Asg 19.3

Working with Pandas DataFrames -- The Essentials (Part One)

(Coding)

Files needed for this assignment:

marketing_campaign.csv (https://drive.google.com/file/d/1Lx2V9-9j_t_9hTSdFpLmmP_5M_1sE4Wf/view?usp=share_link)
houses train.txt (https://drive.google.com/file/d/1E1SzyR0dhIOrMIoKs3UBL8pibFpVwLis/view?usp=share_link)

In [3]:
```python
# set up notebook to display multiple output in one cell

from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

print('The notebook is set up to display multiple output in one cell.')
```

```
The notebook is set up to display multiple output in one cell.
```

In [4]:
```python
# conventional way to import pandas and numpy

import pandas as pd
import numpy as np
```

PART ONE

For Questions 1-7: We will be using the **'marketing_campaign.csv' dataset** and the **customers DataFrame**.

In [ ]:
```
 
```

Question 1:

Read in the dataset **'marketing_campaign.csv'** and store the results in a DataFrame named **customers**.

marketing_campaign.csv (https://drive.google.com/file/d/1Lx2V9-9j_t_9hTSdFpLmmP_5M_1sE4Wf/view?usp=share_link)

```
In [11]: customers = pd.read_csv('marketing_campaign.csv',sep=';')
         print(customers)
```

```
         ID  Year_Birth   Education Marital_Status   Income  Kidhome  \
0      5524        1957  Graduation         Single  58138.0        0   
1      2174        1954  Graduation         Single  46344.0        1   
2      4141        1965  Graduation       Together  71613.0        0   
3      6182        1984  Graduation       Together  26646.0        1   
4      5324        1981         PhD        Married  58293.0        1   
...     ...         ...         ...            ...      ...      ...   
2235  10870        1967  Graduation        Married  61223.0        0   
2236   4001        1946         PhD       Together  64014.0        2   
2237   7270        1981  Graduation       Divorced  56981.0        0   
2238   8235        1956      Master       Together  69245.0        0   
2239   9405        1954         PhD        Married  52869.0        1   

      Teenhome Dt_Customer  Recency  MntWines  ...  NumWebVisitsMonth  \
0            0  2012-09-04       58       635  ...                  7   
1            1  2014-03-08       38        11  ...                  5   
2            0  2013-08-21       26       426  ...                  4   
3            0  2014-02-10       26        11  ...                  6   
4            0  2014-01-19       94       173  ...                  5   
...        ...         ...      ...       ...  ...                ...   
2235         1  2013-06-13       46       709  ...                  5   
2236         1  2014-06-10       56       406  ...                  7   
2237         0  2014-01-25       91       908  ...                  6   
2238         1  2014-01-24        8       428  ...                  3   
2239         1  2012-10-15       40        84  ...                  7   

      AcceptedCmp3  AcceptedCmp4  AcceptedCmp5  AcceptedCmp1  AcceptedCmp2  
\
0                0             0             0             0             0   
1                0             0             0             0             0   
2                0             0             0             0             0   
3                0             0             0             0             0   
4                0             0             0             0             0   
...            ...           ...           ...           ...           ...   
2235             0             0             0             0             0   
2236             0             0             0             1             0   
2237             0             1             0             0             0   
2238             0             0             0             0             0   
2239             0             0             0             0             0   

      Complain  Z_CostContact  Z_Revenue  Response  
0            0              3         11         1  
1            0              3         11         0  
2            0              3         11         0  
3            0              3         11         0  
4            0              3         11         0  
...        ...            ...        ...       ...  
2235         0              3         11         0  
2236         0              3         11         0  
2237         0              3         11         0  
2238         0              3         11         0  
2239         0              3         11         1  
```

```
[2240 rows x 29 columns]
```

Question 2:

Use the following attributes to inspect the **customers** DataFrame.

- index
- columns
- shape
- dtypes

In [18]:
```
print(customers.index)
print(customers.columns)
print(customers.shape)
print(customers.dtypes)
```

```
RangeIndex(start=0, stop=2240, step=1)
Index(['ID', 'Year_Birth', 'Education', 'Marital_Status', 'Income', 'Kidhom
e',
       'Teenhome', 'Dt_Customer', 'Recency', 'MntWines', 'MntFruits',
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
       'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth',
       'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1',
       'AcceptedCmp2', 'Complain', 'Z_CostContact', 'Z_Revenue', 'Respons
e'],
      dtype='object')
(2240, 29)
ID                       int64
Year_Birth               int64
Education               object
Marital_Status          object
Income                 float64
Kidhome                  int64
Teenhome                 int64
Dt_Customer             object
Recency                  int64
MntWines                 int64
MntFruits                int64
MntMeatProducts          int64
MntFishProducts          int64
MntSweetProducts         int64
MntGoldProds             int64
NumDealsPurchases        int64
NumWebPurchases          int64
NumCatalogPurchases      int64
NumStorePurchases        int64
NumWebVisitsMonth        int64
AcceptedCmp3             int64
AcceptedCmp4             int64
AcceptedCmp5             int64
AcceptedCmp1             int64
AcceptedCmp2             int64
Complain                 int64
Z_CostContact            int64
Z_Revenue                int64
Response                 int64
dtype: object
```

Question 3:

Use the following methods to inspect the **customers** DataFrame.

- head()
- tail()

- info()

In [19]:
```python
print(customers.head())
print(customers.tail())
print(customers.info())
```

```
     ID  Year_Birth   Education Marital_Status   Income  Kidhome  Teenhome  
\
0  5524        1957  Graduation         Single  58138.0        0         0   
1  2174        1954  Graduation         Single  46344.0        1         1   
2  4141        1965  Graduation       Together  71613.0        0         0   
3  6182        1984  Graduation       Together  26646.0        1         0   
4  5324        1981         PhD        Married  58293.0        1         0   

  Dt_Customer  Recency  MntWines  ...  NumWebVisitsMonth  AcceptedCmp3  \
0  2012-09-04       58       635  ...                  7             0   
1  2014-03-08       38        11  ...                  5             0   
2  2013-08-21       26       426  ...                  4             0   
3  2014-02-10       26        11  ...                  6             0   
4  2014-01-19       94       173  ...                  5             0   

   AcceptedCmp4  AcceptedCmp5  AcceptedCmp1  AcceptedCmp2  Complain  \
0             0             0             0             0         0   
1             0             0             0             0         0   
2             0             0             0             0         0   
3             0             0             0             0         0   
4             0             0             0             0         0   

   Z_CostContact  Z_Revenue  Response  
0              3         11         1  
1              3         11         0  
2              3         11         0  
3              3         11         0  
4              3         11         0  

[5 rows x 29 columns]
         ID  Year_Birth   Education Marital_Status   Income  Kidhome  \
2235  10870        1967  Graduation        Married  61223.0        0   
2236   4001        1946         PhD       Together  64014.0        2   
2237   7270        1981  Graduation       Divorced  56981.0        0   
2238   8235        1956      Master       Together  69245.0        0   
2239   9405        1954         PhD        Married  52869.0        1   

      Teenhome Dt_Customer  Recency  MntWines  ...  NumWebVisitsMonth  \
2235         1  2013-06-13       46       709  ...                  5   
2236         1  2014-06-10       56       406  ...                  7   
2237         0  2014-01-25       91       908  ...                  6   
2238         1  2014-01-24        8       428  ...                  3   
2239         1  2012-10-15       40        84  ...                  7   

      AcceptedCmp3  AcceptedCmp4  AcceptedCmp5  AcceptedCmp1  AcceptedCmp2  
\
2235             0             0             0             0             0   
2236             0             0             0             1             0   
2237             0             1             0             0             0   
2238             0             0             0             0             0   
2239             0             0             0             0             0   
```

```
      Complain  Z_CostContact  Z_Revenue  Response
2235         0              3         11         0
2236         0              3         11         0
2237         0              3         11         0
2238         0              3         11         0
2239         0              3         11         1

[5 rows x 29 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 29 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   ID                   2240 non-null   int64  
 1   Year_Birth           2240 non-null   int64  
 2   Education            2240 non-null   object 
 3   Marital_Status       2240 non-null   object 
 4   Income               2216 non-null   float64
 5   Kidhome              2240 non-null   int64  
 6   Teenhome             2240 non-null   int64  
 7   Dt_Customer          2240 non-null   object 
 8   Recency              2240 non-null   int64  
 9   MntWines             2240 non-null   int64  
 10  MntFruits            2240 non-null   int64  
 11  MntMeatProducts      2240 non-null   int64  
 12  MntFishProducts      2240 non-null   int64  
 13  MntSweetProducts     2240 non-null   int64  
 14  MntGoldProds         2240 non-null   int64  
 15  NumDealsPurchases    2240 non-null   int64  
 16  NumWebPurchases      2240 non-null   int64  
 17  NumCatalogPurchases  2240 non-null   int64  
 18  NumStorePurchases    2240 non-null   int64  
 19  NumWebVisitsMonth    2240 non-null   int64  
 20  AcceptedCmp3         2240 non-null   int64  
 21  AcceptedCmp4         2240 non-null   int64  
 22  AcceptedCmp5         2240 non-null   int64  
 23  AcceptedCmp1         2240 non-null   int64  
 24  AcceptedCmp2         2240 non-null   int64  
 25  Complain             2240 non-null   int64  
 26  Z_CostContact        2240 non-null   int64  
 27  Z_Revenue            2240 non-null   int64  
 28  Response             2240 non-null   int64  
dtypes: float64(1), int64(25), object(3)
memory usage: 507.6+ KB
None
```

Question 4:

a. Use bracket notation to select the 'Education' Series from the **customers** DataFrame . Show the first 5 rows of this Series.

b. Use dot notation to select the 'Marital_Status' Series from the **customers** DataFrame . Show the first 8 rows of this Series.

In [21]:
```
print(customers['Education'].head())
print(customers.Marital_Status.head(8))
```

```
0    Graduation
1    Graduation
2    Graduation
3    Graduation
4           PhD
Name: Education, dtype: object
0      Single
1      Single
2    Together
3    Together
4     Married
5    Together
6    Divorced
7     Married
Name: Marital_Status, dtype: object
```

Question 5:

a. Create a new 'Age' Series (must use bracket notation to define the Series name) ... use the relationship Age = 2021 - Year_Birth

b. Use an appropriate attribute or method to check that the 'Age' Series was added to the **customers** DataFrame.

In [23]:
```python
Year_Now = 2021
customers['Age'] = int(Year_Now) - customers['Year_Birth']
print(customers.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 30 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   ID                   2240 non-null   int64  
 1   Year_Birth           2240 non-null   int64  
 2   Education            2240 non-null   object 
 3   Marital_Status       2240 non-null   object 
 4   Income               2216 non-null   float64
 5   Kidhome              2240 non-null   int64  
 6   Teenhome             2240 non-null   int64  
 7   Dt_Customer          2240 non-null   object 
 8   Recency              2240 non-null   int64  
 9   MntWines             2240 non-null   int64  
 10  MntFruits            2240 non-null   int64  
 11  MntMeatProducts      2240 non-null   int64  
 12  MntFishProducts      2240 non-null   int64  
 13  MntSweetProducts     2240 non-null   int64  
 14  MntGoldProds         2240 non-null   int64  
 15  NumDealsPurchases    2240 non-null   int64  
 16  NumWebPurchases      2240 non-null   int64  
 17  NumCatalogPurchases  2240 non-null   int64  
 18  NumStorePurchases    2240 non-null   int64  
 19  NumWebVisitsMonth    2240 non-null   int64  
 20  AcceptedCmp3         2240 non-null   int64  
 21  AcceptedCmp4         2240 non-null   int64  
 22  AcceptedCmp5         2240 non-null   int64  
 23  AcceptedCmp1         2240 non-null   int64  
 24  AcceptedCmp2         2240 non-null   int64  
 25  Complain             2240 non-null   int64  
 26  Z_CostContact        2240 non-null   int64  
 27  Z_Revenue            2240 non-null   int64  
 28  Response             2240 non-null   int64  
 29  Age                  2240 non-null   int64  
dtypes: float64(1), int64(26), object(3)
memory usage: 525.1+ KB
None
```

Question 6:

a. Use the appropriate method to calculate summary statistics for the numeric columns in the **customers** DataFrame.

b. Use the appropriate method to calculate summary statistics for the 'object' columns in the **customers** DataFrame.

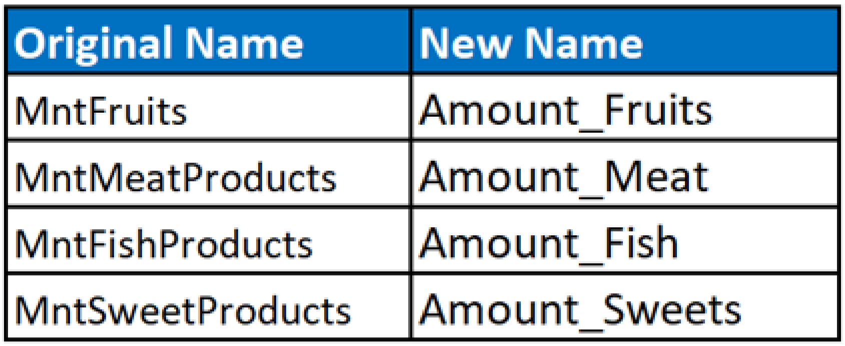
```
In [30]:
print(f"Question A: \n {customers.describe()}")
print("---------------")
print(f"Question B: \n{customers.describe(include=[object])}")
```

```
Question A:
                 ID   Year_Birth         Income      Kidhome      Teenhom
e  \
count   2240.000000  2240.000000    2216.000000  2240.000000  2240.000000
mean    5592.159821  1968.805804   52247.251354     0.444196     0.506250
std     3246.662198    11.984069   25173.076661     0.538398     0.544538
min        0.000000  1893.000000    1730.000000     0.000000     0.000000
25%     2828.250000  1959.000000   35303.000000     0.000000     0.000000
50%     5458.500000  1970.000000   51381.500000     0.000000     0.000000
75%     8427.750000  1977.000000   68522.000000     1.000000     1.000000
max    11191.000000  1996.000000  666666.000000     2.000000     2.000000

           Recency     MntWines    MntFruits  MntMeatProducts  \
count  2240.000000  2240.000000  2240.000000      2240.000000
mean     49.109375   303.935714    26.302232       166.950000
std      28.962453   336.597393    39.773434       225.715373
min       0.000000     0.000000     0.000000         0.000000
25%      24.000000    23.750000     1.000000        16.000000
50%      49.000000   173.500000     8.000000        67.000000
75%      74.000000   504.250000    33.000000       232.000000
max      99.000000  1493.000000   199.000000      1725.000000

       MntFishProducts  ...  AcceptedCmp3  AcceptedCmp4  AcceptedCmp5  \
count      2240.000000  ...   2240.000000   2240.000000   2240.000000
mean         37.525446  ...      0.072768      0.074554      0.072768
std          54.628979  ...      0.259813      0.262728      0.259813
min           0.000000  ...      0.000000      0.000000      0.000000
25%           3.000000  ...      0.000000      0.000000      0.000000
50%          12.000000  ...      0.000000      0.000000      0.000000
75%          50.000000  ...      0.000000      0.000000      0.000000
max         259.000000  ...      1.000000      1.000000      1.000000

       AcceptedCmp1  AcceptedCmp2     Complain  Z_CostContact  Z_Revenue
\
count   2240.000000   2240.000000  2240.000000         2240.0     2240.0
mean       0.064286      0.013393     0.009375            3.0       11.0
std        0.245316      0.114976     0.096391            0.0        0.0
min        0.000000      0.000000     0.000000            3.0       11.0
25%        0.000000      0.000000     0.000000            3.0       11.0
50%        0.000000      0.000000     0.000000            3.0       11.0
75%        0.000000      0.000000     0.000000            3.0       11.0
max        1.000000      1.000000     1.000000            3.0       11.0

          Response          Age
count  2240.000000  2240.000000
mean      0.149107    52.194196
std       0.356274    11.984069
min       0.000000    25.000000
25%       0.000000    44.000000
50%       0.000000    51.000000
75%       0.000000    62.000000
max       1.000000   128.000000
```

```
[8 rows x 27 columns]
---------------
Question B:
         Education Marital_Status Dt_Customer
count         2240           2240        2240
unique           5              8         663
top     Graduation        Married  2012-08-31
freq          1127            864          12
```

Question 7:

Rename four of the columns in the **customers** DataFrame as indicated in the table below.



Original Name	New Name
MntFruits	Amount_Fruits
MntMeatProducts	Amount_Meat
MntFishProducts	Amount_Fish
MntSweetProducts	Amount_Sweets

In [31]:

```python
customers.rename(columns=
                 {
                  'MntFruits':'Amount_Fruits',
                  'MntMeatProducts':'Amount_Meat',
                  'MntFishProducts':'Amount_Fish',
                  'MntSweetProducts':'Amount_Sweets'
                 },inplace=True)
customers.columns
```

Out[31]:

```
Index(['ID', 'Year_Birth', 'Education', 'Marital_Status', 'Income', 'Kidhom
e',
       'Teenhome', 'Dt_Customer', 'Recency', 'MntWines', 'Amount_Fruits',
       'Amount_Meat', 'Amount_Fish', 'Amount_Sweets', 'MntGoldProds',
       'NumDealsPurchases', 'NumWebPurchases', 'NumCatalogPurchases',
       'NumStorePurchases', 'NumWebVisitsMonth', 'AcceptedCmp3',
       'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1', 'AcceptedCmp2',
       'Complain', 'Z_CostContact', 'Z_Revenue', 'Response', 'Age'],
      dtype='object')
```

PART TWO

For Questions 8-11: We will be using the **'houses train.txt' dataset** and the **houses**

dataFrame.

Question 8:

a. Read in the dataset **'house train.txt'** and store the results in a DataFrame named **houses**. See the link below for the dataset.

houses train.txt (https://drive.google.com/file/d/1GCzq0nZt6e4oEhYtWpnGfcAHsMBO4-Oh/view?usp=share_link)

b. Use appropriate attributes and methods to inspect the **houses** DataFrame.

In [47]:
```
houses = pd.read_table('houses train.txt',sep=',')
print(houses.index)
print(houses.columns)
print(houses.shape)
print(houses.dtypes)
```

```
RangeIndex(start=0, stop=1460, step=1)
Index(['Id', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street',
       'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
       'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType',
       'HouseStyle', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAd
d',
       'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType',
       'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual',
       'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1',
       'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating',
       'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF',
       'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBat
h',
       'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
       'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'GarageTyp
e',
       'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQua
l',
       'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'PoolQC',
       'Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleType',
       'SaleCondition', 'SalePrice'],
      dtype='object')
(1460, 81)
Id                 int64
MSSubClass         int64
MSZoning          object
LotFrontage      float64
LotArea            int64
                  ...
MoSold             int64
YrSold             int64
SaleType          object
SaleCondition     object
SalePrice          int64
Length: 81, dtype: object
```

Question 9:

a. Remove the **'MSSubClass' column** from the **houses** DataFrame.

b. Use an appropriate attribute or method to check that the **'MSSubClass' column** was removed from the **houses** DataFrame.

In [39]:
```
del houses['MSSubClass']
houses.columns
```

Out[39]:
```
Index(['Id', 'MSZoning', 'LotFrontage', 'LotArea', 'Street', 'Alley',
       'LotShape', 'LandContour', 'Utilities', 'LotConfig', 'LandSlope',
       'Neighborhood', 'Condition1', 'Condition2', 'BldgType', 'HouseStyle',
       'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd', 'RoofStyl
e',
       'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType', 'MasVnrArea',
       'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual', 'BsmtCond',
       'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1', 'BsmtFinType2',
       'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating', 'HeatingQC',
       'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF',
       'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'HalfBath',
       'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual', 'TotRmsAbvGrd',
       'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType', 'GarageYrBl
t',
       'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual', 'GarageCon
d',
       'PavedDrive', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorc
h',
       'ScreenPorch', 'PoolArea', 'PoolQC', 'Fence', 'MiscFeature', 'MiscVa
l',
       'MoSold', 'YrSold', 'SaleType', 'SaleCondition', 'SalePrice'],
      dtype='object')
```

Question 10:

a. Remove the **'LandContour', 'LandSlope', and 'Functional' columns** from the **houses** DataFrame.

b. Use an appropriate attribute or method to check that the **'LandContour', 'LandSlope', and 'Functional' columns** were removed from the **houses** DataFrame.

In [40]:
```
houses.drop(['LandContour','LandSlope','Functional'],axis=1,inplace=True)
houses.columns
```

Out[40]:
```
Index(['Id', 'MSZoning', 'LotFrontage', 'LotArea', 'Street', 'Alley',
       'LotShape', 'Utilities', 'LotConfig', 'Neighborhood', 'Condition1',
       'Condition2', 'BldgType', 'HouseStyle', 'OverallQual', 'OverallCond',
       'YearBuilt', 'YearRemodAdd', 'RoofStyle', 'RoofMatl', 'Exterior1st',
       'Exterior2nd', 'MasVnrType', 'MasVnrArea', 'ExterQual', 'ExterCond',
       'Foundation', 'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1',
       'BsmtFinSF1', 'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtS
F',
       'Heating', 'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF',
       '2ndFlrSF', 'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBat
h',
       'FullBath', 'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQua
l',
       'TotRmsAbvGrd', 'Fireplaces', 'FireplaceQu', 'GarageType',
       'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQua
l',
       'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'PoolQC',
       'Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleType',
       'SaleCondition', 'SalePrice'],
      dtype='object')
```

Question 11:

a. Remove the first 7 rows from the **houses** DataFrame.

b. Use an appropriate attribute or method to check that the first 7 rows were removed from the **houses** DataFrame.

In [48]:
```python
houses = houses.iloc[7:]
print(houses)
```

```
        Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley LotShape  \
7        8          60       RL          NaN    10382   Pave   NaN      IR1   
8        9          50       RM         51.0     6120   Pave   NaN      Reg   
9       10         190       RL         50.0     7420   Pave   NaN      Reg   
10      11          20       RL         70.0    11200   Pave   NaN      Reg   
11      12          60       RL         85.0    11924   Pave   NaN      IR1   
...    ...         ...      ...          ...      ...    ...   ...      ...   
1455  1456          60       RL         62.0     7917   Pave   NaN      Reg   
1456  1457          20       RL         85.0    13175   Pave   NaN      Reg   
1457  1458          70       RL         66.0     9042   Pave   NaN      Reg   
1458  1459          20       RL         68.0     9717   Pave   NaN      Reg   
1459  1460          20       RL         75.0     9937   Pave   NaN      Reg   

     LandContour Utilities  ... PoolArea PoolQC  Fence MiscFeature MiscVal  \
7            Lvl    AllPub  ...        0    NaN    NaN        Shed     350   
8            Lvl    AllPub  ...        0    NaN    NaN         NaN       0   
9            Lvl    AllPub  ...        0    NaN    NaN         NaN       0   
10           Lvl    AllPub  ...        0    NaN    NaN         NaN       0   
11           Lvl    AllPub  ...        0    NaN    NaN         NaN       0   
...          ...       ...  ...      ...    ...    ...         ...     ...   
1455         Lvl    AllPub  ...        0    NaN    NaN         NaN       0   
1456         Lvl    AllPub  ...        0    NaN  MnPrv         NaN       0   
1457         Lvl    AllPub  ...        0    NaN  GdPrv        Shed    2500   
1458         Lvl    AllPub  ...        0    NaN    NaN         NaN       0   
1459         Lvl    AllPub  ...        0    NaN    NaN         NaN       0   

     MoSold YrSold  SaleType  SaleCondition  SalePrice  
7        11   2009        WD         Normal     200000  
8         4   2008        WD        Abnorml     129900  
9         1   2008        WD         Normal     118000  
10        2   2008        WD         Normal     129500  
11        7   2006       New        Partial     345000  
...     ...    ...       ...            ...        ...  
1455      8   2007        WD         Normal     175000  
1456      2   2010        WD         Normal     210000  
1457      5   2010        WD         Normal     266500  
1458      4   2010        WD         Normal     142125  
1459      6   2008        WD         Normal     147500  

[1453 rows x 81 columns]
```