Final Project - Analyzing Sales Data

Date: 11 November

Author: Natthamon Ratthanasurakarn

Course: `Pandas Foundation`

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

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

```
# 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')
```

```
# TODO - convert order date and ship date to datetime in the original dataframe
```

```
df[['Order Date','Ship Date']] = df[['Order Date','Ship Date']].apply(pd.to_datet
df.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City	...	I
0	1	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	4
1	2	CA-2019-152156	2019-11-08	2019-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	4
2	3	CA-2019-138688	2019-06-12	2019-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	9
3	4	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	3
4	5	US-2018-108966	2018-10-11	2018-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	3

5 rows × 21 columns

```
# 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
```

```
## To explore how many customers in each State
df.groupby('State')['Customer ID'].count().sort_values(ascending=False)
```

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)
```

ans - **9994** rows and **21** columns

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

```
df.isna().sum()
```

ans - There is **11** missing values in **'Postal Code'** Column.

```
# TODO 03 - your friend ask for `California` data, filter it and export csv for h
```

```
df[ df['State'] == 'California' ].to_csv('California_cust.csv')
```

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

```
import datetime as dt
cali_tex_2017 = df[ (df['Order Date'].dt.strftime('%Y') == '2017') & (df['State']
cali_tex_2017.to_csv('cali_tex_2017.csv')
```

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

```
import numpy as np
df[df['Order Date'].dt.strftime('%Y') == '2017']['Sales'].agg(['sum','mean','std'
```

ans

- total sales = 484,247.50
- average sales = 242.97
- standard deviation of sales = 754.05

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

```
df[df['Order Date'].dt.year == 2018].\
    groupby('Segment')['Profit'].sum().\
    sort_values(ascending=False).head(1)
```

ans - **Consumer** segment has the highest profit in 2018.

```
# TODO 07 - which top 5 States have the least total sales between 15 April 2019 -
```

```
# let's say filter Order Date between 2019/04/15 and 2019/12/31
df[ (df['Order Date'] > '2019-04-15') & (df['Order Date'] < '2019-12-31')].\
    groupby('State')['Sales'].sum().\
    sort_values().\
    head(5)
```

ans - top 5 States have the least total sales between '2019-04-15' and '2019-12-31'

1. New Hampshire
2. New Mexico
3. District of Columbia
4. Louisiana
5. South Carolina

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

```
filtered_df = df[df['Order Date'].dt.year == 2019].groupby('Region')['Sales'].sum
total_west_cent = sum(filtered_df[['West','Central']])
total = filtered_df.sum()
proportion = round(((total_west_cent/total)*100),ndigits=2)
print(f"Total sales in West and Central in 2019 : {total_west_cent}.")
print(f"Total sales in 2019: {total}.")
print(f"Proportion : {proportion} %")
```

```
Total sales in West and Central in 2019 : 334909.5525.
Total sales in 2019: 609205.598.
Proportion : 54.97 %
```

```
# TODO 09 - find top 10 popular products in terms of number of orders vs. total s
```

```
filtered_year = df[ df['Order Date'].dt.year.isin([2019,2020]) ]
df_byprod = filtered_year.groupby('Product Name')['Sales'].agg(['count','sum']).r

top10_orders = df_byprod.sort_values('count',ascending=False).head(10)
top10_sales = df_byprod.sort_values('sum',ascending=False).head(10)

print(f"Here is top 10 products in term of 'number of orders' :\n")
print(top10_orders)
print()
print(f"Here is top 10 products in term of 'total sales' :\n")
print(top10_sales)
```

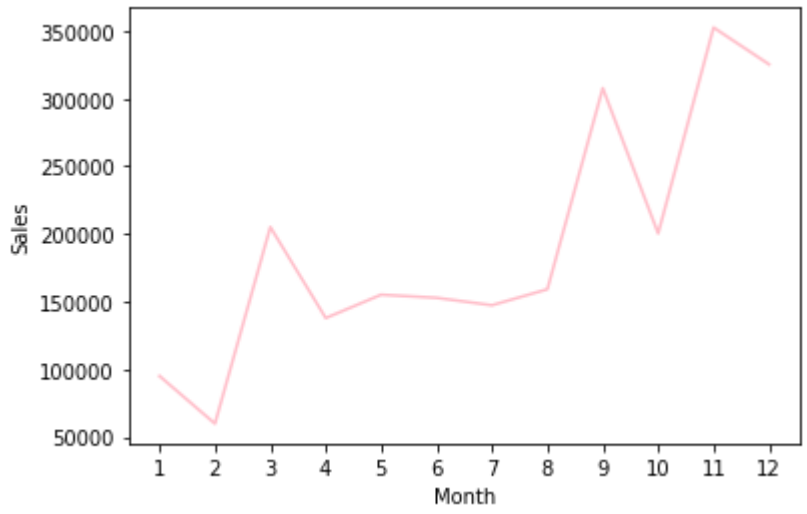
```
Here is top 10 products in term of 'number of orders' :

                                        Product Name  count       sum
512                                 Easy-staple paper     27  1481.728
1412                                          Staples     24   462.068
1406                                  Staple envelope     22   644.936
1413                          Staples in misc. colors     13   357.164
1409                                   Staple remover     12   204.512
1421                          Storex Dura Pro Binders     12   176.418
411                 Chromcraft Round Conference Tables     12  7965.053
732   Global Wood Trimmed Manager's Task Chair, Khaki     11  2793.086
250                           Avery Non-Stick Binders     11   122.128
1410                        Staple-based wall hangings     10   233.392
```

```
Here is top 10 products in term of 'total sales' :

                                            Product Name  count        sum
388               Canon imageCLASS 2200 Advanced Copier      5  61599.824
765                Hewlett Packard LaserJet 3310 Copier      6  16079.732
18     3D Systems Cube Printer, 2nd Generation, Magenta      2  14299.890
```

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

```
df.groupby(df['Order Date'].dt.month)['Sales'].sum().\
    plot(kind='line',xlabel='Month',ylabel='Sales',xticks=list(range(1,13)),color
```

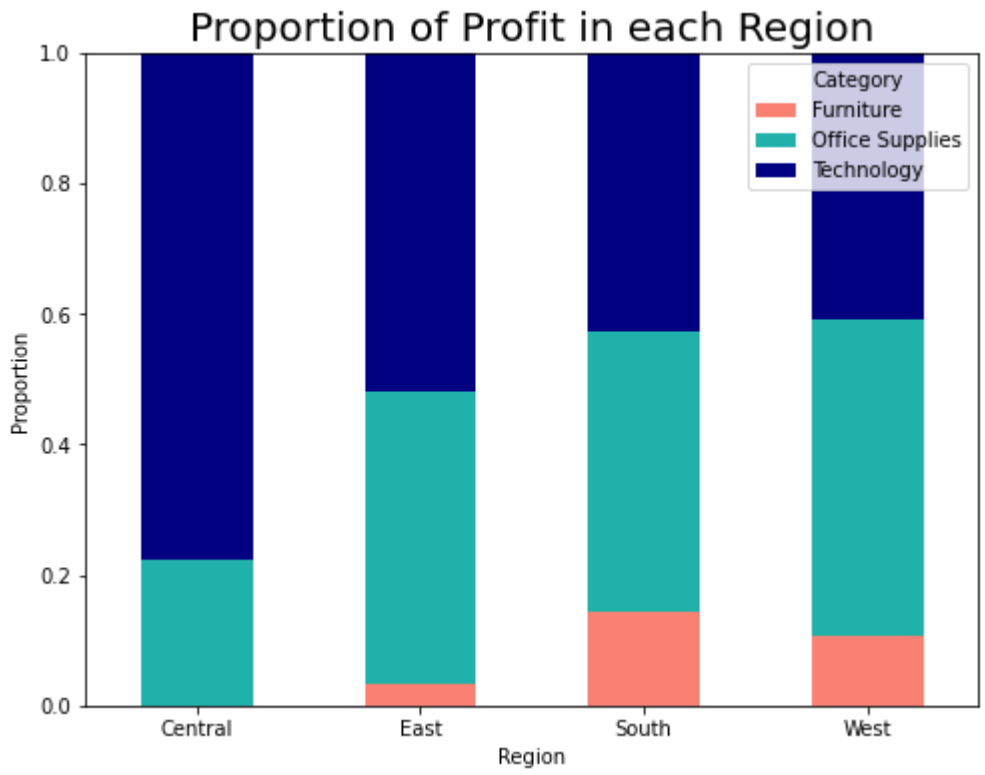
Download




```
# calculate sum of profit grouping by Region and Category
cal_result = df.groupby(['Region','Category'])['Profit'].sum().reset_index()

# calculate proportion
cal_result['proportion'] = cal_result['Profit']/cal_result.groupby('Region')['Pro
print(cal_result)
# transform a long format dataframe to a wide format dataframe
pivot = pd.pivot_table(data=cal_result, index=['Region'], columns=['Category'], v

# create stacked bar chart
ax = pivot.plot.bar(stacked=True,color=['salmon','lightseagreen','navy'] ,figsize
ax.set_title('Proportion of Profit in each Region', fontsize=20)
ax.set_ylim(0,1)
ax.set_xticklabels(['Central','East','South','West'], rotation=0)
```

```
     Region         Category      Profit  proportion
0   Central        Furniture  -2871.0494   -0.072307
1   Central  Office Supplies   8879.9799    0.223641
2   Central       Technology  33697.4320    0.848666
3      East        Furniture   3046.1658    0.033283
4      East  Office Supplies  41014.5791    0.448135
5      East       Technology  47462.0351    0.518582
6     South        Furniture   6771.2061    0.144840
7     South  Office Supplies  19986.3928    0.427522
8     South       Technology  19991.8314    0.427638
9      West        Furniture  11504.9503    0.106116
10     West  Office Supplies  52609.8490    0.485248
11     West       Technology  44303.6496    0.408636


[Text(0, 0, 'Central'),
 Text(1, 0, 'East'),
 Text(2, 0, 'South'),
 Text(3, 0, 'West')]
```

Download




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

```
import numpy as np
df['Profit/Loss'] = 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 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 × 22 columns