In [3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
sales_data = pd.read_csv('D:/Downloads/archive/Online-eCommerce.csv')
```

In [9]:
```
sales_data.head()
```

Out[9]:

	Order_Number	State_Code	Customer_Name	Order_Date	Status	Product	Category	Brand	Cost	Sales	Qu
0	139374.0	AP	Adhir Samal	2020-01-11	Delivered	512 GB M.2	SSD	Samsung	6500.0	8450.0	
1	139375.0	AP	Dannana Jhammi	2020-01-11	Delivered	RYZEN 3rd gen. 3500	CPU	Intel	8500.0	11050.0	
2	139376.0	AS	Vipin Kumar	2020-01-11	Delivered	2GB Graphic Card	Graphic Card	Nvidia	7000.0	9100.0	
3	139377.0	BR	Ranjeet Kumar	2020-01-11	Delivered	16 GB DDR4 RAM	RAM	Hynix	6550.0	8515.0	
4	139378.0	CG	Sajal Singhal	2020-01-11	Order	Standard ATX motherboard	MotherBoard	Gigabyte	7650.0	9945.0	

In [10]:
```
sales_data.info()
sales_data.isnull()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5095 entries, 0 to 5109
Data columns (total 14 columns):
 #   Column               Non-Null Count  Dtype         
---  ------               --------------  -----         
 0   Order_Number         5095 non-null   float64       
 1   State_Code           5095 non-null   object        
 2   Customer_Name        5095 non-null   object        
 3   Order_Date           5095 non-null   datetime64[ns]
 4   Status               5095 non-null   object        
 5   Product              5095 non-null   object        
 6   Category             5095 non-null   object        
 7   Brand                5095 non-null   object        
 8   Cost                 5095 non-null   float64       
 9   Sales                5095 non-null   float64       
 10  Quantity             5095 non-null   float64       
 11  Total_Cost           5095 non-null   float64       
 12  Total_Sales          5095 non-null   float64       
 13  Assigned Supervisor  5095 non-null   object        
dtypes: datetime64[ns](1), float64(6), object(7)
memory usage: 597.1+ KB
```

Out[10]:

	Order_Number	State_Code	Customer_Name	Order_Date	Status	Product	Category	Brand	Cost	Sales	Quantity	Tot
0	False	False	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	
5105	False	False	False	False	False	False	False	False	False	False	False	
5106	False	False	False	False	False	False	False	False	False	False	False	
5107	False	False	False	False	False	False	False	False	False	False	False	
5108	False	False	False	False	False	False	False	False	False	False	False	
5109	False	False	False	False	False	False	False	False	False	False	False	

5095 rows × 14 columns

In [11]:
```
sales_data.dropna(inplace=True)
```

In [12]:
```
sales_data['Order_Date'] = pd.to_datetime(sales_data['Order_Date'], format='%d/%m/%Y', errors='coerce'
```

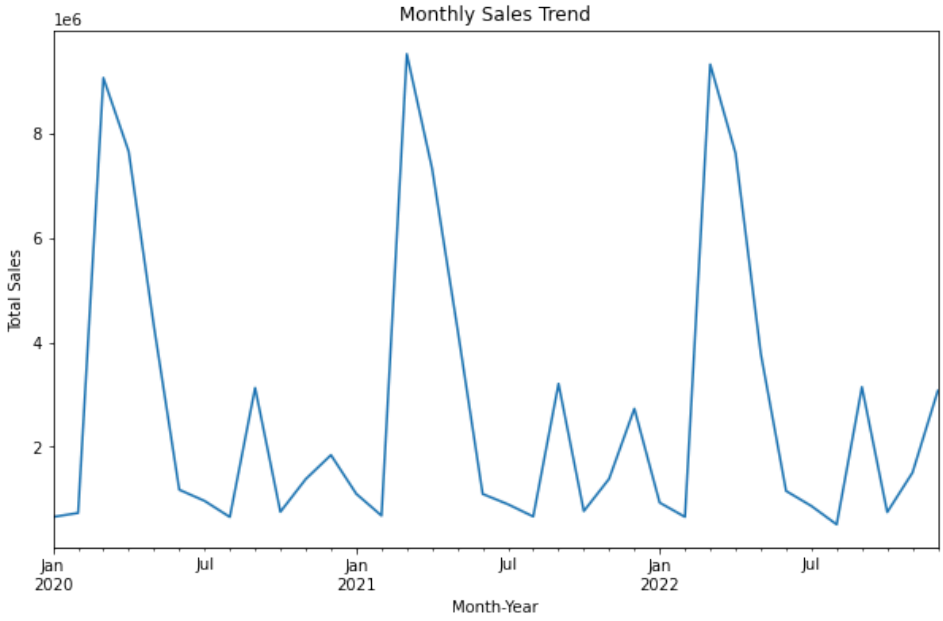
In [13]:
```
total_sales = sales_data['Total_Sales'].sum()
total_cost = sales_data['Total_Cost'].sum()
total_profit = total_sales - total_cost
print(f"Total Sales: {total_sales}, Total Cost: {total_cost}, Total Profit: {total_profit}")
```

```
Total Sales: 99298043.0, Total Cost: 76383110.0, Total Profit: 22914933.0
```

In [14]:
```
top_products = sales_data.groupby('Product')['Total_Sales'].sum().sort_values(ascending=False).head(5)
```

In [15]:
```
sales_data['Month_Year'] = sales_data['Order_Date'].dt.to_period('M')
monthly_sales = sales_data.groupby('Month_Year')['Total_Sales'].sum()
```

In [16]:
```
plt.figure(figsize=(10,6))
monthly_sales.plot(kind='line')
plt.title('Monthly Sales Trend')
plt.xlabel('Month-Year')
plt.ylabel('Total Sales')
plt.show()
```



In [17]:
```
top_categories = sales_data.groupby('Category')['Total_Sales'].sum().sort_values(ascending=False)
```

In [18]:
```
top_categories
```

Out[18]:
```
Category
Monitor          23297105.0
CPU              18760300.0
Graphic Card     13113100.0
HDD              12886250.0
SSD              10191350.0
Mouse             3831893.0
RAM               3154697.0
Motherboard       3106545.0
Cabinet           2947594.0
Printer           2873052.0
Computer Case     1917994.0
NIC               1771484.0
Keyboard          1389895.0
MotherBoard         56784.0
Name: Total_Sales, dtype: float64
```

In [19]:
```
top_products
```

Out[19]:
```
Product
2GB Graphic Card              7198100.0
I7 - intel 12th Generation    6616350.0
26" LCD Display               6558630.0
4GB Graphic card              5915000.0
21" LCD Display               4777500.0
Name: Total_Sales, dtype: float64
```

In [20]:
```
sales_data
```

Out[20]:

	Order_Number	State_Code	Customer_Name	Order_Date	Status	Product	Category	Brand	Cost	Sal
0	139374.0	AP	Adhir Samal	2020-01-11	Delivered	512 GB M.2	SSD	Samsung	6500.0	845(
1	139375.0	AP	Dannana Jhammi	2020-01-11	Delivered	RYZEN 3rd gen. 3500	CPU	Intel	8500.0	1105(
2	139376.0	AS	Vipin Kumar	2020-01-11	Delivered	2GB Graphic Card	Graphic Card	Nvidia	7000.0	910(
3	139377.0	BR	Ranjeet Kumar	2020-01-11	Delivered	16 GB DDR4 RAM	RAM	Hynix	6550.0	851[
4	139378.0	CG	Sajal Singhal	2020-01-11	Order	Standard ATX motherboard	MotherBoard	Gigabyte	7650.0	994[
...	...	...	...	...	...	...	...	...	...	
5105	144464.0	TN	Rahul Kumar Prajapati	2022-12-31	Delivered	406 GB SSD	SSD	Samsung	4500.0	585(
5106	144465.0	TR	Sagar Jeur	2022-12-31	Shipped	Intel i3 11th gen.	CPU	Intel	8500.0	1105(
5107	144466.0	UK	Rhushikesh Mane	2022-12-31	Order	4GB Graphic card	Graphic Card	Nvidia	12500.0	1625(
5108	144467.0	UP	Ashish Kumar	2022-12-31	Processing	158 GB DDR4 RAM	RAM	Hynix	3500.0	455(
5109	144468.0	WB	Javed Akhter	2022-12-31	Delivered	BTX motherboard	Motherboard	Gigabyte	4500.0	585(

5095 rows × 15 columns

In [21]:
```
top_brands = sales_data.groupby('Brand')['Total_Sales'].sum().sort_values(ascending=False)
```

In [22]:
```
top_brands
```

Out[22]:
```
Brand
Intel              18760300.0
Samsung            16166345.0
Dell               14235195.0
Nvidia             13113100.0
Western Digital     8050250.0
Acer                6558630.0
Gigabyte            5886855.0
Hynix               5538520.0
Seagate             4836000.0
MSI                 3205254.0
Asus                2947594.0
Name: Total_Sales, dtype: float64
```

In [23]:
```
sales_by_state = sales_data.groupby('State_Code')['Total_Sales'].sum().sort_values(ascending=False)
```

In [24]:
```
sales_by_state
```

Out[24]:
```
State_Code
MH    17621084.0
UP     9264645.0
GJ     9137726.0
DL     5061953.0
BR     4862221.0
TR     3660657.0
TN     3428763.0
CH     1958697.0
MP     1885910.0
MZ     1874418.0
WB     1843374.0
OR     1811498.0
SK     1800539.0
LD     1780298.0
JK     1774526.0
PY     1763229.0
RJ     1758978.0
ML     1719848.0
UK     1718899.0
DH     1717079.0
AR     1711879.0
CG     1681290.0
JH     1680003.0
DD     1676324.0
AS     1665365.0
PB     1632722.0
HP     1603667.0
KA     1586442.0
AP     1576952.0
GA     1566175.0
NL     1534390.0
MN     1515852.0
KL     1488617.0
HR     1439191.0
AN      494832.0
Name: Total_Sales, dtype: float64
```

In [25]:
```
monthly_sales
```

Out[25]:
```
Month_Year
2020-01     656162.0
2020-02     733759.0
2020-03    9063041.0
2020-04    7655609.0
2020-05    4304261.0
2020-06    1178606.0
2020-07     964808.0
2020-08     656110.0
2020-09    3128957.0
2020-10     754728.0
2020-11    1380717.0
2020-12    1842841.0
2021-01    1100541.0
2021-02     681603.0
2021-03    9519562.0
2021-04    7327775.0
2021-05    4272099.0
2021-06    1097226.0
2021-07     899665.0
2021-08     665470.0
2021-09    3207412.0
2021-10     768599.0
2021-11    1387087.0
2021-12    2730416.0
2022-01     931411.0
2022-02     657696.0
2022-03    9320805.0
2022-04    7621848.0
2022-05    3783416.0
2022-06    1154439.0
2022-07     866970.0
2022-08     513227.0
2022-09    3143790.0
2022-10     748670.0
2022-11    1506596.0
2022-12    3072121.0
Freq: M, Name: Total_Sales, dtype: float64
```

In [26]:
```
top_products
```

Out[26]:
```
Product
2GB Graphic Card              7198100.0
I7 - intel 12th Generation    6616350.0
26" LCD Display               6558630.0
4GB Graphic card              5915000.0
21" LCD Display               4777500.0
Name: Total_Sales, dtype: float64
```

In [27]:
```
top_customers = sales_data.groupby('Customer_Name')['Total_Sales'].sum().sort_values(ascending=False).
```

In [28]:
```
top_customers
```

Out[28]:
```
Customer_Name
Rahul Kumar Prajapati      2974621.0
Vipin Kumar                2865746.0
Sajal Singhal              2842554.0
Rakesh Kumar Sharma        2829840.0
Ramkrishna Das Adhikary    2775162.0
Ashwini Adsare             2757859.0
Amit Singh                 2753348.0
Adhir Samal                2751814.0
Kranti Bheke               2746575.0
Aslam Raza                 2740777.0
Name: Total_Sales, dtype: float64
```

In [29]:
```
sales_data['Profit_Margin'] = (sales_data['Total_Sales'] - sales_data['Total_Cost']) / sales_data['Tot
top_profit_margin_products = sales_data.groupby('Product')['Profit_Margin'].mean().sort_values(ascendi
```

In [30]:
```
top_profit_margin_products
```

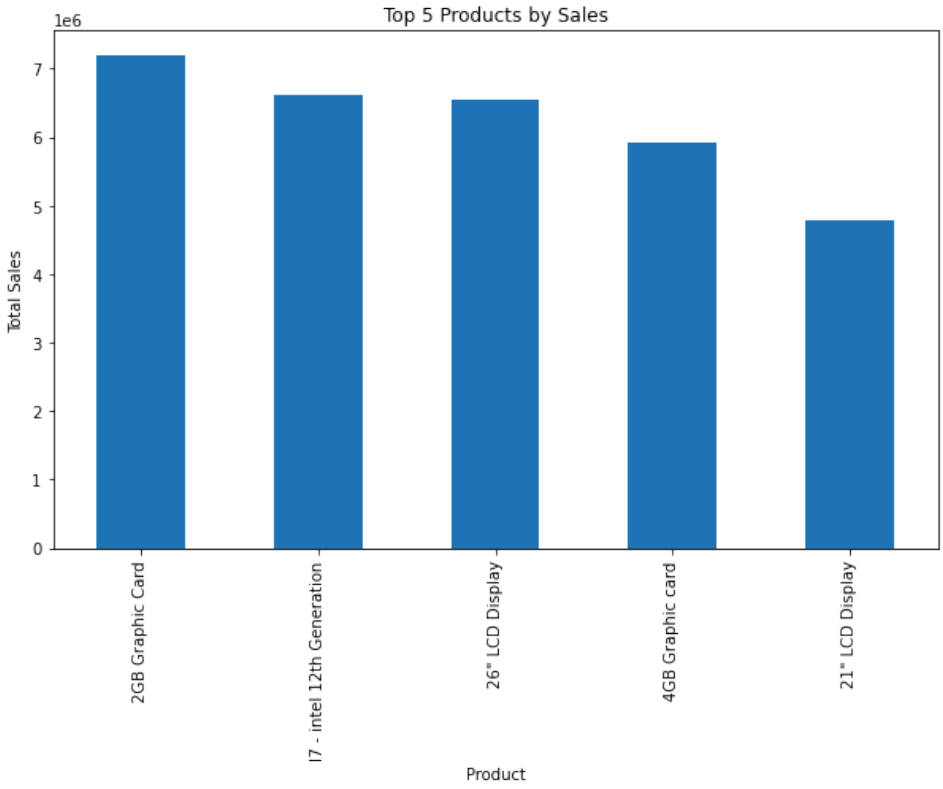
Out[30]:
```
Product
Wireless Mouse      0.230769
4GB Graphic card    0.230769
17" LCD Display     0.230769
21" LCD Display     0.230769
26" LCD Display     0.230769
Name: Profit_Margin, dtype: float64
```

In [31]:
```
sales_by_supervisor = sales_data.groupby('Assigned Supervisor')['Total_Sales'].sum().sort_values(ascen
```

In [32]:
```
sales_by_supervisor
```

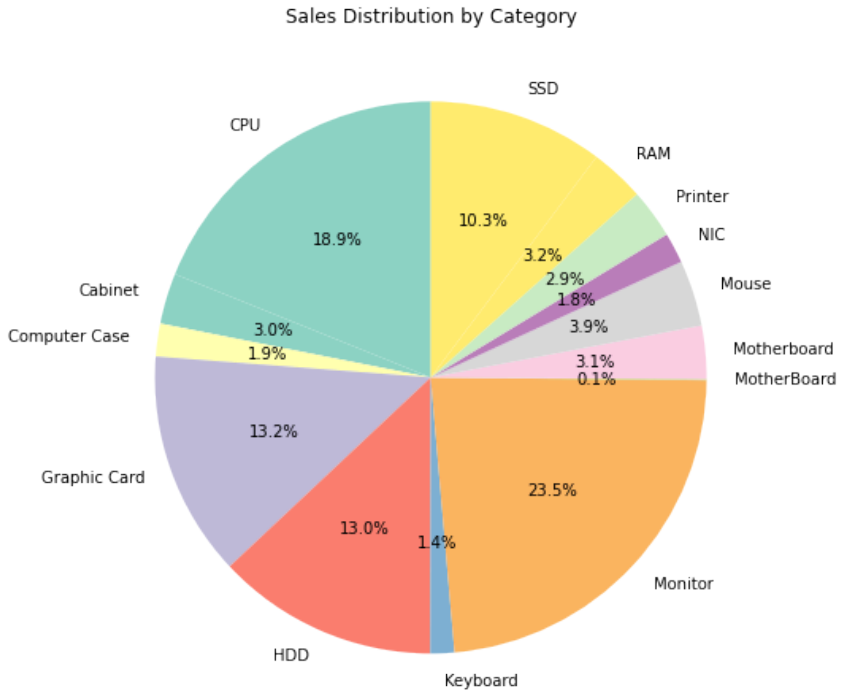
Out[32]:
```
Assigned Supervisor
Aarvi Gupta     18685368.0
Ajay Sharma     17801186.0
Vijay Singh     15939950.0
Roshan Kumar    15887079.0
Aadil Khan      15730767.0
Advika Joshi    15253693.0
Name: Total_Sales, dtype: float64
```

In [33]:
```
# Example: Top Products by Sales
top_products.plot(kind='bar', title='Top 5 Products by Sales', figsize=(10,6))
plt.xlabel('Product')
plt.ylabel('Total Sales')
plt.show()
```



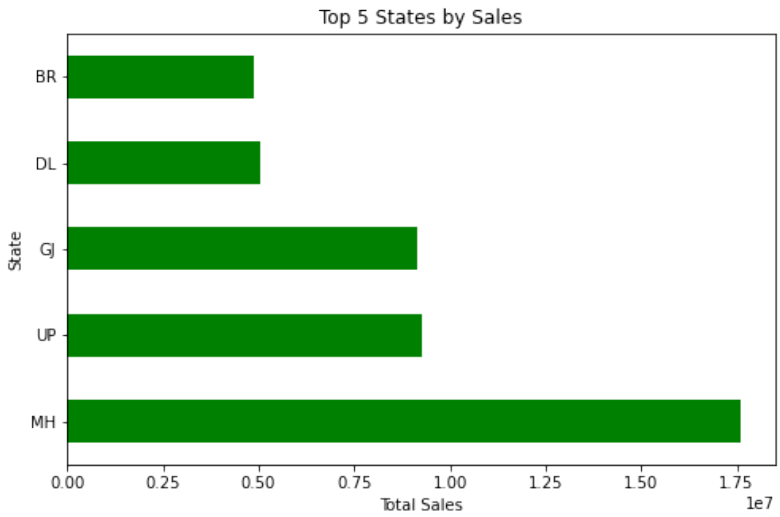
In [35]:
```
category_sales = sales_data.groupby('Category')['Total_Sales'].sum()

plt.figure(figsize=(8,8))
category_sales.plot(kind='pie', autopct='%1.1f%%', startangle=90, cmap='Set3')
plt.title('Sales Distribution by Category')
plt.ylabel('')
plt.show()
```



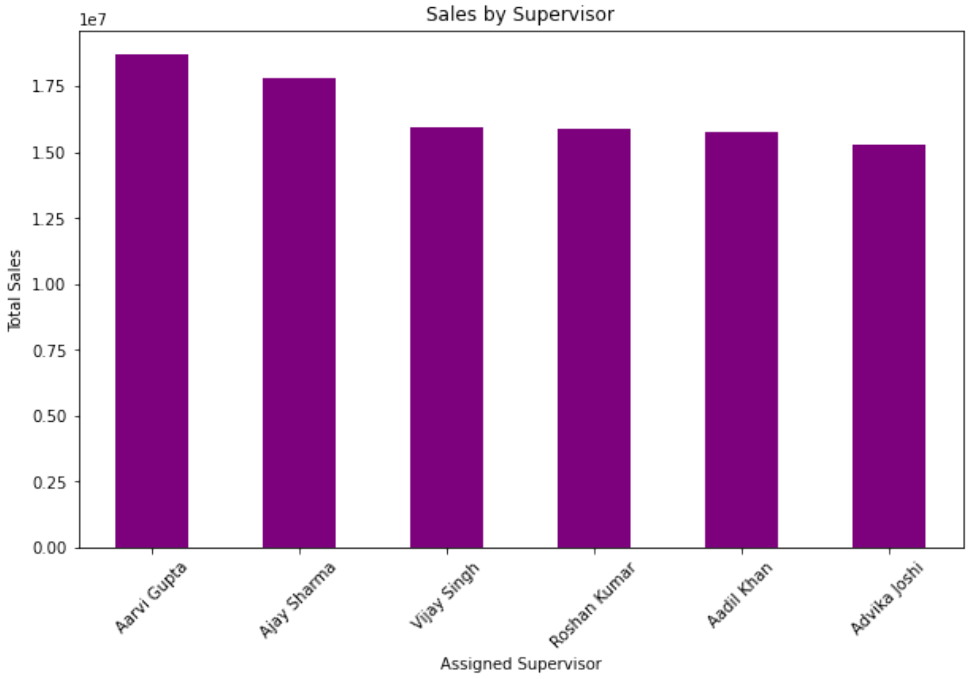
In [36]:
```
top_states = sales_data.groupby('State_Code')['Total_Sales'].sum().sort_values(ascending=False).head(5

plt.figure(figsize=(8,5))
top_states.plot(kind='barh', color='green')
plt.title('Top 5 States by Sales')
plt.xlabel('Total Sales')
plt.ylabel('State')
plt.show()
```



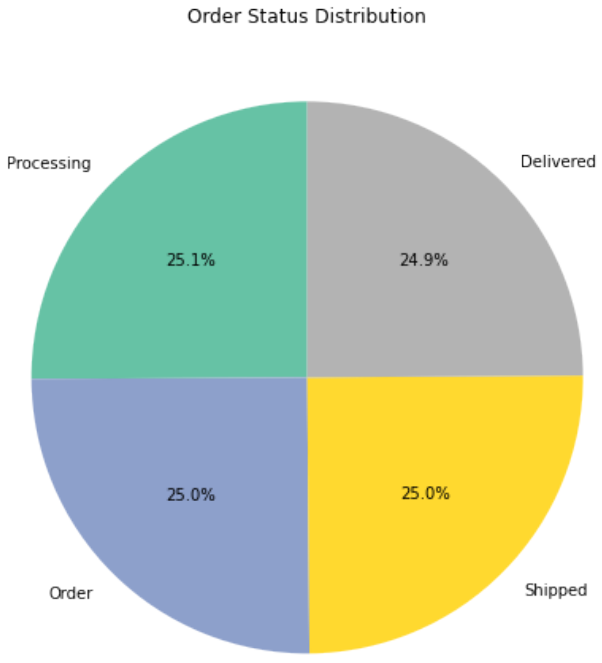
In [37]:
```
supervisor_sales = sales_data.groupby('Assigned Supervisor')['Total_Sales'].sum().sort_values(ascendin


plt.figure(figsize=(10,6))
supervisor_sales.plot(kind='bar', color='purple')
plt.title('Sales by Supervisor')
plt.xlabel('Assigned Supervisor')
plt.ylabel('Total Sales')
plt.xticks(rotation=45)
plt.show()
```



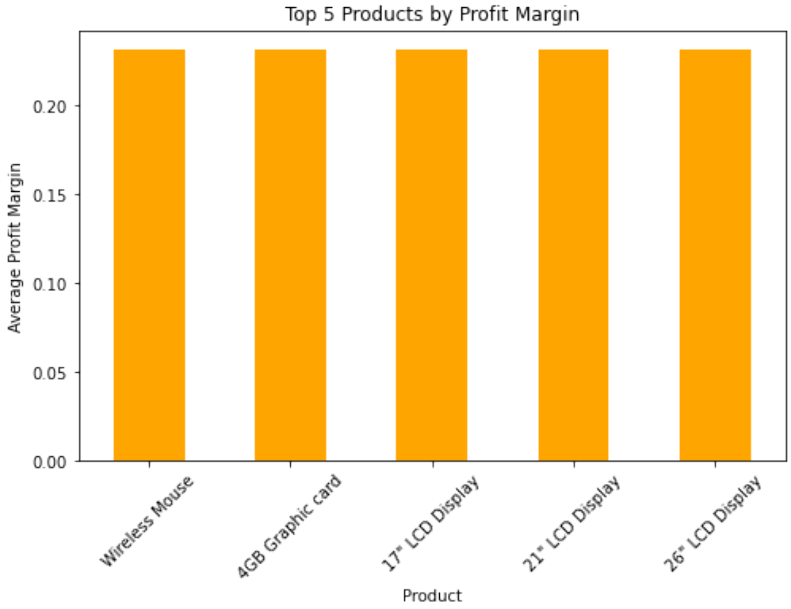
In [38]:
```
status_distribution = sales_data['Status'].value_counts()


plt.figure(figsize=(8,8))
status_distribution.plot(kind='pie', autopct='%1.1f%%', startangle=90, cmap='Set2')
plt.title('Order Status Distribution')
plt.ylabel('')
plt.show()
```



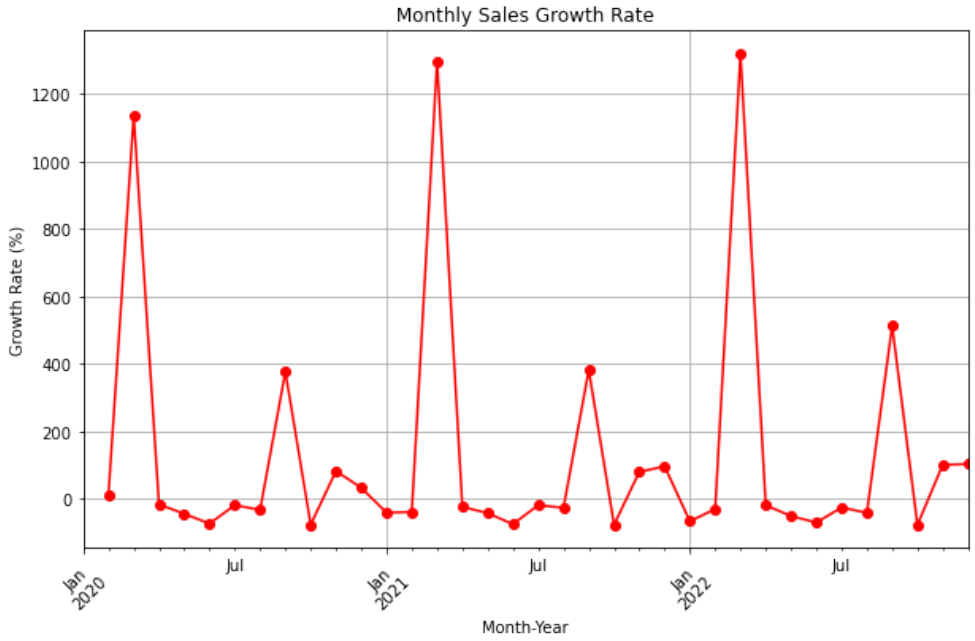
In [39]:
```
sales_data['Profit_Margin'] = (sales_data['Total_Sales'] - sales_data['Total_Cost']) / sales_data['Tot
top_profit_margin_products = sales_data.groupby('Product')['Profit_Margin'].mean().sort_values(ascendi


plt.figure(figsize=(8,5))
top_profit_margin_products.plot(kind='bar', color='orange')
plt.title('Top 5 Products by Profit Margin')
plt.xlabel('Product')
plt.ylabel('Average Profit Margin')
plt.xticks(rotation=45)
plt.show()
```



In [40]:
```
monthly_sales = sales_data.groupby('Month_Year')['Total_Sales'].sum()
sales_growth_rate = monthly_sales.pct_change() * 100


plt.figure(figsize=(10,6))
sales_growth_rate.plot(kind='line', marker='o', color='red')
plt.title('Monthly Sales Growth Rate')
plt.xlabel('Month-Year')
plt.ylabel('Growth Rate (%)')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



In [ ]: