Task 3 - The sparks Foundation

name - smita Rautmare

Perform Exploratory Data Analysis' on dataset 'SampleSuperstore'

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

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

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.dtypes
```

Out[6]:
```
Ship Mode        object
Segment          object
Country          object
City             object
State            object
Postal Code       int64
Region           object
Category         object
Sub-Category     object
Sales           float64
Quantity          int64
Discount        float64
Profit          float64
dtype: object
```

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

Out[7]:

	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 [8]:
```
df.shape
```

Out[8]:
```
(9994, 13)
```

In [9]:
```
df.describe(include="all")
```

Out[9]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Disco
count	9994	9994	9994	9994	9994	9994.000000	9994	9994	9994	9994.000000	9994.000000	9994.000
unique	4	3	1	531	49	NaN	4	3	17	NaN	NaN	N
top	Standard Class	Consumer	United States	New York City	California	NaN	West	Office Supplies	Binders	NaN	NaN	N
freq	5968	5191	9994	915	2001	NaN	3203	6026	1523	NaN	NaN	N
mean	NaN	NaN	NaN	NaN	NaN	55190.379428	NaN	NaN	NaN	229.858001	3.789574	0.156
std	NaN	NaN	NaN	NaN	NaN	32063.693350	NaN	NaN	NaN	623.245101	2.225110	0.206
min	NaN	NaN	NaN	NaN	NaN	1040.000000	NaN	NaN	NaN	0.444000	1.000000	0.000
25%	NaN	NaN	NaN	NaN	NaN	23223.000000	NaN	NaN	NaN	17.280000	2.000000	0.000
50%	NaN	NaN	NaN	NaN	NaN	56430.500000	NaN	NaN	NaN	54.490000	3.000000	0.200
75%	NaN	NaN	NaN	NaN	NaN	90008.000000	NaN	NaN	NaN	209.940000	5.000000	0.200
max	NaN	NaN	NaN	NaN	NaN	99301.000000	NaN	NaN	NaN	22638.480000	14.000000	0.800

Univariate Analysis

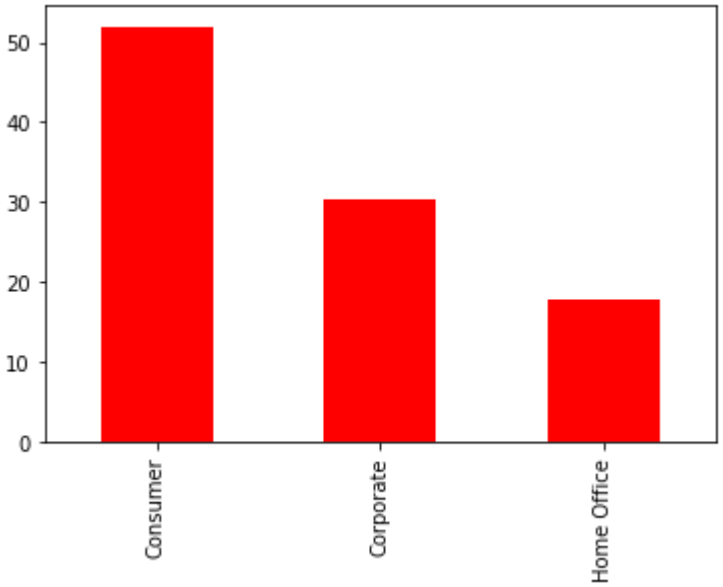
In [10]:
```
df['Segment'].value_counts()
```

Out[10]:
```
Consumer       5191
Corporate      3020
Home Office    1783
Name: Segment, dtype: int64
```

In [12]:
```
df['Segment'].value_counts()/len(df['Segment'])*100
```

Out[12]:
```
Consumer       51.941165
Corporate      30.218131
Home Office    17.840704
Name: Segment, dtype: float64
```

In [13]:
```
S=(df["Segment"].value_counts()/len(df["Segment"])*100).plot(kind='bar',color='r')
```

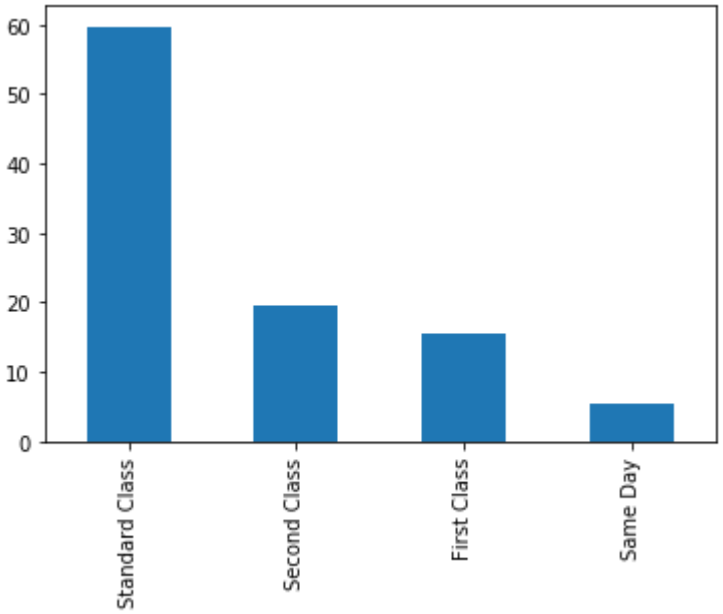


In [14]:
```
M=(df['Ship Mode'].value_counts())/len(df["Ship Mode"])*100
M
```

Out[14]:
```
Standard Class    59.715829
Second Class      19.461677
First Class       15.389234
Same Day           5.433260
Name: Ship Mode, dtype: float64
```

In [15]:
```
M.plot(kind='bar')
```

Out[15]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x21542cbd648>
```

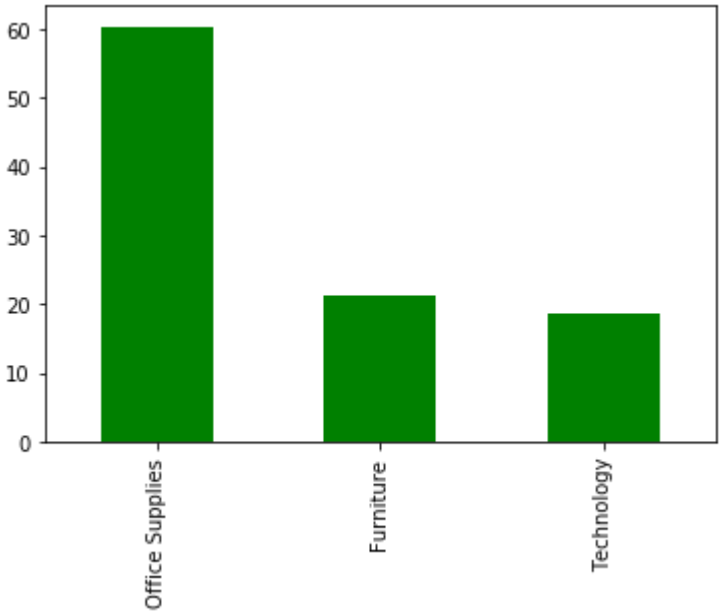


In [16]:
```
df['Category'].value_counts()
```

Out[16]:
```
Office Supplies    6026
Furniture          2121
Technology         1847
Name: Category, dtype: int64
```

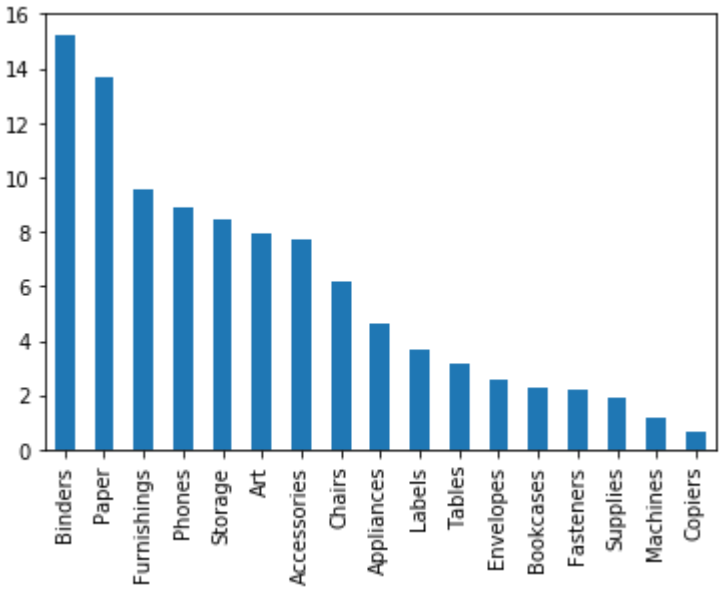
In [17]:
```
c=(df['Category'].value_counts())/len(df['Category'])*100
c.plot(kind='bar',color='g')
```

Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x21542d31688>



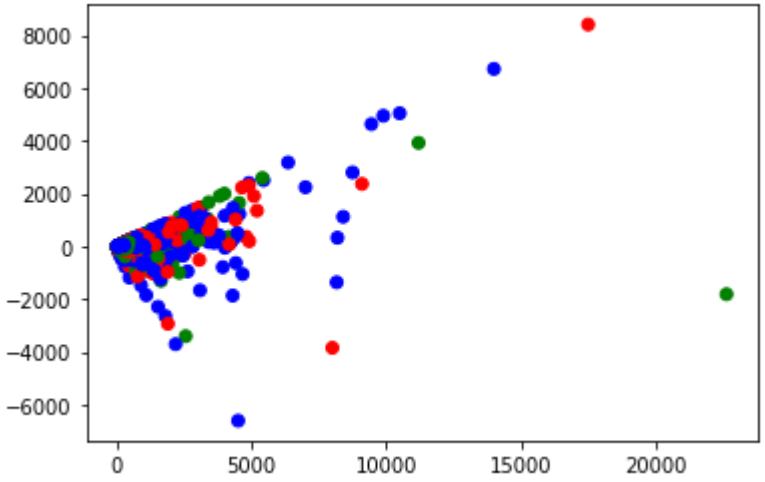
In [18]:
```
((df['Sub-Category'].value_counts())/len(df["Sub-Category"])*100).plot(kind='bar')
```

Out[18]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x21542d98148>
```



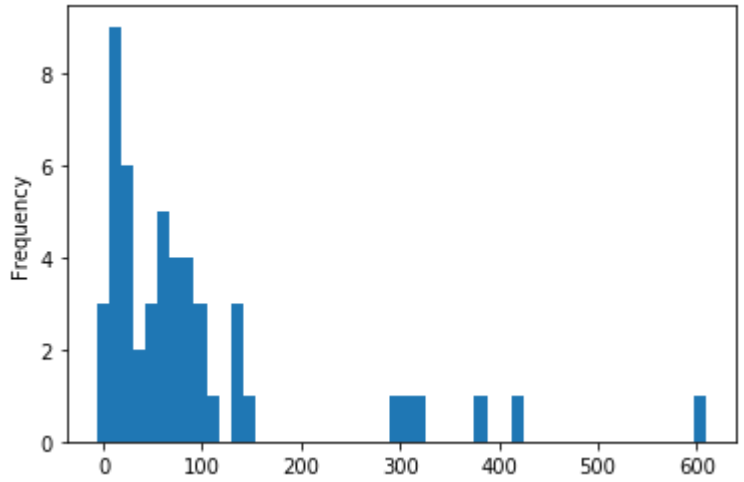
Bivariate Analysis

In [22]:
```
fig,a=plt.subplots()
colors = {'Consumer':'blue', 'Corporate':'red', 'Home Office':'green'}
a.scatter(df['Sales'],df['Profit'],c=df["Segment"].apply(lambda x: colors[x]))
plt.show()
```



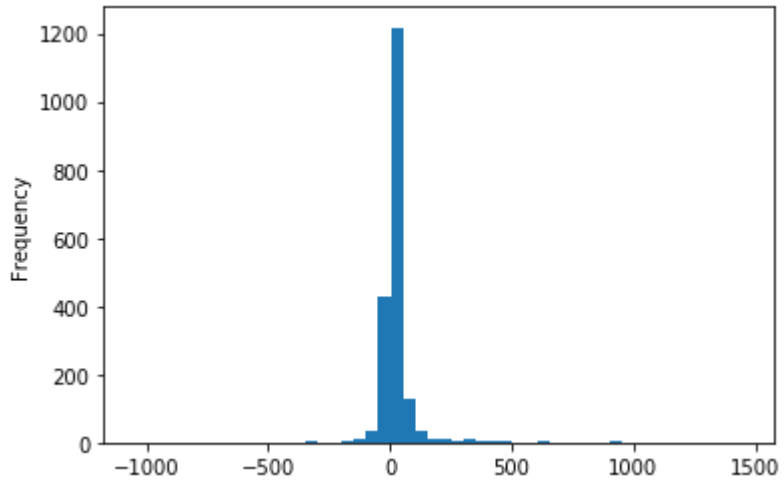
In [24]:
```
tem_df=df.loc[(df['Segment']=='Consumer')&(df["Discount"]==0.1)]
tem_df['Profit'].plot.hist(bins=50)
```

Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x21543f05988>



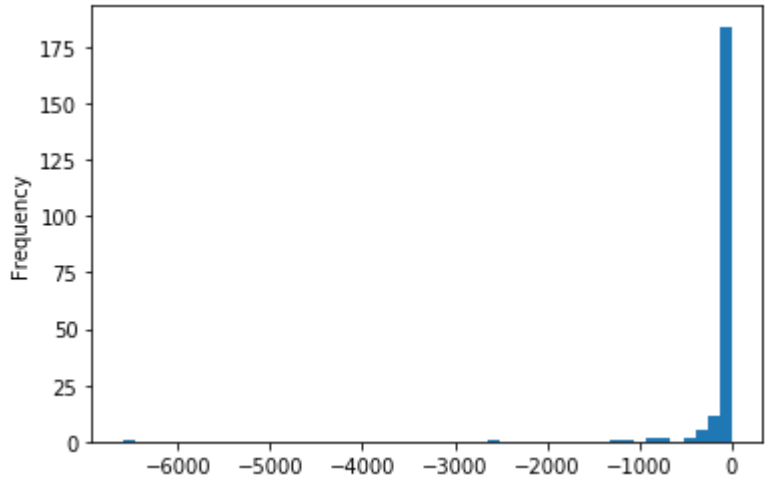
In [25]:
```
tem_df=df.loc[(df['Segment']=='Consumer')&(df["Discount"]==0.2)]
tem_df['Profit'].plot.hist(bins=50)
```

Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x21543feea48>



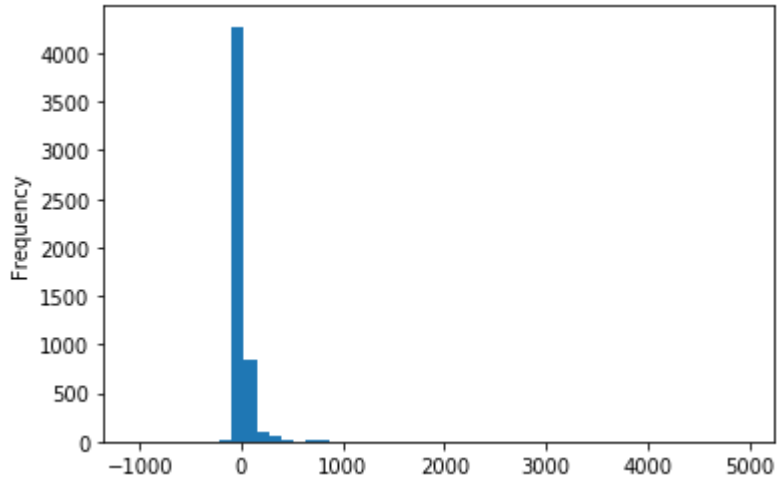
In [27]:
```
tem_df=df.loc[(df['Segment']=='Consumer')&(df["Discount"]==0.7)]
tem_df['Profit'].plot.hist(bins=50)
```

Out[27]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x21544125d08>
```



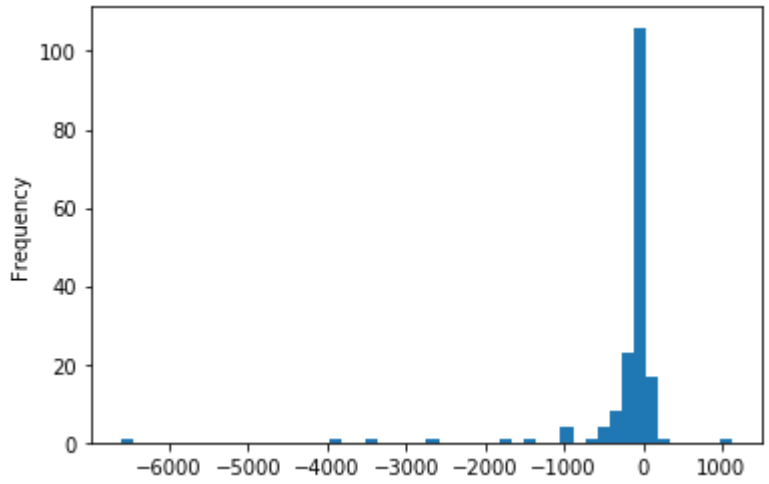
In [32]:
```
tem_df=df.loc[(df['Category']=='Office Supplies')&(df["Discount"]<=0.3)]
tem_df['Profit'].plot.hist(bins=50)
```

Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x215440c7f08>



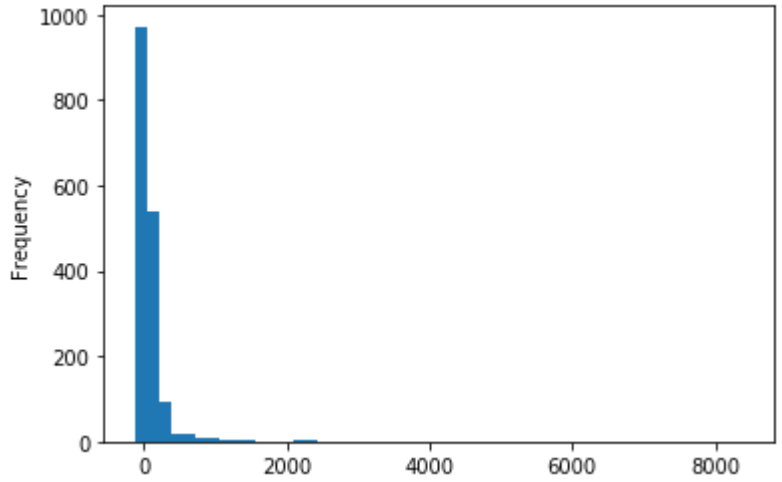
In [37]:
```
tem_df=df.loc[(df['Category']=='Technology')&(df["Discount"]>=0.3)]
tem_df['Profit'].plot.hist(bins=50)
```

Out[37]: <matplotlib.axes._subplots.AxesSubplot at 0x2154422dfc8>



In [38]:
```
tem_df=df.loc[(df['Category']=='Technology')&(df["Discount"]<=0.3)]
tem_df['Profit'].plot.hist(bins=50)
```

Out[38]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2154198bdc8>
```



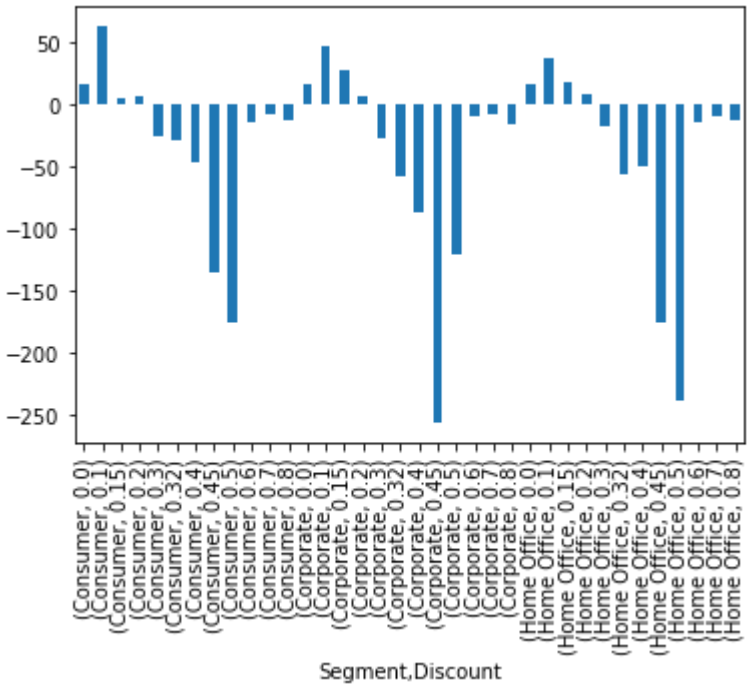
from all observation it is conclude that

when discount<=30% then sales was going to profit

when discount>=30% then superstore has huge loss

In [41]:
```
ans= df.groupby(["Segment","Discount"]).Profit.median()
ans.plot(kind='bar',stacked=True)
```

Out[41]: <matplotlib.axes._subplots.AxesSubplot at 0x215445f5248>



this shows exact scenario of profit of all segment when following discount offerd

In [ ]: