Final Project - Analyzing Sales Data

Date: 11 December 2022

Author: Suppanut A.

Course: `Pandas Foundation`

```
# import data
import pandas as pd
df = pd.read_csv("final project pandas 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	Henderso
1	2	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderso
2	3	CA-2019-138688	6/12/2019	6/16/2019	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles
3	4	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdal
4	5	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdal

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

df.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City	...	P C
0	1	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	4
1	2	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	4
2	3	CA-2019-138688	2019-06-12	2019-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	9
3	4	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	3
4	5	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	3

5 rows × 21 columns

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

```
11
```

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

	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	...
5274	5275	CA-2018-162887	2018-11-07	2018-11-09	Second Class	SV-20785	Stewart Visinsky	Consumer	United States	Burlington	...
8798	8799	US-2019-150140	2019-04-06	2019-04-10	Standard Class	VM-21685	Valerie Mitchum	Home Office	United States	Burlington	...
9146	9147	US-2019-165505	2019-01-23	2019-01-27	Standard Class	CB-12535	Claudia Bergmann	Corporate	United States	Burlington	...
9147	9148	US-2019-165505	2019-01-23	2019-01-27	Standard Class	CB-12535	Claudia Bergmann	Corporate	United States	Burlington	...
9148	9149	US-2019-165505	2019-01-23	2019-01-27	Standard Class	CB-12535	Claudia Bergmann	Corporate	United States	Burlington	...
9386	9387	US-2020-127292	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlington	...
9387	9388	US-2020-127292	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlington	...
9388	9389	US-2020-127292	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlington	...
9389	9390	US-2020-127292	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlington	...
9741	9742	CA-2018-117086	2018-11-08	2018-11-12	Standard Class	QJ-19255	Quincy Jones	Corporate	United States	Burlington	...

11 rows × 21 columns

```
# TODO - Explore this dataset on your owns, ask your own questions
df['State'].unique()
```

```
array(['Kentucky', 'California', 'Florida', 'North Carolina',
       'Washington', 'Texas', 'Wisconsin', 'Utah', 'Nebraska',
       'Pennsylvania', 'Illinois', 'Minnesota', 'Michigan', 'Delaware',
       'Indiana', 'New York', 'Arizona', 'Virginia', 'Tennessee',
       'Alabama', 'South Carolina', 'Oregon', 'Colorado', 'Iowa', 'Ohio',
       'Missouri', 'Oklahoma', 'New Mexico', 'Louisiana', 'Connecticut',
       'New Jersey', 'Massachusetts', 'Georgia', 'Nevada', 'Rhode Island',
       'Mississippi', 'Arkansas', 'Montana', 'New Hampshire', 'Maryland',
       'District of Columbia', 'Kansas', 'Vermont', 'Maine',
       'South Dakota', 'Idaho', 'North Dakota', 'Wyoming',
       'West Virginia'], dtype=object)
```

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)
```

Ans : 9994 rows , 21 columns

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

Ans : There's 11 missing value in column Postal Code

```
# TODO 03 - your friend ask for `California` data, filter it and export csv for h

store_california = df.query('State == "California" ')

store_california.to_csv("store_california.csv")
```

```
# TODO 04 - your friend ask for all order data in `California` and `Texas` in 201

store_cal_tex_2007 = df[ ((df['Order Date'] > '2017-01-01') & (df['Order Date'] <
    & ((df['State'] == 'California') | (df['State'] == 'Texas')) ]

store_cal_tex_2007.to_csv("store_cal_tex_2007.csv")
```

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

df[ (df['Order Date'] > '2017-01-01') & (df['Order Date'] < '2017-12-31') ]['Sale
    .agg(['sum', 'mean', 'std'])
```

```
sum     478994.228100
mean       242.038518
std        755.254933
Name: Sales, dtype: float64
```

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

df[ (df['Order Date'] > '2018-01-01') & (df['Order Date'] < '2018-12-31') ][['Seg
    .groupby('Segment').agg('sum')\
    .sort_values('Profit', ascending=False)
```

	Profit
Segment	
Consumer	28352.4385
Corporate	20357.5646
Home Office	12470.1124

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

df[ (df['Order Date'] > '2019-04-15') & (df['Order Date'] < '2019-12-31') ][['Sta
    .groupby('State').agg('sum')\
    .sort_values('Sales').head()
```

	Sales
State	
New Hampshire	49.05
New Mexico	64.08
District of Columbia	117.07
Louisiana	249.80
South Carolina	502.48

```
# TODO 07 - เปลี่ยนวิธีเขียนโค้ดจาก Agg -> sum()

df[ (df['Order Date'] > '2019-04-15') & (df['Order Date'] < '2019-12-31') ][['Sta
    .groupby('State').sum()\
    .sort_values('Sales').head()
```

	Sales
State	
New Hampshire	49.05
New Mexico	64.08
District of Columbia	117.07
Louisiana	249.80
South Carolina	502.48

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

df[ (df['Order Date'] > '2019-01-01') & (df['Order Date'] < '2019-12-31') ]\
    .query('Region == "West" | Region == "Central"')['Sales'].sum()\
    / df[ (df['Order Date'] > '2019-01-01') & (df['Order Date'] < '2019-12-31') ]
```

```
0.5495805670064725
```

ANS : 54.96%

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

##Part 1 Top Sales

top_sales = df[ (df['Order Date'] > '2019-01-01') & (df['Order Date'] < '2020-12-
    [['Product ID','Sales']]\
    .groupby('Product ID').sum()\
    .sort_values('Sales',ascending=False).head(10)

top_sales
```

	Sales
Product ID	
TEC-CO-10004722	61599.824
TEC-CO-10001449	16079.732
TEC-MA-10001047	14299.890
OFF-BI-10000545	13621.542
OFF-BI-10001359	12737.258
OFF-BI-10004995	12521.108
TEC-PH-10001459	12263.708
FUR-CH-10002024	11846.562
OFF-SU-10002881	11825.902
FUR-CH-10001215	10169.894

```
# TODO 09
##Part 1 Top Sales : join table เพื่อดูชื่อ product

id_name = df[['Product ID', 'Product Name']].value_counts().reset_index()

top_sales = pd.merge(top_sales, id_name, on='Product ID')
```

```
# TODO 09
##Part 1 Top Sales : ลบคอลั่ม 0 ทิ้ง

top_sales = top_sales.drop(0, axis=1)

top_sales
```

	Product ID	Sales	Product Name
0	TEC-CO-10004722	61599.824	Canon imageCLASS 2200 Advanced Copier
1	TEC-CO-10001449	16079.732	Hewlett Packard LaserJet 3310 Copier
2	TEC-MA-10001047	14299.890	3D Systems Cube Printer, 2nd Generation, Magenta
3	OFF-BI-10000545	13621.542	GBC Ibimaster 500 Manual ProClick Binding System
4	OFF-BI-10001359	12737.258	GBC DocuBind TL300 Electric Binding System
5	OFF-BI-10004995	12521.108	GBC DocuBind P400 Electric Binding System
6	TEC-PH-10001459	12263.708	Samsung Galaxy Mega 6.3
7	FUR-CH-10002024	11846.562	HON 5400 Series Task Chairs for Big and Tall
8	OFF-SU-10002881	11825.902	Martin Yale Chadless Opener Electric Letter Op...
9	FUR-CH-10001215	10169.894	Global Troy Executive Leather Low-Back Tilter

```
# TODO 09

##Part 2 Top Orders

top_orders = df[ (df['Order Date'] > '2019-01-01') & (df['Order Date'] < '2020-12
    [['Product ID','Order ID']]\
    .groupby('Product ID').count()\
    .sort_values('Order ID', ascending=False)\
    .head(10)

top_orders
```

	Order ID
Product ID	
TEC-AC-10003832	15
OFF-PA-10001970	13
FUR-TA-10001095	12
FUR-FU-10004864	12
FUR-CH-10003774	11
OFF-BI-10000301	10
FUR-FU-10004848	10
FUR-CH-10001146	10
OFF-ST-10001325	10
FUR-TA-10003473	9

```
# TODO 09
##Part 2 Top Orders : join table เพื่อดูชื่อ product

top_orders = pd.merge(top_orders, id_name, on='Product ID')
```

```
# TODO 09
##Part 2 Top Orders : ลบคอลั่ม 0 ทิ้ง

top_orders = top_orders.drop(0, axis=1)

top_orders
```

	Product ID	Order ID	Product Name
0	TEC-AC-10003832	15	Imation 16GB Mini TravelDrive USB 2.0 Flash Drive
1	TEC-AC-10003832	15	Logitech P710e Mobile Speakerphone
2	OFF-PA-10001970	13	Xerox 1881
3	OFF-PA-10001970	13	Xerox 1908
4	FUR-TA-10001095	12	Chromcraft Round Conference Tables
5	FUR-FU-10004864	12	Howard Miller 14-1/2" Diameter Chrome Round Wa...
6	FUR-FU-10004864	12	Eldon 500 Class Desk Accessories
7	FUR-CH-10003774	11	Global Wood Trimmed Manager's Task Chair, Khaki
8	OFF-BI-10000301	10	GBC Instant Report Kit
9	FUR-FU-10004848	10	Howard Miller 13-3/4" Diameter Brushed Chrome ...
10	FUR-FU-10004848	10	DAX Solid Wood Frames
11	FUR-CH-10001146	10	Global Value Mid-Back Manager's Chair, Gray
12	FUR-CH-10001146	10	Global Task Chair, Black
13	OFF-ST-10001325	10	Sterilite Officeware Hinged File Box
14	FUR-TA-10003473	9	Bretford Rectangular Conference Table Tops

```
# TODO 09 ลองเช็คว่า TEC-AC-10003832 ไม่ได้เป็น Top 10 Sales จริงๆ

df[ (df['Order Date'] > '2019-01-01') & (df['Order Date'] < '2020-12-31') ] \
    [df['Product ID'] == "TEC-AC-10003832" ] \
    ['Sales'].sum()
```

```
9733.798000000003

<ipython-input-25-894ec036fbdb>:3: UserWarning: Boolean Series key will be rein
  df[ (df['Order Date'] > '2019-01-01') & (df['Order Date'] < '2020-12-31') ] \
```

```
# TODO 09 สรุป

display(top_sales)
display(top_orders)
```

	Product ID	Sales	Product Name
0	TEC-CO-10004722	61599.824	Canon imageCLASS 2200 Advanced Copier
1	TEC-CO-10001449	16079.732	Hewlett Packard LaserJet 3310 Copier
2	TEC-MA-10001047	14299.890	3D Systems Cube Printer, 2nd Generation, Magenta
3	OFF-BI-10000545	13621.542	GBC Ibimaster 500 Manual ProClick Binding System
4	OFF-BI-10001359	12737.258	GBC DocuBind TL300 Electric Binding System
5	OFF-BI-10004995	12521.108	GBC DocuBind P400 Electric Binding System
6	TEC-PH-10001459	12263.708	Samsung Galaxy Mega 6.3
7	FUR-CH-10002024	11846.562	HON 5400 Series Task Chairs for Big and Tall
8	OFF-SU-10002881	11825.902	Martin Yale Chadless Opener Electric Letter Op...
9	FUR-CH-10001215	10169.894	Global Troy Executive Leather Low-Back Tilter

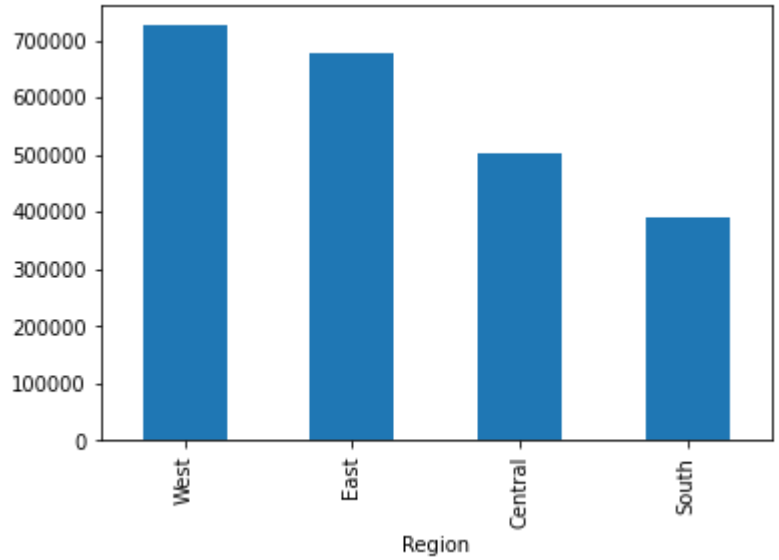
	Product ID	Order ID	Product Name
0	TEC-AC-10003832	15	Imation 16GB Mini TravelDrive USB 2.0 Flash Drive
1	TEC-AC-10003832	15	Logitech P710e Mobile Speakerphone
2	OFF-PA-10001970	13	Xerox 1881
3	OFF-PA-10001970	13	Xerox 1908
4	FUR-TA-10001095	12	Chromcraft Round Conference Tables
5	FUR-FU-10004864	12	Howard Miller 14-1/2" Diameter Chrome Round Wa...
6	FUR-FU-10004864	12	Eldon 500 Class Desk Accessories
7	FUR-CH-10003774	11	Global Wood Trimmed Manager's Task Chair, Khaki
8	OFF-BI-10000301	10	GBC Instant Report Kit
9	FUR-FU-10004848	10	Howard Miller 13-3/4" Diameter Brushed Chrome ...
10	FUR-FU-10004848	10	DAX Solid Wood Frames
11	FUR-CH-10001146	10	Global Value Mid-Back Manager's Chair, Gray
12	FUR-CH-10001146	10	Global Task Chair, Black
13	OFF-ST-10001325	10	Sterilite Officeware Hinged File Box
14	FUR-TA-10003473	9	Bretford Rectangular Conference Table Tops

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

## plot 1

df_region_sales_bar = df.groupby('Region')['Sales'].sum().sort_values(ascending=F
```

Download

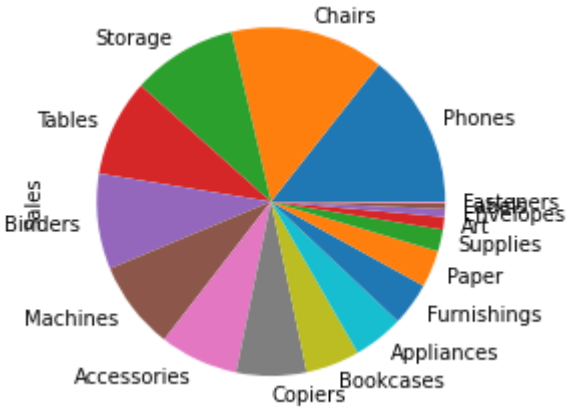



```
# TODO 10

## plot 2

df_category_sales = df.groupby('Sub-Category')['Sales'].sum().sort_values(ascendi
```

Download



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

import numpy as np

df['Profit'].mean()

df['good_business?'] = np.where(df['Profit']>=29 , "Good Business", "Bad Performa

df.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City	...	R
0	1	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	S
1	2	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	S
2	3	CA-2019-138688	2019-06-12	2019-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	V
3	4	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	S
4	5	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	S

5 rows × 22 columns