EXERCISE - A

A) Pandas DataSeries:

1) Write a Pandas program to create and display a one-dimensional array-like object containing an array of data using Pandas module.

   Program:

   ```
   import pandas as pd
   lis=[10,20,30,40,50,60,70]
   obj = pd.Series(arr)
   print(obj)
   ```

   Output:

   ```
   0    10
   1    20
   2    30
   3    40
   4    50
   5    60
   6    70
   dtype: int32
   ```

2) Write a Pandas program to convert a Panda module Serie to Python list and it's type.

Program:

```
import pandas as pd
obj= pd.Series([10,20,30,40,50,60,70])
print(obj, end='\n\n')
print("conversion from Panda module Series to Python list and it's type")
print(obj.tolist())
print(type(obj.tolist()))
```

Output:

```
0    10
1    20
2    30
3    40
4    50
5    60
6    70
dtype: int64

conversion from Panda module Series to Python list and it's type

[10, 20, 30, 40, 50, 60, 70]
<class 'list'>
```

3) Write a Pandas program to add, subtract, multiple and divide two Pandas Series.

Program:

```
import pandas as pd
obj1 = pd.Series([10, 20, 30, 40, 50])
obj2 = pd.Series([-1,-2,-3,-4,-5])
obj = obj1 + obj2
print("Addition two Series:")
print(obj,end='\n\n')
print("Subtraction two Series:")
obj = obj1 - obj2
print(obj,end='\n\n')
print("Multiply two Series:")
obj = obj1 * obj2
print(obj,end='\n\n')
print("Divide Series1 by Series2:")
obj = obj1 / obj2
print(obj,end='\n\n')
```

Output:

```
Addition two Series:
0     9
1    18
2    27
3    36
4    45
dtype: int64

Subtraction two Series:
0    11
1    22
2    33
3    44
4    55
dtype: int64

Multiply two Series:
0    -10
1    -40
2    -90
3   -160
4   -250
dtype: int64

Divide Series1 by Series2:
0   -10.0
1   -10.0
2   -10.0
3   -10.0
4   -10.0
dtype: float64
```

4) Write a Pandas program to convert a NumPy array to a Pandas series.
Sample Series:
NumPy array:
[10 20 30 40 50]
Converted Pandas series:
0 10
1 20
2 30
3 40
4 50
dtype: int64

Program:

```
import numpy as np
import pandas as pd
arr=np.array([10,20,30,40,50])
print(arr, end='\n\n')
obj=pd.Series(arr)
print("conversion from NumPy array to a Pandas series")
print(obj)
```

Output:

```
[10 20 30 40 50]

conversion from NumPy array to a Pandas series
0    10
1    20
2    30
3    40
4    50
dtype: int32
```

5) Write a Pandas program to convert a dictionary to a Pandas series.

Sample Series:

Original dictionary:

{'a': 100, 'b': 200, 'c': 300, 'd': 400, 'e': 800}

Converted series:

a 100

b 200

c 300

d 400

e 800

dtype: int64

Program:

```
import pandas as pd
dic = {'a': 100, 'b': 200, 'c':300, 'd':400, 'e':800}
print(dic,end='\n\n')
obj = pd.Series(dic)
print("conversion from dictionary to a Pandas series:")
print(obj)
```

Output:

```
{'a': 100, 'b': 200, 'c': 300, 'd': 400, 'e': 800}

conversion from dictionary to a Pandas series:
a    100
b    200
c    300
d    400
e    800
dtype: int64
```

6) Write a Pandas program to convert a given Series to an array

Program:

```
import pandas as pd
import numpy as np
obj = pd.Series([10,20,30,40,50])
print(obj,end='\n\n')
print("conversion from a given Series to an array")
a = np.array(obj.tolist())
print (a)
```

Output:

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

conversion from a given Series to an array
[10 20 30 40 50]
```

7) Write a Pandas program to change the index of a given series.

Program:

```
import pandas as pd
lis=[10,20,30,40,50]
s = pd.Series(lis,index=['A','B','C','D','E'])
print("Original Data Series:")
print(s)
s = s.reindex(index = ['C','D','B','E','A'])
print("Data Series after changing the order of index:")
print(s)
```

Output:

```
Original Data Series:
A    10
B    20
C    30
D    40
E    50
dtype: int64
Data Series after changing the order of index:
C    30
D    40
B    20
E    50
A    10
dtype: int64
```

8) Write a Pandas program to get the items of series A present in series B?
Sample series: [1, 2, 3, 4, 5], [4, 5, 6, 7, 8]

Program:

```
import pandas as pd
s1 = pd.Series([1, 2, 3, 4, 5])
s2 = pd.Series([4, 5, 6, 7, 8])
ser1[ser1.isin(ser2)]
```

Output:

```
3    4
4    5
dtype: int64
```

EXERCISE – B

B) Pandas DataFrames:

Consider Sample Python dictionary data and list labels:
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1]
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

1) Write a Pandas program to create and display a DataFrame from a specified dictionary data which has index labels.

Program:

```
import pandas as pd
import numpy as np
exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James',
'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no','yes', 'yes', 'no',
'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data , index=labels)
print(df)
```

Output:

```
        name  score  attempts qualify
a  Anastasia   12.5         1     yes
b       Dima    9.0         3      no
c  Katherine   16.5         2     yes
d      James    NaN         3      no
e      Emily    9.0         2      no
f    Michael   20.0         3     yes
g    Matthew   14.5         1     yes
h      Laura    NaN         1      no
i      Kevin    8.0         2      no
j      Jonas   19.0         1     yes
```

2) Write a Pandas program to change the name 'James' to 'Suresh' in name column of the
DataFrame.

Program:

```
import pandas as pd
import numpy as np
exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James',
'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no','yes', 'yes', 'no',
'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data , index=labels)
print("Original rows:")
print(df)
print("\nChange the name 'James' to ‘Suresh’:")
df['name'] = df['name'].replace('James', 'Suresh')
print(df)
```

Output:

```
Original rows:
        name  score  attempts qualify
a  Anastasia   12.5         1     yes
b       Dima    9.0         3      no
c  Katherine   16.5         2     yes
d      James    NaN         3      no
e      Emily    9.0         2      no
f    Michael   20.0         3     yes
g    Matthew   14.5         1     yes
h      Laura    NaN         1      no
i      Kevin    8.0         2      no
j      Jonas   19.0         1     yes

Change the name 'James' to ‘Suresh’:
        name  score  attempts qualify
a  Anastasia   12.5         1     yes
b       Dima    9.0         3      no
c  Katherine   16.5         2     yes
d     Suresh    NaN         3      no
e      Emily    9.0         2      no
f    Michael   20.0         3     yes
g    Matthew   14.5         1     yes
h      Laura    NaN         1      no
i      Kevin    8.0         2      no
j      Jonas   19.0         1     yes
```

3) Write a Pandas program to insert a new column in existing DataFrame.

Program:

```
import pandas as pd
import numpy as np
exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James',
'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no','yes', 'yes', 'no',
'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data , index=labels)
print("Original rows:")
print(df)
color =
['Red','Blue','Orange','Red','White','White','Blue','Green','Gree
n','Red']
df['color'] = color
print("\nNew DataFrame after inserting the 'color' column")
print(df)
```

Output:

```
Original rows:
        name  score  attempts qualify
a  Anastasia   12.5         1     yes
b       Dima    9.0         3      no
c  Katherine   16.5         2     yes
d      James    NaN         3      no
e      Emily    9.0         2      no
f    Michael   20.0         3     yes
g    Matthew   14.5         1     yes
h      Laura    NaN         1      no
i      Kevin    8.0         2      no
j      Jonas   19.0         1     yes

New DataFrame after inserting the 'color' column
        name  score  attempts qualify   color
a  Anastasia   12.5         1     yes     Red
b       Dima    9.0         3      no    Blue
c  Katherine   16.5         2     yes  Orange
d      James    NaN         3      no     Red
e      Emily    9.0         2      no   White
f    Michael   20.0         3     yes   White
g    Matthew   14.5         1     yes    Blue
h      Laura    NaN         1      no   Green
i      Kevin    8.0         2      no   Green
j      Jonas   19.0         1     yes     Red
```

4) Write a Pandas program to get list from DataFrame column headers.

Program:

```
import pandas as pd
import numpy as np
exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James',
'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no','yes', 'yes', 'no',
'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data , index=labels)
print(list(df.columns.values))
```

Output:

```
['name', 'score', 'attempts', 'qualify']
```

5) Write a Pandas program to select the 'name' and 'score' columns from the following DataFrame.

Program:

```
import pandas as pd
import numpy as np
exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 
'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no','yes', 'yes', 'no', 
'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data , index=labels)
print("Select specific columns:")
print(df[['name', 'score']])
```

Output:

```
Select specific columns:
        name  score
a  Anastasia   12.5
b       Dima    9.0
c  Katherine   16.5
d      James    NaN
e      Emily    9.0
f    Michael   20.0
g    Matthew   14.5
h      Laura    NaN
i      Kevin    8.0
j      Jonas   19.0
```

6) Write a Pandas program to count the number of rows and columns of a Data Frame.

Program:

```
import pandas as pd
exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James',
'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no','yes', 'yes', 'no',
'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data , index=labels)
rows=len(df.axes[0])
cols=len(df.axes[1])
print("Number of Rows: "+ str(rows))
print("Number of Columns: "+ str(cols))
```

Output:

```
Number of Rows: 10
Number of Columns: 4
```

7) Write a Pandas program to select the rows where the score is missing, i.e. is NaN.

Program:

```
import pandas as pd
exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James',
'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no','yes', 'yes', 'no',
'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data , index=labels)
print("Rows where score is missing:")
print(df[df['score'].isnull()])
```

Output:

```
Rows where score is missing:
    name  score  attempts qualify
d  James    NaN         3      no
h  Laura    NaN         1      no
```

8) Write a Pandas program to select the rows where number of attempts in the examination is less than 2 and score greater than 15.

Program:

```
import pandas as pd
import numpy as np
exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James',
'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no','yes', 'yes', 'no',
'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data , index=labels)
print("Number of attempts in the examination is less than 2 and
score greater than 15 :")
print(df[(df['attempts'] < 2) & (df['score'] > 15)])
```

Output:

```
Number of attempts in the examination is less than 2 and score gr
eater than 15 :
    name  score  attempts qualify
j  Jonas   19.0         1     yes
```

EXERCISE – C

C) Pandas Index:

Empid	First name	Last name	Phone	Age	Salary
107	Albert	Watson	1234567890	25	25000
129	Blake	Cooper	1224466790	27	36000
99	Rose	Stephens	1112223336	36	46700
115	Charles	Ward	9490666777	42	78200
119	Alex	Sanders	9490749465	33	35600
113	Louis	Wood	2378911233	43	46700
176	Rory	Kelly	7467133267	44	456700
166	Mia	West	9996677662	47	785360

1. Write a Pandas program to display the default index and set a column as an Index in a given dataframe.

Program:

```
import pandas as pd
df = pd.DataFrame({
'Empid':[107,129,99,115,119,113,176,166],
'First name':['Albert','Blake','Rose','Charles','Alex',
'Louis', 'Rory','Mia'],
'Last name':['Watson','Cooper','Stephens','Ward', 'Sanders',
'Wood','Kelly','West'],
'Phone':[1234567890,1224466790,1112223336,9490666777,
9490749465,2378911233,7467133267,9996677662],
'Age':[25,27,36,42,33,43,44,47],
'Salary':[25000,36000,46700,78200,35600,46700, 456700,
785360]})
print("Default index: \n",df)
df1 = df.set_index('Empid')
print("\nsetting a column as an Index: \n",df1)
```

Output:

```
Default index:
    Empid First name Last name       Phone  Age  Salary
0    107     Albert    Watson  1234567890   25   25000
1    129      Blake    Cooper  1224466790   27   36000
2     99       Rose  Stephens  1112223336   36   46700
3    115    Charles      Ward  9490666777   42   78200
4    119       Alex   Sanders  9490749465   33   35600
5    113      Louis      Wood  2378911233   43   46700
6    176       Rory     Kelly  7467133267   44  456700
7    166        Mia      West  9996677662   47  785360

setting a column as an Index:
       First name Last name       Phone  Age  Salary
Empid
107       Albert    Watson  1234567890   25   25000
129        Blake    Cooper  1224466790   27   36000
99          Rose  Stephens  1112223336   36   46700
115      Charles      Ward  9490666777   42   78200
119         Alex   Sanders  9490749465   33   35600
113        Louis      Wood  2378911233   43   46700
176         Rory     Kelly  7467133267   44  456700
166          Mia      West  9996677662   47  785360
```

2. Write a Pandas program to create a multi Index frame using two columns

Program:

```
import pandas as pd
df = pd.DataFrame({
'Empid':[107,129,99,115,119,113,176,166],
'First name':['Albert','Blake','Rose','Charles','Alex',
'Louis', 'Rory','Mia'],
'Last name':['Watson','Cooper','Stephens','Ward', 'Sanders',
'Wood','Kelly','West'],
'Phone':[1234567890,1224466790,1112223336,9490666777,
9490749465,2378911233,7467133267,9996677662],
'Age':[25,27,36,42,33,43,44,47],
'Salary':[25000,36000,46700,78200,35600,46700, 456700,
785360]})
print("Default index: \n",df)
df1=df.set_index(['Empid','Age'])
print("\n creating multi Index frame using two columns\n",df1)
```

Output:

```
Default index:
    Empid First name Last name       Phone  Age  Salary
0    107     Albert    Watson  1234567890   25   25000
1    129      Blake    Cooper  1224466790   27   36000
2     99       Rose  Stephens  1112223336   36   46700
3    115    Charles      Ward  9490666777   42   78200
4    119       Alex   Sanders  9490749465   33   35600
5    113      Louis      Wood  2378911233   43   46700
6    176       Rory     Kelly  7467133267   44  456700
7    166        Mia      West  9996677662   47  785360

 creating multi Index frame using two columns
           First name Last name       Phone  Salary
Empid Age
107   25      Albert    Watson  1234567890   25000
129   27       Blake    Cooper  1224466790   36000
99    36        Rose  Stephens  1112223336   46700
115   42     Charles      Ward  9490666777   78200
119   33        Alex   Sanders  9490749465   35600
113   43       Louis      Wood  2378911233   46700
176   44        Rory     Kelly  7467133267  456700
166   47         Mia      West  9996677662  785360
```

3. Write a Pandas program to display the default index and set a column as an Index in a given dataframe and then reset the index.

Program:

```
import pandas as pd
df = pd.DataFrame({
'Empid':[107,129,99,115,119,113,176,166],
'First name':['Albert','Blake','Rose','Charles','Alex',
'Louis', 'Rory','Mia'],
'Last name':['Watson','Cooper','Stephens','Ward', 'Sanders',
'Wood','Kelly','West'],
'Phone':[1234567890,1224466790,1112223336,9490666777,
9490749465,2378911233,7467133267,9996677662],
'Age':[25,27,36,42,33,43,44,47],
'Salary':[25000,36000,46700,78200,35600,46700, 456700,
785360]})
print("Default index: \n",df)
df1 = df.set_index('Age')
print("\nsetting a column Age as an Index: \n",df1)
df2 = df1.reset_index()
print("\n reset the index \n",df2)
```

Output:

```
Default index:
    Empid First name Last name       Phone  Age  Salary
0    107     Albert    Watson  1234567890   25   25000
1    129      Blake    Cooper  1224466790   27   36000
2     99       Rose  Stephens  1112223336   36   46700
3    115    Charles      Ward  9490666777   42   78200
4    119       Alex   Sanders  9490749465   33   35600
5    113      Louis      Wood  2378911233   43   46700
6    176       Rory     Kelly  7467133267   44  456700
7    166        Mia      West  9996677662   47  785360

 creating age as new index
      Empid First name Last name       Phone  Salary
Age
25     107     Albert    Watson  1234567890   25000
27     129      Blake    Cooper  1224466790   36000
36      99       Rose  Stephens  1112223336   46700
42     115    Charles      Ward  9490666777   78200
33     119       Alex   Sanders  9490749465   35600
43     113      Louis      Wood  2378911233   46700
44     176       Rory     Kelly  7467133267  456700
47     166        Mia      West  9996677662  785360

 reset the index
    Age  Empid First name Last name       Phone  Salary
0   25    107     Albert    Watson  1234567890   25000
```

```
1   27    129      Blake    Cooper  1224466790   36000
2   36     99       Rose  Stephens  1112223336   46700
3   42    115    Charles      Ward  9490666777   78200
4   33    119       Alex   Sanders  9490749465   35600
5   43    113      Louis      Wood  2378911233   46700
6   44    176       Rory     Kelly  7467133267  456700
7   47    166        Mia      West  9996677662  785360
```

4. Write a Pandas program to convert index of a given dataframe into a column.

Program:

```
import pandas as pd
df = pd.DataFrame({
'Empid':[107,129,99,115,119,113,176,166],
'First name':['Albert','Blake','Rose','Charles','Alex',
'Louis', 'Rory','Mia'],
'Last name':['Watson','Cooper','Stephens','Ward', 'Sanders',
'Wood','Kelly','West'],
'Phone':[1234567890,1224466790,1112223336,9490666777,
9490749465,2378911233,7467133267,9996677662],
'Age':[25,27,36,42,33,43,44,47],
'Salary':[25000,36000,46700,78200,35600,46700, 456700,
785360]})
print("Default index: \n",df)
df.reset_index(level=0,inplace=True)
print("\n reset the index \n",df)
```

Output:

```
Default index:
    Empid First name Last name       Phone  Age  Salary
0    107     Albert    Watson  1234567890   25   25000
1    129      Blake    Cooper  1224466790   27   36000
2     99       Rose  Stephens  1112223336   36   46700
3    115    Charles      Ward  9490666777   42   78200
4    119       Alex   Sanders  9490749465   33   35600
5    113      Louis      Wood  2378911233   43   46700
6    176       Rory     Kelly  7467133267   44  456700
7    166        Mia      West  9996677662   47  785360

 reset the index
    index  Empid First name Last name       Phone  Age  Salary
0      0    107     Albert    Watson  1234567890   25   25000
1      1    129      Blake    Cooper  1224466790   27   36000
2      2     99       Rose  Stephens  1112223336   36   46700
3      3    115    Charles      Ward  9490666777   42   78200
4      4    119       Alex   Sanders  9490749465   33   35600
5      5    113      Louis      Wood  2378911233   43   46700
6      6    176       Rory     Kelly  7467133267   44  456700
7      7    166        Mia      West  9996677662   47  785360
```

5. Write a Pandas program to create an index labels by using 64-bit integers, using floating-point numbers in a given dataframe.

Program:

```
import pandas as pd
df = pd.DataFrame({
'Empid':[107,129,99,115,119,113,176,166],
'First name':['Albert','Blake','Rose','Charles','Alex',
'Louis', 'Rory','Mia'],
'Last name':['Watson','Cooper','Stephens','Ward', 'Sanders',
'Wood','Kelly','West'],
'Phone':[1234567890,1224466790,1112223336,9490666777,
9490749465,2378911233,7467133267,9996677662],
'Age':[25,27,36,42,33,43,44,47],
'Salary':[25000,36000,46700,78200,35600,46700, 456700,
785360]}, index = [1,2,3,4,5,6,7,8])
print("Default index: \n",df)
print("\nViewing the Index:\n",df.index)
df=pd.DataFrame(index=[.1,.2,.3,.4,.5,.6,.7,.8])
print("\nFloating-point Index:\n",df.index)
```

Output:

```
 Default index:
     Empid First name Last name       Phone  Age  Salary
 1    107     Albert    Watson  1234567890   25   25000
 2    129      Blake    Cooper  1224466790   27   36000
 3     99       Rose  Stephens  1112223336   36   46700
 4    115    Charles      Ward  9490666777   42   78200
 5    119       Alex   Sanders  9490749465   33   35600
 6    113      Louis      Wood  2378911233   43   46700
 7    176       Rory     Kelly  7467133267   44  456700
 8    166        Mia      West  9996677662   47  785360

 Viewing the Index:
  Int64Index([1, 2, 3, 4, 5, 6, 7, 8], dtype='int64')

 Floating-point Index:
  Float64Index([0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8], dtype=
 'float64')
```

EXERCISE – D

D) Pandas String and Regular Expressions:

1. Write a Pandas program to convert all the string values to upper, lower cases in a given pandas series. Also find the length of the string values.

   program:

```
import pandas as pd
s = pd.Series(['lower', 'CAPITALS', 'this is a sentence',
'SwApCaSe'])
print("converting the pandas series to uppercase:")
print(s.str.upper())
print("converting the pandas series to lowercase:")
print(s.str.lower())
print("length of the strings in pandas series:")
print(s.str.len())
```

   Output:

```
converting the pandas series to uppercase:
0                 LOWER
1              CAPITALS
2    THIS IS A SENTENCE
3              SWAPCASE
dtype: object
converting the pandas series to lowercase:
0                 lower
1              capitals
2    this is a sentence
3              swapcase
dtype: object
length of the strings in pandas series:
0     5
1     8
2    18
3     8
dtype: int64
```

2. Write a Pandas program to remove whitespaces, left sided whitespaces and right sided whitespaces of the string values of a given pandas series.

program:

```
import pandas as pd
s1 = pd.Series(['   python','   pandas   ','numpy       '])
print("removing whitespaces")
print(s1.str.strip())
print("removing left sided whitespaces")
print(s1.str.lstrip())
print("removing right sided whitespaces")
print(s1.str.rstrip())
```

output:

```
removing whitespaces
0    python
1    pandas
2     numpy
dtype: object
removing left sided whitespaces
0          python
1       pandas
2    numpy
dtype: object
removing right sided whitespaces
0       python
1       pandas
2        numpy
dtype: object
```

3. Write a Pandas program to count of occurrence of a specified substring in a DataFrame column.

program:

```
import pandas as pd
df =pd.DataFrame({'Empid':[107,129,99,115,119,113,176,166],
'First name':['Albert','Blake','Rose','Charles','Alex',
'Louis', 'Rory','Mia'],
'Last name':['Watson','Cooper','Stephens','Ward', 'Sanders',
'Wood','Kelly','West'],
'Phone':[1234567890,1224466790,1112223336,9490666777,
9490749465,2378911233,7467133267,9996677662],
'Age':[25,27,36,42,33,43,44,47],
'Salary':[25000,36000,46700,78200,35600,46700, 456700,
785360]})
df['count']=df['First name'].str.count('a')
print("occurrence of a specified substring 'a' in a column
first name is")
print(df[['First name','count']])
```

output:

```
occurrence of a specified substring 'a' in a column first name
is
  First name  count
0     Albert      0
1      Blake      1
2       Rose      0
3    Charles      1
4       Alex      0
5      Louis      0
6       Rory      0
7        Mia      1
```

4. Write a Pandas program to swap the cases of a specified character column in a given DataFrame.

   program:

```
import pandas as pd
df =pd.DataFrame({'Empid':[107,129,99,115,119,113,176,166],
'First name':['Albert','Blake','Rose','Charles','Alex',
'Louis', 'Rory','Mia'],
'Last name':['Watson','Cooper','Stephens','Ward', 'Sanders',
'Wood','Kelly','West'],
'Phone':[1234567890,1224466790,1112223336,9490666777,
9490749465,2378911233,7467133267,9996677662],
'Age':[25,27,36,42,33,43,44,47],
'Salary':[25000,36000,46700,78200,35600,46700, 456700,
785360]})
df['swap']=df['First name'].str[:2]+df['First
name'].str[2].str.swapcase()+df['First name'].str[3:]
print("swapping the cases of 3rd character in column First
name:")
print(df[['First name','swap']])
```

   output:

```
swapping the cases of 3rd character in column First name:
  First name     swap
0     Albert   AlBert
1      Blake    BlAke
2       Rose     RoSe
3    Charles  ChArles
4       Alex     AlEx
5      Louis    LoUis
6       Rory     RoRy
7        Mia      MiA
```

EXERCISE – E

E) Pandas Joining and merging DataFrame:

1. Write a Pandas program to join the two given dataframes along rows and assign all data.

Program:

```
import pandas as pd
df1 = pd.DataFrame({"student_id":['s1','s2','s3','s4','s5'],
'name':['Danniella Fenton','Ryder Storey','Bryce Jensen','Ed
Bernal','Kwame Morin'],
'marks':[200,210,190,222,199]})
df2 = pd.DataFrame({"student_id":['s4','s5','s6','s7','s8'],
'name':['Scarlette Fisher','Carla Williamson','Dante
Morse','Kaiser William','Madeeha Preston'],
'marks':[201,200,198,219,201]})
print("DataFrame 1 is \n",df1)
print("\nDataFrame 2 is \n",df2)
print("\nDataFrame after joining two dataframes along rows")
print(df1.append(df2,ignore_index=True))
```

output:

```
DataFrame 1 is
student_id              name  marks
0         s1  Danniella Fenton    200
1         s2      Ryder Storey    210
2         s3      Bryce Jensen    190
3         s4         Ed Bernal    222
4         s5       Kwame Morin    199

DataFrame 2 is
student_id              name  marks
0         s4  Scarlette Fisher    201
1         s5  Carla Williamson    200
2         s6       Dante Morse    198
3         s7    Kaiser William    219
4         s8   Madeeha Preston    201

DataFrame after joining two dataframes along rows
student_id              name  marks
0         s1  Danniella Fenton    200
1         s2      Ryder Storey    210
2         s3      Bryce Jensen    190
3         s4         Ed Bernal    222
4         s5       Kwame Morin    199
5         s4  Scarlette Fisher    201
6         s5  Carla Williamson    200
7         s6       Dante Morse    198
8         s7    Kaiser William    219
9         s8   Madeeha Preston    201
```

2. Write a Pandas program to append a list of dictionaries or series to a existing DataFrame and display the combined data.

Program:

```
import pandas as pd
df1 = pd.DataFrame({'student_id':['s1','s2','s3','s4','s5'],
'name':['Danniella Fenton','Ryder Storey','Bryce Jensen','Ed
Bernal','Kwame Morin'],
'marks':[200,210,190,222,199]})
li = [{'student_id':'s6','name':'Scarlette
Fisher','marks':205},
{'student_id':'s7','name':'Carla Williamson','marks':200}]
print("data frame is:\n",df1)
print("\nList of Dictionary is",li[0],li[1],sep='\n')
print("\nthe resultant dataframe is:")
print(df1.append(li,ignore_index=True))
```

output:

```
data frame is:
student_id              name  marks
0         s1  Danniella Fenton    200
1         s2      Ryder Storey    210
2         s3      Bryce Jensen    190
3         s4         Ed Bernal    222
4         s5       Kwame Morin    199

List of Dictionary is
{'student_id': 's6', 'name': 'Scarlette Fisher', 'marks': 205}
{'student_id': 's7', 'name': 'Carla Williamson', 'marks': 200}

the resultant dataframe is:
student_id              name  marks
0         s1  Danniella Fenton    200
1         s2      Ryder Storey    210
2         s3      Bryce Jensen    190
3         s4         Ed Bernal    222
4         s5       Kwame Morin    199
5         s6  Scarlette Fisher    205
6         s7  Carla Williamson    200
```

3. Write a Pandas program to join the two dataframes with matching records from both sides where available.

Program:

```
import pandas as pd
df1 = pd.DataFrame({"student_id":['s1','s2','s3','s4','s5'],
'name':['Danniella Fenton','Ryder Storey','Bryce Jensen','Ed
Bernal','Kwame Morin'],
'marks':[200,210,190,222,199]})
df2 = pd.DataFrame({"student_id":['s4','s5','s6','s7','s8'],
'name':['Scarlette Fisher','Carla Williamson','Dante
Morse','Kaiser William','Madeeha Preston'],
'marks':[201,200,198,219,201]})
print("DataFrame 1 is \n",df1)
print("\nDataFrame 2 is \n",df2)
print("\nDataFrame after joining two dataframes with matching
records from both side are:")
print(df1.merge(df2, how='outer', on='student_id'))
```

output:

```
DataFrame 1 is
student_id              name  marks
0         s1  Danniella Fenton    200
1         s2      Ryder Storey    210
2         s3      Bryce Jensen    190
3         s4         Ed Bernal    222
4         s5       Kwame Morin    199

DataFrame 2 is
student_id              name  marks
0         s4  Scarlette Fisher    201
1         s5  Carla Williamson    200
2         s6       Dante Morse    198
3         s7    Kaiser William    219
4         s8   Madeeha Preston    201

DataFrame after joining two dataframes with matching records
from both side are:
student_id            name_x  marks_x            name_y  marks_y
0         s1  Danniella Fenton    200.0               NaN      NaN
1         s2      Ryder Storey    210.0               NaN      NaN
2         s3      Bryce Jensen    190.0               NaN      NaN
3         s4         Ed Bernal    222.0  Scarlette Fisher    201.0
4         s5       Kwame Morin    199.0  Carla Williamson    200.0
5         s6               NaN      NaN       Dante Morse    198.0
6         s7               NaN      NaN    Kaiser William    219.0
7         s8               NaN      NaN   Madeeha Preston    201.0
```

4. write a pandas as program to join(left join) the two dataframes using keys from left dataframe only

   program:

```
df1 = pd.DataFrame({'key1':['K0','K0','K1','K2'],
'key2':['K0','K1','K0','K1'],
'P':['P0','P1','P2','P3'],
'Q':['Q0','Q1','Q2','Q3']})
df2 = pd.DataFrame({'key1':['K0','K1','K1','K2'],
'key2':['K0','K0','K0','K0'],
'R':['R0','R1','R2','R3'],
'S':['S0','S1','S2','S3']})
print("DataFrame 1 is \n",df1)
print("\nDataFrame 2 is \n",df2)
print("\nDataFrame after joining two dataframes using keys
from left dataframe:")
print(df1.merge(df2,how='left',on=['key1','key2']))
```

   output:

```
DataFrame 1 is
key1 key2   P   Q
0   K0   K0  P0  Q0
1   K0   K1  P1  Q1
2   K1   K0  P2  Q2
3   K2   K1  P3  Q3

DataFrame 2 is
key1 key2   R   S
0   K0   K0  R0  S0
1   K1   K0  R1  S1
2   K1   K0  R2  S2
3   K2   K0  R3  S3

DataFrame after joining two dataframes using keys from left
dataframe:
key1 key2   P   Q    R    S
0   K0   K0  P0  Q0   R0   S0
1   K0   K1  P1  Q1  NaN  NaN
2   K1   K0  P2  Q2   R1   S1
3   K1   K0  P2  Q2   R2   S2
4   K2   K1  P3  Q3  NaN  NaN
```

EXERCISE – F

F) Pandas Time Series:

1. write a pandas program to create

--------------------------(write this code in one program) --------------------------

```
# a) Datetime object for Jan 15 2012.
from datetime import datetime as dt a = dt(2012,1,15)
print("Datetime object for Jan 15 2012:",a)
o/p: Datetime object for Jan 15 2012: 2012-01-15 00:00:00
```

```
# b) Specific date and time of 9:20 pm.
 b = dt(2012,1,15,21,20)
print("Specific date and time of 9:20 pm:",b)
o/p: Specific date and time of 9:20 pm: 2012-01-15 21:20:00
```

```
# c) Local date and time.
 c = dt.now()
print("Local date and time:",c)
o/p: Local date and time: 2022-05-05 05:49:55.109430
```

```
# d) A date without time.
d = dt(2003,2,12).date()
print("A date without time:",d)
o/p: A date without time: 2003-02-12
```

```
# e) Current date.
e = dt.now().date()
print("Current date:",e)
o/p: Current date: 2022-05-05
```

```
# f) Time from a datetime.
f = dt(2003,2,12,5,30).time()
print("Time from a datetime:",f)
o/p: Time from a datetime: 05:30:00
```

```
# g) Current local time.
 g = dt.now().time()
print("Current local time:",g)
o/p: Current local time: 05:41:32.674587
```

2. Write a Pandas program to create a date from a given year, month, day and another date from a given string formats.

   program:

```
from datetime import datetime as dt
year = int(input("Enter year: "))
month = int(input("Enter month: "))
day = int(input("Enter day: "))
a = dt(year,month,day)
print("date from given year,month,day:",a)
string = input("Enter year,month,day in YYYY-MM-DD format:")
b = dt.fromisoformat(string)
print("date from a given string:",b)
```

   Output:

```
Enter year: 2003
Enter month: 02
Enter day: 12
date from given year,month,day: 2003-02-12 00:00:00 Enter
year,month,day in YYYY-MM-DD format: 2002-05-03 date from a
given string: 2002-05-03 00:00:00
```

3. Write a Pandas program to create a time-series with two index labels and random values. Also print the type of the index.

program:

```
from datetime import datetime as dt
import pandas as pd
import numpy as np
labels = [dt(2003,2,12), dt(2002,5,3)]
a =pd.Series(np.random.rand(2),labels)
print("Time-series with two index labels:")
print(a)
print(type(a.index)
```

Output:

```
Time-series with two index labels:
2003-02-12 0.891855
2002-05-03 0.620129
dtype: float64
<class 'pandas.core.indexes.datetimes.DatetimeIndex'>
```

EXERCISE – G

G) Pandas Grouping Aggregate:

Consider dataset:

	school	class	name	date_Of_Birth	age	height	weight	address
S1	s001	V	Alberto Franco	15/05/2002	12	173	35	street1
S2	s002	V	Gino Mcneill	17/05/2002	12	192	32	street2
S3	s003	VI	Ryan Parkes	16/02/1999	13	186	33	street3
S4	s001	VI	Eesha Hinton	25/09/1998	13	167	30	street1
S5	s002	V	Gino Mcneill	11/05/2002	14	151	31	street2
S6	s004	VI	David Parkes	15/09/1997	12	159	32	street4

1. Write a Pandas program to split the following dataframe into groups based on school code. Also check the type of GroupBy object.

   program:

```
import pandas as pd
data = {'school':['s001','s002','s003','s001','s002','s004'],
'class':['V','V','VI','VI','V','VI'],
'name':['Alberto Franco','Gino Mcneill','Ryan Parkes ','Eesha
Hinton','Gino Mcneill','David Parkes'],
'date_Of_Birth':['15/05/2002','17/05/2002','16/02/1999','25/09
/1998','11/05/2002','15/09/1997'],
'age':[12,12,13,13,14,12],
'height':[173,192,186,167,151,159],
'weight':[35,32,33,30,31,32],
'address':['street1','street2','street3','street1','street2','
street4']}
label = ['S1','S2','S3','S4','S5','S6']
df = pd.DataFrame(data,index=label)
#print(df,"\n")
print("split the dataframe into groups based on school
code:\n")
a = df.groupby(['school'])
for i in a:
    print(i,"\n\n")
print("type of GroupBy object:")
print(type(a))
```

output:

```
split the dataframe into groups based on school code:

('s001',school class            name date_Of_Birth  age  height  weight  address
    S1    s001     V  Alberto Franco    15/05/2002   12     173      35  street1
    S4   s001     VI    Eesha Hinton    25/09/1998   13     167      30  street1)


('s002',school class          name date_Of_Birth  age  height  weight  address
    S2    s002     V  Gino Mcneill    17/05/2002   12     192      32  street2
    S5    s002     V  Gino Mcneill    11/05/2002   14     151      31  street2)


('s003',school class         name date_Of_Birth  age  height  weight  address
    S3    s003    VI  Ryan Parkes    16/02/1999   13     186      33  street3)


('s004',school class          name date_Of_Birth  age  height  weight  address
    S6    s004    VI  David Parkes    15/09/1997   12     159      32  street4)


type of GroupBy object:
<class 'pandas.core.groupby.generic.DataFrameGroupBy'>
```

2. Write a Pandas program to split the following dataframe by school code and get mean, min, and max value of age for each school.

program:

```
import pandas as pd
data = {'school':['s001','s002','s003','s001','s002','s004'],
'class':['V','V','VI','VI','V','VI'],
'name':['Alberto Franco','Gino Mcneill','Ryan Parkes ','Eesha
Hinton','Gino Mcneill','David Parkes'],
'date_Of_Birth':['15/05/2002','17/05/2002','16/02/1999','25/09
/1998','11/05/2002','15/09/1997'],
'age':[12,12,13,13,14,12],
'height':[173,192,186,167,151,159],
'weight':[35,32,33,30,31,32],
'address':['street1','street2','street3','street1','street2','
street4']}
label = ['S1','S2','S3','S4','S5','S6']
df = pd.DataFrame(data,index=label)
print("spliting the dataframe by school code and get mean,
min, and max value of age for each school")
print(df.groupby('school')[['age']].agg(['mean','min','max']))
```

output:

```
spliting the dataframe by school code and get mean, min, and
max value of age for each school

             age
        mean min max
school
s001    12.5  12  13
s002    13.0  12  14
s003    13.0  13  13
s004    12.0  12  12
```

EXERCISE – H

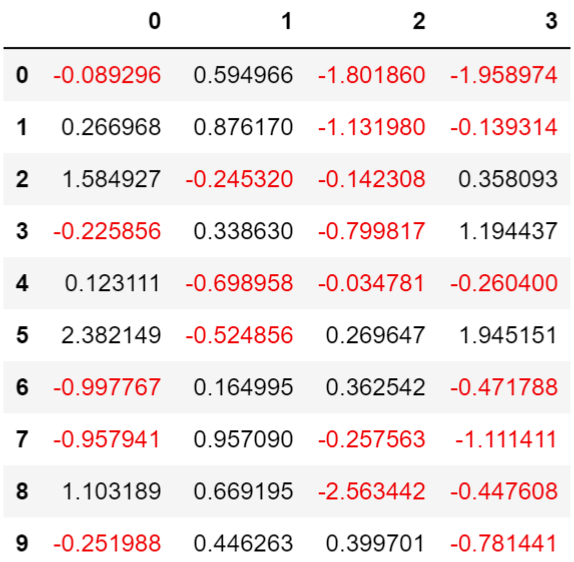
H) Pandas Styling:

1. Create a dataframe of ten rows, four columns with random values. Write a Pandas program to highlight the negative numbers red and positive numbers black.

Program:

```
import pandas as pd
import numpy as np
df = pd.DataFrame(np.random.randn(10,4))
def neg_and_pos(s):
    return ['color : red' if i < 0 else 'color : black' for i
in s]
df.style.apply(neg_and_pos)
```

Output:



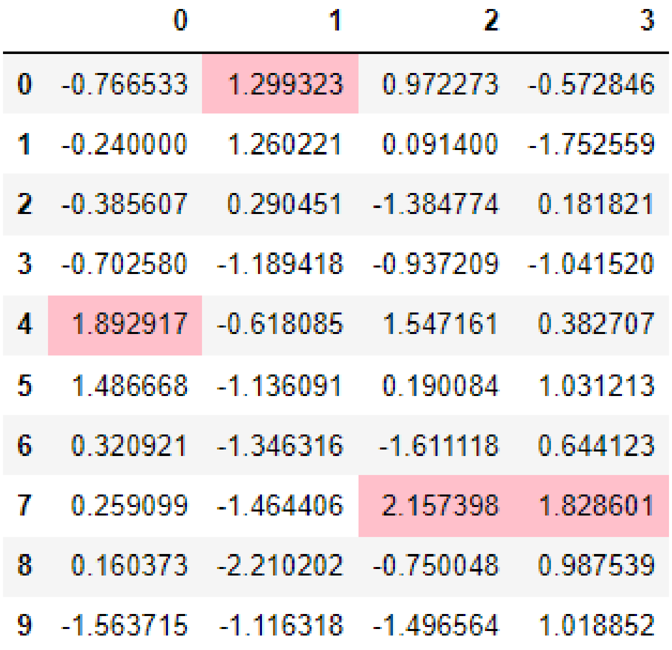
	0	1	2	3
0	-0.089296	0.594966	-1.801860	-1.958974
1	0.266968	0.876170	-1.131980	-0.139314
2	1.584927	-0.245320	-0.142308	0.358093
3	-0.225856	0.338630	-0.799817	1.194437
4	0.123111	-0.698958	-0.034781	-0.260400
5	2.382149	-0.524856	0.269647	1.945151
6	-0.997767	0.164995	0.362542	-0.471788
7	-0.957941	0.957090	-0.257563	-1.111411
8	1.103189	0.669195	-2.563442	-0.447608
9	-0.251988	0.446263	0.399701	-0.781441

2. Create a dataframe of ten rows, four columns with random values. Write a Pandas program to highlight the maximum value in each column.

Program:

```
import pandas as pd
import numpy as np
df = pd.DataFrame(np.random.randn(10,4))
df.style.highlight_max(color='pink')
```

Output:



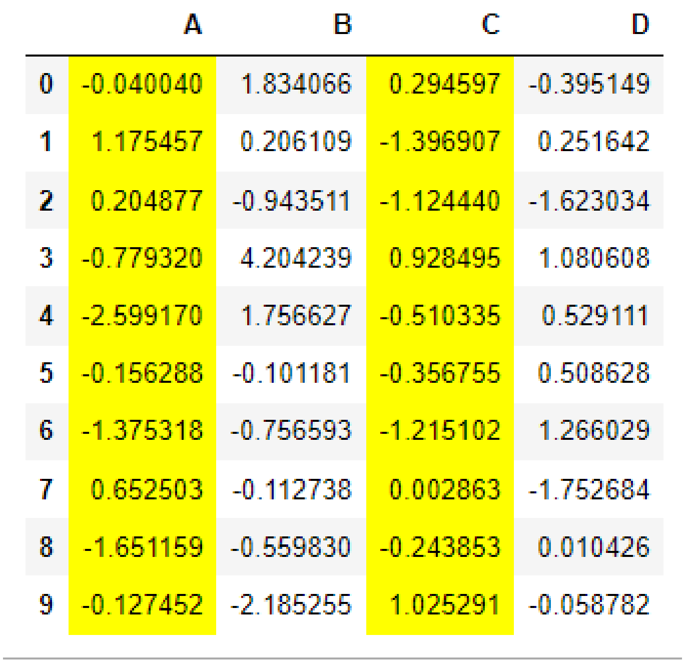
	0	1	2	3
0	-0.766533	1.299323	0.972273	-0.572846
1	-0.240000	1.260221	0.091400	-1.752559
2	-0.385607	0.290451	-1.384774	0.181821
3	-0.702580	-1.189418	-0.937209	-1.041520
4	1.892917	-0.618085	1.547161	0.382707
5	1.486668	-1.136091	0.190084	1.031213
6	0.320921	-1.346316	-1.611118	0.644123
7	0.259099	-1.464406	2.157398	1.828601
8	0.160373	-2.210202	-0.750048	0.987539
9	-1.563715	-1.116318	-1.496564	1.018852

3. Create a dataframe of ten rows, four columns with random values. Write a Pandas program to highlight dataframe's specific columns

program:

```
import pandas as pd
import numpy as np
df =
pd.DataFrame(np.random.randn(10,4),columns=['A','B','C','D'])
def col(s):
    return 'background-color : yellow’
df.style.applymap(col,subset=['A','C'])
```

Output:



	A	B	C	D
0	-0.040040	1.834066	0.294597	-0.395149
1	1.175457	0.206109	-1.396907	0.251642
2	0.204877	-0.943511	-1.124440	-1.623034
3	-0.779320	4.204239	0.928495	1.080608
4	-2.599170	1.756627	-0.510335	0.529111
5	-0.156288	-0.101181	-0.356755	0.508628
6	-1.375318	-0.756593	-1.215102	1.266029
7	0.652503	-0.112738	0.002863	-1.752684
8	-1.651159	-0.559830	-0.243853	0.010426
9	-0.127452	-2.185255	1.025291	-0.058782

Exercise – I

I) Excel:

1. write a pandas program to import excel data into a pandas dataframe

   program:

```
import pandas as pd
df = pd.read_excel(r"C:\ Sample - Superstore.xls")
df
```

   Output:

```
      Row ID        Order ID Order Date  Ship Date       Ship Mode Customer ID
0          1  CA-2016-152156 2016-11-08 2016-11-11    Second Class    CG-12520
1          2  CA-2016-152156 2016-11-08 2016-11-11    Second Class    CG-12520
2          3  CA-2016-138688 2016-06-12 2016-06-16    Second Class    DV-13045
3          4  US-2015-108966 2015-10-11 2015-10-18  Standard Class    SO-20335
4          5  US-2015-108966 2015-10-11 2015-10-18  Standard Class    SO-20335
```

2. Write a Pandas program to find the sum, mean, max, min value of a column of file.

program:

```
import pandas as pd
df = pd.read_excel(r"C:\Sample - Superstore.xls")
print("\nthe total sales:",df['Sales'].sum())
print("\nthe means of  sales:",df['Sales'].mean())
print("\nthe maximum sales:",df['Sales'].max())
print("\nthe minimum sales:",df['Sales'].min())
```

Output:

```
the total sales: 2297200.8603000003
the means of  sales: 229.8580008304938
the maximum sales: 22638.48
the minimum sales: 0.44399999999999995
```

3. write a pandas program to read columns that contains the address of the customer from a given excel file

program:

```
import pandas as pd
import pandas as pd
df = pd.read_excel(r"C:\Sample - Superstore.xls",usecols=['Customer Name','City','State','Postal Code','Region','Country'])
print(df)
```

Output:

```
       Customer Name        Country             City       State
0        Claire Gute  United States        Henderson    Kentucky
1        Claire Gute  United States        Henderson    Kentucky
2    Darrin Van Huff  United States      Los Angeles  California
3     Sean O'Donnell  United States  Fort Lauderdale     Florida
4     Sean O'Donnell  United States  Fort Lauderdale     Florida
```

4. write a Pandas program to import given excel data into a Pandas dataframe and find a list of orders between two specific month and year

program:

```
import pandas as pd
df = pd.read_excel(r"C:\Sample - Superstore.xls")
df[df['Order Date'].between('2016-02-12','2016-05-03')]
```

Output:

```
      Row ID        Order ID Order Date  Ship Date       Ship Mode Customer ID
44        45  CA-2016-118255 2016-03-11 2016-03-13     First Class    ON-18715
45        46  CA-2016-118255 2016-03-11 2016-03-13     First Class    ON-18715
88        89  CA-2016-159695 2016-04-05 2016-04-10    Second Class    GM-14455
151      152  CA-2016-158834 2016-03-13 2016-03-16     First Class    TW-21025
152      153  CA-2016-158834 2016-03-13 2016-03-16     First Class    TW-21025
```

5. Write a Pandas program to import excel data into a Pandas dataframe and identify most sold product

   program:

```
import pandas as pd
df = pd.read_excel(r"C:\Sample - Superstore.xls")
a = df.groupby('Product Name')['Quantity'].sum()
print(a.idxmax(),a.max())
```

   Output:

```
Staples 215
```

Exercise – J

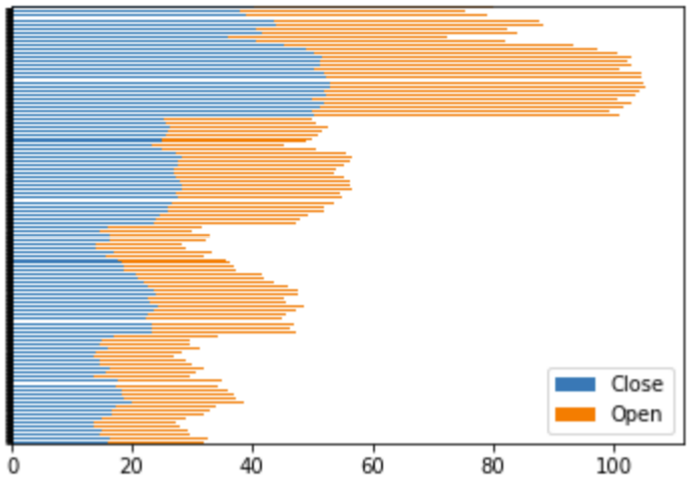
J) Plotting:

1. Write a Pandas program to create a horizontal stacked bar plot of opening, closing stock prices of any stock dataset between two specific dates.

   Program:

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel(r"C:\Stock Portfolio.xlsx")
df2 = df[df['Date'].between('03-05-2002','13-09-2002')]
df2[['Close','Open']].plot.barh(stacked=True)
```

   Output:

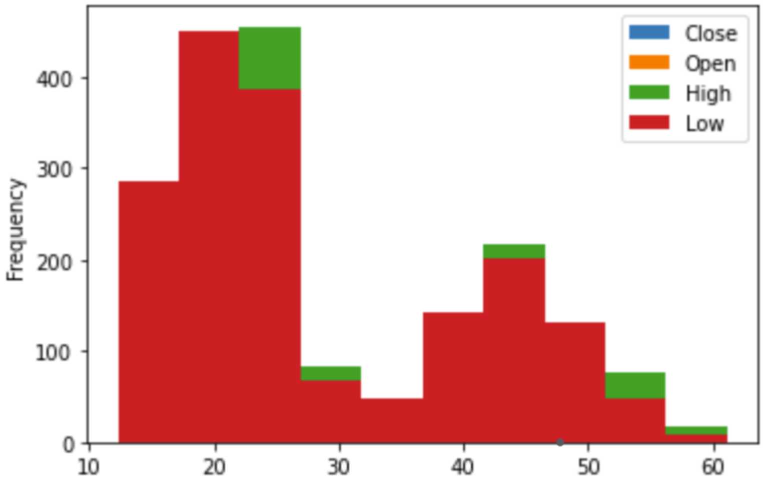


2. Write a Pandas program to create a histograms plot of opening, closing, high, low stock prices of stock dataset between two specific dates.

Program:

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel(r"C:\Stock Portfolio.xlsx")
df2 = df[df['Date'].between('03-05-2002','12-02-2003')]
df2[['Close','Open','High','Low']].plot.hist()
```

Output:

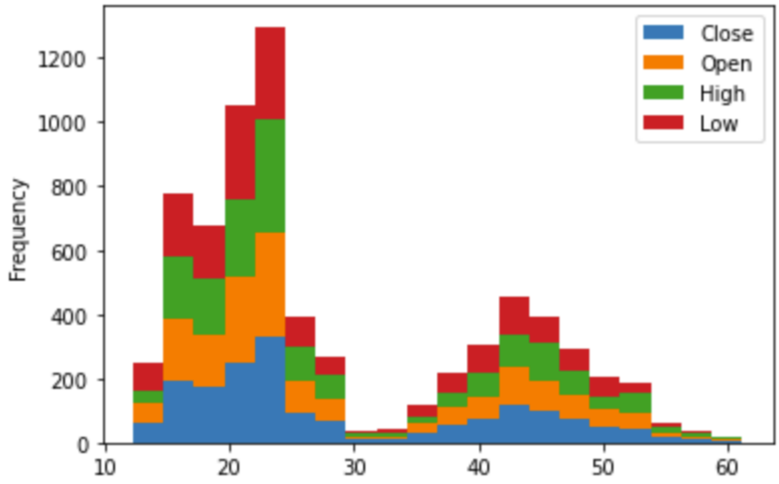


3. Write a Pandas program to create a stacked histograms plot of opening, closing, high, low stock prices of stock dataset between two specific dates with more bins.

Program:

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel(r"C:\Stock Portfolio.xlsx")
df2 = df[df['Date'].between('03-05-2002','12-02-2003')]
df2[['Close','Open','High','Low']].plot.hist(stacked=True,bins=20)
```

Output:



Exercise – K

K) Pandas SQL Query:

1. Write a Pandas program to display all the records of a student file.

Program:

```
import pandas as pd
df = pd.read_csv(r"D:\downloads\Students-data.csv ")
print(df)
```

Output:

```
    gender NationalITy PlaceofBirth       StageID GradeID  VisITedResource
s
0        M          KW       KuwaIT    lowerlevel    G-04                16
1        M          KW       KuwaIT    lowerlevel    G-04                20
2        M          KW       KuwaIT    lowerlevel    G-04                 7
3        M          KW       KuwaIT    lowerlevel    G-04                25
4        M          KW       KuwaIT    lowerlevel    G-04                50
```

2. Write a Pandas program to select distinct department id from employees file.

Program:

```
import pandas as pd
df = pd.read_excel(r"D:\downloads\employee.xlsx")
print("all the unique department numbers in employee list")
print(df['DEPTNO'].unique())
```

Output:

```
all the unique department numbers in employee list
[40 20 30 12 23 24 34]
```