E-COMMERCE

DATA ANALYSIS AND VISUALIZATION USING PYTHON

import pandas as pd  
import plotly.express as px  
import plotly.graph\_objects as go  
import plotly.io as pio  
import plotly.colors as colors  
  
pio.templates.default ='plotly\_white'

data=pd.read\_csv('Sample - Superstore.csv',encoding='latin-1') # encoding='latin-1' it helps the sytem to understand all the special characters like(#,-\_\* etc)  
data

Row ID Order ID Order Date Ship Date Ship Mode \  
0 1 CA-2016-152156 11/8/2016 11/11/2016 Second Class   
1 2 CA-2016-152156 11/8/2016 11/11/2016 Second Class   
2 3 CA-2016-138688 6/12/2016 6/16/2016 Second Class   
3 4 US-2015-108966 10/11/2015 10/18/2015 Standard Class   
4 5 US-2015-108966 10/11/2015 10/18/2015 Standard Class   
... ... ... ... ... ...   
9989 9990 CA-2014-110422 1/21/2014 1/23/2014 Second Class   
9990 9991 CA-2017-121258 2/26/2017 3/3/2017 Standard Class   
9991 9992 CA-2017-121258 2/26/2017 3/3/2017 Standard Class   
9992 9993 CA-2017-121258 2/26/2017 3/3/2017 Standard Class   
9993 9994 CA-2017-119914 5/4/2017 5/9/2017 Second Class   
  
 Customer ID Customer Name Segment Country City \  
0 CG-12520 Claire Gute Consumer United States Henderson   
1 CG-12520 Claire Gute Consumer United States Henderson   
2 DV-13045 Darrin Van Huff Corporate United States Los Angeles   
3 SO-20335 Sean O'Donnell Consumer United States Fort Lauderdale   
4 SO-20335 Sean O'Donnell Consumer United States Fort Lauderdale   
... ... ... ... ... ...   
9989 TB-21400 Tom Boeckenhauer Consumer United States Miami   
9990 DB-13060 Dave Brooks Consumer United States Costa Mesa   
9991 DB-13060 Dave Brooks Consumer United States Costa Mesa   
9992 DB-13060 Dave Brooks Consumer United States Costa Mesa   
9993 CC-12220 Chris Cortes Consumer United States Westminster   
  
 ... Postal Code Region Product ID Category Sub-Category \  
0 ... 42420 South FUR-BO-10001798 Furniture Bookcases   
1 ... 42420 South FUR-CH-10000454 Furniture Chairs   
2 ... 90036 West OFF-LA-10000240 Office Supplies Labels   
3 ... 33311 South FUR-TA-10000577 Furniture Tables   
4 ... 33311 South OFF-ST-10000760 Office Supplies Storage   
... ... ... ... ... ... ...   
9989 ... 33180 South FUR-FU-10001889 Furniture Furnishings   
9990 ... 92627 West FUR-FU-10000747 Furniture Furnishings   
9991 ... 92627 West TEC-PH-10003645 Technology Phones   
9992 ... 92627 West OFF-PA-10004041 Office Supplies Paper   
9993 ... 92683 West OFF-AP-10002684 Office Supplies Appliances   
  
 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   
... ... ... ...   
9989 Ultra Door Pull Handle 25.2480 3   
9990 Tenex B1-RE Series Chair Mats for Low Pile Car... 91.9600 2   
9991 Aastra 57i VoIP phone 258.5760 2   
9992 It's Hot Message Books with Stickers, 2 3/4" x 5" 29.6000 4   
9993 Acco 7-Outlet Masterpiece Power Center, Wihtou... 243.1600 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   
... ... ...   
9989 0.20 4.1028   
9990 0.00 15.6332   
9991 0.20 19.3932   
9992 0.00 13.3200   
9993 0.00 72.9480   
  
[9994 rows x 21 columns]

df=pd.DataFrame(data)  
print(df.columns)

Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',  
 'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State',  
 'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category',  
 'Product Name', 'Sales', 'Quantity', 'Discount', 'Profit'],  
 dtype='object')

df.describe()

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

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 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

# Converting Order Date and Ship Date Into Date Formate

df['Order Date']=pd.to\_datetime(df['Order Date'])  
df['Ship Date']=pd.to\_datetime(df['Ship Date'])

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 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

df

Row ID Order ID Order Date Ship Date Ship Mode \  
0 1 CA-2016-152156 2016-11-08 2016-11-11 Second Class   
1 2 CA-2016-152156 2016-11-08 2016-11-11 Second Class   
2 3 CA-2016-138688 2016-06-12 2016-06-16 Second Class   
3 4 US-2015-108966 2015-10-11 2015-10-18 Standard Class   
4 5 US-2015-108966 2015-10-11 2015-10-18 Standard Class   
... ... ... ... ... ...   
9989 9990 CA-2014-110422 2014-01-21 2014-01-23 Second Class   
9990 9991 CA-2017-121258 2017-02-26 2017-03-03 Standard Class   
9991 9992 CA-2017-121258 2017-02-26 2017-03-03 Standard Class   
9992 9993 CA-2017-121258 2017-02-26 2017-03-03 Standard Class   
9993 9994 CA-2017-119914 2017-05-04 2017-05-09 Second Class   
  
 Customer ID Customer Name Segment Country City \  
0 CG-12520 Claire Gute Consumer United States Henderson   
1 CG-12520 Claire Gute Consumer United States Henderson   
2 DV-13045 Darrin Van Huff Corporate United States Los Angeles   
3 SO-20335 Sean O'Donnell Consumer United States Fort Lauderdale   
4 SO-20335 Sean O'Donnell Consumer United States Fort Lauderdale   
... ... ... ... ... ...   
9989 TB-21400 Tom Boeckenhauer Consumer United States Miami   
9990 DB-13060 Dave Brooks Consumer United States Costa Mesa   
9991 DB-13060 Dave Brooks Consumer United States Costa Mesa   
9992 DB-13060 Dave Brooks Consumer United States Costa Mesa   
9993 CC-12220 Chris Cortes Consumer United States Westminster   
  
 ... Postal Code Region Product ID Category Sub-Category \  
0 ... 42420 South FUR-BO-10001798 Furniture Bookcases   
1 ... 42420 South FUR-CH-10000454 Furniture Chairs   
2 ... 90036 West OFF-LA-10000240 Office Supplies Labels   
3 ... 33311 South FUR-TA-10000577 Furniture Tables   
4 ... 33311 South OFF-ST-10000760 Office Supplies Storage   
... ... ... ... ... ... ...   
9989 ... 33180 South FUR-FU-10001889 Furniture Furnishings   
9990 ... 92627 West FUR-FU-10000747 Furniture Furnishings   
9991 ... 92627 West TEC-PH-10003645 Technology Phones   
9992 ... 92627 West OFF-PA-10004041 Office Supplies Paper   
9993 ... 92683 West OFF-AP-10002684 Office Supplies Appliances   
  
 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   
... ... ... ...   
9989 Ultra Door Pull Handle 25.2480 3   
9990 Tenex B1-RE Series Chair Mats for Low Pile Car... 91.9600 2   
9991 Aastra 57i VoIP phone 258.5760 2   
9992 It's Hot Message Books with Stickers, 2 3/4" x 5" 29.6000 4   
9993 Acco 7-Outlet Masterpiece Power Center, Wihtou... 243.1600 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   
... ... ...   
9989 0.20 4.1028   
9990 0.00 15.6332   
9991 0.20 19.3932   
9992 0.00 13.3200   
9993 0.00 72.9480   
  
[9994 rows x 21 columns]

df['Order Month']=df['Order Date'].dt.month  
df['Order Year']=df['Order Date'].dt.year  
df['Order Day of the week']=df['Order Date'].dt.dayofweek

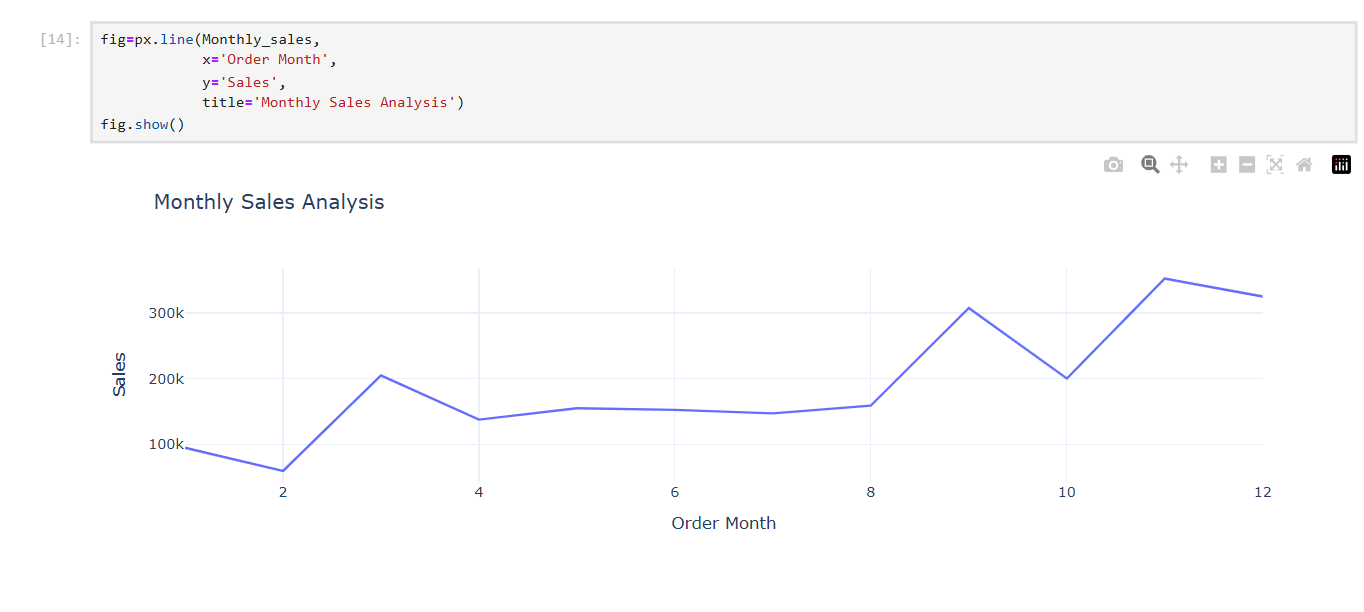
df.head(6)

Row ID Order ID Order Date Ship Date Ship Mode Customer ID \  
0 1 CA-2016-152156 2016-11-08 2016-11-11 Second Class CG-12520   
1 2 CA-2016-152156 2016-11-08 2016-11-11 Second Class CG-12520   
2 3 CA-2016-138688 2016-06-12 2016-06-16 Second Class DV-13045   
3 4 US-2015-108966 2015-10-11 2015-10-18 Standard Class SO-20335   
4 5 US-2015-108966 2015-10-11 2015-10-18 Standard Class SO-20335   
5 6 CA-2014-115812 2014-06-09 2014-06-14 Standard Class BH-11710   
  
 Customer Name Segment Country 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 ...   
5 Brosina Hoffman Consumer United States Los Angeles ...   
  
 Category Sub-Category \  
0 Furniture Bookcases   
1 Furniture Chairs   
2 Office Supplies Labels   
3 Furniture Tables   
4 Office Supplies Storage   
5 Furniture Furnishings   
  
 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   
5 Eldon Expressions Wood and Plastic Desk Access... 48.8600 7   
  
 Discount Profit Order Month Order Year Order Day of the week   
0 0.00 41.9136 11 2016 1   
1 0.00 219.5820 11 2016 1   
2 0.00 6.8714 6 2016 6   
3 0.45 -383.0310 10 2015 6   
4 0.20 2.5164 10 2015 6   
5 0.00 14.1694 6 2014 0   
  
[6 rows x 24 columns]

# Monthly Sales

Monthly\_sales=df.groupby('Order Month')['Sales'].sum().reset\_index()  
Monthly\_sales

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

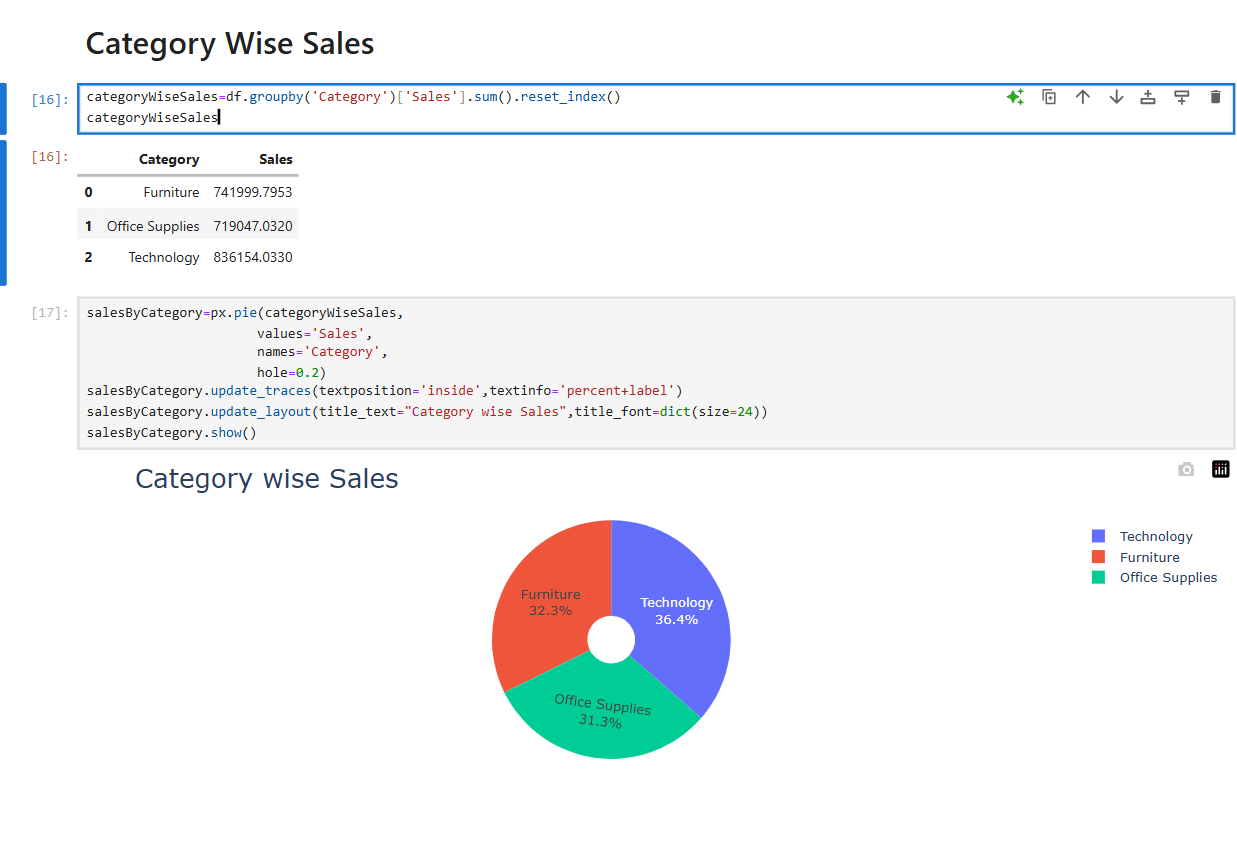
fig=px.line(Monthly\_sales,  
 x='Order Month',  
 y='Sales',  
 title='Monthly Sales Analysis')  
fig.show()

# Category Wise Sales

categoryWiseSales=df.groupby('Category')['Sales'].sum().reset\_index()  
categoryWiseSales

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

salesByCategory=px.pie(categoryWiseSales,  
 values='Sales',  
 names='Category',  
 hole=0.2)  
salesByCategory.update\_traces(textposition='inside',textinfo='percent+label')  
salesByCategory.update\_layout(title\_text="Category wise Sales",title\_font=dict(size=24))  
salesByCategory.show()



# Sub-Category wise Sales

sub\_categorySales=df.groupby('Sub-Category')['Sales'].sum().reset\_index()  
sub\_categorySales

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

subcategorySalesfig=px.bar(sub\_categorySales,  
 x='Sub-Category',  
 y='Sales',  
 title='Sales Analysis By Category',  
 color='Sub-Category',  
 text=sub\_categorySales['Sales'].round(2))  
subcategorySalesfig.show()



# Monthly Wise Profits

monthlyProfits=df.groupby('Order Month')['Profit'].sum().reset\_index()  
monthlyProfits

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

monthlyProfitfig=px.line(monthlyProfits,  
 x='Order Month',  
 y='Profit',  
 title="Monthly Profits",  
 markers='o')  
monthlyProfitfig.show()



# Category Wise Profits

categoryProfits=df.groupby('Category')['Profit'].sum().reset\_index()  
categoryProfits

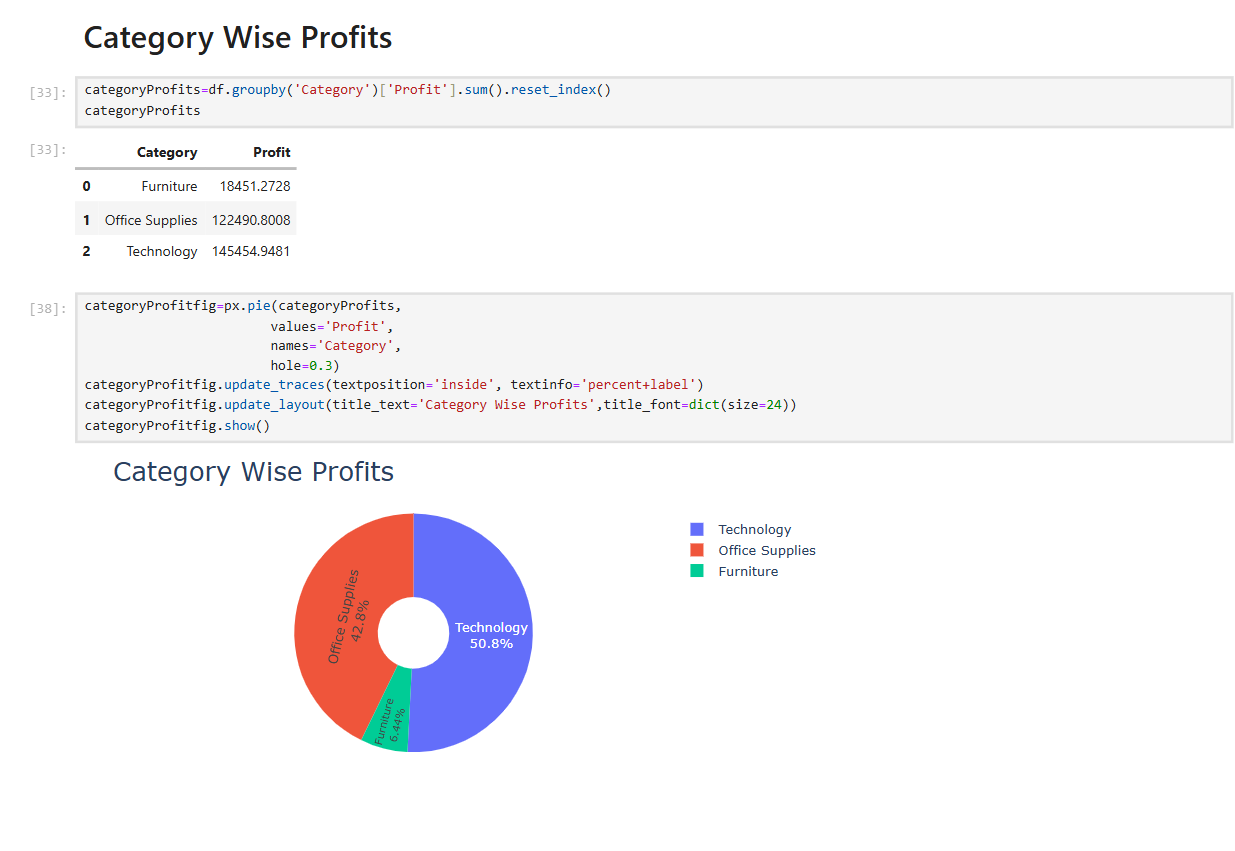
Category Profit  
0 Furniture 18451.2728  
1 Office Supplies 122490.8008  
2 Technology 145454.9481

categoryProfitfig=px.pie(categoryProfits,  
 values='Profit',  
 names='Category',  
 hole=0.3)  
categoryProfitfig.update\_traces(textposition='inside', textinfo='percent+label')  
categoryProfitfig.update\_layout(title\_text='Category Wise Profits',title\_font=dict(size=24))  
categoryProfitfig.show()

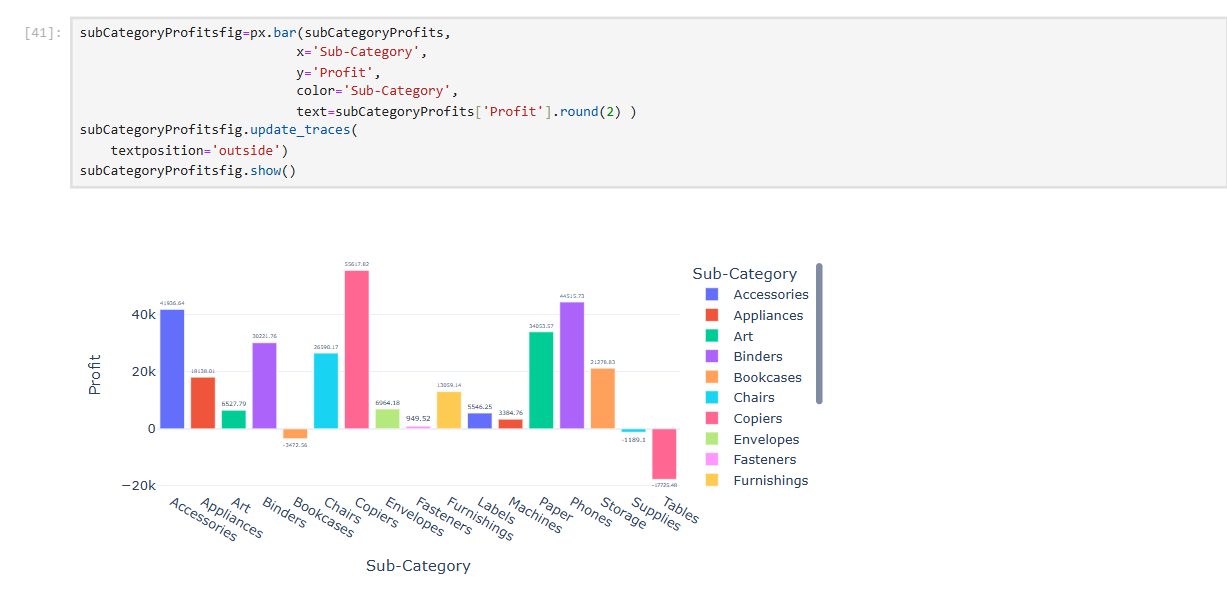
# 

# Sub-category Wise Profits

subCategoryProfits=df.groupby('Sub-Category')['Profit'].sum().reset\_index()  
subCategoryProfits

 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

subCategoryProfitsfig=px.bar(subCategoryProfits,  
 x='Sub-Category',  
 y='Profit',  
 color='Sub-Category',  
 text=subCategoryProfits['Profit'].round(2) )  
subCategoryProfitsfig.update\_traces(   
 textposition='outside')  
subCategoryProfitsfig.show()



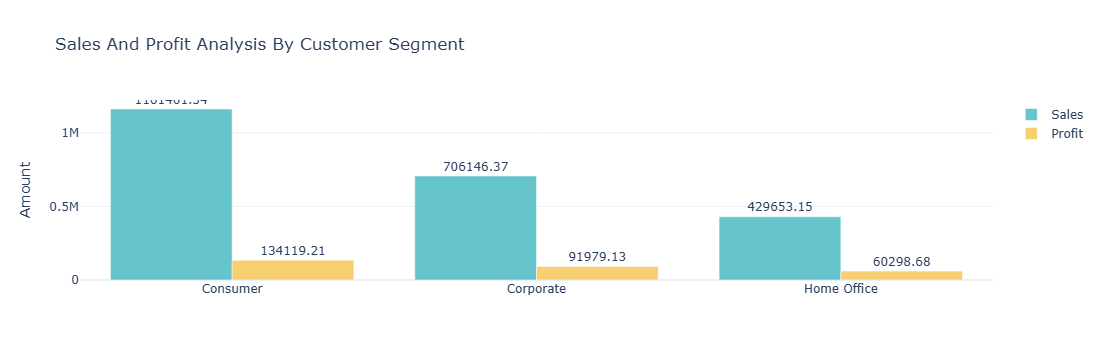
# 

# Sales and Profits Based On Segment

segmentPS=df.groupby('Segment').agg({'Sales':'sum','Profit':'sum'}).reset\_index()  
segmentPS

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

segmentfig=go.Figure()  
color\_palette= colors.qualitative.Pastel  
segmentfig.add\_trace(go.Bar(x=segmentPS['Segment'],  
 y=segmentPS['Sales'],  
 name='Sales',  
 marker\_color=color\_palette[0],  
 text=segmentPS['Sales'].round(2),  
 textposition='outside'))  
segmentfig.add\_trace(go.Bar(x=segmentPS['Segment'],  
 y=segmentPS['Profit'],  
 name='Profit',  
 marker\_color=color\_palette[1],  
 text=segmentPS['Profit'].round(2),  
 textposition='outside'))  
segmentfig.update\_layout(title='Sales And Profit Analysis By Customer Segment',yaxis\_title='Amount')  
segmentfig.show()



# Sales And Profit Ratio

ratio=df.groupby('Segment').agg({'Sales':'sum','Profit':'sum'}).reset\_index()  
ratio['sales/profits']=ratio['Sales']/ratio['Profit']  
print(ratio[['Segment','sales/profits']])

Segment sales/profits  
0 Consumer 8.659471  
1 Corporate 7.677245  
2 Home Office 7.125416