

Exercise 11: Hierarchies of stocks

Previously, you used k-means clustering to cluster companies according to their stock price movements. This time, perform *hierarchical* clustering of the companies. You are given a NumPy array of price movements `movements`, where the rows correspond to companies, and a list of the company names `companies`.

SciPy hierarchical clustering doesn't fit into a sklearn pipeline, so you'll need to use the `normalize()` function from `sklearn.preprocessing` instead of `Normalizer`.

From the course *Transition to Data Science*. [Buy the entire course for just \\$10](#) for many more exercises and helpful video lectures.

Step 1: Load the data (*written for you*)

```
In [1]: import pandas as pd

fn = '../datasets/company-stock-movements-2010-2015-incl.csv'
stocks_df = pd.read_csv(fn, index_col=0)

companies = list(stocks_df.index)
movements = stocks_df.values
```

Step 2: Make the necessary imports:

- `normalize` from `sklearn.preprocessing`.
- `linkage` and `dendrogram` from `scipy.cluster.hierarchy`.
- `matplotlib.pyplot` as `plt`.

Step 3: Rescale the price movements for each stock by using the `normalize()` function on `movements`.

```
In [3]: normalized_movements = normalize(movements)
```

Step 4: Apply the `linkage()` function to `normalized_movements`, using `'complete'` linkage, to calculate the hierarchical clustering. Assign the result to `mergings`.

```
In [4]: mergings = linkage(normalized_movements, method='complete')
```

Step 5: Plot a dendrogram of the hierarchical clustering, using the list `companies` of company names as the labels. In addition, specify the `leaf_rotation=90`, and `leaf_font_size=10` keyword arguments as you did in the previous exercise.

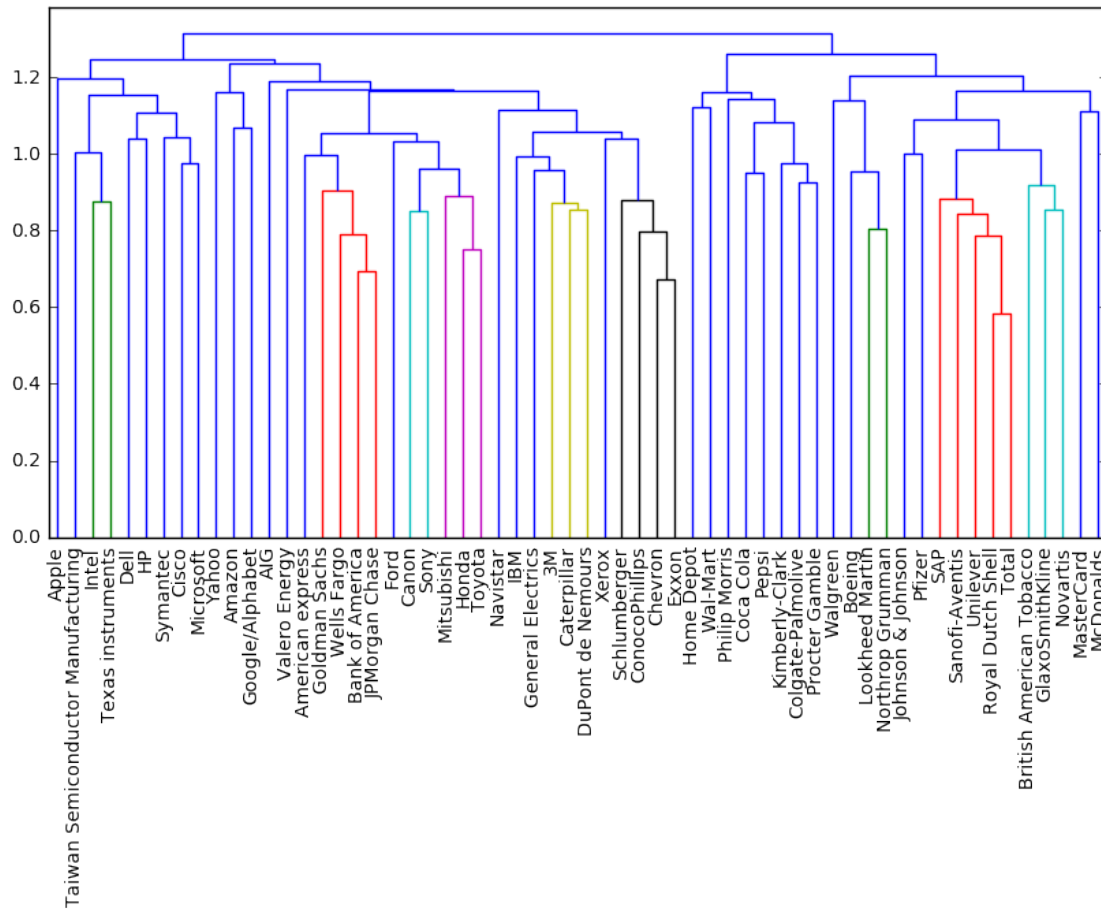
```
In [5]: # the first line is written for you - it sets the size of your plot
plt.figure(figsize=(10, 5))

dendrogram(
    mergings,
```

```

labels=companies,
leaf_rotation=90.,
leaf_font_size=10
)
plt.show()

```



In []: