Final Project - Analyzing Sales Data

Date: 02 February 2023

Author: Khemika Kaewsa-ard

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/Regio
0	1	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States
1	2	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States
2	3	CA-2019-138688	6/12/2019	6/16/2019	Second Class	DV-13045	Darrin Van Huff	Corporate	United States
3	4	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States
4	5	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States

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 datafr

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')
```

```
# 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	Cit
2234	2235	CA-2020-104066	2020-12-05	2020-12-10	Standard Class	QJ-19255	Quincy Jones	Corporate	United States	Bu
5274	5275	CA-2018-162887	2018-11-07	2018-11-09	Second Class	SV-20785	Stewart Visinsky	Consumer	United States	Bu
8798	8799	US-2019-150140	2019-04-06	2019-04-10	Standard Class	VM-21685	Valerie Mitchum	Home Office	United States	Bu
9146	9147	US-2019-165505	2019-01-23	2019-01-27	Standard Class	CB-12535	Claudia Bergmann	Corporate	United States	Bu
9147	9148	US-2019-165505	2019-01-23	2019-01-27	Standard Class	CB-12535	Claudia Bergmann	Corporate	United States	Bu
9148	9149	US-2019-165505	2019-01-23	2019-01-27	Standard Class	CB-12535	Claudia Bergmann	Corporate	United States	Bu
9386	9387	US-2020-	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Bu

		127292								
9387	9388	US-2020-127292	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Bu
9388	9389	US-2020-127292	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Bu
9389	9390	US-2020-127292	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Bu
9741	9742	CA-2018-117086	2018-11-08	2018-11-12	Standard Class	QJ-19255	Quincy Jones	Corporate	United States	Bu

11 rows × 21 columns

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

# Top 5 City

top_city = df['City'].value_counts().reset_index()
top_city.columns = ['City','count']

top_city.head(5)
```

	City	count
0	New York City	915
1	Los Angeles	747
2	Philadelphia	537
3	San Francisco	510
4	Seattle	428

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 mar
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

California = df[df.eq('California').any(1)]

California.to_csv("California_data.csv")

California.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City
2	3	CA-2019-138688	2019-06-12	2019-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles
5	6	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
6	7	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
7	8	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
8	9	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles

5 rows × 21 columns

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

#Texas = df[df.eq('Texas').any(1)]

#Cali_texas = df.query('State == "California" | State == "Texas"')

Cali_texas = df[(df['Order Date'] >= '2017-01-01') & (df['Order Date'] <= '2

Cali_texas.to_csv("California_texas2017.csv")
```

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

sales_date = df[(df['Order Date'] >= '2017-01-01') & (df['Order Date'] <= '2
sales_date['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
df[(df['Order Date'] >= '2018-01-01') & (df['Order Date'] <= '2018-12-31')]\
    .groupby('Segment')['Profit']\
    .agg('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 2

date_lease = df[(df['Order Date'] >= '2019-04-15') & (df['Order Date'] <= '2
date_lease.groupby('State')['Sales'].agg('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 2

propor_date = df[df['Order Date'].dt.year == 2019]
W_C_sales = propor_date.query('Region == "West" | Region == "Central"')['Sal
total_sales = propor_date['Sales'].sum()

result = (W_C_sales/total_sales) * 100
print(f"{result.round(2)}%")
```

```
54.97%
```

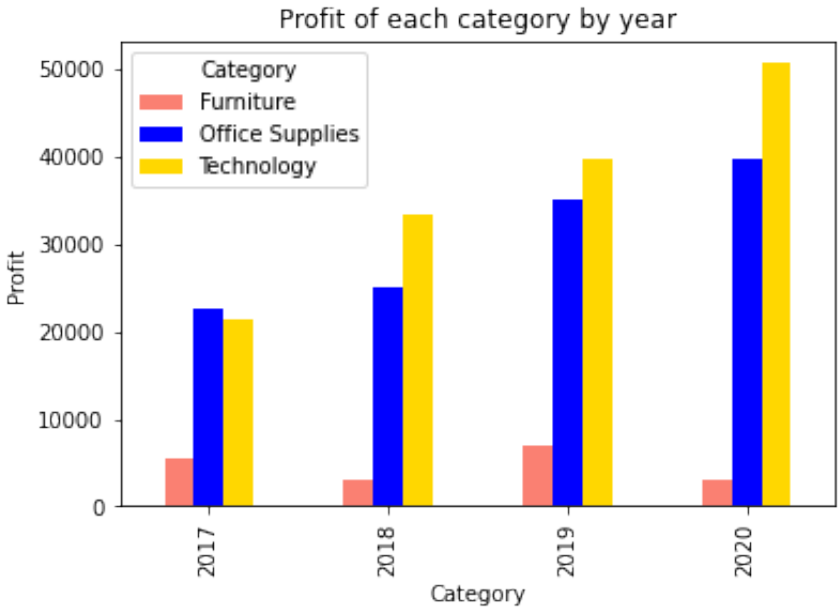
```
# TODO 09 - find top 10 popular products in terms of number of orders vs. to

date = df[(df['Order Date'].dt.year >= 2019) & (df['Order Date'].dt.year <=
top_orders = date.value_counts('Product Name').sort_values(ascending= False)
top_orders.columns = ['Product Name','Number of orders']

top_sales = date.groupby('Product Name')['Sales'].agg('sum').sort_values(asc
top_sales.columns = ['Product Name', 'Total Sales']

topten = pd.concat([top_orders,top_sales],axis=1)
topten
```

	Product Name	Number of orders	Product Name	Total Sales
0	Easy-staple paper	27	Canon imageCLASS 2200 Advanced Copier	61599.82
1	Staples	24	Hewlett Packard LaserJet 3310 Copier	16079.73
2	Staple envelope	22	3D Systems Cube Printer, 2nd Generation, Magenta	14299.89
3	Staples in misc. colors	13	GBC Ibimaster 500 Manual ProClick Binding System	13621.54
4	Staple remover	12	GBC DocuBind TL300 Electric Binding System	12737.26
5	Storex Dura Pro Binders	12	GBC DocuBind P400 Electric Binding System	12521.11
6	Chromcraft Round Conference Tables	12	Samsung Galaxy Mega 6.3	12263.71
7	Global Wood Trimmed Manager's Task Chair, Khaki	11	HON 5400 Series Task Chairs for Big and Tall	11846.56
8	Avery Non-Stick Binders	11	Martin Yale Chadless Opener Electric Letter Op...	11825.90
9	Staple-based wall hangings	10	Global Troy Executive Leather Low-Back Tilter	10169.89

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

# Profit of each category by year

df['Year'] = df['Order Date'].dt.strftime('%Y')
profit_cat = df.groupby(['Category','Year'])['Profit'].agg('sum').reset_inde

profit_year = profit_cat.pivot(index='Year', columns='Category', values='Pro
result = profit_year.plot(kind = 'bar', color=['salmon','blue','gold'],xlabe
result
```

```
<AxesSubplot:title={'center':'Profit of each category by year'}, xlabel='C
```

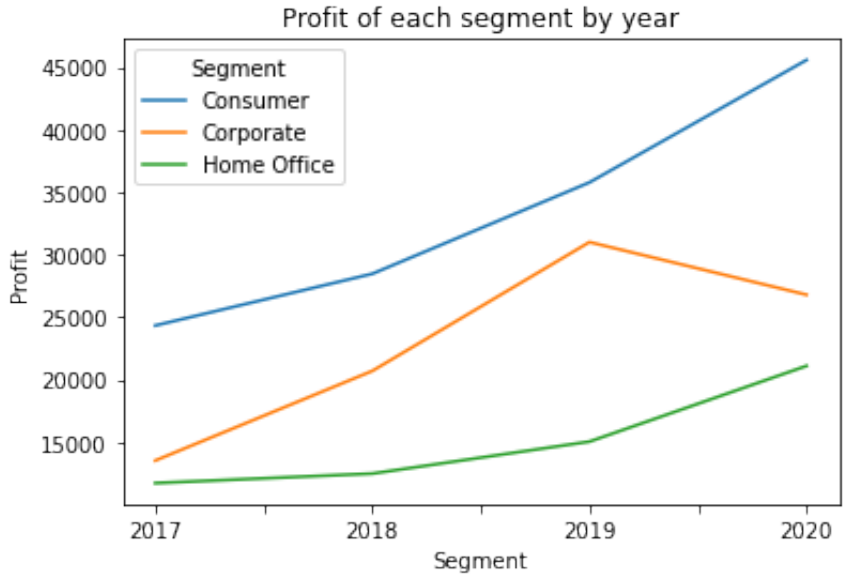
Download



```
df['Year'] = df['Order Date'].dt.strftime('%Y')
profit_seg = df.groupby(['Segment','Year'])['Profit'].agg('sum').reset_index

pro_year = profit_seg.pivot(index='Year', columns='Segment', values='Profit'
result2 = pro_year.plot.line(ylabel= "Profit", xlabel= "Segment", title= "Pr
result2
```

```
<AxesSubplot:title={'center':'Profit of each segment by year'}, xlabel='Se
```

Download




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

import numpy as np
df['new_column'] = np.where(df['Profit'] <0 ,"Loss","Profit")
df.head(10)
```

	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	Hender
1	2	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Hender

2	3	CA-2019-138688	2019-06-12	2019-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles
3	4	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderd
4	5	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderd
5	6	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
6	7	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
7	8	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
8	9	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
9	10	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles

10 rows × 23 columns