Ex2 - Getting and Knowing your Data

This time we are going to pull data directly from the internet. Special thanks to: https://github.com/justmarkham for sharing the dataset and materials.

Step 1. Import the necessary libraries

In [25]:
```
import pandas as pd
import numpy as np
```

Step 2. Import the dataset from this address.

Step 3. Assign it to a variable called chipo.

In [6]:
```
df = pd.read_table("https://raw.githubusercontent.com/justmarkham/DAT8/master/data/chipotle.tsv")
df
```

Out[6]:

	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

Step 4. See the first 10 entries

In [15]:
```
df.head(10)
```

Out[15]:

	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
5	3	1	Chicken Bowl	[Fresh Tomato Salsa (Mild), [Rice, Cheese, Sou...	$10.98
6	3	1	Side of Chips	NaN	$1.69
7	4	1	Steak Burrito	[Tomatillo Red Chili Salsa, [Fajita Vegetables...	$11.75
8	4	1	Steak Soft Tacos	[Tomatillo Green Chili Salsa, [Pinto Beans, Ch...	$9.25
9	5	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Black Beans, Pinto...	$9.25

Step 5. What is the number of observations in the dataset?

In [16]:
```
# Solution 1
len(df)
```

Out[16]: 4622

In [19]:
```
# Solution 2

df.count()[0]
```

Out[19]: 4622

Step 6. What is the number of columns in the dataset?

In [25]:
```
len(df.columns)
```

Out[25]: 5

Step 7. Print the name of all the columns.

In [26]:
```
print(df.columns)
```

```
Index(['order_id', 'quantity', 'item_name', 'choice_description',
       'item_price'],
      dtype='object')
```

Step 8. How is the dataset indexed?

In [27]:
```
df.index
```

Out[27]: RangeIndex(start=0, stop=4622, step=1)

Step 9. Which was the most-ordered item?

In [10]:
```
c = df.groupby('item_name').sum()
c = c.sort_values(['quantity'], ascending = False)
c.head(2)
```

Out[10]:

item_name	order_id	quantity
Chicken Bowl	713926	761
Chicken Burrito	497303	591

Step 10. For the most-ordered item, how many items were ordered?

In [11]:
```
print('For the most-ordered item, ordered were:',str(713926))
```

```
For the most-ordered item, ordered were: 713926
```

Step 11. What was the most ordered item in the choice_description column?

In [13]:
```
d = df.groupby('choice_description').sum()
d = d.sort_values(['quantity'], ascending = False)
d.head(2)
```

Out[13]:

choice_description	order_id	quantity
[Diet Coke]	123455	159
[Coke]	122752	143

Step 12. How many items were orderd in total?

In [15]:
```
total = df.quantity.sum()
total
```

Out[15]: 4972

Step 13. Turn the item price into a float

Step 13.a. Check the item price type

In [16]:
```
df.item_price.dtype
```

Out[16]: dtype('O')

Step 13.b. Create a lambda function and change the type of item price

In [21]:
```
try:
    dollarizer = lambda x: float(x[1:-1])
    df.item_price = df.item_price.apply(dollarizer)

except:TypeError
```

Step 13.c. Check the item price type

In [23]:
```
df.item_price.dtype
```

Out[23]: dtype('float64')

Step 14. How much was the revenue for the period in the dataset?

In [31]:
```
revenue = (df['quantity'] * df['item_price']).sum()

print('Revenue was: $' + str(np.round(revenue,2)))
```

```
Revenue was: $39237.02
```

Step 15. How many orders were made in the period?

In [27]:
```
orders = df.order_id.value_counts().count()
orders
```

Out[27]: 1834

Step 16. What is the average revenue amount per order?

In [33]:
```
# Solution 1

df['revenue'] = df['quantity'] * df['item_price']
d = order_grouped = df.groupby(by=['order_id']).sum()
order_grouped.mean()['revenue']
```

Out[33]: 21.394231188658654

In [34]:
```
# Solution 2

df.groupby("order_id").sum().mean()["revenue"]
```

Out[34]: 21.394231188658654

Step 17. How many different items are sold?

In [35]:
```
df.item_name.value_counts().count()
```

Out[35]: 50

In [ ]:
```

```