5. Multivariate Data Analysis

5.1 Introduction

Multivariate Data Analysis refers to any statistical technique used to analyze data that arises from more than one response variable. This essentially models reality where each situation, product, or decision involves more than a single variable.

When available information is stored in database tables containing rows and columns, Multivariate Analysis can be used to process the information in a meaningful fashion.

This handout tells you how to use the R statistical software to carry out some simple multivariate analyses, with a focus on cluster analysis.

5.2 Reading Multivariate Data into R

The first thing you would want to do to analyze your multivariate data is to read it into R, and plot the data. You can read data into R using the `read.csv()` function (when the data are comma separated).

Example 1:

Data set : wine.txt

The data set contains 178 data points on concentrations of 13 different chemicals in wines grown in the same region in Italy that are derived from three different cultivars.

The part of the data set is given below:

```
> wine<-read.csv("wine.txt",sep=",")
> head(wine)
```

```
  Cultivar V2   V3   V4   V5  V6   V7   V8   V9  V10  V11  V12  V13  V14
1        1 14.23 1.71 2.43 15.6 127 2.80 3.06 0.28 2.29 5.64 1.04 3.92 1065
2        1 13.20 1.78 2.14 11.2 100 2.65 2.76 0.26 1.28 4.38 1.05 3.40 1050
3        1 13.16 2.36 2.67 18.6 101 2.80 3.24 0.30 2.81 5.68 1.03 3.17 1185
4        1 14.37 1.95 2.50 16.8 113 3.85 3.49 0.24 2.18 7.80 0.86 3.45 1480
5        1 13.24 2.59 2.87 21.0 118 2.80 2.69 0.39 1.82 4.32 1.04 2.93  735
6        1 14.20 1.76 2.45 15.2 112 3.27 3.39 0.34 1.97 6.75 1.05 2.85 1450
```

There is one row per wine sample. The first column contains the cultivar of a wine sample (labeled 1, 2 or 3), and the following thirteen columns contain the concentrations of the 13 different chemicals in that sample. In this case the data on 178 samples of data has been read into the data object 'wine'.

5.3 Plotting Multivariate Data

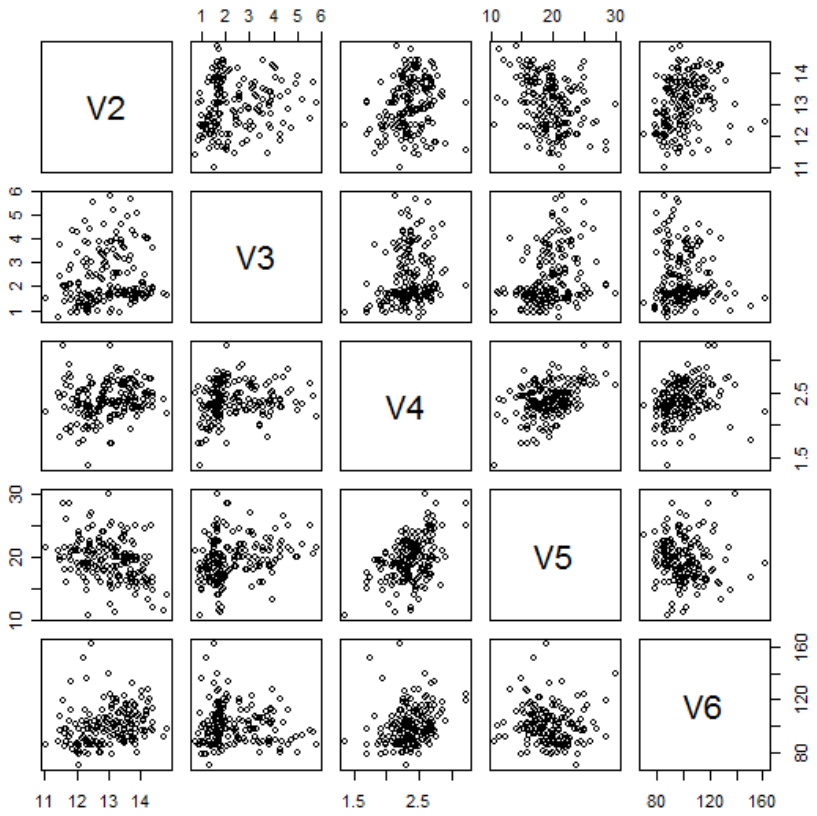
Once you have read a multivariate dataset into R, the next step is usually to plot them.

Matrix scatter plot

One common way of plotting multivariate data is to make a "matrix scatterplot", showing each pair of variables plotted against each other. We can use the `plot()` function to do this.

To use the `plot()` function, you need to specify the variables you want to plot inside the parenthesis. Say for example, that we just want to include the variables corresponding to the concentrations of the first five chemicals. These are stored in columns 2-6 of the data object "wine". We can extract just these columns from the dataset "wine" and plot them as follows:

```
> plot(wine[2:6])
```

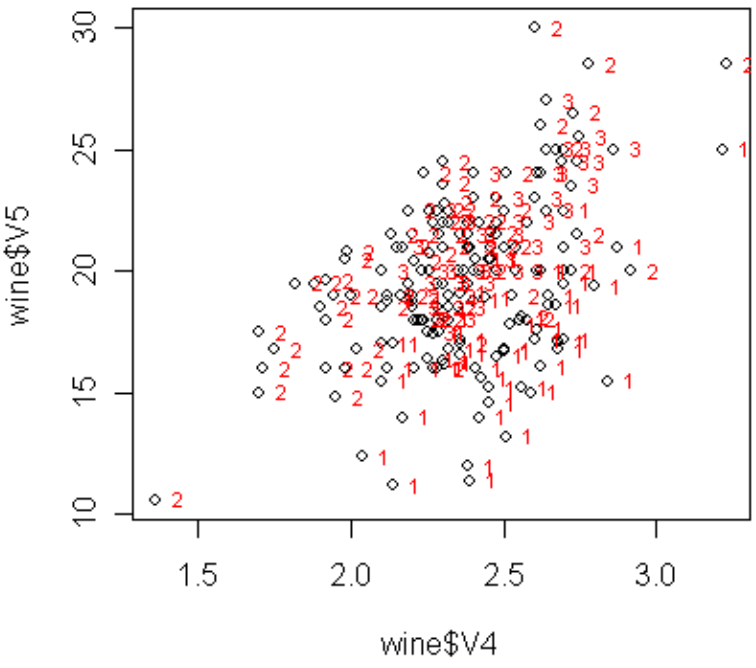


Each of the off-diagonal cell is a scatterplot of two of the five chemicals, for example, the second cell in the first row is a scatterplot of V2 (y-axis) against V3 (x-axis).

After taking an overall idea about the relationships between each pair of variables, you might want to check the relationship between particular two variables more preciously. You can use the "plot" command again for only those two variables. Moreover, if you want to label the data points by their group (the cultivar of wine here), the "text" function together with "plot" function in R would facilitate to plot some text beside every data point. In this case, the cultivar of wine is stored in the column "Cultivar" of the dataset "wine", so we type:

```
> plot(wine$V4, wine$V5)
> text(wine$V4, wine$V5, wine$Cultivar, cex=0.5, pos=4,
col="red")
```

If you look at the help page for the "text" function, you will see that "pos=4" will plot the text just to the right of the symbol for a data point. The "cex=0.5" option will plot the text at half the default size, and the "col=red" option will plot the text in red. This gives us the following plot:



5.4 Calculating Summary Statistics for Multivariate Data

To calculate summary statistics such as the mean and standard deviation for each of the variables in your multivariate data, `sapply()` function can be used.

```
> sapply(wine[2:14],mean)
         V2          V3          V4          V5          V6          V7          V8
 13.0006180   2.3363483   2.3665169  19.4949438  99.7415730   2.2951124   2.0292697

    V9         V10         V11         V12         V13         V14
 0.3618539   1.5908989   5.0580899   0.9574494   2.6116854 746.8932584
```

This tells us that the mean of variable V2 is 13.0006180, the mean of V3 is 2.3363483, and so on.

Similarly, to get the standard deviations of the 13 chemical concentrations, we type:

```
>  sapply(wine[2:14],sd)
         V2          V3          V4          V5          V6          V7          V8
 0.8118265   1.1171461   0.2743440   3.3395638  14.2824835   0.6258510   0.9988587

         V9         V10         V11         V12         V13         V14
 0.1244533   0.5723589   2.3182859   0.2285716   0.7099904 314.9074743
```

5.5 Calculating Correlations for Multivariate Data

It is often of interest to investigate whether any of the variables in a multivariate data set are significantly correlated.

To calculate the linear (Pearson) correlation coefficient for a pair of variables, you can use the `cor.test()` function in R. For example, to calculate the correlation coefficient between concentrations, V7 and V8, we type:

```
> cor.test(wine$V7, wine$V8)

        Pearson's product-moment correlation
data:  wine$V7 and wine$V8
t = 22.8243, df = 176, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.8220088 0.8975164
sample estimates:
cor
0.8645635
```

This tells us that the correlation coefficient is about 0. 864, which is a strong positive correlation. Furthermore, the P-value for the statistical test of whether the correlation coefficient is significantly different from zero is almost 0 (2.2e-16). So, there is enough evidence to suggest that the correlation is significant.

5.6 Cluster Analysis

Cluster analysis comprises a series of methods for determining natural groupings in multivariate data. When a dataset is given with one or more characteristics (i.e. variables), one may need to classify the data into homogeneous groups of individuals or objects. These means within the homogeneous groups' objects are similar to one another, and dissimilar to the objects in other groups or clusters. Therefore in cluster analysis, similarities between data are found according to the characteristics (variables) given in the data and similar data objects are grouped into clusters.

Clustering techniques are based on distance measures, and there are a wide range of clustering approaches. Some of the main types, hierarchical and non-hierarchical are discussed here.

Hierarchical Clustering Methods

Two different methods are considered in hierarchical cluster analysis, which are agglomerative and divisive methods. The agglomerative methods start with the individual objects. Most similar objects are first grouped, and these initial groups are merged according to their similarities. Eventually, as the similarity decreases, all subgroups are merged into a single cluster. The divisive methods start with all of the observations in one cluster and then proceeds to split (partition) them into smaller clusters.

According to the "wine" example explained in previous section, the wine samples are from 3 different cultivators. We can use cluster analysis to see how the 178 samples would be grouped, if we group the data considering the given 13 chemical concentrations.

Several R functions are available to perform both methods (i.e. agglomerative & divisive) of hierarchical clustering including;

- *hclust* ("stats" package)
- *agnes* ("cluster" package) — Agglomerative HC
- *diana* ("cluster" package) - Divisive HC

As an example, let us use "hclust" function in R to cluster the wine samples through agglomerative hierarchical clustering. If you refer the help for "hclust" function in R, the first argument in that is a dissimilarity matrix which can be

obtained using the “dist” function. However, since the values of the numerical variables of the wine data set are in different scales, first we have to standardize them (i.e. bring all the numerical variables into the same scale).

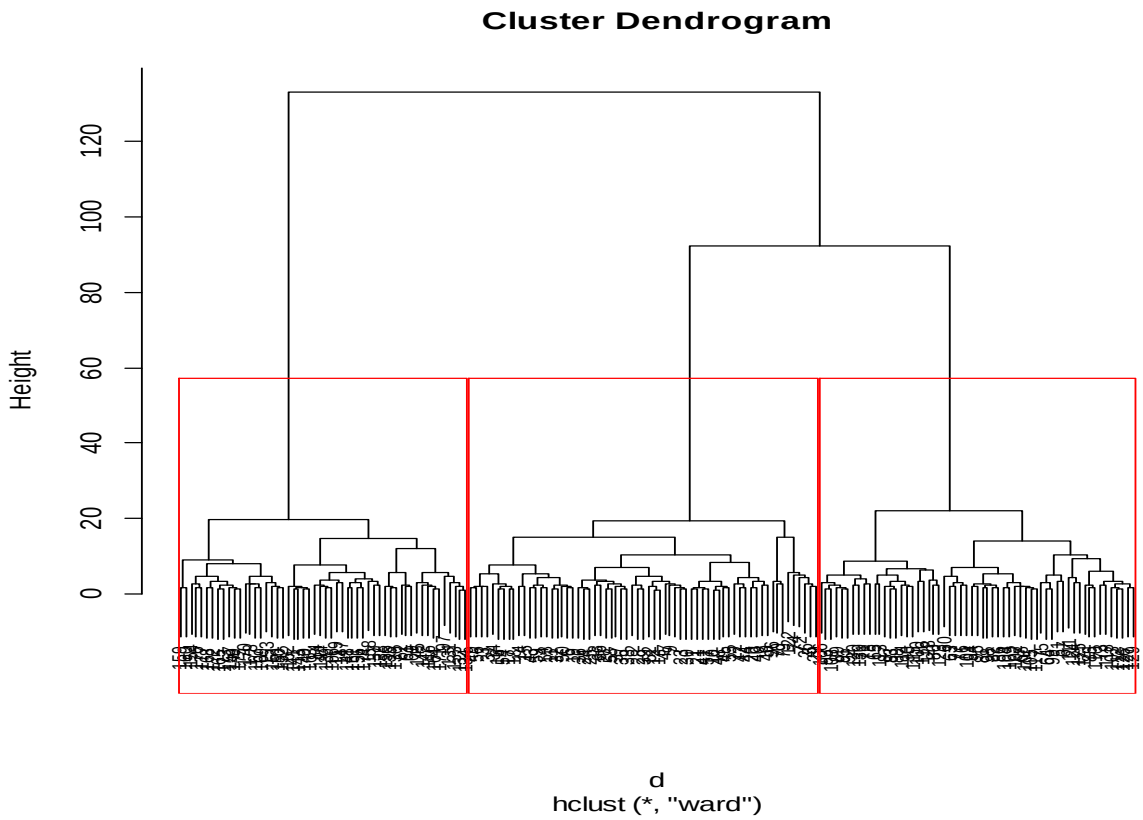
```
wine.std=scale(wine[2:14])#standardizing variables
d<-dist(wine.std, method = "euclidean")#obtaining dissimilarity
matrix
clust<-hclust(d, method="ward.D")#agglomerative clustering
plot(clust,cex=0.7)
rect.hclust(clust, k=3, border="red")
```

Note that in the “dist” function above, the method is given as “euclidean”, but you can choose any of the distances from the list "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski".

Same way for “hclust” function, the method “ward.D” is used here, but it can be one of "ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median" or "centroid". These are different methods of clustering and different algorithms are used in each one.

To divide the resulting dendrogram in to a desired number of clusters, “rect.hclust” function can be used. In this example the number of clusters needed is mentioned as 3, in the color red.

Using the above commands, the following dendrogram can be drawn and it is clearly visible that the data set cluster to 3 clear groupings.



To cut the dendrogram at a desired height (i.e. to obtain a desired number of clusters) and obtain cluster allocations, the "cutree" function can also be used:

```
clust<-hclust(d, method="ward.D")
groups <- cutree(clust, k = 3)
```

The outputs obtained through the "cutree' function can be added as a new variable to the original data to denote the predicted cluster for each observation:

```
wine$clusters <- groups
```

Note: If the clusters to be identified are small, agglomerative hierarchical clustering would perform better whereas, if the clusters to be identified are large, divisive hierarchical clustering would perform better.

Non- Hierarchical Clustering Methods

Non- Hierarchical or partition clustering methods are described here. These methods typically start by dividing all the items into a pre-specified number of clusters. Although there are many different partition methods such as k-means and k-medians methods, here we consider only the k-means method.

We can use the same data set and cluster it using k-means method with the following command. The number of clusters to start with is a subjective judgment and here we are going to use 3 clusters.

```
> kmeans(wine.std,3)
K-means clustering with 3 clusters of sizes 62, 51, 65

Cluster means:
          V2         V3         V4         V5          V6          V7          V8
1  0.8328826 -0.3029551  0.3636801 -0.6084749  0.57596208  0.88274724  0.97506900
2  0.1644436  0.8690954  0.1863726  0.5228924 -0.07526047 -0.97657548 -1.21182921
3 -0.9234669 -0.3929331 -0.4931257  0.1701220 -0.49032869 -0.07576891  0.02075402
           V9         V10        V11        V12        V13        V14
1 -0.56050853  0.57865427  0.1705823  0.4726504  0.7770551  1.1220202
2  0.72402116 -0.77751312  0.9388902 -1.1615122 -1.2887761 -0.4059428
3 -0.03343924  0.05810161 -0.8993770  0.4605046  0.2700025 -0.7517257

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [41] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 2 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3
 [81] 3 3 3 2 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3
[121] 3 1 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[161] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

In the given output, the cluster means for each of the variable is given and the cluster numbers that each of the samples are belonged are also mentioned. So we can see that the dataset clusters to 3 clear groups according to the cultivators, even though there are few points that are clustered to the incorrect groups.

Note: Similar to the techniques explained in this module, there are other multivariate methods such as Multivariate Analysis of Variance (MANOVA), Principal Component Analysis (PCA), Factor Analysis, Discriminant Analysis etc. and there are R functions to do those too. So you can refer R help for further analysis in multivariate methods.