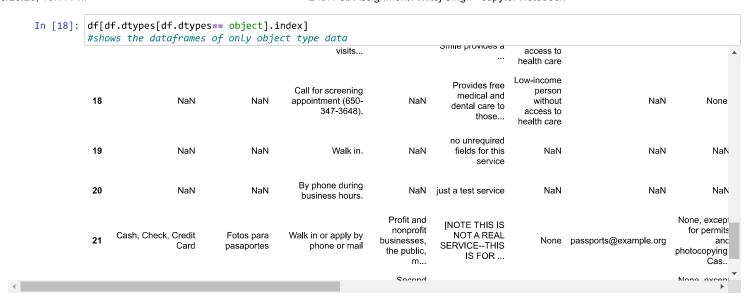
Q1. List any five functions of the pandas library with execution.

d-	f=	pd.	read_csv("Services.	csv")							
	f.head() shows first 5 row data of dataframe											
		id	location_id	program_id	accepted_payments	alternate_name	application_process	audience	description	eligibility	email	interpre
C	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	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	
ı		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	

In [4]: df.tail() #Last rows Out[4]: id location_id program_id accepted_payments alternate_name application_process description eligibility audience income Provides free Call for screening person medical and **18** 19 19 NaN appointment (650-NaN NaN NaN without dental care to 347-3648). access to those... health care no unrequired **19** 20 20 NaN NaN NaN Walk in. NaN fields for this NaN service By phone during **20** 21 21 NaN NaN NaN NaN just a test service NaN business hours. Profit and [NOTE THIS IS nonprofit Cash, Check, Credit Fotos para Walk in or apply by NOT A REAL **21** 22 22 NaN businesses, None passports@ex SERVICE--THIS phone or mail Card pasaportes the public, IS FOR ... Second [NOTE THIS IS service and Walk in or apply by NOT A REAL **22** 23 22 NaN NaN NaN nonprofit None ORGANIZATIONphone or mail businesses, -THIS IS... the p... 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 6.78233 6.711444 NaN std min 1.00000 1.000000 NaN 6.50000 25% 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')



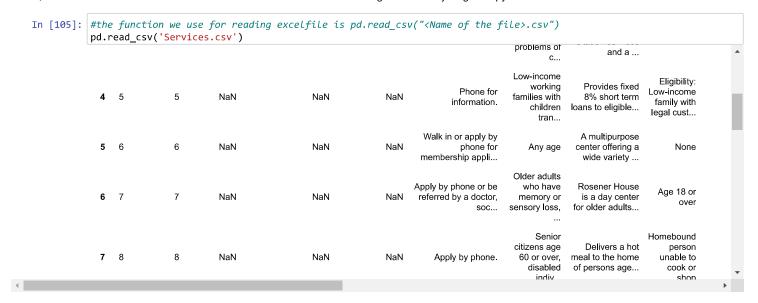
Q2. Given a Pandas DataFrame df with columns 'A', 'B', and 'C', write a Python function to re-index the DataFrame with a new index that starts from 1 and increments by 2 for each row.

```
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]:
                     Α
                                В
                                           С
                Ampere
                             Borde
                                     Chartered
            1
               Amputate
                             Born Compensate
                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']
```

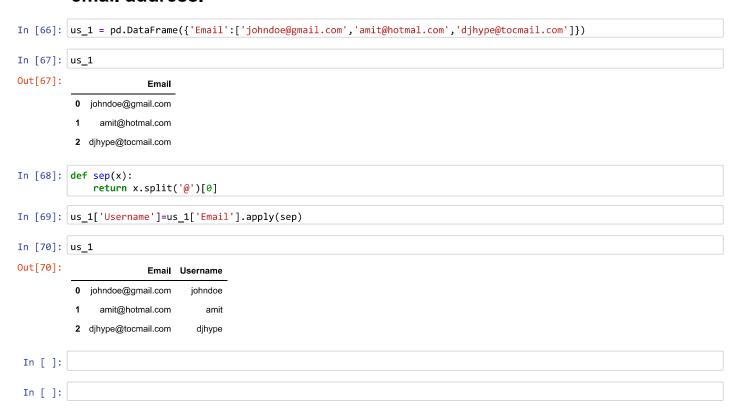
```
In [104]: da,db,dc
Out[104]: (0
                     Ampere
            1
                   Amputate
            2
                  Aviation
            3
                 Aftermath
            Name: A, dtype: object,
            0
                      Borde
            1
                       Born
            2
                  Bookmark
            3
                 Benchmark
            Name: B, dtype: object,
                  Chartered
            1
                 Compensate
                       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
                 Aviation
                Aftermath
           dtype: object
In [110]: d3
Out[110]: 1
                     Borde
                      Born
                 Bookmark
                Benchmark
           dtype: object
In [111]: d4
Out[111]: 1
                 Chartered
                Compensate
                      Clone
                 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):
               12=sum(d2[0:3])
               print(12)
 In [86]: values(d2)
           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
                                                   Amitoj
                                    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
                                                   Amitoj
                                    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
                Amitoj
                                     62
                                     47
               Andrea
            2 micheal
                                     55
                 moris
                                     74
                 Ravi
                                     32
                                     77
                 Xien
                 Jalpa
                                     86
                 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?



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.



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]:
           АВС
         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]:
            а в с
         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 [118]: stats_of(df)
          Median: Values
                             5.5
          dtype: float64
                             13.166667
          Mean : Values
          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]})
Out[121]:
              Values
            0
                  12
            1
                  44
                  66
            3
                  54
                  64
            5
                  42
                   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
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
                35 2023-06-24
           2
                65 2023-06-25
           3
                76 2023-06-26
                87 2023-06-27
                56 2023-06-28
                44 2023-06-29
```

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
           1 2023-01-02
          2 2023-01-03
           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

```
1 10000 04 041
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)]</pre>
               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 2023-01-05 04:00:00 2023-01-06 04:00:00 2023-01-07 04:00:00 2023-01-08 04:00:00 2023-01-09 04:00:00 2023-01-10 04:00:00 2023-01-11 04:00:00 2023-01-12 04:00:00 42 2023-01-13 04:00:00 2023-01-14 04:00:00 2023-01-15 04:00:00 2023-01-16 04:00:00 2023-01-17 04:00:00 2023-01-18 04:00:00 2023-01-19 04:00:00 2023-01-20 04:00:00 2023-01-21 04:00:00 2023-01-22 04:00:00 52 2023-01-23 04:00:00 2023-01-24 04:00:00 2023-01-25 04:00:00 2023-01-26 04:00:00 2023-01-27 04:00:00 2023-01-28 04:00:00 2023-01-29 04:00:00 2023-01-30 04:00:00 2023-01-31 04:00:00 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 []: