In [162…
```
import pandas as pd
import numpy as np
import sklearn
import matplotlib.pyplot as plt
from sklearn.model_selection \
import train_test_split
import seaborn as sns
```

Question#1. for every year (out of 5 years), compute the number of days with positive and negative returns.

In [163…
```
data=pd.read_csv("NVDA.csv")
label=[]
for i in range(len(data)):
    if data["Return"][i]<0:
        label.append("Negative")
    elif data["Return"][i]>0:
        label.append("Positive")
    else:
        label.append("Nan")
data["label"]=label
#print(data)
print("Negative return data for NVDA",data.loc[data['label'] == "Negative"].gro
print("Positive return data for NVDA",data.loc[data['label'] == "Positive"].gro
```

```
Negative return data for NVDA Year
2017    112
2018    124
2019    113
2020    105
2021    115
Name: label, dtype: int64
Positive return data for NVDA Year
2017    138
2018    127
2019    138
2020    148
2021    135
Name: label, dtype: int64
```

Question#2. for each year, compute the average of daily returns μ and compute the percentage of days with returns greater than μ and the proportion of days with returns less than μ. Are there more positive or negative return days? Does it change from year to year? Summarize your results for this question in a table for each year and discuss your findings. Your table should have the following format:

In [164…
```
Q2_mu=[]
Q3_std=[]
year=data["Year"].unique()
Q2_label=["label_2017", "label_2018", "label_2019", "label_2020", "label_2021"]
for i in range(len(year)):
    temp=data.loc[data["Year"]==year[i]]["Return"].mean()
    Q3_std.append(data.loc[data["Year"]==year[i]]["Return"].std())
    Q2_mu.append(temp)
    data[Q2_label[i]]=data.loc[data['Year'] == year[i]]["Return"].apply(lambda
```

```
Nega=[]
Posi=[]
trade_day=[]

for i in range(len(year)):
    Nega.append(data.loc[data[Q2_label[i]]=="Negative"]["Year"].count()/data.lo
    Posi.append(data.loc[data[Q2_label[i]]=="Positive"]["Year"].count()/data.lo
    trade_day.append(data.loc[data["Year"]==year[i]]["Year"].count())
#print(Nega,Posi)
Q2res=pd.DataFrame({"Year":year,
                    'trading days':trade_day,
                    'μ':Q2_mu,
                    'days<μ':Nega,
                    'days>μ':Posi})
Q2res
```

Out[164]:

	Year	trading days	μ	days<μ	days>μ
0	2017	251	0.002882	0.509960	0.490040
1	2018	251	-0.000973	0.470120	0.529880
2	2019	252	0.002595	0.484127	0.515873
3	2020	253	0.003828	0.505929	0.494071
4	2021	251	0.003666	0.525896	0.474104

Quetion#3. for every year, compute the mean and standard deviation of your daily returns. Compute the number of days that your (by absolute value) returns are more than 2 standard deviations from the mean. In other words, if μ = 5 and σ = 2, compute the number of days that your (percent) daily returns are less than 1 (5 - 2*2) or more than 9 (5 + 2*2*). The number of such days per year predicted by normal distribution is less than 5% (out of 252 trading days) - 2.5% below μ–2σ and 2.5% above μ+2σ.

In [165…

```
Q3_Nage=[]
Q3_Posi=[]
Q3_Na_label=["Q3_Na_label_2017", "Q3_Na_label_2018", "Q3_Na_label_2019", "Q3_Na
Q3_Po_label=["Q3_Po_label_2017", "Q3_Po_label_2018", "Q3_Po_label_2019", "Q3_Po
for i in range(len(year)):
    Q3_Nage.append(Q2_mu[i]-(Q3_std[i]*2))
    Q3_Posi.append(Q2_mu[i]+(Q3_std[i]*2))
for i in range(len(year)):
    data[Q3_Po_label[i]]=data.loc[data['Year'] == year[i]]["Return"].apply(lamb
    data[Q3_Na_label[i]]=data.loc[data['Year'] == year[i]]["Return"].apply(lamb


Q3_res_nage=[]
Q3_res_posi=[]
for i in range(len(year)):
    Q3_res_nage.append(data.loc[data[Q3_Na_label[i]]=="Nagetive"]["Year"].count
    Q3_res_posi.append(data.loc[data[Q3_Po_label[i]]=="Positive"]["Year"].count
print(Q3_res_nage,Q3_res_posi)

data.to_csv("Q3.csv")
```

```
[0.027888446215139442, 0.0398406374501992, 0.023809523809523808, 0.02371541501
9762844, 0.0199203187250996] [0.01593625498007968, 0.01195219123505976, 0.0357
1428571428571, 0.015810276679841896, 0.027888446215139442]
```

Question#4. Summarize your findings in a table for each year and discuss your findings

In [166…

```
Q3res=pd.DataFrame({"Year":year,
                    'trading days':trade_day,
                    'μ':Q2_mu,
                    "σ":Q3_std,
                    'days<μ-2σ':Q3_res_nage,
                    'days>μ+2σ':Q3_res_posi})
Q3res
```

Out[166]:

	Year	trading days	μ	σ	days<μ−2σ	days>μ+2σ
0	2017	251	0.002882	0.025267	0.027888	0.015936
1	2018	251	-0.000973	0.030983	0.039841	0.011952
2	2019	252	0.002595	0.025578	0.023810	0.035714
3	2020	253	0.003828	0.036422	0.023715	0.015810
4	2021	251	0.003666	0.028390	0.019920	0.027888