Darshan UNIVERSITY
योग: कर्मसु कौशलम्

Data Mining

Reference:- GitHub (New Restaurant Data Set)

Name - Ritesh Lakhani

Enrollment No - 22010101099

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 [1]:
```
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 [2]:
```
chipo = pd.read_csv('https://raw.githubusercontent.com/justmarkham/DAT8/master/data/chipotle.tsv',sep='\t')
chipo
```

Out[2]:

	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 [3]:
```
chipo.head(10)
```

Out[3]:

	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 [4]:
```
# Solution 1
chipo.shape[0]
```

Out[4]:
```
4622
```

In [5]:
```
# Solution 2
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
```

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

In [6]:
```
num_columns = chipo.shape[1]
print(num_columns)
```

```
5
```

Step 7. Print the name of all the columns.

In [7]:
```
chipo.columns
```

Out[7]:
```
Index(['order_id', 'quantity', 'item_name', 'choice_description',
       'item_price'],
      dtype='object')
```

Step 8. How is the dataset indexed?

In [8]:
```
chipo.index
```

Out[8]:
```
RangeIndex(start=0, stop=4622, step=1)
```

Step 9. Which was the most-ordered item?

In [9]:
```
c = chipo.groupby('item_name').sum()
c = c.sort_values(['quantity'],ascending=False)
c.head(1)
```

Out[9]:

item_name	order_id	quantity	choice_description	item_price
Chicken Bowl	713926	761	[Tomatillo-Red Chili Salsa (Hot), [Black Beans...	16.9810.98 11.258.75 8.4911.25 $8.75 ...

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

In [10]:
```
c = chipo.groupby('item_name').sum()
c = c.sort_values(['quantity'],ascending=False)
c.head(1)
```

Out[10]:

item_name	order_id	quantity	choice_description	item_price
Chicken Bowl	713926	761	[Tomatillo-Red Chili Salsa (Hot), [Black Beans...	16.9810.98 11.258.75 8.4911.25 $8.75 ...

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

In [11]:
```
c = chipo.groupby('choice_description').sum()
c = c.sort_values(['quantity'],ascending=False)
c.head(1)
```

Out[11]:

choice_description	order_id	quantity	item_name	item_price
[Diet Coke]	123455	159	Canned SodaCanned SodaCanned Soda6 Pack Soft D...	2.181.09 1.096.49 2.181.25 1.096.4...

Step 12. How many items were orderd in total?

In [12]:
```
total_items_orders = chipo.quantity.sum()
total_items_orders
```

Out[12]:
```
4972
```

Step 13. Turn the item price into a float

Step 13.a. Check the item price type

In [17]:
```
chipo['item_price'].dtype
```

Out[17]:
```
dtype('float64')
```

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

In [15]:
```
chipo['item_price'] = chipo['item_price'].apply(lambda x: float(x[1:]))
```

Step 13.c. Check the item price type

In [16]:
```
chipo['item_price'].dtype
```

Out[16]:
```
dtype('float64')
```

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

In [18]:
```
x = chipo['quantity'] * chipo['item_price']
total = x.sum()
total
```

Out[18]:
```
39237.02
```

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

In [19]:
```
chipo.order_id.nunique()
```

Out[19]:
```
1834
```

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

In [20]:
```
chipo['total_revenue'] = chipo['quantity']*chipo['item_price']
order_revenue =chipo.groupby('order_id')['total_revenue'].sum()
order_revenue.mean()
```

Out[20]:
```
21.39423118865867
```

Step 17. How many different items are sold?

In [21]:
```
chipo.item_name.nunique()
```

Out[21]:
```
50
```