

INDEX

No.	Title	Date	Remark
	Python		
1.	Create a python series using the python sequence with 5 elements		
2.	Create a Series object to store all vowels individually. Its index should be 1,2,3,4 & 5		
3.	Create a Series object using ndarray that has 5 elements in the range 50 and 100		
4.	Create a Series object using dictionary to that stores the no of students in each section of class 12th of your school		
5.	Create a Series object 'Item' that stores rate of each product as given below: Soap 54 Salt 20 Sugar 39 Write code to modify rate of soap to 44 and sugar to 42. print the changed rate		
6.	Create a Series object 'Item' that stores rate of each product as given below: Soap 54 Salt 20 Sugar 39 Write code to modify rate of soap to 44 and sugar to 42. print the changed rate		

No.	Title	Date	Remark
7.	No of students in class 11 and class 12 in three streams (science, commerce and humanities) are stored in 2 series object class 11 and class 12. write code to find total no of students in class 11 & class 12 stream wise		
8.	Create a Series object 'population' to store population of 5 different metro cities and display the population that are more than 300000		
9.	Create a series 'temp' that stores temperature of seven days in it. Its index should be 'Sunday', 'Monday' Write script to 1. Display temp of first 3 days. 2. Display temp of last 3 days. 3. Display all temp in reverse order like Saturday, Friday,.... 4. Display temp from Tuesday to Friday. 5. Display square of all temperature		
10.	Create a Series object 'employee' that stores salary of 7 employees. Write script to print 1. Total no of elements 2. Series is empty or not 3. Series consist NaN value or not 4. Count Non-NA elements 5. Axis labels		
11.	Create a pandas series from an ndarray		
12.	Given a Series, print all the elements that are above the 75th percentile		

No.	Title	Date	Remark
13.	Create the following dataframe 'Sport' containing sport wise marks for five students. Use 2D dictionary to create dataframe		
14.	Create a dataframe from list containing dictionaries of most economical bike with its name and rate of three companies. Company name should be the row labels		
15.	Consider two series object staff and salaries that stores the number of people in various office branches and salaries distributed in these branches respectively. Write a program to create another Series object that stores average salary per branch and then create a dataframe object from these Series object. After creating dataframe rename all row labels with Branch name		
16.	Write program to do the followings 1. Display row labels of 'sales' 2. Display column label of 'sales' 3. Display last two rows of the 'sales' 4. Display first two rows of the 'sales'.		
17.	Create a dataframe 'sales2' using dictionary as given below and write a program to append 'sales2' to the dataframe 'sales' created in previous practical		
18.	Create a dataframe 'aid' as given below and write program to do following		

No.	Title	Date	Remark
19.	For given dataframe 'aid' write program to write the values of 'aid' to a comma separated file 'aidfigures.csv' on the disk. Do not write the row labels and column labels		
20.	Read the data in the file 'aidfigure.csv' into a dataframe 'aidretrieved' and display it. Now update the row labels and column labels of 'aidretrieved' to be the same as that of 'aid'		
21.	Create a dataframe based on ecommerce data and generate descriptive statistics (mean, median, mode, quartile, and variance)		
22.	Create a data frame for examination result and display row labels, column labels data types of each column and the dimensions		
23.	Filter out rows based on different criteria such as duplicate rows		
24.	Find the sum of each column, or find the column with the lowest mean		
25.	Locate the 3 largest values in a data frame		
26.	Subtract the mean of a row from each element of the row in a Data Frame		
27.	Replace all negative values in a data frame with a 0		
28.	Replace all missing values in a data frame with a 999		

No.	Title	Date	Remark
29.	Importing data between Pandas and MySQL		
30.	Exporting data between Pandas and MySQL		
31.	Given the school result data, analyse the performance of the students on different parameters, e.g subject wise or class wise		
32.	Collect and store total medals won by 10 countries in Olympic games and represent it in form of bar chart with title to compare and analyze data		
33.	.To get proper analysis of sale performance create multiple line chart on a common plot where all bike sale data are plotted. Display appropriate x and y axis labels, legend and chart title.		
34.	Given the school result data, analyses the performance of the student on different parameters, e.g. subject wise or class wise. Create a dataframe for the above, plot appropriate chart with title and legend		
35.	Construct a histogram from given data with 10 bin		
36.	Take data of your interest from an open source (e.g. data.gove.in), aggregate and summarize it. Then plot it using different plotting functions of the Matplotlib library		

No.	Title	Date	Remark
	STRUCTURED QUERY LANGUAGE (SQL)		
37.	Create Database Named Class12		
38.	Open Database Class 12		
39.	Create a student table with the student id, class, section, gender, name, dob, and marks as attributes where the student id is the primary key.		
40.	View the structure of the table		
41.	Insert the details of at least 10 students in the above table.		
42.	Display the details of the student table.		
43.	Delete record of students who secured less than 65 marks.		
44.	Increase marks by 5% for who have studentid more than 1105.		
45.	Display the content of the table of female students.		
46.	Display studentid, Name and Marks whose marks are more than 50.		

No.	Title	Date	Remark
47.	Find the average of marks from the student table		
48.	Find the number of students, who are from section 'A'		
49.	Add a new column email in the above table		
50.	Add the email ids of each student in the created email column		
51.	Display the information of all the students, name contains 'sh'		
52.	Display the information of all the students, name starts with 'sh'		
53.	Display studentid, Name, DOB of who are born in 2005		
54.	Display studentid and DOB of students in ascending order of their names		
55.	Display studentid, Gender, Name, DOB, Marks, Email in descending order of their marks		
56.	Display the unique section available in the table		
57.	Find the total number of customers from each country in the table (customer ID, customer Name, country) using group by.		
58.	For the given table 'Hospital' write SQL command to display name all patient admitted in month of May.		
59.	For the given table 'Hospital' write SQL command to Display patient name in upper case with year of admission.		

No.	Title	Date	Remark
60.	Find min,max and average of marks in the student table		
61.	Delete the details of a particular student from the table student table		
62.	Delete the details of the student table		
63.	Create a new table (name, date of birth) by joining two tables (student id, name) and (student id, date of birth)		
64.	Create a foreign key in one of the two table mentioned above		

Python

1. Create a python series using the python sequence with 5 elements

Input:

```
import pandas as pd
list=[10,20,30,40,50]
s=pd.Series(list)
print(s)
```

Output:

```
0    10
1    20
2    30
3    40
4    50
dtype: int64
```

2. Create a Series object to store all vowels individually. Its index should be 1,2,3,4 & 5

Input:

```
import pandas as pd
v=pd.Series( data: ['a','e','i','o','u'], index=[1,2,3,4,5])
print(v)
```

Output:

```
1    a
2    e
3    i
4    o
5    u
dtype: object
```

3. Create a Series object using ndarray that has 5 elements in the range 50 and 100

Input:

```
import pandas as pd
import numpy as np
s=pd.Series(np.arange(50,100,10))
print(s)
```

Output:

```
0    50
1    60
2    70
3    80
4    90
dtype: int32
```

4. Create a Series object using dictionary to that stores the no of students in each section of class 12th of your school

Input:

```
import pandas as pd
dict= {'A':23,'B':34,'C':36,'D':40,'E':32}
s=pd.Series(dict)
print(s)
```

Output:

```
A    23
B    34
C    36
D    40
E    32
dtype: int64
```

5.Total no of students to be admitted is 350 in Yojna School every year. Write code to create a Series object 'School' that stores these total no of students for the year 2015 to 2022

Input:

```
import pandas as pd
s=pd.Series( data: 350, range (2016, 2023))
print(s)
```

Output:

```
2016    350
2017    350
2018    350
2019    350
2020    350
2021    350
2022    350
dtype: int64
```

6.Create a Series object 'Item' that stores rate of each product as given below: Soap 54 Salt 20 Sugar 39 Write code to modify rate of soap to 44 and sugar to 42. print the changed rate

Input:

```
import pandas as pd
s=pd.Series( data: [54,20,39], index: ['soap','salt','sugar'])
print(s)
s['soap']=44
s['sugar']=42
print(s)
```

Output:

```
soap      54
salt      20
sugar     39
dtype: int64
soap      44
salt      20
sugar     42
dtype: int64
```

7. No of students in class 11 and class 12 in three streams (science, commerce and humanities) are stored in 2 series object class 11 and class 12. write code to find total no of students in class 11 & class 12 stream wise

Input:

```
import pandas as pd
dict1={'Science': 32,'Commerce': 36,'Humanities':20}
dict2={'Science':28,'Commerce':34,'Humanities':22}
Class11=pd.Series(dict1)
Class12=pd.Series(dict2)
print(Class11)
print(Class12)
print('Total Students')
print (Class11 + Class12)
```

Output:

```
Science      32
Commerce     36
Humanities   20
dtype: int64
Science      28
Commerce     34
Humanities   22
dtype: int64
Total Students
Science      60
Commerce     70
Humanities   42
dtype: int64
```

8. Create a Series object 'population' to store population of 5 different metro cities and display the population that are more than 300000

Input:

```
import pandas as pd
population=pd.Series( data: [400000, 25400, 301100, 100500,505000],
                      index: ['Mumbai', 'Kolkata', 'Delhi', 'Chennai','Bangluru'])
print (population)
print('Poplulation more than 300000')
print (population [population>300000])
```

Output:

```
Mumbai      400000
Kolkata      25400
Delhi        301100
Chennai      100500
Bangluru     505000
dtype: int64
Poplulation more than 300000
Mumbai      400000
Delhi        301100
Bangluru     505000
dtype: int64
```

9. Create a series 'temp' that stores temperature of seven days in it. Its index should be 'Sunday', 'Monday' Write script to 1. Display temp of first 3 days. 2. Display temp of last 3 days. 3. Display all temp in reverse order like Saturday, Friday,.... 4. Display temp from Tuesday to Friday. 5. Display square of all temperature

Input:

```
import pandas as pd
Temp=pd. Series( data: [45, 42, 40, 46, 39, 38, 40],
                 index: ['Sunday', 'Monday', 'Tuesday', 'wednesday', 'Thursday', 'Friday', 'Saturday'])
print (Temp)
print("Temp of first three days\n",Temp.head(3))
print ("Temp of last three days\n",Temp.tail(3))
print("Temp in reverse order\n",Temp[::-1])
print("Temp from Tuesday to Friday\n",Temp['Tuesday': 'Friday'])
print("Square of all Temprature\n",Temp*Temp)
```

Output:

```
Sunday      45
Monday      42
Tuesday     40
wednesday   46
Thursday    39
Friday      38
Saturday    40
dtype: int64
Temp of first three days
Sunday      45
Monday      42
Tuesday     40
dtype: int64
Temp of last three days
Thursday    39
Friday      38
Saturday    40
dtype: int64
Temp in reverse order
Saturday    40
Friday      38
Thursday    39
wednesday   46
Tuesday     40
Monday      42
Sunday      45
dtype: int64
```

```
Temp from Tuesday to Friday
Tuesday     40
wednesday   46
Thursday    39
Friday      38
dtype: int64
Square of all Temprature
Sunday      2025
Monday      1764
Tuesday     1600
wednesday   2116
Thursday    1521
Friday      1444
Saturday    1600
dtype: int64
```


10. Create a Series object 'employee' that stores salary of 7 employees. Write script to print 1. Total no of elements 2. Series is empty or not 3. Series consist NaN value or not 4. Count Non-NA elements 5. Axis labels

Input:

```
import pandas as pd
dict={'ram':34000,'hari':42000,'suman':30000,'chandan':45000,'raghu':23000}
Emp=pd.Series(dict)
print(Emp)
print("Total no of Employees",Emp.size)
if Emp.empty:
    print("Series is empty")
else:
    print("Series is not empty")
if Emp.hasnans:
    print ("Series contains NaN elements")
else:
    print("Series does not contains NaN elements")
print("Total no of Non NA elements ",Emp.count())
print("Axis labels\n",Emp.axes)
```

Output:

```
ram      34000
hari     42000
suman    30000
chandan  45000
raghu    23000
dtype: int64
Total no of Employees 5
Series is not empty
Series does not contains NaN elements
Total no of Non NA elements  5
Axis labels
[Index(['ram', 'hari', 'suman', 'chandan', 'raghu'], dtype='object')]
```

11. Create a pandas series from an ndarray

Input:

```
import pandas as pd
import numpy as np
s=pd.Series(np.array([1,3,4,7,8,8,9]))
print(s)
```

Output:

```
0    1
1    3
2    4
3    7
4    8
5    8
6    9
dtype: int32
```


12. Given a Series, print all the elements that are above the 75th percentile

Input:

```
import pandas as pd
import numpy as np
s=pd.Series(np.array([1,3,4,7,8,8,9]))
print(s)
result=s.quantile(q=0.75)
print('75th Percentile of the series is::')
print(result)
print('The elements that are above the 75th percentile::')
print(s[s>result])
```

Output:

```
0    1
1    3
2    4
3    7
4    8
5    8
6    9
dtype: int32
75th Percentile of the series is::
8.0
The elements that are above the 75th percentile::
6    9
dtype: int32
```

13. Create the following dataframe 'Sport' containing sport wise marks for five students. Use 2D dictionary to create dataframe

Input:

```
import pandas as pd
dict={'student':['jai','raj','john','karan','chandu'],
'sport': ['cricket', 'football', 'tennis','kabaddi', 'hockey'],'marks': [80,76,89,92,97]}
sport=pd.DataFrame(dict, index: ['I','II','III','IV','V'])
print(sport)
```

Output:

	student	sport	marks
I	jai	cricket	80
II	raj	football	76
III	john	tennis	89
IV	karan	kabaddi	92
V	chandu	hockey	97

14. Create a dataframe from list containing dictionaries of most economical bike with its name and rate of three companies. Company name should be the row labels

Input:

```
import pandas as pd
list1={'Name':'Sports','Cost':60000}
list2={'Name':'Discover','Cost': 62000}
list3={'Name':'splendor','Cost':63000}
Bike=[list1,list2,list3]
df=pd.DataFrame(Bike, index: ['TVS','Bajaj','Hero'])
print (df)
```

Output:

	Name	Cost
TVS	Sports	60000
Bajaj	Discover	62000
Hero	splendor	63000

15. Consider two series object staff and salaries that stores the number of people in various office branches and salaries distributed in these branches respectively. Write a program to create another Series object that stores average salary per branch and then create a dataframe object from these Series object. After creating dataframe rename all row labels with Branch name

Input:

```
import pandas as pd
staff=pd.Series([20,24,30,18])
salary=pd.Series([240000,336000,450000,270000])
avg=salary/staff
org={'Employees':staff,'Amount':salary,'Average':avg}
df=pd.DataFrame(org)
print("Without Row Label")
print(df)
df.index = ['sale', 'store', 'marketing','maintenance']
print("With Row Label")
print(df)
```

Output:

Without Row Label

	Employees	Amount	Average
0	20	240000	12000.0
1	24	336000	14000.0
2	30	450000	15000.0
3	18	270000	15000.0

With Row Label

	Employees	Amount	Average
sale	20	240000	12000.0
store	24	336000	14000.0
marketing	30	450000	15000.0
maintenance	18	270000	15000.0

16. Write program to do the followings 1. Display row labels of 'sales' 2. Display column label of 'sales' 3. Display last two rows of the 'sales' 4. Display first two rows of the 'sales'.

	2014	2015	2016	2017
<u>Madhu</u>	1000	2000	2400	2800
<u>Kusum</u>	1500	1800	5000	6000
<u>Kinshuk</u>	2000	2200	7000	7000
<u>Ankit</u>	3000	3000	1000	8000
<u>Shruti</u>	4000	4500	1250	9000

Input:

```
import pandas as pd
dict={'2014':[1000,1500,2000,3000,4000], '2015':[2000,1800,2200,3000,4500],
      '2017':[2400,5000,7000,1000,1250], '2017':[2800,6000,7000,8000,9000]}
sale=pd.DataFrame(dict, index: ['Madhu', 'Kusum', 'Kinshuk', 'Ankit', 'Shruti'])
print("----DataFrame----")
print(sale)
print("----Row Labels----")
print(sale.index)
print("----Column Labels----")
print(sale.columns)
print("----Bottom two Rows----")
print(sale.tail (2))
print("----Top two Rows----")
print(sale.head (2))
```

Output:

```
----DataFrame----
      2014  2015  2017
Madhu   1000  2000  2800
Kusum   1500  1800  6000
Kinshuk  2000  2200  7000
Ankit   3000  3000  8000
Shruti   4000  4500  9000
----Row Labels----
Index(['Madhu', 'Kusum', 'Kinshuk', 'Ankit', 'Shruti'], dtype='object')
----Column Labels----
Index(['2014', '2015', '2017'], dtype='object')
----Bottom two Rows----
      2014  2015  2017
Ankit   3000  3000  8000
Shruti   4000  4500  9000
----Top two Rows----
      2014  2015  2017
Madhu   1000  2000  2800
Kusum   1500  1800  6000
```


17. Create a dataframe 'sales2' using dictionary as given below and write a program to append 'sales2' to the dataframe 'sales'

	2018
Madhu	1600
Kusum	1100
Kinshuk	5000
Ankit	3400
Shruti	9000

Input:

```
import pandas as pd
dict={'2014':[1000,1500,2000,3000, 4000],'2015':[2000, 1800,2200,3000,4500],
      '2016':[2400, 5000, 7000, 1000, 1250],'2017':[2800, 6000, 7000, 8000, 9000]}
sale = pd.DataFrame(dict, index=[ 'Madhu', 'Kusum', 'Kinshuk','Ankit','Shruti'])
print("----DataFrame Sale----")
print(sale)
sale2=pd.DataFrame( data: {2018: [1600,1100,5000,5400, 9000]},
index: [ 'Madhu', 'Kusum', 'Kinshuk','Ankit','Shruti'])
print("----DataFrame Sale2----")
print(sale2)
sale=sale.join(sale2)
print("----DataFrame Sale after joining sale2----")
print(sale)
```

Output:

```
----DataFrame Sale----
      2014  2015  2016  2017
Madhu   1000  2000  2400  2800
Kusum   1500  1800  5000  6000
Kinshuk 2000  2200  7000  7000
Ankit   3000  3000  1000  8000
Shruti  4000  4500  1250  9000
----DataFrame Sale2----
      2018
Madhu   1600
Kusum   1100
Kinshuk 5000
Ankit   5400
Shruti  9000
----DataFrame Sale after joining sale2----
      2014  2015  2016  2017  2018
Madhu   1000  2000  2400  2800  1600
Kusum   1500  1800  5000  6000  1100
Kinshuk 2000  2200  7000  7000  5000
Ankit   3000  3000  1000  8000  5400
Shruti  4000  4500  1250  9000  9000
```

18. Create a dataframe 'aid' as given below and write program to do following

1. Display the books and shoes only
2. Display toys only
3. Display quantity in MP and CG for toys and books.
4. Display quantity of books in AP

	Toys	Books	Shoes
MP	7000	4300	6000
UP	3400	3200	1200
AP	7800	5600	3280
CG	4100	2000	3000

Input:

```
import pandas as pd
D={'Toys':{'MP':7000,'UP': 3400,'AP':7800,'CG':4100},
  'Books':{'MP': 4300, 'UP' : 3200, 'AP':5600, 'CG':2000},
  'Shoes':{'MP': 6000, 'UP': 1200, 'AP': 3280, 'CG': 3000}.}
aid=pd.DataFrame (D)
print('----DataFrame----')
print(aid)
print('----Display the books and shoes only----')
print(aid.loc[:,['Books', 'Shoes']])
print('----Display toys only----')
print(aid['Toys'])
print('----Display quantity in MP and CG for toys and books----')
print(aid.loc[['MP','CG'], ['Toys','Books']])
print('----Display quantity of books in AP----')
print(aid.at['AP','Books'])
```

Output:

```
----DataFrame----
   Toys  Books  Shoes
MP  7000   4300   6000
UP  3400   3200   1200
AP  7800   5600   3280
CG  4100   2000   3000
----Display the books and shoes only----
   Books  Shoes
MP    4300   6000
UP    3200   1200
AP    5600   3280
CG    2000   3000
----Display toys only----
MP    7000
UP    3400
AP    7800
CG    4100
Name: Toys, dtype: int64
----Display quantity in MP and CG for toys and books----
   Toys  Books
MP  7000   4300
CG  4100   2000
----Display quantity of books in AP----
5600
```

19. For given dataframe 'aid' in practical 7, write program to write the values of 'aid' to a comma separated file 'aidfigures.csv' on the disk. Do not write the row labels and column labels

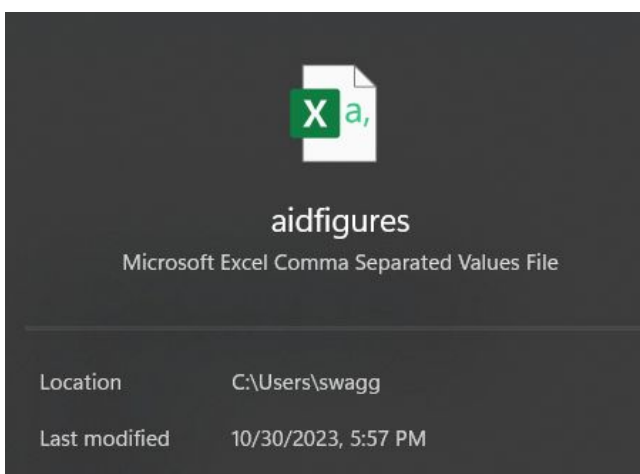
Input:

```
import pandas as pd
dict={'Toys':{'MP':7000,'UP':3400,'AP':7800,'CG':4100},
      'Books':{'MP': 4300,'UP':3200,'AP':5600,'CG':2000},
      'Shoes':{'MP':6000,'UP':1200,'AP':3280,'CG':3000},}
aid=pd.DataFrame (dict)
print(aid)
aid.to_csv(path_or_buf = 'C:/Users/swagg/aidfigures.csv', header = False, index = False)
```

Output:

	Toys	Books	Shoes
MP	7000	4300	6000
UP	3400	3200	1200
AP	7800	5600	3280
CG	4100	2000	3000

	A	B	C
1	7000	4300	6000
2	3400	3200	1200
3	7800	5600	3280
4	4100	2000	3000



20. Read the data in the file 'aidfigure.csv' into a dataframe 'aidretrieved' and display it. Now update the row labels and column labels of 'aidretrieved' to be the same as that of 'aid'

Input:

```
import pandas as pd
aidretrieved = pd.read_csv(filepath_or_buffer='C:/Users/swagg/aidfigures.csv',
names=['Toys', 'Books', 'Shoes'],)
aidretrieved.index=['MP', 'UP', 'AP', 'CG']
print(aidretrieved)
```

Output:

	Toys	Books	Shoes
MP	7000	4300	6000
UP	3400	3200	1200
AP	7800	5600	3280
CG	4100	2000	3000

21. Create a dataframe based on ecommerce data and generate descriptive statistics (mean, median, mode, quartile, and variance)

Input:

```
import pandas as pd
sales={'InvoiceNo':[1001,1002,1003,1004,1005,1006,1007],
'ProductName':['LED', 'AC', 'Deodrant', 'Jeans', 'Books', 'Shoes', 'Jacket'], 'Quantity':[2,1,2,1,2,1,1],
'Price': [65000, 55000, 500, 2500, 950, 3000, 2200]}
df=pd.DataFrame(sales)
print (df['Price'].describe().round(2))
```

Output:

count	7.00
mean	18450.00
std	28543.61
min	500.00
25%	1575.00
50%	2500.00
75%	29000.00
max	65000.00
Name: Price, dtype: float64	

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

Input:

```
import pandas as pd
dic={'Class':['I','II','III','IV','V','VI','VII','VIII','IX','X','XI','XII'],
     'Pass-Percentage':[100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 98.6, 100, 99]}
result=pd.DataFrame(dic)
print(result)
print(result.dtypes)
print('shape of the dataframe is:::::')
print(result.shape)
```

Output:

```
   Class  Pass-Percentage
0      I             100.0
1     II             100.0
2    III             100.0
3     IV             100.0
4      V             100.0
5     VI             100.0
6    VII             100.0
7   VIII             100.0
8     IX             100.0
9      X              98.6
10    XI             100.0
11   XII             99.0
Class              object
Pass-Percentage    float64
dtype: object
shape of the dataframe is:::::
(12, 2)
```

23.Filter out rows based on different criteria such as duplicate rows

Input:

```
import pandas as pd
dic={'Name': ['Rohit', 'Mohit', 'Deepak', 'Rohit', 'Deepak', 'Sohit', 'Geeta'],
'MarksinIP': [85,45,92,85,92,96,84]}
marks=pd.DataFrame(dic)
dr=marks[marks.duplicated (keep=False)]
print(dr)
```

Output:

	Name	MarksinIP
0	Rohit	85
2	Deepak	92
3	Rohit	85
4	Deepak	92

24. Find the sum of each column, or find the column with the lowest mean

Input:

```
import pandas as pd
Profit={'TCS': { 'Qtr1':2500, 'Qtr2': 2000, 'Qtr3': 3000, 'Qtr4':2000},
'WIPRO': { 'Qtr1':2800, 'Qtr2':2400, 'Qtr3':3600, 'Qtr4':2400},
'L&T': { 'Qtr1':2100, 'Qtr2':5700, 'Qtr3':35000, 'Qtr4':2100}}
df=pd.DataFrame(Profit)
print(df)
print('Column wise sum in dataframe is :::')
print(df.sum(axis=0))
print('Column wise mean value are::::::::::::::::::')
print(df.mean (axis=0))
print('Column with minimum mean value is::::::::::::::::::')
print(df.mean (axis=0).idxmin())
```

Output:

```
      TCS  WIPRO  L&T
Qtr1  2500   2800  2100
Qtr2  2000   2400  5700
Qtr3  3000   3600 35000
Qtr4  2000   2400  2100
Column wise sum in dataframe is :::
TCS      9500
WIPRO    11200
L&T     44900
dtype: int64
Column wise mean value are::::::::::::::::::
TCS      2375.0
WIPRO     2800.0
L&T     11225.0
dtype: float64
Column with minimum mean value is::::::::::::::::::
TCS
```

25. Locate the 3 largest values in a data frame

Input:

```
import pandas as pd
dic={'Name': ['Rohit', 'Mohit', 'Deepak', 'Anil', 'Pankaj', 'Sohit', 'Geeta'],
'MarksinIP': [85,45,92,85,98,96,84]}
marks=pd.DataFrame(dic)
print(marks.nlargest ( n: 3, columns: ['MarksinIP']))
```

Output:

	Name	MarksinIP
4	Pankaj	98
5	Sohit	96
2	Deepak	92

26. Subtract the mean of a row from each element of the row in a Data Frame

Input:

```
import pandas as pd
Profit={'TCS':{'Qtr1':2500,'Qtr2':2000,'Qtr3':3000,'Qtr4':2000},
'WIPRO':{'Qtr1':2800, 'Qtr2':2400, 'Qtr3':3600,'Qtr4':2400},
'L&T':{'Qtr1':2100, 'Qtr2':5700, 'Qtr3': 35000, 'Qtr4':2100}}
df=pd.DataFrame(Profit)
print(df)
print('Mean of each row is::::::::::')
print(df.mean(axis=1))
print('Dataframe after Subtracting mean value of each row from each element of that Row is :::')
print(df.sub (df.mean (axis=1), axis=0))
```

Output:

```
      TCS  WIPRO  L&T
Qtr1  2500   2800   2100
Qtr2  2000   2400   5700
Qtr3  3000   3600  35000
Qtr4  2000   2400   2100
Mean of each row is::::::::::
Qtr1      2466.666667
Qtr2      3366.666667
Qtr3     13866.666667
Qtr4      2166.666667
dtype: float64
Dataframe after Subtracting mean value of each row from each element of that Row is :::
      TCS      WIPRO      L&T
Qtr1   33.333333  333.333333 -366.666667
Qtr2 -1366.666667 -966.666667  2333.333333
Qtr3 -10866.666667 -10266.666667  21133.333333
Qtr4  -166.666667   233.333333  -66.666667
```

27. Replace all negative values in a data frame with a 0

Input:

```
import pandas as pd
dic={'Data1':[-5,-2,5,8,9,-6], 'Data2': [2,4,10,15,-5,-8]}
df=pd.DataFrame(dic)
print(df)
print("dataFrame after replacing negative values with 0:::")
df[df<0]=0
print(df)
```

Output:

```
   Data1  Data2
0     -5      2
1     -2      4
2      5     10
3      8     15
4      9     -5
5     -6     -8

dataFrame after replacing negative values with 0:::
   Data1  Data2
0      0      2
1      0      4
2      5     10
3      8     15
4      9      0
5      0      0
```

28. Replace all missing values in a data frame with a 999

Input:

```
import pandas as pd
import numpy as np
empdata={'empid':[101, 102, 103, 104, 105, 106],
'ename': ['Sachin', 'Vinod', 'Lakhbir', np.nan, 'Devinder', 'UmaSelvi'],
'Doj': ['12-01-2012', '15-01-2012', '05-09-2007', '17-01-2012', np.nan, '16-01-2012']}
df = pd.DataFrame(empdata)
print(df)
df=df.fillna({'ename':999, 'Doj':999})
print(df)
```

Output:

	empid	ename	Doj
0	101	Sachin	12-01-2012
1	102	Vinod	15-01-2012
2	103	Lakhbir	05-09-2007
3	104	NaN	17-01-2012
4	105	Devinder	NaN
5	106	UmaSelvi	16-01-2012
	empid	ename	Doj
0	101	Sachin	12-01-2012
1	102	Vinod	15-01-2012
2	103	Lakhbir	05-09-2007
3	104	999	17-01-2012
4	105	Devinder	999
5	106	UmaSelvi	16-01-2012

29.Importing data between Pandas and MySQL

Input:

```
271 import pandas as pd
272 import mysql.connector
273 connection=mysql.connector.connect(host="localhost", user='root',
274                                   passwd=~, database='class12')
275 print(connection)
276 table=pd.read_sql_query( sql: "show tables from class12",connection)
277 print(table)
```

Output:

```
<mysql.connector.connection_cext.MySQLConnection object at 0x102d85c90>
```

```
Tables_in_class12
0      countries
1      Hospital
2      student
```

```
+-----+
| Tables_in_class12 |
+-----+
| countries         |
| Hospital          |
| student           |
+-----+
3 rows in set (0.00 sec)
```

30.Exporting data between Pandas and MySQL

Input:

```
278 import mysql.connector import pandas as pd
279 import numpy as np
280 connection=mysql.connector.connect(host="localhost",user="root",
281                                   passwd=..., database="class12")
282 print(connection)
283 cursor=connection.cursor()
284 cursor.execute('delete from employee')
285 empdata={"empid" : [101, 102, 103, 104, 105, 106],
286          "ename":["Sachin", 'Vinod', 'Lakhbir', 'Anil', 'Devinder', 'Umaselvi'],
287          "Doj":["2012-01-12", '2012-01-15', '2007-09-05", 2012-01-17, 2007-09-05', '2012-01-16']}
288 df=pd.DataFrame(empdata)
289 print(df)
290 for (row, rs) in df.iterrows():
291     empid=str(int (rs[0]))
292     ename = rs [1]
293     Doj=(rs[2])
294     cursor.execute("insert into employee values ("+empid+", '"+ename+"', '"+Doj+"'")")
295 connection.commit()
296 cursor.close()
297 print("Data Transferred Successfully")
```

Output:

```
<mysql.connector.connection_cext.MySQLConnection object at 0x100f67fa8>

   empid  ename      Doj
0    101  Sachin  2012-01-12
1    102  Vinod   2012-01-15
2    103  Lakhbir  2007-09-05
3    104   Anil   2012-01-17
4    105  Devinder 2007-09-05
5    106  Umaselvi 2012-01-16

Data Transferred Successfully
```

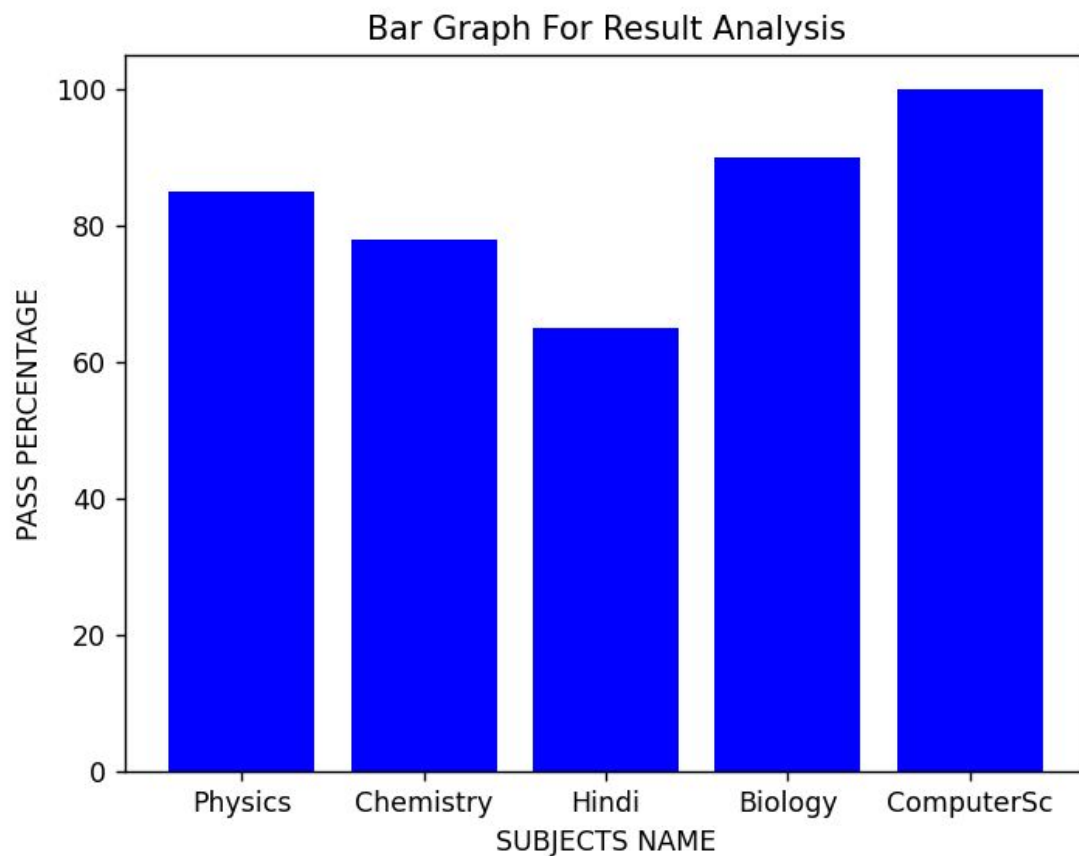
```
mysql> select * from employee;
+----+-----+-----+
| empid | ename | doj |
+----+-----+-----+
| 101 | Sachin | 2012-01-12 |
| 102 | Vinod | 2012-01-15 |
| 103 | Lakhbir | 2007-09-05 |
| 104 | Anil | 2012-01-17 |
| 105 | Devinder | 2007-09-05 |
| 106 | Umaselvi | 2012-01-16 |
+----+-----+-----+
6 rows in set (0.00 sec)
```


31. Given the school result data, analyse the performance of the students on different parameters, e.g subject wise or class wise

Input:

```
import matplotlib.pyplot as plt
Subject=['Physics', 'Chemistry', 'Hindi', 'Biology', 'ComputerSc']
Percentage=[85,78,65,90,100]
plt.bar(Subject, Percentage, align='center', color='blue')
plt.xlabel('SUBJECTS NAME')
plt.ylabel('PASS PERCENTAGE')
plt.title('Bar Graph For Result Analysis')
plt.show()
```

Output:

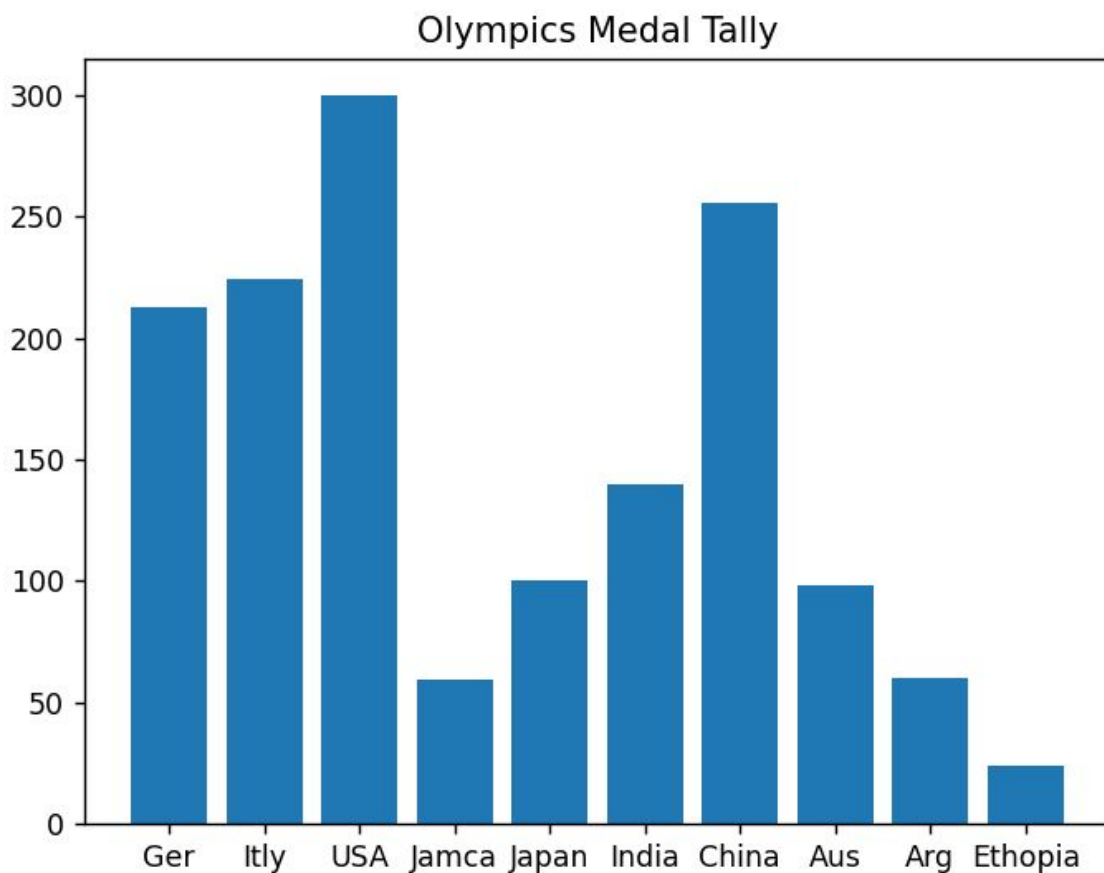


32. Collect and store total medals won by 10 countries in Olympic games and represent it in form of bar chart with title to compare and analyze data

Input:

```
import matplotlib.pyplot as plt
medals=[213,224,300,59,100,140,256,98,60,24]
country=['Ger','Itly','USA','Jamca','Japan','India','China','Aus','Arg','Ethopia']
plt.bar(country, medals)
plt.title('Olympics Medal Tally')
plt.show()
```

Output:



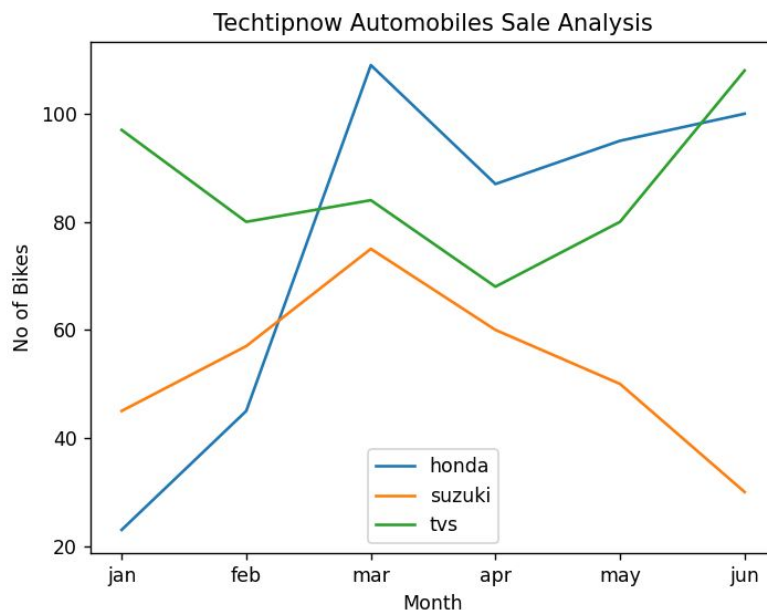
33.To get proper analysis of sale performance create multiple line chart on a common plot where all bike sale data are plotted. Display appropriate x and y axis labels, legend and chart title.

	Jan	Feb	Mar	Apr	May	Jun
Honda	23	45	109	87	95	100
Suzuki	45	57	75	60	50	30
Tvs	97	80	84	68	80	108

Input:

```
import matplotlib.pyplot as plt
month=['jan','feb','mar','apr','may','jun']
honda=[23,45,109,87,95,100]
suzuki=[45,57,75,60,50,30]
tvs=[97,80,84,68,80,108]
plt.plot(*args: month,honda,label='honda')
plt.plot(*args: month, suzuki,label='suzuki')
plt.plot(*args: month,tvs, label = 'tvs')
plt.title('Techtipnow Automobiles Sale Analysis')
plt.xlabel('Month')
plt.ylabel('No of Bikes')
plt.legend(loc = 'lower center')
plt.show()
```

Output:



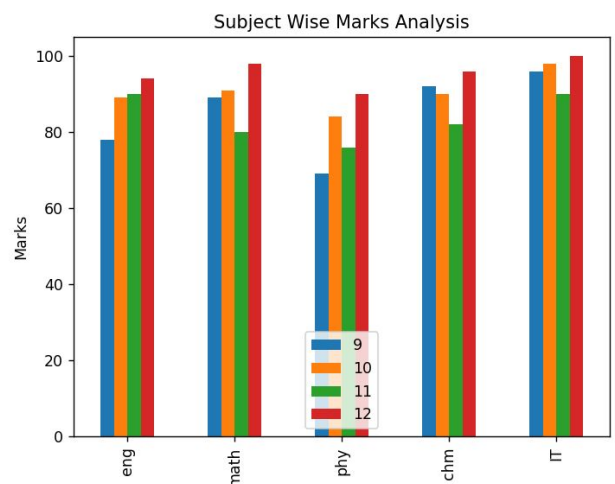
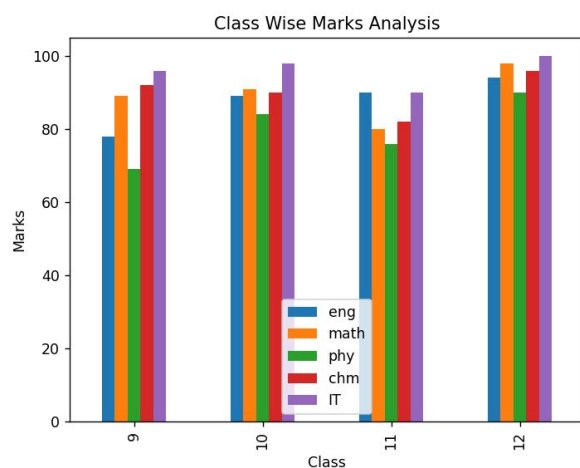
34. Given the school result data, analyse the performance of the student on different parameters, e.g. subject wise or class wise. Create a dataframe for the above, plot appropriate chart with title and legend

	Eng	Math	Phy	Chm	IT
9	78	89	69	92	96
10	89	91	84	90	98
11	90	80	76	82	90
12	94	98	90	96	100

Input:

```
import pandas as pd
import matplotlib.pyplot as plt
d={'eng':[78, 89, 90, 94], 'math':[89,91,80,98], 'phy':[69,84,76,90],
  'chm':[92, 90, 82, 96], 'IT':[96, 98, 90, 100]}
df=pd.DataFrame(d, index=[9,10,11,12])
print(df)
df.plot(kind='bar',title='Class Wise Marks Analysis',xlabel='Class',ylabel='Marks')
df1=df.T
df1.plot(kind='bar',title='Subject Wise Marks Analysis',xlabel='class',ylabel='Marks')
plt.legend(loc='lower center')
plt.show()
```

Output:



35. Construct a histogram from given data with 10 bin

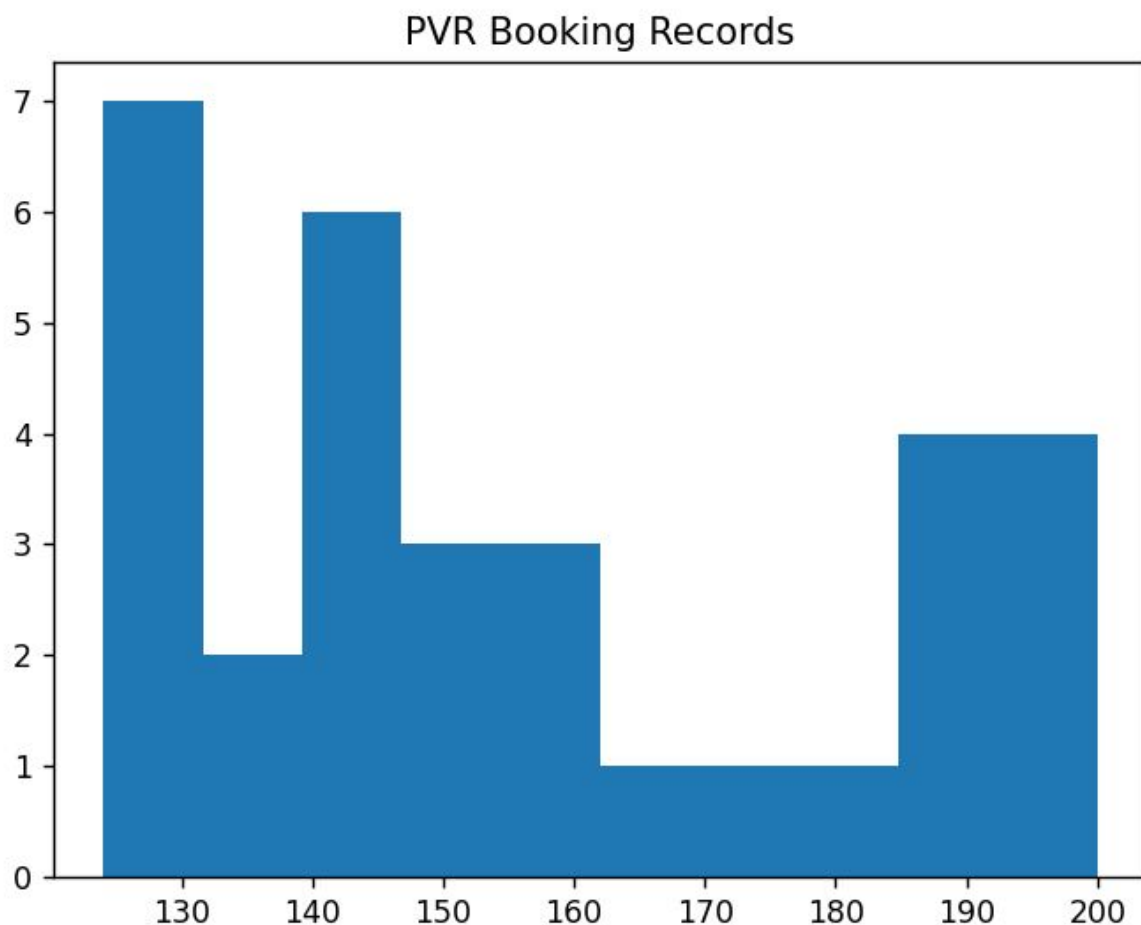
Practical: The following seat bookings are the daily records of a month December from PVR cinemas:

124,124,135,156,128,189,200,150,158, 150,200,124,143,142,130,130, 170,189,200,130,
142,167,180,143,143, 135,156,150,200,189,189,142

Input:

```
import pandas as pd
import matplotlib.pyplot as plt
L=[124,124,135,156,128,189,200,150,158, 150,200,124,143,142,130,130, 170,
189,200,130, 142,167,180,143,143,
135,156,150,200,189,189,142]
plt.hist(L)
plt.title("PVR Booking Records")
plt.show()
```

Output:

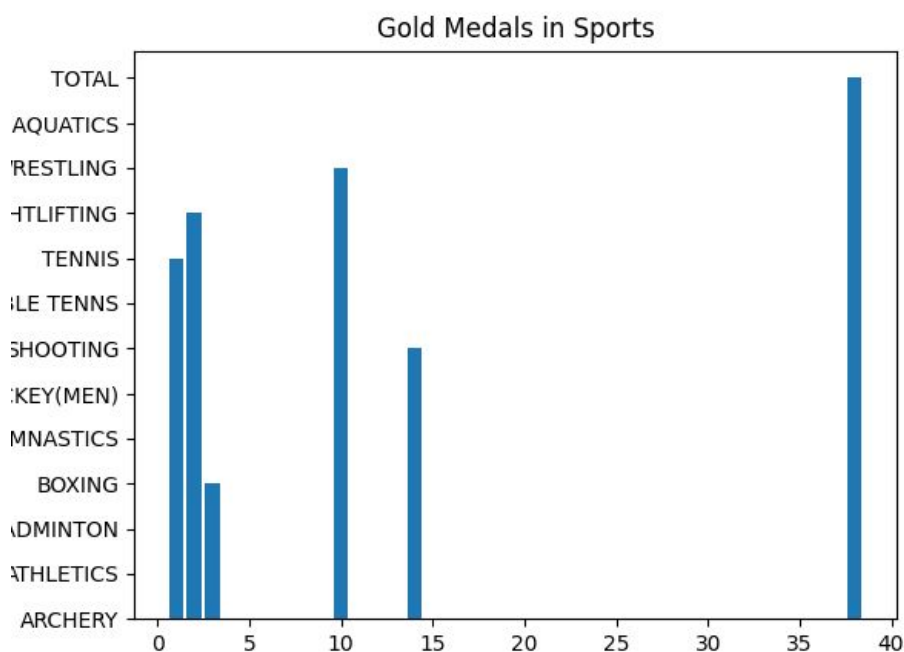


36. Take data of your interest from an open source (e.g. data.gove.in), aggregate and summarize it. Then plot it using different plotting functions of the Matplotlib library

Input:

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv('C:/Users/swagg/opensource1.csv')
medals=df['GOLD']
sports=df['DISCIPLINE']
plt.bar(medals,sports)
plt.title('Gold Medals in Sports')
plt.show()
```

	A	B	C	D	E	F
1	S.No.	DISCIPLINE	GOLD	SILVER	BRONZE	TOTAL
2	1	ARCHERY		3	1	4
3	2	ATHLETICS		2	3	7
4	3	BADMINTON		2	1	4
5	4	BOXING		3	NA	4
6	5	GYMNASTICS	NA		1	1
7	6	HOCKEY(MEN)	NA		1	NA
8	7	SHOOTING		14	11	5
9	8	TABLE TENNS		1	1	3
10	9	TENNIS		1	1	2
11	10	WEIGHTLIFTING		2	2	4
12	11	WRESTLING		10	5	4
13	12	PARALYMPIC – AQUATICS	NA	NA		1
14	TOTAL	TOTAL		38	27	36
						101



Structured Query Language (SQL)

37. Create Database named Class12

```
mysql> create database class12;  
Query OK, 1 row affected (0.02 sec)
```

38. Open Database Class12

```
mysql> use class12;  
Database changed
```

39. Create a student table with the student id, class, section, gender, name, dob, and marks as attributes where the student id is the primary key.

```
mysql> create table student (studentid int(4) primary key, class char(2),  
-> section char(1),  
-> gender char(1),  
-> name varchar(20),  
-> dob date,  
-> marks decimal(5,2));  
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

40.View the structure of the table

```
mysql> desc student;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| studentid  | int           | NO   | PRI | NULL     |       |
| class      | char(2)       | YES  |     | NULL     |       |
| section    | char(1)       | YES  |     | NULL     |       |
| gender     | char(1)       | YES  |     | NULL     |       |
| name       | varchar(20)   | YES  |     | NULL     |       |
| dob        | date          | YES  |     | NULL     |       |
| marks      | decimal(5,2)  | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.01 sec)
```

41.Insert the details of at least 10 students in the above table

```
mysql> insert into student values
-> (1101,'XI','A','M','Aksh','2005/12/23',88.21),
-> (1102,'XI','B','F','Moksha','2005/03/24',77.90),
-> (1103,'XI','A','F','Archi','2006/04/21',76.20),
-> (1104,'XI','B','M','Bhavin','2005/09/15',68.23),
-> (1105,'XI','C','M','Kevin','2005/08/23',66.33),
-> (1106,'XI','C','F','Naadiya','2005/10/27',62.33),
-> (1107,'XI','D','M','Krish','2005/01/23',84.33),
-> (1108,'XI','D','M','Ayush','2005/04/23',55.33),
-> (1109,'XI','C','F','Shruti','2005/06/01',74.33),
-> (1110,'XI','D','F','Shivi','2005/10/19',72.30);
```

42.Display the details of the table student

```
mysql> select * from student;
+-----+-----+-----+-----+-----+-----+-----+
| studentid | class | section | gender | name   | dob       | marks |
+-----+-----+-----+-----+-----+-----+-----+
| 1101      | XI    | A       | M      | Aksh  | 2005-12-23 | 88.21 |
| 1102      | XI    | B       | F      | Moksha | 2005-03-24 | 77.90 |
| 1103      | XI    | A       | F      | Archi  | 2006-04-21 | 76.20 |
| 1104      | XI    | B       | M      | Bhavin | 2005-09-15 | 68.23 |
| 1105      | XI    | C       | M      | Kevin  | 2005-08-23 | 66.33 |
| 1106      | XI    | C       | F      | Naadiya | 2005-10-27 | 62.33 |
| 1107      | XI    | D       | M      | Krish  | 2005-01-23 | 84.33 |
| 1108      | XI    | D       | M      | Ayush  | 2005-04-23 | 55.33 |
| 1109      | XI    | C       | F      | Shruti | 2005-06-01 | 74.33 |
| 1110      | XI    | D       | F      | Shivi  | 2005-10-19 | 72.30 |
+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```


43.Delete record of students who secured less than 65 marks

```
mysql> delete from student where marks <65;  
Query OK, 2 rows affected (0.00 sec)
```

```
mysql> select * from student;
```

studentid	class	section	gender	name	dob	marks
1101	XI	A	M	Aksh	2005-12-23	88.21
1102	XI	B	F	Moksha	2005-03-24	77.90
1103	XI	A	F	Archi	2006-04-21	76.20
1104	XI	B	M	Bhavin	2005-09-15	68.23
1105	XI	C	M	Kevin	2005-08-23	66.33
1107	XI	D	M	Krish	2005-01-23	84.33
1109	XI	C	F	Shruti	2005-06-01	74.33
1110	XI	D	F	Shivi	2005-10-19	72.30

44.Increase marks by 5% for who have studentid more than 1105.

```
mysql> update student set marks=marks+(marks*0.05) where studentid >1105;  
Query OK, 3 rows affected, 3 warnings (0.01 sec)  
Rows matched: 3 Changed: 3 Warnings: 3
```

```
[mysql> select * from student;
```

studentid	class	section	gender	name	dob	marks
1101	XI	A	M	Aksh	2005-12-23	88.21
1102	XI	B	F	Moksha	2005-03-24	77.90
1103	XI	A	F	Archi	2006-04-21	76.20
1104	XI	B	M	Bhavin	2005-09-15	68.23
1105	XI	C	M	Kevin	2005-08-23	66.33
1107	XI	D	M	Krish	2005-01-23	88.55
1109	XI	C	F	Shruti	2005-06-01	78.05
1110	XI	D	F	Shivi	2005-10-19	75.92

8 rows in set (0.00 sec)

45.Display the content of the table of female students

```
[mysql> select * from student where gender = 'f';
```

studentid	class	section	gender	name	dob	marks
1102	XI	B	F	Moksha	2005-03-24	77.90
1103	XI	A	F	Archi	2006-04-21	76.20
1109	XI	C	F	Shruti	2005-06-01	78.05
1110	XI	D	F	Shivi	2005-10-19	75.92

4 rows in set (0.00 sec)

46.Display studentid, name and marks whose marks are more than 50

```
[mysql> select studentid,name,marks from student where marks>50;
```

studentid	name	marks
1101	Aksh	88.21
1102	Moksha	77.90
1103	Archi	76.20
1104	Bhavin	68.23
1105	Kevin	66.33
1107	Krish	88.55
1109	Shruti	78.05
1110	Shivi	75.92

```
8 rows in set (0.00 sec)
```

47.Find the average of marks from the student table

```
[mysql> select avg(marks) from student;
```

avg(marks)
77.423750

```
1 row in set (0.01 sec)
```

48.Find the number of students who are from section A

```
[mysql> select count(section) from student where section='A';
```

count(section)
2

```
1 row in set (0.00 sec)
```

49.Add a new column email in the above table

```
[mysql> alter table student add column email varchar(20);  
Query OK, 0 rows affected (0.04 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

50. Find the number of students who are from section A

```
[mysql> select count(section) from student where section='A';
+-----+
| count(section) |
+-----+
|                2 |
+-----+
1 row in set (0.00 sec)
```

51. Add a new column email in the above table

```
[mysql> alter table student add column email varchar(20);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

52. Add the email ids of each student in the created column email

```
[mysql> update student set email='a@a.com';
Query OK, 8 rows affected (0.00 sec)
Rows matched: 8 Changed: 8 Warnings: 0
```

53.Display studentid,name and DOB of who are born in 2005

```
mysql> select studentid, name, dob from student where dob between '2005-01-01' and '2005-12-31';
```

studentid	name	dob
1101	Aksh	2005-12-23
1102	Moksha	2005-03-24
1104	Bhavin	2005-09-15
1105	Kevin	2005-08-23
1107	Krish	2005-01-23
1109	Shruti	2005-06-01
1110	Shivi	2005-10-19

7 rows in set, 8 warnings (0.01 sec)

54.Display studentid and DOB in ascending order of their names

```
mysql> select studentid, name, dob from student order by name;
```

studentid	name	dob
1101	Aksh	2005-12-23
1103	Archi	2006-04-21
1104	Bhavin	2005-09-15
1105	Kevin	2005-08-23
1107	Krish	2005-01-23
1102	Moksha	2005-03-24
1110	Shivi	2005-10-19
1109	Shruti	2005-06-01

8 rows in set (0.00 sec)

55.Display studentid,gender,name,DOB,marks,email in descending order by their marks

```
mysql> select studentid, gender, name, dob, marks, email from student order
-> by marks desc;
```

studentid	gender	name	dob	marks	email
1107	M	Krish	2005-01-23	88.55	a@a.com
1101	M	Aksh	2005-12-23	88.21	a@a.com
1109	F	Shruti	2005-06-01	78.05	a@a.com
1102	F	Moksha	2005-03-24	77.90	a@a.com
1103	F	Archi	2006-04-21	76.20	a@a.com
1110	F	Shivi	2005-10-19	75.92	a@a.com
1104	M	Bhavin	2005-09-15	68.23	a@a.com
1105	M	Kevin	2005-08-23	66.33	a@a.com

8 rows in set (0.01 sec)

56.Display the unique section available in the table

```
mysql> select distinct section from student;
```

section
A
B
C
D

4 rows in set (0.00 sec)

57. Find the total number of customers from each country in the table (customer ID, customer name, country) using group by

```
mysql> select country, count(cname) as 'Total_Customers' from customer
-> group by country;
```

country	Total_Customers
india	3
pak	3
nepal	1

3 rows in set (0.01 sec)

58. For the given table 'Hospital' write SQL command to display name all patient admitted in month of May.

PID	PNAME	ADMITDATE	DEPT	FEES
AP/PT/001	Rahil Khan	21/04/2019	ENT	250
AP/PT/002	Jitendal Pal	12/05/2019	Cardio	400
AP/PT/003	Suman Lakra	19/05/2019	Cardio	400
AP/PT/004	Chandumal Jain	24/06/2019	Neuro	600

```
mysql> select * from hospital where monthname(admitdate) = 'May';
```

pid	pname	admitdate	dept	fees
ap/pt/002	jitendra pal	2019-05-12	cardio	400
ap/pt/003	suman lakra	2019-05-19	cardio	400

2 rows in set (0.02 sec)

59. For the given table 'Hospital' write SQL command to Display patient name in upper case with year of admission.

```
mysql> select upper(pname) as 'patient name', year(admitdate) as 'admit year'
-> from hospital;
```

patient name	admit year
RAHIL KHAN	2019
JITENDRA PAL	2019
SUMAN LAKRA	2019
CHANDUMAL JAIN	2019

4 rows in set (0.00 sec)

60. Find min, max and average of marks in the student table

```
mysql> select min(marks) as Min_marks, max(marks) as Max_Marks, sum(marks) as Total_Marks, avg(marks) as Average_Marks from student;
```

Min_marks	Max_Marks	Total_Marks	Average_Marks
55.33	88.21	653.19	72.576667

```
1 row in set (0.04 sec)
```

61. Delete the details of a particular student from the table student table

```
mysql> delete from student where name = 'Ayush';
Query OK, 1 row affected (0.05 sec)
```

```
mysql> select * from student;
```

studentid	class	section	gender	name	dob	marks
1101	XII	A	M	Aksh	2005-12-23	88.21
1102	XII	B	F	Moksha	2005-03-24	77.90
1103	XII	A	F	Archi	2006-04-21	76.20
1104	XII	B	M	Bhavin	2005-09-15	68.23
1105	XII	C	M	Kevin	2005-08-23	66.33
1106	XII	C	F	Naadiya	2005-10-27	62.33
1107	XII	D	M	Krish	2005-01-23	84.33
1109	XII	C	F	Shruti	2005-06-01	74.33

```
8 rows in set (0.00 sec)
```

62. Delete the details of the student table

```
mysql> delete from student;
Query OK, 5 rows affected (0.01 sec)
```

63. Create a new table (name, date of birth) by joining two tables (student id, name) and (student id, date of birth)

```
mysql> select * from students;
```

studentid	name
1	SACHIN
2	AMIT

2 rows in set (0.00 sec)

```
mysql> select * from studentbirth;
```

studentid	dob
1	1982-06-17
2	1981-05-20

2 rows in set (0.00 sec)

```
mysql> create table stu2 as select name,dob from students,studentbirth  
-> where students.studentid=studentbirth.studentid;
```

Query OK, 2 rows affected (0.11 sec)

Records: 2 Duplicates: 0 Warnings: 0

```
mysql> select * from stu2;
```

name	dob
SACHIN	1982-06-17
AMIT	1981-05-20

2 rows in set (0.00 sec)

64. Create a foreign key in one of the two table mentioned

```
mysql> alter table orders add foreign key(studentid) references students(studentid);  
Query OK, 2 rows affected (0.00 sec)
```