**COSC 757 Data Mining Lab 1**

**Spring 2014**

The purpose of this lab is to introduce you to the R program for statistical computing and graphics. R, along with Matlab and Python is one of the most widely used programs for data analysis with extensions for various analysis and visualization techniques as well as extensions to interface directly with databases. If you would like to install R on your own machine, it can be downloaded from the following site:

<http://www.r-project.org/>

In this lab, we will be using the wine dataset taken from the UCI Machine Learning Repository <http://archive.ics.uci.edu/ml/datasets/Wine>. You can download the wine dataset (wine.csv) from the Datasets folder under Labs on Blackboard and save it to your local drive.

The wine dataset includes the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

The attributes are as follows:

1) Class

2) Alcohol

3) Malic acid

4) Ash

5) Alcalinity of ash

6) Magnesium

7) Total phenols

8) Flavanoids

9) Nonflavanoid phenols

10) Proanthocyanins

11)Color intensity

12)Hue

13)OD280/OD315 of diluted wines

14)Proline

1) Load data into R

Open R Studio by going to Start > computer science > pgms > rstudio

This should bring up the R Studio GUI where you will see the R Console.

In the R Studio GUI set the directory where you have saved the wine dataset as the working directory.

You can do this by either going to the File menu and Selecting Change dir or entering the following command for example, if you are working in D:\rlab:

> setwd(<”D:\rlab”>)

The wine dataset is in .csv format. You can load the data into R using the following command.

> data <- read.csv(“wine.csv”)

2) Examine the data

Check the dimensionality

> dim(data)

View the variable or column names

> names(data)

View the structure of the data

> str(data)

View the attributes

> attributes(data)

View the first 5 rows

> data[1:5,]

View the first 5 columns

> data[,1:5]

Get Alcohol for the first 10 rows

> data[1:10, data$Alcohol]

Convert last result into a column vector

> cbind(data[1:10, data$Alcohol])

3) Explore Individual Variables

The function summary() returns the minimum, maximum, mean, median, first quartile, and third quartile for numeric variables. For nominal variables, summary returns the frequency of every category.

> summary(data)

The mean, median, and range can be obtained with the corresponding functions mean(), median(), and range(). Quartiles and percentiles can be obtained by using the function quantile().

The default returns the 0, 25th, 50th, 75th, and 100th percentiles

> quantile(data$Total.phenols)

Return the 10th, 20th, and 65th percentiles

> quantile(data$Total.phenols, c(.1, .3, .65))

The inter-quartile range (IQR) can be obtained using the IQR() command

> IQR(data$Hue)

The table() function returns the frequency of a variable

> table(data$Class)

The result can also be viewed in a pie chart

> pie(table(data$Class))

You can observe the dispersion of the data using the standard deviation sd() and variance var().

Obtain the variance of the variable Flavanoids

> var(data$Flavanoids)

Obtain the standard deviation of the variable Flavanoids

> sd(data$Flavanoids)

Or since the standard deviation is the square root of the variance

> sqrt(var(data$Flavanoids))

4) Visualize descriptive statistics with the boxplot and histogram

The boxplot() function creates a boxplot. To create a standard boxplot where the thick line represents the median, the edges of the box represent the 1st and 3rd quartiles, and the ends of the whiskers represent the min and max enter the following command:

> boxplot(data[2:5], range=0)

The default of the boxplot() command is to create a modified boxplot of the data where the line and box represent the same values as in the standard boxplot. However, in the modified boxplot, the whiskers represent 1.5xIQR from the 1st and 3rd quartiles. Outliers are then any values that fall beyond this point and are shown as circles

> boxplot(data[2:5])

It is also easy to create a histogram in R using the hist() function

> hist(data$Ash)

You can suggest the number of breaks that you want. This gives you a higher level of detail in terms of the distribution.

> hist(data$Ash,breaks=100)

A kernel density plot can be obtained by plotting the density() function as follows

> plot(density(data$Ash))

5) Explore and compare multiple variables

The covariance of two variables can be calculated using the cov() function.

> cov(data$Alcohol, data$Ash)

The cov() function can also be used to create a covariance matrix as follows

> cov(data[,2:4])

The same can be done for correlation using the cor() function

> cor(data$Alcohol, data$Ash)

And

> cor(data[2:14])

Relationships between variables can also be visualized using scatterplots. R supports the scatter plot visualization by using the pairs() function.

> pairs(data)

6) Finding similar data objects

The individual wines can be compared using a distance matrix. This is easily computed in R using the dist() function.

> dist(data[2:14], method = ”euclidean”, diag = TRUE, upper = TRUE)