Asg 19.8

Working with Text Data in Pandas

(Coding)

Files needed for this assignment:

menus.csv
food_prices.txt

In [5]:
```
# 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 [3]:
```
# conventional way to import pandas and numpy

import pandas as pd
import numpy as np
```

PART ONE

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

For Questions 1-8: We will be using the **'menu.csv' dataset** and the **menu DataFrame**

Question 1:

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 [4]:
```
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  Calories  \
0  Breakfast                     Egg McMuffin  4.8 oz (136 g)       300
1  Breakfast                 Egg White Delight  4.8 oz (135 g)       250
2  Breakfast                  Sausage McMuffin  3.9 oz (111 g)       370
3  Breakfast         Sausage McMuffin with Egg  5.7 oz (161 g)       450
4  Breakfast  Sausage McMuffin with Egg Whites  5.7 oz (161 g)       400

   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           10.0
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 Value)
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 (Small)
256  Smoothies & Shakes              McFlurry with Oreo Cookies (Medium)
257  Smoothies & Shakes               McFlurry with Oreo Cookies (Snack)
258  Smoothies & Shakes  McFlurry with Reese's Peanut Butter Cups (Medium)
259  Smoothies & Shakes   McFlurry with Reese's Peanut Butter Cups (Snack)

       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 Value)',
       'Carbohydrates', 'Carbohydrates (% Daily Value)', 'Dietary Fiber',
       '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 2:

Change all of the elements in the **Item** Series of the **Menu** DataFrame so that they are represented entirely in upper case letters.

In [5]:
```
menu.Item.str.upper()
```

Out[5]:
```
0                                        EGG MCMUFFIN
1                                   EGG WHITE DELIGHT
2                                    SAUSAGE MCMUFFIN
3                              SAUSAGE MCMUFFIN WITH EGG
4                         SAUSAGE MCMUFFIN WITH EGG WHITES
                                  ...
255                     MCFLURRY WITH OREO COOKIES (SMALL)
256                    MCFLURRY WITH OREO COOKIES (MEDIUM)
257                     MCFLURRY WITH OREO COOKIES (SNACK)
258    MCFLURRY WITH REESE'S PEANUT BUTTER CUPS (MEDIUM)
259     MCFLURRY WITH REESE'S PEANUT BUTTER CUPS (SNACK)
Name: Item, Length: 260, dtype: object
```

Question 3:

Use the string method **contains()** to check the elements in the **Item** Series for the substring 'Sausage'. You should return a boolean Series where a value of True indicates that the item does contain the substring 'Sausage.

In [6]:
```
menu.Item.str.contains('Sausage')
```

Out[6]:
```
0      False
1      False
2       True
3       True
4       True
       ...
255    False
256    False
257    False
258    False
259    False
Name: Item, Length: 260, dtype: bool
```

Question 4:

Filter the **menu** DataFrame by using the Boolean series that you created in Question 3 above ... i.e. return all rows that have items that contain the string "sausage' in the name of the item.

In [7]:
```
menu[menu.Item.str.contains('Sausage')]
```

Out[7]:

	Category	Item	Serving Size	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat
2	Breakfast	Sausage McMuffin	3.9 oz (111 g)	370	200	23.0	35	8.0	42	0.0
3	Breakfast	Sausage McMuffin with Egg	5.7 oz (161 g)	450	250	28.0	43	10.0	52	0.0
4	Breakfast	Sausage McMuffin with Egg Whites	5.7 oz (161 g)	400	210	23.0	35	8.0	42	0.0
10	Breakfast	Sausage Biscuit (Regular Biscuit)	4.1 oz (117 g)	430	240	27.0	42	12.0	62	0.0
11	Breakfast	Sausage Biscuit (Large Biscuit)	4.6 oz (131 g)	480	280	31.0	48	13.0	65	0.0
12	Breakfast	Sausage Biscuit with Egg (Regular Biscuit)	5.7 oz (163 g)	510	290	33.0	50	14.0	71	0.0
13	Breakfast	Sausage Biscuit with Egg (Large Biscuit)	6.2 oz (177 g)	570	330	37.0	57	15.0	74	0.0
14	Breakfast	Sausage Biscuit with Egg Whites (Regular Biscuit)	5.9 oz (167 g)	460	250	27.0	42	12.0	62	0.0
15	Breakfast	Sausage Biscuit with Egg Whites (Large Biscuit)	6.4 oz (181 g)	520	280	32.0	49	13.0	65	0.0
21	Breakfast	Sausage McGriddles	5 oz (141 g)	420	200	22.0	34	8.0	40	0.0
22	Breakfast	Sausage, Egg & Cheese McGriddles	7.1 oz (201 g)	550	280	31.0	48	12.0	61	0.0

	Category	Item	Serving Size	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat
23	Breakfast	Sausage, Egg & Cheese McGriddles with Egg Whites	7.2 oz (205 g)	500	230	26.0	40	10.0	52	0.0
36	Breakfast	Hotcakes and Sausage	6.8 oz (192 g)	520	210	24.0	37	7.0	36	0.0
37	Breakfast	Sausage Burrito	3.9 oz (111 g)	300	150	16.0	25	7.0	33	0.0

14 rows × 24 columns

Question 5:

Change every item in the **Serving Size** Series as indicated in the following example:

4.8 oz (136 g) gets changed to 4.8 oz or 136 g

Note: First you will need to change the name of the **Serving Size** Series to **Serving_Size**. Use the **rename()** method to do this.

In [12]:
```
menu = menu.rename(columns={'Serving Size': 'Serving_Size'})
menu['Serving_Size'] = '4.8 oz (136 g)'
menu
```

Out[12]:

	Category	Item	Serving_Size	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value
0	Breakfast	Egg McMuffin	4.8 oz (136 g)	300	120	13.0	20	5.0	25
1	Breakfast	Egg White Delight	4.8 oz (136 g)	250	70	8.0	12	3.0	15
2	Breakfast	Sausage McMuffin	4.8 oz (136 g)	370	200	23.0	35	8.0	42
3	Breakfast	Sausage McMuffin with Egg	4.8 oz (136 g)	450	250	28.0	43	10.0	52
4	Breakfast	Sausage McMuffin with Egg Whites	4.8 oz (136 g)	400	210	23.0	35	8.0	42
...	...	...	...	...	...	...	...	...	...
255	Smoothies & Shakes	McFlurry with Oreo Cookies (Small)	4.8 oz (136 g)	510	150	17.0	26	9.0	44
256	Smoothies & Shakes	McFlurry with Oreo Cookies (Medium)	4.8 oz (136 g)	690	200	23.0	35	12.0	58
257	Smoothies & Shakes	McFlurry with Oreo Cookies (Snack)	4.8 oz (136 g)	340	100	11.0	17	6.0	29
258	Smoothies & Shakes	McFlurry with Reese's Peanut Butter Cups (Medium)	4.8 oz (136 g)	810	290	32.0	50	15.0	76
259	Smoothies & Shakes	McFlurry with Reese's Peanut Butter Cups (Snack)	4.8 oz (136 g)	410	150	16.0	25	8.0	38

260 rows × 24 columns

Question 6:

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

In [13]:
```
menu.columns = menu.columns.str.replace(' ', '_')
menu
```

Out[13]:

	Category	Item	Serving_Size	Calories	Calories_from_Fat	Total_Fat	Total_Fat_(%_D
0	Breakfast	Egg McMuffin	4.8 oz (136 g)	300	120	13.0	
1	Breakfast	Egg White Delight	4.8 oz (136 g)	250	70	8.0	
2	Breakfast	Sausage McMuffin	4.8 oz (136 g)	370	200	23.0	
3	Breakfast	Sausage McMuffin with Egg	4.8 oz (136 g)	450	250	28.0	
4	Breakfast	Sausage McMuffin with Egg Whites	4.8 oz (136 g)	400	210	23.0	
...	...	...	...	...	...	...	
255	Smoothies & Shakes	McFlurry with Oreo Cookies (Small)	4.8 oz (136 g)	510	150	17.0	
256	Smoothies & Shakes	McFlurry with Oreo Cookies (Medium)	4.8 oz (136 g)	690	200	23.0	
257	Smoothies & Shakes	McFlurry with Oreo Cookies (Snack)	4.8 oz (136 g)	340	100	11.0	
258	Smoothies & Shakes	McFlurry with Reese's Peanut Butter Cups (Medium)	4.8 oz (136 g)	810	290	32.0	
259	Smoothies & Shakes	McFlurry with Reese's Peanut Butter Cups (Snack)	4.8 oz (136 g)	410	150	16.0	

260 rows × 24 columns

Question 7:

Change the data type of the **Calories from Fat** Series of the **menu** DataFrame from an integer to a float.

In [14]:
```
menu['Calories_from_Fat'] = menu['Calories_from_Fat'].astype(float)
menu
```

Out[14]:

	Category	Item	Serving_Size	Calories	Calories_from_Fat	Total_Fat	Total_Fat_(%
0	Breakfast	Egg McMuffin	4.8 oz (136 g)	300	120.0	13.0	
1	Breakfast	Egg White Delight	4.8 oz (136 g)	250	70.0	8.0	
2	Breakfast	Sausage McMuffin	4.8 oz (136 g)	370	200.0	23.0	
3	Breakfast	Sausage McMuffin with Egg	4.8 oz (136 g)	450	250.0	28.0	
4	Breakfast	Sausage McMuffin with Egg Whites	4.8 oz (136 g)	400	210.0	23.0	
...	...	...	...	...	...	...	
255	Smoothies & Shakes	McFlurry with Oreo Cookies (Small)	4.8 oz (136 g)	510	150.0	17.0	
256	Smoothies & Shakes	McFlurry with Oreo Cookies (Medium)	4.8 oz (136 g)	690	200.0	23.0	
257	Smoothies & Shakes	McFlurry with Oreo Cookies (Snack)	4.8 oz (136 g)	340	100.0	11.0	
258	Smoothies & Shakes	McFlurry with Reese's Peanut Butter Cups (Medium)	4.8 oz (136 g)	810	290.0	32.0	
259	Smoothies & Shakes	McFlurry with Reese's Peanut Butter Cups (Snack)	4.8 oz (136 g)	410	150.0	16.0	

260 rows × 24 columns

Question 8:

a. Use the string method **contains()** to check the elements in the **Item** Series for the substring 'Egg'. You should return a boolean Series where a value of True

indicates that the item does contain the substring 'Egg'.

b. Convert the boolean Series from Part(a) to integers (False = 0, True = 1).

In [27]:
```
print(menu.Item.str.contains('Egg'))
menu.Item.str.contains('Egg').astype(int)
```

```
0       True
1       True
2      False
3       True
4       True
       ...
255    False
256    False
257    False
258    False
259    False
Name: Item, Length: 260, dtype: bool
```

Out[27]:
```
0      1
1      1
2      0
3      1
4      1
      ..
255    0
256    0
257    0
258    0
259    0
Name: Item, Length: 260, dtype: int32
```

PART TWO

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

For Questions 9-13: We will be using the **'food_prices.txt' dataset** and the **prices DataFrame**

Question 9:

a. Read in the dataset **'food_prices.txt'** and store the results in a DataFrame named **prices**.

See the links below for access to the dataset.

food_prices.txt

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

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

In [36]:
```
prices = pd.read_table('food_prices.txt', sep=',')
print(prices.head())
print(prices.tail())
print(prices.info())
print(prices.index)
print(prices.columns)
print(prices.shape)
print(prices.dtypes)
```

```
   Food ID   Food Item  Price  Quantity Sold
0        1       Sushi  $3.99             35
1        2     Burrito  $9.99             27
2        3        Taco  2.$99             85
3        4  Quesadilla  $4.25             76
4        5       Pizza  $2.49             54
   Food ID Food Item   Price  Quantity Sold
5        6     Pasta  $13.99             32
6        7     Steak  $24.99             29
7        8     Salad  $11.25             36
8        9     Donut   $0.99             24
9       10     Drink   $1.75             96
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Food ID        10 non-null     int64
 1   Food Item      10 non-null     object
 2   Price          10 non-null     object
 3   Quantity Sold  10 non-null     int64
dtypes: int64(2), object(2)
memory usage: 448.0+ bytes
None
RangeIndex(start=0, stop=10, step=1)
Index(['Food ID', 'Food Item', 'Price', 'Quantity Sold'], dtype='object')
(10, 4)
Food ID           int64
Food Item        object
Price            object
Quantity Sold     int64
dtype: object
```

Question 10:

Convert the **Price** Series of the **prices** DataFrame from a string to a float.

In [37]:
```
prices['Price'] = prices['Price'].apply(lambda x: float(x.split()[0].repl
prices
```

Out[37]:

	Food ID	Food Item	Price	Quantity Sold
0	1	Sushi	3.99	35
1	2	Burrito	9.99	27
2	3	Taco	2.99	85
3	4	Quesadilla	4.25	76
4	5	Pizza	2.49	54
5	6	Pasta	13.99	32
6	7	Steak	24.99	29
7	8	Salad	11.25	36
8	9	Donut	0.99	24
9	10	Drink	1.75	96

Question 11:

Create a new column named **'Revenue'**. The **'Revenue'** column can be derived by using the following formula:

Revenue = Price * Quantity Sold

In [39]:
```
prices['Revenue'] = prices['Price'] * prices['Quantity Sold']
prices
```

Out[39]:

	Food ID	Food Item	Price	Quantity Sold	Revenue
0	1	Sushi	3.99	35	139.65
1	2	Burrito	9.99	27	269.73
2	3	Taco	2.99	85	254.15
3	4	Quesadilla	4.25	76	323.00
4	5	Pizza	2.49	54	134.46
5	6	Pasta	13.99	32	447.68
6	7	Steak	24.99	29	724.71
7	8	Salad	11.25	36	405.00
8	9	Donut	0.99	24	23.76
9	10	Drink	1.75	96	168.00

Question 12

Determine the total amount of revenue that was generated by the food sales found in the **prices** DataFrame.

In [41]:
```
prices.Revenue.sum().astype(int)
```

Out[41]:
```
2890
```

Question 13

Determine the average price of an item in the **prices** DataFrame.

In [45]:
```
print(f"Mean price: {prices.Price.mean():.2f}")
```

```
Mean price: 7.67
```