#Author : Harsha Kumavat

GRIP @The Sparks Foundation Internship - JAN 2022

Data Science & Business Analytics Intern

Task 3 : Exploratory Data Analysis - Retail.

Perform 'Exploratory Data Analysis' on dataset 'SampleSuperstore'.As a business manager, try to find out the weak areas where you can work to make more profit. What all business problems you can derive by exploring the data?

Importing Libraries

In [1]:
```
#Import the necessary libraries
import pandas as pd
import plotly.express as px
from plotly.subplots import make_subplots
import plotly.graph_objects as go
import warnings
warnings.filterwarnings('ignore')
```

Importing the dataset

In [2]:
```
# Load the dataset
df = pd.read_csv('SampleSuperstore (1).csv')
```

In [3]:
```
df.head()  #view first 5 rows of the dataset
```

Out[3]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00	41.9136
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.00	219.5820
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	0.00	6.8714
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.45	-383.0310
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	0.20	2.5164

In [4]:
```
df.tail()    #view last 5 rows of the dataset
```

Out[4]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
9989	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furnishings	25.248	3	0.2	4.1028
9990	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furnishings	91.960	2	0.0	15.6332
9991	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Technology	Phones	258.576	2	0.2	19.3932
9992	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Office Supplies	Paper	29.600	4	0.0	13.3200
9993	Second Class	Consumer	United States	Westminster	California	92683	West	Office Supplies	Appliances	243.160	2	0.0	72.9480

Exploratory Data analysis

In [5]:
```
df.shape    #returns the no of rows and columns
```

Out[5]:
```
(9994, 13)
```

In [6]:
```
df.info()    #Basic summary about the data
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Ship Mode     9994 non-null   object 
 1   Segment       9994 non-null   object 
 2   Country       9994 non-null   object 
 3   City          9994 non-null   object 
 4   State         9994 non-null   object 
 5   Postal Code   9994 non-null   int64  
 6   Region        9994 non-null   object 
 7   Category      9994 non-null   object 
 8   Sub-Category  9994 non-null   object 
 9   Sales         9994 non-null   float64
 10  Quantity      9994 non-null   int64  
 11  Discount      9994 non-null   float64
 12  Profit        9994 non-null   float64
dtypes: float64(3), int64(2), object(8)
memory usage: 1015.1+ KB
```

In [7]:
```
df.isnull().sum()  #checking whether any null values are present
```

Out[7]:
```
Ship Mode       0
Segment         0
Country         0
City            0
State           0
Postal Code     0
Region          0
Category        0
Sub-Category    0
Sales           0
Quantity        0
Discount        0
Profit          0
dtype: int64
```

In [8]:
```
df.duplicated().sum()
```

Out[8]:
```
17
```

In [9]:
```
df.drop_duplicates(inplace=True)
```

In [10]:
```
df.columns
```

Out[10]:
```
Index(['Ship Mode', 'Segment', 'Country', 'City', 'State', 'Postal Code',
       'Region', 'Category', 'Sub-Category', 'Sales', 'Quantity', 'Discount',
       'Profit'],
      dtype='object')
```

In [11]:

```
df.nunique() #gives the count of unique values present in the particular column
```

Out[11]:

```
Ship Mode          4
Segment            3
Country            1
City             531
State             49
Postal Code      631
Region             4
Category           3
Sub-Category      17
Sales           5825
Quantity          14
Discount          12
Profit          7287
dtype: int64
```

In [12]:

```
df.drop(columns='Postal Code',axis=1,inplace=True)
```

In [13]:

```
df.describe() #Statistical summary of data
```

Out[13]:

	Sales	Quantity	Discount	Profit
count	9977.000000	9977.000000	9977.000000	9977.00000
mean	230.148902	3.790719	0.156278	28.69013
std	623.721409	2.226657	0.206455	234.45784
min	0.444000	1.000000	0.000000	-6599.97800
25%	17.300000	2.000000	0.000000	1.72620
50%	54.816000	3.000000	0.200000	8.67100
75%	209.970000	5.000000	0.200000	29.37200
max	22638.480000	14.000000	0.800000	8399.97600

In [14]:

```
df['Ship Mode'].value_counts().to_frame()
```

Out[14]:

	Ship Mode
Standard Class	5955
Second Class	1943
First Class	1537
Same Day	542

In [15]:

```
df['Segment'].value_counts().to_frame()
```

Out[15]:

	Segment
Consumer	5183
Corporate	3015
Home Office	1779

In [16]:

```
df['Country'].value_counts().to_frame()
```

Out[16]:

	Country
United States	9977

In [17]:
```
df['Region'].value_counts().to_frame()
```

Out[17]:
```
         Region
West       3193
East       2845
Central    2319
South      1620
```

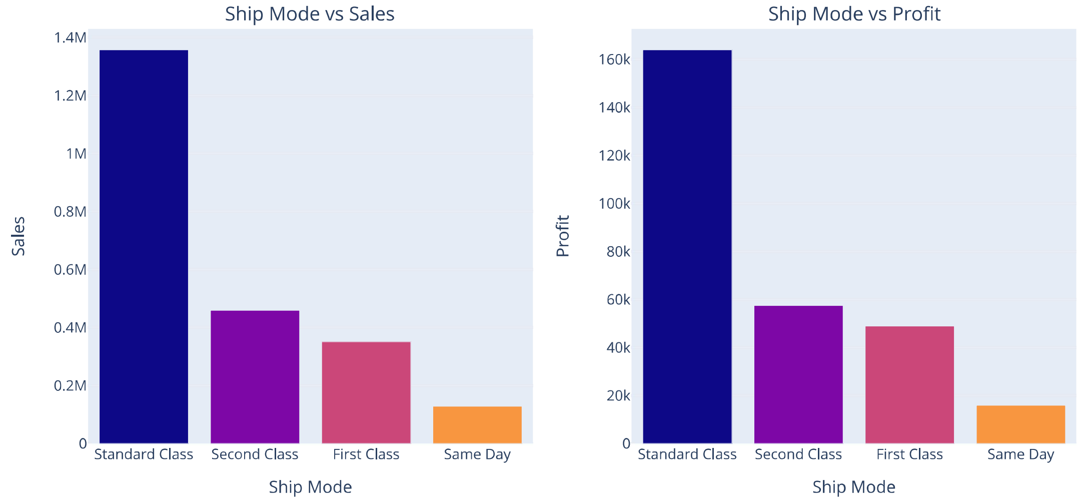
In [18]:
```
a = pd.DataFrame(df.groupby('Ship Mode')['Sales'].sum().sort_values(ascending=False))
a.reset_index(inplace=True)
a.columns=['Ship Mode','Sales']

b = pd.DataFrame(df.groupby('Ship Mode')['Profit'].sum().sort_values(ascending=False))
b.reset_index(inplace=True)
b.columns=['Ship Mode','Profit']

fig = make_subplots(rows=1,cols=2,subplot_titles=("Ship Mode vs Sales","Ship Mode vs Profit", ))
fig.add_trace(go.Bar(x=a['Ship Mode'], y=a['Sales'],marker=dict(color=[1,2,3,4,5])),1, 1)
fig.add_trace(go.Bar(x=b['Ship Mode'], y=b['Profit'],marker=dict(color=[1,2,3,4,5])),1, 2)

fig.update_xaxes(title_text="Ship Mode", row=1, col=1)
fig.update_xaxes(title_text="Ship Mode", row=1, col=2)
fig.update_yaxes(title_text="Sales", row=1, col=1)
fig.update_yaxes(title_text="Profit",row=1, col=2)

fig.update_layout(showlegend=False)
fig.show()
```

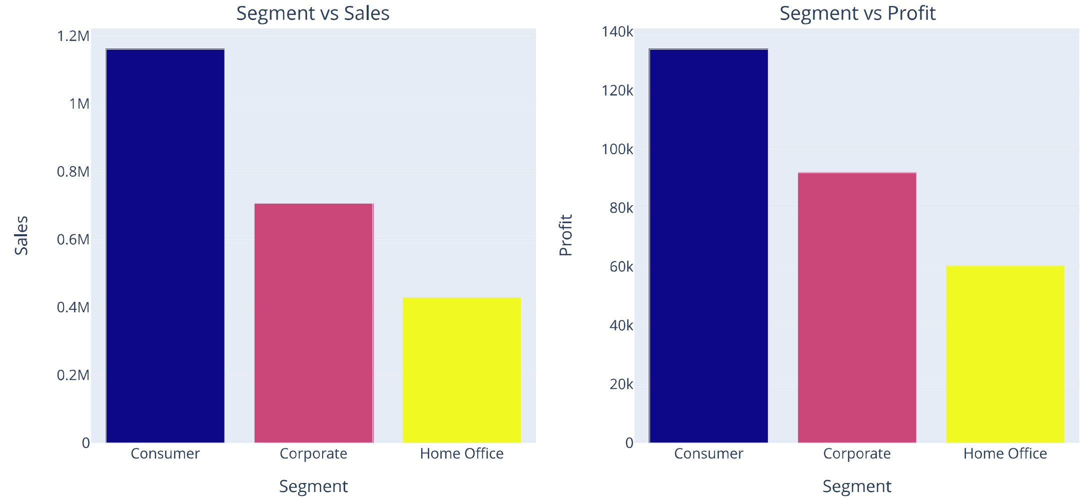


```
In [19]: a = pd.DataFrame(df.groupby('Segment')['Profit'].sum().sort_values(ascending=False))
         a.reset_index(inplace=True)
         a.columns=['Segment','Profit']

         b = pd.DataFrame(df.groupby('Segment')['Sales'].sum().sort_values(ascending=False))
         b.reset_index(inplace=True)
         b.columns=['Segment','Sales']

         fig = make_subplots(rows=1,cols=2,subplot_titles=("Segment vs Sales","Segment vs Profit"))
         fig.add_trace(go.Bar(x=b['Segment'],y=b['Sales'],marker=dict(color=[1,2,3])),1, 1)
         fig.add_trace(go.Bar(x=a['Segment'], y=a['Profit'],marker=dict(color=[1,2,3])),1, 2)
         fig.update_layout(showlegend=False)

         fig.update_xaxes(title_text="Segment", row=1, col=1)
         fig.update_xaxes(title_text="Segment", row=1, col=2)
         fig.update_yaxes(title_text="Sales", row=1, col=1)
         fig.update_yaxes(title_text="Profit",row=1, col=2)
         fig.update_layout(showlegend=False)
         fig.show()
```



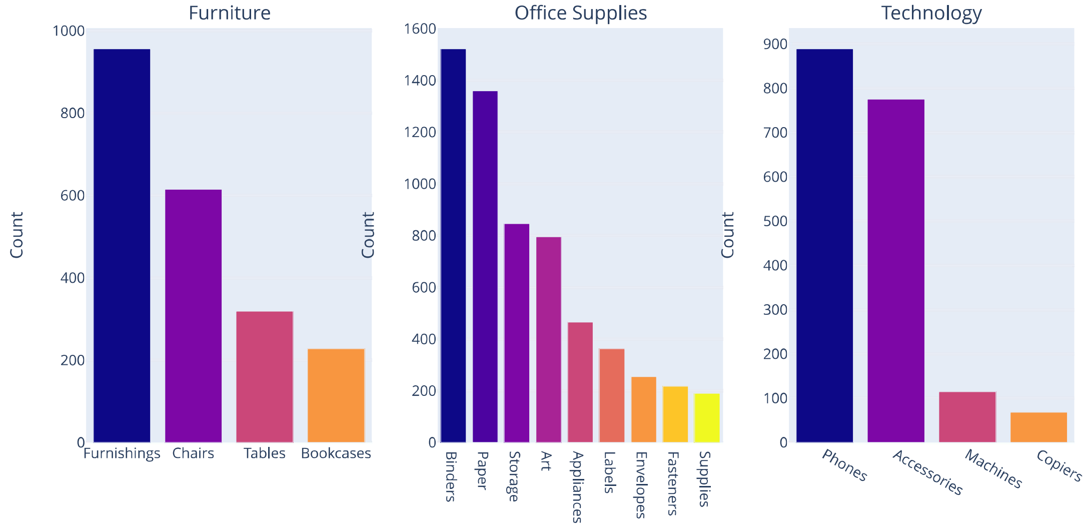
In [20]:
```
df.groupby('Category')['Sub-Category'].value_counts().to_frame()
```

Out[20]:
```
                                Sub-Category
      Category   Sub-Category
     Furniture    Furnishings            956
                       Chairs            615
                       Tables            319
                    Bookcases            228
Office Supplies       Binders           1522
                        Paper           1359
                      Storage            846
                          Art            795
                   Appliances            466
                       Labels            363
                    Envelopes            254
                    Fasteners            217
                     Supplies            190
    Technology         Phones            889
                  Accessories            775
                     Machines            115
                      Copiers             68
```

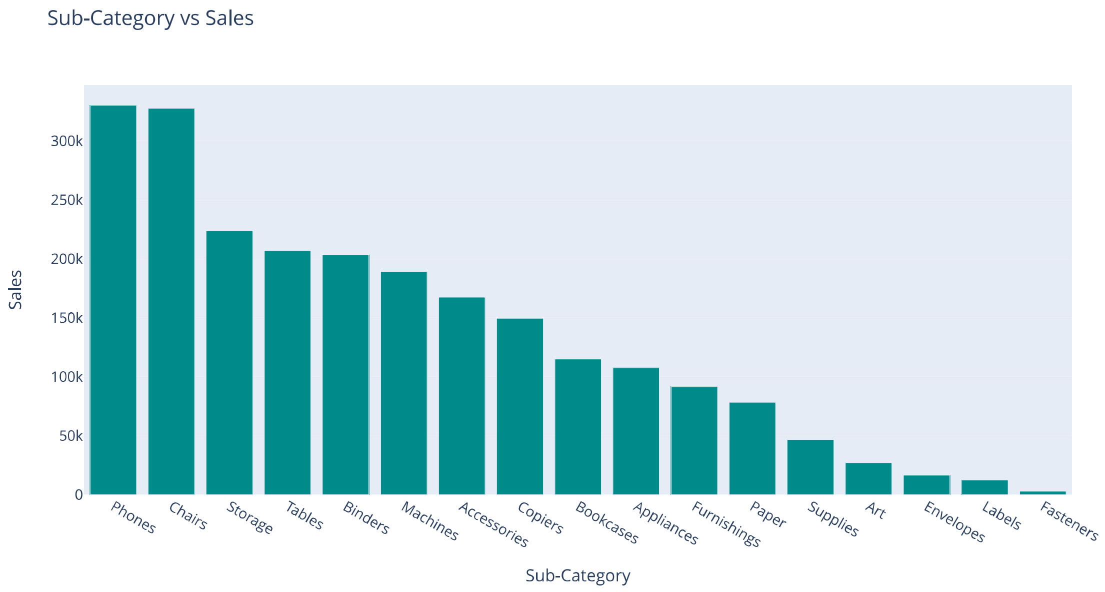
In [21]:
```
Furniture = pd.DataFrame(df[df['Category'] == 'Furniture']['Sub-Category'].value_counts())
Furniture.reset_index(inplace=True)
Furniture.columns = ['Furniture','Count']
Office_Supplies = pd.DataFrame(df[df['Category'] == 'Office Supplies']['Sub-Category'].value_counts())
Office_Supplies.reset_index(inplace=True)
Office_Supplies.columns = ['Office_Supplies','Count']
Technology = pd.DataFrame(df[df['Category'] == 'Technology']['Sub-Category'].value_counts())
Technology.reset_index(inplace=True)
Technology.columns = ['Technology','Count']
```

```
In [22]: fig = make_subplots(rows=1,cols=3,subplot_titles=("Furniture","Office Supplies", "Technology"))
         fig.add_trace(go.Bar(x=Furniture['Furniture'], y=Furniture['Count'],marker=dict(color=[1,2,3,4,5])),1, 1)
         fig.add_trace(go.Bar(x=Office_Supplies['Office_Supplies'], y=Office_Supplies['Count'],marker=dict(color=[1,2,3,4,5,6,7,8,9])),1, 2)
         fig.add_trace(go.Bar(x=Technology['Technology'], y=Technology['Count'],marker=dict(color=[1,2,3,4,5])),1, 3)

         fig.update_yaxes(title_text="Count",row=1, col=2)
         fig.update_yaxes(title_text="Count", row=1, col=1)
         fig.update_yaxes(title_text="Count",row=1, col=3)
         fig.update_layout(showlegend=False)
         fig.show()
```



```
In [23]: a = pd.DataFrame(df.groupby('Sub-Category')['Sales'].sum().sort_values(ascending=False))
         a.reset_index(inplace=True)
         a.columns=['Sub-Category','Sales']
         fig = px.bar(a,y=a['Sales'],x=a['Sub-Category'],title='Sub-Category vs Sales',color_discrete_sequence=['DarkCyan'])
         fig.show()
```



In [24]:
```
data = ['Sales','Quantity','Profit','Discount','State','Category','Sub-Category','Segment']
data=df[data]
data=data.sort_values(by='Profit',ascending=False)
data
df1 = pd.pivot_table(data,index=['Category','Sub-Category'])
df1
```

Out[24]:

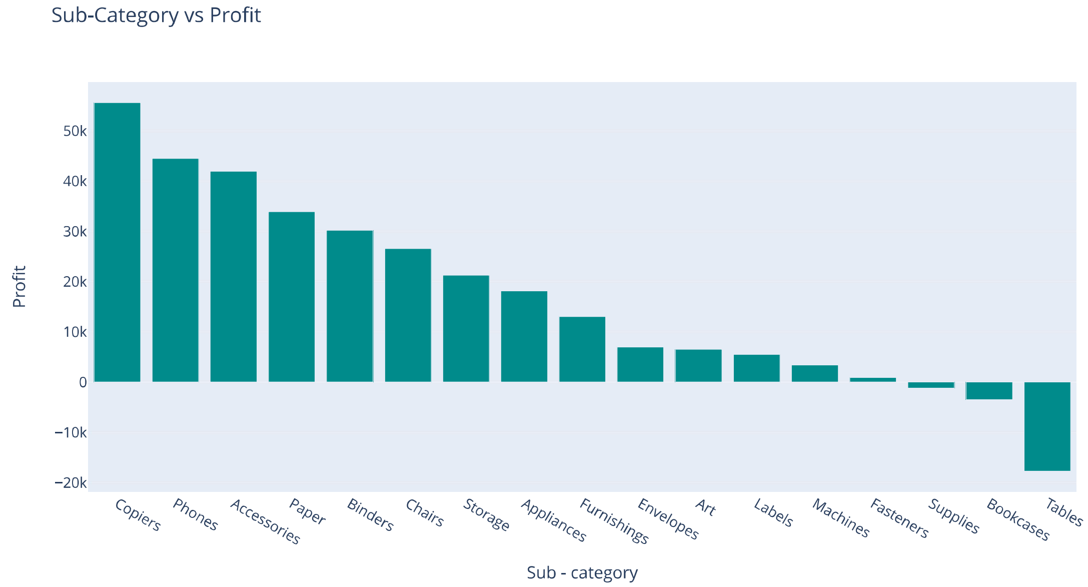
		Discount	Profit	Quantity	Sales
Category	**Sub-Category**				
Furniture	**Bookcases**	0.211140	-15.230509	3.807018	503.859633
	Chairs	0.170244	43.198582	3.822764	532.971969
	Furnishings	0.138494	13.653476	3.723849	95.902745
	Tables	0.261285	-55.565771	3.890282	648.794771
Office Supplies	**Appliances**	0.166524	38.922758	3.710300	230.755710
	Art	0.074969	8.207059	3.768553	34.096896
	Binders	0.372011	19.860710	3.923127	133.645972
	Envelopes	0.080315	27.418019	3.566929	64.867724
	Fasteners	0.082028	4.375660	4.211982	13.936774
	Labels	0.068871	15.224193	3.845730	34.283504
	Paper	0.074908	24.977365	3.785136	57.560075
	Storage	0.074704	25.152277	3.732861	264.590553
	Supplies	0.076842	-6.258418	3.405263	245.650200
Technology	**Accessories**	0.078452	54.111788	3.840000	215.974604
	Copiers	0.161765	817.909190	3.441176	2198.941618
	Machines	0.306087	29.432669	3.826087	1645.553313
	Phones	0.154556	50.073938	3.699663	371.211534

In [25]:
```
data.pivot_table(values='Profit',index='Segment',columns='Discount',aggfunc='median')
```

Out[25]:

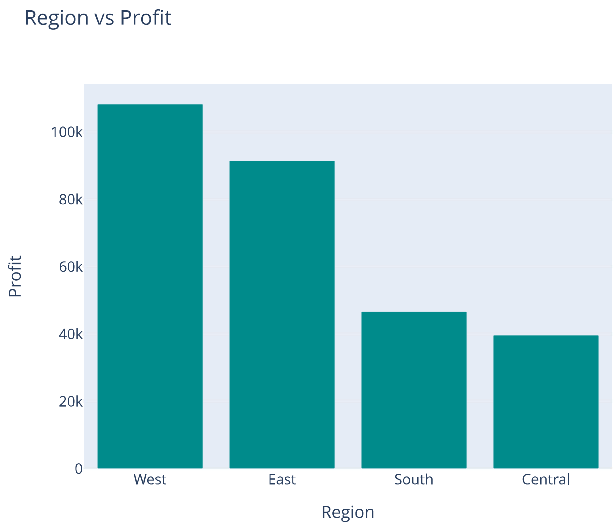
Discount	0.00	0.10	0.15	0.20	0.30	0.32	0.40	0.45	0.50	0.60	0.70	0.80
Segment												
Consumer	16.14600	62.0390	5.12940	6.0433	-26.0562	-30.0980	-47.21360	-135.68145	-175.2600	-14.1323	-8.7276	-13.7175
Corporate	16.35285	46.7908	26.27735	6.7068	-28.2240	-59.0606	-87.27495	-255.58750	-120.5130	-10.4196	-8.9796	-16.7130
Home Office	15.45460	37.2300	16.79860	7.2576	-18.2220	-57.3234	-49.71900	-175.14690	-237.8425	-14.2290	-9.7608	-14.0328

```
In [26]: a = pd.DataFrame(df.groupby('Sub-Category')['Profit'].sum().sort_values(ascending=False))
         a.reset_index(inplace=True)
         a.columns=['Sub - category','Profit']
         fig = px.bar(a,y=a['Profit'],x=a['Sub - category'],title='Sub-Category vs Profit',color_discrete_sequence=['DarkCyan'])
         fig.show()
```

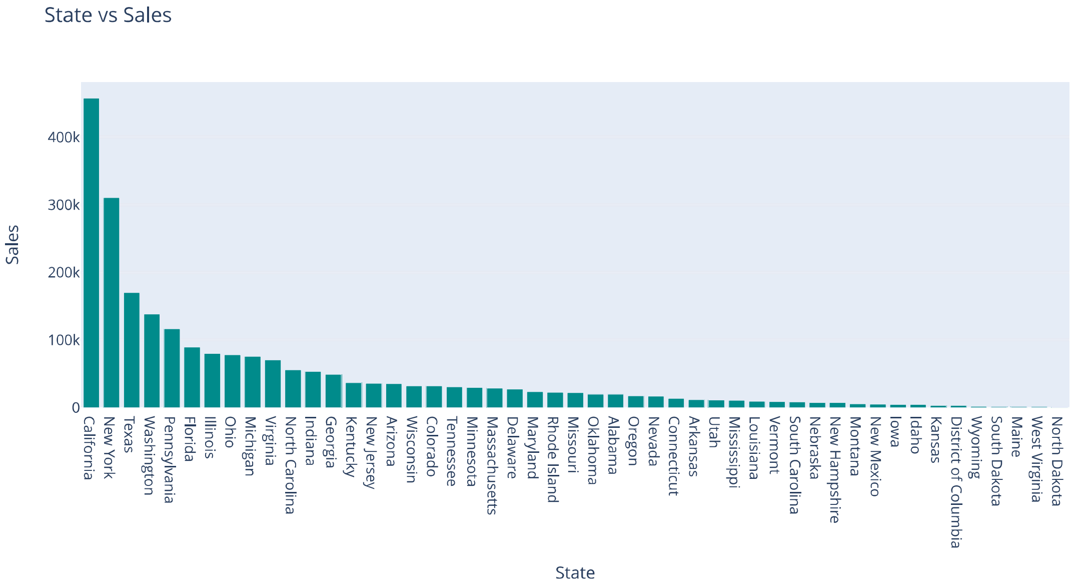


In [27]:

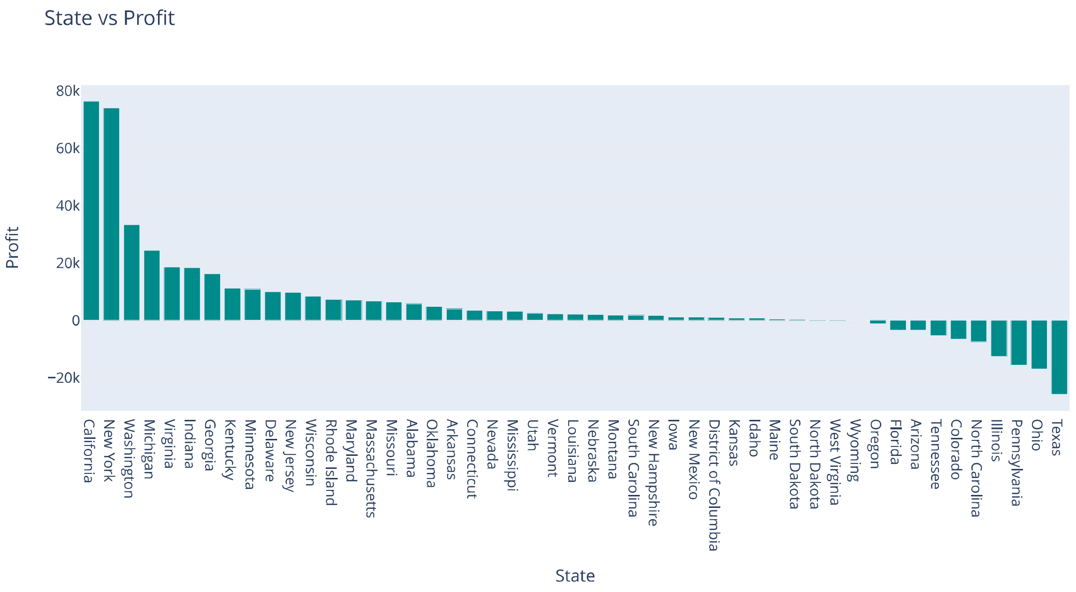
```
a = pd.DataFrame(df.groupby('Region')['Profit'].sum().sort_values(ascending=False))
a.reset_index(inplace=True)
a.columns=['Region','Profit']
fig = px.bar(a,y=a['Profit'],x=a['Region'],title='Region vs Profit',color_discrete_sequence=['DarkCyan'],width=600,height=500)
fig.show()
```



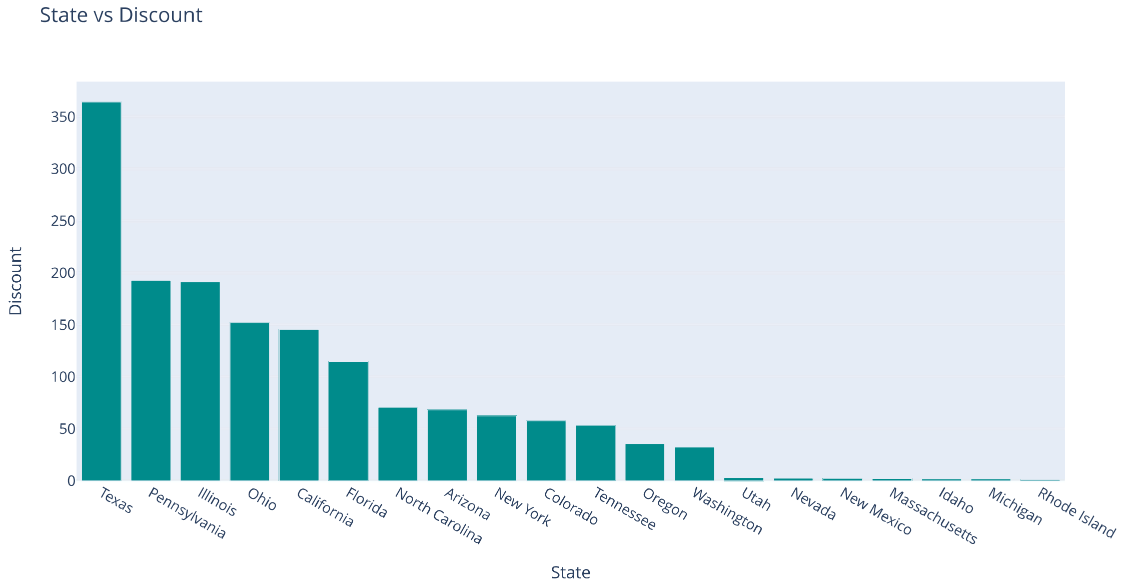
In [28]:
```
a = pd.DataFrame(df.groupby('State')['Sales'].sum().sort_values(ascending=False))
a.reset_index(inplace=True)
a.columns=['State','Sales']
fig = px.bar(a,y=a['Sales'],x=a['State'],title='State vs Sales',color_discrete_sequence=['DarkCyan'])
fig.show()
```



In [29]:
```
a = pd.DataFrame(df.groupby('State')['Profit'].sum().sort_values(ascending=False))
a.reset_index(inplace=True)
a.columns=['State','Profit']
fig = px.bar(a,y=a['Profit'],x=a['State'],title='State vs Profit',color_discrete_sequence=['DarkCyan'])
fig.show()
```



```
In [30]: a = pd.DataFrame(df.groupby('State')['Discount'].sum().sort_values(ascending=False)).head(20)
         a.reset_index(inplace=True)
         a.columns=['State','Discount']
         fig = px.bar(a,y=a['Discount'],x=a['State'],title='State vs Discount',color_discrete_sequence=['darkcyan'])
         fig.show()
```

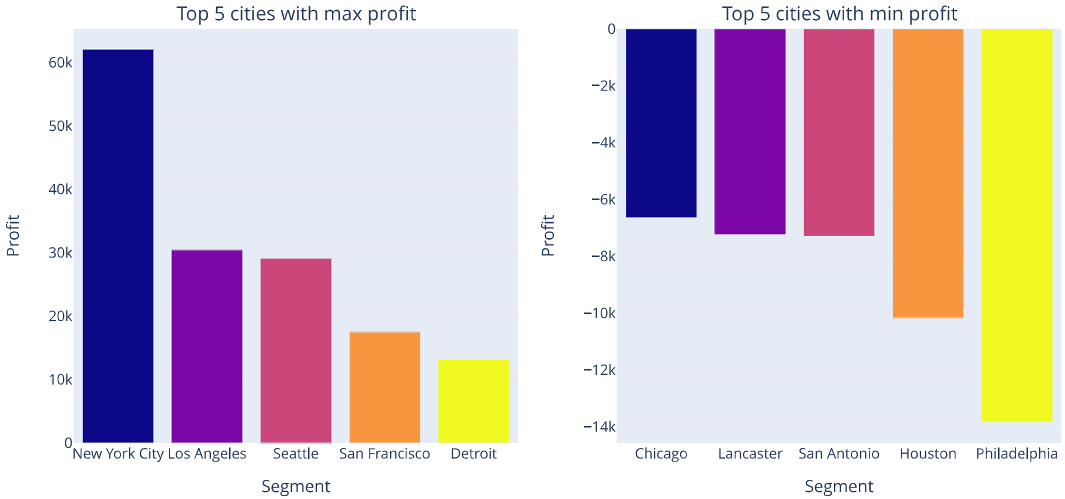


```
In [31]: a = pd.DataFrame(df.groupby('City')['Profit'].sum().sort_values(ascending=False).head(5))
         a.reset_index(inplace=True)
         a.columns=['City','Profit']

         b = pd.DataFrame(df.groupby('City')['Profit'].sum().sort_values(ascending=False).tail(5))
         b.reset_index(inplace=True)
         b.columns=['City','Profit']

         fig = make_subplots(rows=1,cols=2,subplot_titles=("Top 5 cities with max profit","Top 5 cities with min profit"))
         fig.add_trace(go.Bar(x=a['City'],y=a['Profit'],marker=dict(color=[1,2,3,4,5])),1, 1)
         fig.add_trace(go.Bar(x=b['City'], y=b['Profit'],marker=dict(color=[1,2,3,4,5])),1, 2)
         fig.update_layout(showlegend=False)

         fig.update_xaxes(title_text="Segment", row=1, col=1)
         fig.update_xaxes(title_text="Segment", row=1, col=2)
         fig.update_yaxes(title_text="Profit", row=1, col=1)
         fig.update_yaxes(title_text="Profit",row=1, col=2)
         fig.update_layout(showlegend=False)
```



Conclusion

Problem Statement : Find out weak areas where you can work to make profit and what all business problem can be derived by exploring data.

- Standard Class in ShipMode has recorded the highest profit and Same Day has recorded the lowest profit.
- There are 3 segments selling products they are Consumer, Corporate & Home Office where Consumer segment has recorded maximum profit followed by Corporate whereas Home Offices recorded minimum profit.
- In United States the products are sold where West region has recorded maximum profit followed by East and lowest being recorded in Central region.
- Top 5 most sold products Sub-Category wise are Phones, Chairs, Storage, Tables & Binders.
- Top 5 least sold products Sub-Category wise are Fasteners, Labels, Envelopes, Art & Supplies.
- When the discount given on a product is beyond 20% then company is getting a loss instead of gainning profit.
- Maximum profit is gained by Copiers, Phones, Accessories ,Paper, Binders whereas Tables has recorded maximim loss followed by Bookcases & Supplies.Hence discount given on these products can be reduced to increase profit.
- Maximum Sales are from states California, New York & Minimum sales are from North Dakota, West Virginia.
- State California & New Yok has recorded the maximum profit whereas Texas, Ohio, Pennsylvania in these states products has occured loss. So discount given in these states can be reduced to increase profit.
- As maximum sales are in states California, NewYork so sales can be increased in these areas to gain profit and In technology category company is getting benefitted so increase in sales of these category can increase profit.