US SAMPLE STORE RETAIL ANALYSIS

Step1 : *Importing* Libaries

In [1]:
```
#Library
import numpy as np
import pandas as pd
import seaborn as sns
import datetime
import time
import matplotlib.pyplot as plt
import plotly.graph_objs as go
import plotly.offline as py
from plotly.offline import download_plotlyjs, init_notebook_mode, iplot

# py.init_notebook_mode(connected = True)
pd.set_option('display.float_format', lambda x: f'{x:.1f}')
df = pd.read_csv("us_superstore_sales.csv",encoding='latin-1')
```

First 5 Rows of the data

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

Out[2]:

	Row_ID	Order_ID	Order_Date	Ship_Date	Ship_Mode	Customer_ID	Customer_Name	Segment	Country	City	...	Postal_Code	Region	Product_ID	Category	Sub_Catego
0	1	CA-2016-152156	08/11/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-BO-10001798	Furniture	Bookcas
1	2	CA-2016-152156	08/11/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-CH-10000454	Furniture	Cha
2	3	CA-2016-138688	12/06/2016	16/06/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	West	OFF-LA-10000240	Office Supplies	Labe
3	4	US-2015-108966	11/10/2015	18/10/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	FUR-TA-10000577	Furniture	Tabl
4	5	US-2015-108966	11/10/2015	18/10/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	OFF-ST-10000760	Office Supplies	Stora

5 rows × 21 columns

Data Cleaning

In [4]:
```
#Categorical Data
df['Region'] = df['Region'].astype('category')
df['Category'] = df['Category'].astype('category')
df['Sub_Category'] = df['Sub_Category'].astype('category')
df['Product_Name'] = df['Product_Name'].astype('category')
df['Segment'] = df['Segment'].astype('category')
df['Ship_Mode'] = df['Ship_Mode'].astype('category')
df['Country'] = df['Country'].astype('category')
df['State'] = df['State'].astype('category')
df['City'] = df['City'].astype('category')

#Integer Data
df['Quantity'] = df['Quantity'].astype('int64')

df['Discount'] = df['Discount'].fillna(0)
df['Discount'] = df['Discount'].astype('float64')

df['Sales'] = df['Sales'].fillna(0)
df['Sales'] = df['Sales'].astype('float64')

df['Profit'] = df['Profit'].fillna(0)
df['Profit'] = df['Profit'].astype('float64')
```

```
#Date Data
df['Order_Date'] = pd.to_datetime(df['Order_Date'],format='%d/%m/%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   object        
 4   Ship_Mode      9994 non-null   category      
 5   Customer_ID    9994 non-null   object        
 6   Customer_Name  9994 non-null   object        
 7   Segment        9994 non-null   category      
 8   Country        9994 non-null   category      
 9   City           9994 non-null   category      
 10  State          9994 non-null   category      
 11  Postal_Code    9994 non-null   int64         
 12  Region         9994 non-null   category      
 13  Product_ID     9994 non-null   object        
 14  Category       9994 non-null   category      
 15  Sub_Category   9994 non-null   category      
 16  Product_Name   9994 non-null   category      
 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: category(9), datetime64[ns](1), float64(3), int64(3), object(5)
memory usage: 1.1+ MB
```

Solving Business Questions using EDA

1. How much did by year the store make and sell between 2011 and 2014?
2. How much did by month the store make and sell between 2011 and 2014?
3. Which was the most profitable category, and which sold the most?
4. Which was the most profitable sub-category, and which sold the most?
5. Which was the most profitable segment, and which sold the most?
6. Which country bought the most, and which made the most profit?
7. Which country bought the less, and which made the less profit? There was negative profit?

In [5]:
```
# Sales and profit by year
df['Order_Date'] = pd.to_datetime(df['Order_Date'])
df['Order_Year'] = df['Order_Date'].dt.year
grup_y = df.groupby(['Order_Year']).sum().reset_index()
grup_y
```

Out[5]:

	Order_Year	Row_ID	Postal_Code	Sales	Quantity	Discount	Profit
0	2014	9904015	113271247	484247.5	7581	315.5	49544.0
1	2015	10413696	111208247	470532.5	7979	327.1	61618.6
2	2016	12778804	141003420	609205.6	9837	400.3	81795.2
3	2017	16848500	186089738	733215.3	12476	518.2	93439.3

In [6]:
```
# Assuming your DataFrame is named 'df' and contains an 'Order_Year' column
pivot_table = pd.pivot_table(df, values='Sales', index='Order_Year', aggfunc='sum')

# Display the pivot table
print(pivot_table)
```

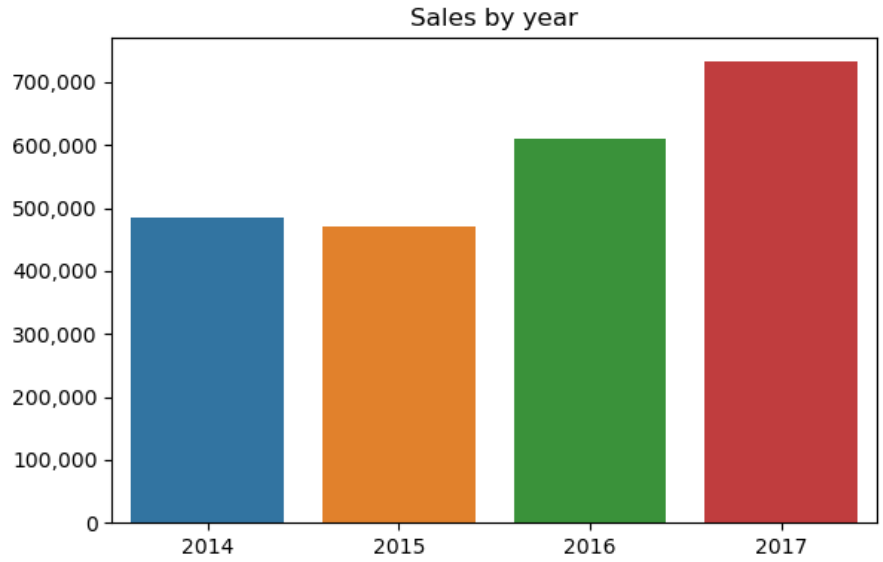
```
              Sales
Order_Year
2014       484247.5
2015       470532.5
2016       609205.6
2017       733215.3
```

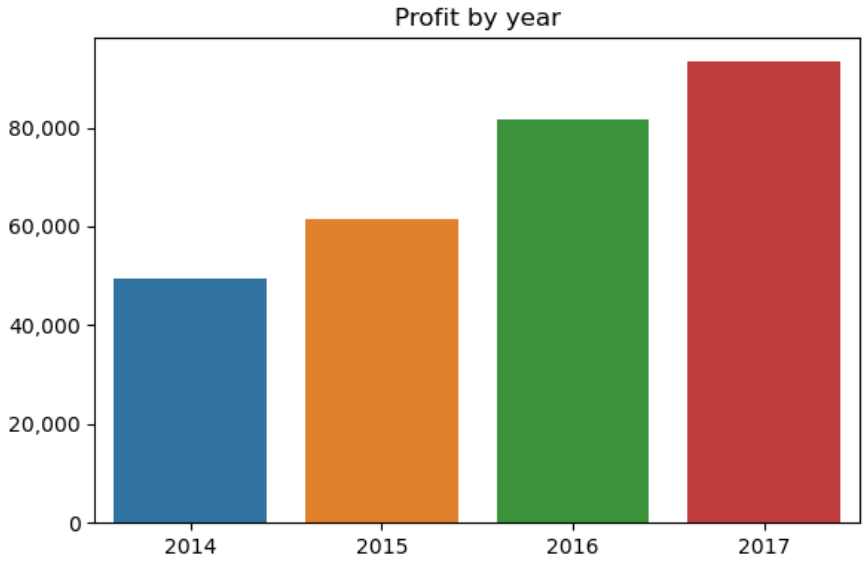
Q1. What is the Sales and Profit growth Year-on-Year?

To answer this question, we can group the data by 'Sales' and calculate the mean price for each Year:

In [7]:
```
# Sales by year
plt.figure(figsize=(12,4), tight_layout=True)
plt.subplot(1,2,1)
g1 = sns.barplot(x= 'Order_Year', y= 'Sales', data= grup_y)
g1.set(xlabel=None, ylabel=None, title='Sales by year')
current_values = plt.gca().get_yticks()
plt.gca().set_yticklabels(['{:,.0f}'.format(x) for x in current_values])
# Profit by year
plt.subplot(1,2,2)
g2 = sns.barplot(x= 'Order_Year', y= 'Profit', data= grup_y)
g2.set(xlabel=None, ylabel=None, title='Profit by year')
current_values = plt.gca().get_yticks()
plt.gca().set_yticklabels(['{:,.0f}'.format(x) for x in current_values])
# plt.savefig('fig1.png')
plt.show()
```

```
C:\Users\Mohan Sharma\AppData\Local\Temp\ipykernel_20640\3970194423.py:7: UserWarning:

FixedFormatter should only be used together with FixedLocator

C:\Users\Mohan Sharma\AppData\Local\Temp\ipykernel_20640\3970194423.py:13: UserWarning:

FixedFormatter should only be used together with FixedLocator
```



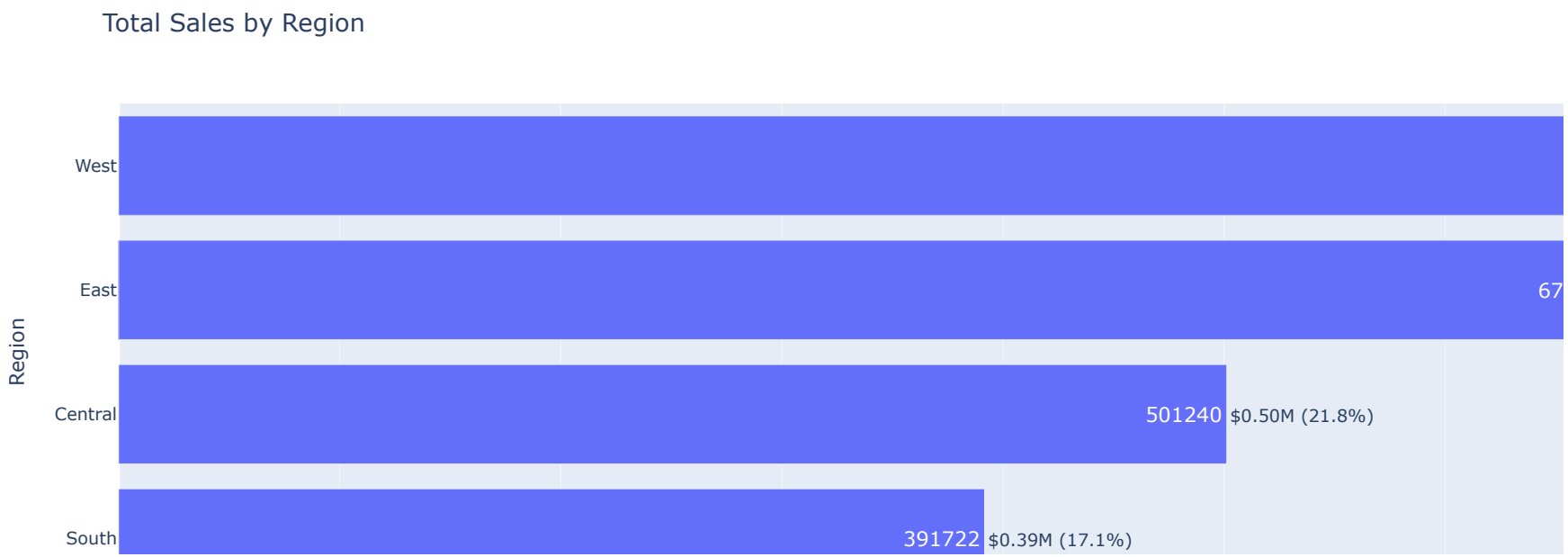


Q1. What region is having the highest sale in US?

To answer this question, we can group the data by 'Region' and calculate the Total Sale for each region:

In [8]:
```
import plotly.graph_objects as go
import plotly.express as px

# Group the data by region and calculate total sales
sales_by_region = df.groupby('Region')['Sales'].sum().reset_index()

# Convert sales to million dollars
sales_by_region['Sales_Million'] = sales_by_region['Sales'] / 1_000_000

# Sort the data by sales in descending order
sales_by_region = sales_by_region.sort_values('Sales', ascending=True)

# Create a horizontal bar chart for sales by region
fig = go.Figure(data=go.Bar(y=sales_by_region['Region'], x=sales_by_region['Sales_Million'], orientation='h'))

# Add labels inside the bars with increased font size
fig.update_traces(text=round(sales_by_region['Sales']), textposition='inside', textfont_size=14)

# Add percentages outside the bars with increased font size
total_sales = sales_by_region['Sales_Million'].sum()
fig.update_layout(annotations=[
    go.layout.Annotation(
        y=r, x=s,
        text=f"${s:.2f}M\n({s/total_sales*100:.1f}%)",
        showarrow=False,
        xanchor='left',
        yanchor='middle',
```

```
        font=dict(size=12)
    ) for r, s in zip(sales_by_region['Region'], sales_by_region['Sales_Million'])
])

# Set the chart title and axes labels
fig.update_layout(
    title='Total Sales by Region',
    xaxis_title='Sales (Million $)',
    yaxis_title='Region'
)

# Show the chart
fig.show()
```



Inference from the Total Sales Bar Plot By Region ;

According to the plot

- Sount region have maximum sales and total contribution of 31.6%.
- West region have minimum sales and total contribution of 17.1 %

Q2. What Category is having the highest sale in US?

To answer this question, we can group the data by 'Category' and calculate the Total Sale for each region:

In [9]:
```
import plotly.graph_objects as go
import plotly.express as px

# Group the data by Category and calculate total sales
sales_by_region = df.groupby('Category')['Sales'].sum().reset_index()

# Convert sales to million dollars
sales_by_region['Sales_Million'] = sales_by_region['Sales'] / 1_000_000

# Sort the data by sales in descending order
sales_by_region = sales_by_region.sort_values('Sales', ascending=True)

# Create a horizontal bar chart for sales by Category
fig = go.Figure(data=go.Bar(y=sales_by_region['Category'], x=sales_by_region['Sales_Million'], orientation='h'))

# Add labels inside the bars with increased font size
fig.update_traces(text=round(sales_by_region['Sales']), textposition='inside', textfont_size=14)

# Add percentages outside the bars with increased font size
total_sales = sales_by_region['Sales_Million'].sum()
fig.update_layout(annotations=[
    go.layout.Annotation(
        y=r, x=s,
        text=f"${s:.2f}M\n({s/total_sales*100:.1f}%)",
        showarrow=False,
        xanchor='left',
        yanchor='middle',
        font=dict(size=12)
    ) for r, s in zip(sales_by_region['Category'], sales_by_region['Sales_Million'])
])

# Set the chart title and axes labels
fig.update_layout(
    title='Total Sales by Category',
    xaxis_title='Sales (Million $)',
    yaxis_title='Category'
)

# Show the chart
fig.show()
```

Inference from the Total Sales Bar Plot By Category ;

According to the plot

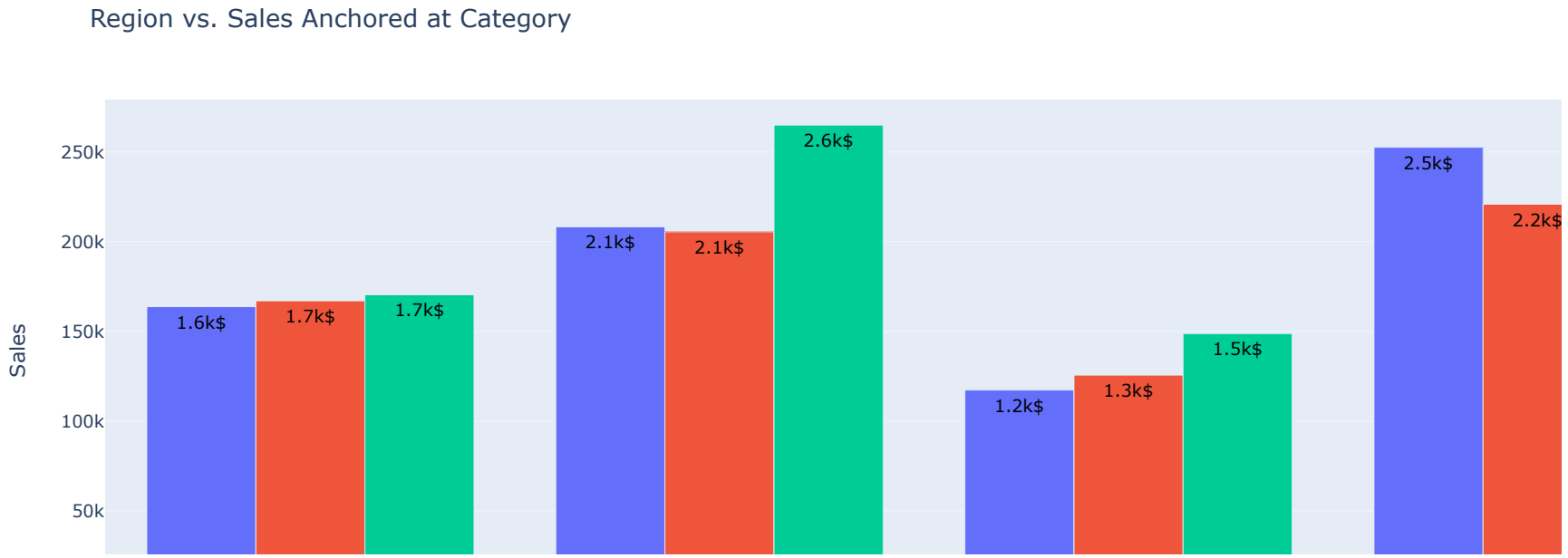
- Technology Product is having the maximum sales and total contribution of 36.4%.
- Office Product is having the minimum sales and total contribution of 31.3 %

Q3. Total Sales by categories across Region?

To answer this question, we can group the data by 'Region' and calculate the Total Sale breakdown by Category for each region:

In [13]:
```
import plotly.graph_objects as go

# Group the data by region and category and calculate total sales
sales_by_region_category = df.groupby(['Region', 'Category'])['Sales'].sum().reset_index()

# Create a bar chart for region vs. sales anchored at category
fig = go.Figure()
```

```
# Add labels as separate bars
for i, category in enumerate(sales_by_region_category['Category'].unique()):
    category_data = sales_by_region_category[sales_by_region_category['Category'] == category]
    x_values = category_data['Region']
    y_values = category_data['Sales']
    labels = [f"{sales/100_000:.1f}k$" for sales in y_values]

    fig.add_trace(go.Bar(
        x=x_values,
        y=y_values,
        text=labels,
        name=category,
        textposition='auto',
        textfont=dict(color='black', size=12),
        showlegend= True,
        opacity=1
    ))

# Set the chart title and axes labels
fig.update_layout(
    title='Region vs. Sales Anchored at Category',
    xaxis_title='Region',
    yaxis_title='Sales'
)

# Show the chart
fig.show()
```

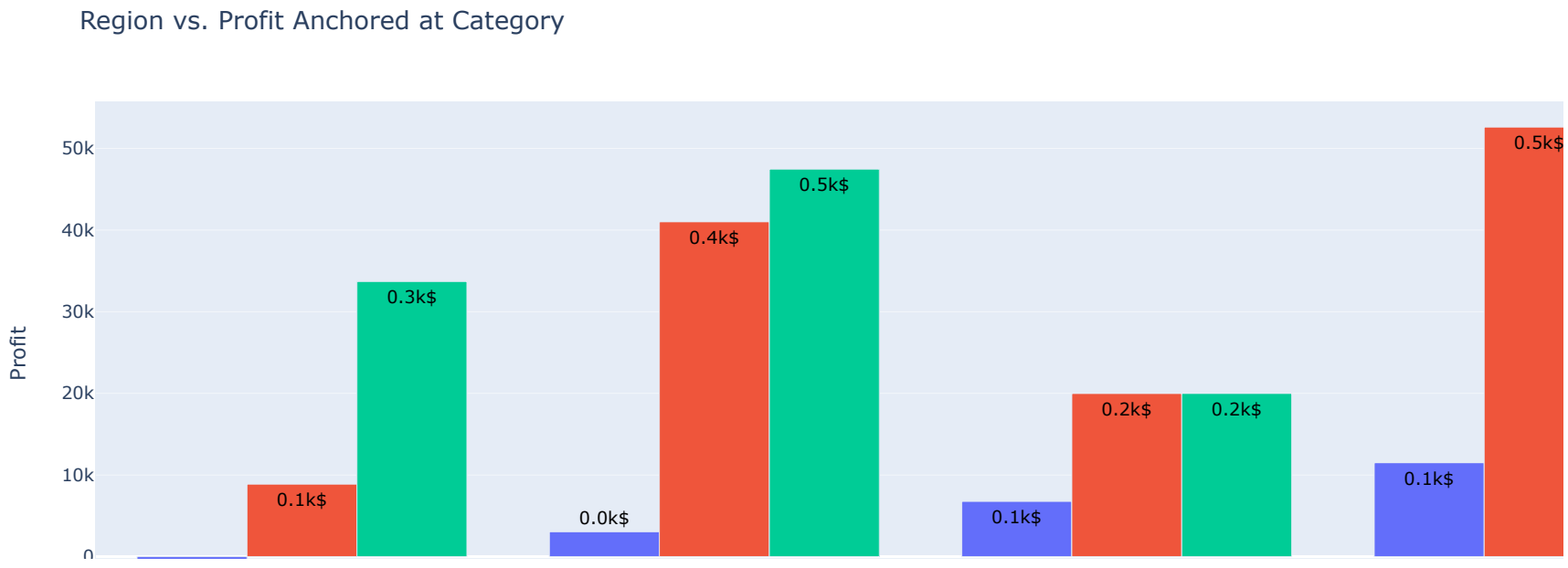


Q4. Total Profit by categories across Region?

To answer this question, we can group the data by 'Region' and calculate the Total profit breakdown by Category for each region:

In [14]:
```
import plotly.graph_objects as go

# Group the data by region and category and calculate total Profit
Profit_by_region_category = df.groupby(['Region', 'Category'])['Profit'].sum().reset_index()

# Create a bar chart for region vs. sales anchored at category
fig = go.Figure()

# Add labels as separate bars
for i, category in enumerate(Profit_by_region_category['Category'].unique()):
    category_data = Profit_by_region_category[Profit_by_region_category['Category'] == category]
    x_values = category_data['Region']
    y_values = category_data['Profit']
    labels = [f"{profit/100_000:.1f}k$" for profit in y_values]

    fig.add_trace(go.Bar(
        x=x_values,
```

```
        y=y_values,
        text=labels,
        name=category,
        textposition='auto',
        textfont=dict(color='black', size=12),
        showlegend= True,
        opacity=1
    ))

# Set the chart title and axes labels
fig.update_layout(
    title='Region vs. Profit Anchored at Category',
    xaxis_title='Region',
    yaxis_title='Profit'
)

# Show the chart
fig.show()
```

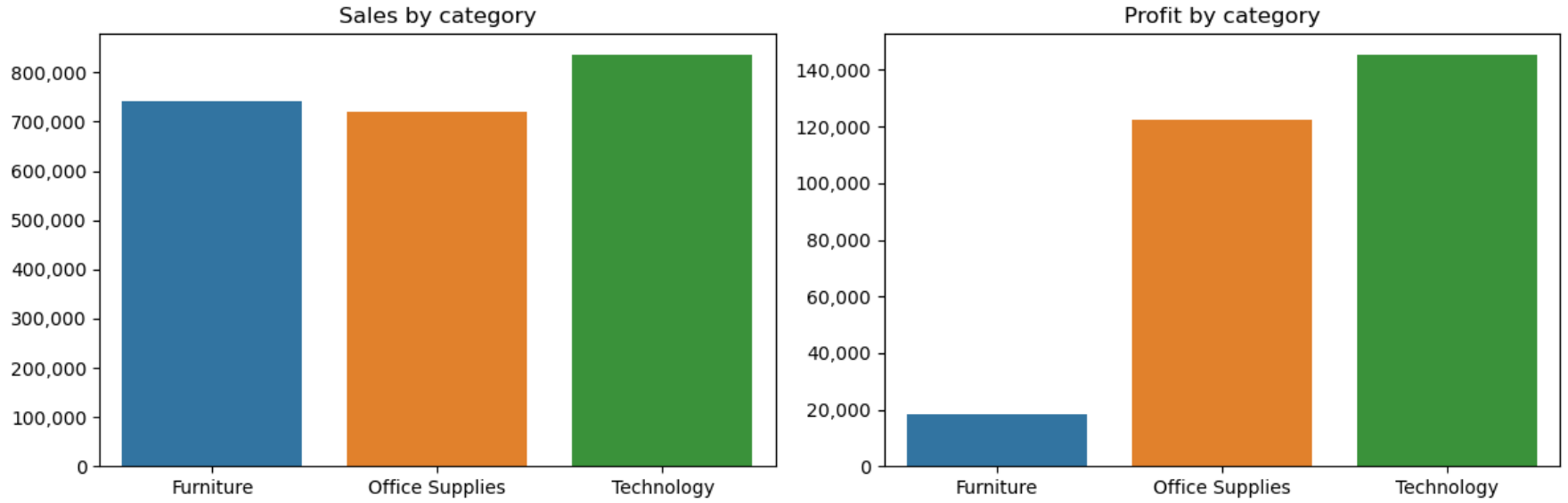


Q5. Which was the most Sales and Profit category, and which sold the most?

To answer this question, we can group the data by 'Region' and calculate the Total profit breakdown by Category for each region:

In [15]:
```
grup_cat = df.groupby(['Category']).sum().reset_index()
grup_cat

plt.figure(figsize=(12,4), tight_layout=True)
plt.subplot(1,2,1)
# Sales by category
g3 = sns.barplot(x='Category', y='Sales', data=grup_cat)
g3.set(xlabel=None, ylabel=None, title='Sales by category')
current_values = plt.gca().get_yticks()
plt.gca().set_yticklabels(['{:,.0f}'.format(x) for x in current_values])
# Profit by category
plt.subplot(1,2,2)
g4 = sns.barplot(x='Category', y='Profit', data=grup_cat)
g4.set(xlabel=None, ylabel=None, title='Profit by category')
current_values = plt.gca().get_yticks()
plt.gca().set_yticklabels(['{:,.0f}'.format(x) for x in current_values])
# plt.savefig('fig4.png')
plt.show()
```

```
C:\Users\Mohan Sharma\AppData\Local\Temp\ipykernel_20640\2779832684.py:10: UserWarning:

FixedFormatter should only be used together with FixedLocator

C:\Users\Mohan Sharma\AppData\Local\Temp\ipykernel_20640\2779832684.py:16: UserWarning:

FixedFormatter should only be used together with FixedLocator
```



Inference from the above bar Plot ;

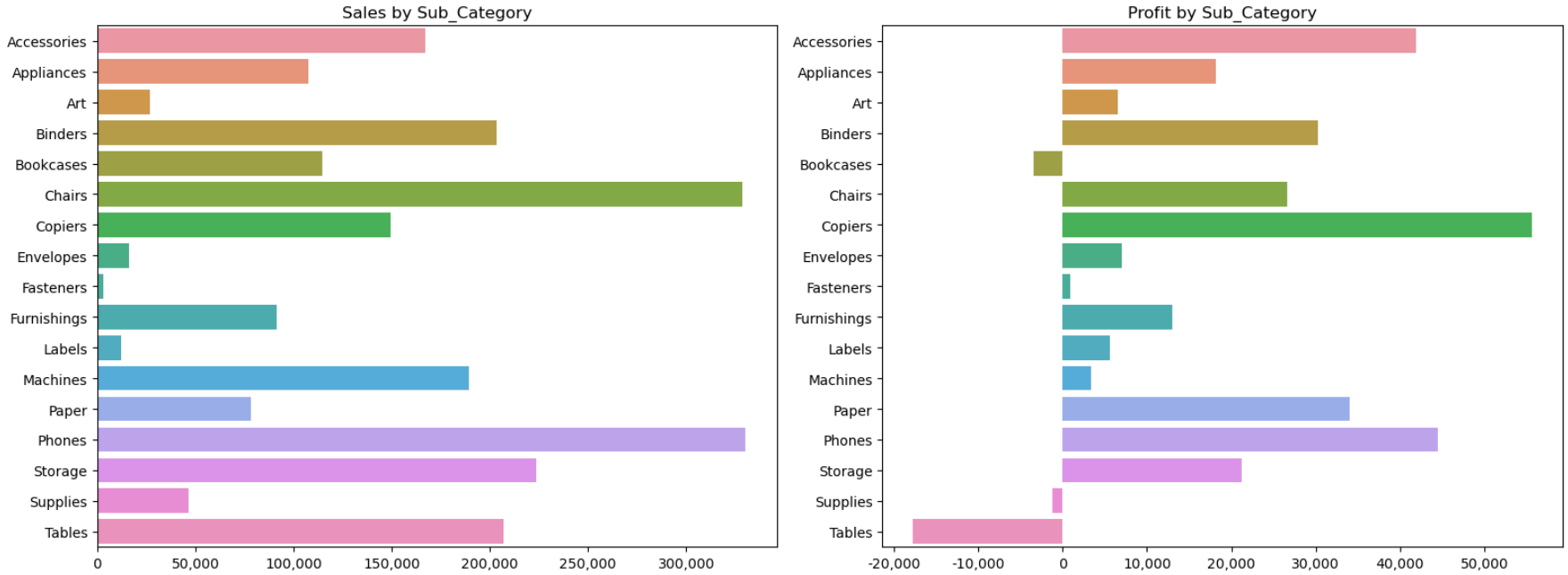
Technology sold the most, and was the most profitable category. But furniture, despite having sold more than Office Supplies, it made less profit than it.

Q6. Which was the most profitable sub-category, and which sold the most?

To answer this question, we can group the data by 'Region' and calculate the Total profit breakdown by Category for each region:

In [16]:
```
grup_subcat = df.groupby(['Sub_Category']).sum().reset_index()

plt.figure(figsize=(16,6), tight_layout=True)
plt.subplot(1,2,1)
# Sales by sub-category
g5 = sns.barplot(y='Sub_Category', x='Sales', data=grup_subcat, orient='h')
g5.set(xlabel=None, ylabel=None, title='Sales by Sub_Category')
current_values = plt.gca().get_xticks()
plt.gca().set_xticklabels(['{:,.0f}'.format(x) for x in current_values])
# Profit by subcategoria
plt.subplot(1,2,2)
g6 = sns.barplot(y='Sub_Category', x='Profit', data=grup_subcat, orient='h')
g6.set(xlabel=None, ylabel=None, title='Profit by Sub_Category')
current_values = plt.gca().get_xticks()
plt.gca().set_xticklabels(['{:,.0f}'.format(x) for x in current_values])
plt.show()
```

```
C:\Users\Mohan Sharma\AppData\Local\Temp\ipykernel_20640\1394607245.py:9: UserWarning:

FixedFormatter should only be used together with FixedLocator

C:\Users\Mohan Sharma\AppData\Local\Temp\ipykernel_20640\1394607245.py:15: UserWarning:

FixedFormatter should only be used together with FixedLocator
```



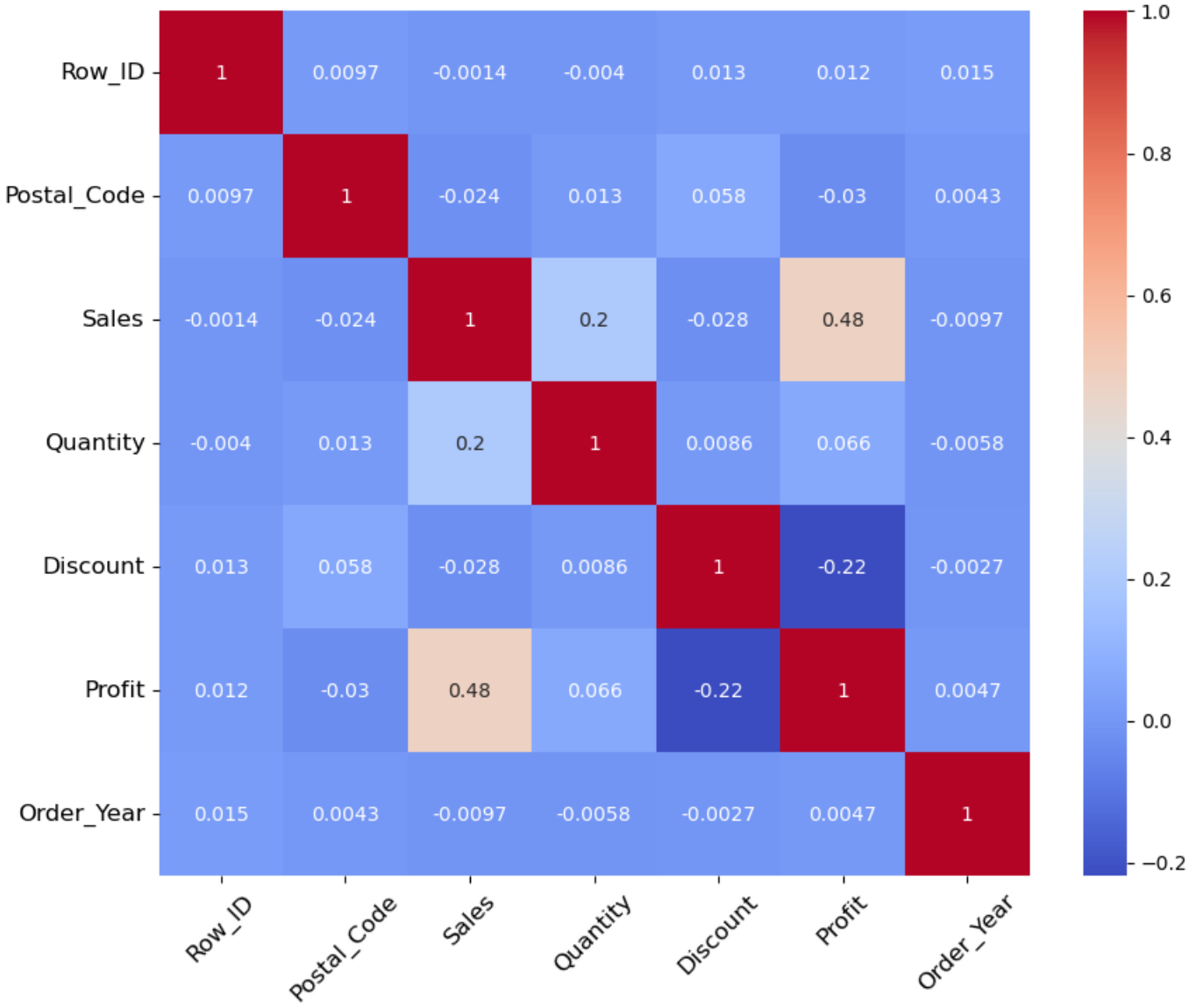
Inference from the above bar Plot ;

1. Looking at the graphs, we can see that the subcategories who sold the most, didn't make the highest profit.
2. Phones was the most sold sub-category, and Copiers was the most profitable sub-category.
3. Tables reached almost 800,000 in sales, but made more than 50,000 in negative profit.

Q7. What factors affect Sale the most?

To answer this question, To answer this question, we can create a correlation matrix using the px.imshow() function in Plotly:

In [24]:
```
# Using Seaborn Library
# Create heatmap

corr_matrix = df.corr()
fig, ax = plt.subplots(figsize=(10,8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', ax=ax)

# Customize plot
# ax.set_title("Correlation Matrix for US Retail Data", fontsize=16)
ax.tick_params(axis='x', labelsize=12, rotation=45)
ax.tick_params(axis='y', labelsize=12, rotation=0)

# Show the plot
plt.show()
```



Inference from the Corr Chart Price with other Cols

According to the correlation matrix, the variables don't really affect each other.

In [ ]: