2 Hands On: Data Quality and Pre-Processing

1. Assessing Data Quality

Load the following packages: dplyr, na.tools, tidyimpute (*version from github decisionpatterns/tidyimpute"*)

Load the carInsurance data set about the insurance risk rating of cars based on several characteristics of each car[1]

(a) Check if there are any missing values.

Tip: use the function any_na().

(b) Count the number of cases that have, at least, one missing value.

Tip: use the function filter_any_na() and then count().

(c) Create a new data set by removing all the cases that have missing values.

Tip: use the function drop_rows_any_na()

(d) Create a new data set by imputing all the missing values with 0.

Tip: explore the variants of the function impute()

(e) Create a new data set by imputing the mean in all the columns which have double type values.

(f) Create a new data set by imputing the mode in all the columns which have integer type values.

(g) Create a new data set by imputing the most frequent value to the column ”nDoors”.

Tip: use the function impute_replace()

(h) Combine the three last imputations to obtain a final dataset. Are there any duplicated cases?

Tip: use the functions distinct() and count()

2. Data Pre-Processing

2. Load the package dlookr. Use the same car insurance data set above and apply the following transformations to the price attribute. Be critical regarding the obtained results.

(a) Apply range-based normalization and z-score normalization.

Tip: use the function transform().

(b) Discretize it into 4 equal-frequency ranges an into 4 equal-width ranges.

Tip: use the function binning().

3. With the seed 111019 obtain the following samples on the car insurance data set.

Tip: use the function sample_frac().

(a) A random sample of 60% of the cases, with replacement

(b) A stratified sample of 60% of the cases of cars, according to the fuelType attribute.

(c) Use the table() function to inspect the distribution of values in each of the two samples above.

[1] Detailed information on this can be found in here.

4. Load the package corrplot and select the numeric attributes of the car insurance data set.

(a) Using the function cor(), obtain the *pearson correlation coefficient* between each pair of variables.

(b) Apply the function cor.mtest() to the previous result to calculate the p-values and confidence intervals of the correlation coefficient for each pair of variables.

(c) Plot the all correlation information using the function corrplot. Explore some of its parameters.

5. Load the data set USJudgeRatings, from the datasets package, containing lawyers' ratings of state judges in the US Superior Court regarding a set of attributes.

(a) Apply the function prcomp() to obtain the principal components. Inspect how each variable is obtained by the linear combination of each component.

(b) Load the package ggbiplot and plot the two first components with the function ggbiplot(). You can label each point with the lawyer's name by setting the labels parameter.