18.6.3

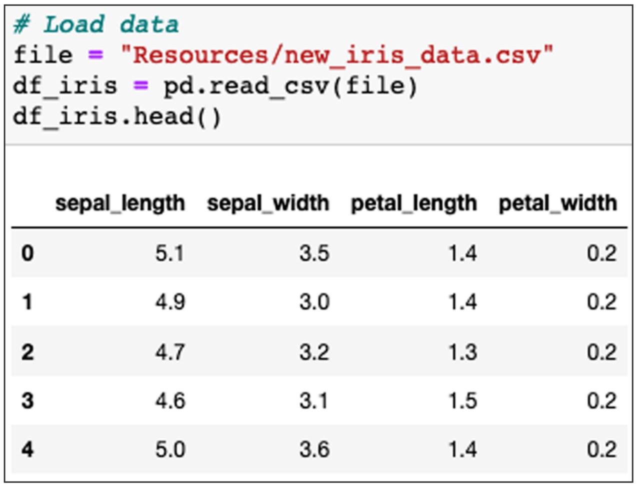
## Running Hierarchical Clustering

**The** two of you are curious to see what hierarchical clustering has to offer over K-means and decide to test it out on the iris dataset.

Open a notebook and enter the following code to import our libraries. (Most of these should look familiar by this point, with the only new one being the AgglomerativeClustering library, the hierarchical clustering algorithm that will replace K-means.):

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import AgglomerativeClustering
import hvplot.pandas
```

Enter the code to load in the iris dataset:



```
# Load data
file = "Resources/new_iris_data.csv"
df_iris = pd.read_csv(file)
df_iris.head()
```

|   | sepal_length | sepal_width | petal_length | petal_width |
|---|---|---|---|---|
| **0** | 5.1 | 3.5 | 1.4 | 0.2 |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 |

**SKILL DRILL**

Apply PCA to reduce the dataset from four features to two.

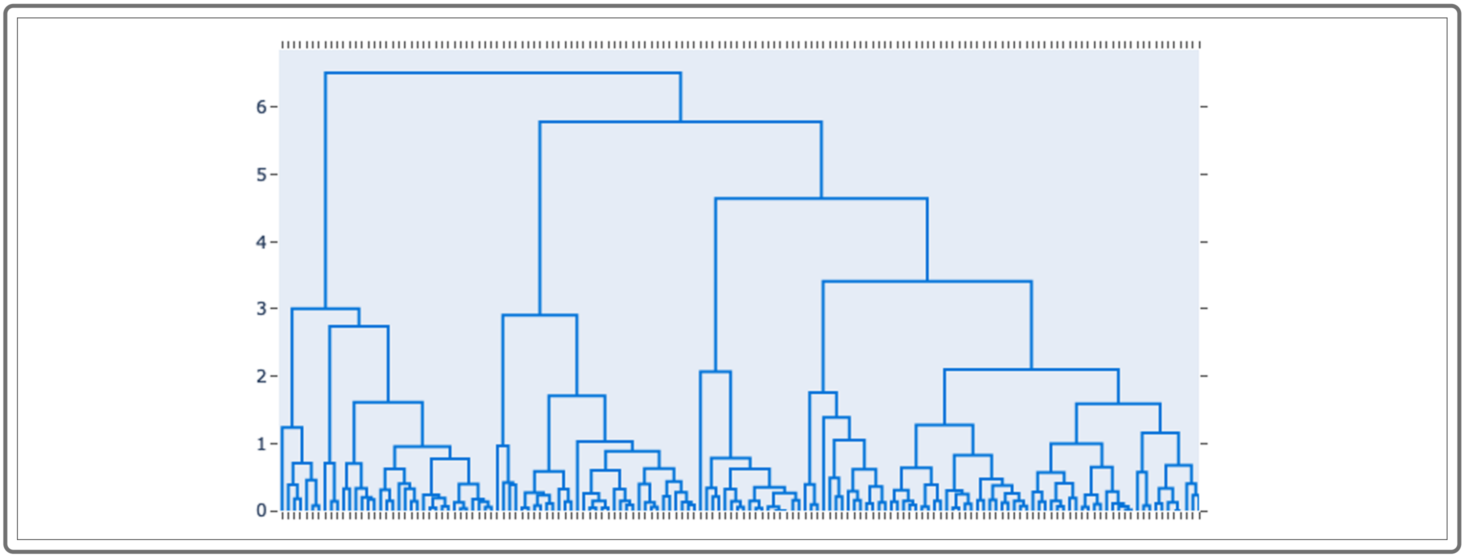
After PCA has been applied, it is time to run the hierarchical clustering algorithm. We start by creating a dendrogram. Enter the code to import the libraries:

```
import plotly.figure_factory as ff
```

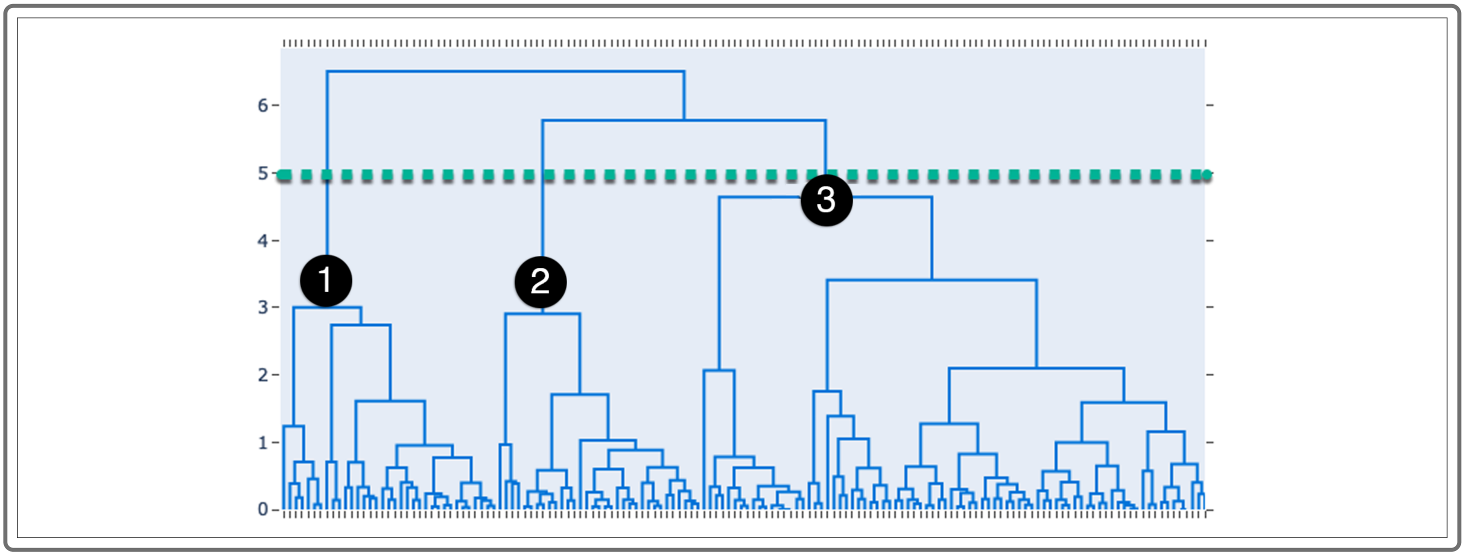
Enter the code to create a dendrogram. We'll pass a `color_threshold` of 0 to make all the branches the same color:

```
# Create the dendrogram
fig = ff.create_dendrogram(df_iris_pca, color_threshold=0)
fig.update_layout(width=800, height=500)
fig.show()
```

The resulting dendrogram will look as follows:



Now it is up to us to determine how many clusters we want to make. Remember, the higher the horizontal lines, the less similarity there is between the clusters. We know the iris dataset contains three clusters. The cutoff will be set at five to obtain three clusters:



**IMPORTANT**

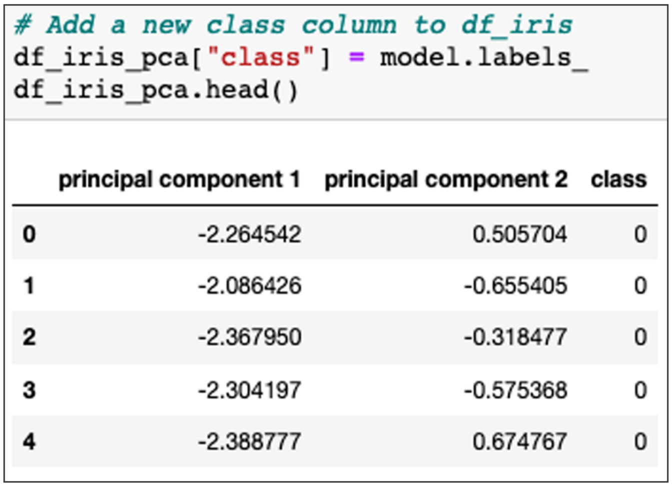
> We knew ahead of time the number of clusters to make; however, the cutoff line on the dendrogram seems high in terms of distances. This is one of the difficulties when using a dendrogram.

Now it's time to run the hierarchical algorithm. Agglomerative clustering is another name for hierarchical clustering. Enter the following code:

```
agg = AgglomerativeClustering(n_clusters=3)
model = agg.fit(df_iris_pca)
```

This will set up our model, and since you're working with a dataset that you're already familiar with, there should be three clustered groups we decided previously, so three will be passed into the `n_clusters` parameter. Then the model is fit against your `df_iris_pca`DataFrame.

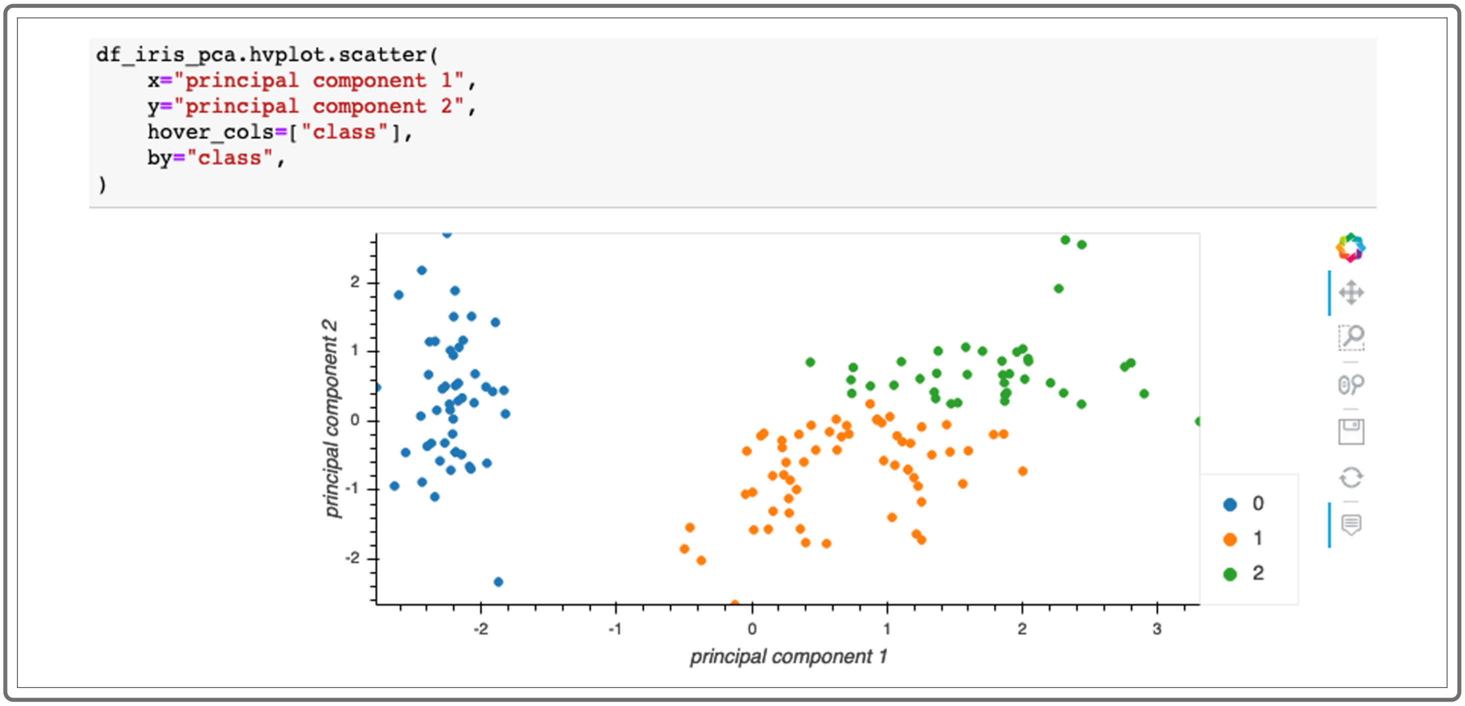
Next, add a class column that will be used to identify the clusters:



```
# Add a new class column to df_iris
df_iris_pca["class"] = model.labels_
df_iris_pca.head()
```

| | principal component 1 | principal component 2 | class |
|---|---|---|---|
| **0** | -2.264542 | 0.505704 | 0 |
| **1** | -2.086426 | -0.655405 | 0 |
| **2** | -2.367950 | -0.318477 | 0 |
| **3** | -2.304197 | -0.575368 | 0 |
| **4** | -2.388777 | 0.674767 | 0 |

Finally, create a plot to show the results of the hierarchical clustering algorithm:



```
df_iris_pca.hvplot.scatter(
    x="principal component 1",
    y="principal component 2",
    hover_cols=["class"],
    by="class",
)
```

You'll see that the process is similar for both types of clustering algorithms, and so are the results. You decide whether to apply the K-means or hierarchical clustering algorithm. In the next section, we'll review the pros and cons of each clustering algorithm.

**SKILL DRILL**

Re-run the algorithm using different cutoffs from the dendrogram.

© 2020 - 2022 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.