

Data preparation

- Are you aware of how much time a data scientist spends in data preparation? Any guesses yet!!! You will be surprised to know that a data scientist spends 80% of the time preparing data and 60% of it is spent in cleaning the data itself.
- This is where you will be spending most of your time, so it's important and a non negotiable step before the data is ready to be fed to the machine learning models.

- How to deal with a raw data with help of pandas and numpy libraries of python?
- What is numerical and categorical variable in data and how to handle it.
- You will know how to find duplicate values , handle missing values and outliers.
- You will know how to scale the data and why it is important with its visualization impact.

Step 1: Load the data set and storing it in data-frame

```
import pandas as pd
```

```
# Set ipython's max row display
```

```
pd.set_option('display.max_row', 1000)
```

```
# Set iPython's max column width to 50
```

```
pd.set_option('display.max_columns', 50)
```

```
#Loading .xlsx type data set to data-frame with meaningful name.
```

```
df_train = pd.read_excel('Data_Train.xlsx',sheet_name="Sheet1")
```

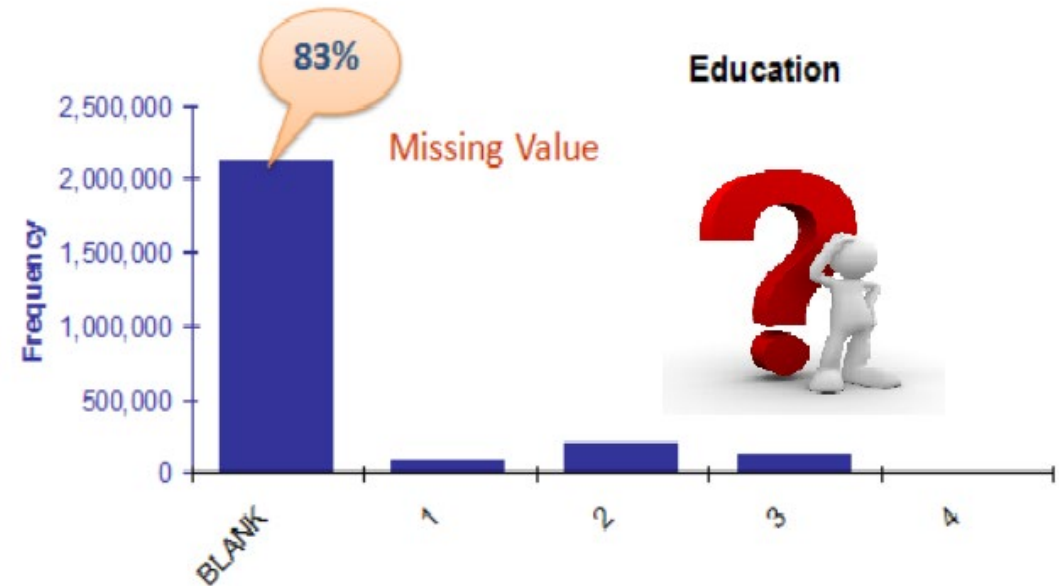
```
df_test = pd.read_excel('Data_Test.xlsx',sheet_name="Sheet1")
```

```
#Loading .csv type data set ,considering the data is in same folder,
```

```
df_csvData = pd.read_csv('horror-train.csv')
```

Step 2: Handling missing data

- Missing values are a common occurrence data and if not handled in the training data set , it can reduce the model fit performance or can lead to a biased model. It can lead to wrong prediction or classification. A missing value signifies a number of different things in your data.



Perhaps the data was not available or not applicable or the event did not happen.

It could be that the person who entered the data did not know the right value, or missed filling in.

Data mining methods vary in the way they treat missing values. There should be a strategy to treat missing values, lets see how we can do it

Remove the missing data

#Method 1: List-wise deletion , is the process of removing the entire data which contains the missing value. Although its a simple process but its disadvantage is reduction of power of the model as the sample size decreases

Number	City	Gender	Age	Income	Illness
1	New York	Male	41	40367	No
2	Los Angeles	Male	54	45084	No
3	New York	Male	42	52483	No
4	Los Angeles	Male	40	40941	No
5	New York	Male	46	50289	No
6	Dallas	Female		50786	No
7	Dallas	Female	32	33155	No
8	Los Angeles	Male	39	30914	No
9	Los Angeles	Male	51	68667	No
10	Los Angeles	Female	30		No
11	Dallas	Female	48	41524	Yes
12	New York	Male	47	54777	No
13	New York	Male	46	62749	No
14	Dallas		42	50894	No
15	Boston	Female	61	38429	No
16	Boston	Male	43	34074	No
17	Dallas	Male	27	50398	No
18	Dallas	Male		46373	Yes
19	New York	Male	47	51137	No
20	New York	Female	35	23688	No
21	New York	Male	57	17378	No

Number	City	Gender	Age	Income	Illness
1	New York	Male	41	40367	No
2	Los Angeles	Male	54	45084	No
3	New York	Male	42	52483	No
4	Los Angeles	Male	40	40941	No
5	New York	Male	46	50289	No
7	Dallas	Female	32	33155	No
8	Los Angeles	Male	39	30914	No
9	Los Angeles	Male	51	68667	No
11	Dallas	Female	48	41524	Yes
12	New York	Male	47	54777	No
13	New York	Male	46	62749	No
15	Boston	Female	61	38429	No
16	Boston	Male	43	34074	No
17	Dallas	Male	27	50398	No
19	New York	Male	47	51137	No
20	New York	Female	35	23688	No
21	New York	Male	57	17378	No

#Method 2: Pair-wise deletion , is the process of removing only specific variables with missing values from the analysis and continue to analyze all other variables without missing values, variables chosen will vary from analysis to analysis based on missingness. One of the disadvantage of this method, it uses different sample size for different variables.

In the above example, for pairwise deletion while performing a correlation we will only perform correlation between city, income and illness and ignore correlation between gender and age but while in list-wise deletion of the missing row would have been done and analysis can be carried out on all three features. Let's see it with some pandas code as well

Number	City	Gender	Age	Income	Illness
1	New York	Male	41	40367	No
2	Los Angeles	Male	54	45084	No
3	New York	Male	42	52483	No
4	Los Angeles	Male	40	40941	No
5	New York	Male	46	50289	No
6	Dallas	Female		50786	No
7	Dallas	Female	32	33155	No
8	Los Angeles	Male	39	30914	No
9	Los Angeles	Male	51	68667	No
10	Los Angeles	Female	30	45919	No
11	Dallas	Female	48	41524	Yes
12	New York	Male	47	54777	No
13	New York	Male	46	62749	No
14	Dallas		42	50894	No
15	Boston	Female	61	38429	No
16	Boston	Male	43	34074	No
17	Dallas	Male	27	50398	No
18	Dallas	Male		46373	Yes
19	New York	Male	47	51137	No
20	New York	Female	35	23688	No
21	New York	Male	57	17378	No

#df is the data frame name where csv data is loaded.

#drop id and use inplace = True only when you want to modify original dataframe, otherwise you can make a copy and use it.

```
df.drop(['Number'],axis=1, inplace=True)
```

#Removing column data

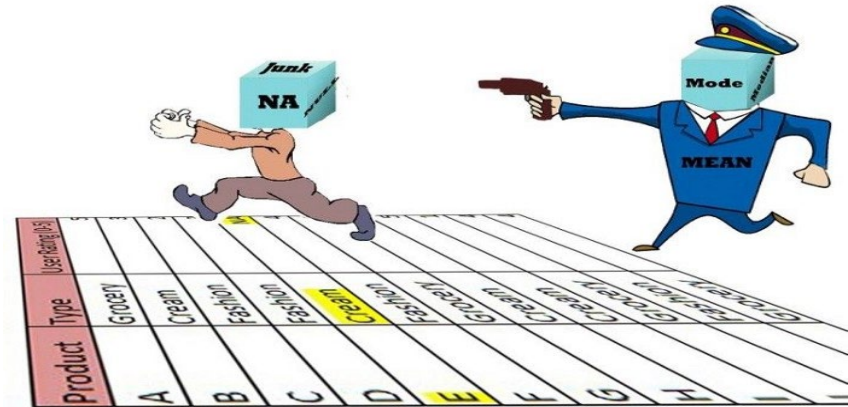
```
df.drop(columns=['City'], inplace=True)
```

#Remove rows which contains null values.

```
df_1 = df.dropna( subset  
=['Gender','Age'] )
```

#Method 3: Retain the Data through imputation

The imputation overcomes the problem of removal of missing records and produces a complete dataset that can be used for machine learning.



#Method 4: Mean , Mode and Median imputation

- Imputation is a way to fill in the missing values with estimated ones. The objective is to employ known relationships that can be identified in the valid values of the data set to assist in estimating the missing values. For numeric data type Mean / Mode / Median imputation is one of the most frequently used methods while for categorical mode is preferred.

Number	City	Gender	Age	Income	Illness
1	New York	Male	41	40367	No
2	Los Angeles	Male	54	45084	No
3	New York	Male	42	52483	No
4	Los Angeles	Male	40	40941	No
5	New York	Male	46	50289	No
6	Dallas	Female		50786	No
7	Dallas	Female	32	33155	No
8	Los Angeles	Male	39	30914	No
9	Los Angeles	Male	51	68667	No
10	Los Angeles	Female	30		No
11	Dallas	Female	48	41524	Yes
12	New York	Male	47	54777	No
13	New York	Male	46	62749	No
14	Dallas	Male	42	50894	No
15	Boston	Female	61	38429	No
16	Boston	Male	43	34074	No
17	Dallas	Male	27	50398	No
18	Dallas	Male		46373	Yes
19	New York	Male	47	51137	No
20	New York	Female	35	23688	No
21	New York	Male	57	17378	No



Number	City	Gender	Age	Income	Illness
1	New York	Male	41	40367	No
2	Los Angeles	Male	54	45084	No
3	New York	Male	42	52483	No
4	Los Angeles	Male	40	40941	No
5	New York	Male	46	50289	No
6	Dallas	Female	mean/median/ mode	50786	No
7	Dallas	Female	32	33155	No
8	Los Angeles	Male	39	30914	No
9	Los Angeles	Male	51	68667	No
10	Los Angeles	Female	30	mean/median/ mode	No
11	Dallas	Female	48	41524	Yes
12	New York	Male	47	54777	No
13	New York	Male	46	62749	No
14	Dallas	Male	42	50894	No
15	Boston	Female	61	38429	No
16	Boston	Male	43	34074	No
17	Dallas	Male	27	50398	No
18	Dallas	Male	mean/median/ mode	46373	Yes
19	New York	Male	47	51137	No
20	New York	Female	35	23688	No
21	New York	Male	57	17378	No

#Filling with mean value

```
df['Income'] = df['Income'].fillna((df['Income'].mean()))
```

#Filling with median value

```
df['Age'] = df['Age'].fillna((df['Age'].median()))
```

#Filling with mode value

```
df['Age'] = df['Age'].fillna((df['Age'].mode()))
```

#Method 5: Forward filling

- Also commonly known as Last observation carried forward (LOCF). It is the process of replacing a missing value with the last observed record. It is widely used as an imputed method in time series data. This method is advantageous as it is easy to communicate, but it is based on the assumption that the value of the outcome remains unchanged by the missing data, which seems very unlikely.

```

import pandas as pd
df = pd.DataFrame([[1, 2, 3], [None, None, 6], [None, 9, None]])
print(df)
'''
      0    1    2
0  1.0  2.0  3.0
1  NaN  NaN  6.0
2  NaN  9.0  NaN
'''

df1 = df[1].fillna(method='ffill')
print(df1)
'''
      0    1    2
0  1.0  2.0  3.0
1  NaN  2.0  6.0
2  NaN  9.0  NaN
'''

df1 = df.fillna(method='ffill')
print(df1)
'''
      0    1    2
0  1.0  2.0  3.0
1  1.0  2.0  6.0
2  1.0  9.0  6.0
'''

df[0].fillna(method='ffill',limit=1 , inplace=True)
print(df)
'''
      0    1    2
0  1.0  2.0  3.0
1  1.0  NaN  6.0
2  NaN  9.0  NaN
'''

```

#Method 6: Backward filling

- As the name suggest, its exact opposite of forward filling and also commonly know as Next Observation Carried Backward (NOCB). It takes the first observation after the missing value and carrying it backward.
- For backward filling, you can replace imputation method to “*bfill*” in Forward fill example.

#Method 7: Linear Interpolation

- Interpolation is a mathematical method that adjusts a function to data and uses this function to extrapolate the missing data. In simple words using a logic to fill the missing values.
- The simplest type of interpolation is the linear interpolation, that makes a mean between the values before the missing data and the value after.
- Of course, we could have a pretty complex pattern in data and linear interpolation could not be enough.
- There are several different types of interpolation. Just in Pandas we have the following options like : 'linear', 'time', 'index', 'values', 'nearest', 'zero', 'slinear', 'quadratic', 'cubic', 'polynomial', 'spline', 'piece wise polynomial' and many more .

Step 3: Check for duplicate value

- One another reason why your model performance might not be accurate can be because of the duplicated data, which can make the data biased and results corrupted.

keep : {'first', 'last', False}, default 'first'

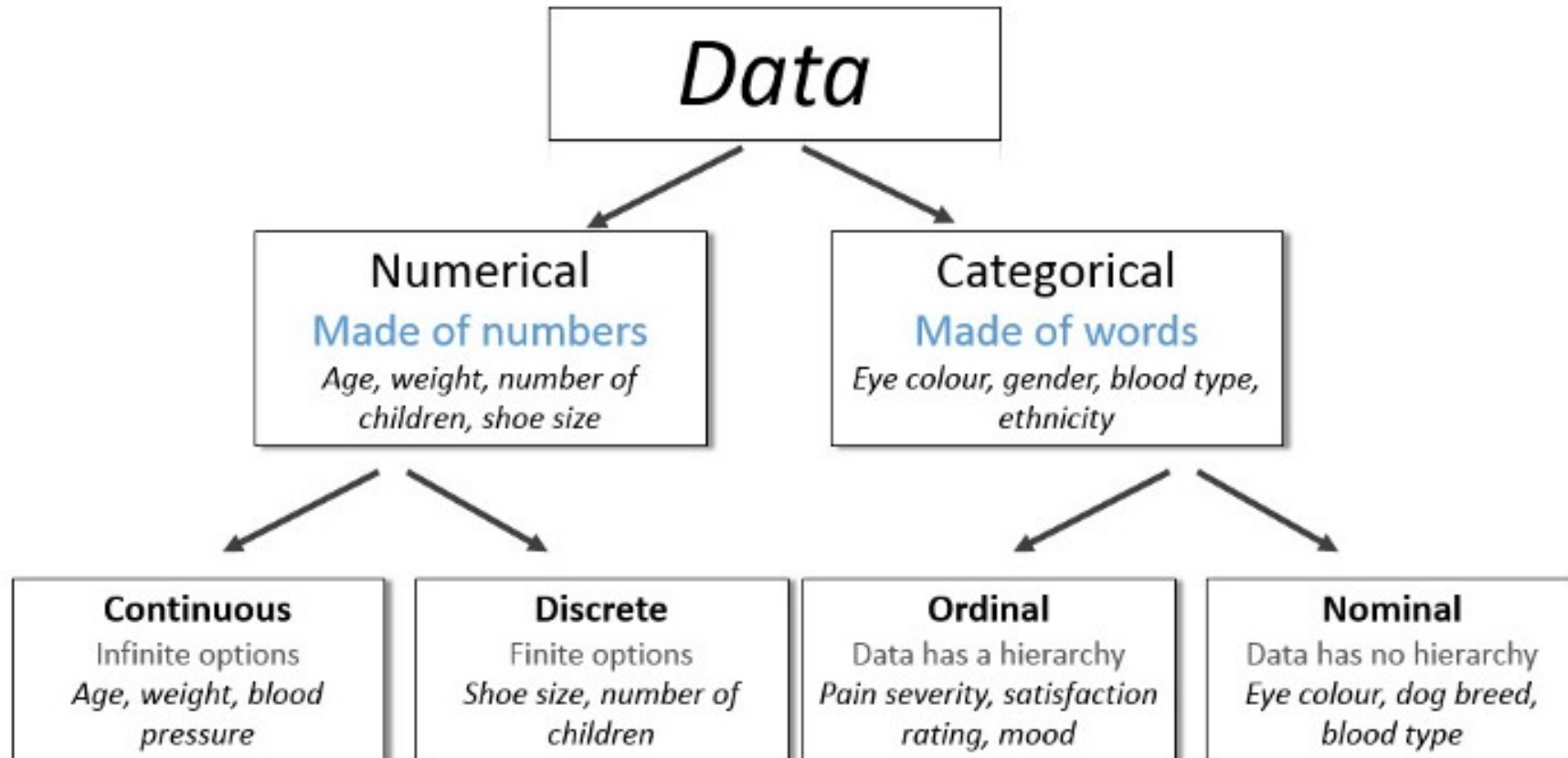
- first : Mark duplicates as True except for the first occurrence.
- last : Mark duplicates as True except for the last occurrence.
- False : Mark all duplicates as True.

```
import pandas as pd
df = pd.DataFrame(['a','b','c','d','a','b','e'])
df[df.duplicated(keep=False)]
'''
```

Out[1]:

```
0
0 a
1 b
4 a
5 b
'''
```

Step 4: Separating categorical and numerical data.



Data Wrangling in Python

- Data wrangling is the process of removing errors and combining complex data sets to make them more accessible and easier to analyze. Due to the rapid expansion of the amount of data and data sources available today, storing and organizing large quantities of data for analysis is becoming increasingly necessary.
- A data wrangling process, also known as a data munging process, consists of reorganizing, transforming and mapping data from one "raw" form into another in order to make it more usable and valuable for a variety of downstream uses including analytics.

Wrangling

- Data wrangling can be defined as the process of cleaning, organizing, and transforming raw data into the desired format for analysts to use for prompt decision-making.
- Also known as data cleaning or data munging, data wrangling enables businesses to tackle more complex data in less time, produce more accurate results, and make better decisions.
- The exact methods vary from project to project depending upon your data and the goal you are trying to achieve. More and more organizations are increasingly relying on data wrangling tools to make data ready for downstream analytics.

importance of using data wrangling tools

- Making raw data usable. Accurately wrangled data guarantees that quality data is entered into the downstream analysis.
- Getting all data from various sources into a centralized location so it can be used.
- Piecing together raw data according to the required format and understanding the business context of data
- Automated data integration tools are used as data wrangling techniques that clean and convert source data into a standard format that can be used repeatedly according to end requirements. Businesses use this standardized data to perform crucial, cross-data set analytics.

- Cleansing the data from the noise or flawed, missing elements
- Data wrangling acts as a preparation stage for the data mining process, which involves gathering data and making sense of it.
- Helping business users make concrete, timely decisions

six iterative steps

Data wrangling software typically performs six iterative steps of Discovering, Structuring, Cleaning, Enriching, Validating, and Publishing data before it is ready for analytics.

Benefits of Data Wrangling

- Data wrangling helps to improve data usability as it converts data into a compatible format for the end system.
- It helps to quickly build data flows within an intuitive user interface and easily schedule and automate the data-flow process.
- Integrates various types of information and their sources (like databases, web services, files, etc.)
- Help users to process very large volumes of data easily and easily share data-flow techniques.

Data Wrangling Tools

- There are different tools for data wrangling that can be used for gathering, importing, structuring, and cleaning data before it can be fed into analytics and BI apps. You can use automated tools for data wrangling, where the software allows you to validate data mappings and scrutinize data samples at every step of the transformation process.
- This helps to quickly detect and correct errors in data mapping. Automated data cleaning becomes necessary in businesses dealing with exceptionally large data sets. For manual data cleaning processes, the data team or data scientist is responsible for wrangling. In smaller setups, however, non-data professionals are responsible for cleaning data before leveraging it.

Some examples of basic data munging tools are:

- Spreadsheets / Excel Power Query - It is the most basic manual data wrangling tool
- OpenRefine - An automated data cleaning tool that requires programming skills
- Tabula – It is a tool suited for all data types
- Google DataPrep – It is a data service that explores, cleans, and prepares data
- Data wrangler – It is a data cleaning and transforming tool

Data Wrangling Examples

Data wrangling techniques are used for various use-cases. The most commonly used examples of data wrangling are for:

- Merging several data sources into one data-set for analysis
- Identifying gaps or empty cells in data and either filling or removing them
- Deleting irrelevant or unnecessary data
- Identifying severe outliers in data and either explaining the inconsistencies or deleting them to facilitate analysis

Businesses also use data wrangling tools to

- Detect corporate fraud
- Support data security
- Ensure accurate and recurring data modeling results
- Ensure business compliance with industry standards
- Perform Customer Behavior Analysis
- Reduce time spent on preparing data for analysis
- Promptly recognize the business value of your data
- Find out data trends

Top tech companies typically look for the following skillsets in data science candidates.

- To be able to perform series of data transformations like merging, ordering, aggregating
- To use data science programming languages like R, [Python](#), Julia, [SQL](#) on specified data sets
- To make logical judgments based on underlying business context

Detecting and Treating Outliers

- One of the most important steps as part of data preprocessing is detecting and treating the outliers as they can negatively affect the statistical analysis and the training process of a machine learning algorithm resulting in lower accuracy.

1. What are Outliers?

We all have heard of the idiom 'odd one out' which means something unusual in comparison to the others in a group.

Similarly, an Outlier is an observation in a given dataset that lies far from the rest of the observations. That means an outlier is vastly larger or smaller than the remaining values in the set.

2. Why do they occur?

- An outlier may occur due to the variability in the data, or due to experimental error/human error.
- They may indicate an experimental error or heavy skewness in the data (heavy-tailed distribution).


3. What do they affect?

- In statistics, we have three measures of central tendency namely Mean, Median, and Mode. They help us describe the data.
- Mean is the accurate measure to describe the data when we do not have any outliers present.
- Median is used if there is an outlier in the dataset.
- Mode is used if there is an outlier AND about $\frac{1}{2}$ or more of the data is the same.
- 'Mean' is the only measure of central tendency that is affected by the outliers which in turn impacts Standard deviation.

Example:

Consider a small dataset, sample= [15, 101, 18, 7, 13, 16, 11, 21, 5, 15, 10, 9]. By looking at it, one can quickly say '101' is an outlier that is much larger than the other values.

+-----+-----+	
with outlier	without outlier
+-----+-----+	
Mean: 20.08	Mean: 12.72
Median: 14.0	Median: 13.0
Mode: 15	Mode: 15
Variance: 614.74	Variance: 21.28
Std dev: 24.79	Std dev: 4.61
+-----+-----+	



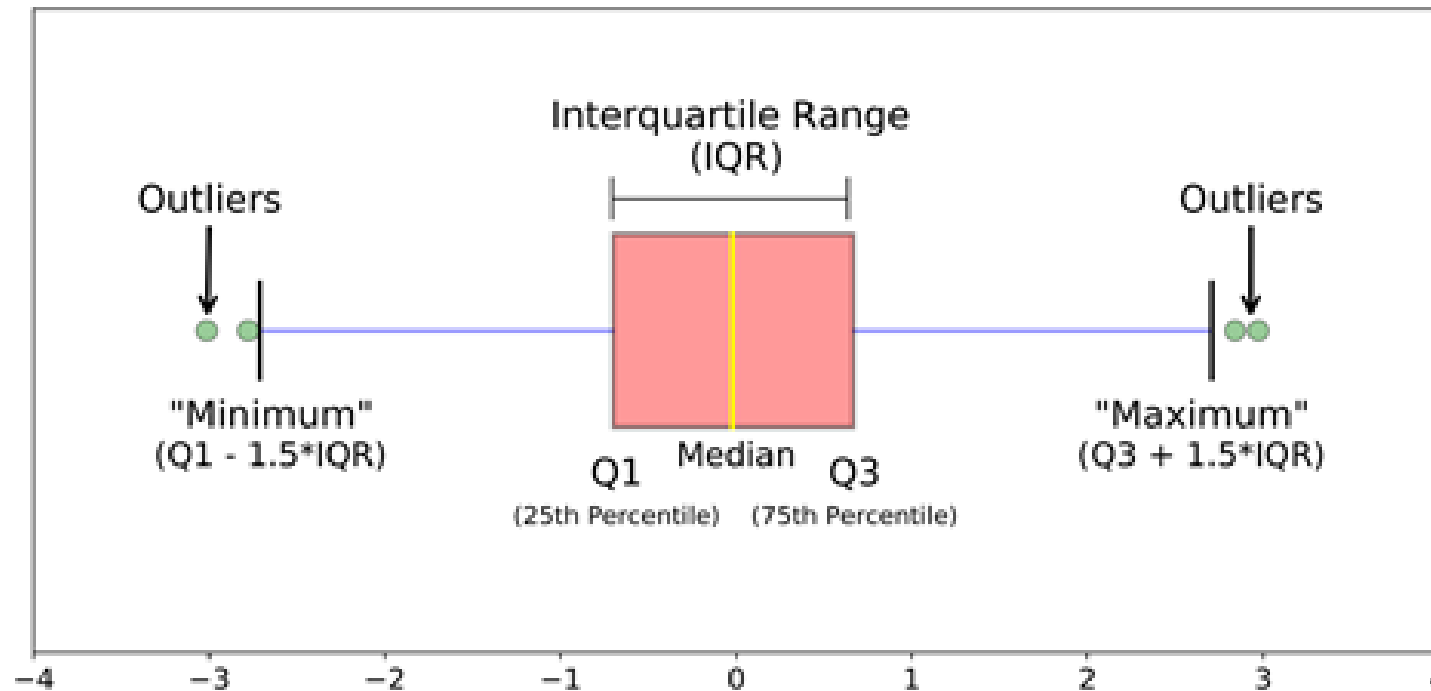
4. Detecting Outliers

If our dataset is small, we can detect the outlier by just looking at the dataset. But what if we have a huge dataset, how do we identify the outliers then? We need to use visualization and mathematical techniques.

Below are some of the techniques of detecting outliers

- Boxplots
- Z-score
- Inter Quantile Range(IQR)

Detecting outliers using the Inter Quartile Range(IQR)



Criteria: data points that lie 1.5 times of IQR above Q3 and below Q1 are outliers.

Steps:

- Sort the dataset in ascending order
- calculate the 1st and 3rd quartiles($Q1$, $Q3$)
- compute $IQR = Q3 - Q1$
- compute lower bound = $(Q1 - 1.5 * IQR)$, upper bound = $(Q3 + 1.5 * IQR)$
- loop through the values of the dataset and check for those who fall below the lower bound and above the upper bound and mark them as outliers

Handling Outliers

Below are some of the methods of treating the outliers

- Trimming/removing the outlier
- Quantile based flooring and capping
- Mean/Median imputation