# AHMEDABAD INSTITUTE OF TECHNOLOGY

## DEPARTMENT OF COMPUTER ENGINEERING 5th SEM

## PYTHON FOR DATA SCIENCE (3150713)

## IMPORTANT QUESTIONS

| | |
|---|---|
| 1 | What is Python? List its features. |
| 2 | What is Dictionary? Explain with example. |
| 3 | Explain role of python in data science / Discuss why python is a first choice for data scientists? |
| 4 | Explain Data science pipeline. |
| 5 | Explain sampling in terms of data science? |
| 6 | Discuss the role of indentation in python. |
| 7 | Explain range() function with suitable examples. |
| 8 | List and explain different coding styles supported by python. |
| 9 | What is Jupyter console? |
| 10 | Define an adjacency matrix. Explain with example. |
| 11 | Explain finding missing data in python. |
| 12 | What is Pandas? Explain. |
| 13 | How Xpath is useful for analysis of HTML data? Explain in brief. |
| 14 | Describe date time transformation using datetime module. |
| 15 | How to read data from CSV files using pandas? |
| 16 | How to read data from a text file using pandas library? |
| 17 | Explain working with n-grams. |
| 18 | Explain the Bag of words model. |
| 19 | Explain TF-DF transformations. |
| 20 | How to represent time on axes? |
| 21 | Define histograms. Write code for creating histogram. |
| 22 | What is pie chart and bar chart? Explain how these charts are used for visualizing the data. |
| 23 | Define axis, ticks and grids. |
| 24 | Define graph. What are methods used for visualizing information. |
| 25 | What is box plots? How scatterplots helps for seeing data patterns? |
| 26 | Describe in detail covariance and correlation. |
| 27 | What is EDA? List the functions performed by EDA. How to measure control tendency? |
| 28 | What is memory profiler? |
| 29 | Explain classes used in Scikit-learn. |
| 30 | Define data wrangling. Explain steps performed in data wrangling process. |

Prepared By : Prof. Ankita Rathod

## 1. What is Python? List its features.

Python is an open source, interpreted, high-level, general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Python is dynamically typed and garbage-collected language. Python was conceived in the late 1980s as a successor to the ABC language. Python was created by Guido van Rossum and first released in 1991.

**Features:**

- Python is an easy language
- Readable. The Python language is designed to make developers life easy
- Interpreted Language
- It is portable
- Dynamically-Typed Language
- Object-Oriented
- Popular and Large Community Support
- Open-Source
- Large Standard Library

## 2. What is Dictionary? Explain with example.

- A dictionary (dict) is an unordered collection of zero or more key-value pairs whose keys are object references that refer to hashable objects, and whose values are object references referring to objects of any type.
- Dictionaries are mutable, so we can easily add, edit, or remove items, but as it is unordered we cannot slice the dictionary.
- Dictionary will be represented with the Curly Brackets '**{ key: value }**'

```
my_dict = { 'key1':'value1', 'key2':'value2' }
```

Note: key-value are separated by colon (:) and pairs of key-value are separated by comma (,)

dictdemo.py

```
my_dict = {'college':"KIRC", 'city':"kalol",'type':"engineering"}
print(my_dict['college'])
print(my_dict.get('city'))
```

values can be accessed using key inside square brackets as well as using get() method
Output : *KIRC*
*kalol*

**Dictionary Methods**

- keys() method will return list of all the keys associated with the Dictionary.

keydemo.py

```
my_dict = {'college':"KIRC", 'city':"kalol",'type':"engineering"}
print(my_dict.keys())
```

Output : *['college', 'city', 'type']*

- values() method will return list of all the values associated with the Dictionary.

valuedemo.py

```
my_dict = {'college':"KIRC", 'city':"kalol",'type':"engineering"}
print(my_dict.values())
```

Output : *['KIRC', 'kalol', 'engineering']*

- items() method will return list of tuples for each key value pair associated with the Dictionary.

itemsdemo.py

```
my_dict = {'college':"KIRC", 'city':"kalol",'type':"engineering"}
print(my_dict.items())
```

Output : *[('college', 'KIRC'), ('city', 'kalol'), ('type', 'engineering')]*

## 3. Explain role of python in data science.

- Data analysts are responsible for interpreting data and analyzing the results utilizing statistical techniques and providing ongoing reports.
- They develop and implement data analyses, data collection systems, and other strategies that optimize statistical efficiency and quality. They are also responsible for acquiring data from primary or secondary data sources and maintaining databases.
- Given the right data sources, analysis requirements, and presentation needs, you can use Python for every part of the data science pipeline. In fact, that's precisely what you do in this book. Every example uses Python to help you understand another part of the data science equation.

### 2.3.1 Considering the Shifting Profile of Data Scientists

- A data scientist simply makes the value of data using processes and tools that are machine-learning dependent. The person uses scientific methods and algorithms to gain insights and extract knowledge from structured and unstructured data.
- For example, a data scientist can use a data analytics and visualization application for extracting useful insights from data.
- Apart from excellent math skills, you will need to improve your model building skills, as well. This will help you to work with other people in solving challenging problems.

- From a business perspective, the necessity of fusing data science and application development is obvious : Businesses must perform various sorts of analysis on the huge databases it has collected, to make sense of the information and use it to predict the future.

### 2.3.2 Working with a Multipurpose, Simple, and Efficient Language

- The programming languages R and Python are perhaps the most famous tools in the data industry and it is very natural to ask the question which one do I choose.
- Python is a complete programming language which not only provides support for native data analytics algorithms but also has a rich library for web developments and software developments. Hence, it comes handy when one is looking for analytics embedded within a web page or a standalone application.

- Python support different coding styles :
  1. Functional : Treats every statement as a mathematical equation and avoids any form of state or mutable data.
  2. Imperative : Performs computations as a direct change to program state. This style is useful for manipulating data structures.
  3. Object-oriented : Relies on data fields that are treated as objects and manipulated only through prescribed methods. Its robust library allows coding professionals to choose from a large number of modules as per their specific requirements
  4. Procedural : Treats tasks as step-by-step iterations where common tasks are placed in functions that are called as needed. This coding style favours iteration, sequencing, selection, and modularization.

## 4. Explain Data science pipeline.

- Data science pipelines are sequences of processing and analysis steps applied to data for a specific purpose.
- They're useful in production projects, and they can also be useful if one expects to encounter the same type of business question in the future, so as to save on design time and coding.

**1. Preparing the data**

- Big data enables organizations to gather, store, manage, and manipulate vast amounts of data at the right speed, and the right time, to gain insights.
- The raw data not only may vary substantially in format, but you may also need to transform it to make all the data sources cohesive and amenable to analysis

**2. Performing exploratory data analysis**

- Exploratory Data Analysis (EDA), also known as Data Exploration, is a step in the Data Analysis Process, where a number of techniques are used to better understand the dataset being used.
- By conducting EDA, you can turn an almost useable dataset into a completely useable dataset. The use of trial and error is part of the data science art.

**3. Learning from data :** Discovery is part of being a data scientist

**4. Visualizing :** Visualization means seeing the patterns in the data and then being able to react to those patterns. visualization is the presentation of quantitative information in a graphical form. In other words, data visualizations turn large and small datasets into visuals that are easier for the human brain to understand and process.

## 5. Explain sampling in terms of data science?

Data sampling is a statistical analysis technique used to select, manipulate and analyze a representative subset of data points to identify patterns and trends in the larger data set being examined. It enables data scientists, predictive modelers and other data analysts to work with a small, manageable amount of data about a statistical population to build and run analytical models more quickly, while still producing accurate findings.

**Advantages and challenges of data sampling**

Sampling can be particularly useful with data sets that are too large to efficiently analyze in full -- for example, in big data analytics applications or surveys. Identifying and analyzing a representative sample is more efficient and cost-effective than surveying the entirety of the data or population.

**Types of data sampling methods**

There are many different methods for drawing samples from data; the ideal one depends on the data set and situation. Sampling can be based on probability, an approach that uses random numbers that correspond to points in the data set to ensure that there is no correlation between points chosen for the sample. Further variations in probability sampling include:

- **Simple random sampling:** Software is used to randomly select subjects from the whole population.
- **Stratified sampling:** Subsets of the data sets or population are created based on a common factor, and samples are randomly collected from each subgroup.

- **Cluster sampling:** The larger data set is divided into subsets (clusters) based on a defined factor, then a random sampling of clusters is analyzed.
- **Multistage sampling:** A more complicated form of cluster sampling, this method also involves dividing the larger population into a number of clusters. Second-stage clusters are then broken out based on a secondary factor, and those clusters are then sampled and analyzed. This staging could continue as multiple subsets are identified, clustered and analyzed.
- **Systematic sampling:** A sample is created by setting an interval at which to extract data from the larger population -- for example, selecting every 10th row in a spreadsheet of 200 items to create a sample size of 20 rows to analyze.

Sampling can also be based on nonprobability, an approach in which a data sample is determined and extracted based on the judgment of the analyst. As inclusion is determined by the analyst, it can be more difficult to extrapolate whether the sample accurately represents the larger population than when probability sampling is used.

Nonprobability data sampling methods include:

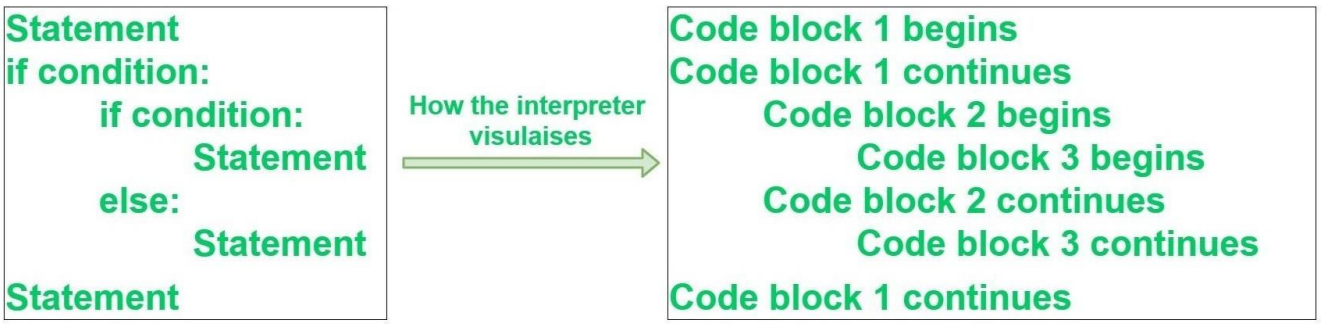
- **Convenience sampling:** Data is collected from an easily accessible and available group.
- **Consecutive sampling:** Data is collected from every subject that meets the criteria until the predetermined sample size is met.
- **Purposive or judgmental sampling:** The researcher selects the data to sample based on predefined criteria.
- **Quota sampling:** The researcher ensures equal representation within the sample for all subgroups in the data set or population.

Once generated, a sample can be used for predictive analytics. For example, a retail business might use data sampling to uncover patterns about customer behavior and predictive modeling to create more effective sales strategies.

## 6. Discuss the role of indentation in python.

- Indentation in Python refers to the (spaces and tabs) that are used at the beginning of a statement. The statements with the same indentation belong to the same group.

- To understand this consider a situation where you are reading a book and all of a sudden all the page numbers from the book went missing. So you don't know, where to continue reading and you will get confused. This situation is similar with Python. Without indentation, Python does not know which statement to execute next or which statement belongs to which block. This will lead to Indentation Error.




In the above example,

- Statement (line 1), if condition (line 2), and statement (last line) belongs to the same block which means that after statement 1, if condition will be executed. and suppose the if condition becomes False then the Python will jump to the last statement for execution.
- The nested if-else belongs to block 2 which means that if nested if becomes False, then Python will execute the statements inside the else condition.
- Statements inside nested if-else belongs to block 3 and only one statement will be executed depending on the if-else condition.
- Python indentation is a way of telling a Python interpreter that the group of statements belongs to a particular block of code. A block is a combination of all these statements. Block can be regarded as the grouping of statements for a specific purpose. Most of the programming languages like C, C++, Java use braces { } to define a block of code. Python uses indentation to highlight the blocks of code. Whitespace is used for indentation in Python. All statements with the same distance to the right belong to the same block of code. If a block has to be more deeply nested, it is simply indented further to the right.

## 7. Explain range() function with suitable examples.

It is an in-built function in Python which returns a sequence of numbers starting from 0 and increments to 1 until it reaches a specified number. The most common use of range function is to iterate sequence type. It is most commonly used in for and while loops.

**Range Parameters**

Following are the range function parameters that we use in python:

- **Start** – This is the starting parameter, it specifies the start of the sequence of numbers in a range function.
- **Stop** – It is the ending point of the sequence, the number will stop as soon as it reaches the stop parameter.

- **Step** – The steps or the number of increments before each number in the sequence is decided by step parameter.

**Syntax**

```
range(start, stop, step)
```

### Example

Create a sequence of numbers from 3 to 5, and print each item in the sequence:

```
x = range(3, 6)
for n in x:
  print(n)
```

**Output**

```
3
4
5
```

## 8. List and explain different coding styles supported by python.

Python is a Programming language that is emerging in a broader way throughout the world. Many programming languages like Java, C++ are object-oriented programming languages that only support object-oriented coding. Unlike other languages, Python provides flexibility for the users to opt for different coding styles.

Different coding styles can be chosen for different problems. There are four different coding styles offered by python. They are

- Functional
- Imperative
- Object-oriented
- Procedural

Let us discuss them individually in more detail.

**Functional coding**

In the functional type of coding, every statement is treated as a Mathematical equation and mutable (able to change) data can be avoided. Most of the programmers prefer this type of coding for recursion and lambda calculus. The merit of this functional coding is it works well for parallel processing, as there is no state to consider. This style of coding is most ly preferred by academics and Data scientists.

**Python Code:**

```
my_list = [1, 5, 4, 6, 8, 11, 3, 12]
new_list = list(map(lambda x: x * 2 , my_list))
print(new_list)
```

**Output:**

[2, 10, 8, 12, 16, 22, 6, 24]

**Imperative coding**

When there is a change in the program, computation occurs. Imperative style of coding is adopted, if we have to manipulate the data structures. This style establishes the program in a simple manner. It is mostly used by Data scientists because of its easy demonstration.

**Python Code:**

```
sum = 0
for x in my_list:
    sum += x
print(sum)
```

**Output:**

50

**Object-oriented coding**

This type of coding relies on the data fields, which are treated as objects. These can be manipulated only through prescribed methods. Python doesn't support this paradigm completely, due to some features like encapsulation cannot be implemented. This type of coding also supports code reuse. Other programming languages generally employ this type of coding.

**Python Code:**

```
class word:
   def test(self):
        print("Python")
string = word()
string.test()
```

**Output:**

Python

**Procedural coding**

Usually, most of the people begin learning a language through procedural code, where the tasks proceed a step at a time. It is the simplest form of coding. It is mostly used by non-programmers

as it is a simple way to accomplish simpler and experimental tasks. It is used for iteration, sequencing, selection, and modularization.

**Python Code:**

```
def add(list):
    sum = 0
    for x in list:
        sum += x
    return sum
print(add(my_list))
```

**Output:**

50

## 9. What is Jupyter console?

- Jupyter is an open source project. The jupyter console is a terminal frontend for kernels using the Jupyter protocol. The console can be installed with: pip install jupyter-console.
- Jupyter is a web application perfect for this task. Jupyter works with Notebooks, documents that mix rich text including beautifully rendered math formulas, blocks of code and code output, including graphics.
- Jupyter main features are:
    1) Inline code execution
    2) Easy idea structuring
    3) Nice displays of pictures and data frame
- Jupyter Notebook is an open source web interface that enables to include text, video, audio, images.
- The main difference between Jupyter console and Jupyter notebook is that the console functions in interactive mode. Whenever you type a line of code, it is immediately executed, and you can see the results.
- If you want to write medium-length pieces of code, do a deep exploration of a dataset to tell a story, the notebook is better. If you want to test out code you are writing, or run quick commands, the console is better.

## 10. Define an adjacency matrix. Explain with example.

- An adjacency matrix represents the connections between nodes of a graph.
- The row and column indices represent the vertices: matrix[i][j] = 1matrix[i][j] = 1 means that there is an edge from vertices ii to jj, and matrix[i][j] = 0matrix[i][j] = 0 denotes that there is no edge between i and j.

- The advantage of the adjacency matrix is that it is simple, and for small graphs it is easy to see which nodes are connected to other nodes.
- Start Python and import NetworkX.
- Different classes exist for directed and undirected networks. Let's create a basic undirected graph:

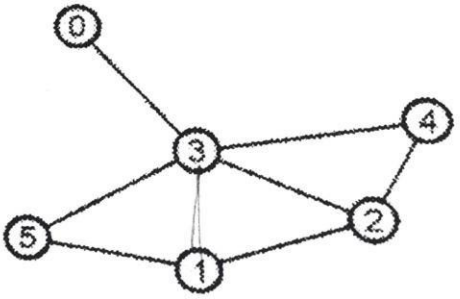
**g = nx.Graph()** # empty graph

- The graph g can be grown in several ways. NetworkX provides many generator functions and facilities to read and write graphs in many formats.

- Example :

```
import networkx as nx

# Create a networkx graph object
my_graph = nx.Graph()

# Add edges to to the graph object
# Each tuple represents an edge between two nodes
my_graph.add_edges_from([ (1,2), (1,3), (3,4), (1,5), (3,5), (4,2), (2,3), (3,0)])

# Draw the resulting graph
nx.draw(my_graph, with_labels=True, font_weight='bold')
```

**Output :**




## 11. Explain finding missing data in python.

- Data can have missing values for a number of reasons such as observations that were not recorded and data corruption. Handling missing data is important as many machine learning algorithms do not support data with missing values.
- You can load the dataset as a Pandas DataFrame and print summary statistics on each attribute.

```
# load and summarize the dataset
from pandas import read_csv
# load the dataset
dataset = read_csv('csv file name', header=None)
# summarize the dataset
print(dataset.describe())
```

- In Python, specifically Pandas, NumPy and Scikit-Learn, we mark missing values as NaN. Values with a NaN value are ignored from operations like sum, count, etc.
- Use the isnull() method to detect the missing values. Pandas Dataframe provides a function isnull(), it returns a new dataframe of same size as calling dataframe, it contains only True & False only. With True at the place NaN in original dataframe and False at other places.

**Encoding missingness:**

- The fillna() function is used to fill NA/NaN values using the specified method.
- Syntax:

```
DataFrame.fillna(value=None, method=None, axis=None, inplace=False, limit=None,
downcast=None, **kwargs)
```

Where

1. value : It is a value that is used to fill the null values.
2. method : A method that is used to fill the null values.
3. axis : It takes int or string value for rows/columns.
4. inplace : If it is True, it fills values at an empty place.
5. limit : It is an integer value that specifies the maximum number of consecutiv forward/backward NaN value fills.
6. downcast : It takes a dict that specifies what to downcast like Float64 to int64.

## 12. What is Pandas? Explain.

- Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame.
- DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.
- Pandas is built on top of the Numpy package, meaning a lot of the structure of the Numpy is used or replicated in Pandas. Data in pandas is often used to feed statistical analysis in Scipy, plotting functions from Matplotlib, and machine learning algorithms in Scikit-learn.
- Pandas is the library for data manipulation and analysis. Usually, it is the starting point for your data science tasks. It allows you to read/write data from/to multiple sources. Process the missing

data, align your data, reshape it, merge and join it with other data, search data, group it, slice it.

## 13. How Xpath is useful for analysis of HTML data? Explain in brief.

- XPath specifies path expression that matches XML data by navigating down the tree. XPath is used as the embedded query language for both XQuery 1.0 and XSLT 2.0.
- XPath provides a common syntax and semantics for functionality shared between XSLT and XPointer. XPath is used in XSL transformations to find information in an XML document. It is used to navigate through elements and attributes in XML documents.
- Parsers are represented by parser objects. There is support for parsing both XML and (broken) HTML.

```
from pandas.io.parsers import TextParser
def parse_options_data(table):
 rows = table.findall('.//tr')
 header = _unpack(rows[0], kind='th')
 data = [_unpack(r) for r in rows[1:]]
 return TextParser(data, names=header).get_chunk()
```

## 14. Describe date time transformation using datetime module.

- Dates are often provided in different formats and must be converted into single format DateTime objects before analysis.
- Python provides two methods of formatting date and time.
  1. str() = it turns a datetime value into a string without any formatting.
  2. strftime() function = it define how user want the datetime value to appear after conversion.

**1. Using pandas.to_datetime() with a date**

```
import pandas as pd
# input in mm.dd.yyyy format
date = ['21.07.2020']
# output in yyyy-mm-dd format
print(pd.to_datetime(date))
```

**2. Using pandas.to_datetime() with a date and time**

```
import pandas as pd
# date (mm.dd.yyyy) and time (H:MM:SS)
date = ['21.07.2020 11:31:01 AM']
# output in yyyy-mm-dd HH:MM:SS
print(pd.to_datetime(date))
```

- We can convert a string to datetime using strptime() function. This function is available in datetime and time modules to parse a string to datetime and time objects respectively.
- Python strptime() is a class method in datetime class. Its syntax is:

  *datetime.strptime(date_string, format)*
- Both the arguments are mandatory and should be string

```
import datetime
format = "%a %b %d %H:%M:%S %Y"
today = datetime.datetime.today()
print 'ISO     :', today
s = today.strftime(format)
print 'strftime:', s
d = datetime.datetime.strptime(s, format)
print 'strptime:', d.strftime(format)
$ python datetime_datetime_strptime.py

ISO     : 2013-02-21 06:35:45.707450
strftime: Thu Feb 21 06:35:45 2013
strptime: Thu Feb 21 06:35:45 2013
```

## 15. How to read data from CSV files using pandas?

- Flat files are data files that contain records with no structured relationships between the records. A flat file presents the easiest kind of file to work with.
- A flat file can be a plain text file having a TSV, CSV format, or a binary file format.
- Comma Separated Values (CSV) files, which contain data values that are separated by ,. For example, a file saved with name "Data" in "CSV" format will appear as "Data.csv". By noticing ".csv" extension we can clearly identify that it is a "CSV" file and data is stored in a tabular format.

- The CSV file used for an example is quite simple :
  a) A header defines each of the fields
  b) Fields are separated by commas
  c) Records are separated by linefeeds
  d) Strings are enclosed in double quotes
  e) Integers and real numbers appear without double quotes
- Applications such as Excel can import and format CSV files so that they become easier to read.
- An **Excel file** uses a complex method to separate data fields and to provide a wealth of information about each field. Excel actually recognizes the header as a header.
- It is an XML-based file format created by Microsoft Excel. The XLSX format was introduced with Microsoft Office 2007.
- In XLSX data is organized under the cells and columns in a sheet. Each XLSX file may contain one or more sheets. So a workbook can contain multiple sheets.
- Let's load the data from XLSX file and define the sheet name. For loading the data you can use the Pandas library in python.

```
# read Excel file into a DataFrame
df = pd.read_excel(r'./Importing files/Indiacity.xlsx')
# print values
df
```

## 16. How to read data from a text file using pandas library?

- Python provides three functions to read data from a text file :
  a) read(n) : This function reads n bytes from the text files or reads the complete information from the file if no number is specified. It is smart enough to handle the delimiters when it encounters one and separates the sentences
  b) readline(n) : This function allows you to read n bytes from the file but not more than one line of information
  c) readlines() : This function reads the complete information in the file but unlike read(), it doesn't bother about the delimiting character and prints them as well in a list format
- The following structure is typical for a CSV file:
  a) Definition of the columns in the header of the table
  b) A character is used to separate individual records
  c) A character is used to separate individual columns (commas, tabs, or spaces)
  d) Field delimiter HTML Special Character within the file to avoid confusion with the separators

## 17. Explain working with n-grams.

- The essential concepts in text mining is n-grams, which are a set of co-occurring or continuous sequence of n items from a sequence of large text or sentence. The item here could be words, letters, and syllables.
- 1-gram is also called as unigrams are the unique words present in the sentence. Bigram(2-gram) is the combination of 2 words. Trigram(3-gram) is 3 words and so on.
- An n-gram is a continuous sequence of items in the text you want to analyze. The items are phonemes, syllables, letters, words, or base pairs. The n in n-gram refers to a size.
- The code below generates n-grams in python :

```
import re

def generate_ngrams(text,n):

  # split sentences into tokens
  tokens=re.split("\\s+",text)
  ngrams=[]

  # collect the n-grams
  for i in range(len(tokens)-n+1):
    temp=[tokens[j] for j in range(i,i+n)]
    ngrams.append(" ".join(temp))

  return ngrams
```

## 18. Explain the Bag of words model.

- Bag of words simply refers to a matrix in which the rows are documents and the columns are words. The values matching a document with a word in the matrix, could be a count of word occurrences within the document or use tf-idf.
- Classifiers are used to train the bag of words and a special kind of algorithm used to break words down into categories.
- Traditionally, text documents are represented in NLP as a bag-of-words. This means that each document is represented as a fixed-length vector with length equal to the vocabulary size.
- Each dimension of this vector corresponds to the count or occurrence of a word in a document. Being able to reduce variable-length documents to fixed-length vectors makes them more amenable for use with a large variety of Machine Learning (ML) models and tasks.
- Fig. 3.7.1 shows turning raw text into a bag of words representation.

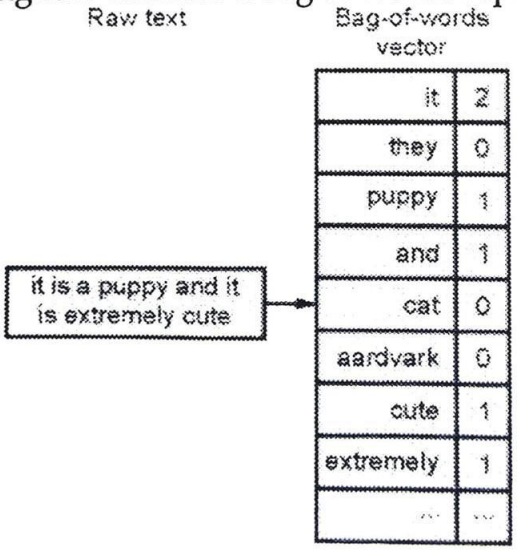



**Fig. 3.7.1 Turning raw text into a bag of words representation**

## 19. Explain TF-DF transformations.

- TF-IDF stands for "Term Frequency, Inverse Data Frequency"
- Term Frequency (tf) gives us the frequency of the word in each document. Inverse Data Frequency (idf) is used to calculate the weight of rare words across all documents.

- Example :

```
from sklearn.feature_extraction.text import TfidfTransformer
>>> from sklearn.feature_extraction.text import CountVectorizer
>>> from sklearn.pipeline import Pipeline
>>> import numpy as np
>>> corpus = ['this is the first document',
...           'this document is the second document',
...           'and this is the third one',
...           'is this the first document']
>>> vocabulary = ['this', 'document', 'first', 'is', 'second', 'the',
...               'and', 'one']
>>> pipe = Pipeline([('count', CountVectorizer(vocabulary=vocabulary)),
...                  ('tfid', TfidfTransformer())]).fit(corpus)
>>> pipe['count'].transform(corpus).toarray()
array([[1, 1, 1, 1, 0, 1, 0, 0],
       [1, 2, 0, 1, 1, 1, 0, 0],
       [1, 0, 0, 1, 0, 1, 1, 1],
       [1, 1, 1, 1, 0, 1, 0, 0]])
>>> pipe['tfid'].idf_
array([1.        , 1.22314355, 1.51082562, 1.        , 1.91629073,
       1.        , 1.91629073, 1.91629073])
>>> pipe.transform(corpus).shape
(4, 8)
```

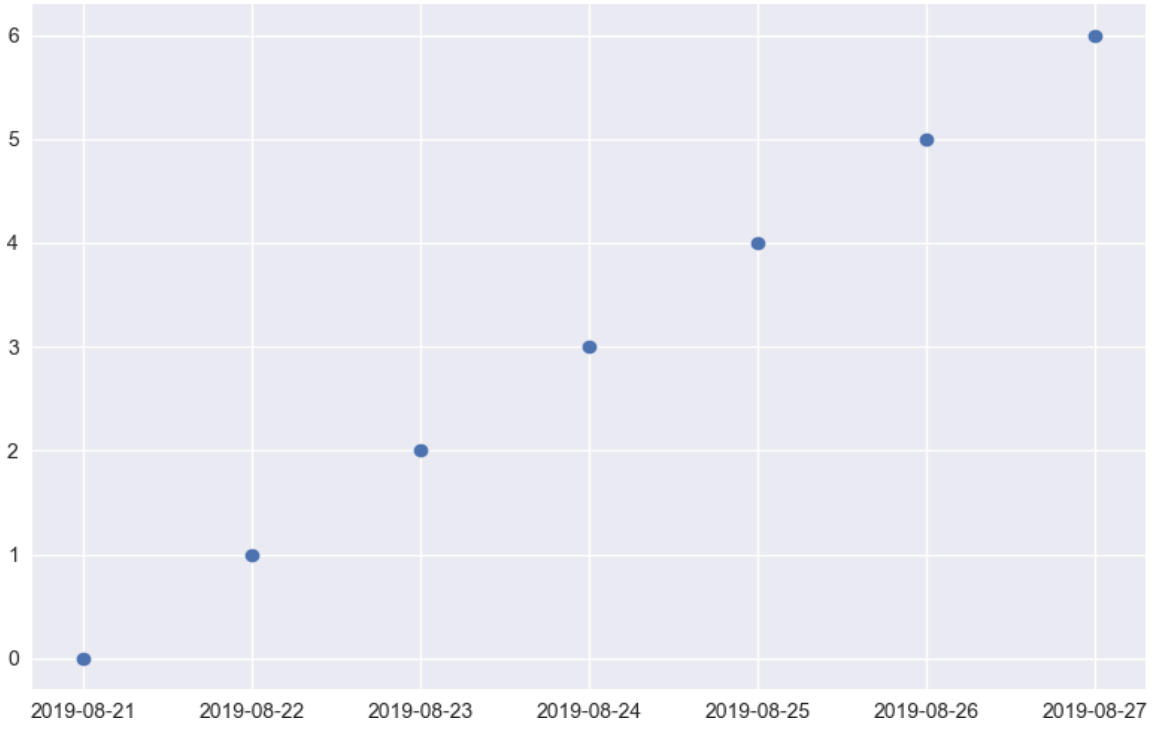
## 20. How to represent time on axes?

- First of all, we will create a scatter plot of dates and values in Matplotlib using plt.plot_date(). We will be using Python's built-in module called datetime(datetime, timedelta) for parsing the dates. So, let us create a python file called 'plot_time_series.py' and make necessary imports.
- Example :

```
# plot_time_series.py

import matplotlib.pyplot as plt
from datetime import datetime, timedelta
plt.style.use('seaborn')

dates = [
```

```
    datetime(2019, 8, 21),
    datetime(2019, 8, 22),
    datetime(2019, 8, 23),
    datetime(2019, 8, 24),
    datetime(2019, 8, 25),
    datetime(2019, 8, 26),
    datetime(2019, 8, 27),
]

y = [0, 1, 2, 3, 4, 5, 6]

plt.plot_date(dates, y)
plt.tight_layout()
plt.show()
```

- This will create a simple scatter plot for the time series data.



## 21. Define histograms. Write code for creating histogram.

- In a histogram, the data are grouped into ranges (e.g. 10–19, 20–29) and then plotted as connected bars. Each bar represents a range of data. The width of each bar is proportional to the width of each category, and the height is proportional to the frequency or percentage of that category.
- It provides a visual interpretation of numerical data by showing the number of data points that fall within a specified range of values called "bins".
- Fig. 4.3.1 shows histogram.

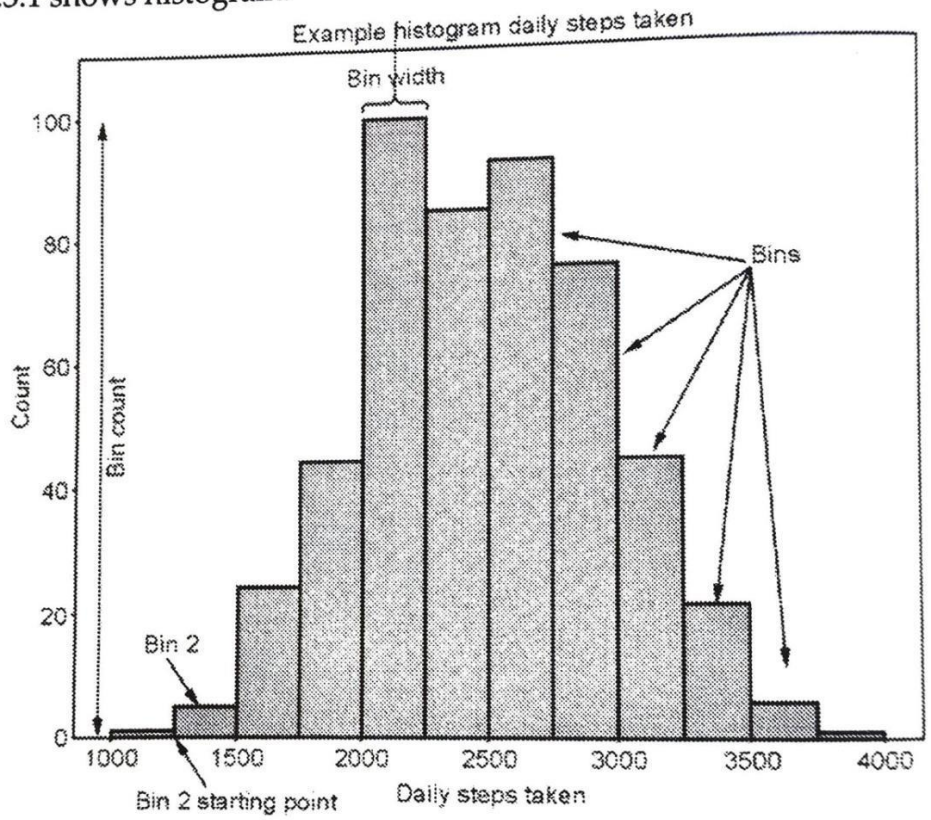



**Fig. 4.3.1 Histogram**

- Histograms can display a large amount of data and the frequency of the data values. The median and distribution of the data can be determined by a histogram. In addition, it can show any outliers or gaps in the data.
- Matplotlib provides a dedicated function to compute and display histograms: plt.hist()
- Code for creating histogram with randomized data:

```
import numpy as np
import matplotlib.pyplot as plt
x = 40 * np.random.randn(50000)
```

```
plt.hist(x, 20, range=(-50, 50), histtype='stepfilled',
align='mid', color='r', label='Test Data')
plt.legend()
plt.title(' Histogram')
plt.show()
```

## 22. What is pie chart and bar chart? Explain how these charts are used for visualizing the data.

**Pie Chart:-**

- A type of graph is which a circle is divided into sectors that each represent a proportion of whole. Each sector shows the relative size of each value.
- A pie chart displays data, information and statistics in an easy to read "pie slice" format with varying slice sizes telling how much of one data element exists.
- Fig. 4.2.1 shows pie chart.

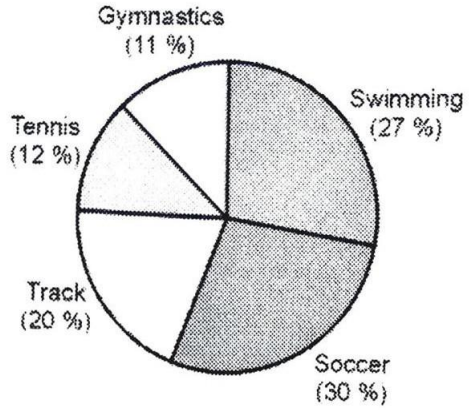



**Fig. 4.2.1 Pie chart**

- Pie chart is also known as circle graph. The bigger the slice, the more of that particular data was gathered. The main use of a pie chart is to show comparisons.
- Pie charts can be drawn using the function pie() in the pyplot module. The below python code example draws a pie chart using the pie() function.
- By default the pie() fucntion of pyplot arranges the pies or wedges in a pie chart in counter clockwise direction.

- Example:

```
# import the pyplot library
import matplotlib.pyplot as plotter

# The slice names of a student distribution pie chart
pieLabels  = 'Rakshita', 'Ritesh', 'Rupali', 'Rutu', 'Rushi', 'Radhika'

# marks data
marksShare    = [59.69, 16, 9.94, 7.79, 5.68, 0.54]
figureObject, axesObject = plotter.subplots()

# Draw the pie chart
axesObject.pie(marksShare, labels=pieLabels, autopct='%1.2f', startangle=90)

# Aspect ratio - equal means pie is a circle
axesObject.axis('equal')

plotter.show()
```

- The essential part of a pie chart is the values. You could create a basic pie chart using just the values as input.

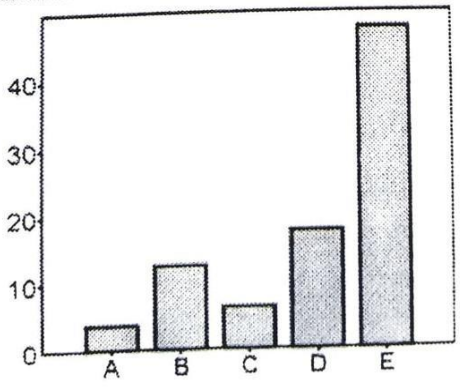
**Bar Chart:-**

- A bar chart is a way of summarizing a set of categorical data. The bar chart displays data using a number of bars, each representing a particular category. The height of each bar is proportional to a specific aggregation.
- The categories could be something like an age group or a geographical location. It is also possible to color or split each bar into another categorical column in the data, which enables you to see the contribution from different categories to each bar or group of bars in the bar chart.
- Bar charts make comparing values easy.
- **Example:**

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

# Make a fake dataset:
height = [3, 12, 5, 18, 45]
bars = ('A', 'B', 'C', 'D', 'E')
y_pos = np.arange(len(bars))
```

```
# Create bars
plt.bar(y_pos, height)

# Create names on the x-axis
plt.xticks(y_pos, bars)

# Show graphic
plt.show()
```

**Output:**

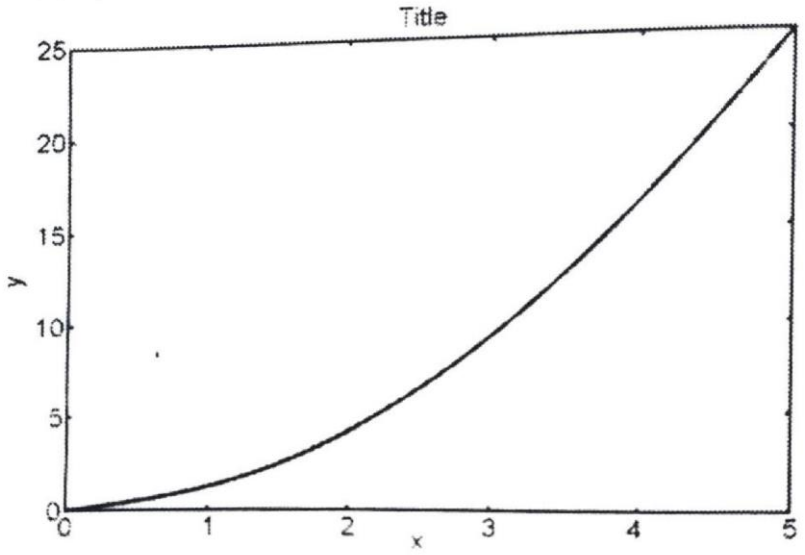



## 23. Define axis, ticks and grids.

- The axes define the x and y plane of the graphic. The x axis runs horizontally, and the y axis runs vertically.
- An axis is added to a plot layer. Axis can be thought of as sets of x and y axis that lines and bars are drawn on. An Axis contains daughter attributes like axis labels, tick labels, and line thickness.
- The following code shows how to obtain access to the axes for a plot.

```
fig = plt.figure()
axes = fig.add_axes([0.1, 0.1, 0.8, 0.8])      # left, bottom, width, height (range 0 to 1)
axes.plot(x, y, 'r')
axes.set_xlabel('x')
axes.set_ylabel('y')
axes.set_title('title');
```

Output:




- A grid can be added to a Matplotlib plot using the plt.grid() command. By defaut, the grid is turned off. To turn on the grid use:

```
plt.grid(True)
```

- The only valid options are plt.grid(True) and plt.grid(False). Note that True and False are capitalized and are not enclosed in quotes.

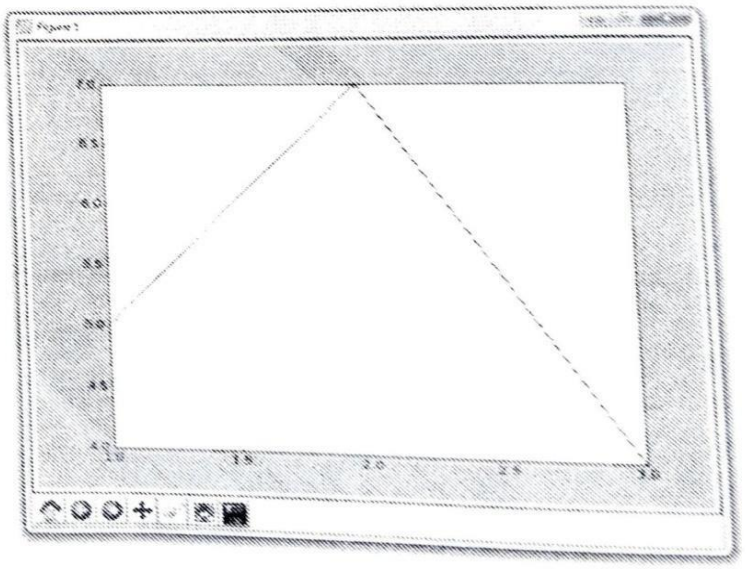
## 24. Define graph. What are methods used for visualizing information.

- Data visualization is the presentation of quantitative information in a graphical form. In other words, data visualizations turn large and small datasets into visuals that are easier for the human brain to understand and process.
- Good data visualizations are created when communication, data science, and design collide. Data visualizations done right offer key insights into complicated datasets in ways that are meaningful and intuitive.
- A graph is simply a visual representation of numeric data. MatPlotLib supports a large number of graph and chart types.
- Matplotlib is a popular Python package used to build plots. Matplotlib can also be used to make 3D plots and animations.
- Line plots can be created in Python with Matplotlib's pyplot library. To build a line plot, first import Matplotlib. It is a standard convention to import Matplotlib's pyplot library as plt.

- To define a plot, you need some values, the matplotlib.pyplot module, and an idea of what you want to display.

```
import matplotlib.pyplot as plt
plt.plot([1,2,3],[5,7,4])
plt.show()
```

- The plt.plot will "draw" this plot in the background, but we need to bring it to the screen when we're ready, after graphing everything we intend to.
- plt.show() : With that, the graph should pop up. If not, sometimes it can pop under or you may have gotten an error. Your graph should look like:



- Methods used for visualizing information are as follows:

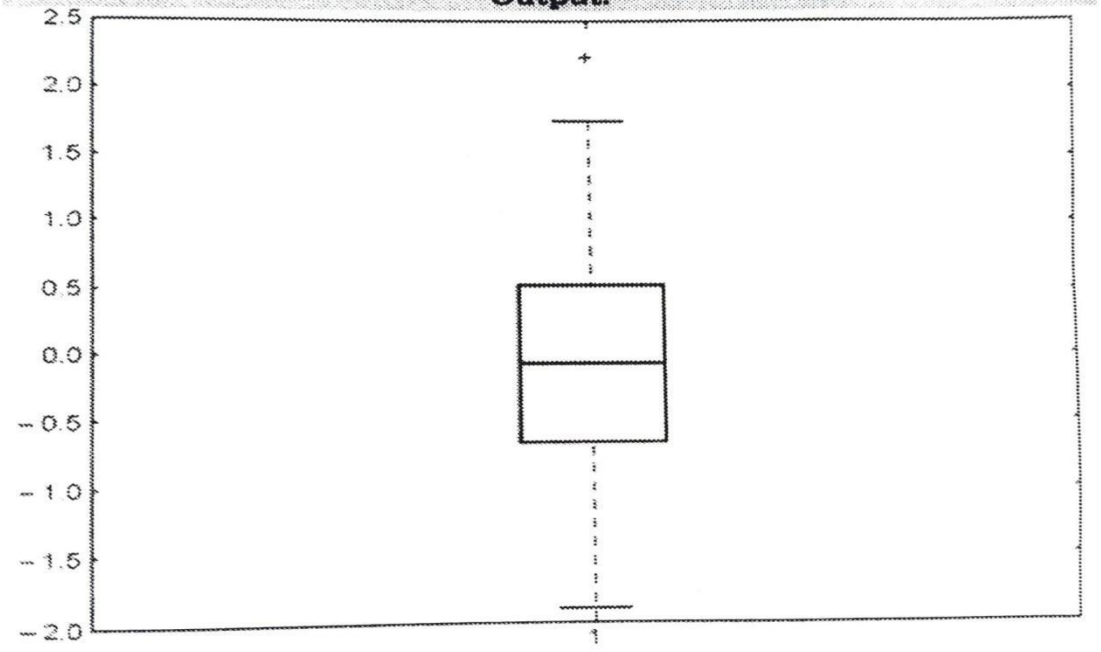
1) Drawing Multiple Lines and Plots
2) Saving your work to Disk
3) Setting the Axis, Ticks, Grids
4) Defining the Line Appearance & Working with Line Style
5) Adding Markers
6) Using Labels, Annotations, and Legends

## 25. What is box plots? How scatterplots helps for seeing data patterns?

- Box plot uses boxes and lines to depict the distributions of one or more groups of numeric data. Box plots are used to show distributions of numeric data values, especially when you want to compare them between multiple groups.
- A box plot may also have lines, called whiskers, indicating data outside the upper and lower quartiles.
- In most cases, it is possible to use numpy or Python objects, but pandas objects are preferable because the associated names will be used to annotate the axes. Additionally, you can use Categorical types for the grouping variables to control the order of plot elements.
- The following example shows how to create a box plot with randomized data:

```
import numpy as np
import matplotlib.pyplot as plt
data = np.random.randn(100)
plt.boxplot(data)
plt.show()
```

**Output:**




## 26. Describe in detail covariance and correlation.

- Correlation gives the changes in one variable due to changes in the other variable.
- Scatter diagram, correlation coefficient and rank correlation coefficient are the common

measures of correlation coefficients.

- The Pearson's correlation is the foundation for complex linear estimation models.

**Using covariance and correlation**

- Covariance is the first measure of the relationship of two variables. It determines whether both variables have a coincident behavior with respect to their mean.
- If the single values of two variables are usually above or below their respective averages, the two variables have a positive association.
- Consider the two plots shown below. In both images, one thousand samples drawn from an underlying joint distribution. Clearly the two distributions are different and shown in Fig. 5.8.1. However, the mean and variance are the same in both the x and the y dimension. What is different ?

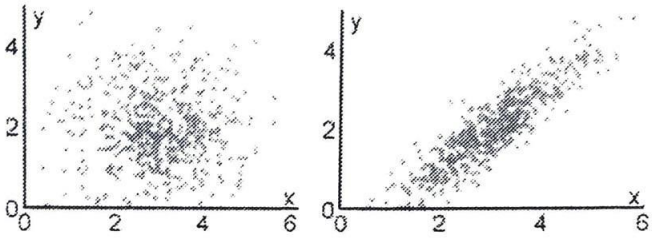



**Fig. 5.8.1 Distribution**

- Covariance is a quantitative measure of the extent to which the deviation of one variable from its mean matches the deviation of the other from its mean.
- Correlation : Covariance is interesting because it is a quantitative measurement of the relationship between two variables. Correlation between two random variables, $\varrho(X,Y)$ is the covariance of the two variables normalized by the variance of each variable. This normalization cancels the units out and normalizes the measure so that it is always in the range [0, 1].
- When people use the term correlation, they are actually referring to a specific type of correlation called "Pearson" correlation. It measures the degree to which there is a linear relationship between the two variables.
- An alternative measure is "Spearman" correlation, which has a formula almost identical to the correlation defined above, with the exception that the underlying random variables are first transformed into their rank
- The difference between variance, covariance, and correlation is :

a. Variance is a measure of variability from the mean

b. Covariance is a measure of relationship between the variability of 2 variables - covariance is scale dependent because it is not standardized

c. Correlation is a of relationship between the variability of of 2 variables - correlation is standardized making it not scale dependent

- The covariance matrix element Cij is the covariance of xi and xj. The element Cii is the variance of xi.
  a. If COV(xi, xj) = 0 then variables are uncorrelated
  b. If COV(xi, xj) > 0 then variables positively correlated
  c. If COV(xi, xj) > < 0 then variables negatively correlated
- **Syntax:**

numpy.cov(m, y=None, rowvar=True, bias=False, ddof=None, fweights=None, aweights=None)
**where :**
**m :** [array_like] A 1D or 2D variables. variables are columns
**y :** [array_like] It has the same form as that of m.
**rowvar** : [bool, optional] If rowvar is True (default), then each row represents a variable, with observations in the columns. Otherwise, the relationship is transposed:
**bias** : Default normalization is False. If bias is True it normalize the data points.
**ddof** : If not None the default value implied by bias is overridden. Note that ddof=1 will return the unbiased estimate, even if both fweights and aweights are specified.
**fweights** : fweight is 1-D array of integer frequency weights
**aweights** : aweight is 1-D array of observation vector weights.
**Returns** : It returns ndarray covariance matrix

## 27. What is EDA? List the functions performed by EDA. How to measure control tendency?

- Exploratory Data Analysis (EDA) is a general approach to exploring datasets by means of simple summary statistics and graphic visualizations in order to gain a deeper understanding of data.
- EDA is an approach/philosophy for data analysis that employs a variety of techniques to

1. Maximize insight into a data set;
2. Uncover underlying structure;
3. Extract important variables;
4. Detect outliers and anomalies;
5. Test underlying assumptions;
6. Develop parsimonious models; and
7. Determine optimal factor settings.

- With EDA, following functions are performed :

1. Describe of user data
2. Closely explore data distributions

3. Understand the relations between variables
4. Notice unusual or unexpected situations
5. Place the data into groups
6. Notice unexpected patterns within groups
7. Take note of group differences

**Measuring Central Tendency:-**

- The three main measures of central tendency are the mean, median and mode. In any given set of data, these may be three very different values or values that are the same or close to the same.
- A measure of central tendency is a single value that attempts to describe a set of data by identifying the central position within that set of data. As such, measures of central tendency are sometimes called measures of central location.
- The mean, median and mode are all valid measures of central tendency, but under different conditions, some measures of central tendency become more appropriate to use than others.
- Example:

```
import pandas as pd
 dataMatrix = {"D1":[135, 137, 136, 138, 138],
               "D2":[43,  42, 42, 42, 42],
               "D3":[72, 73, 72, 72, 73],
              "D4":[100, 102, 100, 103, 104]
            };
    dataFrame = pd.DataFrame(data=dataMatrix);

print("DataFrame:");
print(dataFrame);

print("Mean:Computed column-wise:");
meanData = dataFrame.mean();
print(meanData);

print("Mean:Computed row-wise:");
meanData = dataFrame.mean(axis=1);
print(meanData);

print("Median:Computed column-wise:");
medianData = dataFrame.median();
print(medianData);

print("Median:Computed row-wise:");
medianData = dataFrame.median(axis=1);
print(medianData);
```

```
print("Mode:Computed column-wise:");
modeData = dataFrame.mode();
print(modeData);

print("Mode:Computed row-wise:");
modeData = dataFrame.mode(axis=1);
print(modeData);
```

**Output:**

```
DataFrame:
   D1   D2 D3  D4
0 135 43 72 100
1 137 42 73 102
2 136 42 72 100
3 138 42 72 103
4 138 42 73 104
Mean:Computed column-wise:
D1   136.8
D2    42.2
D3    72.4
D4   101.8
dtype: float64
Mean:Computed row-wise:
0   87.50
1   88.50
2   87.50
3   88.75
4   89.25
dtype: float64
Median:Computed column-wise:
D1   137.0
D2    42.0
D3    72.0
D4   102.0
dtype: float64
Median:Computed row-wise:
0   86.0
1   87.5
2   86.0
3   87.5
4   88.5
dtype: float64
Mode:Computed column-wise:
   D1  D2  D3   D4
0 138  42  72  100
```

```
Mode:Computed row-wise:
   0  1   2   3
0 43 72 100 135
1 42 73 102 137
2 42 72 100 136
3 42 72 103 138
4 42 73 104 138
```

- The median provides the central position in the series of values. When creating a variable, it is a measure less influenced by anomalous cases or by an asymmetric distribution of values around the mean.

## 28. What is memory profiler?

- The memory_profiler module can be used for monitoring memory consumption in a process, or you can use it for a line-by-line analysis of the memory consumption of your code.
- This package is not provided as a default Python package and it requires installation. Use the following commands to install the package and its dependencies from the command line:

```
ipython –m pip install psutil
ipython –m pip install memory_profiler
```

- Once it's installed, we need some code to run it against. The memory_profiler actually works in much the same way as line_profiler in that when you run it, memory_profiler will inject an instance of itself into __builtins__ named profile that you are supposed to use as a decorator on the function you are profiling.

## 29. Explain classes used in Scikit-learn.

- Scikit-learn takes a highly object-oriented approach to machine learning models. Every major Scikit-learn class inherits from ***sklearn.base.BaseEstimator.***
- Scikit-learn features some base classes on which all the algorithms are built. Apart from *BaseEstimator*, the class from which all other classes inherit, there are four class types covering all the basic machine learning functionalities:

1. Classifying
2. Regressing
3. Grouping by clusters
4. Transforming data

- Scikit-learn takes a highly object-oriented approach to machine learning models. Every major Scikit-learn class inherits from ***sklearn.base.BaseEstimator***.
- All objects within scikit-learn share a uniform common basic API consisting of three complementary interfaces: an **estimator** interface for building and fitting models, a **predictor** interface for making predictions and a **transformer** interface for converting data.

**1. Estimators**

- The estimator interface is at the core of the library. It defines instantiation mechanisms of objects and exposes a fit method for learning a model from training data.
- All supervised and unsupervised learning algorithms (e.g., for classification, regression or clustering) are offered as objects implementing this interface. Machine learning tasks like feature extraction, feature selection or dimensionality reduction are also provided as estimators.

**2. Predictors**

- The predictor interface extends the notion of an estimator by adding a predict method that takes an array X test and produces predictions for X test, based on the learned parameters of the estimator.
- In the case of supervised learning estimators, this method typically returns the predicted labels or values computed by the model.

**3. Transformers**

- Since it is common to modify or filter data before feeding it to a learning algorithm, some estimators in the library implement a transformer interface which defines a transform method.
- It takes as input some new data X test and yields as output a transformed version of X test.
- Preprocessing, feature selection, feature extraction and dimensionality reduction algorithms are all provided as transformers within the library.

## 30. Define data wrangling. Explain steps performed in data wrangling process.

- Data Wrangling is the process of transforming data from its original "raw" form into a more digestible format and organizing sets from various sources into a singular coherent whole for further processing.
- Data wrangling is also called as data munging.
- The primary purpose of data wrangling can be described as getting data in coherent shape. In other words, it is making raw data usable. It provides substance for further proceedings.
- Data wrangling covers the following processes :
  1. Getting data from the various source into one place
  2. Piecing the data together according to the determined setting
  3. Cleaning the data from the noise or erroneous, missing elements
- Data wrangling is the process of cleaning, structuring and enriching raw data into a desired format for better decision making in less time
- There are typically six iterative steps that make up the data wrangling process:
  1. Discovering : Before you can dive deeply, you must better understand what is in your data, which will inform how you want to analyze it. How you wrangle customer data, for example, may be informed by where they are located, what they bought, or what promotions they received.
  2. Structuring : This means organizing the data, which is necessary because raw data comes in many different shapes and sizes. A single column may turn into several rows for easier analysis. One column may become two. Movement of data is made for easier computation and analysis.
  3. Cleaning : What happens when errors and outliers skew your data ? You clean the data. What happens when state data is entered as AP or Andhra Pradesh or Arunachal Pradesh? You clean the data. Null values are changed and standard formatting implemented, ultimately increasing data quality.
  4. Enriching : Here you take stock in your data and strategize about how other additional data might augment it. Questions asked during this data wrangling step might be : what new types of data can I derive from what I already have or what other information would better inform my decision making about this current data?
  5. Validating : Validation rules are repetitive programming sequences that verify data consistency, quality, and security. Examples of validation include

ensuring uniform distribution of attributes that should be distributed normally (e.g. birth dates) or confirming accuracy of fields through a check across data.

6. Publishing : Analysts prepare the wrangled data for use downstream, whether by a particular user or software and document any particular steps taken or logic used to wrangle said data. Data wrangling gurus understand that implementation of insights relies upon the ease with which it can be accessed and utilized by others.