

## Practical File- Informatics Practices (Class XII)

**#1 Create a pandas series from a dictionary of values and an ndarray.**

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

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

```
# import the pandas lib as pd
import pandas as pd

# create a dictionary
dictionary = {'A' : 10, 'B' : 20, 'C' : 30}

# create a series
series = pd.Series(dictionary)

print(series)
```

```
A    10
B    20
C    30
dtype: int64
```

**#2. Given a Series, print all the elements that are above the 75th percentile.**

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

```
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
```

**#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**

```
import pandas as pd
dic={'itemcat':['car','Ac','Aircoller','Washing Machine'],
     'itemname':['Ford','Hitachi','Symphony','LG'],
     'expenditure':[7000000,50000,12000,14000]}
quartsales=pd.DataFrame(dic)
print(quartsales)
qs=quartsales.groupby('itemcat')
print('Result after Filtering Dataframe')
print(qs['itemcat','expenditure'].sum())
```

	itemcat	itemname	expenditure
0	car	Ford	7000000
1	Ac	Hitachi	50000
2	Aircoller	Symphony	12000
3	Washing Machine	LG	14000

Result after Filtering Dataframe

	expenditure
itemcat	
Ac	50000
Aircoller	12000
Washing Machine	14000
car	7000000

**#4. Create a data frame based on ecommerce data and generate descriptive statistics (mean, median, mode, quartile, and variance).**

```
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))
```

```
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
```

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

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

```
   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)
```

## #6. Filter out rows based on different criteria such as duplicate rows.

```
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)
# Find duplicate rows
duplicateRow = marks[marks.duplicated(keep=False)]
print(duplicateRow)
```

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

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

```
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()
print('Column wise sum in dataframe is :::')
print(df.sum(axis=0))
# Print mean value of each column
print()
print('Column wise mean value are::::::::::')
print(df.mean(axis=0))
# Returns Column with minimum mean value
print()
print('Column with minimum mean value is::::::::::')
df.mean(axis=0).idxmin()
```

	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'
```

## #8. Locate the 3 largest values in a data frame.

```
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)
# Find 3 Largest Value for MarksinIP Column
print(marks.nlargest(3,['MarksinIP']))
```

	Name	MarksinIP
4	Pankaj	98
5	Sohit	96
2	Deepak	92



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

```
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()
print('Mean of each row is::::::::::')
print(df.mean(axis=1))
print()
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))
```

	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

## #10. Replace all negative values in a data frame with a 0.

```
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()
print("dataFrame after replacing negative values with 0:::")
df[df<0]=0
print(df)
```

	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

## #11. Replace all missing values in a data frame with a 999.

```
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()
print(df)
```

	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

Act

## #12. Importing and exporting data between pandas and CSV file

```
1 # importing pandas module
2 import pandas as pd
3 # making data frame
4 df = pd.read_csv("E:\emp.csv")
5 print(df)
6
```

	empid	ename	doj
0	101	Sachin Bhardwaj	12-01-2012
1	102	Vinod Verma	15-01-2012
2	103	Anand Ganesh	05-09-2007

```
1 import pandas as pd
2 l = [{'Name': 'Sachin', 'SirName': 'Bhardwaj'},
3      {'Name': 'Vinod', 'SirName': 'Verma'},
4      {'Name': 'Rajesh', 'SirName': 'Mishra'}]
5 df1=pd.DataFrame(l)
6 # saving the dataframe
7 df1.to_csv('E:\Dataframe1.csv')
```

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		Name	SirName										
2		0	Sachin	Bhardwaj									
3		1	Vinod	Verma									
4		2	Rajesh	Mishra									
5													

## #13. Importing and exporting data between pandas and MySQL database.

### Importing Data from MySQL to Data Frame.

```
In [1]: import pandas as pd
```

```
In [2]: import mysql.connector
```

```
In [5]: con=mysql.connector.connect(host="localhost",user="root",passwd="root",database="sachin")
print(con)
```

```
<mysql.connector.connection.MySQLConnection object at 0x0000000009C1D7F0>
```

```
In [6]: emp=pd.read_sql_query("show tables from sachin",con)
emp
```

```
Out[6]: Tables_in_sachin
```

0	employee
---	----------

## Exporting data from Data Frame to MYSQL.

```
In [8]: import mysql.connector
import pandas as pd
con=mysql.connector.connect(host="localhost",user="root",passwd="root",database="sachin")
print(con)
c=con.cursor()
print(df)
c.execute("delete from employee")
con.commit()
for(row,rs) in df.iterrows():
    empid=str(int(rs[0]))
    ename=rs[1]
    Doj=(rs[2])
    c.execute("insert into employee values("+ empid +"," + ename +"," + Doj +")")
con.commit()
c.close()
empdata={ 'empid':[101,102,103,104,105,106],
          'ename':['Sachin','Vinod','Lakhbir','Anil','Devinder','UmaSelvi'],
          'Doj':['2012-01-12','2012-01-15','2007-09-05','2012-01-17','2007-09-05','2012-01-16'] }
df=pd.DataFrame(empdata)
print("Dta transfer Successfully")
```

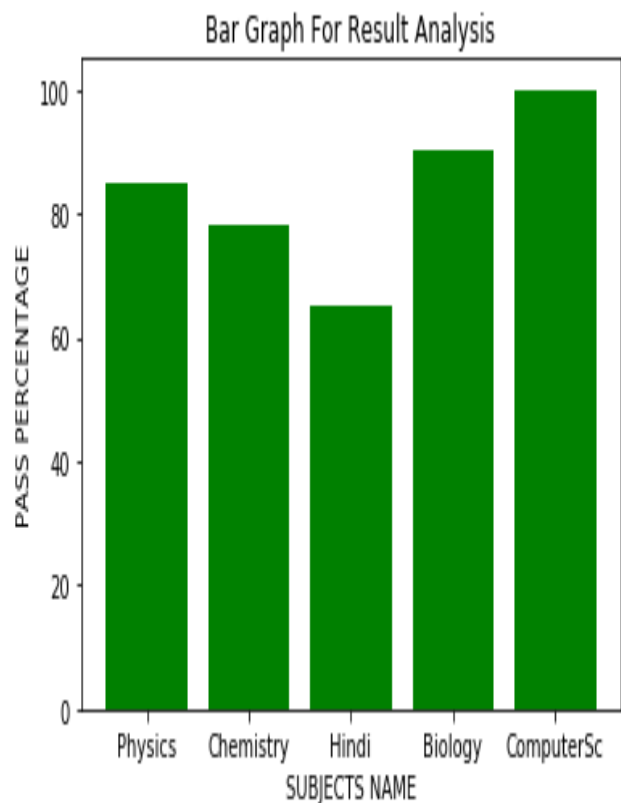
```
<mysql.connector.connection.MySQLConnection object at 0x000001F78BC5A828>
```

```
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
Dta transfer 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 (0.05 sec)
```

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

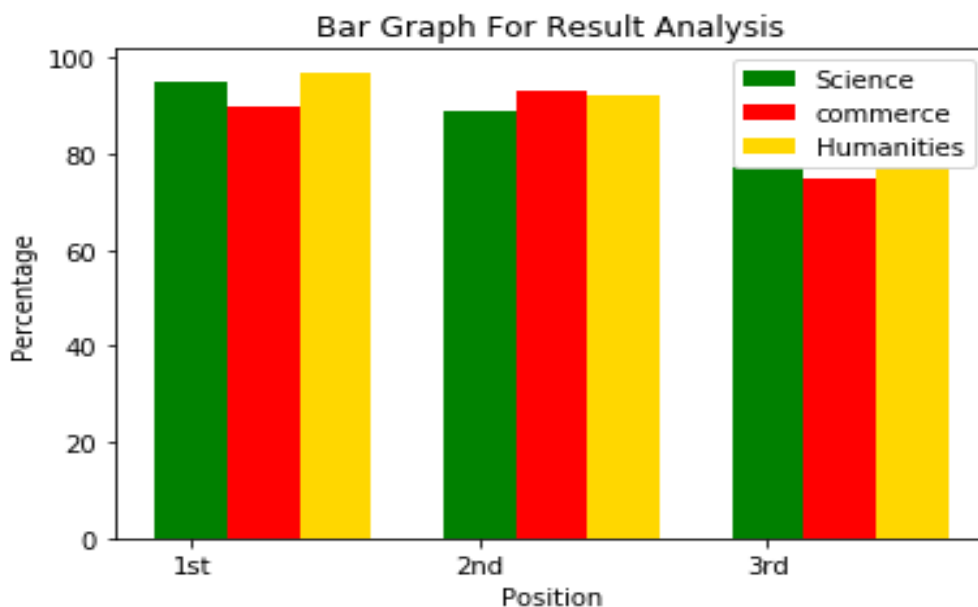
```
import matplotlib.pyplot as plt
Subject=['Physics','Chemistry','Hindi','Biology','ComputerSc']
Percentage=[85,78,65,90,100]
plt.bar(Subject,Percentage,align='center',color='green')
plt.xlabel('SUBJECTS NAME')
plt.ylabel('PASS PERCENTAGE')
plt.title('Bar Graph For Result Analysis')
plt.show()
```



A  
G

**#15. For the Data frames created above, analyze and plot appropriate charts with title and legend.**

```
import matplotlib.pyplot as plt
import numpy as np
s=['1st', '2nd', '3rd']
per_sc=[95,89,77]
per_com=[90,93,75]
per_hum=[97,92,77]
x=np.arange(len(s))
plt.bar(x,per_sc,label='Science',width=0.25,color='green')
plt.bar(x+.25,per_com,label='commerce',width=0.25,color='red')
plt.bar(x+.50,per_hum,label='Humanities',width=0.25,color='gold')
plt.xticks(x,s)
plt.xlabel('Position')
plt.ylabel('Percentage')
plt.title('Bar Graph For Result Analysis')
plt.legend()
plt.show()
```





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

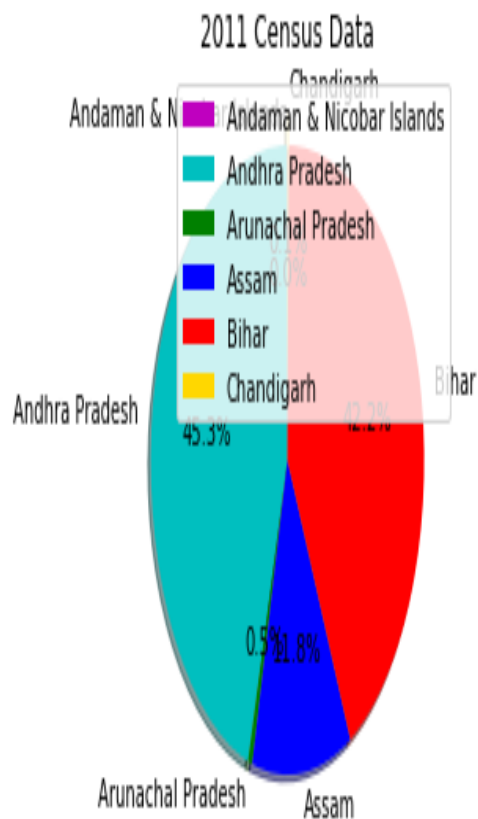
```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv("E:\census.csv")
print(df)
```

	S.No.	State/UT	Total Population of other
0	1	Andaman & Nicobar Islands	47
1	2	Andhra Pradesh	43769
2	3	Arunachal Pradesh	495
3	4	Assam	11374
4	5	Bihar	40827
5	6	Chandigarh	142
6	7	Chhattisgarh	6591
7	8	Dadra & Nagar Haveli	43
8	9	Daman & Diu	59
9	10	Goa	398
10	11	Gujarat	11544
11	12	Haryana	8422
12	13	Himachal Pradesh	2051
13	14	Jammu & Kashmir	4137
14	15	Jharkhand	13463
15	16	Karnataka	20266
16	17	Kerala	3902
17	18	Lakshadweep	2
18	19	Madhya Pradesh	29597
19	20	Maharashtra	40891
20	21	Manipur	1343
21	22	Meghalaya	627
22	23	Mizoram	166
23	24	Nagaland	398
24	25	NCT of Delhi	4213
25	26	Odisha	20332
26	27	Puducherry	252
27	28	Punjab	10243
28	29	Rajasthan	16517
29	30	Sikkim	126
30	31	Tamil Nadu	22364
31	32	Tripura	833
32	33	Uttar Pradesh	137465
33	34	Uttarakhand	4555
34	35	West Bengal	30349
35	Total	Total	487803

```

import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv("E:\census.csv")
slices=(df['Total Population of other'].head(6))
states=(df['State/UT'].head(6))
cols=['m','c','g','b','r','gold']
exp=[0,0,0,0,0,0.1]
plt.pie(slices, labels=states, colors=cols, startangle=90, explode=exp, shadow=True, autopct='%1f%%')
plt.title('2011 Census Data')
plt.legend()
plt.show()

```



**#17. Create a student table with the student id, name, and marks as attributes where the student id is the primary key.**

```
1 create table student
2 (
3 student_id varchar(10) primary key not NULL,
4 name varchar(30),
5 marks integer(5)
6 );
```

**#18. Insert the details of a new student in the above table.**

```
insert into student values( 101, 'Rohit',410);
insert into student values( 102, 'Mohit',425);
insert into student values(103,'Rahul',475);
insert into student values(104,'Virat',495);
```

**#19. Delete the details of a particular student in the above table.**

```
delete from student where name='Rahul';
select * from student;
```

101	Rohit	410
102	Mohit	425
104	Virat	495

**#20.** Use the select command to get the details of the students with marks more than 80.

```
select * from student WHERE marks>80;
```

**#21.** Create a new table (order ID, customer Name, and order Date) by joining two tables (order ID, customer ID, and order Date) and (customer ID, customer

```
mysql> select * from orders;
```

orderid	customerid	orderdate
1	1	2020-10-10
2	1	2020-04-05
3	2	2019-01-01

```
mysql> SELECT * FROM CUSTOMER;
```

customerid	customername	contactname	country
1	AMIT	AMIT	INDIA
2	SUMIT	SUMI	INDIA
3	NITIN	NITIN	AUSTRALIA
4	ROCKY	RAKESH	AUSTRALIA

```
mysql> create table orderdetails as select orderid,customername,orderdate  
-> from orders,customer where orders.customerid=customer.customerid;
```

```
mysql> select * from orderdetails;
```

orderid	customername	orderdate
1	AMIT	2020-10-10
2	AMIT	2020-04-05
3	SUMIT	2019-01-01

**#22. Create a foreign key in one of the two tables mentioned above**

```
mysql> alter table orders add foreign key(customerid) references customer(customerid);  
Query OK, 3 rows affected (1.22 sec)
```

**#23. Find the min, max, sum, and average of the marks in a student marks table.**

```
select Min(marks), Max(marks), Sum(marks), Avg(marks) from student;
```

```
410    495    1805    451.2500
```

**#24. Find the total number of customers from each country in the table (customer ID, customer Name, country) using group by.**

```
mysql> select country,count(*) "TOTAL CUSTOMER" from customer group by country;  
+-----+-----+  
| country | TOTAL CUSTOMER |  
+-----+-----+  
| AUSTRALIA | 2 |  
| INDIA | 2 |  
+-----+-----+
```

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

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

studentid	name
1	SACHIN
2	AMIT

```
2 rows in set (0.01 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 student,studentbirth  
-> where student.studentid=studentbirth.studentid;
```

```
Query OK, 2 rows affected (0.37 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)
```

**#26. Write a SQL query to order the (student ID, marks) table in descending order of the marks.**

```
select student_id, marks from student order by marks desc;
```