Final Project - Analyzing Sales Data

Date: 19/9/2024

Author: Tunyong Subsomboon (Tia)

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	...	Po Co
0	1	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42
1	2	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42
2	3	CA-2019-138688	6/12/2019	6/16/2019	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90
3	4	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33
4	5	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33

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	...	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
...	...	...	...	...	...	...	...	...	...	...	...	.
9989	9990	CA-2017-110422	2017-01-21	2017-01-23	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami	...	3
9990	9991	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	9
9991	9992	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	9
9992	9993	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	9
9993	9994	CA-2020-119914	2020-05-04	2020-05-09	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westminster	...	9

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	...	Postal Code
2234	2235	CA-2020-104066	2020-12-05	2020-12-10	Standard Class	QJ-19255	Quincy Jones	Corporate	United States	Burlington	...	NaN
5274	5275	CA-2018-162887	2018-11-07	2018-11-09	Second Class	SV-20785	Stewart Visinsky	Consumer	United States	Burlington	...	NaN
8798	8799	US-2019-150140	2019-04-06	2019-04-10	Standard Class	VM-21685	Valerie Mitchum	Home Office	United States	Burlington	...	NaN
9146	9147	US-2019-165505	2019-01-23	2019-01-27	Standard Class	CB-12535	Claudia Bergmann	Corporate	United States	Burlington	...	NaN
9147	9148	US-2019-165505	2019-01-23	2019-01-27	Standard Class	CB-12535	Claudia Bergmann	Corporate	United States	Burlington	...	NaN
9148	9149	US-2019-165505	2019-01-23	2019-01-27	Standard Class	CB-12535	Claudia Bergmann	Corporate	United States	Burlington	...	NaN
9386	9387	US-2020-127292	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlington	...	NaN
9387	9388	US-2020-127292	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlington	...	NaN
9388	9389	US-2020-127292	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlington	...	NaN
9389	9390	US-2020-127292	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlington	...	NaN
9741	9742	CA-2018-117086	2018-11-08	2018-11-12	Standard Class	QJ-19255	Quincy Jones	Corporate	United States	Burlington	...	NaN

11 rows × 21 columns

```
# TODO - what date has the highest sales
max_sale_row = df[df['Sales'] == df['Sales'].max()]

result = max_sale_row[['Order Date', 'Quantity', 'Sales']]

print(result)
```

```
     Order Date  Quantity     Sales
2697 2017-03-18         6  22638.48
```

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
num_rows, num_cols = df.shape
print( "Number of rows:", num_rows)
print("Number of columns:", num_cols)
```

```
Number of rows: 9994
Number of columns: 21
```

```
# TODO 02 -
 is there any missing values?, if there is, which colunm? how many nan values?
df[df['Postal Code'].isna()]
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City	...	Postal Code
2234	2235	CA-2020-104066	2020-12-05	2020-12-10	Standard Class	QJ-19255	Quincy Jones	Corporate	United States	Burlington	...	NaN
5274	5275	CA-2018-162887	2018-11-07	2018-11-09	Second Class	SV-20785	Stewart Visinsky	Consumer	United States	Burlington	...	NaN
8798	8799	US-2019-150140	2019-04-06	2019-04-10	Standard Class	VM-21685	Valerie Mitchum	Home Office	United States	Burlington	...	NaN
9146	9147	US-2019-165505	2019-01-23	2019-01-27	Standard Class	CB-12535	Claudia Bergmann	Corporate	United States	Burlington	...	NaN
9147	9148	US-2019-165505	2019-01-23	2019-01-27	Standard Class	CB-12535	Claudia Bergmann	Corporate	United States	Burlington	...	NaN
9148	9149	US-2019-165505	2019-01-23	2019-01-27	Standard Class	CB-12535	Claudia Bergmann	Corporate	United States	Burlington	...	NaN
9386	9387	US-2020-127292	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlington	...	NaN
9387	9388	US-2020-127292	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlington	...	NaN
9388	9389	US-2020-127292	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlington	...	NaN
9389	9390	US-2020-127292	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlington	...	NaN
9741	9742	CA-2018-117086	2018-11-08	2018-11-12	Standard Class	QJ-19255	Quincy Jones	Corporate	United States	Burlington	...	NaN

11 rows × 21 columns

```
# TODO 04 -
 your friend ask for all order data in `California` and `Texas` in 2017 (look at Orde
r Date), send him csv file
California_and_Texas = df[((df['State'] == 'California') |
 (df['State'] == 'Texas')) & (df['Order Date'].dt.year == 2017)]

California_and_Texas.to_csv('California_and_Texas_data_2017.csv')
```

```
# TODO 05 -
 how much total sales, average sales, and standard deviation of sales your company ma
```

```
ke in 2017
df_2017 = df[df['Order Date'].dt.year == 2017]
totalsale_2017 = df_2017['Sales'].sum()
avg_2017 = df_2017['Sales'].mean()
sd_2017 = df_2017['Sales'].std()

print(f'total sales in 2017: {totalsale_2017} $')
print(f'average sales in 2017: {avg_2017} $')
print(f'standard devitation of sales in 2017: {sd_2017}')
```

```
total sales in 2017: 484247.4981 $
average sales in 2017: 242.97415860511794 $
standard devitation of sales in 2017: 754.0533572593683
```

```
# TODO 06 - which Segment has the highest profit in 2018
df2018 = df[df['Order Date'].dt.year== 2018]
# calculate the total profit by segment
segment_profit = df2018.groupby('Segment')['Profit'].sum()
# find the segment with the highest profit
highest_profit_segment = segment_profit.idxmax()
print(highest_profit_segment)
segment_profit = df2018.groupby('Segment')['Profit'].sum()
```

```
Consumer
```

```
# TODO 07 - which top 5 States have the least total sales between 15 April 2019 -
 31 December 2019
filtered_df = df[(df['Order Date'] >= '2019-04-15') & (df['Order Date'] <= '2019-12-
31')]
state_sales = filtered_df.groupby('State')['Sales'].sum()
least5_sales = state_sales.nsmallest(5)
print(least5_sales)
```

```
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
```

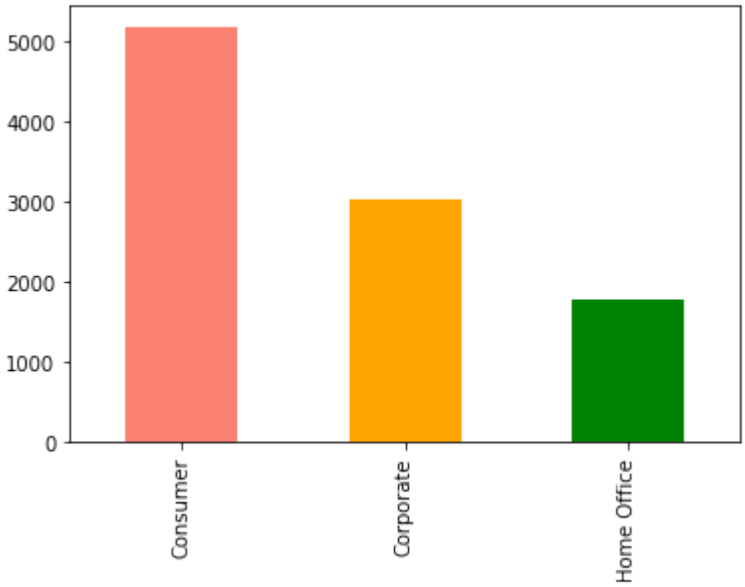
```
# TODO 08 -
 what is the proportion of total sales (%) in West + Central in 2019 e.g. 25%
df2019 = df[df['Order Date'].dt.year== 2019]
# group dataframe by 'Region' and sum 'Sales'
region_sales_2019 = df2019.groupby('Region')['Sales'].sum()

# calculate the total sales in 'West' and 'Central'
west_central_sales = region_sales_2019['West'] + region_sales_2019['Central']

# calculate total sales in 2019
total_sales_2019 = region_sales_2019.sum()

# calculate the proportion
proportion = (west_central_sales / total_sales_2019) * 100

print(f"The proportion of total sales in West + Central in 2019 is {proportion:.2f}%"
)
```

```
The proportion of total sales in West + Central in 2019 is 54.97%
```

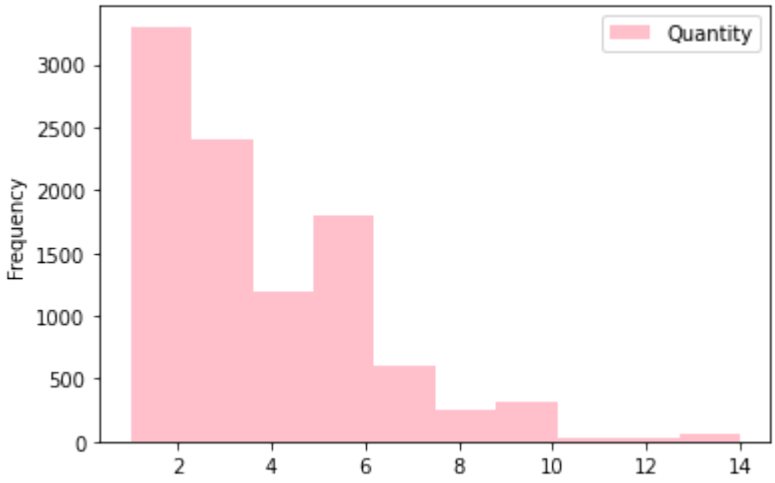
```
# TODO 09 -
 find top 10 popular products in terms of number of orders vs. total sales during 201
9-2020
new_years = df[(df['Order Date'].dt.year >= 2019) & (df['Order Date'].dt.year <= 2020
)]
Q = new_years.groupby('Product Name')['Quantity'].sum()
S = new_years.groupby('Product Name')['Sales'].sum()
result1 = Q.nlargest(10)
result2 = S.nlargest(10)
print(result1)
print(result2)
```

```
Product Name
Staples                                                               124
Easy-staple paper                                                      89
Staple envelope                                                        73
Staples in misc. colors                                                60
Chromcraft Round Conference Tables                                     59
Storex Dura Pro Binders                                                49
Situations Contoured Folding Chairs, 4/Set                             47
Wilson Jones Clip & Carry Folder Binder Tool for Ring Binders, Clear   44
Avery Non-Stick Binders                                                43
Eldon Wave Desk Accessories                                            42
Name: Quantity, dtype: int64
Product Name
Canon imageCLASS 2200 Advanced Copier                 61599.824
Hewlett Packard LaserJet 3310 Copier                  16079.732
3D Systems Cube Printer, 2nd Generation, Magenta      14299.890
GBC Ibimaster 500 Manual ProClick Binding System      13621.542
GBC DocuBind TL300 Electric Binding System            12737.258
GBC DocuBind P400 Electric Binding System             12521.108
Samsung Galaxy Mega 6.3                               12263.708
```

```
# TODO 10 - plot at least 2 plots, any plot you think interesting :)
df['Segment'].value_counts().plot(kind='bar', color=['salmon', 'orange', 'green']);
```




```
df[['Quantity']].plot(kind='hist', color='pink');
```




```
# TODO Bonus -
 use np.where() to create new column in dataframe to help you answer your own questio
ns
#find promotion or not
import numpy as np
df['new_column'] = np.where(df['Discount'] > 0.0, 'Promotion', 'not promotion')
df
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City	...	F
0	1	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	S
1	2	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	S
2	3	CA-2019-138688	2019-06-12	2019-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	V
3	4	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	S
4	5	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	S
...	...	...	...	...	...	...	...	...	...	...	...	.
9989	9990	CA-2017-110422	2017-01-21	2017-01-23	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami	...	S
9990	9991	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	V
9991	9992	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	V
9992	9993	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	V
9993	9994	CA-2020-119914	2020-05-04	2020-05-09	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westminster	...	V

9994 rows × 22 columns