Final Project - Analyzing Sales Data

Date: 25 February 2023

Author: Nutchanon Chaiyakul

Course: `Pandas Foundation`

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

```
df
```

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

...	...	...	...	...	...	...	...	...	...
9989	9990	CA-2017-110422	1/21/2017	1/23/2017	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United
9990	9991	CA-2020-121258	2/26/2020	3/3/2020	Standard Class	DB-13060	Dave Brooks	Consumer	United
9991	9992	CA-2020-121258	2/26/2020	3/3/2020	Standard Class	DB-13060	Dave Brooks	Consumer	United
9992	9993	CA-2020-121258	2/26/2020	3/3/2020	Standard Class	DB-13060	Dave Brooks	Consumer	United
9993	9994	CA-2020-119914	5/4/2020	5/9/2020	Second Class	CC-12220	Chris Cortes	Consumer	United

9994 rows × 21 columns

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

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

5 rows × 21 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
pd.to_datetime(df['Order Date'].head(), format='%m/%d/%Y')
```

```
0   2019-11-08
1   2019-11-08
2   2019-06-12
3   2018-10-11
4   2018-10-11
Name: Order Date, dtype: datetime64[ns]
```

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

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

```
11
```

```
# TODO - filter rows with missing values

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

```
# TODO - Explore this dataset on your owns, ask your own questions
df.groupby('City')['Profit'].sum().sort_values(ascending=False).head(3)
```

```
City
New York City    62036.9837
Los Angeles      30440.7579
Seattle          29156.0967
Name: Profit, dtype: float64
```

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.count()
import pandas as pd
```

```
df
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region
0	1	CA-2019-152156	2019-11-08	2019-11-08	Second Class	CG-12520	Claire Gute	Consumer	United States
1	2	CA-2019-152156	2019-11-08	2019-11-08	Second Class	CG-12520	Claire Gute	Consumer	United States
2	3	CA-2019-138688	2019-06-12	2019-06-12	Second Class	DV-13045	Darrin Van Huff	Corporate	United States
3	4	US-2018-108966	2018-10-11	2018-10-11	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States
4	5	US-2018-108966	2018-10-11	2018-10-11	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States
...	...	...	...	...	...	...	...	...	...
9989	9990	CA-2017-110422	2017-01-21	2017-01-21	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States
9990	9991	CA-2020-121258	2020-02-26	2020-02-26	Standard Class	DB-13060	Dave Brooks	Consumer	United States
9991	9992	CA-2020-121258	2020-02-26	2020-02-26	Standard Class	DB-13060	Dave Brooks	Consumer	United States
9992	9993	CA-2020-121258	2020-02-26	2020-02-26	Standard Class	DB-13060	Dave Brooks	Consumer	United States
9993	9994	CA-2020-119914	2020-05-04	2020-05-04	Second Class	CC-12220	Chris Cortes	Consumer	United States

9994 rows × 21 columns

```
# TODO 02 - is there any missing values?, if there is, which colunm? how mar
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
dtype: int64
```

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

```
import datetime as dt
```

```
# TODO 04 - your friend ask for all order data in `California` and `Texas` i
df['Order Date'] = pd.to_datetime(df['Order Date'], format='%m/%d/%Y')
caltex2017=df[((df['State']=="California") | (df['State']=="Texas")) & (df['
caltex2017.to_csv('caltex_2017_data.csv')
```

```
# TODO 05 - how much total sales, average sales, and standard deviation of s
com_y2017=df[df['Order Date'].dt.strftime("%Y")=='2017']
sum_sales_2017=com_y2017['Sales'].sum()
avg_sales_2017=com_y2017['Sales'].mean()
sd_sales_2017=com_y2017['Sales'].std()
print(f"sum :{sum_sales_2017}")
print(f"average :{avg_sales_2017}")
print(f"sd :{sd_sales_2017}")
```

```
sum :484247.4981
average :242.97415860511794
sd :754.0533572593683
```

```
# TODO 06 - which Segment has the highest profit in 2018
com_y2018=df[df['Order Date'].dt.strftime("%Y")=='2018']
max_segment_2018=com_y2018.groupby('Segment')['Profit'].sum()
max_segment_2018.sort_values(ascending=False).head(1)
```

```
Segment
Consumer    28460.1665
Name: Profit, dtype: float64
```

```
df
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region
0	1	CA-2019-152156	2019-11-08	2019-11-08	Second Class	CG-12520	Claire Gute	Consumer	United States

1	2	CA-2019-152156	2019-11-08	2019-11-08	Second Class	CG-12520	Claire Gute	Consumer	United States
2	3	CA-2019-138688	2019-06-12	2019-06-12	Second Class	DV-13045	Darrin Van Huff	Corporate	United States
3	4	US-2018-108966	2018-10-11	2018-10-11	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States
4	5	US-2018-108966	2018-10-11	2018-10-11	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States
...	...	...	...	...	...	...	...	...	...
9989	9990	CA-2017-110422	2017-01-21	2017-01-21	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States
9990	9991	CA-2020-121258	2020-02-26	2020-02-26	Standard Class	DB-13060	Dave Brooks	Consumer	United States
9991	9992	CA-2020-121258	2020-02-26	2020-02-26	Standard Class	DB-13060	Dave Brooks	Consumer	United States
9992	9993	CA-2020-121258	2020-02-26	2020-02-26	Standard Class	DB-13060	Dave Brooks	Consumer	United States
9993	9994	CA-2020-119914	2020-05-04	2020-05-04	Second Class	CC-12220	Chris Cortes	Consumer	United States

9994 rows × 21 columns

```
# TODO 07 - which top 5 States have the least total sales between 15 April 2
com_year_2019=df.loc[(df['Order Date'] >= '2019-04-15') & (df['Order Date']
com_year_2019.groupby('State')['Sales'].sum().sort_values(ascending=True).he
```

```
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 2
com_y2019=df[df['Order Date'].dt.strftime("%Y")=='2019']
com2019_region=com_y2019.groupby('Region')['Sales'].sum().reset_index()
(com2019_region[com2019_region['Region'].isin(['West','Central'])]['Sales'].
```

```
54.97479891837764
```

```
# TODO 09 - find top 10 popular products in terms of number of orders vs. to
com_year_1920=df.loc[(df['Order Date'] >= '2019-01-01') & (df['Order Date']
top_sale_1920=com_year_1920.groupby('Product Name')['Sales'].sum().sort_valu
top_order_1920=com_year_1920.groupby('Product Name')['Order Date'].count().s

print(top_sale_1920)
print("\n")
print(top_order_1920)
```

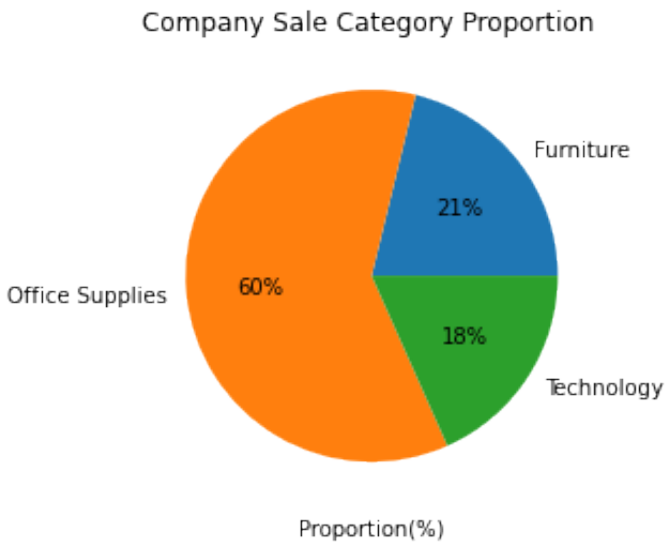
```
Product Name
Canon imageCLASS 2200 Advanced Copier                 61599.824
Hewlett Packard LaserJet 3310 Copier                  16079.732
3D Systems Cube Printer, 2nd Generation, Magenta      14299.890
GBC Ibimaster 500 Manual ProClick Binding System      13621.542
GBC DocuBind TL300 Electric 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
Name: Sales, dtype: float64


Product Name
Easy-staple paper                                   27
Staples                                             24
Staple envelope                                     22
Staples in misc. colors                             13
Staple remover                                      12
```

```
import matplotlib.pyplot as plt
import numpy as np
```

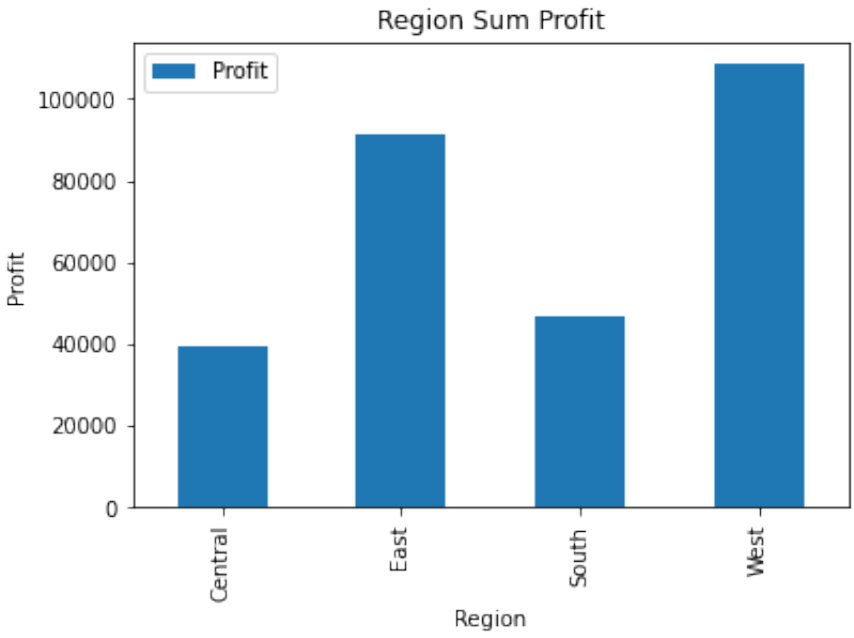
```
# TODO 10 - plot at least 2 plots, any plot you think interesting :)
df_plot=df
df_plot.groupby(['Category']).sum().plot(kind='pie', y='Quantity',autopct='%
plt.xlabel("Proportion(%)")
plt.ylabel("")
plt.title("Company Sale Category Proportion ")
plt.show()
```

```
df_plot.groupby('Region')['Profit'].sum().plot(legend=True,kind='bar')
plt.xlabel("Region")
plt.ylabel("Profit")
plt.title("Region Sum Profit")
plt.show()
```

Download



Download



```
# TODO Bonus - use np.where() to create new column in dataframe to help you
df_sub_category_profit=df_plot.groupby('Sub-Category')['Profit'].sum().reset
df_sub_category_profit['Check_Profit']=np.where(df_sub_category_profit['Prof
df_sub_category_profit
```

	Sub-Category	Profit	Check_Profit
0	Accessories	41936.6357	True
1	Appliances	18138.0054	True
2	Art	6527.7870	True
3	Binders	30221.7633	True
4	Bookcases	-3472.5560	False
5	Chairs	26590.1663	True
6	Copiers	55617.8249	True
7	Envelopes	6964.1767	True
8	Fasteners	949.5182	True
9	Furnishings	13059.1436	True
10	Labels	5546.2540	True
11	Machines	3384.7569	True
12	Paper	34053.5693	True
13	Phones	44515.7306	True
14	Storage	21278.8264	True
15	Supplies	-1189.0995	False
16	Tables	-17725.4811	False