Darshan
UNIVERSITY

योग: कर्मसु कौशलम्

Data Mining

Lab - 4

Mohil Parmar

23010101192

Step 1. Import the necessary libraries

In [ ]:

```
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 [ ]:

```
# .tsv -> tab seprated
chipo = pd.read_csv('https://raw.githubusercontent.com/justmarkham/DAT8/master/data/chipotle.tsv',sep="\t")
chipo
```

Out[ ]:

	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 [4]:

```
chipo.head(10)
```

Out[4]:

	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 [ ]:

```python
# Solution 1

chipo.shape[0]  # number of rows
```

Out[ ]:

```
4622
```

In [8]:

```python
# 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 [ ]: chipo.shape[1] # number of columns
```

```
Out[ ]: 5
```

Step 7. Print the name of all the columns.

```
In [10]: chipo.columns
```

```
Out[10]: Index(['order_id', 'quantity', 'item_name', 'choice_description',
                'item_price'],
               dtype='object')
```

Step 8. How is the dataset indexed?

```
In [11]: chipo.index
```

```
Out[11]: RangeIndex(start=0, stop=4622, step=1)
```

Step 9. Number of Unique Items ?

```
In [16]: chipo["item_name"].nunique()
```

```
Out[16]: 50
```

Step 10. Which was the most-ordered item?

```
In [21]: a = chipo.groupby('item_name')

         b = a.sum()

         ordered  = b.sort_values(['quantity'], ascending=False)
         ordered[['order_id', 'quantity']].head(1)
```

Out[21]:

	order_id	quantity
item_name		
Chicken Bowl	713926	761

Step 11. How many items were orderd in total?

```
In [25]: print(chipo['quantity'].sum())
```

```
4972
```

Step 12. Turn the item price into a float

Step 12.a. Check the item price type

```
In [26]: chipo.item_price.dtype
```

```
Out[26]: dtype('O')
```

Step 12.b. Create a lambda function and change the type of item price

```
In [30]: chipo['item_price'] = chipo['item_price'].apply(lambda x: float(x[1:]))
         chipo
```

Out[30]:

	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 12.c. Check the item price type

In [31]:

```
chipo['item_price'].dtype
```

Out[31]:

```
dtype('float64')
```

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

In [33]:

```
chipo['revenue'] = chipo['quantity']*chipo['item_price']
revenue = chipo['revenue'].sum()
print('Revenue was : $',revenue)
```

```
Revenue was : $ 39237.02
```

Step 15. How many orders were made ?

In [36]:

```
print(chipo['order_id'].nunique())
```

```
1834
```

Step 17. How many different choice descriptions are there?

In [37]:

```
chipo['choice_description'].nunique()
```

Out[37]:

```
1043
```

Step 18. What items have been ordered more than 100 times?

In [53]:

```
a = chipo.groupby('item_name')['quantity'].sum()
a[a>100]
```

Out[53]:

```
item_name
Bottled Water                   211
Canned Soda                     126
Canned Soft Drink               351
Chicken Bowl                    761
Chicken Burrito                 591
Chicken Salad Bowl              123
Chicken Soft Tacos              120
Chips                           230
Chips and Fresh Tomato Salsa    130
Chips and Guacamole             506
Side of Chips                   110
Steak Bowl                      221
Steak Burrito                   386
Name: quantity, dtype: int64
```

Step 19. What is the average revenue amount per order?

In [65]:

```
# Solution 1
a = chipo.groupby('order_id')['revenue'].sum().mean()
print(a)
```

```
21.39423118865867
```

In [67]:

```
# Solution 2
a = chipo.groupby('order_id')['revenue'].sum().sum() / chipo['order_id'].nunique()
print(a)
```

```
21.39423118865867
```