기초데이터과학 (01분반)

Programming assignment 02

1. Pandas Series를 활용하여 결측치를 찾고 합산, 평균을 구하시오.

In [1]:
```
import pandas as pd

# Series 객체를 생성하기 위해 주어진 데이터
d = [100, 200, None, 150, None, 300, 250]
i = ['A', 'B', 'C', 'D', 'E', 'F', 'G']
```

In [2]:
```
# d를 value로 i를 index로 하는 Series 객체 생성
data = pd.Series(d, index=i)
```

1-1. 위에서 생성한 Series 객체에서 결측치가 있는 값들을 찾고 출력하시오.

In [3]:
```
# 주어진 Series 중 결측치가 있는 값들을 찾고 출력
print("결측치가 있는 값들:")
print(data[data.isnull()])
print()
```

```
결측치가 있는 값들:
C   NaN
E   NaN
dtype: float64
```

1-2. 위에서 생성한 Series 객체에서 결측치가 없는 값들을 찾고 출력하시오.

In [4]:
```
# 주어진 Series 중 결측치가 없는 값들을 찾고 출력
print("결측치가 없는 값들:")
print(data[data.notnull()])
print()
```

```
결측치가 없는 값들:
A    100.0
B    200.0
D    150.0
F    300.0
G    250.0
dtype: float64
```

1-3. 위에서 생성한 Series 객체에서 결측치가 있는 값의 개수를 구하고 출력하시오.

In [5]:
```
# 결측치가 있는 값의 개수 구하고 출력
num_missing = data.isnull().sum()
```

```
print(f"결측치가 있는 값의 개수: {num_missing}")
print()
```

```
결측치가 있는 값의 개수: 2
```

1-4. 위에서 생성한 Series 객체에서 결측치가 없는 값들의 평균을 구하고 출력하시오.

In [6]:
```
# 결측치가 없는 값들의 평균을 구하고 출력
mean_value = data[data.notnull()].mean()
print(f"결측치가 없는 값들의 평균: {mean_value}")
print()
```

```
결측치가 없는 값들의 평균: 200.0
```

2. 다음과 같이 주어진 상품 데이터를 이용하여 아래 문제에 맞게 코드를 작성하시오.

In [1]:
```
import pandas as pd

# 주어진 데이터를 이용하여 데이터 프레임 객체를 생성하고 출력
data = {
    'Product': ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J'],
    'Price': [100, 200, 150, 300, 250, 130, 120, 180, 220, 270],
    'Quantity': [10, 5, 8, 6, 9, 15, 12, 7, 10, 8],
    'Category': ['Electronics', 'Clothing', 'Electronics', 'Electronics', \
                 'Clothing', 'Electronics', 'Clothing', 'Electronics', \
                 'Clothing', 'Clothing']
}
df = pd.DataFrame(data)
print("Default:")
print(df)
print()
```

```
Default:
  Product  Price  Quantity     Category
0       A    100        10  Electronics
1       B    200         5     Clothing
2       C    150         8  Electronics
3       D    300         6  Electronics
4       E    250         9     Clothing
5       F    130        15  Electronics
6       G    120        12     Clothing
7       H    180         7  Electronics
8       I    220        10     Clothing
9       J    270         8     Clothing
```

2-1. 위에서 생성한 DataFrame 객체의 행 인덱스를 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J'로 설정하여 출력하시오.

In [2]:
```
# DataFrame의 행 인덱스를 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J'로 설정
# Product 변수(열)의 값이 A, B, ... J 이므로 이 값을 이용

# DataFrame 객체의 사본 생성
df_new = df.copy()
```

```
# df_new 객체의 행 인덱스를 Product 변수의 값으로 설정
# 변수 이름이 함께 추출되지 않도록 pandas.DataFrame.to_numpy 함수 활용
df_new.index = df['Product'].to_numpy()

# DataFrame 출력
print("First Processing:")
print(df_new)
print()
```

```
First Processing:
  Product  Price  Quantity     Category
A       A    100        10  Electronics
B       B    200         5     Clothing
C       C    150         8  Electronics
D       D    300         6  Electronics
E       E    250         9     Clothing
F       F    130        15  Electronics
G       G    120        12     Clothing
H       H    180         7  Electronics
I       I    220        10     Clothing
J       J    270         8     Clothing
```

2-2. 2-1의 DataFrame 객체의 Product 변수의 값을 아래 주어진 리스트를 이용하여 변경하여 출력하시오.

In [3]:
```
# DataFrame의 'Product' 변수의 값을 아래 주어진 d를 이용하여 변경
d = ['Laptop', 'Jacket', 'Smartphone', 'Headphones', 'Sweater', \
     'Tablet', 'Scarf', 'Camera', 'Jeans', 'Hat']
df_new['Product'] = d

# DataFrame 출력
print("Second Processing:")
print(df_new)
print()
```

```
Second Processing:
      Product  Price  Quantity     Category
A      Laptop    100        10  Electronics
B      Jacket    200         5     Clothing
C  Smartphone    150         8  Electronics
D  Headphones    300         6  Electronics
E     Sweater    250         9     Clothing
F      Tablet    130        15  Electronics
G       Scarf    120        12     Clothing
H      Camera    180         7  Electronics
I       Jeans    220        10     Clothing
J         Hat    270         8     Clothing
```

2-3. 2-2의 DataFrame 객체에 Price, Quantity 변수 값의 순위를 구하고 이를 'Price_rank', 'Quantity_rank' 변수로 추가하여 출력하시오.

In [11]:
```
# DataFrame의 열 'Price', 'Quantity'의 순위를 'Price_rank', 'Quantity_rank' 열을
# DataFrame 객체의 어떤 변수(열) 값의 순위를 구하는 것은 pandas.DataFrame.rank 함
df_new['Price_rank'] = df_new['Price'].rank(ascending=False)
df_new['Quantity_rank'] = df_new['Quantity'].rank(ascending=False)

# DataFrame 출력
print("Third Processing:")
```

```
print(df_new)
print()
```

```
Third Processing:
      Product  Price  Quantity     Category  Price_rank  Quantity_rank
A      Laptop    100        10  Electronics        10.0            3.5
B      Jacket    200         5     Clothing         5.0           10.0
C  Smartphone    150         8  Electronics         7.0            6.5
D  Headphones    300         6  Electronics         1.0            9.0
E     Sweater    250         9     Clothing         3.0            5.0
F      Tablet    130        15  Electronics         8.0            1.0
G       Scarf    120        12     Clothing         9.0            2.0
H      Camera    180         7  Electronics         6.0            8.0
I       Jeans    220        10     Clothing         4.0            3.5
J         Hat    270         8     Clothing         2.0            6.5
```

2-4. 2-3의 DataFrame 객체에서 열 'Price_rank', 'Quantity_rank'를 각각 'Price', 'Quantity'의 뒤로 이동하여 출력하시오.

In [12]:
```
# DataFrame의 열 'Price_rank', 'Quantity_rank'를 각각 'Price', 'Quantity'의 뒤로
df_new_new = df_new[['Product', 'Price', 'Price_rank', \
                     'Quantity', 'Quantity_rank', 'Category']]

# DataFrame 출력
print("fourth Processing:")
print(df_new_new)
print()
```

```
fourth Processing:
      Product  Price  Price_rank  Quantity  Quantity_rank     Category
A      Laptop    100        10.0        10            3.5  Electronics
B      Jacket    200         5.0         5           10.0     Clothing
C  Smartphone    150         7.0         8            6.5  Electronics
D  Headphones    300         1.0         6            9.0  Electronics
E     Sweater    250         3.0         9            5.0     Clothing
F      Tablet    130         8.0        15            1.0  Electronics
G       Scarf    120         9.0        12            2.0     Clothing
H      Camera    180         6.0         7            8.0  Electronics
I       Jeans    220         4.0        10            3.5     Clothing
J         Hat    270         2.0         8            6.5     Clothing
```

2-5. 2-4의 DataFrame 객체를 카테고리 별로 분리하여 출력하시오.

In [13]:
```
# 카테고리별로 분리
# pandas.DataFrame.groupby 함수 활용
grouped = (df_new_new).groupby('Category')

# 카테고리별 DataFrame 출력
print("Last Processing:")
for category, group in grouped:
    print(f"Category: {category}")
    print(group, "\n")
```

```
Last Processing:
Category: Clothing
   Product  Price  Price_rank  Quantity  Quantity_rank  Category
B   Jacket    200         5.0         5           10.0  Clothing
E  Sweater    250         3.0         9            5.0  Clothing
G    Scarf    120         9.0        12            2.0  Clothing
I    Jeans    220         4.0        10            3.5  Clothing
J      Hat    270         2.0         8            6.5  Clothing

Category: Electronics
      Product  Price  Price_rank  Quantity  Quantity_rank     Category
A      Laptop    100        10.0        10            3.5  Electronics
C  Smartphone    150         7.0         8            6.5  Electronics
D  Headphones    300         1.0         6            9.0  Electronics
F      Tablet    130         8.0        15            1.0  Electronics
H      Camera    180         6.0         7            8.0  Electronics
```

3. 다음에 주어진 미국의 아기 이름 데이터로 아래 문제에 맞게 코드를 작성하시오.

In [14]:
```python
import pandas as pd

# 주어진 데이터 생성
df = pd.read_csv('https://raw.githubusercontent.com/guipsamora/pandas_exercises/
```

3-1. 위에서 생성한 DataFrame 객체의 앞 3개 행을 출력하시오.

In [15]:
```python
# Data 형태 확인
# DataFrame의 앞 3개 행을 출력
print("앞 3개 행:")
print(df.head(3))
print()
```

```
앞 3개 행:
   Unnamed: 0     Id     Name  Year Gender State  Count
0       11349  11350     Emma  2004      F    AK     62
1       11350  11351  Madison  2004      F    AK     48
2       11351  11352   Hannah  2004      F    AK     46
```

3-2. DataFrame 객체의 앞 행과 열 개수를 출력하시오.

In [16]:
```python
# DataFrame의 행과 열 개수를 출력
print("행과 열 개수:")
print(df.shape)
print()
```

```
행과 열 개수:
(1016395, 7)
```

3-3. DataFrame 객체에서 State가 'AK'이고, Count가 50 이상인 데이터를 추출하여 출력하시오.

In [17]:
```python
# State가 'AK'이고, Count가 50 이상인 데이터를 추출
filtered_data = df.query("State == 'AK' and Count >= 50")
```

```
# 추출한 데이터 출력
print(filtered_data)
```

```
      Unnamed: 0     Id        Name  Year Gender State  Count
0          11349  11350        Emma  2004      F    AK     62
204        11553  11554     Madison  2005      F    AK     53
639        11988  11989    Isabella  2007      F    AK     52
838        12187  12188        Emma  2008      F    AK     51
1063       12412  12413    Isabella  2009      F    AK     57
1275       12624  12625      Sophia  2010      F    AK     60
1276       12625  12626        Emma  2010      F    AK     51
1277       12626  12627    Isabella  2010      F    AK     51
1497       12846  12847      Olivia  2011      F    AK     60
1498       12847  12848        Emma  2011      F    AK     56
1499       12848  12849    Isabella  2011      F    AK     50
1732       13081  13082        Emma  2012      F    AK     57
1733       13082  13083      Sophia  2012      F    AK     56
1948       13297  13298        Emma  2013      F    AK     57
1949       13298  13299      Sophia  2013      F    AK     50
2173       13522  13523        Emma  2014      F    AK     50
2404       24076  24077       Ethan  2004      M    AK     65
2405       24077  24078      Joseph  2004      M    AK     63
2406       24078  24079       James  2004      M    AK     59
2407       24079  24080       Jacob  2004      M    AK     54
2408       24080  24081      Samuel  2004      M    AK     54
2409       24081  24082       Tyler  2004      M    AK     54
2610       24282  24283       Ethan  2005      M    AK     63
2611       24283  24284       Jacob  2005      M    AK     58
2612       24284  24285      Joshua  2005      M    AK     55
2613       24285  24286     Michael  2005      M    AK     52
2826       24498  24499       James  2006      M    AK     67
2827       24499  24500       Jacob  2006      M    AK     55
2828       24500  24501     Michael  2006      M    AK     55
2829       24501  24502      Andrew  2006      M    AK     52
2830       24502  24503       Ethan  2006      M    AK     52
2831       24503  24504       Logan  2006      M    AK     51
3043       24715  24716       Aiden  2007      M    AK     58
3044       24716  24717       Ethan  2007      M    AK     53
3045       24717  24718       Logan  2007      M    AK     53
3046       24718  24719  Alexander  2007      M    AK     52
3047       24719  24720     Michael  2007      M    AK     52
3270       24942  24943       James  2008      M    AK     67
3271       24943  24944       Jacob  2008      M    AK     60
3272       24944  24945     Michael  2008      M    AK     58
3273       24945  24946       Ethan  2008      M    AK     53
3274       24946  24947       Tyler  2008      M    AK     50
3507       25179  25180     Michael  2009      M    AK     58
3508       25180  25181       Ethan  2009      M    AK     57
3744       25416  25417     William  2010      M    AK     57
3745       25417  25418       James  2010      M    AK     51
3746       25418  25419     Michael  2010      M    AK     51
3747       25419  25420       Logan  2010      M    AK     50
3984       25656  25657       Mason  2011      M    AK     58
3985       25657  25658       James  2011      M    AK     52
3986       25658  25659     William  2011      M    AK     52
4224       25896  25897       James  2012      M    AK     52
4476       26148  26149        Liam  2013      M    AK     63
4731       26403  26404        Liam  2014      M    AK     65
4732       26404  26405       James  2014      M    AK     53
```

3-4. 3-3에서 추출한 데이터에서 Gender가 'F'인 데이터를 추출하여 출력하시오.

In [18]:
```
# 그 중 Gender가 'F'인 데이터를 추출
female_data = filtered_data.query("Gender == 'F'")

# 추출한 데이터 출력
print(female_data)
```

```
      Unnamed: 0     Id      Name  Year Gender State  Count
0          11349  11350      Emma  2004      F    AK     62
204        11553  11554   Madison  2005      F    AK     53
639        11988  11989  Isabella  2007      F    AK     52
838        12187  12188      Emma  2008      F    AK     51
1063       12412  12413  Isabella  2009      F    AK     57
1275       12624  12625    Sophia  2010      F    AK     60
1276       12625  12626      Emma  2010      F    AK     51
1277       12626  12627  Isabella  2010      F    AK     51
1497       12846  12847    Olivia  2011      F    AK     60
1498       12847  12848      Emma  2011      F    AK     56
1499       12848  12849  Isabella  2011      F    AK     50
1732       13081  13082      Emma  2012      F    AK     57
1733       13082  13083    Sophia  2012      F    AK     56
1948       13297  13298      Emma  2013      F    AK     57
1949       13298  13299    Sophia  2013      F    AK     50
2173       13522  13523      Emma  2014      F    AK     50
```

3-5. 3-4에서 추출한 데이터에서 Count 기준으로 내림차순 정렬하여 출력하시오.

In [19]:
```
# 추출한 데이터를 Count 기준으로 내림차순 정렬
sorted_data = female_data.sort_values(by='Count', ascending=False)

# 정렬된 데이터 출력
print(sorted_data)
```

```
      Unnamed: 0     Id      Name  Year Gender State  Count
0          11349  11350      Emma  2004      F    AK     62
1275       12624  12625    Sophia  2010      F    AK     60
1497       12846  12847    Olivia  2011      F    AK     60
1063       12412  12413  Isabella  2009      F    AK     57
1732       13081  13082      Emma  2012      F    AK     57
1948       13297  13298      Emma  2013      F    AK     57
1498       12847  12848      Emma  2011      F    AK     56
1733       13082  13083    Sophia  2012      F    AK     56
204        11553  11554   Madison  2005      F    AK     53
639        11988  11989  Isabella  2007      F    AK     52
838        12187  12188      Emma  2008      F    AK     51
1276       12625  12626      Emma  2010      F    AK     51
1277       12626  12627  Isabella  2010      F    AK     51
1499       12848  12849  Isabella  2011      F    AK     50
1949       13298  13299    Sophia  2013      F    AK     50
2173       13522  13523      Emma  2014      F    AK     50
```

3-6. 3-5에서 정렬한 데이터에서 Name과 Count 열만 추출하여 출력하시오.

In [20]:
```
# 'Name'과 'Count' 열만 추출
name_count_data = sorted_data[['Name', 'Count']]

# 추출한 데이터 출력
print(name_count_data)
```

```
          Name  Count
0         Emma     62
1275    Sophia     60
1497    Olivia     60
1063  Isabella     57
1732      Emma     57
1948      Emma     57
1498      Emma     56
1733    Sophia     56
204     Madison     53
639   Isabella     52
838       Emma     51
1276      Emma     51
1277  Isabella     51
1499  Isabella     50
1949    Sophia     50
2173      Emma     50
```

4. 미국의 아기 이름 데이터로 동일한 State와 Name을 가진 중복된 데이터를 제거하고 Count를 기준으로 내림차순 정렬하시오.

In [21]:
```
import pandas as pd

# 주어진 데이터 생성
df = pd.read_csv('https://raw.githubusercontent.com/guipsamora/pandas_exercises/
```

4-1. Name 변수의 값이 Emma인 데이터를 출력하시오.

In [22]:
```
# 데이터 중복 체크하기
# 'Name' 속성이 'Emma'인 데이터 추출
emma = df.query("Name == 'Emma'")

# 추출한 데이터 출력
print(emma)
```

```
         Unnamed: 0       Id  Name  Year Gender State  Count
0             11349    11350  Emma  2004      F    AK     62
205           11554    11555  Emma  2005      F    AK     49
418           11767    11768  Emma  2006      F    AK     49
644           11993    11994  Emma  2007      F    AK     41
838           12187    12188  Emma  2008      F    AK     51
...             ...      ...   ...   ...    ...   ...    ...
1013854     5633487  5633488  Emma  2010      F    WY     23
1013983     5633616  5633617  Emma  2011      F    WY     43
1014104     5633737  5633738  Emma  2012      F    WY     40
1014222     5633855  5633856  Emma  2013      F    WY     30
1014362     5633995  5633996  Emma  2014      F    WY     32

[566 rows x 7 columns]
```

4-2. State 별로 중복된 Name을 제거하고 출력하시오.

In [23]:
```
# State 별로 중복된 Name 제거
# pandas.DataFrame.drop_duplicates 함수 활용
unique_names_per_state = df.drop_duplicates(subset=['State', 'Name'], \
                                            keep='first')

# 중복 제거된 데이터 출력
print(unique_names_per_state)
```

```
         Unnamed: 0       Id     Name  Year Gender State  Count
0             11349    11350     Emma  2004      F    AK     62
1             11350    11351  Madison  2004      F    AK     48
2             11351    11352   Hannah  2004      F    AK     46
3             11352    11353    Grace  2004      F    AK     44
4             11353    11354    Emily  2004      F    AK     41
...             ...      ...      ...   ...    ...   ...    ...
1016379     5647410  5647411    Kyson  2014      M    WY      5
1016385     5647416  5647417     Odin  2014      M    WY      5
1016387     5647418  5647419  Raymond  2014      M    WY      5
1016392     5647423  5647424     Tyce  2014      M    WY      5
1016394     5647425  5647426   Waylon  2014      M    WY      5

[167628 rows x 7 columns]
```

4-3. Count 값을 기준으로 내림차순 정렬하여 State, Name, Count 변수 값만 출력하시오.

In [24]:
```
# Count 기준으로 내림차순 정렬
sorted_unique_names = unique_names_per_state.sort_values(by='Count', \
                                                         ascending=False)

# 'State', 'Name', 'Count' 변수(열)만 출력
print(sorted_unique_names[['State', 'Name', 'Count']])
```

```
        State      Name  Count
62272      CA     Emily   3416
107420     CA     Jacob   3336
62273      CA    Ashley   2926
107425     CA   Matthew   2863
62274      CA  Samantha   2489
...       ...       ...    ...
510465     MO    Adelia      5
510464     MO     Adela      5
510463     MO  Addalynn      5
510459     MO     Aanya      5
1016394    WY    Waylon      5

[167628 rows x 3 columns]
```

5. 치폴레(chipotle) 판매 기록을 담고 있는 데이터를 이용하여 'Steak Burrito'와 'Chicken Bowl'의 평균 판매 가격을 계산할 수 있도록 아래 문제에 맞게 코드를 작성하시오.

In [1]:
```
import pandas as pd

# 주어진 데이터 생성
chipotle_url = 'https://raw.githubusercontent.com/justmarkham/DAT8/master/data/c
chipo = pd.read_csv(chipotle_url, sep = '\t')
```

In [5]:
```
# 데이터의 변수(열) 속성 출력
chipo.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4622 entries, 0 to 4621
Data columns (total 5 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   order_id            4622 non-null   int64 
 1   quantity            4622 non-null   int64 
 2   item_name           4622 non-null   object
 3   choice_description  3376 non-null   object
 4   item_price          4622 non-null   object
dtypes: int64(2), object(3)
memory usage: 180.7+ KB
```

5.1 위에서 생성한 DataFrame 객체의 앞 5개 행을 출력하시오.

In [2]:
```
# Data 형태 확인
# DataFrame chipo의 앞 5개 행을 출력
print("앞 5개 행:")
print(chipo.head(5))
```

```
앞 5개 행:
   order_id  quantity                              item_name  \
0         1         1           Chips and Fresh Tomato Salsa   
1         1         1                                   Izze   
2         1         1                       Nantucket Nectar   
3         1         1  Chips and Tomatillo-Green Chili Salsa   
4         2         2                           Chicken Bowl   

                                choice_description item_price  
0                                              NaN     $2.39  
1                                    [Clementine]     $3.39  
2                                         [Apple]     $3.39  
3                                              NaN     $2.39  
4  [Tomatillo-Red Chili Salsa (Hot), [Black Beans...    $16.98  
```

5.2 위의 DataFrame에서 item_name 값이 'Steak Burrito'인 데이터와 'Chicken Bowl'인 데이터를 추출하시오.

In [12]:
```
# chipo 데이터에서 item_name이 'Steak Burrito'와 'Chicken Bowl'인 데이터 추출
steak_burrito = chipo[chipo['item_name'] == 'Steak Burrito']
chicken_bowl = chipo[chipo['item_name'] == 'Chicken Bowl']

# 결과 출력
print(steak_burrito)
print(chicken_bowl)
```

```
      order_id  quantity      item_name  \
7            4         1  Steak Burrito
9            5         1  Steak Burrito
31          16         1  Steak Burrito
43          20         1  Steak Burrito
46          21         1  Steak Burrito
...        ...       ...           ...
4546      1807         1  Steak Burrito
4607      1829         1  Steak Burrito
4610      1830         1  Steak Burrito
4617      1833         1  Steak Burrito
4618      1833         1  Steak Burrito

                                  choice_description item_price
7     [Tomatillo Red Chili Salsa, [Fajita Vegetables...    $11.75
9     [Fresh Tomato Salsa, [Rice, Black Beans, Pinto...     $9.25
31    [[Roasted Chili Corn Salsa (Medium), Fresh Tom...     $8.99
43    [Fresh Tomato Salsa, [Rice, Pinto Beans, Chees...    $11.75
46    [Tomatillo-Red Chili Salsa (Hot), [Rice, Fajit...     $8.99
...                                                 ...        ...
4546  [Fresh Tomato Salsa, [Rice, Black Beans, Cheese]]     $9.25
4607  [Tomatillo Green Chili Salsa, [Rice, Cheese, S...    $11.75
4610  [Fresh Tomato Salsa, [Rice, Sour Cream, Cheese...    $11.75
4617  [Fresh Tomato Salsa, [Rice, Black Beans, Sour ...    $11.75
4618  [Fresh Tomato Salsa, [Rice, Sour Cream, Cheese...    $11.75

[368 rows x 5 columns]
      order_id  quantity     item_name  \
4            2         2  Chicken Bowl
5            3         1  Chicken Bowl
13           7         1  Chicken Bowl
19          10         1  Chicken Bowl
26          13         1  Chicken Bowl
...        ...       ...           ...
4590      1825         1  Chicken Bowl
4591      1825         1  Chicken Bowl
4595      1826         1  Chicken Bowl
4599      1827         1  Chicken Bowl
4604      1828         1  Chicken Bowl

                                  choice_description item_price
4     [Tomatillo-Red Chili Salsa (Hot), [Black Beans...    $16.98
5     [Fresh Tomato Salsa (Mild), [Rice, Cheese, Sou...    $10.98
13    [Fresh Tomato Salsa, [Fajita Vegetables, Rice,...    $11.25
19    [Tomatillo Red Chili Salsa, [Fajita Vegetables...     $8.75
26    [Roasted Chili Corn Salsa (Medium), [Pinto Bea...     $8.49
...                                                 ...        ...
4590  [Roasted Chili Corn Salsa, [Rice, Black Beans,...    $11.25
4591  [Tomatillo Red Chili Salsa, [Rice, Black Beans...     $8.75
4595  [Tomatillo Green Chili Salsa, [Rice, Black Bea...     $8.75
4599      [Roasted Chili Corn Salsa, [Cheese, Lettuce]]     $8.75
4604  [Fresh Tomato Salsa, [Rice, Black Beans, Chees...     $8.75

[726 rows x 5 columns]
```

5.3 'item_price' 값이 실수형 데이터가 아니므로 연산이 가능하도록 item_price 값에서 '$'를 제거하고 자료형을 float로 변환하여 출력하시오.

In [6]:
```
# 'Steak Burrito'와 'Chicken Bowl' 데이터의 사본 생성
steak_burrito_numeric = steak_burrito.copy()
chicken_bowl_numeric = chicken_bowl.copy()

# 계산을 위해서 'item_price' 값에서 '$'를 제거하고 자료형을 float로 변환한 값을 '
# '$' 제거는 pandas.Series.str.replace 함수 활용
# 자료형 변환은 pandas.DataFrame.astype 함수 활용
steak_burrito_numeric['item_price_numeric'] = \
        steak_burrito.loc[:, 'item_price'].str.replace('$', '') \
        .astype(float)
chicken_bowl_numeric['item_price_numeric'] = \
        chicken_bowl.loc[:, 'item_price'].str.replace('$', '') \
        .astype(float)

# 'Steak Burrito' 데이터의 'item_price_numeric' 열만 출력
print(steak_burrito_numeric['item_price_numeric'])

# 'Chicken Bowl' 데이터의 'item_price_numeric' 열만 출력
print(chicken_bowl_numeric['item_price_numeric'])
```

```
7       11.75
9        9.25
31       8.99
43      11.75
46       8.99
        ...
4546     9.25
4607    11.75
4610    11.75
4617    11.75
4618    11.75
Name: item_price_numeric, Length: 368, dtype: float64
4       16.98
5       10.98
13      11.25
19       8.75
26       8.49
        ...
4590    11.25
4591     8.75
4595     8.75
4599     8.75
4604     8.75
Name: item_price_numeric, Length: 726, dtype: float64
```

5.4 'Steak Burrito'와 'Chicken Bowl'의 총 판매 금액을 계산하여 출력하시오.

In [7]:
```
# 'Steak Burrito'의 총 판매 금액 계산
steak_burrito_sales = steak_burrito_numeric.loc[:, 'item_price_numeric'].sum()

# 'Chicken Bowl'의 총 판매 금액 계산
chicken_bowl_sales = chicken_bowl_numeric.loc[:, 'item_price_numeric'].sum()

# 결과 출력
print(f"'Steak Burrito'의 총 판매 금액: {steak_burrito_sales}")
print(f"'Chicken Bowl'의 총 판매 금액: {chicken_bowl_sales}")
```

```
'Steak Burrito'의 총 판매 금액: 3851.4300000000003
'Chicken Bowl'의 총 판매 금액: 7342.7300000000005
```

5.5 'Steak Burrito'와 'Chicken Bowl'의 총 판매 수량을 계산하여 출력하시오.

In [8]:
```python
# 'Steak Burrito'의 총 판매 수량 계산
steak_burrito_quantity = steak_burrito_numeric.loc[:, 'quantity'].sum()

# 'Chicken Bowl'의 총 판매 수량 계산
chicken_bowl_quantity = chicken_bowl_numeric.loc[:, 'quantity'].sum()

# 결과 출력
print(f"'Steak Burrito'의 총 판매 수량: {steak_burrito_quantity}")
print(f"'Chicken Bowl'의 총 판매 수량: {chicken_bowl_quantity}")
```

```
'Steak Burrito'의 총 판매 수량: 386
'Chicken Bowl'의 총 판매 수량: 761
```

5.6 'Steak Burrito'와 'Chicken Bowl'의 평균 판매 가격을 계산하여 출력하시오.

In [11]:
```python
# 'Steak Burrito'의 평균 가격 계산
# 총 판매 금액을 총 판매 수량으로 나누고 round() 함수를 이용해 소수점 둘째 자리까
steak_burrito_avg_price = round(steak_burrito_sales / steak_burrito_quantity, 2)

# 'Chicken Bowl'의 평균 가격 계산
# 총 판매 금액을 총 판매 수량으로 나누고 round() 함수를 이용해 소수점 둘째 자리까
chicken_bowl_avg_price = round(chicken_bowl_sales / chicken_bowl_quantity, 2)

# 결과 출력
print((f"'Steak Burrito'의 평균 판매 가격: ${steak_burrito_avg_price} "))
print((f"'Chicken Bowl'의 평균 판매 가격: ${chicken_bowl_avg_price} "))
```

```
'Steak Burrito'의 평균 판매 가격: $9.98
'Chicken Bowl'의 평균 판매 가격: $9.65
```