In [63]:

```
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import pandas as pd
```

In [64]:

```
url = 'https://raw.githubusercontent.com/justmarkham/DAT8/master/data/chipotle.tsv'
```

In [74]:

```
df = pd.read_csv(url, sep = '\t',low_memory = False)
df.head()
```

Out[74]:

	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

In [66]:

```
# print total records and type of variables
```

In [67]:

```
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
```

In [68]:

```
#Q1: Which was the most ordered item? and How many items were ordered?
```

In [69]:

```
c = df.groupby("item_name").sum()
c = c.sort_values(["quantity"], ascending = False)
c.head()
```

Out[69]:

	order_id	quantity
item_name		
Chicken Bowl	713926	761
Chicken Burrito	497303	591
Chips and Guacamole	449959	506
Steak Burrito	328437	386

	order_id	quantity
item_name		
Canned Soft Drink	304753	351

In [70]:

```
#Q2: What was the most ordered item in the choice_description column?
```

In [71]:

```
df = df.groupby("choice_description").sum()
df = df.sort_values(["quantity"], ascending = False)
df.head(1)
```

Out[71]:

	order_id	quantity
choice_description		
[Diet Coke]	123455	159

In [72]:

```
#Q3: Turn the item price into a float
```

In [75]:

```
dollar = lambda x: float(x[1:-1])
df.item_price = df.item_price.apply(dollar)
```

In [76]:

```
df.head()
```

Out[76]:

	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

In [77]:

```
#Q3: How much was the revenue for the period in the dataset?
```

In [78]:

```
revenue = (df['quantity']* df['item_price']).sum()
print('Revenue was: ' + str(np.round(revenue,2)))
```

```
Revenue was: 39237.02
```

In [79]:

```
#Q4: print a data frame with only two columns item_name and item_price
```

In [80]:

```
df.loc[:,["item_name","item_price"]]
```

Out[80]:

	item_name	item_price
0	Chips and Fresh Tomato Salsa	2.39

	item_name	item_price
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 × 2 columns

In [81]:

```
#Q5: delete the duplicates in item_name and quantity
```

In [82]:

```
filtered = df.drop_duplicates(['item_name','quantity'])
# select only the products with quantity equals to 1
one_prod = filtered[filtered.quantity == 1]
# select only the item_name and item_price columns
price_per_item = one_prod[['item_name', 'item_price']]
# sort the values from the most to more expensive
price_per_item.sort_values(by = "item_price", ascending = True)
```

Out[82]:

	item_name	item_price
34	Bottled Water	1.09
28	Canned Soda	1.09
263	Canned Soft Drink	1.25
6	Side of Chips	1.69
40	Chips	2.15
0	Chips and Fresh Tomato Salsa	2.39
300	Chips and Tomatillo-Red Chili Salsa	2.39
191	Chips and Roasted Chili-Corn Salsa	2.39
3	Chips and Tomatillo-Green Chili Salsa	2.39
38	Chips and Tomatillo Green Chili Salsa	2.95
233	Chips and Roasted Chili Corn Salsa	2.95
111	Chips and Tomatillo Red Chili Salsa	2.95
674	Chips and Mild Fresh Tomato Salsa	3.00
1	Izze	3.39
2	Nantucket Nectar	3.39
10	Chips and Guacamole	4.45
298	6 Pack Soft Drink	6.49

	item_name	item_price
1414	Salad	7.40
510	Burrito	7.40
673	Bowl	7.40
520	Crispy Tacos	7.40
1653	Veggie Crispy Tacos	8.49
16	Chicken Burrito	8.49
1694	Veggie Salad	8.49
44	Chicken Salad Bowl	8.75
12	Chicken Soft Tacos	8.75
11	Chicken Crispy Tacos	8.75
664	Steak Salad	8.99
3750	Carnitas Salad	8.99
54	Steak Bowl	8.99
33	Carnitas Bowl	8.99
27	Carnitas Burrito	8.99
21	Barbacoa Burrito	8.99
92	Steak Crispy Tacos	9.25
237	Carnitas Soft Tacos	9.25
56	Barbacoa Soft Tacos	9.25
554	Carnitas Crispy Tacos	9.25
8	Steak Soft Tacos	9.25
250	Chicken Salad	10.98
5	Chicken Bowl	10.98
62	Veggie Bowl	11.25
57	Veggie Burrito	11.25
186	Veggie Salad Bowl	11.25
738	Veggie Soft Tacos	11.25
7	Steak Burrito	11.75
168	Barbacoa Crispy Tacos	11.75
39	Barbacoa Bowl	11.75
1132	Carnitas Salad Bowl	11.89
1229	Barbacoa Salad Bowl	11.89
606	Steak Salad Bowl	11.89

In [83]:

```
#Q6: What was the quantity of the most expensive item ordered?
```

In [84]:

```
df.sort_values(["item_price"],ascending = False).head(1)
```

Out[84]:

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

In [85]:

```
# Q7: How many times were a Veggie Salad Bowl ordered?
```

In [86]:

```
df = df[df.item_name == "Veggie Salad Bowl"]
count_row = df.shape[0]
print(count_row)
```

```
18
```

Drinks statistics

In [87]:

```
url = "https://raw.githubusercontent.com/alcor2019/justmarkham/master/data/drinks.csv"
```

In [88]:

```
df2 = pd.read_csv(url,low_memory = False)
df2.head()
```

Out[88]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	continent
0	Afghanistan	0	0	0	0.0	Asia
1	Albania	89	132	54	4.9	Europe
2	Algeria	25	0	14	0.7	Africa
3	Andorra	245	138	312	12.4	Europe
4	Angola	217	57	45	5.9	Africa

In [89]:

```
#Q8: Which continent drinks more beer on average?
```

In [90]:

```
x = df2.groupby("continent")
x.beer_servings.mean()
```

Out[90]:

```
continent
Africa            61.471698
Asia              37.045455
Europe           193.777778
North America    145.434783
Oceania           89.687500
South America    175.083333
Name: beer_servings, dtype: float64
```

In [52]:

```
#Q9: For each continent print the statistics for wine consumption.
```

In [91]:

```
df2.groupby("continent").wine_servings.describe()
```

Out[91]:

	count	mean	std	min	25%	50%	75%	max
continent								
Africa	53.0	16.264151	38.846419	0.0	1.0	2.0	13.00	233.0

	count	mean	std	min	25%	50%	75%	max
continent								
Asia	44.0	9.068182	21.667034	0.0	0.0	1.0	8.00	123.0
Europe	45.0	142.222222	97.421738	0.0	59.0	128.0	195.00	370.0
North America	23.0	24.521739	28.266378	1.0	5.0	11.0	34.00	100.0
Oceania	16.0	35.625000	64.555790	0.0	1.0	8.5	23.25	212.0
South America	12.0	62.416667	88.620189	1.0	3.0	12.0	98.50	221.0

In [ ]: