Chapter 1 Introduction to Multivariate Statistical Analyses

Math 3210

Dr. Zeng

Outline

- What is multivariate statistical analyses
- Selecting appropriate analyses
- Supervised and Unsupervised learning
- Characterizing data for analyses
- Preparing for data analysis

What is multivariate statistical analyses

The expression multivariate analysis is used to describe analyses of data that are multivariate in the sense that numerous observations or variables are obtained for each individual or unit studied. The multivariate statistical analyses is commonly a rewarding activity but it is also a challenge sometimes due to the complicated interrelationships among a large number of variables.

Examples of multivariate analyses

```
Diamonds<-read.table("Diamonds.txt", header=TRUE)
head(Diamonds, 10)
```

```
   IDNO WEIGHT COLOR CLARITY RATER PRICE
1     1   0.30     D     VS2   GIA  1302
2     2   0.30     E     VS1   GIA  1510
3     3   0.30     G    VVS1   GIA  1510
4     4   0.30     G     VS1   GIA  1260
5     5   0.31     D     VS1   GIA  1641
6     6   0.31     E     VS1   GIA  1555
7     7   0.31     F     VS1   GIA  1427
8     8   0.31     G    VVS2   GIA  1427
9     9   0.31     H     VS2   GIA  1126
10   10   0.31     I     VS1   GIA  1126
```

Selecting appropriate analyses

Regression Analysis

In statistical modeling, regression analysis is a set of statistical processes for estimating the relationships among variables.

- Simple linear regression
- Multiple linear regression
- Logistic regression
- Poisson regression

Principal Component Analysis

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The principal components are linear functions of the original variables and will be uncorrelated to each other. It is hoped that the first two or three principal components will explain most of the variation in the response variables.

Factor Analysis

Factor analysis is a useful tool for investigating variable relationships for complex concepts such as socioeconomic status, dietary patterns, or psychological scales. It allows researchers to investigate concepts that are not easily measured directly by collapsing a large number of variables into a few interpretable underlying factors.

Cluster Analysis

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields.

Discriminant analysis

Discriminant analysis techniques are used to classify individuals into one of two or more alternative groups (or populations) on the basis of a set of measurements.

Supervised Learning

- In supervised learning, outcome measurement typically denoted by y and it is called as response variable or dependent variable.
- Typically, there is a set of p predictor variables denoted by $x_1, x_2, ..., x_n$. They are also called as independent variables, regressors, or covariates.
- For each observation of the predictor measurements, there is an associated response measurement y.
- In regression, y is quantitative (e.g. distance, lifetime)
- In classification, y takes values in a finite, unordered set (e.g. model type A/B, survived/died, digit 0-10)
- Normally, the goal of the supervised learning is to predicting the response for future observations or better understanding the relationship between the response and the predictors.

Examples of Supervised learning:

ChickWeight:

```
names(ChickWeight)
```

```
[1] "weight" "Time"   "Chick"  "Diet"
```

```
head(ChickWeight)
```

```
  weight Time Chick Diet
1     42    0     1    1
2     51    2     1    1
3     59    4     1    1
4     64    6     1    1
5     76    8     1    1
6     93   10     1    1
```

- Goal 1: Understand which variable statistically affect the chicken weight.
- Goal 2: Predict the Weight based on the chicken types, diet and time.
- Goal 3: Assess the quality of our predictions and inferences.

Unsupervised Learning

- Unsupervied leanring has no outcome variable y. There is only a set of predictors $x_1, x_2, ...x_n$ or features measured on a set of samples.
- No modeling for the data as there is no response.
- Goals: find patterns, features that have similar characteristics, groups, associations in data. No prediction or inference.

Characterizing data for analyses

This section shows you how to analysis multivariate data numerically and graphically.

Example: The "wine.txt" contains data on concentrations of 13 different chemicals in wines grown in the same region in Italy that are derived from three different cultivars. 173 bottles of wines are sampled. The first column contains the cultivar of a wine sample (labelled 1, 2 or 3), and the following thirteen columns contain the concentrations of the 13 different chemicals in that sample. The columns are separated by commas.

```
wine<-read.table("wine.txt",header=F,sep=",")
```

Plotting Multivariate Data

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 **scatterplotMatrix()** function from the "car" R package to do this. To use this function, we first need to install the "car" R package.

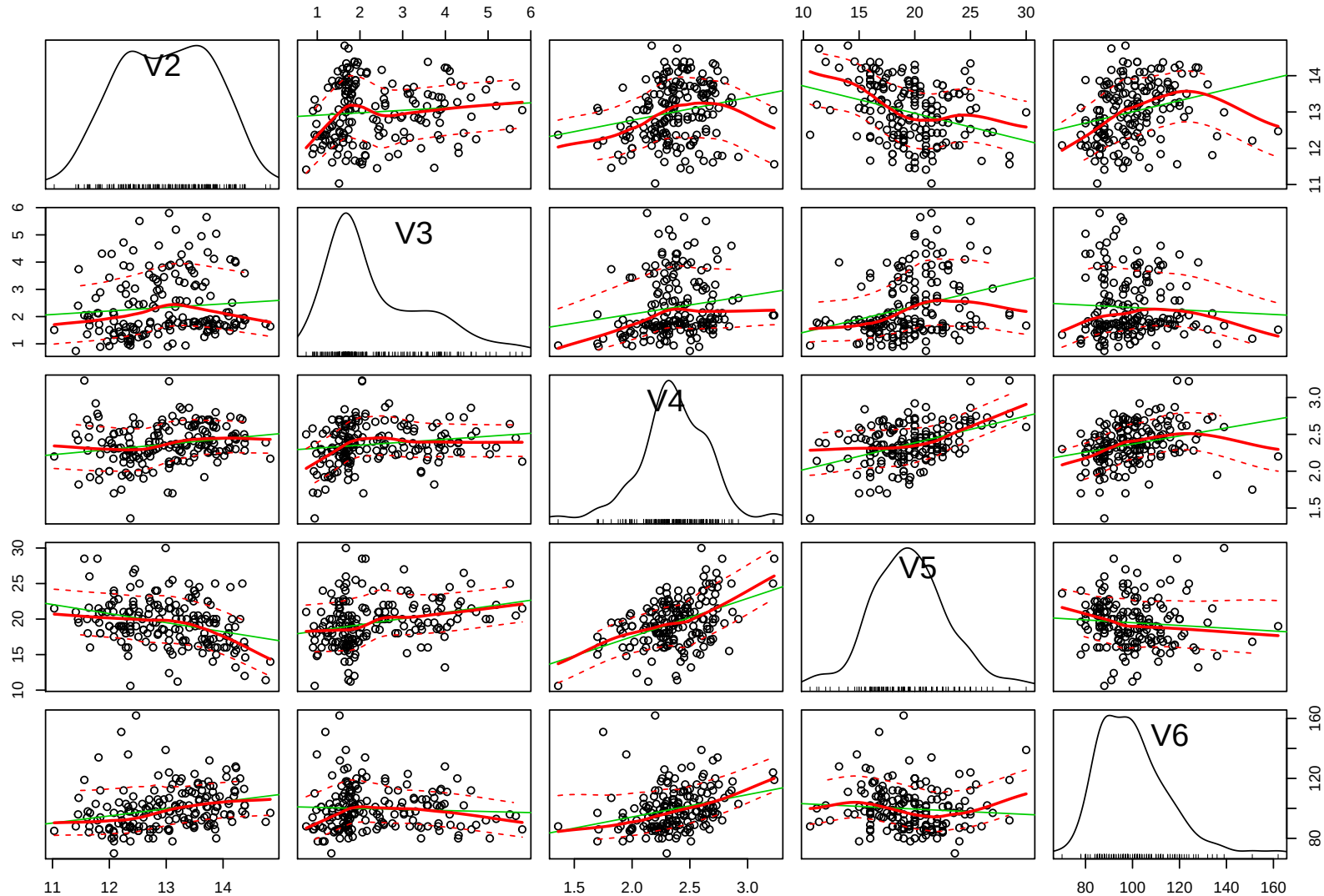
```
if (!require('car'))
{
  install.packages("car",repos = "http://cran.us.r-project.org");
  library(car);
}
```

```
Loading required package: car
```

The scatterplotMatrix() function:

To use the scatterplotMatrix() function, we need to give it as its input the variables that you want included in the plot. For example, we just want to include the variables corresponding to the concentrations of the first five chemicals (V2-V6). To make a matrix scatterplot of just these selected variables using the scatterplotMatrix() function we type:

```
scatterplotMatrix(wine[2:6])
```



Interpretation:

In this matrix scatterplot, the diagonal cells show histograms of each of the variables, in this case the concentrations of the first five chemicals (variables V2, V3, V4, V5, V6).

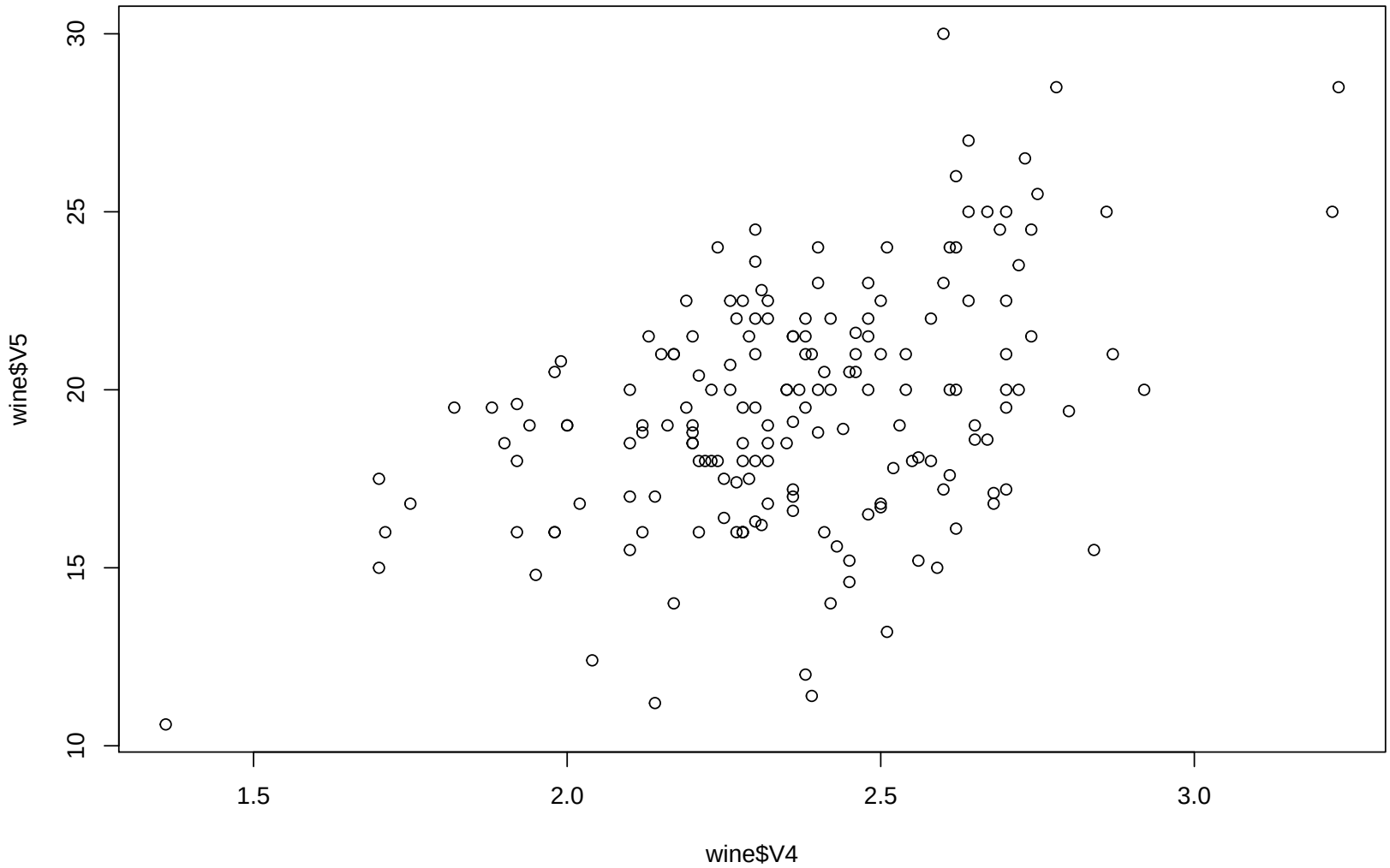
Each of the off-diagonal cells is a scatterplot of two of the five chemicals, for example, the second cell in the first row is a scatterplot of V2 (y-axis) against V3 (x-axis).

The plot() function

If you see an interesting scatterplot for two variables in the matrix scatterplot, you may want to plot that scatterplot in more detail.

For example, you might have observed a positive relationship between V5 and V4 in the matrix scatterplot above. We may therefore decide to examine the relationship between V5 and V4 more closely, by plotting a scatterplot of these two variable.

```r
plot(wine$V4, wine$V5)
```



The text() function

If we want to label the data points by their group (the cultivar of wine here), we can use the "text" function in R to plot some text beside every data point. In this case, the cultivar of wine is stored in the column V1 of the variable "wine".

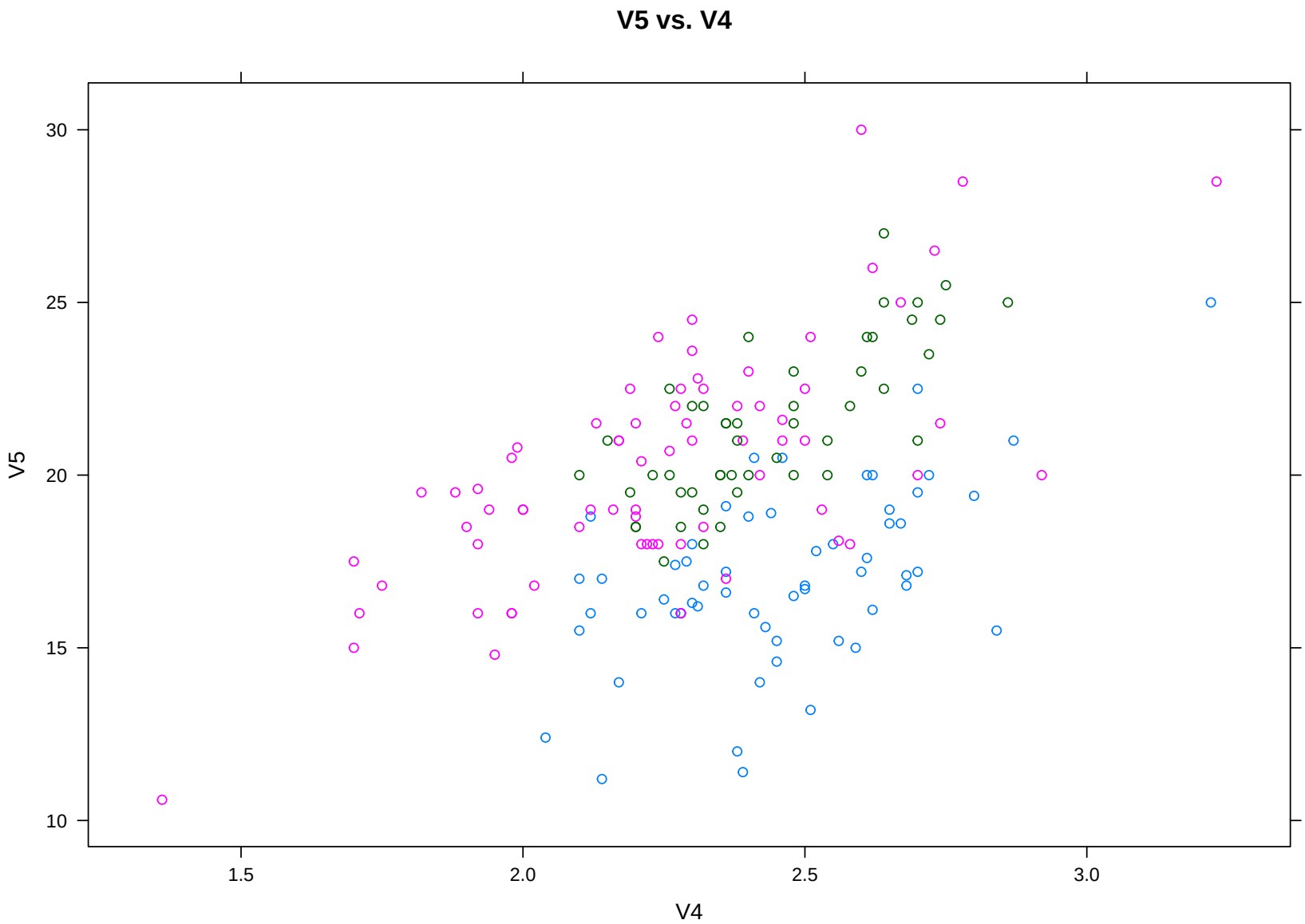
In the following R code, the "pos=4" option will plot the text just to the right of the symbol for a data point. The "cex=0.5" option will plot the text at half the default size, and the "col=red" option will plot the text in red.

```r
plot(wine$V4, wine$V5) ##must run plot() first before text()
text(wine$V4, wine$V5, wine$V1, cex=0.7, pos=4, col="red")
```

The xyplot() function

You can also use the **xyplot()** function, the syntax is:

```
library(lattice)
xyplot( V5 ~ V4, group=V1 , data= wine, type=c("p"), main=" V5 vs. V4")
```



A Profile Plot

Another type of plot that is useful is a "profile plot", which shows the variation in each of the variables, by plotting the value of each of the variables for each of the samples.

The **makeProfilePlot()** below can be used to make a profile plot. To use this function, we need to copy and paste the R codes in the next slide into R. Then we will install the package by using **install.packages("RColorBrewer")** and then load the package using **library(RColorBrewer)**.

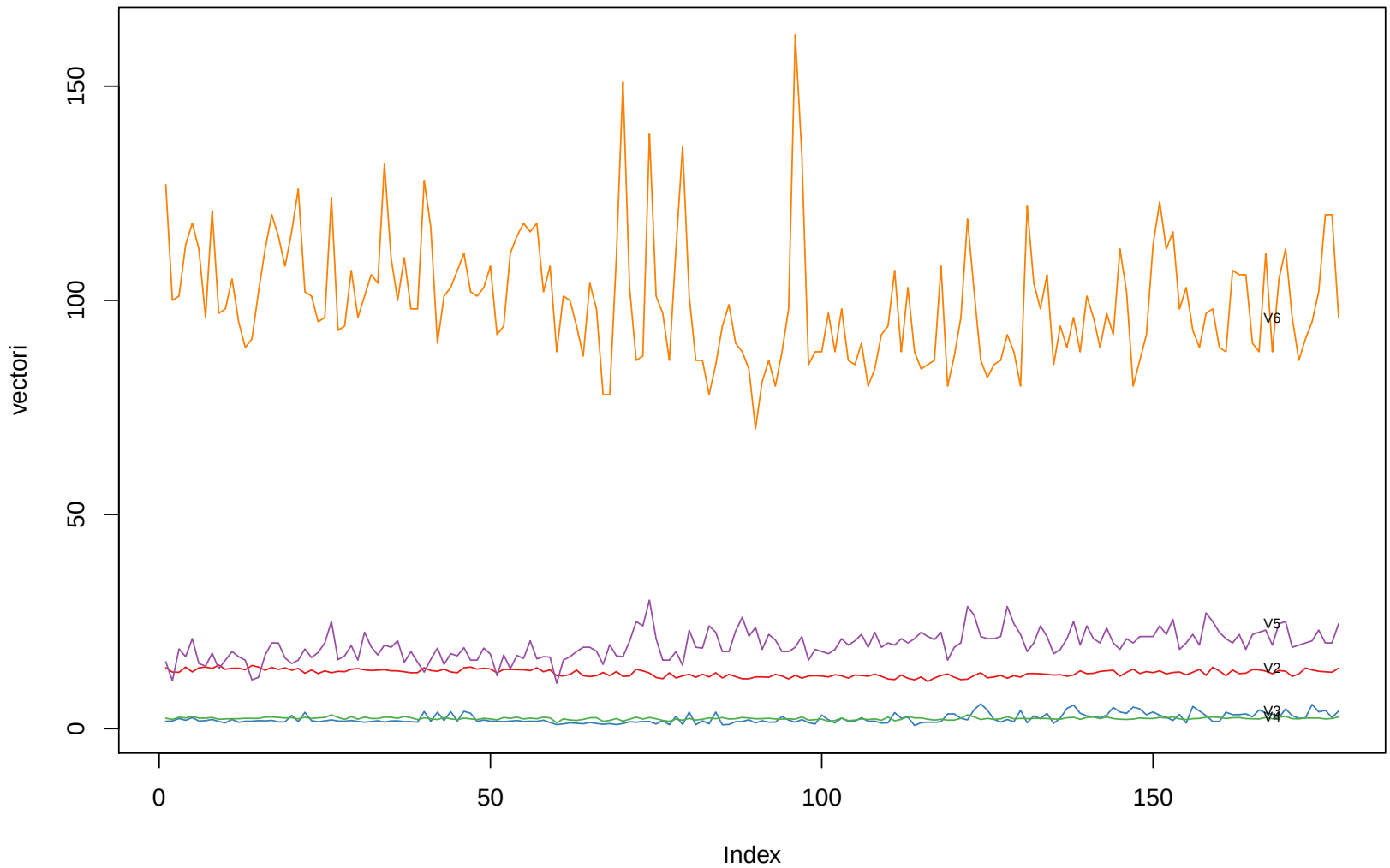
```
makeProfilePlot <- function(mylist,names){
     require(RColorBrewer)
     numvariables <- length(mylist)
     colours <- brewer.pal(numvariables,"Set1")
     mymin <- 1e+20
     mymax <- 1e-20
     for (i in 1:numvariables) {
        vectori <- mylist[[i]]
        mini <- min(vectori)
        maxi <- max(vectori)
        if (mini < mymin) { mymin <- mini }
        if (maxi > mymax) { mymax <- maxi }
     }
     for (i in 1:numvariables) {
        vectori <- mylist[[i]]
        namei <- names[i]
        colouri <- colours[i]
        if (i == 1) { plot(vectori,col=colouri,type="l",
                           ylim=c(mymin,mymax)) }
        else         { points(vectori, col=colouri,type="l") }
        lastxval <- length(vectori)
        lastyval <- vectori[length(vectori)]
        text((lastxval-10),(lastyval),namei,col="black",cex=0.6)
     }}
```

```r
if (!require('RColorBrewer'))
{
  install.packages("RColorBrewer",
                   repos = "http://cran.us.r-project.org");
  library(RColorBrewer);
}
```

```
Loading required package: RColorBrewer
```

For example, to make a profile plot of the concentrations of the first five chemicals in the wine samples (stored in columns V2, V3, V4, V5, V6 of variable "wine"), we type:

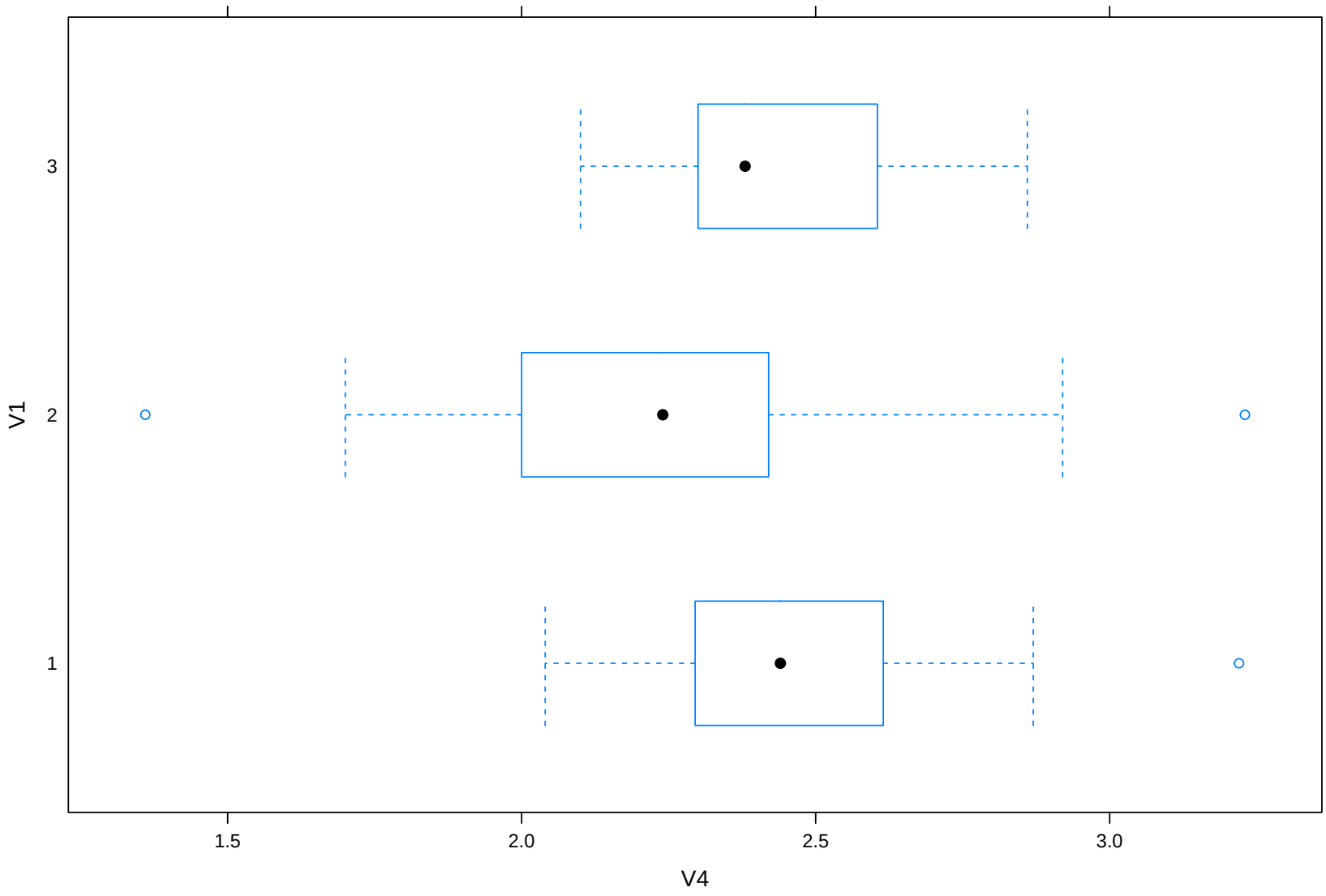
```
names <- c("V2","V3","V4","V5","V6")
mylist <- list(wine$V2,wine$V3,wine$V4,wine$V5,wine$V6)
makeProfilePlot(mylist,names)
```



Boxplot

If you want to compare the performance of a particular chemical (say, V4) among different wine samples, a side-by-side boxplot or histograms

```r
library(lattice)
bwplot(V1 ~ V4 , data= wine)
```



Summary Statistics for Multivariate Data

Next, let's see how to calculate summary statistics for multivariate data. Previously in Chapter 4, We have learned about the **vapply()** function which returns a pre-specified type of outcomes. The **sapply()** function is similar but by default returning a vector.

For example, we want to calculate the mean and standard deviations of each of the 13 chemical concentrations in the wine samples.

```r
sapply(wine[2:14],mean)  ## method 1
```

```
         V2          V3          V4          V5          V6          V7
 13.0006180   2.3363483   2.3665169  19.4949438  99.7415730   2.2951124
         V8          V9         V10         V11         V12         V13
  2.0292697   0.3618539   1.5908989   5.0580899   0.9574494   2.6116854
        V14
746.8932584
```

```r
sapply(wine[2:14],sd)  ## same result
```

```
         V2          V3          V4          V5          V6          V7
  0.8118265   1.1171461   0.2743440   3.3395638  14.2824835   0.6258510
         V8          V9         V10         V11         V12         V13
  0.9988587   0.1244533   0.5723589   2.3182859   0.2285716   0.7099904
        V14
314.9074743
```

```r
## vapply(wine[2:14],mean,FUN.VALUE= 1)  ## method 2
## vapply(wine[2:14],sd,FUN.VALUE= 1)  ## method 2
## summary(wine) ## method3
```

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 for the first two chemicals' concentrations, V2 and V3, we type:

```
cor.test(wine$V2, wine$V3)
```

```
    Pearson's product-moment correlation

data:  wine$V2 and wine$V3
t = 1.2579, df = 176, p-value = 0.2101
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.05342959  0.23817474
sample estimates:
       cor
0.09439694
```

Calculating the variance-covariance matrix

Sometimes, we are interestesd in the correlations of multiple variables. To obtain the variance-covariance matrix, you can use the **cov()** function.

```
cov(wine[2:5])
```

```
            V2         V3         V4          V5
V2  0.65906233 0.08561131 0.04711516 -0.8410929
V3  0.08561131 1.24801540 0.05027704  1.0763317
V4  0.04711516 0.05027704 0.07526464  0.4062083
V5 -0.84109290 1.07633171 0.40620828 11.1526862
```

```
## cov(wine)
```