

Question#1. take year 1 and examine the plot of your labels. Construct a reduced dataset by removing some green and red points so that you can draw a line separating the points. Compute the equation of such a line (many solutions are possible)

```
In [58]: import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
df=pd.read_csv("NVDA_weekly_return_volatility.csv")
df_2017=df.loc[df["Year"]==2017,:]
df_2018=df.loc[df["Year"]==2018,:]
print(df_2018)
```

	Year	Week_Number	mean_return	volatility
52	2018	0	0.76600	0.369974
53	2018	1	0.32060	0.832617
54	2018	2	0.11900	1.415193
55	2018	3	0.88960	0.890441
56	2018	4	-0.47540	1.871303
57	2018	5	-0.72440	4.256168
58	2018	6	0.95060	1.044823
59	2018	7	0.56450	1.593698
60	2018	8	-0.21120	1.086412
61	2018	9	0.74240	0.929728
62	2018	10	-0.39920	1.212748
63	2018	11	-1.61280	1.390139
64	2018	12	1.24725	5.054742
65	2018	13	-0.20420	2.481235
66	2018	14	0.63260	1.479650
67	2018	15	0.41540	1.242980
68	2018	16	0.18520	1.829708
69	2018	17	-0.12540	1.756984
70	2018	18	0.53080	0.782874
71	2018	19	-0.18780	0.624422
72	2018	20	0.41420	0.767087
73	2018	21	0.61625	1.069643
74	2018	22	0.17000	0.998158
75	2018	23	-0.29500	0.724619
76	2018	24	0.05820	0.798730
77	2018	25	-0.35340	1.376620
78	2018	26	0.64500	1.118987
79	2018	27	0.83340	0.893494
80	2018	28	0.16420	1.159775
81	2018	29	0.27900	1.952285
82	2018	30	0.07380	1.296973
83	2018	31	0.17800	0.510863
84	2018	32	-0.18160	0.969596
85	2018	33	0.15480	0.847951
86	2018	34	0.71640	0.649555
87	2018	35	-0.92275	1.353369
88	2018	36	0.93700	0.540086
89	2018	37	0.16340	1.313901
90	2018	38	0.01960	0.345000
91	2018	39	-0.38980	1.138941
92	2018	40	-0.41660	3.300829
93	2018	41	-0.14960	2.072453
94	2018	42	-0.24860	4.083815
95	2018	43	-0.13220	2.117954
96	2018	44	0.65280	2.171354
97	2018	45	-0.13500	1.852363
98	2018	46	-1.20850	2.256683
99	2018	47	1.48820	1.943603
100	2018	48	-1.37200	2.593717
101	2018	49	0.24740	2.099191
102	2018	50	-1.50260	1.837873
103	2018	51	0.62325	4.601512
104	2018	52	1.17500	0.000000

Question#1. take year 1 and examine the plot of your labels. Construct a reduced dataset by removing some green and red points so that you can draw a line separating the points. Compute the equation of such a line (many solutions are possible)

```
In [59]: Q1_label=[]
for i in range(len(df_2017)):
    if df_2017["mean_return"][i]>0:
        Q1_label.append("green")
    elif df_2017["mean_return"][i]<0:
        Q1_label.append("red")
    else:
        Q1_label.append("no change")
df_2017["label"]=Q1_label
#print(df_2017)
df_green=df_2017.loc[(df_2017["label"]=="green"), :]
df_red=df_2017.loc[(df_2017["label"]=="red"), :]
#print(df_green)
plt.plot(df_green["mean_return"],df_green["volatility"],'o',markersize=4.,color='green')
plt.scatter(df_red["mean_return"],df_red["volatility"],s=16.,color=(0.,0.5,0.))
plt.axvline(0)
plt.show()
```

/var/folders/by/3jff2bh0x2ks63rycd0ctbzsh0000gn/T/ipykernel_4915/4058242621.py:

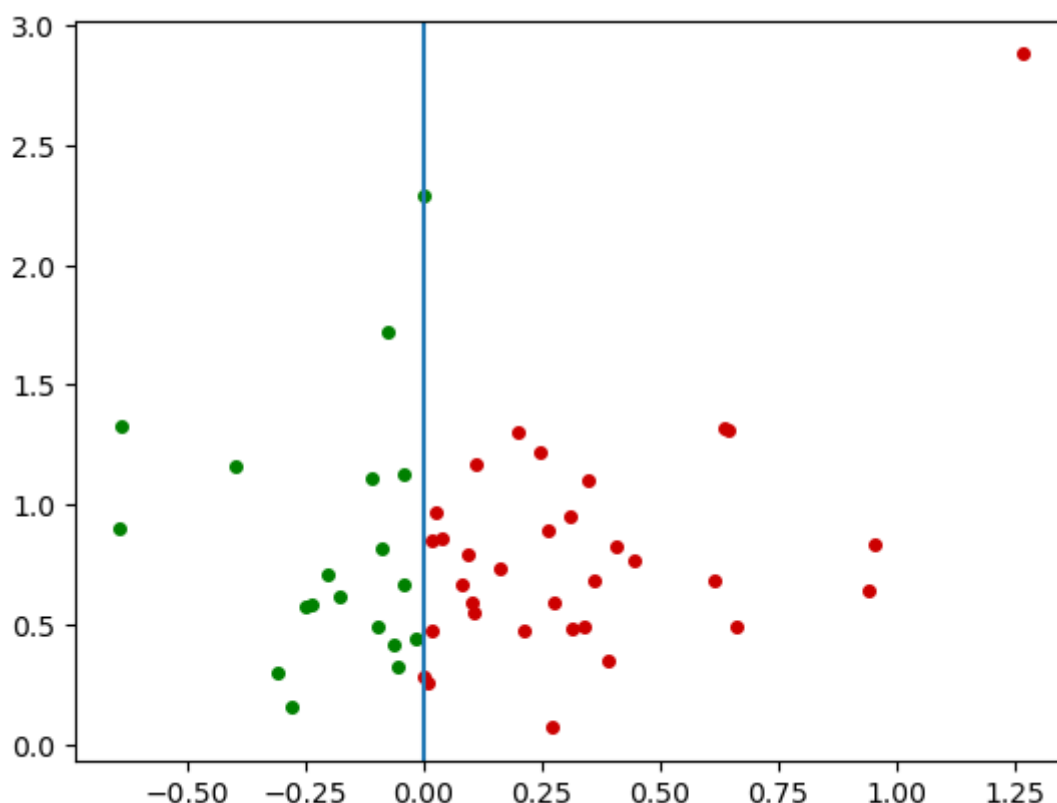
9: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

df_2017["label"]=Q1_label



Question#2. take this line and use it to assign labels for year 2

```
In [60]: Q2_label=[]
for i in range(len(df_2018)):
    if df_2018["mean_return"][i+52]>0:
        Q2_label.append("green")
    elif df_2018["mean_return"][i+52]<0:
```

```

Q2_label.append("red")
else:
    Q2_label.append("no change")
df_2018["label"]=Q2_label
df_green_Q2=df_2018.loc[(df_2018["label"]=="green"), :]
df_red_Q2=df_2018.loc[(df_2018["label"]=="red"), :]
#print(df_green)
plt.plot(df_green_Q2["mean_return"],df_green_Q2["volatility"],'o',markersize=4.)
plt.scatter(df_red_Q2["mean_return"],df_red_Q2["volatility"],s=16.,color=(0.,0.))
plt.axvline(0)
plt.show()

```

/var/folders/by/3jf2bh0x2ks63rycd0ctbzsh0000gn/T/ipykernel_4915/1703682851.py:

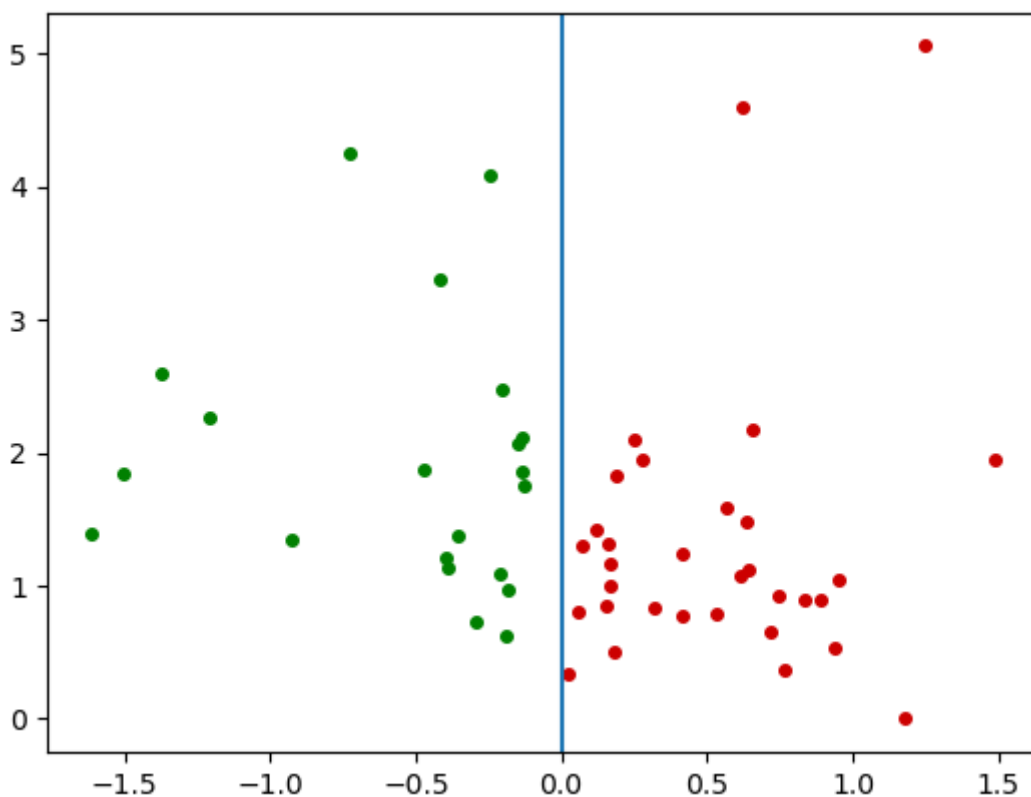
9: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df_2018["label"]=Q2_label
```



Question#3 implement a trading strategy based on your labels for year 2.

In []: