Final Project - Analyzing Sales Data

Date: 13 May 2023

Author: Benjawan Graisriwattana (Nam)

Course: `Pandas Foundation`

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

```
# preview top 5 rows
df
```

	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	Her
1	2	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Her
2	3	CA-2019-138688	6/12/2019	6/16/2019	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los
3	4	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	For Lau
4	5	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	For Lau
...	...	...	...	...	...	...	...	...	...	...
9989	9990	CA-2017-110422	1/21/2017	1/23/2017	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Mia
9990	9991	CA-2020-121258	2/26/2020	3/3/2020	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Cos
9991	9992	CA-2020-121258	2/26/2020	3/3/2020	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Cos
9992	9993	CA-2020-121258	2/26/2020	3/3/2020	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Cos
9993	9994	CA-2020-119914	5/4/2020	5/9/2020	Second Class	CC-12220	Chris Cortes	Consumer	United States	We

9994 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
order_date = pd.to_datetime(df['Order Date'], format='%m/%d/%Y')
ship_date = pd.to_datetime(df['Ship Date'], format='%m/%d/%Y')
```

```
df['Order Date'] = order_date
df['Ship Date'] = ship_date
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	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	.
3	4	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	.
4	5	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	.
...	...	...	...	...	...	...	...	...	...	...	.
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 Mesa	.
9991	9992	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	.
9992	9993	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	.
9993	9994	CA-2020-119914	2020-05-04	2020-05-09	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westminster	.

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()]
```

	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_filtered = df[df['Order Date'].dt.strftime('%Y') == '2020' ]
df_filtered['Profit'].sum()
```

```
93439.26960000001
```

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 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
dtype: int64
```

```
# TODO 03 - your friend ask for `California` data, filter it and export csv for him
data_california = df.query("State == 'California'")
pd.DataFrame(data_california)
data_california.to_csv("data_california.csv")
data_california
```

	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	...
...	...	...	...	...	...	...	...	...	...	...	...
9986	9987	CA-2019-125794	2019-09-29	2019-10-03	Standard Class	ML-17410	Maris LaWare	Consumer	United States	Los Angeles	...
9990	9991	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...
9991	9992	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...
9992	9993	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...
9993	9994	CA-2020-119914	2020-05-04	2020-05-09	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westminster	...

2001 rows × 21 columns

```
# TODO 04 - your friend ask for all order data in `California` and `Texas` in 2017
data_2017 = df[df['Order Date'].dt.strftime('%Y') == '2017']
cali_tax = data_2017.query( "State == 'California' or State == 'Texas'")
cali_tax.to_csv("California_Texas_2017")
```

```
# TODO 05 - how much total sales, average sales, and standard deviation of sales yo
total_sales = data_2017.Sales.sum()
avg_sales = data_2017.Sales.mean()
std_sales = data_2017.Sales.std()
print(total_sales, avg_sales, std_sales)
```

```
484247.4981 242.97415860511794 754.0533572593683
```

```
# TODO 06 - which Segment has the highest profit in 2018
df.sort_values(by = "Profit", ascending = False).head(1)
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City	...	Po Co
6826	6827	CA-2019-118689	2019-10-02	2019-10-09	Standard Class	TC-20980	Tamara Chand	Corporate	United States	Lafayette	...	47

1 rows × 21 columns

```
# TODO 07 - which top 5 States have the least total sales between 15 April 2019 - 3
data_2019 = df.loc[df['Order Date'].between('2019-04-15', '2019-12-31', inclusive =
data_2019.groupby(['State']).sum().sort_values(by = "Sales").head(5)
```

	Row ID	Postal Code	Sales	Quantity	Discount	Profit
State						
New Hampshire	7208	6361.0	49.05	7	0.0	14.6469
New Mexico	23311	352880.0	64.08	11	0.6	24.9520
District of Columbia	11159	100080.0	117.07	18	0.0	50.2118
Louisiana	26138	281839.0	249.80	17	0.0	82.0472
South Carolina	58770	321531.0	502.48	42	0.0	144.1038

```
<ipython-input-142-aad8938822c5>:2: FutureWarning: Boolean inputs to the `inclusi
  data_2019 = df.loc[df['Order Date'].between('2019-04-15', '2019-12-31', inclusi
```

```
# TODO 08 - what is the proportion of total sales (%) in West + Central in 2019 e.g
data_2019 = df[df['Order Date'].dt.strftime('%Y') == '2019']["Sales"].sum()
west_central = df.query( "Region == 'West' or Region == 'Central'")
proportion = (west_central[west_central['Order Date'].dt.strftime('%Y') == '2019'][
print(f"proportion of total sales (%) in West + Central in 2019: {proportion.__rour
```

```
proportion of total sales (%) in West + Central in 2019: 54.97%
```

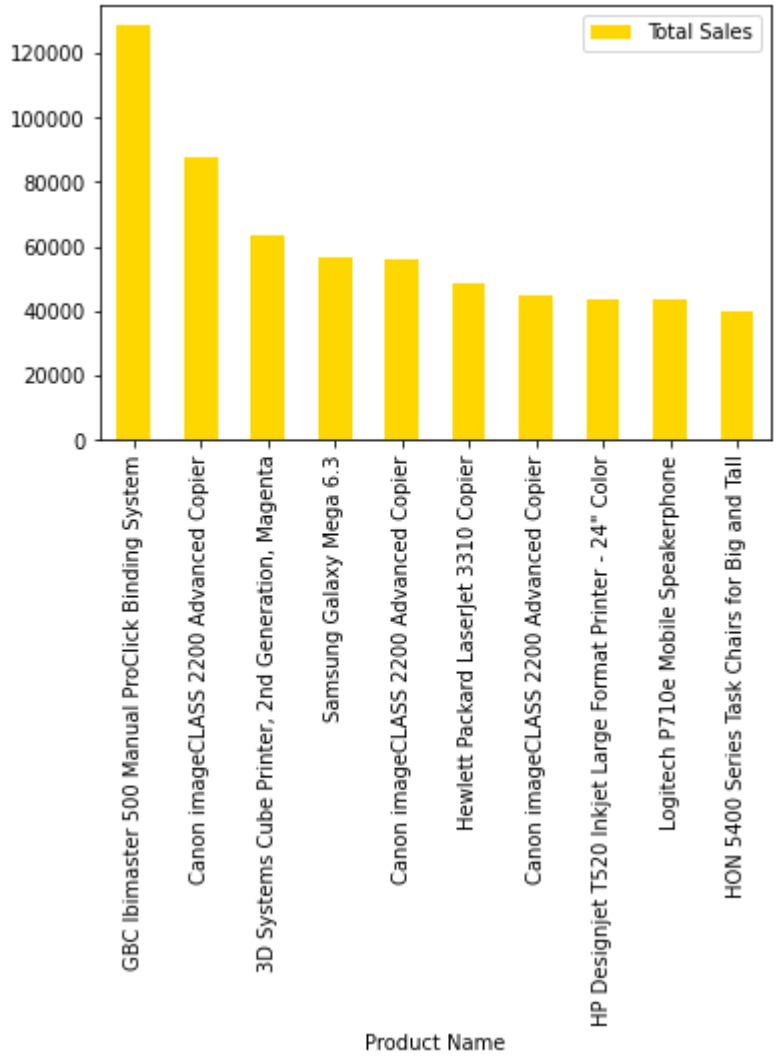
```
# TODO 09 - find top 10 popular products in terms of number of orders vs. total sal
data_2019 = df[df['Order Date'].dt.strftime('%Y').between('2019', '2020')]
data = data_2019.groupby(["Order ID", "Product Name"])[['Sales', 'Quantity']].sum()
data['Total Sales'] = data['Sales'] * data['Quantity']
data_top10 = data.sort_values('Total Sales', ascending=False).reset_index().head(10
data_top10
```

	Order ID	Product Name	Sales	Quantity	Total Sales
0	CA-2019-117121	GBC Ibimaster 500 Manual ProClick Binding System	9892.740	13	128605.620
1	CA-2019-118689	Canon imageCLASS 2200 Advanced Copier	17499.950	5	87499.750
2	US-2019-107440	3D Systems Cube Printer, 2nd Generation, Magenta	9099.930	7	63699.510
3	CA-2020-129021	Samsung Galaxy Mega 6.3	4367.896	13	56782.648
4	CA-2020-140151	Canon imageCLASS 2200 Advanced Copier	13999.960	4	55999.840
5	US-2019-140158	Hewlett Packard LaserJet 3310 Copier	5399.910	9	48599.190
6	CA-2020-127180	Canon imageCLASS 2200 Advanced Copier	11199.968	4	44799.872
7	CA-2019-158841	HP Designjet T520 Inkjet Large Format Printer ...	8749.950	5	43749.750
8	CA-2019-145625	Logitech P710e Mobile Speakerphone	3347.370	13	43515.810
9	CA-2020-118892	HON 5400 Series Task Chairs for Big and Tall	4416.174	9	39745.566

```
# TODO 10 - plot at least 2 plots, any plot you think interesting :)
# Total Sales of 10 top Product
data_top10.plot(kind = 'bar', x = 'Product Name', y = 'Total Sales', color = 'Gold'
```

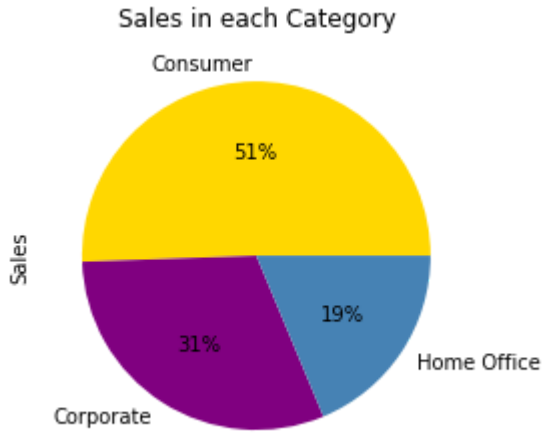
```
<AxesSubplot:xlabel='Product Name'>
```

Download




```
# Pie graph Sales percent of segments in United States
df_new = df.rename(columns={"Country/Region" : "Country"})
df_new.query("Country == 'United States'").groupby("Segment").sum().plot(kind='pie'
                                        ,title = 'Sales in each Category', colors=['g
```

Download




```
# TODO Bonus - use np.where() to create new column in dataframe to help you answer
# Total Sales add vat 7% of each Customers
import numpy as np
df_vat = round(df[['Sales', 'Customer Name', 'Quantity', 'Profit', 'Discount']].gro
df_vat['Total Sales (vat7%)'] = np.where(df_vat['Sales']>0, round(df_vat['Sales'] *
df_vat.head()
```

	Sales	Quantity	Profit	Discount	Total Sales (vat7%)
Customer Name					
Aaron Bergman	886.16	13	129.35	0.40	11519.68
Aaron Hawkins	1744.70	54	365.22	1.00	94212.80
Aaron Smayling	3050.69	48	-253.57	3.55	146429.57
Adam Bellavance	7755.62	56	2054.59	0.80	434313.92
Adam Hart	3250.34	75	281.19	2.70	243772.80