Final Project - Analyzing Sales Data

Date: 13 April 2023

Author: Pasit Chaipatong

Course: `Pandas Foundation`

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

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

	Row	Order			Ship	Customer	Customer			

```
# 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
```

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

	Row	Order	Order	Ship	Ship	Customer	Customer					Po

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

```
False    9983
```

```
# TODO - filter rows with missing values
df[df["Postal Code"].isna()]
```

	Row	Order	Order	Ship	Ship	Customer	Customer				

```
# TODO - Explore this dataset on your owns, ask your own questions
# Top 10 Which Customer pay the most

top_customer = df.groupby("Customer Name")["Profit"]\
    .sum().reset_index().sort_values(by="Profit", ascending = False)\
    .head(10)

print(top_customer)
```

```
           Customer Name     Profit
730          Tamara Chand  8981.3239
```

```
622          Raymond Buch  6976.0959
671          Sanjit Chand  5757.4119
334          Hunter Lopez  5622.4292
6           Adrian Barton  5444.8055
757          Tom Ashbrook  4703.7883
157  Christopher Martinez  3899.8904
431         Keith Dawkins  3038.6254
35            Andy Reiter  2884.6208
194         Daniel Raglin  2869.0760
```

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

#have 21 columns and 9,994 rows
```

```
(9994, 21)
```

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

#found 11 nan values in "Postal Code" Column
```

```
Row ID          0
```

```
# TODO 03 - your friend ask for `California` data, filter it and export csv for him
California = df[df["State"] == "California"]

California.to_csv("California.csv")

California
```

	Row	Order	Order	Ship	Ship	Customer	Customer			

```
# TODO 04 - your friend ask for all order data in `California` and `Texas` in 2017

cal_tex_2017 = df[ (df["State"] == "California") \
        | (df["State"] == "Texas")\
          & (df["Order Date"].dt.year == 2017)]

cal_tex_2017.to_csv("cal_tex_2017.csv")

cal_tex_2017
```

	Row	Order	Order	Ship	Ship	Customer	Customer			

```
# TODO 05 - how much total sales, average sales, and standard deviation of sales yo
df_2017 = df[df["Order Date"].dt.year == 2017]

df_2017["Sales"].agg(["sum", "mean", "std"])
```

sum 484247.498100

```
# 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().reset_index().sort_values(by="Profit", ascending = False)\
    .head()
```

	Segment	Profit

```
# TODO 07 - which top 5 States have the least total sales between 15 April 2019 - 3
df_new = df[(df["Order Date"] >= "2019-04-15") & (df["Order Date"] <= "2019-12-31")]
df_new.groupby("State")["Sales"].sum().reset_index().sort_values(by= "Sales", ascer
```

	State	Sales

```
# TODO 08 - what is the proportion of total sales (%) in West + Central in 2019 e.g

df_2019 = df[df["Order Date"].dt.year == 2019]

west_central_2019 = df_2019.query("Region == 'West' | Region == 'Central'")

result = (west_central_2019["Sales"].sum()/df["Sales"].sum())*100

print(f"The propotion of total sales in West and Central in 2019 is {result:.2f}%")
```

```
The propotion of total sales in West and Central in 2019 is 14.58%
```

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

df_19_20 = df[df["Order Date"].dt.year.isin([2019, 2020])]

#Top10 of order
top10_order = df_19_20.groupby("Product Name")["Quantity"].agg(["sum"]).sort_values

#Top10 of sales
top10_sales = df_19_20.groupby("Product Name")["Sales"].agg(["sum"]).sort_values("s

#Merge dataframe top10_order vs. top10_sales
top10 = pd.merge(top10_order,top10_sales, left_index=True, right_index=True)

#Change columns name
top10.columns = ["Product Name", "Total", "Product Name", "Total"]

top10
```

	Product Name	Total	Product Name	Total

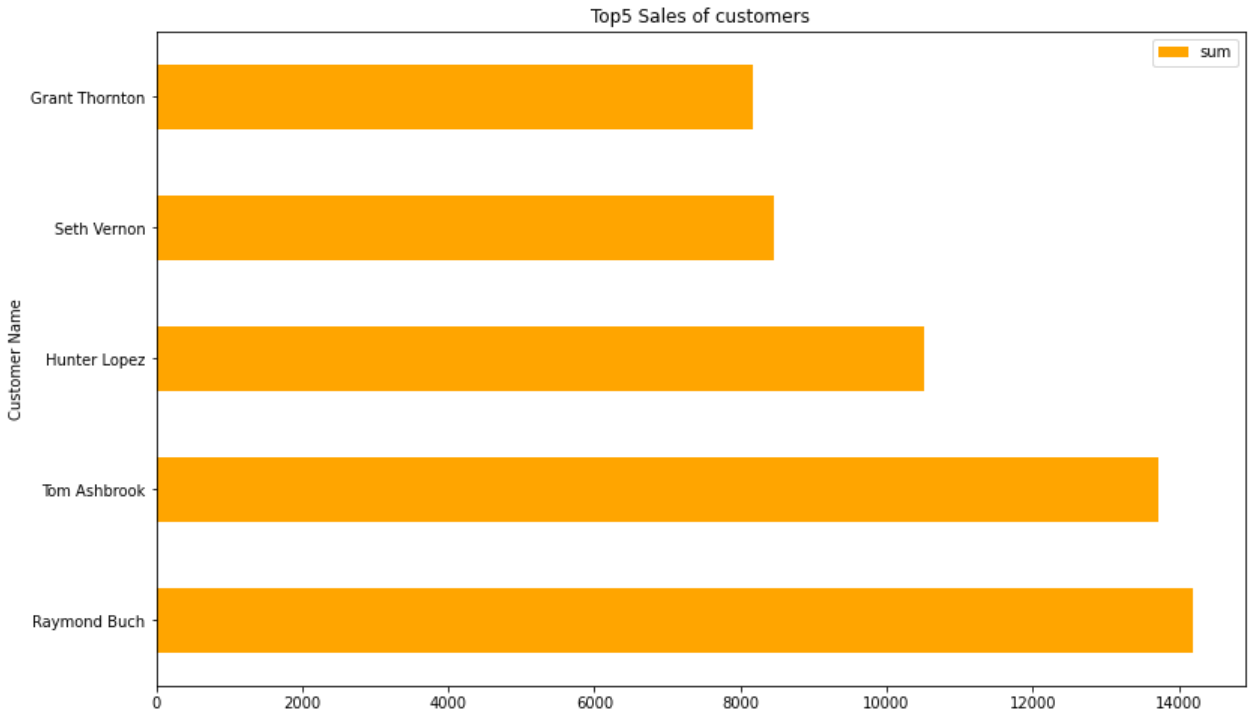
```
# TODO 10 - plot at least 2 plots, any plot you think interesting :)

# First Plot - Top 5 Sales of Customer
df_2020 = df[df["Order Date"].dt.year == 2020]

df_plot = df_2020.groupby("Customer Name")["Sales"].agg(["sum"]).sort_values("sum",

df_plot.plot.barh(figsize= [13,8], title="Top5 Sales of customers", color=["orange"
```

Download

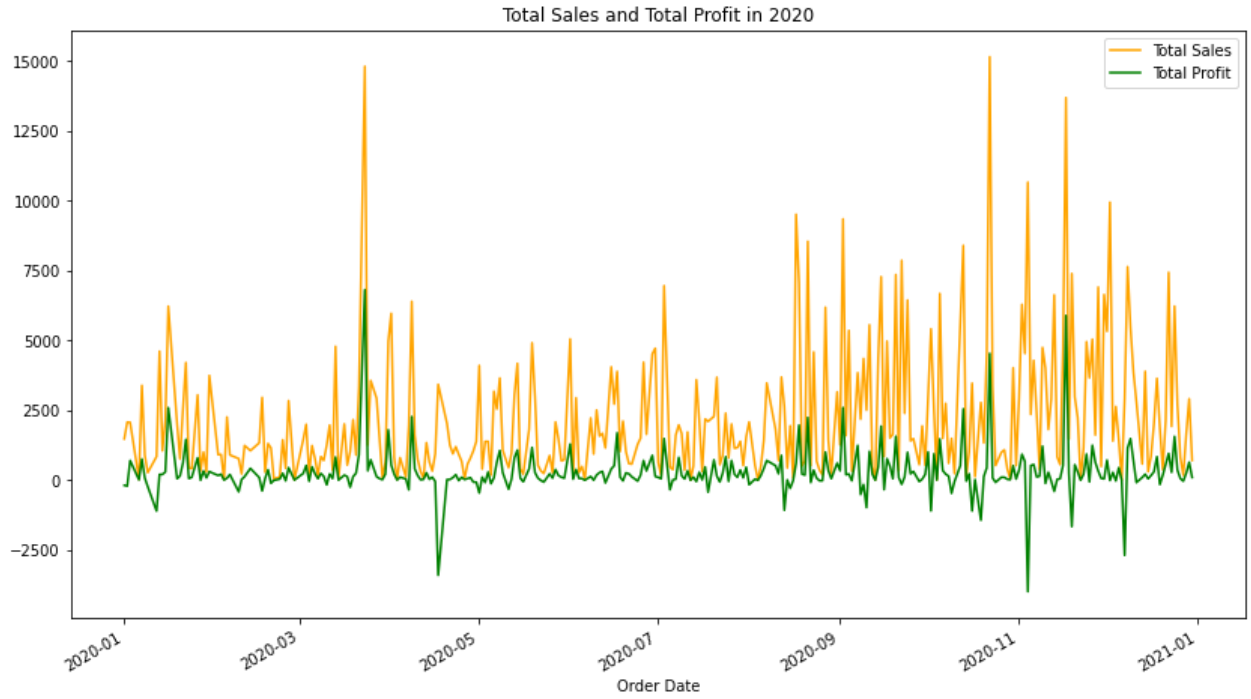



```
df_2020_sales = df_2020.groupby("Order Date")[ ["Sales", "Profit"] ].agg(["sum"]).r

df_2020_sales.columns = ["Order Date", "Total Sales", "Total Profit"]

df_2020_sales.plot(figsize=[14,8], x='Order Date', kind='line', color= ['orange', '
```

Download



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

df["Status"] = np.where(df["Profit"] >0, "Good", "Poor")

df["Status"].value_counts().plot(kind= "bar", color = ["green", "red"]);
```

Download