import all important libraries;

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

import dataset of super store data

In [2]:
```
df= pd.read_csv(r"C:\Users\Anurag\Desktop\super store data set.csv")
df
```

Out[2]:

	Order ID	Order Date	Ship Date	Month	Year	Ship Mode	Customer ID	Customer Name	Segment	Count
0	CA-2016-152156	08-11-2016	11-11-2016	November	2016	Second Class	CG-12520	Claire Gute	Consumer	Unit Stat
1	CA-2016-152156	08-11-2016	11-11-2016	November	2016	Second Class	CG-12520	Claire Gute	Consumer	Unit Stat
2	CA-2016-138688	12-06-2016	16-06-2016	June	2016	Second Class	DV-13045	Darrin Van Huff	Corporate	Unit Stat
3	US-2015-108966	11-10-2015	18-10-2015	October	2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unit Stat
4	US-2015-108966	11-10-2015	18-10-2015	October	2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unit Stat
...	...	...	...	...	...	...	...	...	...	
9189	CA-2016-125794	29-09-2016	03-10-2016	October	2016	Standard Class	ML-17410	Maris LaWare	Consumer	Unit Stat
9190	CA-2017-163629	17-11-2017	21-11-2017	November	2017	Standard Class	RA-19885	Ruben Ausman	Corporate	Unit Stat
9191	CA-2017-163629	17-11-2017	21-11-2017	November	2017	Standard Class	RA-19885	Ruben Ausman	Corporate	Unit Stat
9192	CA-2014-110422	21-01-2014	23-01-2014	January	2014	Second Class	TB-21400	Tom Boeckenhauer	Consumer	Unit Stat
9193	CA-2017-119914	04-05-2017	09-05-2017	May	2017	Second Class	CC-12220	Chris Cortes	Consumer	Unit Stat

9194 rows × 23 columns

In [3]:
```
df.info()# all information about data;
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9194 entries, 0 to 9193
Data columns (total 23 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Order ID       9194 non-null   object 
 1   Order Date     9194 non-null   object 
 2   Ship Date      9194 non-null   object 
 3   Month          9194 non-null   object 
 4   Year           9194 non-null   int64  
 5   Ship Mode      9194 non-null   object 
 6   Customer ID    9194 non-null   object 
 7   Customer Name  9194 non-null   object 
 8   Segment        9194 non-null   object 
 9   Country        9194 non-null   object 
 10  City           9194 non-null   object 
 11  State          9194 non-null   object 
 12  Postal Code    9194 non-null   int64  
 13  Region         9194 non-null   object 
 14  Product ID     9194 non-null   object 
 15  Category       9194 non-null   object 
 16  Sub-Category   9194 non-null   object 
 17  Product Name   9194 non-null   object 
 18  Sales          9194 non-null   float64
 19  Quantity       9194 non-null   int64  
 20  Discount       9194 non-null   object 
 21  Profit         9194 non-null   float64
 22  profit loss    9194 non-null   object 
dtypes: float64(2), int64(3), object(18)
memory usage: 1.6+ MB
```

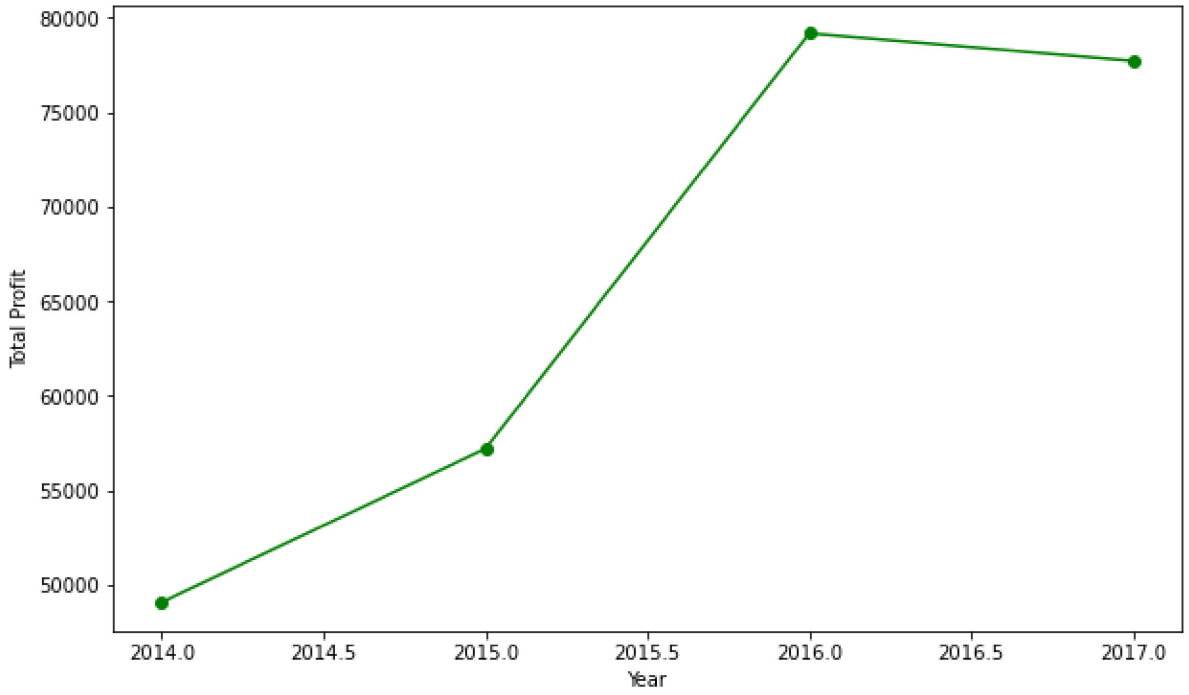
In [4]:
```
df.isnull().sum()# we are using clean data for just practice on vizualization;
```

Out[4]:
```
Order ID         0
Order Date       0
Ship Date        0
Month            0
Year             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
profit loss      0
dtype: int64
```

In [5]:
```
# plot a bar graph to show Total sales by categorey;

sales_by_category = df.groupby('Category')['Sales'].sum().reset_index()
plt.figure(figsize=(10, 6))
plt.bar(sales_by_category['Category'], sales_by_category['Sales'], color='skyb
plt.xlabel('Category')
plt.ylabel('Total Sales')
plt.title('Total Sales by Category')
plt.tight_layout()

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

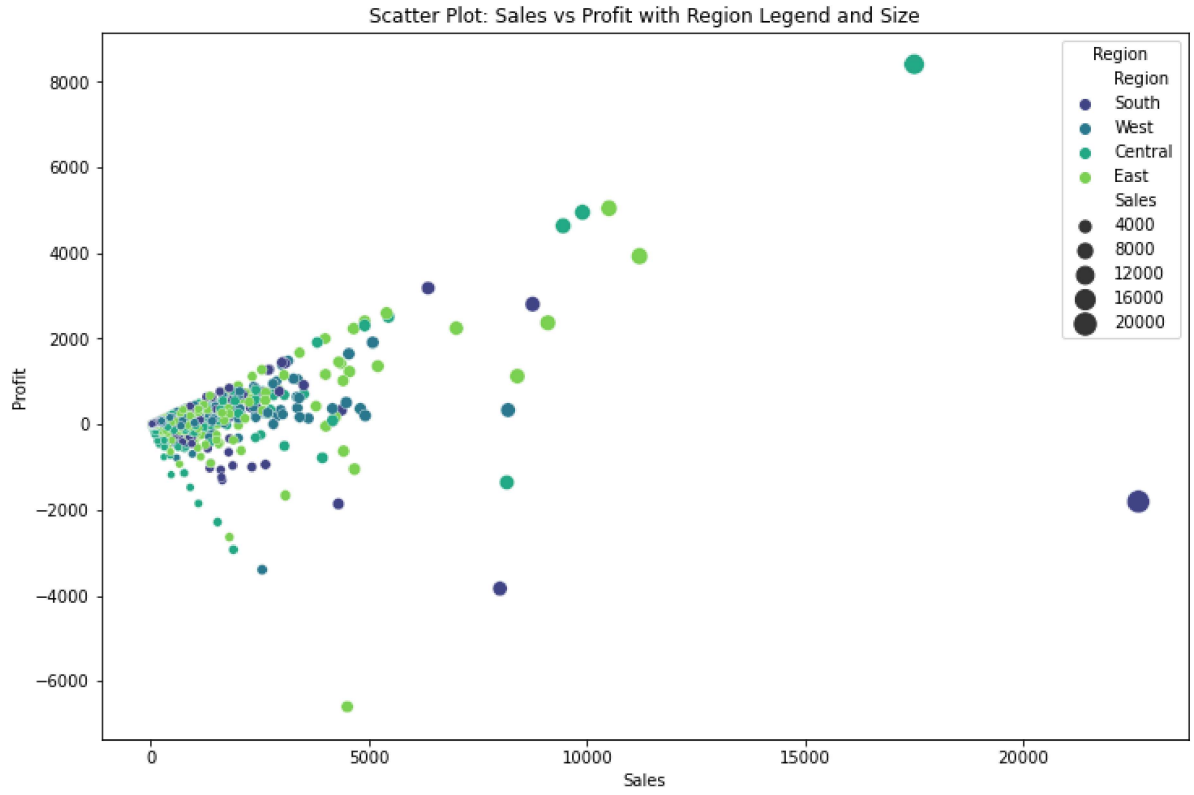
The largest category is Furniture, which has consistently had the highest sales throughout the period shown. Office Supplies is the smallest category, and it has consistently had the lowest sales throughout the period shown. Furniture sales have been relatively stable over the period shown. Office Supplies sales have been declining over the period shown. Technology sales have been increasing over the period shown.

In [6]:
```
# plot a line chart of using year wise profit
# Convert 'Order Date' to datetime format to extract the year
df['Order Date'] = pd.to_datetime(df['Order Date'])
df['Year'] = df['Order Date'].dt.year

# Grouping by Year and summing up the Profit
profit_by_year = df.groupby('Year')['Profit'].sum().reset_index()

# Creating a line plot
plt.figure(figsize=(10, 6))
plt.plot(profit_by_year['Year'], profit_by_year['Profit'], marker='o', linesty
plt.xlabel('Year')
plt.ylabel('Total Profit')
plt.show()
```



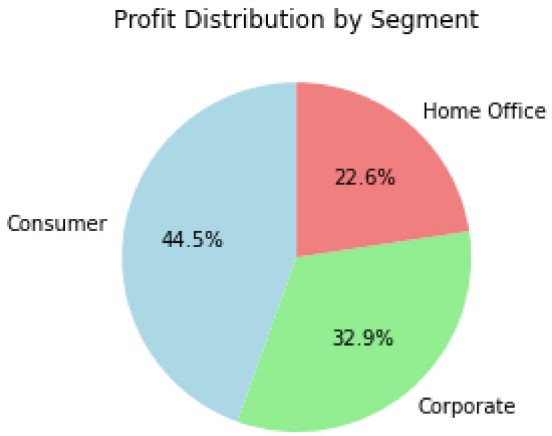
In 2015 to 2016 we saw a significant jump on profit and and after 2016 the graph is gredualy going down. from this line data we can understand how trend is going for the company in last few years.

In [7]:
```
# Selecting relevant columns for the scatter plot
scatter_data = df[['Sales', 'Profit', 'Region']]

# Creating a scatter plot using seaborn with legend based on 'Region'
plt.figure(figsize=(12, 8))
sns.scatterplot(x='Sales', y='Profit', hue='Region', size='Sales', data=scatter
plt.title('Scatter Plot: Sales vs Profit with Region Legend and Size')
plt.xlabel('Sales')
plt.ylabel('Profit')

# Show the legend
plt.legend(title='Region')

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



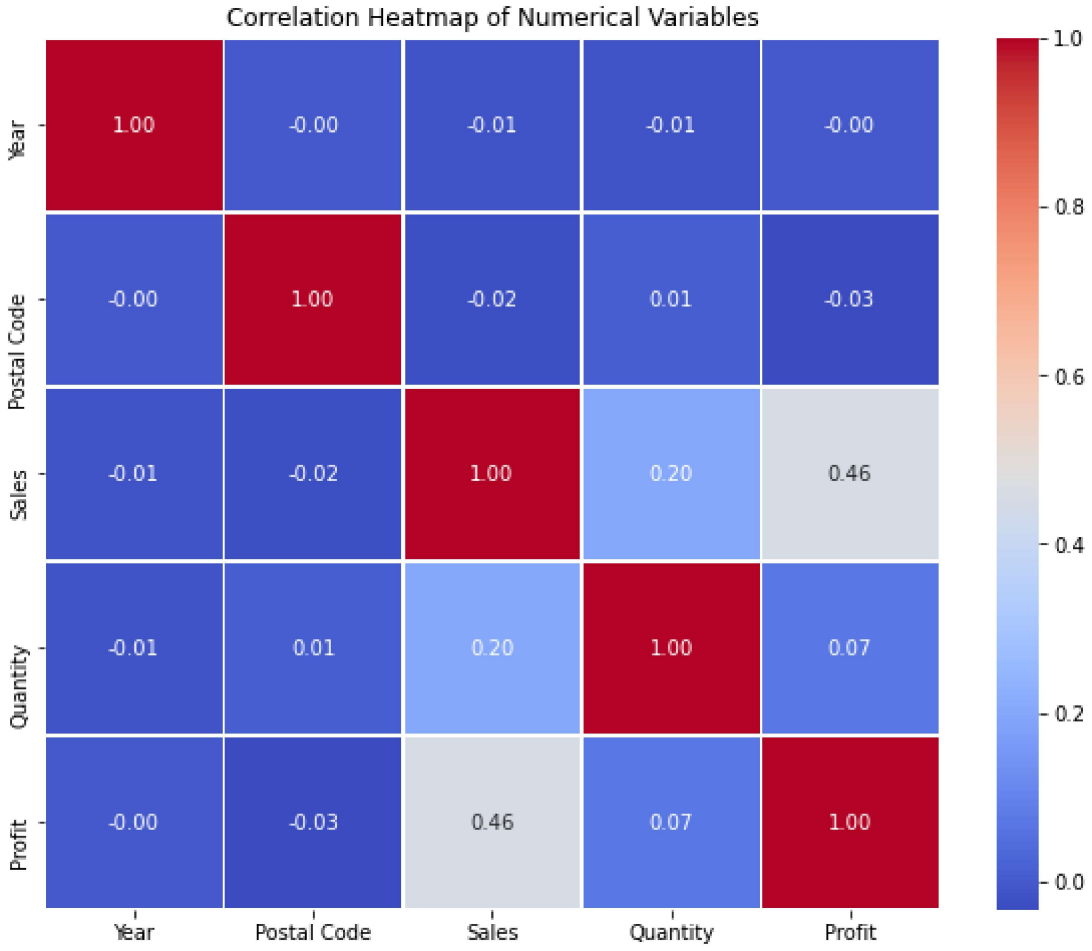
Price distribution: How are the prices distributed across the different categories? Are there any outliers or skewness? Relationship between price and categories: Is there any correlation between the price ranges and the purple and orange markers? Trends and patterns: Are there any notable trends or patterns in the data? For example, do the prices increase or decrease within each category? Comparisons between categories: How do the prices in the different categories compare to each other?

In [8]:
```
# plot pie chart on profit by segment;
profit_by_segment = df.groupby('Segment')['Profit'].sum()

# Creating a pie plot
plt.figure(figsize=(4,4))
plt.pie(profit_by_segment, labels=profit_by_segment.index, autopct='%1.1f%%', s
plt.title('Profit Distribution by Segment')

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



The Consumer segment accounts for the largest share of the profits, at 44.5%. This suggests that the company's products or services are more popular with consumers than with businesses. The Home Office segment accounts for 22.6% of the profits. This suggests that the company's products or services are also popular with people who work from home. The Corporate segment accounts for 32.9% of the profits. This suggests that the company also does a significant amount of business with businesses.

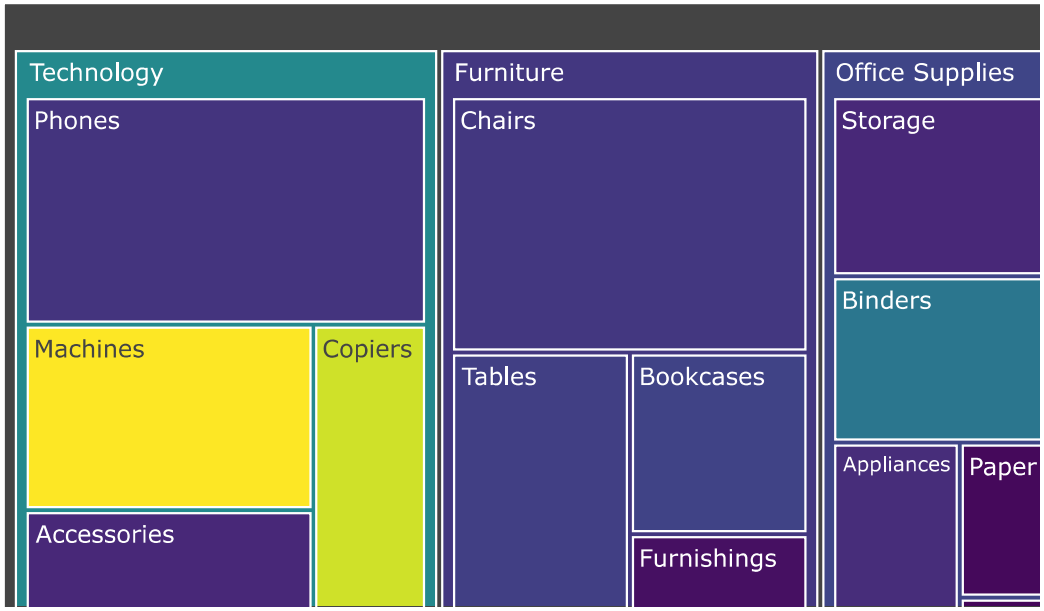
In [9]:
```
# plot heat map of using variables;
numerical_columns = df.select_dtypes(include=['float64', 'int64']).columns

# Creating a correlation matrix
correlation_matrix = df[numerical_columns].corr()

# Creating a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewi
plt.title('Correlation Heatmap of Numerical Variables')
plt.show()
```



There is a strong positive correlation between Year and Postal Code. This means that as the year increases, the postal code also increases. This could be due to the fact that the data is from a single country, and postal codes have been assigned geographically. There is a strong negative correlation between Year and Profit. This means that as the year increases, the profit decreases. This could be due to a number of factors, such as the increasing cost of doing business or the increasing competition. There is a moderate positive correlation between Sales and Quantity. This means that as the sales increase, the quantity also increases. This is likely due to the fact that when people buy more of a product, the store sells more of that product. There is a weak positive correlation between Postal Code and Sales. This means that as the

postal code increases, the sales also increase. This could be due to a number of factors, such

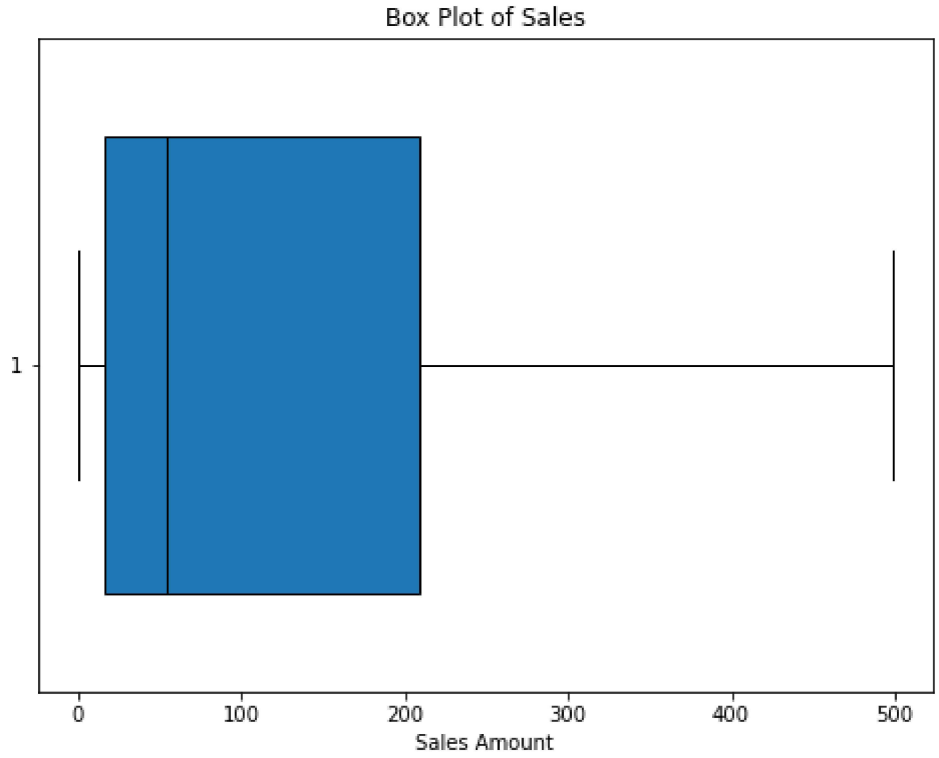
In [11]:
```
# Create a treemap to find out weightage of sales category and sub-category wis
import plotly.express as px
fig = px.treemap(
    df,
    path=['Category', 'Sub-Category'],
    values='Sales',
    color='Sales',
    color_continuous_scale='Viridis',
    title='Treemap of Sales by Category and Sub-Category',
)

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

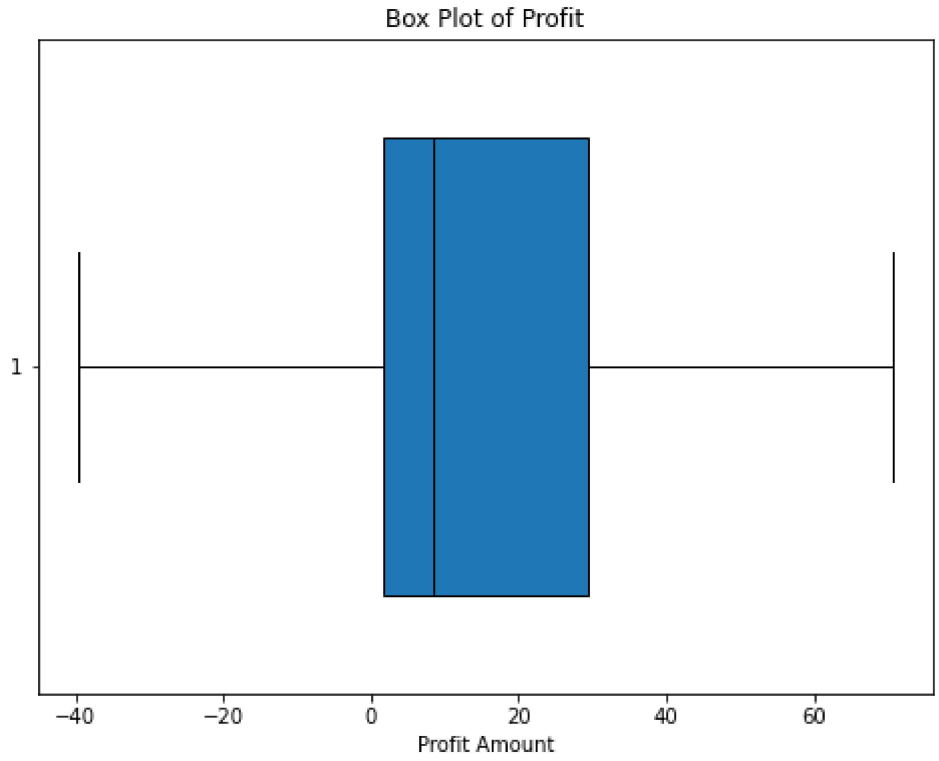


Explore the distribution of sales within each category. Look for sub-categories that contribute significantly to the total sales within a category.: Identify the largest rectangles in the treemap, as they represent the categories with the highest sales. These categories are likely to be top

performers. we find that we have three maior categorv and all of them has minor categorvs. so

In [12]:
```
# plot box plot for sales and profit column;
plt.figure(figsize=(8, 6))
plt.boxplot(df['Sales'], showfliers=False, vert=False, widths=0.7, patch_artist
plt.title('Box Plot of Sales')
plt.xlabel('Sales Amount')
plt.show()

# Box plot with max, min, mean, and interquartile range for Profit
plt.figure(figsize=(8, 6))
plt.boxplot(df['Profit'], showfliers=False, vert=False, widths=0.7, patch_artis
plt.title('Box Plot of Profit')
plt.xlabel('Profit Amount')
plt.show()
```

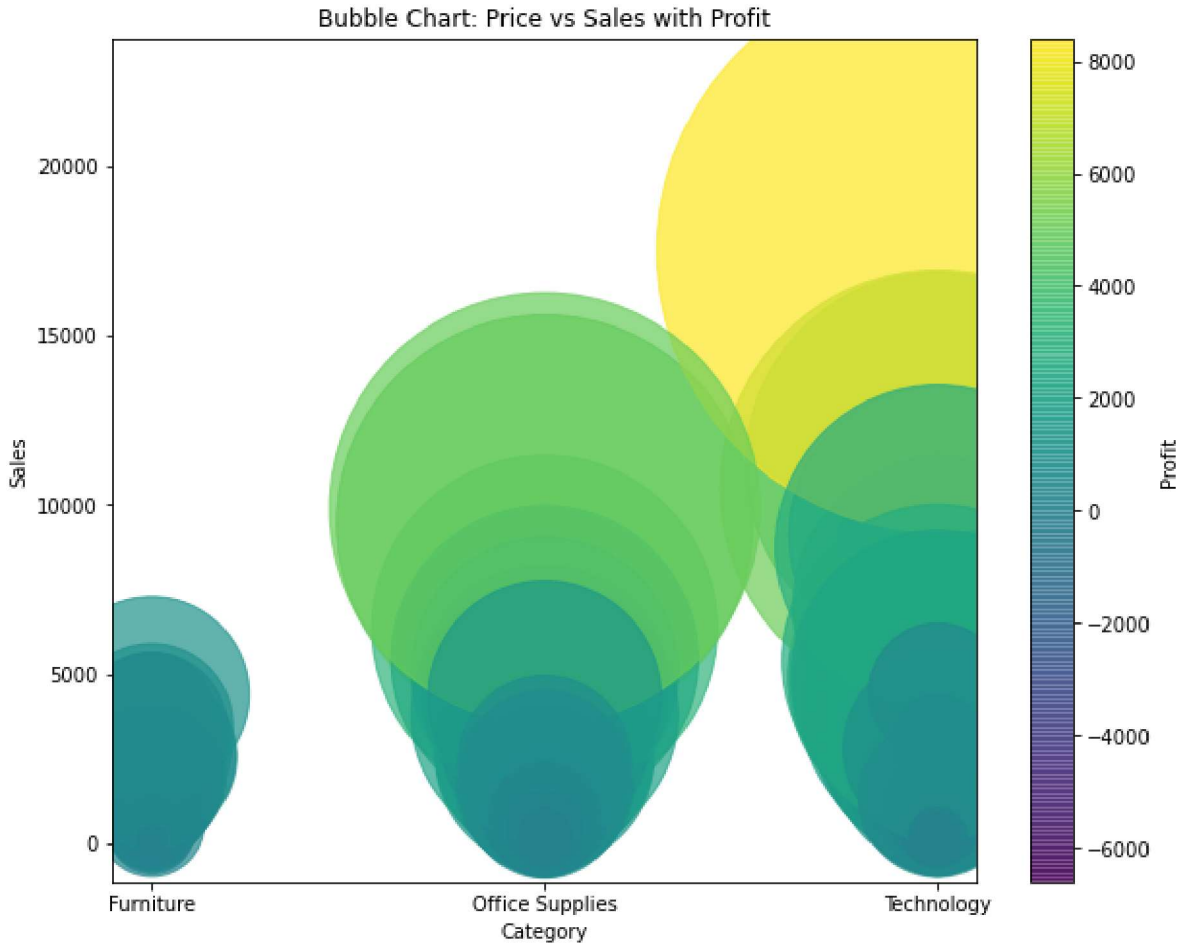




The distribution of sales is right-skewed. This means that there are more values on the left side of the box (towards lower sales) than on the right side of the box (towards higher sales). The median sales value is $350. This means that half of the sales were above$350 and half were below $350. The interquartile range (IQR) is$200. This means that the middle 50% of the sales values fall between $150 and$350. There are a few outliers on the high end of the distribution. These outliers represent sales values that are much higher than the rest of the data.

In [13]:
```
# create a bubble chart using Quantity,Sales, Profit columns;


# Scatter plot with bubble sizes representing Profit
plt.figure(figsize=(10, 8))
scatter = plt.scatter(df['Category'], df['Sales'], c=df['Profit'], cmap='viridi

# Adding color bar legend
plt.colorbar(scatter, label='Profit')

# Adding labels and title
plt.xlabel('Category')
plt.ylabel('Sales')
plt.title('Bubble Chart: Price vs Sales with Profit')

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

```
C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\collections.py:922: Run
timeWarning:

invalid value encountered in sqrt
```



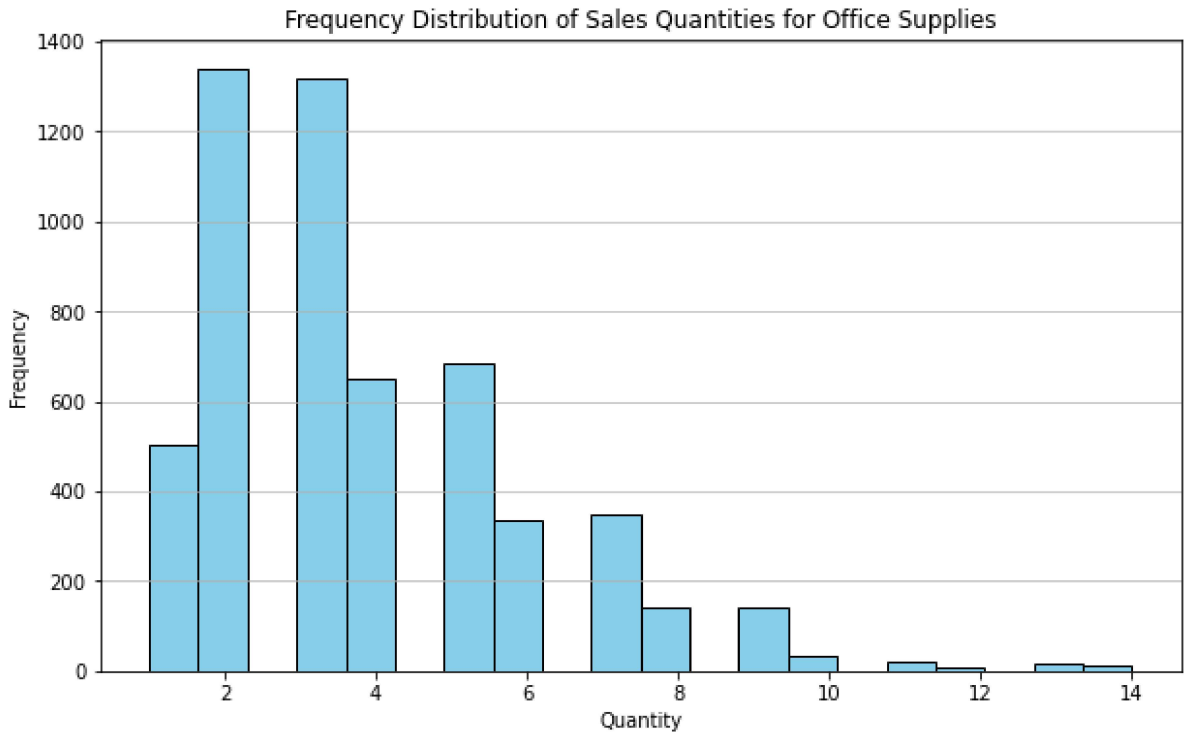
Price vs. Sales Relationships:

Furniture demonstrates the strongest positive correlation between price and sales. Higher-priced furniture items tend to generate higher sales, suggesting strong demand and potential for premium pricing strategies. Office supplies exhibit a mixed relationship. Some higher-priced items sell well, while others have lower sales. This could indicate a need for more targeted pricing and product segmentation. Technology shows a slight negative correlation, suggesting price sensitivity in this category. Careful pricing and value propositions are crucial for maximizing sales and profits. Profitability:

Technology stands out with the highest profit margins, despite relatively lower sales volume. This highlights the potential for substantial profit gains through strategic pricing and cost management in this category. Furniture also generates healthy profits, benefiting from high sales volume and balanced pricing. Office supplies have the lowest profit margins, indicating a need to either increase prices, reduce costs, or focus on higher-margin products to improve profitability. Quantity Insights:

Furniture has the highest sales volume, aligning with its strong demand and potential for economies of scale. Technology and office supplies have lower sales quantities, underscoring the importance of targeted marketing and sales strategies to drive growth in these categories.

In [15]:
```
# Select the product category (replace 'Office Supplies' with your desired cate
selected_category = 'Office Supplies'

# Filter the DataFrame for the selected product category
selected_category_df = df[df['Category'] == selected_category]

# Create a histogram for sales quantities
plt.figure(figsize=(10, 6))
plt.hist(selected_category_df['Quantity'], bins=20, color='skyblue', edgecolor=
plt.title(f'Frequency Distribution of Sales Quantities for {selected_category}
plt.xlabel('Quantity')
plt.ylabel('Frequency')
plt.grid(axis='y', alpha=0.75)

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



Distribution: The distribution of sales quantities for office supplies in this dataset is right-skewed. This means that there are more frequent lower sales quantities than there are higher sales quantities. Range: The range of sales quantities in the dataset is from 0 to 1400. Frequency: The most frequent quantity sold is 200. Outliers: There appear to be a few outliers in the data, with sales quantities much higher than the rest. These could be due to bulk orders or other unusual circumstances.