Module 2 Exercises - Explore Data

Exercise 1:

Use the pandas library to read in the file "travel-times.csv" as a dataframe. Set the dataframe's variable name as "travel_df".

Note: Download the file from here (https://notebooks.azure.com/priesterkc/projects/DABmaterial/tree/Lv1%20Data%20Analytics/datasets).

In [1]:
```
import pandas as pd
import numpy as np

travel_df = pd.read_csv("datasets_travel-times.csv")
```

In [2]:
```
travel_df.head()
```

Out[2]:

	Date	StartTime	DayOfWeek	GoingTo	Distance	MaxSpeed	AvgSpeed	AvgMovingSpeed
0	1/6/2012	16:37	Friday	Home	51.29	127.4	78.3	84.8
1	1/6/2012	08:20	Friday	GSK	51.63	130.3	81.8	88.9
2	1/4/2012	16:17	Wednesday	Home	51.27	127.4	82.0	85.8
3	1/4/2012	07:53	Wednesday	GSK	49.17	132.3	74.2	82.9
4	1/3/2012	18:57	Tuesday	Home	51.15	136.2	83.4	88.1

Exercise 2:

Use the pandas library to read in the file "income_expenses.xlsx" as a dataframe. Set the dataframe's variable name as "expense_df".

Note: Download the file from here (https://notebooks.azure.com/priesterkc/projects/DABmaterial/tree/Lv1%20Data%20Analytics/datasets).

In [3]:
```
expense_df = pd.read_excel("datasets_income_expenses.xlsx")
expense_df.head()
```

Out[3]:

	Income Range	Category	Amount	Class_level
0	$100-200	Rent	26.43	Poor
1	$100-200	Food	59.81	Poor
2	$100-200	Clothes	38.95	Poor
3	$100-200	Taxes	0.14	Poor
4	$100-200	Other Expenses and Savings	13.77	Poor

In [11]:
```
expense_df1 = pd.read_excel("datasets_income_expenses.xlsx", names = ['Student Income', 'Cash', 'Amount', 'Class_level'])
expense_df1.head()
```

Out[11]:

	Student Income	Cash	Amount	Class_level
0	$100-200	Rent	26.43	Poor
1	$100-200	Food	59.81	Poor
2	$100-200	Clothes	38.95	Poor
3	$100-200	Taxes	0.14	Poor
4	$100-200	Other Expenses and Savings	13.77	Poor

In [12]:
```
#add header after load data
expense_df.columns = ['Student Income', 'Expense', 'Amount', 'Class_level']
```

In [13]:
```
expense_df.head()
```

Out[13]:

	Student Income	Expense	Amount	Class_level
0	$100-200	Rent	26.43	Poor
1	$100-200	Food	59.81	Poor
2	$100-200	Clothes	38.95	Poor
3	$100-200	Taxes	0.14	Poor
4	$100-200	Other Expenses and Savings	13.77	Poor

Exercise 3:

Using the lists in the cell below, write code that will zip up the lists and make them into one list, then turn it into a dataframe. Next, export the dataframe as a csv file. Then try exporting the dataframe as an Excel file.

In [14]:
```
names = ['Nike','Adidas','New Balance','Puma','Reebok']
grades = [176,59,47,38,99]
```

In [15]:
```
Final = list(zip(names, grades))
```

In [16]:
```
Final
```

Out[16]:
```
[('Nike', 176),
 ('Adidas', 59),
 ('New Balance', 47),
 ('Puma', 38),
 ('Reebok', 99)]
```

In [19]:
```
#export to CSV files
df = pd.DataFrame(data = Final, columns= ['names', 'grades'])
df.to_csv('studentgrades_ex3.csv',index=False, header=False)
```

In [20]:
```
df_final = pd.read_csv("studentgrades_ex3.csv")
```

In [21]:
```
df_final.head()
```

Out[21]:

	Nike	176
0	Adidas	59
1	New Balance	47
2	Puma	38
3	Reebok	99

In [22]:
```
df = pd.DataFrame(data = Final, columns= ['names', 'grades'])
writer = pd.ExcelWriter('studentgrade_ex3.xlsx', engine='xlsxwriter')
df.to_excel(writer, sheet_name='sheet1')
writer.save()
```

In [23]:
```
df_excel = pd.read_excel("studentgrade_ex3.xlsx")
df_excel.head()
```

Out[23]:

	names	grades
0	Nike	176
1	Adidas	59
2	New Balance	47
3	Puma	38
4	Reebok	99

Exercise 4:

What columns are in the travel_df dataframe? What columns are in the expense_df dataframe?

In [24]:
```
travel_df.columns
```

Out[24]:
```
Index(['Date', 'StartTime', 'DayOfWeek', 'GoingTo', 'Distance', 'MaxSpeed',
       'AvgSpeed', 'AvgMovingSpeed', 'FuelEconomy', 'TotalTime', 'MovingTim
e',
       'Take407All', 'Comments'],
      dtype='object')
```

In [25]:
```
expense_df.columns
```

Out[25]:
```
Index(['Student Income', 'Expense', 'Amount', 'Class_level'], dtype='object')
```

Exercise 5:

Using the expense_df dataframe, sum the expense amount using the group_by function by income range.

In [33]:
```
expense_df.head(35)
```

Out[33]:

	Student Income	Expense	Amount	Class_level
0	$100-200	Rent	26.43	Poor
1	$100-200	Food	59.81	Poor
2	$100-200	Clothes	38.95	Poor
3	$100-200	Taxes	0.14	Poor
4	$100-200	Other Expenses and Savings	13.77	Poor
5	$200-300	Rent	54.88	Poor
6	$200-300	Food	117.24	Poor
7	$200-300	Clothes	57.37	Poor
8	$200-300	Taxes	9.98	Poor
9	$200-300	Other Expenses and Savings	9.98	Poor
10	$300-400	Rent	77.21	Fair
11	$300-400	Food	144.34	Fair
12	$300-400	Clothes	60.42	Fair
13	$300-400	Taxes	15.11	Fair
14	$300-400	Other Expenses and Savings	38.60	Fair
15	$400-500	Rent	78.09	Fair
16	$400-500	Food	160.51	Fair
17	$400-500	Clothes	65.07	Fair
18	$400-500	Taxes	23.86	Fair
19	$400-500	Other Expenses and Savings	106.29	Fair
20	$500-750	Rent	71.11	Comfortable
21	$500-750	Food	169.57	Comfortable
22	$500-750	Clothes	92.99	Comfortable
23	$500-750	Taxes	37.35	Comfortable
24	$500-750	Other Expenses and Savings	185.98	Comfortable
25	$750-1000	Rent	0.00	Comfortable
26	$750-1000	Food	325.60	Comfortable
27	$750-1000	Clothes	167.20	Comfortable
28	$750-1000	Taxes	70.40	Comfortable
29	$750-1000	Other Expenses and Savings	316.80	Comfortable
30	$1000+	Rent	0.00	Well To-Do
31	$1000+	Food	326.25	Well To-Do
32	$1000+	Clothes	180.00	Well To-Do
33	$1000+	Taxes	50.62	Well To-Do
34	$1000+	Other Expenses and Savings	568.13	Well To-Do

In [35]:
```
expense_df.describe()
```

Out[35]:

	Amount
count	35.000000
mean	106.287143
std	120.399752
min	0.000000
25%	31.890000
50%	65.070000
75%	152.425000
max	568.130000

In [34]:
```
#culculate sum of all value in "amount" column, using the group_by function by income range.
expense_df['Amount'].groupby(expense_df['Student Income']).sum()
```

Out[34]:
```
Student Income
$100-200      139.10
$1000+       1125.00
$200-300      249.45
$300-400      335.68
$400-500      433.82
$500-750      557.00
$750-1000     880.00
Name: Amount, dtype: float64
```

Exercise 6:

Using the travel_df dataframe and pivot_table function, get the average total time by day of the week and direction traveled (Home/GSK).

In [36]:
```
travel_df.head()
```

Out[36]:

	Date	StartTime	DayOfWeek	GoingTo	Distance	MaxSpeed	AvgSpeed	AvgMovingSpeed
0	1/6/2012	16:37	Friday	Home	51.29	127.4	78.3	84.8
1	1/6/2012	08:20	Friday	GSK	51.63	130.3	81.8	88.9
2	1/4/2012	16:17	Wednesday	Home	51.27	127.4	82.0	85.8
3	1/4/2012	07:53	Wednesday	GSK	49.17	132.3	74.2	82.9
4	1/3/2012	18:57	Tuesday	Home	51.15	136.2	83.4	88.1

In [37]:
```
#pd.pivot_table?
```

In [38]:
```
# .pivot function in Panda # by default .pivot will calculate a mean
pd.pivot_table(travel_df, index=['DayOfWeek', 'GoingTo'],values=['TotalTime'])
```

Out[38]:

		TotalTime
DayOfWeek	**GoingTo**	
Friday	**GSK**	37.628571
	Home	38.238462
Monday	**GSK**	44.747368
	Home	41.725000
Thursday	**GSK**	40.204167
	Home	42.345000
Tuesday	**GSK**	42.079167
	Home	42.962500
Wednesday	**GSK**	42.087500
	Home	44.300000

Exercise 7:

Choose either the travel_df or expense_df and do some exploratory analysis.

In [39]:
```
travel_df.head()
```

Out[39]:

	Date	StartTime	DayOfWeek	GoingTo	Distance	MaxSpeed	AvgSpeed	AvgMovingSpeed
0	1/6/2012	16:37	Friday	Home	51.29	127.4	78.3	84.8
1	1/6/2012	08:20	Friday	GSK	51.63	130.3	81.8	88.9
2	1/4/2012	16:17	Wednesday	Home	51.27	127.4	82.0	85.8
3	1/4/2012	07:53	Wednesday	GSK	49.17	132.3	74.2	82.9
4	1/3/2012	18:57	Tuesday	Home	51.15	136.2	83.4	88.1

In [40]:
```
travel_df.pivot_table(index = ["GoingTo"],  values = ["MaxSpeed"])
```

Out[40]:

	MaxSpeed
GoingTo	
GSK	127.235238
Home	127.966000