```
import pandas as pd
import numpy as np                          # Read the Data with Pandas
csv_data = pd.read_csv("https://storage.googleapis.com/dqlab-dataset/shopping_data.csv")
print(csv_data)
```

```
     CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-100)
0             1    Male   19                  15                      39
1             2    Male   21                  15                      81
2             3  Female   20                  16                       6
3             4  Female   23                  16                      77
4             5  Female   31                  17                      40
..          ...     ...  ...                 ...                     ...
195         196  Female   35                 120                      79
196         197  Female   45                 126                      28
197         198    Male   32                 126                      74
198         199    Male   32                 137                      18
199         200    Male   30                 137                      83

[200 rows x 5 columns]
```

```
print(csv_data.head())            # show only first five rows
```

```
   CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male   19                  15                      39
1           2    Male   21                  15                      81
2           3  Female   20                  16                       6
3           4  Female   23                  16                      77
4           5  Female   31                  17                      40
5           6  Female   22                  17                      76
6           7  Female   35                  18                       6
7           8  Female   23                  18                      94
```

```
print(csv_data.head(10))              # show only first n rows
```

```
   CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male   19                  15                      39
1           2    Male   21                  15                      81
2           3  Female   20                  16                       6
3           4  Female   23                  16                      77
4           5  Female   31                  17                      40
5           6  Female   22                  17                      76
6           7  Female   35                  18                       6
7           8  Female   23                  18                      94
8           9    Male   64                  19                       3
9          10  Female   30                  19                      72
```

```
print(csv_data.tail())             # show only last five rows
```

```
     CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-100)
195         196  Female   35                 120                      79
196         197  Female   45                 126                      28
197         198    Male   32                 126                      74
198         199    Male   32                 137                      18
199         200    Male   30                 137                      83
```

```
print(csv_data.tail(8))             # show only last n rows
```

```
     CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-100)
192         193    Male   33                 113                       8
193         194  Female   38                 113                      91
194         195  Female   47                 120                      16
195         196  Female   35                 120                      79
196         197  Female   45                 126                      28
197         198    Male   32                 126                      74
198         199    Male   32                 137                      18
199         200    Male   30                 137                      83
```

```
print(csv_data.tail(-4))                    #For negative values of n, this function returns all rows except the first |n| rows
```

```
     CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-100)
4             5  Female   31                  17                      40
5             6  Female   22                  17                      76
6             7  Female   35                  18                       6
7             8  Female   23                  18                      94
8             9    Male   64                  19                       3
..          ...     ...  ...                 ...                     ...
195         196  Female   35                 120                      79
```

```
196         197  Female   45                 126                      28
197         198    Male   32                 126                      74
198         199    Male   32                 137                      18
199         200    Male   30                 137                      83

[196 rows x 5 columns]
```

```
print(csv_data.columns)              #.columns to access the column of the data source
```

```
Index(['CustomerID', 'Genre', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
```

```
csv_data.index
```

```
RangeIndex(start=0, stop=200, step=1)
```

```
csv_data.dtypes                                  #Return the dtypes in the DataFrame
```

	0
CustomerID	int64
Genre	object
Age	int64
Annual Income (k$)	int64
Spending Score (1-100)	int64

dtype: object

```
csv_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   CustomerID              200 non-null    int64
 1   Genre                   200 non-null    object
 2   Age                     200 non-null    int64
 3   Annual Income (k$)      200 non-null    int64
 4   Spending Score (1-100)  200 non-null    int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

```
csv_data.values
```

```
 [74, 'Female', 60, 56.0, 56],
 [75, 'Male', 59, 54.0, 47],
 [76, 'Male', 26, 54.0, 54],
 [77, 'Female', 45, 54.0, 53],
 [78, 'Male', 40, 54.0, 48],
 [79, 'Female', 23, 54.0, 52],
 [80, 'Female', 49, 54.0, 42],
 [81, 'Male', 57, 54.0, 51],
 [82, 'Male', 38, 54.0, 55],
 [83, 'Male', 67, 54.0, 41],
 [84, 'Female', 46, 54.0, 44],
 [85, 'Female', 21, 54.0, 57],
 [86, 'Male', 48, 54.0, 46],
 [87, 'Female', 55, 57.0, 58],
 [88, 'Female', 22, 57.0, 55],
 [89, 'Female', 34, 58.0, 60],
 [90, 'Female', 50, 58.0, 46],
 [91, 'Female', 68, 59.0, 55],
 [92, 'Male', 18, 59.0, 41],
 [93, 'Male', 48, 60.0, 49],
 [94, 'Female', 40, 60.0, 40],
 [95, 'Female', 32, 60.0, 42],
 [96, 'Male', 24, 60.0, 52],
 [97, 'Female', 47, 60.0, 47],
 [98, 'Female', 27, 60.0, 50],
 [99, 'Male', 48, 61.0, 42],
 [100, 'Male', 20, 61.0, 49],
 [101, 'Female', 23, 62.0, 41],
 [102, 'Female', 49, 62.0, 48],
```

```
csv_data.value_counts()
```

CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)	count
1	Male	19	15.0	39	1
138	Male	32	73.0	73	1
128	Male	40	71.0	95	1
129	Male	59	71.0	11	1
130	Male	38	71.0	75	1
...	...	...	...	...	...
70	Female	32	48.0	47	1
71	Male	70	49.0	55	1
72	Female	47	49.0	42	1
73	Female	60	50.0	49	1
200	Male	30	137.0	83	1

200 rows × 1 columns

dtype: int64

```
csv_data.ndim                        #Return an int representing the number of axes / array dimensions
```

```
2
```

```
csv_data.size                       #Return an int representing the number of elements in this object
```

```
1000
```

```
csv_data.shape                                #Return a tuple representing the dimensionality of the DataFrame
```

```
(200, 5)
```

```
csv_data.empty                       # Indicator whether Series/DataFrame is empty
```

```
False
```

```
print(csv_data["Age"])                    #access column using label
```

```
0      19
1      21
2      20
3      23
4      31
       ..
195    35
196    45
197    32
198    32
199    30
Name: Age, Length: 200, dtype: int64
```

```
csv_data.loc[5,'CustomerID']                          # Access a group of rows and columns by label(s)
```

```
6
```

```
csv_data.loc[5]
```

	5
CustomerID	6
Genre	Female
Age	22
Annual Income (k$)	17
Spending Score (1-100)	76

dtype: object

```
print(csv_data.loc[2:5])
```

```
   CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-100)
2           3  Female   20                  16                       6
3           4  Female   23                  16                      77
4           5  Female   31                  17                      40
5           6  Female   22                  17                      76
```

```
print(csv_data.iloc[5])                      #Access row
```

```
CustomerID                    6
Genre                    Female
Age                          22
Annual Income (k$)           17
Spending Score (1-100)       76
Name: 5, dtype: object
```

```
print(csv_data["Age"].iloc[1])
```

```
21
```

```
print("Shows data to 5th to less than 10th in a row:")                  #Data Based on Range
print(csv_data.iloc[5:10])
```

```
Shows data to 5th to less than 10th in a row:
   CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-100)
5           6  Female   22                  17                      76
6           7  Female   35                  18                       6
7           8  Female   23                  18                      94
8           9    Male   64                  19                       3
9          10  Female   30                  19                      72
```

```
# Data Filtering
print(csv_data[csv_data.Genre=='Female'])
```

```
     CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-100)
2             3  Female   20                  16                       6
3             4  Female   23                  16                      77
4             5  Female   31                  17                      40
5             6  Female   22                  17                      76
6             7  Female   35                  18                       6
..          ...     ...  ...                 ...                     ...
191         192  Female   32                 103                      69
193         194  Female   38                 113                      91
```

```
194         195  Female   47                 120                      16
195         196  Female   35                 120                      79
196         197  Female   45                 126                      28

[112 rows x 5 columns]
```

```
print(csv_data[(csv_data.Genre=='Female') & (csv_data.Age>60)])
```

```
     CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-100)
40           41  Female   65                  38                      35
62           63  Female   67                  47                      52
67           68  Female   68                  48                      48
90           91  Female   68                  59                      55
106         107  Female   66                  63                      50
116         117  Female   63                  65                      43
```

```
# str accessor to filter rows based on strings.
csv_data[csv_data.Genre.str.startswith('M')]
```

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
8	9	Male	64	19	3
10	11	Male	67	19	14
14	15	Male	37	20	13
...	...	...	...	...	...
187	188	Male	28	101	68
192	193	Male	33	113	8
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

88 rows × 5 columns

```
# isin method to filter the names that exist in a given list
names = ['M','F','Male']
csv_data[csv_data.Genre.isin(names)]
```

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
8	9	Male	64	19	3
10	11	Male	67	19	14
14	15	Male	37	20	13
...	...	...	...	...	...
187	188	Male	28	101	68
192	193	Male	33	113	8
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

88 rows × 5 columns

```
# query function can pass the conditions as a string
csv_data.query('Genre == "Female" and Age > 60')
```

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
40	41	Female	65	38	35
62	63	Female	67	47	52
67	68	Female	68	48	48
90	91	Female	68	59	55
106	107	Female	66	63	50
116	117	Female	63	65	43

```
# Descriptive statistics include those that summarize the central tendency, dispersion and shape of a dataset’s distribution, excluding NaN
print(csv_data.describe())
```

```
       CustomerID         Age  Annual Income (k$)  Spending Score (1-100)
count  200.000000  200.000000          200.000000              200.000000
mean   100.500000   38.850000           60.560000               50.200000
std     57.879185   13.969007           26.264721               25.823522
min      1.000000   18.000000           15.000000                1.000000
25%     50.750000   28.750000           41.500000               34.750000
50%    100.500000   36.000000           61.500000               50.000000
75%    150.250000   49.000000           78.000000               73.000000
max    200.000000   70.000000          137.000000               99.000000
```

```
# For numeric data, the result’s index will include count, mean, std, min, max as well as lower, 50 and upper percentiles
# For object data (e.g. strings or timestamps), the result’s index will include count, unique, top, and freq.
# The top is the most common value. The freq is the most common value’s frequency. Timestamps also include the first and last items
print(csv_data.describe(include="all"))
```

```
        CustomerID   Genre         Age  Annual Income (k$)  \
count   200.000000     200  200.000000          200.000000
unique         NaN       2         NaN                 NaN
top            NaN  Female         NaN                 NaN
freq           NaN     112         NaN                 NaN
mean    100.500000     NaN   38.850000           60.560000
std      57.879185     NaN   13.969007           26.264721
min       1.000000     NaN   18.000000           15.000000
25%      50.750000     NaN   28.750000           41.500000
50%     100.500000     NaN   36.000000           61.500000
75%     150.250000     NaN   49.000000           78.000000
max     200.000000     NaN   70.000000          137.000000

        Spending Score (1-100)
count               200.000000
unique                    NaN
top                       NaN
freq                      NaN
mean                50.200000
std                 25.823522
min                  1.000000
25%                 34.750000
50%                 50.000000
75%                 73.000000
max                 99.000000
```

```
csv_data.describe(include=[np.number])                          # Including only numeric columns
```

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

```
print(csv_data.describe(exclude=["O"]))    # ignore non-numeric data for processing
```

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

```
csv_data.isnull()
```

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
195	False	False	False	False	False
196	False	False	False	False	False
197	False	False	False	False	False
198	False	False	False	False	False
199	False	False	False	False	False

200 rows × 5 columns

```
csv_data.isna()
```

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
195	False	False	False	False	False
196	False	False	False	False	False
197	False	False	False	False	False
198	False	False	False	False	False
199	False	False	False	False	False

200 rows × 5 columns

```
print(csv_data.isnull().values.any())                # find missing value
```

```
False
```

```
print(csv_data.isna().values.any())
```

```
False
```

```
print(csv_data.duplicated().values.any())
```

```
False
```

```
data_missing = pd.read_csv("https://storage.googleapis.com/dqlab-dataset/shopping_data_missingvalue.csv")
print(data_missing.isnull().values.any())
```

```
True
```

```
data_missing.dropna()
```

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19.0	15.0	39.0
3	4	Female	23.0	16.0	77.0
6	7	Female	35.0	18.0	6.0
7	8	Female	23.0	18.0	94.0
9	10	Female	30.0	19.0	72.0
...	...	...	...	...	...
195	196	Female	35.0	120.0	79.0
196	197	Female	45.0	126.0	28.0
197	198	Male	32.0	126.0	74.0
198	199	Male	32.0	137.0	18.0
199	200	Male	30.0	137.0	83.0

195 rows × 5 columns

```
print(data_missing.isnull().values.any())
```

```
True
```

```
data_missing=data_missing.dropna()
```

```
print(data_missing.isnull().values.any())
```

```
False
```

```
csv_data= csv_data.drop("Genre", axis=1)    # drop column
print(csv_data)
```

```
     CustomerID  Age  Annual Income (k$)  Spending Score (1-100)
0             1   19                  15                      39
1             2   21                  15                      81
2             3   20                  16                       6
3             4   23                  16                      77
4             5   31                  17                      40
..          ...  ...                 ...                     ...
195         196   35                 120                      79
196         197   45                 126                      28
197         198   32                 126                      74
198         199   32                 137                      18
199         200   30                 137                      83

[200 rows x 4 columns]
```

```
csv_data.mean()
```

	0
CustomerID	100.50
Age	38.85
Annual Income (k$)	60.56
Spending Score (1-100)	50.20

dtype: float64

```
csv_data.median()
```

	0
CustomerID	100.5
Age	36.0
Annual Income (k$)	61.5
Spending Score (1-100)	50.0

dtype: float64

```
data_filling=data_missing.fillna(csv_data.mean())

print(data_filling.head(10))
```

```
     CustomerID   Genre   Age  Annual Income (k$)  Spending Score (1-100)
0             1    Male  19.0                15.0                    39.0
3             4  Female  23.0                16.0                    77.0
6             7  Female  35.0                18.0                     6.0
7             8  Female  23.0                18.0                    94.0
9            10  Female  30.0                19.0                    72.0
10           11    Male  67.0                19.0                    14.0
11           12  Female  35.0                19.0                    99.0
12           13  Female  58.0                20.0                    15.0
13           14  Female  24.0                20.0                    77.0
14           15    Male  37.0                20.0                    13.0
```

```
data_filling=data_missing.fillna(csv_data.median())

print(data_filling.head(10))
```

```
     CustomerID   Genre   Age  Annual Income (k$)  Spending Score (1-100)
0             1    Male  19.0                15.0                    39.0
3             4  Female  23.0                16.0                    77.0
6             7  Female  35.0                18.0                     6.0
7             8  Female  23.0                18.0                    94.0
9            10  Female  30.0                19.0                    72.0
10           11    Male  67.0                19.0                    14.0
11           12  Female  35.0                19.0                    99.0
12           13  Female  58.0                20.0                    15.0
13           14  Female  24.0                20.0                    77.0
14           15    Male  37.0                20.0                    13.0
```

```
#In order to convert data types in pandas, there are three basic options:

#Use astype() to force an appropriate dtype
#Create a custom function to convert the data
#Use pandas functions such as to_numeric() or to_datetime()

csv_data['Annual Income (k$)'].astype('float')
```

	Annual Income (k$)
0	15.0
1	15.0
2	16.0
3	16.0
4	17.0
...	...
195	120.0
196	126.0
197	126.0
198	137.0
199	137.0

200 rows × 1 columns

dtype: float64

```
csv_data.dtypes
```

	0
CustomerID	int64
Genre	object
Age	int64
Annual Income (k$)	int64
Spending Score (1-100)	int64

dtype: object

```
csv_data['Annual Income (k$)'] = csv_data['Annual Income (k$)'].astype('float')
csv_data.dtypes
```

	0
CustomerID	int64
Genre	object
Age	int64
Annual Income (k$)	float64
Spending Score (1-100)	int64

dtype: object