Arindam Ghosh

Task-3: Exploratory data analysis

In [2]:
```
#import dataset
import pandas as pd
df = pd.read_csv("E:/task3/SampleSuperstore.csv")
df.head()
```

Out[2]:

	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 [3]:
```
#data analysis
df.shape
```

Out[3]:
```
(9994, 13)
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
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 [7]:
```
df.nunique()
```

Out[7]:
```
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 [9]:
```
df.duplicated().sum()
```

Out[9]:
```
17
```

In [10]:
```
df = df.drop_duplicates()
```

In [12]:
```
df = df.drop(['Postal Code'], axis = 1)
```

In [13]:
```
df.head()
```

Out[13]:

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

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

Out[14]:

	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 [17]:
```
#importing Required libraries
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

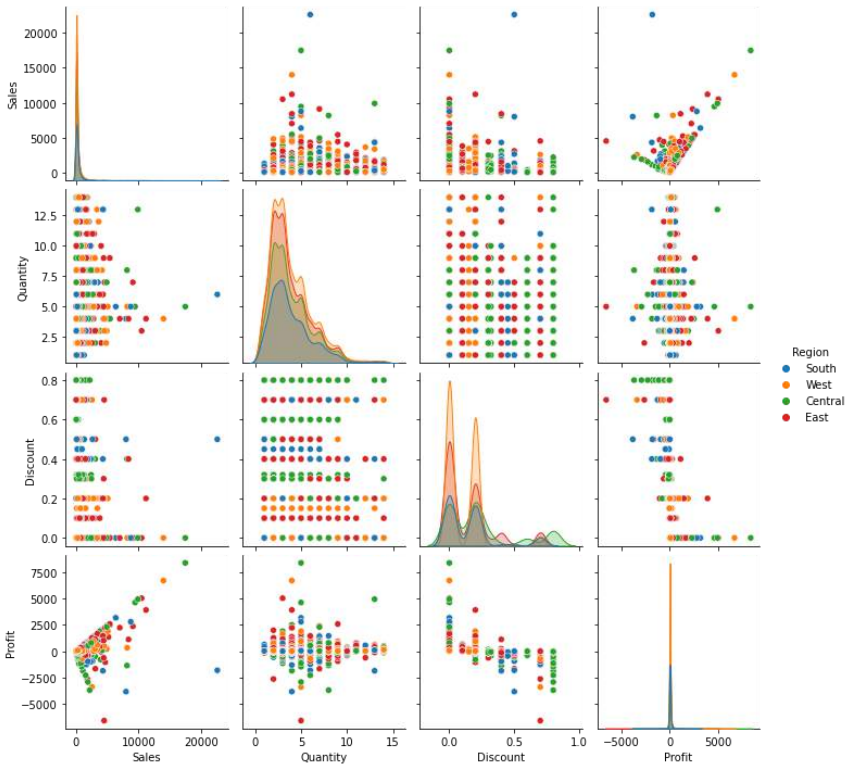
In [18]:
```
#correlation matrix
corr = df.corr()
corr
```

Out[18]:

	Sales	Quantity	Discount	Profit
Sales	1.000000	0.200722	-0.028311	0.479067
Quantity	0.200722	1.000000	0.008678	0.066211
Discount	-0.028311	0.008678	1.000000	-0.219662
Profit	0.479067	0.066211	-0.219662	1.000000

In [19]:
```
sns.pairplot(data = df, hue = 'Region')
```

Out[19]:
```
<seaborn.axisgrid.PairGrid at 0x2721f013c70>
```



In [ ]:
```
```

In [45]:
```
#State wise analysis
df['State'].value_counts().head()
```

Out[45]:
```
California      1996
New York        1127
Texas            983
Pennsylvania     586
Washington       502
Name: State, dtype: int64
```

In [46]:
```
df_mean = df.groupby(['State'])[['Sales','Discount','Profit']].mean()
df_mean.head()
```

Out[46]:

State	Sales	Discount	Profit
Alabama	319.846557	0.000000	94.865989
Arizona	157.508933	0.303571	-15.303235
Arkansas	194.635500	0.000000	66.811452
California	229.246629	0.072946	38.241878
Colorado	176.418231	0.316484	-35.867351

In [47]:
```
df_mean_1 = df_mean.sort_values('Profit')
df_mean_1.head()
```

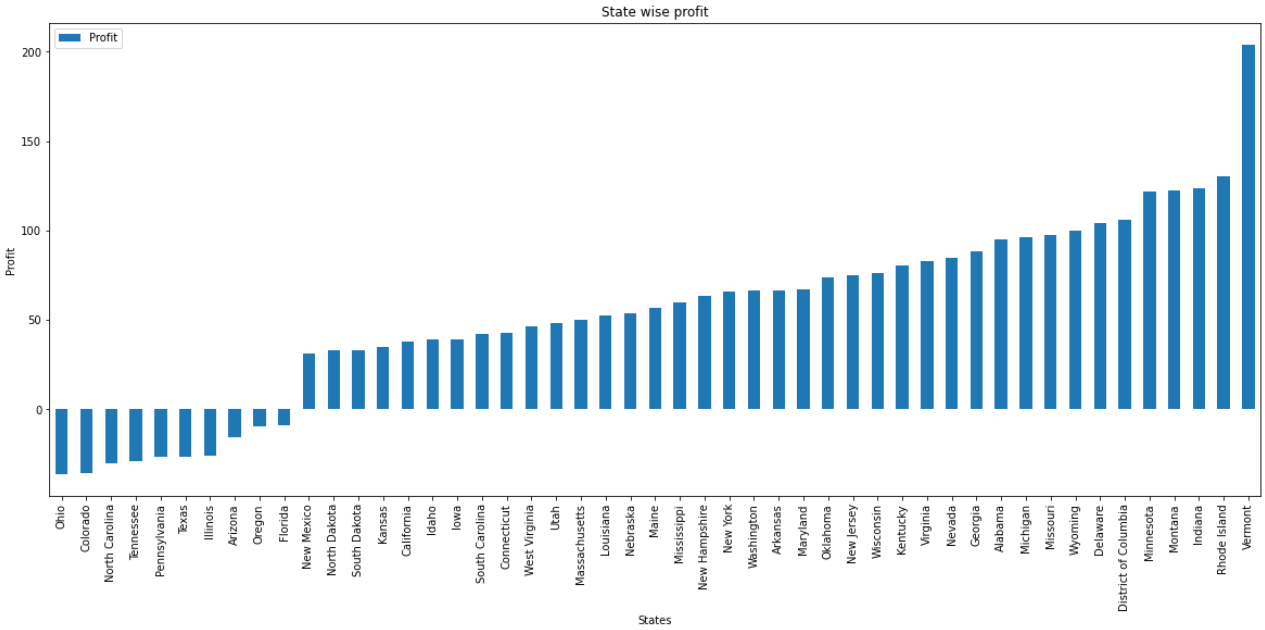
Out[47]:

State	Sales	Discount	Profit
Ohio	166.617017	0.325000	-36.237859
Colorado	176.418231	0.316484	-35.867351
			30.083985

Tennessee	167.551219	0.291257	-29.189583
Pennsylvania	198.799253	0.328840	-26.562122

In [48]:
```
df_mean_1[['Profit']].plot(kind = 'bar',figsize = (20,8))
plt.title('State wise profit')
plt.ylabel('Profit')
plt.xlabel('States')
plt.figure(figsize=(25,20))
```

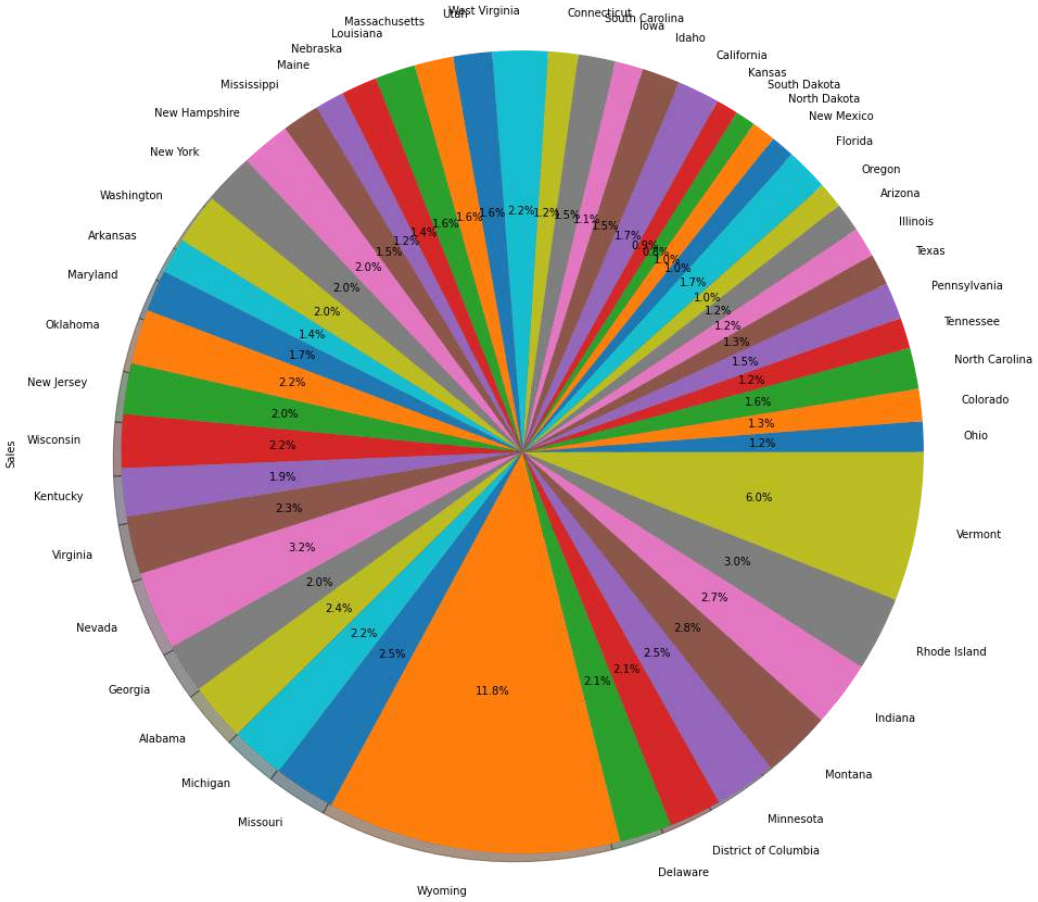
Out[48]: <Figure size 1800x1440 with 0 Axes>



<Figure size 1800x1440 with 0 Axes>

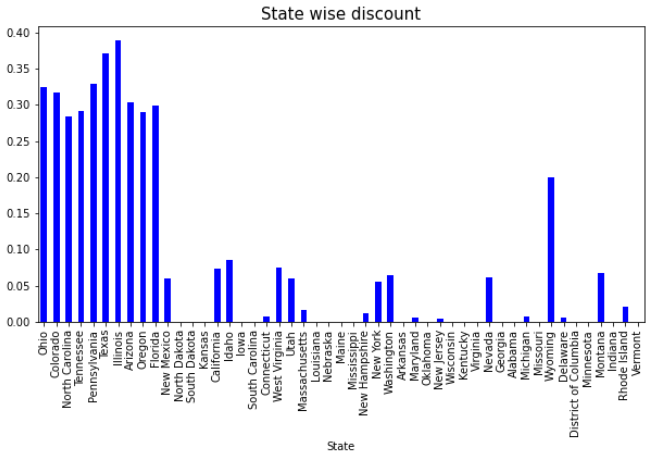
In [49]:
```
df_mean_1['Sales'].plot(kind = 'pie', figsize=(17,17), autopct = '%1.1f%%',startangle=0,shadow=True)
plt.title('State wise Sale', fontsize = 20)
```

Out[49]: Text(0.5, 1.0, 'State wise Sale')



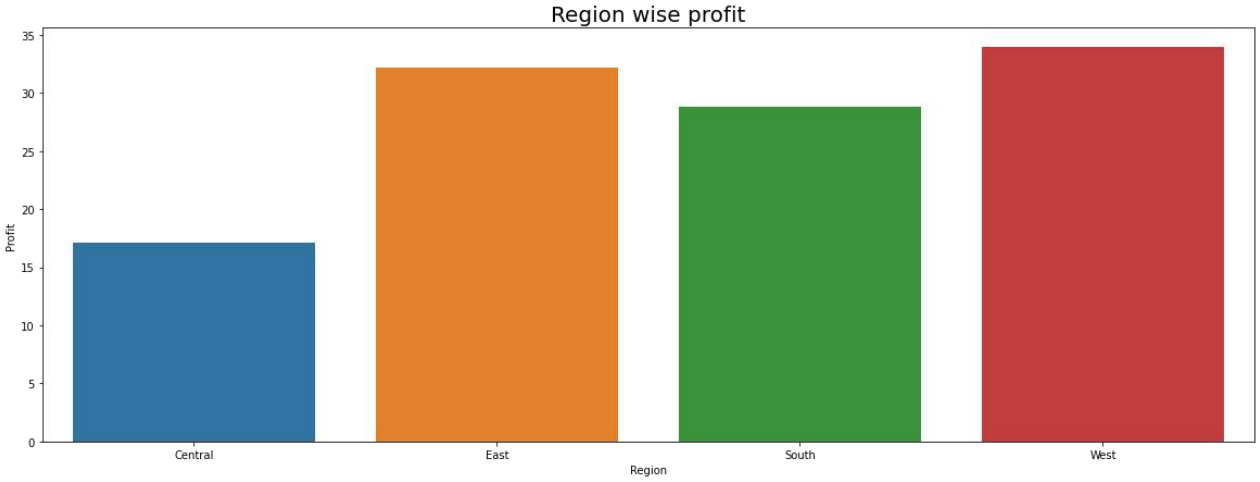
In [50]:
```
df_mean_1['Discount'].plot(kind = 'bar',figsize = (10,5), color = 'b')
plt.title('State wise discount', fontsize=15)
```

Out[50]: Text(0.5, 1.0, 'State wise discount')



In [44]:
```
#Region wise analysis
region_profit = df[['Profit','Region']].groupby(by = 'Region').mean()
plt.figure(figsize = (20,7))
plt.title('Region wise profit', fontsize = 20)
sns.barplot(x = region_profit.index, y = region_profit.Profit , data = region_profit)
```

Out[44]: <AxesSubplot:title={'center':'Region wise profit'}, xlabel='Region', ylabel='Profit'>



In [52]:
```
#Category wise analysis
df['Category'].value_counts()
```

Out[52]:
```
Office Supplies    6012
Furniture          2118
Technology         1847
Name: Category, dtype: int64
```

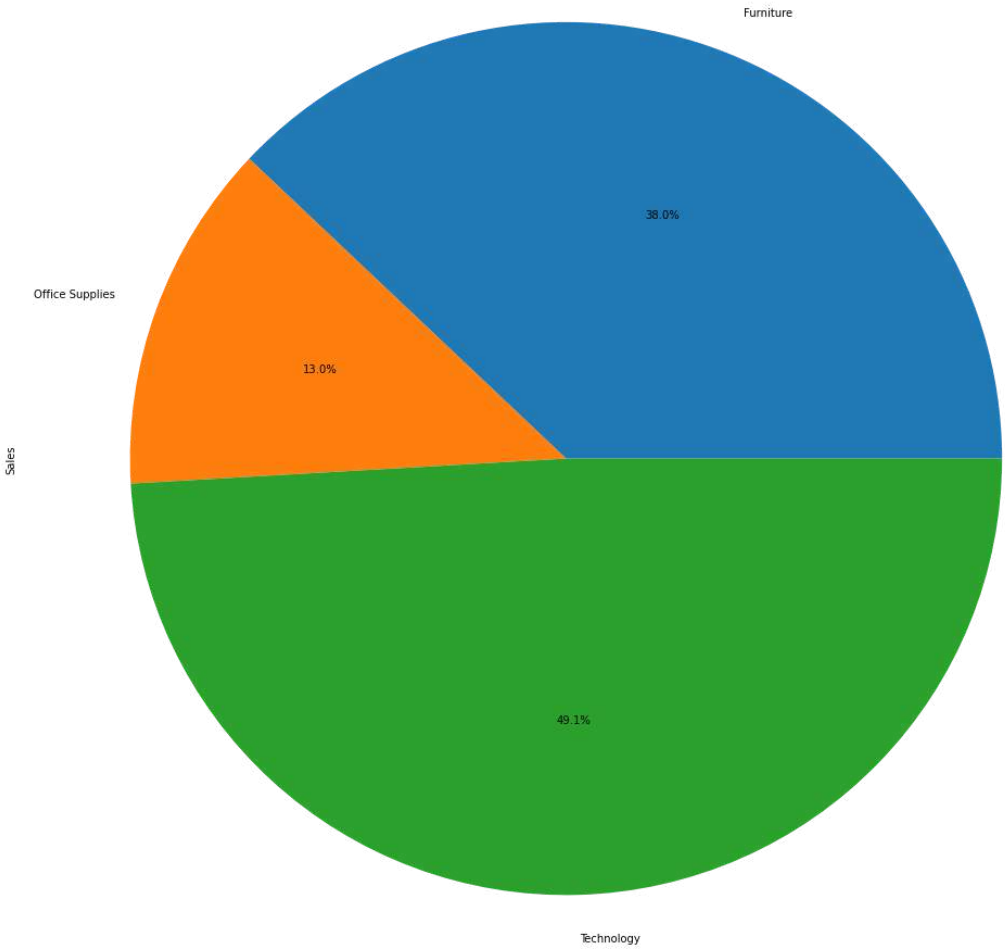
In [54]:
```
df_mean_2 = df.groupby(['Category'])[['Sales','Discount','Profit']].mean()
df_mean_2
```

Out[54]:

Category	Sales	Discount	Profit
Furniture	350.002981	0.174027	8.697740
Office Supplies	119.550107	0.157385	20.353403
Technology	452.709276	0.132323	78.752002

In [56]:
```
df_mean_2['Sales'].plot(kind = 'pie',subplots = True, figsize = (18,20), autopct = '%1.1f%%')
```

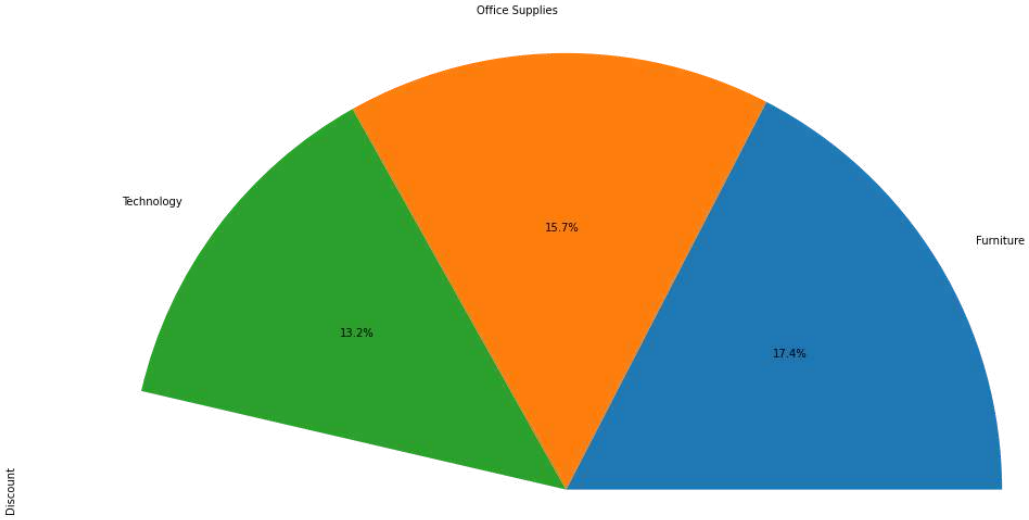
Out[56]:
```
array([<AxesSubplot:ylabel='Sales'>], dtype=object)
```



In [57]:
```
df_mean_2['Discount'].plot(kind = 'pie',subplots = True, figsize = (18,20), autopct = '%1.1f%%')
```

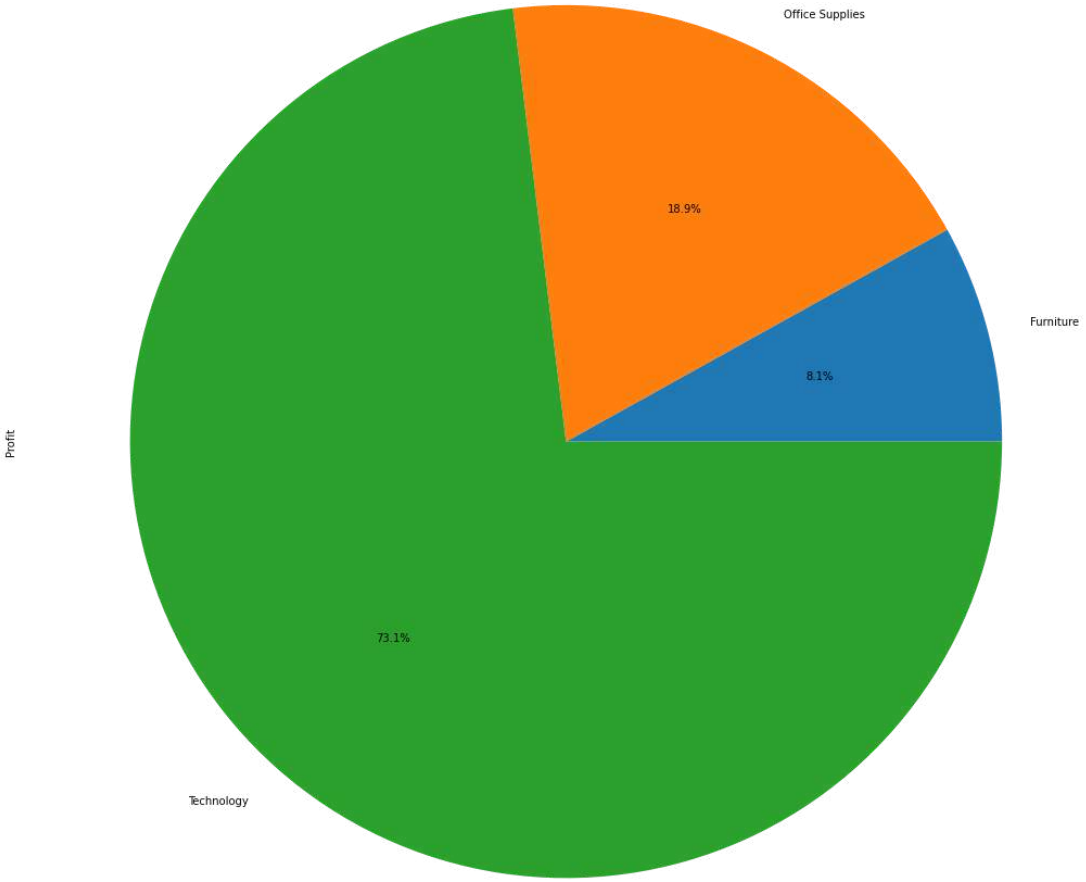
```
C:\Users\ARINDAM\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\core.py:1583: MatplotlibDeprecationWarning: normalize=None does not normalize if the s
um is less than 1 but this behavior is deprecated since 3.3 until two minor releases later. After the deprecation period the default value will be normalize=Tr
ue. To prevent normalization pass normalize=False
  results = ax.pie(y, labels=blabels, **kwds)
```

Out[57]:
```
array([<AxesSubplot:ylabel='Discount'>], dtype=object)
```



In [58]:
```
df_mean_2['Profit'].plot(kind = 'pie',subplots = True, figsize = (18,20), autopct = '%1.1f%%')
```

Out[58]:
```
array([<AxesSubplot:ylabel='Profit'>], dtype=object)
```



In [ ]: