A Journal
For
Data Analytics Using Python

B. Tech (IT)
SEM VI
IT129 SHAH HETVI JATILKUMAR

21ITUON093

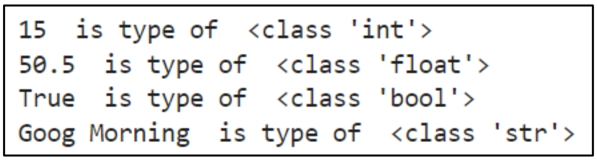
Faculty of Technology
Dharmsinh Desai
University Nadiad

EXPERIMENT – 1

AIM: Introduction to python programming for data analytics.
Tools/Apparatus: Anaconda Python/Spyder IDE

- **Varibles:**

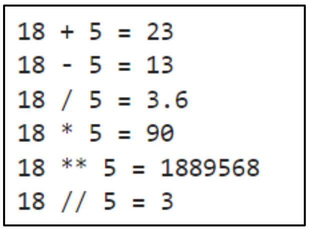
```
a = 15
b = 50.5
c = True
d = "Goog Morning"
print(a," is type of ",type(a))
print(b," is type of ",type(b))
print(c," is type of ",type(c))
print(d," is type of ",type(d))
```



```
15  is type of  <class 'int'>
50.5  is type of  <class 'float'>
True  is type of  <class 'bool'>
Goog Morning  is type of  <class 'str'>
```

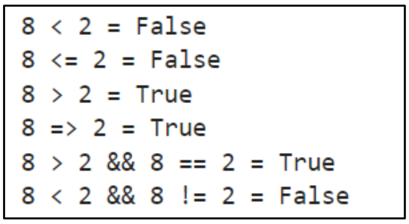
- **Mathematical and Logical Operators**:

```
x = 18
y = 5

print(x,"+",y,"=", x+y)
print(x,"-",y,"=", x-y)
print(x,"/",y,"=", x/y)
print(x,"*",y,"=", x*y)
print(x,"**",y,"=", x**y)
print(x,"//",y,"=", x//y)
```



```
18 + 5 = 23
18 - 5 = 13
18 / 5 = 3.6
18 * 5 = 90
18 ** 5 = 1889568
18 // 5 = 3
```

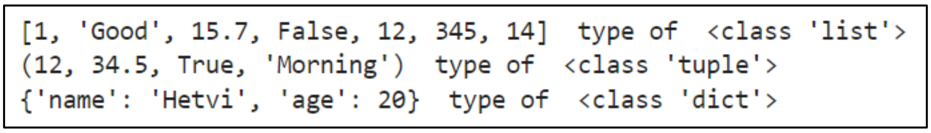
```
x = 8
y = 2
z = 2

print(x,"<",y,"=",x<y)
print(x,"<=",y,"=",x<=y)
print(x,">",y,"=",x>y)
print(x,"=>",y,"=",x>=y)
print(x,">",y,"&&",x,"==",y,"=",x>y and y==z)
print(x,"<",y,"&&",x,"!=",y,"=",x<y or y!=z)
```



```
8 < 2 = False
8 <= 2 = False
8 > 2 = True
8 => 2 = True
8 > 2 && 8 == 2 = True
8 < 2 && 8 != 2 = False
```

- Composite Data Types:

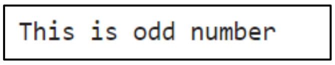
```
list_ = [1,"Good",15.7,False,12,345,14] #mutable
tupple_ = (12,34.5,True,"Morning") #immutable
dictionary_ = {"name" : "Hetvi" , "age" : 20}

print(list_," type of ",type(list_))
print(tupple_," type of ",type(tupple_))
print(dictionary_," type of ",type(dictionary_))
```



```
[1, 'Good', 15.7, False, 12, 345, 14]  type of  <class 'list'>
(12, 34.5, True, 'Morning')  type of  <class 'tuple'>
{'name': 'Hetvi', 'age': 20}  type of  <class 'dict'>
```

- Conditions:

```
a = 17
if(a%2==0):
   print("This is even number")
else:
   print("This is odd number")
```

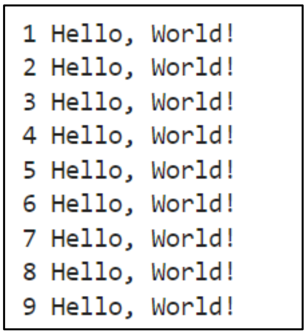


```
This is odd number
```

- Loops:

- While loop:

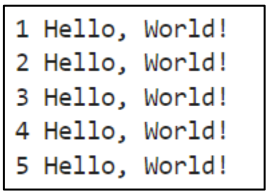
```
i = 1
while(i<10):
   print(i,"Hello, World!")
   i=i+1
```



```
1 Hello, World!
2 Hello, World!
3 Hello, World!
4 Hello, World!
5 Hello, World!
6 Hello, World!
7 Hello, World!
8 Hello, World!
9 Hello, World!
```

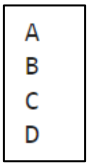
- For in loop:

```
for i in range(1,6):
   print(i,"Hello, World!")
```



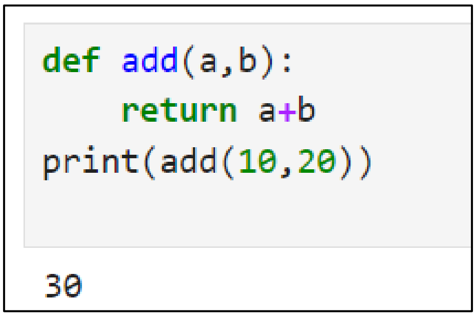
```
1 Hello, World!
2 Hello, World!
3 Hello, World!
4 Hello, World!
5 Hello, World!
```

```
lst = ["A","B","C","D"]
for ele in lst:
    print(ele)
```



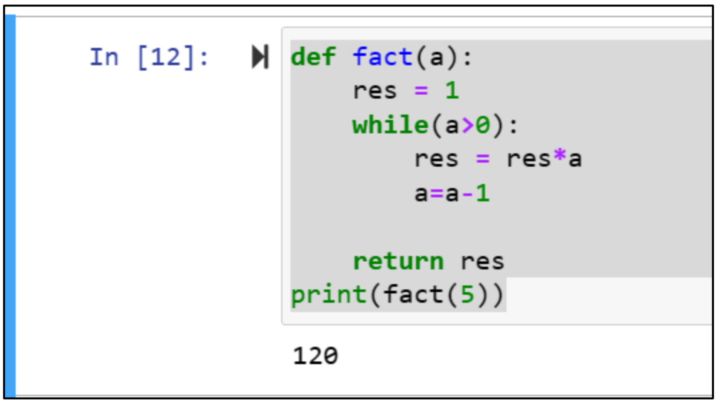
```
A
B
C
D
```

- Functions:



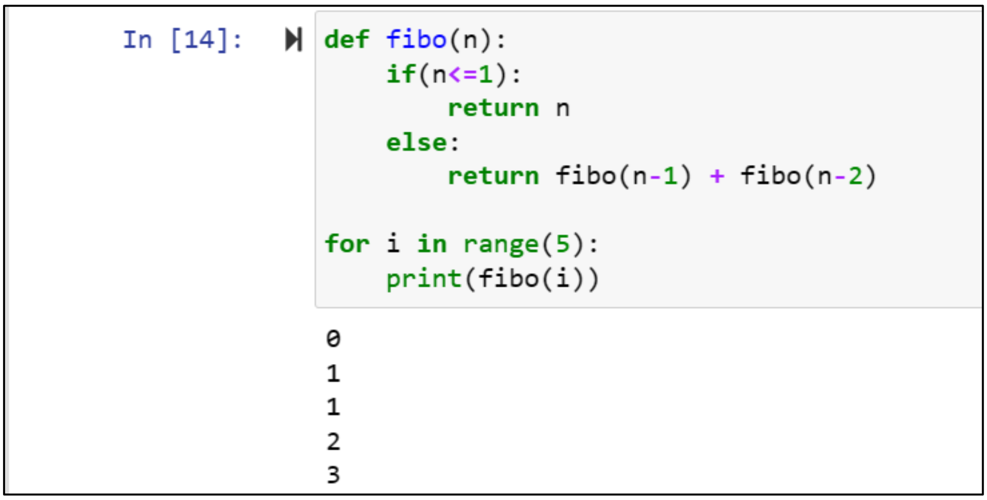
```
def add(a,b):
    return a+b
print(add(10,20))
```

```
30
```



```
In [12]:
def fact(a):
    res = 1
    while(a>0):
        res = res*a
        a=a-1

    return res
print(fact(5))
```

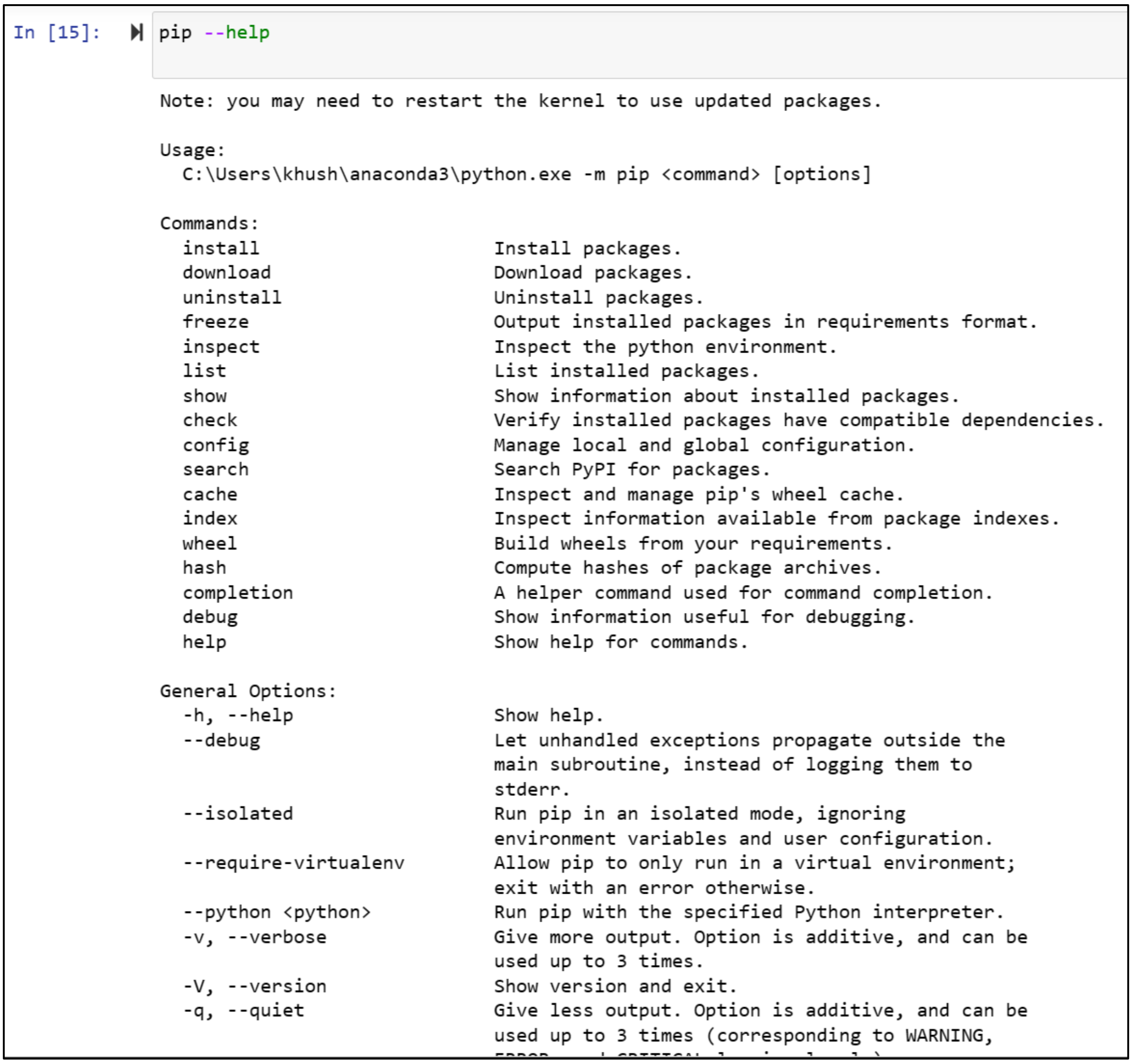
```
120
```



```
In [14]:
def fibo(n):
    if(n<=1):
        return n
    else:
        return fibo(n-1) + fibo(n-2)

for i in range(5):
    print(fibo(i))
```

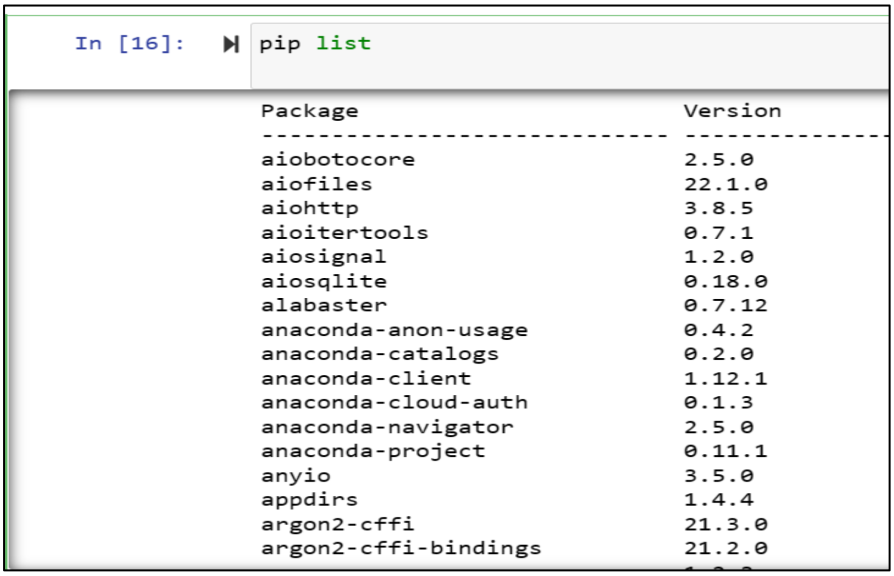
```
0
1
1
2
3
```

- Package Management Using Pip (Python install package):

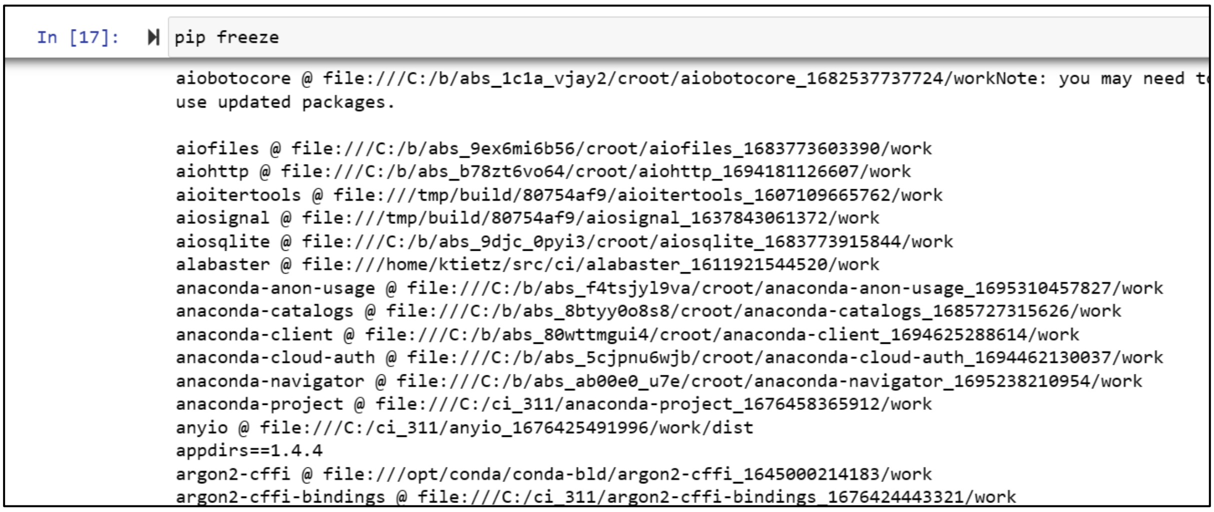


```
In [15]:  pip --help
```

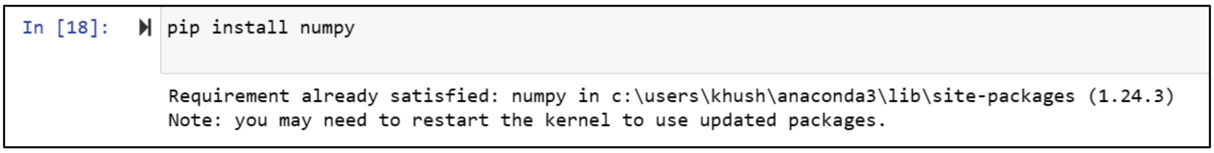
```
Note: you may need to restart the kernel to use updated packages.

Usage:
  C:\Users\khush\anaconda3\python.exe -m pip <command> [options]

Commands:
  install                     Install packages.
  download                    Download packages.
  uninstall                   Uninstall packages.
  freeze                      Output installed packages in requirements format.
  inspect                     Inspect the python environment.
  list                        List installed packages.
  show                        Show information about installed packages.
  check                       Verify installed packages have compatible dependencies.
  config                      Manage local and global configuration.
  search                      Search PyPI for packages.
  cache                       Inspect and manage pip's wheel cache.
  index                       Inspect information available from package indexes.
  wheel                       Build wheels from your requirements.
  hash                        Compute hashes of package archives.
  completion                  A helper command used for command completion.
  debug                       Show information useful for debugging.
  help                        Show help for commands.

General Options:
  -h, --help                  Show help.
  --debug                     Let unhandled exceptions propagate outside the
                              main subroutine, instead of logging them to
                              stderr.
  --isolated                  Run pip in an isolated mode, ignoring
                              environment variables and user configuration.
  --require-virtualenv        Allow pip to only run in a virtual environment;
                              exit with an error otherwise.
  --python <python>           Run pip with the specified Python interpreter.
  -v, --verbose               Give more output. Option is additive, and can be
                              used up to 3 times.
  -V, --version               Show version and exit.
  -q, --quiet                 Give less output. Option is additive, and can be
                              used up to 3 times (corresponding to WARNING,
```



```
In [16]:  pip list

          Package                        Version
          ------------------------------ ---------------
          aiobotocore                    2.5.0
          aiofiles                       22.1.0
          aiohttp                        3.8.5
          aioitertools                   0.7.1
          aiosignal                      1.2.0
          aiosqlite                      0.18.0
          alabaster                      0.7.12
          anaconda-anon-usage            0.4.2
          anaconda-catalogs              0.2.0
          anaconda-client                1.12.1
          anaconda-cloud-auth            0.1.3
          anaconda-navigator             2.5.0
          anaconda-project               0.11.1
          anyio                          3.5.0
          appdirs                        1.4.4
          argon2-cffi                    21.3.0
          argon2-cffi-bindings           21.2.0
```



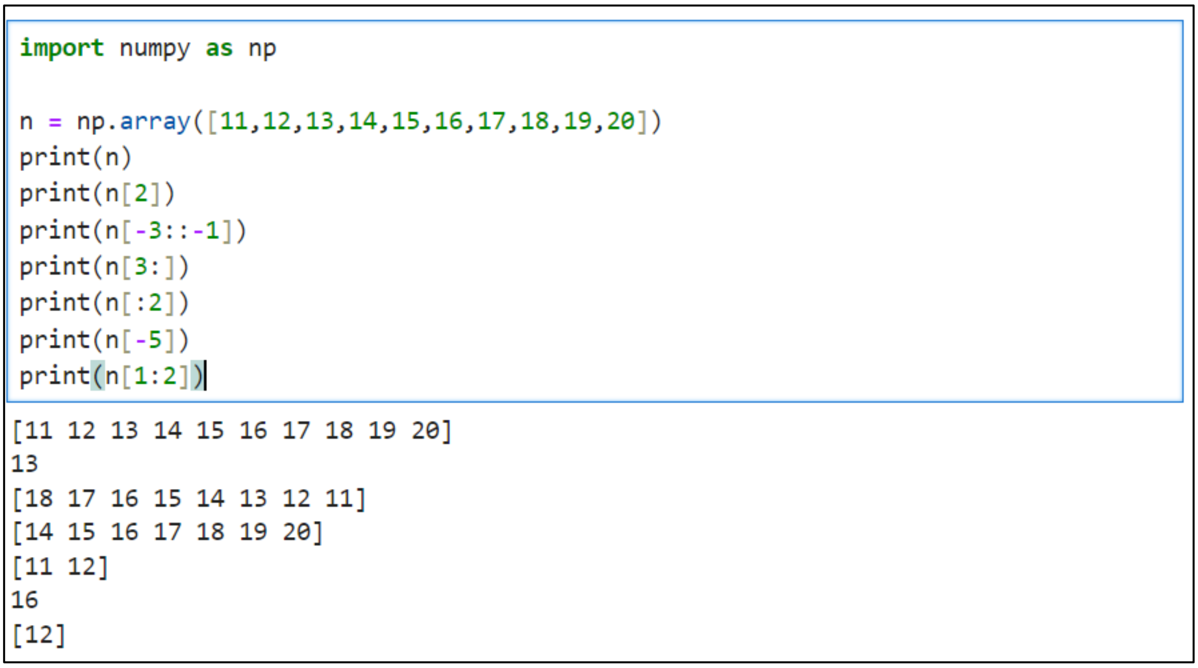
```
In [17]:  pip freeze

          aiobotocore @ file:///C:/b/abs_1c1a_vjay2/croot/aiobotocore_1682537737724/workNote: you may need t
          use updated packages.

          aiofiles @ file:///C:/b/abs_9ex6mi6b56/croot/aiofiles_1683773603390/work
          aiohttp @ file:///C:/b/abs_b78zt6vo64/croot/aiohttp_1694181126607/work
          aioitertools @ file:///tmp/build/80754af9/aioitertools_1607109665762/work
          aiosignal @ file:///tmp/build/80754af9/aiosignal_1637843061372/work
          aiosqlite @ file:///C:/b/abs_9djc_0pyi3/croot/aiosqlite_1683773915844/work
          alabaster @ file:///home/ktietz/src/ci/alabaster_1611921544520/work
          anaconda-anon-usage @ file:///C:/b/abs_f4tsjyl9va/croot/anaconda-anon-usage_1695310457827/work
          anaconda-catalogs @ file:///C:/b/abs_8btyy0o8s8/croot/anaconda-catalogs_1685727315626/work
          anaconda-client @ file:///C:/b/abs_80wttmgui4/croot/anaconda-client_1694625288614/work
          anaconda-cloud-auth @ file:///C:/b/abs_5cjpnu6wjb/croot/anaconda-cloud-auth_1694462130037/work
          anaconda-navigator @ file:///C:/b/abs_ab00e0_u7e/croot/anaconda-navigator_1695238210954/work
          anaconda-project @ file:///C:/ci_311/anaconda-project_1676458365912/work
          anyio @ file:///C:/ci_311/anyio_1676425491996/work/dist
          appdirs==1.4.4
          argon2-cffi @ file:///opt/conda/conda-bld/argon2-cffi_1645000214183/work
          argon2-cffi-bindings @ file:///C:/ci_311/argon2-cffi-bindings_1676424443321/work
```



```
In [18]:  pip install numpy

          Requirement already satisfied: numpy in c:\users\khush\anaconda3\lib\site-packages (1.24.3)
          Note: you may need to restart the kernel to use updated packages.
```

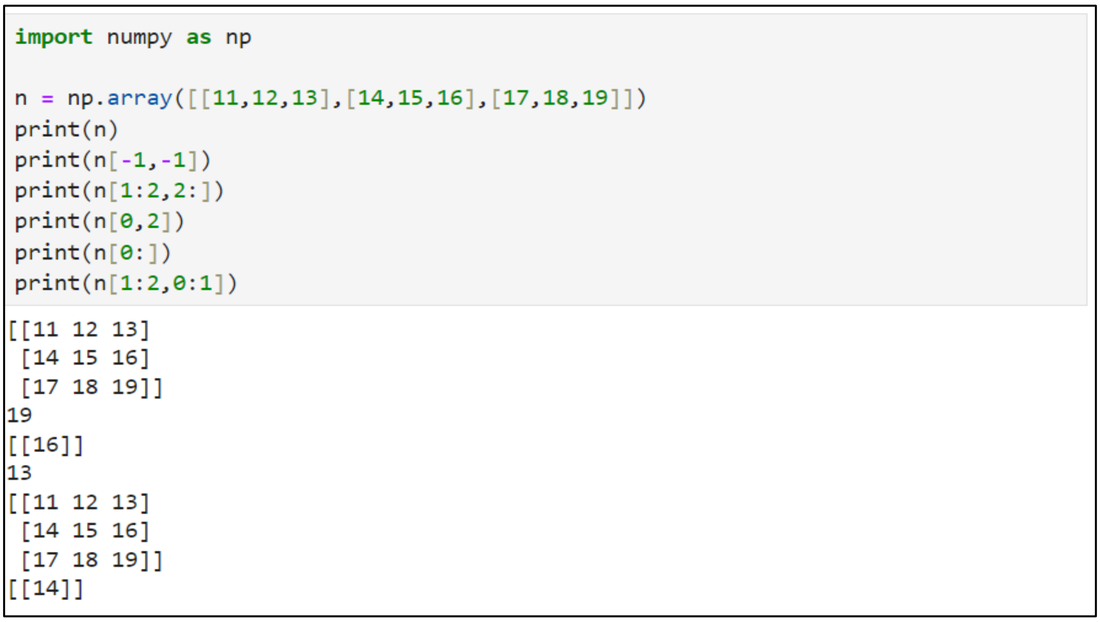
- Handling Multi-dimensional data and element-wise operator using Numpy

  - Import Numpy in the source code, creating data vector and accessing elements of the vector



```
import numpy as np

n = np.array([11,12,13,14,15,16,17,18,19,20])
print(n)
print(n[2])
print(n[-3::-1])
print(n[3:])
print(n[:2])
print(n[-5])
print(n[1:2])
```

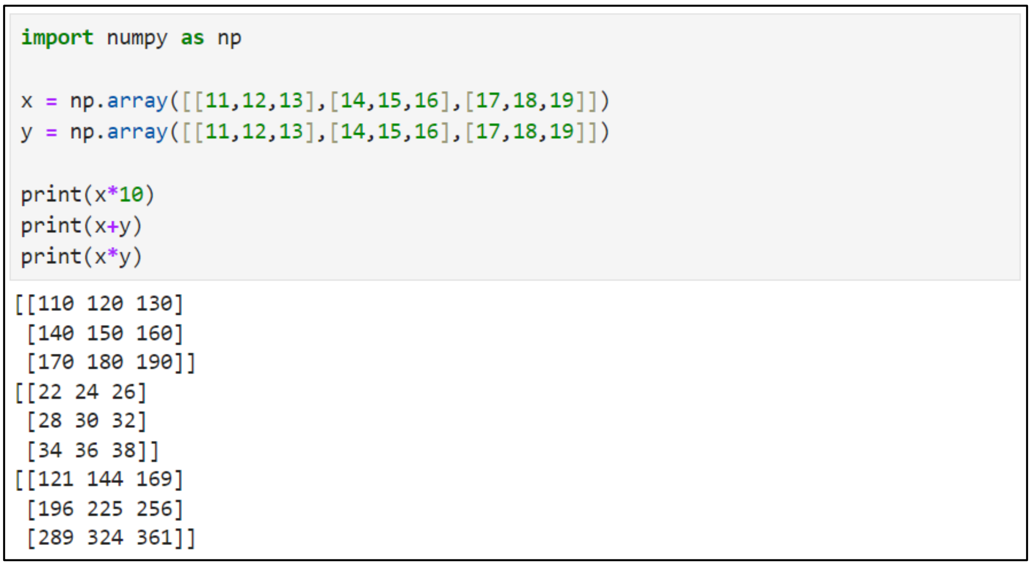
```
[11 12 13 14 15 16 17 18 19 20]
13
[18 17 16 15 14 13 12 11]
[14 15 16 17 18 19 20]
[11 12]
16
[12]
```



```
import numpy as np

n = np.array([[11,12,13],[14,15,16],[17,18,19]])
print(n)
print(n[-1,-1])
print(n[1:2,2:])
print(n[0,2])
print(n[0:])
print(n[1:2,0:1])
```

```
[[11 12 13]
 [14 15 16]
 [17 18 19]]
19
[[16]]
13
[[11 12 13]
 [14 15 16]
 [17 18 19]]
[[14]]
```

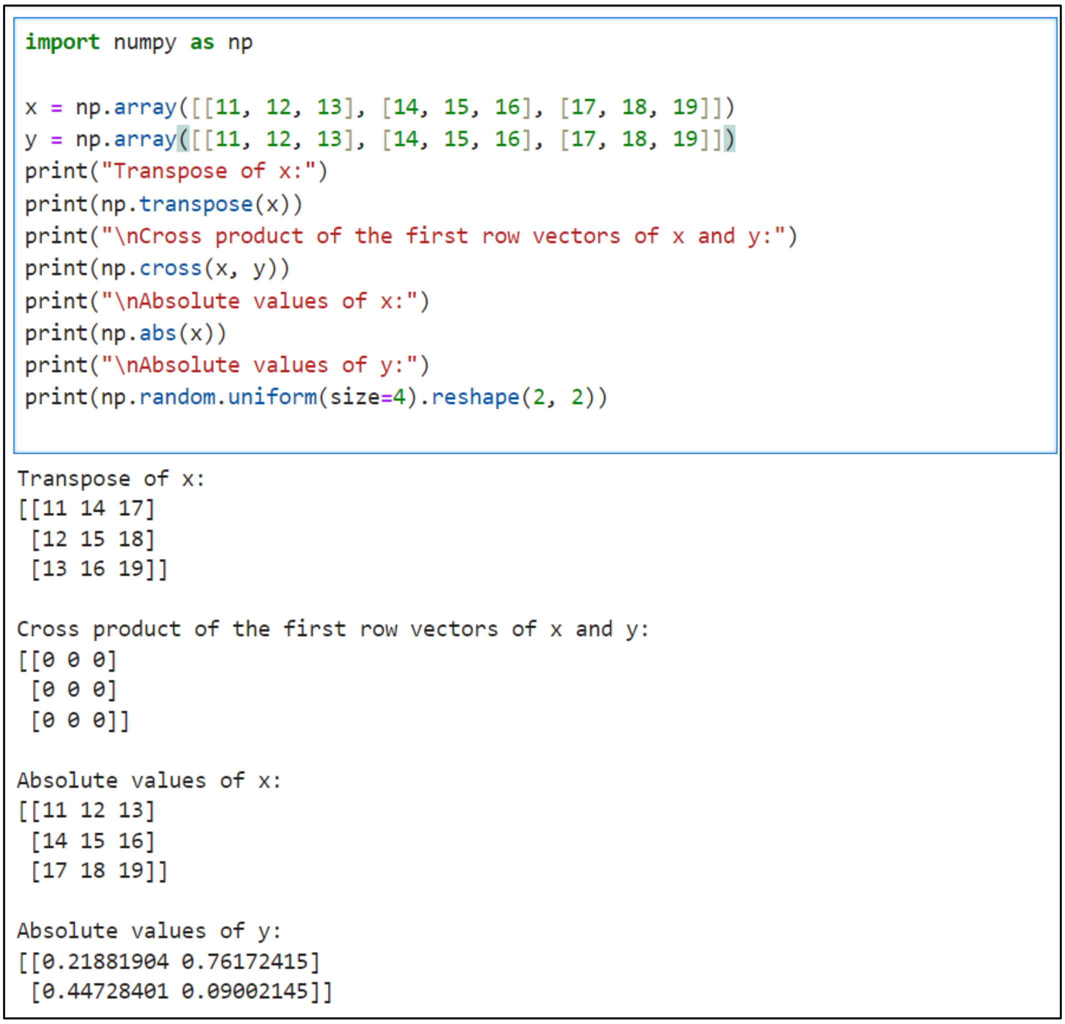
- Element wise operators



```
import numpy as np

x = np.array([[11,12,13],[14,15,16],[17,18,19]])
y = np.array([[11,12,13],[14,15,16],[17,18,19]])

print(x*10)
print(x+y)
print(x*y)
```

```
[[110 120 130]
 [140 150 160]
 [170 180 190]]
[[22 24 26]
 [28 30 32]
 [34 36 38]]
[[121 144 169]
 [196 225 256]
 [289 324 361]]
```

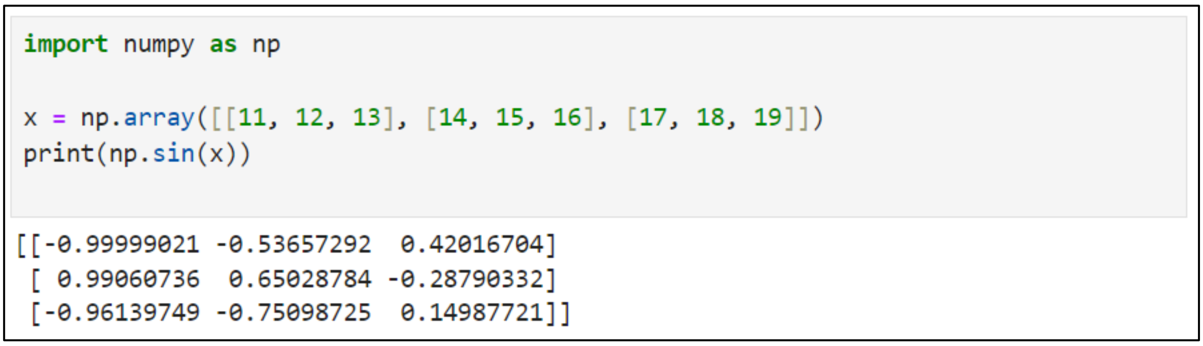
- Matrix Operators



```
import numpy as np

x = np.array([[11, 12, 13], [14, 15, 16], [17, 18, 19]])
y = np.array([[11, 12, 13], [14, 15, 16], [17, 18, 19]])
print("Transpose of x:")
print(np.transpose(x))
print("\nCross product of the first row vectors of x and y:")
print(np.cross(x, y))
print("\nAbsolute values of x:")
print(np.abs(x))
print("\nAbsolute values of y:")
print(np.random.uniform(size=4).reshape(2, 2))
```

```
Transpose of x:
[[11 14 17]
 [12 15 18]
 [13 16 19]]

Cross product of the first row vectors of x and y:
[[0 0 0]
 [0 0 0]
 [0 0 0]]

Absolute values of x:
[[11 12 13]
 [14 15 16]
 [17 18 19]]

Absolute values of y:
[[0.21881904 0.76172415]
 [0.44728401 0.09002145]]
```

- Mathematical Operator



```python
import numpy as np

x = np.array([[11, 12, 13], [14, 15, 16], [17, 18, 19]])
print(np.sin(x))
```

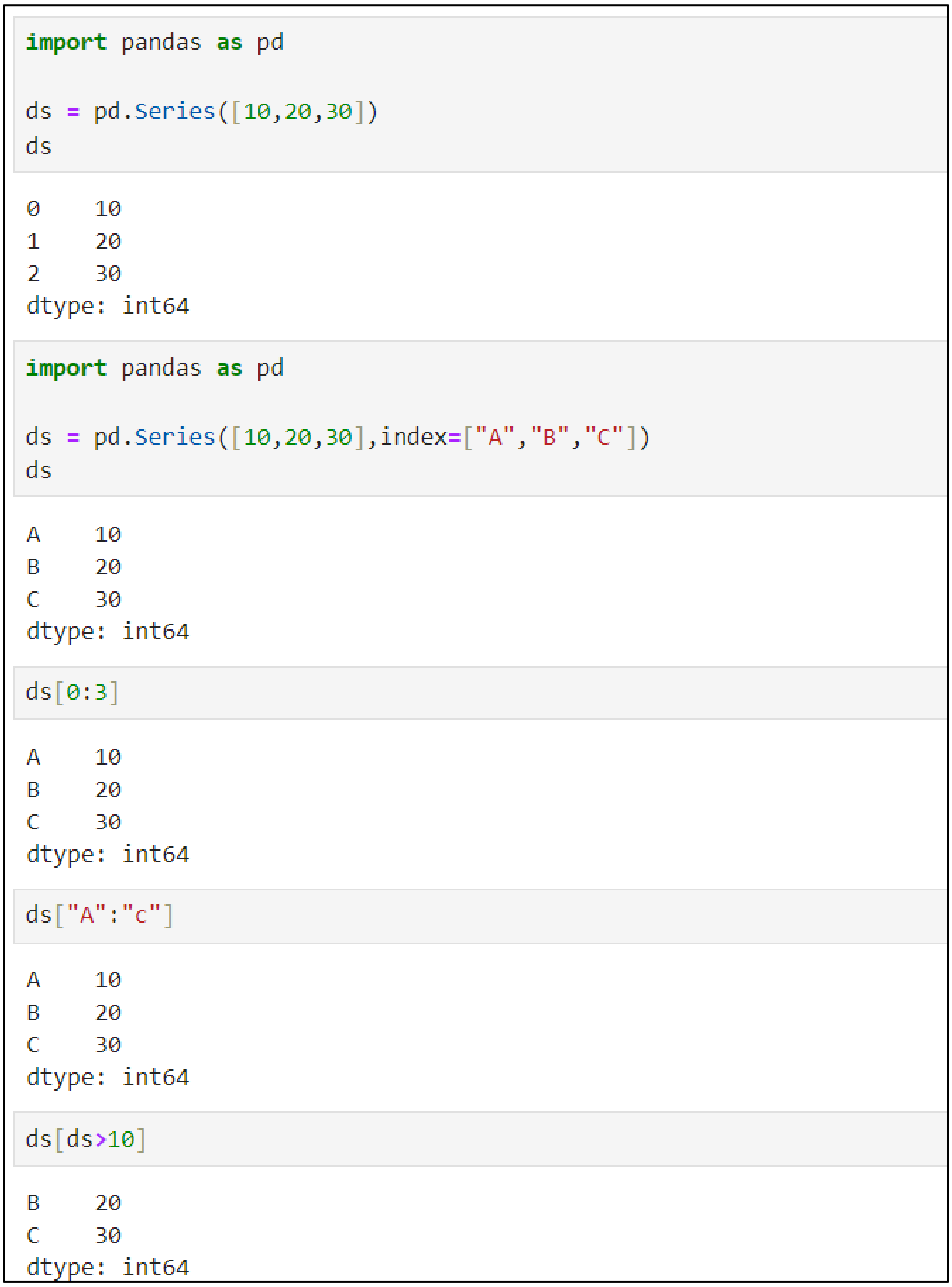
```
[[-0.99999021 -0.53657292  0.42016704]
 [ 0.99060736  0.65028784 -0.28790332]
 [-0.96139749 -0.75098725  0.14987721]]
```

EXPERIMENT – 2

AIM: To perform creating and accessing DataVector, MathsOperations, DataFramecreation, reading & writing, Handling DataFrame using Pandas and Numpy.

Tools/Apparatus: Anaconda Python/ Spyder IDE

- Import pandas, create data series



```
import pandas as pd

ds = pd.Series([10,20,30])
ds
```

```
0    10
1    20
2    30
dtype: int64
```

```
import pandas as pd

ds = pd.Series([10,20,30],index=["A","B","C"])
ds
```

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

```
ds[0:3]
```

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

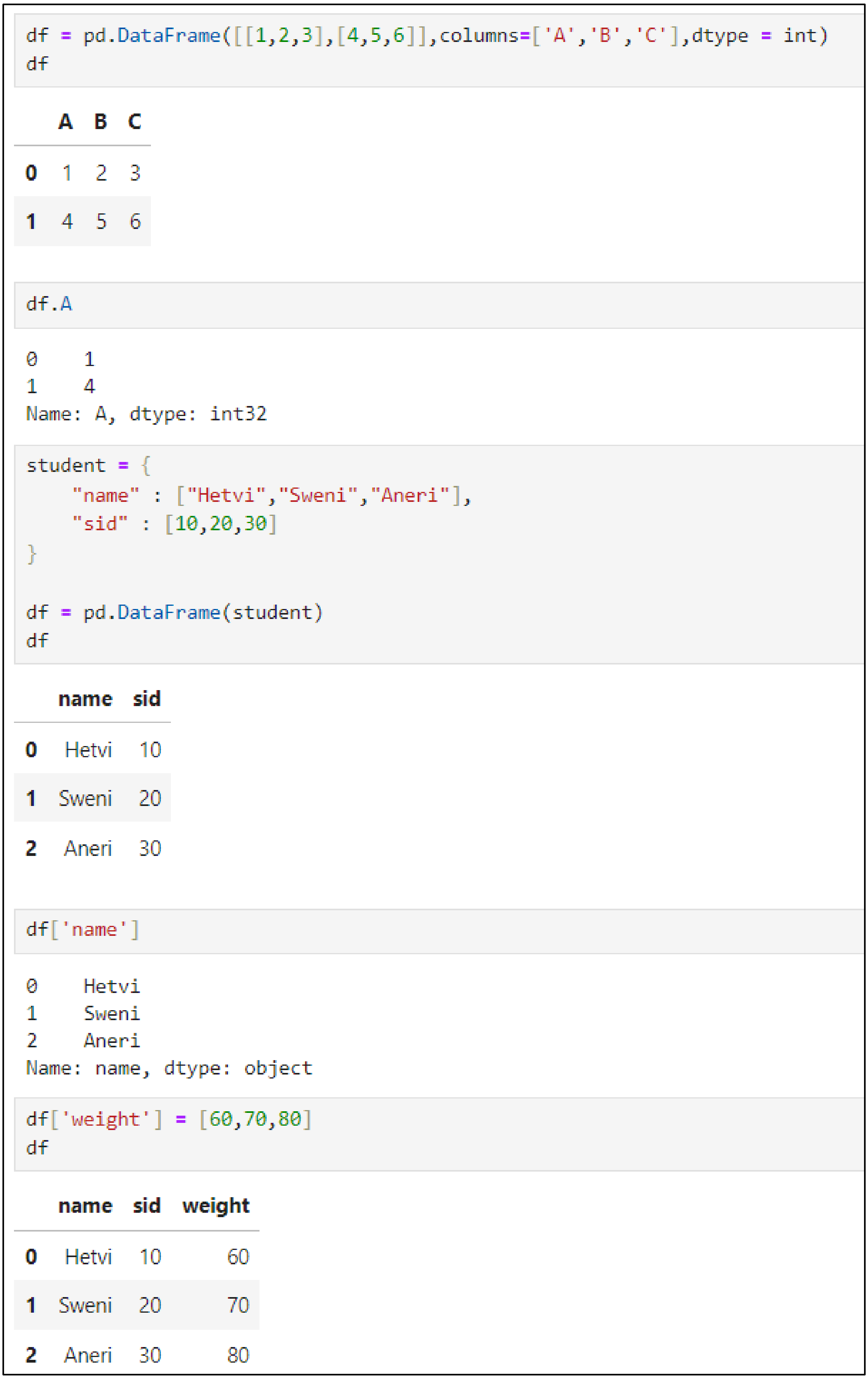
```
ds["A":"c"]
```

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

```
ds[ds>10]
```

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

- Create DataFrame:



```
df = pd.DataFrame([[1,2,3],[4,5,6]],columns=['A','B','C'],dtype = int)
df
```

	A	B	C
0	1	2	3
1	4	5	6

```
df.A
```

```
0    1
1    4
Name: A, dtype: int32
```

```
student = {
    "name" : ["Hetvi","Sweni","Aneri"],
    "sid" : [10,20,30]
}

df = pd.DataFrame(student)
df
```

	name	sid
0	Hetvi	10
1	Sweni	20
2	Aneri	30

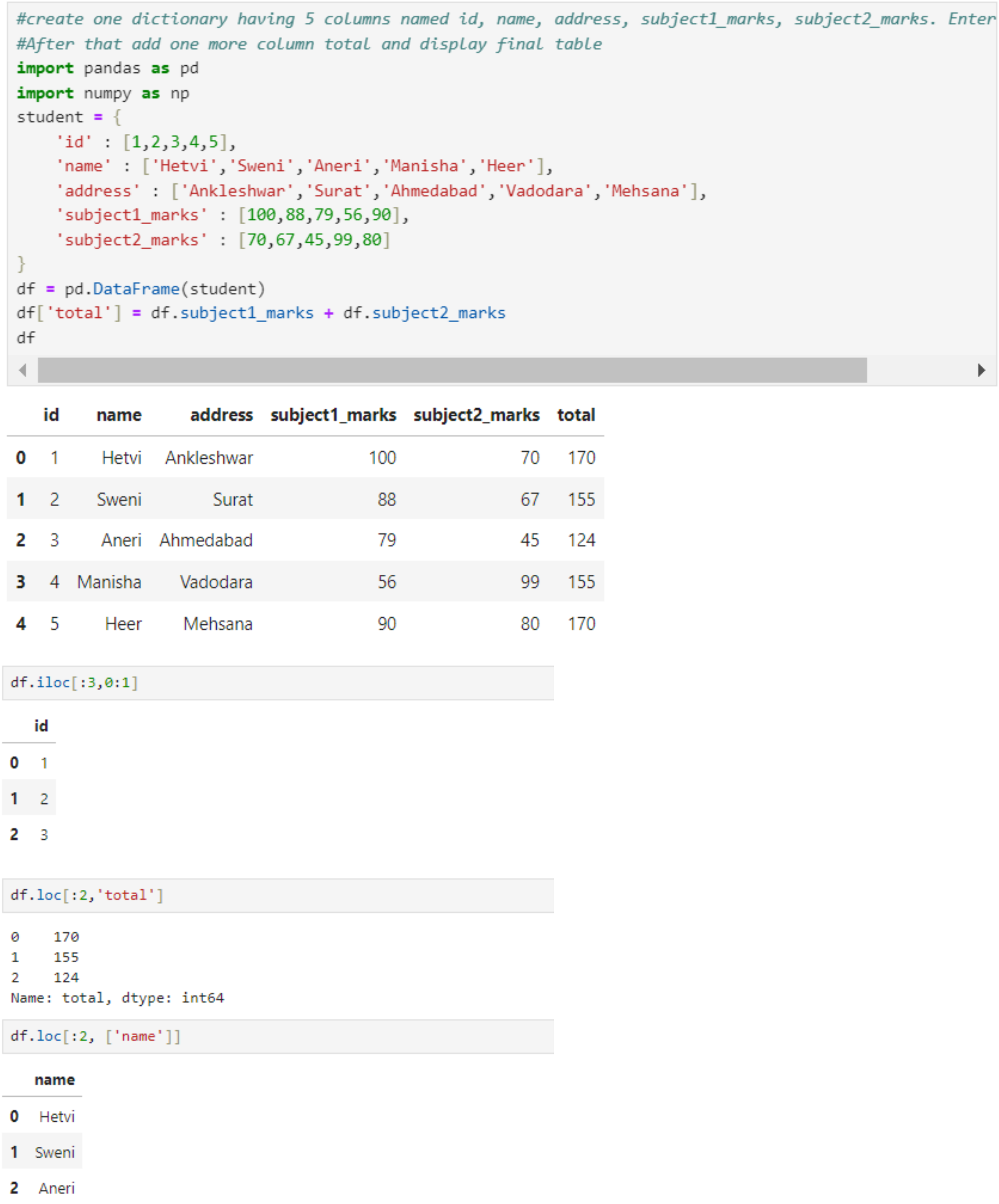
```
df['name']
```

```
0    Hetvi
1    Sweni
2    Aneri
Name: name, dtype: object
```

```
df['weight'] = [60,70,80]
df
```

	name	sid	weight
0	Hetvi	10	60
1	Sweni	20	70
2	Aneri	30	80

Question: Create an dictionary having 5 columns named id,name,address,sub1marks,sub2marks. Enter 5 records after that add one more column total and display final table.



```
#create one dictionary having 5 columns named id, name, address, subject1_marks, subject2_marks. Enter
#After that add one more column total and display final table
import pandas as pd
import numpy as np
student = {
    'id' : [1,2,3,4,5],
    'name' : ['Hetvi','Sweni','Aneri','Manisha','Heer'],
    'address' : ['Ankleshwar','Surat','Ahmedabad','Vadodara','Mehsana'],
    'subject1_marks' : [100,88,79,56,90],
    'subject2_marks' : [70,67,45,99,80]
}
df = pd.DataFrame(student)
df['total'] = df.subject1_marks + df.subject2_marks
df
```

	id	name	address	subject1_marks	subject2_marks	total
0	1	Hetvi	Ankleshwar	100	70	170
1	2	Sweni	Surat	88	67	155
2	3	Aneri	Ahmedabad	79	45	124
3	4	Manisha	Vadodara	56	99	155
4	5	Heer	Mehsana	90	80	170

```
df.iloc[:3,0:1]
```

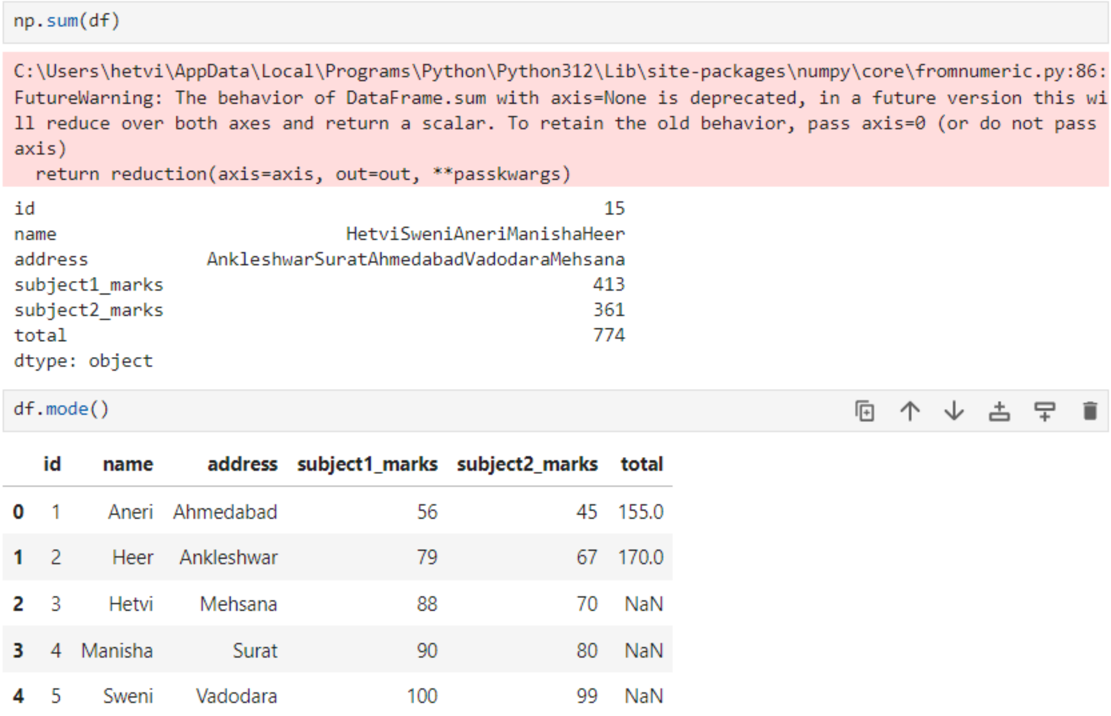
	id
0	1
1	2
2	3

```
df.loc[:2,'total']
```

```
0    170
1    155
2    124
Name: total, dtype: int64
```

```
df.loc[:2, ['name']]
```

	name
0	Hetvi
1	Sweni
2	Aneri



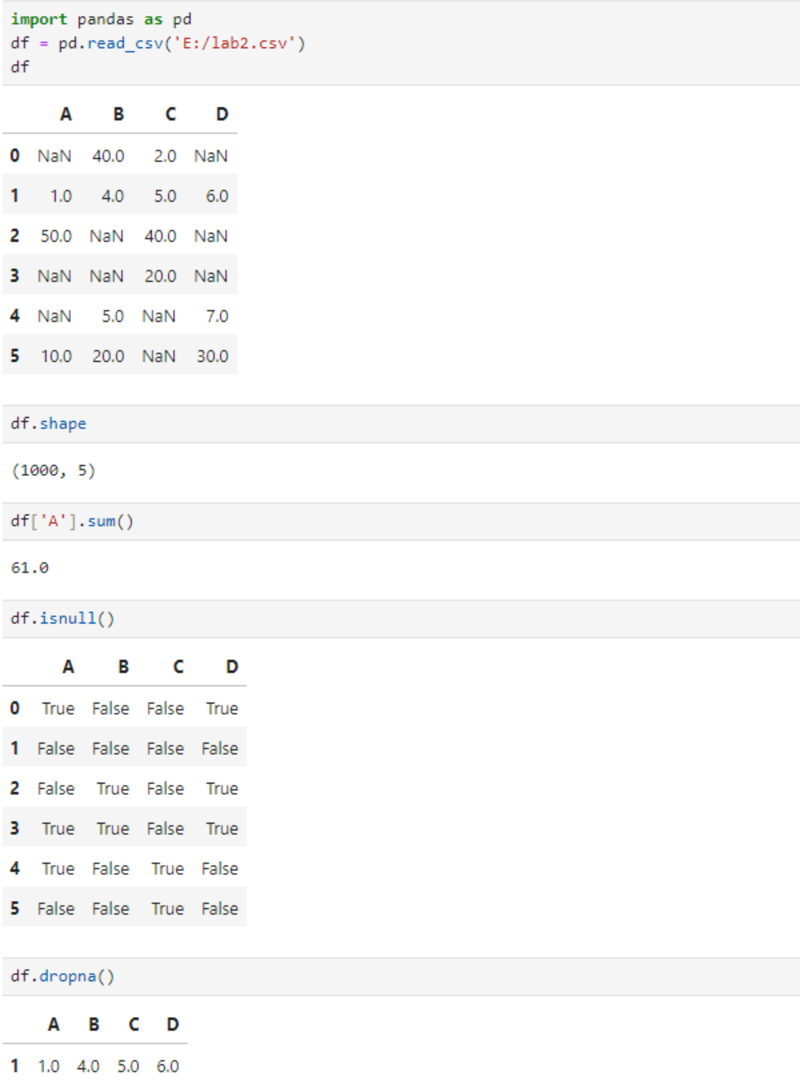
```
np.sum(df)
```

```
C:\Users\hetvi\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy\core\fromnumeric.py:86:
FutureWarning: The behavior of DataFrame.sum with axis=None is deprecated, in a future version this wi
ll reduce over both axes and return a scalar. To retain the old behavior, pass axis=0 (or do not pass
axis)
  return reduction(axis=axis, out=out, **passkwargs)
```

```
id                                                15
name                            HetviSweniAneriManishaHeer
address            AnkleshwarSuratAhmedabadVadodaraMehsana
subject1_marks                                   413
subject2_marks                                   361
total                                            774
dtype: object
```

```
df.mode()
```

	id	name	address	subject1_marks	subject2_marks	total
0	1	Aneri	Ahmedabad	56	45	155.0
1	2	Heer	Ankleshwar	79	67	170.0
2	3	Hetvi	Mehsana	88	70	NaN
3	4	Manisha	Surat	90	80	NaN
4	5	Sweni	Vadodara	100	99	NaN



```
import pandas as pd
df = pd.read_csv('E:/lab2.csv')
df
```

	A	B	C	D
0	NaN	40.0	2.0	NaN
1	1.0	4.0	5.0	6.0
2	50.0	NaN	40.0	NaN
3	NaN	NaN	20.0	NaN
4	NaN	5.0	NaN	7.0
5	10.0	20.0	NaN	30.0

```
df.shape
```

```
(1000, 5)
```

```
df['A'].sum()
```

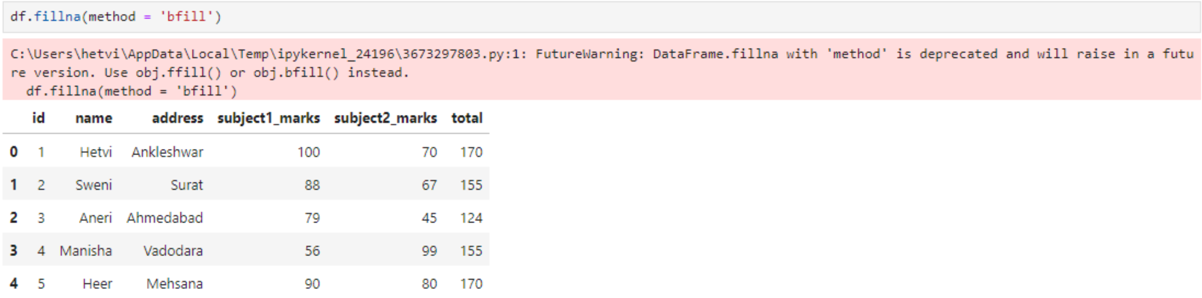
```
61.0
```

```
df.isnull()
```

	A	B	C	D
0	True	False	False	True
1	False	False	False	False
2	False	True	False	True
3	True	True	False	True
4	True	False	True	False
5	False	False	True	False

```
df.dropna()
```

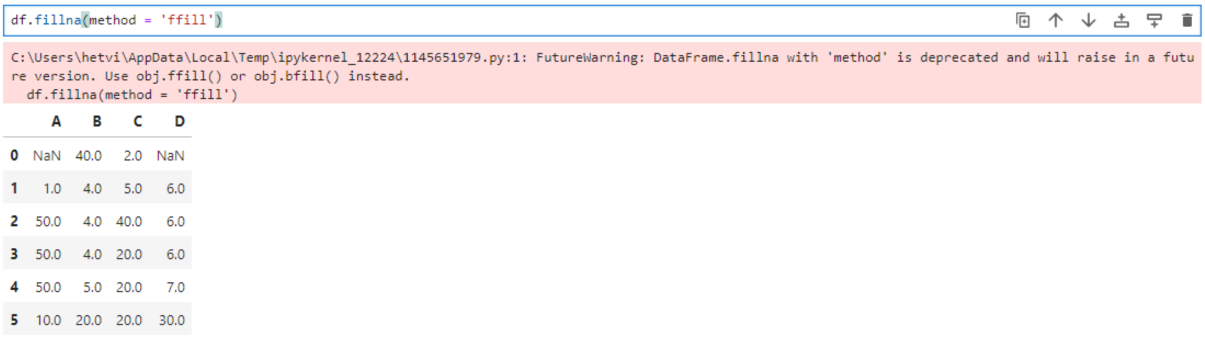
	A	B	C	D
1	1.0	4.0	5.0	6.0



```
df.fillna(method = 'bfill')
```

```
C:\Users\hetvi\AppData\Local\Temp\ipykernel_24196\3673297803.py:1: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a futu
re version. Use obj.ffill() or obj.bfill() instead.
  df.fillna(method = 'bfill')
```

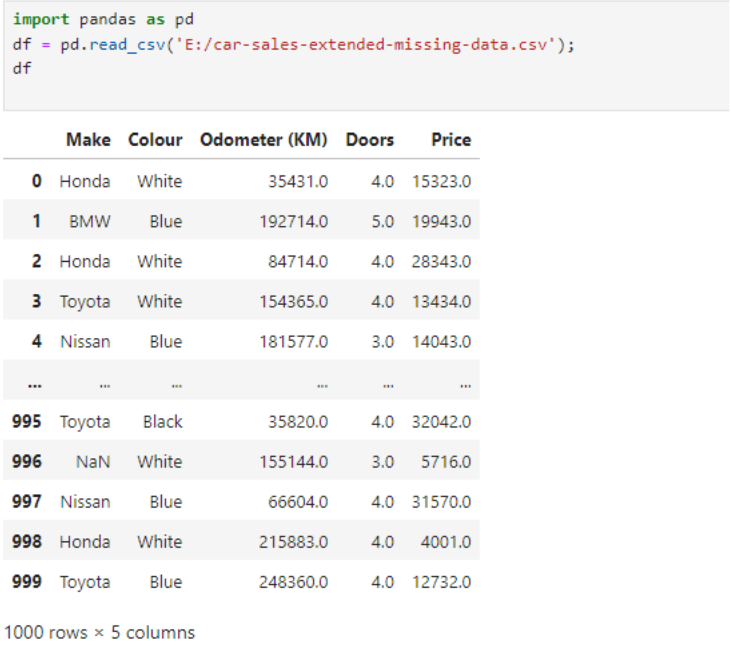
	id	name	address	subject1_marks	subject2_marks	total
0	1	Hetvi	Ankleshwar	100	70	170
1	2	Sweni	Surat	88	67	155
2	3	Aneri	Ahmedabad	79	45	124
3	4	Manisha	Vadodara	56	99	155
4	5	Heer	Mehsana	90	80	170



```
df.fillna(method = 'ffill')
```

```
C:\Users\hetvi\AppData\Local\Temp\ipykernel_12224\1145651979.py:1: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a futu
re version. Use obj.ffill() or obj.bfill() instead.
  df.fillna(method = 'ffill')
```

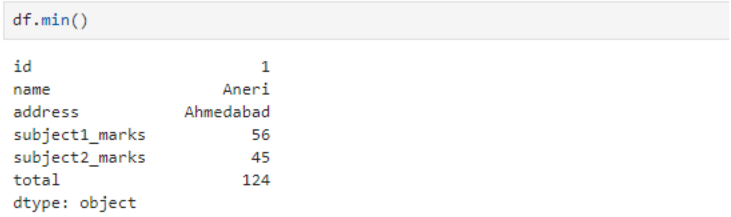
	A	B	C	D
0	NaN	40.0	2.0	NaN
1	1.0	4.0	5.0	6.0
2	50.0	4.0	40.0	6.0
3	50.0	4.0	20.0	6.0
4	50.0	5.0	20.0	7.0
5	10.0	20.0	20.0	30.0



```
import pandas as pd
df = pd.read_csv('E:/car-sales-extended-missing-data.csv');
df
```

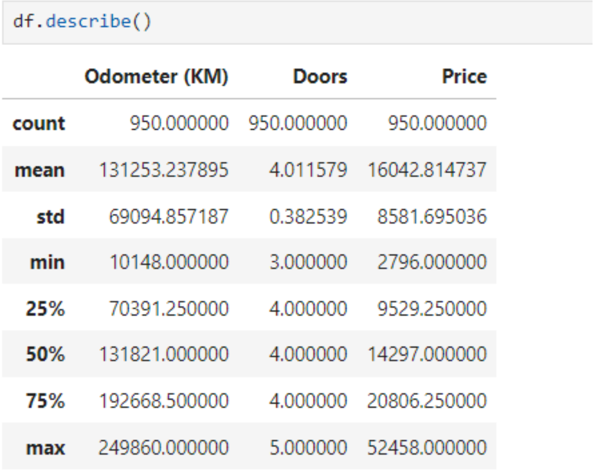
	Make	Colour	Odometer (KM)	Doors	Price
0	Honda	White	35431.0	4.0	15323.0
1	BMW	Blue	192714.0	5.0	19943.0
2	Honda	White	84714.0	4.0	28343.0
3	Toyota	White	154365.0	4.0	13434.0
4	Nissan	Blue	181577.0	3.0	14043.0
...	...	...	...	...	...
995	Toyota	Black	35820.0	4.0	32042.0
996	NaN	White	155144.0	3.0	5716.0
997	Nissan	Blue	66604.0	4.0	31570.0
998	Honda	White	215883.0	4.0	4001.0
999	Toyota	Blue	248360.0	4.0	12732.0

1000 rows × 5 columns



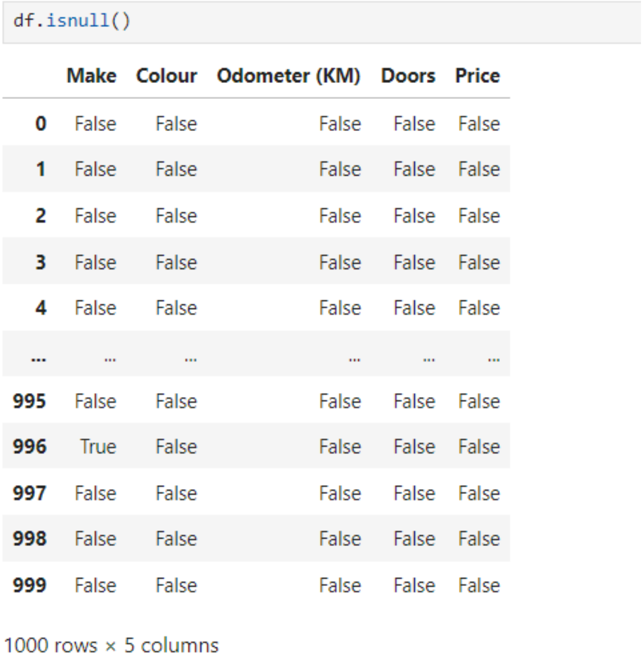
```
df.min()
```

```
id                        1
name                  Aneri
address           Ahmedabad
subject1_marks           56
subject2_marks           45
total                   124
dtype: object
```



```
df.describe()
```

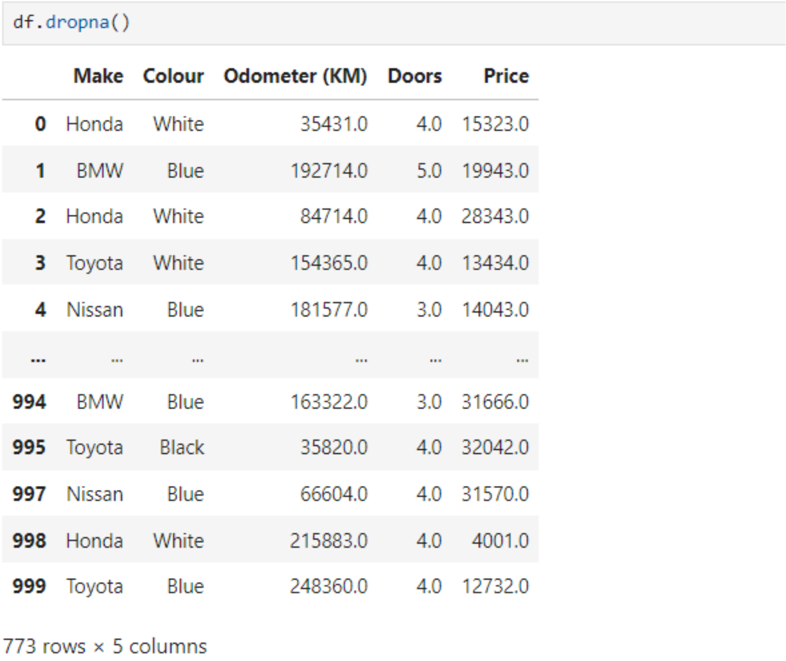
	Odometer (KM)	Doors	Price
count	950.000000	950.000000	950.000000
mean	131253.237895	4.011579	16042.814737
std	69094.857187	0.382539	8581.695036
min	10148.000000	3.000000	2796.000000
25%	70391.250000	4.000000	9529.250000
50%	131821.000000	4.000000	14297.000000
75%	192668.500000	4.000000	20806.250000
max	249860.000000	5.000000	52458.000000



```
df.isnull()
```

	Make	Colour	Odometer (KM)	Doors	Price
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
995	False	False	False	False	False
996	True	False	False	False	False
997	False	False	False	False	False
998	False	False	False	False	False
999	False	False	False	False	False

1000 rows × 5 columns

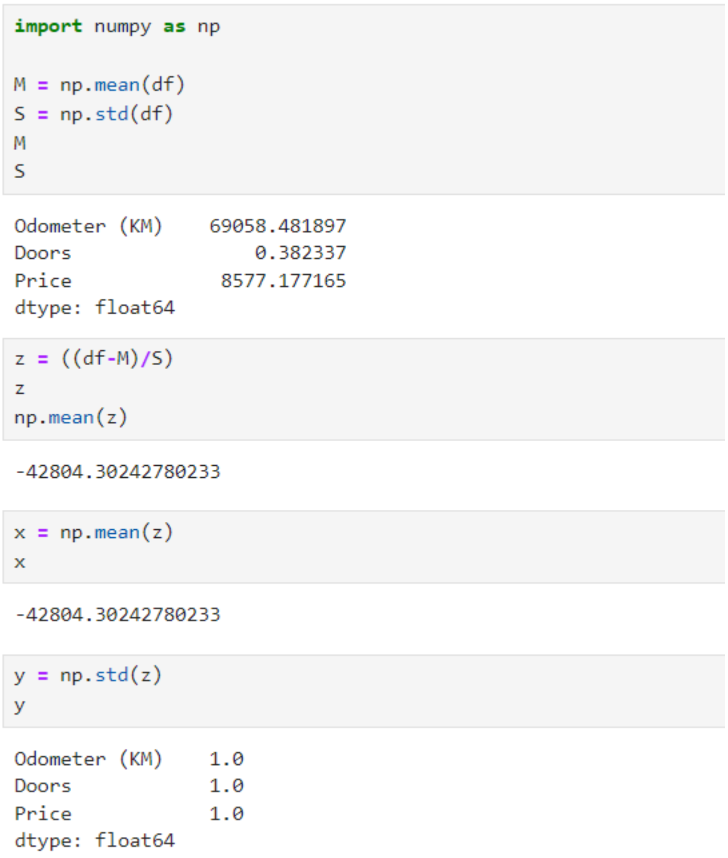


```
df.dropna()
```

	Make	Colour	Odometer (KM)	Doors	Price
0	Honda	White	35431.0	4.0	15323.0
1	BMW	Blue	192714.0	5.0	19943.0
2	Honda	White	84714.0	4.0	28343.0
3	Toyota	White	154365.0	4.0	13434.0
4	Nissan	Blue	181577.0	3.0	14043.0
...	...	...	...	...	...
994	BMW	Blue	163322.0	3.0	31666.0
995	Toyota	Black	35820.0	4.0	32042.0
997	Nissan	Blue	66604.0	4.0	31570.0
998	Honda	White	215883.0	4.0	4001.0
999	Toyota	Blue	248360.0	4.0	12732.0

773 rows × 5 columns

- Z-score Normalization:



```
import numpy as np

M = np.mean(df)
S = np.std(df)
M
S
```

```
Odometer (KM)    69058.481897
Doors                0.382337
Price             8577.177165
dtype: float64
```

```
z = ((df-M)/S)
z
np.mean(z)
```

```
-42804.30242780233
```

```
x = np.mean(z)
x
```

```
-42804.30242780233
```

```
y = np.std(z)
y
```

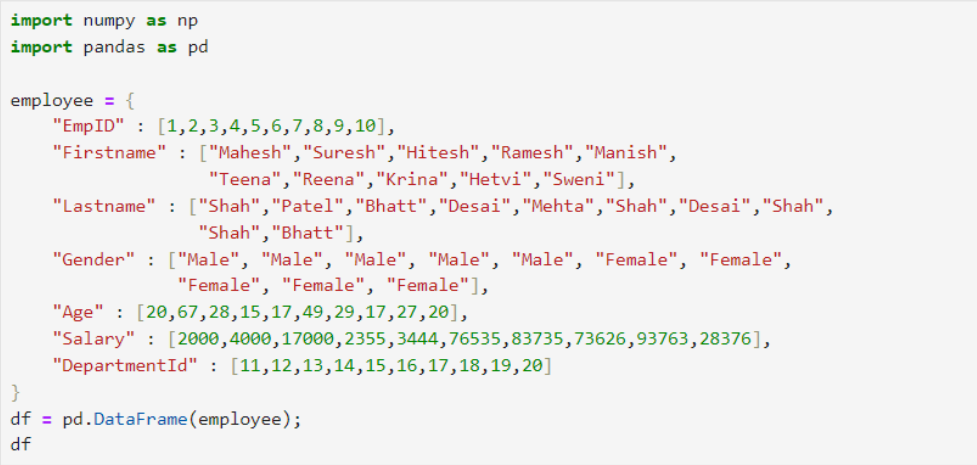
```
Odometer (KM)    1.0
Doors            1.0
Price            1.0
dtype: float64
```

Experiment – 3

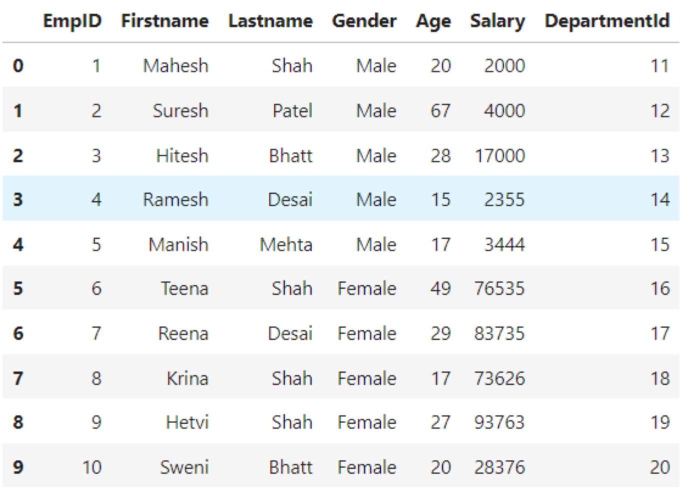
AIM: To perform DataPre-processing and Wrangling tasks like data cleaning, transformation, join, re-shape, String manipulation sample data-set.

Tools/Apparatus: Anaconda Python/ Spyder IDE

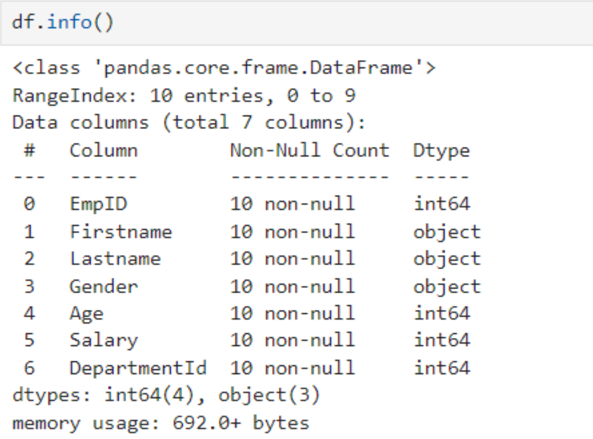
- Data Exploration:



```
import numpy as np
import pandas as pd

employee = {
    "EmpID" : [1,2,3,4,5,6,7,8,9,10],
    "Firstname" : ["Mahesh","Suresh","Hitesh","Ramesh","Manish",
                   "Teena","Reena","Krina","Hetvi","Sweni"],
    "Lastname" : ["Shah","Patel","Bhatt","Desai","Mehta","Shah","Desai","Shah",
                  "Shah","Bhatt"],
    "Gender" : ["Male", "Male", "Male", "Male", "Male", "Female", "Female",
                "Female", "Female", "Female"],
    "Age" : [20,67,28,15,17,49,29,17,27,20],
    "Salary" : [2000,4000,17000,2355,3444,76535,83735,73626,93763,28376],
    "DepartmentId" : [11,12,13,14,15,16,17,18,19,20]
}
df = pd.DataFrame(employee);
df
```

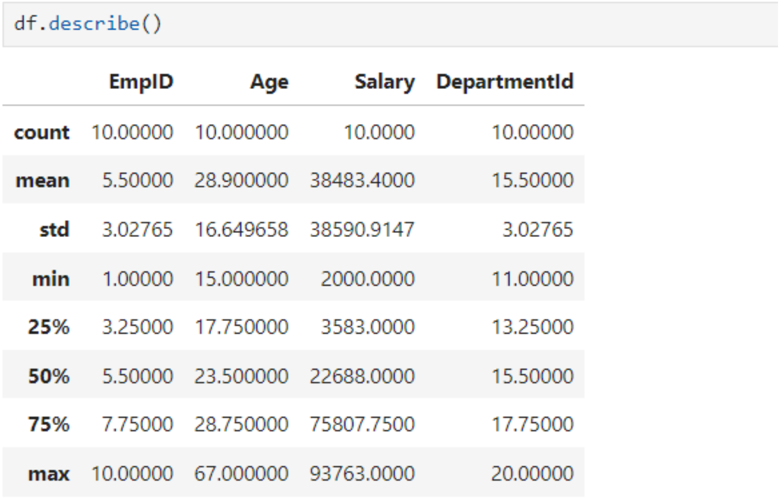


	EmpID	Firstname	Lastname	Gender	Age	Salary	DepartmentId
0	1	Mahesh	Shah	Male	20	2000	11
1	2	Suresh	Patel	Male	67	4000	12
2	3	Hitesh	Bhatt	Male	28	17000	13
3	4	Ramesh	Desai	Male	15	2355	14
4	5	Manish	Mehta	Male	17	3444	15
5	6	Teena	Shah	Female	49	76535	16
6	7	Reena	Desai	Female	29	83735	17
7	8	Krina	Shah	Female	17	73626	18
8	9	Hetvi	Shah	Female	27	93763	19
9	10	Sweni	Bhatt	Female	20	28376	20



```
df.info()
```

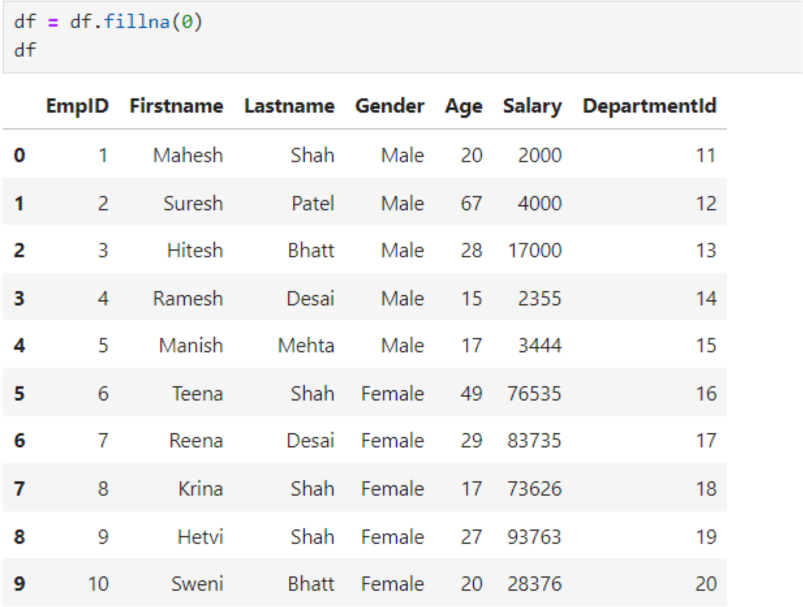
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   EmpID         10 non-null     int64 
 1   Firstname     10 non-null     object
 2   Lastname      10 non-null     object
 3   Gender        10 non-null     object
 4   Age           10 non-null     int64 
 5   Salary        10 non-null     int64 
 6   DepartmentId  10 non-null     int64 
dtypes: int64(4), object(3)
memory usage: 692.0+ bytes
```



```
df.describe()
```

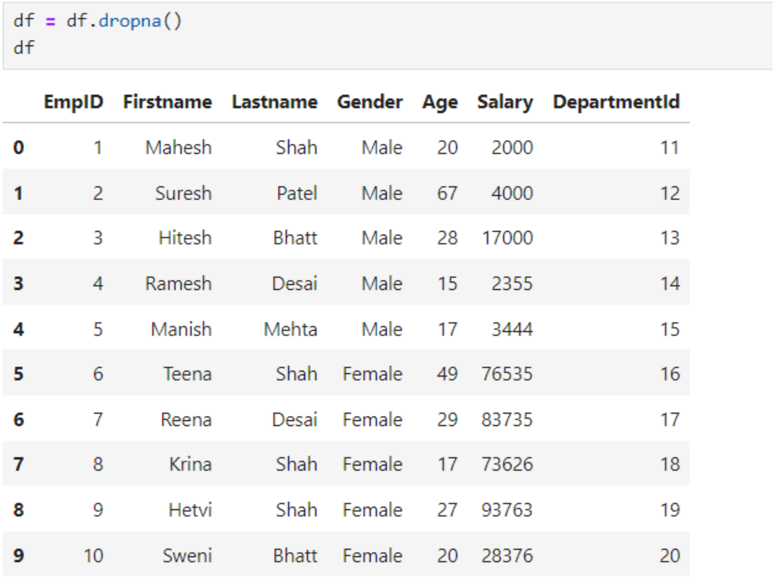
	EmpID	Age	Salary	DepartmentId
count	10.00000	10.000000	10.0000	10.00000
mean	5.50000	28.900000	38483.4000	15.50000
std	3.02765	16.649658	38590.9147	3.02765
min	1.00000	15.000000	2000.0000	11.00000
25%	3.25000	17.750000	3583.0000	13.25000
50%	5.50000	23.500000	22688.0000	15.50000
75%	7.75000	28.750000	75807.7500	17.75000
max	10.00000	67.000000	93763.0000	20.00000

- Dealing With Missing values:



```
df = df.fillna(0)
df
```

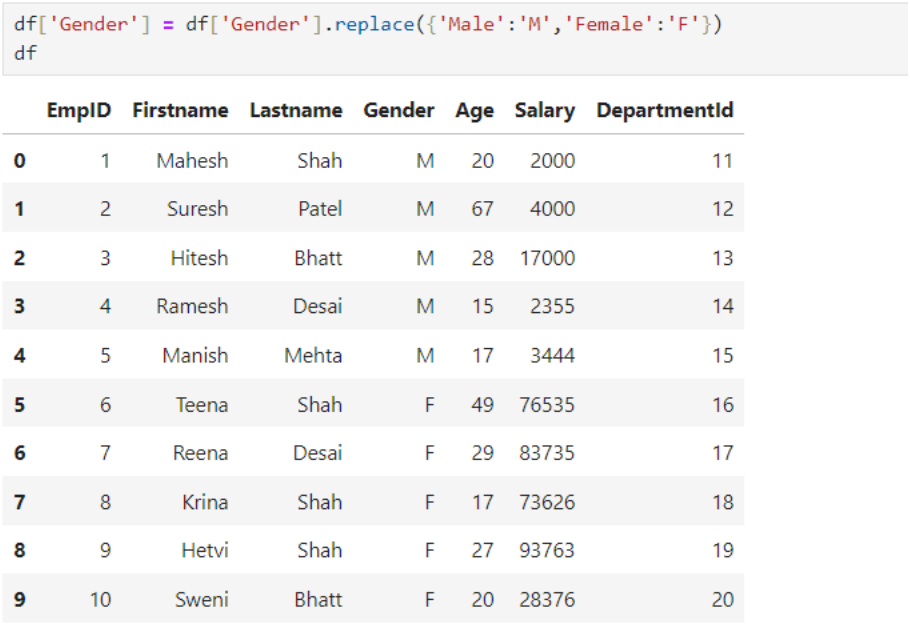
	EmpID	Firstname	Lastname	Gender	Age	Salary	DepartmentId
0	1	Mahesh	Shah	Male	20	2000	11
1	2	Suresh	Patel	Male	67	4000	12
2	3	Hitesh	Bhatt	Male	28	17000	13
3	4	Ramesh	Desai	Male	15	2355	14
4	5	Manish	Mehta	Male	17	3444	15
5	6	Teena	Shah	Female	49	76535	16
6	7	Reena	Desai	Female	29	83735	17
7	8	Krina	Shah	Female	17	73626	18
8	9	Hetvi	Shah	Female	27	93763	19
9	10	Sweni	Bhatt	Female	20	28376	20



```
df = df.dropna()
df
```

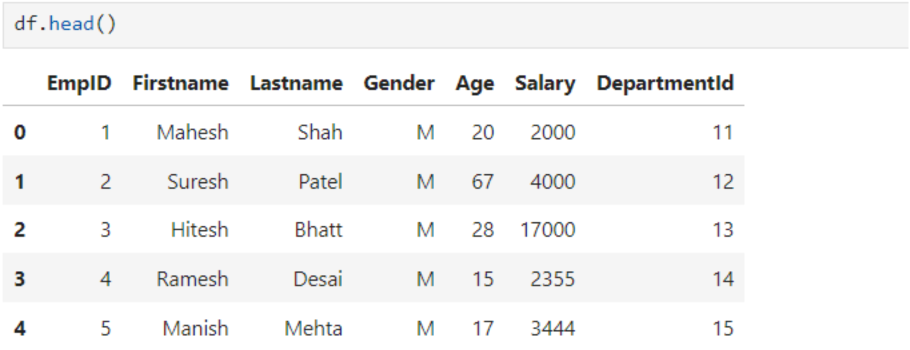
	EmpID	Firstname	Lastname	Gender	Age	Salary	DepartmentId
0	1	Mahesh	Shah	Male	20	2000	11
1	2	Suresh	Patel	Male	67	4000	12
2	3	Hitesh	Bhatt	Male	28	17000	13
3	4	Ramesh	Desai	Male	15	2355	14
4	5	Manish	Mehta	Male	17	3444	15
5	6	Teena	Shah	Female	49	76535	16
6	7	Reena	Desai	Female	29	83735	17
7	8	Krina	Shah	Female	17	73626	18
8	9	Hetvi	Shah	Female	27	93763	19
9	10	Sweni	Bhatt	Female	20	28376	20

- Reshaping data, Filtering data:



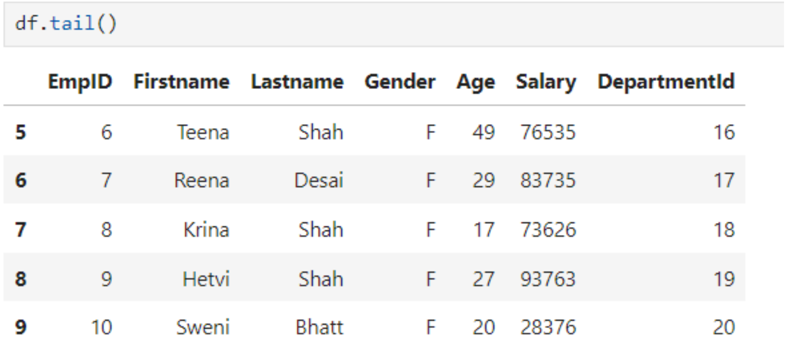
```
df['Gender'] = df['Gender'].replace({'Male':'M','Female':'F'})
df
```

	EmpID	Firstname	Lastname	Gender	Age	Salary	DepartmentId
0	1	Mahesh	Shah	M	20	2000	11
1	2	Suresh	Patel	M	67	4000	12
2	3	Hitesh	Bhatt	M	28	17000	13
3	4	Ramesh	Desai	M	15	2355	14
4	5	Manish	Mehta	M	17	3444	15
5	6	Teena	Shah	F	49	76535	16
6	7	Reena	Desai	F	29	83735	17
7	8	Krina	Shah	F	17	73626	18
8	9	Hetvi	Shah	F	27	93763	19
9	10	Sweni	Bhatt	F	20	28376	20



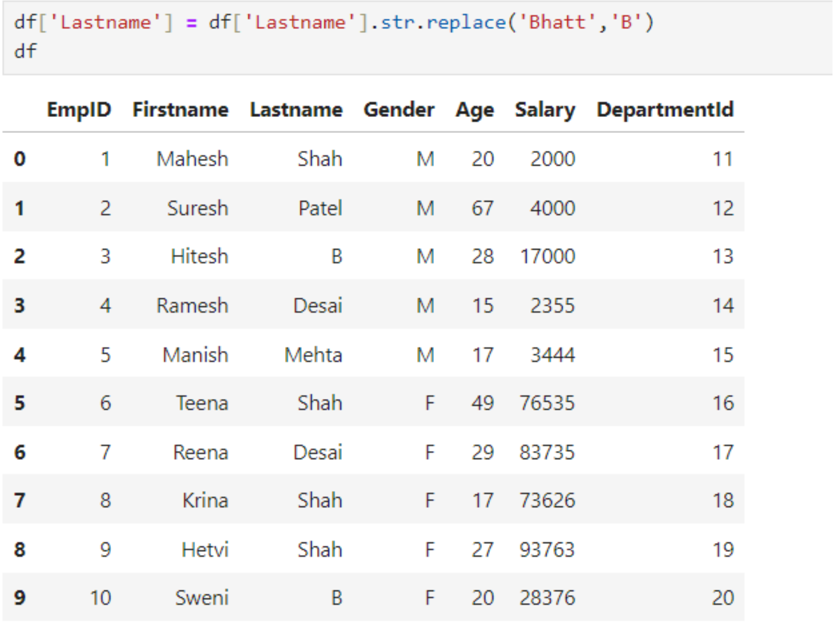
```
df.head()
```

	EmpID	Firstname	Lastname	Gender	Age	Salary	DepartmentId
0	1	Mahesh	Shah	M	20	2000	11
1	2	Suresh	Patel	M	67	4000	12
2	3	Hitesh	Bhatt	M	28	17000	13
3	4	Ramesh	Desai	M	15	2355	14
4	5	Manish	Mehta	M	17	3444	15



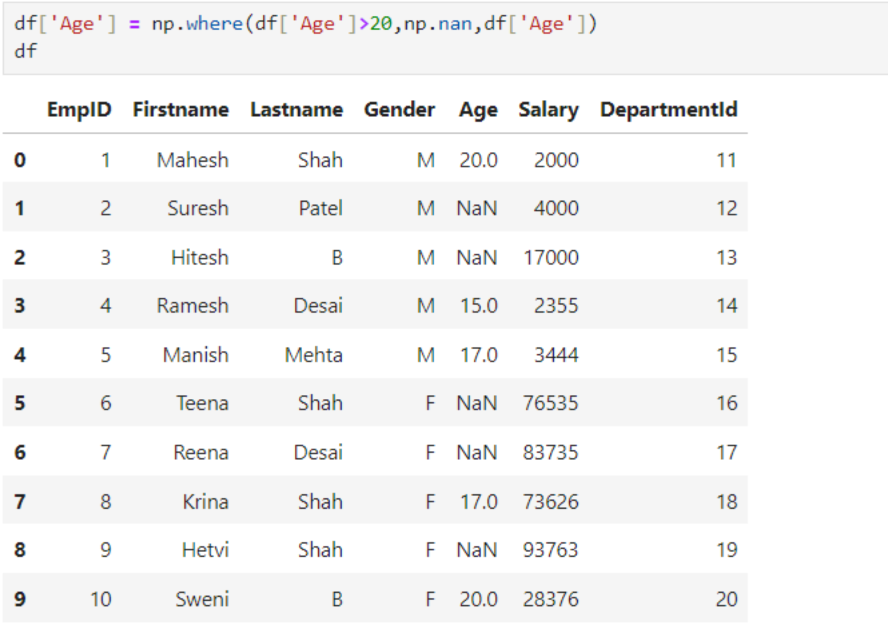
```
df.tail()
```

	EmpID	Firstname	Lastname	Gender	Age	Salary	DepartmentId
5	6	Teena	Shah	F	49	76535	16
6	7	Reena	Desai	F	29	83735	17
7	8	Krina	Shah	F	17	73626	18
8	9	Hetvi	Shah	F	27	93763	19
9	10	Sweni	Bhatt	F	20	28376	20



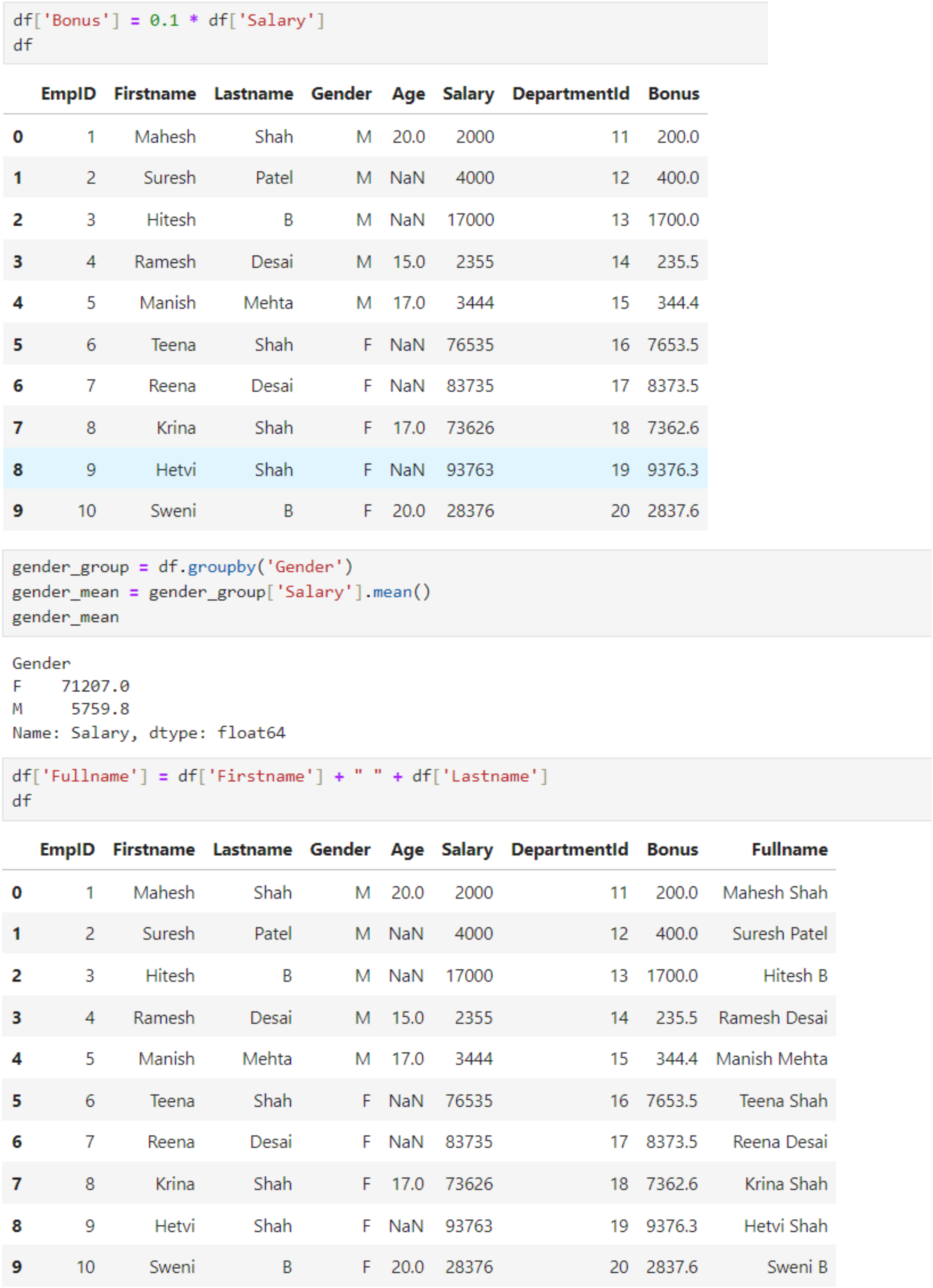
```
df['Lastname'] = df['Lastname'].str.replace('Bhatt','B')
df
```

	EmpID	Firstname	Lastname	Gender	Age	Salary	DepartmentId
0	1	Mahesh	Shah	M	20	2000	11
1	2	Suresh	Patel	M	67	4000	12
2	3	Hitesh	B	M	28	17000	13
3	4	Ramesh	Desai	M	15	2355	14
4	5	Manish	Mehta	M	17	3444	15
5	6	Teena	Shah	F	49	76535	16
6	7	Reena	Desai	F	29	83735	17
7	8	Krina	Shah	F	17	73626	18
8	9	Hetvi	Shah	F	27	93763	19
9	10	Sweni	B	F	20	28376	20



```
df['Age'] = np.where(df['Age']>20,np.nan,df['Age'])
df
```

	EmpID	Firstname	Lastname	Gender	Age	Salary	DepartmentId
0	1	Mahesh	Shah	M	20.0	2000	11
1	2	Suresh	Patel	M	NaN	4000	12
2	3	Hitesh	B	M	NaN	17000	13
3	4	Ramesh	Desai	M	15.0	2355	14
4	5	Manish	Mehta	M	17.0	3444	15
5	6	Teena	Shah	F	NaN	76535	16
6	7	Reena	Desai	F	NaN	83735	17
7	8	Krina	Shah	F	17.0	73626	18
8	9	Hetvi	Shah	F	NaN	93763	19
9	10	Sweni	B	F	20.0	28376	20



```
df['Bonus'] = 0.1 * df['Salary']
df
```

	EmpID	Firstname	Lastname	Gender	Age	Salary	DepartmentId	Bonus
0	1	Mahesh	Shah	M	20.0	2000	11	200.0
1	2	Suresh	Patel	M	NaN	4000	12	400.0
2	3	Hitesh	B	M	NaN	17000	13	1700.0
3	4	Ramesh	Desai	M	15.0	2355	14	235.5
4	5	Manish	Mehta	M	17.0	3444	15	344.4
5	6	Teena	Shah	F	NaN	76535	16	7653.5
6	7	Reena	Desai	F	NaN	83735	17	8373.5
7	8	Krina	Shah	F	17.0	73626	18	7362.6
8	9	Hetvi	Shah	F	NaN	93763	19	9376.3
9	10	Sweni	B	F	20.0	28376	20	2837.6

```
gender_group = df.groupby('Gender')
gender_mean = gender_group['Salary'].mean()
gender_mean
```

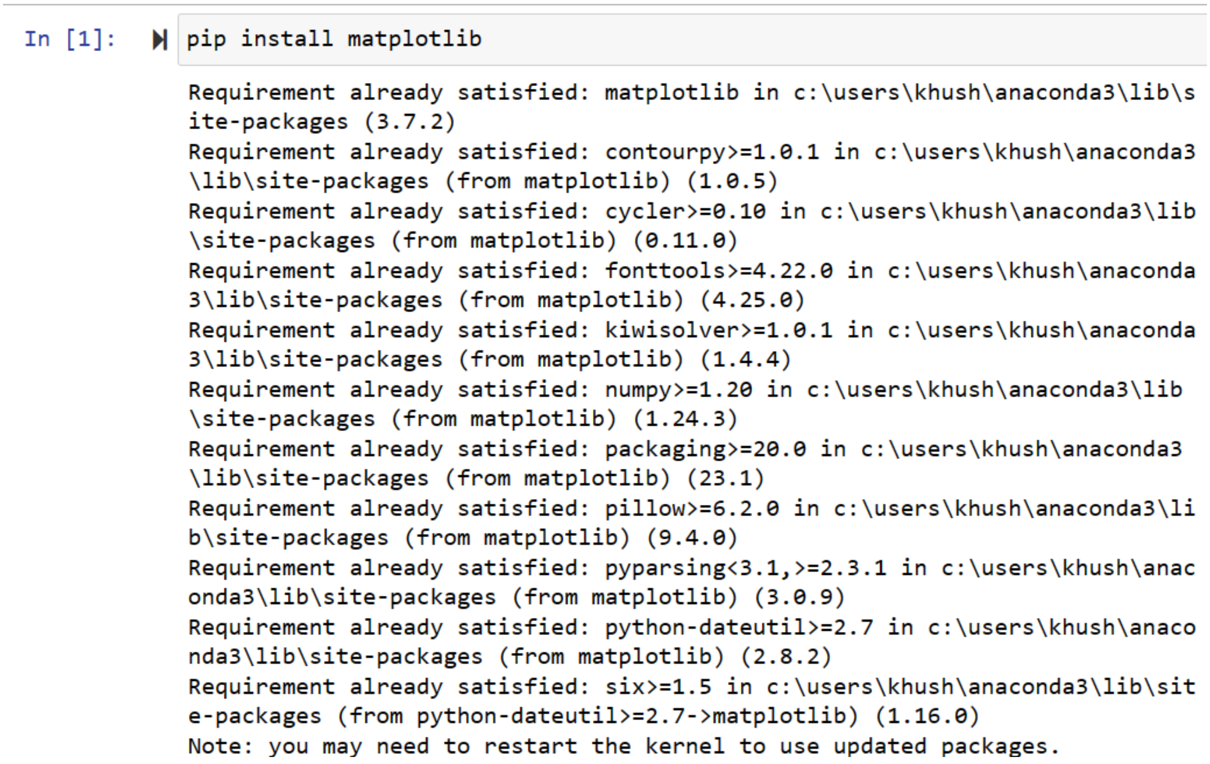
```
Gender
F    71207.0
M     5759.8
Name: Salary, dtype: float64
```

```
df['Fullname'] = df['Firstname'] + " " + df['Lastname']
df
```

	EmpID	Firstname	Lastname	Gender	Age	Salary	DepartmentId	Bonus	Fullname
0	1	Mahesh	Shah	M	20.0	2000	11	200.0	Mahesh Shah
1	2	Suresh	Patel	M	NaN	4000	12	400.0	Suresh Patel
2	3	Hitesh	B	M	NaN	17000	13	1700.0	Hitesh B
3	4	Ramesh	Desai	M	15.0	2355	14	235.5	Ramesh Desai
4	5	Manish	Mehta	M	17.0	3444	15	344.4	Manish Mehta
5	6	Teena	Shah	F	NaN	76535	16	7653.5	Teena Shah
6	7	Reena	Desai	F	NaN	83735	17	8373.5	Reena Desai
7	8	Krina	Shah	F	17.0	73626	18	7362.6	Krina Shah
8	9	Hetvi	Shah	F	NaN	93763	19	9376.3	Hetvi Shah
9	10	Sweni	B	F	20.0	28376	20	2837.6	Sweni B

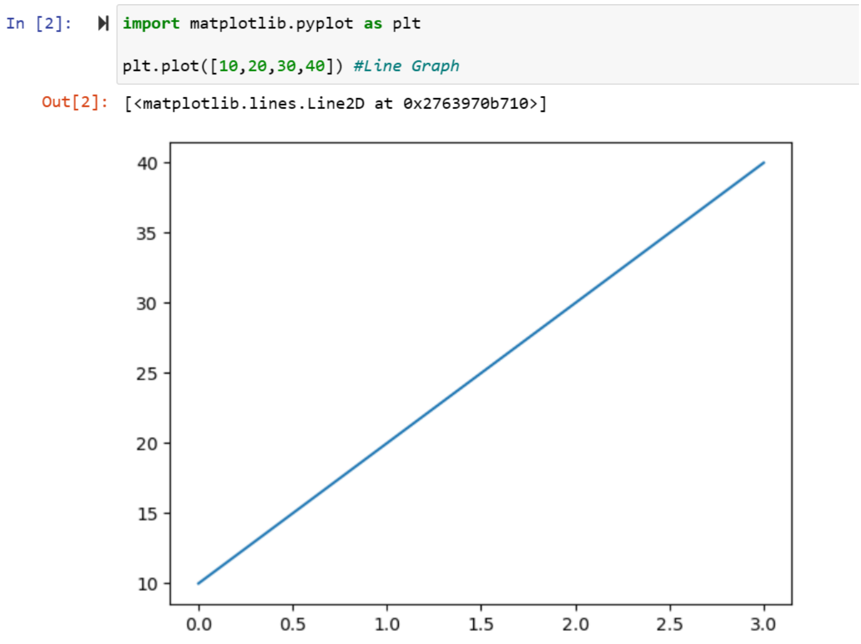
Experiment – 4

AIM: To perform Data visualization and plotting techniques like Lineplot, Barchart, Piechart, Boxchart using Matplotlib libraries.
Tools/Apparatus: Anaconda Python/ Spyder IDE

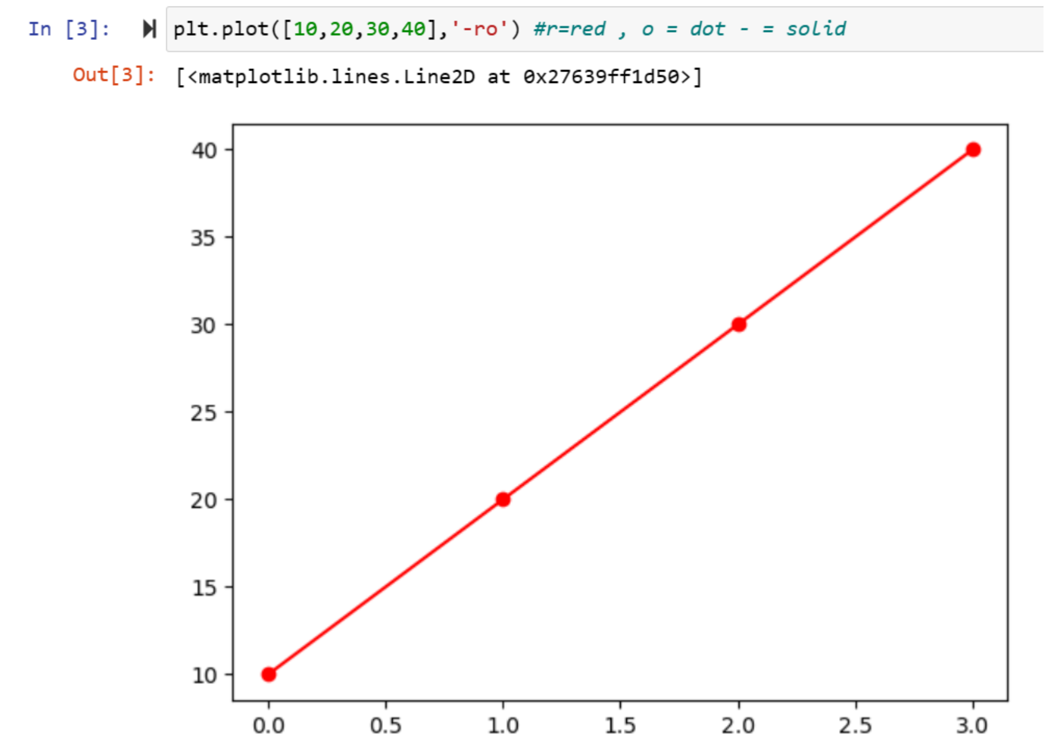


```
In [1]:  pip install matplotlib

Requirement already satisfied: matplotlib in c:\users\khush\anaconda3\lib\s
ite-packages (3.7.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\khush\anaconda3
\lib\site-packages (from matplotlib) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\khush\anaconda3\lib
\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\khush\anaconda
3\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\khush\anaconda
3\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: numpy>=1.20 in c:\users\khush\anaconda3\lib
\site-packages (from matplotlib) (1.24.3)
Requirement already satisfied: packaging>=20.0 in c:\users\khush\anaconda3
\lib\site-packages (from matplotlib) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\khush\anaconda3\li
b\site-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing<3.1,>=2.3.1 in c:\users\khush\anac
onda3\lib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\khush\anaco
nda3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\khush\anaconda3\lib\sit
e-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

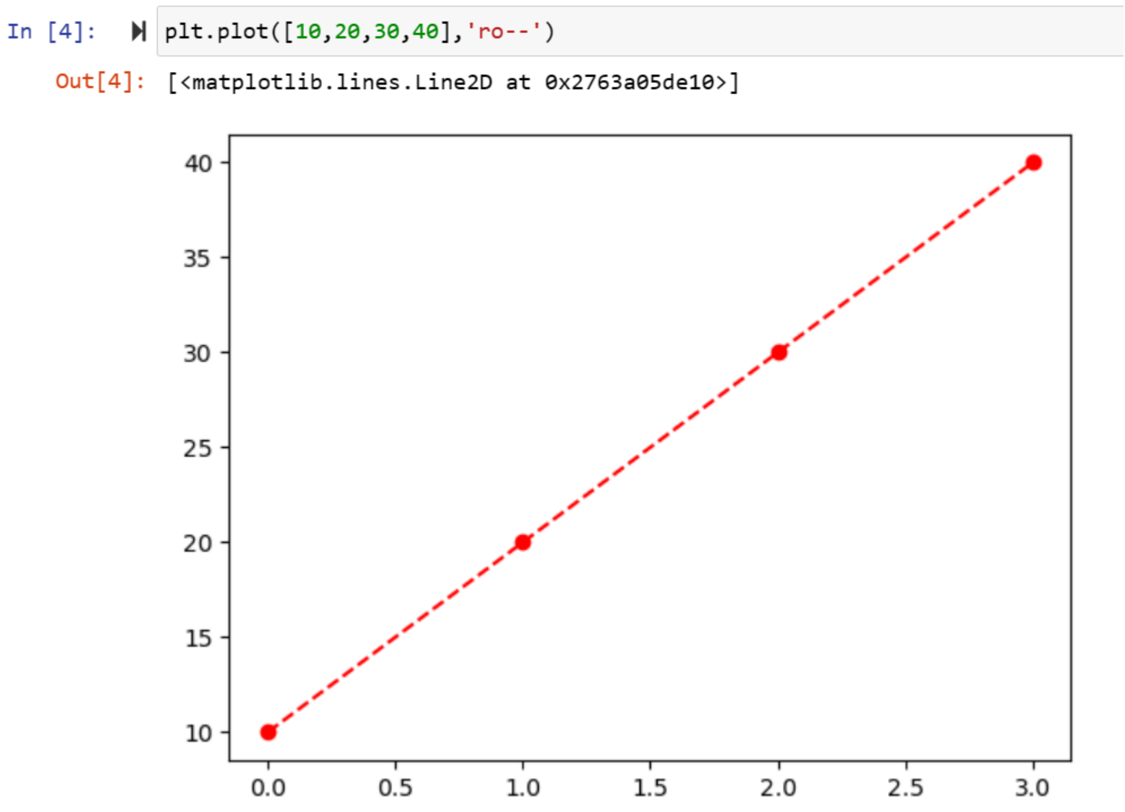
- Line Chart:



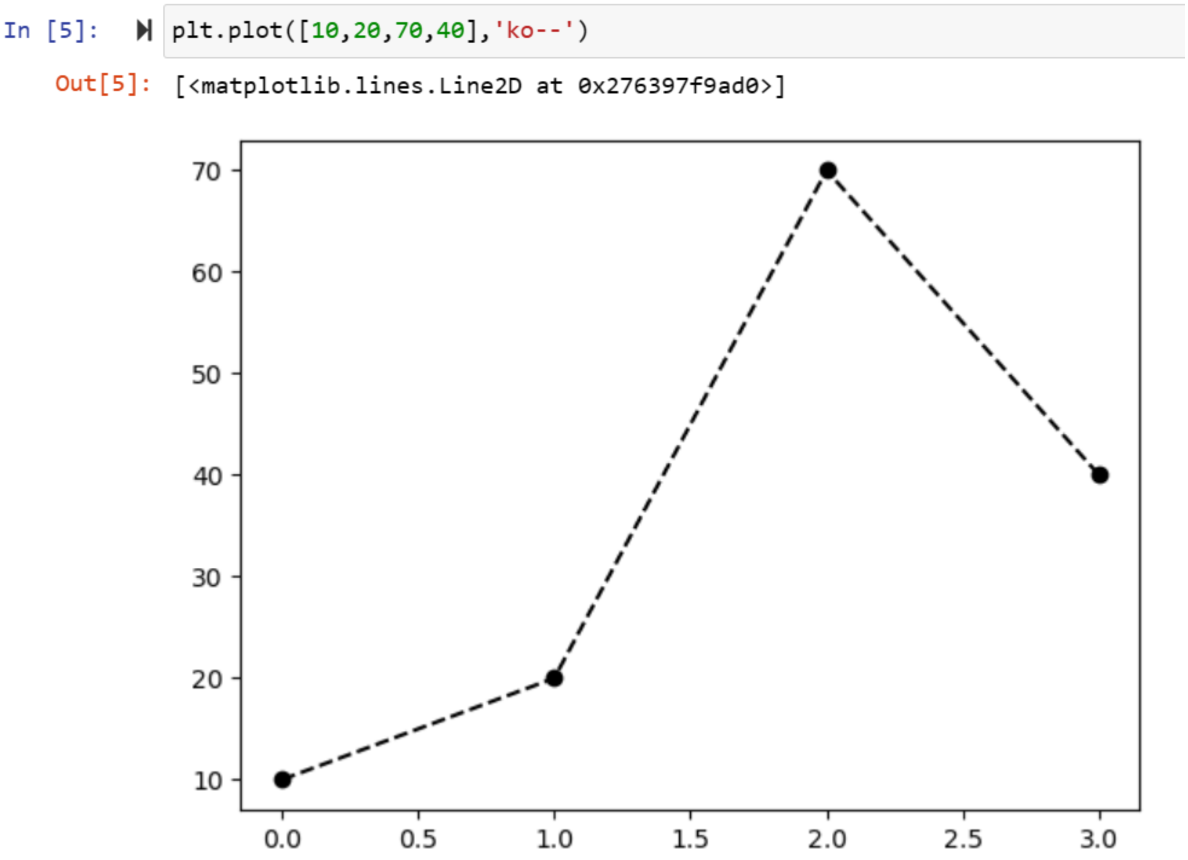
```
In [2]:  import matplotlib.pyplot as plt

         plt.plot([10,20,30,40]) #Line Graph

Out[2]:  [<matplotlib.lines.Line2D at 0x2763970b710>]
```



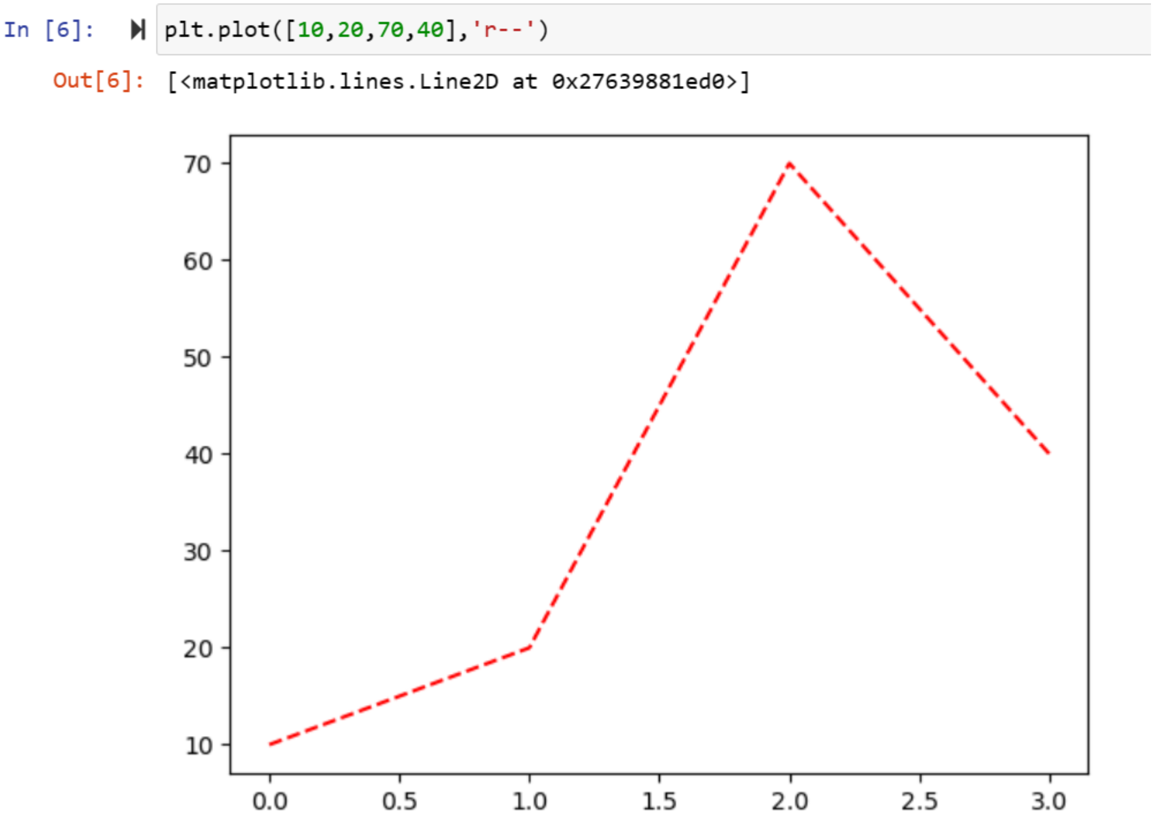
```
In [3]:  plt.plot([10,20,30,40],'-ro') #r=red , o = dot - = solid
Out[3]: [<matplotlib.lines.Line2D at 0x27639ff1d50>]
```



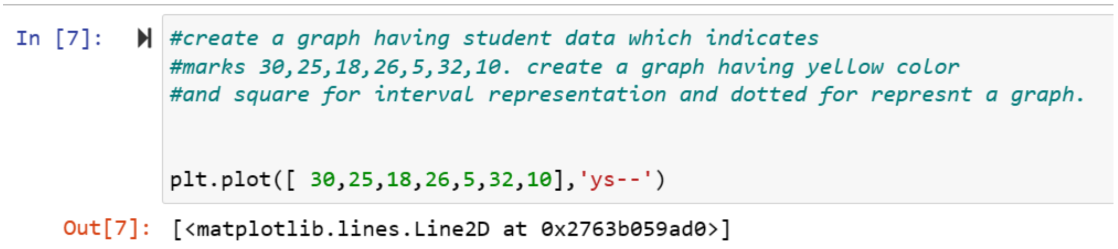
```
In [4]:  plt.plot([10,20,30,40],'ro--')
Out[4]: [<matplotlib.lines.Line2D at 0x2763a05de10>]
```



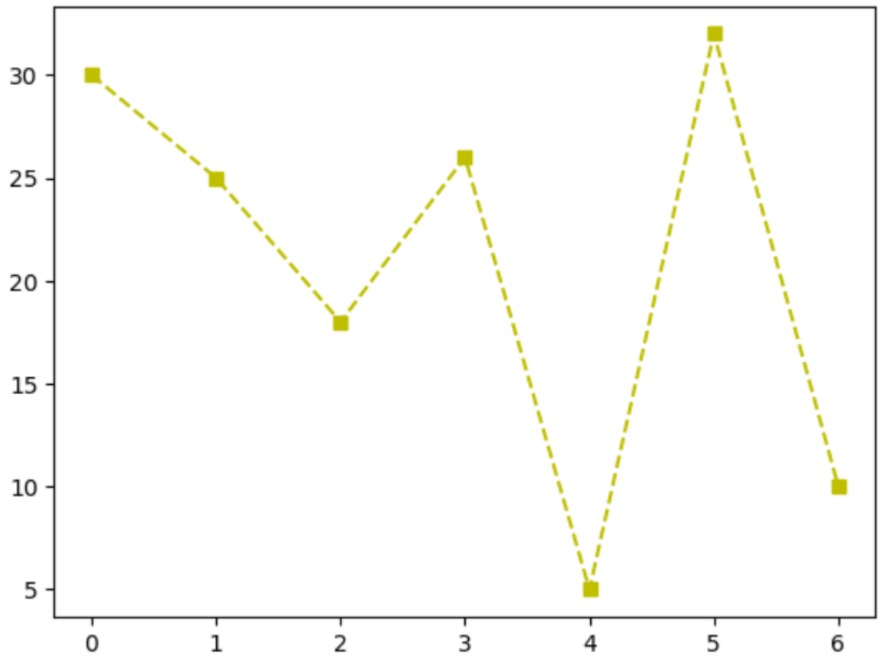
```
In [5]:  plt.plot([10,20,70,40],'ko--')
Out[5]:  [<matplotlib.lines.Line2D at 0x276397f9ad0>]
```

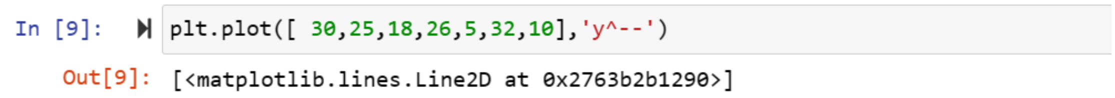


```
In [6]:  plt.plot([10,20,70,40],'r--')
Out[6]:  [<matplotlib.lines.Line2D at 0x27639881ed0>]
```

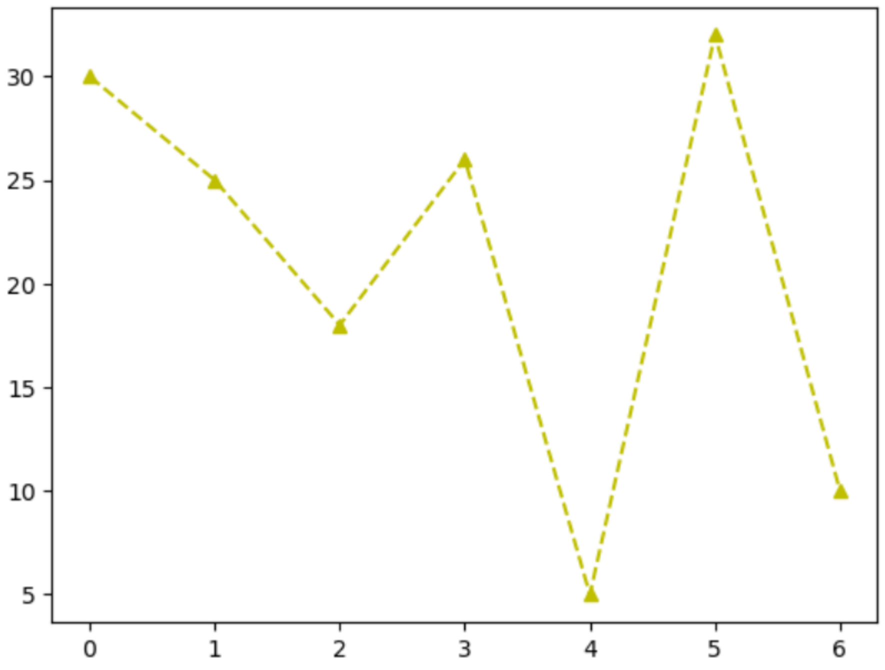


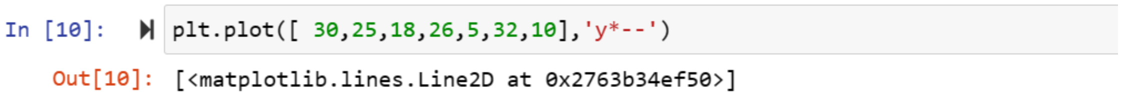
```
In [7]:  #create a graph having student data which indicates
         #marks 30,25,18,26,5,32,10. create a graph having yellow color
         #and square for interval representation and dotted for represnt a graph.


         plt.plot([ 30,25,18,26,5,32,10],'ys--')

Out[7]:  [<matplotlib.lines.Line2D at 0x2763b059ad0>]
```



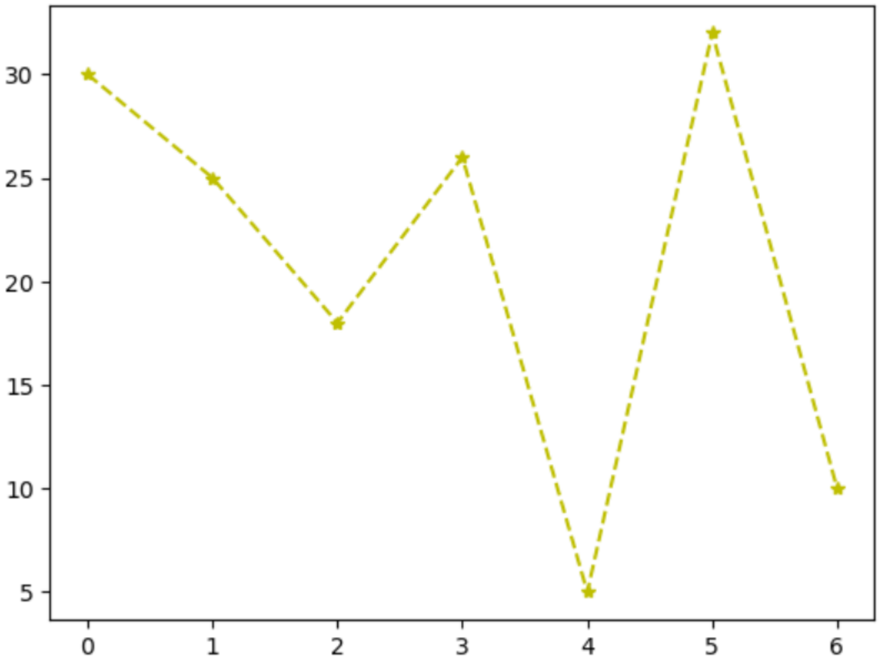


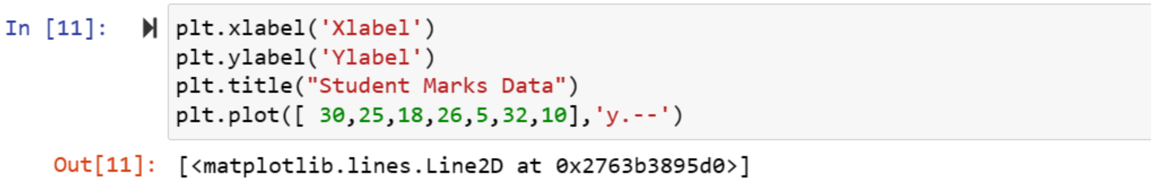
```
In [9]:  plt.plot([ 30,25,18,26,5,32,10],'y^--')

Out[9]:  [<matplotlib.lines.Line2D at 0x2763b2b1290>]
```



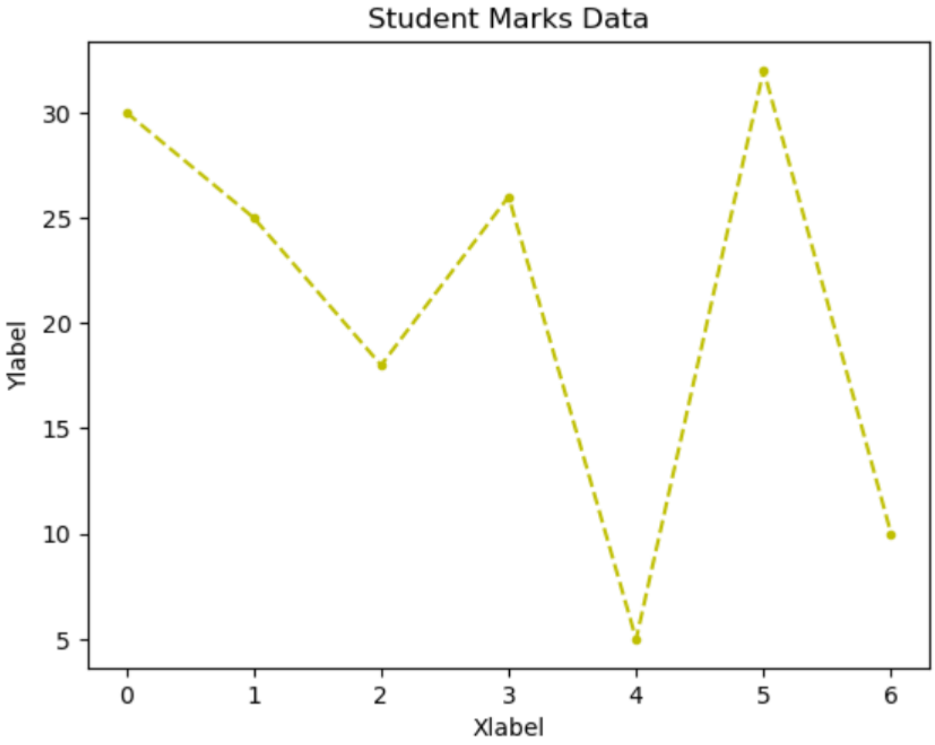


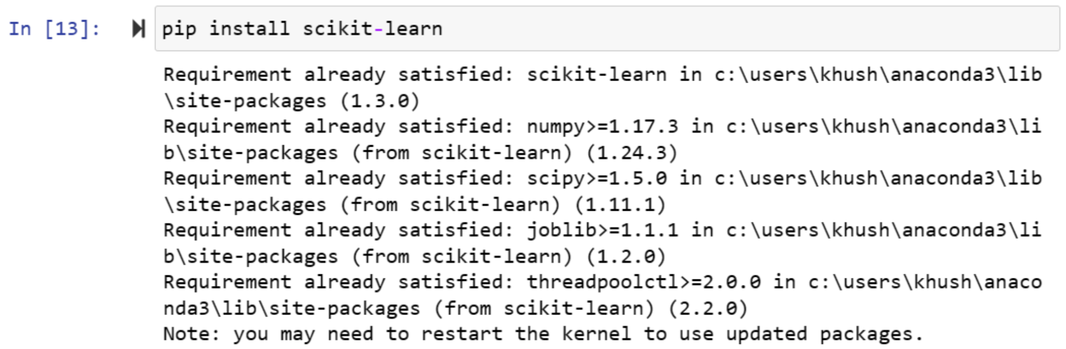
```
In [10]:  plt.plot([ 30,25,18,26,5,32,10],'y*--')
Out[10]:  [<matplotlib.lines.Line2D at 0x2763b34ef50>]
```



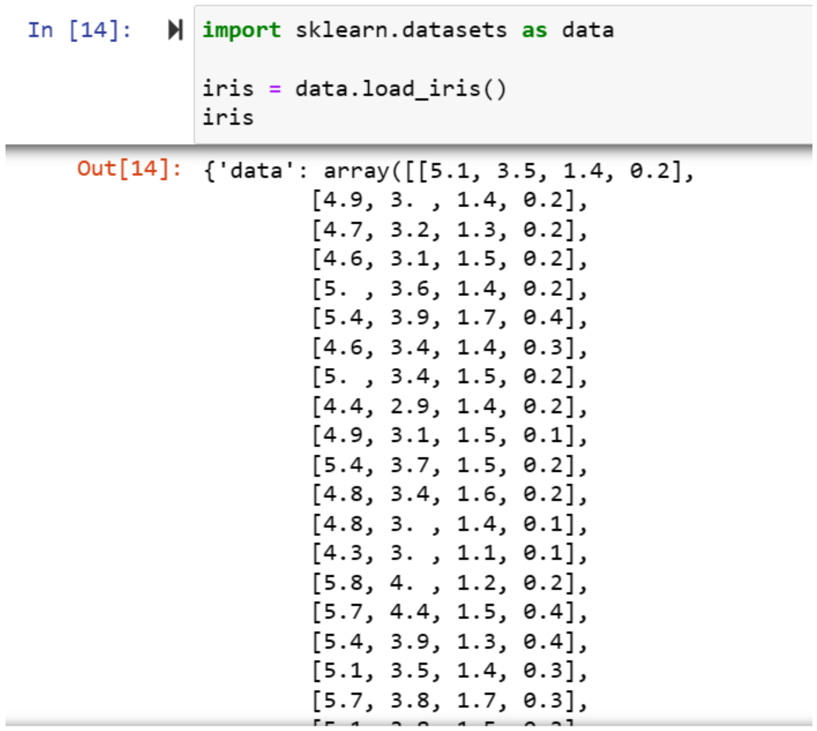


```
In [11]:  plt.xlabel('Xlabel')
          plt.ylabel('Ylabel')
          plt.title("Student Marks Data")
          plt.plot([ 30,25,18,26,5,32,10],'y.--')
Out[11]:  [<matplotlib.lines.Line2D at 0x2763b3895d0>]
```



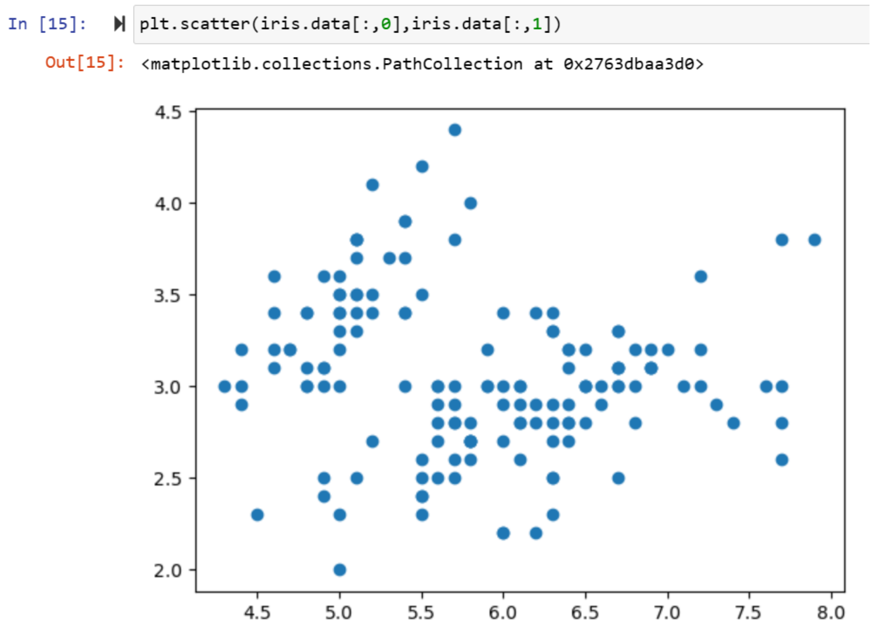


```
In [13]:  pip install scikit-learn

          Requirement already satisfied: scikit-learn in c:\users\khush\anaconda3\lib
          \site-packages (1.3.0)
          Requirement already satisfied: numpy>=1.17.3 in c:\users\khush\anaconda3\li
          b\site-packages (from scikit-learn) (1.24.3)
          Requirement already satisfied: scipy>=1.5.0 in c:\users\khush\anaconda3\lib
          \site-packages (from scikit-learn) (1.11.1)
          Requirement already satisfied: joblib>=1.1.1 in c:\users\khush\anaconda3\li
          b\site-packages (from scikit-learn) (1.2.0)
          Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\khush\anaco
          nda3\lib\site-packages (from scikit-learn) (2.2.0)
          Note: you may need to restart the kernel to use updated packages.
```



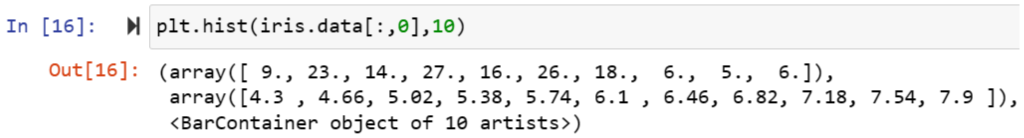
```
In [14]:  import sklearn.datasets as data

          iris = data.load_iris()
          iris

Out[14]:  {'data': array([[5.1, 3.5, 1.4, 0.2],
                  [4.9, 3. , 1.4, 0.2],
                  [4.7, 3.2, 1.3, 0.2],
                  [4.6, 3.1, 1.5, 0.2],
                  [5. , 3.6, 1.4, 0.2],
                  [5.4, 3.9, 1.7, 0.4],
                  [4.6, 3.4, 1.4, 0.3],
                  [5. , 3.4, 1.5, 0.2],
                  [4.4, 2.9, 1.4, 0.2],
                  [4.9, 3.1, 1.5, 0.1],
                  [5.4, 3.7, 1.5, 0.2],
                  [4.8, 3.4, 1.6, 0.2],
                  [4.8, 3. , 1.4, 0.1],
                  [4.3, 3. , 1.1, 0.1],
                  [5.8, 4. , 1.2, 0.2],
                  [5.7, 4.4, 1.5, 0.4],
                  [5.4, 3.9, 1.3, 0.4],
                  [5.1, 3.5, 1.4, 0.3],
                  [5.7, 3.8, 1.7, 0.3],
                  [5.1, 3.8, 1.5, 0.3],
```

- Scatter Plot:

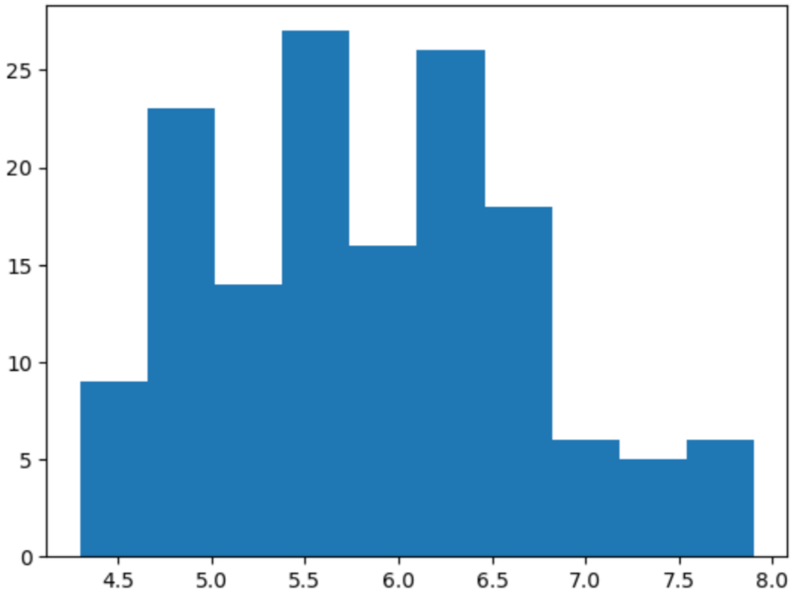


```
In [15]:  plt.scatter(iris.data[:,0],iris.data[:,1])

Out[15]:  <matplotlib.collections.PathCollection at 0x2763dbaa3d0>
```

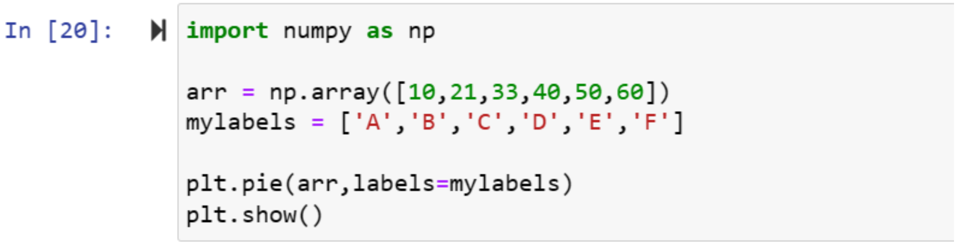
- Bar Chart/Histogram:



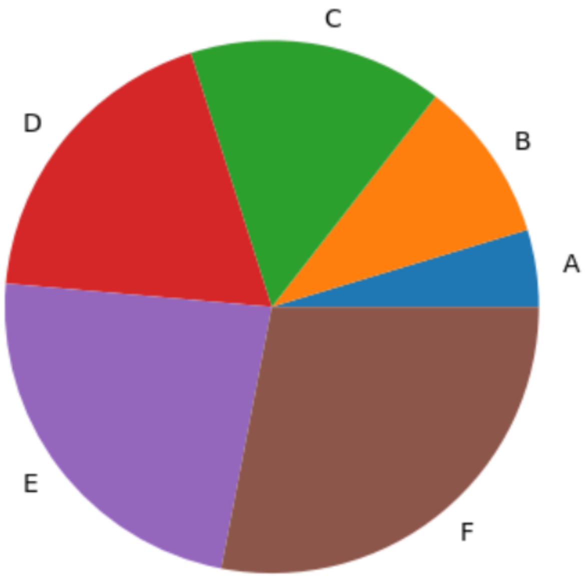
```
In [16]:  plt.hist(iris.data[:,0],10)

Out[16]: (array([ 9., 23., 14., 27., 16., 26., 18.,  6.,  5.,  6.]),
          array([4.3 , 4.66, 5.02, 5.38, 5.74, 6.1 , 6.46, 6.82, 7.18, 7.54, 7.9 ]),
          <BarContainer object of 10 artists>)
```

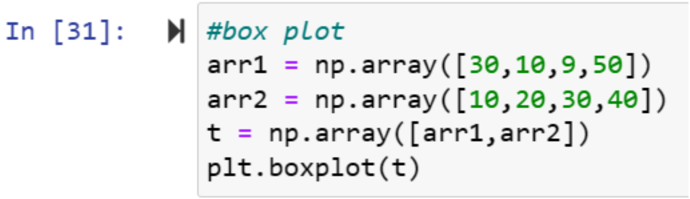


- Pie Chart:



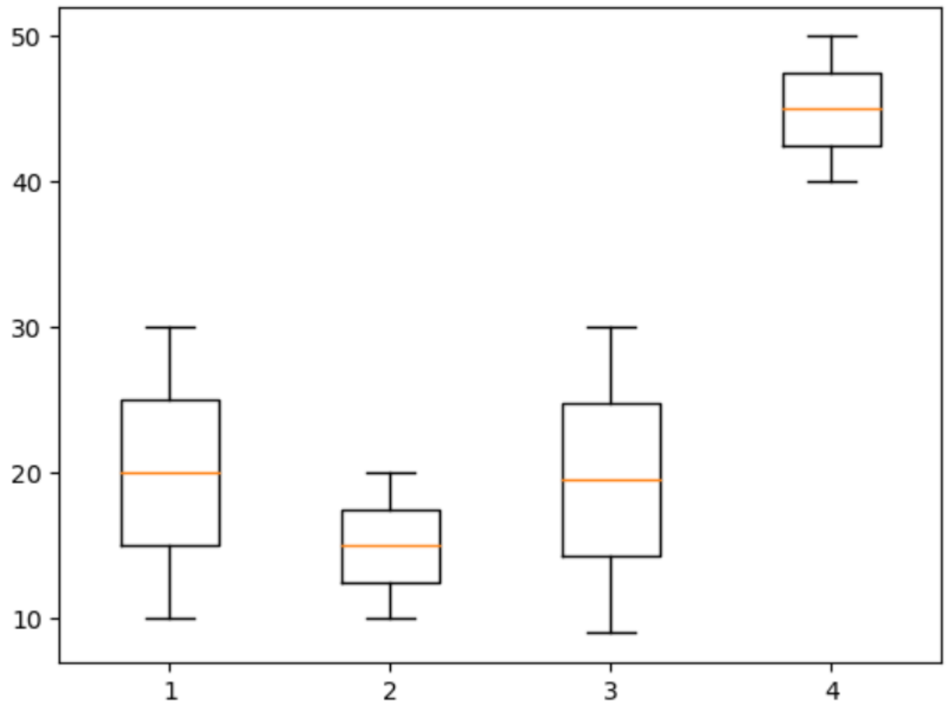
```
In [20]:  import numpy as np

          arr = np.array([10,21,33,40,50,60])
          mylabels = ['A','B','C','D','E','F']

          plt.pie(arr,labels=mylabels)
          plt.show()
```

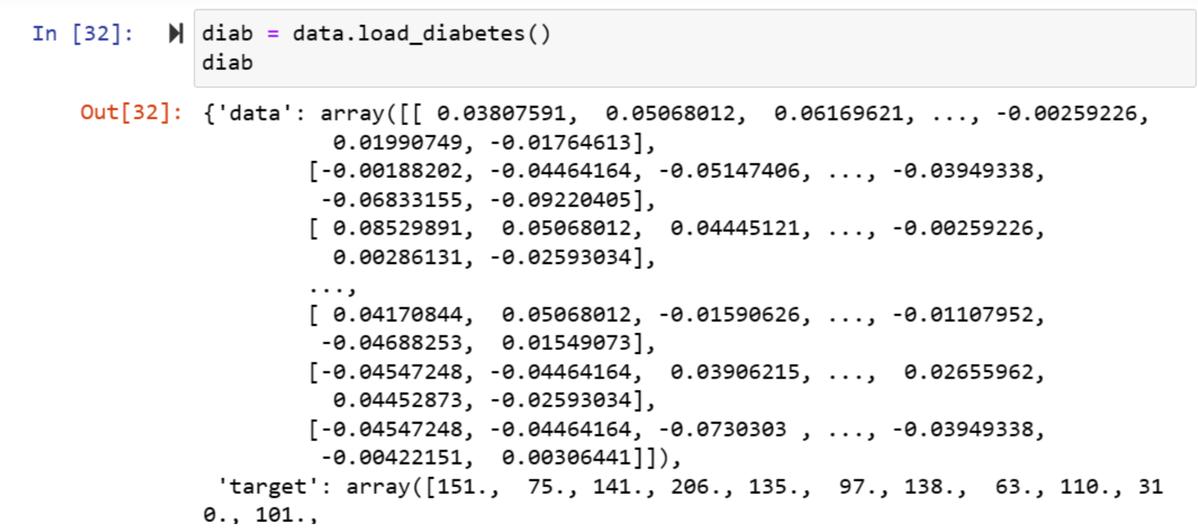




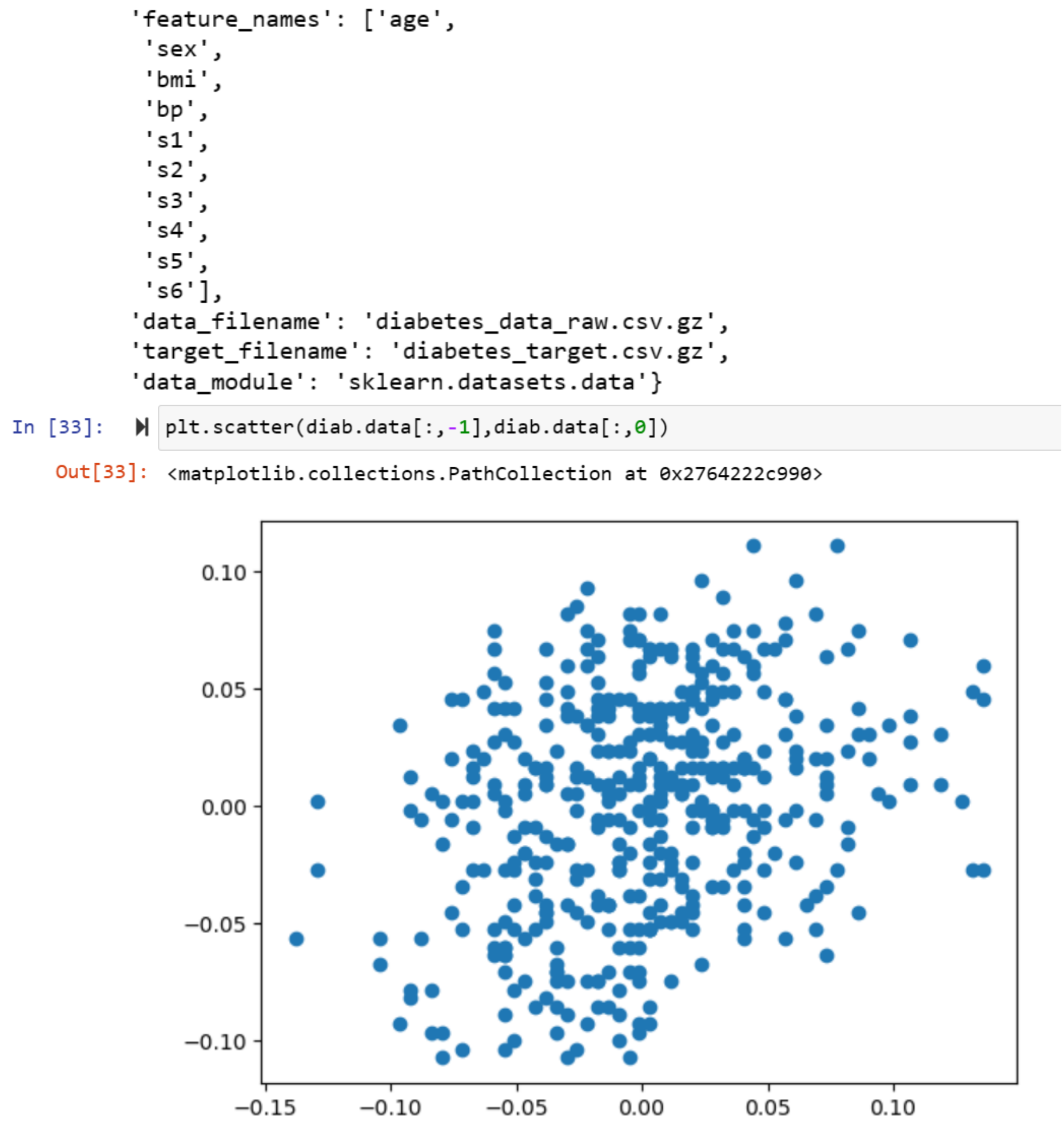
```
In [31]:  #box plot
          arr1 = np.array([30,10,9,50])
          arr2 = np.array([10,20,30,40])
          t = np.array([arr1,arr2])
          plt.boxplot(t)
```

- Box Plot:





```
In [32]:  diab = data.load_diabetes()
          diab

Out[32]: {'data': array([[ 0.03807591,  0.05068012,  0.06169621, ..., -0.00259226,
                   0.01990749, -0.01764613],
                 [-0.00188202, -0.04464164, -0.05147406, ..., -0.03949338,
                  -0.06833155, -0.09220405],
                 [ 0.08529891,  0.05068012,  0.04445121, ..., -0.00259226,
                   0.00286131, -0.02593034],
                 ...,
                 [ 0.04170844,  0.05068012, -0.01590626, ..., -0.01107952,
                  -0.04688253,  0.01549073],
                 [-0.04547248, -0.04464164,  0.03906215, ...,  0.02655962,
                   0.04452873, -0.02593034],
                 [-0.04547248, -0.04464164, -0.0730303 , ..., -0.03949338,
                  -0.00422151,  0.00306441]]),
          'target': array([151.,  75., 141., 206., 135.,  97., 138.,  63., 110., 31
         0., 101.,
```



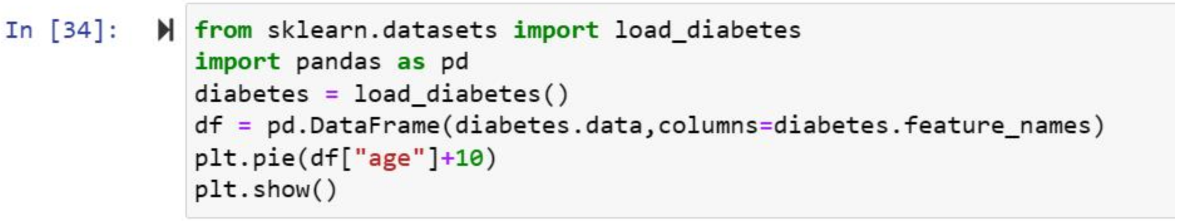
```
'feature_names': ['age',
 'sex',
 'bmi',
 'bp',
 's1',
 's2',
 's3',
 's4',
 's5',
 's6'],
'data_filename': 'diabetes_data_raw.csv.gz',
'target_filename': 'diabetes_target.csv.gz',
'data_module': 'sklearn.datasets.data'}
```

In [33]:

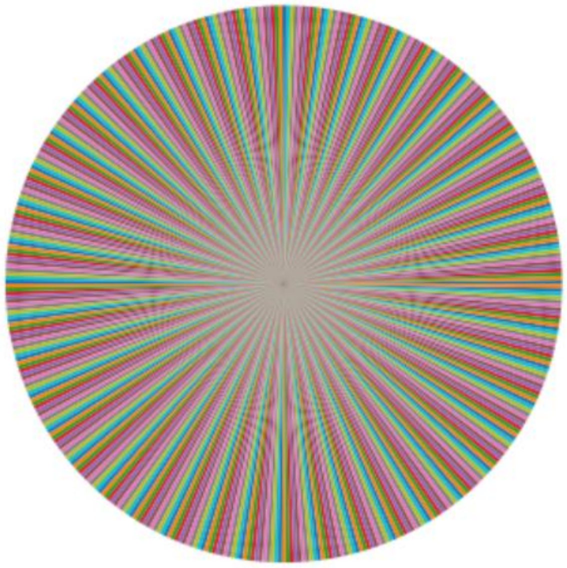
```
plt.scatter(diab.data[:,-1],diab.data[:,0])
```

Out[33]:

```
<matplotlib.collections.PathCollection at 0x2764222c990>
```



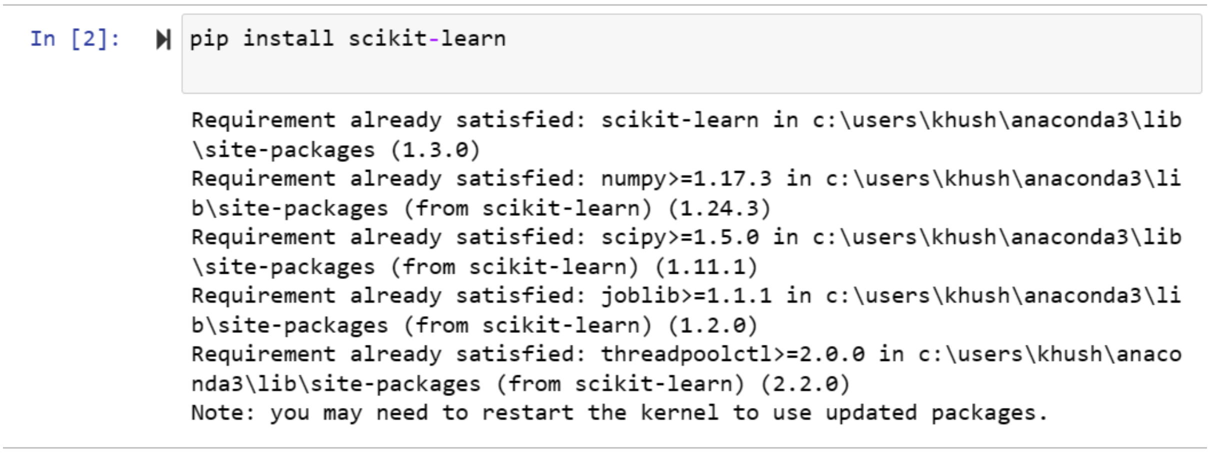
```
In [34]:  from sklearn.datasets import load_diabetes
          import pandas as pd
          diabetes = load_diabetes()
          df = pd.DataFrame(diabetes.data,columns=diabetes.feature_names)
          plt.pie(df["age"]+10)
          plt.show()
```



Experiment – 5

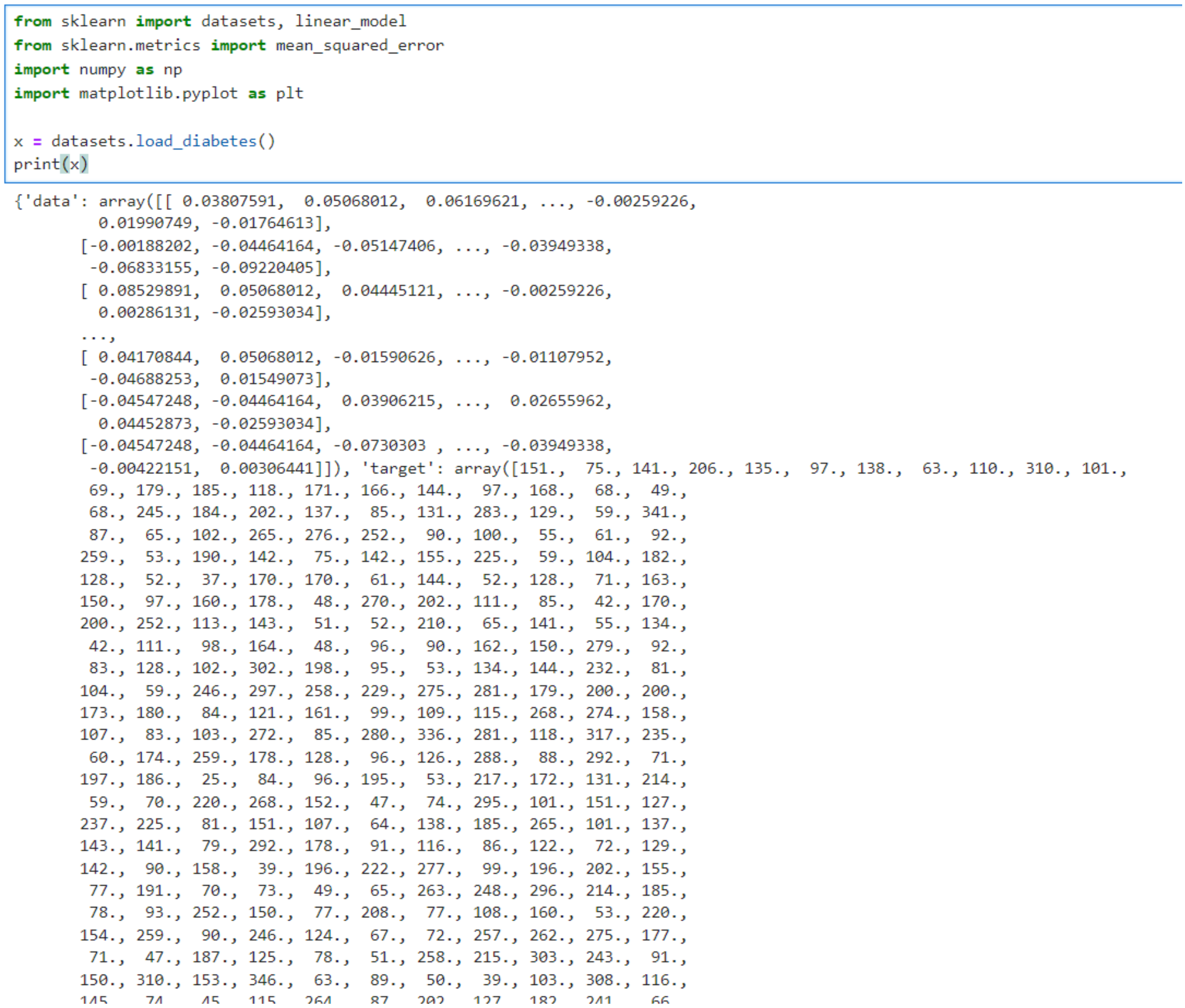
AIM: To perform Regression Analysis using sci-kit learn package in Python.

Tools/Apparatus: Anaconda Python/ Spyder IDE



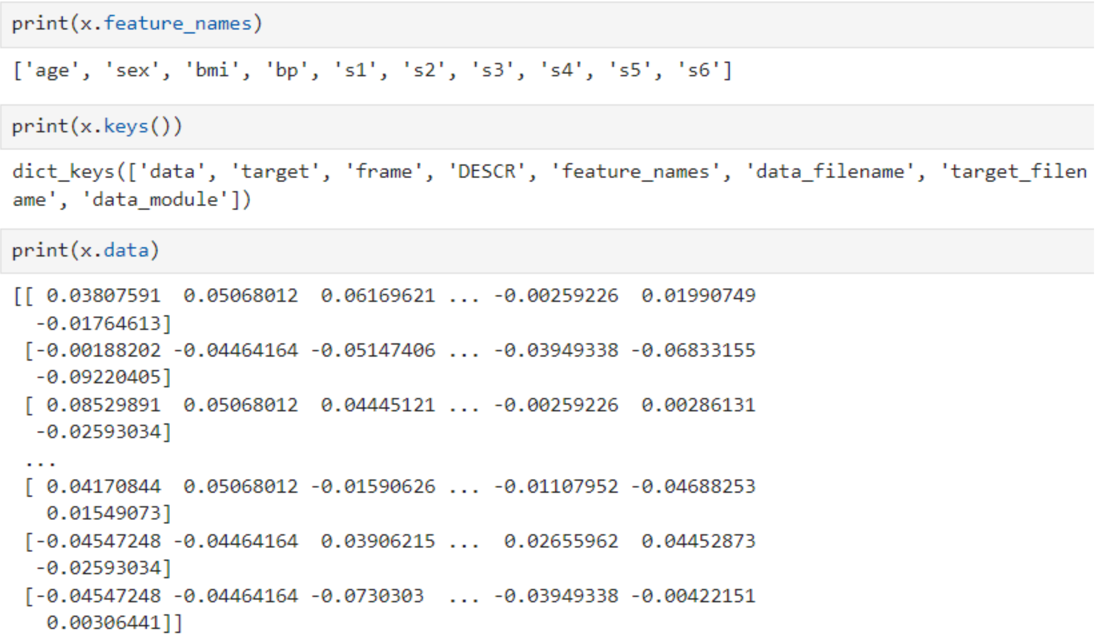
```
In [2]:  pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in c:\users\khush\anaconda3\lib
\site-packages (1.3.0)
Requirement already satisfied: numpy>=1.17.3 in c:\users\khush\anaconda3\li
b\site-packages (from scikit-learn) (1.24.3)
Requirement already satisfied: scipy>=1.5.0 in c:\users\khush\anaconda3\lib
\site-packages (from scikit-learn) (1.11.1)
Requirement already satisfied: joblib>=1.1.1 in c:\users\khush\anaconda3\li
b\site-packages (from scikit-learn) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\khush\anaco
nda3\lib\site-packages (from scikit-learn) (2.2.0)
Note: you may need to restart the kernel to use updated packages.
```



```python
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error
import numpy as np
import matplotlib.pyplot as plt

x = datasets.load_diabetes()
print(x)
```

```
{'data': array([[ 0.03807591,  0.05068012,  0.06169621, ..., -0.00259226,
         0.01990749, -0.01764613],
       [-0.00188202, -0.04464164, -0.05147406, ..., -0.03949338,
        -0.06833155, -0.09220405],
       [ 0.08529891,  0.05068012,  0.04445121, ..., -0.00259226,
         0.00286131, -0.02593034],
       ...,
       [ 0.04170844,  0.05068012, -0.01590626, ..., -0.01107952,
        -0.04688253,  0.01549073],
       [-0.04547248, -0.04464164,  0.03906215, ...,  0.02655962,
         0.04452873, -0.02593034],
       [-0.04547248, -0.04464164, -0.0730303 , ..., -0.03949338,
        -0.00422151,  0.00306441]]), 'target': array([151.,  75., 141., 206., 135.,  97., 138.,  63., 110., 310., 101.,
        69., 179., 185., 118., 171., 166., 144.,  97., 168.,  68.,  49.,
        68., 245., 184., 202., 137.,  85., 131., 283., 129.,  59., 341.,
        87.,  65., 102., 265., 276., 252.,  90., 100.,  55.,  61.,  92.,
       259.,  53., 190., 142.,  75., 142., 155., 225.,  59., 104., 182.,
       128.,  52.,  37., 170., 170.,  61., 144.,  52., 128.,  71., 163.,
       150.,  97., 160., 178.,  48., 270., 202., 111.,  85.,  42., 170.,
       200., 252., 113., 143.,  51.,  52., 210.,  65., 141.,  55., 134.,
        42., 111.,  98., 164.,  48.,  96.,  90., 162., 150., 279.,  92.,
        83., 128., 102., 302., 198.,  95.,  53., 134., 144., 232.,  81.,
       104.,  59., 246., 297., 258., 229., 275., 281., 179., 200., 200.,
       173., 180.,  84., 121., 161.,  99., 109., 115., 268., 274., 158.,
       107.,  83., 103., 272.,  85., 280., 336., 281., 118., 317., 235.,
        60., 174., 259., 178., 128.,  96., 126., 288.,  88., 292.,  71.,
       197., 186.,  25.,  84.,  96., 195.,  53., 217., 172., 131., 214.,
        59.,  70., 220., 268., 152.,  47.,  74., 295., 101., 151., 127.,
       237., 225.,  81., 151., 107.,  64., 138., 185., 265., 101., 137.,
       143., 141.,  79., 292., 178.,  91., 116.,  86., 122.,  72., 129.,
       142.,  90., 158.,  39., 196., 222., 277.,  99., 196., 202., 155.,
        77., 191.,  70.,  73.,  49.,  65., 263., 248., 296., 214., 185.,
        78.,  93., 252., 150.,  77., 208.,  77., 108., 160.,  53., 220.,
       154., 259.,  90., 246., 124.,  67.,  72., 257., 262., 275., 177.,
        71.,  47., 187., 125.,  78.,  51., 258., 215., 303., 243.,  91.,
       150., 310., 153., 346.,  63.,  89.,  50.,  39., 103., 308., 116.,
       145    74    45   115   264    87   202   127   182   241    66
```



```
print(x.feature_names)
```

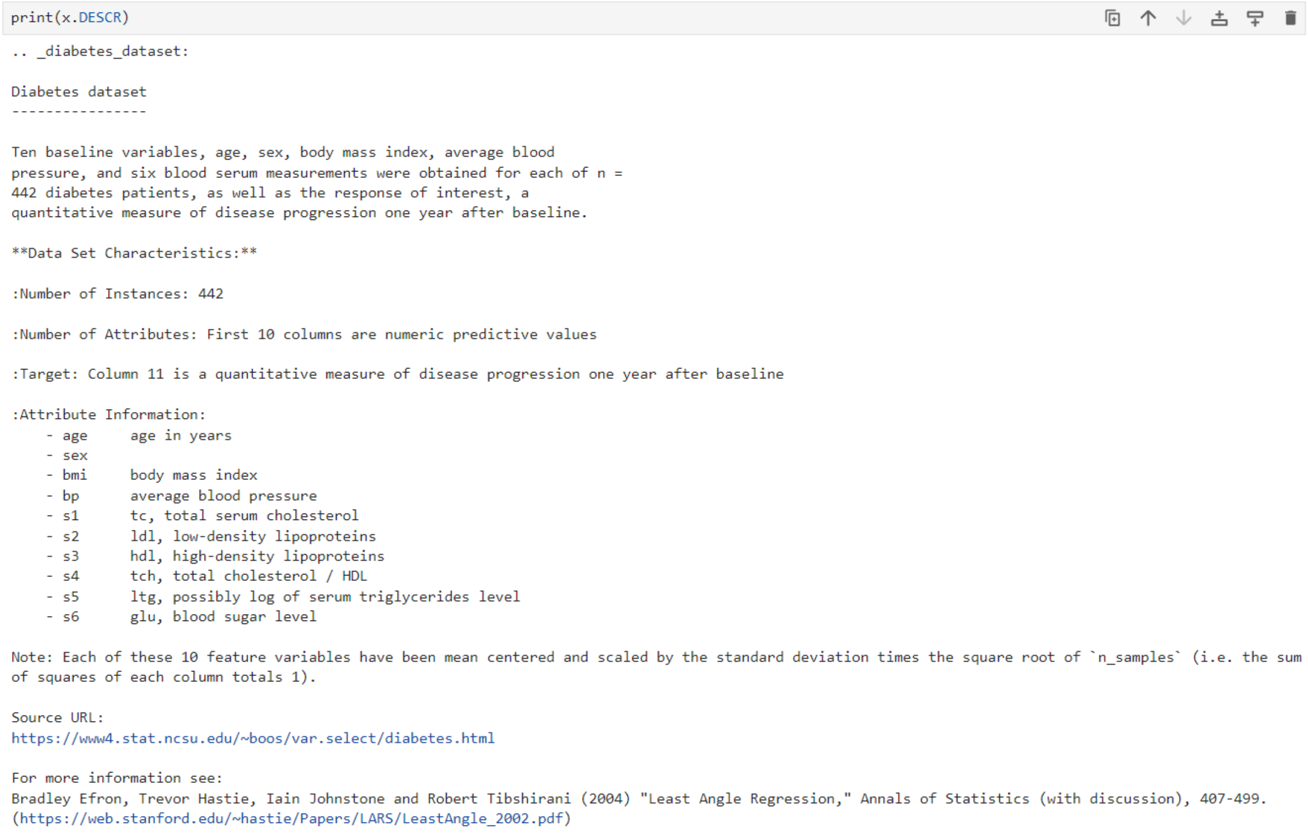
```
['age', 'sex', 'bmi', 'bp', 's1', 's2', 's3', 's4', 's5', 's6']
```

```
print(x.keys())
```

```
dict_keys(['data', 'target', 'frame', 'DESCR', 'feature_names', 'data_filename', 'target_filen
ame', 'data_module'])
```

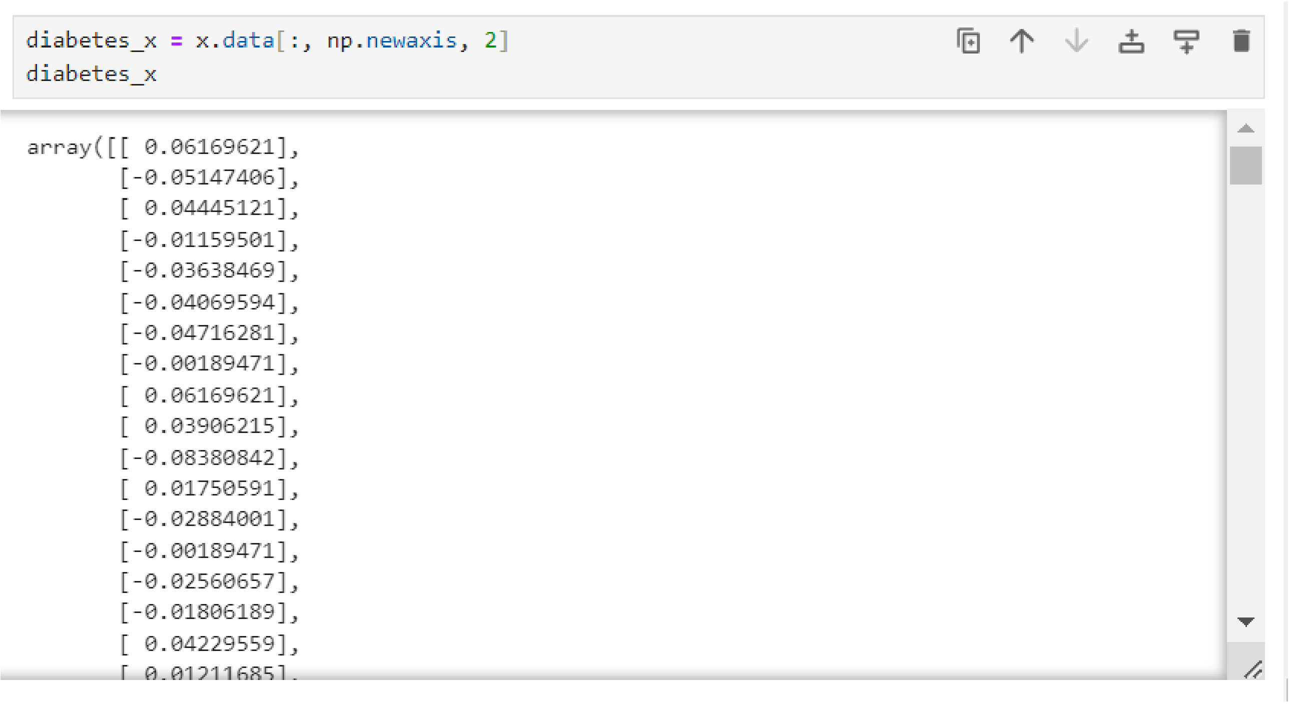
```
print(x.data)
```

```
[[ 0.03807591  0.05068012  0.06169621 ... -0.00259226  0.01990749
  -0.01764613]
 [-0.00188202 -0.04464164 -0.05147406 ... -0.03949338 -0.06833155
  -0.09220405]
 [ 0.08529891  0.05068012  0.04445121 ... -0.00259226  0.00286131
  -0.02593034]
 ...
 [ 0.04170844  0.05068012 -0.01590626 ... -0.01107952 -0.04688253
   0.01549073]
 [-0.04547248 -0.04464164  0.03906215 ...  0.02655962  0.04452873
  -0.02593034]
 [-0.04547248 -0.04464164 -0.0730303  ... -0.03949338 -0.00422151
   0.00306441]]
```



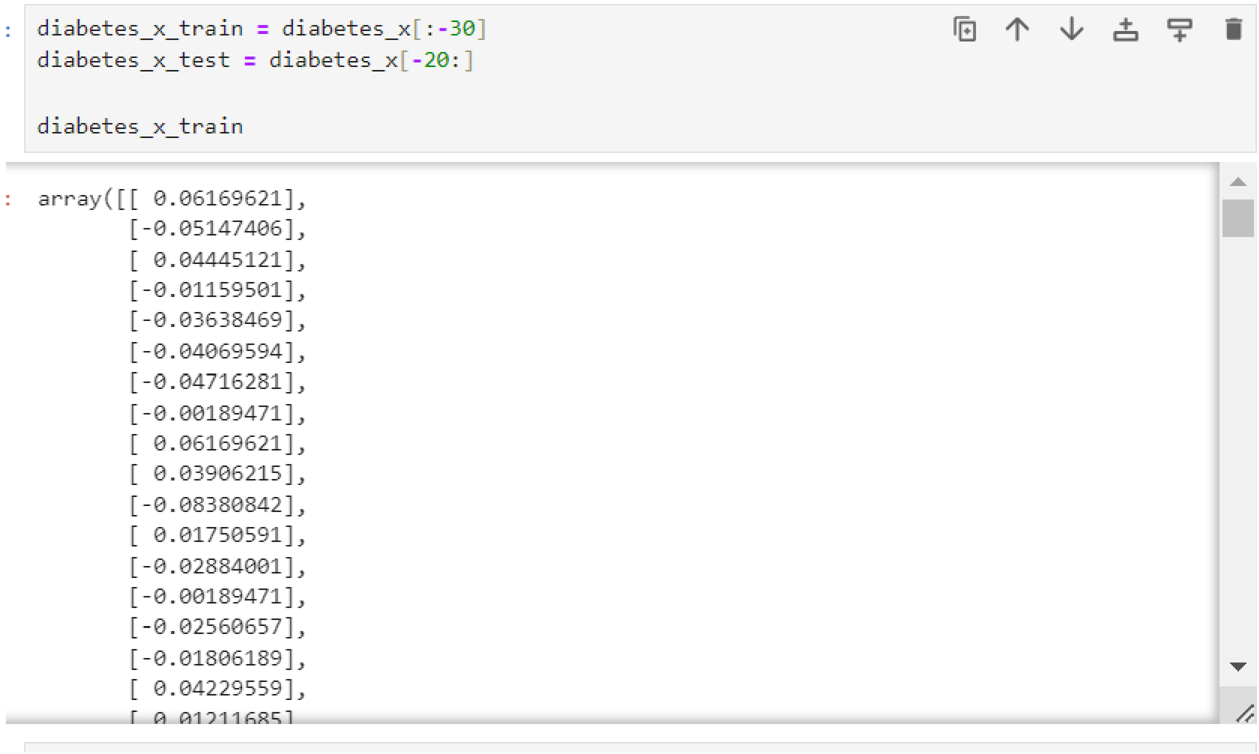
```
print(x.DESCR)
```

```
.. _diabetes_dataset:

Diabetes dataset
----------------

Ten baseline variables, age, sex, body mass index, average blood
pressure, and six blood serum measurements were obtained for each of n =
442 diabetes patients, as well as the response of interest, a
quantitative measure of disease progression one year after baseline.

**Data Set Characteristics:**

:Number of Instances: 442

:Number of Attributes: First 10 columns are numeric predictive values

:Target: Column 11 is a quantitative measure of disease progression one year after baseline

:Attribute Information:
    - age     age in years
    - sex
    - bmi     body mass index
    - bp      average blood pressure
    - s1      tc, total serum cholesterol
    - s2      ldl, low-density lipoproteins
    - s3      hdl, high-density lipoproteins
    - s4      tch, total cholesterol / HDL
    - s5      ltg, possibly log of serum triglycerides level
    - s6      glu, blood sugar level

Note: Each of these 10 feature variables have been mean centered and scaled by the standard deviation times the square root of `n_samples` (i.e. the sum
of squares of each column totals 1).

Source URL:
https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html

For more information see:
Bradley Efron, Trevor Hastie, Iain Johnstone and Robert Tibshirani (2004) "Least Angle Regression," Annals of Statistics (with discussion), 407-499.
(https://web.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.pdf)
```



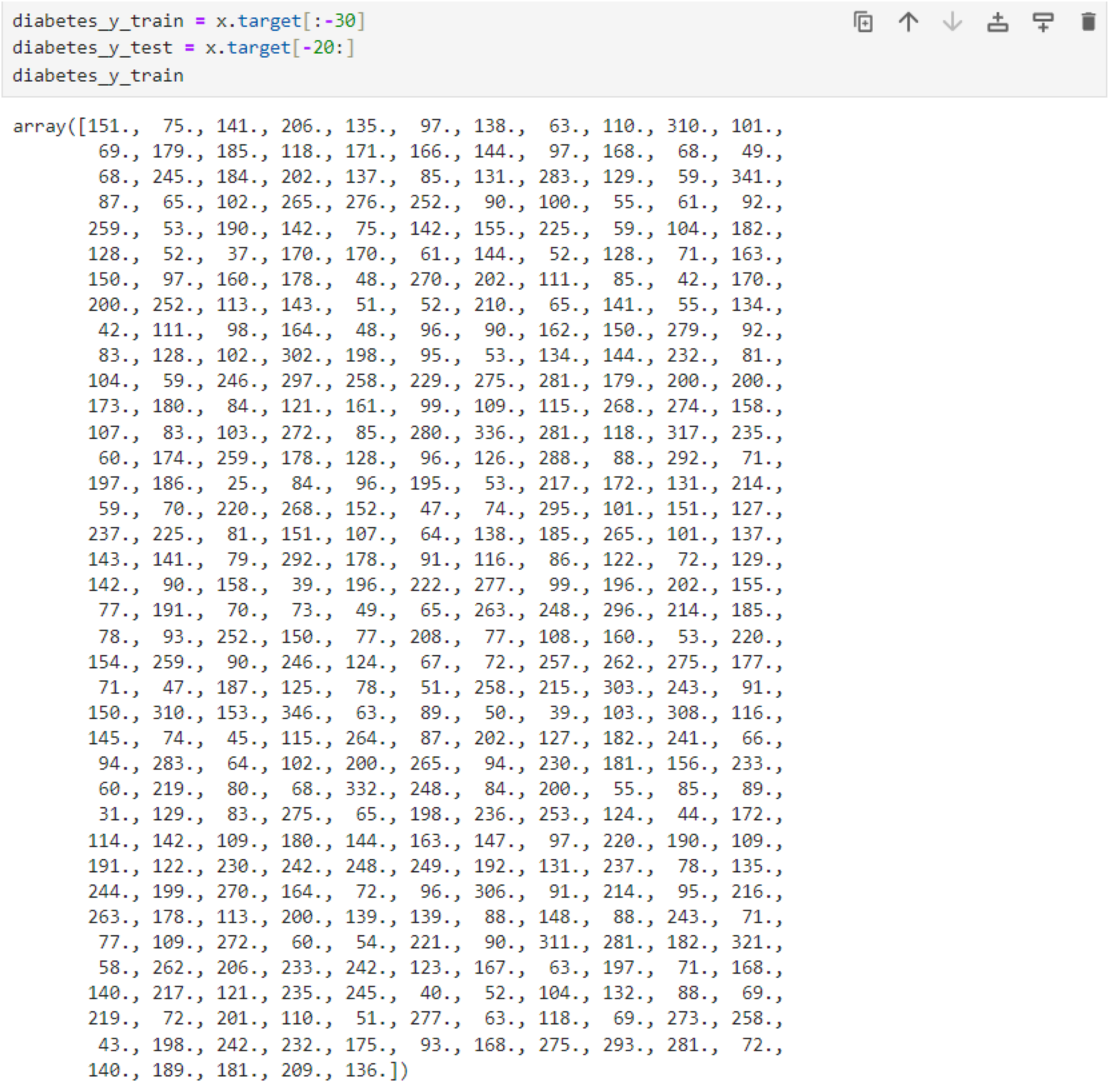
```
diabetes_x = x.data[:, np.newaxis, 2]
diabetes_x
```

```
array([[ 0.06169621],
       [-0.05147406],
       [ 0.04445121],
       [-0.01159501],
       [-0.03638469],
       [-0.04069594],
       [-0.04716281],
       [-0.00189471],
       [ 0.06169621],
       [ 0.03906215],
       [-0.08380842],
       [ 0.01750591],
       [-0.02884001],
       [-0.00189471],
       [-0.02560657],
       [-0.01806189],
       [ 0.04229559],
       [ 0.01211685],
```



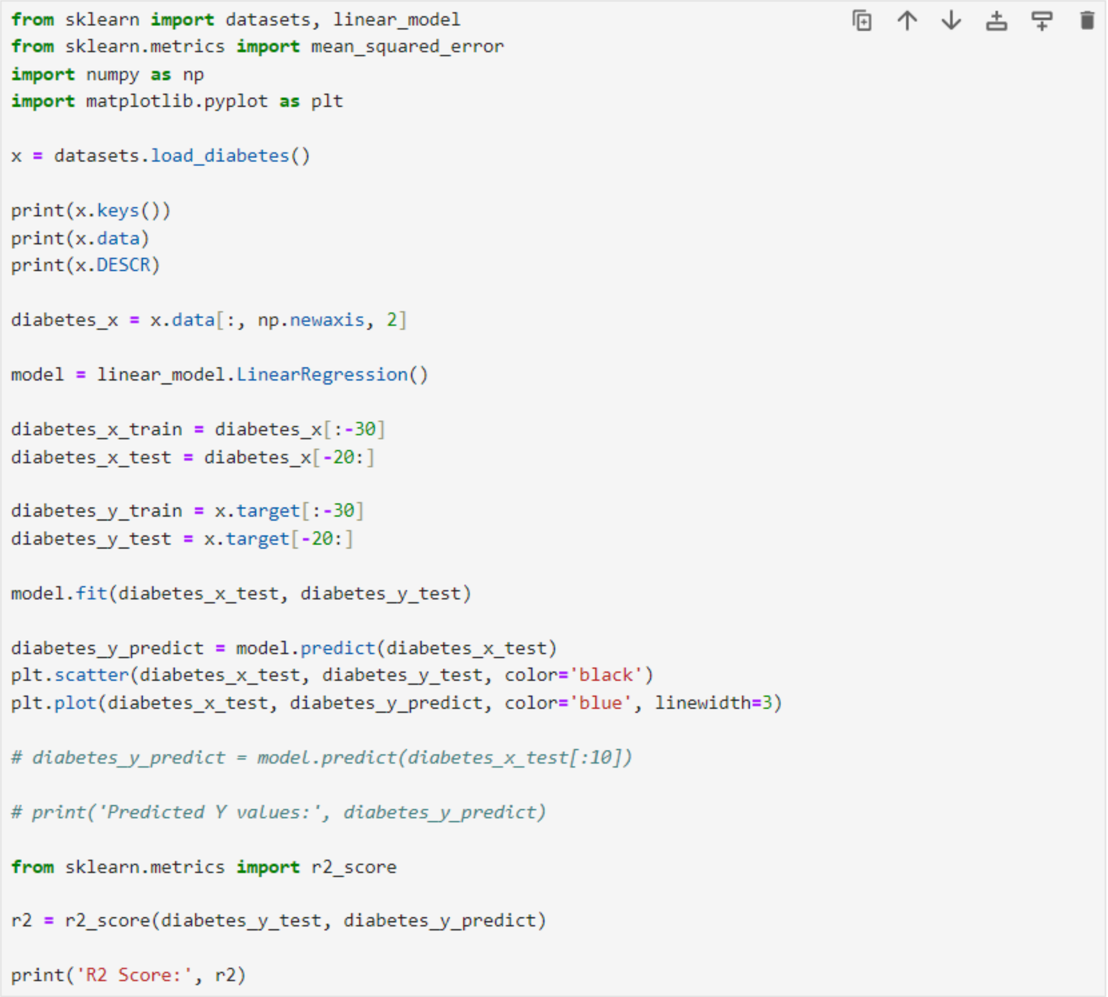
```
diabetes_x_train = diabetes_x[:-30]
diabetes_x_test = diabetes_x[-20:]

diabetes_x_train
```

```
array([[ 0.06169621],
       [-0.05147406],
       [ 0.04445121],
       [-0.01159501],
       [-0.03638469],
       [-0.04069594],
       [-0.04716281],
       [-0.00189471],
       [ 0.06169621],
       [ 0.03906215],
       [-0.08380842],
       [ 0.01750591],
       [-0.02884001],
       [-0.00189471],
       [-0.02560657],
       [-0.01806189],
       [ 0.04229559],
       [ 0.01211685]
```



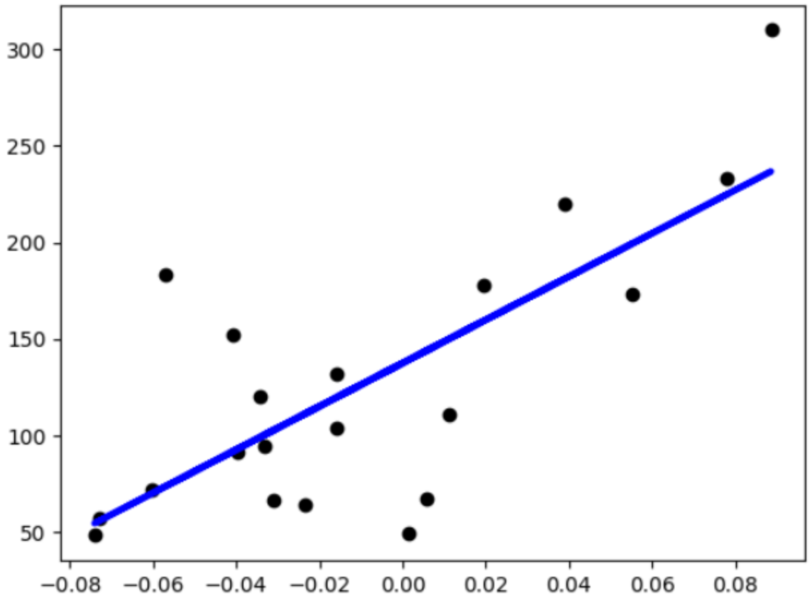
```
diabetes_y_train = x.target[:-30]
diabetes_y_test = x.target[-20:]
diabetes_y_train
```

```
array([151.,  75., 141., 206., 135.,  97., 138.,  63., 110., 310., 101.,
        69., 179., 185., 118., 171., 166., 144.,  97., 168.,  68.,  49.,
        68., 245., 184., 202., 137.,  85., 131., 283., 129.,  59., 341.,
        87.,  65., 102., 265., 276., 252.,  90., 100.,  55.,  61.,  92.,
       259.,  53., 190., 142.,  75., 142., 155., 225.,  59., 104., 182.,
       128.,  52.,  37., 170., 170.,  61., 144.,  52., 128.,  71., 163.,
       150.,  97., 160., 178.,  48., 270., 202., 111.,  85.,  42., 170.,
       200., 252., 113., 143.,  51.,  52., 210.,  65., 141.,  55., 134.,
        42., 111.,  98., 164.,  48.,  96.,  90., 162., 150., 279.,  92.,
        83., 128., 102., 302., 198.,  95.,  53., 134., 144., 232.,  81.,
       104.,  59., 246., 297., 258., 229., 275., 281., 179., 200., 200.,
       173., 180.,  84., 121., 161.,  99., 109., 115., 268., 274., 158.,
       107.,  83., 103., 272.,  85., 280., 336., 281., 118., 317., 235.,
        60., 174., 259., 178., 128.,  96., 126., 288.,  88., 292.,  71.,
       197., 186.,  25.,  84.,  96., 195.,  53., 217., 172., 131., 214.,
        59.,  70., 220., 268., 152.,  47.,  74., 295., 101., 151., 127.,
       237., 225.,  81., 151., 107.,  64., 138., 185., 265., 101., 137.,
       143., 141.,  79., 292., 178.,  91., 116.,  86., 122.,  72., 129.,
       142.,  90., 158.,  39., 196., 222., 277.,  99., 196., 202., 155.,
        77., 191.,  70.,  73.,  49.,  65., 263., 248., 296., 214., 185.,
        78.,  93., 252., 150.,  77., 208.,  77., 108., 160.,  53., 220.,
       154., 259.,  90., 246., 124.,  67.,  72., 257., 262., 275., 177.,
        71.,  47., 187., 125.,  78.,  51., 258., 215., 303., 243.,  91.,
       150., 310., 153., 346.,  63.,  89.,  50.,  39., 103., 308., 116.,
       145.,  74.,  45., 115., 264.,  87., 202., 127., 182., 241.,  66.,
        94., 283.,  64., 102., 200., 265.,  94., 230., 181., 156., 233.,
        60., 219.,  80.,  68., 332., 248.,  84., 200.,  55.,  85.,  89.,
        31., 129.,  83., 275.,  65., 198., 236., 253., 124.,  44., 172.,
       114., 142., 109., 180., 144., 163., 147.,  97., 220., 190., 109.,
       191., 122., 230., 242., 248., 249., 192., 131., 237.,  78., 135.,
       244., 199., 270., 164.,  72.,  96., 306.,  91., 214.,  95., 216.,
       263., 178., 113., 200., 139., 139.,  88., 148.,  88., 243.,  71.,
        77., 109., 272.,  60.,  54., 221.,  90., 311., 281., 182., 321.,
        58., 262., 206., 233., 242., 123., 167.,  63., 197.,  71., 168.,
       140., 217., 121., 235., 245.,  40.,  52., 104., 132.,  88.,  69.,
       219.,  72., 201., 110.,  51., 277.,  63., 118.,  69., 273., 258.,
        43., 198., 242., 232., 175.,  93., 168., 275., 293., 281.,  72.,
       140., 189., 181., 209., 136.])
```



```
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error
import numpy as np
import matplotlib.pyplot as plt

x = datasets.load_diabetes()

print(x.keys())
print(x.data)
print(x.DESCR)

diabetes_x = x.data[:, np.newaxis, 2]

model = linear_model.LinearRegression()

diabetes_x_train = diabetes_x[:-30]
diabetes_x_test = diabetes_x[-20:]

diabetes_y_train = x.target[:-30]
diabetes_y_test = x.target[-20:]

model.fit(diabetes_x_test, diabetes_y_test)

diabetes_y_predict = model.predict(diabetes_x_test)
plt.scatter(diabetes_x_test, diabetes_y_test, color='black')
plt.plot(diabetes_x_test, diabetes_y_predict, color='blue', linewidth=3)

# diabetes_y_predict = model.predict(diabetes_x_test[:10])

# print('Predicted Y values:', diabetes_y_predict)

from sklearn.metrics import r2_score

r2 = r2_score(diabetes_y_test, diabetes_y_predict)

print('R2 Score:', r2)
```

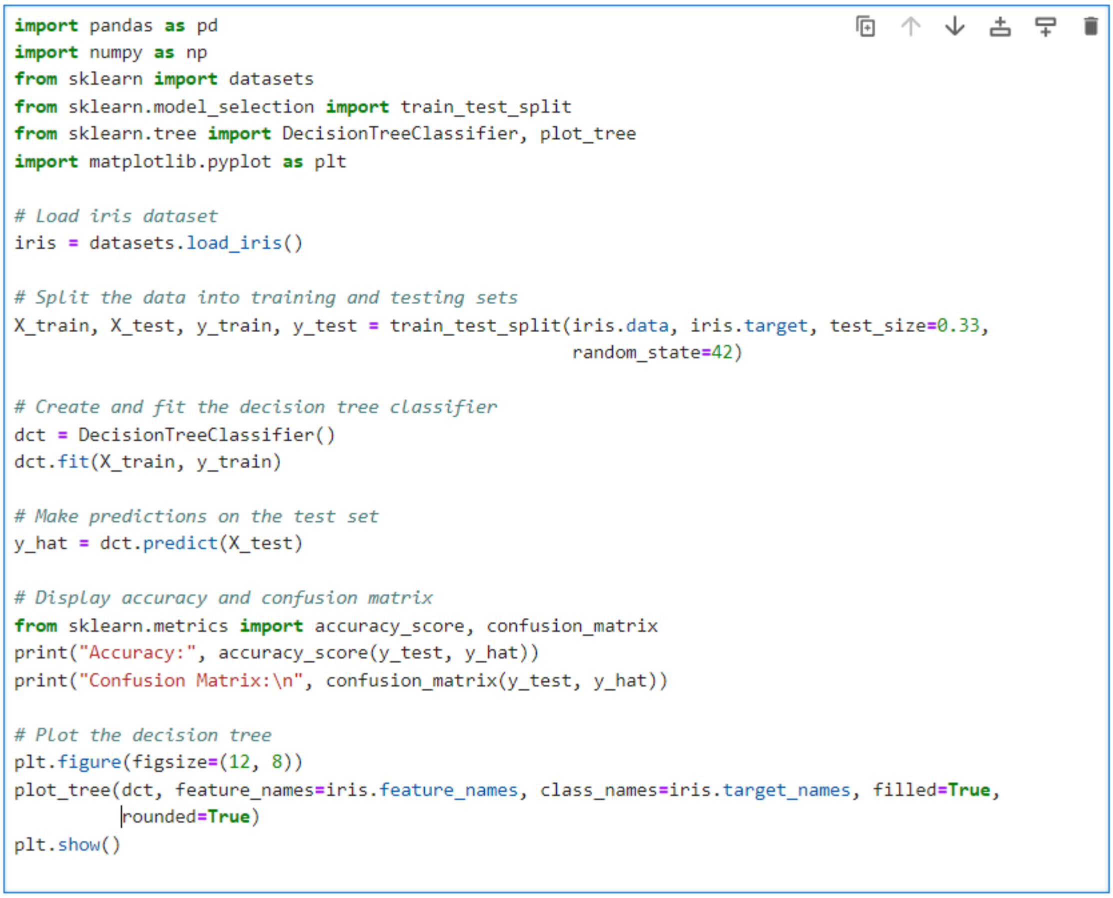
```
R2 Score: 0.5495903864515435
```



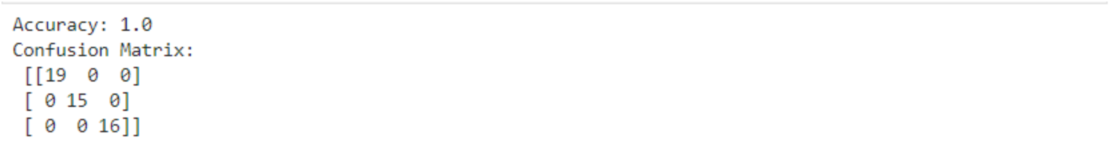
Experiment – 6

AIM: To perform Decision Tree Classification(DCT) using sklearn package in python.

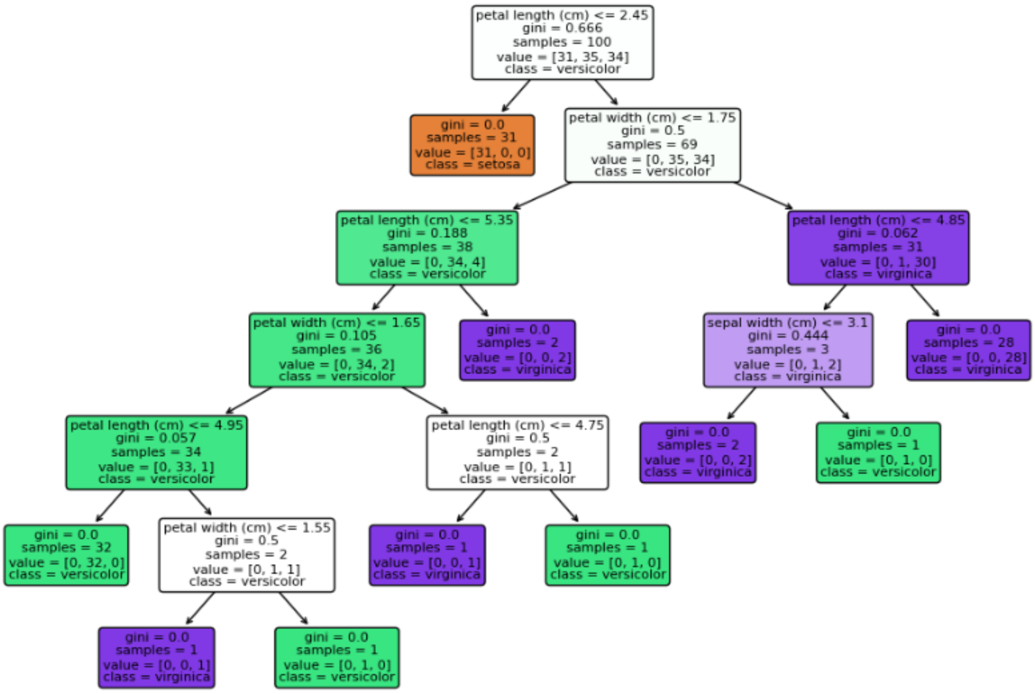
Tools/Apparatus: Anaconda Python/ Spyder IDE

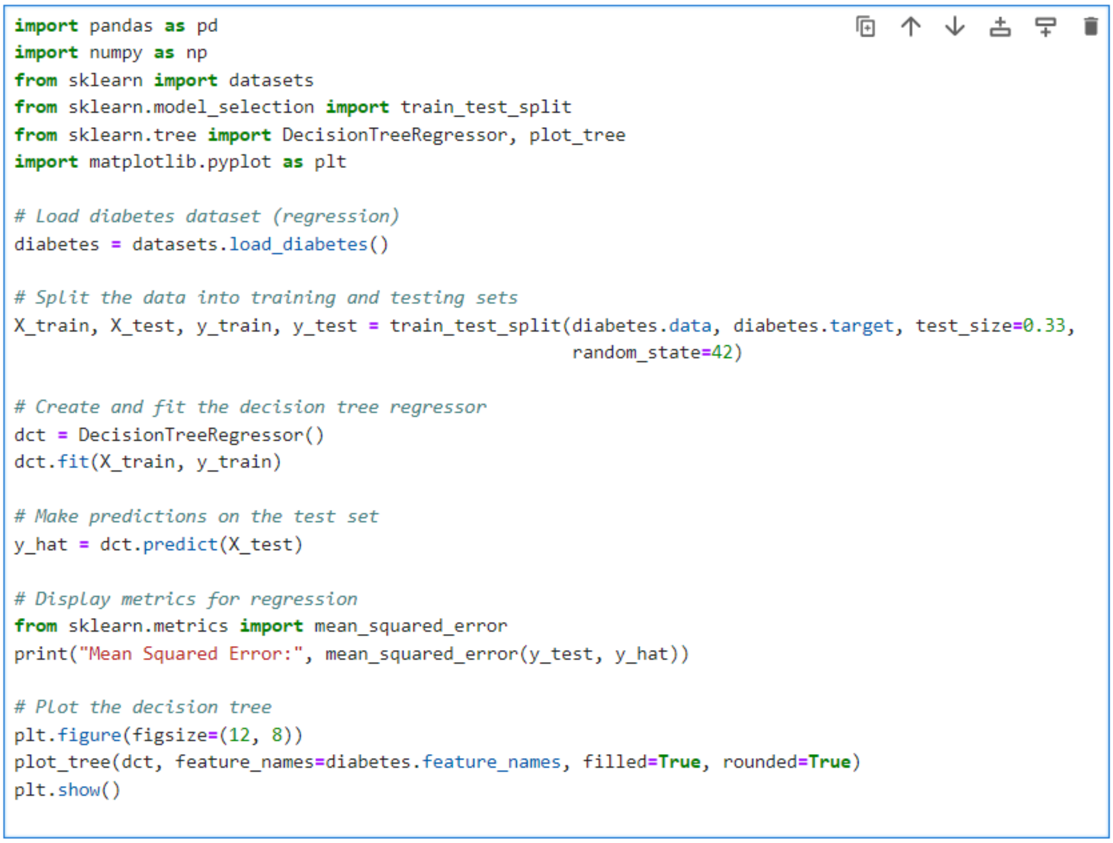


```python
import pandas as pd
import numpy as np
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
import matplotlib.pyplot as plt

# Load iris dataset
iris = datasets.load_iris()

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target, test_size=0.33,
                                                    random_state=42)

# Create and fit the decision tree classifier
dct = DecisionTreeClassifier()
dct.fit(X_train, y_train)

# Make predictions on the test set
y_hat = dct.predict(X_test)

# Display accuracy and confusion matrix
from sklearn.metrics import accuracy_score, confusion_matrix
print("Accuracy:", accuracy_score(y_test, y_hat))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_hat))

# Plot the decision tree
plt.figure(figsize=(12, 8))
plot_tree(dct, feature_names=iris.feature_names, class_names=iris.target_names, filled=True,
          rounded=True)
plt.show()
```

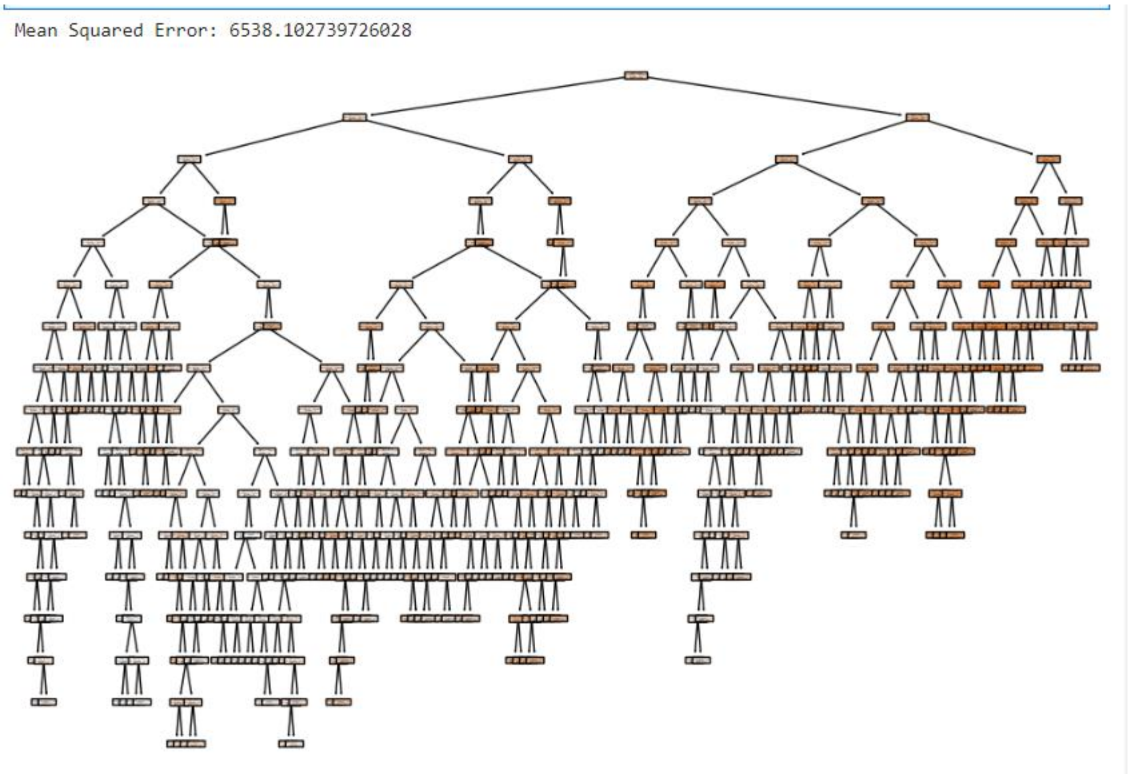


```
Accuracy: 1.0
Confusion Matrix:
 [[19  0  0]
 [ 0 15  0]
 [ 0  0 16]]
```





```python
import pandas as pd
import numpy as np
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor, plot_tree
import matplotlib.pyplot as plt

# Load diabetes dataset (regression)
diabetes = datasets.load_diabetes()

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(diabetes.data, diabetes.target, test_size=0.33,
                                                    random_state=42)

# Create and fit the decision tree regressor
dct = DecisionTreeRegressor()
dct.fit(X_train, y_train)

# Make predictions on the test set
y_hat = dct.predict(X_test)

# Display metrics for regression
from sklearn.metrics import mean_squared_error
print("Mean Squared Error:", mean_squared_error(y_test, y_hat))

# Plot the decision tree
plt.figure(figsize=(12, 8))
plot_tree(dct, feature_names=diabetes.feature_names, filled=True, rounded=True)
plt.show()
```



```
Mean Squared Error: 6538.102739726028
```