**The problem**: Before modeling, we need to check/change numerical, categorical, NAs, one unique value and high cardinality variables.

The new version of funModeling 1.9.2 was released aimed to have assistance during the prior step in creating machine learning models.

**Introduction**

data\_integrity function provide information about the format of all the variables, as well as some short stats about NA values.

This way we can select and transform the variables, keeping them in the format we need.

# install.packages("funModeling")

library(funModeling)

**Load the *messy* data:**



library(tidyverse)

data=read\_delim("<https://raw.githubusercontent.com/pablo14/data-integrity/master/messy_data.txt>", delim = ';')

Now we call to data\_integrity function, which returns an integrity object:

di=data\_integrity(data)

Then, summary function gives us a quick self-explanatory overview :

summary(di)

##

## ◌ {Numerical with NA} num\_vessels\_flour, thal

## ◌ {Categorical with NA} gender

## ● {One unique value} constant

Now we can apply mutate\_at, select, or apply other function over certain and specific columns.

In case we need the variable name as a vector of strings, we can use the RStudio bare-combine add-in:

My keyboard shortcut for this lil' function gets quite the workout…  
"hrbraddins::bare\_combine()" by [@hrbrmstr](https://twitter.com/hrbrmstr?ref_src=twsrc%5Etfw) <https://t.co/8dwqNEso0B> [#rstats](https://twitter.com/hashtag/rstats?src=hash&ref_src=twsrc%5Etfw) [pic.twitter.com/gyqz2mUE0Y](https://t.co/gyqz2mUE0Y)

— Mara Averick (@dataandme) [July 29, 2019](https://twitter.com/dataandme/status/1155842512743030785?ref_src=twsrc%5Etfw)

The high cardinality max value can be changed using the parameter MAX\_UNIQUE

**Accessing all the information**

If we print the integrity object, we can see a lot of information regarding NA, numerical, categorical and other types, alongside the high cardinality variables:

di

## $vars\_num\_with\_NA

## variable q\_na p\_na

## 1 num\_vessels\_flour 4 0.01320132

## 2 thal 2 0.00660066

##

## $vars\_cat\_with\_NA

## variable q\_na p\_na

## 1 gender 1 0.00330033

##

## $vars\_cat\_high\_card

## [1] variable unique

## <0 rows> (or 0-length row.names)

##

## $MAX\_UNIQUE

## [1] 35

##

## $vars\_one\_value

## [1] "constant"

##

## $vars\_cat

## [1] "gender" "has\_heart\_disease"

##

## $vars\_num

## [1] "age" "chest\_pain" "resting\_blood\_pressure"

## [4] "serum\_cholestoral" "fasting\_blood\_sugar" "resting\_electro"

## [7] "max\_heart\_rate" "exer\_angina" "oldpeak"

## [10] "slope" "num\_vessels\_flour" "thal"

## [13] "heart\_disease\_severity" "exter\_angina" "constant"

## [16] "id"

##

## $vars\_char

## [1] "gender" "has\_heart\_disease"

##

## $vars\_factor

## character(0)

##

## $vars\_other

## [1] "has\_heart\_disease2" "fecha" "fecha2"

And each object is accessible to operate quickly:

di$results$vars\_num

## [1] "age" "chest\_pain" "resting\_blood\_pressure"

## [4] "serum\_cholestoral" "fasting\_blood\_sugar" "resting\_electro"

## [7] "max\_heart\_rate" "exer\_angina" "oldpeak"

## [10] "slope" "num\_vessels\_flour" "thal"

## [13] "heart\_disease\_severity" "exter\_angina" "constant"

## [16] "id"

Numerical variables with NA values:

di$results$vars\_num\_with\_NA$variable

## [1] "num\_vessels\_flour" "thal"



Help page:

help("data\_integrity")

**New status function**

This is the internal function used in data\_integrity:

status(heart\_disease)

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

## 1 age 0 0.0000000 0 0.00000000 0 0 integer 41

## 2 gender 0 0.0000000 0 0.00000000 0 0 factor 2

## 3 chest\_pain 0 0.0000000 0 0.00000000 0 0 factor 4

## 4 resting\_blood\_pressure 0 0.0000000 0 0.00000000 0 0 integer 50

## 5 serum\_cholestoral 0 0.0000000 0 0.00000000 0 0 integer 152

## 6 fasting\_blood\_sugar 258 0.8514851 0 0.00000000 0 0 factor 2

## 7 resting\_electro 151 0.4983498 0 0.00000000 0 0 factor 3

## 8 max\_heart\_rate 0 0.0000000 0 0.00000000 0 0 integer 91

## 9 exer\_angina 204 0.6732673 0 0.00000000 0 0 integer 2

## 10 oldpeak 99 0.3267327 0 0.00000000 0 0 numeric 40

## 11 slope 0 0.0000000 0 0.00000000 0 0 integer 3

## 12 num\_vessels\_flour 176 0.5808581 4 0.01320132 0 0 integer 4

## 13 thal 0 0.0000000 2 0.00660066 0 0 factor 3

## 14 heart\_disease\_severity 164 0.5412541 0 0.00000000 0 0 integer 5

## 15 exter\_angina 204 0.6732673 0 0.00000000 0 0 factor 2

## 16 has\_heart\_disease 0 0.0000000 0 0.00000000 0 0 factor 2

It’s another version of df\_status, where percentages are expressed in the range o 0 to 1 (not 0 to 100). More intuitive to use in filters

This is the same object as di$status\_now.

**Next realase?**

It will contain, based on data\_integrity, an automated data quality test suited for the predictive model we need to run.  
Found this task quite important and repetitive when I teach. Hopefully it will save some time!

**Further reading**

All of these topics are covered in deep in the *Data Science Live Book* :

* [Dataset status](https://livebook.datascienceheroes.com/exploratory-data-analysis.html#profiling)
* [Data types in predictive modeling](https://livebook.datascienceheroes.com/data-preparation.html#data_types)
* [High cardinallity variables](https://livebook.datascienceheroes.com/data-preparation.html#high_cardinality_predictive_modeling)
* [Handling Missing data](https://livebook.datascienceheroes.com/data-preparation.html#missing_data)