



Roll No - 130

Enrollment No - 24010101694

Lab - 4

Karan Sonagara

Step 1. Import the necessary libraries

```
In [3]: import pandas as pd  
import numpy as ny
```

Step 2. Import the dataset from this [address](#).

Step 3. Assign it to a variable called chipo.

```
In [5]: df = pd.read_csv("https://raw.githubusercontent.com/justmarkham/DAT8/master/data/chipo.csv")  
df
```

Out[5]:

	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 [7]: `df.head(10)`

Out[7]:

	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 [9]: `# Solution 1`
`print(df.shape[0])`

4622

In [13]: `# Solution 2`
`df.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 [20]: `print(df.shape[1])`

5

Step 7. Print the name of all the columns.

```
In [16]: df.columns
```

```
Out[16]: Index(['order_id', 'quantity', 'item_name', 'choice_description',
               'item_price'],
              dtype='object')
```

Step 8. How is the dataset indexed?

```
In [34]: df.index
```

```
Out[34]: RangeIndex(start=0, stop=4622, step=1)
```

Step 9. Number of Unique Items ?

```
In [50]: df["item_name"].nunique()
```

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

Step 10. Which was the most-ordered item?

```
In [72]: c = df.groupby('item_name')
         c = c.sum()
         c = c.sort_values(['quantity'], ascending = False)
         c[['quantity', 'order_id']].head(1)
```

```
Out[72]:
```

	quantity	order_id
item_name		
Chicken Bowl	761	713926

Step 11. How many items were orderd in total?

```
In [86]: cf = df.quantity.sum()
         cf
```

```
Out[86]: 4972
```

Step 12. Turn the item price into a float

Step 12.a. Check the item price type

```
In [52]: df["item_price"].dtype
```

```
Out[52]: dtype('O')
```

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

```
In [5]: df['item_price'] = df['item_price'].apply(lambda x: float(x.replace('$', '')))
```

Step 12.c. Check the item price type

```
In [7]: df["item_price"].dtype
```

```
Out[7]: dtype('float64')
```

```
In [9]: df['item_price']
```

```
Out[9]: 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: 4622, dtype: float64
```

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

```
In [15]: gt = (df['item_price'] * df['quantity']).sum()
          print(gt)
```

```
39237.02
```

Step 15. How many orders were made ?

```
In [104... total_orders = df['order_id'].nunique()
           print("Total number of orders:", total_orders)
```

```
Total number of orders: 1834
```

Step 17. How many different choice descriptions are there?

```
In [56]: df["choice_description"].nunique()
```

```
Out[56]: 1043
```

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

```
In [100... item_totals = df.groupby('item_name')['quantity'].sum()
           popular_items = item_totals[item_totals > 100]
           print(popular_items)
```

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 [33]: # Solution 1
df["item_revenue"] = df["item_price"] * df["quantity"]
rpd = df.groupby('order_id')['item_revenue'].sum()
ar = rpd.mean()
print(f"Averege Revenue Order: ${ar:.2f}")
```

Averege Revenue Order: \$21.39