Project

Final Project - Analyzing Sales Data

Date: 29 December 2022

Author: Pattara Pisutvacharakul (Fluke)

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	Count
0	1	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	U
1	2	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	U
2	3	CA-2019-138688	6/12/2019	6/16/2019	Second Class	DV-13045	Darrin Van Huff	Corporate	U
3	4	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	U
4	5	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	U

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   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
```

```
 15  Sub-Category    9994 non-null   object
 16  Product Name    9994 non-null   object
 17  Sales           9994 non-null   float64
 18  Quantity        9994 non-null   int64
 19  Discount        9994 non-null   float64
 20  Profit          9994 non-null   float64
dtypes: datetime64[ns](2), float64(4), int64(2), object(13)
memory usage: 1.6+ MB
```

```
#df = df.drop(labels= 'Ship Data', axis =1)
```

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()
df[["Order Date", "Ship Date"]]
```

	Order Date	Ship Date
0	2019-11-08	2019-11-11
1	2019-11-08	2019-11-11
2	2019-06-12	2019-06-16
3	2018-10-11	2018-10-18
4	2018-10-11	2018-10-18
...	...	...
9989	2017-01-21	2017-01-23
9990	2020-02-26	2020-03-03
9991	2020-02-26	2020-03-03
9992	2020-02-26	2020-03-03
9993	2020-05-04	2020-05-09

9994 rows × 2 columns

```
# TODO - count nan in postal code column
df['Postal Code'].isna().sum()
```

```
11
```

```
# TODO - filter rows with missing values
df[df['Postal Code'].isnull()]
```

	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-	2020-01-19	2020-01-23	Standard Class	RM-19375	Raymond Messe	Consumer	United States	Burlington	...

```
# TODO - Explore this dataset on your owns, ask your own questions
# How many sales and quantity of category in each region?
df.groupby(['Region','Category'])['Sales','Quantity'].agg(['sum']).reset_index()
```

```
<ipython-input-101-29c5468e5fc2>:3: FutureWarning: Indexing with multiple keys (implicitly converted to
  df.groupby(['Region','Category'])['Sales','Quantity'].agg(['sum']).reset_index()
```

	Region	Category	Sales	Quantity
			sum	sum
0	Central	Furniture	163797.1638	1827
1	Central	Office Supplies	167026.4150	5409
2	Central	Technology	170416.3120	1544
3	East	Furniture	208291.2040	2214
4	East	Office Supplies	205516.0550	6462
5	East	Technology	264973.9810	1942
6	South	Furniture	117298.6840	1291
7	South	Office Supplies	125651.3130	3800
8	South	Technology	148771.9080	1118
9	West	Furniture	252612.7435	2696
10	West	Office Supplies	220853.2490	7235
11	West	Technology	251991.8320	2335

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 values?
null_col = df.columns[df.isnull().any()]
null_value = df.isnull().sum().sum()
```

```
print(null_col)
print(null_value)
```

```
Index(['Postal Code'], dtype='object')
11
```

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

# export csv file
result.to_csv("California_data.csv")
```

```
# TODO 04 - your friend ask for all order data in `California` and `Texas` in 2017 (look at Order Date), send him csv file
filter_list = ['California' , 'Texas']
result2 = df[(df['Order Date'].dt.strftime('%Y') == '2017') & (df['State'].isin(filter_list))]

result2.to_csv("California_Texas_2017.csv")
```

```
# TODO 05 - how much total sales, average sales, and standard deviation of sales your company make in 2017
total_sales = df[(df['Order Date'].dt.strftime('%Y') == '2017')]['Sales'].sum()
avg_sales = df[(df['Order Date'].dt.strftime('%Y') == '2017')]['Sales'].mean()
std_sales = df[(df['Order Date'].dt.strftime('%Y') == '2017')]['Sales'].std()

print(f"Total Sales: ${round(total_sales,2)} ")
print(f"Avg. Sales: ${round(avg_sales,2)} ")
print(f"SD. sales: ${round(std_sales,2)} ")
```

```
Total Sales: $484247.5
Avg. Sales: $242.97
SD. sales: $754.05
```

```
# TODO 06 - which Segment has the highest profit in 2018
df[(df['Order Date'].dt.strftime('%Y') == '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 - 31 December 2019
df[(df['Order Date'] >= "2019-04-15") & (df['Order Date'] <= "2019-12-31")]\
.groupby('State')['Sales']\
.sum().sort_values()\
.head(5).reset_index()
```

	State	Sales
0	New Hampshire	49.05
1	New Mexico	64.08
2	District of Columbia	117.07
3	Louisiana	249.80
4	South Carolina	502.48

```
# TODO 08 - what is the proportion of total sales (%) in West + Central in 2019 e.g. 25%


#percent_west_central = df['Region'].value_counts(normalize = True).reset_index() # normalize = True -> %
#percent_west_central

region_count = df[(df['Order Date'].dt.strftime('%Y') == '2019')].groupby('Region')['Sales'].sum().reset_index()
region_count['percent'] = (region_count['Sales'] / region_count['Sales'].sum()) * 100
#region_count

# percent in West + Central
result3 = region_count[ (region_count['Region'] == 'Central') | (region_count['Region'] == 'West')  ]['percent'].sum()
print(f"Percent of total sales in West + Central in 2019 = {round(result3,2)}%")
```

```
Percent of total sales in West + Central in 2019 = 54.97%
```

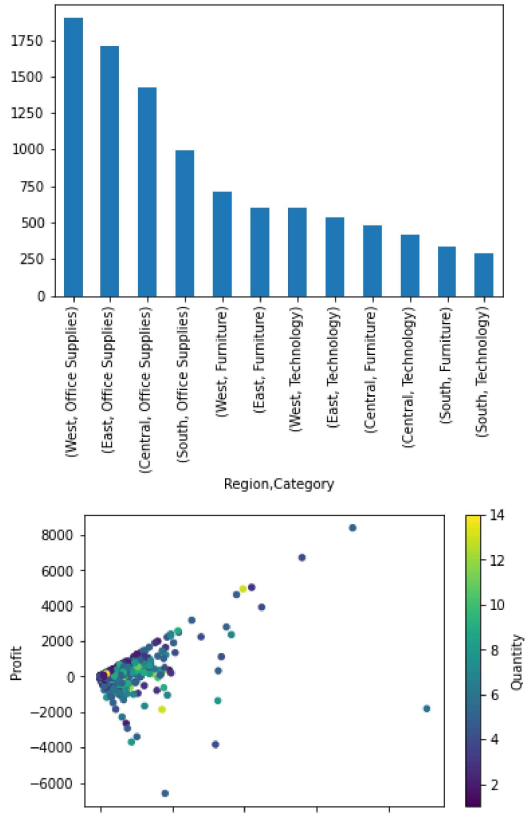
```
# TODO 09 - find top 10 popular products in terms of number of orders vs. total sales during 2019-2020
df[(df['Order Date'] >= "2019-01-01") & (df['Order Date'] <= "2020-12-31")]\
.groupby('Product Name')['Sales']\
.agg(['sum', 'count']).sort_values(by=['sum', 'count'], ascending=False).head(10)
```

Product Name	sum	count
Canon imageCLASS 2200 Advanced Copier	61599.824	5
Hewlett Packard LaserJet 3310 Copier	16079.732	6
3D Systems Cube Printer, 2nd Generation, Magenta	14299.890	2
GBC Ibimaster 500 Manual ProClick Binding System	13621.542	5
GBC DocuBind TL300 Electric Binding System	12737.258	6
GBC DocuBind P400 Electric Binding System	12521.108	4
Samsung Galaxy Mega 6.3	12263.708	5
HON 5400 Series Task Chairs for Big and Tall	11846.562	4
Martin Yale Chadless Opener Electric Letter Opener	11825.902	4
Global Troy Executive Leather Low-Back Tilter	10169.894	7

```
# TODO 10 - plot at least 2 plots, any plot you think interesting :)
# find top 3 orders by category in each region
df[['Region','Category']].value_counts().plot(kind='bar')

import matplotlib.pyplot as plt
df.plot(kind = 'scatter',x='Sales',y='Profit',c='Quantity',colormap='viridis')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6273a484c0>
```




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

# create new column
selected_cols = df[['Order ID','Category','Sub-Category','Product Name', 'Sales', 'Quantity', 'Discount']]
df2 = selected_cols

df2['prices_per_piece'] =  ( df2['Sales'] - df2['Discount'] ) / df2['Quantity']  # mutate
df3 = df2[['Sub-Category', 'Product Name', 'Sales','Quantity','prices_per_piece']]
#np.where( 'prices' > 3000)

df3['Port'] = np.where( (df3['Sales'] >= 230) & (df3['Quantity'] >= 4)  , "Good Port" , "Slow Port") # boolean

```

```
14
15 avg_sale = df['Sales'].mean()
16 avg_qty = df['Quantity'].mean()
17 print(f" Average Sales = {round(avg_sale,2)} ")
18 print(f" Average Quantity = {round(avg_qty,2)} ")
19 df3.head(10)
```

```
 Average Sales = 229.86
 Average Quantity = 3.79
<ipython-input-123-bd20ac1c4104>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.
  df2['prices_per_piece'] =  ( df2['Sales'] - df2['Discount'] ) / df2['Quantity']  # mutate
<ipython-input-123-bd20ac1c4104>:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.
  df3['Port'] = np.where( (df3['Sales'] >= 230) & (df3['Quantity'] >= 4)  , "Good Port" , "Slow Port")
```

	Sub-Category	Product Name	Sales	Quantity	prices_per_piece	Port
0	Bookcases	Bush Somerset Collection Bookcase	261.9600	2	130.980000	Slow Port
1	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs,...	731.9400	3	243.980000	Slow Port
2	Labels	Self-Adhesive Address Labels for Typewriters b...	14.6200	2	7.310000	Slow Port
3	Tables	Bretford CR4500 Series Slim Rectangular Table	957.5775	5	191.425500	Good Port
4	Storage	Eldon Fold 'N Roll Cart System	22.3680	2	11.084000	Slow Port
5	Furnishings	Eldon Expressions Wood and Plastic Desk Access...	48.8600	7	6.980000	Slow Port
						Slow

```
1
```