Final Project - Analyzing Sales Data

Date: 12 January 2023

Author: Nanfah Kongsathein

Course: `Pandas Foundation`

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

```
# preview top 5 rows
df.head(15)
```

	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	Hender
1	2	CA-2019-152156	11/8/2019	11/11/2019	Second Class	CG-12520	Claire Gute	Consumer	United States	Hender
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 Lauderd
4	5	US-2018-108966	10/11/2018	10/18/2018	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderd
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	11	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
11	12	CA-2017-115812	6/9/2017	6/14/2017	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
12	13	CA-2020-114412	4/15/2020	4/20/2020	Standard Class	AA-10480	Andrew Allen	Consumer	United States	Concor
13	14	CA-2019-161389	12/5/2019	12/10/2019	Standard Class	IM-15070	Irene Maddox	Consumer	United States	Seattle
14	15	US-2018-118983	11/22/2018	11/26/2018	Standard Class	HP-14815	Harold Pawlan	Home Office	United States	Fort Wo

15 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]
```

```python
# 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
```

```python
# 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
# Which state is make slae the most
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.

```
# 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
nan = df.isna().sum()
nan
```

```
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
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_sales.csv')
```

```
# TODO 04 - your friend ask for all order data in `California` and `Texas` in 201
result = df[df['Order Date'].dt.year == 2017] \
                .query("State == 'California' | State == 'Texas'")

result.to_csv('cali_tex_2017.csv')
```

```
result.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City	...	Pos Cod
5	6	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	...	900
6	7	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	...	900
7	8	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	...	900
8	9	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	...	900
9	10	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	...	900

5 rows × 21 columns

```
# TODO 05 - how much total sales, average sales, and standard deviation of sales
result2017 = df[df['Order Date'].dt.year == 2017]
sales2017 = result2017['Sales'].agg(['sum','mean','std'])
sales2017
```

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

```
# TODO 06 - which Segment has the highest profit in 2018
result2018 = df[df['Order Date'].dt.year == 2018]
profit2018 = result2018.groupby('Segment')['Profit']\
    .agg('sum')\
    .sort_values(ascending= False)

profit2018
```

```
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 2019 -
date2019 = df.loc[(df['Order Date'] >= '04/15/2019')
                      & (df['Order Date'] <= '12/31/2019')]
date2019.groupby('State')['Sales'].agg('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
region2019 = df[df['Order Date'].dt.year == 2019].groupby('Region')['Sales'].agg(
sum_region2019 = region2019.sum()
west_cen = region2019.iloc[0] + region2019.iloc[3]
sales_west_central = (west_cen/sum_region2019)*100

sales_west_central
```

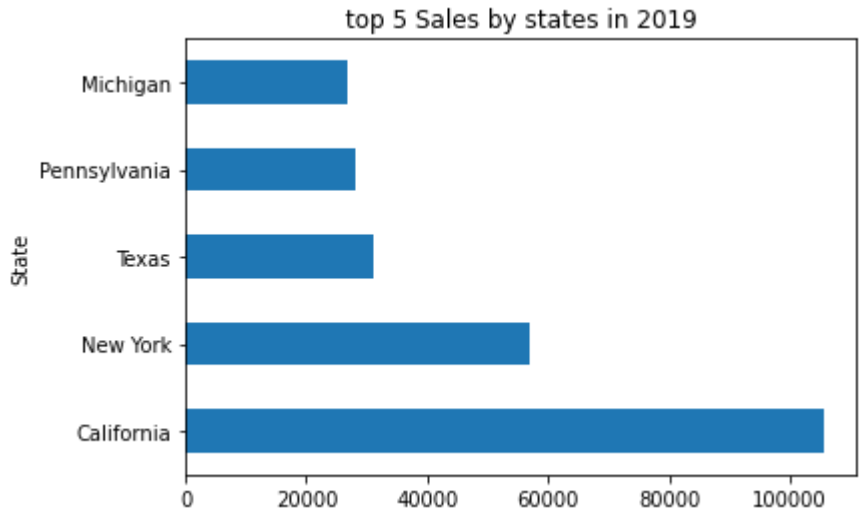
```
54.97479891837763
```

```
# TODO 09 - find top 10 popular products in terms of number of orders vs. total s
df2019_2020 = df.loc[(df['Order Date'] >= '01/01/2019')
                     & (df['Order Date'] <= '12/31/2020')]
top10 = df2019_2020['Product Name'].\
        value_counts().\
        sort_values(ascending=False).\
        head(10).reset_index()


total_sales = df2019_2020.\
              groupby('Product Name')['Sales'].\
              sum().\
              round(2).\
              sort_values(ascending=False).\
              head(10).reset_index()

result_top10 = pd.concat([top10, total_sales], axis=1)
result_top10.columns = ['Products top 10 hignest order','number order','product t
result_top10
```

	Products top 10 hignest order	number order	product top 10 high total sales	Sales
0	Easy-staple paper	27	Canon imageCLASS 2200 Advanced Copier	61599.82
1	Staples	24	Hewlett Packard LaserJet 3310 Copier	16079.73
2	Staple envelope	22	3D Systems Cube Printer, 2nd Generation, Magenta	14299.89
3	Staples in misc. colors	13	GBC Ibimaster 500 Manual ProClick Binding System	13621.54
4	Chromcraft Round Conference Tables	12	GBC DocuBind TL300 Electric Binding System	12737.26
5	Storex Dura Pro Binders	12	GBC DocuBind P400 Electric Binding System	12521.11
6	Staple remover	12	Samsung Galaxy Mega 6.3	12263.71
7	Global Wood Trimmed Manager's Task Chair, Khaki	11	HON 5400 Series Task Chairs for Big and Tall	11846.56
8	Avery Non-Stick Binders	11	Martin Yale Chadless Opener Electric Letter Op...	11825.90
9	Sterilite Officeware Hinged File Box	10	Global Troy Executive Leather Low-Back Tilter	10169.89

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

total_sales_2019 = date2019.groupby('State')['Sales'].agg('sum').sort_values(asce
    plot(kind = 'barh', title='top 5 Sales by states in 2019')
```

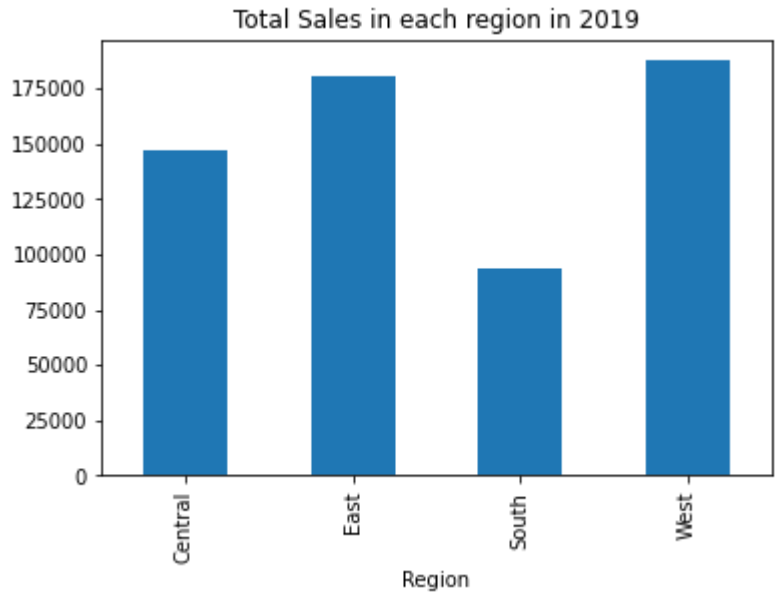
Download



```
region2019.plot(kind = 'bar', title = 'Total Sales in each region in 2019')
```

```
<AxesSubplot:title={'center':'Total Sales in each region in 2019'}, xlabel='Reg
```

Download




```
import numpy as np

# TODO Bonus - use np.where() to create new column in dataframe to help you answe
#Loss and profit
df['profit_loss'] = np.where(df['Profit']>0, "profit","loss")

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	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	...
5	6	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	...
6	7	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	...
7	8	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	...
8	9	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	...
9	10	CA-2017-115812	2017-06-09	2017-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	...

10 rows × 22 columns