Exercise 6: Scaling fish data for clustering

You are given an array `samples` giving measurements of fish. Each row represents asingle fish. The measurements, such as weight in grams, length in centimeters, and the percentage ratio of height to length, have very different scales. In order to cluster this data effectively, you'll need to standardize these features first. In this exercise, you'll build a pipeline to standardize and cluster the data.

This great dataset was derived from the one here, where you can see a description of each measurement.

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 dataset *(this bit is written for you)*.

```
In [2]: import pandas as pd

        df = pd.read_csv('datasets/fish.csv')

        # forget the species column for now - we'll use it later!
        del df['species']
```

Step 2: Call `df.head()` to inspect the dataset:

```
In [3]: df.head()
```

```
Out[3]:    weight  length1  length2  length3  height  width
        0   242.0     23.2     25.4     30.0    38.4   13.4
        1   290.0     24.0     26.3     31.2    40.0   13.8
        2   340.0     23.9     26.5     31.1    39.8   15.1
        3   363.0     26.3     29.0     33.5    38.0   13.3
        4   430.0     26.5     29.0     34.0    36.6   15.1
```

Step 3: Extract all the measurements as a 2D NumPy array, assigning to `samples` (hint: use the `.values` attribute of `df`)

```
In [3]: samples = df.values
```

Step 4: Perform the necessary imports:

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

```
In [4]: from sklearn.pipeline import make_pipeline
        from sklearn.preprocessing import StandardScaler
        from sklearn.cluster import KMeans
```


Step 5: Create an instance of `StandardScaler` called `scaler`.

```
In [5]: scaler = StandardScaler()
```

Step 6: Create an instance of `KMeans` with `4` clusters called `kmeans`.

```
In [6]: kmeans = KMeans(n_clusters=4)
```

Step 7: Create a pipeline called `pipeline` that chains `scaler` and `kmeans`. To do this, you just need to pass them in as arguments to `make_pipeline()`.

```
In [10]: pipeline = make_pipeline(scaler, kmeans)
```

Great job! Now you're all set to transform the fish measurements and perform the clustering. Let's get to it in the next exercise!

```
In [ ]:
```

