

1.

Write a program that create Series from ndarray. Print all values in the Series using loop and also print the sliced values from the Series.

Python Code:

```
import numpy as np

#Create Array
arr = np.array((1, 2, 3), (4, 5, 6), (7, 8, 9))
print('Print all values:')
#Print Array
print(arr)
print('\nPrint all values using loop:')
#print Array in loop
for i in range(0, len(arr)):
    print(arr[i])

print('\nAll elements, slice elements from index 1 to index 2 (not included):')
#Print sliced Array in loop
for i in range(0, len(arr)):
    print(arr[i, 1:2])
```

Output:

```
Print all values:
[[1 2 3]
 [4 5 6]
 [7 8 9]]

Print all values using loop:
[1 2 3]
[4 5 6]
[7 8 9]

All elements, slice elements from index 1 to index 2 (not included):
[2]
[5]
[8]
>>>
```

2.

Write a program that create reads marks in Series and prints the finds average of marks.

Python Code:

```
from numpy import *
arr = array([])
#accept number of subjects from user
n = int(input("Enter the number of subject(s) values you want: "))

#accept marks from all subjects from user
for i in range(n):
    v = input("Enter Marks in subject "+str(i+1)+" : ")
    arr = append(arr, int(v))

print("All Marks entered: ")
print(arr)

sum = 0
avg = 0
perc = 0

#calculate sum of all marks
for j in range(0, len(arr)):
    sum = sum + arr[j]

#print Sum
print("Sum of all marks: ",str(sum))
#Calculate and print Average marks
print("Average marks: ",str(sum/len(arr)))
#Calculate and print Percentage
print("Percentage: ",str((sum/(100*len(arr)))*100))
```

Output:

```
Enter the number of subject(s) values you want: 5
Enter Marks in subject 1 : 95
Enter Marks in subject 2 : 86
Enter Marks in subject 3 : 82
Enter Marks in subject 4 : 83
Enter Marks in subject 5 : 78
All Marks entered:
[95. 86. 82. 83. 78.]
Sum of all marks: 424.0
Average marks: 84.8
Percentage: 84.8
>>>
```

3.

Create a Data Frame quarterly sales where each row contains the item category, item name, and expenditure. Group the rows by the category and print the total expenditure per category. (Read 3 values for each category. Total category should be 4).

Python Code:

```
import pandas as pd

#initialize list of lists
print("initialize list of lists:")
data = [['Software', 'Micosoft', 10000], ['Processor', 'Intel', 550000], ['Motherboard', 'Asus', 1200000],
['Monitor', 'LG', 15000], ['Software', 'Google', 70000], ['Processor', 'Asus', 450000], ['Motherboard', 'Intel', 200000], ['Monitor', 'Dell', 25000]]

#Set column names/header
Col=['Item_Category', 'Name', 'Expenditure']

print("Create the pandas DataFrame:")
qrtsales = pd.DataFrame(data, columns=Col)

print("print dataframe:")
print (qrtsales)

qs=qrtsales.groupby('Item_Category')
print('Result after Filtering Dataframe')
print(qs['Item_Category', 'Expenditure'].sum())
```

Output:

```
initialize list of lists:
Create the pandas DataFrame:
print dataframe:
  Item_Category  Name  Expenditure
0    Software  Micosoft      10000
1    Processor   Intel    550000
2  Motherboard   Asus   1200000
3    Monitor    LG       15000
4    Software   Google    70000
5    Processor   Asus    450000
6  Motherboard   Intel    200000
7    Monitor    Dell     25000
Result after Filtering Dataframe
      Expenditure
Item_Category
Monitor         40000
Motherboard    1400000
Processor     1000000
Software       80000
>>>
```

4.
Create a data frame for examination result and display row labels, column labels data types of each column and the dimensions.

Python Code:

```
import pandas as pd

print("initialize array:")
dic={'Class' : ['I','II','III','IV','V','VI','VII','VIII','IX','X'], 'Pass-Percentage': [100,100,100,100,100,100,100,100,100,100]}

print("Create the pandas DataFrame:")
result = pd.DataFrame(dic)
print(result)
print(result.dtypes)
print('Shape of dataframe is:::')
print(result.shape)
```

Output:

```
initialize array:
Create the pandas DataFrame:
  Class  Pass-Percentage
0     I                100
1    II                100
2   III                100
3    IV                100
4     V                100
5    VI                100
6   VII                100
7  VIII                100
8    IX                100
9     X                100
Class                object
Pass-Percentage      int64
dtype: object
Shape of dataframe is:::
(10, 2)
>>> |
```

5.
Create a DataFrame as shown below and perform following operations on it.

Product	Company	Quantity	Price
CPU	Compaq	40	9000
Mouse	Dell	20	500
Keyboard	HP	15	500
Printer	Epson	5	5700
Hard Disk	Toshiba	10	2000
Plotter	Sony	5	8000

- Add a new column Total Price that contains Total Price of Product (Total Price = Quantity * Price)
- Add a new row to the DataFrame with Data – (Scanner, HP, 2, 9500)
- Display the Product and Total Price of the DataFrame.

Python Code:

```
import pandas as pd
print("initialize list of lists:")
dict = {'Product': ['CPU', 'Mouse', 'Keyboard', 'Printer', 'Hard_Disk', 'Plotter'],
        'Company': ['Compaq', 'Dell', 'HP', 'Epson', 'Toshiba', 'Sony'],
        'Quantity': [40, 20, 15, 5, 10, 5],
        'Price': [900, 500, 500, 5700, 2000, 8000]}
print("Create the pandas DataFrame:")
df = pd.DataFrame(dict)
print("print DataFrame:")
print(df)

#b) Add a new row to the DataFrame with Data - (Scanner, HP, 2, 9500)
df.loc[len(df.index)] = ['Scanner', 'HP', 2, 9500]

print("print NEW DataFrame:")
print(df)

Total_Price=[]
for index, row in df.iterrows():
    Total_Price.append(int(row['Quantity'])*int(row['Price']))
#a) Add a new column Total Price that contains Total Price of Product (Total Price = Quantity * Price)
df['Total_Price'] = Total_Price
#c) Display the Product and Total Price of the DataFrame.
print("print NEW DataFrame with Total_Price:")
print(df)
```

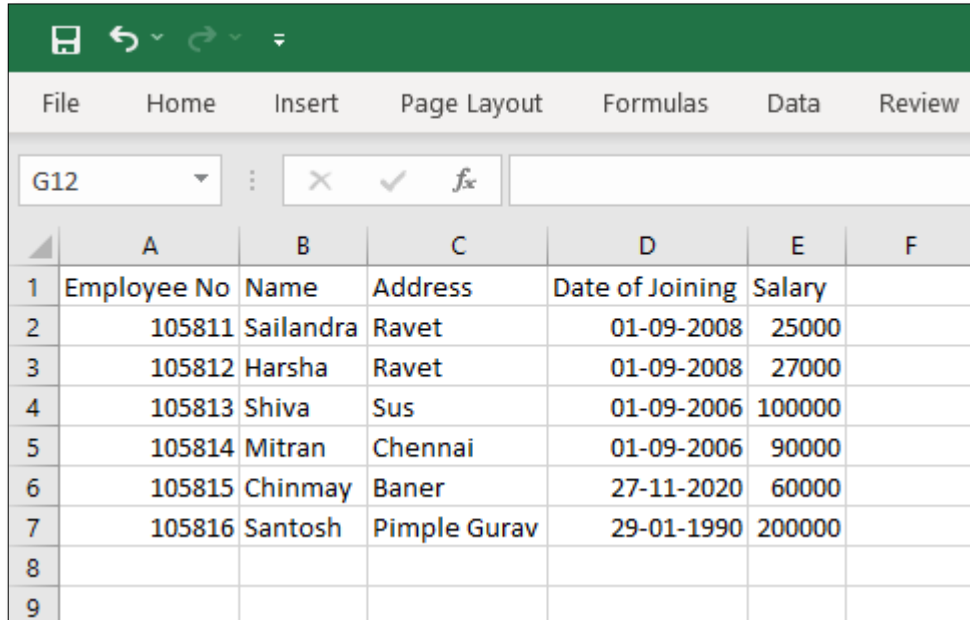
Output:

```
initialize list of lists:
Create the pandas DataFrame:
print DataFrame:
   Product  Company  Quantity  Price
0      CPU   Compaq       40    900
1    Mouse    Dell       20    500
2  Keyboard     HP       15    500
3   Printer   Epson        5   5700
4  Hard_Disk Toshiba       10   2000
5   Plotter    Sony        5   8000
print NEW DataFrame:
   Product  Company  Quantity  Price
0      CPU   Compaq       40    900
1    Mouse    Dell       20    500
2  Keyboard     HP       15    500
3   Printer   Epson        5   5700
4  Hard_Disk Toshiba       10   2000
5   Plotter    Sony        5   8000
6  Scanner     HP         2   9500
print NEW DataFrame with Total_Price:
   Product  Company  Quantity  Price  Total_Price
0      CPU   Compaq       40    900         36000
1    Mouse    Dell       20    500         10000
2  Keyboard     HP       15    500          7500
3   Printer   Epson        5   5700         28500
4  Hard_Disk Toshiba       10   2000         20000
5   Plotter    Sony        5   8000         40000
6  Scanner     HP         2   9500         19000
>>>
```

6.

Create a DataFrame that reads Employee Data (Employee No, Name, Address, Date of Joining, Salary) from CSV File and displays the details.

Python Code:



The screenshot shows a Microsoft Excel spreadsheet with a green header bar and a ribbon menu (File, Home, Insert, Page Layout, Formulas, Data, Review). The active cell is G12. The data is organized in a table with 6 columns: Employee No, Name, Address, Date of Joining, Salary, and an empty column F. The data rows are numbered 1 through 7.

	A	B	C	D	E	F
1	Employee No	Name	Address	Date of Joining	Salary	
2	105811	Sailandra	Ravet	01-09-2008	25000	
3	105812	Harsha	Ravet	01-09-2008	27000	
4	105813	Shiva	Sus	01-09-2006	100000	
5	105814	Mitran	Chennai	01-09-2006	90000	
6	105815	Chinmay	Baner	27-11-2020	60000	
7	105816	Santosh	Pimple Gurav	29-01-1990	200000	
8						
9						

```
import pandas as pd
#reads Employee Data (Employee No, Name, Address, Date of Joining, Salary) from CSV
emp_data = pd.read_csv('006.csv')
#displays the details
print(emp_data.head())
```

Output:

```
Employee No      Name      Address  Date of Joining  Salary
0      105811  Sailandra    Ravet      01-09-2008     25000
1      105812    Harsha    Ravet      01-09-2008     27000
2      105813    Shiva      Sus      01-09-2006    100000
3      105814    Mitran  Chennai      01-09-2006     90000
4      105815   Chinmay    Baner      27-11-2020     60000
>>>
```

7.

Create a DataFrame with details BookID, Book Description, Author, Publisher, Quantity and Price. Store the data into a CSV File with name “Employee_Data”.

Python Code:

```
import pandas as pd
#DataFrame with details BookID, Book Description, Author, Publisher,Quantity and Price
Employee_Data = pd.DataFrame([['001', 'Engg Book', 'Harsha', 'CDAC', '10000', 'Rs.5000'],
                               ['002', 'Medical Book', 'Sail', 'IEEE', '1000', 'Rs.12000'],
                               ['003', 'Engg Book', 'Swati', 'Google', '90000', 'Rs.15000']],
                              columns=['BookID', 'Book Description', 'Author', 'Publisher', 'Quantity', 'Price'])
#Store the data into a CSV File with name "Employee_Data"
Employee_Data.to_csv('Employee_Data.csv')
```

Output:

```
CSV file created with Name: Employee_Data.csv
>>>
```

	A	B	C	D	E	F	G	H
1		BookID	Book Desc	Author	Publisher	Quantity	Price	
2	0	1	Engg Book	Harsha	CDAC	10000	Rs.5000	
3	1	2	Medical B	Sail	IEEE	1000	Rs.12000	
4	2	3	Engg Book	Swati	Google	90000	Rs.15000	
5								
6								
7								

8.

Consider DataFrame df as shown below :

	name	Age	weight	height	runsscored
0	mayur	15	51	5.1	55
1	anil	16	48	5.2	25
2	viraj	17	49	5.1	71
3	viraj	17	51	5.3	53
4	mahesh	16	48	5.1	51
5	viraj	17	59	5.3	50

Write commands to :

- Write command to calculate minimum value for each of the row from subset of dataframe that contains age, weight, height, runsscored
- Write command to calculate mean for last 3 rows.

Suryadatta National School
Class 12 CBSE
Informatics Practices Practicals List

Python Code:

```
import pandas as pd
#DataFrame with details BookID, Book Description, Author, Publisher,Quantity and Price
Employee_Data = pd.DataFrame([[ 'mayur',15,51,5.1,55],
                               [ 'anil',16,48,5.2,25],
                               [ 'viraj',17,49,5.1,71],
                               [ 'viraj',17,51,5.3,53],
                               [ 'mahesh',16,48,5.1,51],
                               [ 'viraj',17,59,5.3,50]],
                              columns=['name', 'Age', 'weight', 'height', 'runsscored'])

print(Employee_Data)

#a) calculate minimum value for each of the row from subset of dataframe that contains age, weight, height, runsscored
minvalue_series = Employee_Data.min()
print(minvalue_series)

#Get Last 03 Rows
Employee_Data_last_3 = Employee_Data.tail(3)

#b) calculate mean for last 3 rows
df_mean = Employee_Data_last_3[['name', 'Age', 'weight', 'height', 'runsscored']].mean()
print(df_mean)

#Store the data into a CSV File with name "008.csv"
Employee_Data.to_csv('008.csv')
```

Output:

```
   name  Age  weight  height  runsscored
0  mayur   15     51     5.1         55
1   anil   16     48     5.2         25
2  viraj   17     49     5.1         71
3  viraj   17     51     5.3         53
4  mahesh  16     48     5.1         51
5  viraj   17     59     5.3         50
name      anil
Age         15
weight      48
height      5.1
runsscored  25
dtype: object
Age      16.666667
weight   52.666667
height    5.233333
runsscored 51.333333
dtype: float64
>>>
```

File Home Insert Page Layout Formulas Data Review							
A1							
	A	B	C	D	E	F	G
1		name	Age	weight	height	runsscored	
2	0	mayur	15	51	5.1	55	
3	1	anil	16	48	5.2	25	
4	2	viraj	17	49	5.1	71	
5	3	viraj	17	51	5.3	53	
6	4	mahesh	16	48	5.1	51	
7	5	viraj	17	59	5.3	50	
8							

9.

Write a code to create following dataframe.

Booking Code	Customer Name	No of Ticket	Ticket Rate	Booking Clerk
B001	Veer	4	100	Manish
B002	Umesh	2	200	Kishor
B003	Lavanya	6	150	Manish
B004	Shobhana	5	250	John
B005	Piyush	3	100	Kishor

Do the following:

- Add column "Total Amount" that calculates total amount of tickets and assign that to new column.
- Add a new row with values (B006 , Vijay, 7, 150, John). Calculate the total amount of tickets and assign it to Total Amount column.

Python Code:

```
import pandas as pd
#DataFrame with details BookID, Book Description, Author, Publisher,Quantity and Price
Booking_Data = pd.DataFrame([[['B001','Veer',4,100,'Manish']],
                              [['B002','Umesh',2,200,'Kishor']],
                              [['B003','Lavanya',6,150,'Manish']],
                              [['B004','Shobhana',5,250,'John']],
                              [['B005','Piyush',3,100,'Kishor']]],
                              columns=['Booking Code','Customer Name','No of Ticket','Ticket Rate','Booking Clerk'])

print(Booking_Data)

#b) Add a new row with values ( B006 , Vijay, 7, 150, John)
Booking_Data.loc[len(Booking_Data.index)] = ['B006','Vijay',7,150,'John']

print("print NEW Booking Data:")
print(Booking_Data)

Total_Amount=[]
for index, row in Booking_Data.iterrows():
    Total_Amount.append(int(row['No of Ticket'])*int(row['Ticket Rate']))
#a) Add column "Total Amount" that calculates total amount of tickets and assign that to new column.
Booking_Data['Total_Amount'] = Total_Amount
print("print NEW DataFrame with Total_Amount:")
print(Booking_Data)
```

Output:

```
Booking Code Customer Name No of Ticket Ticket Rate Booking Clerk
0 B001 Veer 4 100 Manish
1 B002 Umesh 2 200 Kishor
2 B003 Lavanya 6 150 Manish
3 B004 Shobhana 5 250 John
4 B005 Piyush 3 100 Kishor
print NEW Booking Data:
Booking Code Customer Name No of Ticket Ticket Rate Booking Clerk
0 B001 Veer 4 100 Manish
1 B002 Umesh 2 200 Kishor
2 B003 Lavanya 6 150 Manish
3 B004 Shobhana 5 250 John
4 B005 Piyush 3 100 Kishor
5 B006 Vijay 7 150 John
print NEW DataFrame with Total_Amount:
Booking Code Customer Name ... Booking Clerk Total_Amount
0 B001 Veer ... Manish 400
1 B002 Umesh ... Kishor 400
2 B003 Lavanya ... Manish 900
3 B004 Shobhana ... John 1250
4 B005 Piyush ... Kishor 300
5 B006 Vijay ... John 1050

[6 rows x 6 columns]
>>>
```

10.
Read the subjects and marks/grades for different subject of Class 12 and plot column chart on that. Also display title, x-axis and y-axis labels.

Python Code:

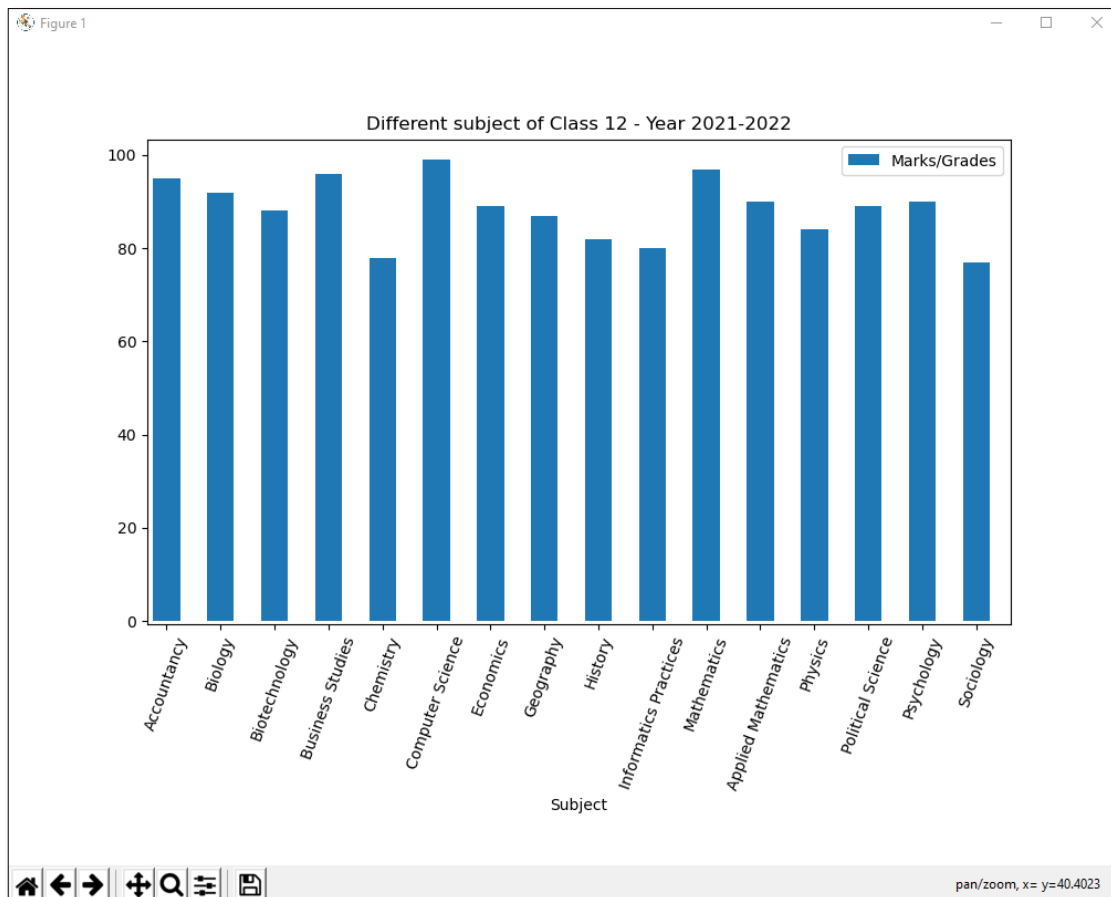
```
import pandas as pd
import matplotlib.pyplot as plot

# A python dictionary
data = {"Subject":['Accountancy',
                  'Biology',
                  'Biotechnology',
                  'Business Studies',
                  'Chemistry',
                  'Computer Science',
                  'Economics',
                  'Geography',
                  'History',
                  'Informatics Practices',
                  'Mathematics',
                  'Applied Mathematics',
                  'Physics',
                  'Political Science',
                  'Psychology',
                  'Sociology'],
        "Marks/Grades": [95, 92, 88, 96, 78, 99, 89, 87, 82, 80, 97, 90, 84, 89, 90, 77]};

# Dictionary loaded into a DataFrame
dataFrame = pd.DataFrame(data=data);

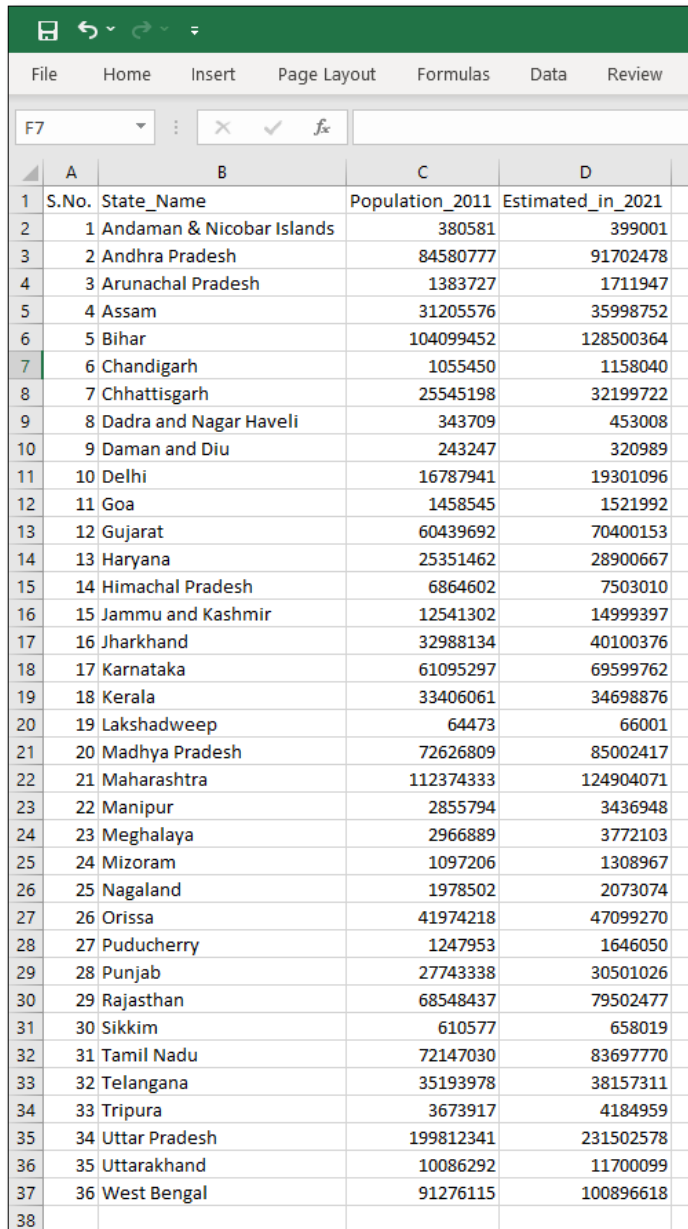
# Draw a vertical bar chart
dataFrame.plot.bar(x="Subject", y="Marks/Grades", rot=70, title="Different subject of Class 12 - Year 2021-2022");
plot.show(block=True);
```

Output:



11.
Read data from open source (e.g. data.gov.in), aggregate and summarize it. Then plot it using different plotting functions of the Matplotlib library.

Python Code:



	A	B	C	D
1	S.No.	State_Name	Population_2011	Estimated_in_2021
2	1	Andaman & Nicobar Islands	380581	399001
3	2	Andhra Pradesh	84580777	91702478
4	3	Arunachal Pradesh	1383727	1711947
5	4	Assam	31205576	35998752
6	5	Bihar	104099452	128500364
7	6	Chandigarh	1055450	1158040
8	7	Chhattisgarh	25545198	32199722
9	8	Dadra and Nagar Haveli	343709	453008
10	9	Daman and Diu	243247	320989
11	10	Delhi	16787941	19301096
12	11	Goa	1458545	1521992
13	12	Gujarat	60439692	70400153
14	13	Haryana	25351462	28900667
15	14	Himachal Pradesh	6864602	7503010
16	15	Jammu and Kashmir	12541302	14999397
17	16	Jharkhand	32988134	40100376
18	17	Karnataka	61095297	69599762
19	18	Kerala	33406061	34698876
20	19	Lakshadweep	64473	66001
21	20	Madhya Pradesh	72626809	85002417
22	21	Maharashtra	112374333	124904071
23	22	Manipur	2855794	3436948
24	23	Meghalaya	2966889	3772103
25	24	Mizoram	1097206	1308967
26	25	Nagaland	1978502	2073074
27	26	Orissa	41974218	47099270
28	27	Puducherry	1247953	1646050
29	28	Punjab	27743338	30501026
30	29	Rajasthan	68548437	79502477
31	30	Sikkim	610577	658019
32	31	Tamil Nadu	72147030	83697770
33	32	Telangana	35193978	38157311
34	33	Tripura	3673917	4184959
35	34	Uttar Pradesh	199812341	231502578
36	35	Uttarakhand	10086292	11700099
37	36	West Bengal	91276115	100896618
38				

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

#Read data from open source (e.g. data.gov.in)
df = pd.read_csv('011.csv', sep=',');
print('Original dataframe:')
#print Original dataframe
print(df)
print('aggregate:')
#calculate and print aggregate
print(df.agg(['sum', 'min']))

x = df.State_Name
y = df.Population_2011
plot.scatter(x, y)
plot.show()
```

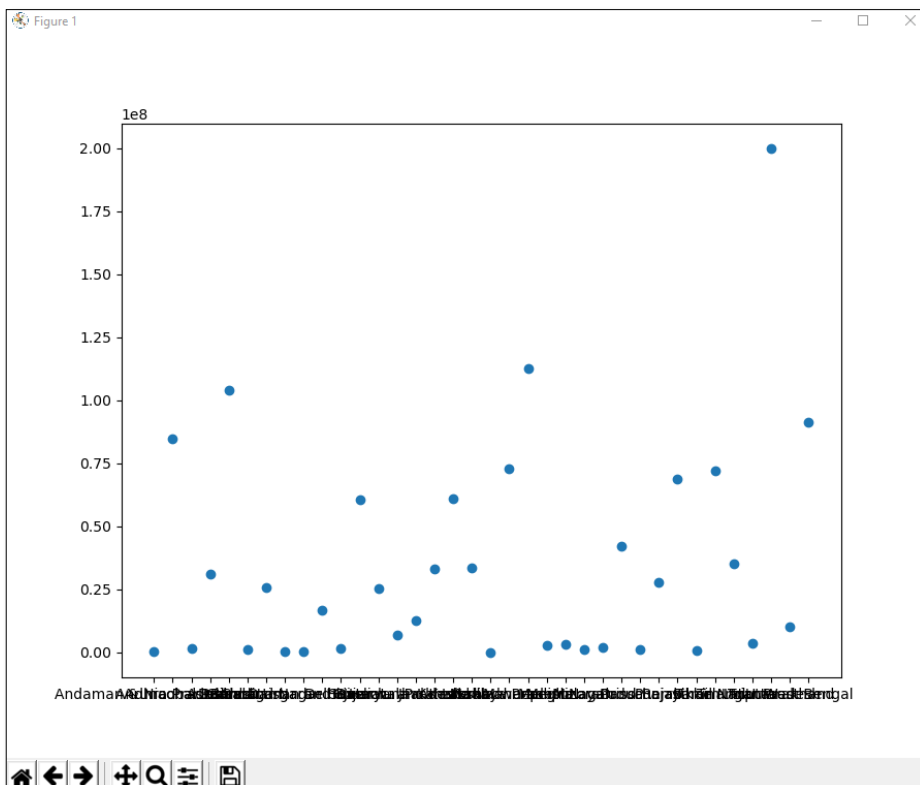
Suryadatta National School
Class 12 CBSE
Informatics Practices Practicals List

Output:

```
Original dataframe:
  S.No.  State_Name  Population_2011  Estimated_in_2021
0      1  Andaman & Nicobar Islands      380581      399001
1      2      Andhra Pradesh      84580777      91702478
2      3      Arunachal Pradesh      1383727      1711947
3      4          Assam      31205576      35998752
4      5          Bihar      104099452      128500364
5      6      Chandigarh      1055450      1158040
6      7      Chhattisgarh      25545198      32199722
7      8  Dadra and Nagar Haveli      343709      453008
8      9      Daman and Diu      243247      320989
9     10          Delhi      16787941      19301096
10     11          Goa      1458545      1521992
11     12          Gujarat      60439692      70400153
12     13          Haryana      25351462      28900667
13     14      Himachal Pradesh      6864602      7503010
14     15  Jammu and Kashmir      12541302      14999397
15     16          Jharkhand      32988134      40100376
16     17          Karnataka      61095297      69599762
17     18          Kerala      33406061      34698876
18     19      Lakshadweep      64473      66001
19     20      Madhya Pradesh      72626809      85002417
20     21      Maharashtra      112374333      124904071
21     22          Manipur      2855794      3436948
22     23      Meghalaya      2966889      3772103
23     24          Mizoram      1097206      1308967
24     25          Nagaland      1978502      2073074
25     26          Orissa      41974218      47099270
26     27      Puducherry      1247953      1646050
27     28          Punjab      27743338      30501026
28     29      Rajasthan      68548437      79502477
29     30          Sikkim      610577      658019
30     31          Tamil Nadu      72147030      83697770
31     32          Telangana      35193978      38157311
32     33          Tripura      3673917      4184959
33     34      Uttar Pradesh      199812341      231502578
34     35      Uttarakhand      10086292      11700099
35     36      West Bengal      91276115      100896618

aggregate:
  S.No.  ... Estimated_in_2021
sum    666  ...      1429579388
min      1  ...           66001

[2 rows x 4 columns]
```



12.

Draw the histogram based on the Production of Corn in different Years

'Year':2003,2005,2007,2009,2011,2013,2015,2017,2019,2021

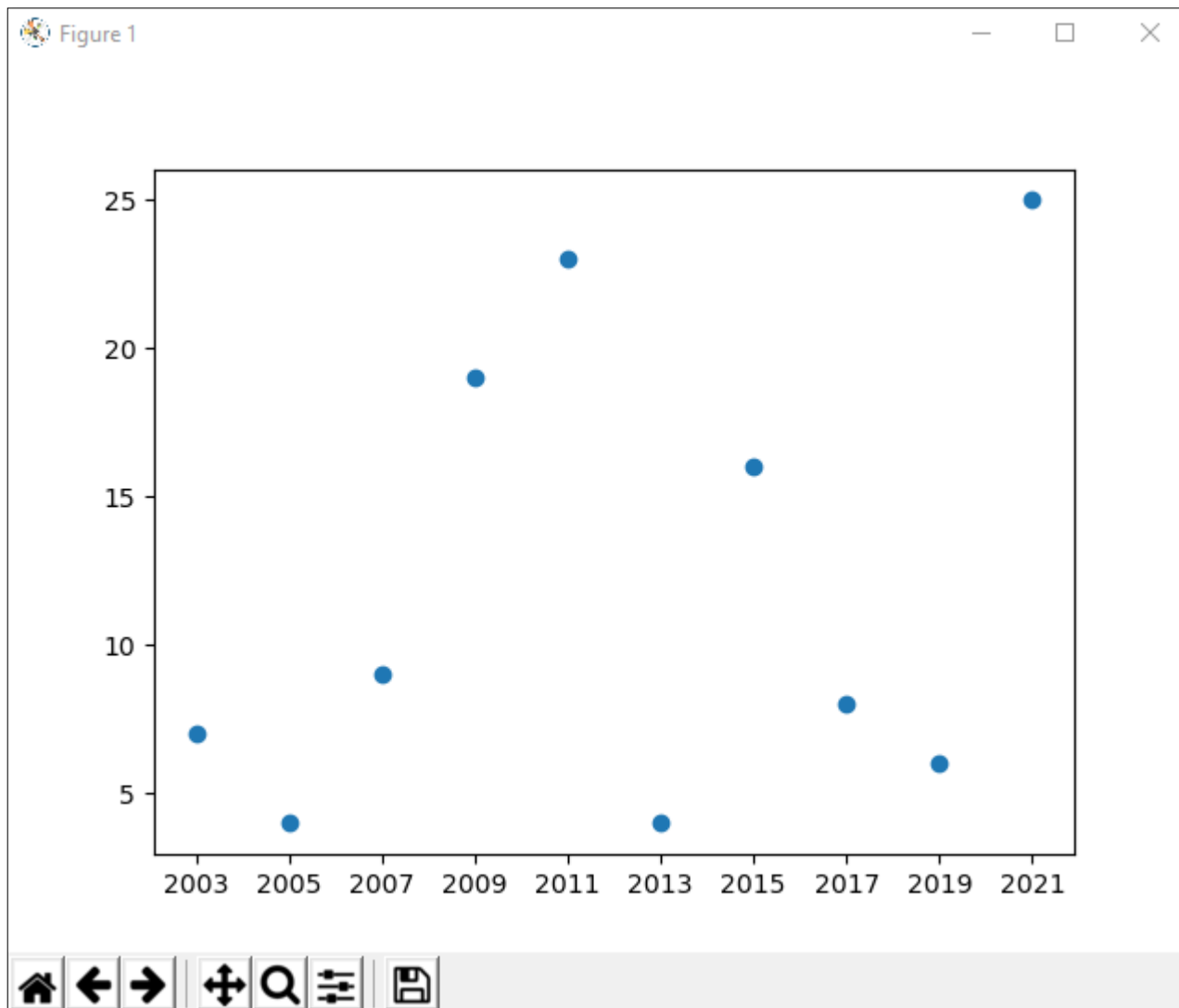
'Production': 7,4,9,19,23,4,16,8,6,25

Python Code:

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

Production_Data = pd.DataFrame({'Year':['2003','2005','2007','2009','2011','2013','2015','2017','2019','2021'],
                                'Production':[7,4,9,19,23,4,16,8,6,25]})
x = Production_Data.Year
y = Production_Data.Production
plt.scatter(x, y)
plt.show()
```

Output:



13.

Observe the given data for Football Players and their goals scored in the matches. Plot them on the bar chart.

Ronaldo	Pele	Messi	Gerd Muller	Eusebio	Maravilla
783	767	755	735	623	575

Python Code:

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

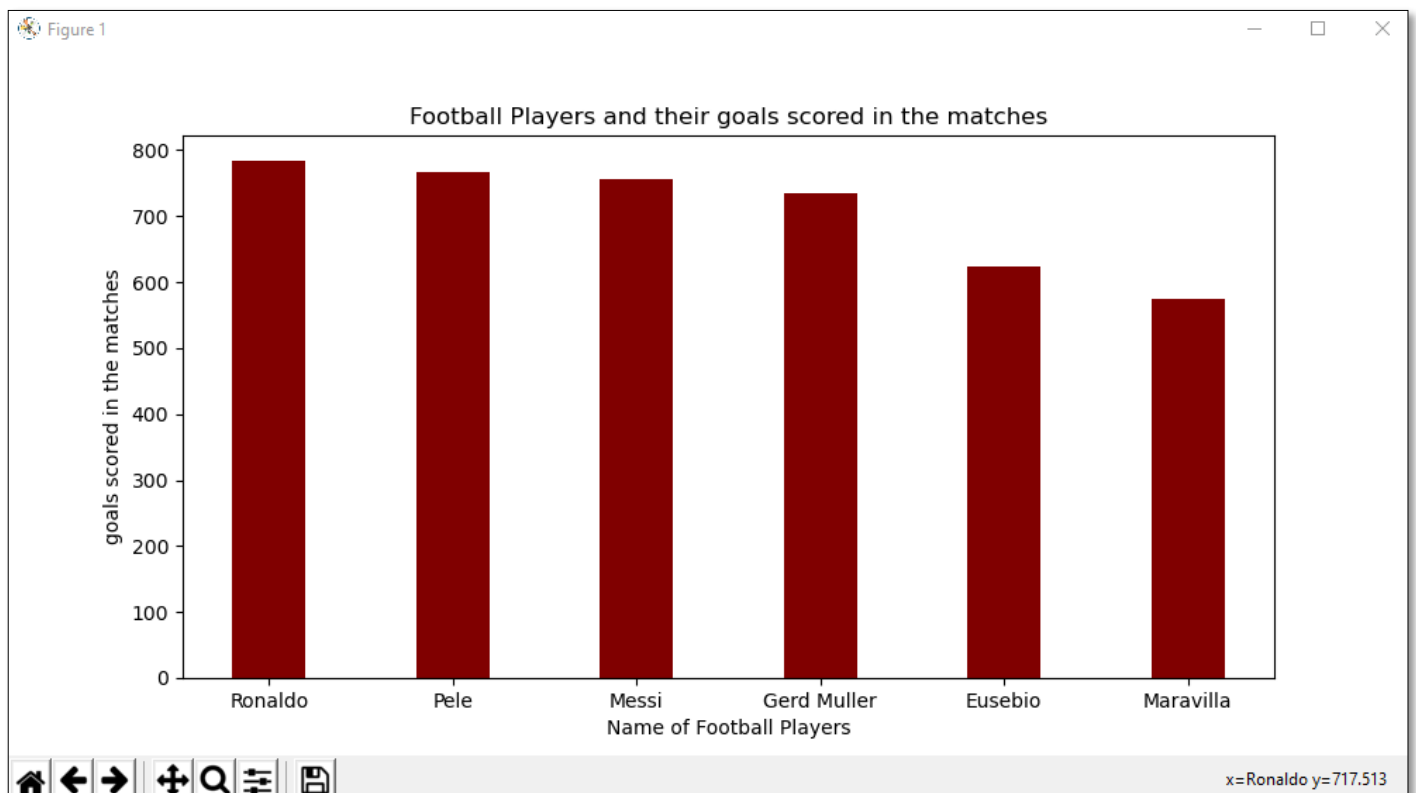
# creating the dataset
data = {'Ronaldo':783, 'Pele':767, 'Messi':755, 'Gerd Muller':735, 'Eusebio':623, 'Maravilla':575}
courses = list(data.keys())
values = list(data.values())

fig = plt.figure(figsize = (10, 5))

# creating the bar plot
plt.bar(courses, values, color='maroon',
        width = 0.4)

plt.xlabel("Name of Football Players")
plt.ylabel("goals scored in the matches")
plt.title("Football Players and their goals scored in the matches")
plt.show()
```

Output:



14.

Consider the data given below for creating bar and line chart

Zones	June	July	Aug	Sept	Oct	Nov
North	140	130	130	190	160	200
South	160	200	130	200	170	190
East	140	180	130	170	190	180
West	180	190	200	120	180	190
Central	110	160	130	110	170	130

- a) Create a bar chart to distribution of rainfall from June to Nov for all zones. b)
Create line chart to observe any trends from June to Nov.

Python Code:

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

barWidth = 0.1
fig = plt.subplots(figsize=(20, 12))

June = [140,160,140,180,110]
July = [130,200,180,190,160]
Aug = [130,130,130,200,130]
Sept = [190,200,170,120,110]
Oct = [160,170,190,180,170]
Nov = [200,190,180,190,130]

#For Bar Chart Start#
br1 = np.arange(len(June))
br2 = [x + barWidth for x in br1]
br3 = [x + barWidth for x in br2]
br4 = [x + barWidth for x in br3]
br5 = [x + barWidth for x in br4]
br6 = [x + barWidth for x in br5]

plt.bar(br1, June, color='r', width = barWidth,edgecolor='grey', label='June')
plt.bar(br2, July, color='g', width = barWidth,edgecolor='grey', label='July')
plt.bar(br3, Aug, color='b', width = barWidth,edgecolor='grey', label='Aug')
plt.bar(br4, Sept, color='gold', width = barWidth,edgecolor='grey', label='Sept')
plt.bar(br5, Oct, color='pink', width = barWidth,edgecolor='grey', label='Oct')
plt.bar(br6, Nov, color='black', width = barWidth,edgecolor='grey', label='Nov')

plt.xlabel('Zone', fontweight='bold', fontsize = 15)
plt.ylabel('rainfall', fontweight='bold', fontsize = 15)
plt.xticks([r + barWidth for r in range(len(June))],['North', 'South', 'East', 'West', 'Central'])

plt.legend()
plt.show()
#For Bar Chart End#

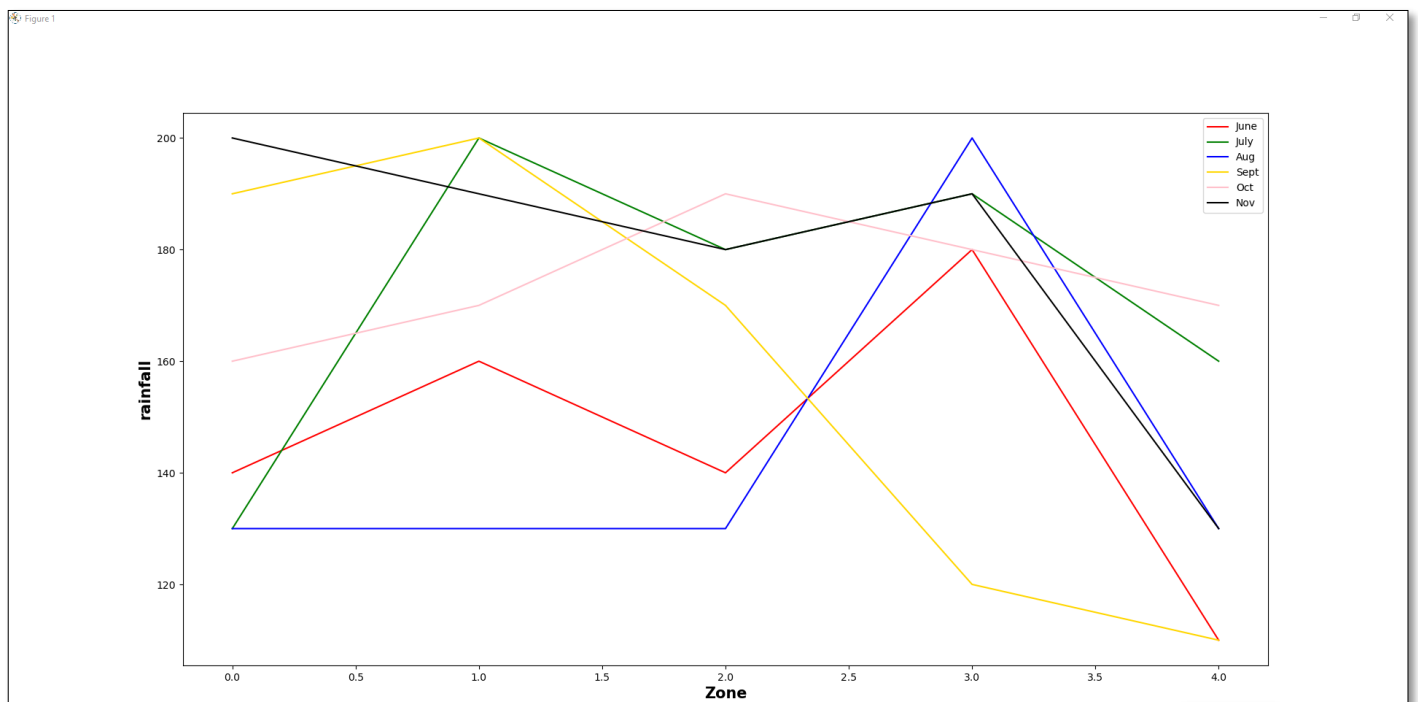
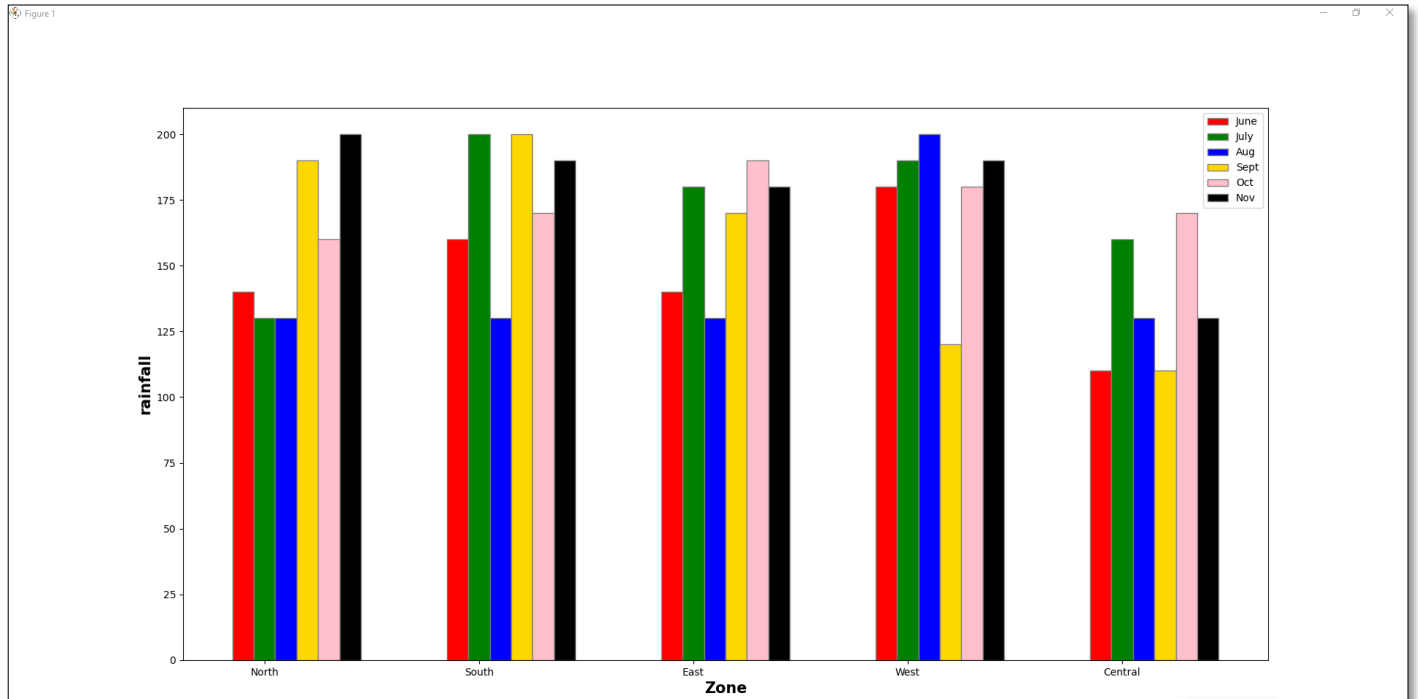
#For Line Chart Start#
plt.plot(June, color='r', label='June')
plt.plot(July, color='g', label='July')
plt.plot(Aug, color='b', label='Aug')
plt.plot(Sept, color='gold', label='Sept')
plt.plot(Oct, color='pink', label='Oct')
plt.plot(Nov, color='black', label='Nov')

plt.xlabel('Zone', fontweight='bold', fontsize = 15)
plt.ylabel('rainfall', fontweight='bold', fontsize = 15)

plt.legend()
plt.show()
#For Line Chart End#
```

Suryadatta National School
Class 12 CBSE
Informatics Practices Practicals List

Output:



15.

Write a program in Python Pandas to create the following DataFrame 'Library'.

Bid	Name	Author	Price	Mem_Name	Issue_Date	Status
B01	Wings of Fire	A.P.J Abdul Kalam	450	Pranjal	2021-04-11	Not Returned
B02	The Monk who sold his Ferrari	Robin Sharma	370	Kunal	2021-03-15	Returned
B03	You Can Win	Shiv Khera	350	Rajat	2021-04-18	Not Returned
B05	Who moved my cheese	Spenser Jhonson	450	Roma	2021-02-27	Returned
B06	Real Success	Patrick Mather Pike	250	Sia	2021-04-23	Not Returned

- Display DataFrame 'Library'.
- Display the Book names having price above 350.

Python Code:

```
import pandas as pd

Library = {
    'Bid': ['B01', 'B02', 'B03', 'B05', 'B06'],
    'Name': ['Wings of Fire', 'The Monk who sold his Ferrari', 'You Can Win', 'Who moved my cheese', 'Real Success'],
    'Author': ['A.P.J Abdul Kalam', 'Robin Sharma', 'Shiv Khera', 'Spenser Jhonson', 'Patrick Mather Pike'],
    'Price': [450, 370, 350, 450, 250],
    'Mem_Name': ['Pranjal', 'Kunal', 'Rajat', 'Roma', 'Sia'],
    'Issue_Date': ['2021-04-11', '2021-03-15', '2021-04-18', '2021-02-27', '2021-04-23'],
    'Status': ['Not Returned', 'Returned', 'Not Returned', 'Returned', 'Not Returned']}
labels = ['a', 'b', 'c', 'd', 'e']

df = pd.DataFrame(Library, columns=['Bid', 'Name', 'Author', 'Price', 'Mem_Name', 'Issue_Date', 'Status'])
print("Given Dataframe :\n", df)

rslt_df = df[df['Price'] > 350]

print('\nResult dataframe :\n', rslt_df)
```

Output:

```
Given Dataframe :
   Bid      Name ... Issue_Date  Status
0  B01  Wings of Fire ... 2021-04-11  Not Returned
1  B02  The Monk who sold his Ferrari ... 2021-03-15  Returned
2  B03      You Can Win ... 2021-04-18  Not Returned
3  B05  Who moved my cheese ... 2021-02-27  Returned
4  B06      Real Success ... 2021-04-23  Not Returned

[5 rows x 7 columns]

Result dataframe :
   Bid      Name ... Issue_Date  Status
0  B01  Wings of Fire ... 2021-04-11  Not Returned
1  B02  The Monk who sold his Ferrari ... 2021-03-15  Returned
3  B05  Who moved my cheese ... 2021-02-27  Returned

[3 rows x 7 columns]
>>>
```