean)
3. creatinine_phosphokinase: level of the CPK enzyme in the blood (mcg/L)
4. diabetes: if the patient has diabetes (boolean)
5. ejection_fraction: percentage of blood leaving the heart at each contraction (percentage)
6. high_blood_pressure: if the patient has hypertension (boolean)
7. platelets: platelets in the blood (kiloplatelets/mL)
8. serum_creatinine: level of serum creatinine in the blood (mg/dL)
9. serum_sodium: level of serum sodium in the blood (mEq/L)
10. sex: woman or man (binary)
11. smoking: if the patient smokes or not (boolean)
12. time: follow-up period (days)
13. DEATH_EVENT: if the patient deceased during the follow-up period



CHAPTER 3: EXPLORATORY DATA ANALYSIS OF MY DATA SET:

Brief Summary:

Exploratory data analysis is a statistical approach for analyzing data sets in order to summarize their important and main characteristics generally by using some visual aids. EDAs are used to gather information and knowledge about the characteristics and features of the data sets, the variables/attributes, the relationships between the attributes, how to solve any underlying problem within the data sets. EDAs helps us to visualize and transform data as well as refining our initial set of questions by generating a new set of questions.

Data Inspection and Summarization of our data set:

First, let's begin with some basic numerical summaries. we always want to begin with a basic summary of the data. This step will always include the use of understanding our variables and understanding the basic statistics of those variables. As mentioned earlier, we know that our dataset contains the medical record of 299 heart failure patients, the patients consisted of 105 women and 194 men, and their ages range between 40 and 95 years old. let's obtain the numerical summaries of each numerical column of data.

The structure of our data set is shown below.

```
> str(mydata)
'data.frame':   299 obs. of  13 variables:
 $ age                     : num  75 55 65 50 65 90 75 60 65 8
 $ anaemia                 : int  0 0 0 1 1 1 1 1 0 1 ...
 $ creatinine_phosphokinase: int  582 7861 146 111 160 47 246
 $ diabetes                : int  0 0 0 0 1 0 0 1 0 0 ...
 $ ejection_fraction       : int  20 38 20 20 20 40 15 60 65 3
 $ high_blood_pressure     : int  1 0 0 0 0 1 0 0 0 1 ...
 $ platelets               : num  265000 263358 162000 210000
 $ serum_creatinine        : num  1.9 1.1 1.3 1.9 2.7 2.1 1.2
 $ serum_sodium            : int  130 136 129 137 116 132 137
 $ sex                     : int  1 1 1 1 0 1 1 1 0 1 ...
 $ smoking                 : int  0 0 1 0 0 1 0 1 0 1 ...
 $ time                    : int  4 6 7 7 8 8 10 10 10 10 ...
 $ DEATH_EVENT             : int  1 1 1 1 1 1 1 1 1 1 ...
```

From the figure above, we can conclude that our data set contains 3 continuous variables (age, platelets, serum creatinine) and 10 discrete variables (anaemia, diabetes, sex, smoking, time, etc.……).

Using the ***summary()*** function, we can get a generalized form of the descriptive statistics of every variable in the data set as shown below.

```
> summary(mydata)
      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        DEATH_EVENT
 Min.   :113.0   Min.   :0.0000   Min.   :0.0000   Min.   :  4.0   Min.   :0.0000
 1st Qu.:134.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 73.0   1st Qu.:0.0000
 Median :137.0   Median :1.0000   Median :0.0000   Median :115.0   Median :0.0000
 Mean   :136.6   Mean   :0.6488   Mean   :0.3211   Mean   :130.3   Mean   :0.3211
 3rd Qu.:140.0   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:203.0   3rd Qu.:1.0000
 Max.   :148.0   Max.   :1.0000   Max.   :1.0000   Max.   :285.0   Max.   :1.0000
>
```

Descriptive Statistics of the data set:

The best course of action is to obtain the summary information for all the variables one after the other. However, the ***summary()*** function falls short in some certain aspect as it does not provide us with things like number of observations, standard deviation, the variance and whether there are missing values in the data. Using another package in R, I was able to get a detailed summary of the descriptive statistics of each variable.

```
> options(scipen = 999)
> ageStat <- round(basicStats(mydata
> names(ageStat)<- c("Summary of age
> ageStat
            Summary of age variable
nobs                            299
NAs                               0
Minimum                          40
Maximum                          95
1. Quartile                      51
3. Quartile                      70
Mean                             61
Median                           60
Sum                           18189
SE Mean                           1
LCL Mean                         59
UCL Mean                         62
Variance                        141
Stdev                            12
Skewness                          0
Kurtosis                          0
>
```

From the figure above, we can see that the mean of the age variable is 61, median is 60, the 1st quartile is 51, 3rd quartile is 70, the variance and the standard deviation are both given as 141 and 12 respectively. There are no missing values in the age column.

The summary of the remaining variables is given below.

```
            Summary of anaemia variable
nobs                            299
NAs                               0
Minimum                           0
Maximum                           1
1. Quartile                       0
3. Quartile                       1
Mean                              0
Median                            0
Sum                             129
SE Mean                           0
LCL Mean                          0
UCL Mean                          0
Variance                          0
Stdev                             0
Skewness                          0
Kurtosis                         -2
>
```

```
            Summary of creatinine variable
nobs                            299
NAs                               0
Minimum                          23
Maximum                        7861
1. Quartile                     116
3. Quartile                     582
Mean                            582
Median                          250
Sum                          173970
SE Mean                          56
LCL Mean                        471
UCL Mean                        692
Variance                     941459
Stdev                           970
Skewness                          4
Kurtosis                         25
>
```

```
            Summary of diabetes variable
nobs                            299
NAs                               0
Minimum                           0
Maximum                           1
1. Quartile                       0
3. Quartile                       1
Mean                              0
Median                            0
Sum                             125
SE Mean                           0
LCL Mean                          0
UCL Mean                          0
Variance                          0
Stdev                             0
Skewness                          0
Kurtosis                         -2
>
```

```
        Summary of high blood pressure variable
nobs                            299.00
NAs                               0.00
Minimum                           0.00
Maximum                           1.00
1. Quartile                       0.00
3. Quartile                       1.00
Mean                              0.35
Median                            0.00
Sum                             105.00
SE Mean                           0.03
LCL Mean                          0.30
UCL Mean                          0.41
Variance                          0.23
Stdev                             0.48
Skewness                          0.62
Kurtosis                         -1.62
>
```

Summary of platelets variable

nobs	299.00
NAs	0.00
Minimum	25100.00
Maximum	850000.00
1. Quartile	212500.00
3. Quartile	303500.00
Mean	263358.03
Median	262000.00
Sum	78744050.75
SE Mean	5656.17
LCL Mean	252226.94
UCL Mean	274489.12
Variance	9565668749.45
Stdev	97804.24
Skewness	1.45
Kurtosis	6.03

>

Summary of smoking variable

nobs	299.00
NAs	0.00
Minimum	0.00
Maximum	1.00
1. Quartile	0.00
3. Quartile	1.00
Mean	0.32
Median	0.00
Sum	96.00
SE Mean	0.03
LCL Mean	0.27
UCL Mean	0.37
Variance	0.22
Stdev	0.47
Skewness	0.76
Kurtosis	-1.42

>

Summary of serum sodium variable

nobs	299.00
NAs	0.00
Minimum	113.00
Maximum	148.00
1. Quartile	134.00
3. Quartile	140.00
Mean	136.63
Median	137.00
Sum	40851.00
SE Mean	0.26
LCL Mean	136.12
UCL Mean	137.13
Variance	19.47
Stdev	4.41
Skewness	-1.04
Kurtosis	3.98

>

Summary of time variable

nobs	299.00
NAs	0.00
Minimum	4.00
Maximum	285.00
1. Quartile	73.00
3. Quartile	203.00
Mean	130.26
Median	115.00
Sum	38948.00
SE Mean	4.49
LCL Mean	121.43
UCL Mean	139.09
Variance	6023.97
Stdev	77.61
Skewness	0.13
Kurtosis	-1.22

>

Summary of sex variable

nobs	299.00
NAs	0.00
Minimum	0.00
Maximum	1.00
1. Quartile	0.00
3. Quartile	1.00
Mean	0.65
Median	1.00
Sum	194.00
SE Mean	0.03
LCL Mean	0.59
UCL Mean	0.70
Variance	0.23
Stdev	0.48
Skewness	-0.62
Kurtosis	-1.62

>

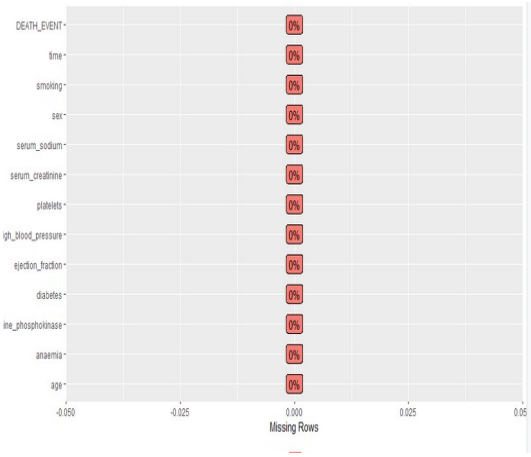
Summary of death event variable

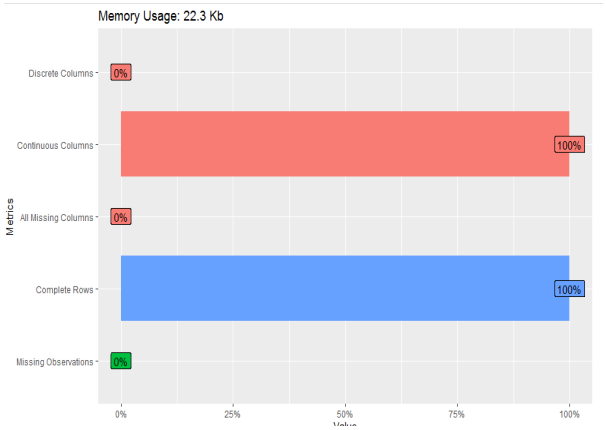
nobs	299.00
NAs	0.00
Minimum	0.00
Maximum	1.00
1. Quartile	0.00
3. Quartile	1.00
Mean	0.32
Median	0.00
Sum	96.00
SE Mean	0.03
LCL Mean	0.27
UCL Mean	0.37
Variance	0.22
Stdev	0.47
Skewness	0.76
Kurtosis	-1.42

>

Missing Values:

Since there are no missing data in any variable column, next step, we check for the presence of outliers in our data set. Using the ***boxplot()*** function, we can determine the presence of outliers and eliminate them from our data set.

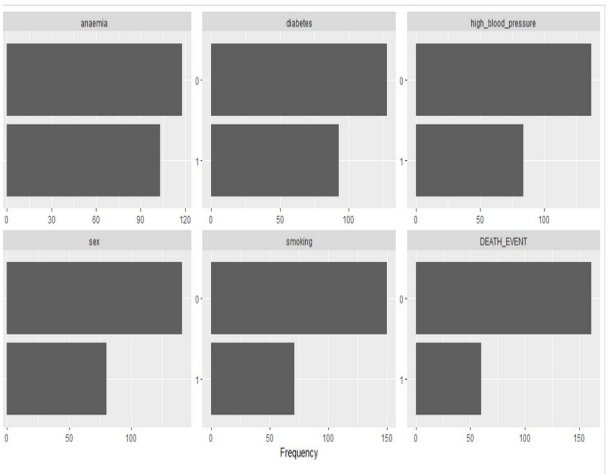




Graphical Representation of my data sets:

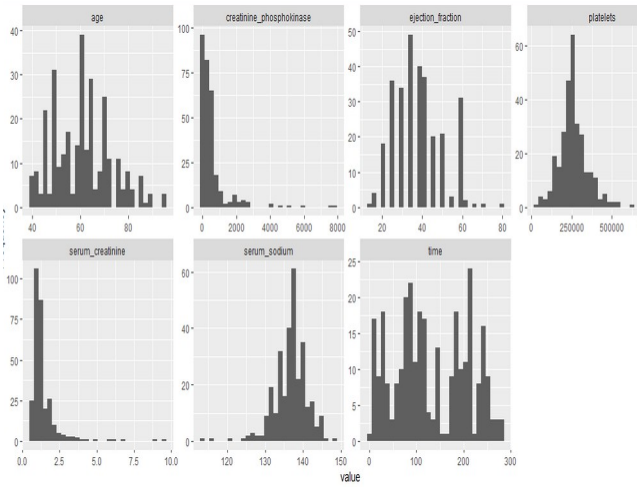
Visualizing Discrete variables:

Categorical variables are known as discrete variables, but since no categorical variable exist in our data set, integers can be thought as discrete. Our current data set contain some discrete variables (i.e. variables with just two values, 0s and 1s), Visualizing the frequency distributions for these discrete features is shown below.



Visualizing the continuous variable in the data set:

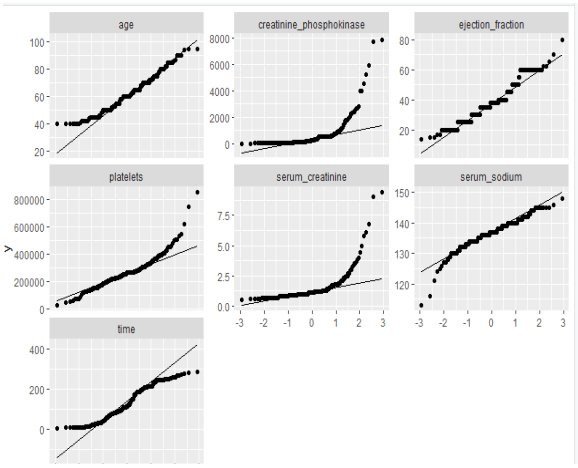
To visualize distributions for all continuous features, we plot the histogram for the numeric variables



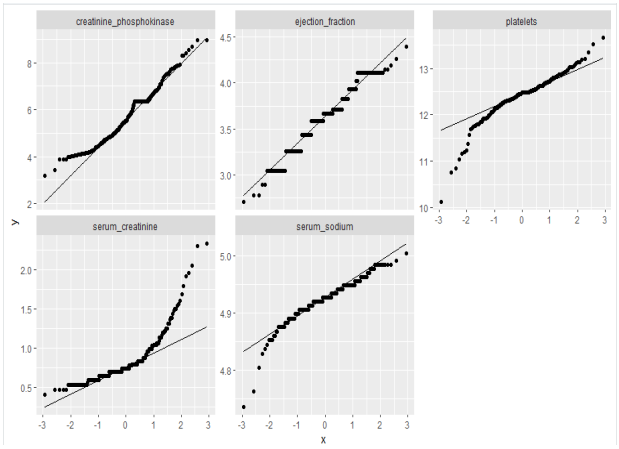
Quantile-Quantile plot for the numeric features:

Quantile-Quantile plot visualizes the deviation from a specific probability distribution. After analyzing these plots, it is often beneficial to apply mathematical transformation (such as log) for models like linear regression.

QQ-plot for our numeric features is shown below.

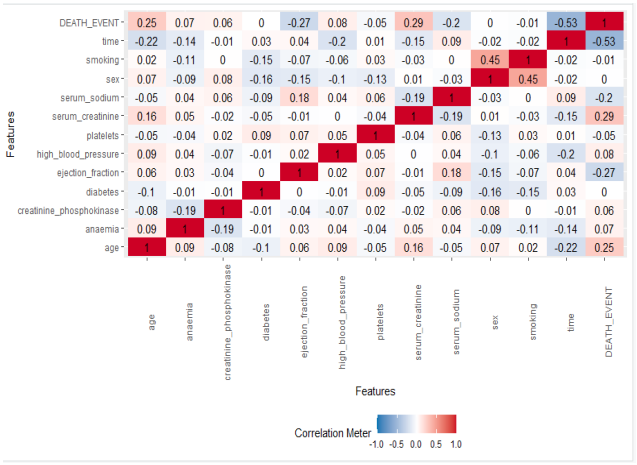


From the plot, we can see that there is a certain skewness on both ends in the creatinine_phosphokinase, platelets, ejection_fraction, serum_creatinine, serum_sodium variables respectively. In order to reduce the skewness, we use a simple log function to transform it and then we plot the qq-plot again.



Correlation plot and analysis:

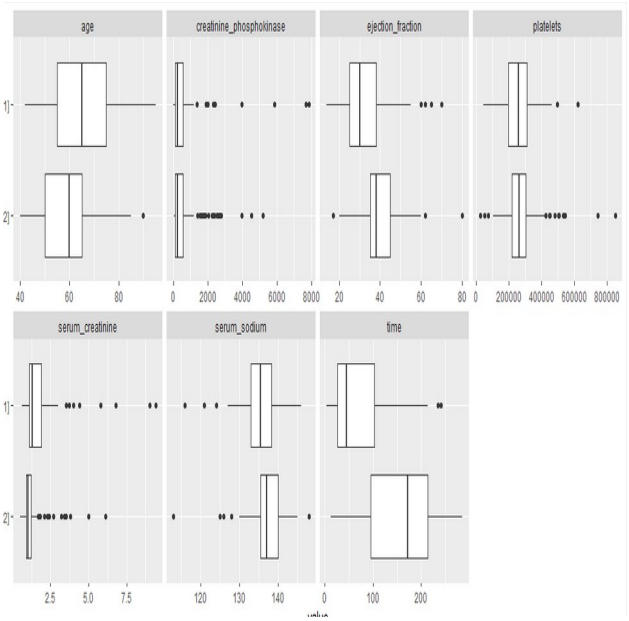
To check if a correlation exists between any variable, we plot a correlation plot for our dataset. This correlation helps us visualize if there is a significant relationship between any variables. A correlation value close to -1 shows a strong negative relationship, while a correlation value close to 1 shows a strong positive relationship. Correlation value close to 0 represents a weak relationship. The correlation plot for our data set is given below.



There is a slightly positive relationship between the sex variable and the smoking variable, while there exists a slightly negative correlation between the time period and the death event variable. The rest correlations are quite insignificant compared to those two.

Boxplot Representation of the Data set:

Visualizing the distribution of all continuous features based on death event with a boxplot:



CHAPTER 4: Data Preprocessing

Min-Max Normalization:

Normalizing variables in a dataset simply means to scale the values such that the variable has a mean of 0 and a standard deviation of 1. In order to understand the relationship between several predictor variables and a response variable, normalization is very important.

Min-Max Normalization of my data set will yield the following results.

```
> #define Min-Max normalization function
> min_max_norm <- function(x) {
+     (x - min(x)) / (max(x) - min(x))
+ }
> mydata <- as.data.frame(lapply(df[1:13], min_max_norm))
> head(mydata)
        age anaemia creatinine_phosphokinase diabetes
1 0.6363636       0              0.071319214        0
2 0.2727273       0              1.000000000        0
3 0.4545455       0              0.015692779        0
4 0.1818182       1              0.011227354        0
5 0.4545455       1              0.017478949        1
6 0.9090909       1              0.003062006        0
  ejection_fraction high_blood_pressure platelets serum_creatin
1        0.09090909                   1 0.2908231       0.15730
2        0.36363636                   0 0.2888326       0.06741
3        0.09090909                   0 0.1659595       0.08988
4        0.09090909                   0 0.2241484       0.15730
5        0.09090909                   0 0.3659838       0.24719
6        0.39393939                   1 0.2168748       0.17977
  serum_sodium sex smoking        time DEATH_EVENT
1   0.48571429   1       0 0.000000000           1
2   0.65714286   1       0 0.007117438           1
3   0.45714286   1       1 0.010676157           1
4   0.68571429   1       0 0.010676157           1
5   0.08571429   0       0 0.014234875           1
6   0.54285714   1       1 0.014234875           1
> |
```

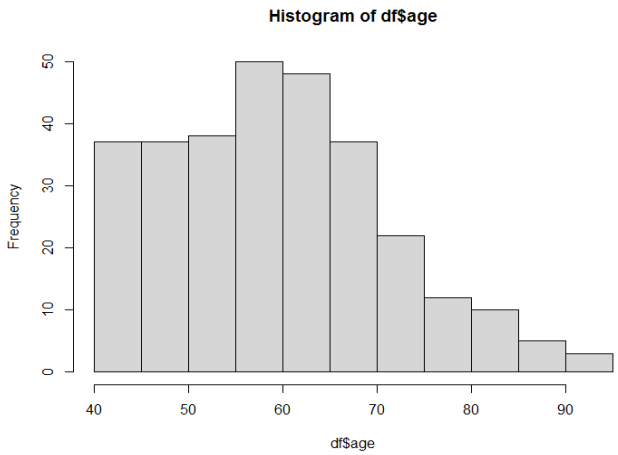
From the figure above, the values of each column range between 0 and 1 with a mean value of 0 and a standard deviation of 1. we can be sure that each variable contributes equally to the analysis.

Z-Score Standardization of my data set:

The drawback of the min-max normalization technique is that it brings the data values towards the mean. If we want to make sure that outliers get weighted more than other values, a z-score standardization is a better technique to implement. Z-score standardization of my data set yields the following results.

```
> #standardization of my dataset
> mydata <- as.data.frame(scale(df[1:13]))
> head(mydata)
         age    anaemia creatinine_phosphokinase   diabetes
1  1.1909487 -0.8696469              0.000165451 -0.8461608
2 -0.4904571 -0.8696469              7.502062717 -0.8461608
3  0.3502458 -0.8696469             -0.449185725 -0.8461608
4 -0.9108085  1.1460462             -0.485257493 -0.8461608
5  0.3502458  1.1460462             -0.434757017  1.1778559
6  2.4520030  1.1460462             -0.551217299 -0.8461608
  ejection_fraction high_blood_pressure          platelets
1       -1.527997920           1.3569966  0.016788339527581
2       -0.007064906          -0.7344569  0.000000007523048
3       -1.527997920          -0.7344569 -1.036335771428699
4       -1.527997920          -0.7344569 -0.545559486711210
5       -1.527997920          -0.7344569  0.650707707287672
6        0.161927651           1.3569966 -0.606906522300896
  serum_creatinine serum_sodium        sex    smoking       time
1       0.48923681   -1.50151891  0.7344569 -0.686531 -1.626775
2      -0.28407611   -0.14173853  0.7344569 -0.686531 -1.601007
3      -0.09074788   -1.72814897  0.7344569  1.451727 -1.588122
4       0.48923681    0.08489153  0.7344569 -0.686531 -1.588122
5       1.26254973   -4.67433977 -1.3569966 -0.686531 -1.575238
6       0.68256504   -1.04825878  0.7344569  1.451727 -1.575238
  DEATH_EVENT
1    1.451727
2    1.451727
3    1.451727
4    1.451727
5    1.451727
6    1.451727
> |
```

Binning the Variable “age” in my dataset:



```
> hist(df$age)
> df$age[1:10]
 [1] 75 55 65 50 65 90 75 60 65 80
> myBins <- cut(df$age, breaks=c(0,50,55,60,65,70,100), labels =
 "B", "C", "D", "E", "F"))
> myBins[1:10]
 [1] F B D A D F F C D F
Levels: A B C D E F
>
```

The Bins 0, 50, 55, 60, 65, 70, 100 were created with 0 being the minimum and 100 being the maximum, and each bin has a label A, B, C, D, E, F respectively. A represents category < 50, B represent category 50 – 55, C represent category 55 -60, D represent category 60 – 65, E represents category 65 – 70, and F represent category 70 – 100. Take for instance, the first value of the age variable 75 falls in the F category, therefore it can be replaced with F as shown in the result above. Hence, we have successfully converted a numerical variable to a categorical variable by binning using the cut command.

Another way we can bin continuous variables into discrete is using quantiles. Each bin to have the same number of observations.

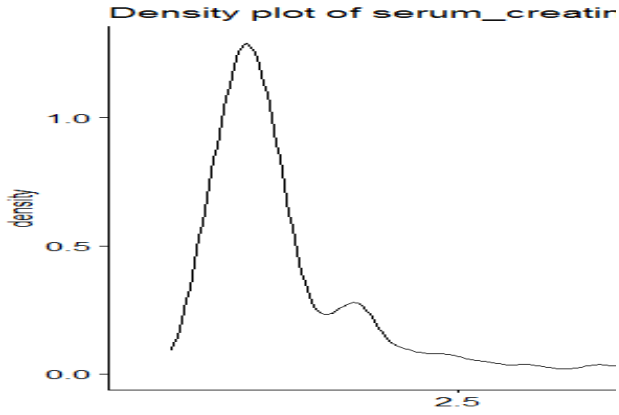
```
(MyQuantileBins = cut(df$age,
                      breaks = unique(quantile(df$age
mbers_of_bins))),
                      include.lowest=TRUE))

_phosphokinase diabetes ejection_fraction
           582         0                20
          7861         0                38
           146         0                20
           111         0                20
           160         1                20
            47         0                40
           246         0                15
           315         1                60
           157         0                65
           123         0                35
atelets serum_creatinine serum_sodium sex
 265000               1.9          130   1
 263358               1.1          136   1
 162000               1.3          129   1
 210000               1.9          137   1
 327000               2.7          116   0
 204000               2.1          132   1
 127000               1.2          137   1
 454000               1.1          131   1
 263358               1.5          138   0
 388000               9.4          133   1
NT MyQuantileBins
 1         (70,95]
 1         (51,60]
 1         (60,70]
 1         [40,51]
 1         (60,70]
 1         (70,95]
 1         (70,95]
 1         (51,60]
 1         (60,70]
 1         (70,95]
```

Natural log transformation of the "serum_creatinine" variable:

Visualization:

Initially, the serum_creatinine variable was positively skewed with a skewness of 4.44. Application of the Normal log transformation fixed this skewness to a substantial level as shown below.

```
skewness(df$serum_creatinine, na.rm = TRUE)
1] 4.43361
```



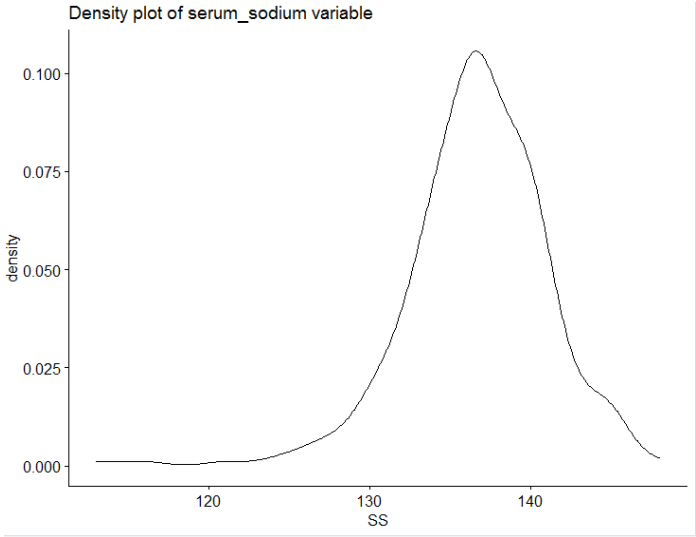
The log10 transformation improves the distribution back to normality.

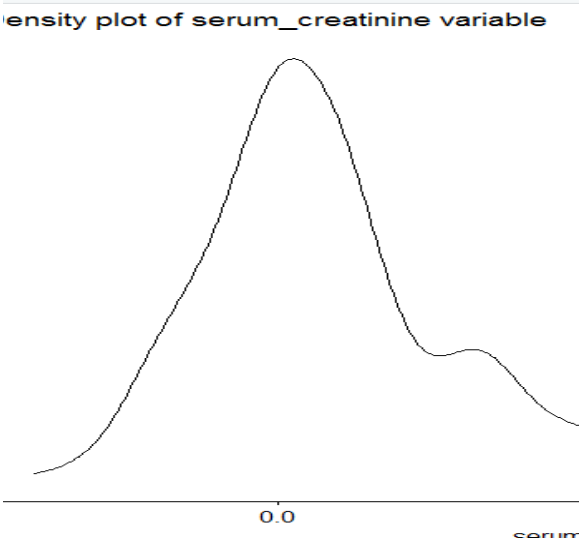
Square root transformation of the serum_sodium variable:

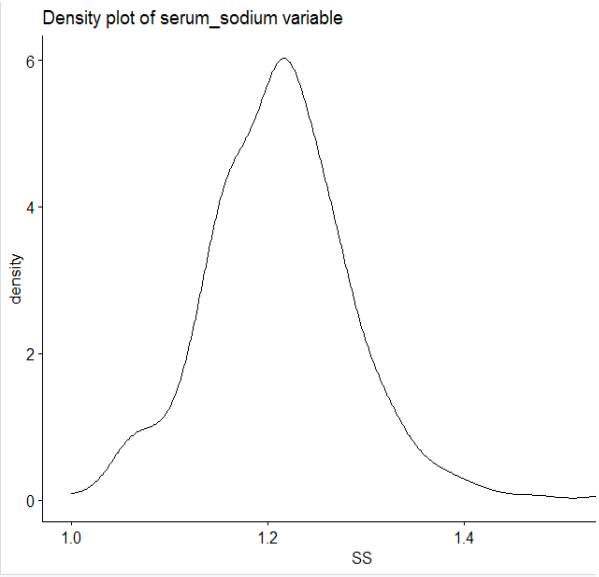
The serum_sodium variable was negatively skewed with a negative skewness of about -1.04, Applying the Square root transformation improves it to normality.

```
> x <- df$serum_sodium
> ggdensity(x,
+           main = "Density plot of serum_sodium variable"
+           xlab = "SS")
> skewness(x)
[1] -1.04287
>
```

```
> #Normal log transformation of the variable
> df$serum_creatinine <- log10(df$serum_creatinine)
> ggdensity(df$serum_creatinine,
+           main = "Density plot of serum_creatinine var
+           xlab = "serum")
> skewness(df$serum_creatinine)
[1] 1.576032
>
```



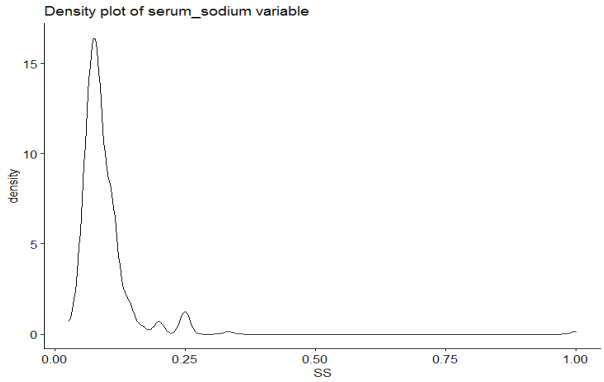





```
> #Square root transformation of the serum_sodium variable
> x <- sqrt(max(x+1)-x)
> ggdensity(x,
+           main = "Density plot of serum_sodium variable",
+           xlab = "SS")
> skewness(x)
[1] 0.7491935
>
```

Applying the inverse log transformation to the negatively skewed serum_sodium variable yields a positively skewed density plot.

```
> #Inverse log transformation of the serum_sodium variable
> s <- df$serum_sodium
> s <- 1/(max(s+1)-s)
> ggdensity(s,
+           main = "Density plot of serum_sodium variable",
+           xlab = "SS")
> skewness(s)
[1] 8.689482
>
```



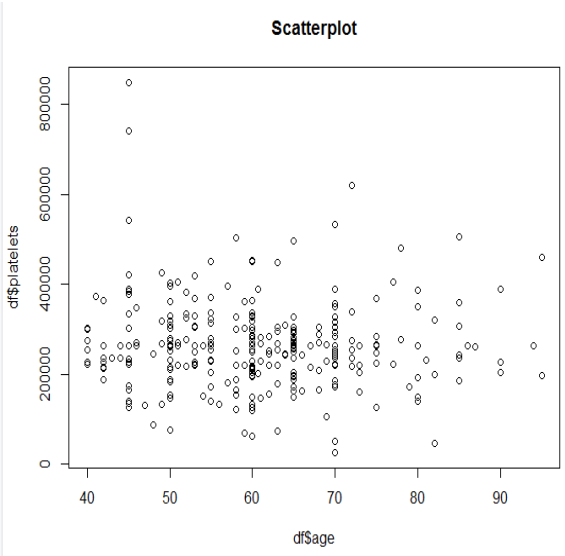
Inverse square root does not improve the data to normality.

CHAPTER 5: REGREssION ANALYSIS:

Modelling the relationship between the Age of a patient and the count of platelets in their blood:

In this situation, the dependent variable (Y) or outcome is the count of blood platelets, while Age is the X variable.

We begin by creating a scatter plot and calculating the Pearson's coefficient for the data to better visualize the scenario.



```
> class(df$age)
[1] "numeric"
> class(df$platelets)
[1] "numeric"
> plot(df$age, df$platelets, main = "Scatterplot")
> cor(df$age, df$platelets)
[1] -0.05235437
>
```

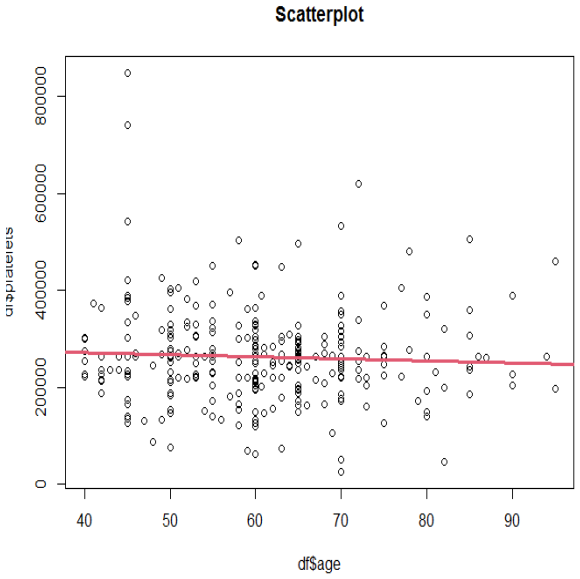
From the above coefficient calculation, we can see that there is a slightly linear negative relationship between the age of the individual and the count of platelets in the blood. Fitting this relationship into a model yields the following

```
> model <- lm(df$platelets ~ df$age)
> summary(model)

Call:
lm(formula = df$platelets ~ df$age)

Residuals:
    Min      1Q  Median      3Q     Max
-234312  -52147   -6022   38061  579826

Coefficients:
            Estimate Std. Error t value            Pr(>|t|)
(Intercept) 289545.8    29531.9   9.804 <0.0000000000000002 ***
df$age        -430.5      476.5  -0.903               0.367
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 97830 on 297 degrees of freedom
Multiple R-squared:  0.002741,  Adjusted R-squared:  -0.0006168
F-statistic: 0.8163 on 1 and 297 DF,  p-value: 0.367
```

Adding the regression line for a better understanding, we clearly see the line sloping downwards towards the right from the left which indicates a negative relationship between the two attributes.

What does this mean? it means that an increase in the age of a patient results in a decrease in the platelet count of the individual and vice versa.

```
> abline(model)
> abline(model, col=2, lwd=3)
>
```



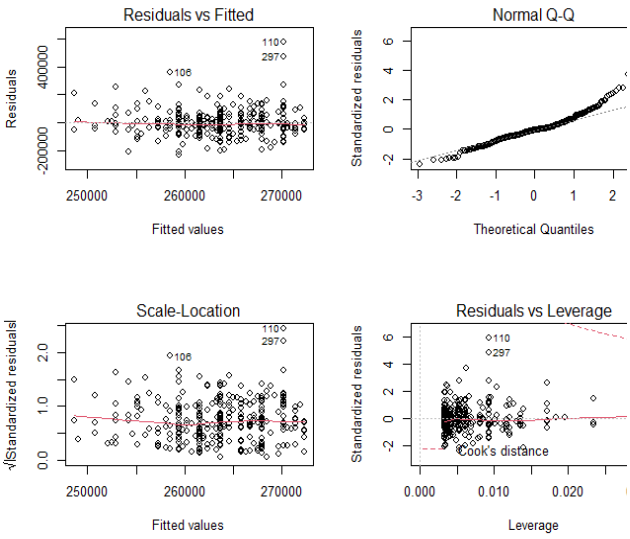
Assumptions made:

From the above model, three assumptions were made;

1. The Y values can be expressed as a linear function of the X variable i.e. the platelet count can be expressed as a linear function of age.
2. Variation of observations around the regression line (the residual SE) is constant.
3. For a given value of X, Y values are normally distributed.

To check on the Validity of my assumptions, we ran a diagnostic test on the model.

```
> par(mfrow=c(2,2))
>
> plot(model)
>
```



- The first plot is the residual diagnostic plot, we can see it is a little bit concise in the middle, this suggests that variance is varies, larger predicted values are associated with larger errors or residuals. The most important aspect of this plot is the red line, we can see that the red line is flat which suggests that the linearity assumption is met.
- The second plot is known as a QQ plot, the y axis is the ordered observed standardized residuals while the x axis is the ordered theoretical residuals. If the points fall roughly on a diagonal line, it indicates that the Y values are normally distributed. Our plot indicates a normally distributed y value. This proves the normally distributed assumption checks out.
- The third plot shows the variance is slightly constant. The red line indicates linearity.

CONCLUSION:

The question to be answered here is, does age determine the count of blood platelets in an individual? The answer is YES. From the analysis, we can see a negative linear relationship exists between age and blood platelets count. As age increases, platelets count tends to decrease, therefore we can assume or make a hypothetical conclusion that age is a major determinant in platelet count reduction. However, this explanation is still hypothetical as many other physiology variables and factors a huge role alongside age in determining platelet count. Further investigation is required to clarify this analysis.