

Ex1 - Filtering and Sorting 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 [3]: ch = pd.read_table("https://raw.githubusercontent.com/justmarkham/DAT8/master/data/chipotle.tsv")
ch
```

```
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
...
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

```
In [11]: ch.shape[0] , ch.shape[1]
```

```
Out[11]: (4622, 5)
```

```
In [13]: ch.describe(include="all")
```

```
Out[13]:
```

	order_id	quantity	item_name	choice_description	item_price
count	4622.000000	4622.000000	4622	3376	4622
unique	NaN	NaN	50	1043	78
top	NaN	NaN	Chicken Bowl	[Diet Coke]	\$8.75
freq	NaN	NaN	726	134	730
mean	927.254868	1.075725	NaN	NaN	NaN
std	528.890796	0.410186	NaN	NaN	NaN
min	1.000000	1.000000	NaN	NaN	NaN
25%	477.250000	1.000000	NaN	NaN	NaN
50%	926.000000	1.000000	NaN	NaN	NaN
75%	1393.000000	1.000000	NaN	NaN	NaN
max	1834.000000	15.000000	NaN	NaN	NaN

Step 4. How many products cost more than \$10.00?

```
In [17]: ch.query('item_price > 10').item_name.nunique()
```

```
Out[17]: 31
```

Step 5. What is the price of each item?

print a data frame with only two columns item_name and item_price

```
In [22]: ch[["item_name" , "item_price"]]
```

```
Out[22]:
```

	item_name	item_price
0	Chips and Fresh Tomato Salsa	2.39
1	Izze	3.39
2	Nantucket Nectar	3.39
3	Chips and Tomatillo-Green Chili Salsa	2.39
4	Chicken Bowl	16.98
...
4617	Steak Burrito	11.75
4618	Steak Burrito	11.75
4619	Chicken Salad Bowl	11.25
4620	Chicken Salad Bowl	8.75
4621	Chicken Salad Bowl	8.75

4622 rows x 2 columns

Step 6. Sort by the name of the item

```
In [25]: ch.sort_values(by="item_name")
```

```
Out[25]:
```

	order_id	quantity	item_name	choice_description	item_price
3389	1360	2	6 Pack Soft Drink	[Diet Coke]	12.98
341	148	1	6 Pack Soft Drink	[Diet Coke]	6.49
1849	749	1	6 Pack Soft Drink	[Coke]	6.49
1860	754	1	6 Pack Soft Drink	[Diet Coke]	6.49
2713	1076	1	6 Pack Soft Drink	[Coke]	6.49
...
2384	948	1	Veggie Soft Tacos	[Roasted Chili Corn Salsa, [Fajita Vegetables,...	8.75
781	322	1	Veggie Soft Tacos	[Fresh Tomato Salsa, [Black Beans, Cheese, Sou...	8.75
2851	1132	1	Veggie Soft Tacos	[Roasted Chili Corn Salsa (Medium), [Black Bea...	8.49
1699	688	1	Veggie Soft Tacos	[Fresh Tomato Salsa, [Fajita Vegetables, Rice,...	11.25
1395	567	1	Veggie Soft Tacos	[Fresh Tomato Salsa (Mild), [Pinto Beans, Rice...	8.49

4622 rows x 5 columns

Step 7. What was the quantity of the most expensive item ordered?

```
In [32]: ch.sort_values(by = "item_price", ascending = False).head(1)
```

```
Out[32]:
```

	order_id	quantity	item_name	choice_description	item_price
3598	1443	15	Chips and Fresh Tomato Salsa	NaN	44.25

Step 8. How many times was a Veggie Salad Bowl ordered?

```
In [33]: ch[ch.item_name == 'Veggie Salad Bowl'].shape[0]
```

```
Out[33]: 18
```

Step 9. How many times did someone order more than one Canned Soda?

```
In [34]: ch[(ch.item_name == 'Canned Soda') & (ch.quantity>1)].shape[0]
```

```
Out[34]: 20
```