Section 3: Correlation in Stocks

In this lesson, the correlation between data of different companies is discussed.

We'll cover the following

The stock behavior of companies dealing in similar services is usually related, and this relation can be measured using correlation.

Correlation

Correlation is a statistical technique that determines how strongly two variables are related to each other and how a change in one would affect the other. It can also be defined as a measure of dependence between two or more quantities.

The two types of correlation, in terms of stock behavior, can be described as follows:

- **Positive correlation**: The stock value of one company goes up, and in correlation with it, the stock values of other companies also go up.
- **Negative correlation**: The stock value of one company goes up, and in correlation with it, the stock values of other companies go down.

Correlation score

For positive correlation, this score is between **0** and **1**, and for negative correlation, this score is between **-1** and **0** (inclusive). A strong positive correlation has a score above **0.4**. It is the same for the negative correlation; it's strong below a **-0.4** value. A score of **1** represents a perfect positive relationship and usually occurs when the correlation is taken with itself.

Correlation for all companies will be calculated with each other to observe what kind of relationship exists between the data of different companies and to see if there is a strong or weak correlation. The **closing price** and **daily return** variables will be used as parameters to get the correlation score for every company.

Before starting, all the company files need to be read in as variables, and their `Time` column needs to be set as the index. The following piece of code does the preprocessing before we actually observe the correlation.

```
import numpy as np
from scipy import stats
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

sys = pd.read_csv('Year_2018/SYS.csv')
ns = pd.read_csv('Year_2018/NETSOL.csv')
ptc = pd.read_csv('Year_2018/PTC.csv')
avn = pd.read_csv('Year_2018/AVN.csv')

sys['Time'] = pd.to_datetime(sys.Time)
ns['Time'] = pd.to_datetime(ns.Time)
ptc['Time'] = pd.to_datetime(ptc.Time)
avn['Time'] = pd.to_datetime(avn.Time)

sys = sys.set_index('Time')
ns = ns.set_index('Time')
ptc = ptc.set_index('Time')
avn = avn.set_index('Time')

## Try printing the data of any company
```

Now that the preprocessing is done and all the data is ready, the correlation can be found between these companies stock behavior. As mentioned above, the **closing price** and **daily return** values will be used to find this correlation.

Correlation with the closing price

The `corr()` function of a `DataFrame` can easily find correlations between its columns. So first, the *closing prices* of all the companies need to be in one `DataFrame`, on which the `corr()` function will be applied.

```
df = pd.DataFrame({'SYS': sys['Close'],
                   'NETSOL': ns['Close'],
                   'PTC': ptc['Close'],
                   'AVN': avn['Close']})
```

```
print("The New DataFrame\n", df.head())

corr = (df.dropna()).corr() # Calculating correlation after dropping null values

print("The correlations\n", corr)
```

On **line 1-4**, a new `DataFrame` is created using the dictionary method, which takes the `Close` column of all four companies.

On **line 8**, the `corr()` function is used to find a correlation between the columns.

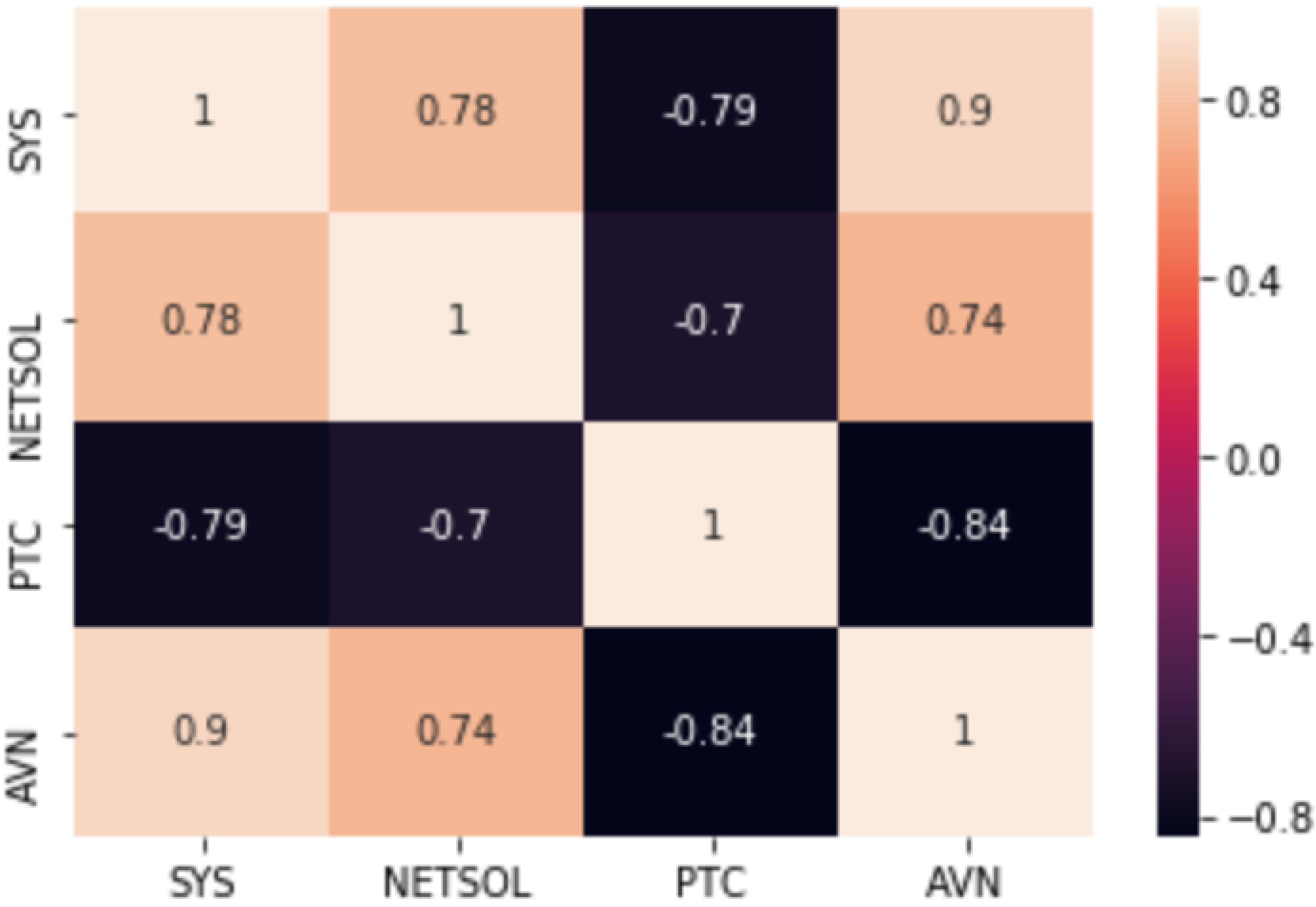
The output shows that the correlation values of all the companies, with respect to the closing price, have been calculated against each other. Also, we can observe that the correlation of the data of a company with itself is **1**. The values below zero represent *negative correlation*, and the values above zero represent *positive correlation*.

Let's visualize these values with a *heatmap*.

```
df = pd.DataFrame({'SYS': sys['Close'],
                   'NETSOL': ns['Close'],
                   'PTC': ptc['Close'],
                   'AVN': avn['Close']})

corr = (df.dropna()).corr()

sns.heatmap(corr, annot = True) # Plotting heatmap
```



On **line 8**, the `heatmap` function of the seaborn package is used to plot correlation values on a *heatmap* with the `annot=True` parameter as discussed in this lesson.

The correlation values are properly separated and easy to comprehend on a *heatmap*. The highest correlation value is of **Systems Ltd** and **Avanceon**, which is **0.9**. This means that these two companies are highly correlated in a positive way. So, if the stock value of one company goes up, the value of the other company also goes up and vice versa.

> **Try to infer more important information about other companies as well!**

Correlation with daily returns

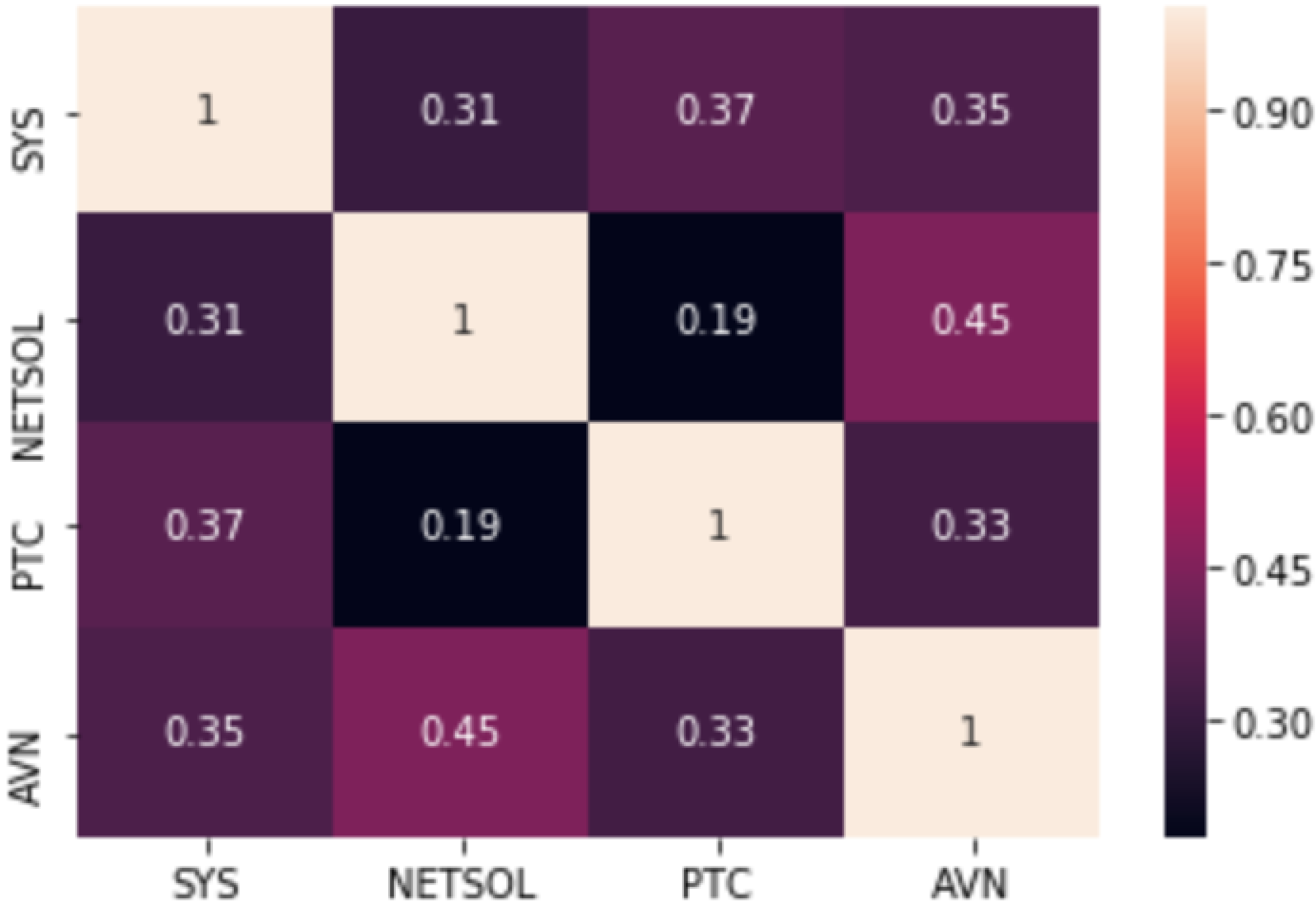
The same `corr()` function and technique are used here, but instead of the *closing price*, the correlation is calculated on *daily returns*.

```
df = pd.DataFrame({'SYS': sys['Close'],
                   'NETSOL': ns['Close'],
                   'PTC': ptc['Close'],
```

```
                  'PTC': ptc['Close'],
                  'AVN': avn['Close']})


all_returns = df.pct_change()
print(all_returns)

corr = (all_returns.dropna()).corr()
sns.heatmap(corr, annot = True)
```



On **line 6**, the `pct_change` function is used to calculate daily returns just like before. Each column is the *closing price* of the companies, so, it calculates the daily returns for all companies in one go.

The daily return values can be viewed by clicking the right button on the output panel.

On **line 10**, the same `heatmap` function of seaborn is used to plot correlation values on a *heatmap*.

From this correlation, we can infer that **NETSOL** and **Avanceon** are correlated in a positive way. Because the score is above **0.4**, it is a strong correlation.

positive way. Because the score is above **0.4**, it is a strong correlation.

In the next lesson, we estimate how much risk is associated with the stock of each company.