Final Project - Analyzing Sales Data

Date: 18 July 2023

Author: Nalin Sae-nim (Ning)

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	Henders
1	2	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Henders
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 Lauderc
4	5	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderc

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
```

	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	Henders
1	2	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henders
2	3	CA-2019-138688	2019-06-12	2019-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Ang
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
...	...	...	...	...	...	...	...	...	...	...
9989	9990	CA-2017-110422	2017-01-21	2017-01-23	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami
9990	9991	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa M
9991	9992	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa M
9992	9993	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa M
9993	9994	CA-2020-119914	2020-05-04	2020-05-09	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westmin

9994 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() == True]
```

	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
# Top 5 Customers

df['Customer Name'].value_counts().head()
```

```
William Brown          37
John Lee               34
Matt Abelman           34
Paul Prost             34
Chloris Kastensmidt    32
Name: Customer Name, dtype: int64
```

```
# How many customer do we have?

df['Customer Name'].value_counts().count()
```

```
793
```

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 # (row, column)

print('columns : ',df.shape[1])
print('rows : ',df.shape[0])
```

```
columns :  21
rows :  9994
```

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

df.isna().sum().sort_values(ascending = False)
```

```
Postal Code       11
Row ID             0
Discount           0
Quantity           0
Sales              0
Product Name       0
Sub-Category       0
Category           0
Product ID         0
Region             0
State              0
Order ID           0
City               0
Country/Region     0
Segment            0
Customer Name      0
Customer ID        0
Ship Mode          0
Ship Date          0
Order Date         0
Profit             0
dtype: int64
```

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

df_califonia = df[df['State'] == 'California']

df_califonia.to_csv('df_califonia.csv')
```

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

# df.query("State == ['California', 'Texas']")[ (df['Order Date'].dt.year == 2017)

# Query Select Califonia & Texas -> Filter year 2017
cc = df.query("State == ['California', 'Texas']")

cc[cc['Order Date'].dt.year == 2017]
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City	...
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	...
...	...	...	...	...	...	...	...	...	...	...	...
9885	9886	CA-2017-112291	2017-04-03	2017-04-08	Standard Class	KE-16420	Katrina Edelman	Corporate	United States	Los Angeles	...
9903	9904	CA-2017-122609	2017-11-12	2017-11-18	Standard Class	DP-13000	Darren Powers	Consumer	United States	Carrollton	...
9904	9905	CA-2017-122609	2017-11-12	2017-11-18	Standard Class	DP-13000	Darren Powers	Consumer	United States	Carrollton	...
9942	9943	CA-2017-143371	2017-12-28	2018-01-03	Standard Class	MD-17350	Maribeth Dona	Consumer	United States	Anaheim	...
9943	9944	CA-2017-143371	2017-12-28	2018-01-03	Standard Class	MD-17350	Maribeth Dona	Consumer	United States	Anaheim	...

632 rows × 21 columns

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

df2017 = df[ (df['Order Date'].dt.year == 2017) ]


print('Total Sales :', df2017['Sales'].sum())
print('Average Sales :', df2017['Sales'].mean())
print('Standard Deviation Sales :', df2017['Sales'].std())
```

```
Total Sales : 484247.4981
Average Sales : 242.97415860511794
Standard Deviation Sales : 754.0533572593683
```

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

df2018 = df[ (df['Order Date'].dt.year == 2018) ]

result = df2018[['Profit', 'Segment']].groupby('Segment').sum('Profit')

result.iloc[0]
```

```
Profit    28460.1665
Name: Consumer, dtype: float64
```

```
# TODO 07 - which top 5 States have the least total sales between 15 April 2019 - 3
# Filter Date

df2019 = df[ (df['Order Date']>= '2019-04-15 00:00:00') & (df['Order Date']<= '2019


# Select Column -> groupby -> sum -> sort -> head

df2019[['Sales', 'State']].groupby('State').sum('Sales').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.g
# Filter Date

year2019 = df[ (df['Order Date']>= '2019-01-01 00:00:00') & (df['Order Date']<= '20

# Sumarise
region19 = year2019['Region'].value_counts()

percent = ((region19[0] + region19[2]) / sum(region19) * 100).__round__()

print('The proportion of total sales (%) in West + Central in 2019 is', percent, '%
```

```
The proportion of total sales (%) in West + Central in 2019 is 54 %
```

```
year2019.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	...
13	14	CA-2019-161389	2019-12-05	2019-12-10	Standard Class	IM-15070	Irene Maddox	Consumer	United States	Seattle	...
21	22	CA-2019-137330	2019-12-09	2019-12-13	Standard Class	KB-16585	Ken Black	Corporate	United States	Fremont	...

5 rows × 21 columns

```
region19
```

```
West       805
East       766
Central    603
South      413
Name: Region, dtype: int64
```

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

# Filter 2019-2020
df20192020 = df[(df['Order Date'].dt.year == 2019) | (df['Order Date'].dt.year == 2

# Top 10 number of order
top_order = df20192020.groupby('Product Name')[['Sales', 'Quantity']]\
    .sum().sort_values('Quantity', ascending = False).head(10).round()

top_order
```

	Sales	Quantity
Product Name		
Staples	462.0	124
Easy-staple paper	1482.0	89
Staple envelope	645.0	73
Staples in misc. colors	357.0	60
Chromcraft Round Conference Tables	7965.0	59
Storex Dura Pro Binders	176.0	49
Situations Contoured Folding Chairs, 4/Set	2612.0	47
Wilson Jones Clip & Carry Folder Binder Tool for Ring Binders, Clear	178.0	44
Avery Non-Stick Binders	122.0	43
Eldon Wave Desk Accessories	216.0	42

```
# Top 10 total sales

top_sales = df20192020.groupby('Product Name')[['Sales', 'Quantity']]\
    .sum().sort_values('Sales', ascending = False).head(10).round()

top_sales
```

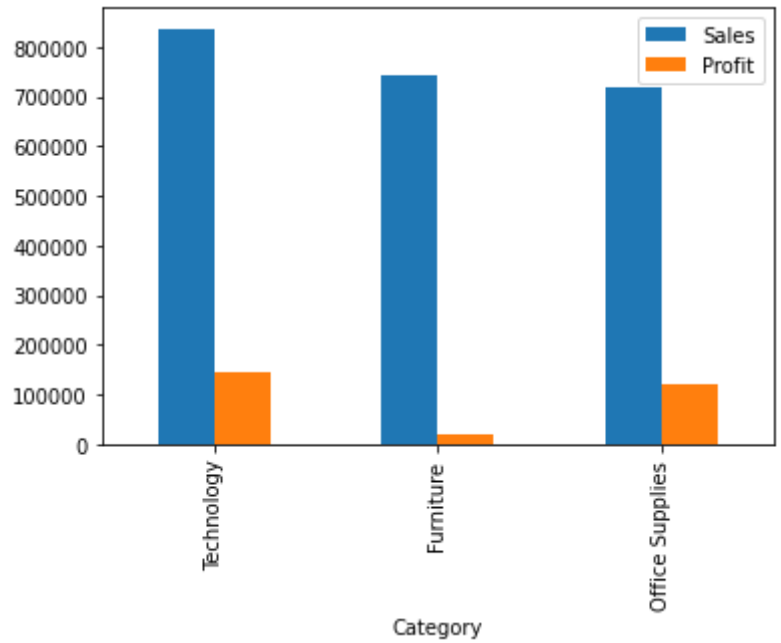
	Sales	Quantity
Product Name		
Canon imageCLASS 2200 Advanced Copier	61600.0	20
Hewlett Packard LaserJet 3310 Copier	16080.0	31
3D Systems Cube Printer, 2nd Generation, Magenta	14300.0	11
GBC Ibimaster 500 Manual ProClick Binding System	13622.0	31
GBC DocuBind TL300 Electric Binding System	12737.0	21
GBC DocuBind P400 Electric Binding System	12521.0	16
Samsung Galaxy Mega 6.3	12264.0	34
HON 5400 Series Task Chairs for Big and Tall	11847.0	21
Martin Yale Chadless Opener Electric Letter Opener	11826.0	16
Global Troy Executive Leather Low-Back Tilter	10170.0	25

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

# Compare Profit vs Sales

df.groupby('Category')[['Sales', 'Profit']].sum().sort_values('Sales', ascending =
```

Download

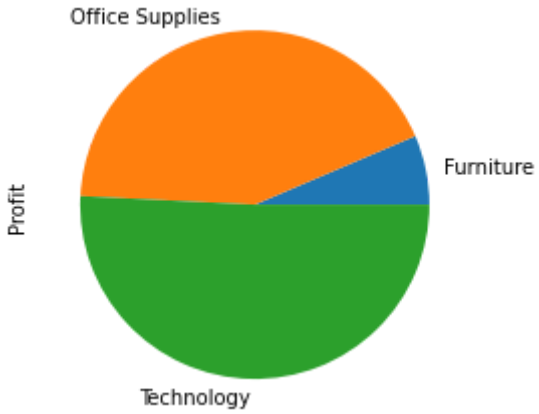


This chart shows the proportion of total profit compared with total sales divided by category. I found that even though the furniture category has total sales more than the office suppliers

category, but the office suppliers category makes more benefit. You can see the proportion of total profit compared with total sales on the chart, makes us know that sometimes we have to focus on total profit more than total sales, like in this case.

```
# Profit by Category
df.groupby('Category')['Profit'].sum().plot(kind = 'pie');
```

Download



This chart shows a profit chart divided by Category. I found the highest profit is Technology, the next is Office Supplies and the last one is Furniture. So, we will know which category that we have to focus on.

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

# Group customers by paid full price and discount price

import numpy as np

df['price'] = np.where(df['Discount']>0, 'discount', 'full price')

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	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henders
1	2	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henders
2	3	CA-2019-138688	2019-06-12	2019-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Ang
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
...	...	...	...	...	...	...	...	...	...	...
9989	9990	CA-2017-110422	2017-01-21	2017-01-23	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami
9990	9991	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa M
9991	9992	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa M
9992	9993	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa M
9993	9994	CA-2020-119914	2020-05-04	2020-05-09	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westmin

9994 rows × 22 columns