Unit-III

Introduction to NumPy, Pandas, Matplotlib. Exploratory Data Analysis (EDA), Data Science life cycle, Descriptive Statistics, Basic tools (plots, graphs and summary statistics) of EDA, Philosophy of EDA. Data Visualization: Scatter plot, bar chart, histogram, boxplot, heat maps, etc.

NumPy:-

NumPy is a Python library used for working with arrays.
It also has functions for working in domain of linear algebra, fourier transform, and matrices.
NumPy stands for Numerical Python.
In Python we have lists that serve the purpose of arrays, but they are slow to process.
NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.
Arrays are very frequently used in data science.

Why is NumPy Faster Than Lists?

NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently.
This behavior is called locality of reference in computer science.
This is the main reason why NumPy is faster than lists. Also it is optimized to work with latest CPU architectures.

```
import numpy
arr = numpy.array([1, 2, 3, 4, 5])
print(arr)
print(type(arr))
```

output –
[1 2 3 4 5]
<class 'numpy.ndarray'>

Dimensions in Arrays -

A dimension in arrays is one level of array depth (nested arrays).
0-D Arrays:
0-D arrays, or Scalars, are the elements in an array. Each value in an array is a 0-D array.
1-D Arrays:
An array that has 0-D arrays as its elements is called uni-dimensional or 1-D array.
2-D Arrays:
An array that has 1-D arrays as its elements is called a 2-D array.
3-D arrays:
An array that has 2-D arrays (matrices) as its elements is called 3-D array.

Example

```
import numpy as np

a = np.array(42)
b = np.array([1, 2, 3, 4, 5])
c = np.array([[1, 2, 3], [4, 5, 6]])
d = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])

print(a.ndim)
print(b.ndim)
print(c.ndim)
print(d.ndim)
```

output –

```
0
1
2
3
```

Array Indexing -

Array indexing is the same as accessing an array element.
You can access an array element by referring to its index number.
The indexes in NumPy arrays start with 0, meaning that the first element has index 0, and the second has index 1 etc.

```
import numpy as np
arr = np.array([1, 2, 3, 4])
print(arr[0])
```

output – 1

```
[1 2]
1
```

Array Slicing -

Slicing in python means taking elements from one given index to another given index.
We pass slice instead of index like this: [*start*:*end*].

We can also define the step, like this: [*start*:*end*:*step*].
If we don't pass start its considered 0
If we don't pass end its considered length of array in that dimension
If we don't pass step its considered 1

```
arr = np.array([1, 2, 3, 4])
print(arr[0:2])
```

output -
[1,2]

Use the minus operator to refer to an index from the end.

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5, 6, 7])
print(arr[-3:-1])
```

output –
[5,6]

Introduction to Pandas:-

Pandas is a Software Libarary in Computer Programming and it is written for the Python Programming Language its work to do data analysis and manipulation.
import pandas as pd
Here, pd is referred to as an alias to the Pandas. However, it is not necessary to import the library using the alias, it just helps in writing less amount code every time a method or property is called.
Pandas generally provide two data structures for manipulating data, They are:

- Series
- DataFrame

Series:

Pandas Series is a one-dimensional labelled array capable of holding data of any type (integer, string, float, python objects, etc.). The axis labels are collectively called indexes. Pandas Series is nothing but a column in an excel sheet. Labels need not be unique but must be a hashable type.

The object supports both integer and label-based indexing and provides a host of methods for performing operations involving the index.

```
import pandas as pd
import numpy as np
ser = pd.Series()
print(ser)
data = np.array(['g', 'e', 'e', 'k', 's'])
ser = pd.Series(data)
print(ser)
```

output -

```
Series([], dtype: float64)
0  g
1  e
2  e
3  k
4  s
dtype: object
```

DataFrame:

Pandas DataFrame is a two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consists of three principal components, the data, rows, and columns.

```
import pandas as pd

studentData = {
   'name' : ['jack', 'Riti', 'Aadi'],
   'age' : [34, 30, 16],
   'city' : ['Sydney', 'Delhi', 'New york'] }

dfObj = pd.DataFrame(studentData)
print(dfObj)
```

output –

	age	city	name
0	34	Sydney	jack
1	30	Delhi	Riti
2	16	New york	Aadi

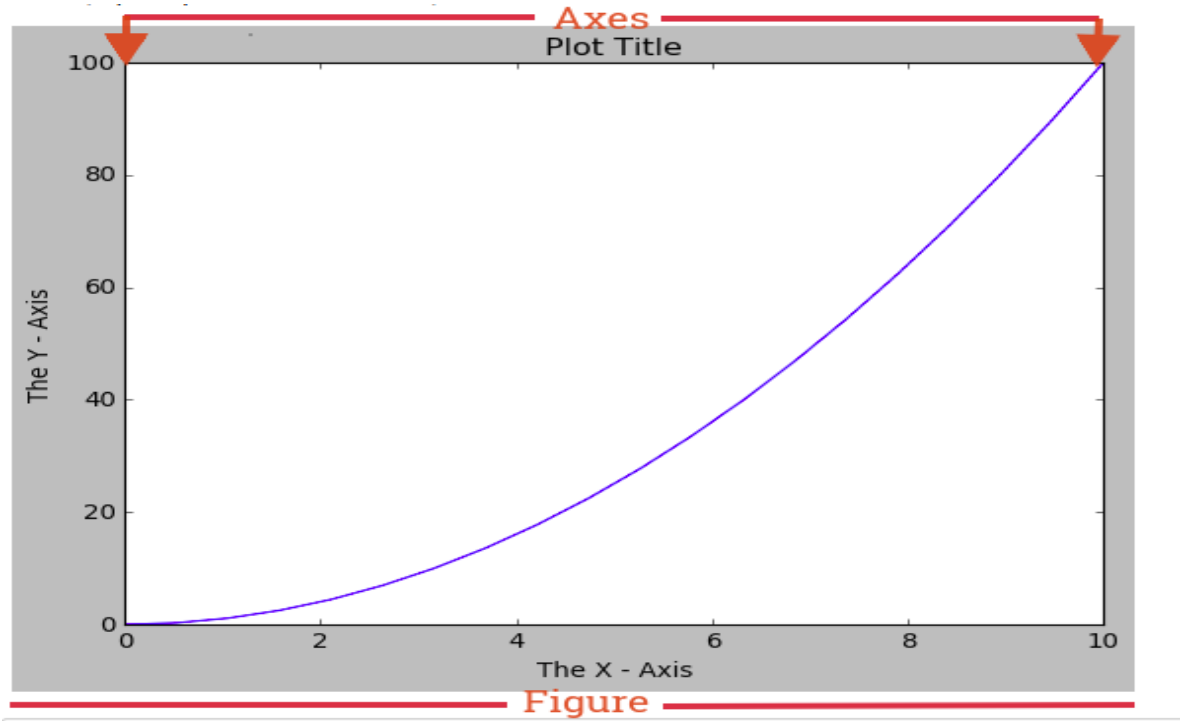
Introduction to Matplotlib:-

Matplotlib is the most popular data visualization library in Python. It allows us to create figures and plots. Plots helps to understand trends, patterns, and to make correlations.
Matplotlib consists of several plots like line, bar, scatter, histogram etc.

Importing matplotlib:
import matplotlib.pyplot as plt

Anatomy of a Plot

There are two key components in a Plot; namely, Figure and Axes.

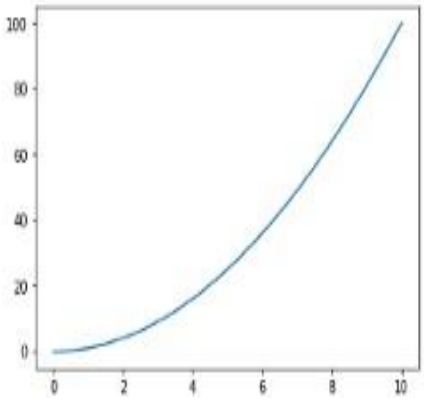


The Figure is the top-level container that acts as the window or page on which everything is drawn. It can contain multiple independent figures, multiple Axes, a subtitle (which is a centered title for the figure), a legend, a color bar, etc.
The Axes is the area on which we plot our data and any labels/ticks associated with it. Each Axes has an X-Axis and a Y-Axis (like in the image above). Let's go ahead to making plots.

Two Approaches for creating Plots

1. **Functional Approach:** Using the basic Matplotlib command, we can easily create a plot. Let's plot an example using two Numpy arrays x and y :

```
In [3]: import numpy as np
        x = np.linspace(0, 10, 20) #Generate 20 datapoints between 0 and 10
        y = x**2                   #Generate array 'y' from square of 'x'

In [4]: plt.plot(x,y)

Out[4]: [<matplotlib.lines.Line2D at 0x7f5a339fcd30>]
```

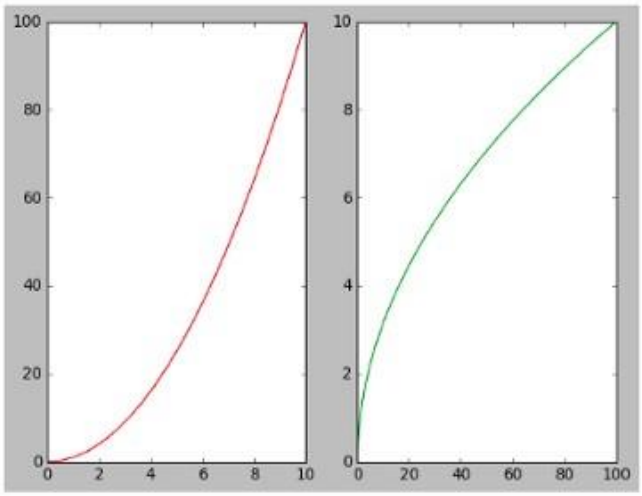


Imagine we needed more than one plot on that canvas. Matplotlib allows us easily create multi-plots on the same figure using the .subplot() method. This .subplot() method takes in three parameters, namely:

- nrows: the number of rows the Figure should have.
- ncols: the number of columns the Figure should have.
- plot_number : which refers to a specific plot in the Figure.

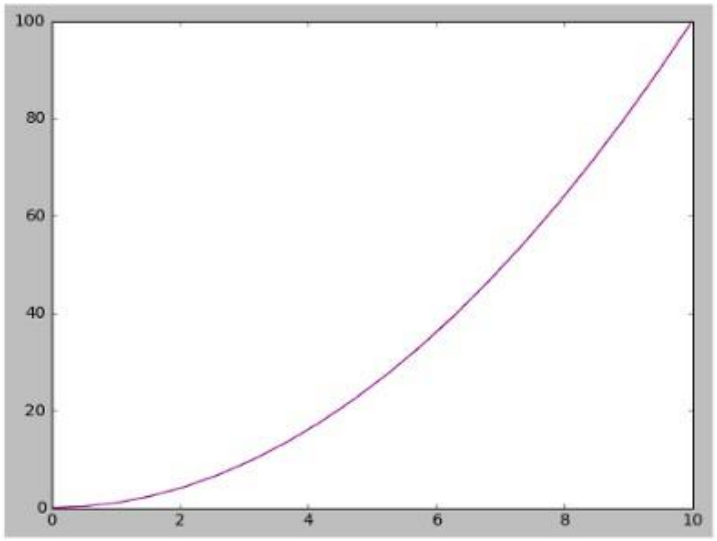
Using .subplot() we will create a two plots on the same canvas:

```
In [13]: # plt.subplot(nrows, ncols, plot_number)
         plt.subplot(1, 2, 1)
         plt.plot(x, y, 'red') # More on color options later

         plt.subplot(1,2,2)
         plt.plot(y, x, 'green');
```



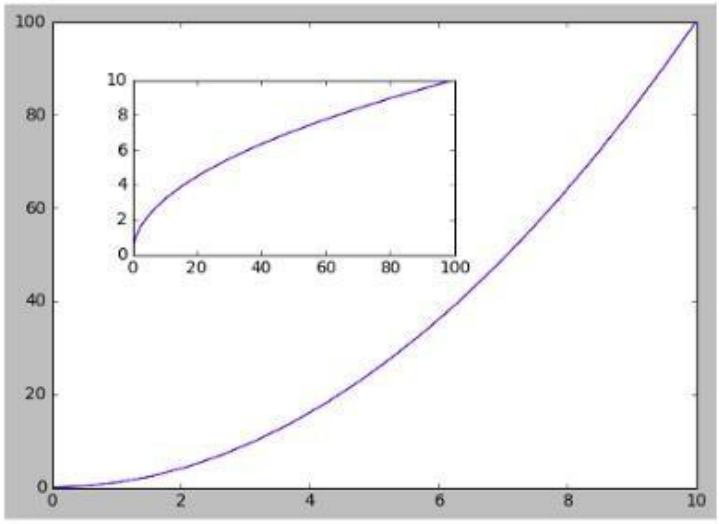
2. **Object oriented Interface:** This is the best way to create plots. The idea here is to create Figure objects and call methods off it. Let's create a blank Figure using the .figure() method.

```
In [18]: fig = plt.figure()
         ax = fig.add_axes([0.1, 0.2, 0.8, 0.9])

         ax.plot(x,y, 'purple')

Out[18]: [<matplotlib.lines.Line2D at 0x7f00630b6208>]
```



Remember, we noted that a Figure can contain multiple figures. Let's try to put in two sets of figures on one canvas:

```
In [26]: fig = plt.figure()

         axes1 = fig.add_axes([0.1, 0.1, 0.8, 0.8])
         axes2 = fig.add_axes([0.2, 0.5, 0.4, 0.3])

         axes1.plot(x,y)
         axes2.plot(y,x)

Out[26]: [<matplotlib.lines.Line2D at 0x7f00630991d0>]
```



Exploratory Data Analysis (EDA):-

What is exploratory data analysis?

The "exploratory" aspect means that your understanding of the problem you are solving, or might solve, is changing as you go.

Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.

EDA assists Data science professionals in various ways:-

1 Getting a better understanding of data
2 Identifying various data patterns
3 Getting a better understanding of the problem statement

Why is exploratory data analysis important in data science?

The main purpose of EDA is to help look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, find interesting relations among the variables.

Data scientists can use exploratory analysis to ensure the results they produce are valid and applicable to any desired business outcomes and goals. EDA also helps stakeholders by confirming they are asking the right questions. EDA can help answer questions about standard deviations, categorical variables, and confidence intervals. Once EDA is complete and insights are drawn, its features can then be used for more sophisticated data analysis or modeling, including machine learning.

Exploratory data analysis tools

Specific statistical functions and techniques you can perform with EDA tools include:

- Clustering and dimension reduction techniques, which help create graphical displays of high-dimensional data containing many variables.
- Univariate visualization of each field in the raw dataset, with summary statistics.
- Bivariate visualizations and summary statistics that allow you to assess the relationship between each variable in the dataset and the target variable you're looking at.
- Multivariate visualizations, for mapping and understanding interactions between different fields in the data.
- K-means Clustering is a clustering method in unsupervised learning where data points are assigned into K groups, i.e. the number of clusters, based on the distance from each group's centroid. The data points closest to a particular centroid will be clustered under the same category. K-means Clustering is commonly used in market segmentation, pattern recognition, and image compression.
- Predictive models, such as linear regression, use statistics and data to predict outcomes.

Types of exploratory data analysis

There are four primary types of EDA:

- **Univariate non-graphical.** This is simplest form of data analysis, where the data being analyzed consists of just one variable. Since it's a single variable, it doesn't deal with

causes or relationships. The main purpose of univariate analysis is to describe the data and find patterns that exist within it.

- **Univariate graphical.** Non-graphical methods don't provide a full picture of the data. Graphical methods are therefore required. Common types of univariate graphics include:
    - Stem-and-leaf plots, which show all data values and the shape of the distribution.
    - Histograms, a bar plot in which each bar represents the frequency (count) or proportion (count/total count) of cases for a range of values.
    - Box plots, which graphically depict the five-number summary of minimum, first quartile, median, third quartile, and maximum.
- **Multivariate nongraphical:** Multivariate data arises from more than one variable. Multivariate non-graphical EDA techniques generally show the relationship between two or more variables of the data through cross-tabulation or statistics.
- **Multivariate graphical:** Multivariate data uses graphics to display relationships between two or more sets of data. The most used graphic is a grouped bar plot or bar chart with each group representing one level of one of the variables and each bar within a group representing the levels of the other variable.

Other common types of multivariate graphics include:

- Scatter plot, which is used to plot data points on a horizontal and a vertical axis to show how much one variable is affected by another.
- Multivariate chart, which is a graphical representation of the relationships between factors and a response.
- Run chart, which is a line graph of data plotted over time.
- Bubble chart, which is a data visualization that displays multiple circles (bubbles) in a two-dimensional plot.
- Heat map, which is a graphical representation of data where values are depicted by color.

Some of the most common data science tools used to create an EDA include:

- **Python:** An interpreted, object-oriented programming language with dynamic semantics. Its high-level, built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for rapid application development, as well as for use as a scripting or glue language to connect existing components together. Python and EDA can be used together to identify missing values in a data set, which is important so you can decide how to handle missing values for machine learning.

- **R:** An open-source programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians in data science in developing statistical observations and data analysis.

Data Science Lifecycle:-

Data Science Lifecycle revolves around the use of machine learning and different analytical strategies to produce insights and predictions from information in order to acquire a commercial enterprise objective. The complete method includes a number of steps like data cleaning,preparation, modelling, model evaluation, etc. It is a lengthy procedure and may additionally take quite a few months to complete. So, it is very essential to have a generic structure to observe for each and every hassle at hand.

Let us understand what is the need for Data Science?

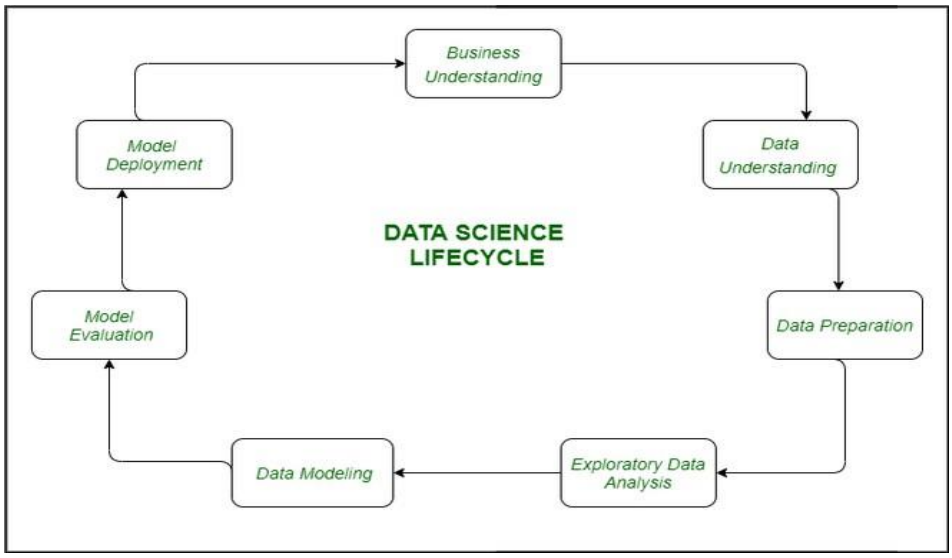
Earlier data used to be much less and generally accessible in a well-structured form, that we could save effortlessly and easily in Excel sheets, and with the help of Business Intelligence tools data can be processed efficiently. But Today we used to deals with large amounts of data every day, which ultimately results in an explosion of records and data.

So it is a very big challenge for any organization to deal with such a massive amount of data generating every second. For handling and evaluating this data we required some very powerful, complex algorithms and technologies and this is where Data science comes into the picture.

The following are some primary motives for the use of Data science technology:

- It helps to convert the big quantity of uncooked and unstructured records into significant insights.
- It can assist in unique predictions such as a range of surveys, elections, etc.
- It also helps in automating transportation such as growing a self-driving car, we can say which is the future of transportation.
- Companies are shifting towards Data science and opting for this technology. Amazon, Netflix, etc, which cope with the big quantity of data, are the use of information science algorithms for higher consumer experience.

The lifecycle of Data Science



1. Business Understanding:

The complete cycle revolves around the enterprise goal. What will you resolve if you do no longer have a specific problem? It is extraordinarily essential to apprehend the commercial enterprise goal sincerely due to the fact that will be your ultimate aim of the analysis. After desirable perception only we can set the precise aim of evaluation that is in sync with the enterprise objective. You need to understand if the customer desires to minimize savings loss, or if they prefer to predict the rate of a commodity, etc.

2. Data Understanding:

After enterprise understanding, the subsequent step is data understanding. This includes a series of all the reachable data. Here you need to intently work with the commercial enterprise group as they are certainly conscious of what information is present, what facts should be used for this commercial enterprise problem, and different information. This step includes describing the data, their structure, their relevance, their records type. Explore the information using graphical plots. Basically, extracting any data that you can get about the information through simply exploring the data.

3. Preparation of Data:

Next comes the data preparation stage. This consists of steps like choosing the applicable data, integrating the data by means of merging the data sets, cleaning it, treating the lacking values through either eliminating them or imputing them, treating inaccurate data through eliminating them, additionally test for outliers the use of box plots and cope with them. Constructing new data, derive new elements from present ones. Format the data into the preferred structure, eliminate undesirable columns and features. Data preparation is the most time-consuming but arguably the most essential step in the complete existence cycle. Your model will be as accurate as your data.

4. Exploratory Data Analysis:

This step includes getting some concept about the answer and elements affecting it, earlier than constructing the real model. Distribution of data inside distinctive variables of a character is explored graphically the usage of bar-graphs, Relations between distinct aspects are captured via graphical representations like scatter plots and warmth maps. Many data visualization strategies are considerably used to discover each and every characteristic individually and by means of combining them with different features.

5. Data Modeling:

Data modeling is the coronary heart of data analysis. A model takes the organized data as input and gives the preferred output. This step consists of selecting the suitable kind of model, whether the problem is a classification problem, or a regression problem or a clustering problem. After deciding on the model family, amongst the number of algorithms amongst that family, we need to cautiously pick out the algorithms to put into effect and enforce them. We need to tune the hyperparameters of every model to obtain the preferred performance. We additionally need to make positive there is the right stability between overall performance and generalizability. We do no longer desire the model to study the data and operate poorly on new data.

6. Model Evaluation:

Here the model is evaluated for checking if it is geared up to be deployed. The model is examined on an unseen data, evaluated on a cautiously thought out set of assessment metrics. We additionally need to make positive that the model conforms to reality. If we do not acquire a quality end result in the evaluation, we have to re-iterate the complete modelling procedure until the preferred stage of metrics is achieved. Any data science solution, a machine learning model, simply like a human, must evolve, must be capable to enhance itself with new data,

adapt to a new evaluation metric. We can construct more than one model for a certain phenomenon, however, a lot of them may additionally be imperfect. The model assessment helps us select and construct an ideal model.

7. Model Deployment:

The model after a rigorous assessment is at the end deployed in the preferred structure and channel. This is the last step in the data science life cycle. Each step in the data science life cycle defined above must be laboured upon carefully. If any step is performed improperly, and hence, have an effect on the subsequent step and the complete effort goes to waste. For example, if data is no longer accumulated properly, you'll lose records and you will no longer be constructing an ideal model. If information is not cleaned properly, the model will no longer work. If the model is not evaluated properly, it will fail in the actual world. Right from Business perception to model deployment, every step has to be given appropriate attention, time, and effort.

Descriptive Statistics:-

Statistics is a branch of mathematics that deals with collecting, interpreting, organization and interpretation of data.

Within statistics, there are two main categories:

1. Descriptive Statistics: In Descriptive Statistics your are describing, presenting, summarizing and organizing your data (population), either through numerical calculations or graphs or tables.

2. Inferential statistics: Inferential Statistics are produced by more complex mathematical calculations, and allow us to infer trends and make assumptions and predictions about a population based on a study of a sample taken from it.

Data availability in today's world is a big boon. However, analyzing them to our needs is the biggest challenge. To be able to analyze the vast resources of data, understanding and describing the data is crucial.
There are different methods through which we can describe data.

What is Descriptive Statistics?

Descriptive Statistics, as the name suggests, describes data. It is a method to collect, organize, summarize, display and analyze sample data taken from a population. Descriptive Statistics, unlike inferential statistics, is not based on probability theory. It paves the way to understand and visualize data better.

Types of Descriptive Statistics

Descriptive Statistics is classified into Measures of Central Tendency and Measure of Dispersion.

A. Measures of Central Tendency

1. Mean/ Average

This measure of central tendency summarizes the data, by considering a value which is an estimate of the total data set. It helps us to ascertain the spread in variables between the minimum and maximum values.

Sample Mean

$$\bar{x} = \frac{\sum x_i}{n}$$

Population Mean

$$\mu = \frac{\sum x_i}{N}$$

Sample Data: 12,18,25,69,45

Sample Mean: [(12+18+25+69+45)/5] = 33.80

Population Data: 55,46,78,12,18,33,28,45,25,69,66

Population Mean: [(55+46+78+12+18+33+28+45+25+69+66)/11)] = 43.18

2. Median

- Median is the middle item in a data set arranged in ascending/descending order.
- If there are n observations then the Median = (n+1)/2 th observation.
- Computational Rule.
- If n is odd, then (n+1)/2 is an integer.
- If n is even, then use an average of n/2 and (n/2) +1th observation.

3. Mode

- Mode is the highest occurring observation.
- The greatest frequency can occur at two or more different values.
- If the data has only two modes, the data is bimodal.
- If the data has more than two modes, the data is multimodal.

4. Percentiles and Quartiles

- The P^{th} percentile in the ordered set is that value below which lies P% (P percent) of the observations in the set.
- The position of the P^{th} percentile is given by (n + 1) P/100, where n is the number of observations in the set.
- Quartiles are special names to percentiles.

Q1 = 25th percentile
Q2 = 50th percentile = median
Q3 = 75th percentile

B. Measures of Dispersion

1. Range

- The range of a data set is the difference between the largest and smallest data values.
- It is the simplest measure of variability.
- It is very sensitive to the smallest and largest data values.
- Range = $X_{max} - X_{min}$

2. Interquartile Range (IQR)

- The interquartile range of a data set is the difference between the third quartile and the first quartile.
- It is the range for the middle 50% of the data.
- It overcomes the sensitivity to extreme data values.

3. Variance

- The variance is a measure of variability that utilizes all the data.

- It is based on the difference between the value of each observation (x_i) and the mean (x for a sample, m for a population).

$$\sigma^2 = \frac{\sum (x_i - \mu)^2}{N}$$

< - Population variance

Sample variance - >

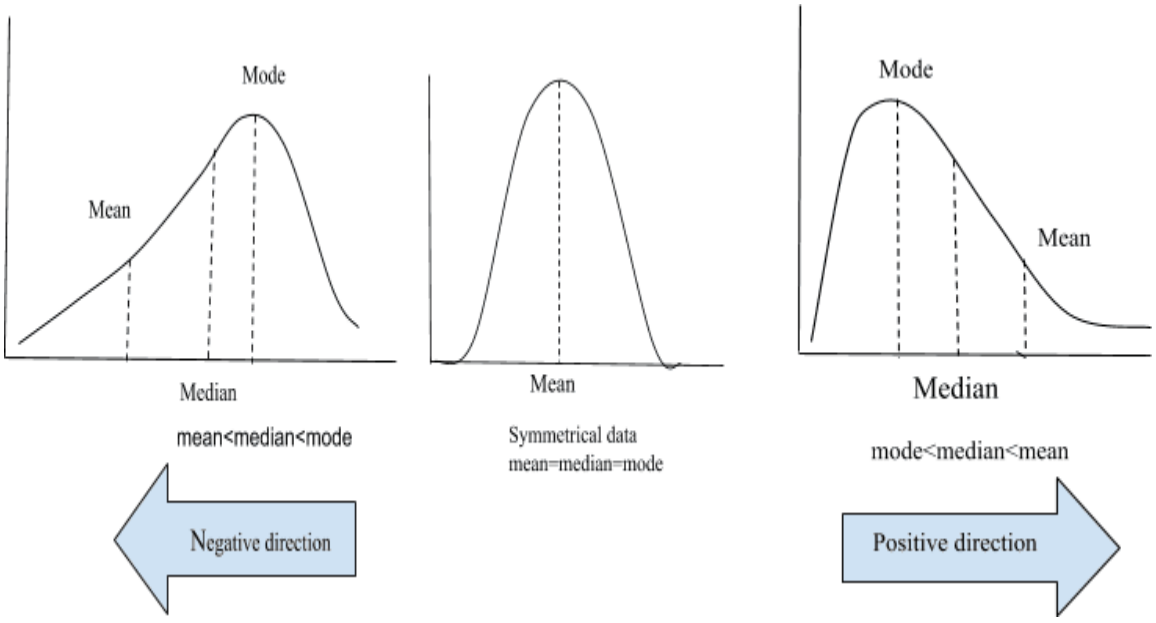
$$s^2 = \frac{\Sigma (x_i - \bar{x})^2}{n-1}$$

4. Standard Deviation

- The standard deviation of a data set is the positive square root of the variance.
- It is measured in the same units as the data, making it more easily comparable, than the variance, to the mean.
- If the data set is a sample, the standard deviation is denoted *s*.
- If the data set is a population, the standard deviation is denoted *s* (sigma).

Skewness -

Skewness characterizes the degree of asymmetry of a distribution around its mean.



The concept of skewness is mainly used to understand the distribution of the data and steps taken to normalize the data for further building of machine learning models.

In case of negatively skewed data, Mean<Median<Mode. This indicates, more data points are to the right of the curve where the data has very high values in large numbers.
In case of positively skewed data, Mode<Median<Mean. This means that more data points are to the left of the curve where the data has very low values in large numbers.

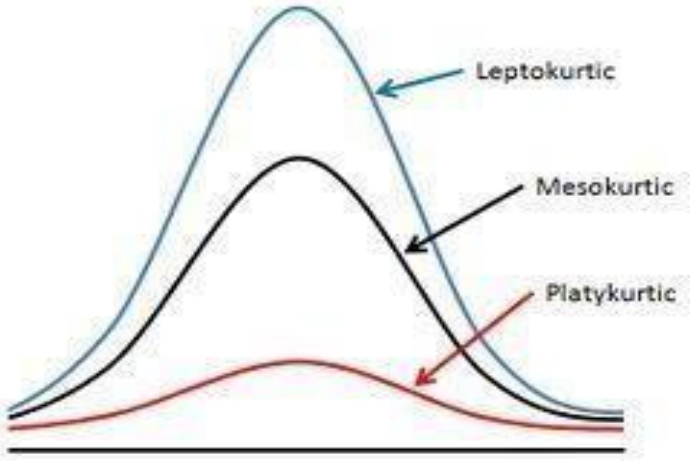
Measure of Skewness

$$\gamma_1 = \frac{\sum(X-\mu)^3}{N\sigma^3}$$

Kurtosis -

Kurtosis characterizes the symmetric distribution through the relative peaks or flatness of the curve.
The main difference between Skewness and Kurtosis is:
Skewness measures the degree of the slope in the frequency distribution.
Kurtosis measures the degree of thickness in the tails of the distribution curve.
There are 3 types of Kurtosis:

- Platykurtic (relatively flat)
- Mesokurtic (normal)
- Leptokurtic (relatively peaked)



Measure of Kurtosis

$$\eta = \beta_2 - 3$$

Where,

$$\beta_2 = \frac{\sum(X-\mu)^4}{N\sigma^4}$$

is the fourth standardized moment or the fourth degree.

Basic tools of EDA:-

The basic tools of EDA are plots, graphs and summary statistics.

Generally speaking, it's a method of systematically going through the data, plotting distributions of all variables (using box plots), plotting time series of data, transforming variables, looking at all pairwise relation- ships between variables using scatterplot matrices, and generating summary statistics for all of them. At the very least that would mean computing their mean, minimum, maximum, the upper and lower quartiles, and identifying outliers.

But as much as EDA is a set of tools, it's also a mindset. And that mindset is about your relationship with the data. You want to understand the data—gain intuition, understand the shape of it, and try to connect your understanding of the process that generated the data to the data itself. EDA happens between you and the data and isn't about proving anything to anyone else yet.

Typical graphical tools used in EDA are:

- Box plot
- Histogram
- Multi-vari chart
- Scatter plot
- Stem-and-leaf plot

Note:
Also explain about different 'plots' in Machine Learning and 'Descriptive statistics'.

Philosophy of EDA:-

There are important reasons anyone working with data should do EDA. Namely, to gain intuition about the data; to make comparisons between distributions; for sanity checking

(making sure the data is on the scale you expect, in the format you thought it should be); to find out where data is missing or if there are outliers; and to summarize the data.
In the context of data generated from logs, EDA also helps with debugging the logging process. For example, “patterns” you find in the data could actually be something wrong in the logging process that needs to be fixed. If you never go to the trouble of debugging, you’ll continue to think your patterns are real. The engineers we’ve worked with are always grateful for help in this area.

In the end, EDA helps you make sure the product is performing as intended.

Although there’s lots of visualization involved in EDA, we distinguish between EDA and data visualization in that EDA is done toward the beginning of analysis, and data visualization as it’s used in our vernacular, is done toward the end to communicate one’s findings. With EDA, the graphics are solely done for you to understand what’s going on. With EDA, you can also use the understanding you get to inform and improve the development of algorithms. For example, suppose you are trying to develop a ranking algorithm that ranks content that you are showing to users. To do this you might want to develop a notion of “popular.”

Before you decide how to quantify popularity (which could be, for example, highest frequency of clicks, or the post with the most number of comments, or comments above some threshold, or some weighted average of many metrics), you need to understand how the data is behaving, and the best way to do that is looking at it and getting your hands dirty.

Plotting data and making comparisons can get you extremely far, and is far better to do than getting a dataset and immediately running a regression just because you know how. It’s been a disservice to analysts and data scientists that EDA has not been enforced as a critical part of the process of working with data.

Data Visualization:-

Data visualization is the process of creating interactive visuals to understand trends, variations, and derive meaningful insights from the data.

This visualization technique aims to identify the Patterns, Trends, Correlations, and Outliers of data sets.

Data visualization is used mainly for data checking and cleaning, exploration and discovery, and communicating results to business stakeholders.

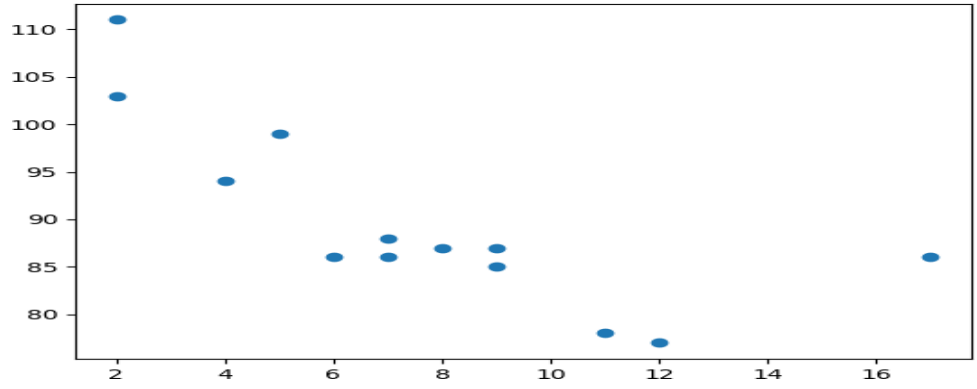
The basic uses of the Data Visualization technique are as follows:

- It is a powerful technique to explore the data with **presentable** and **interpretable** results.
- In the **data mining process**, it acts as a primary step in the pre-processing portion.
- It supports the **data cleaning process** by finding incorrect data and corrupted or missing values.
- It also helps to **construct and select variables**, which means we have to determine which variable to include and discard in the analysis.
- In the process of **Data Reduction**, it also plays a crucial role while combining the categories.

Scatter plot –

A scatter plot is a chart type that is normally used to observe and visually display the relationship between variables. The values of the variables are represented by dots. The positioning of the dots on the vertical and horizontal axis will inform the value of the respective data point; hence, scatter plots make use of Cartesian coordinates to display the values of the variables in a data set.
The relationships observed can either be positive or negative, non-linear or linear, and/or, strong or weak.
The data points or dots, which appear on a scatter plot, represent the individual values of each of the data points and also allow pattern identification when looking at the data holistically.

```
import matplotlib.pyplot as plt
x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]
plt.scatter(x, y)
plt.show()
```

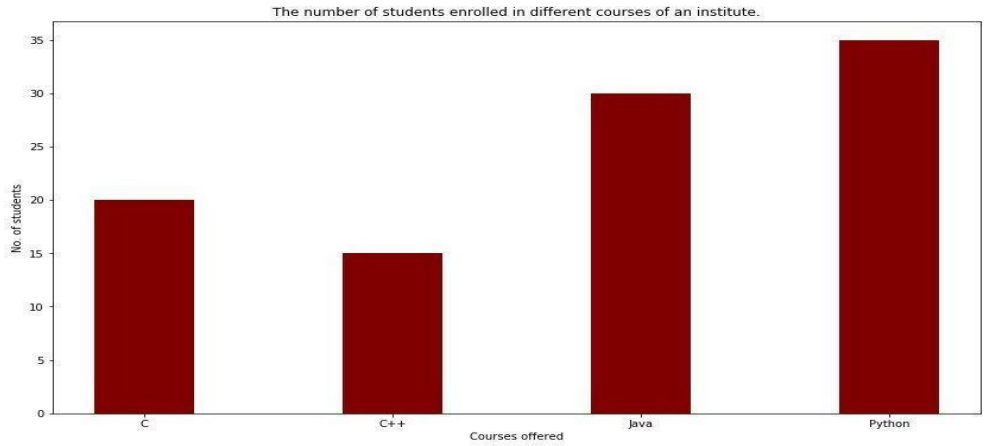


The x-axis represents ages, and the y-axis represents speeds.
What we can read from the diagram is that the two fastest cars were both 2 years old, and the slowest car was 12 years old.

Bar chart –

A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. The bar plots can be plotted horizontally or vertically. A bar chart describes the comparisons between the discrete categories. One of the axis of the plot represents the specific categories being compared, while the other axis represents the measured values corresponding to those categories.
The matplotlib API in Python provides the bar() function which can be used in MATLAB style use or as an object-oriented API.

```
import numpy as np
import matplotlib.pyplot as plt

data = {'C':20, 'C++':15, 'Java':30,'Python':35}
courses =  list(data.keys())
values = list(data.values())
# creating the bar plot
plt.bar(courses, values, color ='maroon',width = 0.4)
plt.xlabel("Courses offered")
plt.ylabel("No. of students enrolled")
plt.title("Students enrolled in different courses")
plt.show()
```



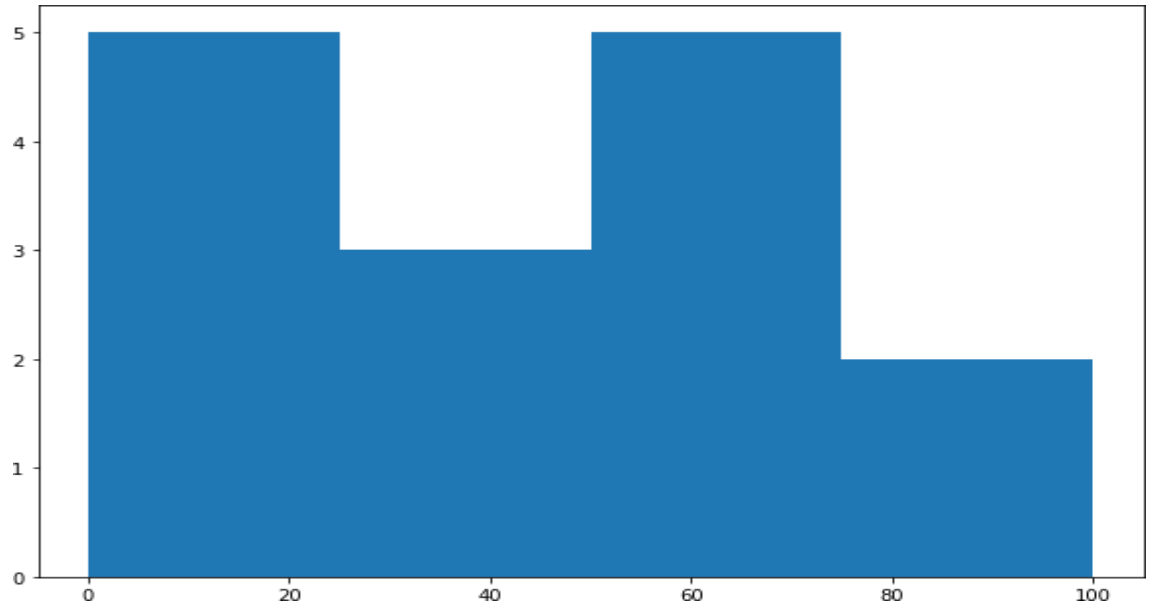
Histogram –

A histogram is basically used to represent data provided in a form of some groups.It is accurate method for the graphical representation of numerical data distribution.It is a type of bar plot where X-axis represents the bin ranges while Y-axis gives information about frequency.

Creating a Histogram

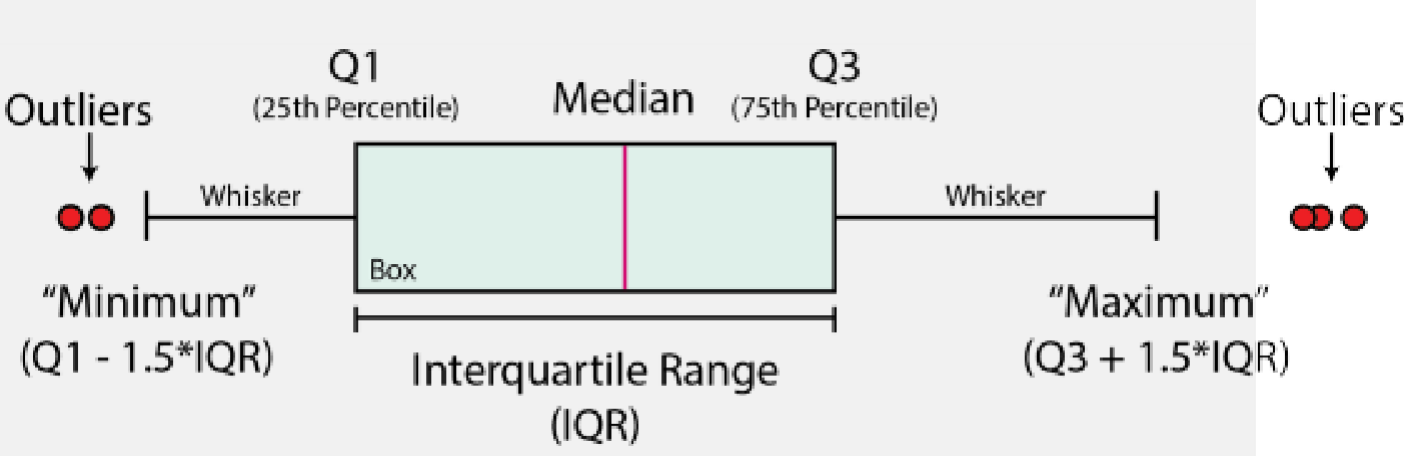
To create a histogram the first step is to create bin of the ranges, then distribute the whole range of the values into a series of intervals, and count the values which fall into each of the intervals.Bins are clearly identified as consecutive, non-overlapping intervals of variables.The matplotlib.pyplot.hist() function is used to compute and create histogram of x.

```
from matplotlib import pyplot as plt
import numpy as np

a = np.array([22, 87, 5, 43, 56,73, 55, 54, 11, 20, 51, 5, 79, 31,27])
# Creating histogram
plt.hist(a, bins = [0, 25, 50, 75, 100])

# Show plot
plt.show()
```



Boxplot –

A boxplot is a graphical and standardised way to display the distribution of data based on five key numbers: The “minimum”, 1st Quartile (25th percentile), median (2nd Quartile./ 50th Percentile), the 3rd Quartile (75th percentile), and the “maximum”. The minimum and maximum values are defined as Q1–1.5 * IQR and Q3 + 1.5 * IQR respectively. Any points that fall outside of these limits are referred to as outliers.



Graphical depiction of a boxplot highlighting key components, including the median, quartiles, outliers, and Interquartile Range. Image created by author.

Boxplots can be used to:

- Identify outliers or anomalous data points
- To determine if our data is skewed
- To understand the spread/range of the data

- To construct a boxplot, we first start with the median value / 50th percentile (Q2). This represents the middle value within our data.
- A box is then formed between the 25th and 75th percentiles (Q1 and Q3 respectively). The range represented by this box is known as the interquartile range (IQR).
- From this box extends lines, which are also known as the whiskers. These extend to Q1–1.5 * IQR and Q3 + 1.5 * IQR or to the last data point if it is less than this value.

Example

Refer 15.a program from PPDS lab record.

Heat maps –

Heatmap is defined as a graphical representation of data using colors to visualize the value of the matrix. In this, to represent more common values or higher activities brighter colors basically reddish colors are used and to represent less common or activity values, darker colors are preferred. Heatmap is also defined by the name of the shading matrix. Heatmaps in Seaborn can be plotted by using the seaborn.heatmap() function.

Syntax:

seaborn.heatmap(data, *, vmin=None, vmax=None, cmap=None, center=None, annot_kws=None, linewidths=0, linecolor='white', cbar=True, **kwargs)

Important Parameters:

data: 2D dataset that can be coerced into an ndarray.
vmin, vmax: Values to anchor the colormap, otherwise they are inferred from the data and other keyword arguments.
cmap: The mapping from data values to color space.
center: The value at which to center the colormap when plotting divergent data.
annot: If True, write the data value in each cell.
fmt: String formatting code to use when adding annotations.
linewidths: Width of the lines that will divide each cell.
linecolor: Color of the lines that will divide each cell.
cbar: Whether to draw a colorbar.

All the parameters except data are optional.

Example

Refer 15.b program from PPDS lab record.