Final Project - Analyzing Sales Data

Date: 27 November 2022

Author: Sirapat Poolsup (Pangpond)

Course: `Pandas Foundation`

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

```
# preview top 5 rows
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	Hendersc
1	2	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Hendersc
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 Lauderda
4	5	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderda
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 × 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

```
# 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
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 Mes
9991	9992	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mes
9992	9993	CA-2020-121258	2020-02-26	2020-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mes
9993	9994	CA-2020-119914	2020-05-04	2020-05-09	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westminst

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.isna().any(axis=1)]
```

	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-	2018-11-08	2018-11-12	Standard Class	QJ-19255	Quincy Jones	Corporate	United States	Burlington	...

```
# TODO - Explore this dataset on your owns, ask your own questions
#Number of customer group by Ship mode and Segment.
result = df[ ['Ship Mode', 'Segment'] ].value_counts().reset_index().sort_values(
result.columns = ['Ship Mode', 'Segment', 'Counts']
print(result)
```

```
         Ship Mode      Segment  Counts
0   Standard Class     Consumer    3085
```

```
1   Standard Class    Corporate    1812
2   Standard Class  Home Office    1071
3     Second Class     Consumer    1020
5     Second Class    Corporate     609
8     Second Class  Home Office     316
7         Same Day     Consumer     317
10        Same Day    Corporate     114
11        Same Day  Home Office     112
4      First Class     Consumer     769
6      First Class    Corporate     485
9      First Class  Home Office     284
```

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[df.columns[df.isna().sum() > 0]].isna().sum()
```

```
Postal Code    11
dtype: int64
```

```
# TODO 03 - your friend ask for `California` data, filter it and export csv for h
California = df[df['State'] == 'California']

California.to_csv('California.csv')
```

```
# TODO 04 - your friend ask for all order data in `California` and `Texas` in 201
```

```
CL_TX_2017 = df [((df['State'] == 'California') | (df['State'] == 'Texas'))\
                 & (df['Order Date'].dt.strftime('%Y') == '2017')]

CL_TX_2017.to_csv('CL_TX_2017.csv')
```

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

df_2017 = df[df['Order Date'].dt.strftime('%Y') == '2017']

sales_2017 = df_2017['Sales'].agg(['sum','mean','std'])

sales_2017
```

```
sum     484247.498100
mean       242.974159
std        754.053357
Name: Sales, dtype: float64
```

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

df[df['Order Date'].dt.strftime('%Y') == '2018']\
[['Profit', 'Segment']].groupby('Segment').sum().sort_values('Profit', ascending=
```

	Profit
Segment	
Consumer	28460.1665

```
# TODO 07 - which top 5 States have the least total sales between 15 April 2019 -
import datetime
df[ ( df['Order Date'] >= datetime.datetime(2019, 4, 15) ) & (df['Order Date'] <=
    [['State', 'Sales']].groupby('State').sum().sort_values(['Sales'], ascending=
```

	Sales
State	
New Hampshire	49.05
New Mexico	64.08
District of Columbia	117.07
Louisiana	249.80

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

(df[(df['Order Date'].dt.strftime('%Y') == '2019') & ((df['Region'] == 'West') |
/ df[(df['Order Date'].dt.strftime('%Y') == '2019')]['Sales'].sum() ) * 100
```

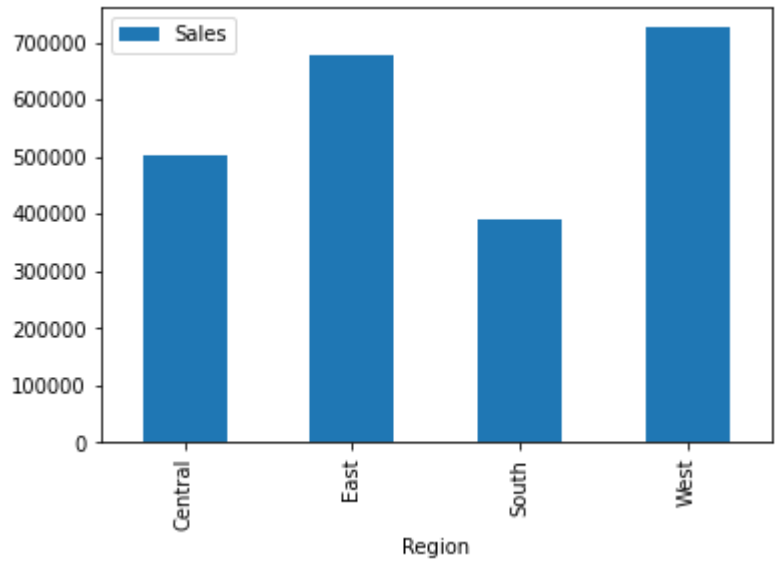
```
54.97479891837763
```

```
# TODO 09 - find top 10 popular products in terms of number of orders vs. total s
top_order = df[(df['Order Date'].dt.strftime('%Y') == '2019') | (df['Order Date']
    .groupby(['Product ID', 'Product Name'])['Order ID'].count().reset_index().so
top_sales = df[(df['Order Date'].dt.strftime('%Y') == '2019') | (df['Order Date']
    .groupby(['Product ID', 'Product Name'])['Sales'].sum().reset_index().sort_va
```

```
# TODO 10 - plot at least 2 plots, any plot you think interesting :)
df[['Region','Sales']].groupby('Region').sum('Sales').plot(kind='bar')
```

```
<AxesSubplot:xlabel='Region'>
```

Download




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