A Cluster of Flowers

In this lab you will be learning the basics of performing cluster analysis in the R programming language. Before you can begin this exercise, you will need to obtain some data. We will be using a classic machine learning data set known as the "iris data" in this lab. You can download this data set from http://archive.ics.uci.edu/ml/machine-learning-databases/iris/, by clicking on the "iris.data" link. This data set consists of measurements from 150 different irises of three different species. Each data point represents one measurement in centimeters. The final element in each line of the data file is the name of the species of iris for that instance. See the below example:

```
5.1,3.5,1.4,0.2,Iris-setosa
```

This line can be divided up into the following components:

5.1	3.5	1.4	0.2	Iris-setosa
V1	V2	V3	V4	V5

V1-V4 all represent measurements taken from this particular iris. V5 indicates that this iris is of the species Iris-setosa. In this exercise, V1-V4 is our prediction variables, and V5 is the classification variable, or the variable we are trying to predict.

Once you save the data file, move it to the directory you'll be working in for this lab.

Most machine learning tasks require you to perform some pre-processing on your data set. However, using CART in R is very user friendly and you will not be required to do any pre-processing on the data for this task. R will accept the data in the Comma Separated Value (CSV) format of the iris data.

Once you have opened the file for your R script, the first step is to read in the data from the `iris.data` file. To do so, you may use a line similar to the following:

```
inData <- read.csv("iris.data", header=FALSE)
```

This line uses the read.csv function to read in a series of comma separated values from the file "iris.data." The second parameter indicates that there is no header line present on the initial row of the file. The result of this line is that the data from the iris.data file is now stored in a data frame.

Next, we must remove the classifications of each instance. Since clustering is an unsupervised machine learning algorithm, it does not require or even know how to handle classifications of instances. To remove the last row from the data frame, use the following line:

```
indata[5] <- NULL
```

Once the extraneous column has been removed, there is only one step left before we can run the clustering algorithm on the remaining data. We must create a distance matrix for our data. A distance matrix is a matrix containing distances between each of the instances in our data set. R

will use this when it performs cluster analysis. For this exercise we will use the default Euclidean distance. It should be noted that there are other distance metrics which can be used, such as Manhattan distance. To create the distance matrix, we use a line similar to:

```
d <- dist(indata)
```

We are now ready to perform cluster analysis on our data. We can do so with the simple line:

```
cluster <- hclust(d)
```

However, on its own this line does us very little good, as R simply returns the command prompt. We can now plot our cluster.

```
plot(cluster)
```

This will plot a dendogram, a visual representation of your cluster analysis. A dendogram is a tree structure depicting which instances are closest to each other. The closer they are, the more similar. Each instance is connected with the instance closest to it. Connections are also made between groups (clusters) of instances. The height of a connection is also important to note. A shorter connection implies that the elements being connected are close together and a taller connection implies more dissimilarity.

Task:

Find another data set (preferably a nicely formatted one to save time) from http://archive.ics.uci.edu/ml/machine-learning-databases/. Repeat the steps you completed in this first task and perform cluster analysis on your new data set. Be sure to show the instructor the dendogram generated from your new data set.