The Sparks Foundation - Data Science & Business Analytics Internship

TASK 3: Perform ‘Exploratory Data Analysis’ on dataset ‘SampleSuperstore’

DONE BY: NOTAM KEDARI

In this task, we will be working on to find the weak areas where we can analyse and work on it to get more profit.

CONTENT - STEPS INVOLVED

STEP 1: IMPORT REQUIRED LIBRARIES

STEP 2: READ AND EXPLORE THE DATASET

STEP 3: PREPROCESSING THE DATA

STEP 4: EXPLORATORY DATA ANALYSIS

OBSERVATIONS FROM THE DATA

STEP 1: IMPORT REQUIRED LIBRARIES

In [1]:
```
# In this step we will import the required libarries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

# To ignore the warnings
import warnings as wg
wg.filterwarnings("ignore")
```

STEP 2: READ AND EXPLORE THE DATASET

In [4]:
```
dset = pd.read_csv("C:\\Users\\NOTAM KEDARI\\Desktop\\SampleSuperstore.csv")
```

In [5]:
```
dset.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

STEP 3: PREPROCESSING THE DATA

In [34]:
```
dset.shape
```

Out[34]:
```
(9994, 12)
```

In [7]:
```
dset.columns
```

Out[7]:
```
Index(['Ship Mode', 'Segment', 'Country', 'City', 'State', 'Postal Code',
       'Region', 'Category', 'Sub-Category', 'Sales', 'Quantity', 'Discount',
       'Profit'],
      dtype='object')
```

In [8]:
```
dset.isnull().sum()
```

Out[8]:
```
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 [10]:
```
dset.describe()
```

Out[10]:

	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]:
```
dset.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 [12]:
```
# checking for duplicate values
dset.duplicated().sum()
```

Out[12]:
```
17
```

In [13]:
```
# dropping the duplicates
dset.drop_duplicates()
dset.head()
```

Out[13]:

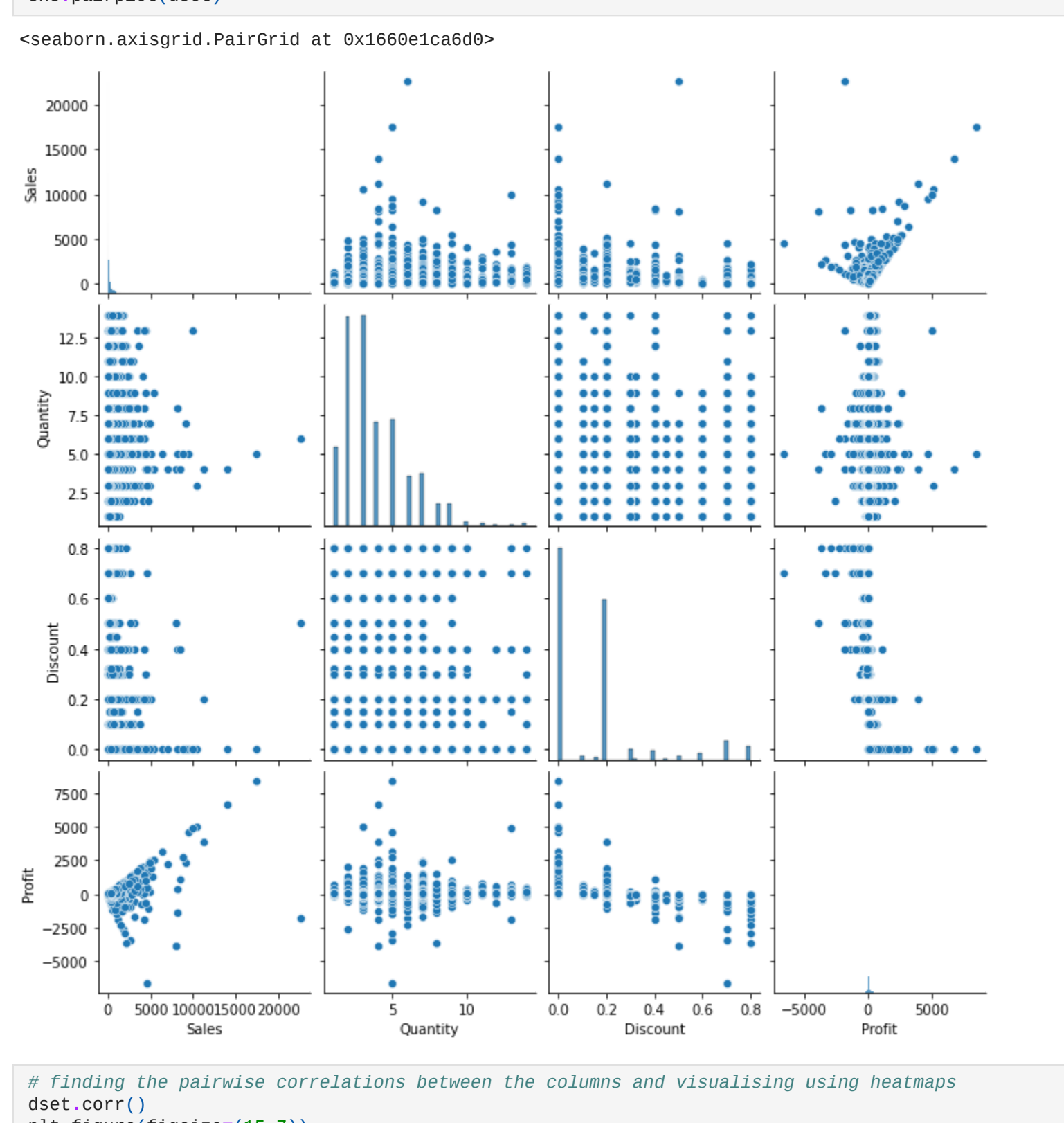
	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00	41.9136
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	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 [14]:
```
#removing the unnecessary columns such as postal code
dset = dataset.drop(['Postal Code'],axis=1)
```

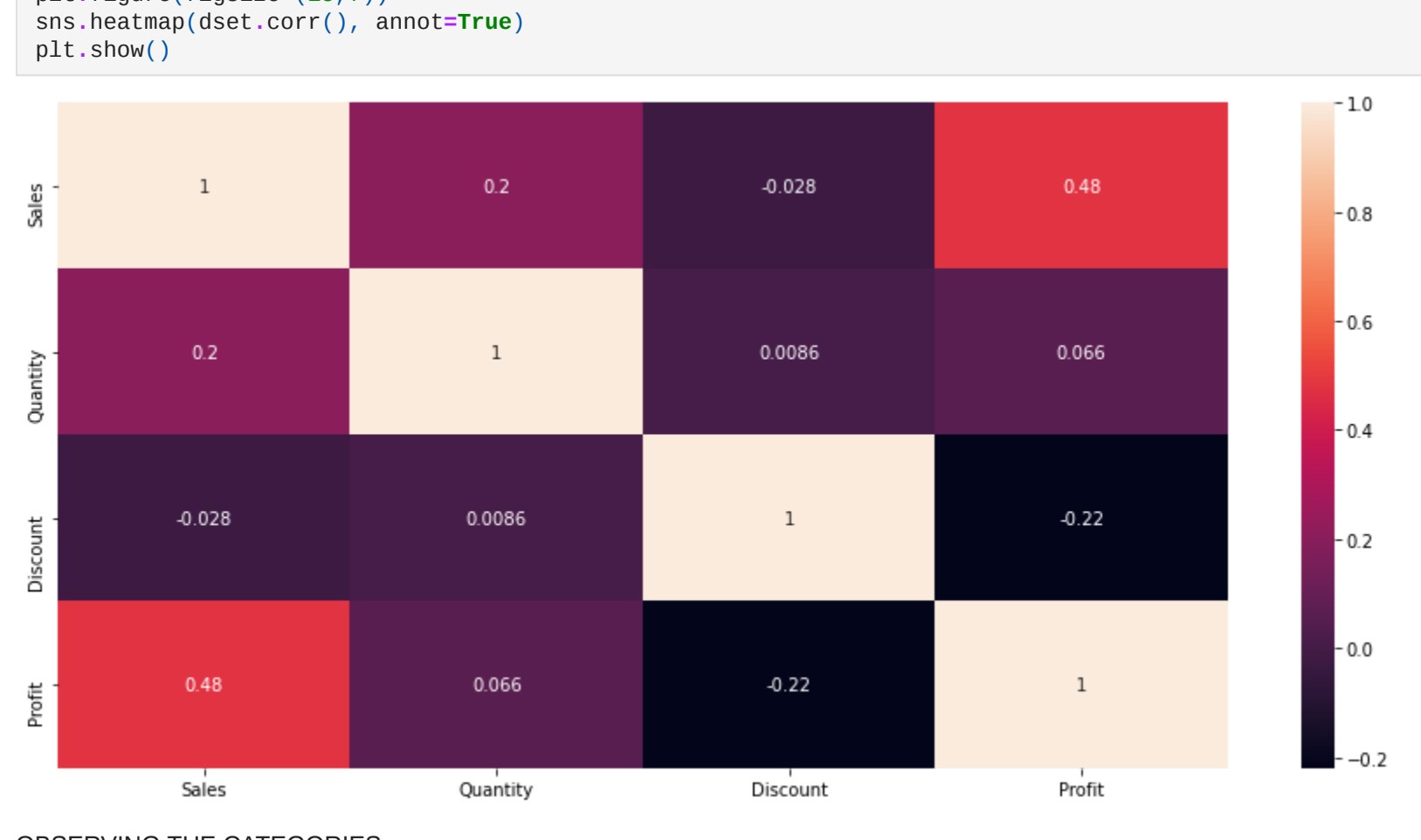
STEP 4: EXPLORATORY DATA ANALYSIS

In [17]:
```
# visualizing the dataset as a whole using the pair plot
import seaborn as sns
sns.pairplot(dset)
```

Out[17]: <seaborn.axisgrid.PairGrid at 0x1660e1ca6d0>

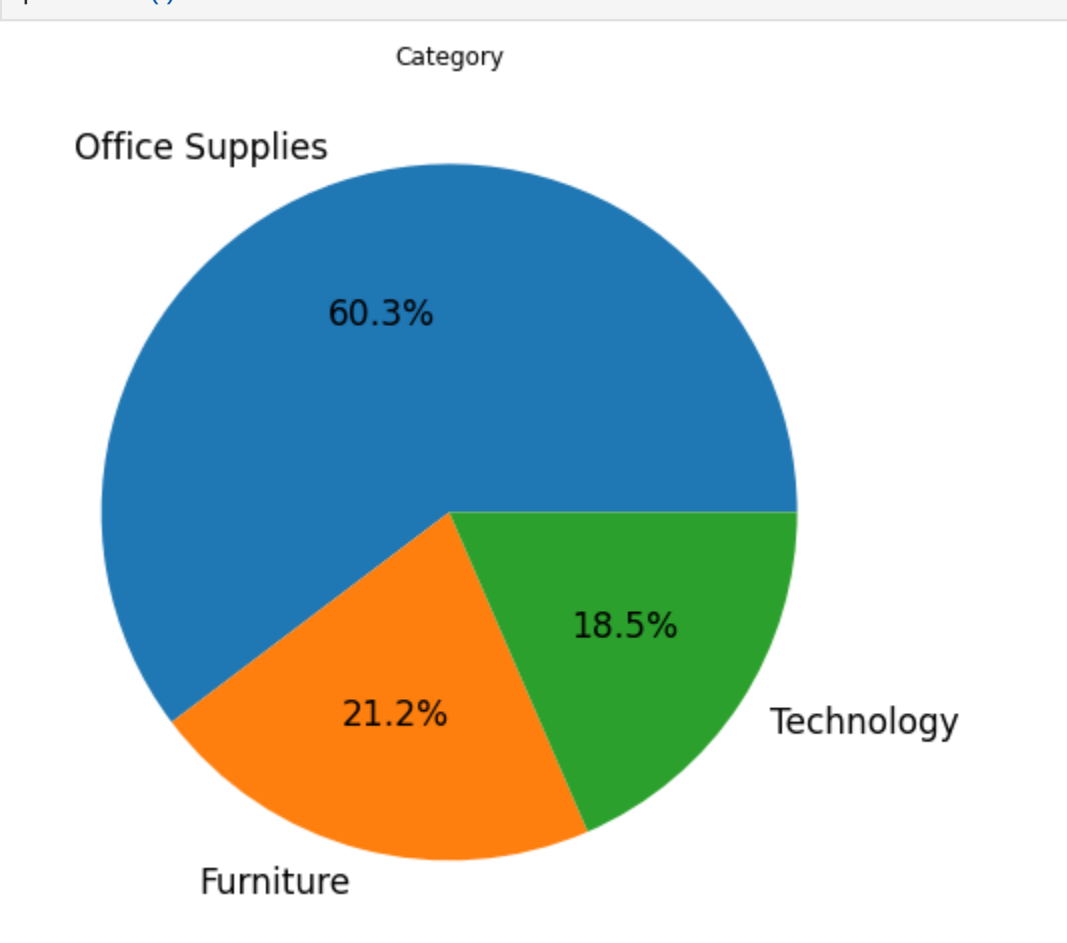


In [22]:
```
# finding the pairwise correlations between the columns and visualising using heatmaps
dset.corr()
plt.figure(figsize=(15,7))
sns.heatmap(dset.corr(), annot=True)
plt.show()
```



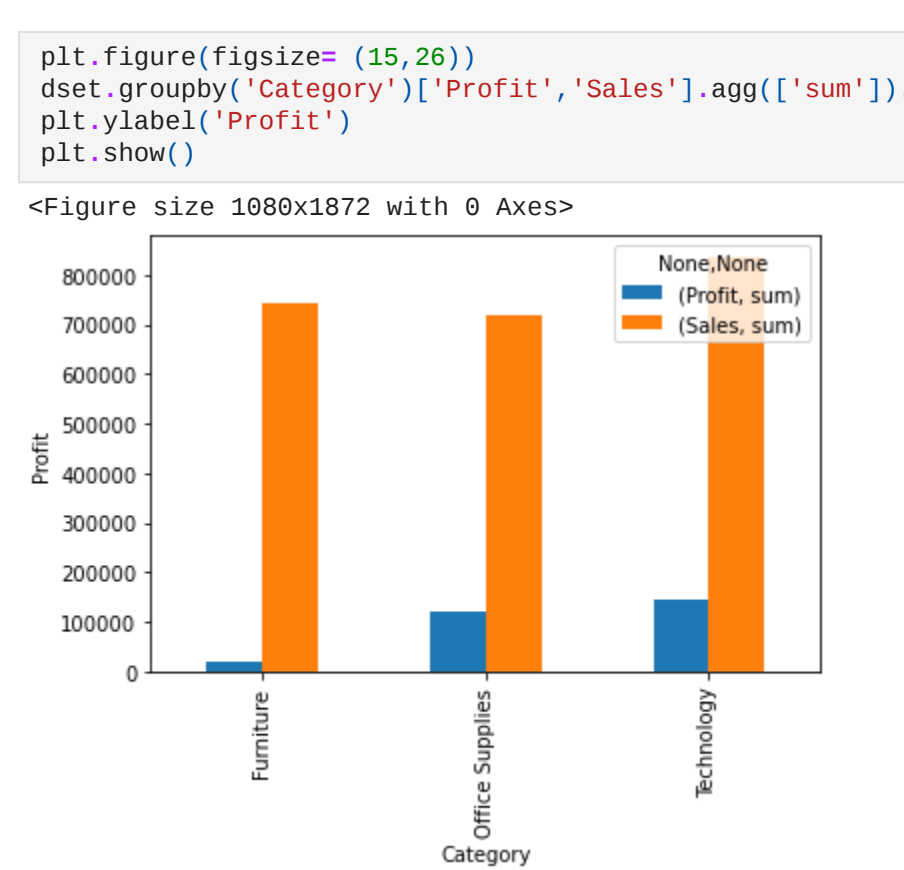
OBSERVING THE CATEGORIES

In [21]:
```
plt.figure(figsize = (8,8))
textprops = {"fontsize":17}
plt.title('Category')
plt.pie(dataset['Category'].value_counts(), labels=dataset['Category'].value_counts().index,autopct='%1.1f%%',textprops = textprops)
plt.show()
```

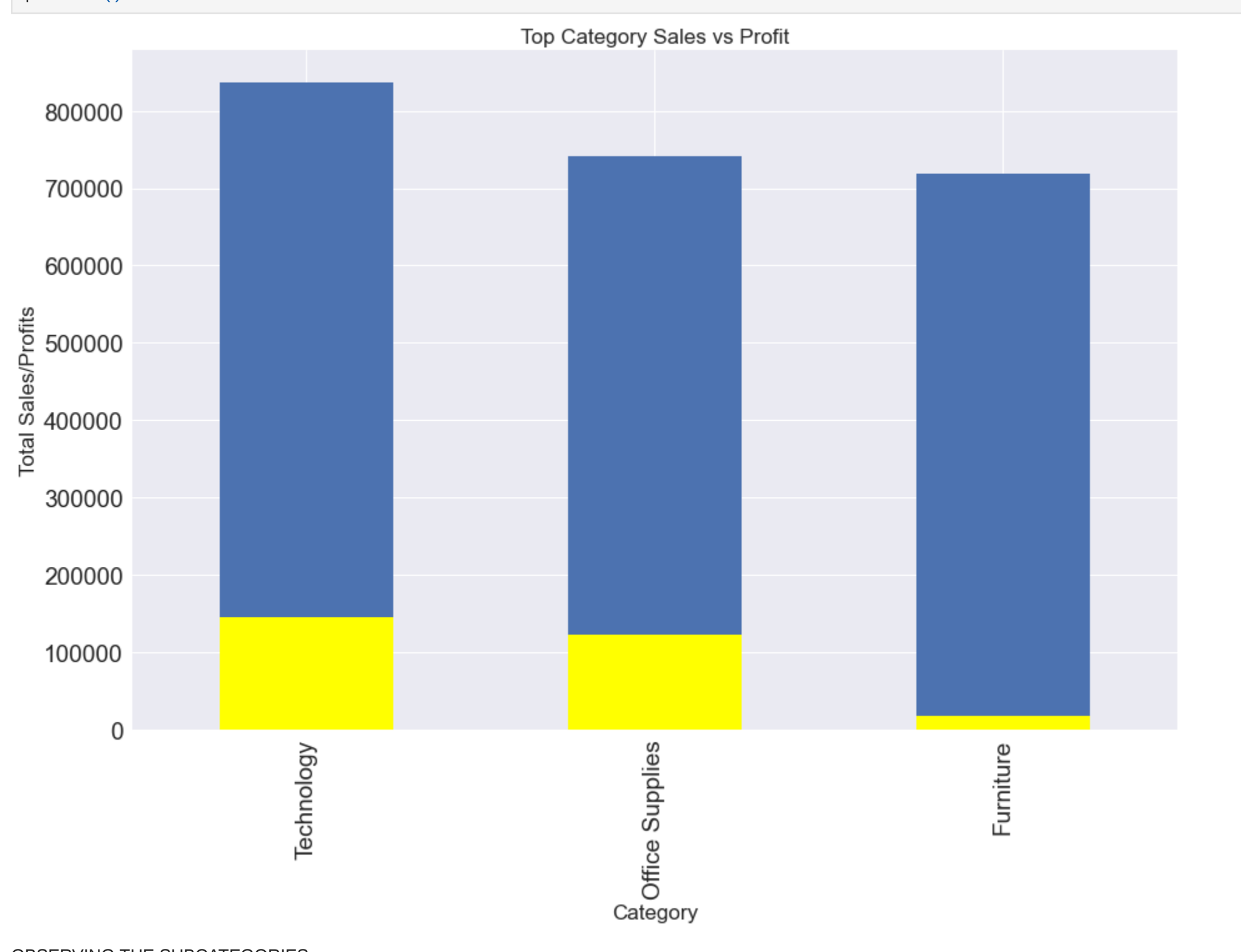


In [23]:
```
plt.figure(figsize= (15,26))
dset.groupby('Category')['Profit','Sales'].agg(['sum']).plot.bar()
plt.ylabel('Profit')
plt.show()
```

<Figure size 1080x1872 with 0 Axes>

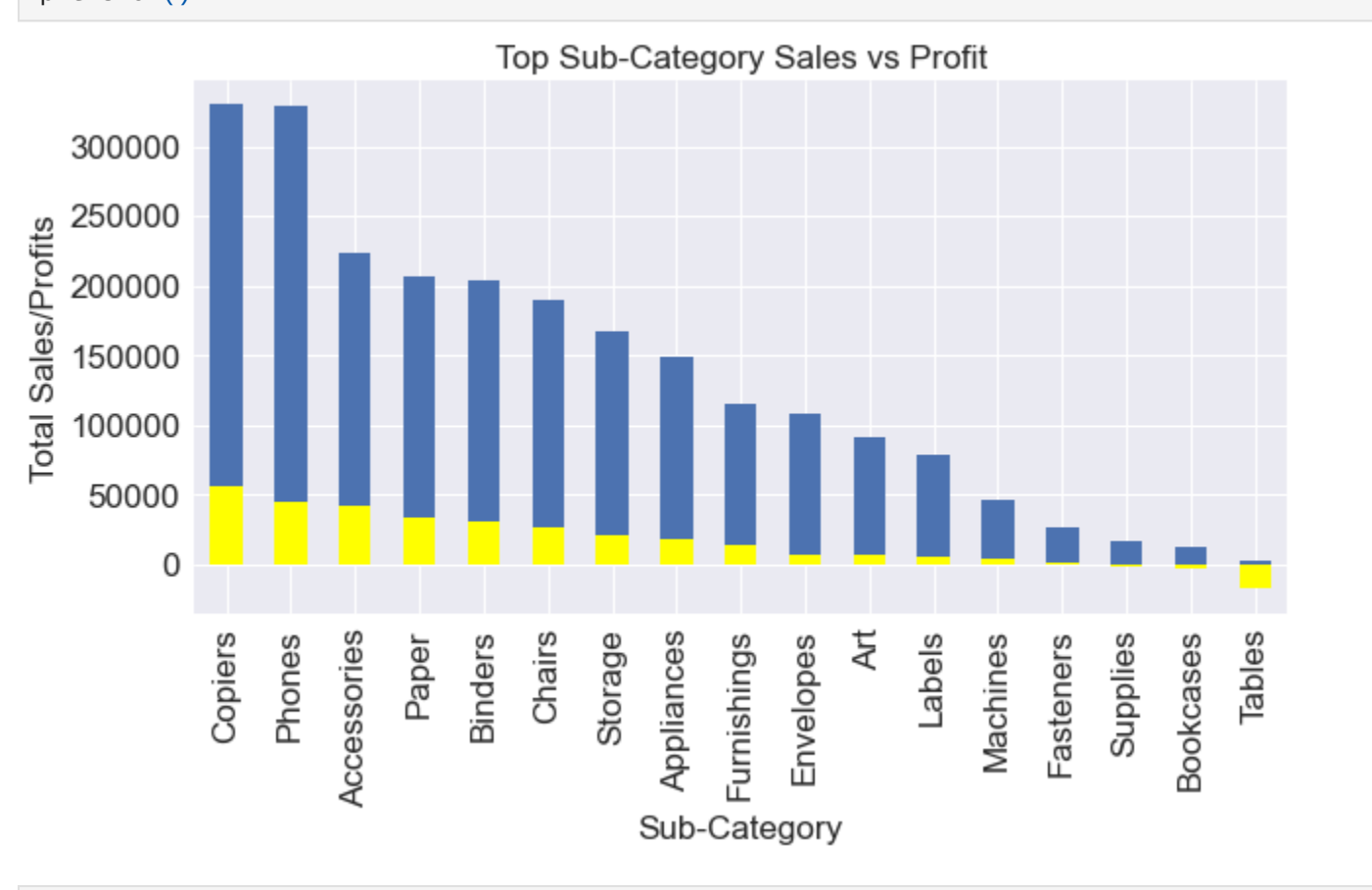


In [25]:
```
# computing top categories in terms of sales from first 100 observations
top_category_s = dset.groupby("Category").Sales.sum().nlargest(n=100)
# computing top categories in terms of profit from first 100 observations
top_category_p = dset.groupby("Category").Profit.sum().nlargest(n=100)

# plotting to see it visually
plt.style.use('seaborn')
top_category_s.plot(kind = 'bar',figsize = (15,10),fontsize = 18)
top_category_p.plot(kind = 'bar',figsize = (15,10),fontsize = 19,color='yellow')
plt.xlabel('Category',fontsize = 17)
plt.ylabel('Total Sales/Profits',fontsize = 17)
plt.title("Top Category Sales vs Profit",fontsize = 17)
plt.show()
```



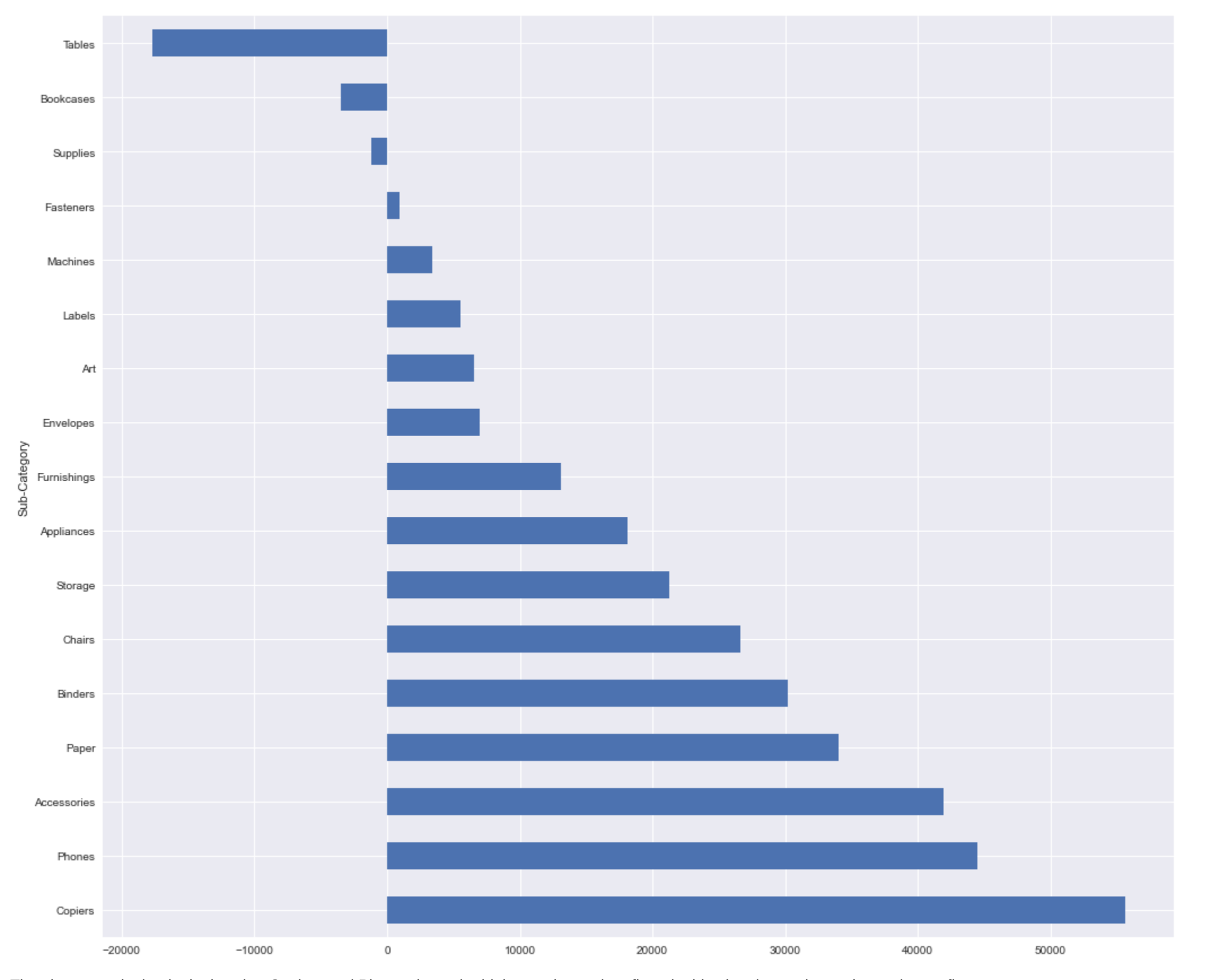
OBSERVING THE SUBCATEGORIES

In [26]:
```
# computing top sub-categories in terms of sales from first 100 observations
top_subcategory_s = dset.groupby("Sub-Category").Sales.sum().nlargest(n = 100)
# computing top sub-categories in terms of profit from first 100 observations
top_subcategory_p = dset.groupby("Sub-Category").Profit.sum().nlargest(n = 100)

# plotting to see it visually
plt.style.use('seaborn')
top_subcategory_s.plot(kind = 'bar',figsize = (10,5),fontsize = 17)
top_subcategory_p.plot(kind = 'bar',figsize = (10,5),fontsize = 17, color = 'yellow')
plt.xlabel('Sub-Category',fontsize = 17)
plt.ylabel('Total Sales/Profits',fontsize = 17)
plt.title("Top Sub-Category Sales vs Profit",fontsize = 17)
plt.show()
```



In [27]:
```
# A more detailed view
plt.figure(figsize=(17,15))
statewise = dset.groupby(['Sub-Category'])['Profit'].sum().nlargest(50)
statewise.plot.barh() # h for horizontal
```

Out[27]: <AxesSubplot:ylabel='Sub-Category'>

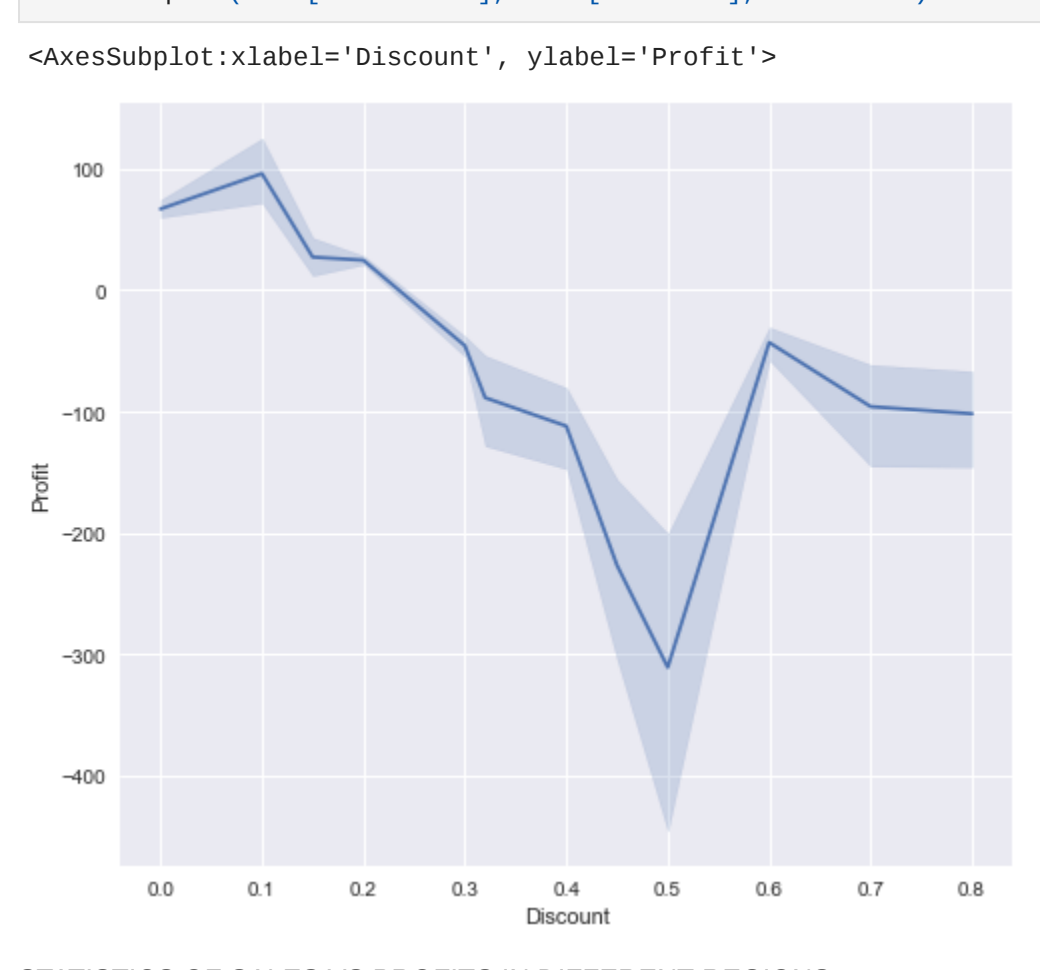


The above graph clearly depicts that Copiers and Phones have the highest sales and profit and tables has less sales and negative profits.

OBSERVING THE DISCOUNT TRENDS

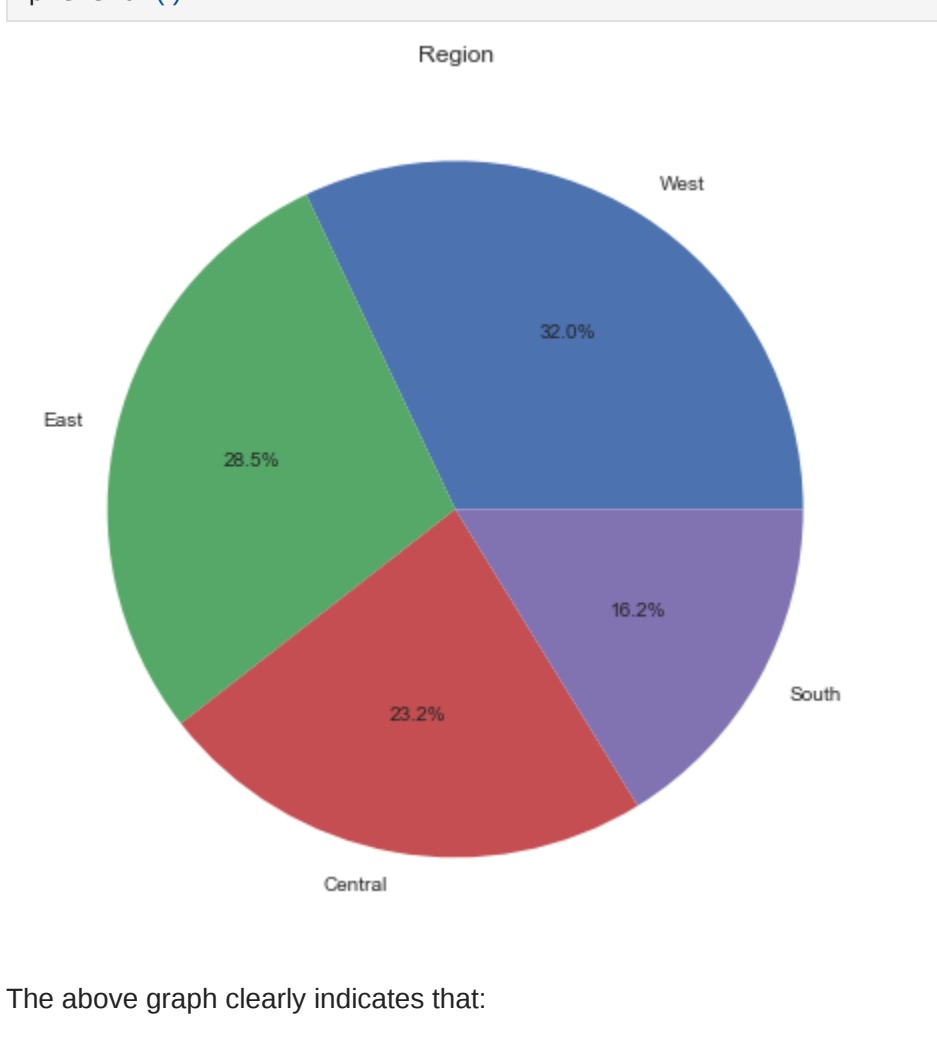
In [36]:
```
plt.figure(figsize=(8,7))
sns.lineplot(dset['Discount'], dset['Profit'], data=dset)
```

Out[36]: <AxesSubplot:xlabel='Discount', ylabel='Profit'>



STATISTICS OF SALES VS PROFITS IN DIFFERENT REGIONS

In [29]:
```
plt.figure(figsize = (8,8))
plt.title('Region')
plt.pie(dset['Region'].value_counts(), labels=dset['Region'].value_counts().index,autopct='%1.1f%%')
plt.show()
```



The above graph clearly indicates that:

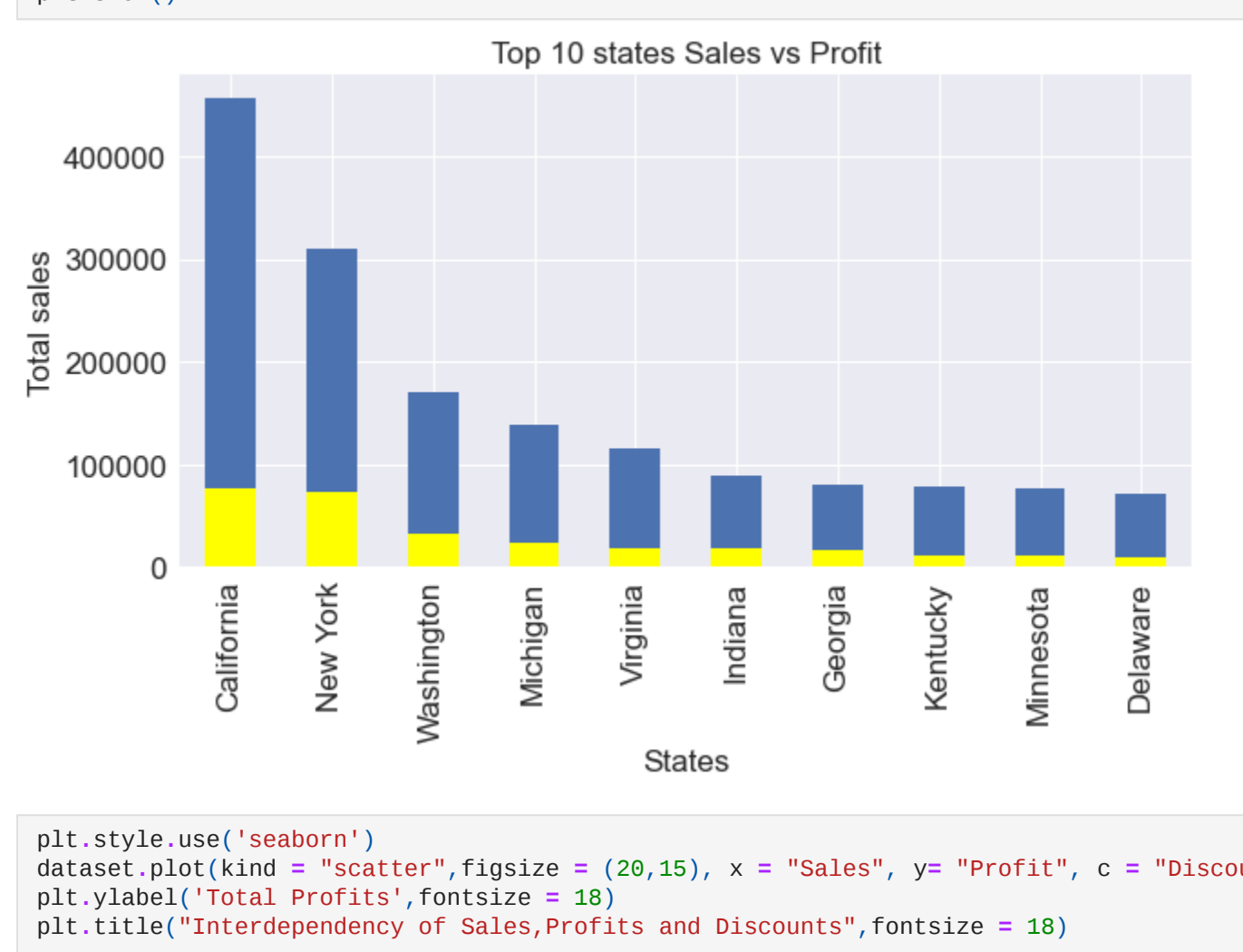
1. West has highest profits of 32.0%

1. Followed by East, with 28.5%

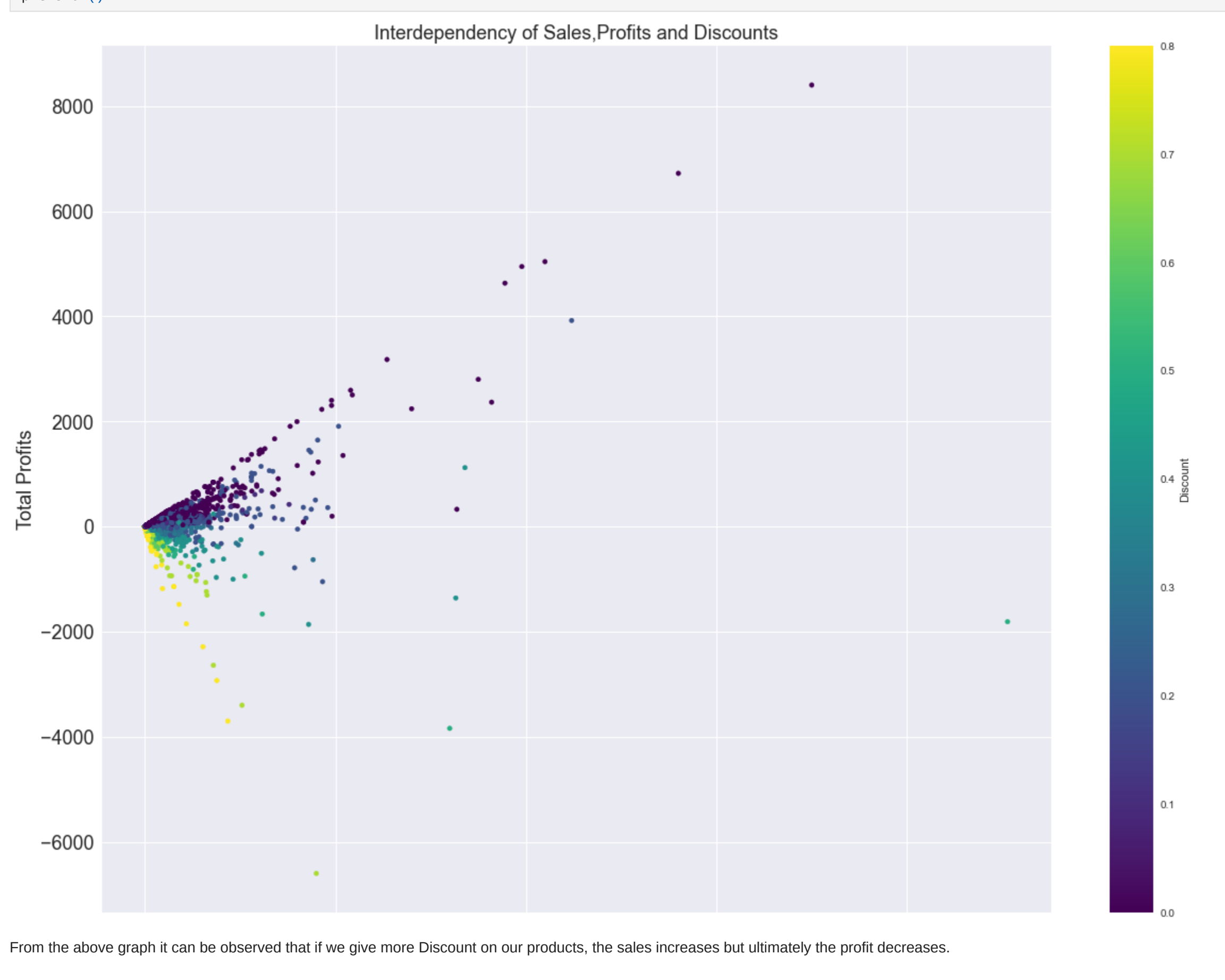
1. Then we have Central region with 23.2%

1. The sales in South are comparitively low, with 16.2%

In [31]:
```
# computing top states in terms of sales from first 10 observations
top_states_s = dset.groupby("State").Sales.sum().nlargest(n=10)

# computing top states in terms of profit from first 10 observations
top_states_p = dset.groupby("State").Profit.sum().nlargest(n = 10)

plt.style.use('seaborn')
top_states_s.plot(kind = 'bar',figsize = (10,5),fontsize = 17)
top_states_p.plot(kind = 'bar',figsize = (10,5),fontsize = 17, color = 'yellow')
plt.xlabel('States',fontsize = 17)
plt.ylabel('Total sales',fontsize = 17)
plt.title("Top 10 states Sales vs Profit",fontsize = 17)
plt.show()
```



In [33]:
```
plt.style.use('seaborn')
dataset.plot(kind = "scatter",figsize = (20,15), x = "Sales", y= "Profit", c = "Discount", s = 20,fontsize = 18, colormap = "viridis")
plt.ylabel('Total Profits',fontsize = 18)
plt.title("Interdependency of Sales,Profits and Discounts",fontsize = 18)

plt.show()
```



From the above graph it can be observed that if we give more Discount on our products, the sales increases but ultimately the profit decreases.

OBSERVATIONS FROM THE DATA

The weak areas to work on, in order to maximise profits:

```
1.We should limit sales of furniture and increase that of technology and office suppliers as furniture has very less profit as compared to sales.
2.The sub-categories sales of tables should be minimized.
3.Increase sales more in the east, as profit is more.
4.We should concentrate more on 'New York' and 'California' to make more profits.
```

DONE BY: NOTAM KEDARI

In [ ]: