

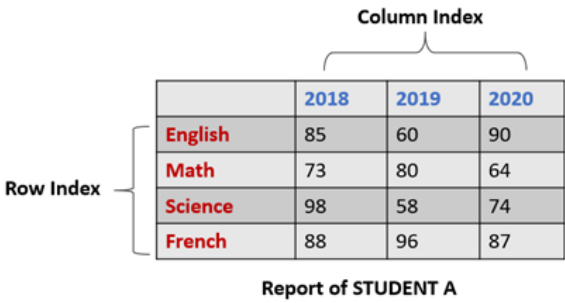
PANDAS DATAFRAME

Analysis and Visualization of large datasets represented in a PANDAS DataFrame

A Pandas DataFrame is a type of a 2 dimensional data structure which can be used to store large datasets in a matrix (tabular) format comprising of rows and columns

Figure below shows a Pandas DataFrame comprising of following details of five students arranged in 5 ROWs x 5 COLUMNs

- Every ROW represents a **Data Record**
- Every COLUMN represents an **ATTRIBUTE or PARAMETER** related to STUDENT



1A. Creating a Pandas DataFrame: pd.DataFrame() Constructor

- Creating a Pandas DataFrame using pd.DataFrame() command
- Inserting Data Elements using Python Dictionary

	Student Name	Roll No.	Physics	Chemistry	Maths
0	Aman	AC-101	78	80	88
1	Vishal	AC-102	67	77	95
2	Anshu	AC-103	85	85	100
3	Shruti	AC-104	96	90	90
4	Rahul	AC-105	56	62	78

```
# Creating a DataFrame from the given dataset using DataFrame() constructor
# Using Python Dictionary to insert data elements

import pandas as pd

StudentData = {
    'Student Name':['Aman', 'Vishal', 'Anshu', 'Shruti', 'Rahul'],
    'Roll No.':['AC-101', 'AC-102', 'AC-103', 'AC-104', 'AC-105'],
    'Physics':[78,67,85,96,56],
    'Chemistry': [80,77,85,90,62],
    'Maths':[88,95,100,90,78]
}

'''StudentData is a variable of type 'Dictionary' which is used to store all the
data-elements to be added in a DataFrame'''

STUDENT_DF = pd.DataFrame(StudentData) # creating data frame STUDENT_DF

display(STUDENT_DF) # printing Data Frame (Data Table)

type(STUDENT_DF)
```

	Student Name	Roll No.	Physics	Chemistry	Maths
0	Aman	AC-101	78	80	88
1	Vishal	AC-102	67	77	95
2	Anshu	AC-103	85	85	100
3	Shruti	AC-104	96	90	90
4	Rahul	AC-105	56	62	78

pandas.core.frame.DataFrame



1B. Saving Pandas DataFrame in an External File

```
import pandas as pd

StudentData = {
    'Student Name':['Aman', 'Vishal', 'Anshu', 'Shruti', 'Rahul'],
    'Roll No.':['AC-101', 'AC-102', 'AC-103', 'AC-104', 'AC-105'],
    'Physics':[78,67,85,96,56],
    'Chemistry': [80,77,85,90,62],
    'Maths':[88,95,100,90,78]
}

STUDENT_DF = pd.DataFrame(StudentData) # creating data frame STUDENT_DF

display(STUDENT_DF)
```

```
STUDENT_DF.to_excel('STUDENT_DF.xlsx') # saving data in excel format

STUDENT_DF.to_csv('STUDENT_DF.csv')    # saving data in csv format

STUDENT_DF.to_json('STUDENT_DF.json')   # saving data in csv format
```

	Student Name	Roll No.	Physics	Chemistry	Maths
0	Aman	AC-101	78	80	88
1	Vishal	AC-102	67	77	95
2	Anshu	AC-103	85	85	100
3	Shruti	AC-104	96	90	90
4	Rahul	AC-105	56	62	78

1C. Renaming Row Index

```
import pandas as pd

StudentData = {
    'Student Name': ['Aman', 'Vishal', 'Anshu', 'Shruti', 'Rahul'],
    'Roll No.': ['AC-101', 'AC-102', 'AC-103', 'AC-104', 'AC-105'],
    'Physics': [78,67,85,96,56],
    'Chemistry': [80,77,85,90,62],
    'Maths': [88,95,100,90,78]
}

STUDENT_DF = pd.DataFrame(StudentData) # creating data frame STUDENT_DF
STUDENT_DF.index= ['Student1','Student2','Student3','Student4', 'Student5' ]

display(STUDENT_DF)
```

	Student Name	Roll No.	Physics	Chemistry	Maths
Student1	Aman	AC-101	78	80	88
Student2	Vishal	AC-102	67	77	95
Student3	Anshu	AC-103	85	85	100
Student4	Shruti	AC-104	96	90	90
Student5	Rahul	AC-105	56	62	78

1D. Setting a particular column as INDEX column using .set_index() function

```
import pandas as pd

StudentData = {
    'Student Name': ['Aman', 'Vishal', 'Anshu', 'Shruti', 'Rahul'],
    'Roll No.': ['AC-101', 'AC-102', 'AC-103', 'AC-104', 'AC-105'],
    'Physics': [78,67,85,96,56],
    'Chemistry': [80,77,85,90,62],
    'Maths': [88,95,100,90,78]
}

STUDENT_DF = pd.DataFrame(StudentData) # creating data frame STUDENT_DF
STUDENT_DF=STUDENT_DF.set_index('Student Name')

display(STUDENT_DF)
```

	Roll No.	Physics	Chemistry	Maths
Student Name				
Aman	AC-101	78	80	88
Vishal	AC-102	67	77	95
Anshu	AC-103	85	85	100
Shruti	AC-104	96	90	90
Rahul	AC-105	56	62	78

DATA FILTERING

1E. Accessing Details of a Particular student by specifying Roll Number

```
import pandas as pd

StudentData = {
    'Student Name': ['Aman', 'Vishal', 'Anshu', 'Shruti', 'Rahul'],
    'Roll No.': ['AC-101', 'AC-102', 'AC-103', 'AC-104', 'AC-105'],
    'Physics': [78,67,85,96,56],
    'Chemistry': [80,77,85,90,62],
    'Maths': [88,95,100,90,78]
}

STUDENT_DF = pd.DataFrame(StudentData) # creating data frame STUDENT_DF
```

```
STUDENT_DF=STUDENT_DF.set_index('Student Name')

# METHOD 1 using loc() function
print("Using Method 1\n")
display(STUDENT_DF.loc[STUDENT_DF['Roll No']=='AC-101'])

# METHOD 2
print("Using Method 2\n")
display(STUDENT_DF[STUDENT_DF['Roll No']=='AC-101'])

# METHOD 3
print("Using Method 3\n")
display(STUDENT_DF.iloc[0])

# METHOD 4
print('Details of Anshu')
display(STUDENT_DF.loc[['Anshu']])
```

Using Method 1

	Roll No	Physics	Chemistry	Maths
Student Name				
Aman	AC-101	78	80	88

Using Method 2

	Roll No	Physics	Chemistry	Maths
Student Name				
Aman	AC-101	78	80	88

Using Method 3

Roll No AC-101
Physics 78
Chemistry 80
Maths 88
Name: Aman, dtype: object
Details of Anshu

	Roll No	Physics	Chemistry	Maths
Student Name				
Anshu	AC-103	85	85	100

1F. List the names of students scoring marks >= 80 in Physics

```
import pandas as pd

StudentData = {
    'Student Name':['Aman', 'Vishal', 'Anshu', 'Shruti', 'Rahul'],
    'RollNo':['AC-101', 'AC-102', 'AC-103', 'AC-104', 'AC-105'],
    'Physics':[78,67,85,96,56],
    'Chemistry': [80,77,85,90,62],
    'Maths':[88,95,100,90,78]
}

STUDENT_DF = pd.DataFrame(StudentData) # creating data frame STUDENT_DF
STUDENT_DF=STUDENT_DF.set_index('Student Name')

display(STUDENT_DF.loc[STUDENT_DF['Physics']>=80])

print("Number of students having MARKS >80 in Physics")
STUDENT_DF1= STUDENT_DF.loc[STUDENT_DF['Physics']>=80]
STUDENT_DF1['RollNo'].count()
```

	RollNo	Physics	Chemistry	Maths
Student Name				
Anshu	AC-103	85	85	100
Shruti	AC-104	96	90	90

Number of students having MARKS >80 in Physics
2

2. Reading data from an External file

```
#2A Reading data from a spreadsheet (.xlsx/.xls) file

import pandas as pd
import openpyxl      # required when using Python IDLE

STUDENT_DF = pd.read_excel('StudentData_LAB4.xlsx')

display(STUDENT_DF)
```

```

#2B Reading data from a JSON format file (.json)

import pandas as pd

STUDENT_DF = pd.read_json('StudentData_LAB4.json')

display(STUDENT_DF)
```

	Student Name	Roll No.	Physics	Chemistry	Maths
Student1	Aman	AC-101	78	80	88
Student2	Vishal	AC-102	67	77	95
Student3	Anshu	AC-103	85	85	100
Student4	Shruti	AC-104	96	90	90
Student5	Rahul	AC-105	56	60	78

```

#2C Reading data from a file (.csv)

import pandas as pd

STUDENT_DF = pd.read_csv('StudentData_LAB4.csv')

display(STUDENT_DF)
```

	Student Name	Roll No.	Physics	Chemistry	Maths
0	Aman	AC-101	78	80	88
1	Vishal	AC-102	67	77	95
2	Anshu	AC-103	85	85	100
3	Shruti	AC-104	96	90	90
4	Rahul	AC-105	56	70	78

3. Basic Analysis of Pandas DataFrame

- .shape
- .size
- .ndim
- .head()
- .tail()
- .info
- Adding new column
- Adding new row
- Concatenation of two dataframes

```
# 3A To determine the shape ( no.of ROWs x no. of COLUMNS) in a given data frame

print("(ROWS, COLUMNS) in the given data set is", STUDENT_DF.shape)
```

(ROWS, COLUMNS) in the given data set is (5, 5)

```
# 3B To determine the size ( no.of data elements) in a given data frame

print("No. of elements in the given data set are", STUDENT_DF.size)
```

No. of elements in the given data set are 25

```
# 3C To determine the dimension of a given data frame

print(STUDENT_DF.ndim)
```

2

```
#3D Displaying first few rows of a dataframe

display(STUDENT_DF.head(2))
```

	Student Name	Roll No.	Physics	Chemistry	Maths
0	Aman	AC-101	78	80	88
1	Vishal	AC-102	67	77	95

```
# 3E Displaying last few rows of a dataframe

display(STUDENT_DF.tail(2))
```

```

    Student Name  Roll No.  Physics  Chemistry  Maths
3      Shruti    AC-104      96          90      90
4      Rahul    AC-105      56          70      78

# 3F Displaying information about the data

import pandas as pd

STUDENT_DF = pd.read_csv('StudentData_LAB4.csv')

print(STUDENT_DF.info())
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  ---
0    Student Name    5 non-null     object
1    Roll No.         5 non-null     object
2    Physics          5 non-null     int64
3    Chemistry        5 non-null     int64
4    Maths            5 non-null     int64
dtypes: int64(3), object(2)
memory usage: 328.0+ bytes
None
```

```

# 3G To display column names (Method 1)

display(STUDENT_DF.columns)

Index(['Student Name', 'Roll No.', 'Physics', 'Chemistry', 'Maths'], dtype='object')
```

```

# 3H To display column names (Method 2)

COL = STUDENT_DF.columns # 'COL' will store a list of column names
for X in COL:
    print(X)

Student Name
Roll No.
Physics
Chemistry
Maths
```

```

#3I To add a new column using insert function

import pandas as pd

STUDENT_DF = pd.read_csv('StudentData_LAB4.csv')

STUDENT_DF.insert(5, 'English',[96, 75, 88, 80, 75]) # to insert marks in English for all the five students in a separate column
display(STUDENT_DF)
```

	Student Name	Roll No.	Physics	Chemistry	Maths	English
0	Aman	AC-101	78	80	88	96
1	Vishal	AC-102	67	77	95	75
2	Anshu	AC-103	85	85	100	88
3	Shruti	AC-104	96	90	90	80
4	Rahul	AC-105	56	70	78	75

```

#3J To add a new column without using insert function

import pandas as pd

STUDENT_DF = pd.read_csv('StudentData_LAB4.csv')

English = [96, 75, 88, 80,75]

STUDENT_DF['English']=English    # to insert marks in English for all the four students in a separate column
display(STUDENT_DF)
```

	Student Name	Roll No.	Physics	Chemistry	Maths	English
0	Aman	AC-101	78	80	88	96
1	Vishal	AC-102	67	77	95	75
2	Anshu	AC-103	85	85	100	88
3	Shruti	AC-104	96	90	90	80
4	Rahul	AC-105	56	70	78	75

```

#3K To add a new row using loc function

import pandas as pd

STUDENT_DF = pd.read_csv('StudentData_LAB4.csv')

print('Original Dataset')
display(STUDENT_DF)
```

```
print('Modified Dataset')
STUDENT_DF.loc[5]=['Manisha','AC-106',88,97,100]
# to add details of a new student 'Manisha' in a separate row
display(STUDENT_DF)
```

Original Dataset

	Student Name	Roll No.	Physics	Chemistry	Maths
0	Aman	AC-101	78	80	88
1	Vishal	AC-102	67	77	95
2	Anshu	AC-103	85	85	100
3	Shruti	AC-104	96	90	90
4	Rahul	AC-105	56	70	78

Modified Dataset

	Student Name	Roll No.	Physics	Chemistry	Maths
0	Aman	AC-101	78	80	88
1	Vishal	AC-102	67	77	95
2	Anshu	AC-103	85	85	100
3	Shruti	AC-104	96	90	90
4	Rahul	AC-105	56	70	78
5	Manisha	AC-106	88	97	100

#3L To concatenate two dataframes (Method 1)

```
import pandas as pd

StudentData1 = {
    'Student Name':['Aman', 'Vishal', 'Anshu'],
    'Roll No':['AC-101', 'AC-102', 'AC-103'],
    'Physics':[78,67,85],
    'Chemistry': [80,77,85],
    'Maths':[88,95,100]
}

StudentData2 = {
    'Student Name':['Shruti', 'Rahul'],
    'Roll No':['AC-104', 'AC-105'],
    'Physics':[96,56],
    'Chemistry': [90,62],
    'Maths':[90,78]
}

STUDENT_DF1=pd.DataFrame(StudentData1)
STUDENT_DF2=pd.DataFrame(StudentData2)

STUDENT_DF = pd.concat([STUDENT_DF1,STUDENT_DF2]) # to combine two dataframes STUDENT_DATA1 and STUDENT_DATA2
display(STUDENT_DF)
```

	Student Name	Roll No	Physics	Chemistry	Maths
0	Aman	AC-101	78	80	88
1	Vishal	AC-102	67	77	95
2	Anshu	AC-103	85	85	100
0	Shruti	AC-104	96	90	90
1	Rahul	AC-105	56	62	78

#3M To concatenate two dataframes (Method 2)

```
import pandas as pd

StudentData1 = {
    'Student Name':['Aman', 'Vishal', 'Anshu'],
    'Roll No':['AC-101', 'AC-102', 'AC-103'],
    'Physics':[78,67,85],
    'Chemistry': [80,77,85],
    'Maths':[88,95,100]
}

StudentData2 = {
    'Student Name':['Shruti', 'Rahul'],
    'Roll No':['AC-104', 'AC-105'],
    'Physics':[96,56],
    'Chemistry': [90,62],
    'Maths':[90,78]
}

STUDENT_DF1=pd.DataFrame(StudentData1)
STUDENT_DF2=pd.DataFrame(StudentData2)

STUDENT_DF=STUDENT_DF1.append(STUDENT_DF2) # to combine two dataframes STUDENT_DATA1 and STUDENT_DATA2
STUDENT_DF=STUDENT_DF.reset_index()
display(STUDENT_DF)
```

```
<ipython-input-37-76ae1bfe44d6>:24: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
STUDENT_DF=STUDENT_DF1.append(STUDENT_DF2) # to combine two dataframes STUDENT_DATA1 and STUDENT_DATA2
```

	index	Student Name	Roll No	Physics	Chemistry	Maths
0	0	Aman	AC-101	78	80	88
1	1	Vishal	AC-102	67	77	95
2	2	Anshu	AC-103	85	85	100
3	0	Shruti	AC-104	96	90	90
4	1	Rahul	AC-105	56	62	78

Activity 1

Analyze the given dataset to calculate Total Marks, Percentage and Cumulative Grade of every student as per the following criteria

- Grade O: Percentage >90
- Grade A+:Percentage >80
- Grade A: Percentage >70
- Grade B: Percentage >60
- Grade C: Percentage >50
- Grade D: Percentage >40
- FAIL: Percentage< 40

Specify Total Marks, Percentage and Grade of every student in a separate column

	Student Name	Roll No.	Physics	Chemistry	Maths
0	Aman	AC-101	78	80	88
1	Vishal	AC-102	67	77	95
2	Anshu	AC-103	85	85	100
3	Shruti	AC-104	96	90	90
4	Rahul	AC-105	56	62	78

```
# To evaluate Total Marks, Percentage and Grade and display it in a separate column
```

```
import pandas as pd
```

```
STUDENT_DF = pd.read_csv('StudentData_LAB4.csv')
```

```
TotalMarks = (STUDENT_DF['Physics']+STUDENT_DF['Chemistry']+STUDENT_DF['Maths'])
```

```
STUDENT_DF['Total Marks']=TotalMarks
```

```
STUDENT_DF['Percentage']=TotalMarks/3
```

```
Grade=[]
```


```
for X in STUDENT_DF['Percentage']:
```

```
    if X>90:
        Grade.append('O')
    elif X>80:
        Grade.append('A+')
    elif X>70:
        Grade.append('A')
    elif X>60:
        Grade.append('B')
    elif X>50:
        Grade.append('C')
    elif X>=40:
        Grade.append('D')
    else:
        Grade.append('Fail')
```

```
STUDENT_DF['Grade']=Grade
```

```
STUDENT_DF=STUDENT_DF.set_index('Student Name')
```

```
display(STUDENT_DF)
```

	Roll No.	Physics	Chemistry	Maths	Total Marks	Percentage	Grade	
Student Name								
Aman	AC-101	78	80	88	246	82.000000	A+	
Vishal	AC-102	67	77	95	239	79.666667	A	
Anshu	AC-103	85	85	100	270	90.000000	A+	
Shruti	AC-104	96	90	90	276	92.000000	O	
Rahul	AC-105	56	70	78	204	68.000000	B	

4. GRAPHICAL VISUALIZATION OF DATAFRAMES

4A. LINE PLOT

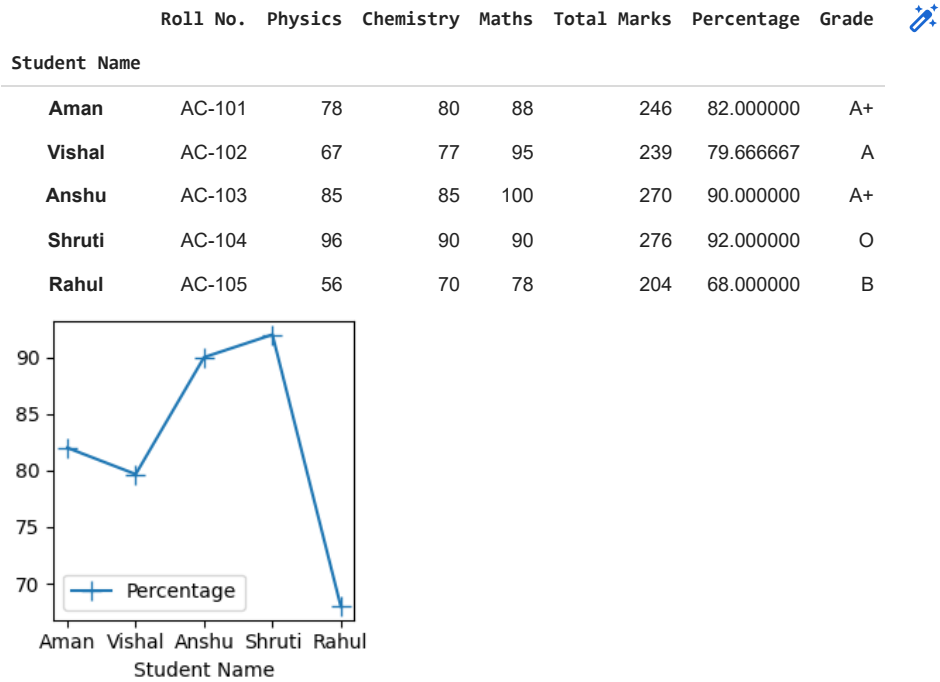
```
import matplotlib.pyplot as plt    # importing matplotlib library

display(STUDENT_DF)
STUDENT_DF['Percentage'].plot.line(x='Student Name',marker='+',ms=10,
                                   figsize=(2.75,2.75))

plt.title='Student Performance'
plt.xlabel='Student Name',
plt.ylabel='Percentage'

plt.legend(loc='lower left')
plt.savefig('Student_DF-LINE.jpg')

plt.show()
```



4B. BAR PLOT

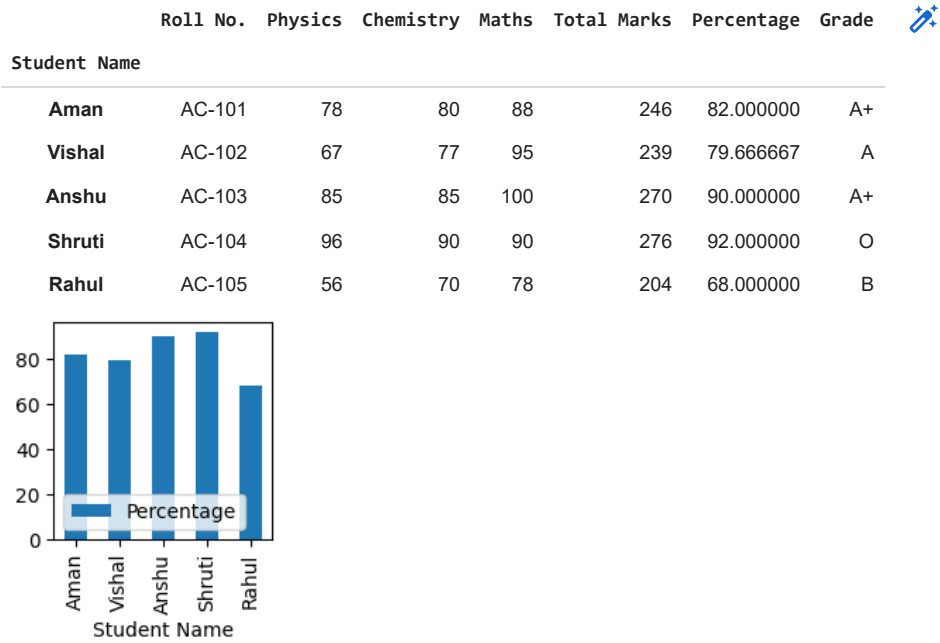
```
import matplotlib.pyplot as plt    # importing matplotlib library

display(STUDENT_DF)
STUDENT_DF['Percentage'].plot.bar(x='Student Name',figsize=(2,2))

plt.xlabel='Student Name',
plt.ylabel='Percentage'
plt.title='Student Performance'
plt.legend(loc='lower left')

plt.savefig('Student_DF-BAR.png')

plt.show()
```



4C BOX PLOT

```
import matplotlib.pyplot as plt    # importing matplotlib library

display(STUDENT_DF)
STUDENT_DF.plot.box(y=['Physics','Chemistry','Maths','Percentage'],
                    figsize=(3.3))
```

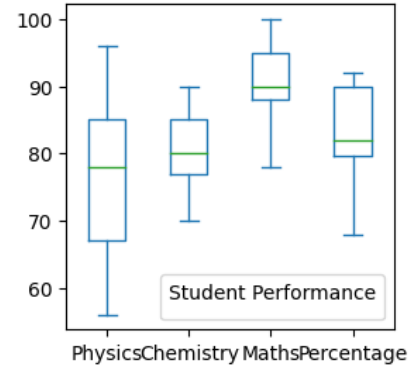


```
plt.xlabel='Student Name',
plt.ylabel='Percentage'
plt.title='Student Performance'
plt.legend(title='Student Performance',loc='lower right')

plt.show()
plt.savefig('Student_DF-BOX.png')
```

	Roll No.	Physics	Chemistry	Maths	Total Marks	Percentage	Grade	
Student Name								
	Aman	AC-101	78	80	88	246	82.000000	A+
	Vishal	AC-102	67	77	95	239	79.666667	A
	Anshu	AC-103	85	85	100	270	90.000000	A+
	Shruti	AC-104	96	90	90	276	92.000000	O
	Rahul	AC-105	56	70	78	204	68.000000	B

WARNING:matplotlib.legend:No artists with labels found to put in legend. Note that artists whose label start with an under



<Figure size 640x480 with 0 Axes>

4D PIE CHART

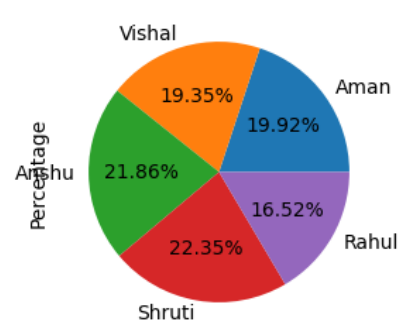
```
import matplotlib.pyplot as plt    # importing matplotlib library

display(STUDENT_DF)

STUDENT_DF['Percentage'].plot.pie(autopct='%1.2f%%',
                                  figsize=(3,3))

plt.savefig('Student_DF-PIE.png')
```

	Roll No.	Physics	Chemistry	Maths	Total Marks	Percentage	Grade	
Student Name								
	Aman	AC-101	78	80	88	246	82.000000	A+
	Vishal	AC-102	67	77	95	239	79.666667	A
	Anshu	AC-103	85	85	100	270	90.000000	A+
	Shruti	AC-104	96	90	90	276	92.000000	O
	Rahul	AC-105	56	70	78	204	68.000000	B




4E HISTOGRAM

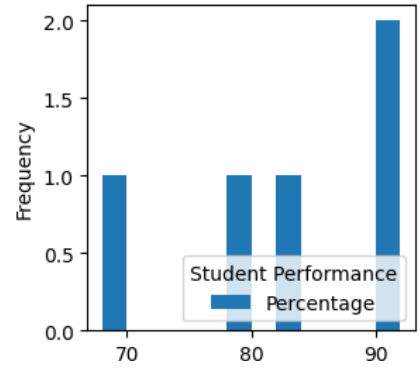
```
import matplotlib.pyplot as plt    # importing matplotlib library

display(STUDENT_DF)
STUDENT_DF['Percentage'].plot.hist(bins=12,figsize=(3,3))

plt.xlabel='Student Name',
plt.ylabel='Percentage'
plt.title='Student Performance'
plt.legend(title='Student Performance',loc='lower right')

plt.show()
plt.savefig('Student_DF-HIST.png')
```

	Roll No.	Physics	Chemistry	Maths	Total Marks	Percentage	Grade	
Student Name								
	Aman	AC-101	78	80	88	246	82.000000	A+
	Vishal	AC-102	67	77	95	239	79.666667	A
	Anshu	AC-103	85	85	100	270	90.000000	A+
	Shruti	AC-104	96	90	90	276	92.000000	O
	Rahul	AC-105	56	70	78	204	68.000000	B



5. SUBPLOT function to display multiple plots

```
import matplotlib.pyplot as plt #importing matplotlib library
```

```
display(STUDENT_DF)
```

```
plt.figure(figsize= (10,15)) # To set PLOT AREA size
```

```
''' Setting Plot size to 15" WIDE X 10" LENGTH
    To accomodate 6 PLOTS arranged in 2 ROWS X 3 COLUMNS
    Each plot of size 5" WIDE X 5" LENGTH'''
```

```
plt.subplot(3,2,1) # SUBPLOT 1
STUDENT_DF['Percentage'].plot.line(ylabel='Percentage')
```

```
plt.xlabel='Student Name',
plt.ylabel='Percentage'
plt.title='Student Performance'
plt.legend(title='Student Performance',loc='lower right')
```

```
plt.subplot(3,2,2) # SUBPLOT 2
STUDENT_DF['Percentage'].plot.bar()
plt.xlabel='Student Name',
plt.ylabel='Percentage'
plt.title='Student Performance'
plt.legend(title='Student Performance',loc='lower right')
```

```
plt.subplot(3,2,3) # SUBPLOT 3
STUDENT_DF['Percentage'].plot.hist(bins=12)
plt.xlabel='Student Name',
plt.ylabel='Percentage'
plt.title='Student Performance'
plt.legend(title='Student Performance',loc='lower right')
```

```
plt.subplot(3,2,4) # SUBPLOT 4
STUDENT_DF['Percentage'].plot.area( )

plt.xlabel='Student Name',
plt.ylabel='Percentage'
plt.title='Student Performance'
plt.legend(title='Student Performance',loc='lower right')
```

```
plt.subplot(3,2,5) # SUBPLOT 5
STUDENT_DF['Percentage'].plot.pie(autopct='%1.2f%%')
```

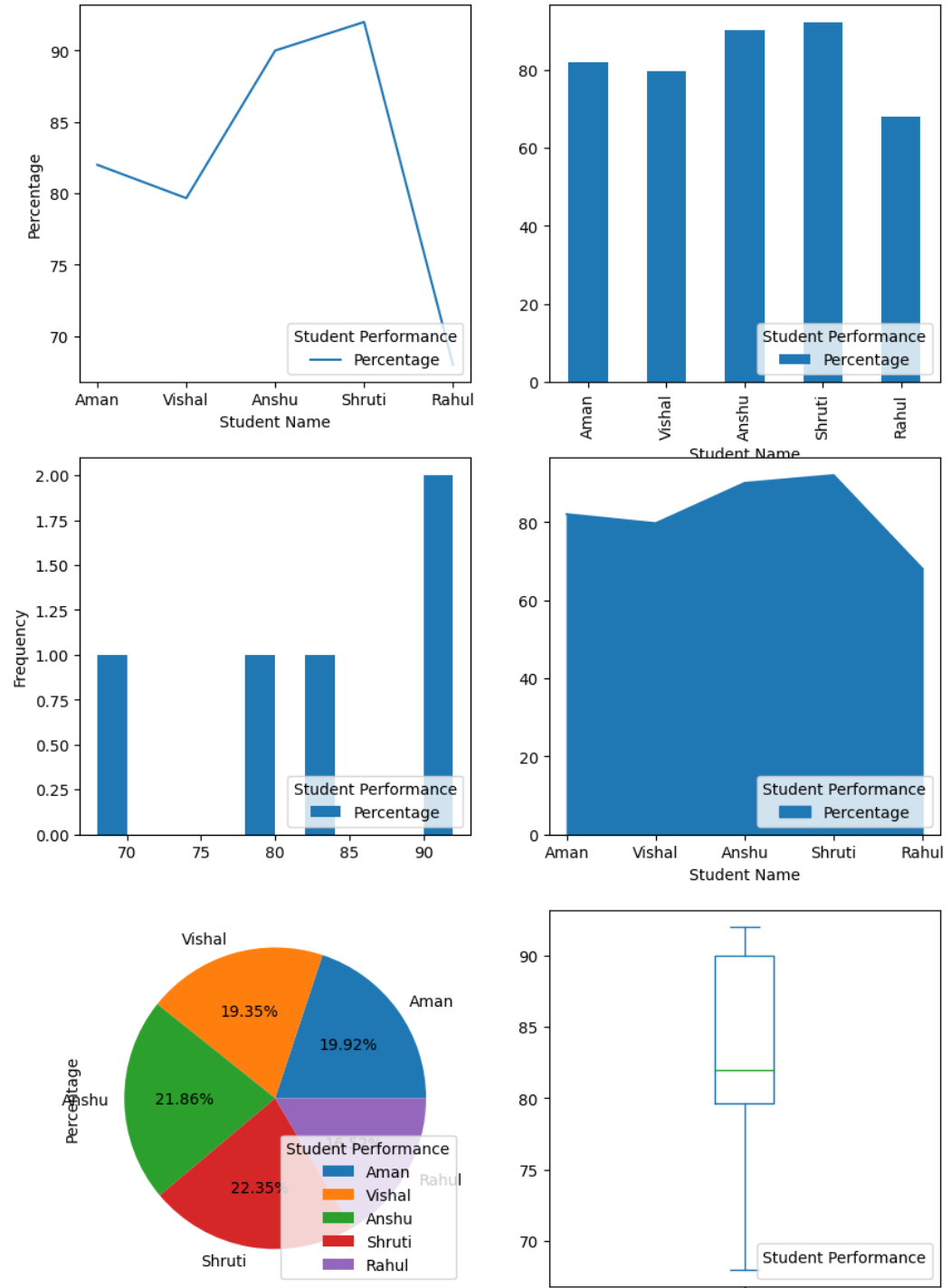
```
plt.xlabel='Student Name',
plt.ylabel='Percentage'
plt.title='Student Performance'
plt.legend(title='Student Performance',loc='lower right')
```

```
plt.subplot(3,2,6) # SUBPLOT 6
STUDENT_DF['Percentage'].plot.box( )
```

```
plt.xlabel='Student Name',
plt.ylabel='Percentage'
plt.title='Student Performance'
plt.legend(title='Student Performance',loc='lower right')
```

	Roll No.	Physics	Chemistry	Maths	Total Marks	Percentage	Grade
Student Name							
Aman	AC-101	78	80	88	246	82.000000	A+
Vishal	AC-102	67	77	95	239	79.666667	A
Anshu	AC-103	85	85	100	270	90.000000	A+
Shruti	AC-104	96	90	90	276	92.000000	O
Rahul	AC-105	56	70	78	204	68.000000	B

WARNING:matplotlib.legend.No artists with labels found to put in legend. Note that artists whose <matplotlib.legend.Legend at 0x7efd59315700>



ACTIVITY 2

Analysis of Time-Series Data

Double-click (or enter) to edit

	Product1	Product2
2023-04-10	100	100
2023-04-11	70	70
2023-04-12	82	82
2023-04-13	125	125
2023-04-14	65	65
2023-04-15	30	30
2023-04-16	50	50

The above table shows sales record of two products, Product 1 and Product 2, monitored across one week from 10-04-2023 to 16-04-2023

1. Create a DataFrame to store detaile of sales of Product 1 and Product

2. Compare Sales of Product 1 and Product 2 through:
- Line plot (scatter plot)

◦ Bar plot

◦ Area plot
3. Analyze sales of Product 1 across one week through a Pie Chart

```
import pandas as pd

Sales_Product1=[100, 70,82,125,65,30,50]
Sales_Product2=[20, 78,65,100,50,80,55]

DATE_SERIES= pd.date_range(start='2023/04/10',
                             end='2023/04/16',
                             freq='D')

display(DATE_SERIES)

DATE_SERIES=pd.Series(DATE_SERIES)

display(DATE_SERIES)
```

```

DatetimeIndex(['2023-04-10', '2023-04-11', '2023-04-12', '2023-04-13',
               '2023-04-14', '2023-04-15', '2023-04-16'],
              dtype='datetime64[ns]', freq='D')
0    2023-04-10
1    2023-04-11
2    2023-04-12
3    2023-04-13
4    2023-04-14
5    2023-04-15
6    2023-04-16
dtype: datetime64[ns]
```

```
import pandas as pd

Sales_Product1=[100, 70,82,125,65,30,50]
Sales_Product2=[20, 78,65,100,50,80,55]

DATE_SERIES= pd.date_range(start='2023/04/10',
                             end='2023/04/16',
                             freq='D')

SALES ={'Product1':Sales_Product1, 'Product2':Sales_Product2}

SALES_RECORD = pd.DataFrame(SALES, index=DATE_SERIES)
SALES_RECORD.index.name='Date '

display(SALES_RECORD)

SALES_RECORD.to_csv('SALES_RECORD.csv')
```

	Product1	Product2
Date		
2023-04-10	100	20
2023-04-11	70	78
2023-04-12	82	65
2023-04-13	125	100
2023-04-14	65	50
2023-04-15	30	80
2023-04-16	50	55

```
import matplotlib.pyplot as plt    # importing matplotlib library

display(SALES_RECORD)

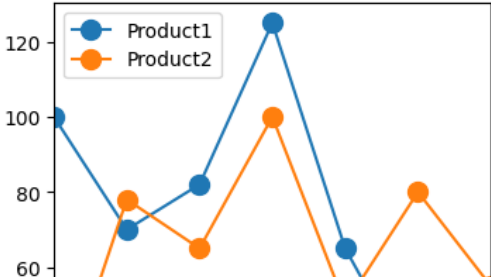
SALES_RECORD.plot.line(y=[ 'Product1', 'Product2'],marker='o',ms=10,
                        figsize=(4,4))

plt.title = 'SALES'
plt.xlabel='Date',
plt.ylabel='Product Sales'

plt.legend(loc='upper left')
plt.savefig('SALES_RECORD-LINE.png')

plt.show()
```

	Product1	Product2
Date		
2023-04-10	100	20
2023-04-11	70	78
2023-04-12	82	65
2023-04-13	125	100
2023-04-14	65	50
2023-04-15	30	80
2023-04-16	50	55



```
import matplotlib.pyplot as plt    # importing matplotlib library

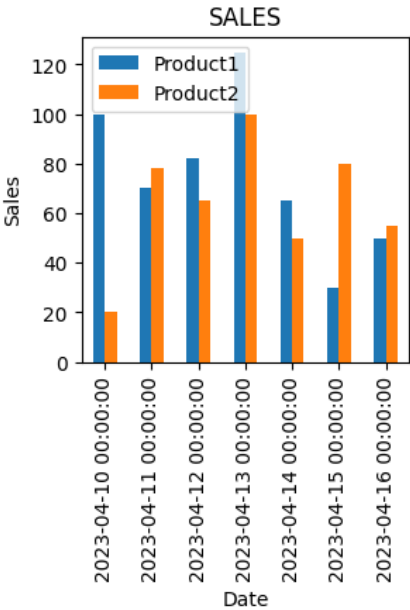
display(SALES_RECORD)

SALES_RECORD.plot.bar(y=[ 'Product1', 'Product2'],figsize=(3,3),
                      xlabel='Date', ylabel='Sales',title='SALES')

plt.legend(loc='upper left')
plt.savefig( 'SALES_RECORD-BAR.png')

plt.show()
plt.savefig( 'SALES_RECORD-BAR.png')
```

	Product1	Product2
Date		
2023-04-10	100	20
2023-04-11	70	78
2023-04-12	82	65
2023-04-13	125	100
2023-04-14	65	50
2023-04-15	30	80
2023-04-16	50	55



<Figure size 640x480 with 0 Axes>

```
import matplotlib.pyplot as plt    # importing matplotlib library

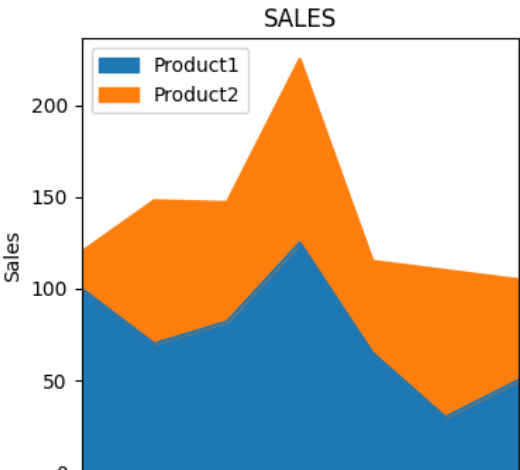
display(SALES_RECORD)

SALES_RECORD.plot.area(y=[ 'Product1', 'Product2'],figsize=(4,4),
                       xlabel='Date', ylabel='Sales',title='SALES')

plt.legend(loc='upper left')
plt.savefig( 'SALES_RECORD-AREA.png')

plt.show()
```

	Product1	Product2
Date		
2023-04-10	100	20
2023-04-11	70	78
2023-04-12	82	65
2023-04-13	125	100
2023-04-14	65	50
2023-04-15	30	80
2023-04-16	50	55



```
import matplotlib.pyplot as plt    # importing matplotlib library

display(SALES_RECORD)

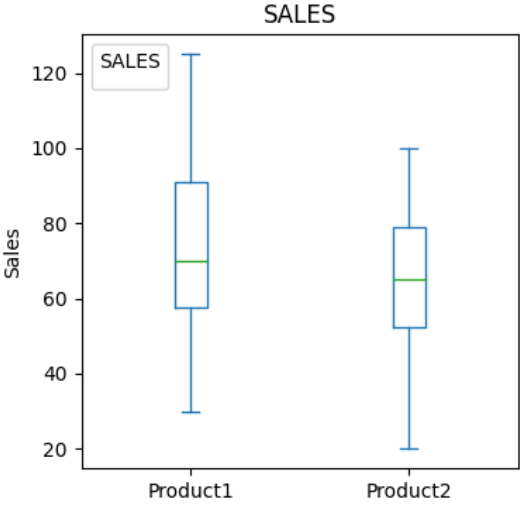
SALES_RECORD.plot.box(y=[ 'Product1', 'Product2'],figsize=(4,4),
                      xlabel='Date', ylabel='Sales',title='SALES')

plt.legend(title='SALES',loc='upper left')
plt.savefig( 'SALES_RECORD-BOX.png')

plt.show()
```

	Product1	Product2
Date		
2023-04-10	100	20
2023-04-11	70	78
2023-04-12	82	65
2023-04-13	125	100
2023-04-14	65	50
2023-04-15	30	80
2023-04-16	50	55

WARNING:matplotlib.legend:No artists with labels found to put in legend. Note that artists whose label start with an under



```
import matplotlib.pyplot as plt    # importing matplotlib library

display(SALES_RECORD)

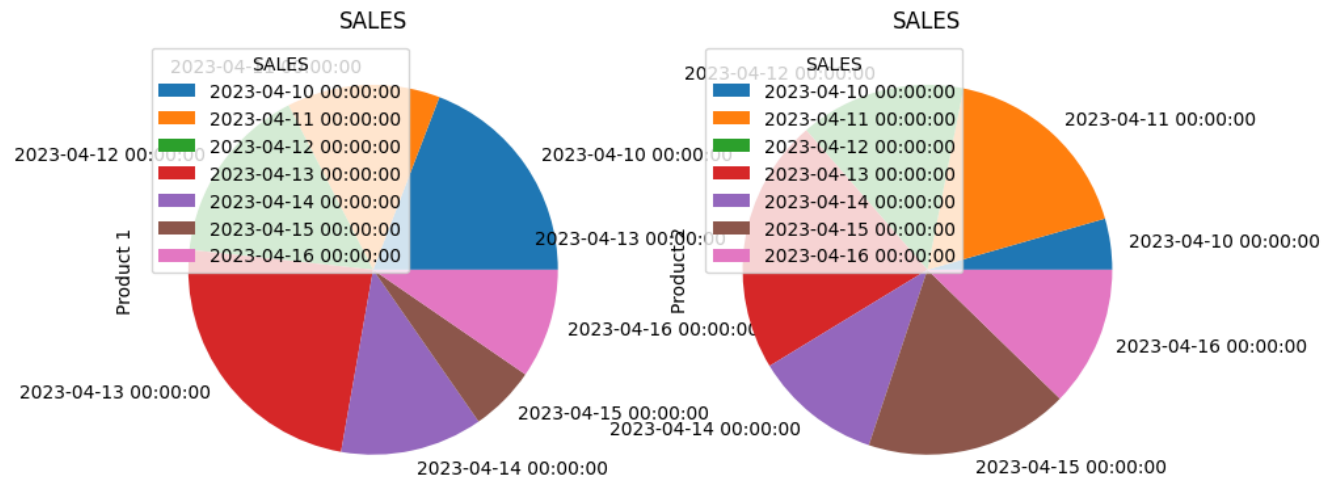
plt.figure(figsize= (10,5)) # To set PLOT AREA size

plt.subplot(1,2,1)
SALES_RECORD['Product1'].plot.pie(xlabel='Date', ylabel='Product 1',title='SALES')
plt.legend(title='SALES',loc='upper left')

plt.subplot(1,2,2)
SALES_RECORD['Product2'].plot.pie(xlabel='Date', ylabel='Product 2',title='SALES')
plt.legend(title='SALES',loc='upper left')
```

```
plt.savefig('SALES_RECORD-PIE.png')
plt.show()
```

	Product1	Product2
Date		
2023-04-10	100	20
2023-04-11	70	78
2023-04-12	82	65
2023-04-13	125	100
2023-04-14	65	50
2023-04-15	30	80
2023-04-16	50	55



✓ 3s completed at 12:54 AM

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.

