import the necessary Libraries

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import plotly.express as px
import matplotlib.pyplot as plt
%matplotlib inline
```

Load the Data

In [2]:
```
salesdata = pd.read_csv('salesdata.csv')
```

In [3]:
```
print(f'Number of rows :{salesdata.shape[0]}')
print(f'Number of columns :{salesdata.shape[1]}')
```

```
Number of rows :9994
Number of columns :9
```

In [4]:
```
salesdata.head()
```

Out[4]:

	Order Date	Customer Name	State	Category	Sub-Category	Product Name	Sales	Quantity	Profit
0	3/1/2014	Darren Powers	Texas	Office Supplies	Paper	Message Book, Wirebound, Four 5 1/2" X 4" Form...	16.45	2	5.55
1	4/1/2014	Phillina Ober	Illinois	Office Supplies	Labels	Avery 508	11.78	3	4.27
2	4/1/2014	Phillina Ober	Illinois	Office Supplies	Storage	SAFCO Boltless Steel Shelving	272.74	3	-64.77
3	4/1/2014	Phillina Ober	Illinois	Office Supplies	Binders	GBC Standard Plastic Binding Systems Combs	3.54	2	-5.49
4	5/1/2014	Mick Brown	Pennsylvania	Office Supplies	Art	Avery Hi-Liter EverBold Pen Style Fluorescent ...	19.54	3	4.88

In [5]:
```
salesdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Order Date     9994 non-null   object
 1   Customer Name  9994 non-null   object
 2   State          9994 non-null   object
 3   Category       9994 non-null   object
 4   Sub-Category   9994 non-null   object
 5   Product Name   9994 non-null   object
 6   Sales          9994 non-null   float64
 7   Quantity       9994 non-null   int64
 8   Profit         9994 non-null   float64
dtypes: float64(2), int64(1), object(6)
memory usage: 702.8+ KB
```

In [6]:
```
salesdata.describe()
```

Out[6]:

	Sales	Quantity	Profit
count	9994.000000	9994.000000	9994.000000
mean	229.858022	3.789574	28.656973
std	623.245131	2.225110	234.260203
min	0.440000	1.000000	-6599.980000
25%	17.280000	2.000000	1.730000
50%	54.490000	3.000000	8.665000
75%	209.940000	5.000000	29.360000
max	22638.480000	14.000000	8399.980000

EDA And analysis

In [7]:
```
salesdata.isna().sum()
```

Out[7]:
```
Order Date       0
Customer Name    0
State            0
Category         0
Sub-Category     0
Product Name     0
Sales            0
Quantity         0
Profit           0
dtype: int64
```

In [8]:
```
# counting the unique values of Customer name
salesdata['Customer Name'].value_counts()
```

Out[8]:
```
William Brown          37
John Lee               34
Matt Abelman           34
Paul Prost             34
Chloris Kastensmidt    32
                       ..
Carl Jackson            1
Anthony O'Donnell       1
Lela Donovan            1
Ricardo Emerson         1
Jocasta Rupert          1
Name: Customer Name, Length: 793, dtype: int64
```

In [9]:
```
#counting the unique values of Customer name
salesdata['State'].value_counts()
```

Out[9]:
```
California              2001
New York                1128
Texas                    985
Pennsylvania             587
Washington               506
Illinois                 492
Ohio                     469
Florida                  383
Michigan                 255
North Carolina           249
Arizona                  224
Virginia                 224
Georgia                  184
Tennessee                183
Colorado                 182
Indiana                  149
Kentucky                 139
Massachusetts            135
New Jersey               130
Oregon                   124
Wisconsin                110
Maryland                 105
Delaware                  96
Minnesota                 89
Connecticut               82
Oklahoma                  66
Missouri                  66
Alabama                   61
Arkansas                  60
Rhode Island              56
Utah                      53
Mississippi               53
Louisiana                 42
South Carolina            42
Nevada                    39
Nebraska                  38
New Mexico                37
Iowa                      30
New Hampshire             27
Kansas                    24
Idaho                     21
Montana                   15
South Dakota              12
Vermont                   11
District of Columbia      10
Maine                      8
North Dakota               7
West Virginia              4
Wyoming                    1
Name: State, dtype: int64
```

In [10]:
```
# counting unique products
salesdata['Product Name'].value_counts()
```

Out[10]:
```
Staple envelope                                                       48
Staples                                                               46
Easy-staple paper                                                     46
Avery Non-Stick Binders                                               20
Staples in misc. colors                                               19
                                                                      ..
AT&T EL51110 DECT                                                      1
Snap-A-Way Black Print Carbonless Speed Message, No Reply Area, Duplicate     1
Cisco 8961 IP Phone Charcoal                                           1
Hunt BOSTON Model 1606 High-Volume Electric Pencil Sharpener, Beige    1
Acco Glide Clips                                                       1
Name: Product Name, Length: 1850, dtype: int64
```

In [11]:
```
# computing the Totale of Profit
Total_profit = salesdata['Profit'].sum()
# Total sales
Total_sales = salesdata['Sales'].sum()
# Total Qty sold
Qty_sold = salesdata['Quantity'].sum()

print(f'The totale of profit : {Total_profit}$')
print("="*70)
print(f'The totale of sales : {Total_sales}')
print("="*70)
print(f'Quantity sold  : {Qty_sold}')
print("="*70)
```

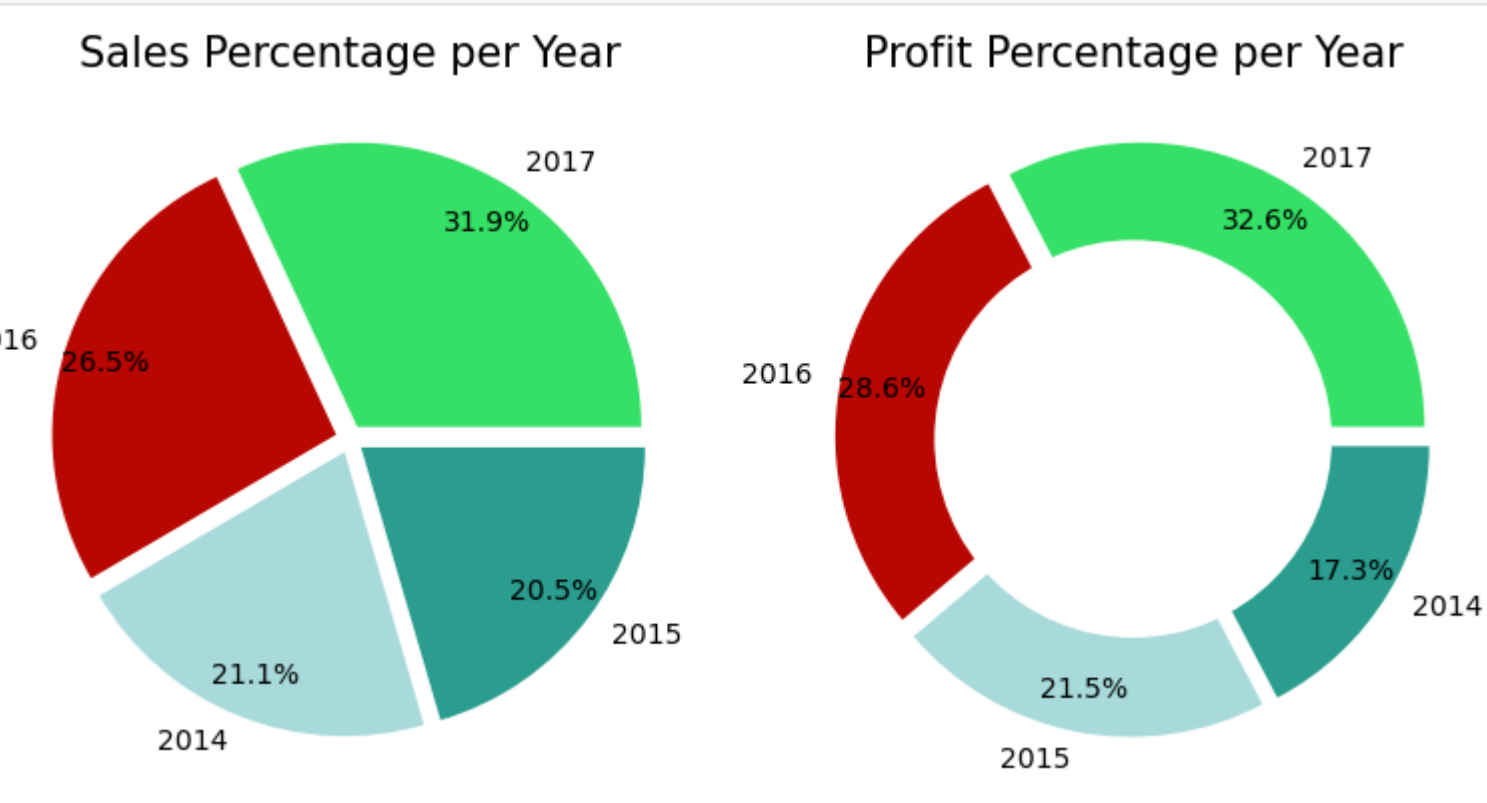
```
The totale of profit : 286397.79$
======================================================================
The totale of sales : 2297201.07
======================================================================
Quantity sold  : 37873
======================================================================
```

In [12]:
```
# convert date order to datetime type
salesdata['Order Date'] = pd.to_datetime(salesdata['Order Date'])
```

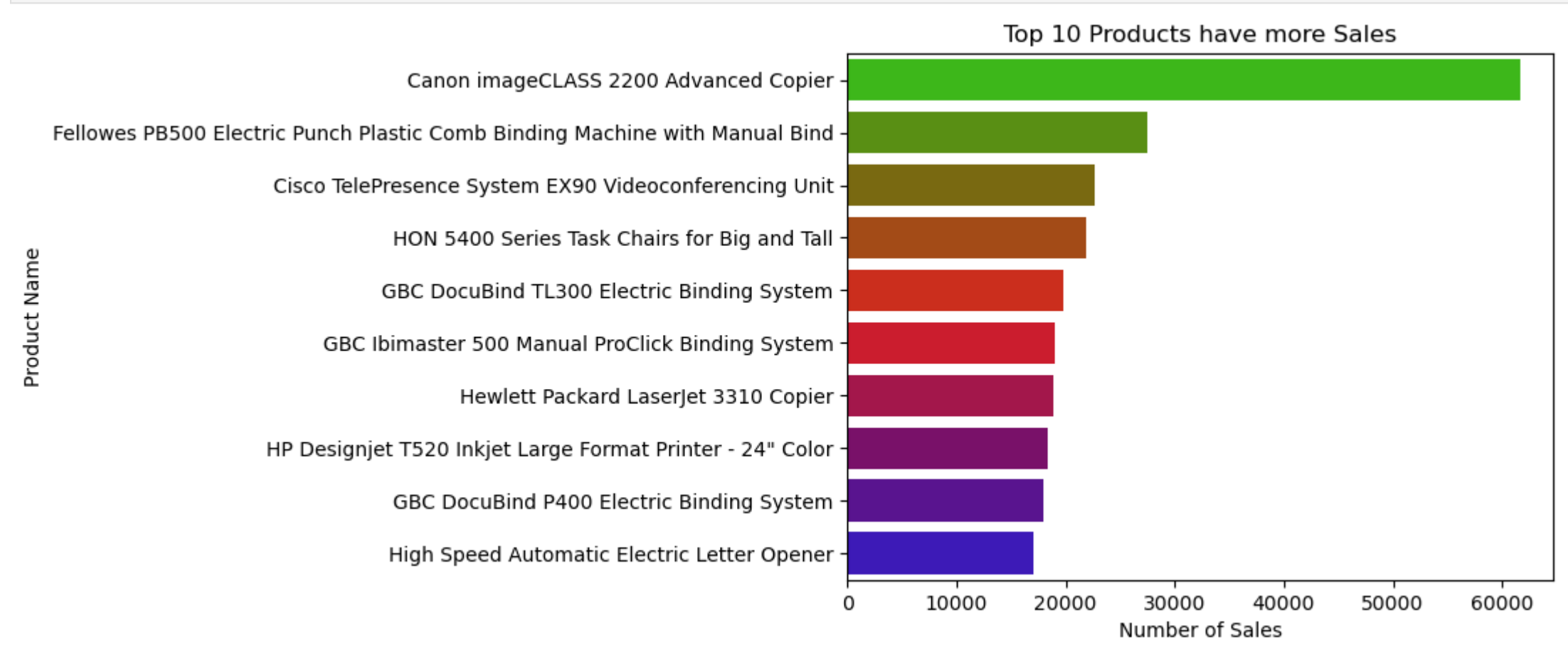
```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_12900\3473090079.py:2: UserWarning: Parsing dates in DD/MM/YYYY format when dayfirst=False (the default) was spe
cified. This may lead to inconsistently parsed dates! Specify a format to ensure consistent parsing.
  salesdata['Order Date'] = pd.to_datetime(salesdata['Order Date'])
```

In [13]:
```
# extract Year from date
salesdata['Year'] = salesdata['Order Date'].dt.year
```

In [29]:
```
colors = ['#34e065', '#b80600', '#a8dadc','#2a9d8f']
explode = (0.05, 0.05, 0.05, 0.05)
fig, ax = plt.subplots(1,2, figsize = (8,5))

#Sales percentage per year
df_sales = salesdata.groupby('Year')['Sales'].sum().sort_values(ascending = False).reset_index()
# Pie Chart
ax[0].pie(df_sales['Sales'],
       colors=colors,
       labels=df_sales['Year']
       ,autopct='%1.1f%%',
       pctdistance=0.85,
       explode=explode )

centre_circle = plt.Circle((0, 0), 0.70, fc='white')
fig = plt.gcf()


fig.gca().add_artist(centre_circle)
ax[0].set_title('Sales Percentage per Year', fontsize = 15)

# PROFIT PERCENTAGE PER YEAR
df_profit = salesdata.groupby('Year')['Profit'].sum().sort_values(ascending = False).reset_index()

ax[1].pie(df_profit['Profit'],
       colors=colors,
       labels=df_profit['Year']
       ,autopct='%1.1f%%',
       pctdistance=0.85,
       explode=explode )

centre_circle = plt.Circle((0, 0), 0.70, fc='white')
fig = plt.gcf()


fig.gca().add_artist(centre_circle)
ax[1].set_title('Profit Percentage per Year', fontsize = 15)

fig.tight_layout()
```



In [15]:
```
# wich product has more sales
df_products = salesdata.groupby('Product Name')['Sales'].sum().sort_values(ascending = False).reset_index().head(10)
sns.barplot(x ='Sales', y ='Product Name', palette = 'brg_r', data = df_products)
plt.title('Top 10 Products have more Sales')
plt.xlabel('Number of Sales')
plt.show()
```



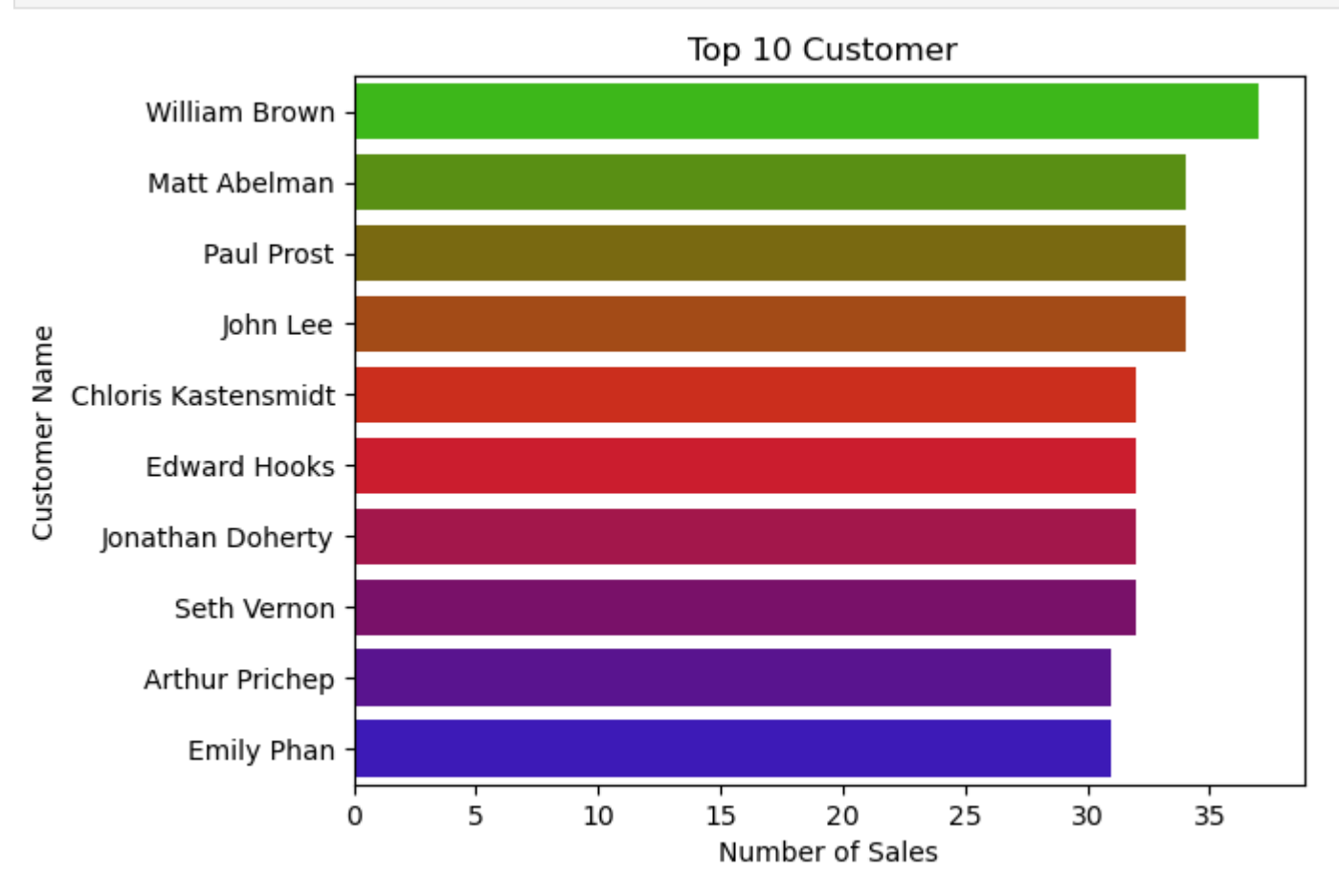
In [28]:
```
df_state = salesdata.groupby('State')['Sales'].sum().sort_values(ascending = False).reset_index()
df_state.head()
```

Out[28]:

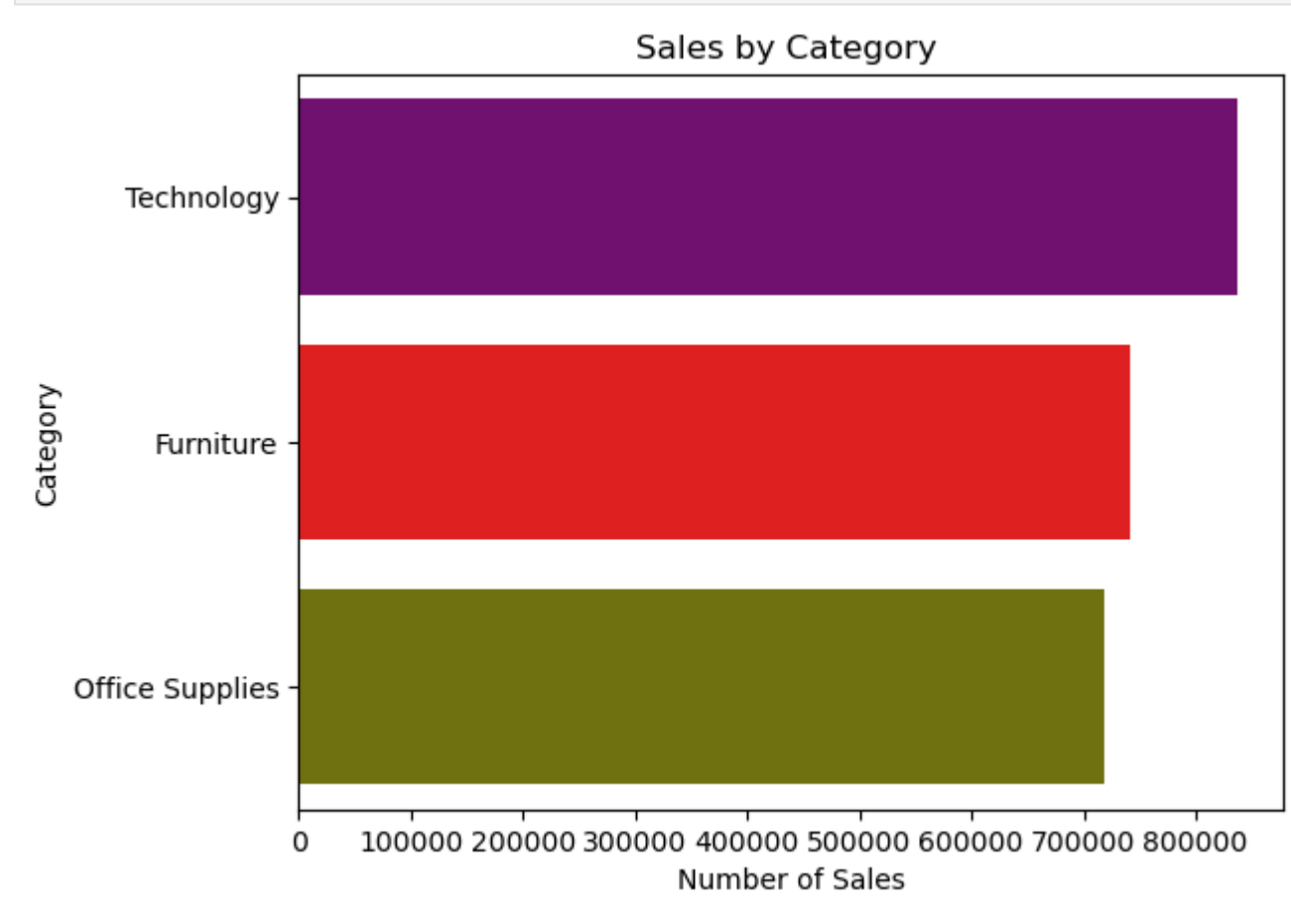
	State	Sales
0	California	457687.68
1	New York	310876.20
2	Texas	170187.98
3	Washington	138641.29
4	Pennsylvania	116512.02

In [27]:
```
# sales by state
# read Geo file
states = pd.read_json('us-states.json').to_json()

# plot the choropleth map
fig = px.choropleth(df_state,
                    geojson= 'https://raw.githubusercontent.com/plotly/datasets/master/geojson-counties-fips.json',
                    locations="State",
                    locationmode="USA-states",
                    color="Sales",
                    color_continuous_scale="Viridis",
                    scope="usa",
                    hover_data={"Sales": ":.2f"})
# fig.update_geos(fitbounds="locations", visible=False)
# fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.update_layout(title='Sales by state')
fig.show()
```

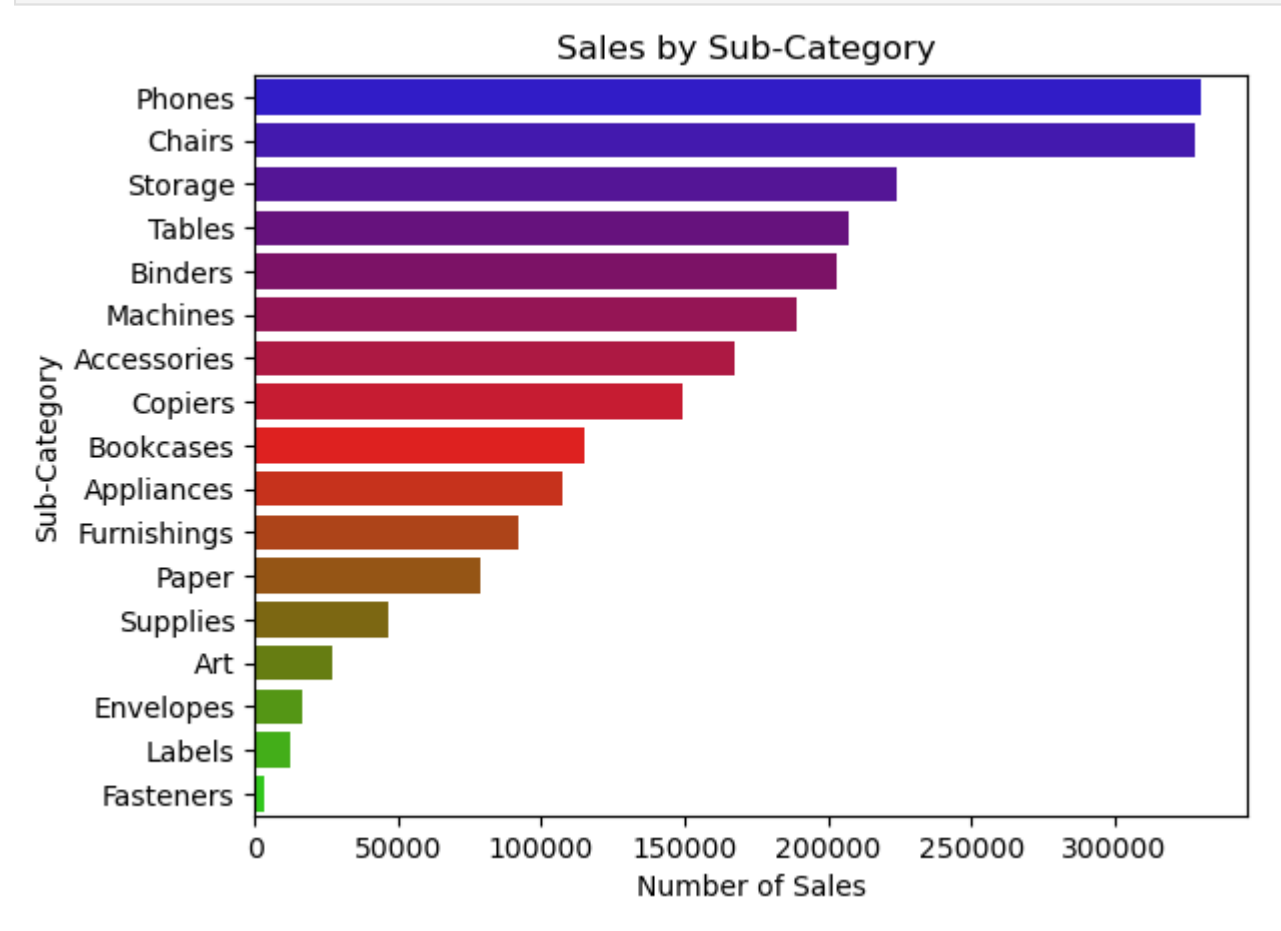
In [25]:
```
# Top 10 Customers
df_top10_customers = salesdata['Customer Name'].value_counts().sort_values(ascending = False).reset_index().head(10)
df_top10_customers.columns = ['Customer Name', 'Sales']

sns.barplot(x ='Sales', y ='Customer Name', palette = 'brg_r', data = df_top10_customers)
plt.title('Top 10 Customers ')
plt.xlabel('Number of Sales')
plt.show()
```



In [41]:
```
# sales by category
df_category = salesdata.groupby('Category')['Sales'].sum().sort_values(ascending = False).reset_index().head(10)
sns.barplot(x ='Sales', y ='Category', palette = 'brg', data = df_category)
plt.title('Sales by Category ')
plt.xlabel('Number of Sales')
plt.show()
```



In [28]:
```
# sales by sub-category
df_subcategory = salesdata.groupby('Sub-Category')['Sales'].sum().sort_values(ascending = False).reset_index()
sns.barplot(x ='Sales', y ='Sub-Category', palette = 'brg', data = df_subcategory)
plt.title('Sales by Sub-Category ')
plt.xlabel('Number of Sales')
plt.show()
```



In [46]:
```
# let's analyz data if there is any relashioship beween sales & profit & Quantity
sns.scatterplot(x = 'Quantity', y = 'Sales', data = salesdata)
plt.title('Quantity vs Sales', fontsize = 16)
plt.show()
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_10284\1739193900.py:2: UserWarning: Ignoring `palette` because no `hue` variable has been assigned.
  sns.scatterplot(x = 'Quantity', y = 'Sales', palette = 'brg', data = salesdata)
```