Demo

February 14, 2021

[1]:
```
#import libraries
import pandas as pd
df=pd .read_excel("C:/Users/OSAGIE/Desktop/Copy of Sample - Superstore.
 ↪xls",sheet_name='Orders')
```

[2]:
```
df.head()
```

[2]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	\
0	1	CA-2018-152156	2018-11-08	2018-11-11	Second Class	CG-12520	
1	2	CA-2018-152156	2018-11-08	2018-11-11	Second Class	CG-12520	
2	3	CA-2018-138688	2018-06-12	2018-06-16	Second Class	DV-13045	
3	4	US-2017-108966	2017-10-11	2017-10-18	Standard Class	SO-20335	
4	5	US-2017-108966	2017-10-11	2017-10-18	Standard Class	SO-20335	

	Customer Name	Segment	Country/Region	City	…	\
0	Claire Gute	Consumer	United States	Henderson	…	
1	Claire Gute	Consumer	United States	Henderson	…	
2	Darrin Van Huff	Corporate	United States	Los Angeles	…	
3	Sean O'Donnell	Consumer	United States	Fort Lauderdale	…	
4	Sean O'Donnell	Consumer	United States	Fort Lauderdale	…	

	Postal Code	Region	Product ID	Category	Sub-Category	\
0	42420.0	South	FUR-BO-10001798	Furniture	Bookcases	
1	42420.0	South	FUR-CH-10000454	Furniture	Chairs	
2	90036.0	West	OFF-LA-10000240	Office Supplies	Labels	
3	33311.0	South	FUR-TA-10000577	Furniture	Tables	
4	33311.0	South	OFF-ST-10000760	Office Supplies	Storage	

	Product Name	Sales	Quantity	\
0	Bush Somerset Collection Bookcase	261.9600	2	
1	Hon Deluxe Fabric Upholstered Stacking Chairs,…	731.9400	3	
2	Self-Adhesive Address Labels for Typewriters b…	14.6200	2	
3	Bretford CR4500 Series Slim Rectangular Table	957.5775	5	
4	Eldon Fold 'N Roll Cart System	22.3680	2	

	Discount	Profit
0	0.00	41.9136

```
1      0.00  219.5820
2      0.00    6.8714
3      0.45 -383.0310
4      0.20    2.5164

[5 rows x 21 columns]
```

[3]:
```
#shape of the dataset
df.shape
```

[3]:
```
(9994, 21)
```

[4]:
```
#information about the dataset
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
```

[5]:
```
from autoviz.AutoViz_Class import AutoViz_Class
AV = AutoViz_Class()
```

```
Imported AutoViz_Class version: 0.0.72. Call using:
```

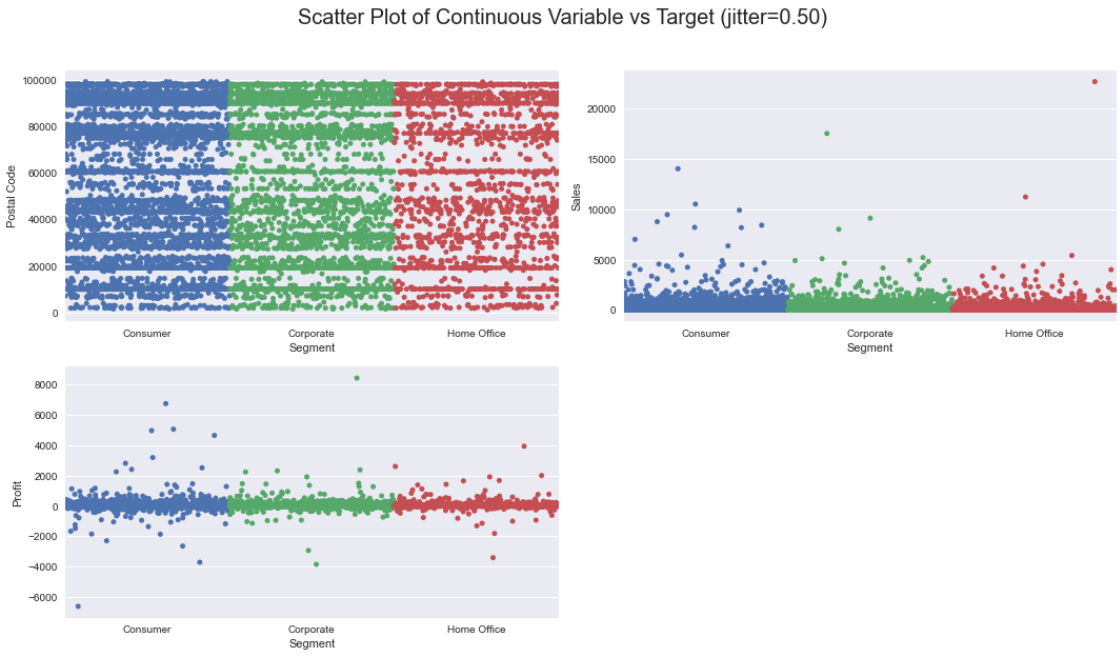
```
    from autoviz.AutoViz_Class import AutoViz_Class
    AV = AutoViz_Class()
    AV.AutoViz(filename, sep=',', depVar='', dfte=None, header=0, verbose=0,
lowess=False,chart_format='svg',max_rows_analyzed=150000,max_cols_analyzed=30)

To remove previous versions, perform 'pip uninstall autoviz'
```

0.1 1.Use the provided data to generate actionable insight using an interactive business intelligence tool

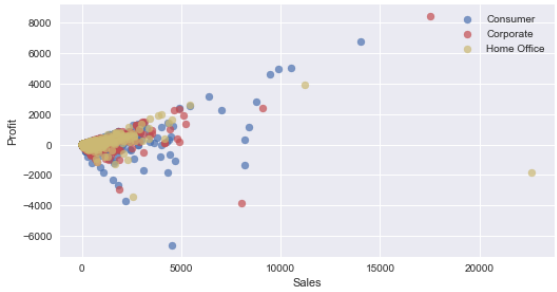
0.1.1 EXPLANATORY DATA ANALYSIS

[6]:
```
df1 = AV.AutoViz("C:/Users/OSAGIE/Desktop/Copy of Sample - Superstore.
↪xls",depVar='Segment',)
```

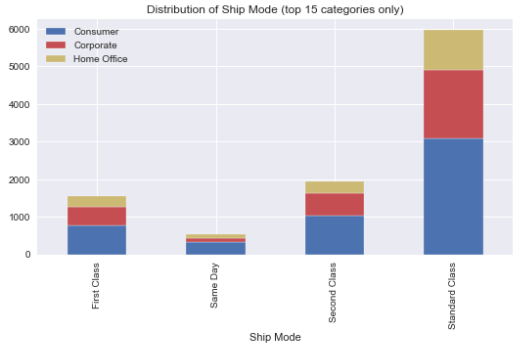
```
Shape of your Data Set: (9994, 21)
Classifying variables in data set…
    20 Predictors classified…
        This does not include the Target column(s)
    8 variables removed since they were ID or low-information variables
Total Number of Scatter Plots = 6
Could not draw Time Series plots
Could not draw Pivot Charts against Dependent Variable
All plots done
Time to run AutoViz (in seconds) = 20.537
```

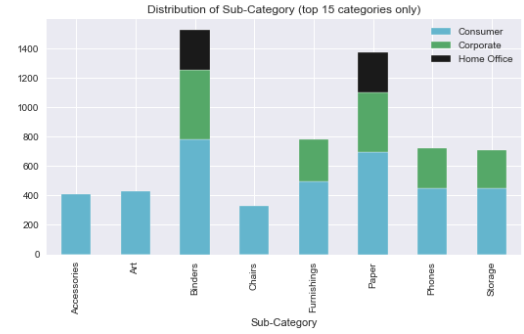


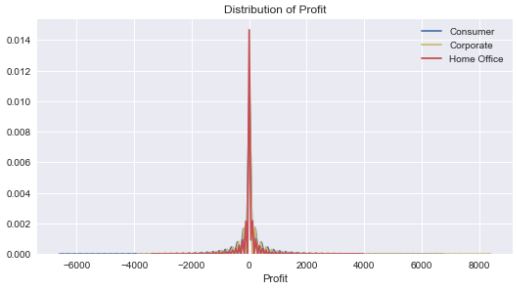
Scatter Plot of each Continuous Variable against Target Variable

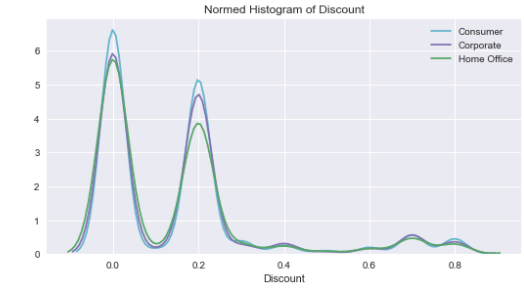


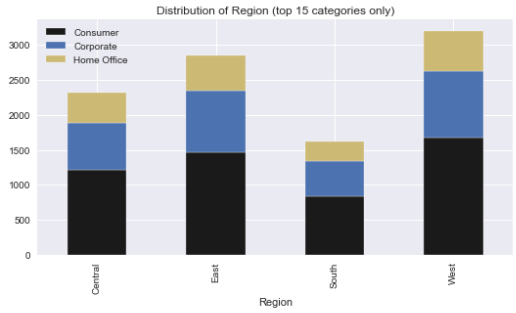
Histograms (KDE plots) of all Continuous Variables

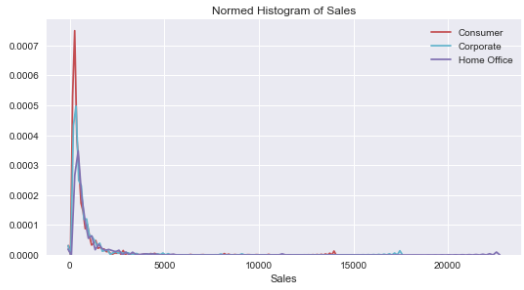


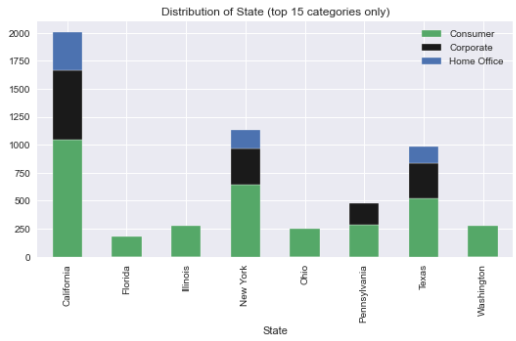


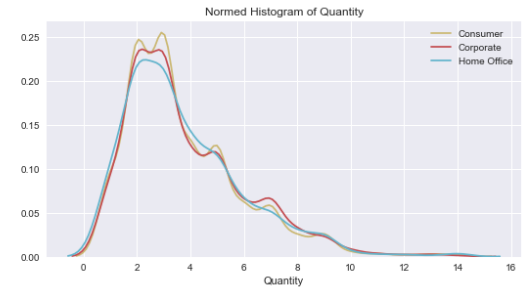


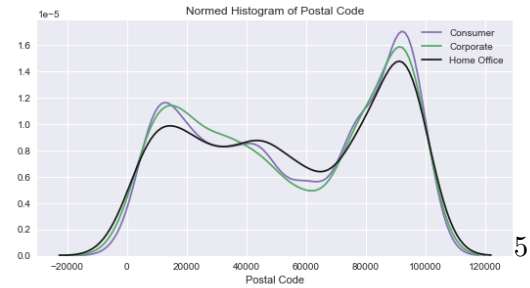


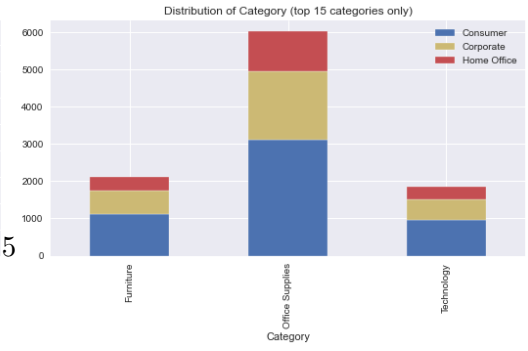






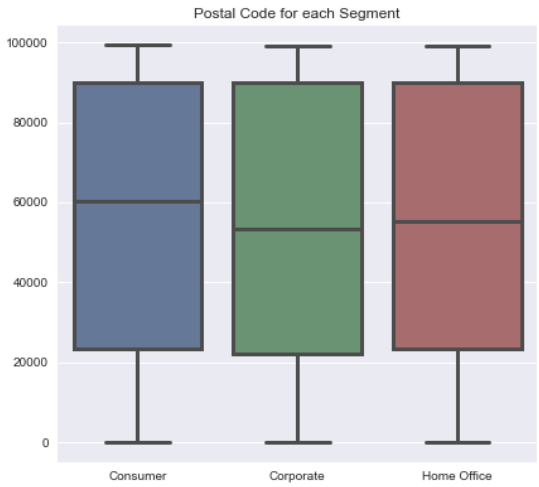


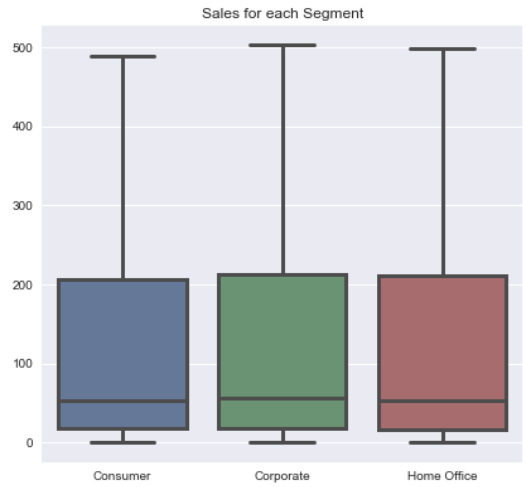


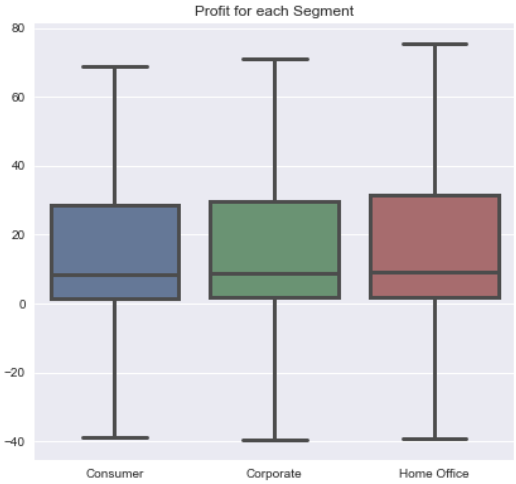


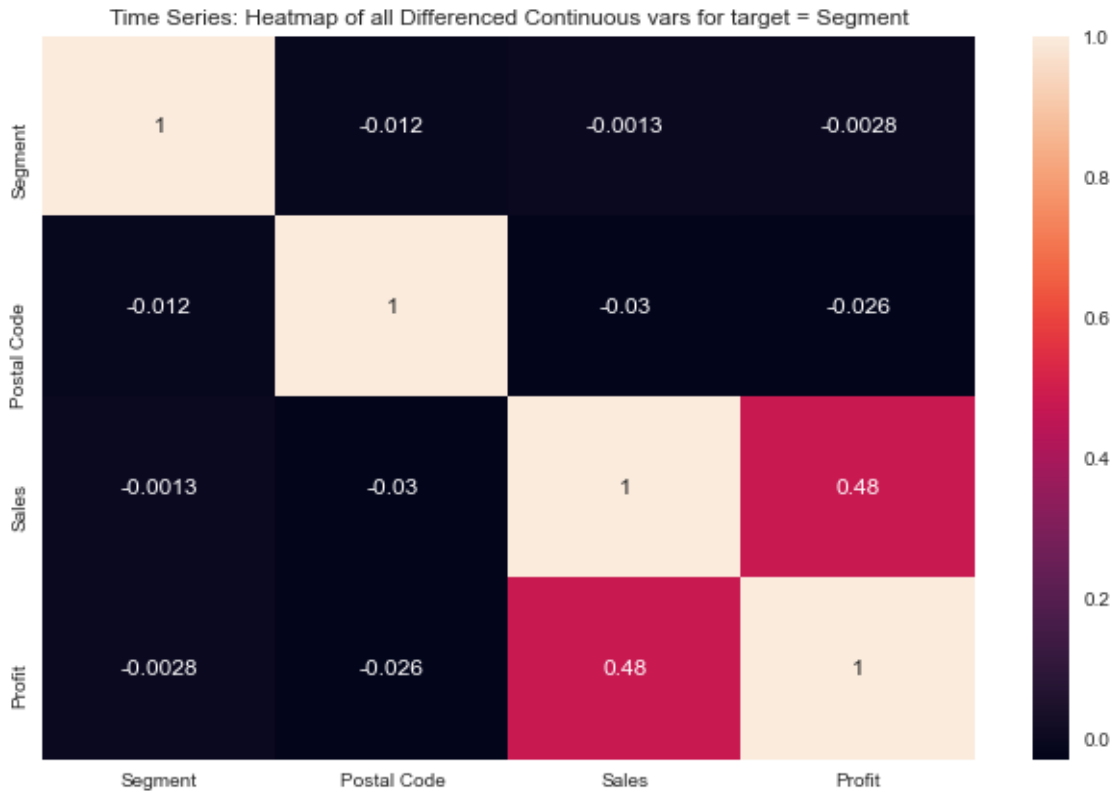

Segment : Distribution of Target Variable
Percentage Distribution of Target = Segment
0.52
0.3
0.18
0.0
0.1
0.2
0.3
0.4
0.5
Consumer
Corporate
Home Office
Freq Distribution of Target Variable = Segment
5191
3020
1783
0
1000
2000
3000
4000
5000
Consumer
Corporate
Home Office

Box Plots without Outliers shown

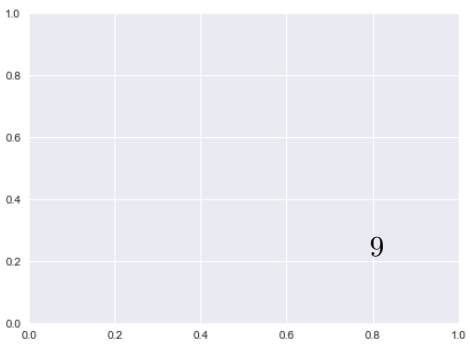




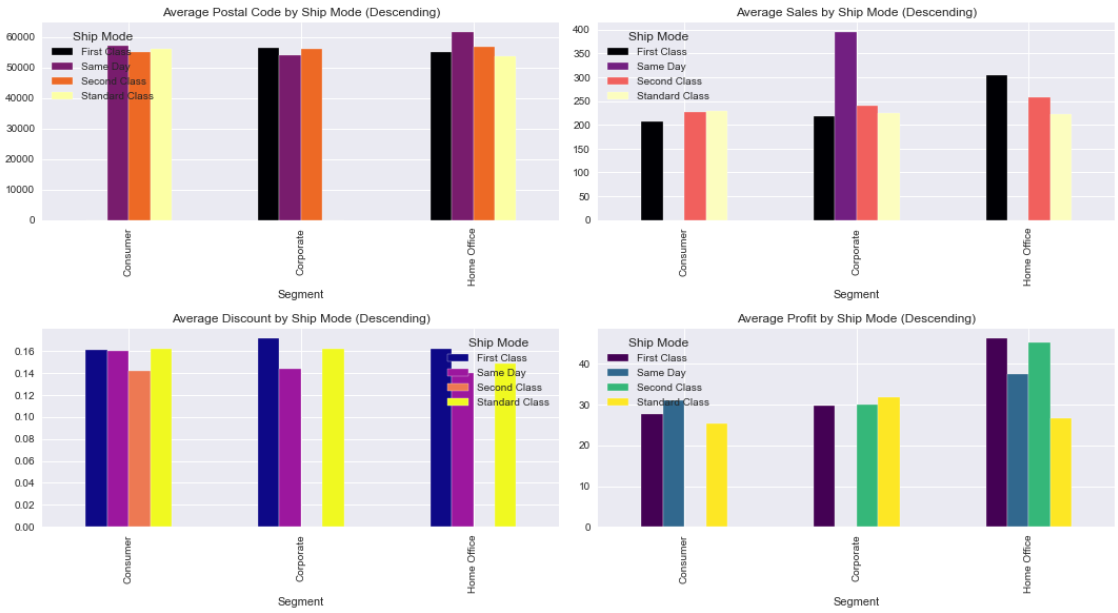



Time Series: Heatmap of all Differenced Continuous vars for target = Segment
Segment
Postal Code
Sales
Profit
1
-0.012
-0.0013
-0.0028
-0.012
1
-0.03
-0.026
-0.0013
-0.03
1
0.48
-0.0028
-0.026
0.48
1
Segment
Postal Code
Sales
Profit
1.0
0.8
0.6
0.4
0.2
0.0

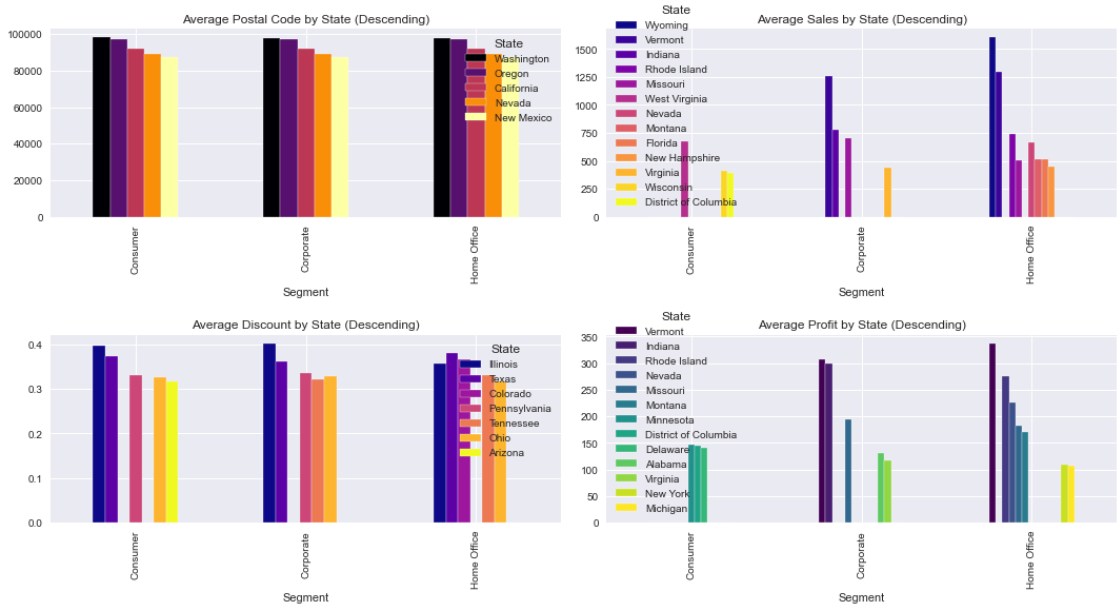
Plots of Number of each Continuous Var by Segment



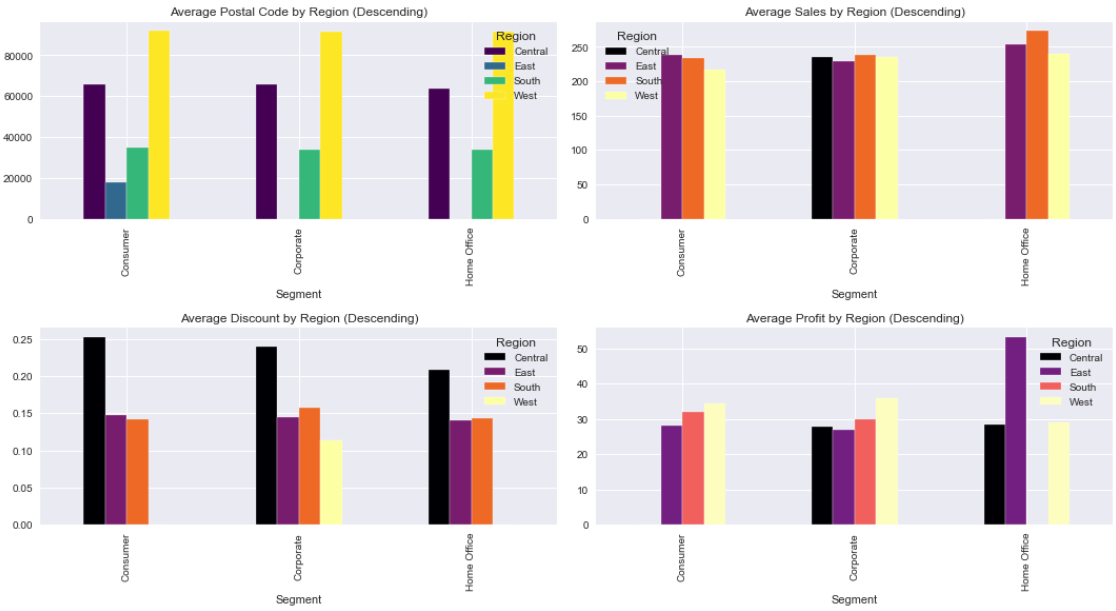
Bar Plots of Continuous Variables by Ship Mode



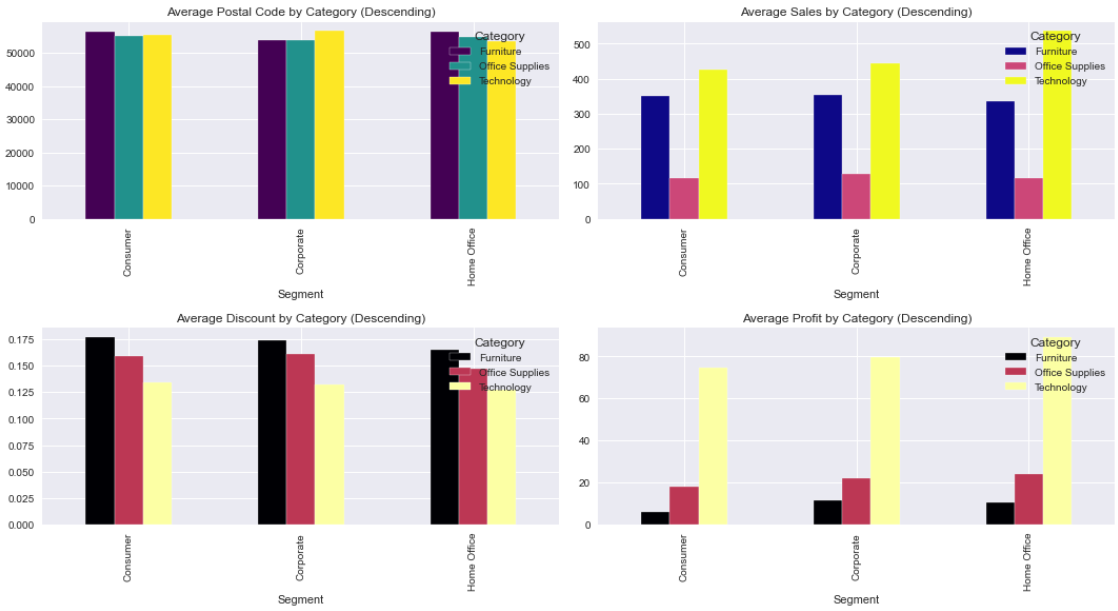
Bar Plots of Continuous Variables by State



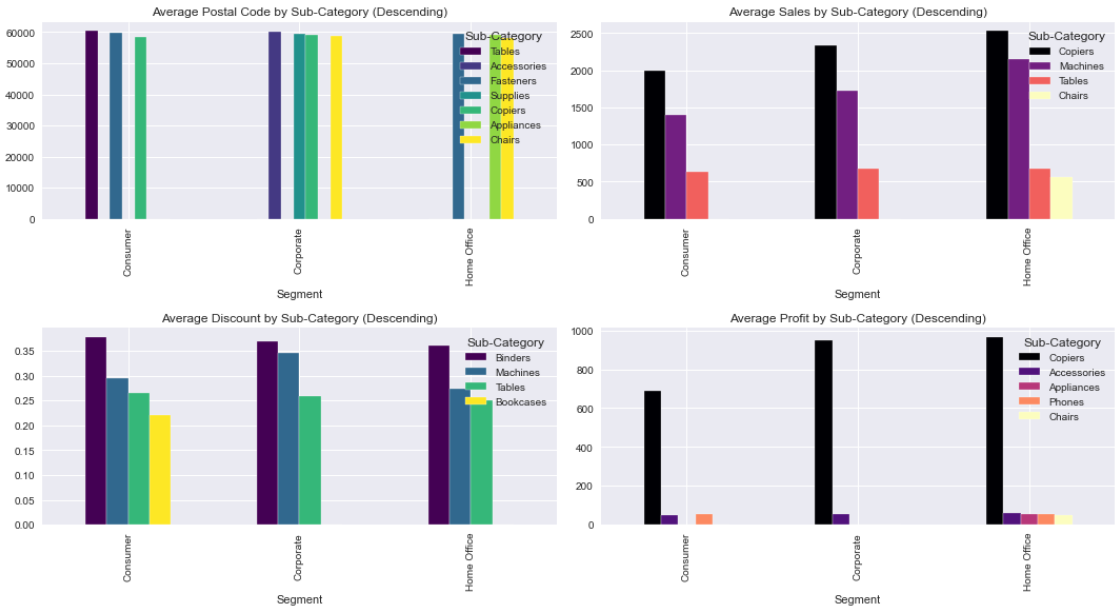
Bar Plots of Continuous Variables by Region

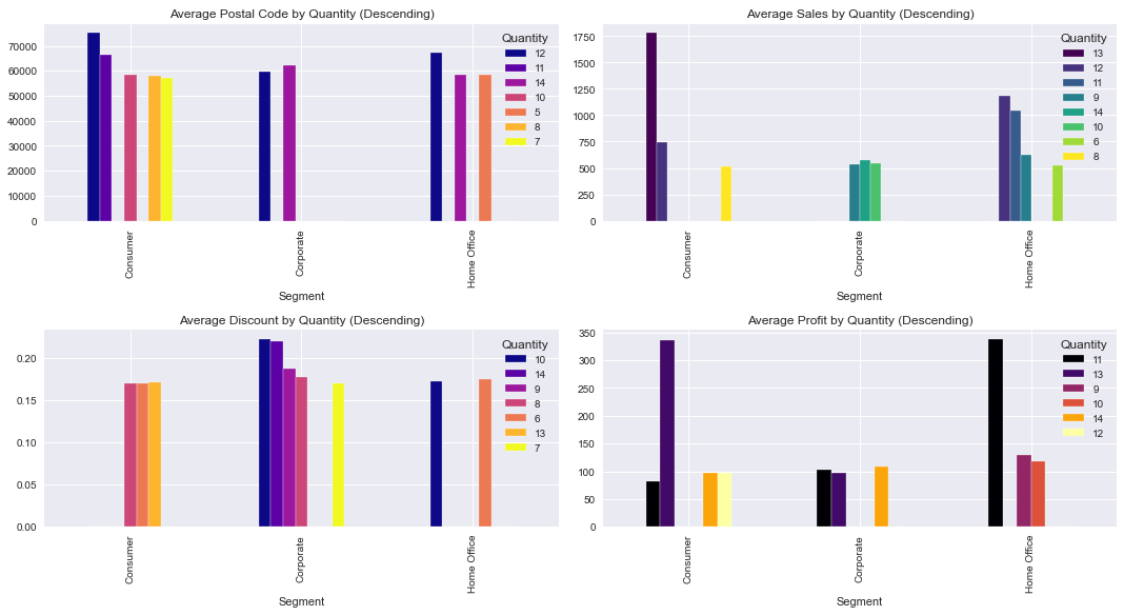


Bar Plots of Continuous Variables by Category



Bar Plots of Continuous Variables by Sub-Category

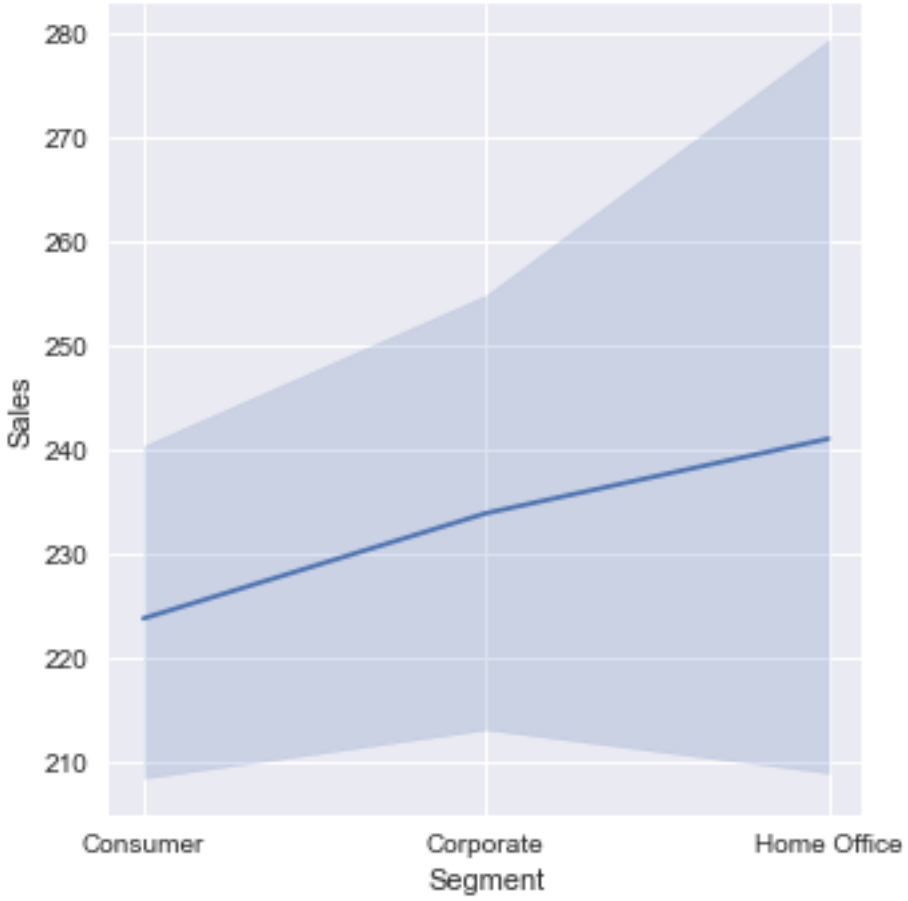




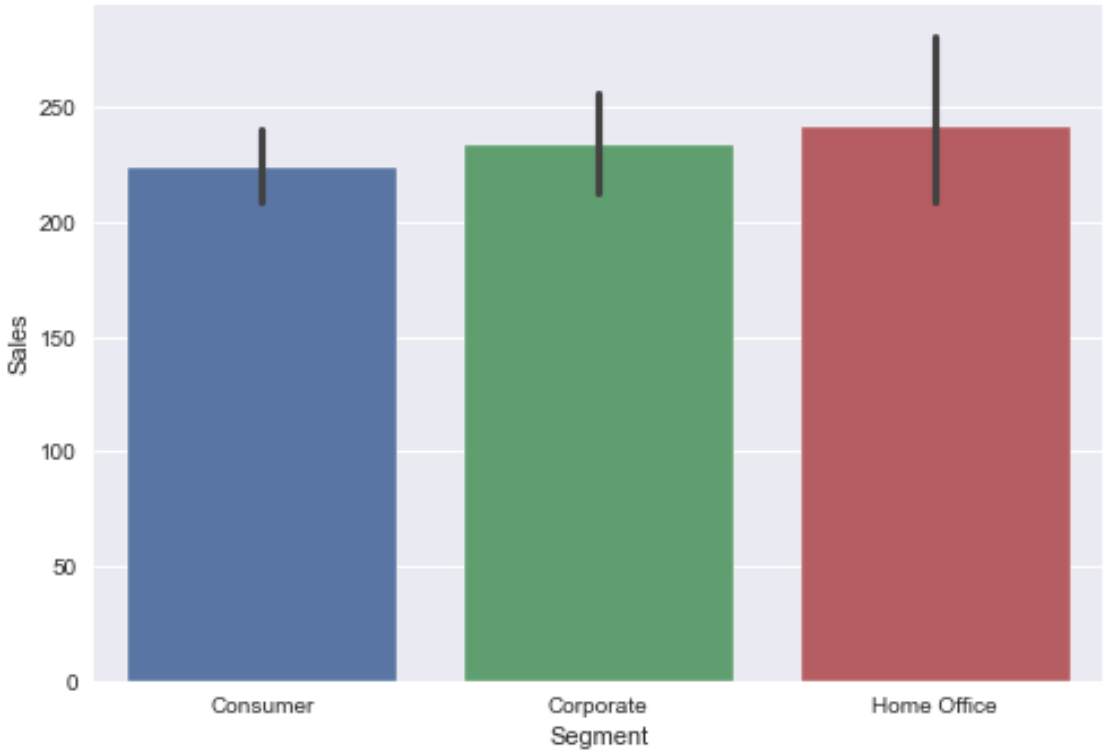
[7]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

0.2 2. Is there a scientific/statistical relationship between customer segment and the revenue from that segment (use the same data provided) and which segment is the most significant?

[8]:
```
#plotting sales and segment as line
sns.relplot(x="Segment", y="Sales",kind="line", data=df);
```



[9]:
```
#plotting sales and segment as boxplot
sns.barplot(x="Segment", y="Sales", data=df);
```



0.3 3. Is there a scientific/statistical relationship between Product and Turn Around Time for Shipping from that segment (use the same data provided)?

[10]:
```
#adding the column for turn time
df['Turnaround_time']=df['Ship Date']-df['Order Date']
```

[11]:
```
df['Turnaround_time']=df['Turnaround_time']/np.timedelta64(1,'D')
```

[12]:
```
#display the first five rows
df.head()
```

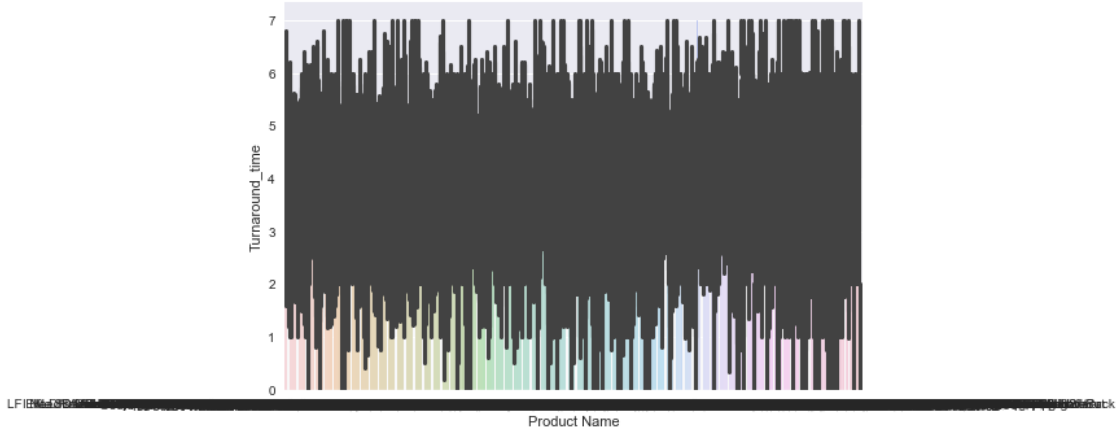
[12]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	\
0	1	CA-2018-152156	2018-11-08	2018-11-11	Second Class	CG-12520	
1	2	CA-2018-152156	2018-11-08	2018-11-11	Second Class	CG-12520	
2	3	CA-2018-138688	2018-06-12	2018-06-16	Second Class	DV-13045	
3	4	US-2017-108966	2017-10-11	2017-10-18	Standard Class	SO-20335	
4	5	US-2017-108966	2017-10-11	2017-10-18	Standard Class	SO-20335	

	Customer Name	Segment	Country/Region	City	…	Region	\
0	Claire Gute	Consumer	United States	Henderson	…	South	
1	Claire Gute	Consumer	United States	Henderson	…	South	
2	Darrin Van Huff	Corporate	United States	Los Angeles	…	West	

```
3   Sean O'Donnell   Consumer  United States  Fort Lauderdale  …  South
4   Sean O'Donnell   Consumer  United States  Fort Lauderdale  …  South

        Product ID         Category Sub-Category  \
0  FUR-BO-10001798        Furniture    Bookcases
1  FUR-CH-10000454        Furniture       Chairs
2  OFF-LA-10000240  Office Supplies       Labels
3  FUR-TA-10000577        Furniture       Tables
4  OFF-ST-10000760  Office Supplies      Storage

                                          Product Name     Sales Quantity  \
0                Bush Somerset Collection Bookcase  261.9600        2
1  Hon Deluxe Fabric Upholstered Stacking Chairs,…  731.9400        3
2  Self-Adhesive Address Labels for Typewriters b…   14.6200        2
3      Bretford CR4500 Series Slim Rectangular Table  957.5775        5
4                    Eldon Fold 'N Roll Cart System   22.3680        2

   Discount    Profit  Turnaround_time
0      0.00   41.9136              3.0
1      0.00  219.5820              3.0
2      0.00    6.8714              4.0
3      0.45 -383.0310              7.0
4      0.20    2.5164              7.0

[5 rows x 22 columns]
```

[13]:
```
#Plotting the graph of product and turnaround time
sns.barplot(x="Product Name", y="Turnaround_time", data=df);
```



The turnAround time for the product can not display properly because of too many product names

0.4 4. Is there a scientific/statistical relationship between Discount and Sales from that segment (use the same data provided)?

[14]:
```
#Creating a column call Discount amount
df['Discount_Amount']=df['Discount']*df['Sales']
```

[15]:
```
#display the first five rows
df.head()
```

[15]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	\
0	1	CA-2018-152156	2018-11-08	2018-11-11	Second Class	CG-12520	
1	2	CA-2018-152156	2018-11-08	2018-11-11	Second Class	CG-12520	
2	3	CA-2018-138688	2018-06-12	2018-06-16	Second Class	DV-13045	
3	4	US-2017-108966	2017-10-11	2017-10-18	Standard Class	SO-20335	
4	5	US-2017-108966	2017-10-11	2017-10-18	Standard Class	SO-20335	

	Customer Name	Segment	Country/Region	City	…	\
0	Claire Gute	Consumer	United States	Henderson	…	
1	Claire Gute	Consumer	United States	Henderson	…	
2	Darrin Van Huff	Corporate	United States	Los Angeles	…	
3	Sean O'Donnell	Consumer	United States	Fort Lauderdale	…	
4	Sean O'Donnell	Consumer	United States	Fort Lauderdale	…	

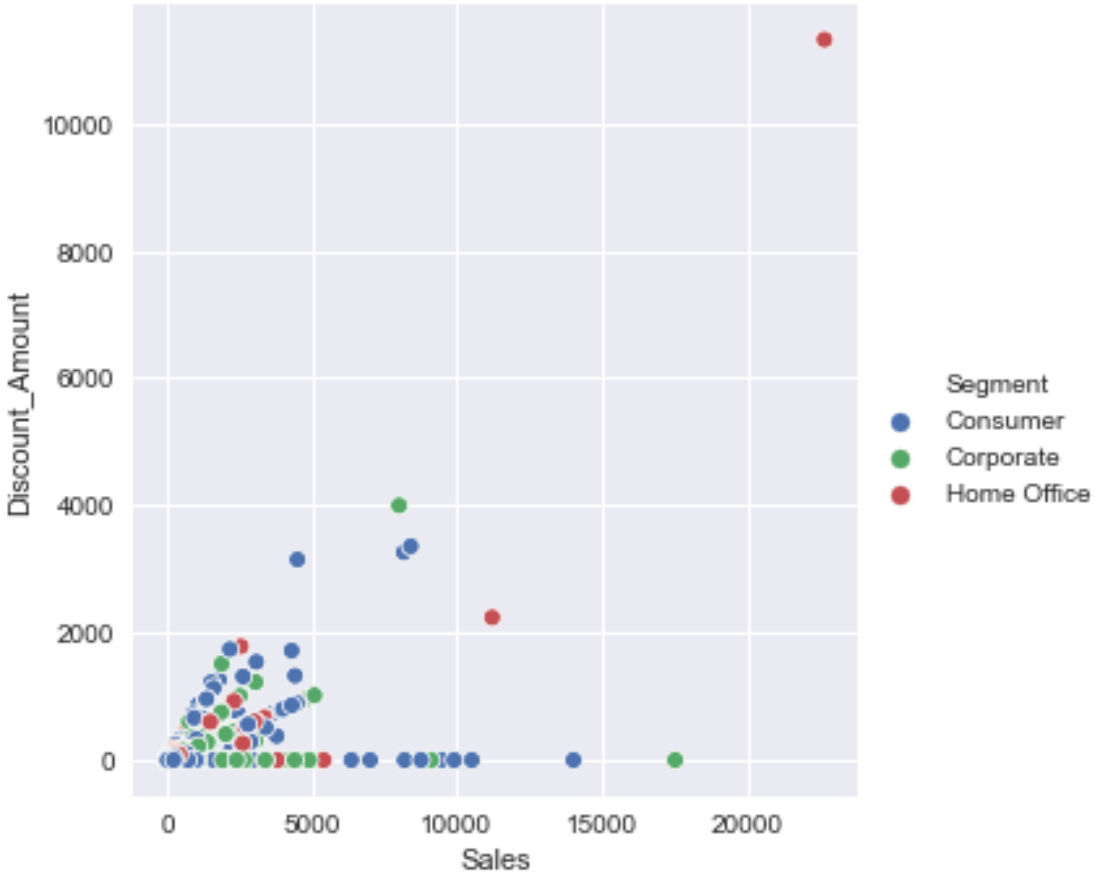
	Product ID	Category	Sub-Category	\
0	FUR-BO-10001798	Furniture	Bookcases	
1	FUR-CH-10000454	Furniture	Chairs	
2	OFF-LA-10000240	Office Supplies	Labels	
3	FUR-TA-10000577	Furniture	Tables	
4	OFF-ST-10000760	Office Supplies	Storage	

	Product Name	Sales	Quantity	\
0	Bush Somerset Collection Bookcase	261.9600	2	
1	Hon Deluxe Fabric Upholstered Stacking Chairs,…	731.9400	3	
2	Self-Adhesive Address Labels for Typewriters b…	14.6200	2	
3	Bretford CR4500 Series Slim Rectangular Table	957.5775	5	
4	Eldon Fold 'N Roll Cart System	22.3680	2	

	Discount	Profit	Turnaround_time	Discount_Amount
0	0.00	41.9136	3.0	0.000000
1	0.00	219.5820	3.0	0.000000
2	0.00	6.8714	4.0	0.000000
3	0.45	-383.0310	7.0	430.909875
4	0.20	2.5164	7.0	4.473600

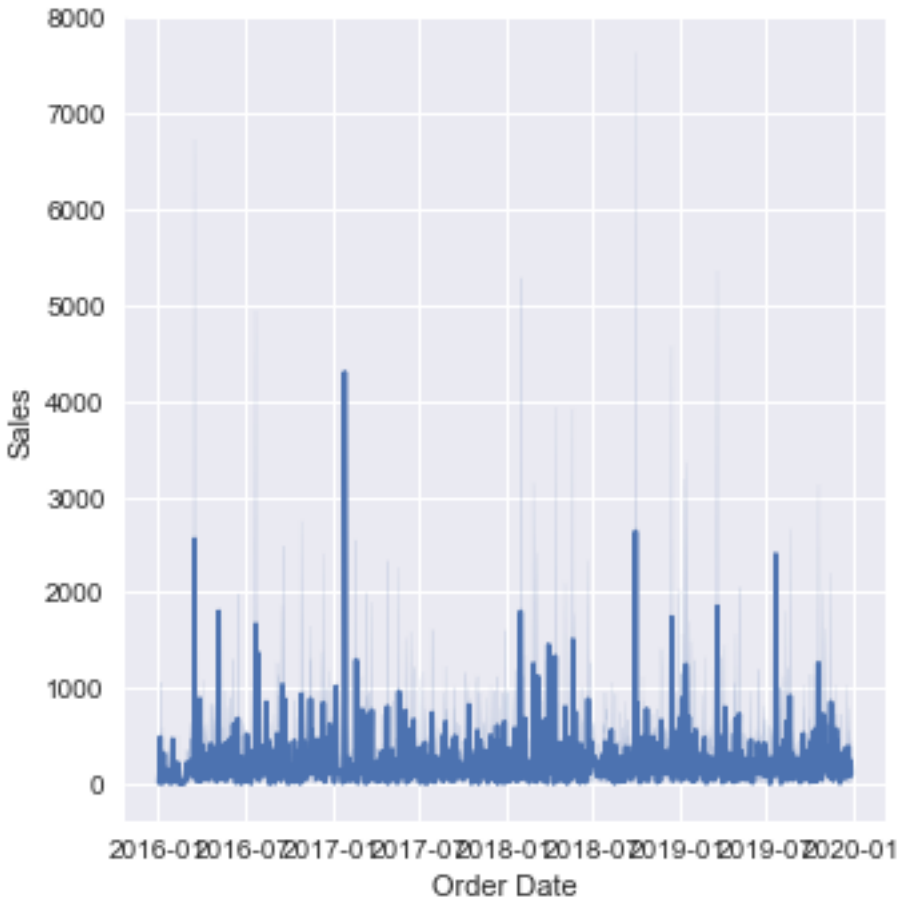
[5 rows x 23 columns]

[16]:
```
# Plotting of the graph for sales and discount
sns.relplot(x="Sales", y="Discount_Amount",hue='Segment', data=df);
```



0.5 5.Based on the same data provided, kindly forecast revenue for each of the product categories for the year 2020

[20]:
```
#plotting the graph of sales and order date as line
sns.relplot(x="Order Date", y="Sales",kind='line', data=df);
```



[21]:
```
#creating a table
df1=df[["Order Date","Sales"]]
```

[22]:
```
#displaying the table
df1.head()
```

[22]:

	Order Date	Sales
0	2018-11-08	261.9600
1	2018-11-08	731.9400
2	2018-06-12	14.6200
3	2017-10-11	957.5775
4	2017-10-11	22.3680

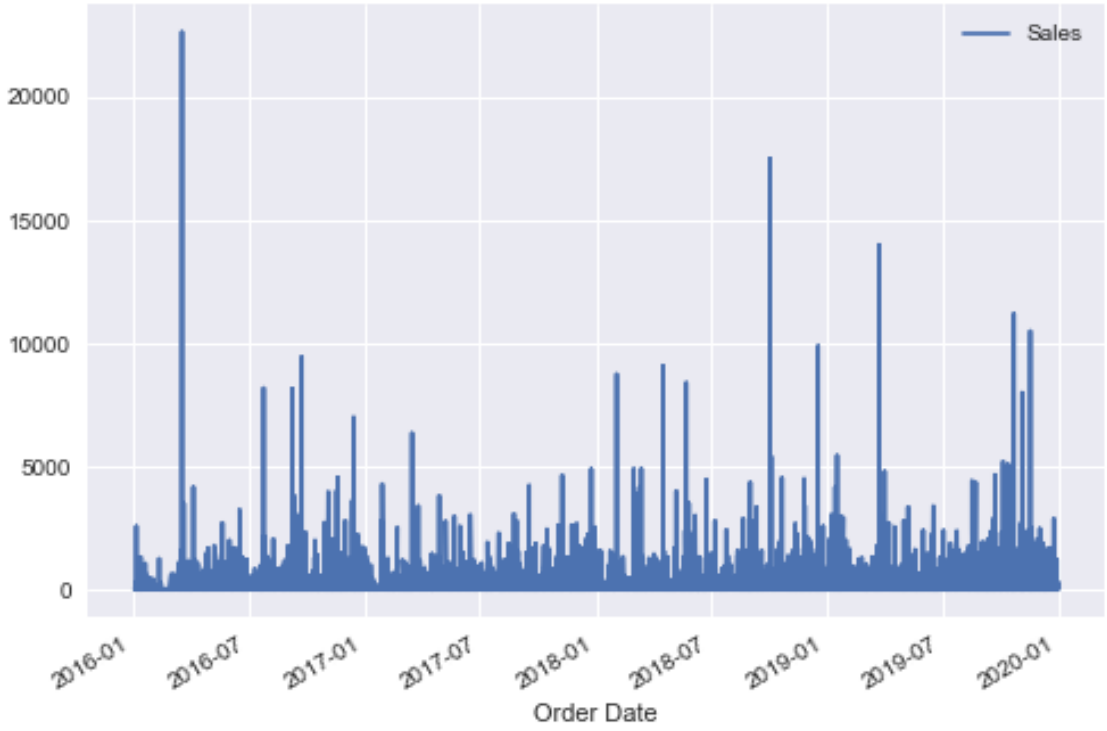
[23]:
```
#setting the ordel date as index
df1 = df1.set_index('Order Date')
```

[24]:
```
#displaying the table
df1.head()
```

[24]:
```
                Sales
Order Date
2018-11-08  261.9600
2018-11-08  731.9400
2018-06-12   14.6200
2017-10-11  957.5775
2017-10-11   22.3680
```

[25]:
```
#plotting the graph of sales and order date
df1.plot()
```

[25]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2a5af1daf10>
```



[28]:
```
#modeling the data for forecasting.
from fbprophet import Prophet
df = df.rename(columns={'Order Date': 'ds', 'Sales': 'y'})
df_model = Prophet(interval_width=0.95)
df_model.fit(df)
```

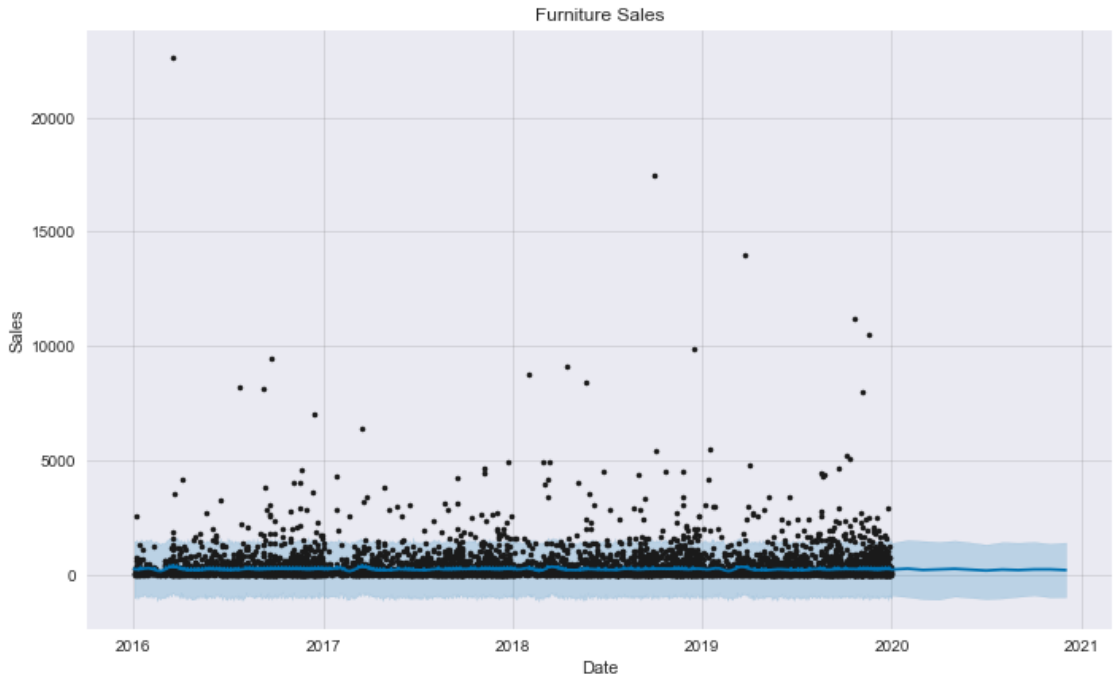
```
INFO:fbprophet:Disabling daily seasonality. Run prophet with
daily_seasonality=True to override this.
```

[28]:
```
<fbprophet.forecaster.Prophet at 0x2a5a641daf0>
```

[33]:
```
#predicting the the value for 2022
df_forecast = df_model.make_future_dataframe(periods=12, freq='MS')
df_forecast = df_model.predict(df_forecast)
```

[34]:
```
#Displaying the graph for 2020 forcast
plt.figure(figsize=(18, 6))
df_model.plot(df_forecast, xlabel = 'Date', ylabel = 'Sales')
plt.title('Furniture Sales');
```

```
<Figure size 1296x432 with 0 Axes>
```



[ ]: