###I. import and read a CSV file

```
import pandas as pd
import numpy as np
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
# I. Import and read a CSV file
df = pd.read_csv('/content/drive/MyDrive/STUDY2/DATA ANALYSIS LAB/LABCYCLE/DATASETS/DataFrames_Dataset_4thQuestion.csv')
#df.rename(columns={'year_of_birth ': 'year_of_birth'}, inplace=True)
#df.rename(columns={'purhcase_date': 'purchase_date'}, inplace=True)
df
```

| | customer_id | year_of_birth | educational_level | marital_status | annual_income | purchase_date | recency | online_purchases | store_purchases | complaints | calls | intercoms |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 20201701 | 1982 | Graduation | Single | 58138.0 | 4/9/2012 | 58 | 8 | 4 | 0 | 3 | 11 |
| 1 | 20201702 | 1950 | Graduation | Married | 46344.0 | 8/3/2014 | 38 | 1 | 2 | 0 | 3 | 11 |
| 2 | 20201703 | 1965 | Graduation | Divorced | 71613.0 | 8/21/2013 | 26 | 8 | 10 | 0 | 3 | 11 |
| 3 | 20201704 | 1984 | Graduation | Relationship | 26646.0 | 10/2/2014 | 26 | 2 | 4 | 0 | 3 | 11 |
| 4 | 20201705 | 1981 | PhD | Widowed | 58293.0 | 1/19/2014 | 94 | 5 | 6 | 0 | 3 | 11 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 494 | 20202195 | 1944 | PhD | Divorced | 55614.0 | 11/27/2013 | 85 | 9 | 6 | 0 | 3 | 11 |
| 495 | 20202196 | 1962 | Master | Divorced | 59432.0 | 4/13/2013 | 88 | 5 | 11 | 0 | 3 | 11 |
| 496 | 20202197 | 1978 | Graduation | Divorced | 55563.0 | 5/4/2014 | 22 | 2 | 3 | 0 | 3 | 11 |
| 497 | 20202198 | 1971 | PhD | Relationship | 43624.0 | 4/21/2013 | 83 | 4 | 4 | 0 | 6 | 11 |
| 498 | 20202199 | 1949 | PhD | Relationship | 41461.0 | 5/22/2014 | 63 | 6 | 11 | 0 | 6 | 11 |

499 rows × 12 columns

###II. To Generate a basic understanding of a given data.

```
# a. Print first 5 rows and last 5 rows
print("First 5 rows:")
print(df.head())
print("Last 5 rows:")
print(df.tail())
```

```
First 5 rows:
   customer_id  year_of_birth educational_level marital_status  annual_income  \
0     20201701           1982        Graduation         Single        58138.0
1     20201702           1950        Graduation        Married        46344.0
2     20201703           1965        Graduation       Divorced        71613.0
3     20201704           1984        Graduation   Relationship        26646.0
4     20201705           1981               PhD        Widowed        58293.0

  purchase_date  recency  online_purchases  store_purchases  complaints  \
0      4/9/2012       58                 8                4           0
1      8/3/2014       38                 1                2           0
2     8/21/2013       26                 8               10           0
3     10/2/2014       26                 2                4           0
4     1/19/2014       94                 5                6           0

   calls  intercoms
0      3         11
1      3         11
2      3         11
3      3         11
4      3         11
Last 5 rows:
     customer_id  year_of_birth educational_level marital_status  \
494     20202195           1944               PhD       Divorced
495     20202196           1962            Master       Divorced
496     20202197           1978        Graduation       Divorced
497     20202198           1971               PhD   Relationship
498     20202199           1949               PhD   Relationship

     annual_income purchase_date  recency  online_purchases  store_purchases  \
494        55614.0    11/27/2013       85                 9                6
495        59432.0     4/13/2013       88                 5               11
496        55563.0      5/4/2014       22                 2                3
497        43624.0     4/21/2013       83                 4                4
498        41461.0     5/22/2014       63                 6               11

     complaints  calls  intercoms
494           0      3         11
495           0      3         11
496           0      3         11
497           0      6         11
498           0      6         11
```

]:
```
# b. Check basic information
print("Basic information:")
print(df.info())
```

]:
```
Basic information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 499 entries, 0 to 498
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   customer_id        499 non-null    int64
 1   year_of_birth      499 non-null    int64
 2   educational_level  499 non-null    object
 3   marital_status     499 non-null    object
 4   annual_income      486 non-null    float64
 5   purchase_date      499 non-null    object
 6   recency            499 non-null    int64
 7   online_purchases   499 non-null    int64
 8   store_purchases    499 non-null    int64
 9   complaints         499 non-null    int64
 10  calls              499 non-null    int64
 11  intercoms          499 non-null    int64
dtypes: float64(1), int64(8), object(3)
memory usage: 46.9+ KB
None
```

]:
```
# c. Extract the shape of the data
print("Data shape:", df.shape)
```

]:
```
Data shape: (499, 12)
```

]:
```
# d. Print unique values of marital status
print("Unique values of marital status:")
print(df['marital_status'].unique())
```

]:
```
Unique values of marital status:
['Single' 'Married' 'Divorced' 'Relationship' 'Widowed' 'Widow']
```

]:
```
# e. Make data consistent for 'widow' and 'widowed'
df['marital_status'] = df['marital_status'].replace({'widow': 'widowed'})
df['marital_status']
```

]:
```
0            Single
1           Married
2          Divorced
3      Relationship
4           Widowed
           ...
494        Divorced
495        Divorced
496        Divorced
497    Relationship
498    Relationship
Name: marital_status, Length: 499, dtype: object
```

]:
```
# f. Check for duplicates and null values
print("Duplicates:")
print(df[df.duplicated()])
print("Null values:")
print(df.isnull().sum())
```

]:
```
Duplicates:
Empty DataFrame
Columns: [customer_id, year_of_birth, educational_level, marital_status, annual_income, purchase_date, recency, online_purchases,
store_purchases, complaints, calls, intercoms]
Index: []
Null values:
customer_id           0
year_of_birth         0
educational_level     0
marital_status        0
annual_income        13
purchase_date         0
recency               0
online_purchases      0
store_purchases       0
complaints            0
calls                 0
intercoms             0
dtype: int64
```

###III. Select and filter data based on conditions:

]:
```
# a. Select a subset of data points (Birthdate, Education, and Income)
subset = df[['year_of_birth', 'educational_level', 'annual_income']]
print("Subset of data:")
print(subset)
```

]:
```
Subset of data:
     year_of_birth educational_level  annual_income
0             1982        Graduation        58138.0
1             1950        Graduation        46344.0
2             1965        Graduation        71613.0
3             1984        Graduation        26646.0
4             1981               PhD        58293.0
..             ...               ...            ...
494           1944               PhD        55614.0
495           1962            Master        59432.0
496           1978        Graduation        55563.0
497           1971               PhD        43624.0
498           1949               PhD        41461.0

[499 rows x 3 columns]
```

]:
```
# b. Retrieve the first seven data points using loc() and iloc()
print("First seven data points using loc:")
print(df.loc[:6])
print("First seven data points using iloc:")
print(df.iloc[:7])
```

]:
```
First seven data points using loc:
   customer_id  year_of_birth educational_level marital_status  annual_income  \
0     20201701           1982        Graduation         Single        58138.0
1     20201702           1950        Graduation        Married        46344.0
2     20201703           1965        Graduation       Divorced        71613.0
3     20201704           1984        Graduation   Relationship        26646.0
4     20201705           1981               PhD        Widowed        58293.0
5     20201706           1967            Master   Relationship        62000.0
6     20201707           1971        Graduation       Divorced        55635.0

  purchase_date  recency  online_purchases  store_purchases  complaints  \
0      4/9/2012       58                 8                4           0
1      8/3/2014       38                 1                2           0
2     8/21/2013       26                 8               10           0
3     10/2/2014       26                 2                4           0
4     1/19/2014       94                 5                6           0
5      9/9/2013       16                 6               10           5
6    11/13/2012       34                 7                7           0

   calls  intercoms
0      3         11
1      3         11
2      3         11
3      3         11
4      3         11
5      3         11
6      3         11
First seven data points using iloc:
   customer_id  year_of_birth educational_level marital_status  annual_income  \
0     20201701           1982        Graduation         Single        58138.0
1     20201702           1950        Graduation        Married        46344.0
2     20201703           1965        Graduation       Divorced        71613.0
3     20201704           1984        Graduation   Relationship        26646.0
4     20201705           1981               PhD        Widowed        58293.0
5     20201706           1967            Master   Relationship        62000.0
6     20201707           1971        Graduation       Divorced        55635.0

  purchase_date  recency  online_purchases  store_purchases  complaints  \
0      4/9/2012       58                 8                4           0
1      8/3/2014       38                 1                2           0
2     8/21/2013       26                 8               10           0
3     10/2/2014       26                 2                4           0
4     1/19/2014       94                 5                6           0
5      9/9/2013       16                 6               10           5
6    11/13/2012       34                 7                7           0

   calls  intercoms
0      3         11
1      3         11
2      3         11
3      3         11
4      3         11
5      3         11
6      3         11
```

]:
```
# c. Filter data using loc() and isin()
filtered_data = df.loc[df['educational_level'].isin(['PhD', 'Master'])]
```

```
print("Filtered data:")
print(filtered_data)
```

]:

```
Filtered data:
     customer_id  year_of_birth educational_level marital_status  \
4       20201705           1981              PhD        Widowed
5       20201706           1967           Master   Relationship
7       20201708           1985              PhD        Married
8       20201709           1974              PhD        Widowed
9       20201710           1950              PhD         Single
..           ...            ...              ...            ...
493     20202194           1964           Master         Single
494     20202195           1944              PhD       Divorced
495     20202196           1962           Master       Divorced
497     20202198           1971              PhD   Relationship
498     20202199           1949              PhD   Relationship

     annual_income purchase_date  recency  online_purchases  store_purchases  \
4          58293.0     1/19/2014       94                 5                6
5          62000.0      9/9/2013       16                 6               10
7          33454.0      8/5/2013       32                 4                4
8          30351.0      6/6/2013       19                 3                2
9           5648.0     3/13/2014       68                 1                0
..             ...           ...      ...               ...              ...
493        58308.0     12/1/2013       77                 2                3
494        55614.0    11/27/2013       85                 9                6
495        59432.0     4/13/2013       88                 5               11
497        43624.0     4/21/2013       83                 4                4
498        41461.0     5/22/2014       63                 6               11

     complaints  calls  intercoms
4             0      3         11
5             5      3         11
7             0      3         11
8             3      3         11
9             5      3         11
..          ...    ...        ...
493           0      3         11
494           0      3         11
495           0      3         11
497           0      6         11
498           0      6         11

[195 rows x 12 columns]
```

]:

```
# d. Customers with income > 75,000 and a Master's degree
filtered_data = df[(df['annual_income'] > 75000) & (df['educational_level'] == 'Master')]
print("Customers with income > 75,000 and Master's degree:")
print(filtered_data)
```

]:

```
Customers with income > 75,000 and Master's degree:
     customer_id  year_of_birth educational_level marital_status  \
18      20201719           1980            Master         Single
51      20201752           1964            Master         Single
55      20201756           1955            Master        Married
60      20201761           1982            Master         Single
76      20201777           1993            Master        Married
109     20201810           1993            Master         Single
120     20201821           1957            Master   Relationship
140     20201841           1987            Master         Single
217     20201918           1985            Master        Widowed
277     20201978           1981            Master         Single
305     20202006           1983            Master        Widowed
423     20202124           1973            Master   Relationship
435     20202136           1983            Master   Relationship

     annual_income purchase_date  recency  online_purchases  store_purchases  \
18         76995.0     3/28/2013       91                11                9
51         79143.0     11/8/2012        2                 6               13
55         82384.0    11/19/2012       55                 3               13
60         75777.0      4/7/2013       12                 3               11
76         75251.0     8/27/2012       34                 7                5
109        89058.0     7/12/2012       18                 5                4
120        88193.0     6/20/2013       65                 6               10
140        92859.0    10/19/2012       46                 5               12
217        83790.0    11/15/2013       81                 8                6
277        77882.0     4/30/2014       29                 3                5
305        80950.0     3/28/2013       44                 6                9
423        82584.0      4/6/2013       26                 3                8
435        82634.0     6/21/2013       49                 1                3

     complaints  calls  intercoms
18            4      3         11
51            0      3         11
55            0      3         11
60            0      3         11
```

```
76          0       3          11
109         0       3           7
120         0       5          11
140         0       3           2
217         0       3          11
277         0       3          11
305         0       3          11
423         0       3          11
435         0       0          11
```

###IV. Apply various data operation tools such as creating new variables or changing data types:

]:
```
# a. Set a new index with the variable of interest
df = df.set_index('customer_id')
df
```

]:

| | year_of_birth | educational_level | marital_status | annual_income | purchase_date | recency | online_purchases | store_purchases | complaints | calls | intercoms |
|---|---|---|---|---|---|---|---|---|---|---|---|
| customer_id | | | | | | | | | | | |
| **20201701** | 1982 | Graduation | Single | 58138.0 | 4/9/2012 | 58 | 8 | 4 | 0 | 3 | 11 |
| **20201702** | 1950 | Graduation | Married | 46344.0 | 8/3/2014 | 38 | 1 | 2 | 0 | 3 | 11 |
| **20201703** | 1965 | Graduation | Divorced | 71613.0 | 8/21/2013 | 26 | 8 | 10 | 0 | 3 | 11 |
| **20201704** | 1984 | Graduation | Relationship | 26646.0 | 10/2/2014 | 26 | 2 | 4 | 0 | 3 | 11 |
| **20201705** | 1981 | PhD | Widowed | 58293.0 | 1/19/2014 | 94 | 5 | 6 | 0 | 3 | 11 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **20202195** | 1944 | PhD | Divorced | 55614.0 | 11/27/2013 | 85 | 9 | 6 | 0 | 3 | 11 |
| **20202196** | 1962 | Master | Divorced | 59432.0 | 4/13/2013 | 88 | 5 | 11 | 0 | 3 | 11 |
| **20202197** | 1978 | Graduation | Divorced | 55563.0 | 5/4/2014 | 22 | 2 | 3 | 0 | 3 | 11 |
| **20202198** | 1971 | PhD | Relationship | 43624.0 | 4/21/2013 | 83 | 4 | 4 | 0 | 6 | 11 |
| **20202199** | 1949 | PhD | Relationship | 41461.0 | 5/22/2014 | 63 | 6 | 11 | 0 | 6 | 11 |

499 rows × 11 columns

]:
```
# b. Sort the data frame by 'annual_income' in descending order
df = df.sort_values(by='annual_income', ascending=False)
df['annual_income']
```

]:

| | year_of_birth | educational_level | marital_status | annual_income | purchase_date | recency | online_purchases | store_purchases | complaints | calls | intercoms |
|---|---|---|---|---|---|---|---|---|---|---|---|
| customer_id | | | | | | | | | | | |
| **20201865** | 1952 | PhD | Relationship | 157243.0 | 1/3/2014 | 98 | 0 | 0 | 0 | 3 | 11 |
| **20201953** | 1997 | Graduation | Widowed | 102692.0 | 5/4/2013 | 5 | 6 | 13 | 0 | 3 | 11 |
| **20201904** | 1989 | PhD | Single | 102160.0 | 2/11/2012 | 54 | 7 | 10 | 0 | 4 | 11 |
| **20201825** | 1957 | Graduation | Single | 101970.0 | 12/3/2013 | 69 | 6 | 13 | 0 | 3 | 11 |
| **20202125** | 1967 | PhD | Married | 93027.0 | 4/13/2013 | 77 | 7 | 5 | 0 | 3 | 11 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **20201793** | 1973 | Master | Relationship | NaN | 11/23/2013 | 87 | 2 | 8 | 0 | 3 | 7 |
| **20201829** | 1961 | PhD | Married | NaN | 11/7/2013 | 23 | 6 | 7 | 0 | 3 | 11 |
| **20201834** | 1980 | Graduation | Relationship | NaN | 11/8/2013 | 96 | 6 | 7 | 0 | 3 | 2 |
| **20202013** | 1990 | Graduation | Single | NaN | 3/6/2013 | 69 | 6 | 12 | 0 | 3 | 11 |
| **20202020** | 1997 | Graduation | Divorced | NaN | 8/23/2013 | 67 | 2 | 10 | 0 | 3 | 11 |

499 rows × 11 columns

]:
```
# c. Create a new variable for the sum of purchases
df['total_purchases'] = df['online_purchases'] + df['store_purchases']
df['total_purchases']
```

]:
```
0      12
1       3
2      18
3       6
4      11
       ..
494    15
495    16
496     5
497     8
498    17
Name: total_purchases, Length: 499, dtype: int64
```

]:
```
# d. Change the datatype of 'purchase_date' to datetime
df['purchase_date'] = pd.to_datetime(df['purchase_date'])
print(df['purchase_date'].dtypes)
```

]:
```
datetime64[ns]
```

]:
```
# e. Determine the age based on 'year_of_birth'
current_year = pd.Timestamp.now().year
```

```
df['age'] = current_year - df['year_of_birth']
df['age']
```

```
customer_id
20201865    71
20201953    26
20201904    34
20201825    66
20202125    56
            ..
20201793    50
20201829    62
20201834    43
20202013    33
20202020    26
Name: age, Length: 499, dtype: int64
```

```
# f. Create the week date from 'purchase_date'
df['week_date'] = df['purchase_date'].dt.strftime('%Y-%U')
df['week_date']
```

```
customer_id
20201865    2014-00
20201953    2013-17
20201904    2012-06
20201825    2013-48
20202125    2013-14
             ...
20201793    2013-46
20201829    2013-44
20201834    2013-44
20202013    2013-09
20202020    2013-33
Name: week_date, Length: 499, dtype: object
```

###V. perform data aggregation using group by and pivot table methods

```
# a. Group by educational level and calculate the mean of income, recency, online purchases, and store purchases
grouped_data = df.groupby('educational_level').agg({
    'annual_income': 'mean',
    'recency': 'mean',
    'online_purchases': 'mean',
    'store_purchases': 'mean'
})
print("Grouped data by educational level:")
print(grouped_data)
```

```
Grouped data by educational level:
                   annual_income    recency  online_purchases  store_purchases
educational_level
Basic               19514.571429  53.571429          1.571429         2.857143
Graduation          51607.827309  47.171206          3.887160         5.840467
High School         44154.717949  58.400000          3.450000         4.600000
Master              51191.700000  45.000000          4.049383         5.691358
PhD                 55878.990991  49.008772          4.429825         6.298246
```

```
# b. Use pivot_table to find aggregated sum of purchases and mean of recency per education and marital status group
pivot_table_data = pd.pivot_table(df, values=['online_purchases', 'store_purchases', 'recency'],
                                  index=['educational_level', 'marital_status'],
                                  aggfunc={'online_purchases': 'sum', 'store_purchases': 'sum', 'recency': 'mean'})
print("Pivot table data:")
pivot_table_data
```

```
Pivot table data:
```

| | | online_purchases | recency | store_purchases |
|---|---|---|---|---|
| educational_level | marital_status | | | |
| Basic | Divorced | 4 | 68.333333 | 9 |
| | Relationship | 6 | 39.333333 | 9 |
| | Single | 1 | 52.000000 | 2 |
| Graduation | Divorced | 195 | 54.897959 | 286 |
| | Married | 249 | 42.701493 | 403 |
| | Relationship | 179 | 48.196078 | 285 |
| | Single | 258 | 44.278689 | 365 |
| | Widow | 12 | 61.000000 | 22 |
| | Widowed | 106 | 46.760000 | 140 |
| High School | Divorced | 16 | 64.666667 | 15 |
| | Married | 49 | 66.866667 | 71 |
| | Relationship | 42 | 49.615385 | 57 |
| | Single | 23 | 49.000000 | 31 |

| | | online_purchases | recency | store_purchases |
|---|---|---|---|---|
| educational_level | marital_status | | | |
| | Widow | 4 | 96.000000 | 4 |
| | Widowed | 4 | 52.000000 | 6 |
| Master | Divorced | 52 | 60.083333 | 82 |
| | Married | 87 | 50.315789 | 106 |
| | Relationship | 61 | 36.800000 | 98 |
| | Single | 86 | 42.761905 | 132 |
| | Widow | 9 | 14.000000 | 5 |
| | Widowed | 33 | 40.000000 | 38 |
| PhD | Divorced | 106 | 41.350000 | 126 |
| | Married | 91 | 60.000000 | 145 |
| | Relationship | 147 | 43.161290 | 217 |
| | Single | 71 | 49.315789 | 102 |
| | Widow | 3 | 25.000000 | 3 |
| | Widowed | 87 | 53.684211 | 125 |