Final Project - Analyzing Sales Data

Date: 14 July 2023

Author: Chavinee Prasertpong **Course**: `Pandas Foundation`

```
# import data
import pandas as pd
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	Henderson
1	2	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson
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 Lauderdale
4	5	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale

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'].head(), format='%m/%d/%Y')
df['Ship_date'] = pd.to_datetime(df['Ship Date'].head(), format='%m/%d/%Y')
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	Henderson
1	2	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson
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 Lauderdale
4	5	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale

5 rows × 23 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	12/5/2020	12/10/2020	Standard Class	QJ-19255	Quincy Jones	Corporate	United States	Burlingt
5274	5275	CA-2018-162887	11/7/2018	11/9/2018	Second Class	SV-20785	Stewart Visinsky	Consumer	United States	Burlingt
8798	8799	US-2019-150140	4/6/2019	4/10/2019	Standard Class	VM-21685	Valerie Mitchum	Home Office	United States	Burlingt
9146	9147	US-2019-165505	1/23/2019	1/27/2019	Standard Class	CB-12535	Claudia Bergmann	Corporate	United States	Burlingt
9147	9148	US-2019-165505	1/23/2019	1/27/2019	Standard Class	CB-12535	Claudia Bergmann	Corporate	United States	Burlingt
9148	9149	US-2019-165505	1/23/2019	1/27/2019	Standard Class	CB-12535	Claudia Bergmann	Corporate	United States	Burlingt
9386	9387	US-2020-127292	1/19/2020	1/23/2020	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlingt
9387	9388	US-2020-127292	1/19/2020	1/23/2020	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlingt
9388	9389	US-2020-127292	1/19/2020	1/23/2020	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlingt
9389	9390	US-2020-127292	1/19/2020	1/23/2020	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlingt
9741	9742	CA-2018-117086	11/8/2018	11/12/2018	Standard Class	QJ-19255	Quincy Jones	Corporate	United States	Burlingt

11 rows × 23 columns

```
# TODO - Explore this dataset on your owns, ask your own questions
sales_seg = df.groupby(['Segment','Category'])['Sales'].sum().sort_values(ascending
```

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

rows,columns = df.shape
print(rows)
print(columns)
```

```
9994
23
```

```
# TODO 02 - is there any missing values?, if there is, which colunm? how many nan v
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
Order_date       9989
Ship_date        9989
dtype: int64
```

```
# TODO 03 - your friend ask for `California` data, filter it and export csv for him
cal_df = df[df['State'] == 'California']
cal_df.to_csv("California_df.csv")
```

```
# TODO 04 - your friend ask for all order data in `California` and `Texas` in 2017
df['Order Date'] = pd.to_datetime(df['Order Date'], format='%m/%d/%Y')
df2017 = df[df['Order Date'].dt.year==2017]
cal_texas_2017 = df2017.query('State == "California" | State == "Texas"')
cal_texas_2017.to_csv('Cal_texas_2017.csv')
```

```
# TODO 05 - how much total sales, average sales, and standard deviation of sales yo
df[df['Order Date'].dt.year == 2017]['Sales'].agg(['sum','mean','std'])
```

```
sum     484247.498100
mean       242.974159
std        754.053357
Name: Sales, dtype: float64
```

```
# TODO 06 - which Segment has the highest profit in 2018
data_order2018 = df[df['Order Date'].dt.year == 2018]
data_order2018.groupby('Segment')['Profit'].sum().sort_values(ascending=False)
```

```
Segment
Consumer       28460.1665
Corporate      20688.3248
Home Office    12470.1124
Name: Profit, dtype: float64
```

```
# TODO 07 - which top 5 States have the least total sales between 15 April 2019 - 3
data_order2019 = df[(df['Order Date'] >= "2019-04-15") & (df['Order Date'] <= "2019
data_order2019.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 e.g
orders_2019 = df[df['Order Date'].dt.year == 2019]
total_sales_2019 = orders_2019['Sales'].sum()

total_sales_wc = orders_2019.query('Region == "West" | Region == "Central"')['Sales
portion_wc = ((total_sales_wc/total_sales_2019)*100).round(2)
print(f"{portion_wc} %")
```

```
54.97 %
```

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

order201920 = df[(df['Order Date'] >= "2019") & (df['Order Date'] <= "2020")]
topsales = order201920.groupby(['Product Name','Quantity'])['Sales'].sum().reset_ir
    .sort_values('Sales',ascending = False).head(10)
topsales
```

	Product Name	Quantity	Sales
500	Canon imageCLASS 2200 Advanced Copier	5	17499.950
844	GBC Ibimaster 500 Manual ProClick Binding System	13	9892.740
19	3D Systems Cube Printer, 2nd Generation, Magenta	7	9099.930
1039	High Speed Automatic Electric Letter Opener	3	8842.662
1011	HP Designjet T520 Inkjet Large Format Printer ...	5	8749.950
499	Canon imageCLASS 2200 Advanced Copier	4	8399.976
497	Canon PC1060 Personal Laser Copier	5	5599.920
836	GBC DocuBind P400 Electric Binding System	4	5443.960
1037	Hewlett Packard LaserJet 3310 Copier	9	5399.910
209	Ativa V4110MDD Micro-Cut Shredder	7	4899.930

```
toporder = order201920.groupby(['Product Name','Quantity'])['Sales'].sum().reset_ir
    .sort_values('Quantity',ascending = False).head(10)
toporder
```

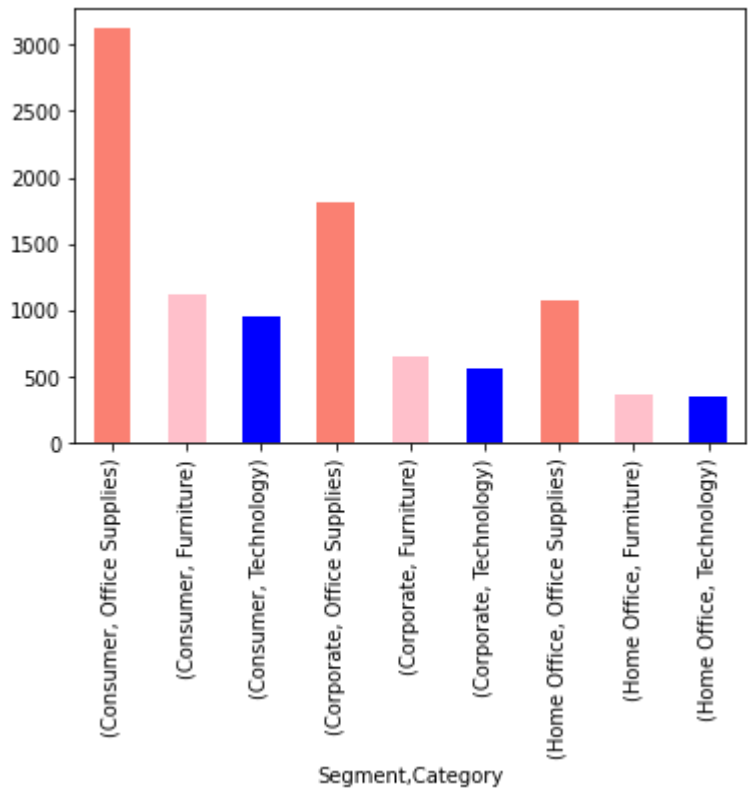
	Product Name	Quantity	Sales
1325	Longer-Life Soft White Bulbs	14	43.120
724	Electrix Architect's Clamp-On Swing Arm Lamp, ...	14	1336.440
2054	Wilson Jones Turn Tabs Binder Tool for Ring Bi...	14	53.984
1368	Metal Folding Chairs, Beige, 4/Carton	14	427.644
1173	Ibico Recycled Linen-Style Covers	14	437.472
2196	Xerox 1964	14	319.760
1641	Pressboard Covers with Storage Hooks, 9 1/2" x...	14	13.748
1978	Ultra Door Push Plate	14	54.992
1178	Iceberg Mobile Mega Data/Printer Cart	13	1564.290
1003	Great White Multi-Use Recycled Paper (20Lb. an...	13	62.192

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

df.groupby('Segment')['Category'].value_counts().plot(kind='bar',color= ['salmon','
```

```
<Axes: xlabel='Segment,Category'>
```

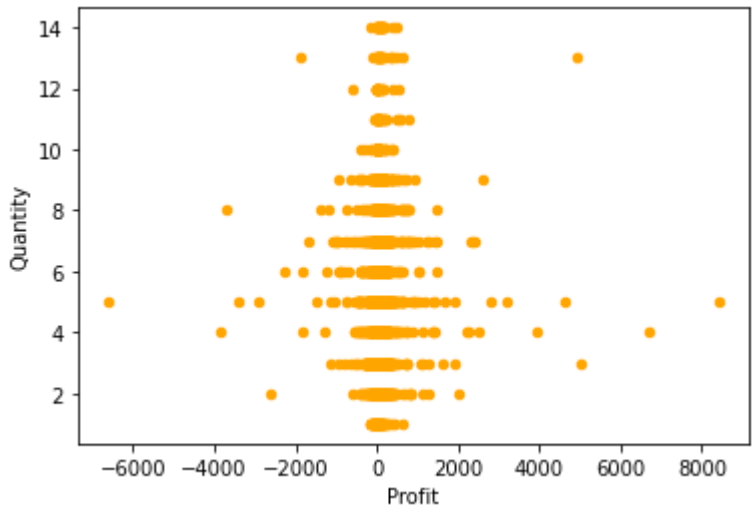
Download



```
df[['Profit','Quantity']]\
    .plot(x='Profit', y='Quantity', kind="scatter",color='orange')
```

```
<Axes: xlabel='Profit', ylabel='Quantity'>
```

Download




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

import numpy as np

df['Profit_sum'] = np.where(df['Profit']>15, "Profit","Loss")
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	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Henders
1	2	CA-2019-152156	2019-11-08	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Henders
2	3	CA-2019-138688	2019-06-12	6/16/2019	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Ange
3	4	US-2018-108966	2018-10-11	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderda
4	5	US-2018-108966	2018-10-11	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderda
...	...	...	...	...	...	...	...	...	...	...
9989	9990	CA-2017-110422	2017-01-21	1/23/2017	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami
9990	9991	CA-2020-121258	2020-02-26	3/3/2020	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Me
9991	9992	CA-2020-121258	2020-02-26	3/3/2020	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Me
9992	9993	CA-2020-121258	2020-02-26	3/3/2020	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Me
9993	9994	CA-2020-119914	2020-05-04	5/9/2020	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westmin

9994 rows × 24 columns

```
df['Quantity_sum'] = np.where(df['Quantity']>=5, "True","False")
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	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Henders
1	2	CA-2019-152156	2019-11-08	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Henders
2	3	CA-2019-138688	2019-06-12	6/16/2019	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Ange
3	4	US-2018-108966	2018-10-11	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderda
4	5	US-2018-108966	2018-10-11	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderda
...	...	...	...	...	...	...	...	...	...	...
9989	9990	CA-2017-110422	2017-01-21	1/23/2017	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami
9990	9991	CA-2020-121258	2020-02-26	3/3/2020	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Me
9991	9992	CA-2020-121258	2020-02-26	3/3/2020	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Me
9992	9993	CA-2020-121258	2020-02-26	3/3/2020	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Me
9993	9994	CA-2020-119914	2020-05-04	5/9/2020	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westmin

9994 rows × 25 columns