

```
import pandas as pd
import numpy as np
```

Uploading the data

```
chipotle = pd.read_csv('/content/drive/MyDrive/chipotle.tsv', sep = '\t')
chipotle
```

order_id quantity			item_name	choice_description	item_price
0	1	1	Chips and Fresh Tomato Salsa	NaN	\$2.39
1	1	1	Izze	[Clementine]	\$3.39
2	1	1	Nantucket Nectar	[Apple]	\$3.39
3	1	1	Chips and Tomatillo-Green Chili Salsa	NaN	\$2.39
4	2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans...	\$16.98
...
4617	1833	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Black Beans, Sour ...	\$11.75
4618	1833	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Sour Cream, Cheese...	\$11.75
4619	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Pinto...	\$11.25
4620	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Lettu...	\$8.75
4621	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Pinto...	\$8.75

4622 rows × 5 columns

Creating a copy

```
new_df = chipotle.copy()
new_df
```

order_id quantity			item_name	choice_description	item_price
0	1	1	Chips and Fresh Tomato Salsa	NaN	\$2.39
1	1	1	Izze	[Clementine]	\$3.39
2	1	1	Nantucket Nectar	[Apple]	\$3.39
3	1	1	Chips and Tomatillo-Green Chili Salsa	NaN	\$2.39
4	2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans...	\$16.98
...
4617	1833	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Black Beans, Sour ...	\$11.75
4618	1833	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Sour Cream, Cheese...	\$11.75
4619	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Pinto...	\$11.25
4620	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Lettu...	\$8.75
4621	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Pinto...	\$8.75

4622 rows × 5 columns

```
print(new_df.shape)
print(new_df.dtypes)

(4622, 5)
order_id      int64
quantity      int64
item_name      object
choice_description  object
item_price     object
dtype: object
```

```
new_df['item_price'] = new_df['item_price'].astype(str)
print(new_df.dtypes)
```

```
order_id      int64
quantity      int64
item_name      object
choice_description  object
item_price     object
dtype: object
```

Analyzing the choice description column

```
new_df['choice_description'] = new_df.choice_description.str.replace("[", " ")
print(new_df.choice_description)
```

```
0      NaN
1      Clementine]
2      Apple]
3      NaN
4      Tomatillo-Red Chili Salsa (Hot),  Black Beans...
```

```
...
4617  Fresh Tomato Salsa,  Rice, Black Beans, Sour ...
4618  Fresh Tomato Salsa,  Rice, Sour Cream, Cheese...
4619  Fresh Tomato Salsa,  Fajita Vegetables, Pinto...
4620  Fresh Tomato Salsa,  Fajita Vegetables, Lettu...
4621  Fresh Tomato Salsa,  Fajita Vegetables, Pinto...
```

Name: choice_description, Length: 4622, dtype: object

<ipython-input-59-0e419f601591>:1: FutureWarning: The default value of regex will change from True to False in a future version. In addition, the escape sequence should be used when passing a regular expression.

```
new_df['choice_description'] = new_df.choice_description.str.replace("[", " ")
```

```
new_df['choice_description'] = new_df.choice_description.str.replace("]", " ")
print(new_df.choice_description)
```

```
0      NaN
1      Clementine
2      Apple
3      NaN
4      Tomatillo-Red Chili Salsa (Hot),  Black Beans...
```

```
...
4617  Fresh Tomato Salsa,  Rice, Black Beans, Sour ...
4618  Fresh Tomato Salsa,  Rice, Sour Cream, Cheese...
4619  Fresh Tomato Salsa,  Fajita Vegetables, Pinto...
4620  Fresh Tomato Salsa,  Fajita Vegetables, Lettu...
4621  Fresh Tomato Salsa,  Fajita Vegetables, Pinto...
```

Name: choice_description, Length: 4622, dtype: object

<ipython-input-60-f4c1b7addf8b>:1: FutureWarning: The default value of regex will change from True to False in a future version. In addition, the escape sequence should be used when passing a regular expression.

```
new_df['choice_description'] = new_df.choice_description.str.replace("]", " ")
```

```
new_df['choice_description'] = new_df.choice_description.str.replace("(", " ")
print(new_df.choice_description)
```

```
0      NaN
1      Clementine
2      Apple
3      NaN
4      Tomatillo-Red Chili Salsa  Hot),  Black Beans...
```

```
...
4617  Fresh Tomato Salsa,  Rice, Black Beans, Sour ...
4618  Fresh Tomato Salsa,  Rice, Sour Cream, Cheese...
4619  Fresh Tomato Salsa,  Fajita Vegetables, Pinto...
4620  Fresh Tomato Salsa,  Fajita Vegetables, Lettu...
4621  Fresh Tomato Salsa,  Fajita Vegetables, Pinto...
```

Name: choice_description, Length: 4622, dtype: object

<ipython-input-61-54b29a6172bc>:1: FutureWarning: The default value of regex will change from True to False in a future version. In addition, the escape sequence should be used when passing a regular expression.

```
new_df['choice_description'] = new_df.choice_description.str.replace("(", " ")
```

```
new_df['choice_description'] = new_df.choice_description.str.replace(" ", " ")
print(new_df.choice_description)
```

```
0      nan
1      Clementine
2      Apple
3      nan
4      Tomatillo-Red Chili Salsa Hot),  Black Beans, ...
```

```
...
4617  Fresh Tomato Salsa, Rice, Black Beans, Sour C...
```

```

4618    Fresh Tomato Salsa, Rice, Sour Cream, Cheese,...
4619    Fresh Tomato Salsa, Fajita Vegetables, Pinto ...
4620    Fresh Tomato Salsa, Fajita Vegetables, Lettuce
4621    Fresh Tomato Salsa, Fajita Vegetables, Pinto ...
Name: choice_description, Length: 4622, dtype: object

```

```

new_df['choice_description'] = new_df.choice_description.str.replace(")", " ")
print(new_df.choice_description)

```

```

0          nan
1    Clementine
2        Apple
3          nan
4    Tomatillo-Red Chili Salsa Hot , Black Beans, ...
...
4617    Fresh Tomato Salsa, Rice, Black Beans, Sour C...
4618    Fresh Tomato Salsa, Rice, Sour Cream, Cheese,...
4619    Fresh Tomato Salsa, Fajita Vegetables, Pinto ...
4620    Fresh Tomato Salsa, Fajita Vegetables, Lettuce
4621    Fresh Tomato Salsa, Fajita Vegetables, Pinto ...
Name: choice_description, Length: 4622, dtype: object

```

```

<ipython-input-66-7c59b57467d3>:1: FutureWarning: The default value of regex will change from True to False in a future version. In addi
new_df['choice_description'] = new_df.choice_description.str.replace(")", " ")

```

```
print(new_df.dtypes)
```

```

order_id      int64
quantity      int64
item_name     object
choice_description  object
item_price    object
dtype: object

```

```

new_df['choice_description'] = new_df['choice_description'].astype(str)
print(new_df.dtypes)

```

```

order_id      int64
quantity      int64
item_name     object
choice_description  object
item_price    object
dtype: object

```

describing the dataset

```
new_df.describe()
```

	order_id	quantity
count	4622.000000	4622.000000
mean	927.254868	1.075725
std	528.890796	0.410186
min	1.000000	1.000000
25%	477.250000	1.000000
50%	926.000000	1.000000
75%	1393.000000	1.000000
max	1834.000000	15.000000

checking the null values

```
new_df.isnull().sum()
```

```

order_id      0
quantity      0
item_name     0
choice_description  0
item_price    0
dtype: int64

```



checking for duplicate values

checking for duplicate values

```
rows_duplicated = new_df.duplicated()
print(rows_duplicated)
```

```
0      False
1      False
2      False
3      False
4      False
...
4617   False
4618   False
4619   False
4620   False
4621   False
Length: 4622, dtype: bool
```

dropping the duplicate values

```
newdf = new_df.drop_duplicates()
print(newdf)
```

```
   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
...      ...      ...
4617    1833         1      Steak Burrito
4618    1833         1      Steak Burrito
4619    1834         1  Chicken Salad Bowl
4620    1834         1  Chicken Salad Bowl
4621    1834         1  Chicken Salad 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
...      ...      ...
4617  Fresh Tomato Salsa, Rice, Black Beans, Sour C...    $11.75
4618  Fresh Tomato Salsa, Rice, Sour Cream, Cheese,...    $11.75
4619  Fresh Tomato Salsa, Fajita Vegetables, Pinto ...    $11.25
4620  Fresh Tomato Salsa, Fajita Vegetables, Lettuce     $8.75
4621  Fresh Tomato Salsa, Fajita Vegetables, Pinto ...     $8.75

[4563 rows x 5 columns]
```

updating the item price column

newdf.item_price

```
0      $2.39
1      $3.39
2      $3.39
3      $2.39
4     $16.98
...
4617   $11.75
4618   $11.75
4619   $11.25
4620    $8.75
4621    $8.75
Name: item_price, Length: 4563, dtype: object
```

```
newdf['item_price'] = newdf.item_price.str.replace("$", " ")
newdf['item_price'] = newdf.item_price
print(newdf)
```

```
   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
...
4617	1833	1	Steak Burrito
4618	1833	1	Steak Burrito
4619	1834	1	Chicken Salad Bowl
4620	1834	1	Chicken Salad Bowl
4621	1834	1	Chicken Salad 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
...
4617	Fresh Tomato Salsa, Rice, Black Beans, Sour C...	11.75
4618	Fresh Tomato Salsa, Rice, Sour Cream, Cheese,...	11.75
4619	Fresh Tomato Salsa, Fajita Vegetables, Pinto ...	11.25
4620	Fresh Tomato Salsa, Fajita Vegetables, Lettuce	8.75
4621	Fresh Tomato Salsa, Fajita Vegetables, Pinto ...	8.75

```
[4563 rows x 5 columns]
<ipython-input-74-08274dacf174>:1: FutureWarning: The default value of regex will change from True to False in a future version. In addi
newdf['item_price'] = newdf.item_price.str.replace("$", " ")
<ipython-input-74-08274dacf174>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
newdf['item_price'] = newdf.item_price.str.replace("$", " ")
<ipython-input-74-08274dacf174>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
newdf['item_price'] = newdf.item_price
```

```
newdf['choice_description'].mode()

0    nan
Name: choice_description, dtype: object
```

newdf

order_id	quantity	item_name	choice_description	item_price	
0	1	1	Chips and Fresh Tomato Salsa	nan	2.39
1	1	1	Izze	Clementine	3.39
2	1	1	Nantucket Nectar	Apple	3.39
3	1	1	Chips and Tomatillo-Green Chili Salsa	nan	2.39
4	2	2	Chicken Bowl	Tomatillo-Red Chili Salsa Hot , Black Beans, ...	16.98
...
4617	1833	1	Steak Burrito	Fresh Tomato Salsa, Rice, Black Beans, Sour C...	11.75
4618	1833	1	Steak Burrito	Fresh Tomato Salsa, Rice, Sour Cream, Cheese,...	11.75
4619	1834	1	Chicken Salad	Fresh Tomato Salsa, Fajita	11.25


```
newdf['quantity'].isnull().sum()

0
```

Analyzing the choice description column

```
newdf.loc[:, 'choice_description'] = newdf['choice_description'].str.split(',')
newdf = newdf.explode('choice_description')
```

<ipython-input-78-c75da727a4fb>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
newdf.loc[:, 'choice_description'] = newdf['choice_description'].str.split(',')


```
unique_choice_description = newdf['choice_description'].unique()
print("Unique Choice Description after Standardizing: ")
print(unique_choice_description)
```

```
Unique Choice Description after Standardizing:
['nan' ' Clementine ' ' Apple ' ' Tomatillo-Red Chili Salsa Hot '
 ' Black Beans' ' Rice' ' Cheese' ' Sour Cream '
 ' Fresh Tomato Salsa Mild ' ' Sour Cream' ' Guacamole' ' Lettuce '
 ' Tomatillo Red Chili Salsa' ' Fajita Vegetables' ' Pinto Beans'
 ' Tomatillo Green Chili Salsa' ' Fresh Tomato Salsa'
 ' Roasted Chili Corn Salsa' ' Guacamole '
 ' Tomatillo-Green Chili Salsa Medium ' ' Sprite '
 ' Tomatillo-Red Chili Salsa Hot ' ' Pomegranate Cherry ' ' Grapefruit '
 ' Roasted Chili Corn Salsa Medium ' ' Fajita Veggies'
 ' Roasted Chili Corn Salsa Medium ' ' Dr. Pepper '
 ' Fresh Tomato Salsa Mild ' ' Rice ' ' Cheese ' ' Blackberry '
 ' Mountain Dew ' ' Lettuce' ' Tomatillo Red Chili Salsa '
 ' Pineapple Orange Banana ' ' Tomatillo-Green Chili Salsa Medium '
 ' Diet Dr. Pepper ' ' Coca Cola ' ' Diet Coke ' ' Peach Orange ' ' Coke '
 ' Fajita Veggies ' ' Fajita Vegetables ' ' Fresh Tomato Mild '
 ' Lemonade ' ' Braised Carnitas' ' Cilantro-Lime Rice ' ' Black Beans '
 ' Adobo-Marinated and Grilled Chicken' ' Salsa' ' Cilantro-Lime Rice'
 ' Braised Barbacoa' ' Vegetarian Black Beans'
 ' Adobo-Marinated and Grilled Steak '
 ' Adobo-Marinated and Grilled Chicken ' ' Tomatillo Green Chili Salsa '
 ' Roasted Chili Corn Salsa ' ' Nestea '
 ' Adobo-Marinated and Grilled Steak' ' Fresh Tomato Salsa '
 ' Roasted Chili Corn Medium ' ' White Rice' ' Pinto Beans '
 ' Tomatillo Red Chili Hot ' ' Brown Rice' ' Vegetarian Black Beans '
 ' Tomatillo Green Chili Medium ' ' White Rice ' ' Brown Rice ']
```

```
newdf.isnull().sum()
```

```
order_id      0
quantity      0
item_name      0
choice_description  0
item_price     0
dtype: int64
```

```
new_df['choice_description'] = new_df.choice_description.str.replace("nan", "Rice")
print(new_df.choice_description)          #Replacing the null values with Rice as it's a staple food and preferred by everyone
```

```
0          Rice
1      Clementine
2          Apple
3          Rice
4  Tomatillo-Red Chili Salsa Hot , Black Beans, ...
...
4617  Fresh Tomato Salsa, Rice, Black Beans, Sour C...
4618  Fresh Tomato Salsa, Rice, Sour Cream, Cheese,...
4619  Fresh Tomato Salsa, Fajita Vegetables, Pinto ...
4620  Fresh Tomato Salsa, Fajita Vegetables, Lettuce
4621  Fresh Tomato Salsa, Fajita Vegetables, Pinto ...
Name: choice_description, Length: 4622, dtype: object
```

```
newdf
```



	order_id	quantity	item_name	choice_description	item_price	
	0	1	1	Chips and Fresh Tomato Salsa	nan	2.39
	1	1	1	Izze	Clementine	3.39
	2	1	1	Nantucket Nectar	Apple	3.39
	3	1	1	Chips and Tomatillo-Green Chili Salsa	nan	2.39
	4	2	2	Chicken Bowl	Tomatillo-Red Chili Salsa Hot	16.98

newdf.dtypes

```
order_id      int64
quantity      int64
item_name      object
choice_description  object
item_price     object
dtype: object
16844 rows x 5 columns
```

checking inconsistencies in item_price and quantity

newdf['quantity']

```
0      1
1      1
2      1
3      1
4      2
...
4620    1
4621    1
4621    1
4621    1
4621    1
Name: quantity, Length: 16844, dtype: int64
```

newdf['quantity'].isnull().sum()

```
0
```

newdf['item_price']

```
0      2.39
1      3.39
2      3.39
3      2.39
4     16.98
...
4620    8.75
4621    8.75
4621    8.75
4621    8.75
4621    8.75
Name: item_price, Length: 16844, dtype: object
```

```
newdf['item_price'] = newdf['item_price'].astype(float)
newdf['item_price']
```

```
0      2.39
1      3.39
2      3.39
3      2.39
4     16.98
...
4620    8.75
4621    8.75
4621    8.75
4621    8.75
4621    8.75
Name: item_price, Length: 16844, dtype: float64
```

```
newdf['item_price'] = newdf.item_price.round(decimals = 1)
newdf['item_price']
```

```

0      2.4
1      3.4
2      3.4
3      2.4
4      17.0
...
4620    8.8
4621    8.8
4621    8.8
4621    8.8
4621    8.8
Name: item_price, Length: 16844, dtype: float64

```

```

newdf['item_price'] = newdf['item_price'].astype(int)
newdf['item_price']

```

```

0      2
1      3
2      3
3      2
4      17
..
4620    8
4621    8
4621    8
4621    8
4621    8
Name: item_price, Length: 16844, dtype: int64

```

Cross-reference

```
newdf['order_id'].isnull().sum()
```

```
0
```

```

unique_order_ids = newdf['order_id'].unique()
print("Unique Order IDs: ")
print(unique_order_ids)

```

```

Unique Order IDs:
[ 1  2  3 ... 1832 1833 1834]

```

```

order_id_stats = newdf['order_id'].describe()
print(order_id_stats)

```

```

count    16844.000000
mean       926.329316
std       525.701346
min         1.000000
25%       482.000000
50%       926.000000
75%      1385.000000
max      1834.000000
Name: order_id, dtype: float64

```

```

cr_ref = newdf.groupby('order_id')['quantity'].count().reset_index(name = 'count_per_order')
print("Cross Reference Result: ")
print(cr_ref)

```

```

Cross Reference Result:
  order_id  count_per_order
0         1                4
1         2                5
2         3                7
3         4               13
4         5                8
...      ...             ...
1829     1830               11
1830     1831                9
1831     1832                5
1832     1833               13
1833     1834               12

```

```
[1834 rows x 2 columns]
```


standardizing item name column

```
newdf['item_name'] = newdf['item_name'].str.lower().str.strip()
print("Dataframe after standardizing 'Item Name': ")
print(newdf)
```

```
Dataframe after standardizing 'Item Name':
   order_id  quantity item_name \
0          1         1  chips and fresh tomato salsa
1          1         1           ize
2          1         1  nantucket nectar
3          1         1  chips and tomatillo-green chili salsa
4          2         2      chicken bowl
...      ...      ...      ...
4620      1834         1      chicken salad bowl
4621      1834         1      chicken salad bowl
4621      1834         1      chicken salad bowl
4621      1834         1      chicken salad bowl
4621      1834         1      chicken salad bowl
```

```
   choice_description  item_price
0                  nan           2
1          Clementine           3
2             Apple           3
3                  nan           2
4  Tomatillo-Red Chili Salsa Hot       17
...      ...      ...
4620          Lettuce           8
4621  Fresh Tomato Salsa           8
4621  Fajita Vegetables           8
4621          Pinto Beans           8
4621          Lettuce           8
```

[16844 rows x 5 columns]

```
unique_item_names = newdf['item_name'].unique()
print("Unique Item Names after Standardizing: ")
print(unique_item_names)
```

```
Unique Item Names after Standardizing:
['chips and fresh tomato salsa' 'ize' 'nantucket nectar'
 'chips and tomatillo-green chili salsa' 'chicken bowl' 'side of chips'
 'steak burrito' 'steak soft tacos' 'chips and guacamole'
 'chicken crispy tacos' 'chicken soft tacos' 'chicken burrito'
 'canned soda' 'barbacoa burrito' 'carnitas burrito' 'carnitas bowl'
 'bottled water' 'chips and tomatillo green chili salsa' 'barbacoa bowl'
 'chips' 'chicken salad bowl' 'steak bowl' 'barbacoa soft tacos'
 'veggie burrito' 'veggie bowl' 'steak crispy tacos'
 'chips and tomatillo red chili salsa' 'barbacoa crispy tacos'
 'veggie salad bowl' 'chips and roasted chili-corn salsa'
 'chips and roasted chili corn salsa' 'carnitas soft tacos'
 'chicken salad' 'canned soft drink' 'steak salad bowl'
 '6 pack soft drink' 'chips and tomatillo-red chili salsa' 'bowl'
 'burrito' 'crispy tacos' 'carnitas crispy tacos' 'steak salad'
 'chips and mild fresh tomato salsa' 'veggie soft tacos'
 'carnitas salad bowl' 'barbacoa salad bowl' 'salad' 'veggie crispy tacos'
 'veggie salad' 'carnitas salad']
```

Quantity and Price Relationships

```
stats_quantity = newdf['quantity'].describe()
stats_price = newdf['item_price'].describe()
print("Descriptive Statistics for Quantity: ")
print(stats_quantity)
print("\n Descriptive Statistics for Item Price: ")
print(stats_price)
```

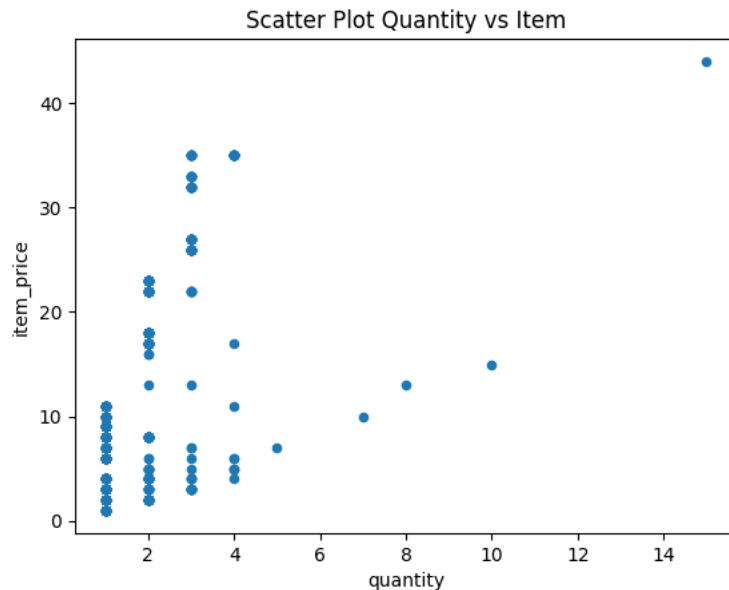
```
Descriptive Statistics for Quantity:
count    16844.000000
mean         1.053610
std         0.293469
min         1.000000
25%         1.000000
50%         1.000000
75%         1.000000
max         15.000000
Name: quantity, dtype: float64
```

Descriptive Statistics for Item Price:

```
count    16844.000000
mean       9.168844
std        3.394151
min         1.000000
25%         8.000000
50%         9.000000
75%        11.000000
max        44.000000
Name: item_price, dtype: float64
```

```
newdf.plot.scatter(x = 'quantity', y = 'item_price', title = 'Scatter Plot Quantity vs Item')
```

```
<Axes: title={'center': 'Scatter Plot Quantity vs Item'}, xlabel='quantity', ylabel='item_price'>
```



Checking the data integrity

```
stats_quantity = newdf['quantity'].describe()
stats_price = newdf['item_price'].describe()
print("Descriptive Statistics for Quantity: ")
print(stats_quantity)
print("\n Descriptive Statistics for Item Price: ")
print(stats_price)
```

Descriptive Statistics for Quantity:

```
count    16844.000000
mean       1.053610
std        0.293469
min         1.000000
25%         1.000000
50%         1.000000
75%         1.000000
max        15.000000
Name: quantity, dtype: float64
```

Descriptive Statistics for Item Price:

```
count    16844.000000
mean       9.168844
std        3.394151
min         1.000000
25%         8.000000
50%         9.000000
75%        11.000000
max        44.000000
Name: item_price, dtype: float64
```

```
quantity_item_relation = newdf.groupby('item_name')['quantity'].sum().reset_index(name= 'total_quantity')
print("Quantity-Item Relationship: ")
print(quantity_item_relation)
```

Quantity-Item Relationship:

```
   item_name  total_quantity
0  6 pack soft drink         55
```

1	barbacoa bowl	347
2	barbacoa burrito	467
3	barbacoa crispy tacos	54
4	barbacoa salad bowl	48
5	barbacoa soft tacos	125
6	bottled water	204
7	bowl	21
8	burrito	35
9	canned soda	124
10	canned soft drink	340
11	carnitas bowl	441
12	carnitas burrito	322
13	carnitas crispy tacos	36
14	carnitas salad	6
15	carnitas salad bowl	35
16	carnitas soft tacos	173
17	chicken bowl	4125
18	chicken burrito	3078
19	chicken crispy tacos	218
20	chicken salad	40
21	chicken salad bowl	559
22	chicken soft tacos	482
23	chips	227
24	chips and fresh tomato salsa	130
25	chips and guacamole	501
26	chips and mild fresh tomato salsa	1
27	chips and roasted chili corn salsa	23
28	chips and roasted chili-corn salsa	18
29	chips and tomatillo green chili salsa	45
30	chips and tomatillo red chili salsa	48
31	chips and tomatillo-green chili salsa	33
32	chips and tomatillo-red chili salsa	24
33	crispy tacos	2
34	izze	19
35	nantucket nectar	29
36	salad	6
37	side of chips	110
38	steak bowl	1238
39	steak burrito	2076
40	steak crispy tacos	151
41	steak salad	19
42	steak salad bowl	163
43	steak soft tacos	233
44	veggie bowl	541
45	veggie burrito	580
46	veggie crispy tacos	5
47	veggie salad	39
48	veggie salad bowl	103
49	veggie soft tacos	48

```
price_item_relation = newdf.groupby('item_name')['item_price'].mean().reset_index(name='average_price')
print("\nPrice-Item Relationship:")
print(price_item_relation)
```

Price-Item Relationship:		
	item_name	average_price
0	6 pack soft drink	6.129630
1	barbacoa bowl	9.951009
2	barbacoa burrito	9.591006
3	barbacoa crispy tacos	10.440000
4	barbacoa salad bowl	10.291667
5	barbacoa soft tacos	9.624000
6	bottled water	1.490323
7	bowl	13.818182
8	burrito	7.000000
9	canned soda	1.215686
10	canned soft drink	1.182759
11	carnitas bowl	10.683453
12	carnitas burrito	9.823344
13	carnitas crispy tacos	10.666667
14	carnitas salad	9.000000
15	carnitas salad bowl	10.428571
16	carnitas soft tacos	9.184971
17	chicken bowl	9.792587
18	chicken burrito	9.682491
19	chicken crispy tacos	9.615385
20	chicken salad	9.125000
21	chicken salad bowl	10.686391
22	chicken soft tacos	9.206074
23	chips	2.182692
24	chips and fresh tomato salsa	3.145455
25	chips and guacamole	4.219409

26	chips and mild fresh tomato salsa	3.000000
27	chips and roasted chili corn salsa	3.090909
28	chips and roasted chili-corn salsa	2.000000
29	chips and tomatillo green chili salsa	3.093023
30	chips and tomatillo red chili salsa	3.086957
31	chips and tomatillo-green chili salsa	2.129032
32	chips and tomatillo-red chili salsa	2.631579
33	crispy tacos	7.000000
34	izze	3.000000
35	nantucket nectar	3.222222
36	salad	7.000000
37	side of chips	1.158416
38	steak bowl	10.418428
39	steak burrito	10.263558
40	steak crispy tacos	10.013699
41	steak salad	8.894737
42	steak salad bowl	11.235294
43	steak soft tacos	9.569565
44	veggie bowl	9.845283
45	veggie burrito	9.575972
46	veggie crispy tacos	8.000000
47	veggie salad	8.000000
48	veggie salad bowl	9.893204
49	veggie soft tacos	10.162791

```
newdf['total_price_calculated'] = newdf['quantity'] * newdf['item_price']
discrepancies = newdf[newdf['total_price_calculated'] != newdf['item_price']]
print("\nOrders with Total Price Discrepancies:")
print(discrepancies[['order_id', 'item_name', 'quantity', 'item_price', 'total_price_calculated']])
```

Orders with Total Price Discrepancies:

	order_id	item_name	quantity	item_price	\
4	2	chicken bowl	2	17	
4	2	chicken bowl	2	17	
4	2	chicken bowl	2	17	
4	2	chicken bowl	2	17	
4	2	chicken bowl	2	17	
...	
4561	1813	chicken salad bowl	2	17	
4561	1813	chicken salad bowl	2	17	
4561	1813	chicken salad bowl	2	17	
4561	1813	chicken salad bowl	2	17	
4582	1822	bottled water	2	3	

	total_price_calculated
4	34
4	34
4	34
4	34
4	34
...	...
4561	34
4561	34
4561	34
4561	34
4582	6

[758 rows x 5 columns]

Handling categorial data

```
df_encoded = pd.concat([newdf, pd.get_dummies(newdf['item_name'], prefix='item')], axis=1)
df_encoded
```

```

item_6
pack soft drink
item_barbacoa bowl
item_barbacoa burrito
item_barbacoa crispy tacos
...
item_steak crisp taco

ity item_name choice_description item_price total_price_calculated
1 chips and fresh tomato salsa nan 2 2 0 0 0 0 ...
1 izee Clementine 3 3 0 0 0 0 ...
1 nantucket nectar Apple 3 3 0 0 0 0 ...
1 chips and tomatillo-green chili salsa nan 2 2 0 0 0 0 ...
2 chicken bowl Tomatillo-Red Chili Salsa Hot 17 34 0 0 0 0 ...
... ... ... ... ...
1 chicken Lettuce 8 8 0 0 0 0 ...
Consistent quantity and price units
1 salad bowl Fresh Tomato Salsa 8 8 0 0 0 0 ...
newdf['total_price'] = newdf['quantity'] * newdf['item_price']
print("DataFrame with Total Price:")
print(newdf)

DataFrame with Total Price:
order_id quantity item_name \
0 1 1 chips and fresh tomato salsa
1 1 1 izee
2 1 1 nantucket nectar
3 1 1 chips and tomatillo-green chili salsa
4 2 2 chicken bowl
... ... ...
4620 1834 1 chicken salad bowl
4621 1834 1 chicken salad bowl
4621 1834 1 chicken salad bowl
4621 1834 1 chicken salad bowl
4621 1834 1 chicken salad bowl

choice_description item_price total_price_calculated \
0 nan 2 2
1 Clementine 3 3
2 Apple 3 3
3 nan 2 2
4 Tomatillo-Red Chili Salsa Hot 17 34
... ... ...
4620 Lettuce 8 8
4621 Fresh Tomato Salsa 8 8
4621 Fajita Vegetables 8 8
4621 Pinto Beans 8 8
4621 Lettuce 8 8

total_price
0 2
1 3
2 3
3 2
4 34
... ...
4620 8
4621 8
4621 8
4621 8
4621 8

[16844 rows x 7 columns]
```

