Project - Analyzing Sales Data

Date: 20 April 2023

Author: Phatcahra Jariyasit (Boss)

This project entails the data analysis process to uncover valuable business insights.\

Dataset : "sample-store"

- import data
- check data
- analysis data
- visualization
- conclusion

```
# First we gonna import library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

print("complete")
```

```
complete
```

```
# import data

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	Hende
1	2	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Hende
2	3	CA-2019-138688	6/12/2019	6/16/2019	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angele
3	4	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauder
4	5	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauder

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 dataframe
```

```
df[["Order Date", "Ship Date"]] = df[["Order Date","Ship Date"]].apply(pd.to_dat
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	Henderso
1	2	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderso
2	3	CA-2019-138688	2019-06-12	2019-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angel
3	4	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdal
4	5	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdal
...	...	...	...	...	...	...	...	...	...	...
9989	9990	CA-2017-110422	2017-01-21	2017-01-23	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami
9990	9991	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mes
9991	9992	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mes
9992	9993	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mes
9993	9994	CA-2020-119914	2020-05-04	2020-05-09	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westminst

9994 rows × 21 columns

```
# TODO - count nan in postal code column
```

```
df["Postal Code"].isna().sum()
```

```
11
```

```
# TODO - filter rows with missing values
```

```
df.dropna()
```

	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	Henderso
1	2	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderso
2	3	CA-2019-138688	2019-06-12	2019-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angel
3	4	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdal
4	5	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdal
...	...	...	...	...	...	...	...	...	...	...
9989	9990	CA-2017-110422	2017-01-21	2017-01-23	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami
9990	9991	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mes
9991	9992	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mes
9992	9993	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mes
9993	9994	CA-2020-119914	2020-05-04	2020-05-09	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westmins

9983 rows × 21 columns

```
# TODO - Explore this dataset on your owns, ask your own questions
```

```
df.groupby('Product Name')['Profit'].sum().reset_index().sort_values(by = "Profi
```

	Product Name	Profit
404	Canon imageCLASS 2200 Advanced Copier	25199.9280
649	Fellowes PB500 Electric Punch Plastic Comb Bin...	7753.0390
804	Hewlett Packard LaserJet 3310 Copier	6983.8836
400	Canon PC1060 Personal Laser Copier	4570.9347
786	HP Designjet T520 Inkjet Large Format Printer ...	4094.9766

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, 21)
```

```
# TODO 02 - is there any missing values?, if there is, which colunm? how many nc
```

```
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 for
```

```
df[df["State"] == "California"]
df.to_csv("df_california")
```

```
# TODO 04 - your friend ask for all order data in `California` and `Texas` in 20
```

```
df[ (df["Order Date"].dt.year == 2017) & \
  (df["State"] == "California") | (df["State"] == "Texas")  ]

df.to_csv("cali_tex_2017")
```

```
# TODO 05 - how much total sales, average sales, and standard deviation of sales
```

```
df_2017 = df[ df["Order Date"].dt.year == 2017 ]
round(df_2017["Sales"].agg(["sum", "mean", "std"]), 2)
```

```
sum     484247.50
mean       242.97
std        754.05
Name: Sales, dtype: float64
```

```
# TODO 06 - which Segment has the highest profit in 2018
```

```
df_2018 = df[ df["Order Date"].dt.year == 2018 ]
df_2018.groupby("Segment")["Profit"].sum().sort_values(ascending = False).head(1
```

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

```
# TODO 07 - which top 5 States have the least total sales between 15 April 2019
```

```
newdf = df [ (df["Order Date"] >= "2019-04-15" ) & (df["Order Date"] <= "2019-12
newdf.groupby("State")["Sales"].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 2019
```

```
df2019 = df[ df["Order Date"].dt.year == 2019 ]
df2019_wes_cen = df2019[["Region", "Sales"]].query( "Region == ['West', 'Central
result = df2019_wes_cen["Sales"] / df["Sales"].sum()
print(f"{round(result*100, 2)} %")
```

```
14.58 %
```

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

```
df19_20 = df[df["Order Date"].dt.year.isin([2019, 2020])]
top_product = df19_20.groupby("Product Name")[["Quantity", "Sales"]].sum().reset
              .sort_values(by="Quantity", ascending=False).head(10)
top_product
```

	Product Name	Quantity	Sales
1412	Staples	124	462.068
512	Easy-staple paper	89	1481.728
1406	Staple envelope	73	644.936
1413	Staples in misc. colors	60	357.164
411	Chromcraft Round Conference Tables	59	7965.053
1421	Storex Dura Pro Binders	49	176.418
1364	Situations Contoured Folding Chairs, 4/Set	47	2612.064
1532	Wilson Jones Clip & Carry Folder Binder Tool f...	44	178.060
250	Avery Non-Stick Binders	43	122.128
562	Eldon Wave Desk Accessories	42	215.924

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

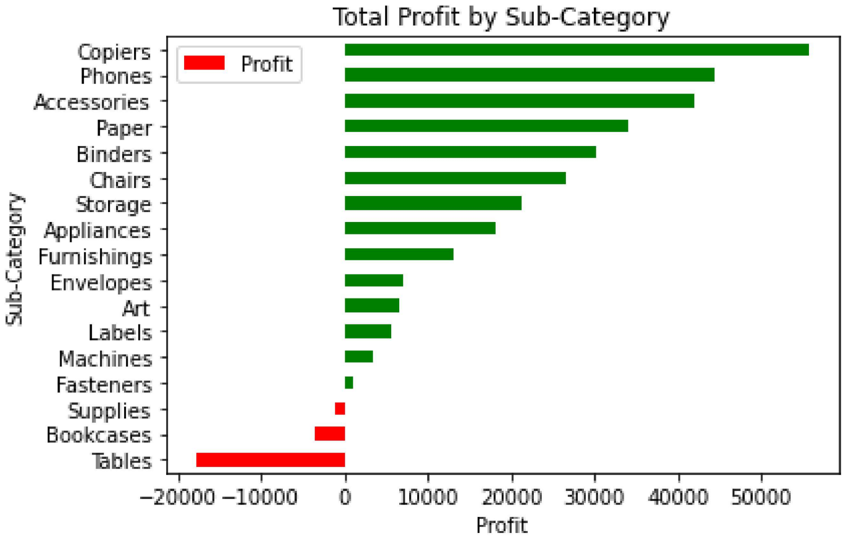
```
import matplotlib.pyplot as plt

# 1 st Plot about total profit of all sub-category.

df_subcat = df.groupby("Sub-Category")["Profit"].sum().reset_index().sort_values
df_subcat.plot(kind = "barh", x = "Sub-Category", y = "Profit",\
               color = df_subcat['Profit'].apply(lambda x: 'r' if x < 0 else 'g'

# set the axis labels and chart title
plt.xlabel("Profit")
plt.ylabel("Sub-Category")
plt.title("Total Profit by Sub-Category");
```

Download

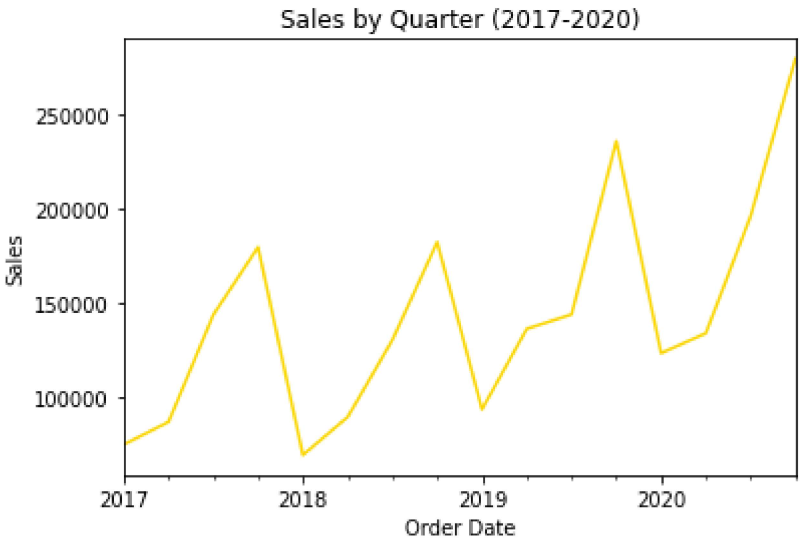


```
import pandas as pd
import matplotlib.pyplot as plt

df1 = pd.read_csv("sample-store.csv")
df1["Order Date"] = pd.to_datetime(df["Order Date"])
df1["Order Date"] = df1["Order Date"].dt.to_period('Q')
df_line = df1.groupby("Order Date")["Sales"].sum()

plot = df_line.plot(kind = "line", x = 'Sales', y = 'Order Date', color = 'gold'
plot.set_xlabel("Order Date")
plot.set_ylabel("Sales")
plot.set_title("Sales by Quarter (2017-2020)");
```

Download

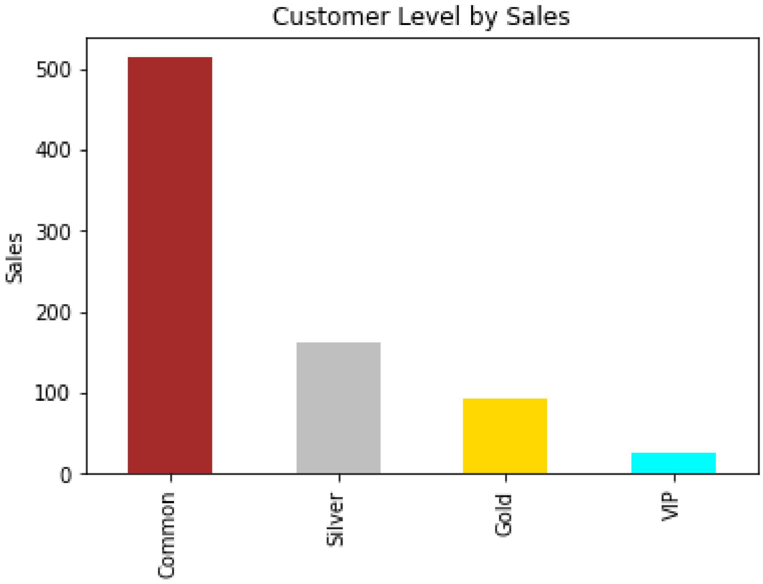


```
#  TODO Bonus - use np.where() to create new column in dataframe to help you ans
```

```
import numpy as np
df3 = df.groupby(['Customer Name'])['Sales'].sum().reset_index()
df3["Customer_Level"] = np.where(df3["Sales"] > 9000, "VIP",\
                        np.where(df3["Sales"] > 5000, "Gold",\
                        np.where(df3["Sales"] > 3000, "Silver", "Common")))

df_chart = df3["Customer_Level"].value_counts()
chart = df_chart.plot(kind = "bar", color = ["brown", "silver", "gold", "cyan"])
chart.set_ylabel("Sales")
chart.set_title("Customer Level by Sales");
```

Download


Customer Level by Sales
Sales
500
400
300
200
100
0
Common
Silver
Gold
VIP