

In [8]: `# Exercise 1`

```
import os
os.getcwd()

os.chdir("C:\\Users\\kriti\\Downloads\\Lab8")

import numpy as np
import pandas as pd

chipo = pd.read_csv("chipotle.txt", delimiter = "\t")
```

In [9]: `# Exercise 2(a)`

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

	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
5	3	1	Chicken Bowl
6	3	1	Side of Chips
7	4	1	Steak Burrito
8	4	1	Steak Soft Tacos
9	5	1	Steak Burrito

  

	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
5	[Fresh Tomato Salsa (Mild), [Rice, Cheese, Sou...	\$10.98
6	NaN	\$1.69
7	[Tomatillo Red Chili Salsa, [Fajita Vegetables...	\$11.75
8	[Tomatillo Green Chili Salsa, [Pinto Beans, Ch...	\$9.25
9	[Fresh Tomato Salsa, [Rice, Black Beans, Pinto...	\$9.25

In [10]: `# Exercise 2(b)`

```
print(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
None
```

In [11]: `# Exercise 2(c)`

```
print(chipo.shape)
```

```
(4622, 5)
```

In [12]: `# Exercise 2(d)`

```
print(chipo.columns)
```

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

In [13]: `# Exercise 2(e)`

```
print(chipo.index)
```

```
RangeIndex(start=0, stop=4622, step=1)
```

In [20]: `# Exercise 3`

```
most_ordered = chipo.groupby("item_name").sum()
```

```
sortedq = most_ordered.sort_values(['quantity'], ascending = False)
print(sortedq.head(1))
```

	order_id	quantity
item_name		
6 Pack Soft Drink	52322	55
Barbacoa Bowl	53972	66
Barbacoa Burrito	74718	91
Barbacoa Crispy Tacos	5613	12
Barbacoa Salad Bowl	9708	10
	order_id	quantity
item_name		
Chicken Bowl	713926	761

C:\Users\kriti\AppData\Local\Temp\ipykernel\_14516\3480030507.py:3: FutureWarning: The default value of numeric\_only in DataFrameGroupBy.sum is deprecated. In a future version, numeric\_only will default to False. Either specify numeric\_only or select only columns which should be valid for the function.

```
most_ordered = chipo.groupby("item_name").sum()
```

In [19]: `# Exercise 4`

```
most_ordered_c = chipo.groupby("choice_description").sum()
```

```
most_ordered_c = most_ordered.sort_values(['quantity'], ascending = False)
print(most_ordered_c.head(1))
```

	order_id	quantity
item_name		
Chicken Bowl	713926	761

C:\Users\kriti\AppData\Local\Temp\ipykernel\_14516\339233465.py:3: FutureWarning: The default value of numeric\_only in DataFrameGroupBy.sum is deprecated. In a future version, numeric\_only will default to False. Either specify numeric\_only or select only columns which should be valid for the function.

```
most_ordered_c = chipo.groupby("choice_description").sum()
```

In [21]: `# Exercise 5`

```
total_items_ordered = chipo.quantity.sum()
print(total_items_ordered)
```

4972

In [23]: *# Exercise 6*

```
revenue = chipo.item_price.sum()
chipo["actual_price"] = chipo["item_price"].apply(lambda x:float(x[1:]))
chipo["revenue"] = chipo["actual_price"] * chipo['quantity']
chipo["revenue"].sum()
```

Out[23]: 39237.02

In [24]: *# Exercise 7*

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

1834

In [26]: *# Exercise 8*

```
sum_of_order = chipo.groupby("order_id").sum()

x = sum_of_order["actual_price"].mean()
print(x)
```

18.811428571428568

C:\Users\kriti\AppData\Local\Temp\ipykernel\_14516\3103887736.py:3: FutureWarning: The default value of numeric\_only in DataFrameGroupBy.sum is deprecated. In a future version, numeric\_only will default to False. Either specify numeric\_only or select only columns which should be valid for the function.

```
sum_of_order = chipo.groupby("order_id").sum()
```

In [27]: *# Exercise 9*

```
print(chipo["item_name"].nunique())
```

50

In [ ]: