Assignment NO:- 04.

Title:-

Write python code that loads any dataset from (www.data.gov.in) & does some basic data cleaning. Add component on data set.

Problem statement:-

To python code for that loads any dataset & plot the graph.

Pre-lab:-

A basic understanding of computer programming terminology. A basic understanding of any of programming languages will help to understanding the python programming & data science concepts.

Theory:-

1) Step 1 Acquiring Data:-

step one, acquiring data. The first step in the data science process is to acquiring the data. The first step in acquiring data is to determine what data is available.

2) step 2-A: Exploring Data:-

Exploring Data after you've put together the data that you need for your application you might to tempted to immediatly build models to analyze the data.

3) Step 2-B: Pre-Processing Data:—

The raw data that you get directly from your sources are never in the format that you need to perform analysis on. A very important part of data preparation is to address quality of issues in your data. Real-world data is messy!.

4) Step 3- Analysing Data:—

Now that you have your data nicely prepared, the next step is to analyze the data. There are different types of problems, & so there are different types of analysis techniques.

5) Step 4: Communicating Results:—

step four, reporting insights. The fourth step in our data science process is reporting the insights gained from our analysis The techniques that we discuss & explore in data can be used here as well.

Dropping colamns in the Dataframe:—

often, you'll find that not all the categories of data in a dataset are useful to you. Let's let us now see how we can handle missing values (say NA or NaN) using Pandas.

```
# import the pandas library.
import pandas as pd.
import numpy as np.
```

df = pd.DataFrame(np.random.random(5,3), index=['a', 'b', 'c', 'f', 'h'], columns=(['one', 'two', 'three'])

df = df.index(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])
print df.

Its output is as follows -

	one	two	three
a	0.077988	0.476149	0.965836
b	NaN	NaN	NaN
c	-0.390208	-0.551605	-2.301950
d	NaN	NaN	NaN
e	-2.000303	-0.788201	1.510272
f	-0.930230	-0.670473	1.146615
g	NaN	NaN	NaN
h	0.085100	0.532791	0.887415

Example;

```
import pandas as pd
import numpy as np

df = pd.DataFrame(np.random.randn(5,3), index=(['a', 'c', 'e', 'f', 'h'], columns=['one', 'two', 'three'])

df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])
print df['one'].isnull()
```

Its output is as follows -

a false
b True
c false
d True

e False
f False
g True
h False.
Name: one, dtype: bool.
cleaning / filling Missing Data.
Replace NaN with a scalar value
The following program shows how you replace "NaN" with "0
import pandas as pd
import numpy as np
df = df.reindex(['a', 'b', 'c'])
print df
print("NaN replaced with '0':")
print df.fillna(0)
Its output is as follows-

	one	two	three
a	0.576991	-0.741695	0.553172
b	NaN	NaN	NaN
c	0.744328	-1.735166	1.749580.

NaN replaced with '0'.

	one	two	three
a	-0.576991	-0.741695	0.553172.
b	0.000000	0.000000	0.000000
c	0.744328	-1.735166	1.749580

Fill NA forward & Backward.
Using the concepts of filling discussed in the ReIndexing chapter we will fill the missing values

Method
Action
Pad / fill
Fill methods Backward
Example
Import pandas as pd
Import numpy as np

df = pd DataFrame (np.random.randn(5, 3), index = ['a', 'c', 'e', 'f', 'h'], columns = ['one', 'two', 'three'])
df = df.reindex (['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])
print df.fillna (method = 'pad')
Its o/p is as follows -

	one	two	three
a	0.077988	0.476149	0.965836
b	0.077988	0.476149	0.965836
c	-0.390208	-0.551605	-2.301910
d	-0.390208	-0.551605	-2.301910
e	-2.000303	-0.788201	1.510072
f	-0.930230	-0.670473	1.146815
g	-0.930230	-0.670473	1.146815
h	0.085100	0.532791	0.887415

Drop missing values
Example
import pandas as pd
import numpy as np
df = df.reindex (['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])
print df.dropna()
its o/p is as follows -

	one	two	three
g	0.017988	0.476149	0.965836
h	-0.390280	-0.551605	-2.301950
i	-2.022303	-0.788201	1.510072
j	-0.930280	-0.670473	1.146615
k	0.085100	0.532791	0.887415

Replace Missing (or) Generic Values
Replace No with a scalar value is equivalent behaviour of the fillna() fun.

Ex;

```
import pandas as pd
import numpy as np
df = pd.DataFrame({'one':[10,20,30,40,50,2000],
'two':[1000,0,30,40,50,60]})
print df.replace({1000:10,2000:60})
```

Its dp is as follows -

	one	two
0	10	10
1	20	0
2	30	30
3	40	40
4	50	50
5	60	60

Data visualization:

Ex: Training dataset(ML)

```
>>> import pandas as pd
>>> d=pd.read.csv("regress.csv")
>>> d.corr()
```

	x	y
x	1	1
y	1	1

```
>>> d.x.mean()
4.5
>>> d.y.mean()
4.5
>>> d.y.std()
2.449489742783178
>>> d.x.std()
2.449489742783178
>>> b1 = 1*(d.y.std()/d.x.std())
>>> bo = d.y.mean()-(b1*d.x.mean())
>>> b1
1.0
>>> bo
0.0
>>> d.y = b1*d.x+bo
>>> d
   0  1
   1  2
   2  3
   3  4
   4  5
   5  6
   6  7
   7  8
Name: x, dtype: int64
>>> d.y
0    1
1    2
2    3
3    4
4    5
```

```
5    6
6    7
7    8.
Name: 1, dtype: float64.
# Testing
>>> d=pd.read.csv("regress.csv")
>>> d.y = b1*d.x+b0.
>>> d.y
0    1
1    2
2    3
3    4
4    5
5    6
6    7
7    8
8    9
9    10
Name: y; dtype: float64.
>>> d.x
0    1
1    2
2    3
3    4
4    5
5    6
6    7
7    8
8    9
9    10
Name: x, dtyp: float64.
```

Post-Lab:-

Students will be able to aquire different data & perform statical operation on it & display graph of it. In this way student perform datascience operations on any data (IPL dataset.csv).

Conclusion:—

Thus student can implement note book for (perform steps & step4 data science steps for any data (IPL-data) by using python tools like pandas, matplotlib, numpy tec.