Worksheet 4

This worksheet is due Monday night of Week 3. You are encouraged to work in groups of up to 3 total students, but each student should submit their own file. (It's fine for everyone in the group to upload the same file.)

These questions refer to the attached vending machines csv file, `vend.csv`.

Goal

The goal of this worksheet is to make a 8-by-3 pandas DataFrame. The columns for this DataFrame will be named "Month", "transactions" and "total price". The entries will correspond to the number of transactions in that month, as well as the combined price for all those transactions.

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

- Load the attached `vend.csv` dataset using `pd.read_csv`, and store it with the variable name `df`.

In [2]:
```
df = pd.read_csv("../Data/vend.csv")
```

- Look at the first few rows of `df` using the `head` method.

In [3]:
```
df.head()
```

Out[3]:

	Status	Device ID	Location	Machine	Product	Category	Transaction	TransDa
0	Processed	VJ300320611	Brunswick Sq Mall	BSQ Mall x1366 - ATT	Red Bull - Energy Drink - Sugar Free	Carbonated	14515778905	Saturd January 20
1	Processed	VJ300320611	Brunswick Sq Mall	BSQ Mall x1366 - ATT	Red Bull - Energy Drink - Sugar Free	Carbonated	14516018629	Saturd January 20
2	Processed	VJ300320611	Brunswick Sq Mall	BSQ Mall x1366 - ATT	Takis - Hot Chilli Pepper & Lime	Food	14516018629	Saturd January 20
3	Processed	VJ300320611	Brunswick Sq Mall	BSQ Mall x1366 - ATT	Takis - Hot Chilli Pepper & Lime	Food	14516020373	Saturd January 20
4	Processed	VJ300320611	Brunswick Sq Mall	BSQ Mall x1366 - ATT	Red Bull - Energy Drink - Sugar Free	Carbonated	14516021756	Saturd January 20

- Which of the columns contains date-like values? Convert this column to actual date values (i.e., values that are recognized as dates by pandas) using the pandas function `to_datetime`. (You shouldn't need to use a for loop or anything like that, just use the entire column as an input to the pandas `to_datetime` function.) Save the resulting pandas Series as a new column in `df`, named `"Date"`.

In [4]:
```
df["Date"] = pd.to_datetime(df["TransDate"])
df.head()
```

Out[4]:

	Status	Device ID	Location	Machine	Product	Category	Transaction	TransDa
0	Processed	VJ300320611	Brunswick Sq Mall	BSQ Mall x1366 - ATT	Red Bull - Energy Drink - Sugar Free	Carbonated	14515778905	Saturd January 20
1	Processed	VJ300320611	Brunswick Sq Mall	BSQ Mall x1366 - ATT	Red Bull - Energy Drink - Sugar Free	Carbonated	14516018629	Saturd January 20
2	Processed	VJ300320611	Brunswick Sq Mall	BSQ Mall x1366 - ATT	Takis - Hot Chilli Pepper & Lime	Food	14516018629	Saturd January 20
3	Processed	VJ300320611	Brunswick Sq Mall	BSQ Mall x1366 - ATT	Takis - Hot Chilli Pepper & Lime	Food	14516020373	Saturd January 20
4	Processed	VJ300320611	Brunswick Sq Mall	BSQ Mall x1366 - ATT	Red Bull - Energy Drink - Sugar Free	Carbonated	14516021756	Saturd January 20

- Get the corresponding month name for each date, by using the `dt` accessor and the method `month_name` on the new "Date" column. Save this as a new column in `df`, named `"Month"`.

In [5]:
```
df["Month"] = df["Date"].dt.month_name()
df.head()
```

Out[5]:

	Status	Device ID	Location	Machine	Product	Category	Transaction	TransDa
0	Processed	VJ300320611	Brunswick Sq Mall	BSQ Mall x1366 - ATT	Red Bull - Energy Drink - Sugar Free	Carbonated	14515778905	Saturd January 20
1	Processed	VJ300320611	Brunswick Sq Mall	BSQ Mall x1366 - ATT	Red Bull - Energy Drink - Sugar Free	Carbonated	14516018629	Saturd January 20
2	Processed	VJ300320611	Brunswick Sq Mall	BSQ Mall x1366 - ATT	Takis - Hot Chilli Pepper & Lime	Food	14516018629	Saturd January 20
3	Processed	VJ300320611	Brunswick Sq Mall	BSQ Mall x1366 - ATT	Takis - Hot Chilli Pepper & Lime	Food	14516020373	Saturd January 20
4	Processed	VJ300320611	Brunswick Sq Mall	BSQ Mall x1366 - ATT	Red Bull - Energy Drink - Sugar Free	Carbonated	14516021756	Saturd January 20

- Write a function `month_count` which takes as input a month name like "March" and as output returns how many transactions occurred in this month (use Boolean indexing... there are other ways, but we'll use a similar approach below to get the total pr).

For example, the output for "March" should be `633`.

In [6]:
```
def month_count(month_name):
    return np.sum(df["Month"] == month_name)
```

- Write a function `total_price` which takes as input a month name like "March" and as output returns the sum of the "RPrice" value for all transactions occuring in that month.

For example, the output for "March" should be `1117.75`.

In [7]:
```
def total_price(month_name):
    return np.sum(df["RPrice"][df["Month"] == month_name])
```

- Make an empty DataFrame, that will eventually be the submission for this worksheet. Use `pd.DataFrame()`, nothing inside the parentheses. Name this empty DataFrame `df_out`.

In [8]:
```
df_out = pd.DataFrame()
```

- Put a "Month" column in `df_out` containing the months from `df`, each listed one time. (Use `df["Month"].unique()` to get each value one time.)

In [9]:
```
df_out["Month"] = df["Month"].unique()
```

- Apply the `month_count` function to each value in `df_out["Month"]` by calling `df_out["Month"].map(month_count)`. Store the result as a new column in `df_out` named `"transactions"`.

In [10]:
```
df_out["transactions"] = df_out["Month"].map(month_count)
```

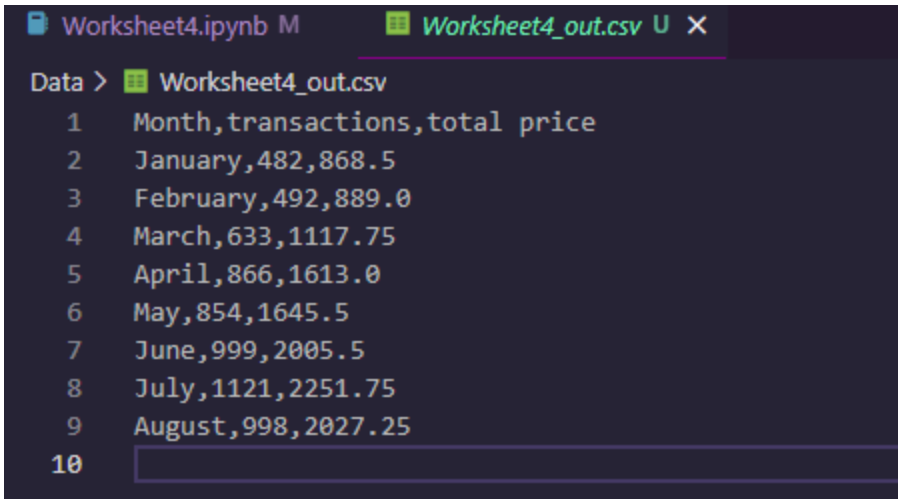
- Using the same strategy, make a column named `"total price"` in `df_out`, obtained by applying the `total_price` function to each entry in the `"Month"` column of `df_out`.

In [11]:
```
df_out["total price"] = df_out["Month"].map(total_price)
```

- The resulting DataFrame `df_out` should have 8 rows and 3 columns. Save it to a file named `df_out.csv` using the pandas DataFrame method `to_csv`. (You need to give the desired file name as the first argument to the `to_csv` method.) Specify the keyword argument `index=False` so that the numbers `0` through `7` from the index are not stored.

In [12]:
```
df_out.to_csv("../Data/Worksheet4_out.csv", index=False)
```

- Submit this csv file on Canvas.


Worksheet4.ipynb M
Worksheet4_out.csv U
Data > Worksheet4_out.csv
1 Month,transactions,total price
2 January,482,868.5
3 February,492,889.0
4 March,633,1117.75
5 April,866,1613.0
6 May,854,1645.5
7 June,999,2005.5
8 July,1121,2251.75
9 August,998,2027.25
10