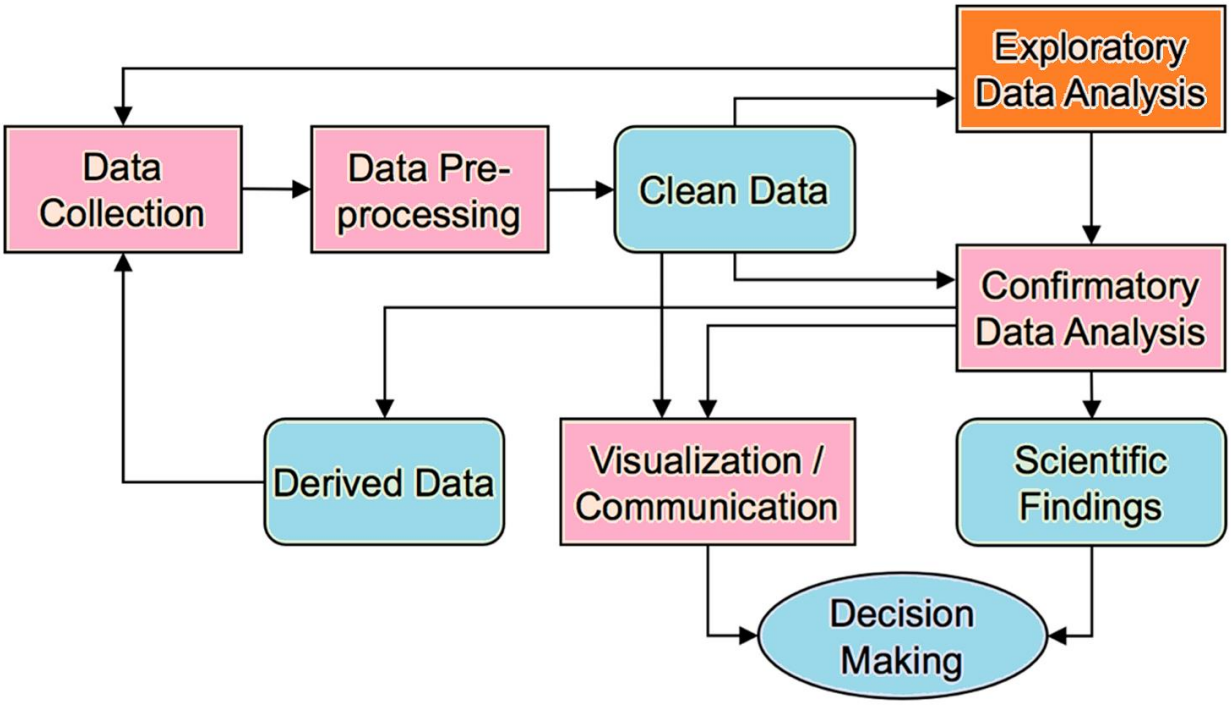
Chapter 1: Prepare the problem, Structural Investigation, Quantitative Investigation, Outliers, Content Investigation

Introduction

In this chapter, we will cover the topics related to Exploratory Data Analysis. But before that, we will be working on a problem statement. A problem will be addressed in terms of a use case scenario from a dataset. Then that dataset will be analyzed thoroughly.

Initial analysis will be done on the datasets to investigate some of the data features. EDA is a data analysis method that allows data scientists to explore and summarize the important characteristics of a dataset by visualizing the findings. EDA describes how to best manipulate the data sources. When used with big data, it allows data scientists to discover patterns, spot anomalies, and test assumptions. It can be viewed as:



In the upcoming exercises, we will be analyzing the dataset having price records of houses in Miami. Different features associated with the trend of prices along with the nature of data structure, content, and quality will be observed. Moreover, outliers will also be covered in the upcoming exercises.

You can learn more about the EDA from here: https://youtu.be/YRBdTw9TZPE

Investigations

- Structure Investigation.
- Quality Investigation.
- Content Investigation.

Preparing the problem

If you have a large dataset with more than 100 features, it will be overwhelming to try to generate visualizations for all of them. In such a case, variable selection will become very important. The EDA is often seen as the hardest, and most time-consuming part of the process. But it is highly recommended for anyone who's working on large data with more than 100 features to do variable selection first before jumping into EDA.

In this chapter, we will be exploring the dataset from Miami. The dataset contains information on 13,932 single-family homes sold in Miami in 2016. Besides publicly available information, the dataset creator Steven C. Bourassa has added distance variables, aviation noise as well as latitude and longitude. This dataset will be explored in terms of its features. Numeric and categorical variables will be extracted from it.

Exercise 1.01: Importing the Libraries and Loading the Data

In this exercise, we will import the essential libraries and load the Miami housing dataset. Initial data cleaning will be done here:

Open the following link to get started (Empty Jupyter Environment) or start with your own local or Hosted Jupyter environment:

https://mybinder.org/v2/gh/fenago/jupyter/HEAD

RESOURCES AND REFERENCES

The code is located here:

datawrangling/Exercise_1_01_Loading_Data.ipynb at main · fenago/datawrangling (github.com)

The code repository is located here:

datawrangling/Chapter 1 at main · fenago/datawrangling (github.com)

Structural Investigation

This step allows user to explore the general structure of a dataset and understand what types of features it contains.

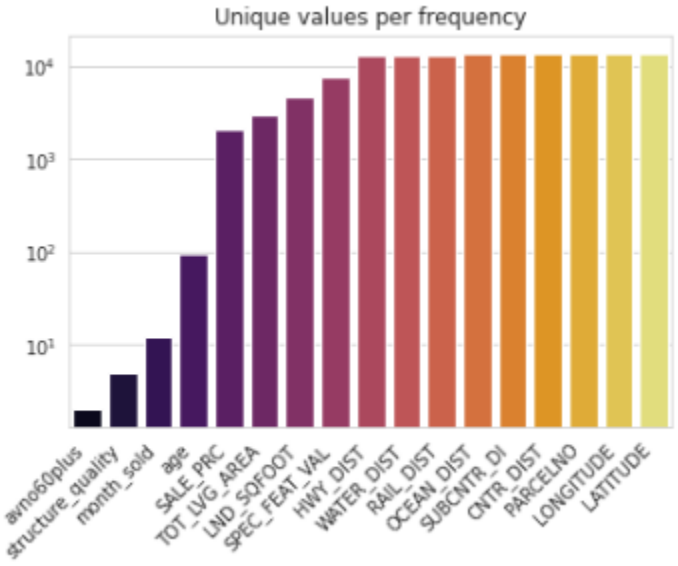
The structure of any dataset is important to understand the possible choices of feature extraction. The dimensions of columns and rows affects various features or attributes that can be used for classification or prediction, as well as other statistical analyses. We can also draw conclusions based on which columns are excluded from our data, e.g., if they are not represented at all in our dataset then we don't need to consider them when using those columns for classification or prediction. So, it is important to have a look at the general structure of the data.

This investigation covers these two parameters:

1. Structure of Non-numerical Features
2. Structure of Numerical Features

We will be covering structural investigation in the upcoming exercise.

The following output is obtained from the barplot, numerical features analysis having unique values per frequency:



The general structure thus obtained is as:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13932 entries, 0 to 13931
Data columns (total 17 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   LATITUDE           13932 non-null  float64
 1   LONGITUDE          13932 non-null  float64
 2   PARCELNO           13932 non-null  int64
 3   SALE_PRC           13932 non-null  float64
 4   LND_SQFOOT         13932 non-null  int64
 5   TOT_LVG_AREA       13932 non-null  int64
 6   SPEC_FEAT_VAL      13932 non-null  int64
 7   RAIL_DIST          13932 non-null  float64
 8   OCEAN_DIST         13932 non-null  float64
 9   WATER_DIST         13932 non-null  float64
 10  CNTR_DIST          13932 non-null  float64
 11  SUBCNTR_DI         13932 non-null  float64
 12  HWY_DIST           13932 non-null  float64
 13  age                13932 non-null  int64
 14  avno60plus         13932 non-null  int64
 15  month_sold         13932 non-null  int64
 16  structure_quality  13932 non-null  int64
dtypes: float64(9), int64(8)
memory usage: 1.8 MB
```

Exercise 1.02: Structure Investigation

Open the following link to get started (Empty Jupyter Environment) or start with your own local or Hosted Jupyter environment:

https://mybinder.org/v2/gh/fenago/jupyter/HEAD

RESOURCES AND REFERENCES

The code is located here:

datawrangling/Exercise_1_02_Structural_Investigation.ipynb at main · fenago/datawrangling (github.com)

The code repository is located here:

datawrangling/Chapter 1 at main · fenago/datawrangling (github.com)

Quality Investigation

The first step in any data analysis process is to perform the checks of quality on the data beforehand. In this case, we will be focusing on the overall quality of dataset. Before looking into the details, we can look at some general statistics of this dataset and check if there are any unexpected entries or duplicate rows.

Removal of Duplicates

Duplicates are entries that represent the same sample point multiple times. It is common when two people take a measurement at the same location of your dataset, or even when two recordings are made by different people under similar conditions at the same time. Ways exist to eliminate duplicates from your records: delete all duplicates in a dataset and/or detect duplicates only as necessary.

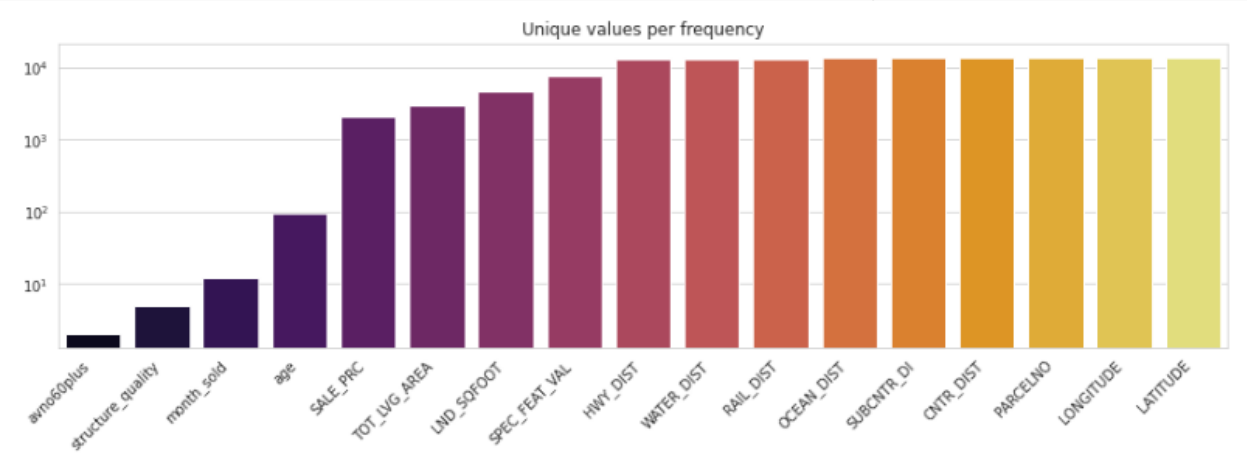
Duplicates can be dropped through ***.drop_duplicates().***

Missing Values

Missing data can be a critical problem with any dataset, but it is very important to make sure they are handled correctly when cleaning or analyzing the data. When you begin data cleaning, missing values are a big problem. For some datasets, tackling first the features and then the samples might be better. Furthermore, the threshold at which it is decided to drop missing values per feature or sample should change in different circumstances.

Per Sample

There are multiple options to consider missing values per sample. The most straightforward one is to simply visualize the result of ***df_X.isna().***



Another better approach is to use the missingno library to get the plot.

Per Feature

For missing values per feature, some pandas features can be used to quickly identify the ratio of missing values per feature. It can be implemented as:

```
df_X.isna().mean().sort_values().plot(
    kind="bar", figsize=(15, 4),
    title="Percentage of missing values per feature",
    ylabel="Ratio of missing values per feature");
```

Unwanted Entries and recording Errors

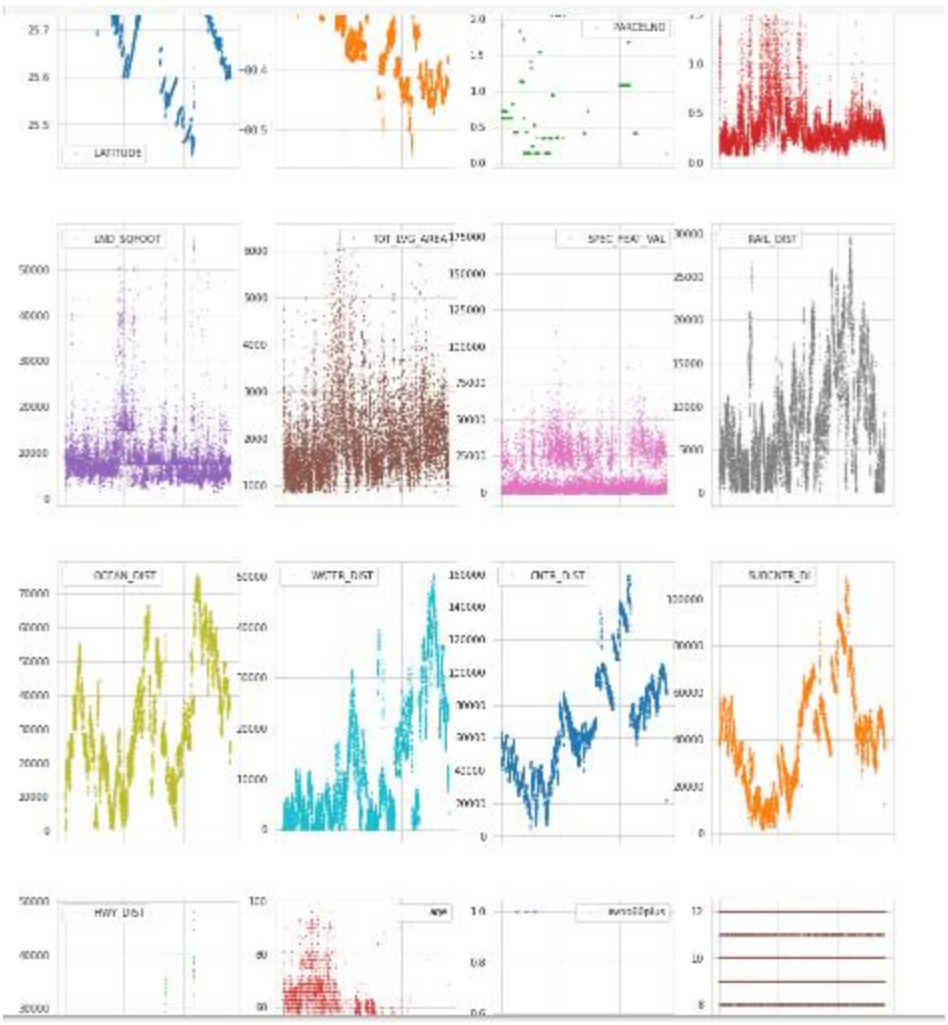
A dataset without unwanted entries is clean and accurate, but it's hard to tell if a dataset can contain such "**unwanted entries**". Unwanted entries can be caused by incorrect or incomplete recordings or by changes in the data after being defined, for instance, because of adding information to a dataset. They need to be removed.

Numerical Features

You can use the ***.plot***() function to show the results of your data analysis in the form of a scatter plot, histogram, frequency distribution, or bar chart. To get started, pass it the name of one or more parameters to customize your plot. The following parameters can be used:

- **lw=0:** lw stands for line width.
- **marker=".":** Instead of lines, we tell the plot to use . as markers for each data point
- **subplots=True:** Subplots tell pandas to plot each feature in a separate subplot
- **layout=(-1, 4):** This parameter tells pandas how many rows and columns to use for the subplots.
- **figsize=(15, 30), markersize=1:** To make sure that the figure is big enough we recommend having a figure height of roughly the number of features, and adjusting the markersize accordingly.

```
df_X.plot(lw=0, marker=".", subplots=True, layout=(-1, 4),
          figsize=(15, 30), markersize=1);
```



We will be handling the duplicates and missing values in the upcoming exercise.

Non-numerical Features

As our dataset has no non-numerical features, it has displayed the following result:

0
1
2
3
4

Exercise 1.03: Quality Investigation

Open the following link to get started (Empty Jupyter Environment) or start with your own local or Hosted Jupyter environment:

https://mybinder.org/v2/gh/fenago/jupyter/HEAD

RESOURCES AND REFERENCES

The code is located here:

datawrangling/Exercise_1_03_Quality_Investigation.ipynb at main · fenago/datawrangling (github.com)

The code repository is located here:

datawrangling/Chapter 1 at main · fenago/datawrangling (github.com)

Content Investigation

The general structure and quality of the data have been observed in the last section. Let's now explore more and take a look at the actual content. Such an investigation is mostly done feature by feature. But it can become cumbersome in the case of 20-30 features.

The following three approaches will give you an idea about the content-related investigations:

Feature Distribution

One of the methods to analyze the content of the dataset involves the value distribution of each feature. It helps to guide EDA as well and provides a lot of information that can help in data cleaning as well as feature transformation. And it can be implemented for the numerical features through the histogram plots. Pandas offer a built-in function for histograms as depicted here:

```
df_X.hist(bins=25, figsize=(15, 25), layout=(-1, 5), edgecolor="black")
```

Most frequent entry

Dataset may contain some features that contain entries of just one kind. Using the .mode() function, the ratio of the most frequent entry for each feature can be extracted and we can visualize that information.

Skewed value distributions

Some numerical features can show non-Gaussian distributions. In such a case, a data scientist may think to transform these values to make them more normally distributed. A "log transformation" can be done in the case of right-skewed data.

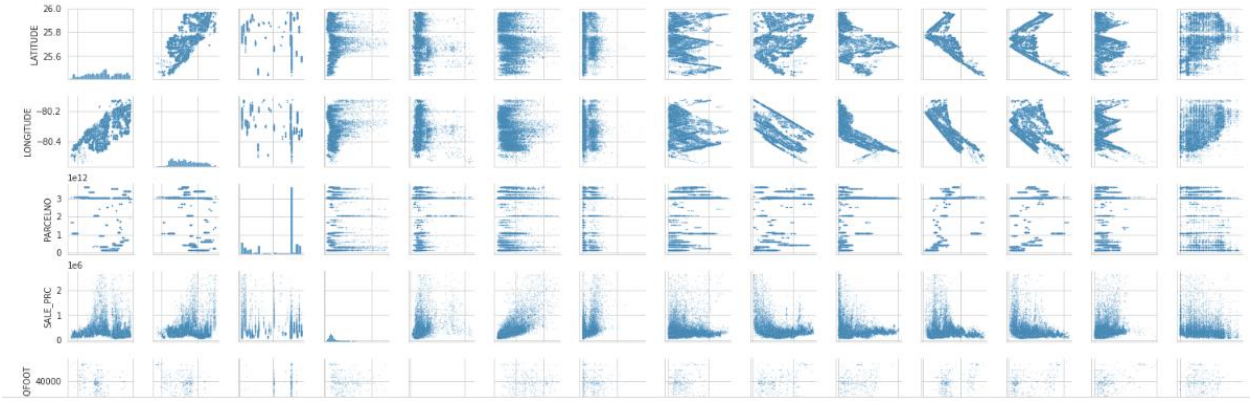
Feature Patterns

The investigation of feature-specific patterns is the next step. The goal includes the following two cases:

1. Data Inspection is the analysis of the quality, and completeness of the information that is stored. It must be achieved first.
2. The relationships between features will help us in understanding the dataset.

Continuous Features:

Seaborn's "pairplot" can be used to visualize the relationships between the continuous features. It can be a time-taking process to create all subplots. So, it is recommended that it may not be used with more than 10 features. In our case, we have 17 features, we can somehow use pairplot. This can be seen in our upcoming exercise.



Discrete and Ordinal features

It is more difficult to find patterns in the discrete or ordinal feature. But some quick pandas and seaborn features can help to get a general overview of a dataset.

```
# Create a new data frame that doesn't contain the numerical continuous features

df_discrete = df[cols_continuous[~cols_continuous].index]

df_discrete.shape
```

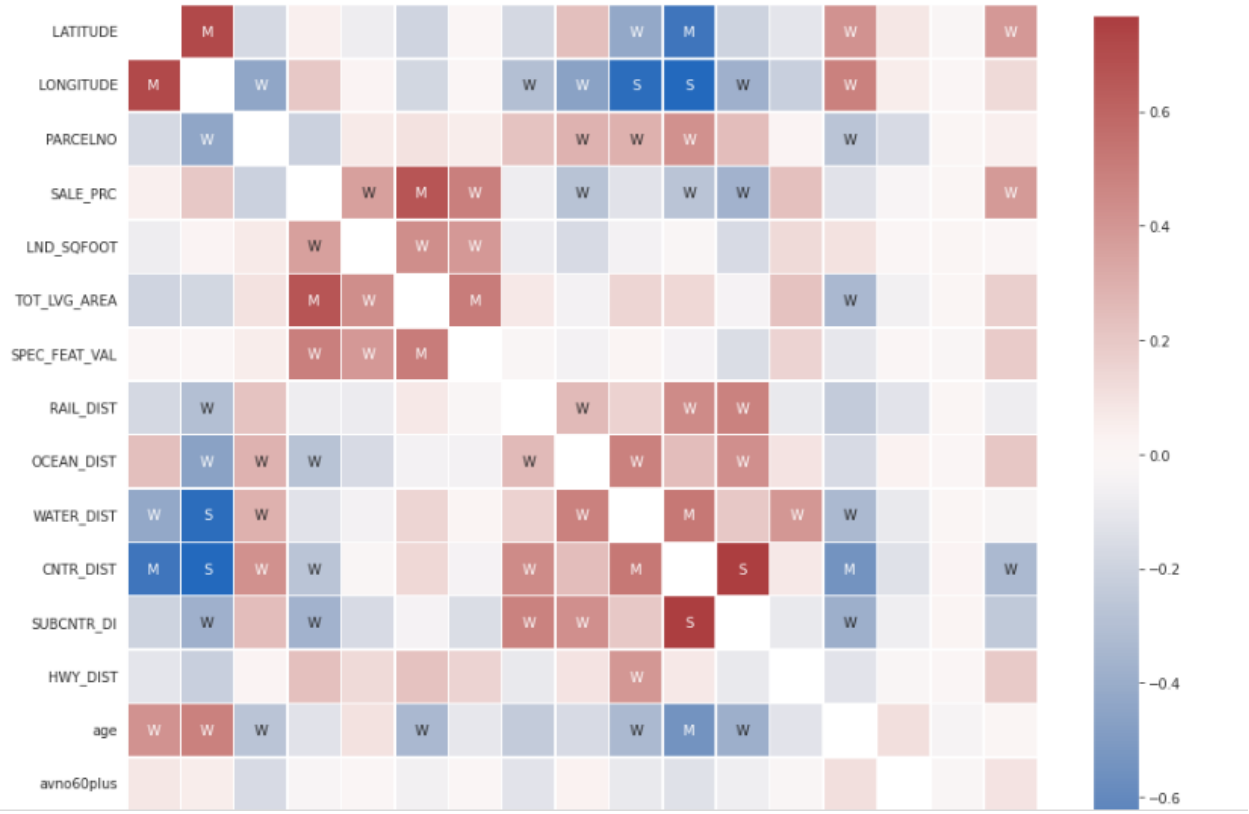
Feature Relationships

It involves the relationships between the features. It can be done through the ".**corr**()" method. It can be implemented as follows:

```
# Computes feature correlation

df_corr = df_X.corr(method="pearson")
```

The “**Spearman**” method can also be used instead of the “**Pearson**” method, depending upon the dataset. Whereas the Pearson correlation evaluates the linear relationship between two continuous variables. The Spearman correlation evaluates the monotonic relationship based on the ranked values for each feature. And to help with the interpretation of this correlation matrix, let's use **seaborn's** .**heatmap**() to visualize it.



Exercise 1.04: Content Investigation

Open the following link to get started (Empty Jupyter Environment) or start with your own local or Hosted Jupyter environment:

https://mybinder.org/v2/gh/fenago/jupyter/HEAD

RESOURCES AND REFERENCES

The code is located here:

datawrangling/Exercise_1_04_Content_Investigation.ipynb at main · fenago/datawrangling (github.com)

The code repository is located here:

datawrangling/Chapter 1 at main · fenago/datawrangling (github.com)

Quantity Investigation – Outliers

IQR method is used by the Box plot to display data and outliers. But a mathematical formula will be needed to get a list of an identified outlier.

IQR – InterQuartile Range

IQR is the range of values that resides in the middle of scores. In case of skewed distribution and when the median is used instead of the mean to show the central tendency, the suitable measure of variability is the InterQuartile Range, which includes:

- Q1: Lower Quartile Part
- Q2: Median
- Q3: Upper Quartile Part

In other words, IQR is equal to the difference between 75th and 25th percentiles, or between upper and lower quartiles, **IQR = Q3 – Q1.** It is a measure of dispersion similar to standard deviation or variance but is much more robust.

We will be covering outliers in the next exercise. Outliers will be filtered out by keeping only valid values.

Exercise 1.05: Quantity Investigation - Outliers

Open the following link to get started (Empty Jupyter Environment) or start with your own local or Hosted Jupyter environment:

https://mybinder.org/v2/gh/fenago/jupyter/HEAD

RESOURCES AND REFERENCES

The code is located here:

datawrangling/Exercise_1_05__Outliers.ipynb at main · fenago/datawrangling (github.com)

The code repository is located here:

datawrangling/Chapter 1 at main · fenago/datawrangling (github.com)

Activity 1.01: Loading the "Future Land Use of Miami" Dataset and performing Quality and Structure Investigation on it.

In this activity, you will have to check the working on another dataset. This dataset contains the records of Future Land Usage from Miami. We will be performing Structure and Quality Investigations to test the skill.

Open the following link to get started (Empty Jupyter Environment) with the aforementioned instructions:

https://mybinder.org/v2/gh/fenago/jupyter/HEAD

RESOURCES AND REFERENCES

To check the solution, you can access it from here:

datawrangling/Activity_1_01_Structure_Quality_Investigations.ipynb at main · fenago/datawrangling (github.com)

Conclusion

In this chapter, we have covered all the preprocessing techniques for a dataset. EDA must be performed right after the tasks performed in this chapter. In this way, it becomes easier to deal with a large number of features in a dataset. A proper and detailed EDA takes more time. It is an iterative process. So, a proper start must be taken to avoid vagueness in the upcoming process. This chapter is revolving around such techniques. It focuses on the EDA basics as well through exercises.

In the next chapter, we will cover EDA in detail including Univariate Analysis, Bivariate Analysis, and multivariate Analysis. Furthermore, the techniques learned in this chapter will be applied in the next chapter. Moreover, these techniques should be applied everywhere before performing the tasks related to Exploratory Data Analysis.