# Data preprocessing

Francesco Pugliese, PhD

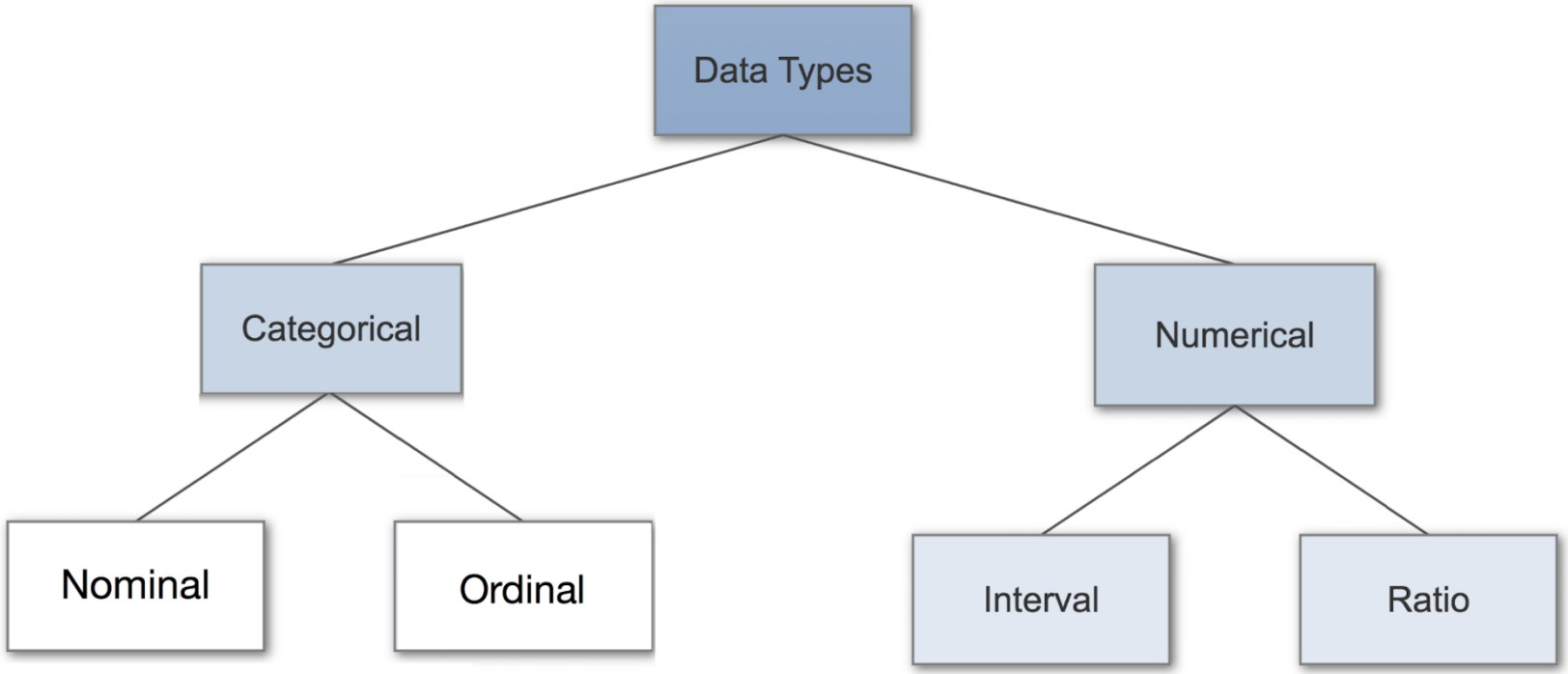
neural1977@gmail.com

# Table of contents

# Types of data

- ✓ Having a good understanding of the different data types, also called measurement scales, is a crucial prerequisite for doing **Exploratory Data Analysis** (EDA), since you can use certain statistical measurements only for specific data types.
- ✓ You also need to know which data type you are dealing with to choose the right visualization method.
- ✓ Think of data types as a way to categorize different types of variables.
- ✓ Taking a broader perspective, data is classified into numerical and categorical data.
    - ✓ **Numerical**: As the name suggests, this is numeric data that is quantifiable. Numerical data is further divided into **discrete** and **continuous**
    - ✓ **Categorical**: The data is a string or non-numeric data that is qualitative in nature. Categorical data is further divided into **ordered** and **nominal**

# Types of data



# **Discrete data** (Numerical)

- ✓ To explain in simple terms, any numerical data that is **countable**.
- ✓ Discrete data can only **take certain values** (such as 1, 2, 3, 4, etc).
- ✓ Values are **distinct** and **separate**
- ✓ can’t be measured but it can be counted
- ✓ **Example**: the number of people in a family or the number of students in a class.
- ✓ basically represents information that can be categorized into a classification

# **Continuous data** (Numerical)

- ✓ Any numerical data that is **measurable** is called continuous.
- ✓ **Example**: the height of a person or the time taken to reach a destination.
- ✓ Continuous data can take **virtually any value** (for example, 1.25, 3.8888, and 77.1276).
- ✓ You can summarise your data using percentiles, median, interquartile range, mean, mode, standard deviation, and range.
- ✓ **Visualisation Methods**: To visualise continuous data, you can use a **histogram** or a **box-plot**. With a histogram, you can check the central tendency, variability, modality, and kurtosis of a distribution. Note that a histogram can't show you if you have any **outliers**. This is why we also use box-plots.

# **Nominal data** (Categorical)

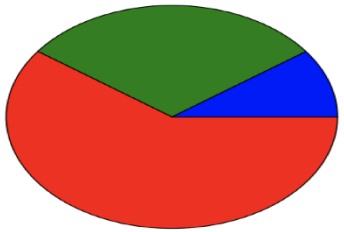
- ✓ Any categorical data that has **no order**.
- ✓ **Example**: gender, country etc..,
- ✓ In Data Science, you can use **one hot encoding**, to transform nominal data into a numeric feature.
- ✓ When you are dealing with nominal data, you collect information through:
  - ✓ **Frequencies**: The Frequency is the rate at which something occurs over a period of time or within a dataset.
  - ✓ **Proportion**: You can easily calculate the proportion by dividing the frequency by the total number of events. (e.g how often something happened divided by how often it could happen)
  - ✓ **Percentage**.
- ✓ **Visualisation Methods**: To visualise nominal data you can use a **pie** chart or a **bar** chart.

# **Ordinal data** (Categorical)

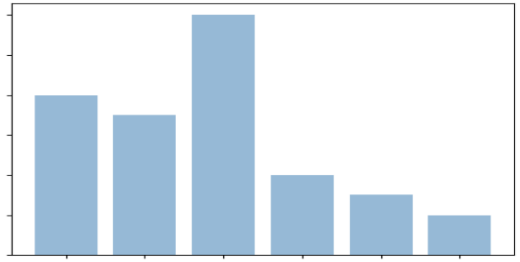
- ✓ Any categorical data that has **some order** associated with it.
- ✓ **Example**: **movie ratings** (excellent, good, bad, worst) and **feedback** (happy, not bad, bad).
- ✓ You can think of ordered data as being something you could **mark on a scale**.
- ✓ Observe that the differences between the values is not really known.
- ✓ Due to this reason they are usually used to measure non-numeric features like happiness, customer satisfaction and so on.
- ✓ you can summarise your ordinal data with **frequencies**, **proportions**, **percentages**. And you can **visualise** it with **pie** and **bar** charts. Additionally, you can use **percentiles**, **median**, **mode** and the **interquartile range** to summarise your data.
- ✓ you can use one label encoding, to transform ordinal data into a numeric feature.

# Data visualization types

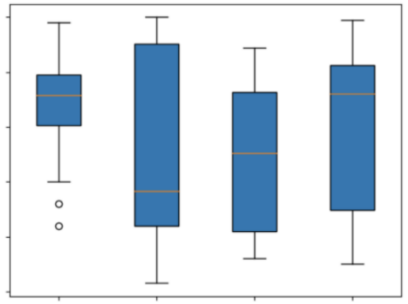
Pie Chart



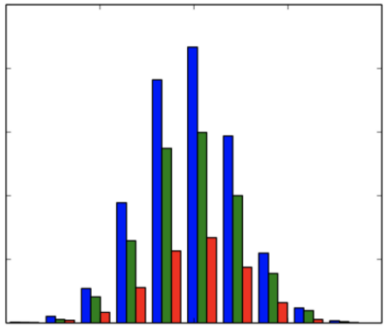
Bar Chart



Boxplot



Histogram



# Why Data preprocessing ?

- ✓ Real-world data is often incomplete, inconsistent, and lacking in certain behaviors or trends, and is likely to contain many errors. i.e

    - ✓ **Incomplete**: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - ✓ **Noisy**: containing errors or outliers
    - ✓ **Inconsistent**: containing discrepancies in codes or names

    Data preprocessing is a proven method of resolving such issues.

- ✓ Data preprocessing, a technique that focuses more on Data cleaning, Data integration, Data transformation, Data dimensionality reduction, Data discretization.

# Data Wrangling

- ✓ When we perform data wrangling, we are taking our input data from its original state to a format where we can perform **meaningful** analysis on it.

- ✓ **Data manipulation** is another way to refer to this process.

- ✓ There is no set list or order of operations; the only **goal** is that the data post-wrangling is more useful to us than when we started.

- ✓ There are three common tasks involved in the data wrangling process:
  - ✓ Data cleaning
  - ✓ Data transformation
  - ✓ Data enrichment

  Let us see what each of these 3 tasks involve

# Data cleaning

- ✓ An initial round of data cleaning on our data frame will often give us the bare minimum we need to start exploring our data.

- ✓ Some essential data cleaning tasks to master include the following:
  - ✓ Renaming
  - ✓ Sorting and reordering
  - ✓ Data type conversions
  - ✓ Deduplicating data
  - ✓ Addressing missing or invalid data
  - ✓ Filtering to the desired subset of data

- ✓ Data cleaning is the best starting point for data wrangling since having the data stored as the correct data types and easy-to-reference names will open up many avenues for exploration and wrangling opportunities, such as **summary statistics**, **sorting**, and **filtering**.

# Data transformation

- ✓ Frequently, we will reach the data transformation stage after some initial data cleaning, but it is entirely possible that our dataset is **unusable** in its current shape, and we must **restructure** it before attempting to do any data cleaning.

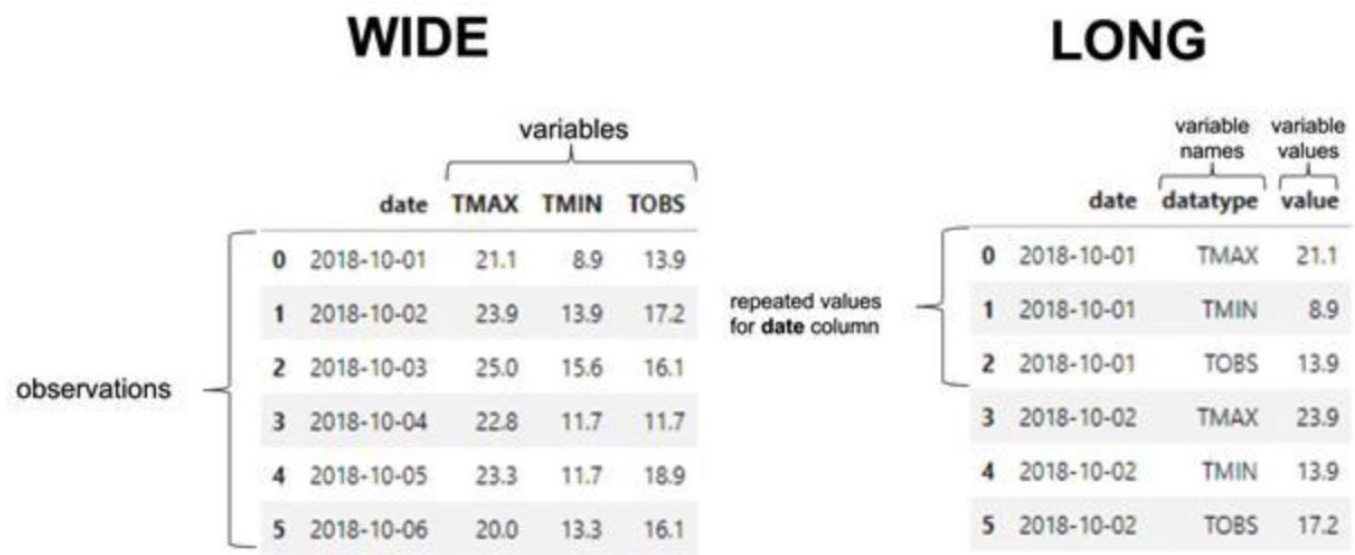
- ✓ We mainly focus on changing our **data's structure** to facilitate our **downstream** analyses; this usually involves changing which data goes along the rows and which goes down the columns.

- ✓ Most data we will find is either in a **wide** format or a **long** format; each of these formats has its merits, and it's important to know which one we will need for our analysis.

- ✓ Often, people will record and present data in the wide format, but there are certain visualizations that require the data to be in the long format.

- ✓ The wide format is preferred for analysis and database design, while the long format is considered poor design because each column should be its own data type and have a singular meaning. When building an API, the long format may be chosen if flexibility is required.

# Wide vs Long format



# Data enrichment

- ✓ When we're looking to enrich the data, we can either merge new data with the original data (by appending new rows or columns) or use the original data to create new data.

- ✓ The following are ways to enhance our data using the original data:

  - ✓ **Adding new columns**: Using functions on the data from existing columns to create new values

  - ✓ **Binning**: Turning continuous data or discrete data with many distinct values into range buckets, which makes the column discrete while letting us control the number of possible values in the column

  - ✓ **Aggregating**: Rolling up the data and summarizing it

  - ✓ **Resampling**: Aggregating time series data at specific intervals

# Data Normalization

- ✓ Also referred to as **Column Normalization**. Data Normalization usually means to scale a variable to have a value between 0 and 1, and we can archive that by using the following formula:

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

- ✓ **Example**: Let us consider a list of numberic values, x[] = [3.5, 3.0, 3.2, 3.1, 3.6, 3.7, 3.4, 3.4, 2.9, 2.7] and apply data normalization using above equation as follows,

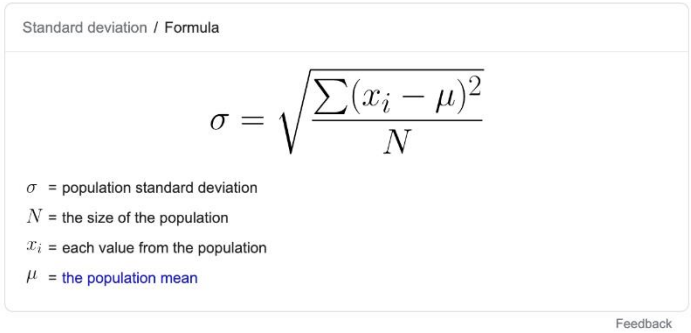
  x max = max(numAry) = 3.7
  x min = min(numAry) = 2.7
  x[] new = [0.7, 0.2, 0.5, 0.3, 0.8, 1.0, 0.6, 0.6, 0.1, 0.0]

# Data Standardization

- ✓ Standardization transforms data to have a mean of zero and a standard deviation of 1. Data points can be standardized with the following formula:

$$z_i = \frac{x_i - \bar{x}}{S}$$

Where S is standard deviation

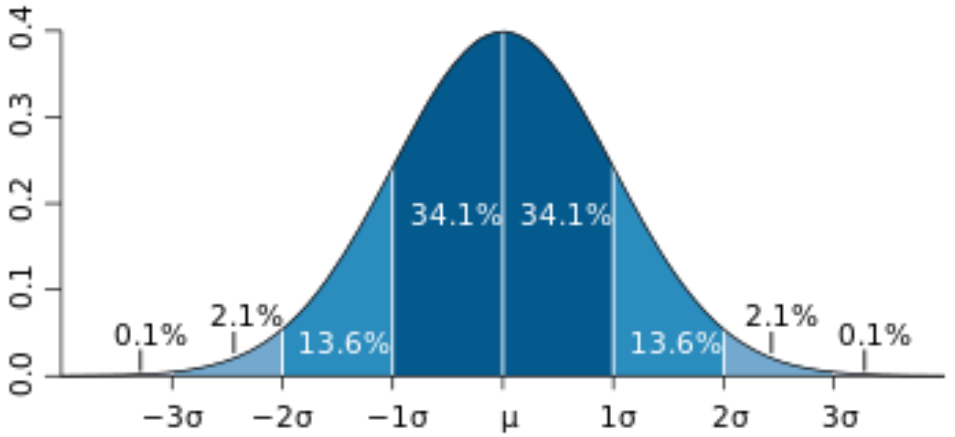



- ✓ **Example**: Let us consider a list of numberic values, x[] = [3.5, 3.0, 3.2, 3.1, 3.6, 3.7, 3.4, 3.4, 2.9, 2.7] and apply data standardization using above equation as follows,

$\bar{x}$ = mean(x[]) = 3.25
S = standard deviation(x[]) = 0.3240370349203930 4
x[] new = [0.77, -0.77, -0.15, -0.46, 1.08, 1.38, 0.46, 0.46, -1.0, -1.6]

# Data Normalization

- ✓ What we meant by standard deviation by 1 is all the points will lie between -1σ and 1σ(as shown below).
- ✓ As 68% of the values lie within one standard deviation of the mean. Here we can say that one standard deviation is ling between -1.5 and 1.5.




- ✓ Column Standardization is also called as Mean Centering.
- ✓ Sometimes it’s also known as z-score.

# Encoding categorical data

- ✓ There are some algorithms that can work well with categorical data, such as decision trees. But most machine learning algorithms cannot operate directly with categorical data.

- ✓ These algorithms require the input and output both to be in numerical form. If the output to be predicted is categorical, then after prediction we convert them back to categorical data from numerical data.

- ✓ There are three simple methods of encoding categorical data:
  - ✓ Replacing
  - ✓ Label Encoding
  - ✓ One-Hot Encoding

# Encoding categorical data

- ✓ **Replacing**:
  - ✓ This is a technique in which we replace the categorical data with a number.
  - ✓ This is a simple manual replacement and does not involve much logical processing

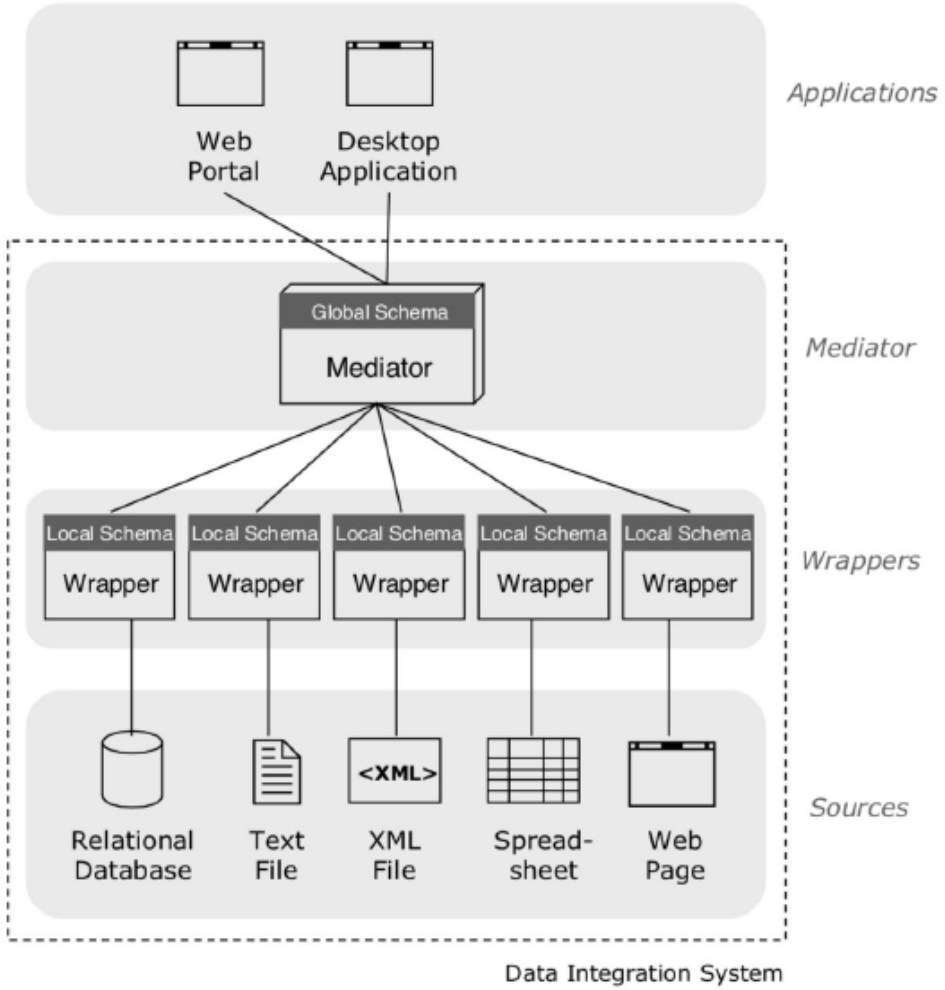
- ✓ **Label encoding**:
  - ✓ This is a technique in which we replace each value in a categorical column with numbers from 0 to N-1. For example, say we've got a list of employee names in a column. After performing label encoding, each employee name will be assigned a numeric label.
  - ✓ Label encoding is the best method to use for **ordinal data**.
  - ✓ The scikit-learn library provides **LabelEncoder()**, which helps with label encoding.

- ✓ **One hot encoding**:
  - ✓ Here, the label-encoded data is further divided into n number of columns. Here, n denotes the total number of unique labels generated while performing label encoding.
  - ✓ For example, say that three new labels are generated through label encoding. Then, while performing one-hot encoding, the columns will be divided into three parts.

# Data Integration

- ✓ Data Integration (or Information Integration) is the problem of **finding and combining data from different sources**.
- ✓ Process of combining data from different sources into a single, unified view.
- ✓ Integration begins with the ingestion process, and includes steps such as cleansing, ETL mapping, and transformation.
- ✓ Data integration ultimately enables analytics tools to **produce effective, actionable business intelligence**.
- ✓ Abstracting out the differences between individual systems, a typical view-based data integration system (VDIS) conforms to the architecture shown in the next slide.

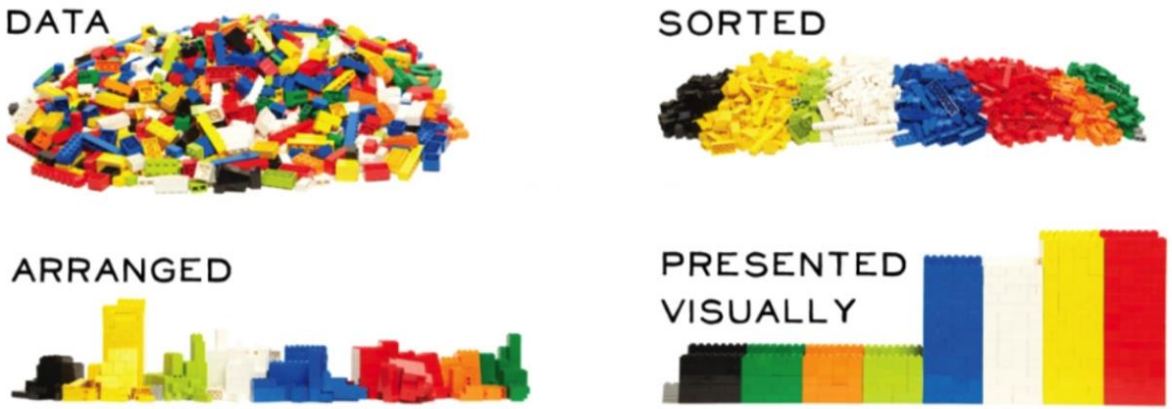
# View based data integration system (VDIS)



# View based data integration system (VDIS)

- ✓ **Sources**: store the data in a variety of formats (relational databases, text files etc.).
- ✓ **Wrappers**: solve the heterogeneity in the formats by transforming each source's data model to a common data model used by the integration system.
- ✓ The wrapped data sources are usually referred to as local or source databases, the structure of which is described by corresponding local/source schemas. This is in contrast to the unified view exported by the mediator, also called global/target database.
- ✓ Finally, mappings expressed in a certain mapping language (depicted as lines between the wrapped sources and the mediator) specify the relationship between the wrapped data sources (i.e. the local schemas) and the unified view exported by the mediator (global schema).

# Exploratory data analysis(EDA)




"Torture the data, and it will confess to anything."

— Ronald Coase

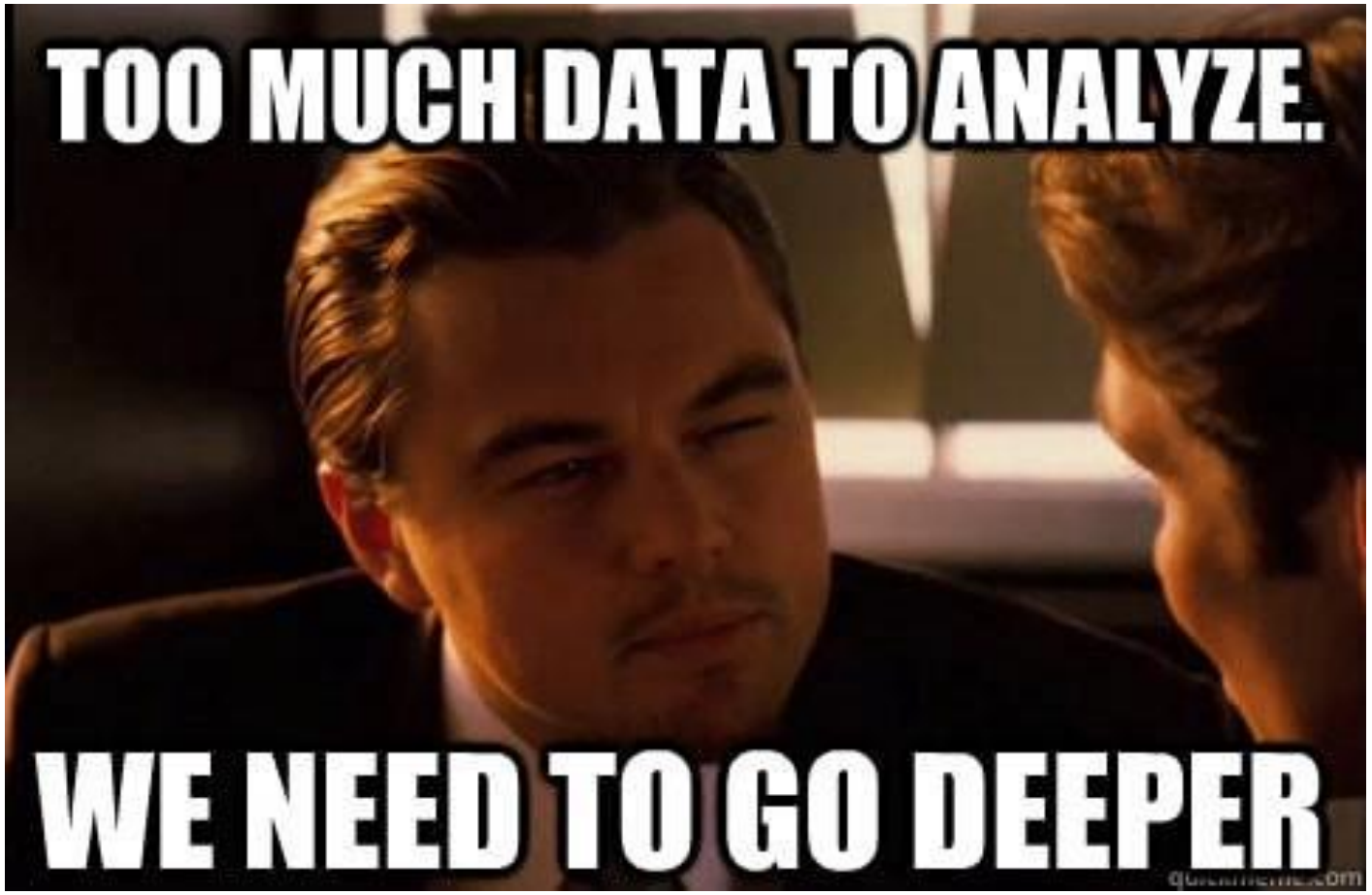
# Exploratory data analysis(EDA)

- ✓ Exploratory Data Analysis (EDA) is the process of **visualizing** and **analyzing** data to extract insights from it. In other words, EDA is the process of **summarizing** important characteristics of data in order to gain better **understanding** of the dataset.

- ✓ Methods for EDA:
  - ✓ Descriptive Statistics
  - ✓ Grouping of Data
  - ✓ Handling missing values in dataset
  - ✓ ANOVA: Analysis of variance Correlation

  Lets understand each of the above methods

# Exploratory data analysis(EDA)



# **Method 1:** Descriptive statistics

- ✓ Descriptive statistics analysis helps to describe the basic features of dataset and obtain a brief summary of the data.

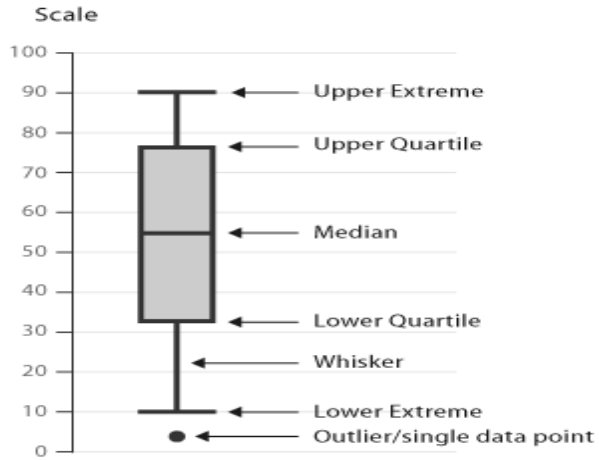
- ✓ The describe() method in Pandas library helps us to have a brief summary of the dataset. It automatically calculates basic statistics for all numerical variables excluding values.

- ✓ But, what if we have **categorical** data? How can we get a summary of categorical data? The **value_counts()** method will be useful in this case.

- ✓ To analyse the numerical data we can make use of different plots such as,
  - ✓ Box plot
  - ✓ Scatter plots
  - ✓ Histograms

# Plots for Descriptive statistics

- ✓ **Box-plot**:
  - ✓ Box plot shows us the median of the data, which represents where the middle data point is.
  - ✓ The upper and lower quartiles represent the 75 and 25 percentile of the data respectively.
  - ✓ The upper and lower extremes shows us the extreme ends of the distribution of our data.
  - ✓ Finally, it also represents outliers, which occur outside the upper and lower extremes.



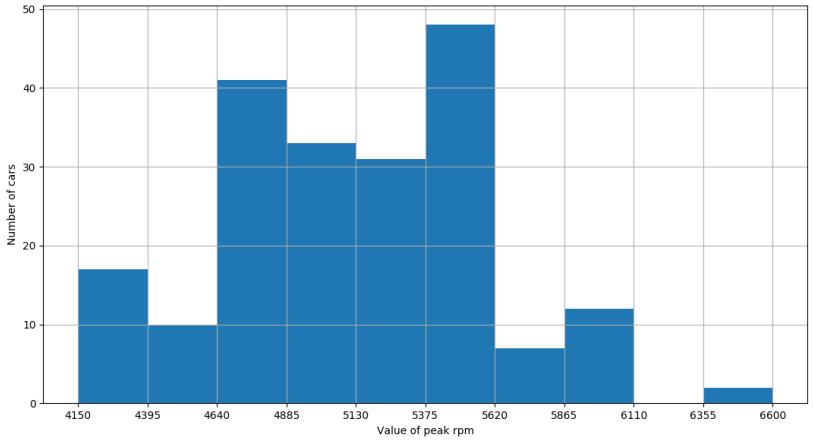
# Plots for Descriptive statistics

- ✓ **Scatter-plot**:
  - ✓ Scatter plots represent each relationship between two continuous variables as individual data point in a 2D graph.

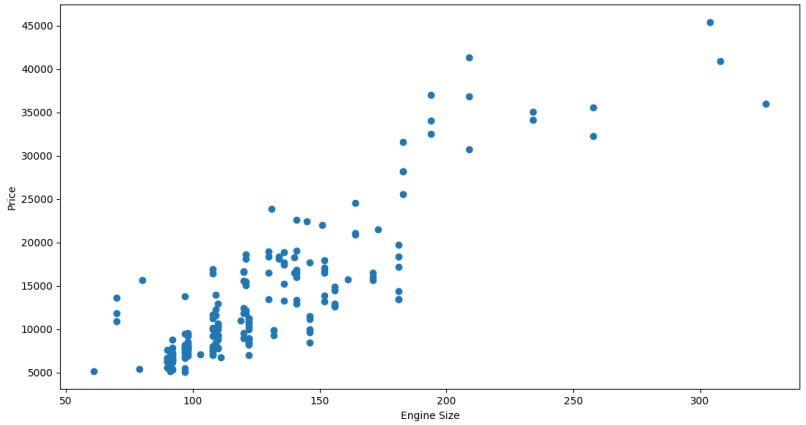
- ✓ **Histogram**:
  - ✓ shows us the frequency distribution of a variable.
  - ✓ It partitions the spread of numeric data into parts called as “bins” and then counts the number of data points that fall into each bin.
  - ✓ So, the vertical axis actually represents the number of data points in each bin.

# Plots for Descriptive statistics

Histogram




Scatter plot



# Other methods for EDA

- ✓ **Grouping of data**: The groupby() method from Pandas library helps us to accomplish this task.

- ✓ **Handling missing values**:
  - ✓ When no data value is stored for a feature in a particular observation, we say this feature has missing values.
  - ✓ Examining this is important because when some of your data is missing, it can lead to weak or biased analysis.
  - ✓ We can detect missing values by applying isnull() method over the dataframe.
  - ✓ The isnull() method returns a rectangular grid of boolean values which tells us if a particular cell in the dataframe has missing value or not.

# Handling Missing values

| | symboling | normalized-losses | make | aspiration | num-of-doors | body-style | drive-wheels | engine-location | wheel-base | length | ... | compression-ratio | horsepower | peak-rpm | city-mpg | highway-mpg | pric |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | False | False | False | False | False | False | False | False | False | False | ... | False | False | False | False | False | Fals |
| 1 | False | False | False | False | False | False | False | False | False | False | ... | False | False | False | False | False | Fals |
| 2 | False | False | False | False | False | False | False | False | False | False | ... | False | False | False | False | False | Fals |
| 3 | False | False | False | False | False | False | False | False | False | False | ... | False | False | False | False | False | Fals |
| 4 | False | False | False | False | False | False | False | False | False | False | ... | False | False | False | False | False | Fals |
| 5 | False | False | False | False | False | False | False | False | False | False | ... | False | False | False | False | False | Fals |
| 6 | False | False | False | False | False | False | False | False | False | False | ... | False | False | False | False | False | Fals |
| 7 | False | False | False | False | False | False | False | False | False | False | ... | False | False | False | False | False | Fals |

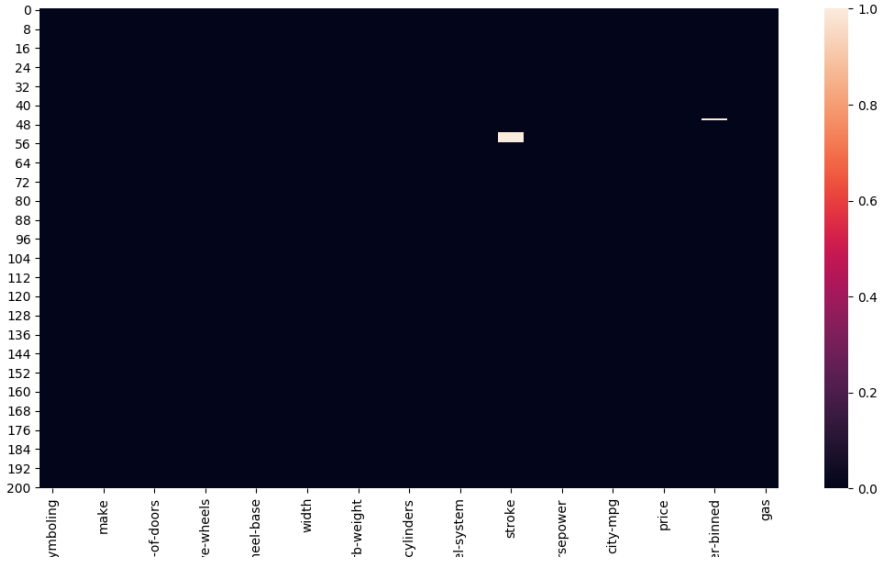
As you can see, analysing a grid of this style to detect missing value is not very convenient, so we will use heatmaps to visually detect these missing values.

# Using Heatmaps for Missing values

- ✓ Heatmap takes a rectangular data grid as input and then assigns a color intensity to each data cell based on the data value of the cell.
- ✓ This is a great way to get visual clues about the data.
- ✓ We will generate a heatmap of the output of isnull() in order to detect missing values.

**Example**: sns.heatmap(df.isnull())
plt.show()

# Using Heatmaps for Missing values




This indicates that “stroke” and “horsepower-binned” columns have few missing values.

# Handling Missing values

We can handle missing values in many ways:

- ✓ **Delete**: You can delete the rows with the missing values or delete the whole column which has missing values. The **dropna()** method from Pandas library can be used to accomplish this task.

- ✓ **Impute**: Deleting data might cause huge amount of information loss. So, **replacing** data might be a better option than deleting. One standard replacement technique is to replace missing values with the **average** value of the entire column. For example, we can replace the missing values in "stroke" column with the mean value of stroke column. The **fillna()** method from Pandas library can be used to accomplish this task.

- ✓ **Predictive filling**: Alternatively, you can choose to fill missing values through predictive filling. The interpolate() method will perform a **linear interpolation** in order to "guess" the missing values and fill the results in the dataset.

# ANOVA (Analysis of Variance)

- ✓ ANOVA is a statistical method which is used for figuring out the relation between different groups of categorical data.

- ✓ The ANOVA test, gives us two measures as result:

    - ✓ **F-test score**: It calculates the variation between sample group means divided by variation within sample group.
    - ✓ **P value**: It shows us the confidence degree. In other words, it tells us whether the obtained result is statistically significant or not.

- ✓ The ANOVA test can be performed using the **f_oneway()** method from Scipy library .

# Correlation

- ✓ Correlation is a statistical metric for measuring to what extent different variables are interdependent.

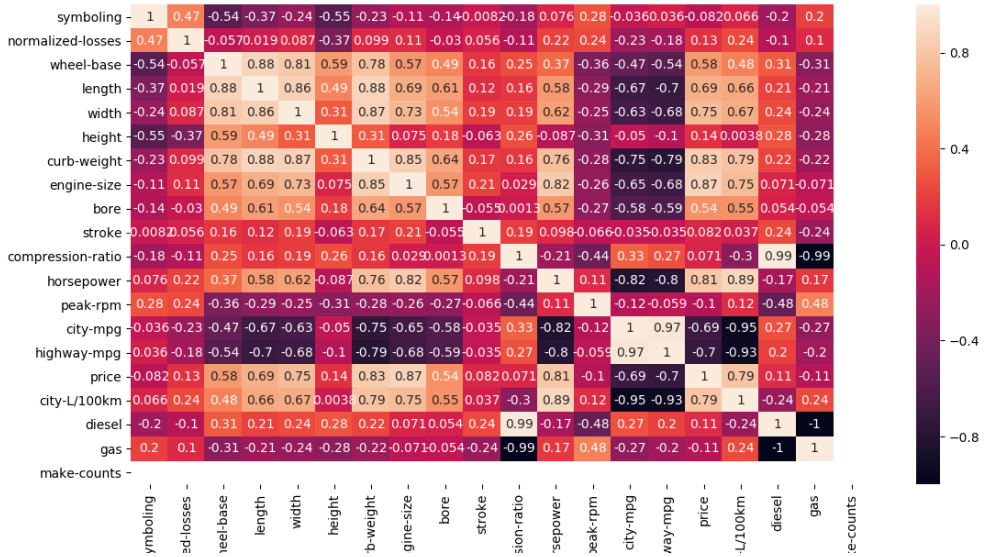
- ✓ In other words, when we look at two variables over time, if one variable changes, how does this effect change in the other variable?

- ✓ For example, smoking is known to be correlated with lung cancer. Since, smoking increases the chances of lung cancer. Another example would be the relationship between the number of hours a student studies and the score obtained by that student. Because, we expect the student who studies more to obtain higher marks in the exam.

- ✓ We can see the correlation between different variables using the corr() function.

- ✓ Then we can plot a heatmap over this output to visualize the results.
  - ✓ **Example**: correlation_matrix = df.corr()
    sns.heatmap(correlation_matrix, annot=True)
    plt.show()

# Heatmap for Correlation




From the above heatmap, we can see that engine size and price are positively correlated(score of 0.87) with each other while, highway-mpg and price are negatively correlated(score of -0.7) with each other. In other words, it tells us that cars with larger engine sizes will be costlier than cars with small engine sizes. It also tells us that expensive cars generally have less MPG as compared to cheaper cars.

# References:


- ✓ https://medium.com/analytics-vidhya/introduction-to-data-wrangling-88c1b5e747cb - Data wrangling
- ✓ https://medium.com/code-heroku/introduction-to-exploratory-data-analysis-eda-c0257f888676 – EDA
- ✓ https://medium.com/@klopmp/etl-using-python-and-pandas-90804bc541ee - ETL
- ✓ https://medium.com/analytics-vidhya/normalization-vs-standardization-8937f45b3e20 – Data preprocessing
- ✓ https://towardsdatascience.com/data-types-in-statistics-347e152e8bee – Data types
- ✓ https://subscription.packtpub.com/book/data/9781838552862/1/ch01lvl1sec07/data-transformation – Encoding categorical data
- ✓ https://medium.com/cracking-the-data-science-interview/an-introduction-to-big-data-data-integration-40715baa7961 – Data integration

# Francesco **Pugliese**