Exercise 8: Clustering stocks using KMeans

In this exercise, you'll cluster companies using their daily stock price movements (i.e. the dollar difference between the closing and opening prices for each trading day). You are given a NumPy array `movements` of daily price movements from 2010 to 2015, where each row corresponds to a company, and each column corresponds to a trading day.

Some stocks are more expensive than others. To account for this, include a `Normalizer` at the beginning of your pipeline. The Normalizer will separately transform each company's stock price to a relative scale before the clustering begins.

Normalizer vs StandardScaler

Note that `Normalizer()` is different to `StandardScaler()`, which you used in the previous exercise. While `StandardScaler()` standardizes **features** (such as the features of the fish data from the previous exercise) by removing the mean and scaling to unit variance, `Normalizer()` rescales **each sample** - here, each company's stock price - independently of the other.

This dataset was obtained from the Yahoo! Finance API.

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 [ ]: import pandas as pd

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

Step 2: Inspect the first few rows of the DataFrame `stocks_df` by calling its `head()` function.

```
In [ ]:
```

Step 3: Extract the NumPy array `movements` from the DataFrame and the list of company names (*written for you*)

```
In [ ]: companies = list(stocks_df.index)
        movements = stocks_df.values
```

Step 4: Make the necessary imports:

- `Normalizer` from `sklearn.preprocessing`.
- `KMeans` from `sklearn.cluster`.
- `make_pipeline` from `sklearn.pipeline`.

```
In [ ]:
```


Step 3: Create an instance of `Normalizer` called `normalizer`.

```
In [ ]:
```

Step 4: Create an instance of `KMeans` called `kmeans` with `14` clusters.

```
In [ ]:
```

Step 5: Using `make_pipeline()`, create a pipeline called `pipeline` that chains `normalizer` and `kmeans`.

```
In [ ]:
```

Step 6: Fit the pipeline to the `movements` array.

```
In [ ]:
```

In the next exercise: Let's check out your clustering!

