2.1 DESCRIPTION OF DATA :-

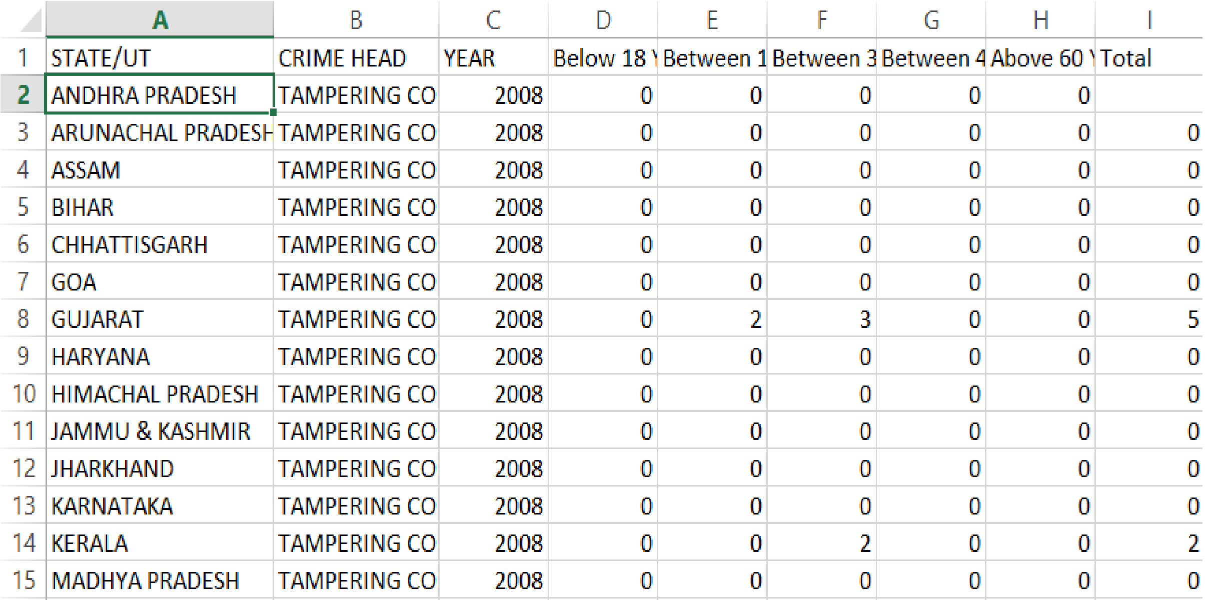
DEFINATION OF DATA SET

A **data set** (or **dataset**) is a collection of data. Most commonly a data set corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable, and each row corresponds to a given member of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Each value is known as a datum. The data set may comprise data for one or more members, corresponding to the number of rows.

The term data set may also be used more loosely, to refer to the data in a collection of closely related tables, corresponding to a particular experiment or event. Less used names for this kind of data sets are **data corpus** and **data stock**.

Data sets that are so large that traditional **data** processing applications are inadequate to deal with them are known as big **data**. In the open **data** discipline, **data set** is the unit to measure the information released in a public open **data** repository.

- **Display the whole dataset records**

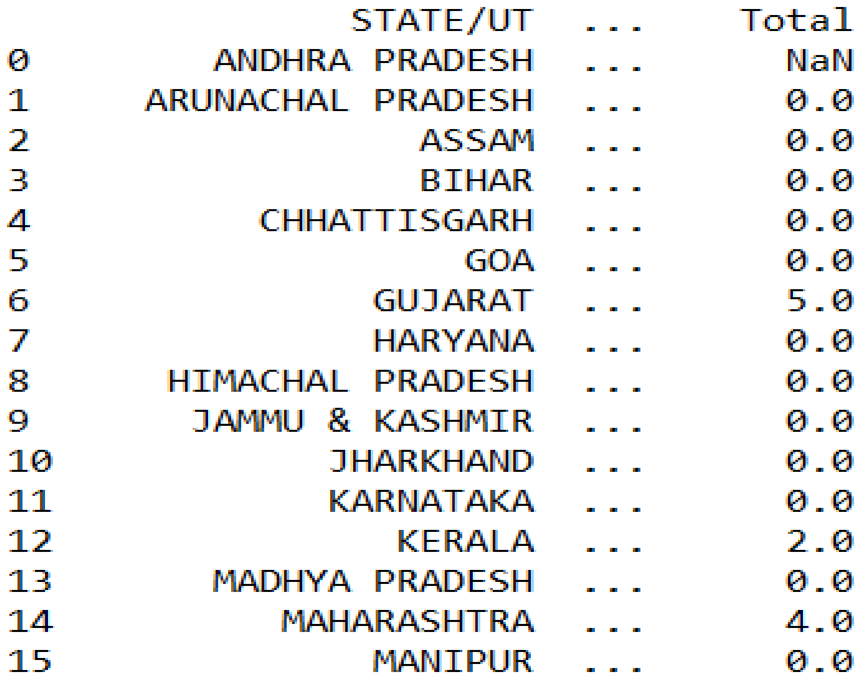


	A	B	C	D	E	F	G	H	I
1	STATE/UT	CRIME HEAD	YEAR	Below 18 Y	Between 1	Between 3	Between 4	Above 60 Y	Total
2	ANDHRA PRADESH	TAMPERING CO	2008	0	0	0	0	0	
3	ARUNACHAL PRADESH	TAMPERING CO	2008	0	0	0	0	0	0
4	ASSAM	TAMPERING CO	2008	0	0	0	0	0	0
5	BIHAR	TAMPERING CO	2008	0	0	0	0	0	0
6	CHHATTISGARH	TAMPERING CO	2008	0	0	0	0	0	0
7	GOA	TAMPERING CO	2008	0	0	0	0	0	0
8	GUJARAT	TAMPERING CO	2008	0	2	3	0	0	5
9	HARYANA	TAMPERING CO	2008	0	0	0	0	0	0
10	HIMACHAL PRADESH	TAMPERING CO	2008	0	0	0	0	0	0
11	JAMMU & KASHMIR	TAMPERING CO	2008	0	0	0	0	0	0
12	JHARKHAND	TAMPERING CO	2008	0	0	0	0	0	0
13	KARNATAKA	TAMPERING CO	2008	0	0	0	0	0	0
14	KERALA	TAMPERING CO	2008	0	0	2	0	0	2
15	MADHYA PRADESH	TAMPERING CO	2008	0	0	0	0	0	0

Command:

```
path='C://Users/MALVIKA/Desktop/18mcl2/Project_datascience/cyber_Crime.csv'
cc = pd.read_csv(path, encoding = 'latin1')
print(cc)
```

Output:



```
                 STATE/UT  ...    Total
0          ANDHRA PRADESH  ...      NaN
1       ARUNACHAL PRADESH  ...      0.0
2                   ASSAM  ...      0.0
3                   BIHAR  ...      0.0
4            CHHATTISGARH  ...      0.0
5                     GOA  ...      0.0
6                 GUJARAT  ...      5.0
7                 HARYANA  ...      0.0
8        HIMACHAL PRADESH  ...      0.0
9         JAMMU & KASHMIR  ...      0.0
10              JHARKHAND  ...      0.0
11              KARNATAKA  ...      0.0
12                 KERALA  ...      2.0
13         MADHYA PRADESH  ...      0.0
14            MAHARASHTRA  ...      4.0
15                MANIPUR  ...      0.0
```

2.2 CATEGORY OF DATA :-

Nominal Data

- What is nominal data?

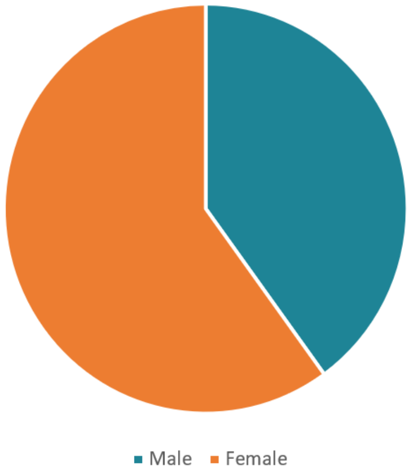
In statistics, nominal data (also known as nominal scale) is a type of data th sed to label variables without providing any quantitative value. It is the simplest form of a scale of measure. Unlike ordinal data, nominal data cannot be ordered and cannot be measured.

Dissimilar to interval or ratio data, nominal data cannot be manipulated using available mathematical operators. Thus, the only measure of central tendency for such type of data is the mode.

Nominal data can be both qualitative and quantitative. However, the quantitative labels lack a numerical value or relationship (e.g., identification number). On the other hand, various types of qualitative data can be represented in nominal form. They may include words, letters, and symbols. Also, names of people, gender, and nationality are just a few of the most common examples of nominal data.

Nominal data can be analysed using the grouping method. The variables can be grouped together into categories, and for each category, the frequency or percentage can be calculated. The data can also be presented visually such as by using a pie chart.

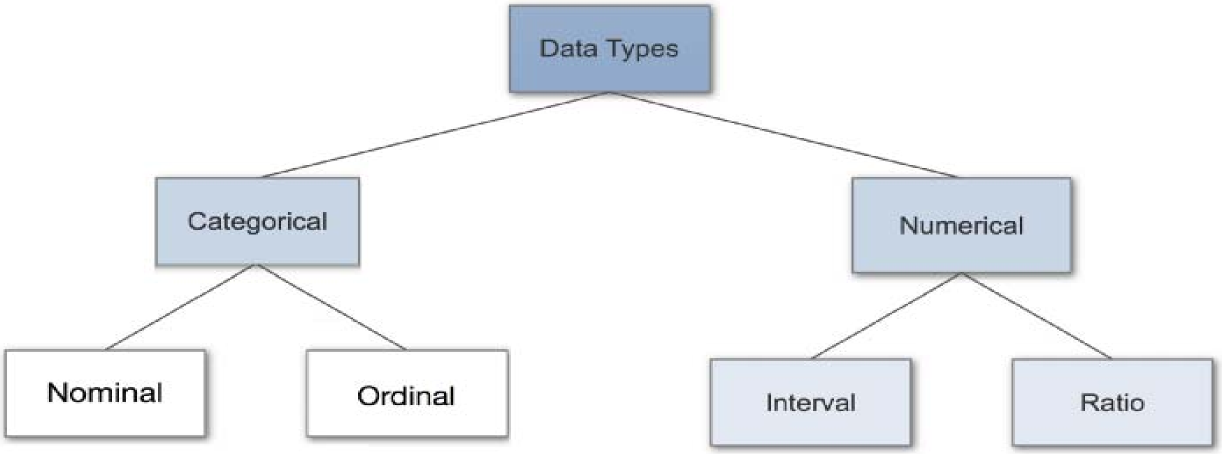
- **EXAMPLE**



Dissimilar to interval or ratio data, nominal data cannot be manipulated using available mathematical operators. Thus, the only measure of central tendency for such type of data is the mode.

❖ Types of Data in Data Science

Data Types are an important concept of statistics, which needs to be understood, to correctly apply statistical measurements to your data and therefore to correctly conclude certain assumptions about it. This blog post will introduce you to the different data types you need to know, to do proper exploratory data analysis (EDA), which is one of the most underestimated parts of a machine learning project.

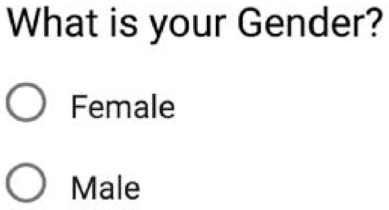


➔ Categorical Data

Categorical data represents characteristics. Therefore it can represent things like a person's gender, language etc. Categorical data can also take on numerical values (Example: 1 for female and 0 for male). Note that those numbers don't have mathematical meaning.

- **Nominal Data**

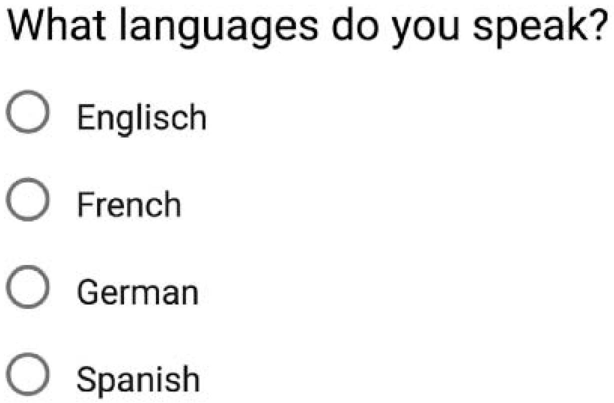
Nominal values represent discrete units and are used to label variables, that have no quantitative value. Just think of them as „labels". Note that nominal data that has no order. Therefore if you would change the order of its values, the meaning would not change. You can see two examples of nominal features below:



What is your Gender?

○ Female

○ Male



What languages do you speak?

○ Englisch

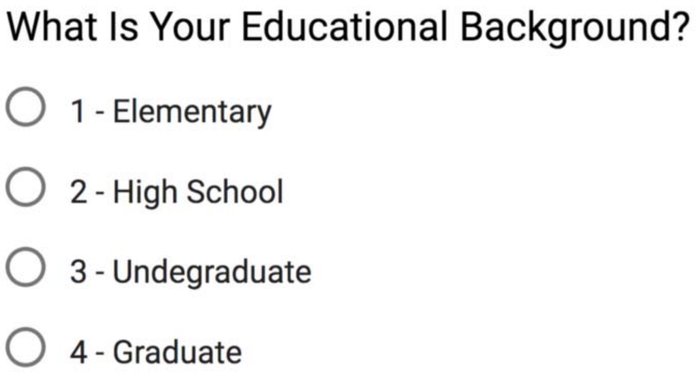
○ French

○ German

○ Spanish

- **Ordinal Data**

Ordinal values represent discrete and ordered units. It is therefore nearly the same as nominal data, except that it's ordering matters. You can see an example below:



What Is Your Educational Background?

○ 1 - Elementary

○ 2 - High School

○ 3 - Undegraduate

○ 4 - Graduate

➔ **Numerical Data:**

1. Discrete Data

We speak of discrete data if its values are distinct and separate. In other words: We speak of discrete data if the data can only take on certain values. This type of data **can't be measured but it can be counted**. It basically represents information that can be categorized into a classification. An example is the number of heads in 100 coin flips.

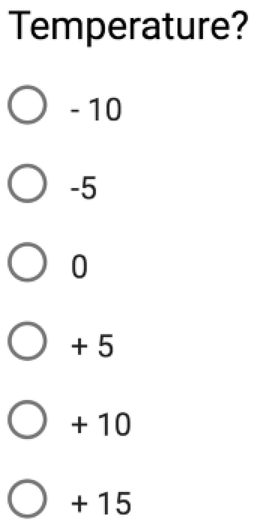
You can check by asking the following two questions whether you are dealing with discrete data or not: Can you count it and can it be divided up into smaller and smaller parts?

2. Continuous Data

Continuous Data represents measurements and therefore their values **can't be counted but they can be measured**. An example would be the height of a person, which you can describe by using intervals on the real number line.

- **Interval Data:**

Interval values represent **ordered units that have the same difference**. Therefore we speak of interval data when we have a variable that contains numeric values that are ordered and where we know the exact differences between the values. An example would be a feature that contains temperature of a given place like you can see below:



Temperature?

○ - 10

○ -5

○ 0

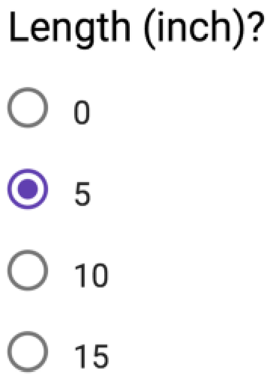
○ + 5

○ + 10

○ + 15

- **Ratio Data:**

Ratio values are also ordered units that have the same difference. Ratio values are **the same as interval values, with the difference that they do have an absolute zero**. Good examples are height, weight, length etc.



Length (inch)?

○ 0

● 5

○ 10

○ 15

In our Data Set States Column is Nominal Data. Crime Head is Ordinal Data. And Crime Rate is Desecrate Series.

➔NUMERICAL DATA

Numerical data is information that is something that is measurable. It is always collected in number form, although there are other types of data that can appear in number form. An example of numerical data would be the number of people that attended the movie theater over the course of a month.

One of the ways you can identify numerical data is by seeing if the data can be added together. In fact, you should be able to perform just about any mathematical operation on numerical data.

You can also put data in ascending (least to greatest) and descending (greatest to least) order. Data can only be numerical if the answers can be represented in fraction and/or decimal form. If you have to group the information into categories, then it is considered categorical.

➔BINARY DATA

Binary data is a type of data that is represented or displayed in the binary numeral system.

Binary data is the only category of data that can be directly understood and executed by a computer. It is numerically represented by a combination of zeros and ones.

Binary data is considered the native data/language of a computer and it interacts with the lowest abstraction layer of its hardware. This type of data is produced whenever a process is performed on a computer.

The application requesting the process sends instructions in a high-level language that is ultimately converted into binary data to be executed or sent to the processor.

➔ORDINAL DATA

Ordinal data is a categorical, statistical data type where the variables have natural, ordered categories and the distances between the categories is not known.

These data exist on an **ordinal scale**, one of four levels of measurement described by S. S. Stevens in 1946. The ordinal scale is distinguished from the nominal scale by having a ranking.

It also differs from interval and ratio scales by not having category widths that represent equal increments of the underlying attribute.

2.3 IMPORT LIBRARY AND READING THE DATASETS VIEWING METHODS :-

IMPORT LIBRARY

1. Pandas

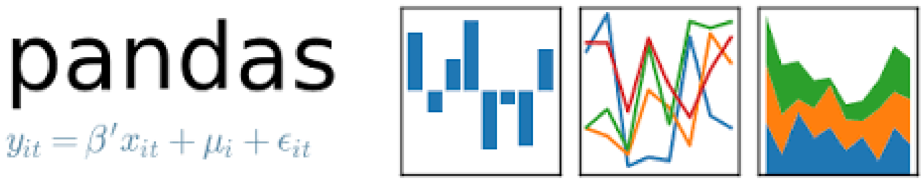
A pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

✓ **Key features of pandas**

- Fast and efficient Data Frame object with default and customized indexing.
- Tools for loading data into in-memory data objects from different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of date sets.
- Label-based slicing, indexing and sub setting of large data sets.
- Columns from a data structure can be deleted or inserted.
- Group by data for aggregation and transformations.
- High performance merging and joining of data.
- Time Series functionality.

Syntax : import pandas as pd



➔ We are using pandas in cyber crime data set for create data frame
➔ We also use for lode csv file in python code
➔ Also use for data processing.

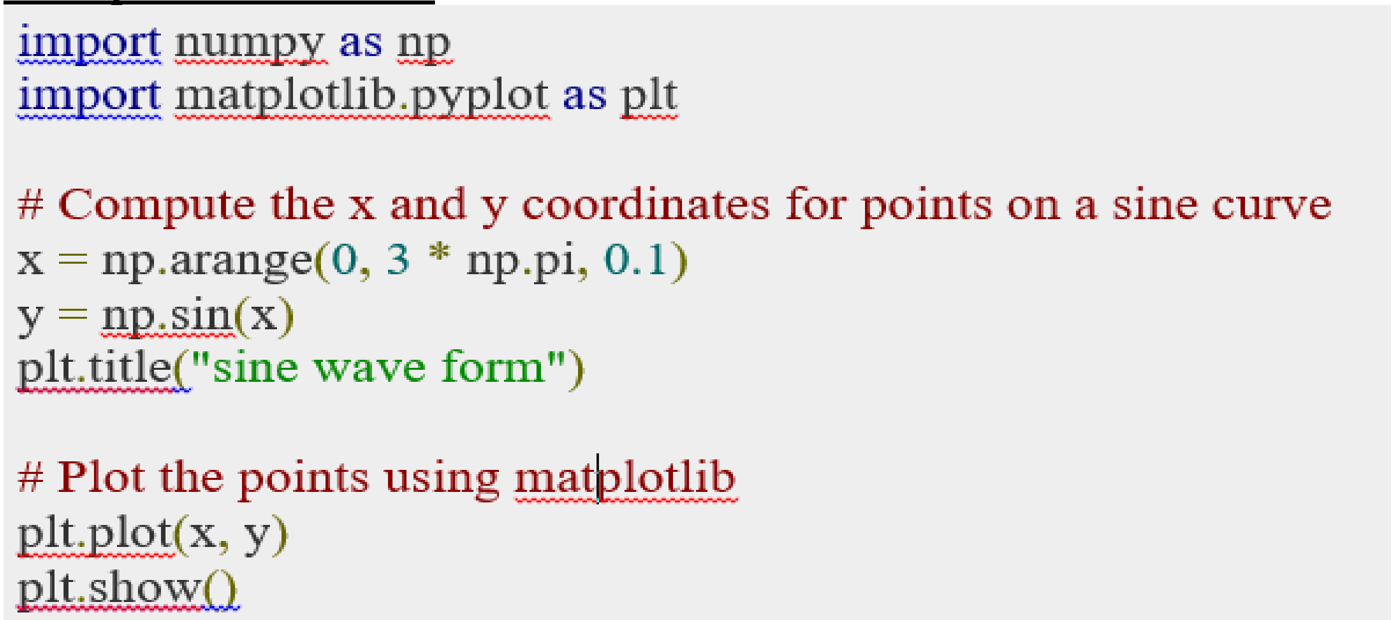
2. Matplotlib

Matplotlib is a python library used to create 2D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc. It supports a very wide variety of graphs and plots namely - histogram, bar charts, power spectra, error charts etc. It is used along with NumPy to provide an environment that is an effective open source alternative for MatLab. It can also be used with graphics toolkits like PyQt and wxPython.

Syntax : import matplotlib.pyplot as plt

➔ We are using matplotlib to plotting graph of x,y

Example of MatPlotLib



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

# Compute the x and y coordinates for points on a sine curve
x = np.arange(0, 3 * np.pi, 0.1)
y = np.sin(x)
plt.title("sine wave form")

# Plot the points using matplotlib
plt.plot(x, y)
plt.show()
```

3. Numpy

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed.

NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

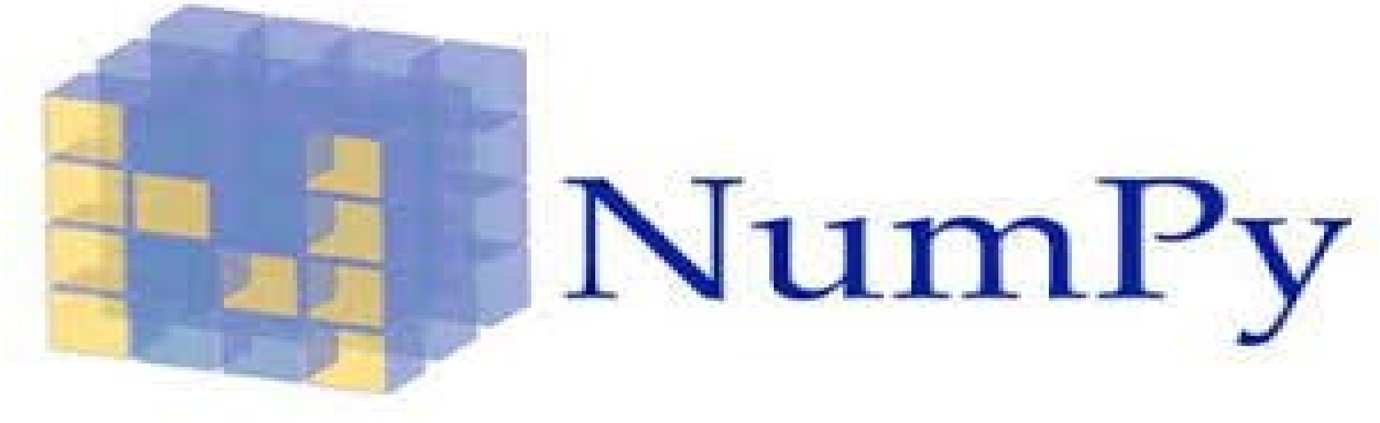
Numeric, the ancestor of NumPy, was developed by Jim Hugunin. Another package Num array was also developed, having some additional functionalities. In 2005, Travis Oliphant created NumPy package by incorporating the features of Numarray into Numeric package. There are many contributors to this open source project.

➔ **Operations using numpy**

- Mathematical and logical operations on arrays.
- Fourier transforms and routines for shape manipulation.

Syntax : import numpy as np

➔ Here, we are using numpy for linear algebra



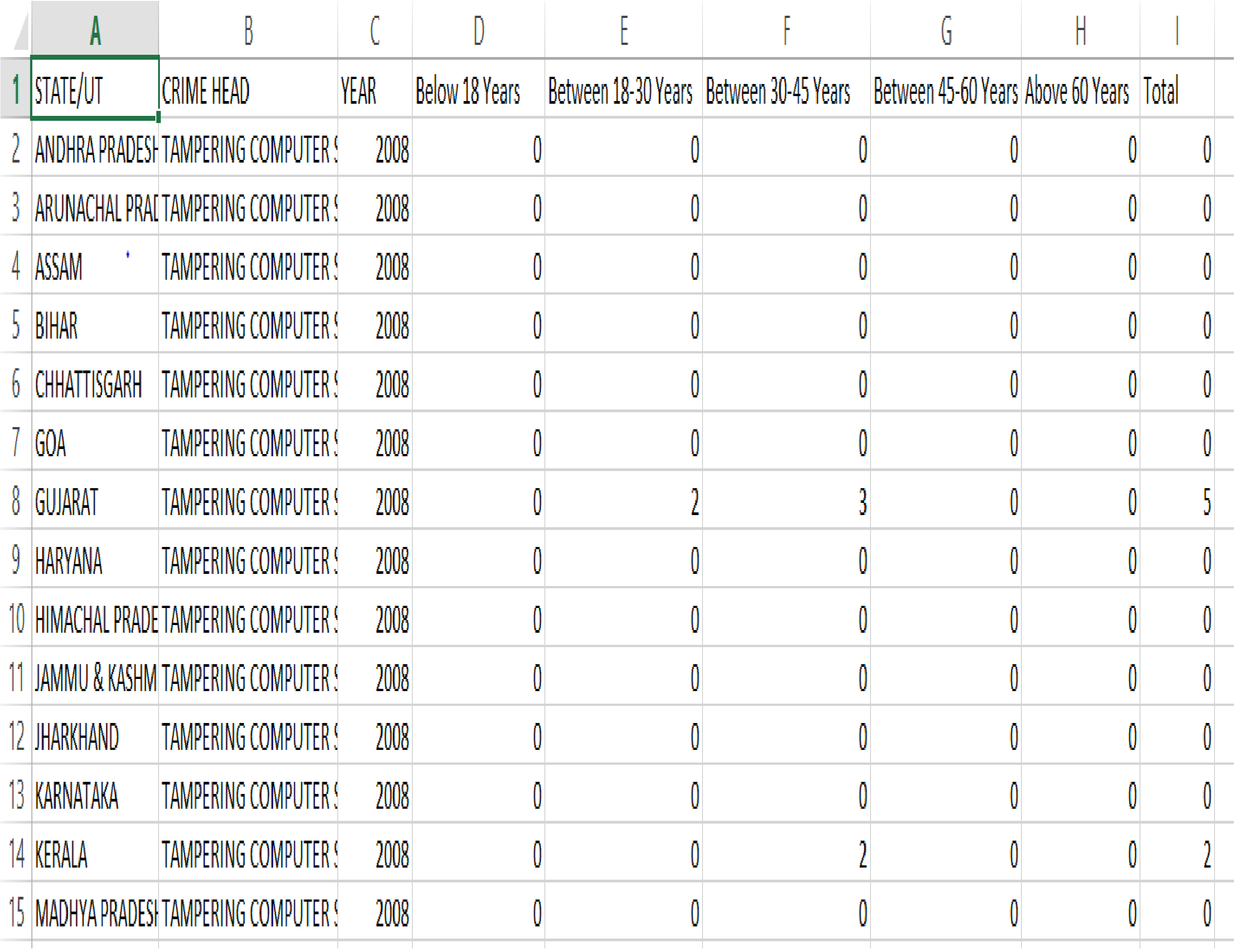
4. Sklearn

- Sklearn-learn (formerly scikits.learn) is a free software machine learning library for the Python programming language.
- It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, *k*-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.
- Scikit-learn was initially developed by David Cournapeau as a Google summer of code project in 2007. Later Matthieu Brucher joined the project and started to use it as a part of his thesis work.
- Scikit-learn is largely written in Python, with some core algorithms written in Cython to achieve performance. Support vector machines are implemented by a Cython wrapper around LIBSVM; logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR.

Syntax : import sklearn as sk

- ➔ Here, we are using sklearn to splitting training and testing data
- ➔ Also we calculate mean error and variance calculation

READING THE DATASETS



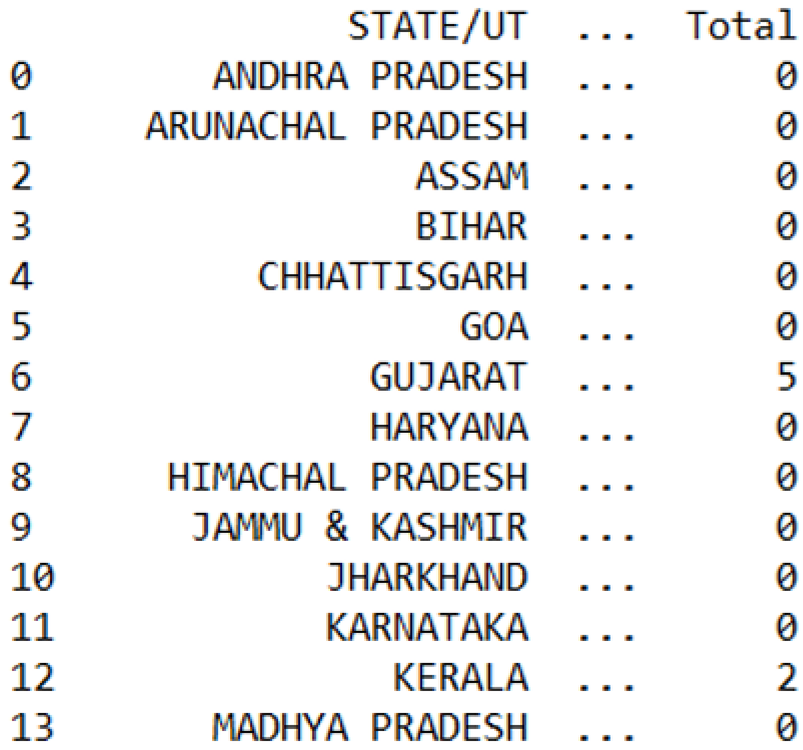
	A	B	C	D	E	F	G	H	I
1	STATE/UT	CRIME HEAD	YEAR	Below 18 Years	Between 18-30 Years	Between 30-45 Years	Between 45-60 Years	Above 60 Years	Total
2	ANDHRA PRADESH	TAMPERING COMPUTER S	2008	0	0	0	0	0	0
3	ARUNACHAL PRAD	TAMPERING COMPUTER S	2008	0	0	0	0	0	0
4	ASSAM	TAMPERING COMPUTER S	2008	0	0	0	0	0	0
5	BIHAR	TAMPERING COMPUTER S	2008	0	0	0	0	0	0
6	CHHATTISGARH	TAMPERING COMPUTER S	2008	0	0	0	0	0	0
7	GOA	TAMPERING COMPUTER S	2008	0	0	0	0	0	0
8	GUJARAT	TAMPERING COMPUTER S	2008	0	2	3	0	0	5
9	HARYANA	TAMPERING COMPUTER S	2008	0	0	0	0	0	0
10	HIMACHAL PRADE	TAMPERING COMPUTER S	2008	0	0	0	0	0	0
11	JAMMU & KASHM	TAMPERING COMPUTER S	2008	0	0	0	0	0	0
12	JHARKHAND	TAMPERING COMPUTER S	2008	0	0	0	0	0	0
13	KARNATAKA	TAMPERING COMPUTER S	2008	0	0	0	0	0	0
14	KERALA	TAMPERING COMPUTER S	2008	0	0	2	0	0	2
15	MADHYA PRADESH	TAMPERING COMPUTER S	2008	0	0	0	0	0	0

Lode the data set and print

- **Command :**

```
path='C://Users/MALVIKA/Desktop/18mcl2/Project_datascience/cyber_Crime.csv'

cc = pd.read_csv(path, encoding = 'latin1')

print(cc)
```

- **Output :**



```
                  STATE/UT  ...  Total
0           ANDHRA PRADESH  ...      0
1        ARUNACHAL PRADESH  ...      0
2                    ASSAM  ...      0
3                    BIHAR  ...      0
4             CHHATTISGARH  ...      0
5                      GOA  ...      0
6                  GUJARAT  ...      5
7                  HARYANA  ...      0
8         HIMACHAL PRADESH  ...      0
9          JAMMU & KASHMIR  ...      0
10               JHARKHAND  ...      0
11               KARNATAKA  ...      0
12                  KERALA  ...      2
13          MADHYA PRADESH  ...      0
```

VIEWING METHODS

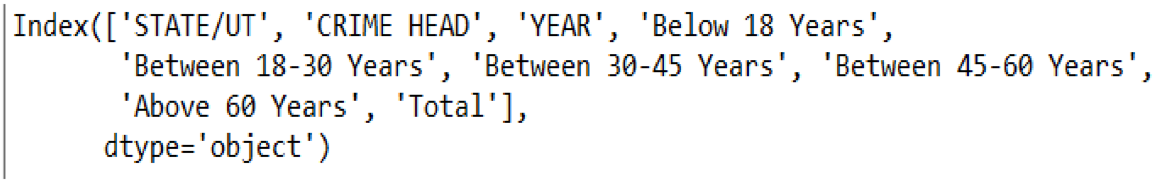
Attributes of a class are function objects that define corresponding methods of its instances. They are used to implement access controls of the classes.

1. Display the dataset columns

- **Command :**

  print(cc.columns)

- **Output :**



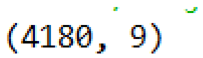
```
Index(['STATE/UT', 'CRIME HEAD', 'YEAR', 'Below 18 Years',
       'Between 18-30 Years', 'Between 30-45 Years', 'Between 45-60 Years',
       'Above 60 Years', 'Total'],
      dtype='object')
```

2. Display total rows and columns

- **Command :**

  print(cc.shape)

- **Output :**



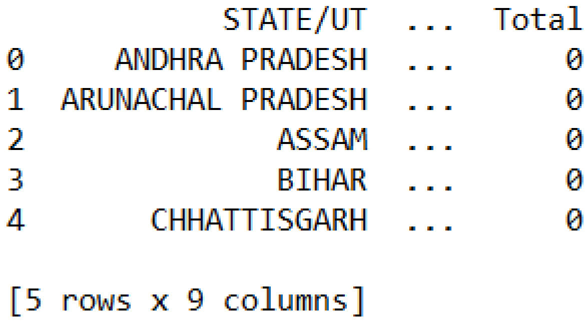
```
(4180, 9)
```

3. Display the dataset first five rows

- **Command :**

  print(cc.head())

- **Output :**



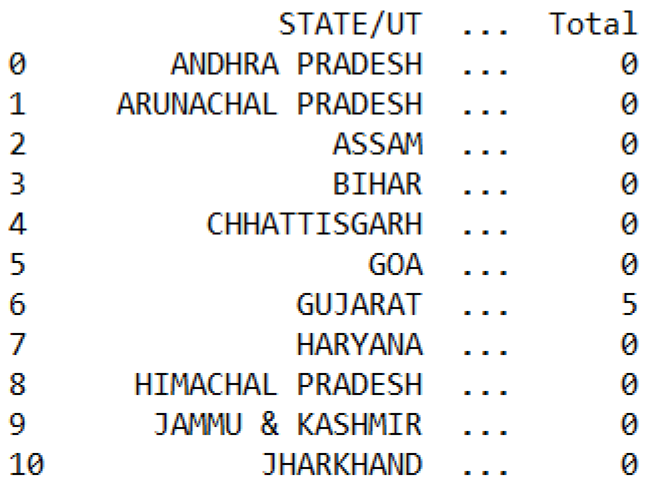
```
            STATE/UT  ...  Total
0     ANDHRA PRADESH  ...      0
1  ARUNACHAL PRADESH  ...      0
2              ASSAM  ...      0
3              BIHAR  ...      0
4       CHHATTISGARH  ...      0

[5 rows x 9 columns]
```

4. **Display the whole dataset records**

- **Command :**

  print(cc)

- **Output :**



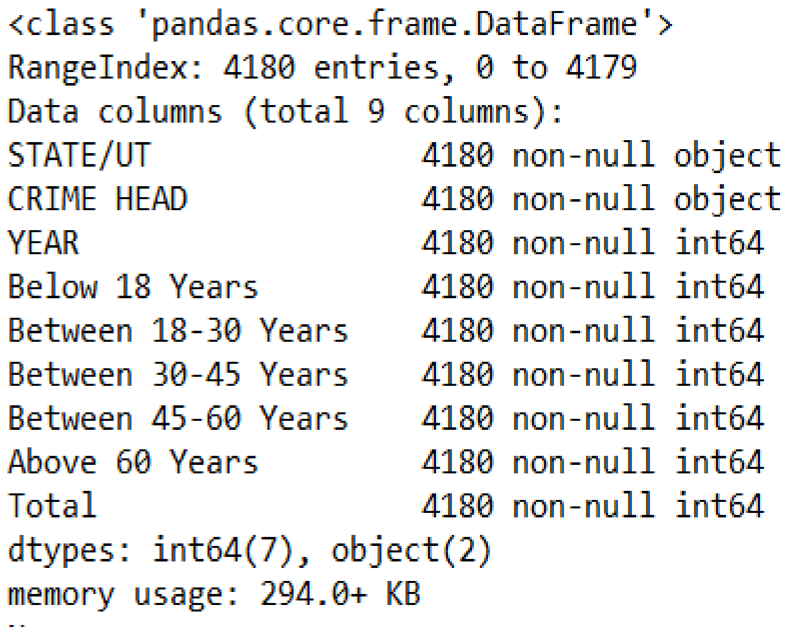
```
                STATE/UT  ...  Total
0         ANDHRA PRADESH  ...      0
1      ARUNACHAL PRADESH  ...      0
2                  ASSAM  ...      0
3                  BIHAR  ...      0
4           CHHATTISGARH  ...      0
5                    GOA  ...      0
6                GUJARAT  ...      5
7                HARYANA  ...      0
8       HIMACHAL PRADESH  ...      0
9        JAMMU & KASHMIR  ...      0
10             JHARKHAND  ...      0
```

5. **Display the dataset information**

- **Command :**

  print(cc.info())

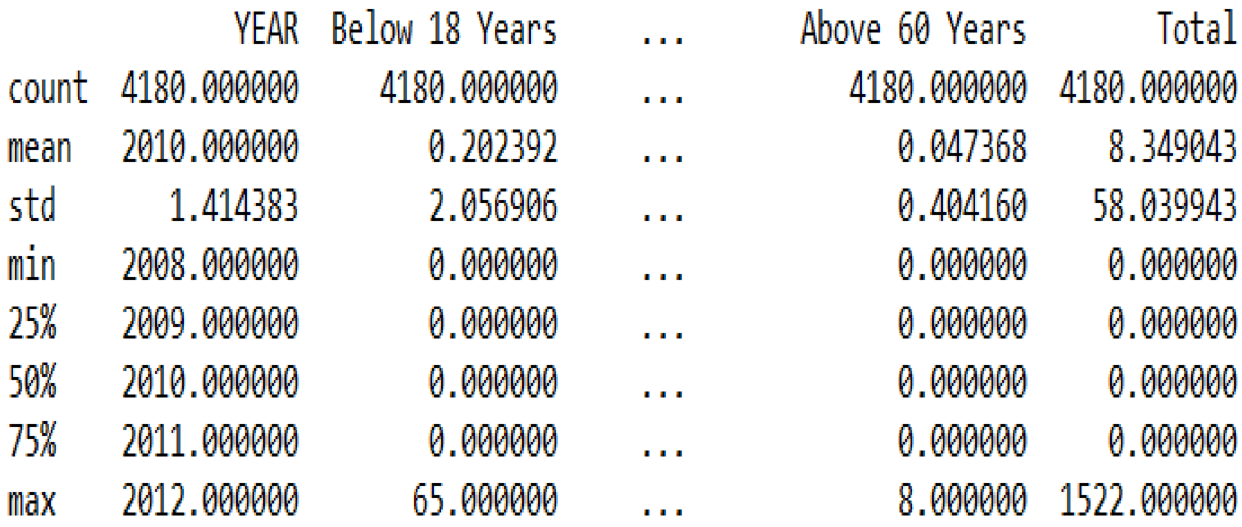
- **Output :**



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4180 entries, 0 to 4179
Data columns (total 9 columns):
STATE/UT               4180 non-null object
CRIME HEAD             4180 non-null object
YEAR                   4180 non-null int64
Below 18 Years         4180 non-null int64
Between 18-30 Years    4180 non-null int64
Between 30-45 Years    4180 non-null int64
Between 45-60 Years    4180 non-null int64
Above 60 Years         4180 non-null int64
Total                  4180 non-null int64
dtypes: int64(7), object(2)
memory usage: 294.0+ KB
```

6. **Display the dataset mean, medium and other values**
   - **Command :**

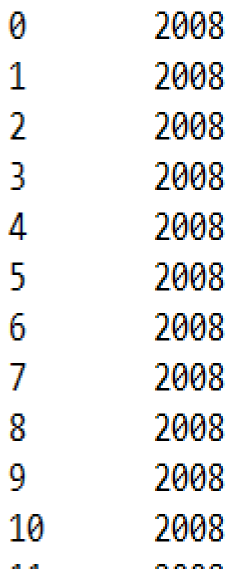
     print(cc.describe())
   - **Output :**



```
              YEAR  Below 18 Years    ...     Above 60 Years        Total
count  4180.000000     4180.000000    ...        4180.000000  4180.000000
mean   2010.000000        0.202392    ...           0.047368     8.349043
std       1.414383        2.056906    ...           0.404160    58.039943
min    2008.000000        0.000000    ...           0.000000     0.000000
25%    2009.000000        0.000000    ...           0.000000     0.000000
50%    2010.000000        0.000000    ...           0.000000     0.000000
75%    2011.000000        0.000000    ...           0.000000     0.000000
max    2012.000000       65.000000    ...           8.000000  1522.000000
```

7. **Display the dataset single column value**
   - **Command :**

     print(cc.YEAR)
   - **Output :**



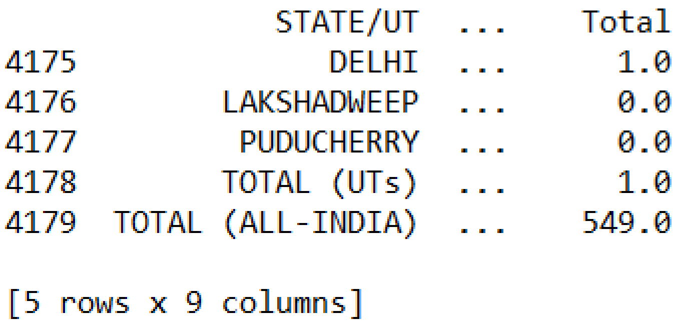
```
0        2008
1        2008
2        2008
3        2008
4        2008
5        2008
6        2008
7        2008
8        2008
9        2008
10       2008
```

8. Display the last five records

- **Command :**

  print(cc.tail())

- **Output :**



```
                 STATE/UT  ...   Total
4175                DELHI  ...     1.0
4176          LAKSHADWEEP  ...     0.0
4177           PUDUCHERRY  ...     0.0
4178          TOTAL (UTs)  ...     1.0
4179   TOTAL (ALL-INDIA)   ...   549.0

[5 rows x 9 columns]
```