Hi there!

tl;dr: Exploratory data analysis (**EDA**) the very first step in a data project. We will create a code-template to achieve this with one function.

**Introduction**

EDA consists of univariate (1-variable) and bivariate (2-variables) analysis.  
In this post we will review some functions that lead us to the analysis of the first case.

* Step 1 – First approach to data
* Step 2 – Analyzing categorical variables
* Step 3 – Analyzing numerical variables
* Step 4 – Analyzing numerical and categorical at the same time

Covering some key points in a basic EDA:

* Data types
* Outliers
* Missing values
* Distributions (numerically and graphically) for both, numerical and categorical variables.

**Type of analysis results**

They can be two: informative or operative.

Informative – For example plots, or any long variable summary. We cannot filter data from it, but give us a lot of information at once. Most used on the **EDA** stage.

Operative – The results can be used to take an action directly on the data workflow (for example, selecting any variables whose percentage of missing values are below 20%). Most used in the **Data Preparation** stage.

**Setting-up**

Uncoment in case you don’t have any of these libraries:

# install.packages("tidyverse")

# install.packages("funModeling")

# install.packages("Hmisc")

A newer version of funModeling has been released on Ago-1, please update 

Now load the needed libraries…

library(funModeling)

library(tidyverse)

library(Hmisc)

**tl;dr (code)**

Run all the functions in this post in one-shot with the following function:

basic\_eda <- function(data)

{

glimpse(data)

df\_status(data)

freq(data)

profiling\_num(data)

plot\_num(data)

describe(data)

}

Replace data with *your* data, and that’s it!:

basic\_eda(my\_amazing\_data)

**Creating the data for this example**

Using the heart\_disease data (from funModeling package). We will take only 4 variables for legibility.

data=heart\_disease %>% select(age, max\_heart\_rate, thal, has\_heart\_disease)

**Step 1 – First approach to data**

Number of observations (rows) and variables, and a head of the first cases.

glimpse(data)

## Observations: 303

## Variables: 4

## $ age 63, 67, 67, 37, 41, 56, 62, 57, 63, 53, 57, ...

## $ max\_heart\_rate 150, 108, 129, 187, 172, 178, 160, 163, 147,...

## $ thal 6, 3, 7, 3, 3, 3, 3, 3, 7, 7, 6, 3, 6, 7, 7,...

## $ has\_heart\_disease no, yes, yes, no, no, no, yes, no, yes, yes,...

Getting the metrics about data types, zeros, infinite numbers, and missing values:

df\_status(data)

## variable q\_zeros p\_zeros q\_na p\_na q\_inf p\_inf type unique

## 1 age 0 0 0 0.00 0 0 integer 41

## 2 max\_heart\_rate 0 0 0 0.00 0 0 integer 91

## 3 thal 0 0 2 0.66 0 0 factor 3

## 4 has\_heart\_disease 0 0 0 0.00 0 0 factor 2

df\_status returns a table, so it is easy to keep with variables that match certain conditions like:  
+ Having at least 80% of non-NA values (p\_na < 20)  
+ Having less than 50 unique values (unique <= 50)

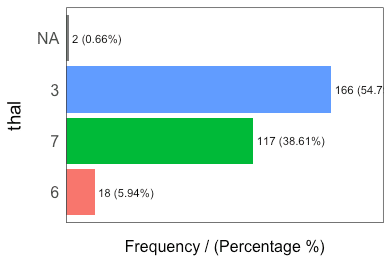
💡 TIPS:

* Are all the variables in the correct data type?
* Variables with lots of zeros or NAs?
* Any high cardinality variable?

**Step 2 – Analyzing categorical variables**

freq function runs for all factor or character variables automatically:

freq(data)



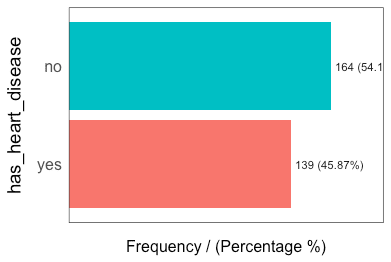
## thal frequency percentage cumulative\_perc

## 1 3 166 54.79 55

## 2 7 117 38.61 93

## 3 6 18 5.94 99

## 4 2 0.66 100



## has\_heart\_disease frequency percentage cumulative\_perc

## 1 no 164 54 54

## 2 yes 139 46 100

## [1] "Variables processed: thal, has\_heart\_disease"

TIPS:

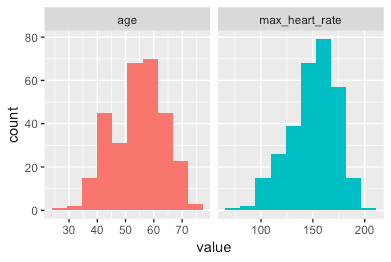
* If freq receives one variable –freq(data$variable)– it retruns a table. Useful to treat high cardinality variables (like zip code).
* Export the plots to jpeg into current directory: freq(data, path\_out = ".")
* Does all the categories make sense?
* Lots of missing values?
* Always check absolute and relative values

**Step 3 – Analyzing numerical variables**

We will see: plot\_num and profiling\_num. Both run automatically for all numerical/integer variables:

**Graphically**

plot\_num(data)



Export the plot to jpeg: plot\_num(data, path\_out = ".")

TIPS:

* Try to identify high-unbalanced variables
* Visually check any variable with outliers

**Quantitatively**

profiling\_num runs for all numerical/integer variables automatically:

data\_prof=profiling\_num(data)

## variable mean std\_dev variation\_coef p\_01 p\_05 p\_25 p\_50 p\_75 p\_95

## 1 age 54 9 0.17 35 40 48 56 61 68

## 2 max\_heart\_rate 150 23 0.15 95 108 134 153 166 182

## p\_99 skewness kurtosis iqr range\_98 range\_80

## 1 71 -0.21 2.5 13 [35, 71] [42, 66]

## 2 192 -0.53 2.9 32 [95.02, 191.96] [116, 176.6]

TIPS:

* Try to describe each variable based on its distribution (also useful for reporting)
* Pay attention to variables with high standard deviation.
* Select the metrics that you are most familiar with: data\_prof %>% select(variable, variation\_coef, range\_98): A high value in variation\_coef may indictate outliers. range\_98 indicates where most of the values are.

**Step 4 – Analyzing numerical and categorical at the same time**

describe from Hmisc package.

library(Hmisc)

describe(data)

## data

##

## 4 Variables 303 Observations

## ---------------------------------------------------------------------------

## age

## n missing distinct Info Mean Gmd .05 .10

## 303 0 41 0.999 54.44 10.3 40 42

## .25 .50 .75 .90 .95

## 48 56 61 66 68

##

## lowest : 29 34 35 37 38, highest: 70 71 74 76 77

## ---------------------------------------------------------------------------

## max\_heart\_rate

## n missing distinct Info Mean Gmd .05 .10

## 303 0 91 1 149.6 25.73 108.1 116.0

## .25 .50 .75 .90 .95

## 133.5 153.0 166.0 176.6 181.9

##

## lowest : 71 88 90 95 96, highest: 190 192 194 195 202

## ---------------------------------------------------------------------------

## thal

## n missing distinct

## 301 2 3

##

## Value 3 6 7

## Frequency 166 18 117

## Proportion 0.55 0.06 0.39

## ---------------------------------------------------------------------------

## has\_heart\_disease

## n missing distinct

## 303 0 2

##

## Value no yes

## Frequency 164 139

## Proportion 0.54 0.46

## ---------------------------------------------------------------------------

Really useful to have a quick picture for all the variables. But is not as operative as freq and profiling\_num when we want to use its results to change our data workflow.