Asg 19.5

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

(Coding)

This assignment focuses on the material found in ...

- Presentation 7: Working with Pandas DataFrames -- The Essentials (Part Three)

- Lessons 12, 15, 19, and 20 of the video series found below.

Python pandas Q&A video series by Data School

YouTube playlist and GitHub repository

Table of contents

1. What is pandas?
2. How do I read a tabular data file into pandas?
3. How do I select a pandas Series from a DataFrame?
4. Why do some pandas commands end with parentheses (and others don't)?
5. How do I rename columns in a pandas DataFrame?

6. How do I remove columns from a pandas DataFrame?
7. How do I sort a pandas DataFrame or a Series?
8. How do I filter rows of a pandas DataFrame by column value?
9. How do I apply multiple filter criteria to a pandas DataFrame?
10. Your pandas questions answered!
11. How do I use the "axis" parameter in pandas?
12. How do I use string methods in pandas?
13. How do I change the data type of a pandas Series?
14. When should I use a "groupby" in pandas?
15. How do I explore a pandas Series?
16. How do I handle missing values in pandas?
17. What do I need to know about the pandas index? (Part 1)
18. What do I need to know about the pandas index? (Part 2)
19. How do I select multiple rows and columns from a pandas DataFrame?
20. When should I use the "inplace" parameter in pandas?
21. How do I make my pandas DataFrame smaller and faster?
22. How do I use pandas with scikit-learn to create Kaggle submissions?
23. More of your pandas questions answered!
24. How do I create dummy variables in pandas?
25. How do I work with dates and times in pandas?
26. How do I find and remove duplicate rows in pandas?
27. How do I avoid a SettingWithCopyWarning in pandas?
28. How do I change display options in pandas?
29. How do I create a pandas DataFrame from another object?
30. How do I apply a function to a pandas Series or DataFrame?

Files needed for this assignment:

marketing_campaign.csv
menu.csv

In [76]:
```
# 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 [77]:
```
# conventional way to import pandas
import pandas as pd
```

PART ONE

<div class="alert alert-block alert-info"

For Questions 1-9: We will be using the **'marketing_campaign.csv' dataset** and the **customers DataFrame**

Question 1:

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

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

In [78]:
```
customers = pd.read_csv('marketing_campaign.csv',sep=";")
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', 'Kidhome',
       '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', 'Response'],
      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 2:

Use the **describe method** to get a count of values, the number of unique values, the most common value, and the frequency of the most common value in the **Marital_Status** Series.

Documentation for **describe**

In [79]:
```
customers.Marital_Status.describe()
```

Out[79]:
```
count        2240
unique          8
top       Married
freq          864
Name: Marital_Status, dtype: object
```

Question 3:

Use the **value_counts method** to count how many times each unique value in the **Marital_Status** Series occurs.

Documentation for **value_counts**

In [80]:
```
customers.Marital_Status.value_counts()
```

Out[80]:
```
Married     864
Together    580
Single      480
Divorced    232
Widow        77
Alone         3
Absurd        2
YOLO          2
Name: Marital_Status, dtype: int64
```

Question 4:

Use the **normalize parameter** of the **value_counts method** to display percentages instead of raw counts of how many times each unique value in the **Marital_Status** Series occurs.

Documentation for **value_counts**

In [81]:
```
customers.Marital_Status.value_counts(normalize=True)
```

Out[81]:
```
Married     0.385714
Together    0.258929
Single      0.214286
Divorced    0.103571
Widow       0.034375
Alone       0.001339
Absurd      0.000893
YOLO        0.000893
Name: Marital_Status, dtype: float64
```

Question 5:

Use the **unique method** to display the unique values in the **Marital_Status** Series.

Documentation for **unique**

In [82]:
```
customers.Marital_Status.unique()
```

Out[82]:
```
array(['Single', 'Together', 'Married', 'Divorced', 'Widow', 'Alone',
       'Absurd', 'YOLO'], dtype=object)
```

Question 6:

Use the **nunique method** to count the number of unique values in the Series in the **Marital_Status** Series.

Documentation for **nunique**

In [83]:
```
uniquearr = customers.Marital_Status.nunique()
uniquearr
```

Out[83]:
```
8
```

Question 7:

Use the Pandas **crosstab function** to compute a cross-tabulation of the two Series, **Education** and **Marital_Status** Series.

Documentation for **crosstab**

In [84]:
```
pd.crosstab(customers.Education,customers.Marital_Status)
```

Out[84]:

Marital_Status	Absurd	Alone	Divorced	Married	Single	Together	Widow	YOLO
Education								
2n Cycle	0	0	23	81	37	57	5	0
Basic	0	0	1	20	18	14	1	0
Graduation	1	1	119	433	252	286	35	0
Master	1	1	37	138	75	106	12	0
PhD	0	1	52	192	98	117	24	2

Question 8:

Use the **describe method** to calculate a statistical summary for the **Income** Series.

Documentation for **describe**

In [85]:
```
customers.Income.describe()
```

Out[85]:
```
count      2216.000000
mean      52247.251354
std       25173.076661
min        1730.000000
25%       35303.000000
50%       51381.500000
75%       68522.000000
max      666666.000000
Name: Income, dtype: float64
```

Question 9:

Use the **value_counts method** to count how many times each unique value in the **Income** Series occurs.

Documentation for **value_counts**

Note: The **value_counts** method is more useful for categorical data than it is for numerical data.

In [86]:
```
customers.Income.value_counts()
```

Out[86]:
```
7500.0     12
35860.0     4
37760.0     3
83844.0     3
63841.0     3
           ..
40760.0     1
41452.0     1
6835.0      1
33622.0     1
52869.0     1
Name: Income, Length: 1974, dtype: int64
```

Run the next code cell to allow plots to appear in the notebook

In [87]:
```
# allow plots to appear in the notebook
%matplotlib inline
```

Question 10:

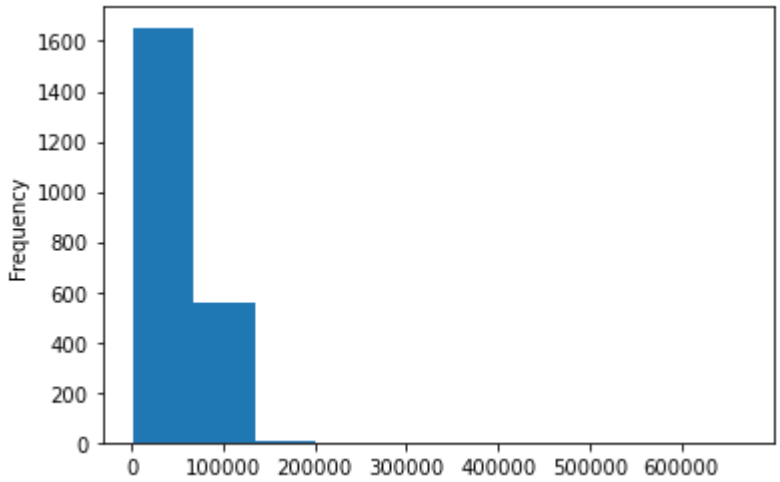
Make a histogram of the **Income** Series.

Documentation for **plot**

Note: Histograms are used to show the distribution of a numerical variable.

In [88]:
```
customers.Income.plot(kind='hist')
```

Out[88]:
```
<AxesSubplot:ylabel='Frequency'>
```



Question 11:

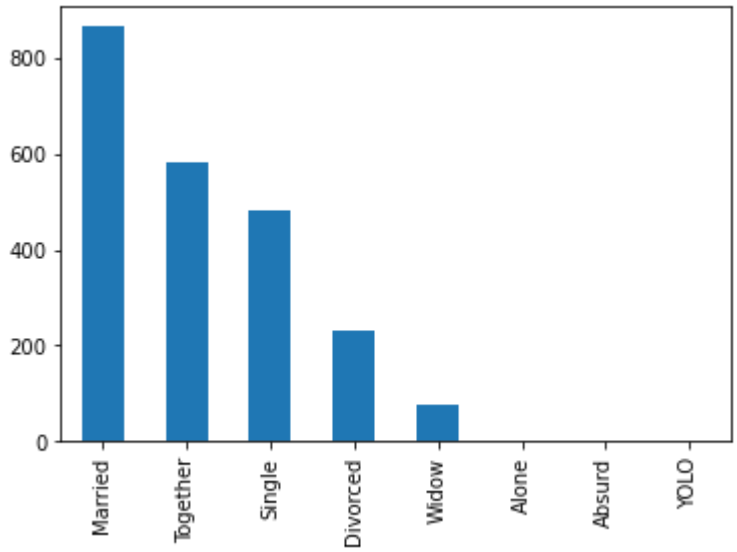
Make a bar graph of the **Marital_Status** Series.

Documentation for **plot**

Note: Bar graphs are used to show the distribution of a categorical variable.

In [89]:
```
customers.Marital_Status.value_counts().plot(kind='bar')
```

Out[89]:
```
<AxesSubplot:>
```



PART TWO

<div class="alert alert-block alert-info"

For Questions 12-18: We will be using the **'menu.csv' dataset** and the **menu DataFrame**

Question 12:

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

See the links below for access to the dataset and for information about the dataset.

Nutrition Facts for McDonald's Menu

b. Use appropriate methods and attributes to inspect the **customers** DataFrame. Consider the following options.

- head()
- tail()
- info()
- index
- columns
- shape
- dtypes

In [90]:
```
menu = pd.read_csv('menu.csv')
print(menu.head())
print(menu.tail())
print(menu.info())
print(menu.index)
print(menu.columns)
print(menu.shape)
print(menu.dtypes)
```

```
    Category                            Item    Serving Size  Calori
es  \
0  Breakfast                      Egg McMuffin  4.8 oz (136 g)       3
00
1  Breakfast                 Egg White Delight  4.8 oz (135 g)       2
50
2  Breakfast                  Sausage McMuffin  3.9 oz (111 g)       3
70
3  Breakfast         Sausage McMuffin with Egg  5.7 oz (161 g)       4
50
4  Breakfast  Sausage McMuffin with Egg Whites  5.7 oz (161 g)       4
00

   Calories from Fat  Total Fat  Total Fat (% Daily Value)  Saturated
Fat  \
0                120       13.0                          20
5.0
1                 70        8.0                          12
3.0
2                200       23.0                          35
8.0
3                250       28.0                          43           1
0.0
4                210       23.0                          35
8.0

   Saturated Fat (% Daily Value)  Trans Fat  ...  Carbohydrates  \
0                             25        0.0  ...             31
1                             15        0.0  ...             30
2                             42        0.0  ...             29
3                             52        0.0  ...             30
4                             42        0.0  ...             30

   Carbohydrates (% Daily Value)  Dietary Fiber  \
0                             10              4
1                             10              4
2                             10              4
3                             10              4
4                             10              4

   Dietary Fiber (% Daily Value)  Sugars  Protein  Vitamin A (% Daily
Value)  \
0                             17       3       17
10
1                             17       3       18
6
2                             17       2       14
8
3                             17       2       21
15
4                             17       2       21
6

   Vitamin C (% Daily Value)  Calcium (% Daily Value)  Iron (% Daily V
alue)
0                          0                       25
15
1                          0                       25
8
2                          0                       25
```

```
10
3                          0                        30
15
4                          0                        25
10

[5 rows x 24 columns]
              Category
Item  \
255  Smoothies & Shakes                 McFlurry with Oreo Cookies (Sm
all)
256  Smoothies & Shakes                McFlurry with Oreo Cookies (Med
ium)
257  Smoothies & Shakes                 McFlurry with Oreo Cookies (Sn
ack)
258  Smoothies & Shakes  McFlurry with Reese's Peanut Butter Cups (Med
ium)
259  Smoothies & Shakes   McFlurry with Reese's Peanut Butter Cups (Sn
ack)

        Serving Size  Calories  Calories from Fat  Total Fat  \
255  10.1 oz (285 g)       510                150       17.0
256  13.4 oz (381 g)       690                200       23.0
257   6.7 oz (190 g)       340                100       11.0
258  14.2 oz (403 g)       810                290       32.0
259   7.1 oz (202 g)       410                150       16.0

     Total Fat (% Daily Value)  Saturated Fat  Saturated Fat (% Daily
Value)  \
255                         26            9.0
44
256                         35           12.0
58
257                         17            6.0
29
258                         50           15.0
76
259                         25            8.0
38

     Trans Fat  ...  Carbohydrates  Carbohydrates (% Daily Value)  \
255        0.5  ...            80                              27
256        1.0  ...           106                              35
257        0.0  ...            53                              18
258        1.0  ...           114                              38
259        0.0  ...            57                              19

     Dietary Fiber  Dietary Fiber (% Daily Value)  Sugars  Protein  \
255              1                              4      64       12
256              1                              5      85       15
257              1                              2      43        8
258              2                              9     103       21
259              1                              5      51       10

     Vitamin A (% Daily Value)  Vitamin C (% Daily Value)  \
255                         15                          0
256                         20                          0
257                         10                          0
258                         20                          0
259                         10                          0
```

```
     Calcium (% Daily Value)  Iron (% Daily Value)
255                       40                     8
256                       50                    10
257                       25                     6
258                       60                     6
259                       30                     4

[5 rows x 24 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 260 entries, 0 to 259
Data columns (total 24 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Category                        260 non-null    object
 1   Item                            260 non-null    object
 2   Serving Size                    260 non-null    object
 3   Calories                        260 non-null    int64
 4   Calories from Fat               260 non-null    int64
 5   Total Fat                       260 non-null    float64
 6   Total Fat (% Daily Value)       260 non-null    int64
 7   Saturated Fat                   260 non-null    float64
 8   Saturated Fat (% Daily Value)   260 non-null    int64
 9   Trans Fat                       260 non-null    float64
 10  Cholesterol                     260 non-null    int64
 11  Cholesterol (% Daily Value)     260 non-null    int64
 12  Sodium                          260 non-null    int64
 13  Sodium (% Daily Value)          260 non-null    int64
 14  Carbohydrates                   260 non-null    int64
 15  Carbohydrates (% Daily Value)   260 non-null    int64
 16  Dietary Fiber                   260 non-null    int64
 17  Dietary Fiber (% Daily Value)   260 non-null    int64
 18  Sugars                          260 non-null    int64
 19  Protein                         260 non-null    int64
 20  Vitamin A (% Daily Value)       260 non-null    int64
 21  Vitamin C (% Daily Value)       260 non-null    int64
 22  Calcium (% Daily Value)         260 non-null    int64
 23  Iron (% Daily Value)            260 non-null    int64
dtypes: float64(3), int64(18), object(3)
memory usage: 48.9+ KB
None
RangeIndex(start=0, stop=260, step=1)
Index(['Category', 'Item', 'Serving Size', 'Calories', 'Calories from
Fat',
       'Total Fat', 'Total Fat (% Daily Value)', 'Saturated Fat',
       'Saturated Fat (% Daily Value)', 'Trans Fat', 'Cholesterol',
       'Cholesterol (% Daily Value)', 'Sodium', 'Sodium (% Daily Valu
e)',
       'Carbohydrates', 'Carbohydrates (% Daily Value)', 'Dietary Fibe
r',
       'Dietary Fiber (% Daily Value)', 'Sugars', 'Protein',
       'Vitamin A (% Daily Value)', 'Vitamin C (% Daily Value)',
       'Calcium (% Daily Value)', 'Iron (% Daily Value)'],
      dtype='object')
(260, 24)
Category                         object
Item                             object
Serving Size                     object
Calories                          int64
Calories from Fat                 int64
```

```
Total Fat                          float64
Total Fat (% Daily Value)            int64
Saturated Fat                      float64
Saturated Fat (% Daily Value)        int64
Trans Fat                          float64
Cholesterol                          int64
Cholesterol (% Daily Value)          int64
Sodium                               int64
Sodium (% Daily Value)               int64
Carbohydrates                        int64
Carbohydrates (% Daily Value)        int64
Dietary Fiber                        int64
Dietary Fiber (% Daily Value)        int64
Sugars                               int64
Protein                              int64
Vitamin A (% Daily Value)            int64
Vitamin C (% Daily Value)            int64
Calcium (% Daily Value)              int64
Iron (% Daily Value)                 int64
dtype: object
```

Question 13:

Use the **str.replace() method** to replace all spaces with underscores in the column names of the **menu** DataFrame.

In [91]:
```
menu.columns=menu.columns.str.replace(' ','_')
menu
```

Out[91]:

	Category	Item	Serving_Size	Calories	Calories_from_Fat	Total_Fat	T
0	Breakfast	Egg McMuffin	4.8 oz (136 g)	300	120	13.0	
1	Breakfast	Egg White Delight	4.8 oz (135 g)	250	70	8.0	
2	Breakfast	Sausage McMuffin	3.9 oz (111 g)	370	200	23.0	
3	Breakfast	Sausage McMuffin with Egg	5.7 oz (161 g)	450	250	28.0	
4	Breakfast	Sausage McMuffin with Egg Whites	5.7 oz (161 g)	400	210	23.0	
...	...	...	...	...	...	...	
255	Smoothies & Shakes	McFlurry with Oreo Cookies (Small)	10.1 oz (285 g)	510	150	17.0	
256	Smoothies & Shakes	McFlurry with Oreo Cookies (Medium)	13.4 oz (381 g)	690	200	23.0	
257	Smoothies & Shakes	McFlurry with Oreo Cookies (Snack)	6.7 oz (190 g)	340	100	11.0	
258	Smoothies & Shakes	McFlurry with Reese's Peanut Butter Cups (Medium)	14.2 oz (403 g)	810	290	32.0	
259	Smoothies & Shakes	McFlurry with Reese's Peanut Butter Cups (Snack)	7.1 oz (202 g)	410	150	16.0	

260 rows × 24 columns

Use for Questions 14-19:

Documentation for **loc**

Question 14:

Use the **loc function** to select row 0 and all columns of the **menu** DataFrame.

In [92]:
```
menu.loc[0,:]
```

Out[92]:
```
Category                                Breakfast
Item                                Egg McMuffin
Serving_Size                     4.8 oz (136 g)
Calories                                     300
Calories_from_Fat                            120
Total_Fat                                   13.0
Total_Fat_(%_Daily_Value)                     20
Saturated_Fat                                5.0
Saturated_Fat_(%_Daily_Value)                 25
Trans_Fat                                    0.0
Cholesterol                                  260
Cholesterol_(%_Daily_Value)                   87
Sodium                                       750
Sodium_(%_Daily_Value)                        31
Carbohydrates                                 31
Carbohydrates_(%_Daily_Value)                 10
Dietary_Fiber                                  4
Dietary_Fiber_(%_Daily_Value)                 17
Sugars                                         3
Protein                                       17
Vitamin_A_(%_Daily_Value)                     10
Vitamin_C_(%_Daily_Value)                      0
Calcium_(%_Daily_Value)                       25
Iron_(%_Daily_Value)                          15
Name: 0, dtype: object
```

Question 15:

Use the **loc function** to select rows 0 through 4 (inclusive) and all columns of the **menu** DataFrame.

In [93]:
```
menu.loc[0:5,:]
```

Out[93]:

	Category	Item	Serving_Size	Calories	Calories_from_Fat	Total_Fat	T
0	Breakfast	Egg McMuffin	4.8 oz (136 g)	300	120	13.0	
1	Breakfast	Egg White Delight	4.8 oz (135 g)	250	70	8.0	
2	Breakfast	Sausage McMuffin	3.9 oz (111 g)	370	200	23.0	
3	Breakfast	Sausage McMuffin with Egg	5.7 oz (161 g)	450	250	28.0	
4	Breakfast	Sausage McMuffin with Egg Whites	5.7 oz (161 g)	400	210	23.0	
5	Breakfast	Steak & Egg McMuffin	6.5 oz (185 g)	430	210	23.0	

6 rows × 24 columns

Question 16:

Use the **loc function** to select rows 0 through 4 (inclusive) for the **'Item'** and **'Calories"** columns of the **menu** DataFrame.

In [94]:
```
menu.loc[0:5,['Item','Calories']]
```

Out[94]:

	Item	Calories
0	Egg McMuffin	300
1	Egg White Delight	250
2	Sausage McMuffin	370
3	Sausage McMuffin with Egg	450
4	Sausage McMuffin with Egg Whites	400
5	Steak & Egg McMuffin	430

Question 17:

Use the **loc function** to select rows 0 through 4 (inclusive) for the columns **'Item'** through **'Total_Fat"** (inclusive) from the **menu** DataFrame.

In [95]:
```
menu.loc[0:5,['Item','Total_Fat']]
```

Out[95]:

	Item	Total_Fat
0	Egg McMuffin	13.0
1	Egg White Delight	8.0
2	Sausage McMuffin	23.0
3	Sausage McMuffin with Egg	28.0
4	Sausage McMuffin with Egg Whites	23.0
5	Steak & Egg McMuffin	23.0

Question 18:

Redo Question 17, but without using the **loc method**. Accomplish the same thing by using the **head** and **drop** methods.

In [96]:
```
all_cols = list(menu.columns)
all_cols.remove('Item')
all_cols.remove('Total_Fat')
menu.head(6).drop(columns=all_cols)
```

Out[96]:

	Item	Total_Fat
0	Egg McMuffin	13.0
1	Egg White Delight	8.0
2	Sausage McMuffin	23.0
3	Sausage McMuffin with Egg	28.0
4	Sausage McMuffin with Egg Whites	23.0
5	Steak & Egg McMuffin	23.0

Question 19:

Use the **loc function** to select the rows in which the **'Category'** is **'Desserts'** and the column is **'Calories'**.

In [97]:
```
menu.loc[menu['Category']=='Desserts','Category']
```

Out[97]:
```
103    Desserts
104    Desserts
105    Desserts
106    Desserts
107    Desserts
108    Desserts
109    Desserts
Name: Category, dtype: object
```

Use for Questions 20-22:

Documentation for **iloc**

Question 20:

Use the **iloc function** to select the rows in positions 0, 2, and 5 and the columns in positions 2 and 4.

In [98]:
```
menu.iloc[[0,2,5],[2,4]]
```

Out[98]:

	Serving_Size	Calories_from_Fat
0	4.8 oz (136 g)	120
2	3.9 oz (111 g)	200
5	6.5 oz (185 g)	210

Question 21:

Use the **iloc function** to select the rows in positions 2 through 5 (exclusive) and the columns in positions 0 through 3 (exclusive).

In [99]:
```
menu.iloc[2:5,0:3]
```

Out[99]:

	Category	Item	Serving_Size
2	Breakfast	Sausage McMuffin	3.9 oz (111 g)
3	Breakfast	Sausage McMuffin with Egg	5.7 oz (161 g)
4	Breakfast	Sausage McMuffin with Egg Whites	5.7 oz (161 g)

Question 22:

Use the **iloc function** to select the rows in positions 4 through 7 (exclusive) and all of the columns.

In [100]:
```
menu.iloc[4:7,:]
```

Out[100]:

	Category	Item	Serving_Size	Calories	Calories_from_Fat
4	Breakfast	Sausage McMuffin with Egg Whites	5.7 oz (161 g)	400	210
5	Breakfast	Steak & Egg McMuffin	6.5 oz (185 g)	430	210
6	Breakfast	Bacon, Egg & Cheese Biscuit (Regular Biscuit)	5.3 oz (150 g)	460	230

3 rows × 24 columns

Question 23:

Use the **drop method** to remove the **'Serving_Size'** column without affecting the **menu DataFrame**.

In [101]:
```
menu.drop(['Serving_Size'],axis=1)
```

Out[101]:

	Category	Item	Calories	Calories_from_Fat	Total_Fat
0	Breakfast	Egg McMuffin	300	120	13.0
1	Breakfast	Egg White Delight	250	70	8.0
2	Breakfast	Sausage McMuffin	370	200	23.0
3	Breakfast	Sausage McMuffin with Egg	450	250	28.0
4	Breakfast	Sausage McMuffin with Egg Whites	400	210	23.0
...	...	...	...	...	...
255	Smoothies & Shakes	McFlurry with Oreo Cookies (Small)	510	150	17.0
256	Smoothies & Shakes	McFlurry with Oreo Cookies (Medium)	690	200	23.0
257	Smoothies & Shakes	McFlurry with Oreo Cookies (Snack)	340	100	11.0
258	Smoothies & Shakes	McFlurry with Reese's Peanut Butter Cups (Medium)	810	290	32.0
259	Smoothies & Shakes	McFlurry with Reese's Peanut Butter Cups (Snack)	410	150	16.0

260 rows × 23 columns

Question 24:

Use the **head method** to confirm that the **'Serving_Size'** column was not actually removed from

the **menu DataFrame**.

In [102]:
```
menu.head()
```

Out[102]:

	Category	Item	Serving_Size	Calories	Calories_from_F
0	Breakfast	Egg McMuffin	4.8 oz (136 g)	300	1
1	Breakfast	Egg White Delight	4.8 oz (135 g)	250	
2	Breakfast	Sausage McMuffin	3.9 oz (111 g)	370	2
3	Breakfast	Sausage McMuffin with Egg	5.7 oz (161 g)	450	2
4	Breakfast	Sausage McMuffin with Egg Whites	5.7 oz (161 g)	400	2

5 rows × 24 columns

Question 25:

Use the **drop method** to remove the **'Serving_Size'** column and have it affect the **menu DataFrame**.

In [103]:
```
menu.drop(['Serving_Size'],axis=1,inplace=True)
```

Question 26:

Use the **head method** to confirm that the **'Serving_Size'** column was actually removed from the **menu DataFrame**.

In [104]:
```
menu.head()
```

Out[104]:

	Category	Item	Calories	Calories_from_Fat	Total_Fa
0	Breakfast	Egg McMuffin	300	120	13.
1	Breakfast	Egg White Delight	250	70	8.
2	Breakfast	Sausage McMuffin	370	200	23.
3	Breakfast	Sausage McMuffin with Egg	450	250	28.
4	Breakfast	Sausage McMuffin with Egg Whites	400	210	23.

5 rows × 23 columns