**Exploratory Data Analysis in R: Data Summarising, Visualization, and Predictive Model**

**Extracting Meaning of a Dataset**

Mulher com flor amarela

Descrição gerada automaticamente

Photo by [Courtney Cook](https://unsplash.com/@courtneymcook?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com?utm_source=medium&utm_medium=referral)

[Rashida Nasrin Sucky](https://rashida00.medium.com/?source=post_page-----da79439b3952--------------------------------)

May 26·2021

Exploratory data analysis is unavoidable to understand any dataset. It includes data summarization, visualization, some statistical analysis, and predictive analysis. This article will focus on data storytelling or exploratory data analysis using R and different packages of R.

This article will cover:

1. The Summarization and Visualization of Some Key Points
2. Some Basic Statistics
3. Predictive Model

If you are a regular follower of my articles, you might have seen another exploratory data analysis project using the same dataset before in Python. Here is the [link](https://towardsdatascience.com/exploratory-data-analysis-visualization-and-prediction-model-in-python-241b954e1731).

I am using the same dataset here for performing an exploratory data analysis in R. The approach is different though.

Let’s start! [Feel free to download the dataset from this link](https://www.kaggle.com/andrewmvd/heart-failure-clinical-data).

## Summarizing and Visualization

I have the dataset in the same folder as my RStudio file. So, here I can read the CSV file in the RStudio:

df = read.csv("heart\_failure\_clinical\_records\_dataset.csv")

I am not showing a screenshot because the dataset is pretty big. These are the columns in the dataset:

colnames(df)

Output:

[1] "age" "anaemia" "creatinine\_phosphokinase" "diabetes" "ejection\_fraction"   
 [6] "high\_blood\_pressure" "platelets" "serum\_creatinine" "serum\_sodium" "sex"   
[11] "smoking" "time" "DEATH\_EVENT"

The dataset has different health parameters, age, sex, and the DEATH\_EVENT. There are many ways to approach a dataset. I want to start by seeing a correlation plot. It will be nice to start by seeing which variables are correlated. It requires ‘corrplot’ library.

library(corrplot)

corrplot(cor(df), type = "upper")

Gráfico, Linha do tempo, Gráfico de dispersão

Descrição gerada automaticamente

I will use this plot as a guide to decide the next steps.

We have six continuous variables in this dataset. Let’s check the distribution of them. First, those six variables will be separated as a different dataset, and then using the base R, histograms will be made:

df1 = df[, c('age', 'creatinine\_phosphokinase', 'ejection\_fraction', 'platelets', 'serum\_creatinine', 'serum\_sodium')]

hist.data.frame(df1, title = "Histograms of all Numerical Variables")

Gráfico, Histograma

Descrição gerada automaticamente

The distribution provides an idea about these six variables. Having some data will be even more helpful. I will use the summary function that gives some basic statistical parameters.

summary(df1)

Output:

# age creatinine\_phosphokinase ejection\_fraction platelets serum\_creatinine

# Min. :40.00 Min. : 23.0 Min. :14.00 Min. : 25100 Min. :0.500

# 1st Qu.:51.00 1st Qu.: 116.5 1st Qu.:30.00 1st Qu.:212500 1st Qu.:0.900

# Median :60.00 Median : 250.0 Median :38.00 Median :262000 Median :1.100

# Mean :60.83 Mean : 581.8 Mean :38.08 Mean :263358 Mean :1.394

# 3rd Qu.:70.00 3rd Qu.: 582.0 3rd Qu.:45.00 3rd Qu.:303500 3rd Qu.:1.400

# Max. :95.00 Max. :7861.0 Max. :80.00 Max. :850000 Max. :9.400

# serum\_sodium

# Min. :113.0

# 1st Qu.:134.0

# Median :137.0

# Mean :136.6

# 3rd Qu.:140.0

# Max. :148.0

The same way the categorical variables are taken as a separate dataset for convenience:

df2 = df[, c('anaemia', 'diabetes', 'high\_blood\_pressure', 'sex', 'smoking', 'DEATH\_EVENT')]

head(df2)

Tabela

Descrição gerada automaticamente com confiança média

Though the correlation plot shows no correlation between DEATH\_EVENT and the ‘sex’ variable. I still want to see the number of death in males and females. Before that, it is a good idea to change the 0 and 1 of those columns to some meaningful strings.

df$death = ifelse(df$DEATH\_EVENT == 1, "Yes", "No")

df$sex = ifelse(df$sex == 1, "Male", "Female")table(df$sex, df$death)

Texto

Descrição gerada automaticamente

Overall, there are more males than females in the dataset. What is the percentage of males and females?

data = table(df$sex)

data1 = round(data/sum(data)\*100)

data1 = paste(names(data1), data1)

paste(data1, "%", sep = "")Output:

[1] "Female 35%" "Male 65%"

Just to learn more about the population in the dataset, let’s check the distribution of ages of male and female population separately:

ggplot(df, aes(x = age)) +   
 geom\_histogram(fill = "white", colour = "black") +   
 facet\_grid(sex ~ .)

Gráfico, Gráfico de barras

Descrição gerada automaticamente

The histograms above show that the male population is older in general than the female population.

Serum creatinine and serum sodium show some correlation. A scatter plot will show it clearly:

ggplot(df) + geom\_point(aes(x = serum\_creatinine, y = serum\_sodium, colour = death, shape = death)) + ggtitle("Serum Creatinine vs Serum Sodium")

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

Because of a few outliers, most of the data is clustered in one place. In this type of situation, a good practice is to check the same plot without those outliers as well.

df\_scr = df[df$serum\_creatinine < 4.0 & df$serum\_sodium > 120,]ggplot(df\_scr) + geom\_point(aes(x = serum\_creatinine, y = serum\_sodium, colour = death, shape = death)) + ggtitle("Serum Creatinine vs Serum Sodium")

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

The graph clearly shows the relation between serum sodium and serum creatinine. The dots are separated by death events using different colors and shapes.

The next plot is the ejection fraction vs death event. It will show a jitter plot which actually a modified version of the scatter plot. When one of the variables in the scatter plot is categorical, dots stay on a straight line and they overlap. It becomes hard to understand. That’s why in jitter plot they deflect the points a little bit that gives a better understanding of the data. Also, we will add a red dot that will show the mean of the data.

Tabela

Descrição gerada automaticamente com confiança média

If you notice I trimmed the data a little bit. I cut down two data points that were outliers.

The time variable has a strong correlation with death events. I would like to see a boxplot that will show the difference in ‘time’ for death events and no death events:

ggplot(df, aes(death, time))+geom\_point() + labs(title="Death Event with Time Variable Segregated by Gender",  
 x = "Death",  
 y = "Time") +  
 geom\_boxplot(fill='steelblue', col="black", notch=TRUE) + facet\_wrap(~ sex)

Gráfico

Descrição gerada automaticamente

Creatinine phosphokinase and death event has some correlation. The next plot explores that with histogram:

ggplot(df, aes(x=creatinine\_phosphokinase, fill=death)) + geom\_histogram(bins=20) + labs(title = "Distribution of Creatinine Phosphokinase", x = "Creatinine Phosphokinase", y = "Count")

Gráfico

Descrição gerada automaticamente

Anemia is also related to creatinine phosphokinase. The distribution can be segregated by anaemia to observe the distribution separately by anaemia:

ggplot(df, aes(x=creatinine\_phosphokinase, fill=death)) + geom\_histogram(bins=20)+facet\_wrap(~anaemia) + labs(title = "Distribution of Creatinine Phosphokinase", x = "Creatinine Phosphokinase", y = "Count")

Gráfico, Gráfico de barras, Histograma

Descrição gerada automaticamente

The distribution is quite different for the population with anaemia and without anaemia.

The ‘time’ variable has a strong correlation with death events. This age vs time scatter-plot shows death events with different colors.

ggplot(df, aes(x = age, y = time, col = death))+geom\_point() + labs(title = "Age vs Time", x = "Age", y = "Time")

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

The following plot shows the distribution of serum creatinine segregated by different colors for death events:

ggplot(df, aes(x = serum\_creatinine, fill = death))+geom\_histogram() + labs(title = "Distribution of Serum Creatinine different colors for death event", x = "Serum Creatinine", y = "Count")

Gráfico, Histograma

Descrição gerada automaticamente

A very few outliers are there at the far right. Let’s check the same distribution without those outliers:

df\_sc = df[df$serum\_creatinine < 4.0,]

ggplot(df\_sc, aes(x = serum\_creatinine, fill = death))+

geom\_histogram() +

labs(title = "Distribution of Serum Creatinine different colors for death event",

x = "Serum Creatinine", y = "Count")

Gráfico, Histograma

Descrição gerada automaticamente

The distribution is much more clear now.

The ejection fraction may be different for death events. It will also be interesting to see if ejection fraction is different based on gender as well.

ggplot(df, aes(death, ejection\_fraction, fill = as.factor(sex))) +

geom\_bar(stat = "summary", fun = "median", col = "black",

position = "dodge") +

geom\_point(position = position\_dodge(0.9)) +

labs(title = "Ejection Fraction per Death Event",

x = "Death", y = "Ejection Fraction")

Gráfico, Gráfico de caixa estreita

Descrição gerada automaticamente

Those black points show the variance in the data.

The next plot explores the relationship between Anaemia and creatinine phosphokinase separately for death events and no death events.

ggplot(df, aes(x = as.factor(anaemia), y = creatinine\_phosphokinase, fill = death)) +

geom\_violin() +

stat\_summary(aes(x= as.factor(anaemia), y = creatinine\_phosphokinase), fun = median, geom='point', colour = "red", size = 3)+

facet\_wrap(~death)+

geom\_jitter(position = position\_jitter(0.1), alpha = 0.2) +

labs(title = "Creatinine Phosphokinase for Anaemia State Segregated by Death Event", x = "Anaemia", y ="Creatinine Phosphokinase")

Gráfico

Descrição gerada automaticamente

In this plot, you can see several different angles at the same time. violin plot shows the distribution and outliers. At the same time, I put the jitter plot in it with the red dot that points to the median of the respective data.

**Predictive Model**

In this section, we will try to see if we can predict the death event using the other features in the dataset. First, we should check the proportions of the death events:

data = table(df$death)  
round(data/sum(data), 2)

Output:

No Yes   
0.68 0.32

For this predictive model, I will import the dataset one more time because I made several changes in the dataset I imported before.

df1 = read.csv("heart\_failure\_clinical\_records\_dataset.csv")

Now, I will split the dataset into training data and test data.

library(caTools)

set.seed(1243)  
split = sample.split(df1$DEATH\_EVENT, SplitRatio = 0.75)

This split should provide boolean values for each DEATH\_EVENT. 75% of them are True and 25% False. Based on this split value, training\_data and test\_data will be separated:

training\_data = subset(d, split == TRUE)  
test\_data = subset(d, split == FALSE)

Dataset is ready for the classifier. I choose a support vector machine classifier for this project.

library(e1071)

svm\_fit = svm(formula = DEATH\_EVENT ~ .,  
 data = training\_data,  
 type = 'C-classification',  
 kernel = 'linear'  
 )

Training data were fitted in the classifier and the classifier is trained now. Now the classifier can be tested with the test data. Using the test data, I will predict the DEATH\_EVENT. So, we need to exclude the DEATH\_EVENT from the test data.

y\_pred = predict(svm\_fit, newdata = test\_data[-13])

The prediction is done. Now let’s check the accuracy of the prediction. ConfusionMatrix function is used from the ‘caret’ package here. Because this ConfusionMatrix function provides so much information with just one line of code:

library(caret)

confusionMatrix(y\_pred, as.factor(test\_data$DEATH\_EVENT))

Output:

Confusion Matrix and StatisticsReference  
Prediction 0 1  
 0 47 5  
 1 4 19  
   
 Accuracy : 0.88   
 95% CI : (0.7844, 0.9436)  
 No Information Rate : 0.68   
 P-Value [Acc > NIR] : 5.362e-05   
   
 Kappa : 0.7212   
   
 Mcnemar's Test P-Value : 1   
   
 Sensitivity : 0.9216   
 Specificity : 0.7917   
 Pos Pred Value : 0.9038   
 Neg Pred Value : 0.8261   
 Prevalence : 0.6800   
 Detection Rate : 0.6267   
 Detection Prevalence : 0.6933   
 Balanced Accuracy : 0.8566   
   
 'Positive' Class : 0

Overall accuracy is 88%.

Look at the confusion matrix on top! It predicts 47 no death events accurately and 4 no death events were wrongly predicted as death events. On the other hand, 19 death events were predicted accurately and 5 death events were wrongly predicted as no death events.

Here true positive is 19, false positive is 4, and false negative is 5. Using this information F1 score is calculated below:

true\_positive/(true\_positive + 0.5\*(false\_positive + false\_negative)) = 0.81

I am not commenting on if the prediction model worked well or bad. It depends on the expectation and expectation varies project by project.

**Conclusion**

Exploratory data analysis is very dynamic. You can approach a dataset in so many different ways. This was one of my approaches. If you have checked my work on the same dataset in Python before, you will see a different approach. Please feel free to try out some other types of visualization and classifier of your choice on it. Thanks for reading!

Feel free to follow me on [Twitter](https://twitter.com/rashida048) and like my [Facebook](https://www.facebook.com/rashida.smith.161) page.