Final Project - Analyzing Sales Data

Date: 2 November 2022

Author: Sirinthip Ngamchaluay (Fern)

Course: `Pandas Foundation`

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

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

```
# 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')
```

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

```
Convert original dataframe to datatime of order date and ship date.
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City
0	1	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Hendersor
1	2	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Hendersor
2	3	CA-2019-138688	2019-06-12	2019-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angele
3	4	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale

```
# TODO - count nan in postal code column
print("Missing Value of 'Postal Code' Column has ",df['Postal Code'].isna().sum(
```

```
Missing Value of 'Postal Code' Column has  11  rows.
```

```
# TODO - filter rows with missing values
print("Filter rows with missing values")
df[df.isna().any(axis=1)] #axis = 1 refers to columns
```

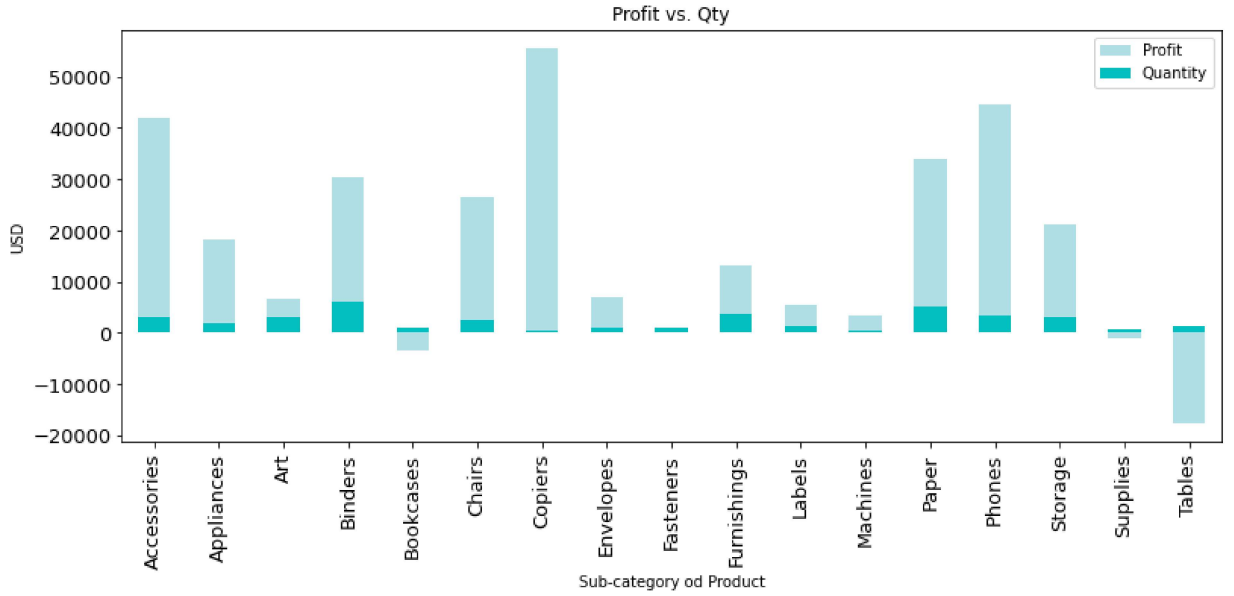
```
Filter rows with missing values
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City	...
2234	2235	CA-2020-104066	2020-12-05	2020-12-10	Standard Class	QJ-19255	Quincy Jones	Corporate	United States	Burlington	...

```
# TODO - Explore this dataset on your owns, ask your own questions
clean_df = df
p1 = clean_df.groupby('Sub-Category')['Profit'].sum().plot(kind='bar',color='powd
p2 = clean_df.groupby('Sub-Category')['Quantity'].sum().plot(kind='bar',color='c'
p1.set_ylabel("USD", fontsize=10)
p1.set_xlabel("Sub-category od Product", fontsize=10)
print('According to sales data in 2017 to 2020, which products were the most prof
print('Profitable product was Copiers Phones and Accessories.On the other hand, l
```

```
According to sales data in 2017 to 2020, which products were the most profitabl
Profitable product was Copiers Phones and Accessories.On the other hand, losing
```

Download



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
print(f"rows: {df.shape[0]} ,columns: {df.shape[1]}")
```

```
rows: 9994 ,columns: 26
```

```
# TODO 02 - is there any missing values?, if there is, which colunm? how many no
print('Column Name      Count Missing Value\n_____________________________________
print((df.isna().sum()).sort_values(ascending=False))
#Postal Code column have 11 rows contain missing values.
```

```
Column Name      Count Missing Value
_____________________________________
Postal Code       11
Row ID             0
Discount           0
Quantity           0
Sales              0
Product Name       0
Sub-Category       0
Category           0
Product ID         0
Region             0
State              0
Order ID           0
City               0
Country/Region     0
Segment            0
Customer Name      0
Customer ID        0
Ship Mode          0
```

```
# TODO 03 - your friend ask for `California` data, filter it and export csv for h
california_df = clean_df[clean_df['State']=='California']
california_df.to_csv('California_dataset.csv')
```

```
# TODO 04 - your friend ask for all order data in `California` and `Texas` in 201
California_Texas_2017 = clean_df[((clean_df['State'] =='California') | (clean_df[
California_Texas_2017.to_csv('California_Texas_2017.csv')
```

```
# TODO 05 - how much total sales, average sales, and standard deviation of sales
print('Sales in 2017\n',clean_df[(clean_df['Order Date'].dt.year == 2017)]['Sales
```

```
Sales in 2017
   index          Sales
0    sum  484247.498100
1   mean     242.974159
2    std     754.053357
```

```
# TODO 06 - which Segment has the highest profit in 2018
profit_seg = clean_df[(clean_df['Order Date'].dt.year == 2018)].groupby('Segment'
    .sort_values('Profit',ascending=False)
print('In 2018, Consumer segment has the highest profit was 28,460.17 USD')
profit_seg
```

```
In 2018, Consumer segment has the highest profit was 28,460.17 USD
```

	Segment	Profit
0	Consumer	28460.1665
1	Corporate	20688.3248
2	Home Office	12470.1124

```
# TODO 07 - which top 5 States have the least total sales between 15 April 2019
# Filter data between two dates
filter_date2019 = clean_df.loc[(clean_df['Order Date'] >= '04-15-2019') & (clear
sale_least_2019 = filter_date2019.groupby('State')['Sales'].sum().reset_index().
print('top 5 States have the least total sales between 15 April 2019 - 31 Decemb
sale_least_2019
```

```
top 5 States have the least total sales between 15 April 2019 - 31 December 20
```

	State	Sales
26	New Hampshire	49.05
28	New Mexico	64.08
7	District of Columbia	117.07
16	Louisiana	249.80
36	South Carolina	502.48

```
# TODO 08 - what is the proportion of total sales (%) in West + Central in 2019 e
sale2019_region = clean_df[(clean_df['Order Date'].dt.year == 2019)].groupby('Reg
sale2019_region['Propotion Sales(%)']=(sale2019_region['Sales']*100/(sale2019_reg
propotion_west_central = sale2019_region[(sale2019_region['Region'] == 'West') |
print('Proportion of total sales (%) in 2019\n-----------------------------------
```

```
Proportion of total sales (%) in 2019
-----------------------------------------
     Region        Sales  Propotion Sales(%)
0  Central  147429.3760               24.20
1     East  180685.8220               29.66
2    South   93610.2235               15.37
3     West  187480.1765               30.77

West + Central =  54.97 %
```

```
# TODO 09 - find top 10 popular products in terms of number of orders vs. total
sales_1920 = clean_df[(clean_df['Order Date'].dt.year == 2019) | (clean_df['Orde
top10qty = sales_1920.groupby(['Product Name','Sub-Category'])[['Quantity']].sum
```

```
top10sales = sales_1920.groupby(['Product Name','Sub-Category'])[['Sales']].sum(
display("Sales period: 2019-2020","Top 10 of number of orders",top10qty,"Top 10
```

'Sales period: 2019-2020'

'Top 10 of number of orders'

	Product Name	Sub-Category	Quantity
1412	Staples	Fasteners	124
512	Easy-staple paper	Paper	89
1406	Staple envelope	Envelopes	73
1413	Staples in misc. colors	Art	60
411	Chromcraft Round Conference Tables	Tables	59
1421	Storex Dura Pro Binders	Binders	49
1364	Situations Contoured Folding Chairs, 4/Set	Chairs	47
1532	Wilson Jones Clip & Carry Folder Binder Tool f...	Binders	44
250	Avery Non-Stick Binders	Binders	43
562	Eldon Wave Desk Accessories	Furnishings	42

'Top 10 of total sales'

	Product Name	Sub-Category	Sales
388	Canon imageCLASS 2200 Advanced Copier	Copiers	61599.824
765	Hewlett Packard LaserJet 3310 Copier	Copiers	16079.732
18	3D Systems Cube Printer, 2nd Generation, Magenta	Machines	14299.890
651	GBC Ibimaster 500 Manual ProClick Binding System	Binders	13621.542
649	GBC DocuBind TL300 Electric Binding System	Binders	12737.258
646	GBC DocuBind P400 Electric Binding System	Binders	12521.108
1310	Samsung Galaxy Mega 6.3	Phones	12263.708
746	HON 5400 Series Task Chairs for Big and Tall	Chairs	11846.562
987	Martin Yale Chadless Opener Electric Letter Op...	Supplies	11825.902
729	Global Troy Executive Leather Low-Back Tilter	Chairs	10169.894

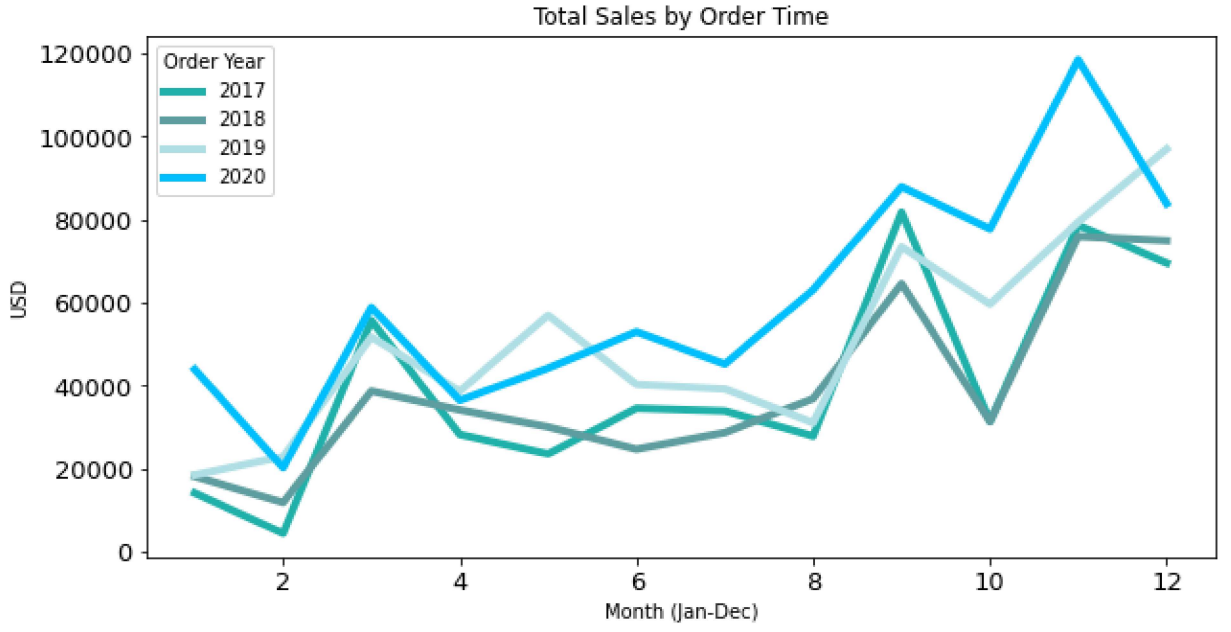
```
# TODO 10 - plot at least 2 plots, any plot you think interesting :)
clean_df
```

```
#Create New Column
clean_df['Order Year'] = clean_df['Order Date'].dt.year
clean_df['Order Month'] = clean_df['Order Date'].dt.month
clean_df['Quarter'] = clean_df['Order Date'].dt.quarter
```

```
#Prepare Data for Chart 1 & Chart2
monthly_report = clean_df.groupby(['Order Year','Order Month'])[['Sales']].sum()
Q4_sales = clean_df[clean_df['Quarter']==4]
Q4_sales_subcat = Q4_sales.groupby(['Sub-Category'])[['Sales','Profit','Quantity
```

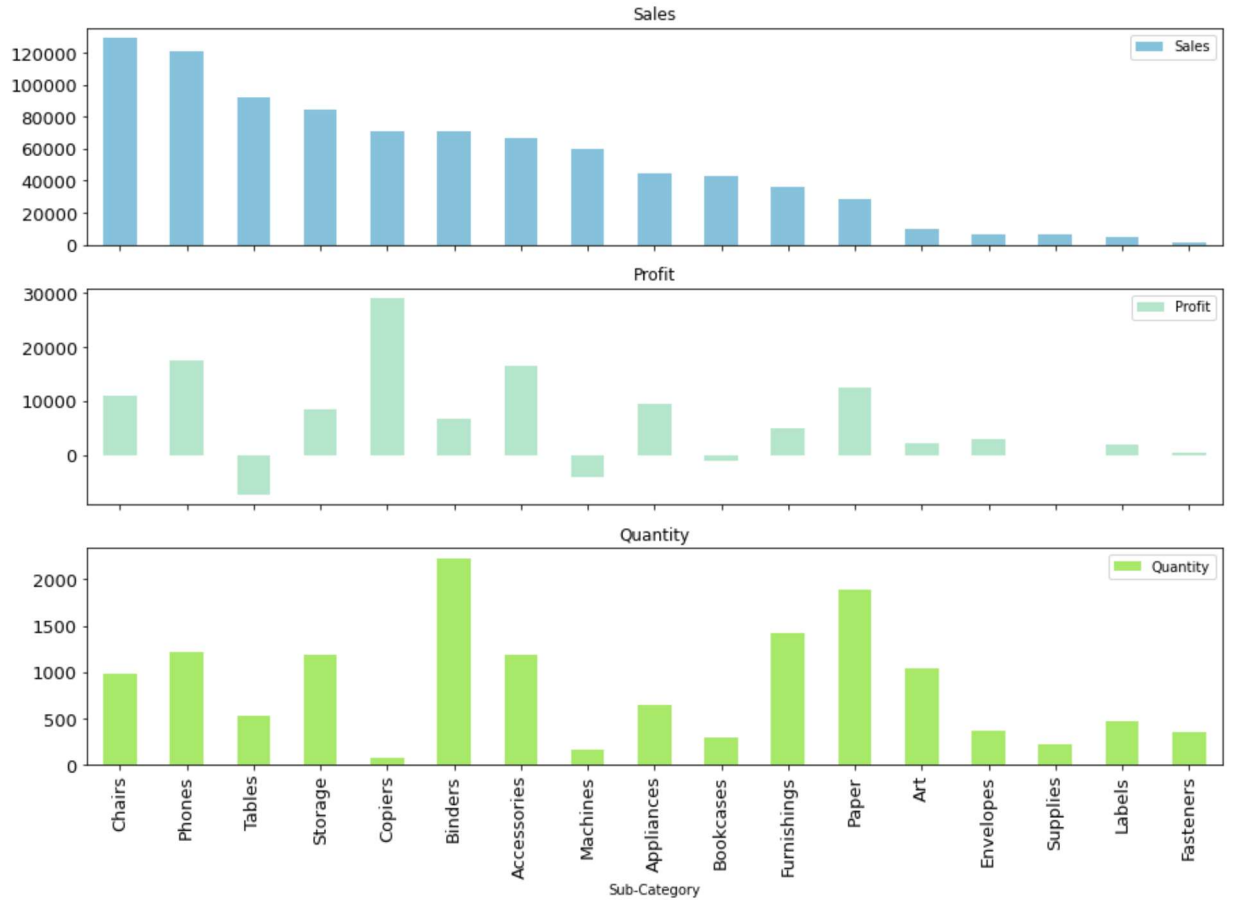
```
#Chart 1 Total Sales by Order Time >>> Q4 of every year,the most total sales.
import matplotlib.pyplot as plt
chart1_sale_year = monthly_report.pivot(index='Order Month', columns='Order Year'
chart1_sale_year.plot(color = ['lightseagreen','cadetblue','powderblue','deepskyb
plt.title('Total Sales by Order Time')
plt.xlabel('Month (Jan-Dec)')
plt.ylabel('USD')
plt.show()
```

Download



```
#Chart 2 Sub-Category best seller in Q4(Oct-Dec)
Q4_sales_subcat.plot.bar(x='Sub-Category',subplots=True,figsize=(15, 10), legend=
```

Download




```
# TODO Bonus - use np.where() to create new column in dataframe to help you answe
#use np.where() to create new column in dataframe
import numpy as np
clean_df['Discount_yn']= np.where(clean_df['Discount']==0,'no','yes')
clean_df['Profit_loss']= np.where(clean_df['Profit']>0,'Profit','Non-profit')
```

```
clean_df
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City
0	1	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson
1	2	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson
2	3	CA-2019-138688	2019-06-12	2019-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angele
3	4	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale

```
# TODO Bonus - use np.where() to create new column in dataframe to help you answe
#QA: Which Sub-Category had a discount but still lost in 2020?
compare_dc_2020 = clean_df[clean_df['Order Year']==2020].groupby(['Sub-Category',
compare_dc_2020['Label of Profit']= np.where(compare_dc_2020['Profit']>0,'Profit'
print("Compare sub-category between discount and profit groups\nSub-category had
```

```
Compare sub-category between discount and profit groups
Sub-category had a discount but still lost: Tables, Machines, Binders, Bookc
```

	Sub-Category	Discount_yn	Quantity	Sales	Profit	Label of Prof
33	Tables	yes	320	44565.5025	-10693.9664	Non-profi
32	Tables	no	70	16328.0400	2553.2717	Profi
30	Supplies	no	127	9168.4500	493.4000	Profi
31	Supplies	yes	65	6880.9600	-1448.7128	Non-profi
28	Storage	no	676	49556.1700	8461.6445	Profi
29	Storage	yes	346	20121.4480	-1058.8438	Non-profi
27	Phones	yes	749	65161.9360	1953.5210	Profi
26	Phones	no	341	40178.5800	10895.8040	Profi
24	Paper	no	1030	18565.3100	8875.5893	Profi
25	Paper	yes	672	9129.4080	3165.2541	Profi
23	Machines	yes	84	32144.2550	-7182.5017	Non-profi
22	Machines	no	37	11400.4200	4313.2861	Profi