```
import pandas as pd
```

SERIES

```
mydata1=["ravi","uma","ria","maria"]
ser1=pd.Series(mydata1)
print(ser1)
```

```
0     ravi
1      uma
2      ria
3    maria
dtype: object
```

```
mydata1=["ravi","uma","ria","maria"]
roll=["A","B","C","D"]
ser2=pd.Series(mydata1,roll)
print(ser2)
```

```
A     ravi
B      uma
C      ria
D    maria
dtype: object
```

```
print(ser1[0])
```

```
ravi
```

```
ser1[1]
```

```
'uma'
```

```
ser2.to_csv=(r"C:\Priya\Mydata1.csv")
```

DATAFRAMES

```
mydict={"Names":["Virat","Rohit","Rishab","ABD"],
        "Age":[25,30,28,29,34],
        "Place":["Delhi","Mumbai","odisha","US"]
}
print(mydict)
```

```
{'Names': ['Virat', 'Rohit', 'Rishab', 'ABD'], 'Age': [25, 30, 28, 29,
34], 'Place': ['Delhi', 'Mumbai', 'odisha', 'US']}
```

```
df1=pd.read_csv(r"C:\Users\User\Documents\python02\sampledata.csv")
print(df1)
```

```
       Name Dept  Sem1  Sem2  Sem3
0       Sam  iSE   6.7  6.70   8.1
```

```
1     Priya  iSE   9.0  7.29   8.5
2    Ananya  iSE   7.9  7.30   8.9
3  Srinidhi  iSE   8.9  7.50   8.4
4      Jeev  Ise   9.5  8.90   7.9
```

```
df1.head()
```

```
       Name Dept  Sem1  Sem2  Sem3
0       Sam  iSE   6.7  6.70   8.1
1     Priya  iSE   9.0  7.29   8.5
2    Ananya  iSE   7.9  7.30   8.9
3  Srinidhi  iSE   8.9  7.50   8.4
4      Jeev  Ise   9.5  8.90   7.9
```

```
diab_df=pd.read_csv(r"C:\Users\User\Documents\python02\
diabetcsvsmall.csv")
diab_df.head()
```

```
   preg  plas  pres  skin  insu  mass   pedi  age            class
0   6.0   148  72.0  35.0     0  33.6  0.627   50  tested_positive
1   1.0    85  66.0  29.0     0  26.6  0.351   31  tested_negative
2   8.0   183  64.0   0.0     0  23.3  0.672   32  tested_positive
3   1.0    89  66.0  23.0    94  28.1  0.167   21  tested_negative
4   0.0   137  40.0  35.0   168  43.1  2.288   33  tested_positive
```

```
diab_df.tail()
```

```
     preg  plas  pres  skin  insu  mass   pedi  age            class
97    1.0    71  48.0   NaN    76  20.4  0.323   22  tested_negative
98    6.0    93  50.0  30.0    64  28.7  0.356   23  tested_negative
99    NaN   122  90.0  51.0   220  49.7  0.325   31  tested_positive
100   1.0   163  72.0   0.0     0  39.0  1.222   33  tested_positive
101   1.0   151  60.0   0.0     0  26.1  0.179   22  tested_negative
```

ACCESSING

```
diab_df.loc[12:19,"age"]
```

```
12    57
13    59
14    51
15    32
16    31
17    31
18    33
19    32
Name: age, dtype: int64
```

```
diab_df.loc[12:19]
```

```
    preg  plas  pres  skin  insu  mass   pedi  age            class
12  10.0   139  80.0   0.0     0  27.1  1.441   57  tested_negative
13   1.0   189  60.0  23.0   846  30.1  0.398   59  tested_positive
14   5.0   166  72.0  19.0   175  25.8  0.587   51  tested_positive
15   7.0   100   0.0   0.0     0  30.0  0.484   32  tested_positive
16   0.0   118  84.0  47.0   230  45.8  0.551   31  tested_positive
17   7.0   107  74.0   0.0     0  29.6  0.254   31  tested_positive
18   1.0   103  30.0  38.0    83  43.3  0.183   33  tested_negative
19   1.0   115  70.0  30.0    96  34.6  0.529   32  tested_positive
```

```
diab_df.loc[3:8]
```

```
   preg  plas  pres  skin  insu  mass   pedi  age            class
3   1.0    89  66.0  23.0    94  28.1  0.167   21  tested_negative
4   0.0   137  40.0  35.0   168  43.1  2.288   33  tested_positive
5   5.0   116  74.0   0.0     0  25.6  0.201   30  tested_negative
6   3.0    78  50.0  32.0    88  31.0  0.248   26  tested_positive
7  10.0   115   0.0   0.0     0  35.3  0.134   29  tested_negative
8   2.0   197  70.0  45.0   543  30.5  0.158   53  tested_positive
```

```
diab_df.iloc[12:19, 3:8] #dataframe .iloc[row_range, column_range]
```

```
    skin  insu  mass   pedi  age
12   0.0     0  27.1  1.441   57
13  23.0   846  30.1  0.398   59
14  19.0   175  25.8  0.587   51
15   0.0     0  30.0  0.484   32
16  47.0   230  45.8  0.551   31
17   0.0     0  29.6  0.254   31
18  38.0    83  43.3  0.183   33
```

FEATURE ENGINEERING

preg ,plas.insu,skin,mass,pedi,age,==> independent(Feature) class==> dependent(Target)

```
diab_df.rename(columns={"plas":"glocouse"},inplace= True)
diab_df.head()
diab_df.rename(columns={"glocouse":"preg1",1},inplace= True)
diab_df.head()
```

```
  Cell In[106], line 3
    diab_df.rename(columns={"glocouse":"preg1",1},inplace= True)
                                               ^
SyntaxError: ':' expected after dictionary key
```

```
diab_df['glucose_in_mmol']=diab_df['glocouse']/18.018
```

```
---------------------------------------------------------------------
-----
KeyError                                  Traceback (most recent call
last)
File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3805,
in Index.get_loc(self, key)
   3804 try:
-> 3805     return self._engine.get_loc(casted_key)
   3806 except KeyError as err:

File index.pyx:167, in pandas._libs.index.IndexEngine.get_loc()

File index.pyx:191, in pandas._libs.index.IndexEngine.get_loc()

File index.pyx:234, in
pandas._libs.index.IndexEngine._get_loc_duplicates()

File index.pyx:242, in
pandas._libs.index.IndexEngine._maybe_get_bool_indexer()

File index.pyx:134, in pandas._libs.index._unpack_bool_indexer()

KeyError: 'glocouse'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call
last)
Cell In[104], line 1
----> 1 diab_df['glucose_in_mmol']=diab_df['glocouse']/18.018

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:4102, in
DataFrame.__getitem__(self, key)
   4100 if self.columns.nlevels > 1:
   4101     return self._getitem_multilevel(key)
-> 4102 indexer = self.columns.get_loc(key)
   4103 if is_integer(indexer):
   4104     indexer = [indexer]

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3812,
in Index.get_loc(self, key)
   3807     if isinstance(casted_key, slice) or (
   3808         isinstance(casted_key, abc.Iterable)
   3809         and any(isinstance(x, slice) for x in casted_key)
   3810     ):
   3811         raise InvalidIndexError(key)
-> 3812     raise KeyError(key) from err
   3813 except TypeError:
   3814     # If we have a listlike key, _check_indexing_error will
raise
   3815     #  InvalidIndexError. Otherwise we fall through and re-
```

```
raise
   3816     #  the TypeError.
   3817     self._check_indexing_error(key)

KeyError: 'glocouse'
```

```
diab_df.head(99)
```

```
    glocouse  plas  pres  skin  insu  mass   pedi  age
class  \
0        6.0   148  72.0  35.0     0  33.6  0.627   50
tested_positive
1        1.0    85  66.0  29.0     0  26.6  0.351   31
tested_negative
2        8.0   183  64.0   0.0     0  23.3  0.672   32
tested_positive
3        1.0    89  66.0  23.0    94  28.1  0.167   21
tested_negative
4        0.0   137  40.0  35.0   168  43.1  2.288   33
tested_positive
..       ...   ...   ...   ...   ...   ...    ...  ...                ..
.
94       2.0   142  82.0  18.0    64  24.7  0.761   21
tested_negative
95       6.0   144  72.0  27.0   228  33.9  0.255   40
tested_negative
96       2.0    92  62.0  28.0     0  31.6  0.130   24
tested_negative
97       1.0    71  48.0   NaN    76  20.4  0.323   22
tested_negative
98       6.0    93  50.0  30.0    64  28.7  0.356   23
tested_negative

    glucose_in_mmol
0            0.3330
1            0.0555
2            0.4440
3            0.0555
4            0.0000
..              ...
94           0.1110
95           0.3330
96           0.1110
97           0.0555
98           0.3330

[99 rows x 10 columns]
```

Filter and Groups

```
fil_age_30less=diab_df[diab_df['age']<30]
fil_age_30less.head()
```

```
    glocouse  plas  pres  skin  insu  mass   pedi  age
class  \
3        1.0    89  66.0  23.0    94  28.1  0.167   21
tested_negative
6        3.0    78  50.0  32.0    88  31.0  0.248   26
tested_positive
7       10.0   115   0.0   0.0     0  35.3  0.134   29
tested_negative
20       3.0   126  88.0  41.0   235  39.3  0.704   27
tested_negative
23       9.0   119  80.0  35.0     0  29.0  0.263   29
tested_positive
```

```
    glucose_in_mmol
3          0.055500
6          0.166500
7          0.555001
20         0.166500
23         0.499500
```

```
glucose_below100=diab_df[diab_df['glocouse']<100]
glucose_below100.head()
```

```
   glocouse  plas  pres  skin  insu  mass   pedi  age            class
\
0       6.0   148  72.0  35.0     0  33.6  0.627   50  tested_positive

1       1.0    85  66.0  29.0     0  26.6  0.351   31  tested_negative

2       8.0   183  64.0   0.0     0  23.3  0.672   32  tested_positive

3       1.0    89  66.0  23.0    94  28.1  0.167   21  tested_negative

4       0.0   137  40.0  35.0   168  43.1  2.288   33  tested_positive
```

```
   glucose_in_mmol
0           0.3330
1           0.0555
2           0.4440
3           0.0555
4           0.0000
```

create a filter data set which has only the rows with age between 20 and 30

```
age_bet_20_30=diab_df[(diab_df['age']>20) & (diab_df['age']<30)]
age_bet_20_30.head()
```

```
    preg  preg  pres  skin  insu  mass   pedi  age            class  \
3    1.0    89  66.0  23.0    94  28.1  0.167   21  tested_negative
6    3.0    78  50.0  32.0    88  31.0  0.248   26  tested_positive
7   10.0   115   0.0   0.0     0  35.3  0.134   29  tested_negative
20   3.0   126  88.0  41.0   235  39.3  0.704   27  tested_negative
23   9.0   119  80.0  35.0     0  29.0  0.263   29  tested_positive

    glucose_in_mmol
3          0.055500
6          0.166500
7          0.555001
20         0.166500
23         0.499500
```

Grouping

```
grouped_by_class_age=diab_df.groupby('class')['age'].mean()

grouped_by_class_age
```

```
class
tested_negative    31.238095
tested_positive    40.589744
Name: age, dtype: float64
```

```
group_class_ins=diab_df.groupby('class')['insu'].mean()
group_class_ins
```

```
class
tested_negative     52.571429
tested_positive    114.692308
Name: insu, dtype: float64
```

```
group_class_min=diab_df.groupby('class')['age'].min()
group_class_min
```

```
class
tested_negative    21
tested_positive    25
Name: age, dtype: int64
```

Handling Null

```
diab_df.isnull().sum()
```

```
preg                1
preg                0
pres                1
skin                1
insu                0
```

```
mass               1
pedi               1
age                0
class              0
glucose_in_mmol    1
dtype: int64
```

```
diab_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 102 entries, 0 to 101
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   preg             101 non-null    float64
 1   preg             102 non-null    int64  
 2   pres             101 non-null    float64
 3   skin             101 non-null    float64
 4   insu             102 non-null    int64  
 5   mass             101 non-null    float64
 6   pedi             101 non-null    float64
 7   age              102 non-null    int64  
 8   class            102 non-null    object 
 9   glucose_in_mmol  101 non-null    float64
dtypes: float64(6), int64(3), object(1)
memory usage: 8.1+ KB
```

```
diab_df.dropna()
```

```
     preg  preg  pres  skin  insu  mass   pedi  age  
class  \
0     6.0   148  72.0  35.0     0  33.6  0.627   50  tested_positive

1     1.0    85  66.0  29.0     0  26.6  0.351   31  tested_negative

2     8.0   183  64.0   0.0     0  23.3  0.672   32  tested_positive

3     1.0    89  66.0  23.0    94  28.1  0.167   21  tested_negative

4     0.0   137  40.0  35.0   168  43.1  2.288   33  tested_positive

..    ...   ...   ...   ...   ...   ...    ...  ...                ...

95    6.0   144  72.0  27.0   228  33.9  0.255   40  tested_negative

96    2.0    92  62.0  28.0     0  31.6  0.130   24  tested_negative

98    6.0    93  50.0  30.0    64  28.7  0.356   23  tested_negative

100   1.0   163  72.0   0.0     0  39.0  1.222   33  tested_positive
```

```
101   1.0   151  60.0   0.0     0  26.1  0.179   22  tested_negative
```

```
     glucose_in_mmol
0             0.3330
1             0.0555
2             0.4440
3             0.0555
4             0.0000
..               ...
95            0.3330
96            0.1110
98            0.3330
100           0.0555
101           0.0555

[98 rows x 10 columns]
```

```
diab_df.dropna(inplace=True)
```

```
diab_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 98 entries, 0 to 101
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   preg             98 non-null     float64
 1   preg             98 non-null     int64  
 2   pres             98 non-null     float64
 3   skin             98 non-null     float64
 4   insu             98 non-null     int64  
 5   mass             98 non-null     float64
 6   pedi             98 non-null     float64
 7   age              98 non-null     int64  
 8   class            98 non-null     object 
 9   glucose_in_mmol  98 non-null     float64
dtypes: float64(6), int64(3), object(1)
memory usage: 8.4+ KB
```

```
diab_df.isnull().sum()
```

```
preg              0
preg              0
pres              0
skin              0
insu              0
mass              0
pedi              0
age               0
class             0
```

```
glucose_in_mmol    0
dtype: int64
```

```
diab_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 98 entries, 0 to 101
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   preg             98 non-null     float64
 1   preg             98 non-null     int64  
 2   pres             98 non-null     float64
 3   skin             98 non-null     float64
 4   insu             98 non-null     int64  
 5   mass             98 non-null     float64
 6   pedi             98 non-null     float64
 7   age              98 non-null     int64  
 8   class            98 non-null     object 
 9   glucose_in_mmol  98 non-null     float64
dtypes: float64(6), int64(3), object(1)
memory usage: 8.4+ KB
```

```
diab_df.drop_duplicates()
```

```
     preg  preg  pres  skin  insu  mass   pedi  age  
class  \
0     6.0   148  72.0  35.0     0  33.6  0.627   50  tested_positive

1     1.0    85  66.0  29.0     0  26.6  0.351   31  tested_negative

2     8.0   183  64.0   0.0     0  23.3  0.672   32  tested_positive

3     1.0    89  66.0  23.0    94  28.1  0.167   21  tested_negative

4     0.0   137  40.0  35.0   168  43.1  2.288   33  tested_positive

..    ...   ...   ...   ...   ...   ...    ...  ...                ...

95    6.0   144  72.0  27.0   228  33.9  0.255   40  tested_negative

96    2.0    92  62.0  28.0     0  31.6  0.130   24  tested_negative

98    6.0    93  50.0  30.0    64  28.7  0.356   23  tested_negative

100   1.0   163  72.0   0.0     0  39.0  1.222   33  tested_positive

101   1.0   151  60.0   0.0     0  26.1  0.179   22  tested_negative


     glucose_in_mmol
```

```
0              0.3330
1              0.0555
2              0.4440
3              0.0555
4              0.0000
..                ...
95             0.3330
96             0.1110
98             0.3330
100            0.0555
101            0.0555

[96 rows x 10 columns]
```

```
diab_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 98 entries, 0 to 101
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   preg             98 non-null     float64
 1   preg             98 non-null     int64  
 2   pres             98 non-null     float64
 3   skin             98 non-null     float64
 4   insu             98 non-null     int64  
 5   mass             98 non-null     float64
 6   pedi             98 non-null     float64
 7   age              98 non-null     int64  
 8   class            98 non-null     object 
 9   glucose_in_mmol  98 non-null     float64
dtypes: float64(6), int64(3), object(1)
memory usage: 8.4+ KB
```

```
diab_df.drop_duplicates(inplace=True)
```

```
diab_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 96 entries, 0 to 101
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   preg             96 non-null     float64
 1   preg             96 non-null     int64  
 2   pres             96 non-null     float64
 3   skin             96 non-null     float64
 4   insu             96 non-null     int64  
 5   mass             96 non-null     float64
```

```
 6   pedi              96 non-null     float64
 7   age               96 non-null     int64
 8   class             96 non-null     object
 9   glucose_in_mmol   96 non-null     float64
dtypes: float64(6), int64(3), object(1)
memory usage: 8.2+ KB
```

Reading other formates

```
dia_ex=pd.read_excel(r"C:\Users\User\Documents\python02\
diabetes.xlsx")
dia_ex.head()
```

```
   preg  plas  pres  skin  insu  mass   pedi  age            class
0     6   148    72    35     0  33.6  0.627   50  tested_positive
1     1    85    66    29     0  26.6  0.351   31  tested_negative
2     8   183    64     0     0  23.3  0.672   32  tested_positive
3     1    89    66    23    94  28.1  0.167   21  tested_negative
4     0   137    40    35   168  43.1  2.288   33  tested_positive
```

```
dia_ex=pd.read_excel(r"C:\Users\User\Documents\python02\
diabetes.xlsx",sheet_name="dora")
dia_ex.head()
```

```
  Dead Alive
0  yes    no
1  yes    no
2  yes    no
3  yes    no
4  yes    no
```

```
dia_ex=pd.read_excel(r"C:\Users\User\Documents\python02\
diabetes.xlsx",sheet_name="Hello")
dia_ex.head()
```

```
Empty DataFrame
Columns: [hello, guys, how, are ]
Index: []
```

```
gra_txt=pd.read_csv(r"C:\Users\User\Documents\python02\
grades.txt",sep=' ')
gra_txt.head()
```

```
    Names Initials  SEM1  SEM2  SEM3 Grade
0     Joe        K   9.8  10.0   9.9    A+
1  Rajesh        M   8.9   9.1   9.3     A
2  Kissan        V   9.9   9.3   9.2     A
3    Mary        N   7.7   8.0   7.1     B
4    Jeen        K   9.8   9.1   9.9    A+
```

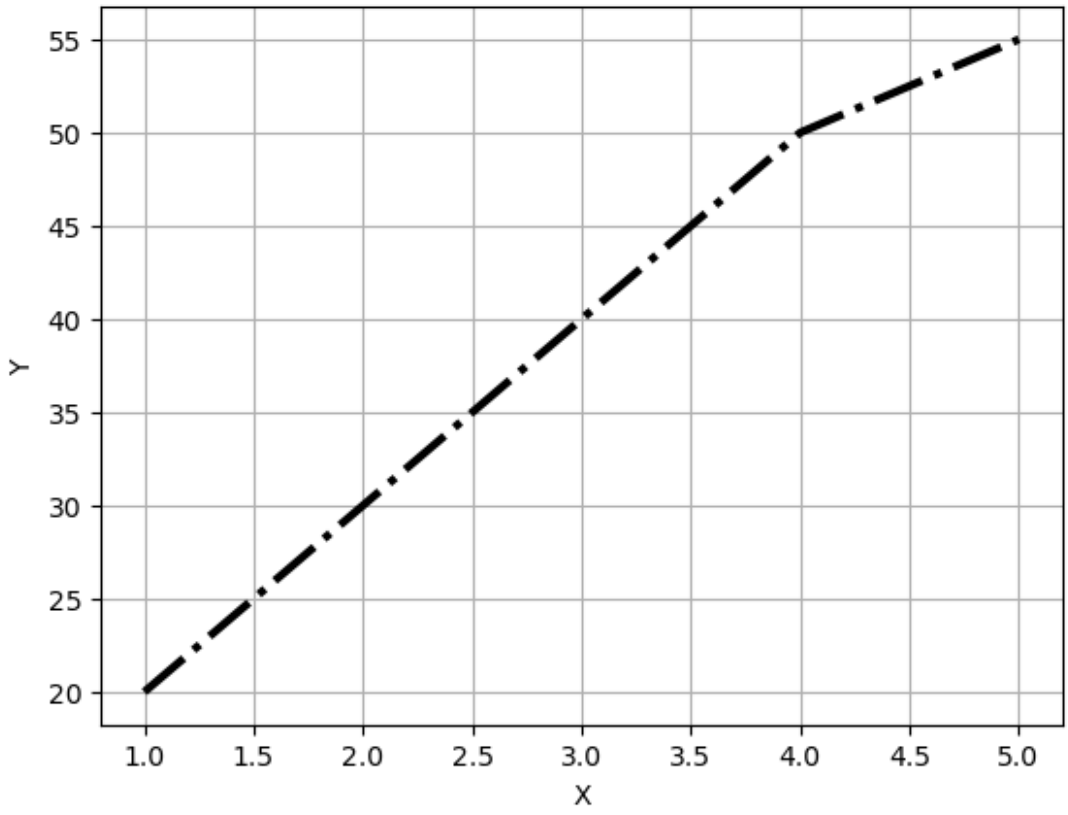
Modifiying datatype

```
gra_txt['SEM1_int']=gra_txt['SEM1'].astype(int)
gra_txt.head()
```

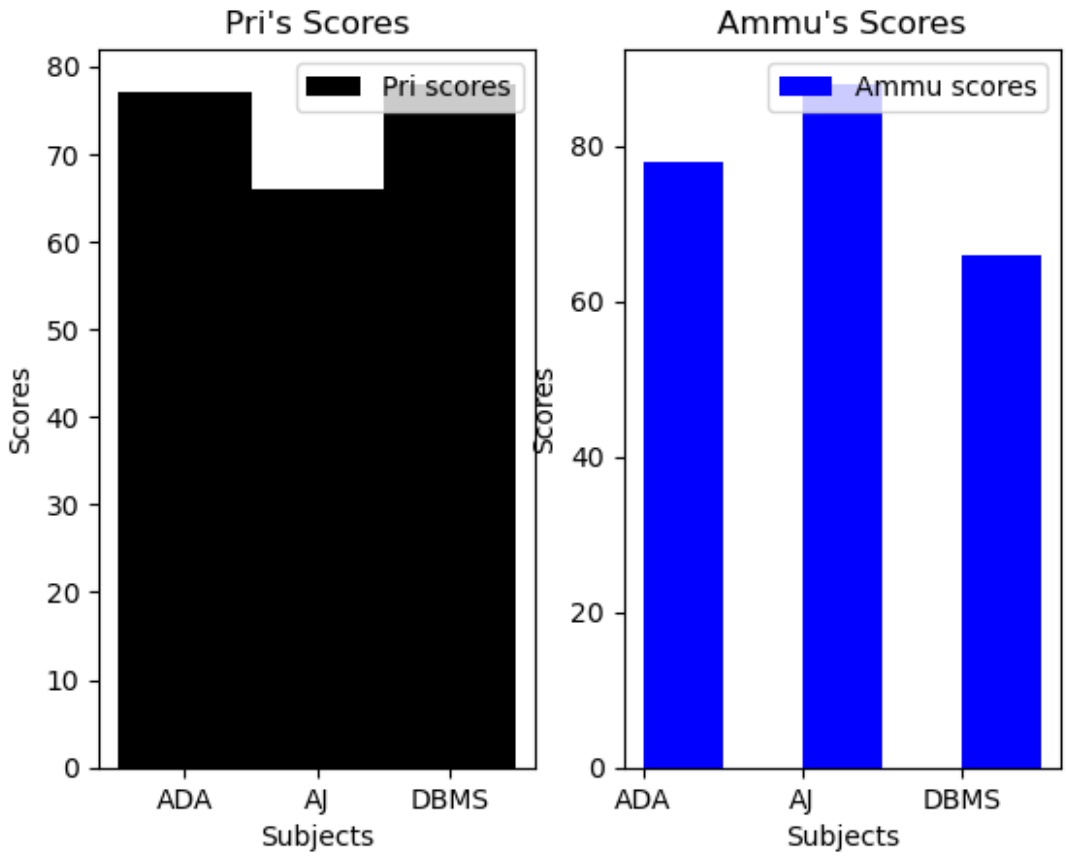
	Names	Initials	SEM1	SEM2	SEM3	Grade	SEM1_int
0	Joe	K	9.8	10.0	9.9	A+	9
1	Rajesh	M	8.9	9.1	9.3	A	8
2	Kissan	V	9.9	9.3	9.2	A	9
3	Mary	N	7.7	8.0	7.1	B	7
4	Jeen	K	9.8	9.1	9.9	A+	9

Matplotlib

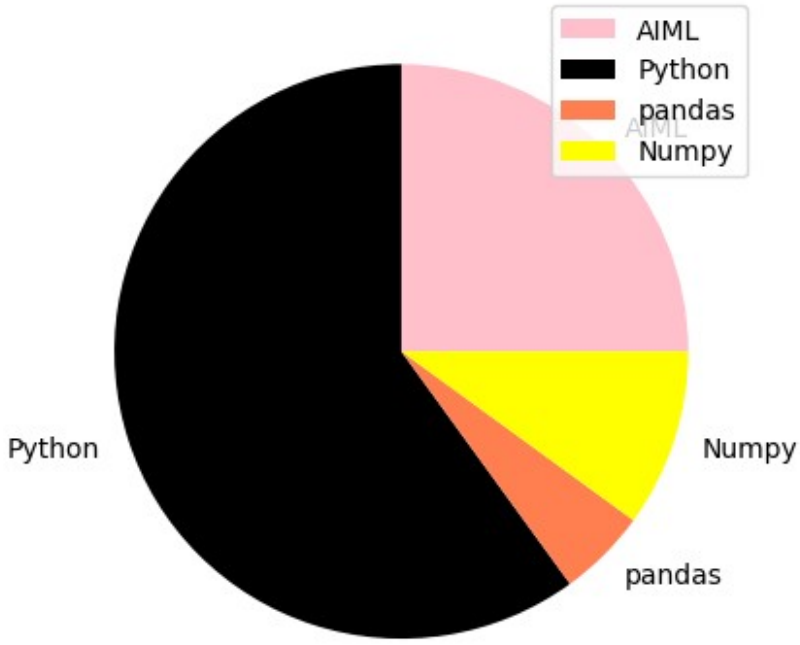
```
x=[1,2,3,4,5]
y=[20,30,40,50,55]

import matplotlib.pyplot as plt

plt.plot(x,y, color='k',label='XY',linestyle="-.",linewidth=3)
plt.xlabel('X')
plt.ylabel('Y')
plt.grid()
plt.show()
```



```
import matplotlib.pyplot as plt

sub = ["ADA", "AJ", "DBMS"]
Pri = [77, 66, 78]
Ammu = [78, 88, 66]
plt.subplot(1, 2, 1)
plt.bar(sub, Pri, color="k", label="Pri
scores",width=1,align="center")
plt.xlabel("Subjects")
plt.ylabel("Scores")
plt.title("Pri's Scores")
plt.legend()

plt.subplot(1, 2, 2)
plt.bar(sub, Ammu, color="b", label="Ammu
scores",width=0.5,align="edge")
plt.xlabel("Subjects")
plt.ylabel("Scores")
plt.title("Ammu's Scores")
plt.legend()

plt.show()
```




```
import numpy as np
a=np.array([25,60,5,10])
labe=["AIML","Python","pandas","Numpy"]
color=['pink','black','coral','yellow']

plt.pie(a,labels=labe,colors=color)
plt.legend()
plt.show()
```


AIML
Python
pandas
Numpy
AIML
Python
Numpy
pandas