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

In [1]:
```
import pandas as pd
```

In [2]:
```
df=pd.read_csv("Services.csv")
```

In [3]:
```
df.head()
#shows first 5 row data of dataframe
```

Out[3]:

	id	location_id	program_id	accepted_payments	alternate_name	application_process	audience	description	eligibility	email	...	interpreta
0	1	1	NaN	NaN	NaN	Walk in or apply by phone.	Older adults age 55 or over, ethnic minorities...	A walk-in center for older adults that provide...	Age 55 or over for most programs, age 60 or ov...	NaN	...	
1	2	2	NaN	NaN	NaN	Apply by phone for an appointment.	Residents of San Mateo County age 55 or over	Provides training and job placement to eligibl...	Age 55 or over, county resident and willing an...	NaN	...	
2	3	3	NaN	NaN	NaN	Phone for information (403-4300 Ext. 4322).	Older adults age 55 or over who can benefit fr...	Offers supportive counseling services to San M...	Resident of San Mateo County age 55 or over	NaN	...	
3	4	4	NaN	NaN	NaN	Apply by phone.	Parents, children, families with problems of c...	Provides supervised visitation services and a ...	None	NaN	...	
4	5	5	NaN	NaN	NaN	Phone for information.	Low-income working families with children tran...	Provides fixed 8% short term loans to eligible...	Eligibility: Low-income family with legal cust...	NaN	...	

5 rows × 22 columns

In [4]:
```
df.tail()
#Last rows
```

Out[4]:

	id	location_id	program_id	accepted_payments	alternate_name	application_process	audience	description	eligibility	
18	19	19	NaN	NaN	NaN	Call for screening appointment (650-347-3648).	NaN	Provides free medical and dental care to those...	Low-income person without access to health care	
19	20	20	NaN	NaN	NaN	Walk in.	NaN	no unrequired fields for this service	NaN	
20	21	21	NaN	NaN	NaN	By phone during business hours.	NaN	just a test service	NaN	
21	22	22	NaN	Cash, Check, Credit Card	Fotos para pasaportes	Walk in or apply by phone or mail	Profit and nonprofit businesses, the public, m...	[NOTE THIS IS NOT A REAL SERVICE--THIS IS FOR ...	None	passports@ex
22	23	22	NaN	NaN	NaN	Walk in or apply by phone or mail	Second service and nonprofit businesses, the p...	[NOTE THIS IS NOT A REAL ORGANIZATION--THIS IS...	None	

5 rows × 22 columns

In [5]:
```
#describes the stats but only describes whose data type is int or float
df.describe()
```

Out[5]:

	id	location_id	program_id
count	23.00000	23.000000	0.0
mean	12.00000	11.956522	NaN
std	6.78233	6.711444	NaN
min	1.00000	1.000000	NaN
25%	6.50000	6.500000	NaN
50%	12.00000	12.000000	NaN
75%	17.50000	17.500000	NaN
max	23.00000	22.000000	NaN

In [6]:
```
df.columns
#shows the index of column
```

Out[6]:
```
Index(['id', 'location_id', 'program_id', 'accepted_payments',
       'alternate_name', 'application_process', 'audience', 'description',
       'eligibility', 'email', 'fees', 'funding_sources',
       'interpretation_services', 'keywords', 'languages', 'name',
       'required_documents', 'service_areas', 'status', 'wait_time', 'website',
       'taxonomy_ids'],
      dtype='object')
```

In [18]:
```
df[df.dtypes[df.dtypes== object].index]
#shows the dataframes of only object type data
```

			visits...		...	access to health care		
18	NaN	NaN	Call for screening appointment (650-347-3648).	NaN	Provides free medical and dental care to those...	Low-income person without access to health care	NaN	None
19	NaN	NaN	Walk in.	NaN	no unrequired fields for this service	NaN	NaN	NaN
20	NaN	NaN	By phone during business hours.	NaN	just a test service	NaN	NaN	NaN
21	Cash, Check, Credit Card	Fotos para pasaportes	Walk in or apply by phone or mail	Profit and nonprofit businesses, the public, m...	[NOTE THIS IS NOT A REAL SERVICE--THIS IS FOR ...	None	passports@example.org	None, excep for permits and photocopying Cas..
				Second				None, excep

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 [58]:
```
Mylist=[['Ampere','Borde','Chartered'],['Amputate','Born','Compensate'],['Aviation','Bookmark','Clone'],['Aftermath
```

In [59]:
```
Mylist
```

Out[59]:
```
[['Ampere', 'Borde', 'Chartered'],
 ['Amputate', 'Born', 'Compensate'],
 ['Aviation', 'Bookmark', 'Clone'],
 ['Aftermath', 'Benchmark', 'Copyright']]
```

In [60]:
```
df=pd.DataFrame(Mylist,columns=['A','B','C'])
```

In [61]:
```
df
```

Out[61]:

	A	B	C
0	Ampere	Borde	Chartered
1	Amputate	Born	Compensate
2	Aviation	Bookmark	Clone
3	Aftermath	Benchmark	Copyright

In [85]:
```
inc_list = []
for i in range(0,9):
    if i%2!=0:
        inc_list.append(i)
```

In [86]:
```
inc_list
```

Out[86]:
```
[1, 3, 5, 7]
```

In [103]:
```
da = df['A']
db = df['B']
dc = df['C']
```

In [104]:
```
da,db,dc
```

Out[104]:
```
(0       Ampere
 1     Amputate
 2     Aviation
 3    Aftermath
 Name: A, dtype: object,
 0        Borde
 1         Born
 2     Bookmark
 3    Benchmark
 Name: B, dtype: object,
 0     Chartered
 1    Compensate
 2         Clone
 3     Copyright
 Name: C, dtype: object)
```

In [108]:
```
d2=pd.Series(list(da),index=inc_list)
d3=pd.Series(list(db),index=inc_list)
d4=pd.Series(list(dc),index=inc_list)
```

In [109]:
```
d2
```

Out[109]:
```
1       Ampere
3     Amputate
5     Aviation
7    Aftermath
dtype: object
```

In [110]:
```
d3
```

Out[110]:
```
1        Borde
3         Born
5     Bookmark
7    Benchmark
dtype: object
```

In [111]:
```
d4
```

Out[111]:
```
1     Chartered
3    Compensate
5         Clone
7     Copyright
dtype: object
```

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 [4]:
```
import pandas as pd
```

In [5]:
```
df=pd.DataFrame({'Values':[12,15,17,19,15]})
```

In [79]:
```
d2=df['Values']
```

In [85]:
```
def values(list):
    l2=sum(d2[0:3])
    print(l2)
```

In [86]:
```
values(d2)
```

```
44
```

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 [89]:
```
df = pd.DataFrame({'Text':['Amitoj','Amitoj Makes new music','Gaming is my new style but I make gaming music too']}
#creating the dataframe
```

In [90]:
```
df
```

Out[90]:

	Text
0	Amitoj
1	Amitoj Makes new music
2	Gaming is my new style but I make gaming music...

In [91]:
```
df['Word_Count']=df['Text'].apply(lambda x:len(x.split()))
#function for counting words
```

In [92]:
```
df
```

Out[92]:

	Text	Word_Count
0	Amitoj	1
1	Amitoj Makes new music	4
2	Gaming is my new style but I make gaming music...	11

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

DataFrame.size() : It shows the the multiplication of number of rows and columns present in the dataframe DataFrame.size() : It shows both no. of rows as well as number of columns present in the dataframe

In [99]:
```
df1 = {
     'Name':['Amitoj','Andrea','micheal','moris','Ravi','Xien','Jalpa','Zilpe'],
   'Mathematics_score':[62,47,55,74,32,77,86,55]}
```

In [100]:
```
df=pd.DataFrame(df1)
```

In [101]:
```
df
```

Out[101]:

	Name	Mathematics_score
0	Amitoj	62
1	Andrea	47
2	micheal	55
3	moris	74
4	Ravi	32
5	Xien	77
6	Jalpa	86
7	Zilpe	55

In [102]:
```
df.shape
```

Out[102]:
```
(8, 2)
```

In [104]:
```
df.size
```

Out[104]:
```
16
```

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

In [105]:
```
#the function we use for reading excelfile is pd.read_csv("<Name of the file>.csv")
pd.read_csv('Services.csv')
```

							problems of c...	and a ...	
4	5	5	NaN	NaN	NaN	Phone for information.	Low-income working families with children tran...	Provides fixed 8% short term loans to eligible...	Eligibility: Low-income family with legal cust...
5	6	6	NaN	NaN	NaN	Walk in or apply by phone for membership appli...	Any age	A multipurpose center offering a wide variety ...	None
6	7	7	NaN	NaN	NaN	Apply by phone or be referred by a doctor, soc...	Older adults who have memory or sensory loss, ...	Rosener House is a day center for older adults...	Age 18 or over
7	8	8	NaN	NaN	NaN	Apply by phone.	Senior citizens age 60 or over, disabled indiv...	Delivers a hot meal to the home of persons age...	Homebound person unable to cook or shop

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.

In [66]:
```
us_1 = pd.DataFrame({'Email':['johndoe@gmail.com','amit@hotmal.com','djhype@tocmail.com']})
```

In [67]:
```
us_1
```

Out[67]:

	Email
0	johndoe@gmail.com
1	amit@hotmal.com
2	djhype@tocmail.com

In [68]:
```
def sep(x):
    return x.split('@')[0]
```

In [69]:
```
us_1['Username']=us_1['Email'].apply(sep)
```

In [70]:
```
us_1
```

Out[70]:

	Email	Username
0	johndoe@gmail.com	johndoe
1	amit@hotmal.com	amit
2	djhype@tocmail.com	djhype

In [ ]:

In [ ]:

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 [31]:
```
import pandas as pd
df = pd.DataFrame({'A':[3,8,6,2,9],'B':[5,2,9,3,1],'C':[1,7,4,5,2]})
```

In [32]:
```
df
```

Out[32]:

	A	B	C
0	3	5	1
1	8	2	7
2	6	9	4
3	2	3	5
4	9	1	2

In [33]:
```
def filt_num(df):
    selected_rows = df[(df['A']>5) & (df['B']<10)]
    return selected_rows
```

In [34]:
```
filt_num(df)
```

Out[34]:

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

In [ ]:

In [ ]:

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 [106]:
```
df=pd.DataFrame({'Values':[1,4,6,5,4,2,4,6,7,9,45,65]})
```

In [107]:
```
df
```

Out[107]:

	Values
0	1
1	4
2	6
3	5
4	4
5	2
6	4
7	6
8	7
9	9
10	45
11	65

In [117]:
```
def stats_of(df):
    print("Median: ",df.median())
    print("Mean : ",df.mean())
    print("Std Deviation: ",df.std())
```

In [118]:
```
stats_of(df)
```

```
Median:  Values    5.5
dtype: float64
Mean :  Values    13.166667
dtype: float64
Std Deviation:  Values    20.112562
dtype: float64
```

In [121]:
```
dm=pd.DataFrame({'Values':[12,44,66,54,64,42,4,6,7,9,45,65]})
dm
```

Out[121]:

	Values
0	12
1	44
2	66
3	54
4	64
5	42
6	4
7	6
8	7
9	9
10	45
11	65

In [122]:
```
stats_of(dm)
```

```
Median:  Values    43.0
dtype: float64
Mean :  Values    34.833333
dtype: float64
Std Deviation:  Values    25.376558
dtype: float64
```

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 [43]:
```
date=pd.date_range(start='2023-08-23',end='2023-08-29')
```

In [44]:
```
date
```

Out[44]:
```
DatetimeIndex(['2023-08-23', '2023-08-24', '2023-08-25', '2023-08-26',
               '2023-08-27', '2023-08-28', '2023-08-29'],
              dtype='datetime64[ns]', freq='D')
```

In [45]:
```
df=pd.DataFrame({'Sales':[24,35,65,76,87,56,44],'Date':date,})
```

In [42]:
```
df
```

Out[42]:

	Sales	Date
0	24	2023-06-23
1	35	2023-06-24
2	65	2023-06-25
3	76	2023-06-26
4	87	2023-06-27
5	56	2023-06-28
6	44	2023-06-29

In [47]:
```
df['Sales'].rolling(window=7).mean()
```

Out[47]:
```
0          NaN
1          NaN
2          NaN
3          NaN
4          NaN
5          NaN
6    55.285714
Name: Sales, dtype: float64
```

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.

In [92]:
```
date = pd.date_range(start='2023-01-01',end='2023-01-05')
```

In [93]:
```
df=pd.DataFrame({'Date':date})
```

In [94]:
```
df
```

Out[94]:

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

In [95]:
```
weekname=['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday']
```

In [96]:
```
df['Weekday']=df['Date'].dt.weekday
```

In [97]:
```
df
```

Out[97]:

	Date	Weekday
0	2023-01-01	6
1	2023-01-02	0
2	2023-01-03	1
3	2023-01-04	2
4	2023-01-05	3

In [98]:
```
df['Weekday']=df['Weekday'].map({0:'Monday',1:'Tuesday',2:'Wednesday',3:'Thursday',4:'Friday',5:'Saturday',6:'Sunday
```

In [99]:
```
df
```

Out[99]:

	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

Q12. Given a Pandas DataFrame df with a column 'Date' that contains timestamps, write a Python function to select all rows where the date

In [101]:
```
date=pd.date_range(start='2022--01',end='2023-02-10')
```

In [104]:
```
df=pd.DataFrame({'Date': date})
```

In [105]:
```
df
```

Out[105]:

	Date
0	2022-12-01
1	2022-12-02
2	2022-12-03
3	2022-12-04
4	2022-12-05
...	...
67	2023-02-06
68	2023-02-07
69	2023-02-08
70	2023-02-09
71	2023-02-10

72 rows × 1 columns

In [144]:
```
def filter_date(df,start_date,end_date):
    df['Date']=pd.to_datetime(df['Date'])
    td=pd.Timedelta(days=1,hours=4)
    selected_rows=df[(df['Date']>=start_date) & (df['Date']<=end_date)]
    return selected_rows+td
```

In [145]:
```
filter_date(df,'2023-01-01','2023-01-31')
```

Out[145]:

	Date
31	2023-01-02 04:00:00
32	2023-01-03 04:00:00
33	2023-01-04 04:00:00
34	2023-01-05 04:00:00
35	2023-01-06 04:00:00
36	2023-01-07 04:00:00
37	2023-01-08 04:00:00
38	2023-01-09 04:00:00
39	2023-01-10 04:00:00
40	2023-01-11 04:00:00
41	2023-01-12 04:00:00
42	2023-01-13 04:00:00
43	2023-01-14 04:00:00
44	2023-01-15 04:00:00
45	2023-01-16 04:00:00
46	2023-01-17 04:00:00
47	2023-01-18 04:00:00
48	2023-01-19 04:00:00
49	2023-01-20 04:00:00
50	2023-01-21 04:00:00
51	2023-01-22 04:00:00
52	2023-01-23 04:00:00
53	2023-01-24 04:00:00
54	2023-01-25 04:00:00
55	2023-01-26 04:00:00
56	2023-01-27 04:00:00
57	2023-01-28 04:00:00
58	2023-01-29 04:00:00
59	2023-01-30 04:00:00
60	2023-01-31 04:00:00
61	2023-02-01 04:00:00

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

It is necessary that we import pandas library by entering this given function: "import pandas as pd"

In [ ]:
```
```