LAB CYCLE 1

Experiment No:1
Date : 12/12/2021
AIM:

1. Review of Python Programming and Matrix Operations

    1.1. Create two 3X4 matrices a and b using numpy arrays. Perform the following operations: Display the number of dimensions of the matrices a and b, find the shape of the matrices a and b, find a+b and a-b, a*b(elementwise), multiply the matrix a and its transpose (matrix multiplication), add the value 10 to all elements of a, find transpose of b, calculate average, mean and standard deviation of the elements of the matrix b, find the maximum element in each column and each row of the matrix a, minimum value of the matrix b, reshape b with dimension 2x6 and find the transpose of a.

    1.2. Create a row vector row_a, and a column vector col_a. Create the transpose of col_a, calculate the dot product of col_a with itself. Add the vectors row_a and col_a (broadcasting).

    1.3. Create a dictionary of data. Convert the dictionary to a feature matrix. Display the feature matrix and its column names.

2. Data handling

    You are given the dataset ecom.csv. It contains various properties of E-commerce transactions. Create a pandas dataframe for the dataset. Use appropriate functions to show the shape (number of feature vectors x number of features) of the dataset. Use appropriate slicing functions to show the head and tail ends of the dataset, to display the feature vector corresponding to the row number 180 and to display the set of tuples where mode of shipment is flight and weight is more than 7000 gms. Find the mean, median, mode and variance of the customer rating. Generate descriptive statistics of the numeric features in the dataset.

3. Data Visualization

    Create a dataframe from the python dictionary consisting of three attributes and values: age:[12, 14, 3,8, 7, 5,12, 18,21,19], height:[140, 150, 110, 130, 135, 120, 150, 170, 178, 180], and weight:[40, 50, 10, 30, 35, 20, 50, 70, 78, 80].
    Draw a scatter plot with age and height on the x and y axis respectively.
    Draw a bubble plot with age and height on the x and y axis respectively.
    Draw a density plot for the attribute weight
    Draw a histogram for the attribute age.
    Draw a boxplot for the attribute height.

SOURCE CODE

In [3]:

```
import numpy as np
a= np.array([[1,2,3,1],[1,1,1,2],[1,2,2,1]])

b= np.array([[1,1,3,2],[1,2,1,2],[1,1,2,1]])

print("Matrix a=\n",a)

print("Matrix b=\n",b)
```

```
Matrix a=
 [[1 2 3 1]
 [1 1 1 2]
 [1 2 2 1]]
Matrix b=
 [[1 1 3 2]
 [1 2 1 2]
 [1 1 2 1]]
```

In [4]:

```
a.ndim

b.ndim
```

Out[4]:

```
2
```

In [5]:

```
a.shape

b.shape
```

Out[5]:

```
(3, 4)
```

In [6]:

```
print("Matrix a+b =\n",np.add(a,b))

print("Matrix a-b =\n",np.subtract(a,b))
```

```
Matrix a+b =
 [[2 3 6 3]
 [2 3 2 4]
 [2 3 4 2]]
Matrix a-b =
 [[ 0  1  0 -1]
 [ 0 -1  0  0]
 [ 0  1  0  0]]
```

In [7]:

```
print("Matrix a+b =\n",np.add(a,b))
print("Matrix a-b =\n",np.subtract(a,b))
print("a*b=\n",a*b)
```

```
Matrix a+b =
 [[2 3 6 3]
 [2 3 2 4]
 [2 3 4 2]]
Matrix a-b =
 [[ 0  1  0 -1]
 [ 0 -1  0  0]
 [ 0  1  0  0]]
a*b=
 [[1 2 9 2]
 [1 2 1 4]
 [1 2 4 1]]
```

In [8]:

```
t_a=a.T
```

```
print("transpose a=\n",t_a)
print("Multiplication=\n",np.matmul(a,t_a))
```

```
transpose a=

 [[1 1 1]
 [2 1 2]
 [3 1 2]
 [1 2 1]]
Multiplication=
 [[15  8 12]
 [ 8  7  7]
 [12  7 10]]
```

In [9]:

```
print("Add 10 to all elements in a=\n",np.add(a,10))
```

```
Add 10 to all elements in a=
 [[11 12 13 11]
 [11 11 11 12]
 [11 12 12 11]]
```

In [10]:

```
t_b=b.T
print("Transpose of b=\n",t_b)
```

```
Transpose of b=
 [[1 1 1]
 [1 2 1]
 [3 1 2]
 [2 2 1]]
```

In [11]:

```
print("Average of b = ",np.average(b))
print("Mean of b = ",np.mean(b))
print("Satndard deviation of b= ",np.std(b))
```

```
Average of b =  1.5
Mean of b =  1.5
Satndard deviation of b=  0.6454972243679028
```

In [12]:

```
print("Max element in each column of a= ",np.max(a,axis=0))
print("Max element in each row of a= ",np.max(a,axis=1))
```

```
Max element in each column of a=  [1 2 3 2]
Max element in each row of a=  [3 2 2]
```

In [13]:

```
r_b=b.reshape(2,6)
print("Reshape b as 2X6 = \n",r_b)
```

```
Reshape b as 2X6 =
 [[1 1 3 2 1 2]
 [1 2 1 1 2 1]]
```

In [18]:

```
row_a=np.array([[11,22,33]])
col_a=np.array([[10],[20],[30],[40]])
print("Row vector row_a=\n",row_a)
print("Column vector col_a=\n",col_a)
```

```
Row vector row_a=
 [[11 22 33]]
Column vector col_a=
 [[10]
 [20]
 [30]
 [40]]
```

In [26]:

```
col_a_Trns=col_a.T
print("Transpose of col_a=\n",col_a_Trns)
print("Dot product(col_a.col_a_Trns)=\n",np.dot(col_a,col_a_Trns))
```

```
Transpose of col_a=
 [[10 20 30 40]]
Dot product(col_a.col_a_Trns)=
 [[ 100  200  300  400]
 [ 200  400  600  800]
 [ 300  600  900 1200]
 [ 400  800 1200 1600]]
```

In [27]:

```
print("Adding row_a and col_a=\n",np.add(row_a,col_a))
```

```
Adding row_a and col_a=
 [[21 32 43]
 [31 42 53]
 [41 52 63]
 [51 62 73]]
```

In [28]:

```
from sklearn.feature_extraction import DictVectorizer
```

In [29]:

```
data_dict=[{'White':4,'Black':7},
          {'Orange':3,'White':9},
          {'Red':1,'White':5},
          {'Black':5,'Red':1}]
```

In [30]:

```
dictvectorizer=DictVectorizer(sparse=False)
features=dictvectorizer.fit_transform(data_dict)
```

In [33]:

```
print("Feature matrix=\n",features)
print("feature matrix column names=\n",dictvectorizer.get_feature_names())
```

```
Feature matrix=
 [[7. 0. 0. 4.]
 [0. 3. 0. 9.]
 [0. 0. 1. 5.]
 [5. 0. 1. 0.]]
feature matrix column names=
 ['Black', 'Orange', 'Red', 'White']
```

In [34]:

```
############################################################################
#################################
```

In [35]:

```
#2.DATA HANDLING
```

In [37]:

```
import pandas as pd
import matplotlib.pyplot as plt
```

In [39]:

```
ecom=pd.read_csv("Downloads/ecom.csv")
```

In [41]:

```
print("Shape(number of freature vectors x number of features)= ",ecom.shape)
```

```
Shape(number of freature vectors x number of features)=  (10999, 12)
```

In [42]:

```
print("Head set of dataset = \n",ecom.head())
```

```
Head set of dataset = 
    ID Warehouse_block Mode_of_Shipment  Customer_care_calls  Customer_rat
ing  \
0    1               D           Flight                    4
```

```
2
1   2               F           Flight                    4
5
2   3               A           Flight                    2
2
3   4               B           Flight                    3
3
4   5               C           Flight                    2
2

   Cost_of_the_Product  Prior_purchases Product_importance Gender  \
0                  177                3                low      F
1                  216                2                low      M
2                  183                4                low      M
3                  176                4             medium      M
4                  184                3             medium      F

   Discount_offered  Weight_in_gms  Reached.on.Time_Y.N
0                44           1233                    1
1                59           3088                    1
2                48           3374                    1
3                10           1177                    1
4                46           2484                    1
```

```
In [43]: print("Tail set of dataset = \n",ecom.tail())
```

```
Tail set of dataset =
           ID Warehouse_block Mode_of_Shipment  Customer_care_calls  \
10994  10995               A             Ship                    4
10995  10996               B             Ship                    4
10996  10997               C             Ship                    5
10997  10998               F             Ship                    5
10998  10999               D             Ship                    2

       Customer_rating  Cost_of_the_Product  Prior_purchases  \
10994                1                  252                5
10995                1                  232                5
10996                4                  242                5
10997                2                  223                6
10998                5                  155                5

      Product_importance Gender  Discount_offered  Weight_in_gms  \
10994             medium      F                 1           1538
10995             medium      F                 6           1247
10996                low      F                 4           1155
10997             medium      M                 2           1210
10998                low      F                 6           1639

       Reached.on.Time_Y.N
10994                    1
10995                    0
10996                    0
10997                    0
10998                    0
```

```
In [44]: print("Feature vector corresponding to the row 180 : \n",ecom.iloc[180])
```

```
Feature vector corresponding to the row 180 :
 ID                        181
Warehouse_block              D
Mode_of_Shipment          Ship
Customer_care_calls          4
Customer_rating              1
Cost_of_the_Product        161
Prior_purchases              7
Product_importance      medium
Product_importance     medium
Gender                      F

Discount_offered           18
Weight_in_gms            1294
Reached.on.Time_Y.N         1
Name: 180, dtype: object
```

In [45]:
```
print("The set of tuples where mode of shipment is flight and weight is more
than 7000 gms: \n\n",
     ecom.loc[(ecom["Mode_of_Shipment"]=="flight") & (ecom["Weight_in_gms"]>
7000)])
```

```
The set of tuples where mode of shipment is flight and weight is more than
7000 gms:

 Empty DataFrame
Columns: [ID, Warehouse_block, Mode_of_Shipment, Customer_care_calls, Cust
omer_rating, Cost_of_the_Product, Prior_purchases, Product_importance, Gen
der, Discount_offered, Weight_in_gms, Reached.on.Time_Y.N]
Index: []
```

In [46]:
```
print("Mean of customer rating = ",ecom.Customer_rating.mean())
print("Median of customer rating = ",ecom.Customer_rating.median())
print("Mode of customer rating = ",ecom.Customer_rating.mode())
print("Varience of customer rating = ",ecom.Customer_rating.var())
```

```
Mean of customer rating =  2.9905445949631786
Median of customer rating =  3.0
Mode of customer rating =  0    3
dtype: int64
Varience of customer rating =  1.9982739259753057
```

In [48]:
```
print("Descriptive statistics of the numeric features in the dataset : \n\n"
,ecom.describe())
```

```
Descriptive statistics of the numeric features in the dataset :

                ID  Customer_care_calls  Customer_rating  Cost_of_the_Pro
duct  \
count  10999.00000          10999.000000     10999.000000         10999.000
000
mean    5500.00000              4.054459         2.990545           210.196
836
std     3175.28214              1.141490         1.413603            48.063
272
min        1.00000              2.000000         1.000000            96.000
000
25%     2750.50000              3.000000         2.000000           169.000
000
50%     5500.00000              4.000000         3.000000           214.000
000
75%     8249.50000              5.000000         4.000000           251.000
000
max    10999.00000              7.000000         5.000000           310.000
000

       Prior_purchases  Discount_offered  Weight_in_gms  Reached.on.Time_
Y.N
count     10999.000000      10999.000000   10999.000000         10999.0000
00
mean          3.567597         13.373216    3634.016729             0.5966
91
std           1.522860         16.205527    1635.377251             0.4905
84
min           2.000000          1.000000    1001.000000             0.0000
```

```
00
25%          3.000000        4.000000    1839.500000           0.0000
00
50%          3.000000        7.000000    4149.000000           1.0000
00
75%          4.000000       10.000000    5050.000000           1.0000
00
max         10.000000       65.000000    7846.000000           1.0000
00
```

In [ ]:

In [ ]:

In [49]:

```
##########################################################################
############################
```

In [50]:

```
#3.Data Visualization
```

In [ ]:

In [ ]:

In [51]:

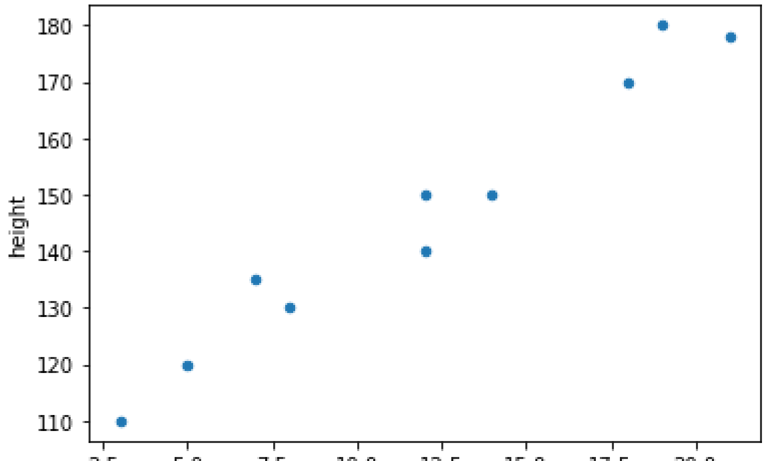
```
data={'age':[12,14,3,8,7,5,12,18,21,19],'height':[140,150,110,130,135,120,15
0,170,178,180],'weight':[40,50,10,30,35,20,50,70,78,80]}
df=pd.DataFrame(data)
print("DataFrame:\n\n",df)
```

```
DataFrame:

    age  height  weight
0   12     140      40
1   14     150      50
2    3     110      10
3    8     130      30
4    7     135      35
5    5     120      20
6   12     150      50
7   18     170      70
8   21     178      78
9   19     180      80
```

In [52]:

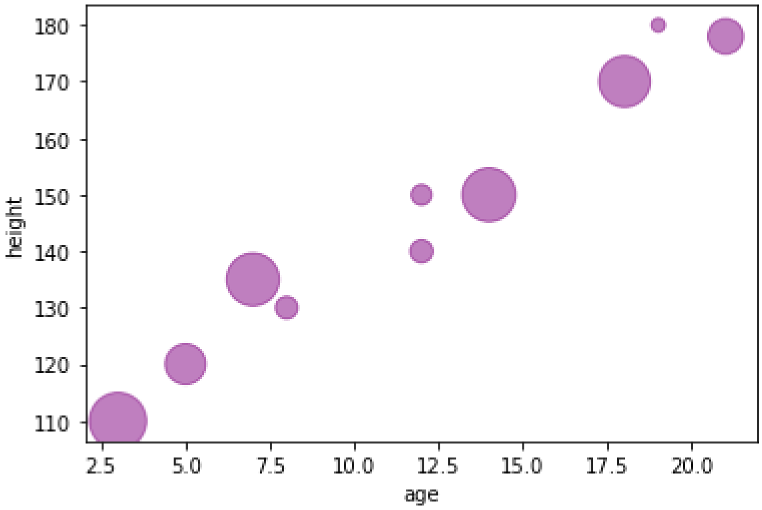
```
print("Scatter plot with age and height on the x and y axis : \n\n",df.plot.
scatter(x='age',y='height'))
```

```
Scatter plot with age and height on the x and y axis :

 AxesSubplot(0.125,0.125;0.775x0.755)
```



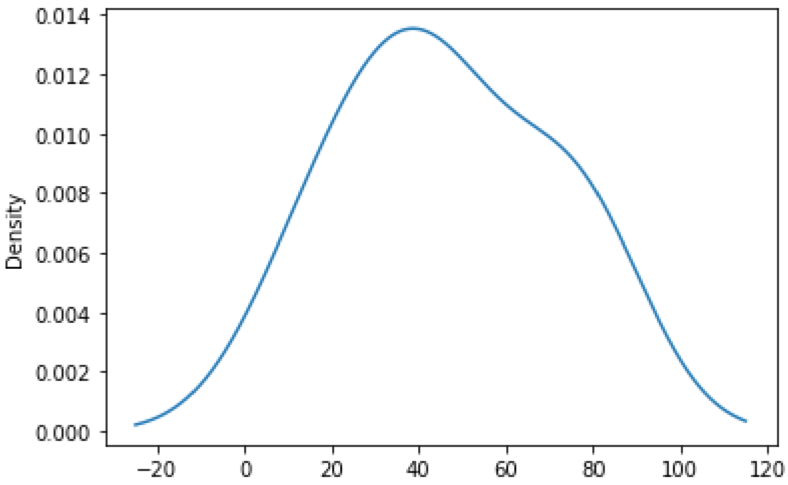
```
In [66]: z=np.random.rand(40)
         print("Bubble plot with age and height on the x and y axis:\n\n",df.plot.sca
         tter('age','height',color='purple',

         alpha=0.5,s=z*1000))
```

```
Bubble plot with age and height on the x and y axis:

 AxesSubplot(0.125,0.125;0.775x0.755)
```



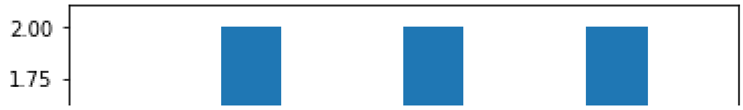
```
In [61]: print("Density plot for the attribute weight",df.weight.plot.density())
```

```
Density plot for the attribute weight AxesSubplot(0.125,0.125;0.775x0.755)
```



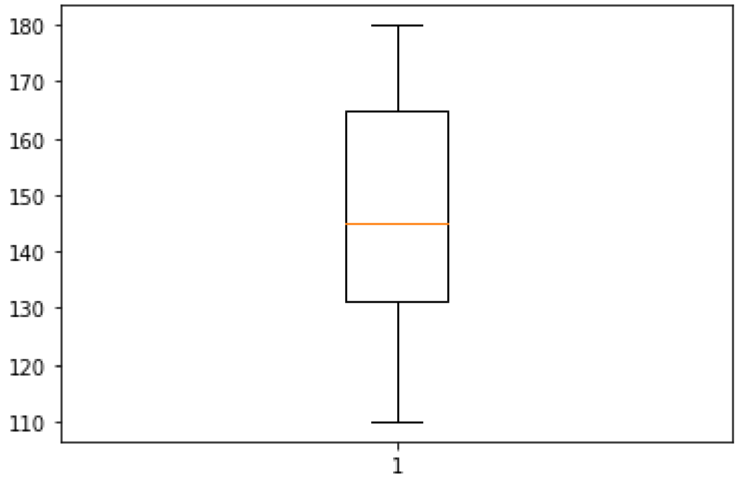
```
In [59]: print("Histogram for the attribute age:\n\n",plt.hist(df.age))
```

```
Histogram for the attribute age:

 (array([1., 1., 2., 0., 0., 2., 1., 0., 2., 1.]), array([ 3. ,  4.8,  6.
6,  8.4, 10.2, 12. , 13.8, 15.6, 17.4, 19.2, 21. ]), <a list of 10 Patch o
bjects>)
```



```
In [60]: print("Boxplot for the attribute height:\n\n",plt.boxplot(df.height))
```

```
Boxplot for the attribute height:

 {'whiskers': [<matplotlib.lines.Line2D object at 0x0000023F1FB04988>, <ma
tplotlib.lines.Line2D object at 0x0000023F1FB04F08>], 'caps': [<matplotli
b.lines.Line2D object at 0x0000023F1FB04B08>, <matplotlib.lines.Line2D obj
ect at 0x0000023F1FB0AA08>], 'boxes': [<matplotlib.lines.Line2D object at
0x0000023F1FB04148>], 'medians': [<matplotlib.lines.Line2D object at 0x000
0023F1FB0AF88>], 'fliers': [<matplotlib.lines.Line2D object at 0x0000023F1
FB0AB88>], 'means': []}
```



```
In [ ]:
```