Final Project - Analyzing Sales Data

Date: 29 November 2022

Author: Kulchaya Panacharas (Mink)

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
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 (row, column)
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.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   datetime64[ns]
 3   Ship Date       9994 non-null   datetime64[ns]
 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
```

```
# 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
# How many sales of each Ship Mode that generate in 2018?
df_2018 = df[df['Order Date'].dt.year == int(2018)]
df_2018.groupby('Ship Mode')['Sales'].sum().sort_values(ascending=False).reset_in
```

	Ship Mode	Sales
0	Standard Class	284558.85
1	Second Class	89102.73
2	First Class	69259.44
3	Same Day	27611.49

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

# subset the values from df.shape
print("number of columns : ", df.shape[1])
print("number of rows : ", df.shape[0])
```

```
number of columns :  21
number of rows :  9994
```

```
# TODO 02 - is there any missing values?, if there is, which colunm? how many nan

count_na = df['Postal Code'].isna().sum()
print('Postal Code column has ' + str(count_na) + ' missing values')
```

```
Postal Code column has 11 missing values
```

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

result_California.to_csv('result_California.csv')
```

```
# TODO 04 - your friend ask for all order data in `California` and `Texas` in 201
result_2017_Cali_Tex = df[((df['State'] == 'California') | (df['State'] == 'Texas
        (df['Order Date'].dt.year == int(2017))].reset_index(drop=True)

result_2017_Cali_Tex.to_csv('result_2017_Cali_Tex.csv')
```

```
# TODO 05 - how much total sales, average sales, and standard deviation of sales
df_2017 = df[df['Order Date'].dt.year == int(2017)]
print("Total sales that company make in 2017 is", df_2017['Sales'].sum().round(2)
print("Average sales that company make in 2017 is", df_2017['Sales'].mean().round
print("Standard deviation of sales that company make in 2017 is", df_2017['Sales'
```

```
Total sales that company make in 2017 is 484247.5
Average sales that company make in 2017 is 242.97
Standard deviation of sales that company make in 2017 is 754.05
```

```
# TODO 06 - which Segment has the highest profit in 2018
df_2018 = df[df['Order Date'].dt.year == int(2018)]
answer = df_2018.groupby('Segment')['Profit'].sum()[0].round(2)

print("Consumer has the highest profit in 2018 as" , answer)
```

```
Consumer has the highest profit in 2018 as 28460.17
```

```
# TODO 07 - which top 5 States have the least total sales between 15 April 2019 -
datefix = (df['Order Date'] >= '2019-04-15') & (df['Order Date'] <= '2019-12-31')
new_df = df.loc[datefix]

new_df.groupby('State')['Sales'].sum().sort_values().head(5)
```

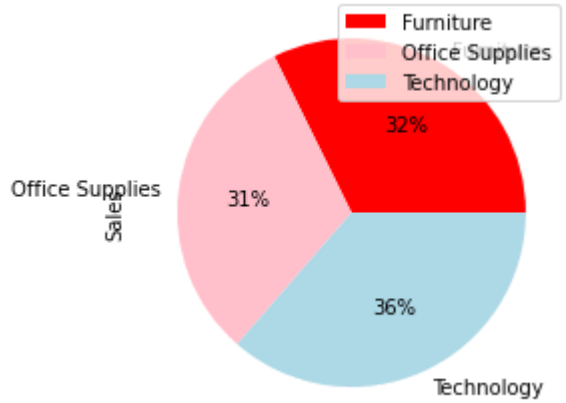
```
State
New Hampshire            49.05
New Mexico               64.08
District of Columbia    117.07
Louisiana               249.80
South Carolina          502.48
Name: Sales, dtype: float64
```

```
# TODO 08 - what is the proportion of total sales (%) in West + Central in 2019 e
df_2019 = df[df['Order Date'].dt.year == int(2019)]
Total = df_2019['Sales'].sum()
Central = df_2019.groupby('Region')['Sales'].sum()[0]
West = df_2019.groupby('Region')['Sales'].sum()[3]

Proportion = (((Central+West)/Total) * 100).round(2)

print(f"The proportion of total sales in West + Central in 2009 is", Proportion,
```

```
The proportion of total sales in West + Central in 2009 is 54.97 %
```

```
# TODO 10 - plot at least 2 plots, any plot you think interesting :)

# pie chart to see the propotion of sales by category
df.groupby(['Category']).sum().plot(kind='pie', y='Sales', autopct='%1.0f%%', col
```

Download




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

df['Status'] = np.where(df['Profit'] > 0, 'Profit', 'loss')
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 × 22 columns