In [61]:

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sn
import numpy as np
from scipy.spatial.distance import squareform
from scipy.spatial.distance import pdist
```

Shape of data set

In [2]:

```python
df=pd.read_csv("pricesperday1.csv")
df=df.dropna()
print('shape of data'),df.shape
df
```

```
shape of data
```

Out[2]:

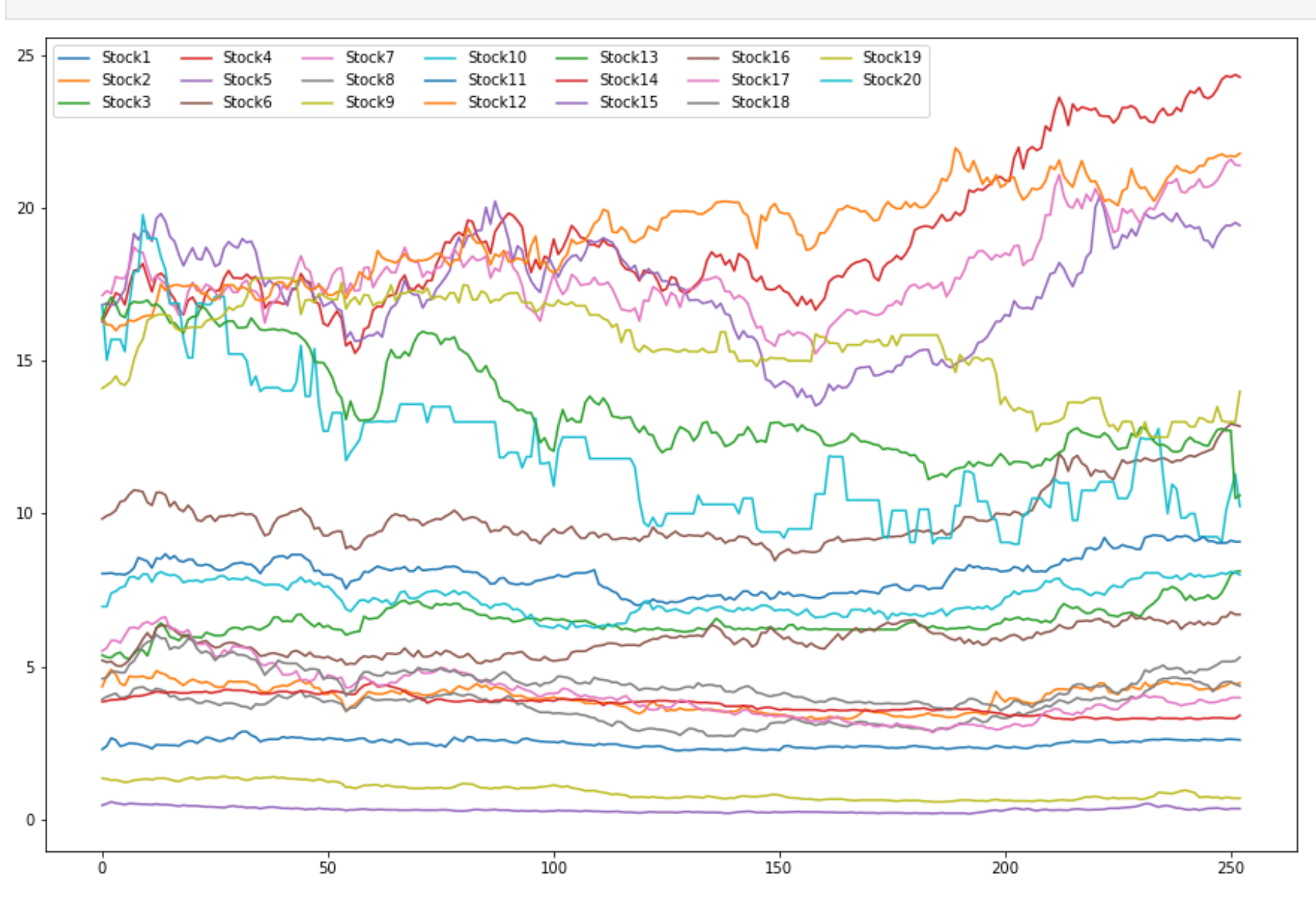
	Stock1	Stock2	Stock3	Stock4	Stock5	Stock6	Stock7	Stock8	Stock9	Stock10	Stock11	Stock12	Stock13	Stock14	Stock15	Stock16	Stock17	Stock18	Stock19	Stock20
0	2.29	4.34	5.37	16.28	16.83	9.83	17.14	3.93	1.34	16.80	8.04	16.42	16.36	3.85	0.46	5.20	5.51	4.60	14.10	6.96
1	2.40	4.70	5.30	16.57	16.88	9.93	17.29	4.01	1.32	15.02	8.04	16.18	16.82	3.87	0.51	5.14	5.61	4.64	14.20	6.96
2	2.66	4.90	5.29	16.86	16.84	9.99	17.20	4.07	1.28	15.70	8.06	16.17	17.08	3.91	0.57	5.16	5.85	4.82	14.30	7.40
3	2.59	4.74	5.41	17.22	16.83	10.11	17.75	4.12	1.29	15.70	8.02	15.99	16.86	3.91	0.53	5.04	5.85	4.84	14.50	7.46
4	2.39	4.44	5.46	17.14	17.08	10.39	17.70	3.97	1.25	15.70	8.02	16.16	16.50	3.93	0.51	5.00	5.77	4.80	14.26	7.58
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
248	2.58	4.34	7.41	24.20	19.28	12.66	21.16	4.46	0.68	9.06	9.04	21.76	12.78	3.32	0.35	6.62	3.90	5.16	13.04	8.06
249	2.62	4.38	7.68	24.32	19.43	12.80	21.46	4.50	0.72	10.14	9.04	21.68	12.72	3.30	0.32	6.62	3.90	5.16	13.00	8.02
250	2.62	4.48	8.00	24.28	19.42	12.94	21.59	4.50	0.70	10.70	9.12	21.70	12.72	3.30	0.35	6.78	3.98	5.16	13.00	8.10
251	2.61	4.42	8.10	24.36	19.52	12.90	21.41	4.42	0.69	11.30	9.08	21.68	10.50	3.30	0.35	6.70	3.98	5.18	13.00	8.08
252	2.60	4.46	8.12	24.28	19.43	12.86	21.39	4.36	0.69	10.24	9.08	21.78	10.60	3.40	0.35	6.70	3.98	5.30	14.00	8.00

253 rows × 20 columns

Plotting all stocks

In [28]:

```python
# Timeseries for the first stock
# df['Stock1'].plot(figsize=(12,5))
df.plot()
plt.legend(ncol=7, loc="upper left")
plt.rcParams["figure.figsize"] = (15,10)
```

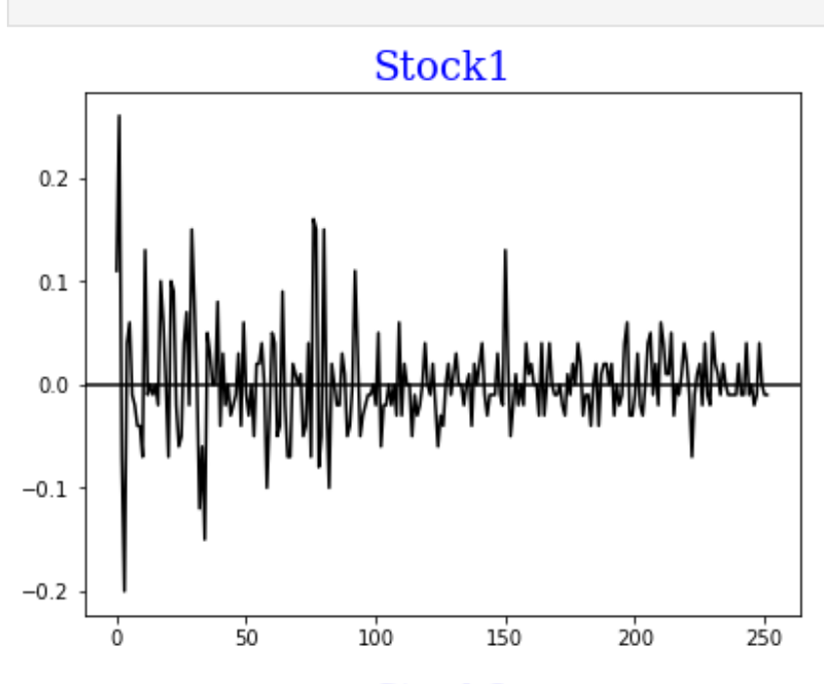


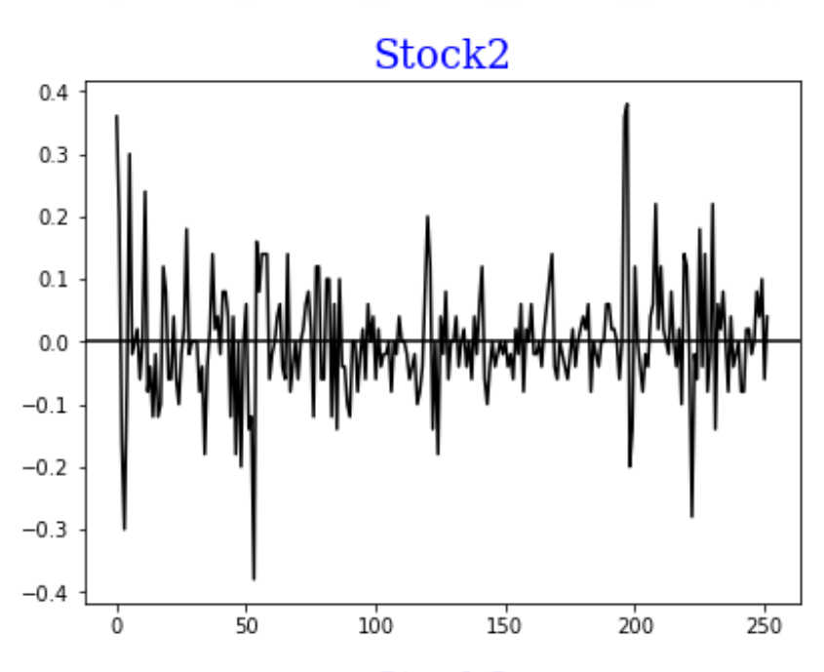
1. Plotting logarithmic returns of stocks

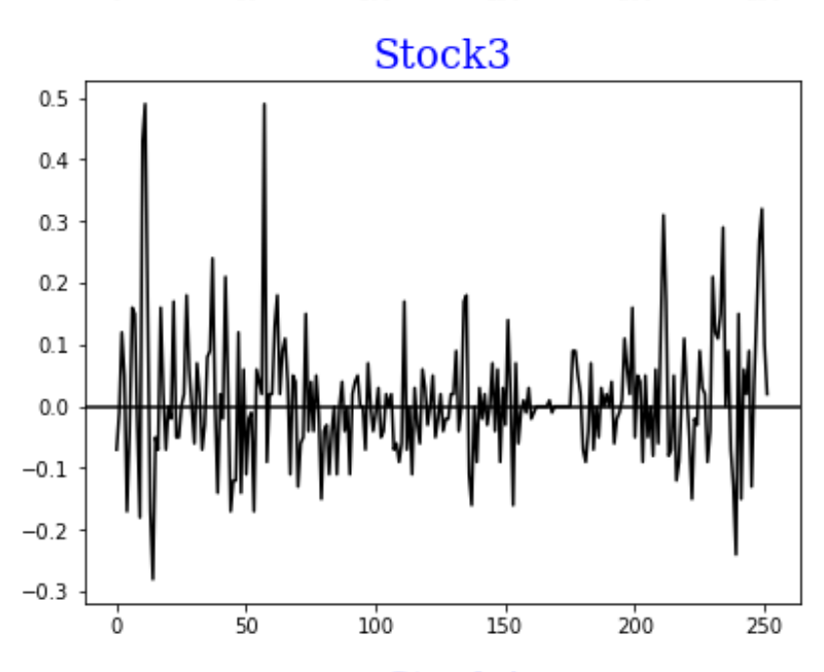
$$r(t) = \ln\left(P\_i\ (t)\right) - \ln\left(P\_i\ (t-1)\right)$$

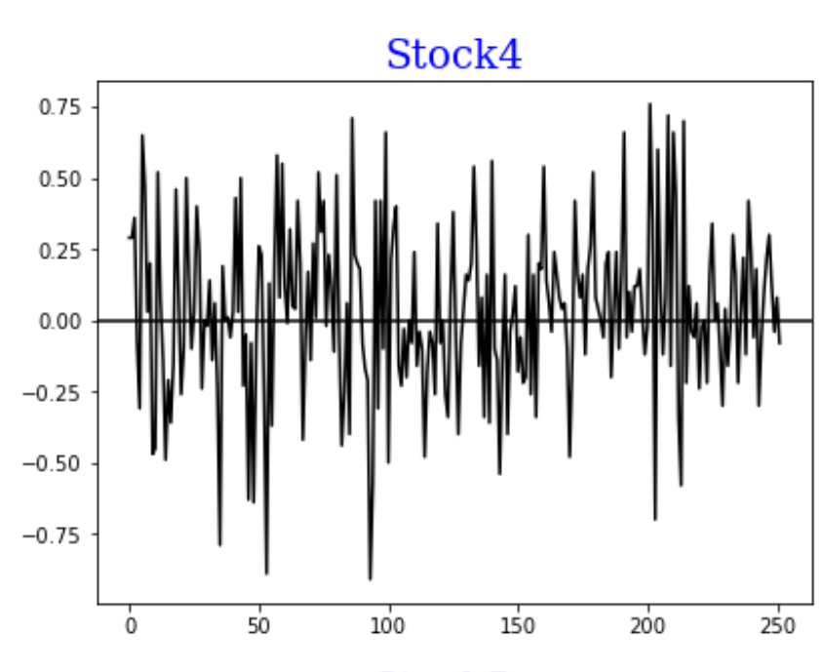
In [31]:

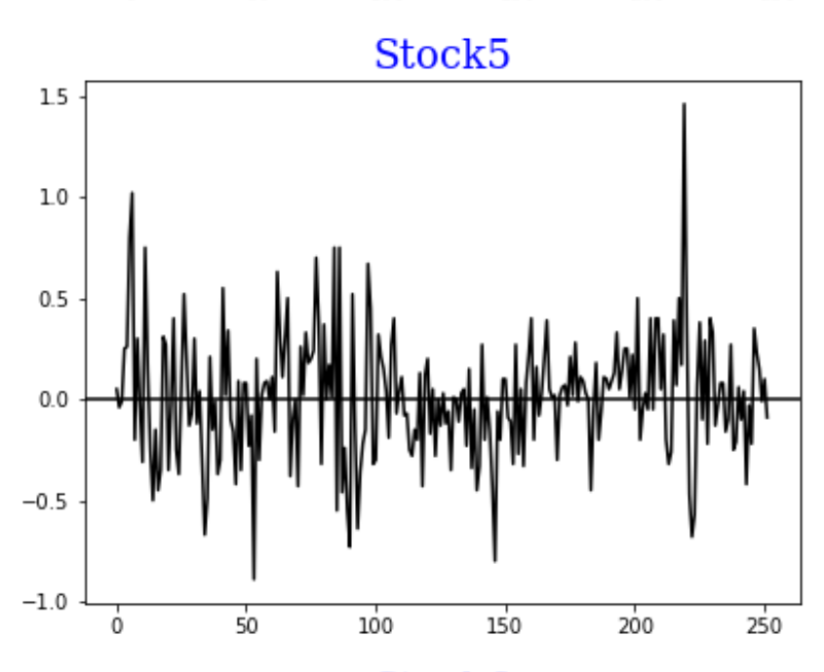
```python
# Log Returns plots
plt.rcParams["figure.figsize"] = (6.4, 4.8)
for i in range (1,21):
    temp=df['Stock'+str(i)]
    df['Stock'+str(i)]
    k=[]

    for j in range(1,len(temp)):
        k.append(temp[j]-temp[j-1])
        font1 = {'family':'serif','color':'blue','size':20}
    plt.figure()
    plt.title('Stock'+str(i), fontdict = font1)
    plt.plot(k,c='black')
    plt.axhline(y=0, c='k')
```

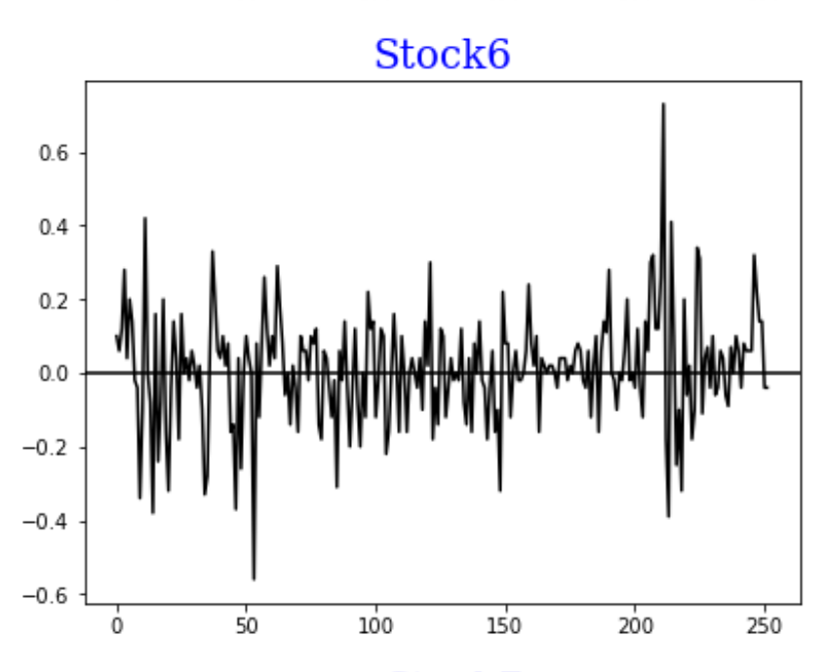


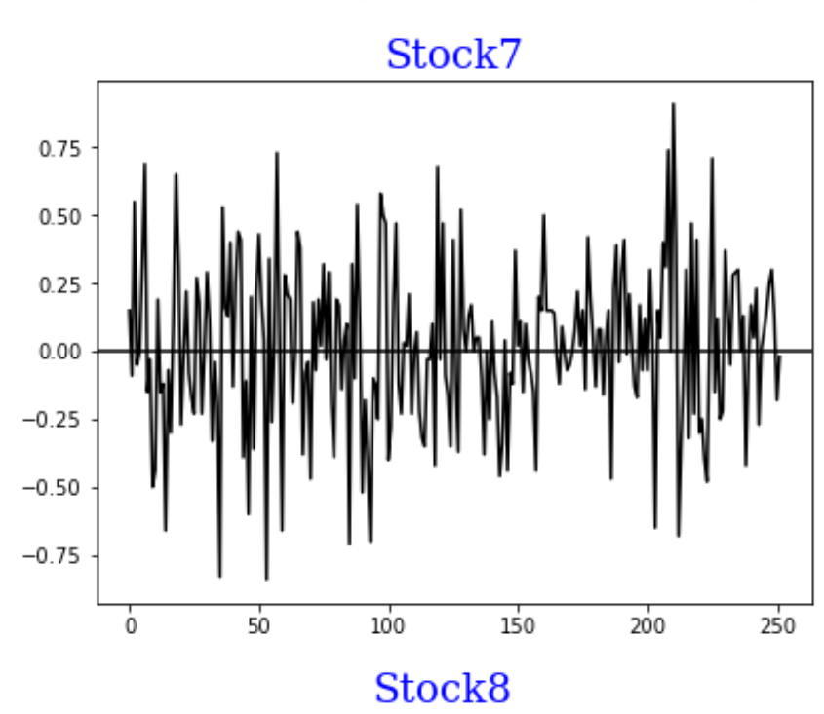


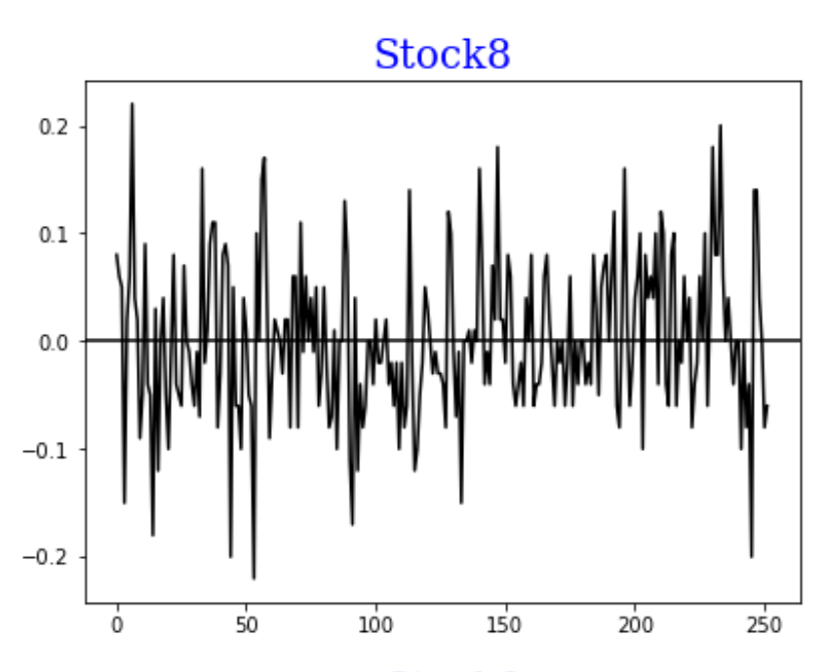


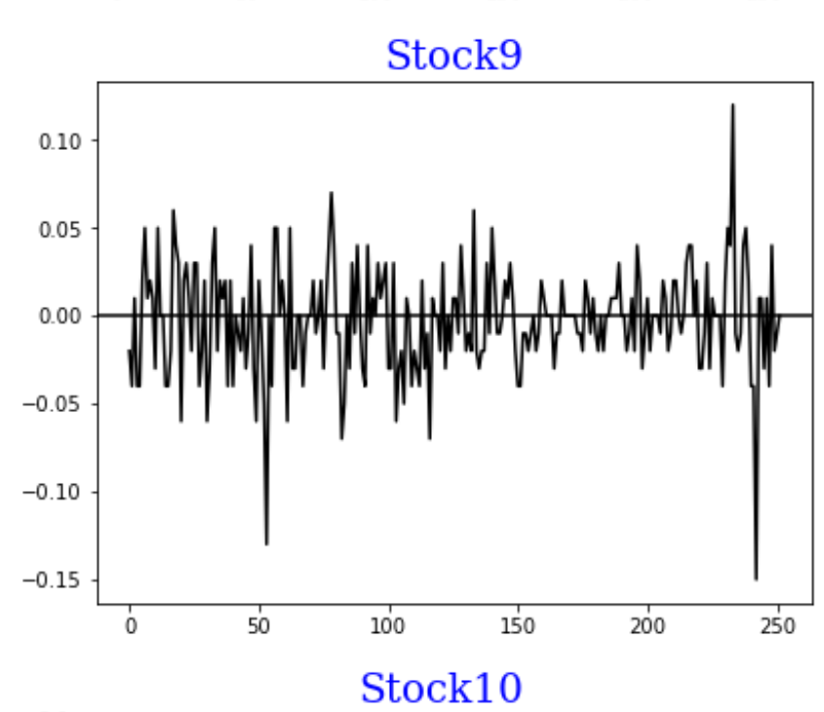


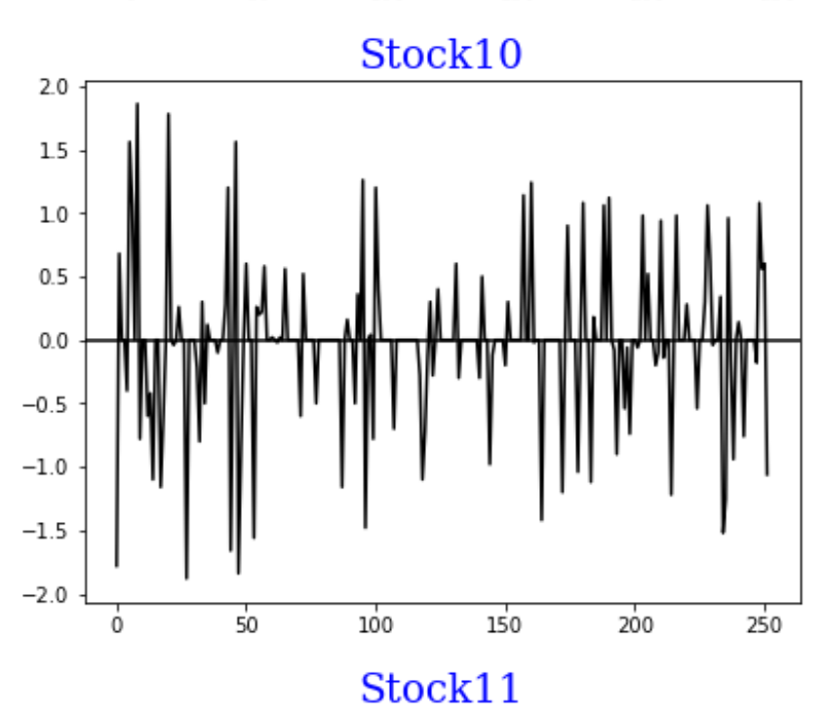


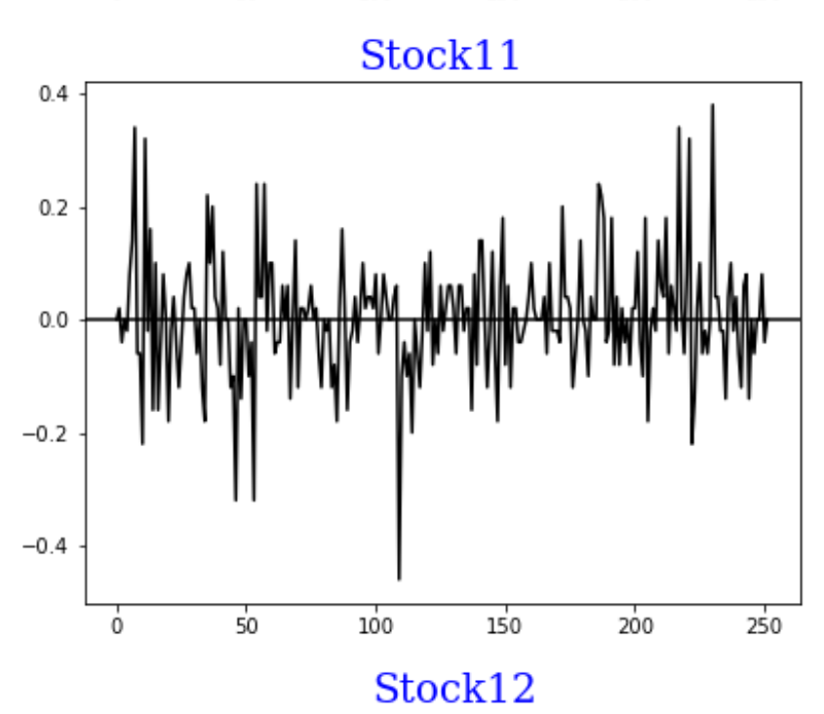


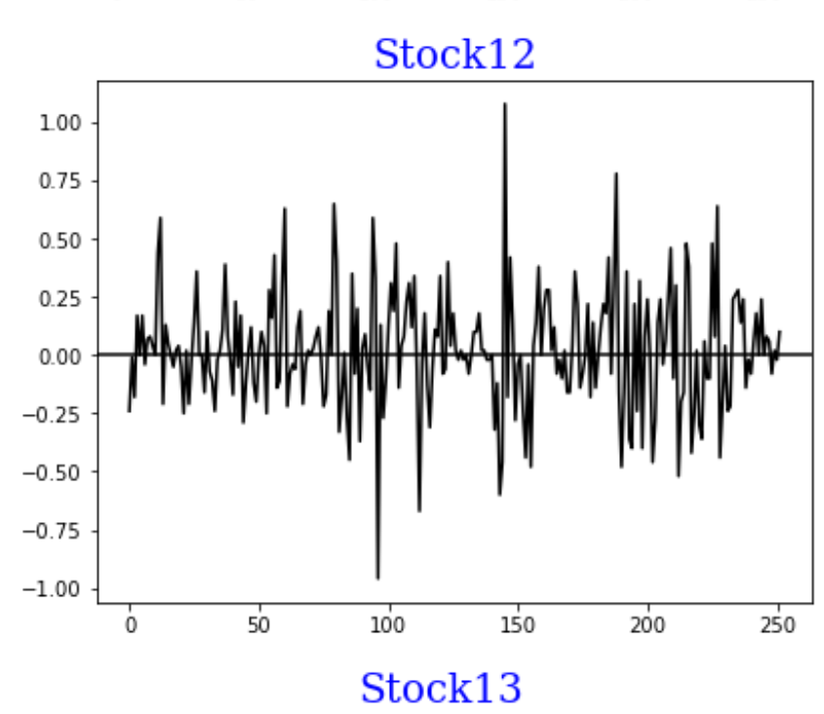


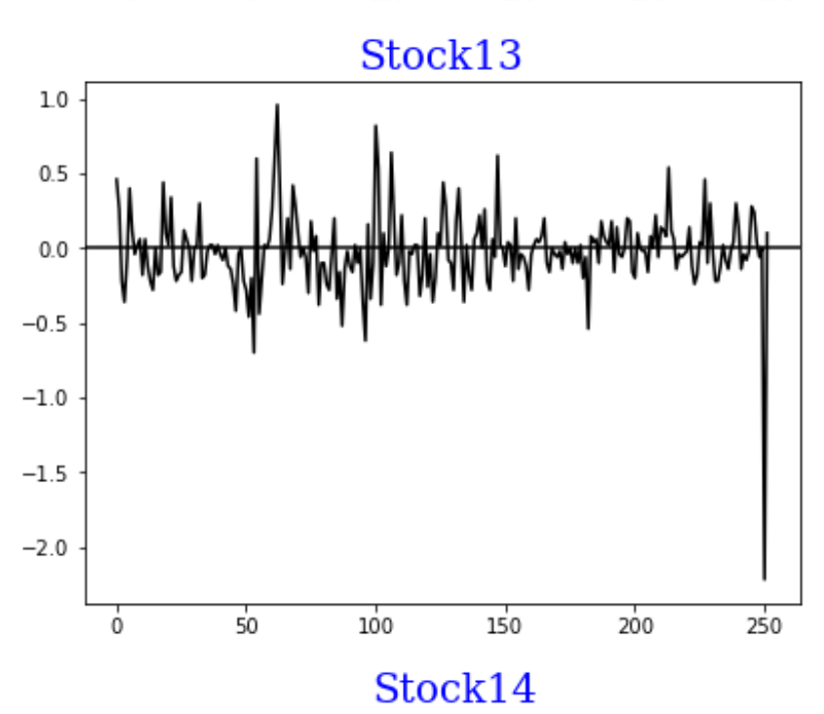


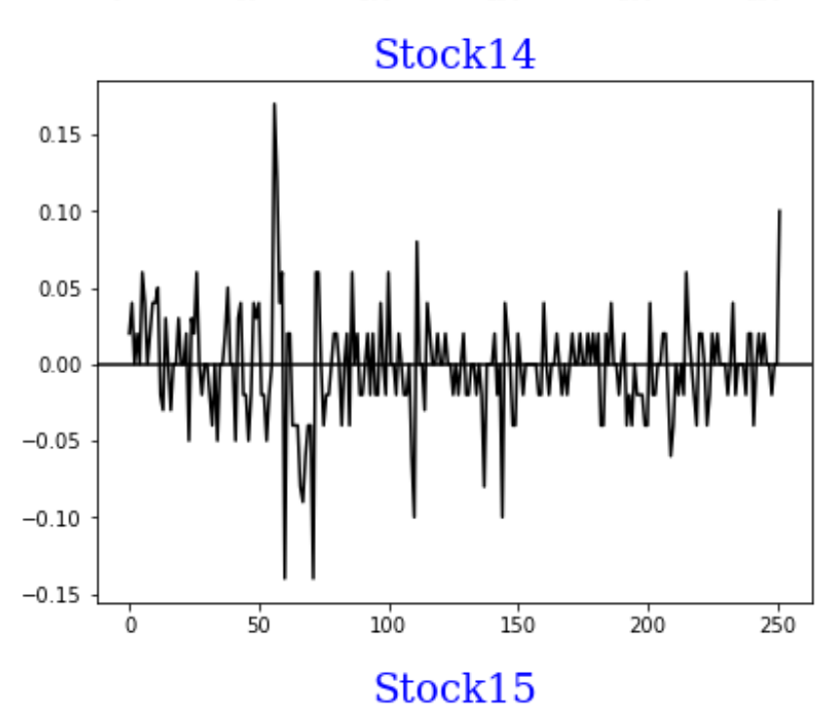


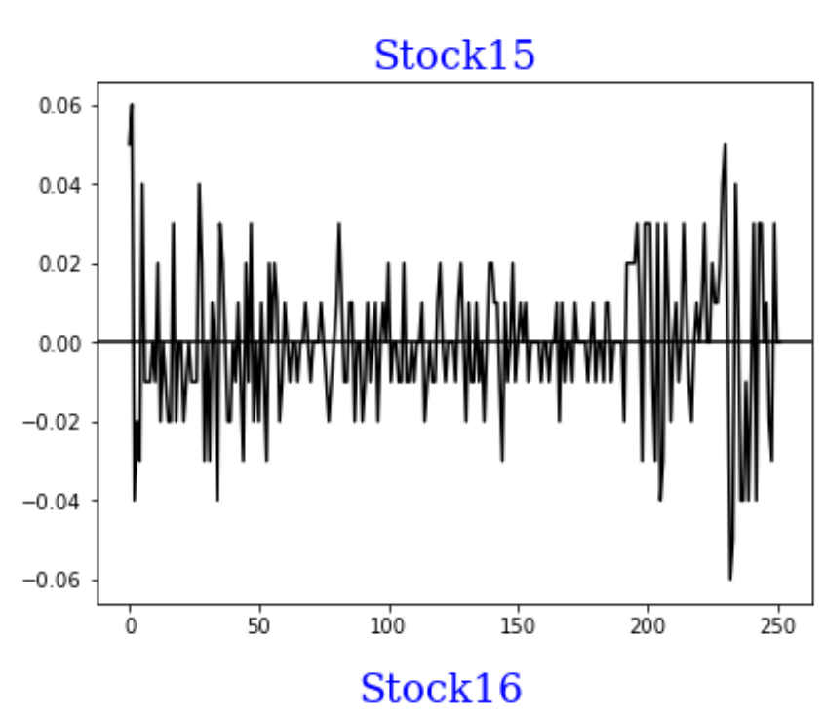


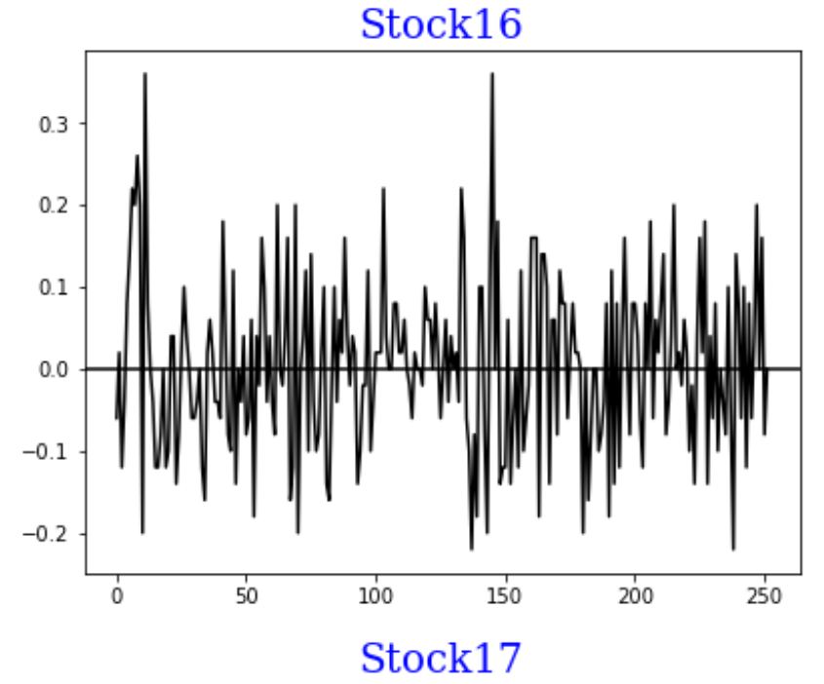


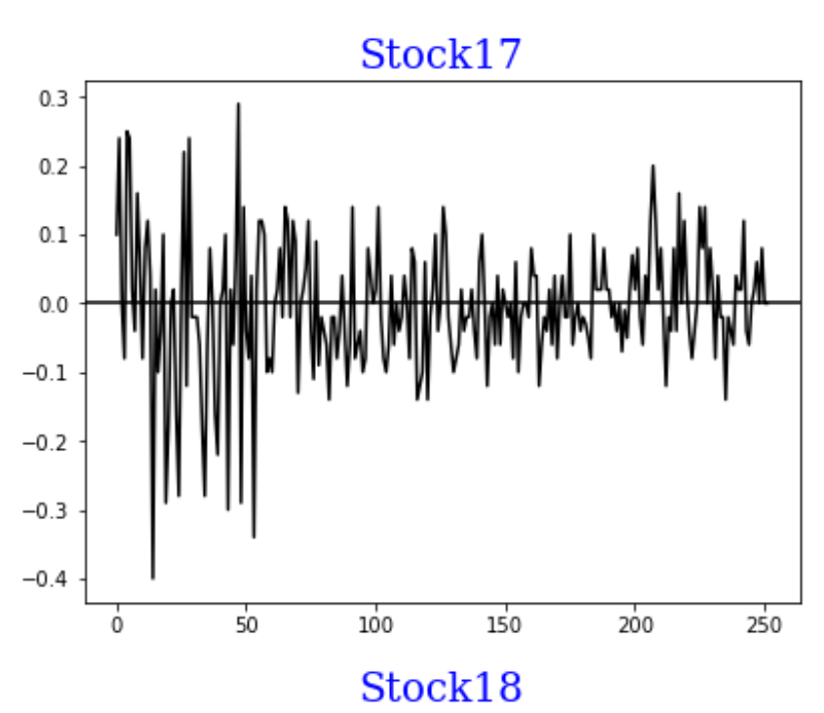


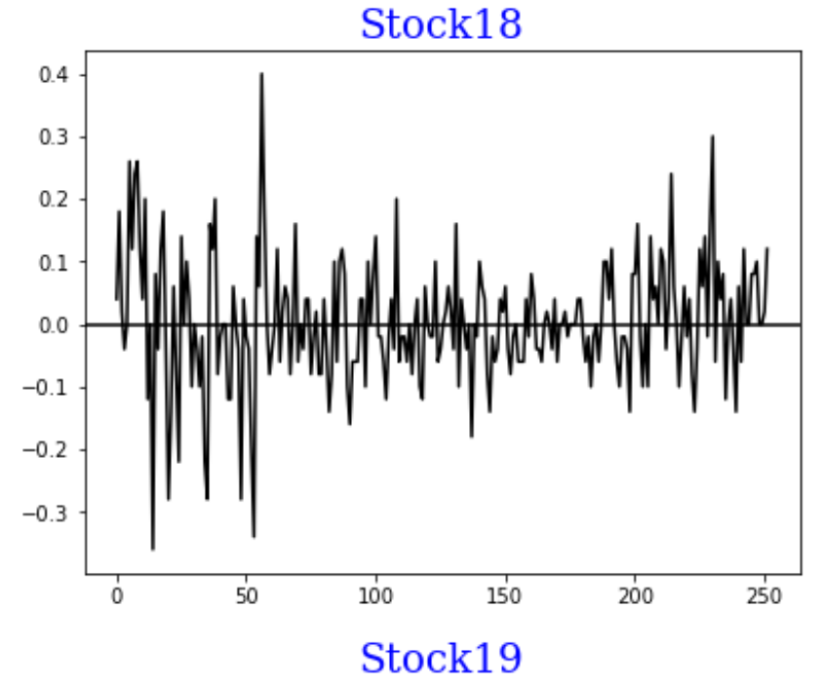


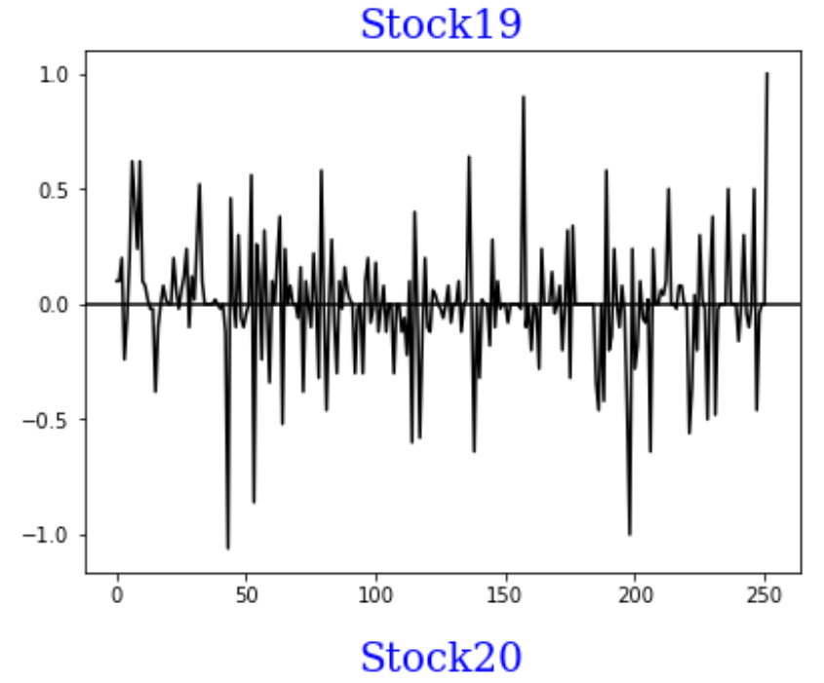


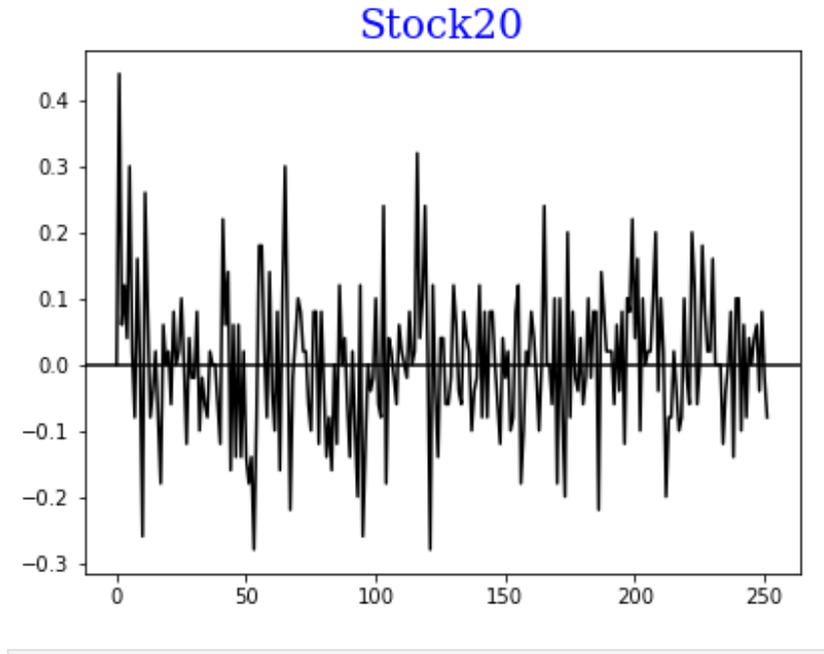












In [32]:

```python
# LogReturns for dataset
returns = pd.DataFrame()
for i in df:
    returns[i] = np.log(df[i]).diff()
returns = returns[1:]
```

In [37]:

```python
# First 5 rows to check
returns.head()
```

Out[37]:

	Stock1	Stock2	Stock3	Stock4	Stock5	Stock6	Stock7	Stock8	Stock9	Stock10	Stock11	Stock12	Stock13	Stock14	Stock15	Stock16	Stock17	Stock18	Stock19	Stock20
1	0.046917	0.079688	-0.013121	0.017656	0.002966	0.010122	0.008713	0.020152	-0.015038	-0.111996	0.000000	-0.014724	0.027729	0.005181	0.103184	-0.011606	0.017986	0.008658	0.007067	0.000000
2	0.102857	0.041673	-0.001889	0.017350	-0.002372	0.006024	-0.005219	0.014852	-0.030772	0.044278	0.002484	-0.000618	0.015340	0.010283	0.111226	0.003884	0.041891	0.038060	0.007018	0.061301
3	-0.026668	-0.033198	0.022431	0.021128	-0.000594	0.011940	0.031476	0.012210	0.007782	0.000000	-0.004975	-0.011194	-0.012964	0.000000	-0.072759	-0.023530	0.000000	0.004141	0.013889	0.008075
4	-0.080365	-0.065383	0.009200	-0.004657	0.014745	0.027319	-0.002821	-0.037087	-0.031499	0.000000	0.000000	0.010576	-0.021584	0.005102	-0.038466	-0.007968	-0.013770	-0.008299	-0.016690	0.015958
5	0.016598	-0.013606	-0.031631	-0.018252	0.015108	0.003842	0.000000	0.005025	-0.032523	-0.025808	-0.002497	0.000000	-0.004860	0.000000	-0.060625	0.015873	0.042415	0.000000	-0.004216	0.005263

In [38]:

```python
# Some usefull info
returns.describe()
```

Out[38]:

	Stock1	Stock2	Stock3	Stock4	Stock5	Stock6	Stock7	Stock8	Stock9	Stock10	Stock11	Stock12	Stock13	Stock14	Stock15	Stock16	Stock17	Stock18
count	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000
mean	0.000504	0.000108	0.001641	0.001586	0.000570	0.001066	0.000879	0.000412	-0.002634	-0.001965	0.000483	0.001121	-0.001722	-0.000493	-0.001084	0.001006	-0.001291	0.000562
std	0.019063	0.023351	0.016276	0.016112	0.017641	0.014967	0.016909	0.020100	0.033968	0.046414	0.013173	0.013085	0.020671	0.008849	0.054133	0.017995	0.021501	0.021623
min	-0.080365	-0.102516	-0.045835	-0.055866	-0.055003	-0.061288	-0.049845	-0.060127	-0.189242	-0.127527	-0.058024	-0.051624	-0.191800	-0.036086	-0.127833	-0.036905	-0.076099	-0.080804
25%	-0.008448	-0.011959	-0.007639	-0.007010	-0.010848	-0.004606	-0.009189	-0.013163	-0.027151	0.000000	-0.006519	-0.006574	-0.010996	-0.005256	-0.037922	-0.011539	-0.014118	-0.012487
50%	0.000000	0.000000	0.000000	0.001700	0.000914	0.002088	0.001071	0.000000	0.000000	0.000000	0.000000	0.000000	-0.001616	0.000000	0.000000	0.000000	0.000000	0.000000
75%	0.008386	0.010169	0.007639	0.011121	0.011403	0.009045	0.010813	0.012960	0.016294	0.000000	0.005775	0.008542	0.005518	0.005158	0.033068	0.012500	0.010963	0.012474
max	0.102857	0.099530	0.081373	0.037894	0.075723	0.062979	0.045001	0.059034	0.146603	0.113329	0.042088	0.056206	0.066323	0.040724	0.111226	0.060830	0.061395	0.090151

In [39]:

```python
# Correlation on log returns for distance matrix
returns.corr()
```

Out[39]:

	Stock1	Stock2	Stock3	Stock4	Stock5	Stock6	Stock7	Stock8	Stock9	Stock10	Stock11	Stock12	Stock13	Stock14	Stock15	Stock16	Stock17	Stock18	Stock19	Stock20
Stock1	1.000000	0.377839	0.171774	0.216850	0.221269	0.255995	0.229337	0.256582	0.125978	-0.054438	0.211700	0.065065	0.132234	0.099646	0.183430	0.159003	0.238807	0.294642	0.029851	0.207010
Stock2	0.377839	1.000000	0.277360	0.426473	0.467085	0.463014	0.411252	0.453182	0.297355	0.065245	0.364480	0.177061	0.220949	0.171380	0.321374	0.291933	0.309953	0.476994	0.089607	0.180868
Stock3	0.171774	0.277360	1.000000	0.287389	0.272800	0.381025	0.330232	0.361002	0.160443	0.045320	0.320430	0.141259	0.090682	0.188111	0.130964	0.291137	0.206442	0.345680	0.113781	0.139721
Stock4	0.216850	0.426473	0.287389	1.000000	0.521728	0.604618	0.678161	0.428515	0.256852	0.051339	0.395201	0.259782	0.167238	0.164697	0.120278	0.423767	0.263393	0.453944	0.008734	0.337458
Stock5	0.221269	0.467085	0.272800	0.521728	1.000000	0.502389	0.554488	0.307282	0.165065	0.115580	0.267480	0.188631	0.173909	0.110458	0.070095	0.415977	0.418520	0.390268	0.084001	0.285113
Stock6	0.255995	0.463014	0.381025	0.604618	0.502389	1.000000	0.640949	0.414285	0.247489	0.038199	0.417388	0.214264	0.199881	0.135685	0.135823	0.336827	0.359402	0.487185	-0.002956	0.259426
Stock7	0.229337	0.411252	0.330232	0.678161	0.554488	0.640949	1.000000	0.475594	0.204573	0.039390	0.381230	0.211911	0.164529	0.142993	0.121317	0.325456	0.315053	0.455064	0.063936	0.260097
Stock8	0.256582	0.453182	0.361002	0.428515	0.307282	0.414285	0.475594	1.000000	0.345534	0.130136	0.354568	0.186186	0.214405	0.178222	0.240717	0.330682	0.286723	0.476378	0.074344	0.221328
Stock9	0.125978	0.297355	0.160443	0.256852	0.165065	0.247489	0.204573	0.345534	1.000000	0.092303	0.246101	0.112208	0.031189	0.137983	0.038295	0.160354	0.100330	0.212269	0.048774	-0.000163
Stock10	-0.054438	0.065245	0.045320	0.051339	0.115580	0.038199	0.039390	0.130136	0.092303	1.000000	0.023142	0.087340	0.019197	0.151407	-0.002496	0.008427	0.129203	0.161440	-0.011897	0.142851
Stock11	0.211700	0.364480	0.320430	0.395201	0.267480	0.417388	0.381230	0.354568	0.246101	0.023142	1.000000	0.154793	0.173781	0.202775	0.184302	0.313174	0.286097	0.426041	0.061622	0.117991
Stock12	0.065065	0.177061	0.141259	0.259782	0.188631	0.214264	0.211911	0.186186	0.112208	0.087340	0.154793	1.000000	0.135647	0.139284	0.121475	0.359775	0.185530	0.231858	0.091630	0.245859
Stock13	0.132234	0.220949	0.090682	0.167238	0.173909	0.199881	0.164529	0.214405	0.031189	0.019197	0.173781	0.135647	1.000000	0.088221	0.133373	0.240060	0.218746	0.214153	0.121640	0.142896
Stock14	0.099646	0.171380	0.188111	0.164697	0.110458	0.135685	0.142993	0.178222	0.137983	0.151407	0.202775	0.139284	0.088221	1.000000	0.061661	0.266222	0.111222	0.321123	0.079039	0.152063
Stock15	0.183430	0.321374	0.130964	0.120278	0.070095	0.135823	0.121317	0.240717	0.038295	-0.002496	0.184302	0.121475	0.133373	0.061661	1.000000	0.092444	0.151881	0.312703	0.079372	0.114221
Stock16	0.159003	0.291933	0.291137	0.423767	0.415977	0.336827	0.325456	0.330682	0.160354	0.008427	0.313174	0.359775	0.240060	0.266222	0.092444	1.000000	0.364853	0.397461	0.150776	0.315896
Stock17	0.238807	0.309953	0.206442	0.263393	0.418520	0.359402	0.315053	0.286723	0.100330	0.129203	0.286097	0.185530	0.218746	0.111222	0.151881	0.364853	1.000000	0.491345	0.154533	0.270237
Stock18	0.294642	0.476994	0.345680	0.453944	0.390268	0.487185	0.455064	0.476378	0.212269	0.161440	0.426041	0.231858	0.214153	0.321123	0.312703	0.397461	0.491345	1.000000	0.191529	0.331595
Stock19	0.029851	0.089607	0.113781	0.008734	0.084001	-0.002956	0.063936	0.074344	0.048774	-0.011897	0.061622	0.091630	0.121640	0.079039	0.079372	0.150776	0.154533	0.191529	1.000000	0.029188
Stock20	0.207010	0.180868	0.139721	0.337458	0.285113	0.259426	0.260097	0.221328	-0.000163	0.142851	0.117991	0.245859	0.142896	0.152063	0.114221	0.315896	0.270237	0.331595	0.029188	1.000000

2. Plotting correlation matrix for the data set

$$\rho_{ij} = \frac{\langle r_i r_j\rangle - \langle r_i\rangle\langle r_j\rangle}{\sqrt{\left(\langle r_i^2\rangle - \langle r_i\rangle^2\right)\left(\langle r_j^2\rangle - \langle r_j\rangle^2\right)}}$$

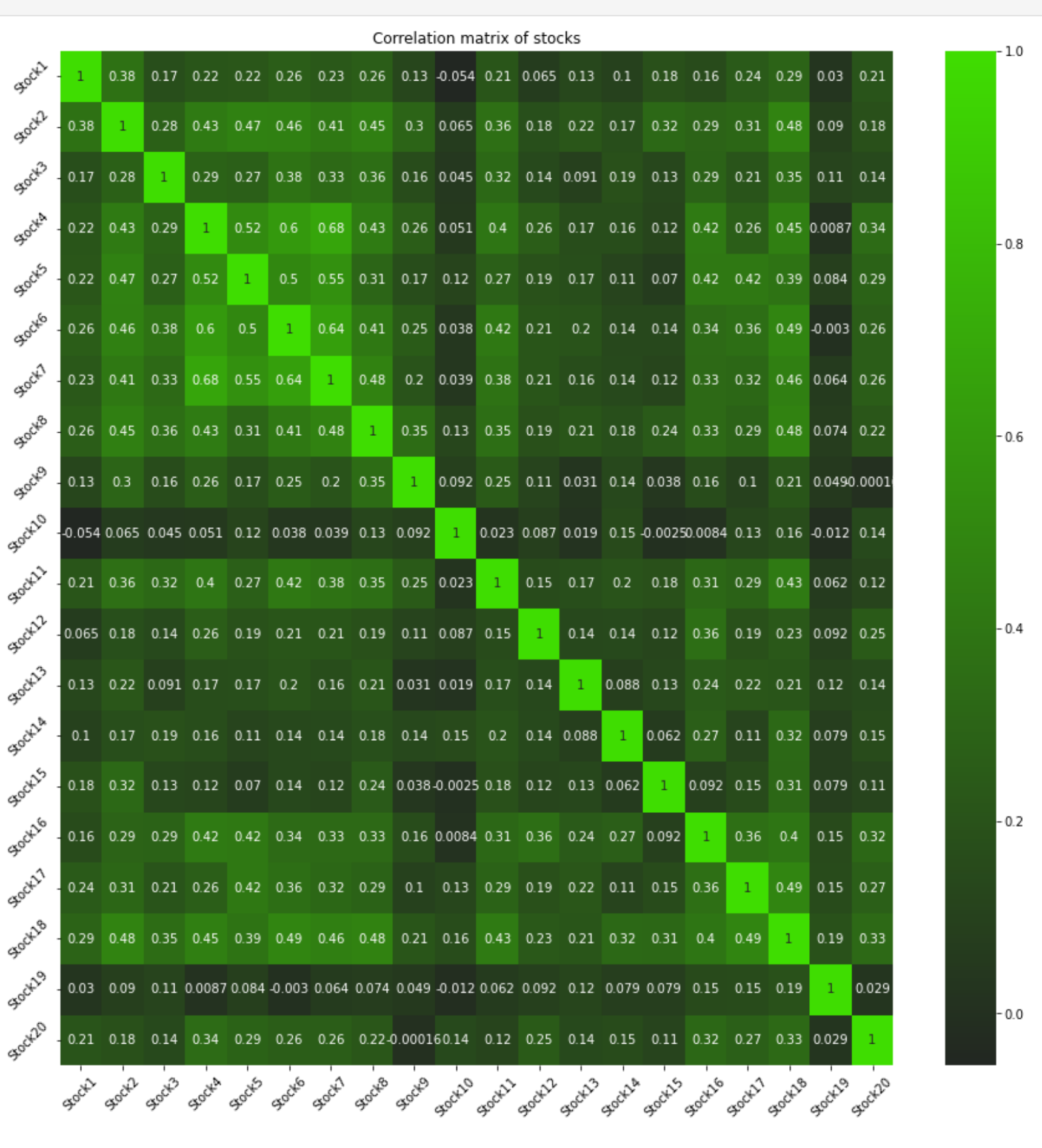
In the 2nd part of the exercise, the correlations between the stocks had to be calculated. These correlations are shown as a heatmap. We can use this heatmap to easily distinguish some elements between the stocks, such as e.g. that Stock10 and Stock19 show a very reduced correlation compared to the rest, as can be seen from their dark color. At the same time, the first 8 stocks seem to show an increased correlation between them, especially Stocks5-7.

In [40]:

```python
# Dataset for crosscorrelation matrix
corrMatrix = returns.corr()
```

In [57]:

```python
#Correlation matrix of dataset
fig, ax = plt.subplots(figsize=(15,15))
# sn.heatmap(corrMatrix, annot=True)
cmap = sn.dark_palette('#3fdd01',as_cmap=True)
map = sn.heatmap(corrMatrix,annot=True,cmap=cmap)
plt.yticks(rotation=45)
plt.xticks(rotation=45)
plt.title("Correlation matrix of stocks")
plt.show()
```



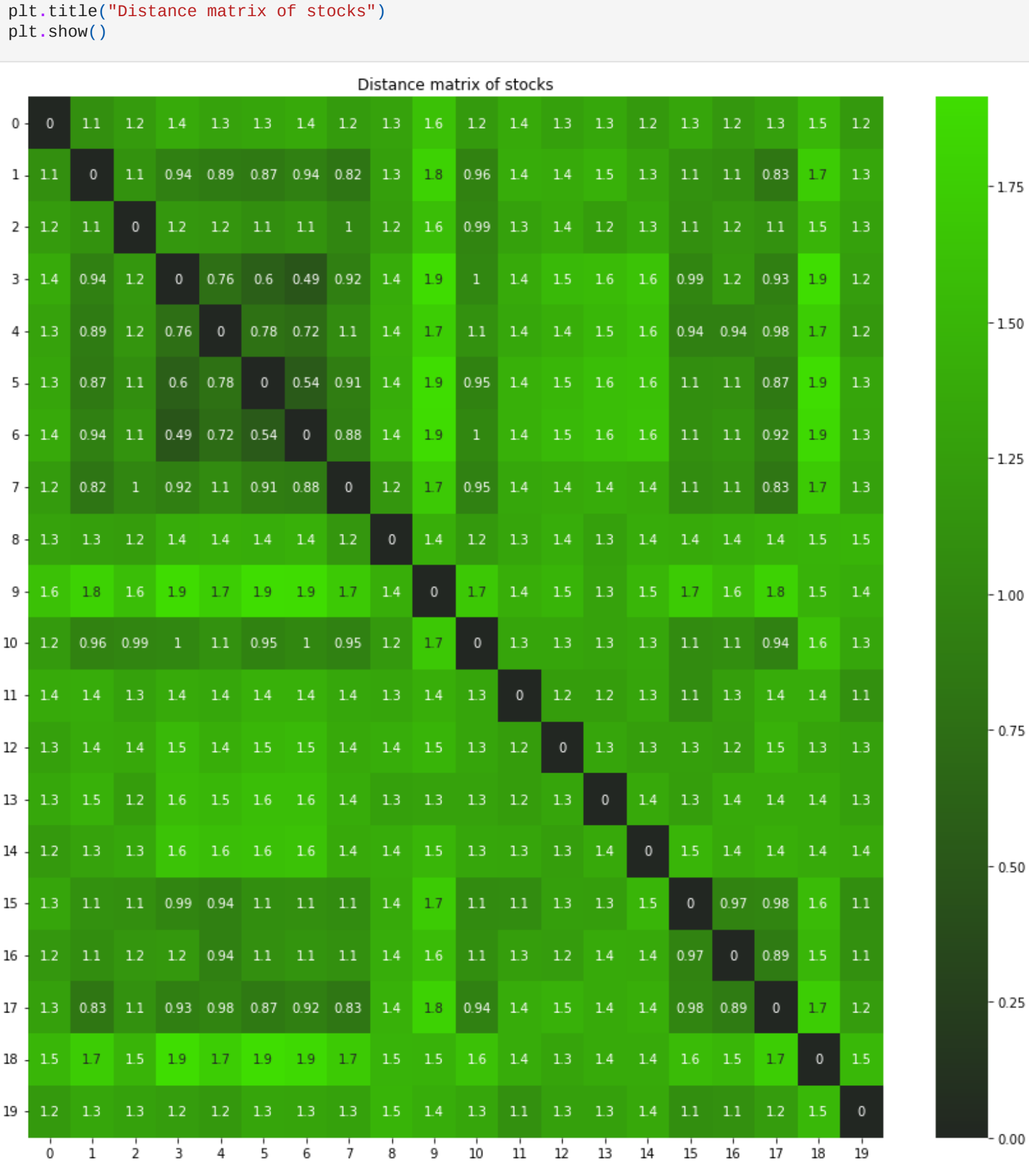
3. Plotting distance matrix for the data set

$$d_{ij} = \sqrt{2\left(1 - \rho_{ij}\right)}$$

In the third part of the exercise, we calculate the distance between the shares which can be seen below. As expected the matrix seems to be the exact opposite to the correlations matrix. The values ranges from 0 to almost 2 depending on the distance the shares have to each other.

In [119…

```python
A=squareform(pdist(corrMatrix.loc[['Stock1','Stock2','Stock3','Stock4','Stock5','Stock6','Stock7','Stock8','Stock9',
                                   'Stock10','Stock11','Stock12','Stock13','Stock14','Stock15','Stock16','Stock17',
                                   'Stock18',
                                  'Stock19','Stock20']]))
fig, ax = plt.subplots(figsize=(15,15))
cmap = sn.dark_palette('#3fdd01',as_cmap=True)
map = sn.heatmap(A,annot=True,cmap=cmap)
plt.yticks(rotation=360)
# plt.xticks(rotation=45)
plt.title("Distance matrix of stocks")
plt.show()
```



In [ ]: