Ex2 - Getting and Knowing your Data

This time we are going to pull data directly from the internet. Special thanks to: https://github.com/justmarkham (https://github.com/justmarkham) for sharing the dataset and materials.

Step 1. Import the necessary libraries

In [30]:
```
import pandas as pd
import numpy as np
```

Step 2. Import the dataset from this address (https://raw.githubusercontent.com/justmarkham/DAT8/master/data/ch

Step 3. Assign it to a variable called chipo.

In [155]:
```
db = pd.read_csv('https://raw.githubusercontent.com/justmarkham/DAT8/master/da
db
```

Out[155]:

order_id	quantity	item_name	choice_description	item_price
1	1	Chips and Fresh Tomato Salsa	NaN	$2.39
1	1	Izze	[Clementine]	$3.39
1	1	Nantucket Nectar	[Apple]	$3.39
1	1	Chips and Tomatillo-Green Chili Salsa	NaN	$2.39
2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans...	$16.98
...	...	...	...	...
1833	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Black Beans, Sour ...	$11.75
1833	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Sour Cream, Cheese...	$11.75
1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Pinto...	$11.25
1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Lettu...	$8.75
1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Pinto...	$8.75

4622 rows × 4 columns

Step 4. See the first 10 entries

In [14]:
```
db.head(10)
```

Out[14]:

order_id	quantity	item_name	choice_description	item_price
1	1	Chips and Fresh Tomato Salsa	NaN	$2.39
1	1	Izze	[Clementine]	$3.39
1	1	Nantucket Nectar	[Apple]	$3.39
1	1	Chips and Tomatillo-Green Chili Salsa	NaN	$2.39
2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans...	$16.98
3	1	Chicken Bowl	[Fresh Tomato Salsa (Mild), [Rice, Cheese, Sou...	$10.98
3	1	Side of Chips	NaN	$1.69
4	1	Steak Burrito	[Tomatillo Red Chili Salsa, [Fajita Vegetables...	$11.75
4	1	Steak Soft Tacos	[Tomatillo Green Chili Salsa, [Pinto Beans, Ch...	$9.25
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 [18]:
```
db.shape[0]
```

Out[18]:
```
4622
```

In [31]:
```
db.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 4622 entries, 1 to 1834
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   quantity            4622 non-null   int64 
 1   item_name           4622 non-null   object
 2   choice_description  3376 non-null   object
 3   item_price          4622 non-null   object
dtypes: int64(1), object(3)
memory usage: 180.5+ KB
```

Step 6. What is the number of columns in the dataset?

In [19]:
```
db.shape[1]
```

Out[19]:
```
4
```

Step 7. Print the name of all the columns.

In [26]:
```
print(db.columns)
```

```
Index(['quantity', 'item_name', 'choice_description', 'item_price'], dtype='o
bject')
```

Step 8. How is the dataset indexed?

In [27]:
```
db.index
```

Out[27]:
```
Index([   1,    1,    1,    1,    2,    3,    3,    4,    4,    5,
       ...
       1831, 1831, 1831, 1832, 1832, 1833, 1833, 1834, 1834, 1834],
      dtype='int64', name='order_id', length=4622)
```

Step 9. Which was the most-ordered item?

In [60]:
```
x = db.groupby('item_name')
# x.first()
x = x.sum()
x = x.sort_values(['quantity'],ascending=False)
x.head(1)
```

Out[60]:

item_name	quantity	choice_description	item_price
Chicken Bowl	761	[Tomatillo-Red Chili Salsa (Hot), [Black Beans...	16.9810.98 11.258.75 8.4911.25 $8.75 ...

Step 10. For the most-ordered item, how many items were ordered?

In [74]:
```
y = db.groupby('item_name',)
y = y.sum()
y = y.sort_values(['quantity'],ascending=False)
y.head(1)
# y.columns
```

Out[74]:

item_name	quantity	choice_description	item_price
Chicken Bowl	761	[Tomatillo-Red Chili Salsa (Hot), [Black Beans...	16.9810.98 11.258.75 8.4911.25 $8.75 ...

Step 11. What was the most ordered item in the choice_description column?

In [83]:
```
x = db.groupby('choice_description').sum()
x = x.sort_values('quantity',ascending=False)
x.head(1)
```

Out[83]:

choice_description	quantity	item_name	item_price
[Diet Coke]	159	Canned SodaCanned SodaCanned Soda6 Pack Soft D...	2.181.09 1.096.49 2.181.25 1.096.4...

Step 12. How many items were orderd in total?

In [85]:
```
x = db['quantity'].sum()
print(x)
```

```
4972
```

Step 13. Turn the item price into a float

Step 13.a. Check the item price type

In [87]:
```
db.item_price.dtype
```

Out[87]:
```
dtype('O')
```

Step 13.b. Create a lambda function and change the type of item price

In [170]:
```
changer = lambda x : float(x[1:-1])
db.item_price = db.item_price.apply(changer)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[170], line 2
      1 changer = lambda x : float(x[1:-1])
----> 2 db.item_price = db.item_price.apply(changer)

File ~\anaconda3\Lib\site-packages\pandas\core\series.py:4630, in Series.appl
y(self, func, convert_dtype, args, **kwargs)
   4520 def apply(
   4521     self,
   4522     func: AggFuncType,
   (...)
   4525     **kwargs,
   4526 ) -> DataFrame | Series:
   4527     """
   4528     Invoke function on values of Series.
   4529
   (...)
   4628     dtype: float64
   4629     """
-> 4630     return SeriesApply(self, func, convert_dtype, args, kwargs).apply
()

File ~\anaconda3\Lib\site-packages\pandas\core\apply.py:1025, in SeriesApply.
apply(self)
   1022     return self.apply_str()
   1024 # self.f is Callable
-> 1025 return self.apply_standard()

File ~\anaconda3\Lib\site-packages\pandas\core\apply.py:1076, in SeriesApply.
apply_standard(self)
   1074     else:
   1075         values = obj.astype(object)._values
-> 1076         mapped = lib.map_infer(
   1077             values,
   1078             f,
   1079             convert=self.convert_dtype,
   1080         )
   1082 if len(mapped) and isinstance(mapped[0], ABCSeries):
   1083     # GH#43986 Need to do list(mapped) in order to get treated as nes
ted
   1084     #  See also GH#25959 regarding EA support
   1085     return obj._constructor_expanddim(list(mapped), index=obj.index)

File ~\anaconda3\Lib\site-packages\pandas\_libs\lib.pyx:2834, in pandas._lib
s.lib.map_infer()

Cell In[170], line 1, in <lambda>(x)
----> 1 changer = lambda x : float(x[1:-1])
      2 db.item_price = db.item_price.apply(changer)

TypeError: 'float' object is not subscriptable
```

Step 13.c. Check the item price type

In [94]:
```
db.item_price.dtype
```

Out[94]:
```
dtype('float64')
```

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

In [164]:
```
revenue = (db['quantity']* db['item_price']).sum()
print("revenue",str(np.round(revenue,2)))
```

```
revenue 39237.02
```

Step 15. How many orders were made in the period?

In [165]:
```
x = pd.read_csv('https://raw.githubusercontent.com/justmarkham/DAT8/master/dat
ans = x.order_id.value_counts().count()
print("total orders :",ans)
```

```
total orders : 1834
```

Step 16. What is the average revenue amount per order?

In [177]:
```
db['revenue'] = db['quantity']*db['item_price']
xy = db.groupby(by=['order_id']).sum()
# xy.head()
xy['revenue'].mean()
```

Out[177]:
```
21.39423118865867
```

In [ ]:
```

```

Step 17. How many different items are sold?

In [179]:
```
db['item_name'].value_counts().count()
```

Out[179]:
```
50
```