Final Project - Analyzing Sales Data

Date: 8 February 2022

Author: Jeerapa Tamnitra

Course: `Pandas Foundation`

```
# import data
import pandas as pd
df = pd.read_csv("sample-store.csv")
```

```
# preview top 5 rows
df.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City
0	1	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Hend
1	2	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Hend
2	3	CA-2019-138688	6/12/2019	6/16/2019	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Ange
3	4	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Laude
4	5	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Laude

5 rows × 27 columns

```
# shape of dataframe
df.shape
```

```
(9994, 21)
```

```
# see data frame information using .info()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Row ID          9994 non-null   int64  
 1   Order ID        9994 non-null   object 
 2   Order Date      9994 non-null   object 
 3   Ship Date       9994 non-null   object 
 4   Ship Mode       9994 non-null   object 
 5   Customer ID     9994 non-null   object 
 6   Customer Name   9994 non-null   object 
 7   Segment         9994 non-null   object 
 8   Country/Region  9994 non-null   object 
 9   City            9994 non-null   object 
 10  State           9994 non-null   object 
 11  Postal Code     9983 non-null   float64
 12  Region          9994 non-null   object 
 13  Product ID      9994 non-null   object 
 14  Category        9994 non-null   object 
```

We can use `pd.to_datetime()` function to convert columns 'Order Date' and 'Ship Date' to datetime.

```
# example of pd.to_datetime() function
df['oder_date'] = pd.to_datetime(df['Order Date'],format='%m/%d/%Y')
```

```
# TODO - convert order date and ship date to datetime in the original datafram
df['ship_date'] = pd.to_datetime(df['Ship Date'].head(),format='%m/%d/%Y')
```

```
# TODO - count nan in postal code column
df['Postal Code'].isna().sum()
```

```
11
```

```
# TODO - filter rows with missing values
df[df.isna().any(axis=1)].head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City
5	6	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
6	7	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
7	8	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
8	9	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
9	10	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles

5 rows × 27 columns

```
# TODO - Explore this dataset on your owns, ask your own questions
df['Orderdate_y'] = pd.to_datetime(df['Order Date']).dt.strftime('%Y')
df['Orderdate_m'] = pd.to_datetime(df['Order Date']).dt.strftime('%m')
df.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City
0	1	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Hend
1	2	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Hend
2	3	CA-2019-138688	6/12/2019	6/16/2019	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Ange
3	4	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Laude
4	5	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Laude

5 rows × 27 columns

Data Analysis Part

Answer 10 below questions to get credit from this course. Write `pandas` code to find answers.

```
# TODO 01 - how many columns, rows in this dataset
df.shape
```

```
(9994, 25)
```

```
# TODO 02 - is there any missing values?, if there is, which colunm? how many
df.isna().sum()
```

```
Row ID             0
Order ID           0
Order Date         0
Ship Date          0
Ship Mode          0
Customer ID        0
Customer Name      0
Segment            0
Country/Region     0
City               0
State              0
Postal Code       11
Region             0
Product ID         0
Category           0
Sub-Category       0
Product Name       0
Sales              0
Quantity           0
Discount           0
Profit             0
oder_date          0
ship_date       9989
Orderdate_y        0
Orderdate_m        0
dtype: int64
```

```
# TODO 03 - your friend ask for `California` data, filter it and export csv fo
df_Califonia = df[df['State'] == 'California']
df_Califonia.to_csv('Califonia.csv')
```

```
# TODO 04 - your friend ask for all order data in `California` and `Texas` in
df_Cal_Tex_in2017 = df[((df['State'] == 'California') | (df['State'] == 'Texas
df_Cal_Tex_in2017.to_csv('California_Texas in2017.csv')
```

```
# TODO 05 - how much total sales, average sales, and standard deviation of sal
df[df['Orderdate_y'] == '2017'].groupby('Orderdate_y')['Sales'].agg(['sum','me
```

	sum	mean	std
Orderdate_y			
2017	484247.4981	242.974159	754.053357

```
# TODO 06 - which Segment has the highest profit in 2018
df[df['Orderdate_y'] =="2018"].groupby(['Orderdate_y','Segment'])['Profit']\
        .agg('sum').sort_values(ascending=False).head(1).reset_index()
```

	Orderdate_y	Segment	Profit
0	2018	Consumer	28460.1665

```
# TODO 07 - which top 5 States have the least total sales between 15 April 201
df[(df['oder_date'] >= '2019-04-15') & (df['oder_date'] <= '2019-12-31')]\
        .groupby('State')['Sales'].agg('sum').sort_values().head(5)
```

```
State
New Hampshire          49.05
New Mexico            64.08
District of Columbia    117.07
Louisiana            249.80
South Carolina          502.48
Name: Sales, dtype: float64
```

```
# TODO 08 - what is the proportion of total sales (%) in West + Central in 201
Total_sale = df[df['Orderdate_y'] == '2019'] ['Sales'].sum()
Total_sale

Total_sale_WC = df[((df['Region']=='West') | (df['Region']=='Central')) & (df[
Total_sale_WC

result =(Total_sale_WC /Total_sale)*100

result
```

```
54.97479891837763
```

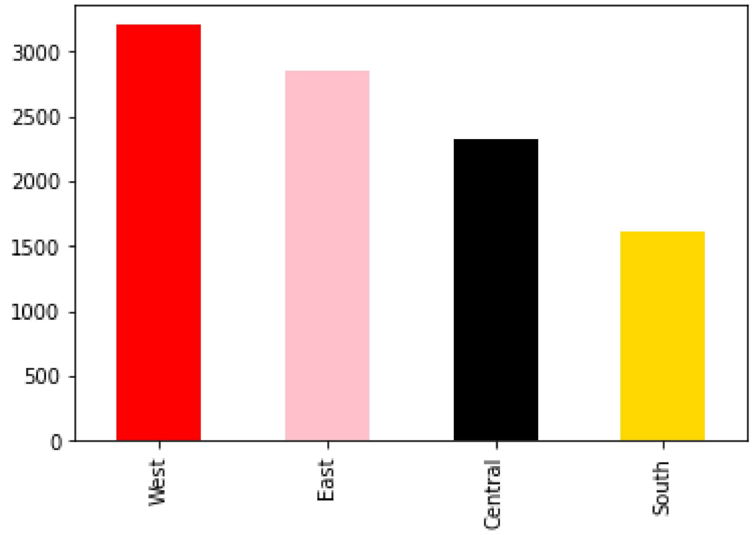
```
# TODO 09 - find top 10 popular products in terms of number of orders vs. tota

y19_20 =df.query('Orderdate_y == ["2019","2020"]')
y19_20

y19_20.groupby(['Product Name'])['Sales'].agg(['sum']).sort_values(ascending=F
```

	sum
Product Name	
Canon imageCLASS 2200 Advanced Copier	61599.824
Hewlett Packard LaserJet 3310 Copier	16079.732
3D Systems Cube Printer, 2nd Generation, Magenta	14299.890
GBC Ibimaster 500 Manual ProClick Binding System	13621.542
GBC DocuBind TL300 Electric Binding System	12737.258
GBC DocuBind P400 Electric Binding System	12521.108
Samsung Galaxy Mega 6.3	12263.708
HON 5400 Series Task Chairs for Big and Tall	11846.562
Martin Yale Chadless Opener Electric Letter Opener	11825.902
Global Troy Executive Leather Low-Back Tilter	10169.894

```
# TODO 10 - plot at least 2 plots, any plot you think interesting :)
df['Region'].value_counts().plot(kind = 'bar',color =['Red','Pink','Black','go
```

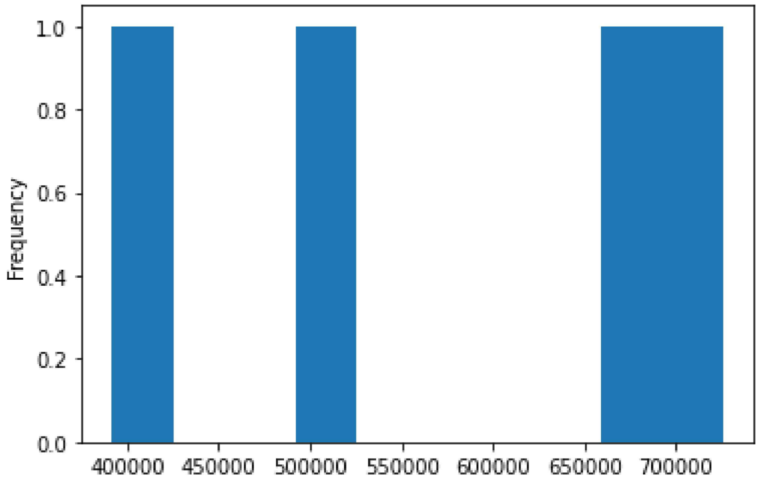
```
<AxesSubplot:>
```

Download



```
df.groupby('Region') ['Sales'].sum().plot(kind = 'hist');
```

Download



```
# TODO Bonus - use np.where() to create new column in dataframe to help you an
import numpy as np

# which sales are more than mean sales?
sales_avg = np.mean(df['Sales']).round()

df['More Mean Sales'] = np.where(df["Sales"] >= sales_avg , True, False)
df.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City
0	1	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Hend
1	2	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Hend
2	3	CA-2019-138688	6/12/2019	6/16/2019	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Ange
3	4	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Laude
4	5	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Laude

5 rows × 27 columns