E commerce sales analysis by python

importing libraries

In [356…

```
import pandas as pd
#  data cleaning, manipulation, and analysis
```

In [357…

```
import numpy as np
#  numerical computation
import plotly.express as px
# px visualization library
import matplotlib.pyplot as plt
#  basic plotting
import plotly.graph_objects as go
#  go advance and customer graph
import plotly.io as pio
#  pio help to customize  the graph templates
import plotly.colors as colors
#  colors for colors and Pick and use predefined color scales and color sets
pio.templates.default = "plotly_white"
# This line sets the default style (theme) for all your Plotly charts to use the
```

creating the data frame

In [358…

```
data=pd.read_csv("Sample - Superstore.csv ",encoding='latin-1') # Load the datas
```

In [359…

```
data
```

Out[359…

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segme
0	1	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consun
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consun
2	3	CA-2016-138688	6/12/2016	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corpora
3	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consun
4	5	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consun
...	...	...	...	...	...	...	...	
9989	9990	CA-2014-110422	1/21/2014	1/23/2014	Second Class	TB-21400	Tom Boeckenhauer	Consun
9990	9991	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	DB-13060	Dave Brooks	Consun
9991	9992	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	DB-13060	Dave Brooks	Consun
9992	9993	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	DB-13060	Dave Brooks	Consun
9993	9994	CA-2017-119914	5/4/2017	5/9/2017	Second Class	CC-12220	Chris Cortes	Consun

9994 rows × 21 columns

Let’s start by looking at the descriptive statistics of the dataset

In [360…

```
data.head() # Display the first few rows of the DataFrame
```

Out[360…

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Co
0	1	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	U
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	U
2	3	CA-2016-138688	6/12/2016	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	U
3	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	U
4	5	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	U

5 rows × 21 columns

In [361…

```
data.describe()
#  for descriptive statistics
```

Out[361…

	Row ID	Postal Code	Sales	Quantity	Discount	Profit
count	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000
mean	4997.500000	55190.379428	229.858001	3.789574	0.156203	28.656896
std	2885.163629	32063.693350	623.245101	2.225110	0.206452	234.260108
min	1.000000	1040.000000	0.444000	1.000000	0.000000	-6599.978000
25%	2499.250000	23223.000000	17.280000	2.000000	0.000000	1.728750
50%	4997.500000	56430.500000	54.490000	3.000000	0.200000	8.666500
75%	7495.750000	90008.000000	209.940000	5.000000	0.200000	29.364000
max	9994.000000	99301.000000	22638.480000	14.000000	0.800000	8399.976000

In [362…

```
data.info() #  for basic information about the DataFrame
```

```
<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        9994 non-null   object 
 9   City           9994 non-null   object 
 10  State          9994 non-null   object 
 11  Postal Code    9994 non-null   int64  
 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: float64(3), int64(3), object(15)
memory usage: 1.6+ MB
```

In [363…

```
data.isnull().sum()
#  checking null values in the dataset
```

Out[363...

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

Converting Date Columns

In [364...

```
# converting date(object) column in date
data['Order Date']=pd.to_datetime(data["Order Date"])
```

In [365...

```
# converting date(object) column in date
data['Ship Date']=pd.to_datetime(data["Ship Date"])
```

In [366...

```
data.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        9994 non-null   object        
 9   City           9994 non-null   object        
 10  State          9994 non-null   object        
 11  Postal Code    9994 non-null   int64         
 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(3), int64(3), object(13)
memory usage: 1.6+ MB
```

In [367…

```
# recall the data
data
```

Out[367…

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Coun
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	Unit Stat
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	Unit Stat
2	3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	Unit Stat
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unit Stat
4	5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unit Stat
...	...	...	...	...	...	...	...	...	
9989	9990	CA-2014-110422	2014-01-21	2014-01-23	Second Class	TB-21400	Tom Boeckenhauer	Consumer	Unit Stat
9990	9991	CA-2017-121258	2017-02-26	2017-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	Unit Stat
9991	9992	CA-2017-121258	2017-02-26	2017-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	Unit Stat
9992	9993	CA-2017-121258	2017-02-26	2017-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	Unit Stat
9993	9994	CA-2017-119914	2017-05-04	2017-05-09	Second Class	CC-12220	Chris Cortes	Consumer	Unit Stat

9994 rows × 21 columns

question 1:

1. you need to find the monthly sales of store aand also find which month has the highest sales and which month had the lowest sales

Adding New Date-Based Columns

In [368…

```
data["Order Month"]=data["Order Date"].dt.month
data["Order Year"]=data["Order Date"].dt.year
data["order Day of Week"]=data["Order Date"].dt.dayofweek       #
```

In [369…

```
data
```

Out[369…

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Count
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	Unit Stat
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	Unit Stat
2	3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	Unit Stat
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unit Stat
4	5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unit Stat
...	...	...	...	...	...	...	...	...	
9989	9990	CA-2014-110422	2014-01-21	2014-01-23	Second Class	TB-21400	Tom Boeckenhauer	Consumer	Unit Stat
9990	9991	CA-2017-121258	2017-02-26	2017-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	Unit Stat
9991	9992	CA-2017-121258	2017-02-26	2017-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	Unit Stat
9992	9993	CA-2017-121258	2017-02-26	2017-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	Unit Stat
9993	9994	CA-2017-119914	2017-05-04	2017-05-09	Second Class	CC-12220	Chris Cortes	Consumer	Unit Stat

9994 rows × 24 columns

In [370…

```
data.head()
```

Out[370…

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Hei
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Hei
2	3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lau
4	5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lau

5 rows × 24 columns

Monthly Sales Analysis

In [371…

```
sales_by_month = data.groupby("Order Month")["Sales"].sum().reset_index()
```

In [372…

```
sales_by_month
```

Out[372…

	Order Month	Sales
0	1	94924.8356
1	2	59751.2514
2	3	205005.4888
3	4	137762.1286
4	5	155028.8117
5	6	152718.6793
6	7	147238.0970
7	8	159044.0630
8	9	307649.9457
9	10	200322.9847
10	11	352461.0710
11	12	325293.5035

In [373…

```
sales_by_month.max()
```

Out[373…

```
Order Month        12.000
Sales          352461.071
dtype: float64
```

In [374…

```
sales_by_month.min()
```

Out[374…

```
Order Month        1.0000
Sales          59751.2514
dtype: float64
```

In [375…

```
fig=px.line(sales_by_month,
           x="Order Month",
           y="Sales",
title="Monthly Sales Analytics")
fig.show()
```

question 2

2. You need to analyze sales based on product categories and determine which category has the lowest sales and which category has the highest sales.

In [376…

```
data.head()
```

Out[376...

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Hei
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Hei
2	3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lau
4	5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lau

5 rows × 24 columns

Sales Analysis by Category

In [377...

```
sales_by_category=data.groupby("Category")["Sales"].sum().reset_index()
```

In [378...

```
sales_by_category
```

Out[378...

	Category	Sales
0	Furniture	741999.7953
1	Office Supplies	719047.0320
2	Technology	836154.0330

In [379...

```
sales_by_category.max()
```

Out[379...

```
Category     Technology
Sales        836154.033
dtype: object
```

In [380…

```
sales_by_category.min()
```

Out[380…

```
Category      Furniture
Sales        719047.032
dtype: object
```

In [381…

```
 # names = What the slices are called (labels)
# 🔵 values = How big each slice is (size)
fig=px.pie(sales_by_category,
          names="Category",
          values="Sales",
           hole=0.4,
          title="Sales By Categories",
          color_discrete_sequence=px.colors.qualitative.Pastel)
fig.show()
```

question 3:

3. The sales analysis needs to be done based on sub-categories

In [382…

```
data.head()
```

Out[382...

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Her
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Her
2	3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lau
4	5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lau

5 rows × 24 columns

sales analysis by sub category

In [383...

```
sales_by_subcategory=data.groupby("Sub-Category")["Sales"].sum().reset_index()
```

In [384...

```
sales_by_subcategory
```

Out[384...

	Sub-Category	Sales
0	Accessories	167380.3180
1	Appliances	107532.1610
2	Art	27118.7920
3	Binders	203412.7330
4	Bookcases	114879.9963
5	Chairs	328449.1030
6	Copiers	149528.0300
7	Envelopes	16476.4020
8	Fasteners	3024.2800
9	Furnishings	91705.1640
10	Labels	12486.3120
11	Machines	189238.6310
12	Paper	78479.2060
13	Phones	330007.0540
14	Storage	223843.6080
15	Supplies	46673.5380
16	Tables	206965.5320

In [385...

```
sales_by_subcategory.max()
```

Out[385...

```
Sub-Category         Tables
Sales            330007.054
dtype: object
```

In [386...

```
sales_by_subcategory.min()
```

Out[386...

```
Sub-Category    Accessories
Sales               3024.28
dtype: object
```

In [387...

```
# fig by line in plotly graph
fig=px.line(sales_by_subcategory,
            x="Sub-Category",
            y="Sales",
            title="sales by sub_ Category")
fig.show()
```

In [388...

```
# by pie in plotly
fig=px.pie(sales_by_subcategory,
          names="Sub-Category",
          values="Sales",
           title="Sales by sub categories",
          hole=0.5)
fig.show()
```

In [389…

```
# bar graph in plotly
fig=px.bar(sales_by_subcategory,
           x="Sub-Category",
            y="Sales",
          title="Sales by sub categories")
fig.show()
```

In [390…

```
data.head()
```

Out[390…

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Hei
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Hei
2	3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	.
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lau
4	5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lau

5 rows × 24 columns

question 4:

4. You need to analyze the monthly profit from sales and determine which month had the highest profit.

In [391…

```
data.head()
```

Out[391…

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Hei
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Hei
2	3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lau
4	5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lau

5 rows × 24 columns

Monthly Profit Analysis

In [392…

```
monthly_profit =data.groupby("Order Month")["Profit"].sum().reset_index()
```

In [393…

```
monthly_profit
```

Out[393…

	Order Month	Profit
0	1	9134.4461
1	2	10294.6107
2	3	28594.6872
3	4	11587.4363
4	5	22411.3078
5	6	21285.7954
6	7	13832.6648
7	8	21776.9384
8	9	36857.4753
9	10	31784.0413
10	11	35468.4265
11	12	43369.1919

In [394…

```
monthly_profit.max()
```

Out[394…

```
Order Month       12.0000
Profit         43369.1919
dtype: float64
```

In [395…

```
monthly_profit.min()
```

Out[395…

```
Order Month       1.0000
Profit         9134.4461
dtype: float64
```

In [396…

```
# bar graph
fig=px.bar(monthly_profit,
          x="Order Month",y="Profit",
          title="Mothly Profit By sales")
fig.show()
```

In [397…

```
# line graph
fig=px.line(monthly_profit,
          x="Order Month",y="Profit",
          title="Mothly Profit By sales")
fig.show()
```

In [398…

```
# #  question 5
#  Analyze the profit by category and sub-category.
```

profit by category

In [399…

```
profit_by_category=data.groupby("Category")["Profit"].sum().reset_index()
```

In [400…

```
profit_by_category
```

Out[400…

	Category	Profit
0	Furniture	18451.2728
1	Office Supplies	122490.8008
2	Technology	145454.9481

In [401…

```
# figure in pie graph
fig=px.pie(profit_by_category,
            values="Profit",
           names="Category",
            title="profit by categories",
          hole=0.5)
fig.update_traces(textposition="inside",textinfo="percent+label")
fig.update_layout(title_text="Profit Analysis By Category", title_font=dict(size

fig.show()
```

profit by sub category

In [402…

```
profit_by_sub_category=data.groupby("Sub-Category")["Profit"].sum().reset_index(
```

In [403…

```
profit_by_sub_category
```

Out[403...

	Sub-Category	Profit
0	Accessories	41936.6357
1	Appliances	18138.0054
2	Art	6527.7870
3	Binders	30221.7633
4	Bookcases	-3472.5560
5	Chairs	26590.1663
6	Copiers	55617.8249
7	Envelopes	6964.1767
8	Fasteners	949.5182
9	Furnishings	13059.1436
10	Labels	5546.2540
11	Machines	3384.7569
12	Paper	34053.5693
13	Phones	44515.7306
14	Storage	21278.8264
15	Supplies	-1189.0995
16	Tables	-17725.4811

In [404...

```
#  fig in bar graph in plotly library
fig=px.bar(profit_by_sub_category,
            y="Profit",
           x="Sub-Category",
            title="profit by sub_categories",
        )

fig.show()
```

In [405...

```
# line graph
fig=px.line(profit_by_sub_category,
            x="Sub-Category",
            y="Profit",
            title="Profit by Sub-Category")
fig.show()
```

question 6:

6.Analyze the sales and profit by customer segment

call the data

In [406…

```
data.head()
```

Out[406…

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Hei
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Hei
2	3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lau
4	5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lau

5 rows × 24 columns

In [407…

```
sales_profit_by_segment=data.groupby("Segment").agg({"Sales":"sum","Profit":"sum
```

In [408…

```
sales_profit_by_segment
```

Out[408…

	Segment	Sales	Profit
0	Consumer	1.161401e+06	134119.2092
1	Corporate	7.061464e+05	91979.1340
2	Home Office	4.296531e+05	60298.6785

In [409…

```
import plotly.graph_objects as go
fig=go.Figure()
fig.add_traces(go.Bar(x=sales_profit_by_segment["Segment"],
                y=sales_profit_by_segment["Sales"],
              name="Sales"))


fig.add_traces(go.Bar(x=sales_profit_by_segment["Segment"],
              y=sales_profit_by_segment["Profit"],
              name="profit"))
fig.update_layout(title="sales and profit analysis by customer segment column",
                 xaxis_title="Customer segment" , yaxis_title="Amount")
fig.show()


#  # go.figure is Multiple bar graphs (or other charts) together in one chart (c
# go.Figure() = an empty whiteboard 🧼
# go.Bar(...) = stickers (bars) 📊 you place on that whiteboard
# fig.show() = shows the whiteboard (your final chart)
```

question 7:

7. Analyze the sales to profit ratio

In [410…

```
sales_profit_by_segmemt = data.groupby("Segment").agg({"Sales":"sum","Profit":"s
```

In [411…

```
sales_profit_by_segmemt
```

Out[411…

	Sales	Profit
Segment		
Consumer	1.161401e+06	134119.2092
Corporate	7.061464e+05	91979.1340
Home Office	4.296531e+05	60298.6785

In [412…

```
sales_profit_by_segmemt["Sales_to_Profit_Ratio"]=sales_profit_by_segmemt["Sales"
```

In [413…

```
sales_profit_by_segmemt["Sales_to_Profit_Ratio"]
```

Out[413…

```
Segment
Consumer       8.659471
Corporate      7.677245
Home Office    7.125416
Name: Sales_to_Profit_Ratio, dtype: float64
```

Geographical Analysis

question 8:

8. find Sales and profit by region?

In [414…

```
# Sales and profit by region
region_analysis = data.groupby("Region")[["Sales", "Profit"]].sum().reset_index(
print(region_analysis)
```

```
    Region        Sales        Profit
0  Central  501239.8908   39706.3625
1     East  678781.2400   91522.7800
2    South  391721.9050   46749.4303
3     West  725457.8245  108418.4489
```

Customer Analysis

question 9:

9. find top 10 customers by sales?

In [415…

```
# Top 10 customers by sales
top_customers = data.groupby("Customer Name")["Sales"].sum().sort_values(ascendi
print(top_customers)
```

```
Customer Name
Sean Miller           25043.050
Tamara Chand          19052.218
Raymond Buch          15117.339
Tom Ashbrook          14595.620
Adrian Barton         14473.571
Ken Lonsdale          14175.229
Sanjit Chand          14142.334
Hunter Lopez          12873.298
Sanjit Engle          12209.438
Christopher Conant    12129.072
Name: Sales, dtype: float64
```

Discount Impact

question 10:

10. how does discount affect profit?

In [416…

```
# Correlation between discount and profit
discount_profit_corr = data["Discount"].corr(data["Profit"])
print("Correlation between Discount and Profit:", discount_profit_corr)
```

```
Correlation between Discount and Profit: -0.21948745637176834
```

Shipping Analysis

question 11:

10. Calculate shipping duration

In [417…

```
# Calculate shipping duration
data["Shipping Duration"] = (data["Ship Date"] - data["Order Date"]).dt.days
print(data[["Shipping Duration", "Sales", "Profit"]].head())
```

```
   Shipping Duration     Sales    Profit
0                  3  261.9600   41.9136
1                  3  731.9400  219.5820
2                  4   14.6200    6.8714
3                  7  957.5775 -383.0310
4                  7   22.3680    2.5164
```

In [418…

```
#Visualize Shipping Duration Distribution
import plotly.express as px
fig = px.histogram(data, x="Shipping Duration", nbins=20, title="Distribution of
fig.show()
```

Conclusion:

The E-commerce sales analysis provided valuable insights into sales, profit, and customer behavior. Key findings include:

- **Monthly Trends:** Identified months with highest and lowest sales and profit.
- **Category Analysis:** Determined top-performing and underperforming product categories and sub-categories.
- **Customer Segments:** Analyzed sales and profit distribution across customer segments.
- **Geographical Insights:** Compared sales and profit by region.
- **Discount Impact:** Found correlation between discounts and profit.
- **Shipping Analysis:** Explored shipping duration and its effect on customer satisfaction.

These insights can help optimize inventory, marketing strategies, and logistics to