THE SPARKS FOUNDATION

Batch :September2022

Name : Varsha Reddy

Task 3 -Exploratory Data Analysis

Statement of the Problem : Perform Exploratory Data Analysis on dataset"Sample SuperStore"

Dataset : https://bit.ly/3i4rbWl

Importing Libraries for Data Manipulation

In [1]:
```
import numpy as np
import pandas as pd
```

Importing Libraries for Data Visualization

In [2]:
```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [3]:
```
import plotly.offline as py
import plotly.graph_objs as go
import plotly.express as px
from collections import Counter
import math
import warnings
warnings.filterwarnings('ignore')
```

Importing the Dataset

In [4]:
```
df = pd.read_csv("SampleSuperstore.csv")
```

Processing the Data

In [5]:
```
df.head()
```

Out[5]:

	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 [6]:
```
df.tail()
```

Out[6]:

	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

In [7]:
```
df.info()
```

```
<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 [8]:
```
df.describe()
```

Out[8]:

	Postal Code	Sales	Quantity	Discount	Profit
count	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000
mean	55190.379428	229.858001	3.789574	0.156203	28.656896
std	32063.693350	623.245101	2.225110	0.206452	234.260108
min	1040.000000	0.444000	1.000000	0.000000	-6599.978000
25%	23223.000000	17.280000	2.000000	0.000000	1.728750
50%	56430.500000	54.490000	3.000000	0.200000	8.666500
75%	90008.000000	209.940000	5.000000	0.200000	29.364000
max	99301.000000	22638.480000	14.000000	0.800000	8399.976000

In [9]:
```
df.shape
```

Out[9]:
```
(9994, 13)
```

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()
```

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.isnull().sum()
```

Out[12]:
```
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 [13]:
```
print(df.duplicated().sum())
```

```
17
```

In [14]:
```
df.drop_duplicates()
```

Out[14]:

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9989	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furnishings	25.2480	3	0.20	4.1028
9990	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furnishings	91.9600	2	0.00	15.6332
9991	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Technology	Phones	258.5760	2	0.20	19.3932
9992	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Office Supplies	Paper	29.6000	4	0.00	13.3200
9993	Second Class	Consumer	United States	Westminster	California	92683	West	Office Supplies	Appliances	243.1600	2	0.00	72.9480

9977 rows × 13 columns

In [15]:
```
df.Country.value_counts()[:10]
```

Out[15]:
```
United States    9994
Name: Country, dtype: int64
```

In [16]:
```
df.State.value_counts()[:10]
```

Out[16]:
```
California        2001
New York          1128
Texas              985
Pennsylvania       587
Washington         506
Illinois           492
Ohio               469
Florida            383
Michigan           255
North Carolina     249
Name: State, dtype: int64
```

In [17]:
```
df.City.value_counts()[:10]
```

Out[17]:
```
New York City    915
Los Angeles      747
Philadelphia     537
San Francisco    510
Seattle          428
Houston          377
Chicago          314
Columbus         222
San Diego        170
Springfield      163
Name: City, dtype: int64
```

In [18]:
```
df.Profit.value_counts(normalize=True)
```

Out[18]:
```
0.0000     0.006504
6.2208     0.004303
9.3312     0.003802
5.4432     0.003202
3.6288     0.003202
             ...
83.2508    0.000100
16.1096    0.000100
7.1988     0.000100
1.6510     0.000100
72.9480    0.000100
Name: Profit, Length: 7287, dtype: float64
```

In [19]:
```
df[df["Discount"] == 1].mean()
```

Out[19]:
```
Ship Mode       NaN
Segment         NaN
Country         NaN
City            NaN
State           NaN
Postal Code     NaN
Region          NaN
Category        NaN
Sub-Category    NaN
Sales           NaN
Quantity        NaN
Discount        NaN
Profit          NaN
dtype: object
```

In [20]:
```
df.fillna(0).head()
```

Out[20]:

	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 [21]:
```
print("Country with the most Profit:",df['Country'].value_counts().idxmax())
```

```
Country with the most Profit: United States
```

In [22]:
```
print("City with the most Profit:",df['City'].value_counts().idxmax())
```

```
City with the most Profit: New York City
```

In [23]:
```
print("State with the most Profit:",df['State'].value_counts().idxmax())
```

```
State with the most Profit: California
```

In [24]:
```
pd.set_option('display.max_columns',None)
df.head()
```

Out[24]:

	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 [25]:
```
df.corr()
```

Out[25]:

	Postal Code	Sales	Quantity	Discount	Profit
Postal Code	1.000000	-0.023854	0.012761	0.058443	-0.029961
Sales	-0.023854	1.000000	0.200795	-0.028190	0.479064
Quantity	0.012761	0.200795	1.000000	0.008623	0.066253
Discount	0.058443	-0.028190	0.008623	1.000000	-0.219487
Profit	-0.029961	0.479064	0.066253	-0.219487	1.000000

In [26]:
```
col = ['Postal Code']
df = df.drop(columns = col,axis = 1)
```

In [27]:
```
df.corr()
```

Out[27]:

	Sales	Quantity	Discount	Profit
Sales	1.000000	0.200795	-0.028190	0.479064
Quantity	0.200795	1.000000	0.008623	0.066253
Discount	-0.028190	0.008623	1.000000	-0.219487
Profit	0.479064	0.066253	-0.219487	1.000000

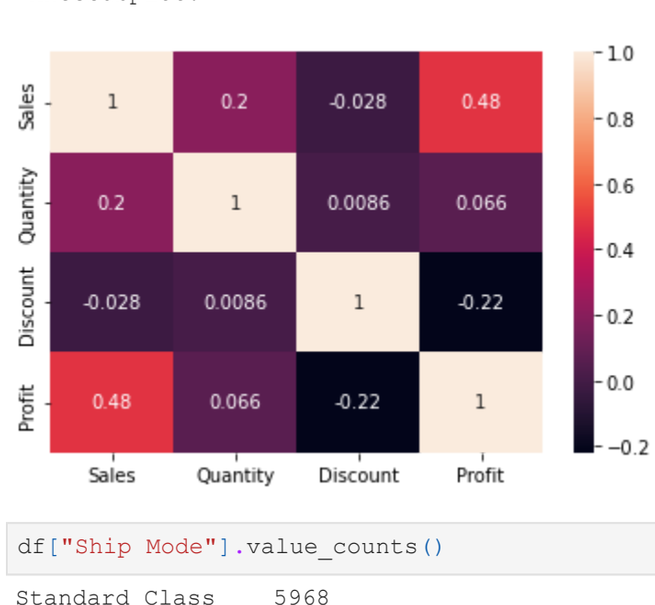
In [28]:
```
correlation = df.corr()
correlation
```

Out[28]:

	Sales	Quantity	Discount	Profit
Sales	1.000000	0.200795	-0.028190	0.479064
Quantity	0.200795	1.000000	0.008623	0.066253
Discount	-0.028190	0.008623	1.000000	-0.219487
Profit	0.479064	0.066253	-0.219487	1.000000

In [29]:
```
sns.heatmap(correlation,annot=True)
```

Out[29]:
```
<AxesSubplot:>
```

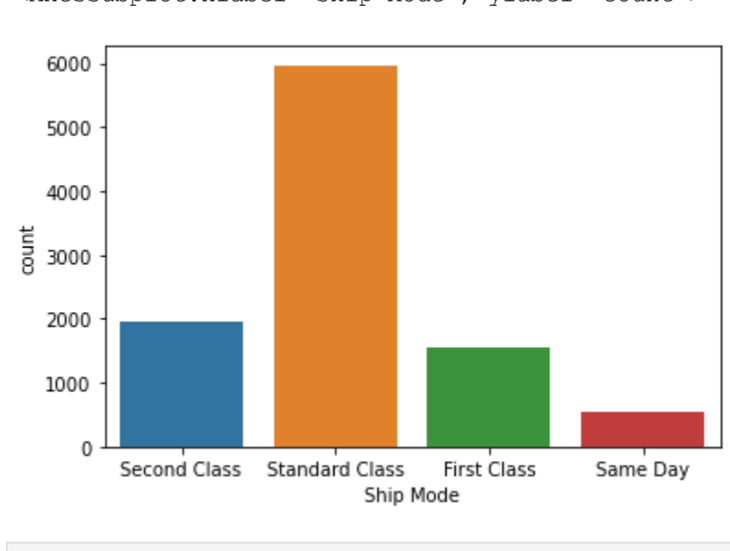


In [30]:
```
df["Ship Mode"].value_counts()
```

Out[30]:
```
Standard Class    5968
Second Class      1945
First Class       1538
Same Day           543
Name: Ship Mode, dtype: int64
```

In [31]:
```
sns.countplot(x=df["Ship Mode"])
```

Out[31]:
```
<AxesSubplot:xlabel='Ship Mode', ylabel='count'>
```

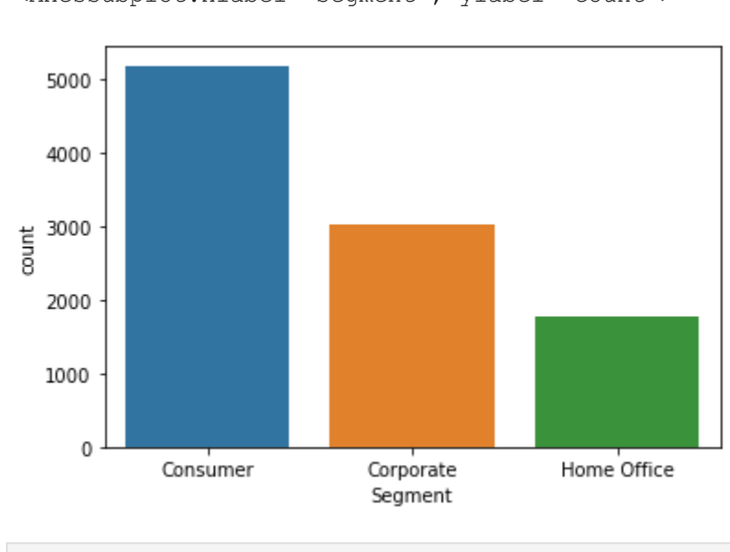


In [32]:
```
df["Segment"].value_counts()
```

Out[32]:
```
Consumer       5191
Corporate      3020
Home Office    1783
Name: Segment, dtype: int64
```

In [33]:
```
sns.countplot(x=df["Segment"])
```

Out[33]:
```
<AxesSubplot:xlabel='Segment', ylabel='count'>
```

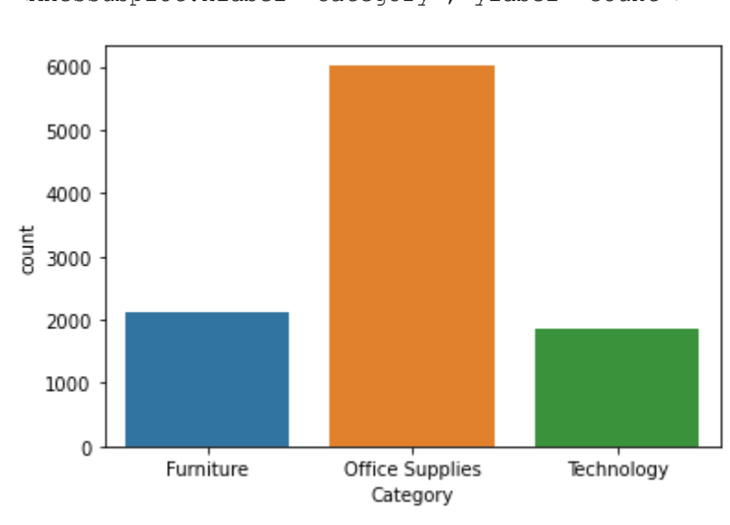


In [34]:
```
df["Category"].value_counts()
```

Out[34]:
```
Office Supplies    6026
Furniture          2121
Technology         1847
Name: Category, dtype: int64
```

In [35]:
```
sns.countplot(x=df["Category"])
```

Out[35]:
```
<AxesSubplot:xlabel='Category', ylabel='count'>
```



Based on the above analysis, Office supplies consists of the most items. Let's now examine its subcategories

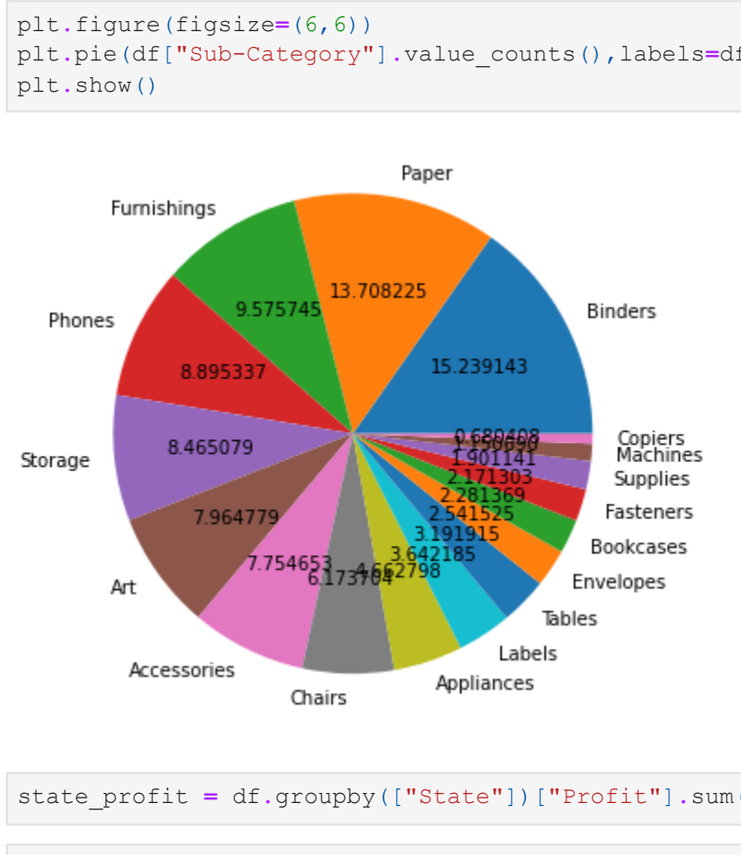
In [36]:
```
df["Sub-Category"].value_counts()
```

Out[36]:
```
Binders        1523
Paper          1370
Furnishings     957
Phones          889
Storage         846
Art             796
Accessories     775
Chairs          617
Appliances      466
Labels          364
Tables          319
Envelopes       254
Bookcases       228
Fasteners       217
Supplies        190
Machines        115
Copiers          68
Name: Sub-Category, dtype: int64
```

In [37]:
```
print("Highest Sub-Category:",df['Sub-Category'].value_counts().idxmax())
```

```
Highest Sub-Category: Binders
```

In [38]:
```
plt.figure(figsize=(6,6))
plt.pie(df["Sub-Category"].value_counts(),labels=df["Sub-Category"].value_counts().index,autopct="%2f")
plt.show()
```



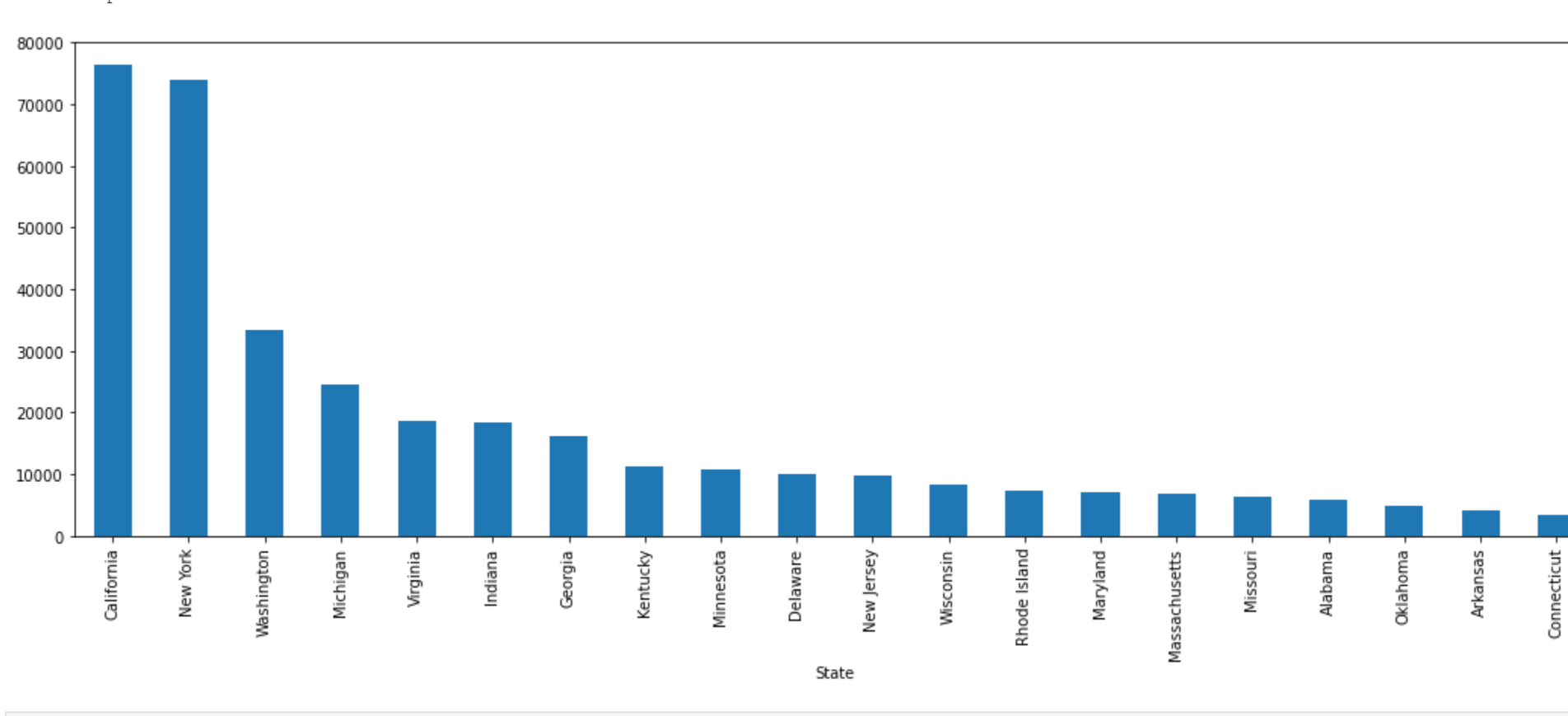
In [39]:
```
state_profit = df.groupby(["State"])["Profit"].sum().nlargest(20)
```

In [40]:
```
state_profit
```

Out[40]:
```
State
California       76381.3871
New York         74038.5486
Washington       33402.6517
Michigan         24463.1876
Virginia         18597.9504
Indiana          18382.9363
Georgia          16250.0433
Kentucky         11199.6966
Minnesota        10823.1874
Delaware          9977.3748
New Jersey        9772.9138
Wisconsin         8401.8004
Rhode Island      7285.6293
Maryland          7031.1788
Massachusetts     6785.5016
Missouri          6436.2105
Alabama           5786.8253
Oklahoma          4853.9560
Arkansas          4008.6871
Connecticut       3511.4918
Name: Profit, dtype: float64
```

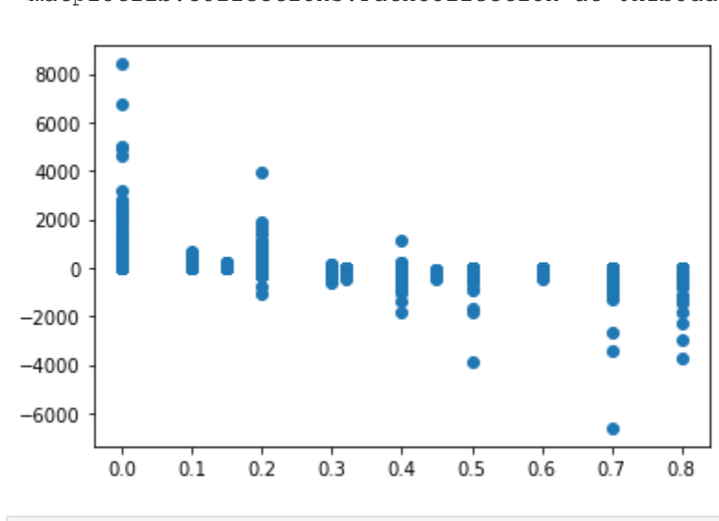
In [41]:
```
plt.figure(figsize=(18,6))
state_profit.plot.bar()
```

Out[41]:
```
<AxesSubplot:xlabel='State'>
```



In [42]:
```
plt.scatter(data = df,x="Discount",y="Profit")
```

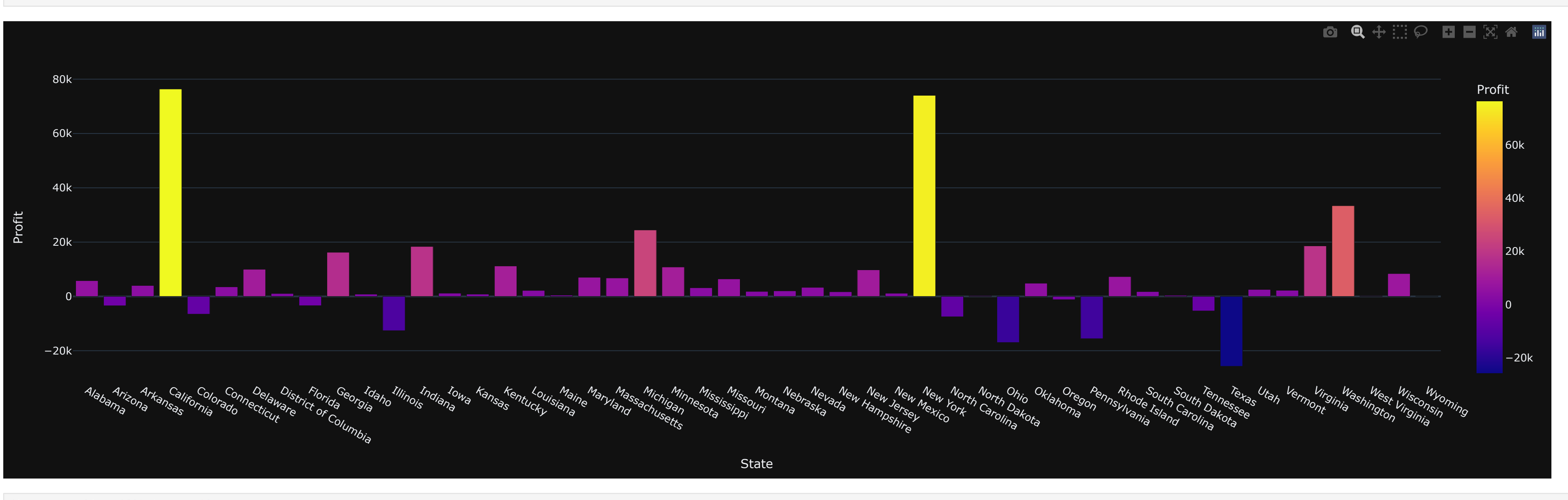
Out[42]:
```
<matplotlib.collections.PathCollection at 0x1b5dd82a850>
```



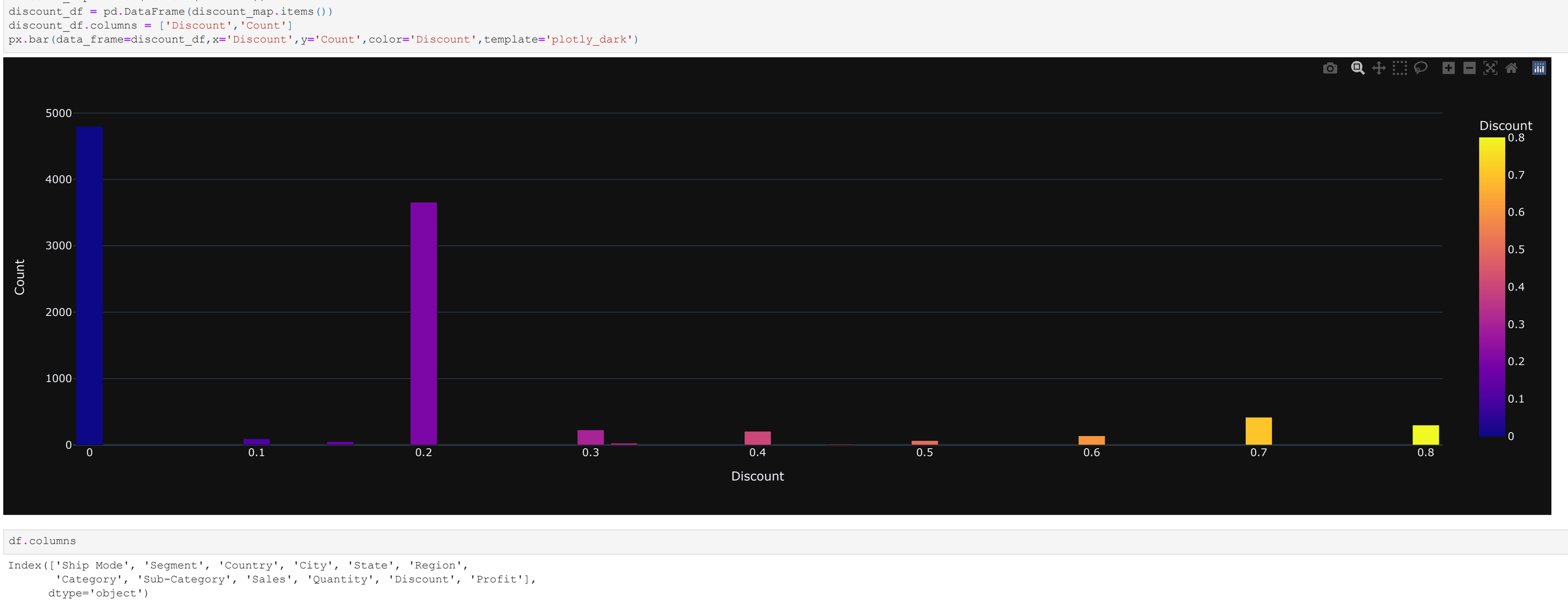
In [43]:
```
df_con = df.select_dtypes(include=[np.number])
```

In [44]:
```
state_profit = df.groupby('State').Profit.sum().to_frame().reset_index()
state_profit .columns = ['State','Profit']
```

In [45]:
```
px.bar(data_frame=state_profit ,x="State",y="Profit",color="Profit",template='plotly_dark')
```



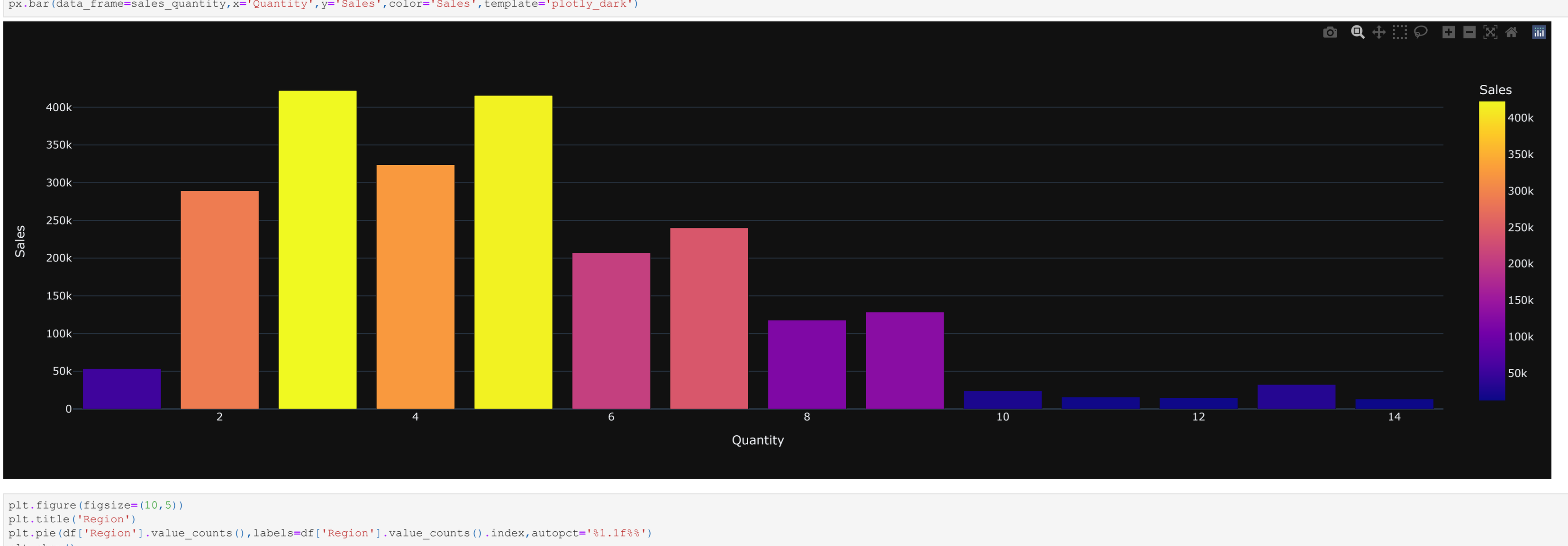
In [46]:
```
discount = list(df['Discount'])
discount_map = dict(Counter(discount))
discount_df = pd.DataFrame(discount_map.items())
discount_df.columns = ['Discount','Count']
px.bar(data_frame=discount_df,x='Discount',y='Count',color='Discount',template='plotly_dark')
```



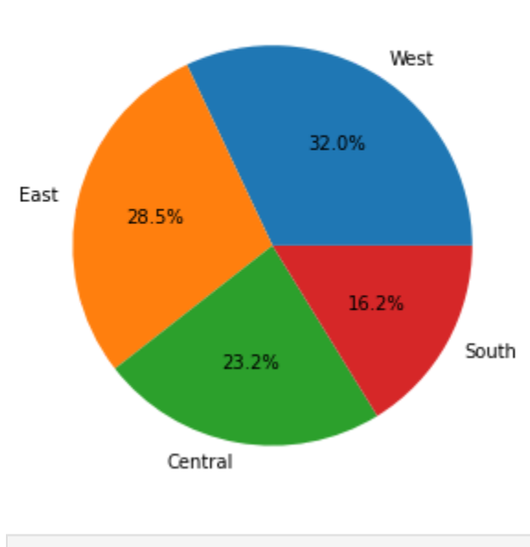
In [47]:
```
df.columns
```

Out[47]:
```
Index(['Ship Mode', 'Segment', 'Country', 'City', 'State', 'Region',
       'Category', 'Sub-Category', 'Sales', 'Quantity', 'Discount', 'Profit'],
      dtype='object')
```

In [48]:
```
sales_quantity= df.groupby('Quantity').Sales.sum().to_frame().reset_index()
sales_quantity.columns=['Quantity','Sales']
px.bar(data_frame=sales_quantity,x='Quantity',y='Sales',color='Sales',template='plotly_dark')
```



In [49]:
```
plt.figure(figsize=(10,5))
plt.title('Region')
plt.pie(df['Region'].value_counts(),labels=df['Region'].value_counts().index,autopct='%1.1f%%')
plt.show()
```



In [52]:
```
plt.style.use('seaborn')
df.plot(kind="scatter",x="Sales",y='Profit',c='Discount',s=20,fontsize=15,colormap='viridis')
plt.xlabel("The interdependency of sales, profit, and discounts",fontsize=15)
plt.ylabel("Total Profits",fontsize=15)
plt.show()
```

In [ ]: