# Exercise 1

אהרון שרים, 052328523

לביא ליזרוביץ, 065957383

using numpy, pandas, sklearn

# General look at the data

First thing we did was to run a crosstab of the features with the 'Vote' column, or the mean function per 'Vote' category

Some features show a difference in distribution in relation to the category, such as "Will\_only\_vote\_for\_large\_party":

|  |  |  |  |
| --- | --- | --- | --- |
| Large | Maybe | No | Yes |
| Vote |  |  |  |
| Blues | 0 | 0 | 904 |
| Browns | 0 | 1843 | 0 |
| Greens | 0 | 1617 | 0 |
| Greys | 547 | 0 | 0 |
| Oranges | 533 | 0 | 0 |
| Pinks | 0 | 1039 | 0 |
| Purples | 0 | 2057 | 0 |
| Reds | 536 | 0 | 0 |
| Whites | 0 | 335 | 0 |
| Yellows | 0 | 0 | 527 |

Or "AVG\_lottary\_expanses":

|  |  |
| --- | --- |
| Blues | 191621.1 |
| Browns | 52893.8 |
| Greens | 38129.14 |
| Greys | 61310.12 |
| Oranges | 55700.64 |
| Pinks | 71426.43 |
| Purples | 55050.97 |
| Reds | 58810.35 |
| Whites | 58480.08 |
| Yellows | 69470.84 |

While others show no immediate difference between voting categories, such as "Financial\_agenda\_matters":

|  |  |  |
| --- | --- | --- |
| Matters | No | Yes |
| Blues | 416 | 481 |
| Browns | 904 | 934 |
| Greens | 800 | 816 |
| Greys | 77 | 268 |
| Oranges | 257 | 271 |
| Pinks | 528 | 504 |
| Purples | 1006 | 1045 |
| Reds | 273 | 265 |
| Whites | 180 | 152 |
| Yellows | 251 | 279 |

Or "Financial\_balance\_score\_(0-1)":

|  |  |
| --- | --- |
| Blues | 0.499682 |
| Browns | 0.49922 |
| Greens | 0.501186 |
| Greys | 0.513089 |
| Oranges | 0.487941 |
| Pinks | 0.498352 |
| Purples | 0.501359 |
| Reds | 0.503831 |
| Whites | 0.50445 |
| Yellows | 0.502071 |

All in all, we've identified 18 features which showed some difference in distribution by the 'Vote' category, and decided we would probably want to keep all of them, and tread them differently when finding outliers and filling gaps.

Next, we run pairwise correlation between features. There was very low correlation between most of them, except for two pairs:

Yearly\_IncomeK | Avg\_size\_per\_room | 0.977327

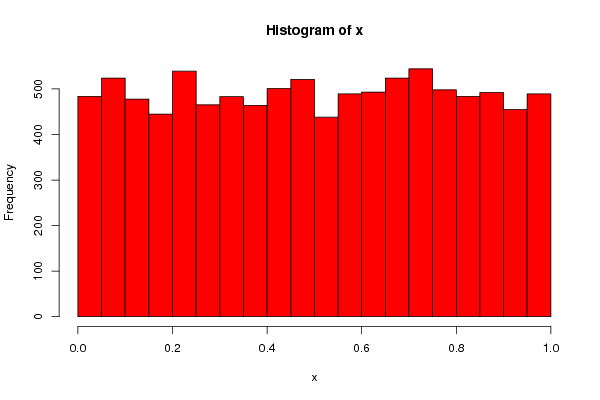
Garden\_sqr\_meter\_per\_person\_in\_residancy\_area | Avg\_monthly\_expense\_on\_pets\_or\_plants | 0.989534

We will use this correlation to fill gaps in the data, and then remove one of those features as it contains no new information.

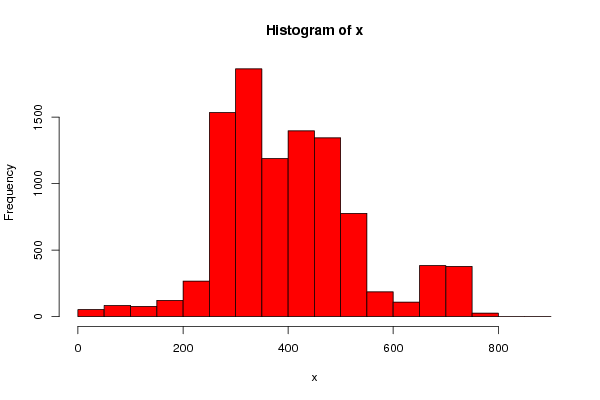
Next, we checked the distribution of each attribute using a quantile - quantile histogram with 20 bins.

Most of them "looked" liked Gaussian distribution, with a few that looked uniform. We will use this information later when choosing a method to scale the data.

For example "financial balance score" looks uniformly distributed:



While "Garden\_sqr\_meter\_per\_person\_in\_residancy\_area" looks almost normal:



# Removing outliers and filling nans

At first, we intended to remove outlier lines after classifying them as such using k-nearest-neighbors, but after realizing we were not supposed to remove any lines, we chose a different method:

Running on each numerical feature, we remove each value which is more than 3 standard deviations from the mean – category based mean if we identified it as meaningful at the first step, global mean if not.

Filling the missing cells is done in two steps. First, for the features we recognize as linear dependent, we use values in one, where available, to fill missing values in the other.

The second step fills the median/mode for each feature in the missing values space – and same as before, if we recognize the feature as one that has different distribution across classes, we use the class specific data point.

# Scaling the data

Here we decided to use different strategies based on the nature of the feature:

Those which were more or less uniform, and also in the range of 1-10, we scaled down by a factor of 10.

Percentile attributes were just scaled down to be fractions.

Features which we found to be nearly uniform, we use Min-Max to scale it to [0, 1]

All the rest of the numerical features are either normal-like or long tailed, so we use z-score to scale them to [-1, 1] , although all of them are positive.

# Feature selection

We used two filter methods and one wrapper method.

We filter the features using both correlation and mutual information vs the category, and take the top 60% feature.

We use backward propagation to find the best features that can fit the data using a support vector machine.

We then take the features we recognized as important in stage 1, and add to each any feature that was selected by the wrapper method and at least one filter method.