



DARSHAN ACADEMY, MEERUT

Sawan Ashram, West End Road, Meerut

A Practical Record File

**is submitted for the partial fulfilment of
AISSCE Examination 2024-25**



Under the guidance of Gopal Singh (PGT - Computer)

Submitted By: Krishna Bansal

Roll Number: 21716370

TABLE OF CONTENTS

S. No.	Assignment	Page No.
Programs using Pandas		
SERIES		
1	Creating a Python program to create a series using a dictionary	3
2	Creating a Python program to create a series using scalar value	4
3	Creating a Python program to create a series using NumPy array	5
4	Creating a python program for modifying or updating existing values of series object	6
5	Creating a Python program for performing mathematical operations on two Series objects	8
6	Creating a Python program for calculating per capita income of four zones using Series	10
7	Creating a Python program to display attributes of a Series	11
8	Creating a Python program using head() and tail() in Series	12
DATAFRAME		
9	Creating a Python program for creating a DataFrame using a nested list	13
10	Creating a python program for accessing values of rows and columns of a dataframe.	14
11	Creating a Python program for accessing values of rows and columns of a dataframe.	16
12	Creating a python program to perform operations on a DataFrame (rename, count, update, replace)	17
13	Creating a Python program to filter the data of a DataFrame	19
14	Creating a Python program to display the attributes of the DataFrame	20
15	Creating a Python program to display the data of a DataFrame row-wise and column-wise using iterrows() and iteritems()	22
16	Creating a Python program to perform writing and reading operations in a CSV file	25
Programs using Matplotlib – Data Visualization		
17	Creating a python program for plotting a line chart	26
18	Creating a python program for plotting a bar chart from a CSV file	27
19	Creating a python program for plotting a multiple bar chart from a CSV file	28
20	Creating a python program for plotting Histogram	29
SQL Queries		
21	SQL COMMANDS EXERCISE – 1 (Basic Queries – I)	30
22	SQL COMMANDS EXERCISE – 2 (Basic Queries – II)	32
23	SQL COMMANDS EXERCISE – 3 (Aggregate functions, Order by Group by, Having Clause)	33
24	SQL COMMANDS EXERCISE – 4 (Math Functions)	34
25	SQL COMMANDS EXERCISE – 5 (Text Functions)	35
26	SQL COMMANDS EXERCISE – 6 (Date Functions)	36

1) To write a Python program to create a Series to store 5 students Percentage Using dictionary and print all the elements that are above 75 percentage.

Source Code:

```
import pandas as pd

dic={1:'Apple',2:'Google',3:'Microsoft',4:'Meta',5:'Nvidia',6:'Amazon'}
Ser=pd.Series(dic)

print(Ser)
```

Output:

```
1      Apple
2     Google
3  Microsoft
4       Meta
5     Nvidia
6     Amazon
dtype: object
```

2) To write a Python program to create a Series object that stores the Initial budget allocated (50000/- each) for the four quarters of the year: Qtr1, Qtr2, Qtr3 and Qtr4.

Source Code:

```
import pandas as pd  
  
Ser=pd.Series(50000,index=['Qtr1','Qtr2','Qtr3','Qtr4'])  
  
print(Ser)
```

Output:

```
Qtr1    50000  
Qtr2    50000  
Qtr3    50000  
Qtr4    50000  
dtype: int64
```

3) To write a Python program to create a Series object that stores the Employee names as index and their Salary as values.

Source Code:

```
import pandas as pd

Emp_name=['Amit','Raghav','Ramesh','Suresh']
Emp_salary=[50000,65000,45500,60000]
Ser=pd.Series(Emp_salary,index=Emp_name)

print(Ser)|
```

Output:

```
Amit      50000
Raghav    65000
Ramesh    45500
Suresh    60000
dtype: int64
```

4) To Write a Python program to create a Series object with Employee names as the index and their salaries as values. Accept the name of the employee whose salary needs to be changed, along with the new salary, and update it in the Series.

Source Code:

```
import pandas as pd

Emp_name=['Amit', 'Raghav', 'Ramesh', 'Suresh']
Emp_salary=[50000, 65000, 45500, 60000]
Ser=pd.Series(Emp_salary, index=Emp_name)

print(Ser)

name=input('\nEnter the name of employee whose salary needs to change: ')
if name in Ser:
    salary=int(input('Enter the salary: '))
    Ser[name]=salary
    print()
    print('Employee Salary updated successfully!!!!')
    print(Ser)
else:
    print('Employee name not found!')
```

Output:

```
Amit      50000
Raghav    65000
Ramesh    45500
Suresh    60000
dtype: int64

Enter the name of employee whose salary needs to change: Amit
Enter the salary: 70000

Employee Salary updated successfully!!!!
Amit      70000
Raghav    65000
Ramesh    45500
Suresh    60000
dtype: int64
```

```
Amit      50000
Raghav    65000
Ramesh    45500
Suresh    60000
dtype: int64
```

```
Enter the name of employee whose salary needs to change: Disha
Employee name not found!
```

5) To create two series S1 and S2, to perform following mathematical Operations on Two Series objects:

(i) Addition (ii) Subtraction (iii) Multiplication (iv) Division Also show the result.

Source Code:

```
import pandas as pd

ind=['A','B','C','D']
data=[56,65,45.5,60]
S1=pd.Series(data,index=ind)

ind=['A','B','C','D']
data=[76,97,35,98.7]
S2=pd.Series(data,index=ind)

print('ADDITION of two Series')
print(S1+S2)

print('\nSUBTRACTION of two Series')
print(S1-S2)

print('\nMULTIPLICATION of two Series')
print(S1*S2)

print('\nDIVISION of two Series')
print(S1/S2)
```


Output:

```
ADDITION of two Series
A      132.0
B      162.0
C       80.5
D      158.7
dtype: float64

SUBTRACTION of two Series
A      -20.0
B      -32.0
C       10.5
D      -38.7
dtype: float64

MULTIPLICATION of two Series
A      4256.0
B      6305.0
C      1592.5
D      5922.0
dtype: float64

DIVISION of two Series
A       0.736842
B       0.670103
C       1.300000
D       0.607903
dtype: float64
```

6) To write a Python program to create two series i.e. population and average income of Four Zones, and then calculate per capita income storing in third pandas series print it.

Source Code:

```
import pandas as pd

zones=['North','South','East','West']
population=[23498,78659,23412,34098]
avg_salary=[2364000,1674000,1290000,4506900]

population_sr=pd.Series(population,index=zones)
avgsalary_sr=pd.Series(avg_salary,index=zones)

per_capita_income=avgsalary_sr/population

print("Per capita Income")
print()
print(per_capita_income)
```

Output:

```
Per capita Income

North      9.958294
South      2.123088
East        5.125577
West       13.399613
dtype: float64
```

7) To write a Python program to create a Series using list and display the following attributes of the Series: (i) index (ii) dtype (iii) size (iv) shape (v) hasnans

Source Code:

```
import pandas as pd

ls=[34,89,76,56,90]
Sr=pd.Series(ls)

print('Index of Series')
print(Sr.index)

print('\ndtype of Series')
print(Sr.dtype)

print('\nSize of Series')
print(Sr.size)

print('\nShape of Series')
print(Sr.shape)

print('\nSeries has nans values')
print(Sr.hasnans)|
```

Output:

```
Index of Series
RangeIndex(start=0, stop=5, step=1)

dtype of Series
int64

Size of Series
5

Shape of Series
(5,)

Series has nans values
False
```

8) To write a Python program to create a Series using list of Marks of 10 students and display first 5 Students' marks and Last 2 Students' marks from Series object.

Source Code:

```
import pandas as pd

marks=[76,89,79,67,90,79,73,99,93,78]
ind=['Student'+str(i) for i in range(1,11)]

Ser=pd.Series(marks,index=ind)

print('First 5 students')
print(Ser.head())

print('\nLast 2 students')
print(Ser.tail(2))
```

Output:

```
First 5 students
Student1      76
Student2      89
Student3      79
Student4      67
Student5      90
dtype: int64

Last 2 students
Student9       93
Student10      78
dtype: int64
```

9) To write a Python program to create a panda's Data Frame for the following table Using Nested list:

	Name	Age
Stu1	Arun	21
Stu2	Bala	23
Stu3	Mukesh	22

Source Code:

```
import pandas as pd

data=[['Arun',21],['Bala',23],['Mukesh',22]]
ind=['Stu1','Stu2','Stu3']
columns=['Name','Age']

Df=pd.DataFrame(data,columns=columns,index=ind)

print(Df)|
```

Output:

```
      Name  Age
Stu1  Arun   21
Stu2  Bala   23
Stu3  Mukesh  22
```

10) To write a Python program to create a panda's DataFrame called DF for the following table Using Dictionary of List and perform the following operations:

	Toys	Books	Uniform	Shoes
AP	7916	61896	610	8810
OD	8508	8208	508	6798
MP	7226	6149	611	9611
UP	7617	6157	457	6457

- (i) To Display only column 'Toys' from DataFrame DF.
- (ii) To Display the row details of 'AP' and 'OD' from DataFrame DF.
- (iii) To Display the column 'Books' and 'Uniform' for 'M.P' and 'U.P' from DataFrame DF.
- (iv) To Display consecutive 3 rows and 3 columns from DataFrame DF.

Source Code:

```
import pandas as pd

data={'Toys':[7916,8508,7226,7617],
      'Books':[61896,8208,6149,6157],
      'Uniform':[610,508,611,457],
      'Shoes':[8810,6798,9611,6457]}
ind=['AP','OD','MP','UP']
Df=pd.DataFrame(data,index=ind)

print("Display only column 'Toys' from DataFrame DF")
print(Df['Toys'])

print("\nDisplay the row details of 'AP' and 'OD' from DataFrame DF")
print(Df.loc['AP':'OD'])

print("\nDisplay the column 'Books' and 'Uniform' for 'M.P' and 'U.P' from DataFrame DF")
print(Df.loc['MP':'UP', 'Books':'Uniform'])

print("\nDisplay consecutive 3 rows and 3 columns from DataFrame DF")
print(Df.iloc[0:3,0:3])
```

Output:

```
Display only column 'Toys' from DataFrame DF
```

```
AP      7916
```

```
OD      8508
```

```
MP      7226
```

```
UP      7617
```

```
Name: Toys, dtype: int64
```

```
Display the row details of 'AP' and 'OD' from DataFrame DF
```

```
      Toys  Books  Uniform  Shoes
```

```
AP   7916  61896      610   8810
```

```
OD   8508   8208      508   6798
```

```
Display the column 'Books' and 'Uniform' for 'M.P' and 'U.P' from DataFrame DF
```

```
      Books  Uniform
```

```
MP    6149      611
```

```
UP    6157      457
```

```
Display consecutive 3 rows and 3 columns from DataFrame DF
```

```
      Toys  Books  Uniform
```

```
AP   7916  61896      610
```

```
OD   8508   8208      508
```

```
MP   7226   6149      611
```

11) To write a Python program to create a panda's DataFrame called DF for the following table Using Dictionary of List and perform the following operations:

	Toys	Books	Uniform	Shoes
AP	7916	61896	610	8810
OD	8508	8208	508	6798
MP	7226	6149	611	9611
UP	7617	6157	457	6457

(i) Insert a new column "Bags" with values as [5891, 8628, 9785, 4475].

(ii) Delete the row details of M.P from DataFrame DF.

Source Code:

```
import pandas as pd

data={'Toys':[7916,8508,7226,7617],'Books':[61896,8208,6149,6157],
      'Uniform':[610,508,611,457],'Shoes':[8810,6798,9611,6457]}
ind=['AP','OD','MP','UP']
Df=pd.DataFrame(data,index=ind)

print("Inserting a new column 'Bags'")
Df['Bags']=[5891, 8628, 9785, 4475]
print(Df)

print('\nDeleting the row details of M.P')
Df=Df.drop('MP',axis=0)
print(Df)
```

Output:

```
Inserting a new column 'Bags'
   Toys  Books  Uniform  Shoes  Bags
AP  7916  61896     610   8810  5891
OD  8508   8208     508   6798  8628
MP  7226   6149     611   9611  9785
UP  7617   6157     457   6457  4475

Deleting the row details of M.P
   Toys  Books  Uniform  Shoes  Bags
AP  7916  61896     610   8810  5891
OD  8508   8208     508   6798  8628
UP  7617   6157     457   6457  4475
```


12) To write a Python program to create a pandas DataFrame to analyze number of Government and Private medical college and their Total seats, Fees statewide details using the dataset available at www.data.gov.in. Also, perform the following operations.

- (i) To Change the name of the state AP to Andhra.
- (ii) To Count and Display Non-NaN values of each column.
- (iii) To Count and Display Non-NaN values of each row.
- (iv) To Increase the fees of all colleges by 5%
- (v) To Replace all NaN values with 0.

Source Code:

```
import pandas as pd
import numpy as np

S1 = pd.Series([17, 30, 7150, 25000], index=['GC', 'PC', 'TS', 'Fees']) # AP
S2 = pd.Series([6, np.NaN, 726, 33000], index=['GC', 'PC', 'TS', 'Fees']) # Assam
S3 = pd.Series([np.NaN, 1, 100, 27000], index=['GC', 'PC', 'TS', 'Fees']) # Sikkim
S4 = pd.Series([22, 24, 46, 23000], index=['GC', 'PC', 'TS', 'Fees']) # TN
S5 = pd.Series([7, np.NaN, 673, 15000], index=['GC', 'PC', 'TS', 'Fees']) # AIIMS

df = pd.DataFrame({'AP': S1, 'Assam': S2, 'Sikkim': S3, 'TN': S4, 'AIIMS': S5})

print('original Dataframe')
print(df)

print('\nColumn Name of AP changed to Andhra')
df=df.rename(columns={'AP':'Andhra'})
print(df)

print('\nCount non-Nan values in each column')
print(df.count())

print('\nCount non-Nan values in each row')
print(df.count(1))

print('\nIncreasing fees by 5%')
df.loc['Fees']+=df.loc['Fees']*5*10**-2
print(df)

print('\nFilling NaN values by 0')
df=df.fillna(0)
print(df)
```

Output:

original Dataframe

	AP	Assam	Sikkim	TN	AIIMS
GC	17	6.0	NaN	22	7.0
PC	30	NaN	1.0	24	NaN
TS	7150	726.0	100.0	46	673.0
Fees	25000	33000.0	27000.0	23000	15000.0

Column Name of AP changed to Andhra

	Andhra	Assam	Sikkim	TN	AIIMS
GC	17	6.0	NaN	22	7.0
PC	30	NaN	1.0	24	NaN
TS	7150	726.0	100.0	46	673.0
Fees	25000	33000.0	27000.0	23000	15000.0

Count non-Nan values in each column

```
Andhra      4
Assam       3
Sikkim      3
TN           4
AIIMS       3
dtype: int64
```

Count non-Nan values in each row

```
GC          4
PC          3
TS          5
Fees        5
dtype: int64
```

Increasing fees by 5%

	Andhra	Assam	Sikkim	TN	AIIMS
GC	17	6.0	NaN	22	7.0
PC	30	NaN	1.0	24	NaN
TS	7150	726.0	100.0	46	673.0
Fees	26250	34650.0	28350.0	24150	15750.0

Filling NaN values by 0

	Andhra	Assam	Sikkim	TN	AIIMS
GC	17	6.0	0.0	22	7.0
PC	30	0.0	1.0	24	0.0
TS	7150	726.0	100.0	46	673.0
Fees	26250	34650.0	28350.0	24150	15750.0

13) To write a Python program to create a panda's DataFrame called DF for the following table using Dictionary of List and display the details of students whose Percentage is more than 85.

	Stu_Name	Degree	Percentage
S1	Anu	MBA	90
S2	Arun	MCA	85
S3	Bala	ME	91
S4	Charan	M.Sc.	76
S5	Mannu	MCA	84

Source Code:

```
import pandas as pd

data={'Stu_Name':['Anu','Arun','Bala','Charan','Mannu'],
      'Degree':['MBA','MCA','ME','M.Sc','MCA'],
      'Percentage':[90,85,91,76,84]}
Df=pd.DataFrame(data,index=['S1','S2','S3','S4','S5'])

print(Df[Df['Percentage']>85])
```

Output:

```
Stu_Name Degree Percentage
S1      Anu    MBA         90
S3     Bala    ME         91
```

14) To write a Python program to create a DataFrame using Dictionary of list and display the following attributes of the DataFrame: (i) index (ii) columns (iii) axes (iv) dtypes (v) shape (vi) dimension (vii) T

Source Code:

```
import pandas as pd

data={ 'Name': ['Anu', 'Arun', 'Bala', 'Charan', 'Mannu'],
       'Degree': ['MBA', 'MCA', 'ME', 'M.Sc', 'MCA'],
       'Percentage': [90, 85, 91, 76, 84] }
Df=pd.DataFrame(data,index=['S1', 'S2', 'S3', 'S4', 'S5'])

print('DataFrame')
print(Df)

print('\nIndex')
print(Df.index)

print('\nColumns')
print(Df.columns)

print('\nAxes')
print(Df.axes)

print('\ndtype')
print(Df.dtypes)

print('\nShape')
print(Df.shape)

print('\nDimension')
print(Df.ndim)

print('\nTranspose')
print(Df.T)
```

Output:

```
DataFrame
   Name Degree Percentage
S1   Anu   MBA         90
S2  Arun   MCA         85
S3   Bala   ME          91
S4 Charan  M.Sc         76
S5  Mannu   MCA         84

Index
Index(['S1', 'S2', 'S3', 'S4', 'S5'], dtype='object')

Columns
Index(['Name', 'Degree', 'Percentage'], dtype='object')

Axes
[Index(['S1', 'S2', 'S3', 'S4', 'S5'], dtype='object'), Index(['Name', 'Degree', 'Percentage'], dtype='object')]

dtype
Name          object
Degree         object
Percentage     int64
dtype: object

Shape
(5, 3)

Dimension
2

Transpose
   S1  S2  S3  S4  S5
Name  Anu  Arun  Bala  Charan  Mannu
Degree  MBA  MCA  ME  M.Sc  MCA
Percentage  90  85  91  76  84
```

15) To write a Python program to create a panda's DataFrame called Students for the following table and demonstrate iterrows and iteritems.

	Stu_Name	Degree	Percentage
S1	Anu	MBA	90
S2	Arun	MCA	85
S3	Bala	ME	91
S4	Charan	M.Sc.	76
S5	Mannu	MCA	84

Source Code:

iteritems:

```
import pandas as pd

index = ["s" + str(i) for i in range(1, 6)]
data = {
    'Stu_Name': ['Anu', 'Arun', 'Bala', 'Charan', 'Mannu'],
    'Degree': ['MBA', 'MCA', 'ME', 'M.Sc.', 'MCA'],
    'Percentage': [90, 85, 91, 76, 84]
}

df = pd.DataFrame(data, index=index)

for column_name, series in df.items():
    print(f"Column: {column_name}")
    print(series)
    print()
```

iterrows:

```
import pandas as pd

index = ["s" + str(i) for i in range(1, 6)]
data = {
    'Stu_Name': ['Anu', 'Arun', 'Bala', 'Charan', 'Mannu'],
    'Degree': ['MBA', 'MCA', 'ME', 'M.Sc.', 'MCA'],
    'Percentage': [90, 85, 91, 76, 84]
}

df = pd.DataFrame(data, index=index)

for index, row in df.iterrows():
    print(f"Index: {index}")
    print(row)
    print()
```

Output:

iteritems:

```
Column: Stu_Name
s1      Anu
s2      Arun
s3      Bala
s4      Charan
s5      Mannu
Name: Stu_Name, dtype: object

Column: Degree
s1      MBA
s2      MCA
s3      ME
s4      M.Sc.
s5      MCA
Name: Degree, dtype: object

Column: Percentage
s1      90
s2      85
s3      91
s4      76
s5      84
Name: Percentage, dtype: int64
```

iterrows:

```
Index: s1  
Stu_Name      Anu  
Degree        MBA  
Percentage     90  
Name: s1, dtype: object
```

```
Index: s2  
Stu_Name      Arun  
Degree        MCA  
Percentage     85  
Name: s2, dtype: object
```

```
Index: s3  
Stu_Name      Bala  
Degree        ME  
Percentage     91  
Name: s3, dtype: object
```

```
Index: s4  
Stu_Name      Charan  
Degree        M.Sc.  
Percentage     76  
Name: s4, dtype: object
```

```
Index: s5  
Stu_Name      Mannu  
Degree        MCA  
Percentage     84  
Name: s5, dtype: object
```


16) To Write a Python program to store the details of Employess' such as Empno, Name, Salary into a Employee.csv file. Also, write a code to read employee details from csv file.

Source Code:

```
import pandas as pd

Emp={'Empno':[1001,1002,1003,1004],
     'Emp_name':['Amit','Rahul','Vansh','Varun'],
     'Salary':[56000,98000,56000,50000]}
df=pd.DataFrame(Emp)

df.to_csv('Employee.csv',index=False)

DF=pd.read_csv('Employee.csv')
print(DF)
```

Output:

	Empno	Emp_name	Salary
0	1001	Amit	56000
1	1002	Rahul	98000
2	1003	Vansh	56000
3	1004	Varun	50000

17) To write a Python program to plot a Line chart to depict the changing weekly Onion and Brinjal prices for four weeks. Also, give appropriate axes labels, title and keep marker style as Diamond and marker edge color as 'red' for Onion

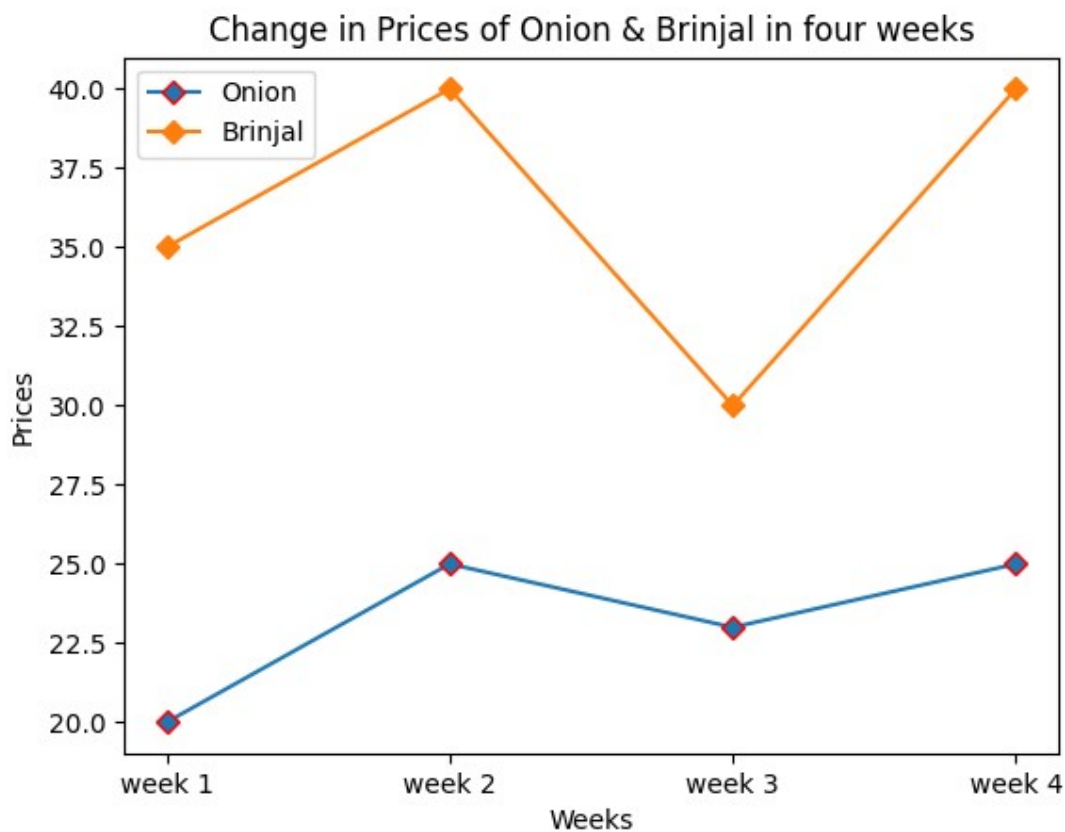
Source Code:

```
import matplotlib.pyplot as plt

weeks=['week 1','week 2','week 3','week 4']
onion_price=[20,25,23,25]
brinjal_price=[35,40,30,40]

plt.plot(weeks,onion_price,label='Onion',marker='D',markeredgecolor='red')
plt.plot(weeks,brinjal_price,label='Brinjal',marker='D')
plt.xlabel('Weeks')
plt.ylabel('Prices')
plt.title('Change in Prices of Onion & Brinjal in four weeks')
plt.legend()
plt.show()
```

Output:



18) To write a Python program to create a DataFrame for subject-wise average, save it to a CSV file, and then draw a bar chart using Matplotlib with a width of each bar as 0.25, specifying different colors for each bar. Additionally, provide a proper title and axes labels for the bar chart.

Source Code:

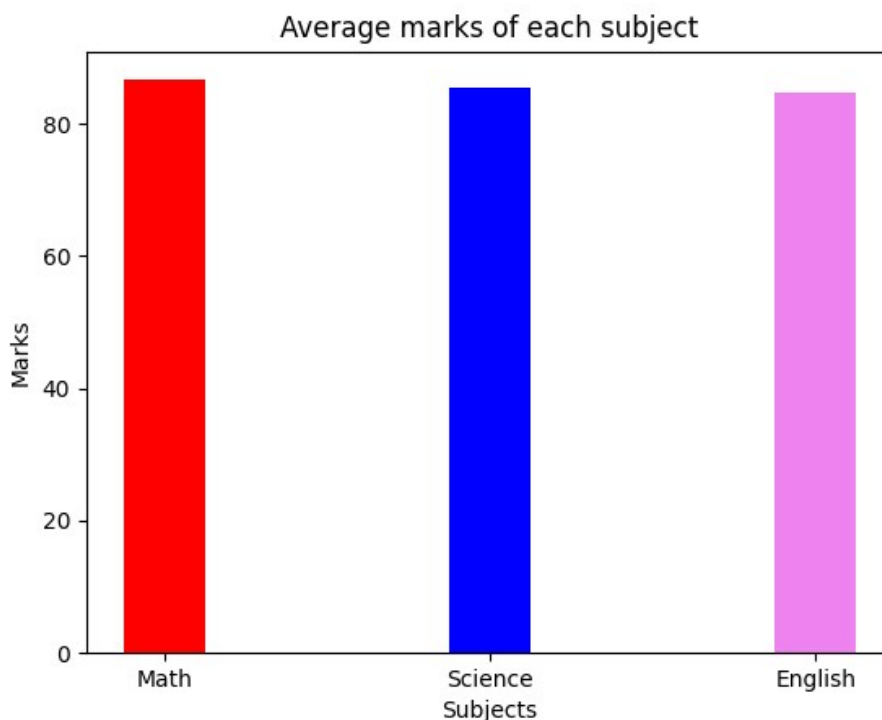
```
import matplotlib.pyplot as plt
import pandas as pd

data = {
    'Student': ['Anu', 'Arun', 'Bala', 'Charan', 'Mannu'],
    'Math': [85, 90, 78, 88, 92],
    'Science': [88, 84, 91, 79, 85],
    'English': [82, 87, 90, 84, 80]
}
df=pd.DataFrame(data)

df_avg=df[['Math','Science','English']].mean()
df_avg.to_csv('subject_averages.csv')

plt.bar(df_avg.index,df_avg.values,width=0.25,color=['red','blue','violet'])
plt.title('Average marks of each subject')
plt.xlabel('Subjects')
plt.ylabel('Marks')
plt.show()
```

Output:



19) To write a Python program to plot a multiple bar chart From CSV file using Matplotlib for subject wise Scores of Class A, Class B, and Class C. Different colors represent each class, and subjects include English, Accountancy, Economics, BST and IP. Proper labels, a title and a legend are displayed on the chart.

Source Code:

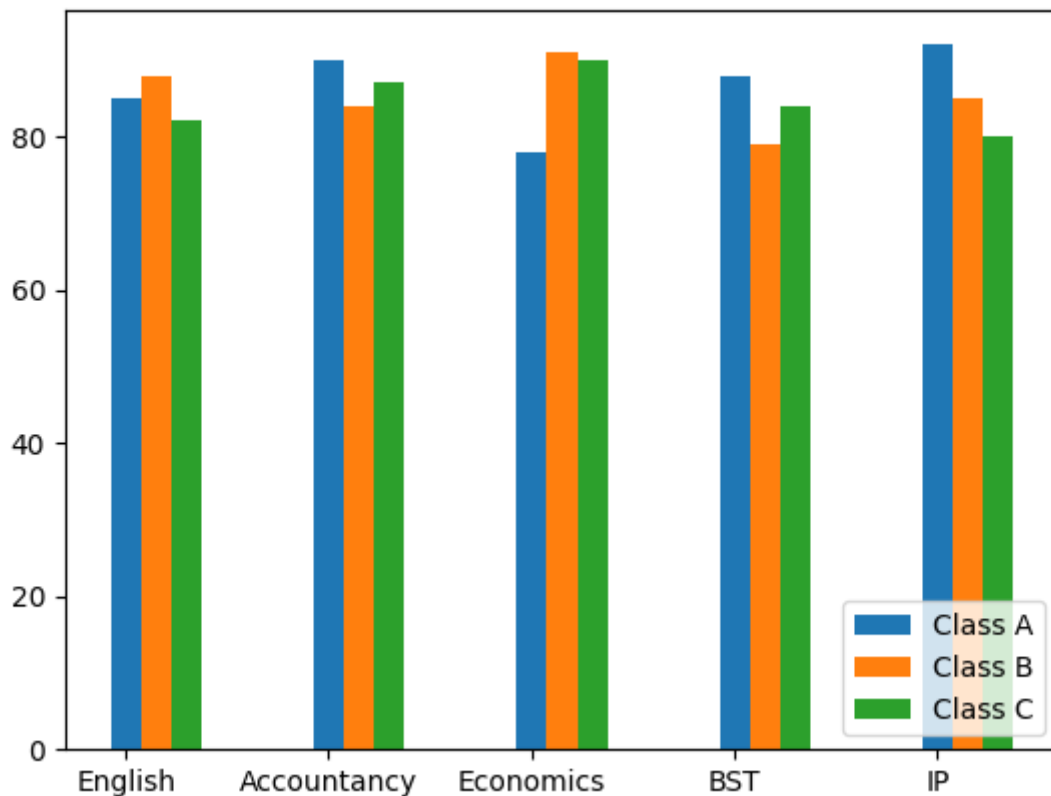
```
import matplotlib.pyplot as plt
import pandas as pd

scores=pd.read_csv('subject_averages.csv')
print(scores)

w=0.15

plt.bar(scores['Subjects'],scores['Class A'],width=w,label='Class A')
plt.bar([x + w for x in range(len(scores))],scores['Class B'],width=w,label='Class B')
plt.bar([x + 2*w for x in range(len(scores))],scores['Class C'],width=w,label='Class C')
plt.legend(loc='lower right')
plt.show()
```

Output:



20) To write a Python program to plot a Histogram for the following class interval or range. Also, give appropriate axes name, title and edge color as 'red'.

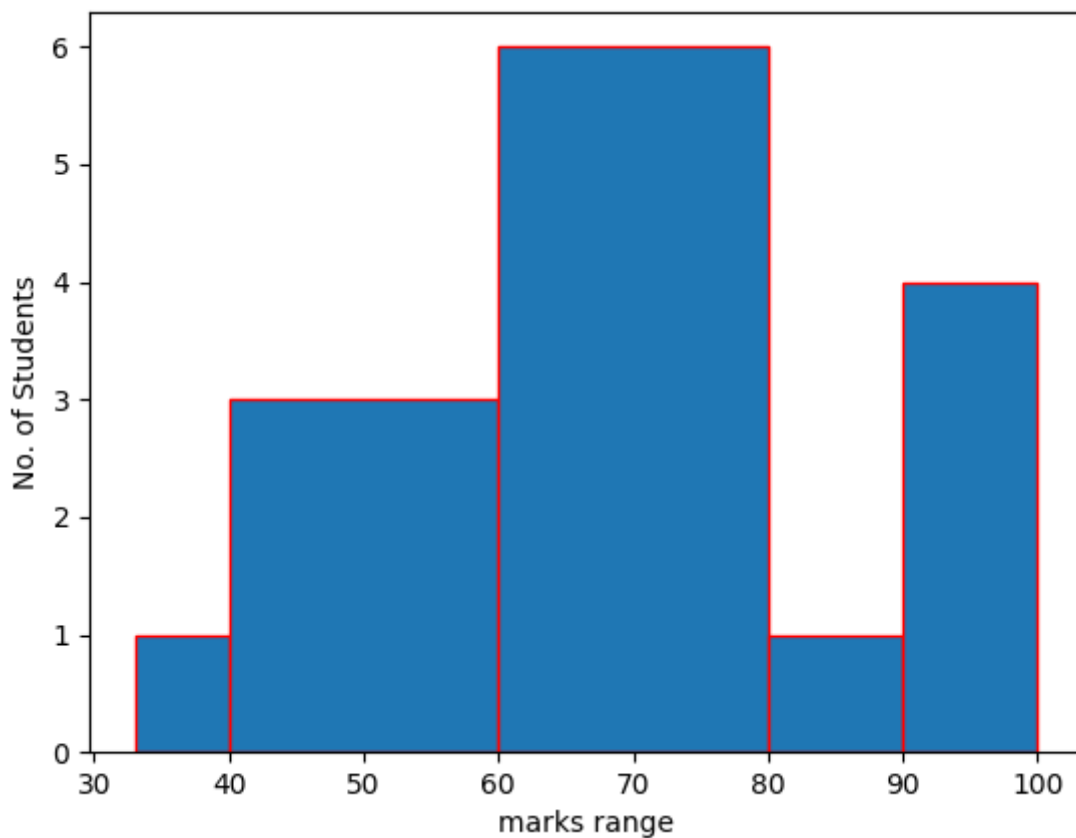
Source Code:

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

marks=[78,67,98,67,56,34,45,67,76,78,98,54,87,23,11,90,95]

plt.hist(marks,bins=[33,40,60,80,90,100],edgecolor='red')
plt.xlabel('marks range')
plt.ylabel('No. of Students')
plt.show()
```

Output:



SQL COMMANDS EXERCISE – 1

21) To write Queries for the following Questions based on the given table:

EmpID	Name	Gender	Age	Dept	DOJ	Salary	City
1	Praveen	M	25	Sales	1989-06-08	20000	Chennai
2	Arun	M	29	Marketing	1989-09-26	22000	Chennai
3	Usha	F	27	Finance	1994-08-09	25000	Bangalore
4	Bala	M	31	Sales	1990-03-23	27000	NULL
5	Rani	F	28	Marketing	1990-04-23	27000	Mumbai
6	Nisha	F	26	NULL	1991-02-24	18000	Bangalore
7	Manoj	M	32	Finance	1982-05-06	30000	Goa

- Write a Query to Create a new database in the name of "EMPS".
- Write a Query to Open the database EMPS.
- Write a Query to create the above table called: Info
- Write a Query to list all the existing database names
- Write a Query to List all the tables that exists in the current database.
- Write a Query to insert all the rows of above table into Info table.
- Write a Query to display all the details of the Employees from the above table 'INFO'.

Queries:

- CREATE DATABASE EMPS;
- use EMPS;
- CREATE TABLE Info
(EmpID INT(8) PRIMARY KEY,
Name varchar(20) NOT NULL,
Gender char(1),
Dept varchar(20),
DOJ Date,
Salary DECIMAL(10, 2) ,
City varchar(10));
- SHOW DATABASES;
- SHOW TABLES;
- Insert into Info(EmpID,Name,Gender,Dept,DOJ,Salary,City)
Values
(1, 'Praveen', 'M', 25, 'Sales', '1989-06-08', 20000, 'Chennai'), (

(2, 'Arun', 'M', 29, 'Marketing', '1989-09-26', 22000, 'Chennai'),
(3, 'Usha', 'F', 27, 'Finance', '1994-08-09', 25000, 'Bangalore'),
(4, 'Bala', 'M', 31, 'Sales', '1990-03-23', 27000, NULL),
(5, 'Rani', 'F', 28, 'Marketing', '1990-04-23', 27000, 'Mumbai'),
(6, 'Nisha', 'F', 26, NULL, '1991-02-24', 18000, 'Bangalore'),
(7, 'Manoj', 'M', 32, 'Finance', '1982-05-06', 30000, 'Goa');

g) SELECT * from Info;

SQL COMMANDS EXERCISE – 2

22) To write Queries for the following Questions based on the given table:

EmpID	Name	Gender	Age	Dept	DOJ	Salary	City
1	Praveen	M	25	Sales	1989-06-08	20000	Chennai
2	Arun	M	29	Marketing	1989-09-26	22000	Chennai
3	Usha	F	27	Finance	1994-08-09	25000	Bangalore
4	Bala	M	31	Sales	1990-03-23	27000	NULL
5	Rani	F	28	Marketing	1990-04-23	27000	Mumbai
6	Nisha	F	26	NULL	1991-02-24	18000	Bangalore
7	Manoj	M	32	Finance	1982-05-06	30000	Goa

- Write a Query to Display Employees' name and City from the above table.
- Write a Query to Display all details of Employees who are living in Chennai.
- Write a Query to get the name and salary of the employee whose salary is above 15000 and gender is not male.
- Write a query to update increase 10% Salary of an employee whose City is 'CHENNAI' and Gender is 'MALE'.
- Write a Query to delete the details of Employee Id 6.

- SELECT Name, City FROM Info;
- SELECT * FROM Info WHERE City = 'Chennai';
- SELECT Name, Salary FROM Info WHERE Salary > 15000 AND Gender != 'M';
- UPDATE Info
SET Salary = Salary * 1.10
WHERE City = 'Chennai' AND Gender = 'M';
- DELETE FROM Info WHERE EmpID = 6;

SQL COMMANDS EXERCISE – 3

(Aggregate Functions, Order By, Group By, Having Clause)

23) To write Queries for the following Questions based on the given table:

EmpID	Name	Gender	Age	Dept	DOJ	Salary	City
1	Praveen	M	25	Sales	1989-06-08	20000	Chennai
2	Arun	M	29	Marketing	1989-09-26	22000	Chennai
3	Usha	F	27	Finance	1994-08-09	25000	Bangalore
4	Bala	M	31	Sales	1990-03-23	27000	NULL
5	Rani	F	28	Marketing	1990-04-23	27000	Mumbai
6	Nisha	F	26	NULL	1991-02-24	18000	Bangalore
7	Manoj	M	32	Finance	1982-05-06	30000	Goa

- Write a Query to list names of Employees in Descending order.
- Write a Query to find a total salary of all employees.
- Write a Query to display maximum salary and minimum salary of employees.
- Write a Query to count the number of employees earning more than 25000.
- Write a query to display sum of salary of the employees grouped by department wise.
- Write a query to display the department names where number of employees are greater than or equal to 2.

- SELECT Name FROM Info
ORDER BY Name DESC;
- SELECT SUM(Salary) AS Total_Salary FROM Info;
- SELECT MAX(Salary) AS Max_Salary, MIN(Salary) AS Min_Salary FROM Info;
- SELECT COUNT(*) AS Employees_Above_25000 FROM Info WHERE Salary > 25000;
- SELECT Dept, SUM(Salary) AS Total_Salary FROM Info
GROUP BY Dept;
- SELECT Dept FROM Info
GROUP BY Dept
HAVING COUNT(*) >= 2;

COMMANDS EXERCISE – 4

(Mathematical Functions)

24) To write Queries for the following Questions based on the given table:

Rollno	Name	Gender	AGE	DEPT	DOA	Percentage
1	Arun	M	16	CS	1997-01-10	97.8
2	ANKIT	M	17	IP	1998-06-24	95.2
3	ANU	F	15	HINDI	1996-07-12	81
4	BALA	F	17	IP	1999-07-01	75.5
5	CHARAN	M	16	HINDI	1997-06-27	92.13
6	DEEPA	F	16	HISTORY	1997-06-27	60.54
7	DINESH	M	15	CS	1996-08-25	82

- (a) Write a Query to Display square of age that got admission in the month of August.
- (b) Write a Query to display Remainder of column Percentage divide by 3.
- (c) Write a Query to display Student names and their Percentage in round figure.
- (d) Display Name, Percentage and round up the remainder marks up to 2 decimal places.

- a) `SELECT Name, AGE * AGE AS Age_Square FROM Students`
`WHERE MONTH(DOA) = 8;`
- b) `SELECT Name, Percentage % 3 AS Remainder FROM Students;`
- c) `SELECT Name, ROUND(Percentage) AS Rounded_Percentage FROM Students;`
- d) `SELECT Name, Percentage, ROUND(Percentage % 3, 2) AS Remainder_Rounded`
`FROM Students;`

SQL COMMANDS EXERCISE – 5

(Text Functions)

25) To write Queries for the following Questions based on the given table:

Rollno	Name	Gender	AGE	DEPT	DOA	Percentage
1	Arun	M	16	CS	1997-01-10	97.8
2	ANKIT	M	17	IP	1998-06-24	95.2
3	ANU	F	15	HINDI	1996-07-12	81
4	BALA	F	17	IP	1999-07-01	75.5
5	CHARAN	M	16	HINDI	1997-06-27	92.13
6	DEEPA	F	16	HISTORY	1997-06-27	60.54
7	DINESH	M	15	CS	1996-08-25	82

- (a) Write a Query to display Department name in lower case letters.
- (b) Write a Query to display department name and its respective number of characters in Dept column.
- (c) Write a Query to display first 2 characters of the column Name.
- (d) Write a Query to display first 2 characters of the column Name.
- (e) Write a query to display the names of all students and extract five characters from the third position of the 'Name' field.

- a) `SELECT LOWER(DEPT) AS Department_Lower_Case FROM Students;`
- b) `SELECT DEPT, LENGTH(DEPT) AS Dept_Character_Count FROM Students;`
- c) `SELECT SUBSTRING(Name, 1, 2) AS First_Two_Characters FROM Students;`
- d) `SELECT SUBSTRING(Name, 1, 2) AS First_Two_Characters FROM Students;`
- e) `SELECT Name, SUBSTRING(Name, 3, 5) AS Extracted_Characters FROM Students;`

SQL COMMANDS EXERCISE – 6

(Date Functions)

26) To write Queries for the following Questions based on the given table:

Rollno	Name	Gender	AGE	DEPT	DOA	Percentage
1	Arun	M	16	CS	1997-01-10	97.8
2	ANKIT	M	17	IP	1998-06-24	95.2
3	ANU	F	15	HINDI	1996-07-12	81
4	BALA	F	17	IP	1999-07-01	75.5
5	CHARAN	M	16	HINDI	1997-06-27	92.13
6	DEEPA	F	16	HISTORY	1997-06-27	60.54
7	DINESH	M	15	CS	1996-08-25	82

- (a) Write a Query to display student name and month of date of admission of all students.
- (b) Write a Query to display Student name and day name of the students' DOA of the table STU.
- (c) Write a query to display the joining year of IP students.
- (d) Write a Query to Display the month for the date_of_birth of all students.
- (e) Write a query to display the names of the students who joined in the month of June.

- a) `SELECT Name, MONTH(DOA) AS Admission_Month FROM Students;`
- b) `SELECT Name, DAYNAME(DOA) AS Day_Of_Admission FROM Students;`
- c) `SELECT Name, YEAR(DOA) AS Joining_Year FROM Students WHERE DEPT = 'IP';`
- d) `SELECT Name, MONTH(DOA) AS Birth_Month FROM Students;`
- e) `SELECT Name FROM Students WHERE MONTH(DOA) = 6;`