Final Project - Analyzing Sales Data

Date: 8 December 2022

Author: Wichuorn Phimjam

Course: `Pandas Foundation`

```
# import data
import pandas as pd
import numpy as np
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	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', 'Ship Date']] = df[['Order Date','Ship Date']].apply(pd.to_date
df.head(5)
```

	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

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
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 h
df_California = df.query('State == "California"')
df_California.to_csv('df_California.csv')
```

```
# TODO 04 - your friend ask for all order data in `California` and `Texas` in 201
df_cali_Tex_2017 = df[((df['State']=='California') | (df['State']=='Texas')) & (d
df_cali_Tex_2017.to_csv('df_cali_Tex_2017.csv')
```

	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
df_year_2017 = df[df['Order Date'].dt.year == 2017]
df_year_2017['Sales'].agg(['sum', 'mean', 'std']).reset_index()
```

	index	Sales
0	sum	484247.498100
1	mean	242.974159
2	std	754.053357

```
# TODO 06 - which Segment has the highest profit in 2018
df_2018 = df[df['Order Date'].dt.year == 2018]
df_2018.groupby('Segment')['Profit'].sum().reset_index()
```

	Segment	Profit
0	Consumer	28460.1665
1	Corporate	20688.3248
2	Home Office	12470.1124

```
# TODO 07 - which top 5 States have the least total sales between 15 April 2019 -
sale_state = df[(df['Order Date'] >= '2019-04-15') & (df['Order Date'] <= '2019-1
sale_state.sort_values(['Sales'], ascending=False).head()
```

	State	Sales
3	California	105632.9565
29	New York	56873.9340
38	Texas	31114.3390
34	Pennsylvania	28207.2940
20	Michigan	26675.8110

```
# TODO 08 - what is the proportion of total sales (%) in West + Central in 2019 e
df_2019 = df[df['Order Date'].dt.year == 2019]
df_2019_West = df_2019[df_2019['Region'] == 'West'].groupby('Region')['Sales'].su
df_2019_Central = df_2019[df_2019['Region'] == 'Central'].groupby('Region')['Sale
df_2019_total = df_2019['Sales'].sum()
df_west_central = df_2019_Central['Sales']+df_2019_West['Sales']
(df_west_central*100)/df_2019_total
```

```
0    54.974799
Name: Sales, dtype: float64
```

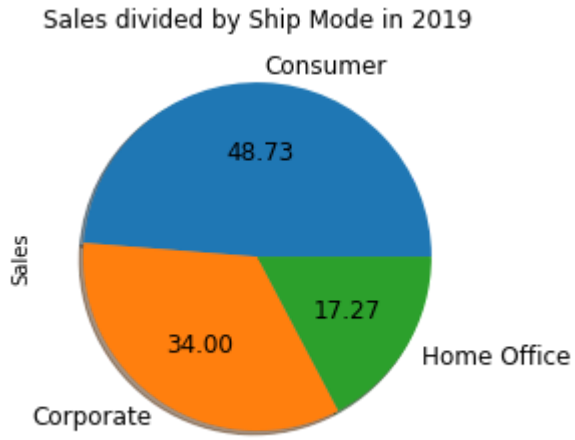
```
# TODO 09 - find top 10 popular products in terms of number of orders vs. total s
df_2019_2020 = df[(df['Order Date'].dt.year == 2019) | (df['Order Date'].dt.year
df_count = df_2019_2020['Product Name'].value_counts().reset_index().head(10)
df_sales = df_2019_2020.groupby('Product Name')['Sales'].sum()
df_sales = df_sales.sort_values(ascending=False).reset_index().head(10)
print(f'{df_count}\n {df_sales}')
```

```
                                        index  Product Name
0                              Easy-staple paper            27
1                                        Staples            24
2                                Staple envelope            22
3                        Staples in misc. colors            13
4              Chromcraft Round Conference Tables            12
5                        Storex Dura Pro Binders            12
6                                  Staple remover            12
7  Global Wood Trimmed Manager's Task Chair, Khaki            11
8                          Avery Non-Stick Binders            11
9             Sterilite Officeware Hinged File Box            10
                                     Product Name      Sales
0              Canon imageCLASS 2200 Advanced Copier  61599.824
1               Hewlett Packard LaserJet 3310 Copier  16079.732
2   3D Systems Cube Printer, 2nd Generation, Magenta  14299.890
3   GBC Ibimaster 500 Manual ProClick Binding System  13621.542
4         GBC DocuBind TL300 Electric Binding System  12737.258
5          GBC DocuBind P400 Electric Binding System  12521.108
6                           Samsung Galaxy Mega 6.3  12263.708
7       HON 5400 Series Task Chairs for Big and Tall  11846.562
```

```
# TODO 10 - plot at least 2 plots, any plot you think interesting :)
df[df['Order Date'].dt.year == 2019].groupby('Segment')['Sales']\
    .sum().plot.pie(title = 'Sales divided by Ship Mode in 2019',
                    autopct = '%.2f',
                    fontsize = 12,
                    shadow = True)
```

```
<AxesSubplot:title={'center':'Sales divided by Ship Mode in 2019'}, ylabel='Sal
```

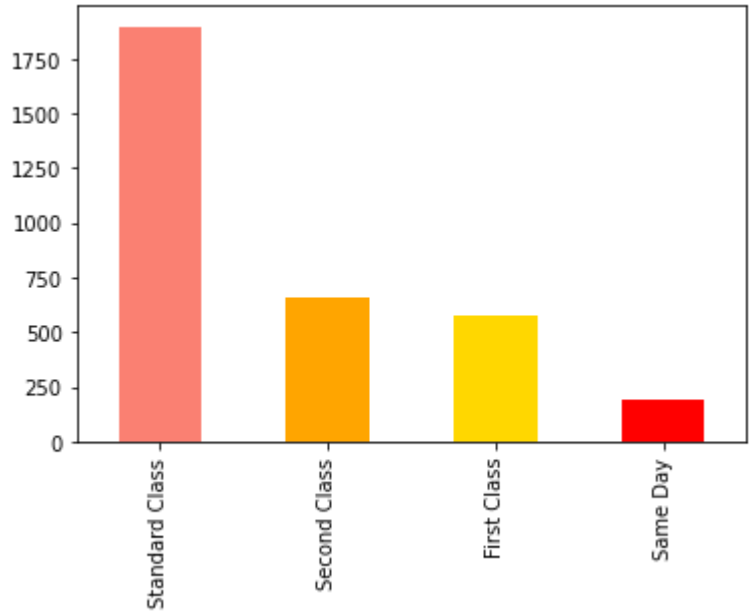
Download




```
df_2020 = df[df['Order Date'].dt.year == 2020]
df_2020['Ship Mode'].value_counts().plot(kind = 'bar', color = ['salmon', 'orange
print("What is the most popular ship mode in 2020?")
```

```
What is the most popular ship mode in 2020?
```

Download




```
# TODO Bonus - use np.where() to create new column in dataframe to help you answe
mean_sale = np.mean(df['Sales'])
df['new_column'] = np.where(df['Sales'] > mean_sale, "Good customer", "Normal cus
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	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	6	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
6	7	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
7	8	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
8	9	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
9	10	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles

10 rows × 23 columns