import dataset

```
import pandas as pd
df = pd.read_csv("/content/Mall_Customers4.CSV")
print(df.info())
print(df.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 203 entries, 0 to 202
Data columns (total 7 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   CustomerID              201 non-null    float64
 1   Gender                  200 non-null    object 
 2   Age                     200 non-null    float64
 3   Annual Income (k$)      201 non-null    float64
 4   Spending Score (1-100)  199 non-null    float64
 5   Unnamed: 5              0 non-null      float64
 6   Date                    201 non-null    object 
dtypes: float64(5), object(2)
memory usage: 11.2+ KB
None
       CustomerID         Age  Annual Income (k$)  Spending Score (1-100)  \
count  201.000000  200.000000          201.000000              199.000000   
mean   100.059701   38.825000           60.353234               50.904523   
std     58.070788   14.000516           26.362466               25.698444   
min      1.000000   18.000000           15.000000                1.000000   
25%     50.000000   28.000000           40.000000               35.000000   
50%    100.000000   36.000000           61.000000               50.000000   
75%    150.000000   49.000000           78.000000               73.000000   
max    200.000000   70.000000          137.000000               99.000000   

       Unnamed: 5  
count         0.0  
mean          NaN  
std           NaN  
min           NaN  
25%           NaN  
50%           NaN  
75%           NaN  
max           NaN  
```

```
# CLEAN & FILL MISSING VALUES

# Clean column names (remove extra spaces)
df.columns = df.columns.str.strip()

# Standardize column names
df.rename(columns=lambda x: x.strip(), inplace=True)

# Fill Gender
if any(df.columns.str.contains("Gender", case=False)):
    gender_col = [c for c in df.columns if "gender" in c.lower()][0]
    df[gender_col] = df[gender_col].replace('', pd.NA)
    df[gender_col] = df[gender_col].fillna("Female")

# Fill Age
if any(df.columns.str.contains("Age", case=False)):
    age_col = [c for c in df.columns if "age" in c.lower()][0]
    df[age_col] = pd.to_numeric(df[age_col], errors='coerce')
    df[age_col] = df[age_col].fillna(30)
```

```
#  Fill Spending Score
if any(df.columns.str.contains("Spending", case=False)):
    spend_col = [c for c in df.columns if "spending" in c.lower()][0]
    df[spend_col] = pd.to_numeric(df[spend_col], errors='coerce')
    df[spend_col] = df[spend_col].fillna(72)

#  Drop Unwanted Column (if exists)
if "unnamed:_5" in df.columns:
    df = df.drop(columns=["unnamed:_5"])

# Fill Missing CustomerID
if "customerid" in df.columns:
    df['customerid'] = pd.to_numeric(df['customerid'], errors='coerce')
    df['customerid'] = df['customerid'].fillna(df['customerid'].max() + 1)

# Fill Annual Income
if any(df.columns.str.contains("annual", case=False)):
    income_col = [c for c in df.columns if "annual" in c.lower()][0]
    df[income_col] = pd.to_numeric(df[income_col], errors='coerce')
    df[income_col] = df[income_col].fillna(df[income_col].median())

# Fill Date
if "date" in df.columns:
    df['date'] = df['date'].fillna(method='ffill')



# View updated dataset
print(df.head(25))
print("\nTotal Rows after cleaning:", len(df))
```

```
    customerid  gender   age  annual_income_(k$)  spending_score_(1-100)  \
0          1.0    Male  19.0                15.0                    39.0
1          2.0    Male  21.0                15.0                    81.0
2          3.0  Female  20.0                16.0                     6.0
3          4.0  Female  23.0                16.0                    77.0
4          5.0  Female  30.0                17.0                    40.0
5          6.0  Female  22.0                17.0                    76.0
6          7.0  Female  35.0                18.0                    72.0
7          8.0  Female  23.0                18.0                    94.0
8          9.0    Male  64.0                19.0                     3.0
9         10.0  Female  30.0                19.0                    72.0
10        11.0  Female  67.0                19.0                    72.0
11        12.0  Female  35.0                19.0                    99.0
13        13.0  Female  58.0                20.0                    15.0
14        14.0  Female  24.0                20.0                    77.0
15        15.0    Male  37.0                20.0                    13.0
16        16.0    Male  22.0                20.0                    79.0
17        17.0  Female  35.0                21.0                    35.0
18        18.0    Male  20.0                21.0                    66.0
19        19.0    Male  52.0                23.0                    29.0
20        20.0  Female  35.0                23.0                    98.0
21        21.0    Male  35.0                24.0                    35.0
22        22.0    Male  25.0                24.0                    73.0
23        23.0  Female  46.0                25.0                     5.0
24        24.0    Male  31.0                25.0                    73.0
25        25.0  Female  54.0                28.0                    14.0

         date
0  2024-01-01
1  2024-01-02
2  2024-01-03
3  2024-01-04
```

```
4  2024-01-05
5  2024-01-06
6  2024-01-07
7  2024-01-08
8  2024-01-09
9  2024-01-10
10 2024-01-11
11 2024-01-12
13 2024-01-14
14 2024-01-15
15 2024-01-16
16 2024-01-17
17 2024-01-18
18 2024-01-19
19 2024-01-20
20 2024-01-21
21 2024-01-22
22 2024-01-23
23 2024-01-24
24 2024-01-25
25 2024-01-26

Total Rows after cleaning: 201
/tmp/ipython-input-3397940517.py:44: FutureWarning: Series.fillna with 'method' is deprecated and wi
  df['date'] = df['date'].fillna(method='ffill')
```

```
duplicates_removed = df.duplicated().sum()  # duplicate rows count
df = df.drop_duplicates()
print(df)
```

```
     customerid  gender   age  annual_income_(k$)  spending_score_(1-100)  \
0           1.0    Male  19.0                15.0                    39.0
1           2.0    Male  21.0                15.0                    81.0
2           3.0  Female  20.0                16.0                     6.0
3           4.0  Female  23.0                16.0                    77.0
4           5.0  Female  30.0                17.0                    40.0
..          ...     ...   ...                 ...                     ...
197       197.0  Female  45.0               126.0                    28.0
198       198.0    Male  32.0               126.0                    74.0
199       199.0    Male  32.0               137.0                    18.0
200       200.0    Male  30.0               137.0                    83.0
201       201.0  Female  30.0                61.5                    72.0

          date
0   2024-01-01
1   2024-01-02
2   2024-01-03
3   2024-01-04
4   2024-01-05
..         ...
197 2024-07-16
198 2024-07-17
199 2024-07-18
200 2024-07-19
201 2024-07-19

[201 rows x 6 columns]
```

```
# Lowercase column names and remove spaces
df.columns = df.columns.str.lower().str.replace(" ", "_")

# Convert Date to datetime
df['date'] = pd.to_datetime(df['date'], format='%d-%m-%Y')
```

```
summary = {
    "Total rows": len(df),
    "Missing values per column": df.isnull().sum().to_dict(),
    "Duplicates removed": duplicates_removed,
    "Data types": df.dtypes.to_dict()
}

summary
```

```
{'Total rows': 201,
 'Missing values per column': {'customerid': 0,
  'gender': 0,
  'age': 0,
  'annual_income_(k$)': 0,
  'spending_score_(1-100)': 0,
  'date': 0},
 'Duplicates removed': np.int64(0),
 'Data types': {'customerid': dtype('float64'),
  'gender': dtype('O'),
  'age': dtype('float64'),
  'annual_income_(k$)': dtype('float64'),
  'spending_score_(1-100)': dtype('float64'),
  'date': dtype('<M8[ns]')}}
```