Q1. List any five functions of the pandas library with execution.

In [3]:

```
import pandas as pd
a = pd.Series([1,2,3])
print(a)

b = pd.DataFrame({'a':[1,2,3],'b':[4,5,6]})
print(b)

c  = pd.read_csv("https://raw.githubusercontent.com/datasciencedojo/datasets/master/tita
print(c)

print(c.head(1))

print(c.tail(1))
```

```
0    1
1    2
2    3
dtype: int64
   a  b
0  1  4
1  2  5
2  3  6
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           ...       ...     ...
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3

                                                 Name     Sex   Age  SibS
p  \
0                             Braund, Mr. Owen Harris    male  22.0
1
1    Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0
1
2                              Heikkinen, Miss. Laina  female  26.0
0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0
1
4                            Allen, Mr. William Henry    male  35.0
0
..                                                ...     ...   ...
...
886                            Montvila, Rev. Juozas    male  27.0
0
887                      Graham, Miss. Margaret Edith  female  19.0
0
888          Johnston, Miss. Catherine Helen "Carrie"  female   NaN
1
889                              Behr, Mr. Karl Howell    male  26.0
0
890                                Dooley, Mr. Patrick    male  32.0
0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     ...               ...      ...   ...      ...
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
   PassengerId  Survived  Pclass                      Name   Sex   Age  Sib
```

```
Sp  \
0            1         0       3  Braund, Mr. Owen Harris  male  22.0
1

   Parch     Ticket  Fare Cabin Embarked
0      0  A/5 21171  7.25   NaN        S
     PassengerId  Survived  Pclass                 Name   Sex   Age  SibSp
\
890          891         0       3  Dooley, Mr. Patrick  male  32.0      0

     Parch  Ticket  Fare Cabin Embarked
890      0  370376  7.75   NaN        Q
```

Q2. Given a Pandas DataFrame df with columns 'A', 'B', and 'C', write a Python function to re-index the DataFrame with a new index that starts from 1 and increments by 2 for each row.

In [9]:

```
import pandas as pd

def reindex_dataframe(df):

    new_index = range(1, 2*len(df)+1, 2)
    df = df.set_index(pd.Index(new_index))

    return df


df = pd.DataFrame({'A': [1, 2, 3, 4], 'B': [5, 6, 7, 8], 'C': [9, 10, 11, 12]})

new_df = reindex_dataframe(df)

print(new_df)
```

```
   A  B   C
1  1  5   9
3  2  6  10
5  3  7  11
7  4  8  12
```

Q3. You have a Pandas DataFrame df with a column named 'Values'. Write a Python function that iterates over the DataFrame and calculates the sum of the first three values in the 'Values' column. The function should print the sum to the console.

In [10]:

```
c['Age'][0:3].sum()
```

Out[10]:

```
86.0
```

Q4. Given a Pandas DataFrame df with a column 'Text', write a Python function to create a new column 'Word_Count' that contains the number of words in each row of the 'Text' column.

In [14]:

```
b = [{"text" : "There are many variations of passages of Lorem Ipsum available, but the
k = pd.DataFrame(b)
k

k['Word_Count'] = k['text'].apply(lambda x : len(x.split()))
k
```

Out[14]:

	text	Word_Count
0	There are many variations of passages of Lorem...	121

Q5. How are DataFrame.size() and DataFrame.shape() different?

DataFrame.shape():- The shape of a DataFrame is a tuple of array dimensions that tells the number of rows and columns of a given DataFrame.

DataFrame.size():- The size property will return the size of a pandas DataFrame, which is the exact number of data cells in your DataFrame

Q6. Which function of pandas do we use to read an excel file?

df1 = pd.read_excel()

Q7. You have a Pandas DataFrame df that contains a column named 'Email' that contains email addresses in the format 'username@domain.com'. Write a Python function that creates a new column 'Username' in df that contains only the username part of each email address.

The username is the part of the email address that appears before the '@' symbol. For example, if the email address is 'john.doe@example.com', the 'Username' column should contain 'john.doe'. Your function should extract the username from each email address and store it in the new 'Username' column.

In [17]:

```
df = pd.DataFrame({'email':['vishal123@gmail.com','guautam456@gmail.com','varun895@gmail
df['Username'] = df['email'].apply(lambda email: email.split('@')[0])
df
```

Out[17]:

	email	Username
0	vishal123@gmail.com	vishal123
1	guautam456@gmail.com	guautam456
2	varun895@gmail.com	varun895

Q8. You have a Pandas DataFrame df with columns 'A', 'B', and 'C'. Write a Python function that selects all rows where the value in column 'A' is greater than 5 and the value in column 'B' is less than 10. The function should return a new DataFrame that contains only the selected rows.

In [44]:

```
data = {'A' : [3,8,6,2,9],
      'B' : [5,2,9,3,1],
      'C' : [1,7,4,5,2]}
df2 = pd.DataFrame(data)
df2
df2[(df2['A'] > 5)&(df2['B'] < 10)]
```

Out[44]:

	A	B	C
1	8	2	7
2	6	9	4
4	9	1	2

Q9. Given a Pandas DataFrame df with a column 'Values', write a Python function to calculate the mean, median, and standard deviation of the values in the 'Values' column.

In [48]:

```
c  = pd.read_csv("https://raw.githubusercontent.com/datasciencedojo/datasets/master/tita
c
```

Out[48]:

```
49.693428597180905
```

In [49]:

```
c['Fare'].mean()
```

Out[49]:

```
32.2042079685746
```

In [50]:

```
c['Fare'].median()
```

Out[50]:

```
14.4542
```

In [51]:

```
c['Fare'].std()
```

Out[51]:

```
49.693428597180905
```

Q10. Given a Pandas DataFrame df with a column 'Sales' and a column 'Date', write a Python function to create a new column 'MovingAverage' that contains the moving average of the sales for the past 7 days for each row in the DataFrame. The moving average should be calculated using a window of size 7 and should include the current day.

In [52]:

```
data = pd.DataFrame({"Sales": [500,100,300,200,400,1000,1500,700,900,600,800],
       "Date":["01-01-2023","02-01-2023","03-01-2023","04-01-2023","05-01-2023","06-01-2
              ,"11-01-2023"]})

data['Moving_average'] = data['Sales'].rolling(window = 7,min_periods = 1).mean()
data
```

Out[52]:

	Sales	Date	Moving_average
0	500	01-01-2023	500.000000
1	100	02-01-2023	300.000000
2	300	03-01-2023	300.000000
3	200	04-01-2023	275.000000
4	400	05-01-2023	300.000000
5	1000	06-01-2023	416.666667
6	1500	07-01-2023	571.428571
7	700	08-01-2023	600.000000
8	900	09-01-2023	714.285714
9	600	10-01-2023	757.142857
10	800	11-01-2023	842.857143

Q11. You have a Pandas DataFrame df with a column 'Date'. Write a Python function that creates a new column 'Weekday' in the DataFrame. The 'Weekday' column should contain the weekday name (e.g. Monday, Tuesday) corresponding to each date in the 'Date' column. For example, if df contains the following values:

Date 0 2023-01-01 1 2023-01-02 2 2023-01-03 3 2023-01-04 4 2023-01-05

Your function should create the following DataFrame:

Date Weekday 0 2023-01-01 Sunday 1 2023-01-02 Monday 2 2023-01-03 Tuesday 3 2023-01-04 Wednesday 4 2023-01-05 Thursday

The function should return the modified DataFrame.

In [53]:

```
data = pd.DataFrame({"Date":["2023-01-01","2023-01-02","2023-01-03","2023-01-04","2023-0
data['Date'] = pd.to_datetime(data['Date'])
data['Weekdays'] = data['Date'].dt.strftime('%A')
data
```

Out[53]:

	Date	Weekdays
0	2023-01-01	Sunday
1	2023-01-02	Monday
2	2023-01-03	Tuesday
3	2023-01-04	Wednesday
4	2023-01-05	Thursday

Q12. Given a Pandas DataFrame df with a column 'Date' that contains timestamps, write a Python function to select all rows where the date is between '2023-01-01' and '2023-01-31'.

In [55]:

```
df = pd.DataFrame()
df["Date"] = pd.date_range(start= "2023-01-01" , end = "2023-02-28")
df[(df['Date'] >= '2023-01-01') & (df['Date'] <= '2023-01-31')]
```

Out[55]:

	Date
0	2023-01-01
1	2023-01-02
2	2023-01-03
3	2023-01-04
4	2023-01-05
5	2023-01-06
6	2023-01-07
7	2023-01-08
8	2023-01-09
9	2023-01-10
10	2023-01-11
11	2023-01-12
12	2023-01-13
13	2023-01-14
14	2023-01-15
15	2023-01-16
16	2023-01-17
17	2023-01-18
18	2023-01-19
19	2023-01-20
20	2023-01-21
21	2023-01-22
22	2023-01-23
23	2023-01-24
24	2023-01-25
25	2023-01-26
26	2023-01-27
27	2023-01-28
28	2023-01-29
29	2023-01-30
30	2023-01-31

Q13. To use the basic functions of pandas, what is the first and foremost necessary library that needs to be imported?

In [57]:

```
import pandas as pd
```

In [ ]:

```

```