Final Project - Analyzing Sales Data

Date: 11 December 2022

Author: Suppanut A.

Course: `Pandas Foundation (DataRockie)`

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

```python
# 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

## Which states are included in this dataset?

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.

Question 1

```
# TODO 01 - how many columns, rows in this dataset
df.shape
```

```
(9994, 21)
```

Ans : 9994 rows , 21 columns

Question 2

```
# 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

Question 3

```
# 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")
```

Question 4

```
# 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")
```

Question 5

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

df[ df['Order Date'].dt.strftime('%Y') == "2017" ]['Sales'].agg(['sum', 'mean', '
```

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

Question 6

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

df[ df['Order Date'].dt.strftime('%Y') == "2018" ][['Segment','Profit']]\
    .groupby('Segment').agg('sum')\
    .sort_values('Profit', ascending=False).round(2)
```

	Profit
Segment	
Consumer	28460.17
Corporate	20688.32
Home Office	12470.11

Ans: Consumer Segment has the highest profit in 2018

Question 7

```
# 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

Question 8

```
# 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
```

```
# TODO 08 - เปลี่ยนมาใช้ dt.strftime

df[ df['Order Date'].dt.strftime('%Y') == "2019" ]\
    .query('Region == "West" | Region == "Central"')['Sales'].sum()\
    / df[ df['Order Date'].dt.strftime('%Y') == "2019" ]['Sales'].sum()
```

```
0.5497479891837763
```

ANS : 54.97%

Question 9

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

## Filter data frame 2019-2020

df_19_20 = df[ (df['Order Date'] > '2019-01-01') & (df['Order Date'] < '2020-12-3

df_19_20.head()
```

	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 Angeles	...
12	13	CA-2020-114412	2020-04-15	2020-04-20	Standard Class	AA-10480	Andrew Allen	Consumer	United States	Concord	...
13	14	CA-2019-161389	2019-12-05	2019-12-10	Standard Class	IM-15070	Irene Maddox	Consumer	United States	Seattle	...

5 rows × 21 columns

```
# TODO 09
## Top Orders
df_top_orders = df_19_20.groupby(['Product ID','Product Name'])['Order ID'].count

df_top_orders
```

	Product ID	Product Name	Order ID
0	FUR-TA-10001095	Chromcraft Round Conference Tables	12
1	FUR-CH-10003774	Global Wood Trimmed Manager's Task Chair, Khaki	11
2	OFF-BI-10000301	GBC Instant Report Kit	10
3	OFF-ST-10001325	Sterilite Officeware Hinged File Box	10
4	OFF-BI-10001989	Premium Transparent Presentation Covers by GBC	9
5	FUR-TA-10003473	Bretford Rectangular Conference Table Tops	9
6	OFF-BI-10004364	Storex Dura Pro Binders	9
7	TEC-AC-10004510	Logitech Desktop MK120 Mouse and keyboard Combo	9
8	OFF-BI-10004236	XtraLife ClearVue Slant-D Ring Binder, White, 3"	9
9	FUR-CH-10000454	Hon Deluxe Fabric Upholstered Stacking Chairs,...	9

```
# TODO 09
## Top Sales
df_top_sales = df_19_20.groupby(['Product ID','Product Name'])['Sales'].sum().sor

df_top_sales
```

	Product ID	Product Name	Sales
0	TEC-CO-10004722	Canon imageCLASS 2200 Advanced Copier	61599.82
1	TEC-CO-10001449	Hewlett Packard LaserJet 3310 Copier	16079.73
2	TEC-MA-10001047	3D Systems Cube Printer, 2nd Generation, Magenta	14299.89
3	OFF-BI-10000545	GBC Ibimaster 500 Manual ProClick Binding System	13621.54
4	OFF-BI-10001359	GBC DocuBind TL300 Electric Binding System	12737.26
5	OFF-BI-10004995	GBC DocuBind P400 Electric Binding System	12521.11
6	TEC-PH-10001459	Samsung Galaxy Mega 6.3	12263.71
7	FUR-CH-10002024	HON 5400 Series Task Chairs for Big and Tall	11846.56
8	OFF-SU-10002881	Martin Yale Chadless Opener Electric Letter Op...	11825.90
9	FUR-CH-10001215	Global Troy Executive Leather Low-Back Tilter	10169.89

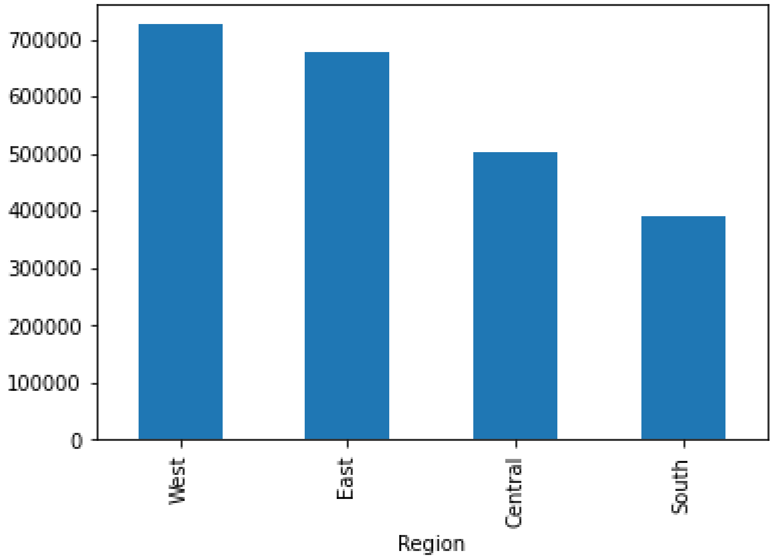
Question 10

```
# 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

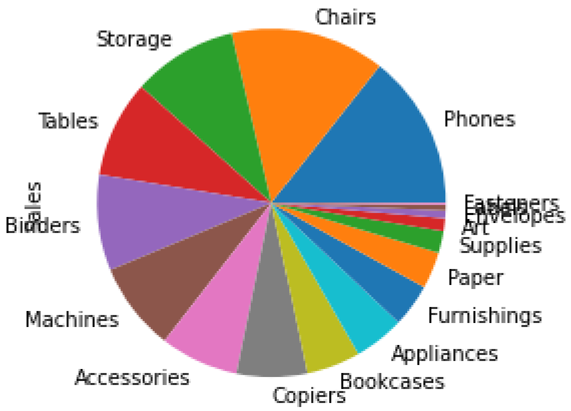


```
# TODO 10

## plot 2

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

Download



Bonus Question

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

import numpy as np

df['Profit'].mean() ## mean=29

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

df[['Order ID','Product Name','Category','Sub-Category','Sales','Profit','good_bu
```

	Order ID	Product Name	Category	Sub-Category	Sales	Profit	good_business?
0	CA-2019-152156	Bush Somerset Collection Bookcase	Furniture	Bookcases	261.9600	41.9136	Good Business
1	CA-2019-152156	Hon Deluxe Fabric Upholstered Stacking Chairs,...	Furniture	Chairs	731.9400	219.5820	Good Business
2	CA-2019-138688	Self-Adhesive Address Labels for Typewriters b...	Office Supplies	Labels	14.6200	6.8714	Bad Performance
3	US-2018-108966	Bretford CR4500 Series Slim Rectangular Table	Furniture	Tables	957.5775	-383.0310	Bad Performance
4	US-2018-108966	Eldon Fold 'N Roll Cart System	Office Supplies	Storage	22.3680	2.5164	Bad Performance