Final Project - Analyzing Sales Data

Date: 25 February 2023

Author: Patipan Saktanawat (Pattsk138)

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

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 datafram
df['new_order_date'] = pd.to_datetime(df['Order Date'], format='%m/%d/%Y')
df['new_ship_date'] = pd.to_datetime(df['Ship Date'], format='%m/%d/%Y')
```

```
df[['new_order_date','new_ship_date']]
```

	new_order_date	new_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

```
df[df['State']=='Texas']
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region
14	15	US-2018-118983	11/22/2018	11/26/2018	Standard Class	HP-14815	Harold Pawlan	Home Office	United States
15	16	US-2018-118983	11/22/2018	11/26/2018	Standard Class	HP-14815	Harold Pawlan	Home Office	United States
34	35	CA-2020-107727	10/19/2020	10/23/2020	Second Class	MA-17560	Matt Abelman	Home Office	United States
35	36	CA-2019-117590	12/8/2019	12/10/2019	First Class	GH-14485	Gene Hale	Corporate	United States
36	37	CA-2019-117590	12/8/2019	12/10/2019	First Class	GH-14485	Gene Hale	Corporate	United States
...	...	...	...	...	...	...	...	...	...
9919	9920	CA-2019-149272	3/15/2019	3/19/2019	Standard Class	MY-18295	Muhammed Yedwab	Corporate	United States
9920	9921	CA-2019-149272	3/15/2019	3/19/2019	Standard Class	MY-18295	Muhammed Yedwab	Corporate	United States
9961	9962	CA-2018-168088	3/19/2018	3/22/2018	First Class	CM-12655	Corinna Mitchell	Home Office	United States
9962	9963	CA-2018-168088	3/19/2018	3/22/2018	First Class	CM-12655	Corinna Mitchell	Home Office	United States
9972	9973	CA-2019-130225	9/11/2019	9/17/2019	Standard Class	RC-19960	Ryan Crowe	Consumer	United States

985 rows × 23 columns

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

```
# TODO - filter rows with missing values
clean_df = df.dropna().reset_index()
```

```
# TODO - Explore this dataset on your owns, ask your own questions
df['State'].value_counts()
```

```
California              2001
New York                1128
Texas                    985
Pennsylvania             587
Washington               506
Illinois                 492
Ohio                     469
Florida                  383
Michigan                 255
North Carolina           249
Arizona                  224
Virginia                 224
Georgia                  184
Tennessee                183
Colorado                 182
Indiana                  149
Kentucky                 139
Massachusetts            135
New Jersey               130
Oregon                   124
Wisconsin                110
Maryland                 105
Delaware                  96
Minnesota                 89
Connecticut               82
Oklahoma                  66
Missouri                  66
Alabama                   61
Arkansas                  60
Rhode Island              56
Utah                      53
Mississippi               53
Louisiana                 42
South Carolina            42
Nevada                    39
Nebraska                  38
New Mexico                37
Iowa                      30
New Hampshire             27
Kansas                    24
Idaho                     21
Montana                   15
South Dakota              12
Vermont                   11
District of Columbia      10
Maine                      8
North Dakota               7
West Virginia              4
Wyoming                    1
Name: State, dtype: int64
```

Data Analysis Part

Answer 10 below questions to get credit from this course. Write `pandas` code to find answers.

```python
# TODO 01 - how many columns, rows in this dataset
df.shape
```

```
(9994, 23)
```

```python
# TODO 02 - is there any missing values?, if there is, which colunm? how many
df.isna().sum()
```

```
Row ID           0
Order ID         0
Order Date       0
Ship Date        0
Ship Mode        0
Customer ID      0
Customer Name    0
Segment          0
Country/Region   0
City             0
State            0
Postal Code      11
Region           0
Product ID       0
Category         0
Sub-Category     0
Product Name     0
Sales            0
Quantity         0
Discount         0
Profit           0
new_order_date   0
new_ship_date    0
dtype: int64
```

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

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City
2	3	CA-2019-138688	6/12/2019	6/16/2019	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los A
5	6	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los A
6	7	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los A
7	8	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los A
8	9	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los A
...	...	...	...	...	...	...	...	...	...	...
9986	9987	CA-2019-125794	9/29/2019	10/3/2019	Standard Class	ML-17410	Maris LaWare	Consumer	United States	Los A
9990	9991	CA-2020-121258	2/26/2020	3/3/2020	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Cost
9991	9992	CA-2020-121258	2/26/2020	3/3/2020	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Cost
9992	9993	CA-2020-121258	2/26/2020	3/3/2020	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Cost
9993	9994	CA-2020-119914	5/4/2020	5/9/2020	Second Class	CC-12220	Chris Cortes	Consumer	United States	West

2001 rows × 23 columns

```
# TODO 04 - your friend ask for all order data in `California` and `Texas` in
df['order_year'] = df['new_order_date'].apply(lambda x: x.strftime('%Y'))

calitexas = df[((df['State'] == 'California') | (df['State'] == 'Texas')) & (d
calitexas
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	Ci
5	6	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Lo Ar
6	7	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Lo Ar
7	8	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Lo Ar
8	9	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Lo Ar
9	10	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Lo Ar
...	...	...	...	...	...	...	...	...	...	...
9885	9886	CA-2017-112291	4/3/2017	4/8/2017	Standard Class	KE-16420	Katrina Edelman	Corporate	United States	Lo Ar
9903	9904	CA-2017-122609	11/12/2017	11/18/2017	Standard Class	DP-13000	Darren Powers	Consumer	United States	Ca
9904	9905	CA-2017-122609	11/12/2017	11/18/2017	Standard Class	DP-13000	Darren Powers	Consumer	United States	Ca
9942	9943	CA-2017-143371	12/28/2017	1/3/2018	Standard Class	MD-17350	Maribeth Dona	Consumer	United States	Ar
9943	9944	CA-2017-143371	12/28/2017	1/3/2018	Standard Class	MD-17350	Maribeth Dona	Consumer	United States	Ar

632 rows × 24 columns

```
# TODO 05 - how much total sales, average sales, and standard deviation of sal
import pandas as pd
import numpy as np
total_sales = np.sum(df['Sales'])
avg_sales = np.mean(df['Sales'])
stdv_sales = np.std(df['Sales'])
data = round(pd.DataFrame({'total':[total_sales],'average':[avg_sales],'SD':[s
print(data)
```

```
        total  average      SD
0  2297200.86   229.86  623.21
```

```
result = round(df['Sales'].agg(['sum','mean','std']),2)
print(result)
```

```
sum     2297200.86
mean        229.86
std         623.25
Name: Sales, dtype: float64
```

```
# TODO 06 - which Segment has the highest profit in 2018
df[df['order_year'] == '2018'].groupby('Segment')['Profit'].sum().sort_values(
```

```
Segment
Consumer      28460.1665
Corporate     20688.3248
Home Office   12470.1124
Name: Profit, dtype: float64
```

```
# TODO 07 - which top 5 States have the least total sales between 15 April 201
df['Order Date'] = pd.to_datetime(df['Order Date'],format='%m/%d/%Y')
df_range = df[(df['new_order_date'] >= '2019-04-15') & (df['new_order_date'] <
print (df_range)
```

```
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
```

```python
# TODO 08 - what is the proportion of total sales (%) in West + Central in 201
df_sale = df[df['new_order_date'].dt.year == 2019].groupby('Region')['Sales'].
df_sale['%'] = (100* df_sale['Sales']/df_sale['Sales'].sum()).round(2)
ans = df_sale[(df_sale['Region'] == 'West') | (df_sale['Region'] == 'Central')
print (ans)
```

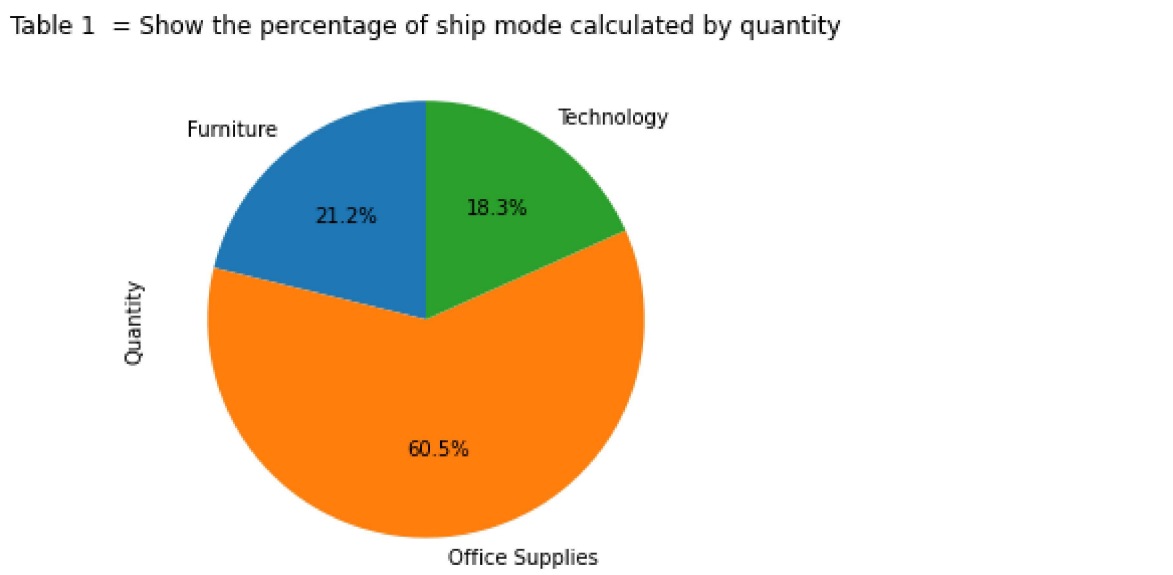
```
54.97
```

```python
# TODO 09 - find top 10 popular products in terms of number of orders vs. tota
df2 = df[(df['new_order_date'].dt.year >= 2019) & (df['new_order_date'].dt.yea

print ("\n Top 10 product in terms of number of orders during 2019-2020")
product = pd.DataFrame(df2.value_counts('Product Name').sort_values(ascending
product.columns = ['Product Name','Total orders']
print(product)


print ("\n Top 10 product in terms of total sales during 2019-2020")
sales = pd.DataFrame(df2.groupby('Product Name')['Sales'].sum().sort_values(as
sales.columns = ['Product Name','Total Sales']
print(sales)
```

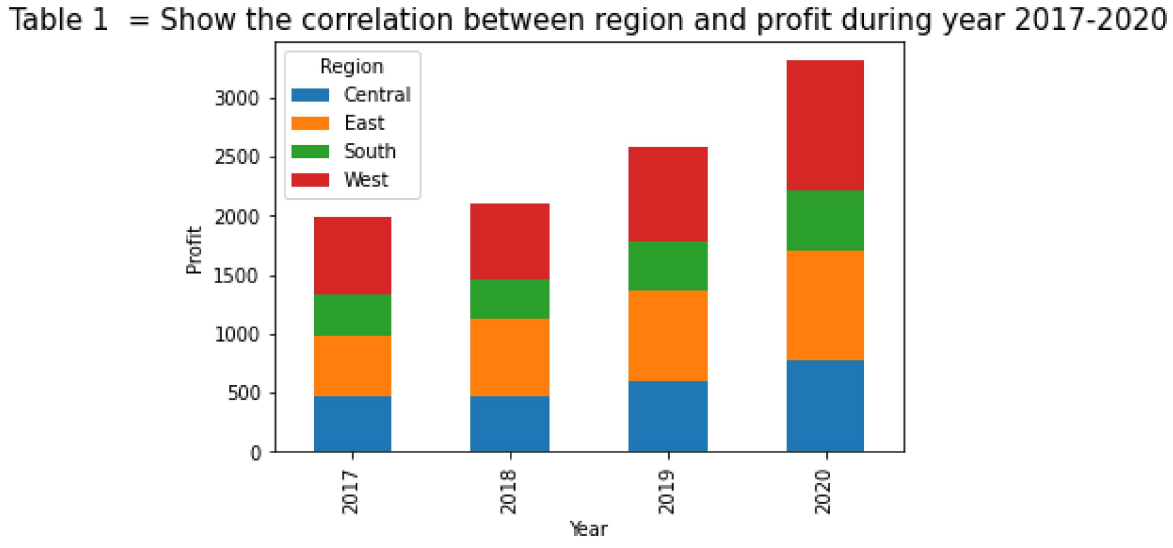
```
 Top 10 product in terms of number of orders during 2019-2020
                                     Product Name  Total orders
0                               Easy-staple paper            27
1                                         Staples            24
2                                 Staple envelope            22
3                         Staples in misc. colors            13
4                                  Staple remover            12
5                         Storex Dura Pro Binders            12
6                Chromcraft Round Conference Tables            12
7  Global Wood Trimmed Manager's Task Chair, Khaki            11
8                         Avery Non-Stick Binders            11
9                       Staple-based wall hangings            10

 Top 10 product in terms of total sales during 2019-2020
                                       Product Name  Total Sales
0              Canon imageCLASS 2200 Advanced Copier     61599.82
1               Hewlett Packard LaserJet 3310 Copier     16079.73
2   3D Systems Cube Printer, 2nd Generation, Magenta     14299.89
3   GBC Ibimaster 500 Manual ProClick Binding System     13621.54
```

```
# TODO 10 - plot at least 2 plots, any plot you think interesting :)
import matplotlib as mpl
import matplotlib.pyplot as plt

df.groupby('Category')['Quantity'].sum()\
.plot.pie(y='Category',  figsize=(5,5), autopct='%1.1f%%', startangle=90)\
.set_title('Table 1  = Show the percentage of ship mode calculated by quantity
plt.show()

table =pd.DataFrame( df.groupby(['order_year','Region'])['Profit'].sum().reset
df.groupby(['order_year','Region']).size().unstack().plot(kind='bar', stacked=
plt.xlabel("Year")
plt.ylabel("Profit")
plt.title('Table 1  = Show the correlation between region and profit during ye
plt.show()
```

Download



Download



```python
# TODO Bonus - use np.where() to create new column in dataframe to help you an
df['profitgambatte'] = np.where(df['Profit'] >= 50,'Great','Gambatte')
```

```python
print(df[['Profit','profitgambatte']])
```

```
        Profit profitgambatte
0      41.9136       Gambatte
1     219.5820          Great
2       6.8714       Gambatte
3    -383.0310       Gambatte
4       2.5164       Gambatte
...        ...            ...
9989    4.1028       Gambatte
9990   15.6332       Gambatte
9991   19.3932       Gambatte
9992   13.3200       Gambatte
9993   72.9480          Great

[9994 rows x 2 columns]
```