Lab 7: Clustering Algorithms

This lab is a continuation fo Lab 6 in which you analyzed the Philadelphia historical daily temperatures from file 'historical_temperature_philly.csv'. In this lab you will use a couple of clustering algorithms to gain more insight into the Philadelphia weather.

Hierarchical clustering

The following instructions explain how to apply hierarchical clustering on monthly temperatures. Here, every year is treated as a 12-dimensional example. The clustering algorithms groups similar examples together until all examples are grouped in one big cluster. Use the following code to perform the clustering.

```
# Create the required matrix
years = np.arange(1874,2015)
monthly_maxave_mtx = matrix_generation.monthly_ave_mtx(years, 'Max')


# Hierarchical Clustering
from scipy.spatial.distance import pdist, squareform
from scipy.cluster.hierarchy import linkage, dendrogram
data_dist_flat = pdist(monthly_maxave_mtx, 'euclidean')
data_linkage = linkage(data_dist_flat, method='average')
```

Task 1. Explain what happens after every line of the code above.

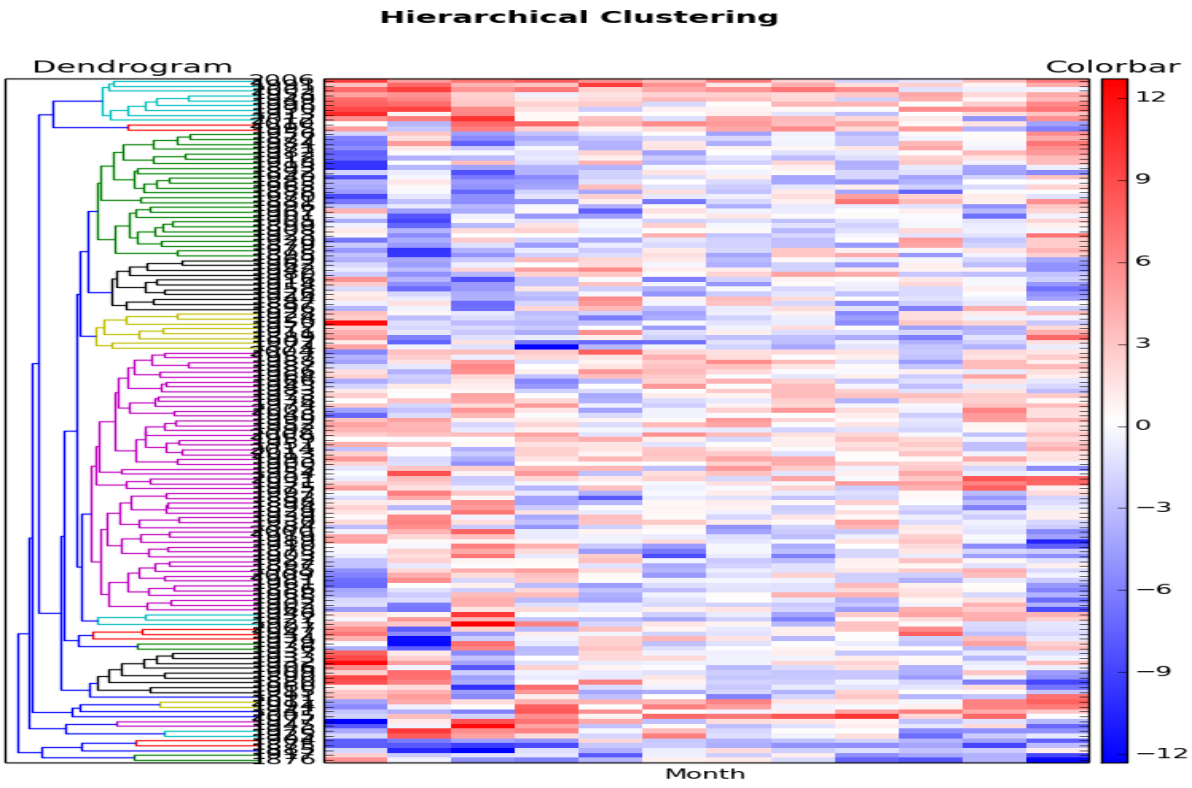
The following code helps you visualize the results. It calculates and then plots the dendrogram together with the heatmap of the examples.

```
# Plot dendrogram and the heatmap of temperatures.
# Dendrogram
%pylab qt
fig = plt.figure(figsize=(8,10))
ax1 = fig.add_axes([0.05,0.1,0.2,0.8])
Z = dendrogram(data_linkage,orientation='right')
plt.xticks([])
plt.yticks([])
plt.title('Dendrogram')
# Heatmap
ax2 = fig.add_axes([0.30, 0.1,0.6,0.8])
idx = Z['leaves']
colmean = monthly_maxave_mtx.mean(axis=0)
data_substracted = monthly_maxave_mtx - colmean[None, :]
data_ordered = data_substracted[idx, :]
im = ax2.matshow(data_ordered, aspect='auto', origin='lower',
cmap=plt.cm.bwr)
```

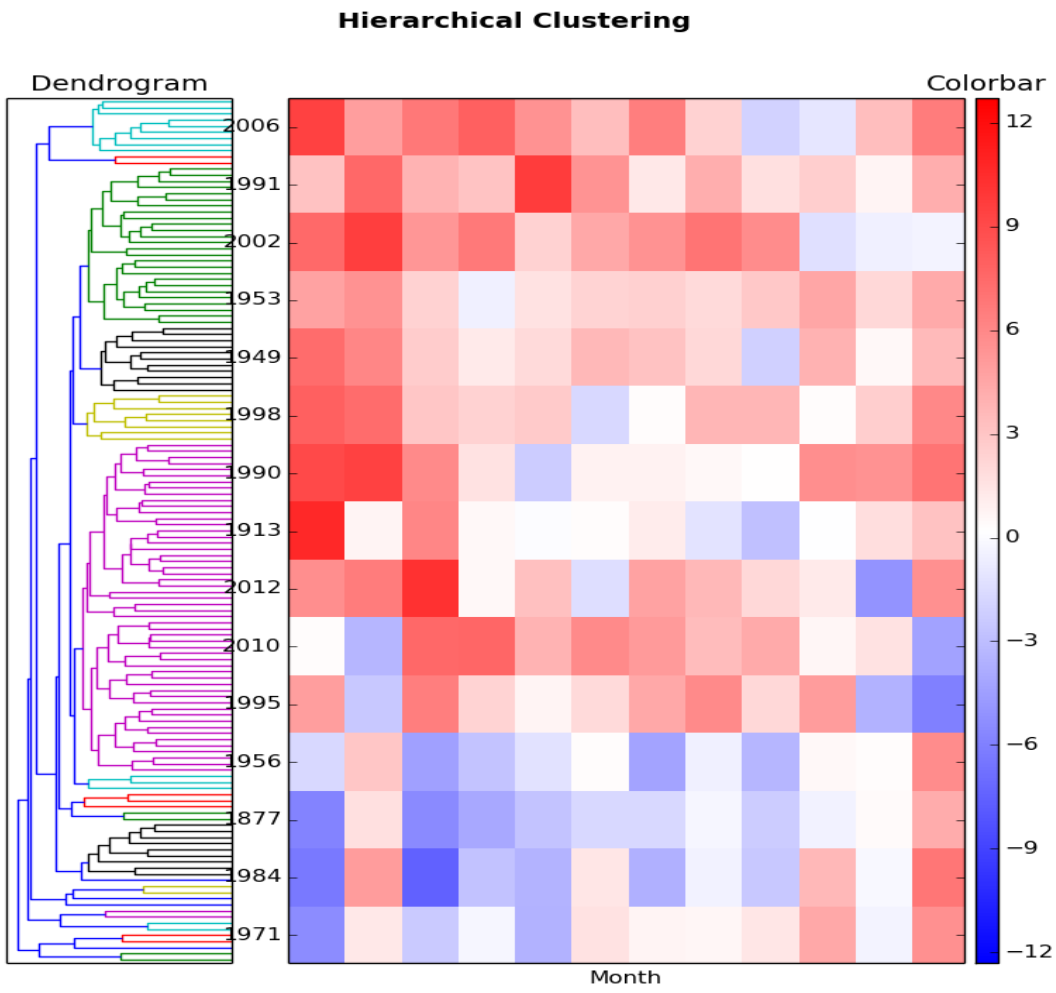
```
plt.yticks(np.arange(len(years)),years[idx])
plt.xticks([])
#plt.title('Heatmap')
plt.xlabel('Month')

# Colorbar
ax3 = fig.add_axes([0.91,0.1,0.02,0.8])
plt.colorbar(im, cax=ax3)
plt.title('Colorbar')
plt.suptitle('Hierarchical Clustering', fontweight='bold', fontsize=14)
```

This code should create a figure such as the one below:



You can zoom in a certain subset of the plot by using the zoom-in tool. An example of a zoomed-in figure is below.



Task 2. Discuss what happens after every line of code you just ran.

Task 3. Explain what is shown on the figure, both the left and the right sides.

Task 4. Find the cluster of years to which year 2014 belongs. Summarize in one-two sentences how was the last year's weather. Which other years had the similar weather. Can you identify the cluster with the hottest years and the cluster with the coldest years. What did you use to identify the hottest and the coldest years. How many years from the 21st century belong to the hottest and the coldest clusters you identified.

The following code allows you to automatically create the clusters from the hierarchical clustering you already performed:

```
cluster_assigns = fcluster(data_linkage, 10, 'maxclust')
np.bincount(cluster_assigns)
```

The code above created 10 clusters by cutting the dendrogram at an appropriate height.

Task 5. Find the size of each cluster.Is the distribution of cluster sizes uniform or skewed?

In case the cluster sizes are skewed, the Python package allows you to find more natural division of data. To accomplish this, you can use the 'inconsistency coefficients' criteria. The definition of 'inconsistency' can be found here and here. To implement it, type the following code:

```
cluster_assigns = fcluster(data_linkage, 0.8, 'inconsistent')
np.bincount(cluster_assigns)
```

Task 6. How many clusters did this create? Find the sizes of your clusters and comment on their distribution.

The line above created 53 clusters, with sizes ranging from 2 to 6.

Task 7. Repeat Task 4 by finding to which cluster the year 2014 was assigned and discuss the properties of this cluster. Also, identify the hottest and the coldest cluster and check how many 21st century years are assigned to them.

K-means clustering

K-means clustering is an alternative to hierarchical clustering. The following code allows you to create k clusters:

```
from scipy.cluster.vq import *
centers, cluster_assigns = kmeans2(monthly_maxave_mtx, 10, iter =
100, minit='random')
```

Task 8. How many clusters did you create? What are their sizes and is their distribution uniform?

Task 9. Plot a heatmap for each cluster.

Task 10. Repeat analysis from Tasks 4 and 7.