Final Project - Analyzing Sales Data with Pandas Python

Date: November 2022

```
# 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

```
# 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')
```

```
# 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	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 Angele

```
# TODO - count nan in postal code column

df['Postal Code'].isna().sum()
```

```
11
```

```
# TODO - filter rows with missing values
##Which row(s) has(ve) missing values?
### 'Postal Code'

df.isna().sum()
```

```
# TODO - Explore this dataset on your owns, ask your own questions
## Which category make the most profit?

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

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()
#Postal Code - 11 values
```

```
# TODO 03 - your friend ask for `California` data, filter it and export csv for h
result = df.query("State == 'California'").reset_index()
result
```

	index	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	...	
0	2	3	CA-2019-138688	2019-06-12	2019-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	...	
1	5	6	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	...	
2	6	7	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	...	
3	7	8	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	...	
4	8	9	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	...	
...	...	...	...	...	...	...	...	...	...	...	...	
1996	9986	9987	CA-2019-125794	2019-09-29	2019-10-03	Standard Class	ML-17410	Maris LaWare	Consumer	United States	...	
1997	9990	9991	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	...	
			CA-	2020-	2020-	Standard	DB-	Dave				

```
# TODO 04 - your friend ask for all order data in `California` and `Texas` in 201
## Long
df['Order Date'] = pd.to_datetime(df['Order Date'], format = '%m/%d/%Y')

california2017 = df[(df['Order Date'] >= '2017-01-01') & (df['Order Date'] <= '20

california2017
```

	index	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	...
0	5	6	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	...
1	6671	6672	CA-2017-154837	2017-08-23	2017-08-27	Second Class	RB-19645	Robert Barroso	Corporate	United States	...
2	6623	6624	CA-2017-130449	2017-09-06	2017-09-09	First Class	VP-21760	Victoria Pisteka	Corporate	United States	...
3	6622	6623	CA-2017-130449	2017-09-06	2017-09-09	First Class	VP-21760	Victoria Pisteka	Corporate	United States	...
4	6311	6312	CA-2017-112837	2017-09-11	2017-09-16	Standard Class	LW-17125	Liz Willingham	Consumer	United States	...
...	...	...	...	...	...	...	...	...	...	...	...
410	3164	3165	CA-2017-153969	2017-09-21	2017-09-25	Standard Class	HF-14995	Herbert Flentye	Consumer	United States	...
411	3163	3164	CA-2017-153969	2017-09-21	2017-09-25	Standard Class	HF-14995	Herbert Flentye	Consumer	United States	...
412	3162	3163	CA-2017-153969	2017-09-21	2017-09-25	Standard Class	HF-14995	Herbert Flentye	Consumer	United States	...
413	3207	3208	CA-2017-158372	2017-11-10	2017-11-16	Standard Class	RD-19900	Ruben Dartt	Consumer	United States	...
414	9943	9944	CA-2017-143371	2017-12-28	2018-01-03	Standard Class	MD-17350	Maribeth Dona	Consumer	United States	...

```
# TODO 05 - how much total sales, average sales, and standard deviation of sales
df[(df['Order Date'].dt.strftime('%Y') == '2017')]['Sales']\
    .agg(['sum','mean','std']).round(2)
```

```
# TODO 06 - which Segment has the highest profit in 2018
filter2018 = df[(df['Order Date'].dt.strftime('%Y') == '2018')]
filter2018.groupby('Segment')['Profit'].sum()
```

```
# TODO 07 - which top 5 States have the least total sales between 15 April 2019
df['Order Date'] = pd.to_datetime(df['Order Date'], format='%m/%d/%Y')
fil_date = df[(df['Order Date'] >= '2019-04-15') & (df['Order Date'] <= '2019-12
fil_date.groupby('State')['Sales'].sum().sort_values().head(5)
```

```
# TODO 08 - what is the proportion of total sales (%) in West + Central in 2019 e

filter2019 = df[(df['Order Date'].dt.strftime('%Y') == '2019')]
region = ['West','Central']
total_region_sales_sum = filter2019[filter2019['Region'].isin(region)]['Sales'].s
total_sales_sum = filter2019['Sales'].sum()
result = ((total_region_sales_sum/total_sales_sum)*100).round(2)
print(result,"%")
```

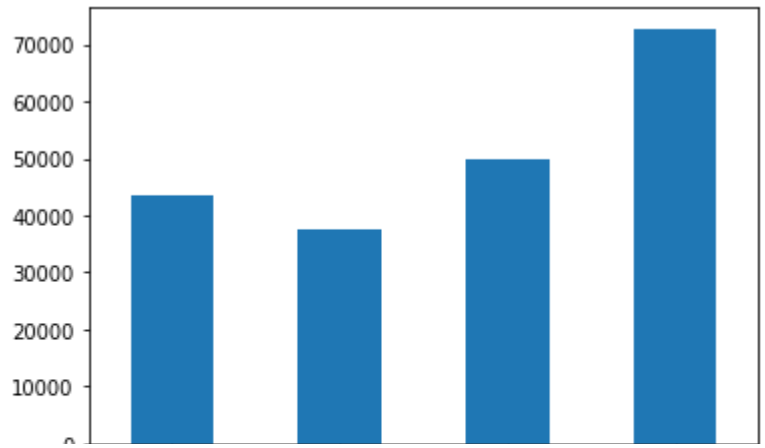
```
54.97 %
```

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

filtertime = df[(df['Order Date'].dt.strftime('%Y') >= '2019') & (df['Order Date'
Sub_cat = filter2019.groupby('Sub-Category')['Quantity'].count().reset_index()
Total_sales = filter2019.groupby('Sub-Category')['Sales'].sum().round(2).reset_in
result = pd.merge(Sub_cat,Total_sales)
result.sort_values(['Quantity','Sales'], ascending = False).head(10)
```

	Sub-Category	Quantity	Sales
3	Binders	415	49683.32
12	Paper	366	20661.89
9	Furnishings	257	27874.12
13	Phones	225	78962.03
14	Storage	210	58788.70
0	Accessories	186	41895.85
2	Art	183	5960.91
5	Chairs	165	83918.64
1	Appliances	114	26050.32

```
# TODO 10 - plot at least 2 plots, any plot you think interesting :)
## 01 Binders sales from 2017 - 2020

filter2017 = df[(df['Order Date'].dt.strftime('%Y') == '2017')]
filter2018 = df[(df['Order Date'].dt.strftime('%Y') == '2018')]
filter2019 = df[(df['Order Date'].dt.strftime('%Y') == '2019')]
filter2020 = df[(df['Order Date'].dt.strftime('%Y') == '2020')]

binders2017 = filter2017[filter2017['Sub-Category'] == 'Binders']
binders2018 = filter2018[filter2018['Sub-Category'] == 'Binders']
binders2019 = filter2019[filter2019['Sub-Category'] == 'Binders']
binders2020 = filter2020[filter2020['Sub-Category'] == 'Binders']

binder_sum_2017 = binders2017['Sales'].sum().round(2)
binder_sum_2018 = binders2018['Sales'].sum().round(2)
binder_sum_2019 = binders2019['Sales'].sum().round(2)
binder_sum_2020 = binders2020['Sales'].sum().round(2)

result = pd.Series([binder_sum_2017,binder_sum_2018,binder_sum_2019,binder_sum_20
        , index=['Binder Sales 2017','Binder Sales 2018','Binder Sales 2019','Bin
result.columns = ['Sales']

result.plot(kind = 'bar');
```

Download




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

filter2019_1 = df[df['Order Date'].dt.strftime('%Y') == '2019']
report = filter2019_1.groupby('Sub-Category')['Sales'].sum().reset_index()
report['Good Sales'] = np.where(report['Sales'] <= 50000, "Bad", "Good")
report
```

	Sub-Category	Sales	Good Sales
0	Accessories	41895.8540	Bad
1	Appliances	26050.3150	Bad
2	Art	5960.9080	Bad
3	Binders	49683.3250	Bad
4	Bookcases	26275.4665	Bad
5	Chairs	83918.6450	Good
6	Copiers	49599.4100	Bad
7	Envelopes	4729.8900	Bad
8	Fasteners	960.1340	Bad
9	Furnishings	27874.1240	Bad
10	Labels	2827.2400	Bad
11	Machines	55906.8860	Good
12	Paper	20661.8940	Bad
13	Phones	78962.0300	Good
14	Storage	58788.7000	Good
15	Supplies	14277.5760	Bad
16	Tables	60833.2005	Good