Multivariate Data Analysis

Multivariate data sets

This is a convenient point to look at some multivariate data sets and briefly ponder the type of question that might be of interest in each case. The first data set consists of chest, waist, and hip measurements on a sample of men and women and the measurements for 20 individuals are shown in Table 1.2. Two questions might be addressed by such data;

. Could body size and body shape be summarised in some way by combining the three measurements into a single number?

. Are there subtypes of body shapes amongst the men and amongst the women within which individuals are of similar shapes and between which body shapes differ?

Table 1.2: `measure` data. Chest, waist, and hip measurements on 20 individuals (in inches).

chest	waist	hips	gender	chest	waist	hips	gender
34	30	32	male	36	24	35	female
37	32	37	male	36	25	37	female
38	30	36	male	34	24	37	female
36	33	39	male	33	22	34	female
38	29	33	male	36	26	38	female
43	32	38	male	37	26	37	female
40	33	42	male	34	25	38	female
38	30	40	male	36	26	37	female
40	30	37	male	38	28	40	female
41	32	39	male	35	23	35	female

The first question might be answered by **principal components analysis**
The second question could be investigated using **cluster analysis.**

Our second set of multivariate data consists of the results of chemical analysis on Romano-British pottery made in three different regions (region 1 contains kiln 1, region 2 contains kilns 2 and 3, and region 3 contains kilns 4 and 5). The complete data set, which we shall meet consists of the chemical analysis results on 45 pots, shown in Table 1.3.

One question that might be posed about these data is whether the chemical profiles of each pot suggest different types of pots and if any such types are related to kiln or region

Table 1.3: pottery data (continued).

Al2O3	Fe2O3	MgO	CaO	Na2O	K2O	TiO2	MnO	BaO	kiln
18.6	7.85	2.33	0.87	0.38	3.17	0.98	0.081	0.018	1
16.9	7.87	1.83	1.31	0.53	3.09	0.95	0.092	0.023	1
18.9	7.58	2.05	0.83	0.13	3.29	0.98	0.072	0.015	1
18.0	7.50	1.94	0.69	0.12	3.14	0.93	0.035	0.017	1
17.8	7.28	1.92	0.81	0.18	3.15	0.90	0.067	0.017	1
14.4	7.00	4.30	0.15	0.51	4.25	0.79	0.160	0.019	2
13.8	7.08	3.43	0.12	0.17	4.14	0.77	0.144	0.020	2
14.6	7.09	3.88	0.13	0.20	4.36	0.81	0.124	0.019	2
11.5	6.37	5.64	0.16	0.14	3.89	0.69	0.087	0.009	2
13.8	7.06	5.34	0.20	0.20	4.31	0.71	0.101	0.021	2
10.9	6.26	3.47	0.17	0.22	3.40	0.66	0.109	0.010	2
10.1	4.26	4.26	0.20	0.18	3.32	0.59	0.149	0.017	2
11.6	5.78	5.91	0.18	0.16	3.70	0.65	0.082	0.015	2
11.1	5.49	4.52	0.29	0.30	4.03	0.63	0.080	0.016	2
13.4	6.92	7.23	0.28	0.20	4.54	0.69	0.163	0.017	2
12.4	6.13	5.69	0.22	0.54	4.65	0.70	0.159	0.015	2
13.1	6.64	5.51	0.31	0.24	4.89	0.72	0.094	0.017	2
11.6	5.39	3.77	0.29	0.06	4.51	0.56	0.110	0.015	3
11.8	5.44	3.94	0.30	0.04	4.64	0.59	0.085	0.013	3
18.3	1.28	0.67	0.03	0.03	1.96	0.65	0.001	0.014	4
15.8	2.39	0.63	0.01	0.04	1.94	1.29	0.001	0.014	4
18.0	1.50	0.67	0.01	0.06	2.11	0.92	0.001	0.016	4
18.0	1.88	0.68	0.01	0.04	2.00	1.11	0.006	0.022	4
20.8	1.51	0.72	0.07	0.10	2.37	1.26	0.002	0.016	4
17.7	1.12	0.56	0.06	0.06	2.06	0.79	0.001	0.013	5
18.3	1.14	0.67	0.06	0.05	2.11	0.89	0.006	0.019	5
16.7	0.92	0.53	0.01	0.05	1.76	0.91	0.004	0.013	5
14.8	2.74	0.67	0.03	0.05	2.15	1.34	0.003	0.015	5
19.1	1.64	0.60	0.10	0.03	1.75	1.04	0.007	0.018	5

: Tubb, A., et al., *Archaeometry*, 22, 153–171, 1980. With permission.

Our third set of multivariate data involves the examination scores of a large number of college students in six subjects; the scores for five subjects are shown in Table 1.4. Here the main question of interest might be whether the exam scores reect some underlying trait in a student that cannot be measured directly, perhaps \general intelligence"?

The question could be investigated by using exploratory factor analysis

Table 1.4: exam data. Exam scores for five psychology students.

subject	maths	english	history	geography	chemistry	physics
1	60	70	75	58	53	42
2	80	65	66	75	70	76
3	53	60	50	48	45	43
4	85	79	71	77	68	79
5	45	80	80	84	44	46

The main reason why we should analyse a multivariate data set using multivariate methods rather than looking at each variable separately using one or another familiar univariate method is that any structure or pattern in the data is as likely to be implied either by “relationships" between the variables or by the relative “closeness" of different units as by their different variable values; in some cases perhaps by both.

In the first case, any structure or pattern uncovered will be such that it “links" together the columns of the data matrix, X, in some way, and in the second case a possible structure that might be discovered is that involving interesting subsets of the units.

The question now arises as to how we quantify the relationships between the variables and how we measure the distances between different units.

In a multivariate data set with q observed variables, there are q variances and $q(q-1)/2$ covariances. These quantities can be conveniently arranged in a $q \times q$ symmetric matrix, $\mathbf{\Sigma}$, where

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1q} \\ \sigma_{21} & \sigma_2^2 & \dots & \sigma_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{q1} & \sigma_{q2} & \dots & \sigma_q^2 \end{pmatrix}.$$

Note that $\sigma_{ij} = \sigma_{ji}$. This matrix is generally known as the *variance-covariance matrix* or simply the *covariance matrix* of the data.

For a set of multivariate observations, perhaps sampled from some population, the matrix $\mathbf{\Sigma}$ is estimated by

$$\mathbf{S} = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top,$$

where $\mathbf{x}_i^\top = (x_{i1}, x_{i2}, \dots, x_{iq})$ is the vector of (numeric) observations for the ith individual and $\bar{\mathbf{x}} = n^{-1} \sum_{i=1}^{n} \mathbf{x}_i$ is the mean vector of the observations. The diagonal of $\mathbf{S}$ contains the sample variances of each variable, which we shall denote as s_i^2.

shall denote as S_i.

The covariance matrix for the data in Table 1.2 can be obtained using the `var()` function in R; however, we have to "remove" the categorical variable `gender` from the `measure` data frame by subsetting on the numerical variables first:

```
R> cov(measure[, c("chest", "waist", "hips")])

      chest  waist  hips
chest 6.632  6.368 3.000
waist 6.368 12.526 3.579
hips  3.000  3.579 5.945
```

If we require the separate covariance matrices of men and women, we can use

```
R> cov(subset(measure, gender == "female")[,
+              c("chest", "waist", "hips")])

      chest waist  hips
chest 2.278 2.167 1.556
waist 2.167 2.989 2.756
hips  1.556 2.756 3.067

R> cov(subset(measure, gender == "male")[,
+              c("chest", "waist", "hips")])

       chest  waist  hips
chest 6.7222 0.9444 3.944
waist 0.9444 2.1000 3.078
hips  3.9444 3.0778 9.344
```

where the `subset()` returns all observations corresponding to females (first statement) or males (second statement).

The covariance is often difficult to interpret because it depends on the scales on which the two variables are measured; consequently, it is often standardised by dividing by the product of the standard deviations of the two variables to give a quantity called the *correlation coefficient*, ρ_{ij}, where

$$\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j},$$

where $\sigma_i = \sqrt{\sigma_i^2}$.

The advantage of the correlation is that it is independent of the scales of the two variables. The correlation coefficient lies between -1 and $+1$ and gives a measure of the *linear* relationship of the variables X_i and X_j. It is positive if high values of X_i are associated with high values of X_j and negative if high values of X_i are associated with low values of X_j. If the relationship between two variables is non-linear, their correlation coefficient can be misleading.

With q variables there are $q(q-1)/2$ distinct correlations, which may be arranged in a $q \times q$ correlation matrix the diagonal elements of which are unity. For observed data, the correlation matrix contains the usual estimates of the ρs, namely Pearson's correlation coefficient, and is generally denoted by $\mathbf{R}$. The matrix may be written in terms of the sample covariance matrix $\mathbf{S}$

$$\mathbf{R} = \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{-1/2},$$

where $\mathbf{D}^{-1/2} = \text{diag}(1/s_1, \ldots, 1/s_q)$ and $s_i = \sqrt{s_i^2}$ is the sample standard deviation of variable i. (In most situations considered in this book, we will be dealing with covariance and correlation matrices of full rank, q, so that both matrices will be *non-singular*, that is, invertible, to give matrices $\mathbf{S}^{-1}$ or $\mathbf{R}^{-1}$.)

The sample correlation matrix for the three variables in Table 1.1 is obtained by using the function `cor()` in R:

```
R> cor(measure[, c("chest", "waist", "hips")])

       chest  waist   hips
chest 1.0000 0.6987 0.4778
waist 0.6987 1.0000 0.4147
hips  0.4778 0.4147 1.0000
```

Distances

For some multivariate techniques such as multidimensional scaling and cluster analysis the concept of distance between the units in the data is often of considerable interest and importance. So, given the variable values for two units, say unit i and unit j, what serves as a measure of distance between them? The most common measure used is Euclidean distance, which is defined as

The multivariate normal density function

For a vector of q variables, $\mathbf{x}^\top = (x_1, x_2, \ldots, x_q)$, the multivariate normal density function takes the form

$$f(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-q/2} \det(\boldsymbol{\Sigma})^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right\},$$

where $\boldsymbol{\Sigma}$ is the population covariance matrix of the variables and $\boldsymbol{\mu}$ is the vector of population mean values of the variables. The simplest example of the *multivariate normal density function* is the bivariate normal density with $q = 2$; this can be written explicitly as

$$\begin{aligned} &f((x_1, x_2); (\mu_1, \mu_2), \sigma_1, \sigma_2, \rho) = \\ &\quad (2\pi\sigma_1\sigma_2(1-\rho^2))^{-1/2} \exp\left\{-\frac{1}{2(1-\rho^2)} \times \right. \\ &\quad \left. \left(\left(\frac{x_1 - \mu_1}{\sigma_1}\right)^2 - 2\rho \frac{x_1 - \mu_1}{\sigma_1} \frac{x_2 - \mu_2}{\sigma_2} + \left(\frac{x_2 - \mu_2}{\sigma_2}\right)^2\right)\right\}, \end{aligned}$$

where μ_1 and μ_2 are the population means of the two variables, σ_1^2 and σ_2^2 are the population variances, and ρ is the population correlation between the two variables X_1 and X_2. Figure 1.1 shows an example of a bivariate normal density function with both means equal to zero, both variances equal to one, and correlation equal to 0.5.

The population mean vector and the population covariance matrix of a multivariate density function are estimated from a sample of multivariate observations as described in the previous subsections.

One property of a multivariate normal density function that is worth mentioning here is that *linear combinations* of the variables (i.e., $y = a_1X_1 + a_2X_2 + \cdots + a_qX_q$, where $a_1, a_2, \ldots, a_q$ is a set of scalars) are themselves normally distributed with mean $\mathrm{a}^\top \boldsymbol{\mu}$ and variance $\mathrm{a}^\top \Sigma \mathrm{a}$, where $\mathrm{a}^\top = (a_1, a_2, \ldots, a_q)$. Linear combinations of variables will be of importance

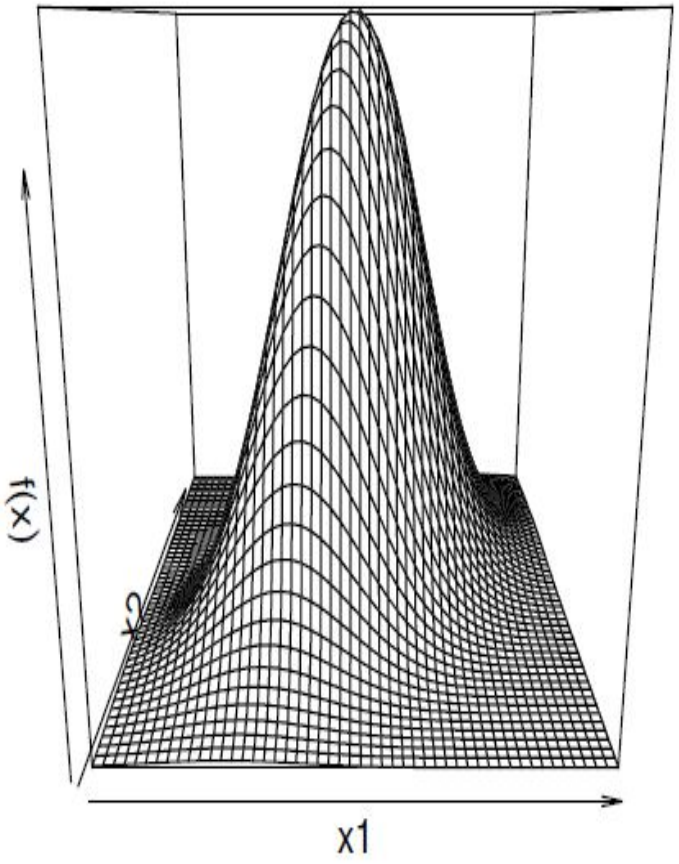


Fig. 1.1. Bivariate normal density function with correlation $\rho = 0.5$.

For many multivariate methods to be described in later chapters, the assumption of multivariate normality is not critical to the results of the analysis, but there may be occasions when testing for multivariate normality may be of interest. A start can be made perhaps by assessing each variable separately for univariate normality using a probability plot . Such plots are commonly applied in univariate analysis and involve ordering the observations and then plotting them against the appropriate values of an assumed cumulative distribution function. There are two basic types of plots for comparing two probability distributions, the probability-probability plot and the quantile-quantile plot. The diagram in Figure 1.2 may be used for describing each type.

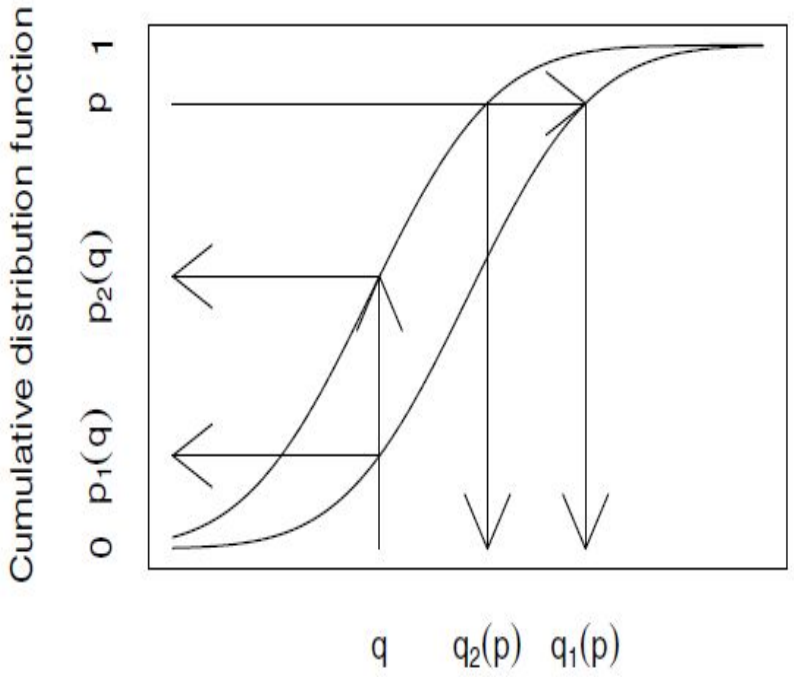


Fig. 1.2. Cumulative distribution functions and quantiles.

A plot of points whose coordinates are the cumulative probabilities $p_1(q)$ and $p_2(q)$ for different values of q with

$$p_1(q) = \mathsf{P}(X_1 \leq q),$$
$$p_2(q) = \mathsf{P}(X_2 \leq q),$$

for random variables X_1 and X_2 is a probability-probability plot, while a plot of the points whose coordinates are the quantiles $(q_1(p), q_2(p))$ for different values of p with

$$q_1(p) = p_1^{-1}(p),$$
$$q_2(p) = p_2^{-1}(p),$$

is a quantile-quantile plot. For example, a quantile-quantile plot for investigating the assumption that a set of data is from a normal distribution would involve plotting the ordered sample values of variable 1 (i.e.,$x_{(1)1}, x_{(2)1}, \ldots, x_{(n)1}$) against the quantiles of a standard normal distribution, $\Phi^{-1}(p(i))$, where usually

$$p_i = \frac{i - \frac{1}{2}}{n} \quad \Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2} du.$$

This is known as a *normal probability plot*.

For multivariate data, normal probability plots may be used to examine each variable separately, although marginal normality does not necessarily imply that the variables follow a multivariate normal distribution. Alternatively (or additionally), each multivariate observation might be converted to a single number in some way before plotting. For example, in the specific case of assessing a data set for multivariate normality, each q-dimensional observation, $\mathbf{x}_i$, could be converted into a *generalised distance*, d_i^2, giving a measure of the distance of the particular observation from the mean vector of the complete sample, $\bar{\mathbf{x}}$; d_i^2 is calculated as

$$d_i^2 = (\mathbf{x}_i - \bar{\mathbf{x}})^\top \mathbf{S}^{-1} (\mathbf{x}_i - \bar{\mathbf{x}}),$$

where $\mathbf{S}$ is the sample covariance matrix. This distance measure takes into account the different variances of the variables and the covariances of pairs of variables. If the observations do arise from a multivariate normal distribution, then these distances have approximately a *chi-squared distribution* with q degrees of freedom, also denoted by the symbol χ_q^2. So plotting the ordered distances against the corresponding quantiles of the appropriate chi-square distribution should lead to a straight line through the origin.

We will now assess the body measurements data in Table 1.2 for normality, although because there are only 20 observations in the sample there is really too little information to come to any convincing conclusion. Figure 1.3 shows separate probability plots for each measurement; there appears to be no evidence of any departures from linearity. The chi-square plot of the 20 generalised distances in Figure 1.4 does seem to deviate a little from linearity, but with so few observations it is hard to be certain. The plot is set up as follows. We first extract the relevant data

```
R> x <- measure[, c("chest", "waist", "hips")]
```

and estimate the means of all three variables (i.e., for each column of the data) and the covariance matrix

```
R> cm <- colMeans(x)
R> S <- cov(x)
```

The differences d_i have to be computed for all units in our data, so we iterate over the rows of `x` using the `apply()` function with argument `MARGIN = 1` and, for each row, compute the distance d_i:

```
R> d <- apply(x, MARGIN = 1, function(x)
+             t(x - cm) %*% solve(S) %*% (x - cm))
```

The sorted distances can now be plotted against the appropriate quantiles of the χ^2_3 distribution obtained from `qchisq()`; see Figure 1.4.

```
R> qqnorm(measure[,"chest"], main = "chest"); qqline(measure[,"chest"])
R> qqnorm(measure[,"waist"], main = "waist"); qqline(measure[,"waist"])
R> qqnorm(measure[,"hips"], main = "hips"); qqline(measure[,"hips"])
```

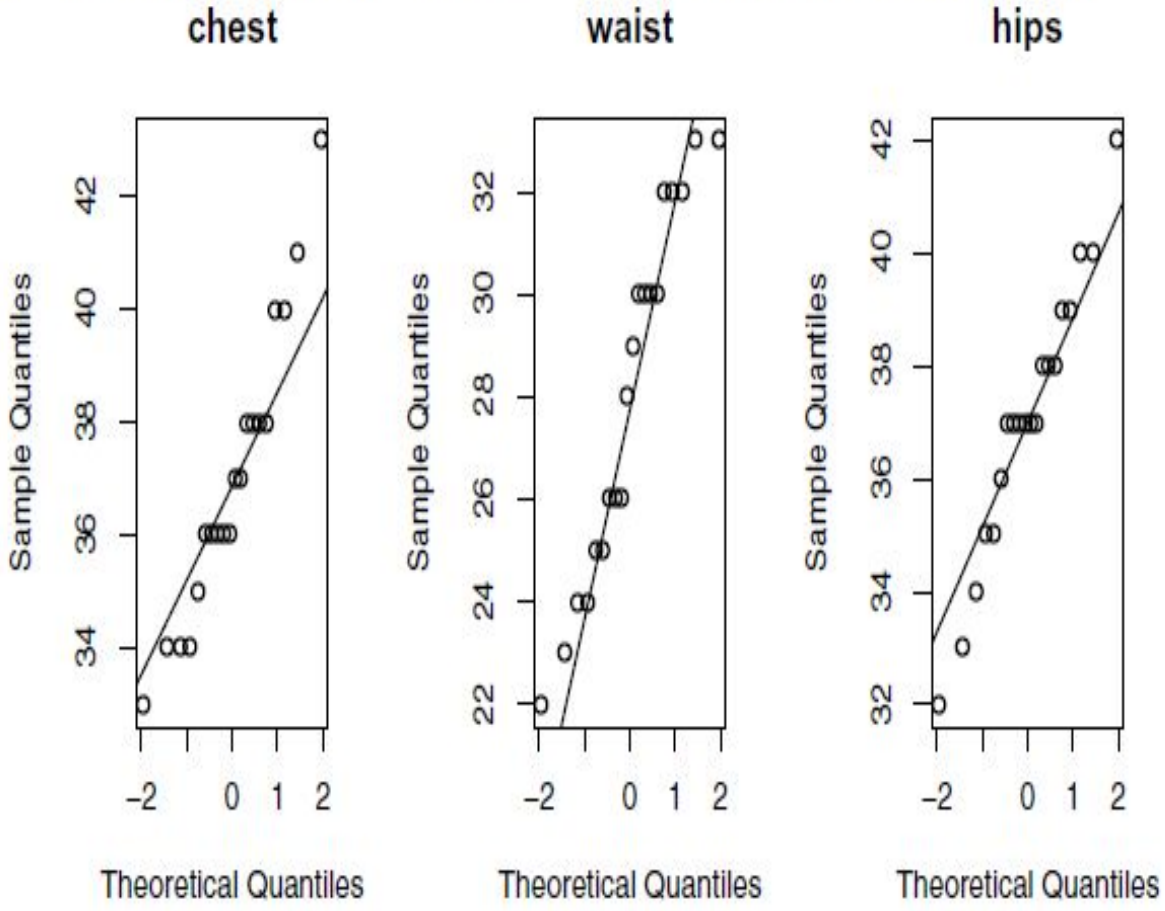


Fig. 1.3. Normal probability plots of chest, waist, and hip measurements.

```
R> plot(qchisq((1:nrow(x) - 1/2) / nrow(x), df = 3), sort(d),
+       xlab = expression(paste(chi[3]^2, " Quantile")),
+       ylab = "Ordered distances")
R> abline(a = 0, b = 1)
```

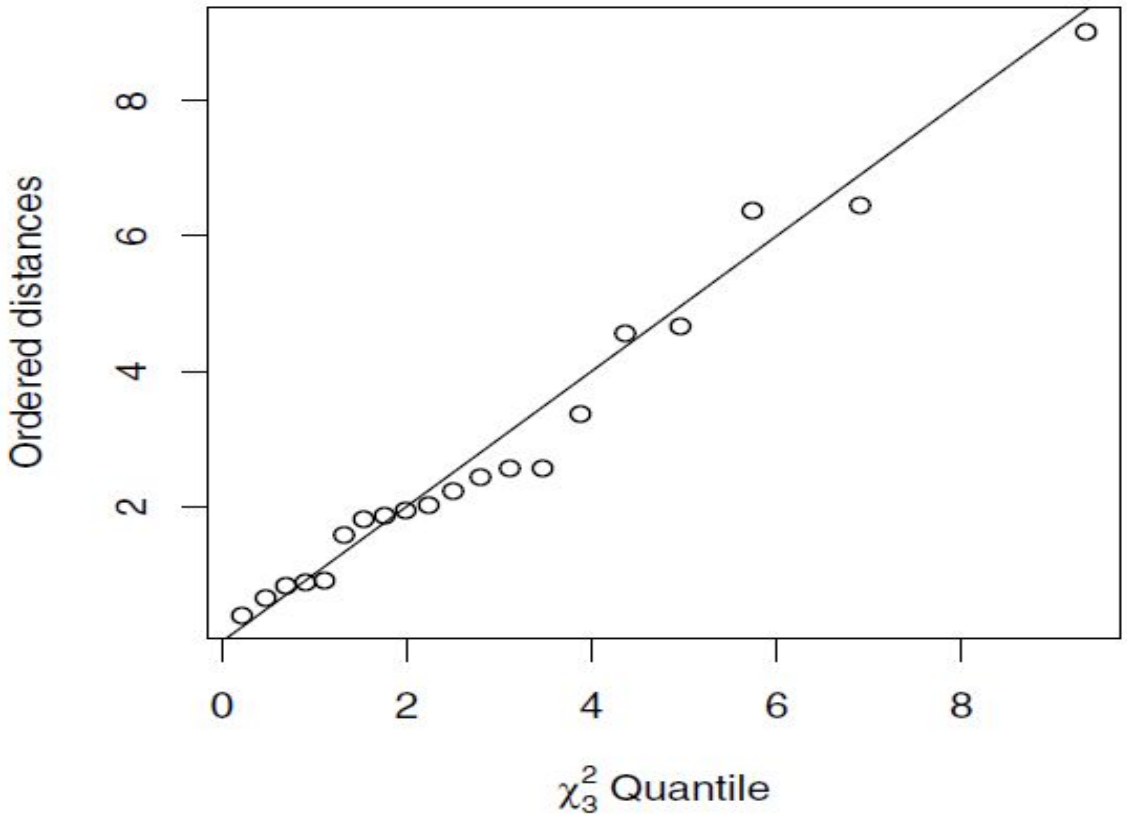


Fig. 1.4. Chi-square plot of generalised distances for body measurements data.

The final set of data we shall consider in this section was collected in a study of air pollution in cities in the USA. The following variables were obtained for 41 US cities:

`SO2`: SO_2 content of air in micrograms per cubic metre;
`temp`: average annual temperature in degrees Fahrenheit;
`manu`: number of manufacturing enterprises employing 20 or more workers;
`popul`: population size (1970 census) in thousands;
`wind`: average annual wind speed in miles per hour;
`precip`: average annual precipitation in inches;
`predays`: average number of days with precipitation per year.

Table 1.5: `USairpollution` data. Air pollution in 41 US cities.

	SO2	temp	manu	popul	wind	precip	predays
Albany	46	47.6	44	116	8.8	33.36	135
Albuquerque	11	56.8	46	244	8.9	7.77	58
Atlanta	24	61.5	368	497	9.1	48.34	115
Baltimore	47	55.0	625	905	9.6	41.31	111
Buffalo	11	47.1	391	463	12.4	36.11	166
Charleston	31	55.2	35	71	6.5	40.75	148
Chicago	110	50.6	3344	3369	10.4	34.44	122
Cincinnati	23	54.0	462	453	7.1	39.04	132
Cleveland	65	49.7	1007	751	10.9	34.99	155
Columbus	26	51.5	266	540	8.6	37.01	134
Dallas	9	66.2	641	844	10.9	35.94	78
Denver	17	51.9	454	515	9.0	12.95	86
Des Moines	17	49.0	104	201	11.2	30.85	103
Detroit	35	49.9	1064	1513	10.1	30.96	129
Hartford	56	49.1	412	158	9.0	43.37	127

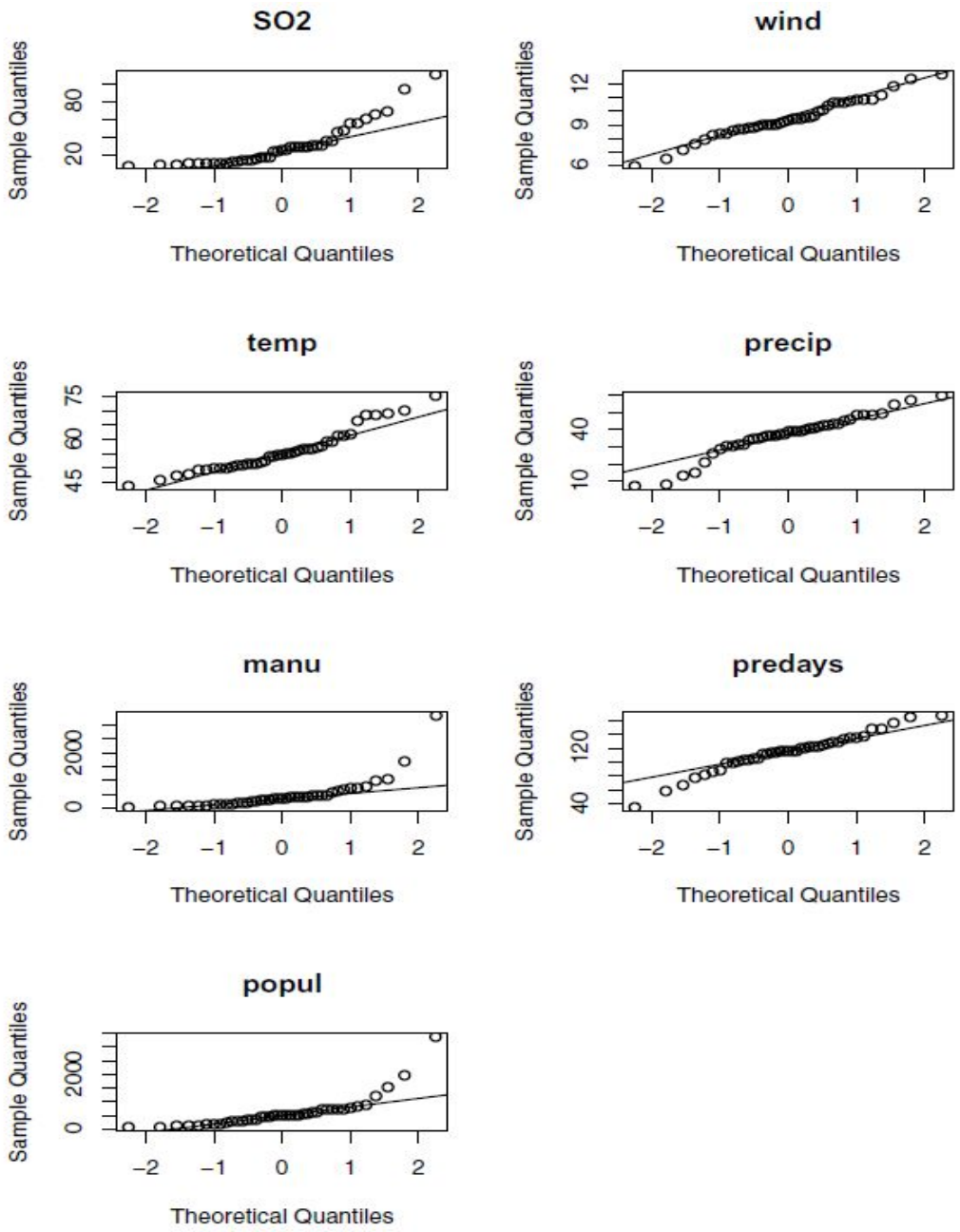


Fig. 1.5. Normal probability plots for `USairpollution` data.

The resulting seven plots are arranged on one page by a call to the `layou` matrix; see Figure 1.5. The plots for SO_2 concentration and precipitatio both deviate considerably from linearity, and the plots for manufacturing an population show evidence of a number of outliers. But of more importanc is the chi-square plot for the data, which is given in Figure 1.6; the R coc is identical to the code used to produce the chi-square plot for the bod measurement data. In addition, the two most extreme points in the plot hav been labelled with the city names to which they correspond using `text()`.

```
R> x <- USairpollution
R> cm <- colMeans(x)
R> S <- cov(x)
R> d <- apply(x, 1, function(x) t(x - cm) %*% solve(S) %*% (x - cm))
R> plot(qc <- qchisq((1:nrow(x) - 1/2) / nrow(x), df = 6),
+        sd <- sort(d),
+        xlab = expression(paste(chi[6]^2, " Quantile")),
+        ylab = "Ordered distances", xlim = range(qc) * c(1, 1.1))
R> oups <- which(rank(abs(qc - sd), ties = "random") > nrow(x) - 3)
R> text(qc[oups], sd[oups] - 1.5, names(oups))
R> abline(a = 0, b = 1)
```

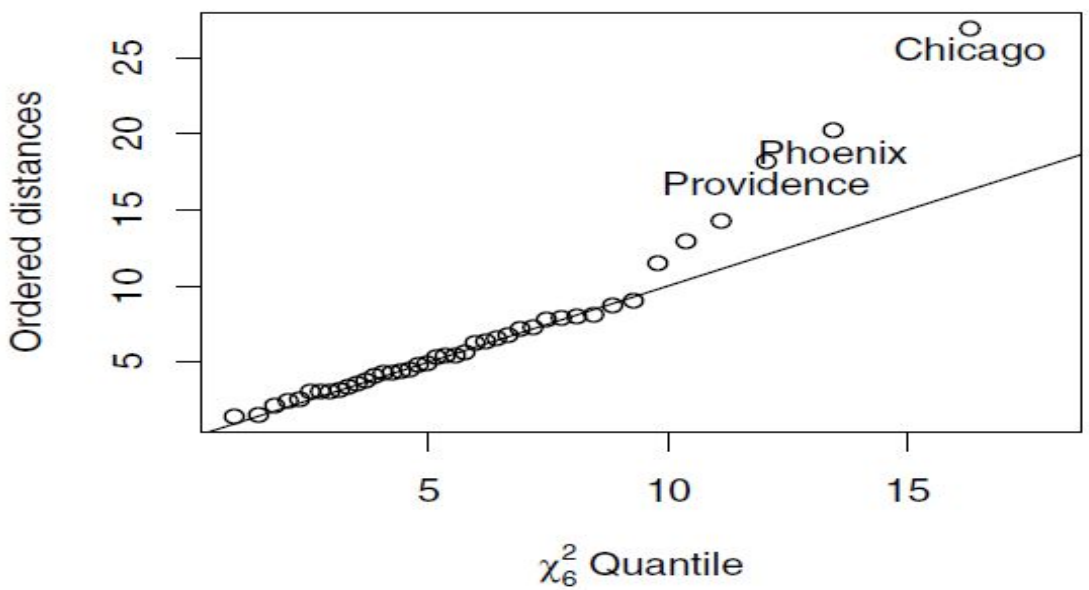


Fig. 1.6. χ^2 plot of generalised distances for `USairpollution` data.

This example illustrates that the chi-square plot might also be useful for detecting possible outliers in multivariate data, where informally outliers are "abnormal" in the sense of deviating from the natural data variability. Outlier identification is important in many applications of multivariate analysis either because there is some specific interest in finding anomalous observations or as a pre-processing task before the application of some multivariate method in order to preserve the results from possible misleading effects produced by these observations. A number of methods for identifying multivariate outliers have been suggested–see, for example, Rocke and Woodruff (1996)

Data graphics visually display measured quantities by means of the combined use of points, lines, a coordinate system, numbers, symbols, words, shading and color.

Some of the advantages of graphical methods have been listed by Schmid (1954):

- In comparison with other types of presentation, well-designed charts are more effective in creating interest and in appealing to the attention of the reader.
- Visual relationships as portrayed by charts and graphs are more easily grasped and more easily remembered.
- The use of charts and graphs saves time since the essential meaning of large measures of statistical data can be visualised at a glance.
- Charts and graphs provide a comprehensive picture of a problem that makes for a more complete and better balanced understanding than could be derived from tabular or textual forms of presentation.
- Charts and graphs can bring out hidden facts and relationships and can stimulate, as well as aid, analytical thinking and investigation.

```
R> pairs(USairpollution, pch = ".", cex = 1.5)
```

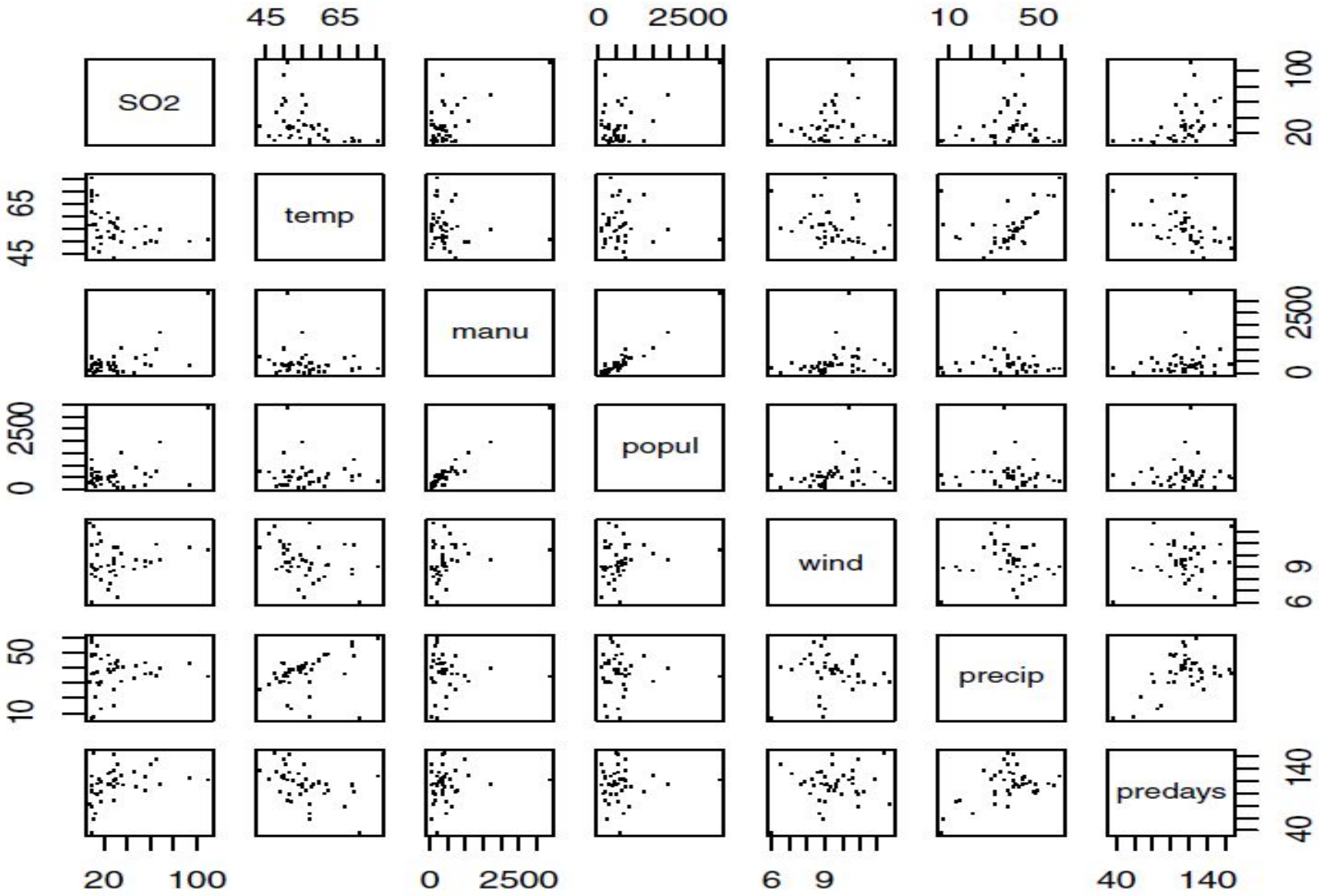


Fig. 2.10. Scatterplot matrix of the air pollution data.

```
R> round(cor(USairpollution), 4)

            SO2    temp    manu   popul    wind  precip predays
SO2      1.0000 -0.4336  0.6448  0.4938  0.0947  0.0543  0.3696
temp    -0.4336  1.0000 -0.1900 -0.0627 -0.3497  0.3863 -0.4302
manu     0.6448 -0.1900  1.0000  0.9553  0.2379 -0.0324  0.1318
popul    0.4938 -0.0627  0.9553  1.0000  0.2126 -0.0261  0.0421
wind     0.0947 -0.3497  0.2379  0.2126  1.0000 -0.0130  0.1641
precip   0.0543  0.3863 -0.0324 -0.0261 -0.0130  1.0000  0.4961
predays  0.3696 -0.4302  0.1318  0.0421  0.1641  0.4961  1.0000
```

Focussing on the correlations between `SO2` and the six other variables, we see that the correlation for `SO2` and `precip` is very small and that for `SO2` and `predays` is moderate.

Positive Definite Matrices

The study of the variation and interrelationships in multivariate data is often based upon distances and the assumption that the data are multivariate normally distributed. Squared distances and the multivariate normal density can be expressed in terms of matrix products called *quadratic forms* . Consequently, it should not be surprising that quadratic forms play a central role in multivariate analysis. We consider quadratic forms that are always nonnegative and the associated *positive definite* matrices. Results involving quadratic forms and symmetric matrices are, in many cases, a direct consequence of an expansion for symmetric matrices known as the *spectral decomposition.* The spectral decomposition of *k x k* symmetric matrix A is given by

$$\underset{(k\times k)}{\mathbf{A}} = \lambda_1 \underset{(k\times 1)}{\mathbf{e}_1}\underset{(1\times k)}{\mathbf{e}_1'} + \lambda_2 \underset{(k\times 1)}{\mathbf{e}_2}\underset{(1\times k)}{\mathbf{e}_2'} + \cdots + \lambda_k \underset{(k\times 1)}{\mathbf{e}_k}\underset{(1\times k)}{\mathbf{e}_k'} \qquad (2\text{-}16)$$

where $\lambda_1, \lambda_2, \ldots, \lambda_k$ are the eigenvalues of $\mathbf{A}$ and $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k$ are the associated normalized eigenvectors. (See also Result 2A.14 in Supplement 2A). Thus, $\mathbf{e}_i'\mathbf{e}_i = 1$ for $i = 1, 2, \ldots, k$, and $\mathbf{e}_i'\mathbf{e}_j = 0$ for $i \neq j$.

Example 2.10 (The spectral decomposition of a matrix) Consider the symmetric matrix

$$\mathbf{A} = \begin{bmatrix} 13 & -4 & 2 \\ -4 & 13 & -2 \\ 2 & -2 & 10 \end{bmatrix}$$

The eigenvalues obtained from the characteristic equation $|\mathbf{A} - \lambda\mathbf{I}| = 0$ are $\lambda_1 = 9$, $\lambda_2 = 9$, and $\lambda_3 = 18$ (Definition 2A.30). The corresponding eigenvectors $\mathbf{e}_1, \mathbf{e}_2$, and $\mathbf{e}_3$ are the (normalized) solutions of the equations $\mathbf{A}\mathbf{e}_i = \lambda_i\mathbf{e}_i$ for $i = 1, 2, 3$. Thus, $\mathbf{A}\mathbf{e}_1 = \lambda\mathbf{e}_1$ gives

$$\begin{bmatrix} 13 & -4 & 2 \\ -4 & 13 & -2 \\ 2 & -2 & 10 \end{bmatrix} \begin{bmatrix} e_{11} \\ e_{21} \\ e_{31} \end{bmatrix} = 9 \begin{bmatrix} e_{11} \\ e_{21} \\ e_{31} \end{bmatrix}$$

or

$$\begin{aligned} 13e_{11} - 4e_{21} + 2e_{31} &= 9e_{11} \\ -4e_{11} + 13e_{21} - 2e_{31} &= 9e_{21} \\ 2e_{11} - 2e_{21} + 10e_{31} &= 9e_{31} \end{aligned}$$

Moving the terms on the right of the equals sign to the left yields three homogeneous equations in three unknowns, but two of the equations are redundant. Selecting one of the equations and arbitrarily setting $e_{11} = 1$ and $e_{21} = 1$, we find that $e_{31} = 0$. Consequently, the normalized eigenvector is $\mathbf{e}_1' = [1/\sqrt{1^2 + 1^2 + 0^2}, 1/\sqrt{1^2 + 1^2 + 0^2}, 0/\sqrt{1^2 + 1^2 + 0^2}] = [1/\sqrt{2}, 1/\sqrt{2}, 0]$, since the sum of the squares of its elements is unity. You may verify that $\mathbf{e}_2' = [1/\sqrt{18}, -1/\sqrt{18}, -4/\sqrt{18}]$ is also an eigenvector for $9 = \lambda_2$, and $\mathbf{e}_3' = [2/3, -2/3, 1/3]$ is the normalized eigenvector corresponding to the eigenvalue $\lambda_3 = 18$. Moreover, $\mathbf{e}_i'\mathbf{e}_j = 0$ for $i \neq j$.

The spectral decomposition of $\mathbf{A}$ is then

$$\mathbf{A} = \lambda_1 \mathbf{e}_1 \mathbf{e}_1' + \lambda_2 \mathbf{e}_2 \mathbf{e}_2' + \lambda_3 \mathbf{e}_3 \mathbf{e}_3'$$

or

$$\begin{bmatrix} 13 & -4 & 2 \\ -4 & 13 & -2 \\ 2 & -2 & 10 \end{bmatrix} = 9 \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}$$

$$+ 9 \begin{bmatrix} \frac{1}{\sqrt{18}} \\ \frac{-1}{\sqrt{18}} \\ \frac{-4}{\sqrt{18}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{18}} & \frac{-1}{\sqrt{18}} & \frac{-4}{\sqrt{18}} \end{bmatrix} + 18 \begin{bmatrix} \frac{2}{3} \\ -\frac{2}{3} \\ \frac{1}{3} \end{bmatrix} \begin{bmatrix} \frac{2}{3} & -\frac{2}{3} & \frac{1}{3} \end{bmatrix}$$

$$= 9 \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 \end{bmatrix} + 9 \begin{bmatrix} \frac{1}{18} & -\frac{1}{18} & -\frac{4}{18} \\ -\frac{1}{18} & \frac{1}{18} & \frac{4}{18} \\ -\frac{4}{18} & \frac{4}{18} & \frac{16}{18} \end{bmatrix}$$

$$+ 18 \begin{bmatrix} \frac{4}{9} & -\frac{4}{9} & \frac{2}{9} \\ -\frac{4}{9} & \frac{4}{9} & -\frac{2}{9} \\ \frac{2}{9} & -\frac{2}{9} & \frac{1}{9} \end{bmatrix}$$

as you may readily verify. ∎

The spectral decomposition is an important analytical tool. With it, we are very easily able to demonstrate certain statistical results. The first of these is a matrix explanation of distance, which we now develop.

Because $\mathbf{x}'\mathbf{A}\mathbf{x}$ has only squared terms x_i^2 and product terms $x_i x_k$, it is called a *quadratic form*. When a $k \times k$ symmetric matrix $\mathbf{A}$ is such that

$$0 \leq \mathbf{x}'\mathbf{A}\mathbf{x} \tag{2-17}$$

for all $\mathbf{x}' = [x_1, x_2, \ldots, x_k]$, both the matrix $\mathbf{A}$ and the quadratic form are said to be *nonnegative definite*. If equality holds in (2-17) only for the vector $\mathbf{x}' = [0, 0, \ldots, 0]$, then $\mathbf{A}$ or the quadratic form is said to be *positive definite*. In other words, $\mathbf{A}$ is positive definite if

$$0 < \mathbf{x}'\mathbf{A}\mathbf{x} \tag{2-18}$$

for all vectors $\mathbf{x} \neq \mathbf{0}$.

A Square-Root Matrix

The spectral decomposition allows us to express the inverse of a square matrix in terms of its eigenvalues and eigenvectors, and this leads to a useful *square-root matrix*.

Let $\mathbf{A}$ be a $k \times k$ positive definite matrix with the spectral decomposition $\mathbf{A} = \sum_{i=1}^{k} \lambda_i \mathbf{e}_i \mathbf{e}_i'$. Let the normalized eigenvectors be the columns of another matrix $\mathbf{P} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k]$. Then

$$\underset{(k\times k)}{\mathbf{A}} = \sum_{i=1}^{k} \lambda_i \underset{(k\times 1)}{\mathbf{e}_i} \underset{(1\times k)}{\mathbf{e}_i'} = \underset{(k\times k)}{\mathbf{P}} \underset{(k\times k)}{\mathbf{\Lambda}} \underset{(k\times k)}{\mathbf{P}'} \tag{2-20}$$

where $\mathbf{PP}' = \mathbf{P}'\mathbf{P} = \mathbf{I}$ and $\mathbf{\Lambda}$ is the diagonal matrix

$$\underset{(k\times k)}{\mathbf{\Lambda}} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_k \end{bmatrix} \quad \text{with } \lambda_i > 0$$

Thus,

$$\mathbf{A}^{-1} = \mathbf{P}\boldsymbol{\Lambda}^{-1}\mathbf{P}' = \sum_{i=1}^{k} \frac{1}{\lambda_i}\mathbf{e}_i\mathbf{e}_i' \tag{2-21}$$

since $(\mathbf{P}\boldsymbol{\Lambda}^{-1}\mathbf{P}')\mathbf{P}\boldsymbol{\Lambda}\mathbf{P}' = \mathbf{P}\boldsymbol{\Lambda}\mathbf{P}'(\mathbf{P}\boldsymbol{\Lambda}^{-1}\mathbf{P}') = \mathbf{P}\mathbf{P}' = \mathbf{I}$.

Next, let $\boldsymbol{\Lambda}^{1/2}$ denote the diagonal matrix with $\sqrt{\lambda_i}$ as the ith diagonal element. The matrix $\sum_{i=1}^{k} \sqrt{\lambda_i}\,\mathbf{e}_i\mathbf{e}_i' = \mathbf{P}\boldsymbol{\Lambda}^{1/2}\mathbf{P}'$ is called the *square root* of $\mathbf{A}$ and is denoted by $\mathbf{A}^{1/2}$.

The square-root matrix, of a positive definite matrix $\mathbf{A}$,

$$\mathbf{A}^{1/2} = \sum_{i=1}^{k} \sqrt{\lambda_i}\,\mathbf{e}_i\mathbf{e}_i' = \mathbf{P}\boldsymbol{\Lambda}^{1/2}\mathbf{P}' \tag{2-22}$$

has the following properties:

1. $(\mathbf{A}^{1/2})' = \mathbf{A}^{1/2}$ (that is, $\mathbf{A}^{1/2}$ is symmetric).
2. $\mathbf{A}^{1/2}\mathbf{A}^{1/2} = \mathbf{A}$.
3. $(\mathbf{A}^{1/2})^{-1} = \sum_{i=1}^{k} \frac{1}{\sqrt{\lambda_i}}\mathbf{e}_i\mathbf{e}_i' = \mathbf{P}\boldsymbol{\Lambda}^{-1/2}\mathbf{P}'$, where $\boldsymbol{\Lambda}^{-1/2}$ is a diagonal matrix with $1/\sqrt{\lambda_i}$ as the ith diagonal element.
4. $\mathbf{A}^{1/2}\mathbf{A}^{-1/2} = \mathbf{A}^{-1/2}\mathbf{A}^{1/2} = \mathbf{I}$, and $\mathbf{A}^{-1/2}\mathbf{A}^{-1/2} = \mathbf{A}^{-1}$, where $\mathbf{A}^{-1/2} = (\mathbf{A}^{1/2})^{-1}$.

Random Vectors and Matrices

A *random vector* is a vector whose elements are random variables. Similarly, a *random matrix* is a matrix whose elements are random variables. The expected value of a random matrix (or vector) is the matrix (vector) consisting of the expected values of each of its elements. Specifically, let $\mathbf{X} = \{X_{ij}\}$ be an $n \times p$ random matrix. Then the expected value of $\mathbf{X}$, denoted by $E(\mathbf{X})$, is the $n \times p$ matrix of numbers (if they exist)

$$E(\mathbf{X}) = \begin{bmatrix} E(X_{11}) & E(X_{12}) & \cdots & E(X_{1p}) \\ E(X_{21}) & E(X_{22}) & \cdots & E(X_{2p}) \\ \vdots & \vdots & \ddots & \vdots \\ E(X_{n1}) & E(X_{n2}) & \cdots & E(X_{np}) \end{bmatrix} \tag{2-23}$$

where, for each element of the matrix,[2]

$$E(X_{ij}) = \begin{cases} \displaystyle\int_{-\infty}^{\infty} x_{ij} f_{ij}(x_{ij})\, dx_{ij} & \text{if } X_{ij} \text{ is a continuous random variable with probability density function } f_{ij}(x_{ij}) \\ \displaystyle\sum_{\text{all } x_{ij}} x_{ij} p_{ij}(x_{ij}) & \text{if } X_{ij} \text{ is a discrete random variable with probability function } p_{ij}(x_{ij}) \end{cases}$$

Example 2.12 (Computing expected values for discrete random variables) Suppose $p = 2$ and $n = 1$, and consider the random vector $\mathbf{X}' = [X_1, X_2]$. Let the discrete random variable X_1 have the following probability function:

x_1	-1	0	1
$p_1(x_1)$	.3	.3	.4

Then $E(X_1) = \sum_{\text{all } x_1} x_1 p_1(x_1) = (-1)(.3) + (0)(.3) + (1)(.4) = .1.$

Similarly, let the discrete random variable X_2 have the probability function

x_2	0	1
$p_2(x_2)$	.8	.2

Then $E(X_2) = \sum_{\text{all } x_2} x_2 p_2(x_2) = (0)(.8) + (1)(.2) = .2.$

Thus,

$$E(\mathbf{X}) = \begin{bmatrix} E(X_1) \\ E(X_2) \end{bmatrix} = \begin{bmatrix} .1 \\ .2 \end{bmatrix}$$

■

Two results involving the expectation of sums and products of matrices follow directly from the definition of the expected value of a random matrix and the univariate properties of expectation, $E(X_1 + Y_1) = E(X_1) + E(Y_1)$ and $E(cX_1) = cE(X_1)$. Let $\mathbf{X}$ and $\mathbf{Y}$ be random matrices of the same dimension, and let $\mathbf{A}$ and $\mathbf{B}$ be conformable matrices of constants. Then (see Exercise 2.40)

$$E(\mathbf{X} + \mathbf{Y}) = E(\mathbf{X}) + E(\mathbf{Y}) \tag{2-24}$$

$$E(\mathbf{AXB}) = \mathbf{A}E(\mathbf{X})\mathbf{B}$$

Mean Vectors and Covariance Matrices

Suppose $\mathbf{X}' = [X_1, X_2, \ldots, X_p]$ is a $p \times 1$ random vector. Then each element of $\mathbf{X}$ is a random variable with its own marginal probability distribution. (See Example 2.12.) The marginal means μ_i and variances σ_i^2 are defined as $\mu_i = E(X_i)$ and $\sigma_i^2 = E(X_i - \mu_i)^2$, $i = 1, 2, \ldots, p$, respectively. Specifically,

$$\mu_i = \begin{cases} \displaystyle\int_{-\infty}^{\infty} x_i f_i(x_i)\, dx_i & \text{if } X_i \text{ is a continuous random variable with probability density function } f_i(x_i) \\ \displaystyle\sum_{\text{all } x_i} x_i p_i(x_i) & \text{if } X_i \text{ is a discrete random variable with probability function } p_i(x_i) \end{cases}$$

$$\sigma_i^2 = \begin{cases} \displaystyle\int_{-\infty}^{\infty} (x_i - \mu_i)^2 f_i(x_i)\, dx_i & \text{if } X_i \text{ is a continuous random variable with probability density function } f_i(x_i) \\ \displaystyle\sum_{\text{all } x_i} (x_i - \mu_i)^2 p_i(x_i) & \text{if } X_i \text{ is a discrete random variable with probability function } p_i(x_i) \end{cases} \tag{2-25}$$

It will be convenient in later sections to denote the marginal variances by σ_{ii} rather than the more traditional σ_i^2, and consequently, we shall adopt this notation.

The behavior of any pair of random variables, such as X_i and X_k, is described by their joint probability function, and a measure of the linear association between them is provided by the covariance

$$\sigma_{ik} = E(X_i - \mu_i)(X_k - \mu_k)$$

$$= \begin{cases} \displaystyle\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x_i - \mu_i)(x_k - \mu_k) f_{ik}(x_i, x_k)\, dx_i\, dx_k & \text{if } X_i, X_k \text{ are continuous random variables with the joint density function } f_{ik}(x_i, x_k) \\ \displaystyle\sum_{\text{all } x_i}\sum_{\text{all } x_k} (x_i - \mu_i)(x_k - \mu_k) p_{ik}(x_i, x_k) & \text{if } X_i, X_k \text{ are discrete random variables with joint probability function } p_{ik}(x_i, x_k) \end{cases} \tag{2-26}$$

and μ_i and μ_k, $i, k = 1, 2, \ldots, p$, are the marginal means. When $i = k$, the covariance becomes the marginal variance.

More generally, the collective behavior of the p random variables $X_1, X_2, \ldots, X_p$ or, equivalently, the random vector $\mathbf{X}' = [X_1, X_2, \ldots, X_p]$, is described by a joint probability density function $f(x_1, x_2, \ldots, x_p) = f(\mathbf{x})$. As we have already noted in

f(x) will often be the multivariate normal density function.

If the joint probability $P[X_i \le x_i \text{ and } X_k \le x_k]$ can be written as the product of the corresponding marginal probabilities, so that

$$P[X_i \le x_i \text{ and } X_k \le x_k] = P[X_i \le x_i]P[X_k \le x_k] \tag{2-27}$$

for all pairs of values x_i, x_k, then X_i and X_k are said to be *statistically independent.* When X_i and X_k are continuous random variables with joint density $f_{ik}(x_i, x_k)$ and marginal densities $f_i(x_i)$ and $f_k(x_k)$, the independence condition becomes

$$f_{ik}(x_i, x_k) = f_i(x_i)f_k(x_k)$$

for all pairs (x_i, x_k).

The p continuous random variables $X_1, X_2, \ldots, X_p$ are *mutually statistically independent* if their joint density can be factored as

$$f_{12\cdots p}(x_1, x_2, \ldots, x_p) = f_1(x_1)f_2(x_2)\cdots f_p(x_p) \tag{2-28}$$

for all p-tuples $(x_1, x_2, \ldots, x_p)$.

Statistical independence has an important implication for covariance. The factorization in (2-28) implies that $\text{Cov}(X_i, X_k) = 0$. Thus,

$$\text{Cov}(X_i, X_k) = 0 \qquad \text{if } X_i \text{ and } X_k \text{ are independent} \tag{2-29}$$

The converse of (2-29) is not true in general; there are situations where $\mathrm{Cov}(X_i, X_k) = 0$, but X_i and X_k are not independent.

The means and covariances of the $p \times 1$ random vector $\mathbf{X}$ can be set out as matrices. The expected value of each element is contained in the vector of means $\boldsymbol{\mu} = E(\mathbf{X})$, and the p variances σ_{ii} and the $p(p-1)/2$ distinct covariances $\sigma_{ik}(i < k)$ are contained in the symmetric variance-covariance matrix $\boldsymbol{\Sigma} = E(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})'$. Specifically,

$$E(\mathbf{X}) = \begin{bmatrix} E(X_1) \\ E(X_2) \\ \vdots \\ E(X_p) \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix} = \boldsymbol{\mu} \tag{2-30}$$

and

$$= E(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})'$$

$$= E\left(\begin{bmatrix} X_1 - \mu_1 \\ X_2 - \mu_2 \\ \vdots \\ X_p - \mu_p \end{bmatrix} [X_1 - \mu_1, X_2 - \mu_2, \ldots, X_p - \mu_p] \right)$$

$$= E\begin{bmatrix} (X_1 - \mu_1)^2 & (X_1 - \mu_1)(X_2 - \mu_2) & \cdots & (X_1 - \mu_1)(X_p - \mu_p) \\ (X_2 - \mu_2)(X_1 - \mu_1) & (X_2 - \mu_2)^2 & \cdots & (X_2 - \mu_2)(X_p - \mu_p) \\ \vdots & \vdots & \ddots & \vdots \\ (X_p - \mu_p)(X_1 - \mu_1) & (X_p - \mu_p)(X_2 - \mu_2) & \cdots & (X_p - \mu_p)^2 \end{bmatrix}$$

$$
= \begin{bmatrix} E(X_1-\mu_1)^2 & E(X_1-\mu_1)(X_2-\mu_2) & \cdots & E(X_1-\mu_1)(X_p-\mu_p) \\ E(X_2-\mu_2)(X_1-\mu_1) & E(X_2-\mu_2)^2 & \cdots & E(X_2-\mu_2)(X_p-\mu_p) \\ \vdots & \vdots & \ddots & \vdots \\ E(X_p-\mu_p)(X_1-\mu_1) & E(X_p-\mu_p)(X_2-\mu_2) & \cdots & E(X_p-\mu_p)^2 \end{bmatrix}
$$

or

$$
\boldsymbol{\Sigma} = \mathrm{Cov}(\mathbf{X}) = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \cdots & \sigma_{pp} \end{bmatrix} \tag{2-31}
$$

Example 2.13 (Computing the covariance matrix) Find the covariance matrix for the two random variables X_1 and X_2 introduced in Example 2.12 when their joint probability function $p_{12}(x_1, x_2)$ is represented by the entries in the body of the following table:

x_1 \ x_2	0	1	$p_1(x_1)$
-1	.24	.06	.3
0	.16	.14	.3
1	.40	.00	.4
$p_2(x_2)$	.8	.2	1

We have already shown that $\mu_1 = E(X_1) = .1$ and $\mu_2 = E(X_2) = .2$. (See Example 2.12.) In addition,

$$\sigma_{11} = E(X_1 - \mu_1)^2 = \sum_{\text{all } x_1} (x_1 - .1)^2 p_1(x_1)$$

$$= (-1 - .1)^2(.3) + (0 - .1)^2(.3) + (1 - .1)^2(.4) = .69$$

$$\sigma_{22} = E(X_2 - \mu_2)^2 = \sum_{\text{all } x_2} (x_2 - .2)^2 p_2(x_2)$$

$$= (0 - .2)^2(.8) + (1 - .2)^2(.2)$$

$$= .16$$

$$\sigma_{12} = E(X_1 - \mu_1)(X_2 - \mu_2) = \sum_{\text{all pairs } (x_1, x_2)} (x_1 - .1)(x_2 - .2) p_{12}(x_1, x_2)$$

$$= (-1 - .1)(0 - .2)(.24) + (-1 - .1)(1 - .2)(.06)$$

$$+ \cdots + (1 - .1)(1 - .2)(.00) = -.08$$

$$\sigma_{21} = E(X_2 - \mu_2)(X_1 - \mu_1) = E(X_1 - \mu_1)(X_2 - \mu_2) = \sigma_{12} = -.08$$

Consequently, with $\mathbf{X}' = [X_1, X_2]$,

$$\boldsymbol{\mu} = E(\mathbf{X}) = \begin{bmatrix} E(X_1) \\ E(X_2) \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} .1 \\ .2 \end{bmatrix}$$

and

$$
\begin{aligned}
\boldsymbol{\Sigma} &= E(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})' \\
&= E\begin{bmatrix} (X_1 - \mu_1)^2 & (X_1 - \mu_1)(X_2 - \mu_2) \\ (X_2 - \mu_2)(X_1 - \mu_1) & (X_2 - \mu_2)^2 \end{bmatrix} \\
&= \begin{bmatrix} E(X_1 - \mu_1)^2 & E(X_1 - \mu_1)(X_2 - \mu_2) \\ E(X_2 - \mu_2)(X_1 - \mu_1) & E(X_2 - \mu_2)^2 \end{bmatrix} \\
&= \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix} = \begin{bmatrix} .69 & -.08 \\ -.08 & .16 \end{bmatrix}
\end{aligned}
$$

■

We note that the computation of means, variances, and covariances for *discrete* random variables involves summation (as in Examples 2.12 and 2.13), while analogous computations for *continuous* random variables involve integration.

Because $\sigma_{ik} = E(X_i - \mu_i)(X_k - \mu_k) = \sigma_{ki}$, it is convenient to write the matrix appearing in (2-31) as

$$
\boldsymbol{\Sigma} = E(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})' = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{12} & \sigma_{22} & \cdots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1p} & \sigma_{2p} & \cdots & \sigma_{pp} \end{bmatrix} \tag{2-32}
$$

We shall refer to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ as the *population mean* (vector) and *population variance–covariance* (matrix), respectively.

Properties of the Covariance Matrix $\Sigma = \mathsf{Var}(X)$

$$\Sigma = (\sigma_{X_i X_j}), \quad \sigma_{X_i X_j} = \mathsf{Cov}(X_i, X_j), \quad \sigma_{X_i X_i} = \mathsf{Var}(X_i)$$

$$\Sigma = \mathsf{E}(XX^\top) - \mu\mu^\top$$

$$\Sigma \geq 0$$

$$\mathsf{Var}(a^\top X) = a^\top \mathsf{Var}(X) a = \sum_{i,j} a_i a_j \sigma_{X_i X_j}$$

$$\mathsf{Var}(\mathcal{A}X + b) = \mathcal{A}\,\mathsf{Var}(X)\mathcal{A}^\top$$

$$\mathsf{Cov}(X + Y, Z) = \mathsf{Cov}(X, Z) + \mathsf{Cov}(Y, Z)$$

$$\mathsf{Var}(X + Y) = \mathsf{Var}(X) + \mathsf{Cov}(X, Y) + \mathsf{Cov}(Y, X) + \mathsf{Var}(Y)$$

$$\mathsf{Cov}(\mathcal{A}X, \mathcal{B}Y) = \mathcal{A}\,\mathsf{Cov}(X, Y)\mathcal{B}^\top.$$

Theorem 4.6 *Let $X \sim N_p(\mu, \Sigma)$ and $\mathcal{A}(p \times p)$, $c \in \mathbb{R}^p$, where $\mathcal{A}$ is nonsingular. Then $Y = \mathcal{A}X + c$ is again a p-variate Normal, i.e.*

$$Y \sim N_p(\mathcal{A}\mu + c, \mathcal{A}\Sigma\mathcal{A}^\top). \tag{4.50}$$

The multinormal distribution with mean μ and covariance $\Sigma > 0$ has the density

$$f(x) = |2\pi\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right\}. \tag{4.47}$$

We write $X \sim N_p(\mu, \Sigma)$.

How is this multinormal distribution with mean μ and covariance Σ related to the multivariate standard normal $N_p(0, \mathcal{I}_p)$? Through a linear transformation using

Theorem 4.5 *Let $X \sim N_p(\mu, \Sigma)$ and $Y = \Sigma^{-1/2}(X - \mu)$ (Mahalanobis transformation). Then*

$$Y \sim N_p(0, \mathcal{I}_p),$$

i.e. the elements $Y_j \in \mathbb{R}$ are independent, one-dimensional $N(0, 1)$ variables.

How can we create $N_p(\mu, \Sigma)$ variables on the basis of $N_p(0, \mathcal{I}_p)$ variables? We use the inverse linear transformation

$$X = \Sigma^{1/2} Y + \mu. \tag{4.49}$$

we can also check that $\mathsf{E}(X) = \mu$ and $\mathsf{Var}(X) = \Sigma$.

Theorem 4.7 *If $X \sim N_p(\mu, \Sigma)$, then the variable $U = (X-\mu)^\top \Sigma^{-1}(X-\mu)$ has a χ^2_p distribution.*

Theorem 4.8 *The characteristic function (cf) of a multinormal $N_p(\mu, \Sigma)$ is given by*

$$\varphi_X(t) = \exp\left(i\, t^\top \mu - \frac{1}{2} t^\top \Sigma t\right). \tag{4.52}$$

Geometry of the $N_p(\mu, \Sigma)$ Distribution

From (4.47) we see that the density of the $N_p(\mu, \Sigma)$ distribution is constant on ellipsoids of the form

$$(x-\mu)^\top \Sigma^{-1}(x-\mu) = d^2. \tag{4.51}$$

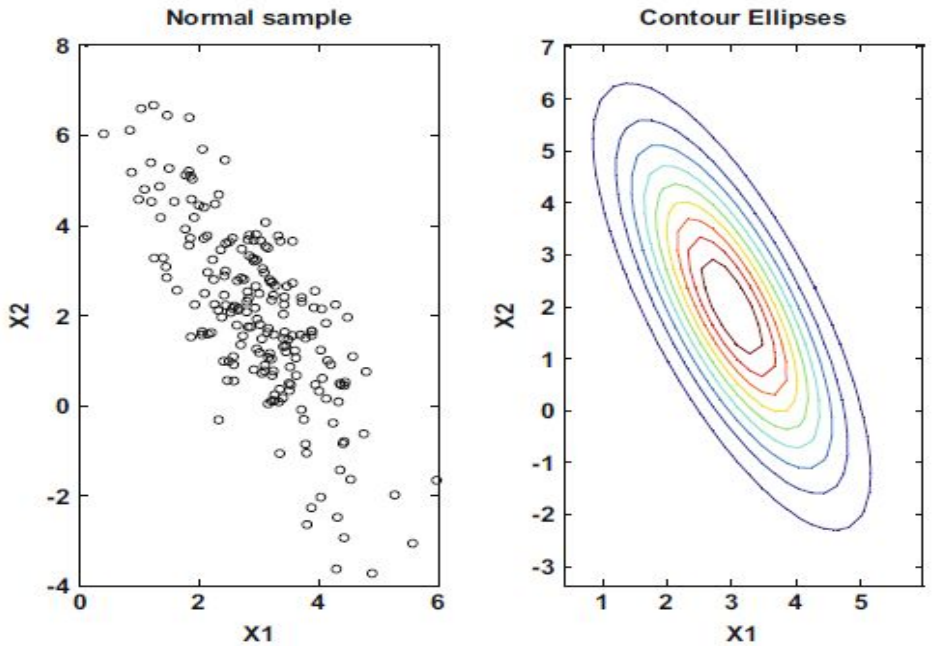


Fig. 4.3 Scatterplot of a normal sample and contour ellipses for $\mu = \binom{3}{2}$ and $\Sigma = \begin{pmatrix} 1 & -1.5 \\ -1.5 & 4 \end{pmatrix}$
MVAcontnorm

Singular Normal Distribution

Suppose that we have $\text{rank}(\Sigma) = k < p$, where p is the dimension of X. We define the (singular) density of X with the aid of the G-Inverse Σ^- of Σ,

$$f(x) = \frac{(2\pi)^{-k/2}}{(\lambda_1 \cdots \lambda_k)^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu)^\top \Sigma^- (x-\mu)\right\} \tag{4.53}$$

where

1. x lies on the hyperplane $\mathcal{N}^\top(x-\mu) = 0$ with $\mathcal{N}(p \times (p-k)) : \mathcal{N}^\top \Sigma = 0$ and $\mathcal{N}^\top \mathcal{N} = \mathcal{I}_k$.
2. Σ^- is the G-Inverse of Σ, and $\lambda_1, \ldots, \lambda_k$ are the nonzero eigenvalues of Σ.

What is the connection to a multinormal with k-dimensions? If

$$Y \sim N_k(0, \Lambda_1) \text{ and } \Lambda_1 = \text{diag}(\lambda_1, \ldots, \lambda_k), \tag{4.54}$$

then an orthogonal matrix $\mathcal{B}(p \times k)$ with $\mathcal{B}^\top \mathcal{B} = \mathcal{I}_k$ exists that means $X = \mathcal{B}Y + \mu$ where X has a singular pdf of the form (4.53).

It is frequently informative to separate the information contained in variances σ_{ii} from that contained in measures of association and, in particular, the measure of association known as the *population correlation coefficient* ρ_{ik}. The correlation coefficient ρ_{ik} is defined in terms of the covariance σ_{ik} and variances σ_{ii} and σ_{kk} as

$$\rho_{ik} = \frac{\sigma_{ik}}{\sqrt{\sigma_{ii}}\sqrt{\sigma_{kk}}} \tag{2-33}$$

The correlation coefficient measures the amount of *linear* association between the random variables X_i and X_k.

Let the population correlation matrix be the $p \times p$ symmetric matrix

$$\boldsymbol{\rho} = \begin{bmatrix} \frac{\sigma_{11}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{11}}} & \frac{\sigma_{12}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{22}}} & \cdots & \frac{\sigma_{1p}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{pp}}} \\ \frac{\sigma_{12}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{22}}} & \frac{\sigma_{22}}{\sqrt{\sigma_{22}}\sqrt{\sigma_{22}}} & \cdots & \frac{\sigma_{2p}}{\sqrt{\sigma_{22}}\sqrt{\sigma_{pp}}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\sigma_{1p}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{pp}}} & \frac{\sigma_{2p}}{\sqrt{\sigma_{22}}\sqrt{\sigma_{pp}}} & \cdots & \frac{\sigma_{pp}}{\sqrt{\sigma_{pp}}\sqrt{\sigma_{pp}}} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1p} \\ \rho_{12} & 1 & \cdots & \rho_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1p} & \rho_{2p} & \cdots & 1 \end{bmatrix} \tag{2-34}$$

and let the $p \times p$ *standard deviation* matrix be

$$\mathbf{V}^{1/2} = \begin{bmatrix} \sqrt{\sigma_{11}} & 0 & \cdots & 0 \\ 0 & \sqrt{\sigma_{22}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{\sigma_{pp}} \end{bmatrix} \tag{2-35}$$

Then it is easily verified (see Exercise 2.23) that

$$\mathbf{V}^{1/2}\boldsymbol{\rho}\mathbf{V}^{1/2} = \boldsymbol{\Sigma} \tag{2-36}$$

and

$$\boldsymbol{\rho} = (\mathbf{V}^{1/2})^{-1}\boldsymbol{\Sigma}(\mathbf{V}^{1/2})^{-1} \tag{2-37}$$

That is, $\boldsymbol{\Sigma}$ can be obtained from $\mathbf{V}^{1/2}$ and $\boldsymbol{\rho}$, whereas $\boldsymbol{\rho}$ can be obtained from $\boldsymbol{\Sigma}$. Moreover, the expression of these relationships in terms of matrix operations allows the calculations to be conveniently implemented on a computer.

The advantage of the correlation is that it is independent of the scales of the two variables. The correlation coefficient lies between -1 and $+1$ and gives a measure of the *linear* relationship of the variables X_i and X_j. It is positive if high values of X_i are associated with high values of X_j and negative if high values of X_i are associated with low values of X_j. If the relationship between two variables is non-linear, their correlation coefficient can be misleading.

With q variables there are $q(q-1)/2$ distinct correlations, which may be arranged in a $q \times q$ correlation matrix the diagonal elements of which are unity. For observed data, the correlation matrix contains the usual estimates of the ρs, namely Pearson's correlation coefficient, and is generally denoted by $\mathbf{R}$. The matrix may be written in terms of the sample covariance matrix $\mathbf{S}$

$$\mathbf{R} = \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{-1/2},$$

where $\mathbf{D}^{-1/2} = \mathrm{diag}(1/s_1, \ldots, 1/s_q)$ and $s_i = \sqrt{s_i^2}$ is the sample standard deviation of variable i. (In most situations considered in this book, we will be dealing with covariance and correlation matrices of full rank, q, so that both matrices will be *non-singular*, that is, invertible, to give matrices $\mathbf{S}^{-1}$ or $\mathbf{R}^{-1}$.)

The sample correlation matrix for the three variables in Table 1.1 is obtained by using the function `cor()` in R:

```
R> cor(measure[, c("chest", "waist", "hips")])
```

```
       chest  waist   hips
chest 1.0000 0.6987 0.4778
waist 0.6987 1.0000 0.4147
hips  0.4778 0.4147 1.0000
```

Example 2.14 (Computing the correlation matrix from the covariance matrix)

Suppose

$$\Sigma = \begin{bmatrix} 4 & 1 & 2 \\ 1 & 9 & -3 \\ 2 & -3 & 25 \end{bmatrix} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{bmatrix}$$

Obtain $\mathbf{V}^{1/2}$ and $\boldsymbol{\rho}$.

Here

$$\mathbf{V}^{1/2} = \begin{bmatrix} \sqrt{\sigma_{11}} & 0 & 0 \\ 0 & \sqrt{\sigma_{22}} & 0 \\ 0 & 0 & \sqrt{\sigma_{33}} \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{bmatrix}$$

and

$$(\mathbf{V}^{1/2})^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{5} \end{bmatrix}$$

Consequently, from (2-37), the correlation matrix $\boldsymbol{\rho}$ is given by

$$(\mathbf{V}^{1/2})^{-1}\Sigma(\mathbf{V}^{1/2})^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{5} \end{bmatrix} \begin{bmatrix} 4 & 1 & 2 \\ 1 & 9 & -3 \\ 2 & -3 & 25 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{5} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \frac{1}{6} & \frac{1}{5} \\ \frac{1}{6} & 1 & -\frac{1}{5} \\ \frac{1}{5} & -\frac{1}{5} & 1 \end{bmatrix}$$

The zero-order Pearson correlation between two random variables Y_i and Y_j is given by

$$\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j} = \frac{\operatorname{cov}\left(Y_i, Y_j\right)}{\sqrt{\operatorname{var}\left(Y_i\right) \operatorname{var}\left(Y_j\right)}} \quad \text{where} \quad -1 \leq \rho_{ij} \leq 1$$

The correlation matrix for the random p-vector $\mathbf{Y}$ is

$$\mathbf{P} = \left[\rho_{ij}\right] \tag{3.2.4}$$

Letting $(\operatorname{diag} \Sigma)^{-1/2}$ represent the diagonal matrix with diagonal elements equal to the square root of the diagonal elements of Σ, the relationship between $\mathbf{P}$ and Σ is established

$$\mathbf{P} = (\operatorname{diag} \Sigma)^{-1/2} \Sigma (\operatorname{diag} \Sigma)^{-1/2}$$
$$\Sigma = (\operatorname{diag} \Sigma)^{1/2} \mathbf{P} (\operatorname{diag} \Sigma)^{1/2}$$

Because the correlation matrix does not depend on the scale of the random variables, it is used to express relationships among random variables measured on different scales. Furthermore, since the $|\Sigma| = |(\operatorname{diag} \Sigma)^{1/2} \mathbf{P} (\operatorname{diag} \Sigma)^{1/2}|$ we have that $0 \leq |\mathbf{P}|^2 \leq 1$. Takeuchi, et al. (1982, p. 246) call the $|\mathbf{P}|$ the generalized alienation coefficient. If the elements of $\mathbf{Y}$ are independent its value is one and if elements are dependent it value is zero. Thus, the determinant of the correlation matrix may be interpreted as an overall measure of association or nonassociation.

Partitioning the Covariance Matrix

In general, we can partition the p characteristics contained in the $p \times 1$ random vector $\mathbf{X}$ into, for instance, two groups of size q and $p - q$, respectively. For example, we can write

$$\mathbf{X} = \begin{bmatrix} X_1 \\ \vdots \\ X_q \\ \hdashline X_{q+1} \\ \vdots \\ X_p \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} X_1 \\ \vdots \\ X_q \end{matrix}}\right\} q \\ \left.\vphantom{\begin{matrix} X_{q+1} \\ \vdots \\ X_p \end{matrix}}\right\} p - q \end{matrix} = \begin{bmatrix} \mathbf{X}^{(1)} \\ \hdashline \mathbf{X}^{(2)} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\mu} = E(\mathbf{X}) = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_q \\ \hdashline \mu_{q+1} \\ \vdots \\ \mu_p \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu}^{(1)} \\ \hdashline \boldsymbol{\mu}^{(2)} \end{bmatrix} \tag{2-38}$$

From the definitions of the transpose and matrix multiplication,

$$(\mathbf{X}^{(1)} - \boldsymbol{\mu}^{(1)})(\mathbf{X}^{(2)} - \boldsymbol{\mu}^{(2)})'$$

$$= \begin{bmatrix} X_1 - \mu_1 \\ X_2 - \mu_2 \\ \vdots \\ X_q - \mu_q \end{bmatrix} [X_{q+1} - \mu_{q+1}, X_{q+2} - \mu_{q+2}, \ldots, X_p - \mu_p]$$

Upon taking the expectation of the matrix $(\mathbf{X}^{(1)} - \boldsymbol{\mu}^{(1)})(\mathbf{X}^{(2)} - \boldsymbol{\mu}^{(2)})'$, we get

$$E(\mathbf{X}^{(1)} - \boldsymbol{\mu}^{(1)})(\mathbf{X}^{(2)} - \boldsymbol{\mu}^{(2)})' = \begin{bmatrix} \sigma_{1,q+1} & \sigma_{1,q+2} & \cdots & \sigma_{1p} \\ \sigma_{2,q+1} & \sigma_{2,q+2} & \cdots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{q,q+1} & \sigma_{q,q+2} & \cdots & \sigma_{qp} \end{bmatrix} = \boldsymbol{\Sigma}_{12} \quad (2\text{-}39)$$

which gives all the covariances, σ_{ij}, $i = 1, 2, \ldots, q$, $j = q+1, q+2, \ldots, p$, between a component of $\mathbf{X}^{(1)}$ and a component of $\mathbf{X}^{(2)}$. Note that the matrix $\boldsymbol{\Sigma}_{12}$ is not necessarily symmetric or even square.

Making use of the partitioning in Equation (2–38), we can easily demonstrate that

$$(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})'$$

$$= \begin{bmatrix} (\underset{(q\times 1)}{\mathbf{X}^{(1)}} - \boldsymbol{\mu}^{(1)})(\underset{(1\times q)}{\mathbf{X}^{(1)}} - \boldsymbol{\mu}^{(1)})' & (\underset{(q\times 1)}{\mathbf{X}^{(1)}} - \boldsymbol{\mu}^{(1)})(\underset{(1\times(p-q))}{\mathbf{X}^{(2)}} - \boldsymbol{\mu}^{(2)})' \\ (\underset{((p-q)\times 1)}{\mathbf{X}^{(2)}} - \boldsymbol{\mu}^{(2)})(\underset{(1\times q)}{\mathbf{X}^{(1)}} - \boldsymbol{\mu}^{(1)})' & (\underset{((p-q)\times 1)}{\mathbf{X}^{(2)}} - \boldsymbol{\mu}^{(2)})(\underset{(1\times(p-q))}{\mathbf{X}^{(2)}} - \boldsymbol{\mu}^{(2)})' \end{bmatrix}$$

and consequently,

$$\underset{(p\times p)}{\boldsymbol{\Sigma}} = E(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})' = \begin{array}{c} \\ q \\ p-q \end{array} \begin{array}{c} \begin{array}{cc} q & p-q \end{array} \\ \left[\begin{array}{c|c} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \hline \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{array}\right] \\ (p\times p) \end{array}$$

$$= \left[\begin{array}{ccc|ccc} \sigma_{11} & \cdots & \sigma_{1q} & \sigma_{1,q+1} & \cdots & \sigma_{1p} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{q1} & \cdots & \sigma_{qq} & \sigma_{q,q+1} & \cdots & \sigma_{qp} \\ \hline \sigma_{q+1,1} & \cdots & \sigma_{q+1,q} & \sigma_{q+1,q+1} & \cdots & \sigma_{q+1,p} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \cdots & \sigma_{pq} & \sigma_{p,q+1} & \cdots & \sigma_{pp} \end{array}\right] \quad (2\text{-}40)$$

Note that $\boldsymbol{\Sigma}_{12} = \boldsymbol{\Sigma}_{21}'$. The covariance matrix of $\mathbf{X}^{(1)}$ is $\boldsymbol{\Sigma}_{11}$, that of $\mathbf{X}^{(2)}$ is $\boldsymbol{\Sigma}_{22}$, and that of elements from $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ is $\boldsymbol{\Sigma}_{12}$ (or $\boldsymbol{\Sigma}_{21}$).

It is sometimes convenient to use the $\text{Cov}\,(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ notation where

$$\text{Cov}\,(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) = \boldsymbol{\Sigma}_{12}$$

is a matrix containing all of the covariances between a component of $\mathbf{X}^{(1)}$ and a component of $\mathbf{X}^{(2)}$.

Partitioning a random p-vector into two subvectors: $\mathbf{Y} = [\mathbf{Y}_1', \mathbf{Y}_2']'$, the covariance matrix of the partitioned vector is

$$\operatorname{cov}(\mathbf{Y}) = \begin{bmatrix} \operatorname{cov}(\mathbf{Y}_1, \mathbf{Y}_1) & \operatorname{cov}(\mathbf{Y}_1, \mathbf{Y}_2) \\ \operatorname{cov}(\mathbf{Y}_2, \mathbf{Y}_1) & \operatorname{cov}(\mathbf{Y}_2, \mathbf{Y}_2) \end{bmatrix} = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

where $\Sigma_{ij} = \operatorname{cov}(\mathbf{Y}_i, \mathbf{Y}_j)$. To evaluate whether $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are uncorrelated, the following theorem is used.

Theorem 3.2.3 *The random vectors $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are uncorrelated if and only if $\Sigma_{12} = \mathbf{0}$.*

The individual components of $\mathbf{Y}_i$ are uncorrelated if and only if Σ_{ii} is a diagonal matrix. If $\mathbf{Y}_i$ has cumulative distribution function $(c.d.f)$, $F_{\mathbf{Y}_i}(\mathbf{y}_i)$, with mean $\boldsymbol{\mu}_i$ and covariance matrix Σ_{ii}, we write $\mathbf{Y}_i \sim (\boldsymbol{\mu}_i, \Sigma_{ii})$.

Definition 3.2.1 *Two (absolutely) continuous random vectors $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are (statistically) independent if the probability density function of $\mathbf{Y} = [\mathbf{Y}_1', \mathbf{Y}_2']'$ is obtained from the product of the marginal densities of $\mathbf{Y}_1$ and $\mathbf{Y}_2$:*

$$f_{\mathbf{Y}}(\mathbf{y}) = f_{\mathbf{Y}_1}(\mathbf{y}_1)\, f_{\mathbf{Y}_2}(\mathbf{y}_2)$$

Lengths, Distances,

Knowledge of vector lengths, distances and angles between vectors helps one to understand relationships among multivariate vector observations. However, prior to discussing these concepts, the inner (scalar or dot) product of two vectors needs to be defined.

Definition 2.3.2 *The inner product of two vectors* $\mathbf{x}$ *and* $\mathbf{y}$, *each with n elements, is the scalar quantity*

$$\mathbf{x}'\mathbf{y} = \sum_{i=1}^{n} x_i y_i$$

In textbooks on linear algebra, the inner product may be represented as $(\mathbf{x}, \mathbf{y})$ or $\mathbf{x} \cdot \mathbf{y}$. Given Definition 2.3.2, inner products have several properties

If $\mathbf{x} = \mathbf{y}$ in Definition 2.3.2, then $\mathbf{x}'\mathbf{x} = \sum_{i=1}^{n} x_i^2$. The quantity $(\mathbf{x}'\mathbf{x})^{1/2}$ is called the Euclidean vector norm or length of $\mathbf{x}$ and is represented as $\|\mathbf{x}\|$. Thus, the norm of $\mathbf{x}$ is the positive square root of the inner product of a vector with itself. The norm squared of $\mathbf{x}$ is represented as $||\mathbf{x}||^2$. The Euclidean distance or length between two vectors $\mathbf{x}$ and $\mathbf{y}$ in V_n is $\|\mathbf{x} - \mathbf{y}\| = [(\mathbf{x} - \mathbf{y})'(\mathbf{x} - \mathbf{y})]^{1/2}$.

$$d_{ij} = \sqrt{\sum_{k=1}^{q} (x_{ik} - x_{jk})^2},$$

where x_{ik} and $x_{jk}, k = 1, \ldots, q$ are the variable values for units i and j, respectively. Euclidean distance can be calculated using the `dist()` function in R.

When the variables in a multivariate data set are on different scales, it makes more sense to calculate the distances *after* some form of standardisation. Here we shall illustrate this on the body measurement data and divide each variable by its standard deviation using the function `scale()` before applying the `dist()` function—the necessary R code and output are

```
R> dist(scale(measure[, c("chest", "waist", "hips")],
+         center = FALSE))

      1    2    3    4    5    6    7    8    9   10   11
2  0.17
3  0.15 0.08
4  0.22 0.07 0.14
5  0.11 0.15 0.09 0.22
6  0.29 0.16 0.16 0.19 0.21
7  0.32 0.16 0.20 0.13 0.28 0.14
8  0.23 0.11 0.11 0.12 0.19 0.16 0.13
9  0.21 0.10 0.06 0.16 0.12 0.11 0.17 0.09
10 0.27 0.12 0.13 0.14 0.20 0.06 0.09 0.11 0.09
11 0.23 0.28 0.22 0.33 0.19 0.34 0.38 0.25 0.24 0.32
12 0.22 0.24 0.18 0.28 0.18 0.30 0.32 0.20 0.20 0.28 0.06

...
```

(Note that only the distances for the first 12 observations are shown in the output.)

Trace and the Euclidean Matrix Norm

An important operation for square matrices is the trace operator. For a square matrix $\mathbf{A}_{n\times n} = [a_{ij}]$, the trace of $\mathbf{A}$, represented as $\text{tr}(\mathbf{A})$, is the sum of the diagonal elements of $\mathbf{A}$. Hence,

$$\text{tr}\,(\mathbf{A}) = \sum_{i=1}^{n} a_{ii} \tag{2.4.6}$$

Theorem 2.4.3 *For square matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *and scalars* α *and* β*, the following properties hold for the trace of a matrix.*

1. $\text{tr}(\alpha\mathbf{A}+\beta\mathbf{B}) = \alpha\,\text{tr}(\mathbf{A}) + \beta\,\text{tr}(\mathbf{B})$
2. $\text{tr}(\mathbf{AB}) = \text{tr}\,(\mathbf{BA})$
3. $\text{tr}(\mathbf{A}') = \text{tr}(\mathbf{A})$
4. $\text{tr}(\mathbf{A}'\mathbf{A}) = \text{tr}(\mathbf{AA}') = \sum_{i,j} a_{ij}^2$ *and equals* 0*, if and only if* $\mathbf{A} = \mathbf{0}$.

Property (4) is an important property for matrices since it generalizes the Euclidean vector norm squared to matrices. The Euclidean norm squared of $\mathbf{A}$ is defined as

$$\|\mathbf{A}\|^2 = \sum_i \sum_j a_{ij}^2 = \text{tr}(\mathbf{A}'\mathbf{A}) = \text{tr}(\mathbf{AA}')$$

The Euclidean matrix norm is defined as

$$\|\mathbf{A}\| = \left\{\operatorname{tr}\left(\mathbf{A}'\mathbf{A}\right)\right\}^{1/2} = \left\{\operatorname{tr}\left(\mathbf{A}\mathbf{A}'\right)\right\}^{1/2} \tag{2.4.7}$$

and is zero only if $\mathbf{A} = \mathbf{0}$.

Solving the characteristic equation $\left|\mathbf{A}'\mathbf{A} - \lambda\mathbf{I}\right| = 0$, the Euclidean norm becomes $\|\mathbf{A}\|_2 = \left\{\sum_i \lambda_i\right\}^{1/2}$ where λ_i is a root of $\mathbf{A}'\mathbf{A}$. The spectral norm is the square root of the maximum root of $\mathbf{A}'\mathbf{A}$. Thus, $\|\mathbf{A}\|_s = \max\sqrt{\lambda_i}$. Extending the Minkowski vector norm to a matrix, a general matrix (L_p norm) norm is $\|\mathbf{A}\|_p = \left\{\sum_i \lambda_i^{p/2}\right\}^{1/p}$ where λ_i are the roots of $\mathbf{A}'\mathbf{A}$, also called the von Neumann norm. For $p = 2$, it reduces to the Euclidean norm.

While Euclidean distances and norms are useful concepts in statistics since they help to visualize statistical sums of squares, non-Euclidean distance and non-Euclidean norms are often useful in multivariate analysis. We have seen that the Euclidean norm generalizes to a more general function that maps a vector to a scalar. In a similar manner, we may generalize the concept of distance. A non-Euclidean distance important in multivariate analysis is the statistical or Mahalanobis distance.

To motivate the definition, consider a normal random variable X with mean zero and variance one, $X \sim N(0, 1)$. An observation $\mathbf{x}_o$ that is two standard deviations from the mean lies a distance of two units from the origin since the $\|\mathbf{x}_o\| = (0^2 + 2^2)^{1/2} = 2$ and the probability that $0 \leq x \leq 2$ is 0.4772. Alternatively, suppose $Y \sim N(0, 4)$ where the distance from the origin for $\mathbf{y}_o = \mathbf{x}_o$ is still 2. However, the probability that $0 \leq y \leq 2$ becomes 0.3413 so that y is closer to the origin than x. To compare the distances, we must take into account the variance of the random variables. Thus, the squared distance between x_i and x_j is defined as

$$D_{ij}^2 = (x_i - x_j)^2/\sigma^2 = (x_i - x_j)(\sigma^2)^{-1}(x_i - x_j) \tag{2.3.5}$$

where σ^2 is the population variance. For our example, the point $\mathbf{x}_o$ has a squared statistical distance $D_{ij}^2 = 4$ while the point $\mathbf{y}_o = 2$ has a value of $D_{ij}^2 = 1$ which maintains the inequality in probabilities in that Y is "closer" to zero statistically than X. D_{ij} is the distance between x_i and x_j, in the metric of σ^2 called the Mahalanobis distance between x_i and x_j. When $\sigma^2 = 1$, Mahalanobis' distance reduces to the Euclidean distance.

we defined the Mahalanobis distance for a random variable. It was an "adjusted" Euclidean distance which represented statistical closeness in the metric of $1/\sigma^2$ or $\left(\sigma^2\right)^{-1}$. With the first two moments of a random vector defined, suppose we want to calculate the distance between $\mathbf{Y}$ and $\boldsymbol{\mu}$. Generalizing (2.3.5), the Mahalanobis distance between $\mathbf{Y}$ and $\boldsymbol{\mu}$ in the metric of Σ is

$$D_{\Sigma}(\mathbf{Y}, \boldsymbol{\mu}) = [(\mathbf{Y} - \boldsymbol{\mu})'\Sigma^{-1}(\mathbf{Y} - \boldsymbol{\mu})]^{1/2} \tag{3.2.5}$$

If $\mathbf{Y} \sim (\boldsymbol{\mu}_1, \Sigma)$ and $\mathbf{X} \sim (\boldsymbol{\mu}_2, \Sigma)$, then the Mahalanobis distance between $\mathbf{Y}$ and $\mathbf{X}$, in the metric of Σ, is the square root of

$$D_{\Sigma}^2(\mathbf{X}, \mathbf{Y}) = (\mathbf{X} - \mathbf{Y})'\Sigma^{-1}(\mathbf{X} - \mathbf{Y})$$

which is invariant under linear transformations $\mathbf{z}_{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{a}$ and $\mathbf{z}_{\mathbf{Y}} = \mathbf{A}\mathbf{Y} + \mathbf{b}$. The covariance matrix Σ of $\mathbf{X}$ and $\mathbf{Y}$ becomes $\Omega = \mathbf{A}\Sigma\mathbf{A}'$ under the transformations so that $D_{\Sigma}^2(\mathbf{X}, \mathbf{Y}) = \mathbf{z}_{\mathbf{X}}'\Omega\mathbf{z}_{\mathbf{Y}} = D_{\Omega}^2(\mathbf{z}_{\mathbf{X}}, \mathbf{z}_{\mathbf{Y}})$.

The Mahalanobis distances, D, arise in a natural manner when investigating the separation between two or more multivariate populations, the topic of discriminant analysis

Having defined the mean and covariance matrix for a random vector $\mathbf{Y}_{p\times 1}$ and the first two moments of a random vector, we extend the classical measures of skewness and kurtosis, $E[(Y-\mu)^3]/\sigma^3 = \mu_3'/\sigma^3$ and $E[(Y-\mu)^4]/\sigma^4 = \mu_4'/\sigma^4$ of a univariate variable Y, respectively, to the multivariate case. Following Mardia (1970), multivariate skewness and kurtosis measures for a random p-variate vector $\mathbf{Y}_p \sim (\boldsymbol{\mu}, \Sigma)$ are, respectively, defined as

$$\beta_{1,p} = E\left\{(\mathbf{Y}-\boldsymbol{\mu})'\,\Sigma^{-1}\,(\mathbf{X}-\boldsymbol{\mu})\right\}^3 \tag{3.2.6}$$

$$\beta_{2,p} = E\left\{(\mathbf{Y}-\boldsymbol{\mu})'\,\Sigma^{-1}\,(\mathbf{Y}-\boldsymbol{\mu})\right\}^2 \tag{3.2.7}$$

where $\mathbf{Y}_p$ and $\mathbf{X}_p$ are identically and independent identically distributed (*i.i.d.*). Because $\beta_{1,p}$ and $\beta_{2,p}$ have the same form as Mahalanobis' distance, they are also seen to be invariant under linear transformations.

The multivariate measures of skewness and kurtosis are natural generalizations of the univariate measures

$$\sqrt{\beta_1} = \sqrt{\beta_{1,1}} = \mu_3'/\sigma^3 \tag{3.2.8}$$

and

$$\beta_2 = \beta_{2,1} = \mu_4'/\sigma^4 \tag{3.2.9}$$

For a univariate normal random variable, $\gamma_1 = \sqrt{\beta_1} = 0$ and $\gamma_2 = \beta_2 - 3 = 0$.

The Multivariate Normal (MVN) Distribution

This is the joint density function of an independent multivariate normal distribution, written as $\mathbf{Y} \sim N_p(\boldsymbol{\mu}, \sigma^2\mathbf{I})$, where the mean vector and covariance matrix are

$$E(\mathbf{Y}) = \boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix}, \quad \text{and} \quad \text{cov}(\mathbf{Y}) = \begin{bmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & \sigma^2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \sigma^2 \end{bmatrix} = \sigma^2\mathbf{I}_p,$$

respectively.

More generally, replacing $\sigma^2\mathbf{I}_p$ with a positive definite covariance matrix Σ, a generalization of the independent multivariate normal density to the multivariate normal (MVN) distribution is established

$$f(\mathbf{y}) = (2\pi)^{-p/2} |\Sigma|^{-1/2} \exp\left\{-(\mathbf{y}-\boldsymbol{\mu})' \Sigma^{-1} (\mathbf{y}-\boldsymbol{\mu})/2\right\} \qquad -\infty < y_i < \infty \tag{3.3.2}$$

Theorem 3.3.1 *A random p-vector* $\mathbf{Y}$ *is said to have a p-variate normal or multivariate normal (MVN) distribution with mean* $\boldsymbol{\mu}$ *and p.d. covariance matrix* Σ *written* $\mathbf{Y} \sim N_p(\boldsymbol{\mu}, \Sigma)$, *if it has the joint density function given in (3.3.2). If* Σ *is not p.d., the density function of* $\mathbf{Y}$ *does not exist and* $\mathbf{Y}$ *is said to have a singular multivariate normal distribution.*

Observe that the joint density of the MVN distribution is constant whenever the quadratic form in the exponent is constant. The constant density ellipsoid $(\mathbf{Y} - \boldsymbol{\mu})' \Sigma^{-1} (\mathbf{Y} - \boldsymbol{\mu}) = c$ has center at $\boldsymbol{\mu}$ while Σ determines its shape and orientation. In the bivariate case,

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}, \boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$$

For the MVN to be nonsingular, we need $\sigma_1^2 > 0, \sigma_2^2 > 0$ and the $|\Sigma| = \sigma_1^2\sigma_2^2\left(1 - \rho^2\right) > 0$ so that $-1 < \rho < 1$. Then

$$\Sigma^{-1} = \frac{1}{1 - \rho^2} \begin{bmatrix} \frac{1}{\sigma_1^2} & \frac{-\rho}{\sigma_1\sigma_2} \\ \frac{-\rho}{\sigma_1\sigma_2} & \frac{1}{\sigma_2^2} \end{bmatrix}$$

and the joint probability density of $\mathbf{Y}$ yields the bivariate normal density

$$f(\mathbf{y}) = \frac{\exp\left\{\frac{-1}{2(1-\rho^2)}\left[\left(\frac{y_1-\mu_1}{\sigma_1}\right)^2 - 2\rho\left(\frac{y_1-\mu_1}{\sigma_1}\right)\left(\frac{y_2-\mu_2}{\sigma_2}\right) + \left(\frac{y_2-\mu_2}{\sigma_2}\right)^2\right]\right\}}{2\pi\sigma_1\sigma_2\left(1-\rho^2\right)^{1/2}}$$

Letting $Z_i = \left(Y_i - \mu_i\right)/\sigma_i (i = 1, 2)$, the joint bivariate normal becomes the standard bivariate normal

$$f(\mathbf{z}) = \frac{\exp\frac{-1}{2(1-\rho^2)}\left(z_1^2 - 2\rho z_1 z_2 + z_2^2\right)}{2\pi\left(1-\rho^2\right)^{1/2}} \qquad -\infty < z_i < \infty$$

The exponent in the standard bivariate normal distribution is a quadratic form

$$Q = [z_1, z_2]' \, \Sigma^{-1} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \frac{z_1^2 - 2\rho z_1 z_2 + z_2^2}{1-\rho^2} > 0$$

where

$$\Sigma^{-1} = \frac{1}{1-\rho^2}\begin{bmatrix} 1 & -\rho \\ -\rho & 1 \end{bmatrix} \quad \text{and} \quad \Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

which generates concentric ellipses about the origin. Setting $\rho = 1/2$, the ellipses have the form $Q = z_1^2 - z_1 z_2 + z_2^2$ for $Q > 0$. Graphing this function in the plane with axes z_1 and z_2 for $Q = 1$ yields the constant density ellipse with semi-major axis a and semi-minor axis b, Figure 3.3.1.

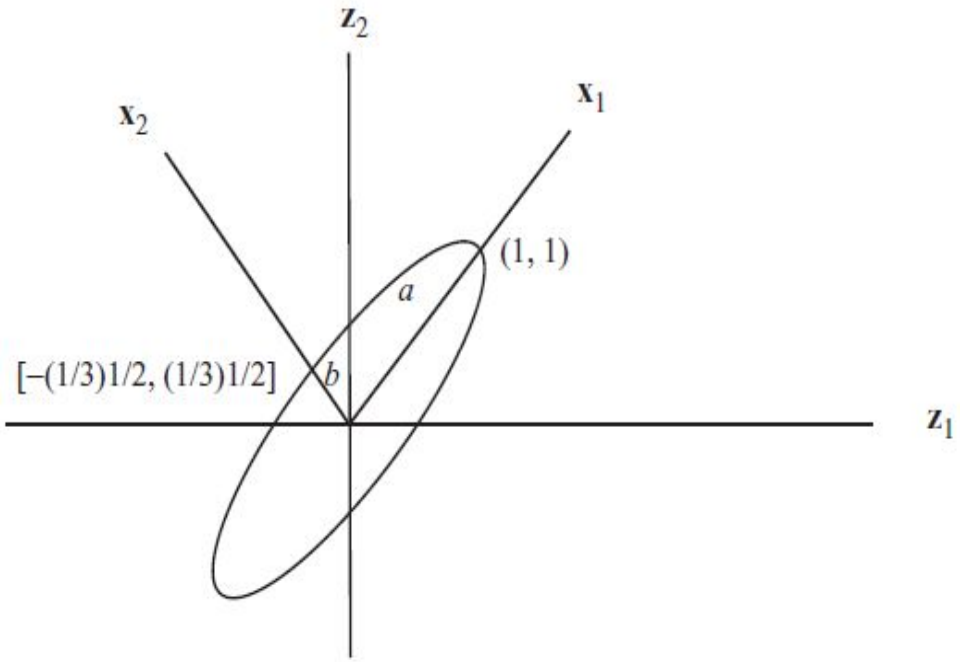


FIGURE 3.3.1. $\mathbf{z}'\Sigma^{-1}\mathbf{z} = z_1^2 - z_1 z_2 + z_2^2 = 1$

Performing an orthogonal rotation of $\mathbf{x} = \mathbf{P}'\mathbf{z}$, the quadratic form for the exponent of the standard MVN becomes

$$\mathbf{z}'\Sigma^{-1}\mathbf{z} = \lambda_1^* x_1^2 + \lambda_2^* x_2^2 = \frac{1}{\lambda_2} x_1^2 + \frac{1}{\lambda_1} x_2^2$$

where $\lambda_1 = 1 + \rho = 3/2$ and $\lambda_2 = 1 - \rho = 1/2$ are the roots of $|\Sigma - \lambda\mathbf{I}| = 0$ and $\lambda_2^* = 1/\lambda_1$ and $\lambda_1^* = 1/\lambda_2$ are the roots of $|\ \Sigma^{-1} - \lambda^*\mathbf{I} = 0$. From analytic geometry, the equation of an ellipse, for $Q = 1$, is given by

$$\left(\frac{1}{b}\right)^2 x_1^2 + \left(\frac{1}{a}\right)^2 x_2^2 = 1$$

Hence $a^2 = \lambda_1$ and $b^2 = \lambda_2$ so that each half-axis is proportional to the inverse of the squared lengths of the eigenvalues of Σ. As Q varies, concentric ellipsoids are generated so that $a = \sqrt{Q\lambda_1}$ and $b = \sqrt{Q\lambda_2}$.

Theorem 3.3.2 *Properties of normally distributed random variables.*

1. *Linear combinations of the elements of* $\mathbf{Y} \sim \mathbf{N}[\boldsymbol{\mu}, \Sigma]$ *are normally distributed. For a constant vector* $\mathbf{a} \neq \mathbf{0}$ *and* $\mathbf{X} = \mathbf{a}'\mathbf{Y}$, *then* $\mathbf{X} \sim N_1\left(\mathbf{a}'\boldsymbol{\mu}, \mathbf{a}'\Sigma\mathbf{a}\right)$.

2. *The normal distribution of* $\mathbf{Y}_p \sim N_p[\boldsymbol{\mu}, \Sigma]$ *is invariant to linear transformations. For a constant matrix* $\mathbf{A}_{q\times p}$ *and vector* $\mathbf{b}_{q\times 1}$, $\mathbf{X} = \mathbf{A}\mathbf{Y}_p + \mathbf{b} \sim N_q(\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{A}\Sigma\mathbf{A}')$.

3. *Partitioning* $\mathbf{Y} = [\mathbf{Y}_1', \mathbf{Y}_2']'$, $\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$, *the subvectors of* $\mathbf{Y}$ *are normally distributed.* $\mathbf{Y}_1 \sim N_{p_1}\left[\boldsymbol{\mu}_1, \Sigma_{11}\right]$ *and* $\mathbf{Y}_2 \sim N_{p_2}\left[\boldsymbol{\mu}_2, \Sigma_{22}\right]$ *where* $p_1 + p_2 = p$. *More generally, all marginal distributions for any subset of random variables are normally distributed. However, the converse is not true, marginal normality does not imply multivariate normality.*

4. *The random subvectors* $\mathbf{Y}_1$ *and* $\mathbf{Y}_2$ *of* $\mathbf{Y} = [\mathbf{Y}_1', \mathbf{Y}_2']'$ *are independent if and only if* $\Sigma = \text{diag}[\Sigma_{11}, \Sigma_{22}]$. *Thus, uncorrelated normal subvectors are independent under multivariate normality.*

5. *The conditional distribution of* $\mathbf{Y}_1 \mid \mathbf{Y}_2$ *is normally distributed,*

$$\mathbf{Y}_1 \mid \mathbf{Y}_2 \sim N_{p_1}\left[\boldsymbol{\mu}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{y}_2 - \boldsymbol{\mu}_1), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right]$$

Writing the mean of the conditional normal distribution as

$$\begin{aligned}\boldsymbol{\mu} &= (\boldsymbol{\mu}_1 - \Sigma_{12}\Sigma_{22}^{-1}\boldsymbol{\mu}_2) + \Sigma_{12}\Sigma_{22}^{-1}\mathbf{y}_2 \\ &= \boldsymbol{\mu}_0 + \mathbf{B}_1'\mathbf{y}_2\end{aligned}$$

$\boldsymbol{\mu}$ *is called the regression function of* $\mathbf{Y}_1$ *on* $\mathbf{Y}_2 = \mathbf{y}_2$ *with regression coefficients* $\mathbf{B}_1'$. *The matrix* $\Sigma_{11.2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$ *is called the partial covariance matrix with elements* $\sigma_{ij.\,p_1+1,\ldots,\,p_1+p_2}$. *A similar result holds for* $\mathbf{Y}_2 \mid \mathbf{Y}_1$.

6. *Letting* $\mathbf{Y}_1 = Y$, *a single random variable and letting the random vector* $\mathbf{Y}_2 = \mathbf{X}$, *a random vector of independent variables, the population coefficient of determination or population squared multiple correlation coefficient is defined as the maximum correlation between* $\mathbf{Y}$ *and linear functions* $\boldsymbol{\beta}'\mathbf{X}$. *The population coefficient of determination or the squared population multiple correlation coefficient is*

$$\rho^2_{Y\mathbf{X}} = \boldsymbol{\sigma}'_{Y\mathbf{X}}\Sigma^{-1}_{\mathbf{XX}}\boldsymbol{\sigma}_{\mathbf{X}Y}/\sigma_{YY}$$

If the random vector $\mathbf{Z} = (Y, \mathbf{X}')'$ *follows a multivariate normal distribution, the population coefficient of determination is the square of the zero-order correlation between the random variable* Y *and the population predicted value of* Y *which we see from (5) has the form* $\widehat{Y} = \mu_Y + \boldsymbol{\sigma}'_{Y\mathbf{X}}\Sigma^{-1}_{\mathbf{XX}}(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{X}})$.

7. *For* $\mathbf{X} = \Sigma^{-1/2}(\mathbf{Y} - \boldsymbol{\mu})$ *where* $\Sigma^{-1/2}$*is the symmetric positive definite square root of* Σ^{-1} *then* $\mathbf{X} \sim N_p(\mathbf{0}, \mathbf{I})$ or $X_i \sim IN(0, 1)$.

8. *If* $\mathbf{Y}_1$ and $\mathbf{Y}_2$ *are independent multivariate normal random vectors, then the sum* $\mathbf{Y}_1 + \mathbf{Y}_2 \sim N(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2, \Sigma_{11} + \Sigma_{22})$. *More generally, if* $\mathbf{Y}_i \sim IN_p(\boldsymbol{\mu}_i, \Sigma_i)$ and $a_1, a_2, \ldots, a_n$ *are fixed constants, then the sum of n p-variate vectors*

$$\sum_{i=1}^{n} a_i \mathbf{Y}_i \sim N_p\left[\sum_{i=1}^{n} a_i \mu_i, \sum_{i=1}^{n} a_i^2 \Sigma_i\right]$$

Theorem 3.3.3 *If* $\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_n$ *are independent MVN random vectors with common mean* $\boldsymbol{\mu}$ *and covariance matrix* Σ, then $\overline{\mathbf{Y}} = \sum_{i=1}^{n} \mathbf{Y}_i / n$ *is MVN with mean* $\boldsymbol{\mu}$ *and covariance matrix* Σ/n, $\overline{\mathbf{Y}} \sim N_p(\boldsymbol{\mu}, \Sigma/n)$.

Estimating $\boldsymbol{\mu}$ and Σ

From Theorem 3.3.3, observe that for a random sample from a normal population that $\overline{\mathbf{Y}}$ is an unbiased and consistent estimator of $\boldsymbol{\mu}$, written as $\widehat{\boldsymbol{\mu}} = \overline{\mathbf{Y}}$. Having estimated $\boldsymbol{\mu}$, the $p \times p$ sample covariance matrix is

$$
\begin{aligned}
\mathbf{S} &= \sum_{i=1}^{n} (\mathbf{y}_i - \overline{\mathbf{y}})(\mathbf{y}_i - \overline{\mathbf{y}})' / (n-1) \\
&= \sum_{i=1}^{n} [(\mathbf{y}_i - \boldsymbol{\mu}) - (\overline{\mathbf{y}} - \boldsymbol{\mu})][(\mathbf{y}_i - \boldsymbol{\mu}) - (\overline{\mathbf{y}} - \boldsymbol{\mu})]' / (n-1) \\
&= \left[\sum_{i=1}^{n} (\mathbf{y}_i - \boldsymbol{\mu})(\mathbf{y}_i - \boldsymbol{\mu})' + n\left(\overline{\mathbf{y}}_i - \boldsymbol{\mu}\right)\left(\overline{\mathbf{y}}_i - \boldsymbol{\mu}\right)' \right] / (n-1) \qquad (3.3.3)
\end{aligned}
$$

where $E(\mathbf{S}) = \Sigma$ so that $\mathbf{S}$ is an unbiased estimator of Σ. Representing the sample as a matrix $\mathbf{Y}_{n \times p}$ so that

$$
\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1' \\ \mathbf{y}_2' \\ \vdots \\ \mathbf{y}_n' \end{bmatrix}
$$

$\mathbf{S}$ may be written as

$$
\begin{aligned}
(n-1)\,\mathbf{S} &= \mathbf{Y}'\left[\mathbf{I}_n - \mathbf{1}_n\left(\mathbf{1}_n'\mathbf{1}_n\right)^{-1}\mathbf{1}_n'\right]\mathbf{Y} \\
&= \mathbf{Y}'\mathbf{Y} - n\overline{\mathbf{y}}\overline{\mathbf{y}}' \qquad (3.3.4)
\end{aligned}
$$

where $\mathbf{I}_n$ is the identity matrix and $\mathbf{1}_n$ is a vector of n 1s. While the matrix $\mathbf{S}$ is an unbiased estimator, a biased estimator, called the maximum likelihood estimator under normality is $\widehat{\Sigma} = \frac{(n-1)}{n}\mathbf{S} = \sum_{i=1}^{n}(\mathbf{y}_i - \overline{\mathbf{y}})(\mathbf{y}_i - \overline{\mathbf{y}})'/n = \mathbf{E}/n$. The matrix $\mathbf{E}$ is called the sum of squares and cross-products matrix, SSCP and the $|\mathbf{S}|$ is the sample estimate of the generalized variance.

In Theorem 3.3.3, we assumed that the observations $\mathbf{Y}_i$ represent a sample from a normal distribution. More generally, suppose $\mathbf{Y}_i \sim (\boldsymbol{\mu}, \Sigma)$ is an independent sample from any distribution with mean $\boldsymbol{\mu}$ and covariance matrix Σ. Theorem 3.3.4 is a multivariate version of the Central Limit Theorem (CLT).

Theorem 3.3.4 *Let $\{\mathbf{Y}_i\}_{i=1}^{\infty}$ be a sequence of random p-vectors with finite mean $\boldsymbol{\mu}$ and covariance matrix Σ. Then*

$$
n^{1/2}(\overline{\mathbf{Y}} - \boldsymbol{\mu}) = n^{-1/2}\sum_{i=1}^{n}(\mathbf{Y}_i - \boldsymbol{\mu}) \xrightarrow{d} N_p(\mathbf{0}, \Sigma)
$$

Theorem 3.3.4 is used to show that $\mathbf{S}$ is a consistent estimator of Σ. To obtain the distribution of a random matrix $\mathbf{Y}_{n\times p}$, the vec $(\cdot)$ operator is used. Assuming a random sample of n p-vectors $\mathbf{Y}_i \sim (\boldsymbol{\mu}, \Sigma)$, consider the random matrix $\mathbf{X}_i = (\mathbf{Y}_i - \boldsymbol{\mu})(\mathbf{Y}_i - \boldsymbol{\mu})'$. By Theorem 3.3.4,

$$n^{-1/2} \sum_{i=1}^{n} [\mathrm{vec}(\mathbf{X}_i) - \mathrm{vec}\,(\Sigma)] \xrightarrow{d} N_{p^2}\,(\mathbf{0}, \Omega)$$

where

$$\Omega = \mathrm{cov}[\mathrm{vec}(\mathbf{X}_i)]$$

and

$$n^{-1/2}\,(\overline{\mathbf{y}} - \boldsymbol{\mu}) \xrightarrow{d} N_p\,(\mathbf{0}, \Sigma)$$

so that

$$n^{-1/2}[\mathrm{vec}(\mathbf{E}) - n\ \mathrm{vec}(\Sigma)] \xrightarrow{d} N_{p^2}(\mathbf{0}, \Omega).$$

Because $\mathbf{S} = (n-1)^{-1}\mathbf{E}$ and the replacement of n by $n-1$ does not effect the limiting distribution, we have the following theorem.

Theorem 3.3.5 *Let $\{\mathbf{Y}_i\}_{i=1}^{\infty}$ be a sequence of independent and identically distributed $p \times 1$ vectors with finite fourth moments and mean $\boldsymbol{\mu}$ and covariance matrix Σ. Then*

$$(n-1)^{-1/2}\,\mathrm{vec}(\mathbf{S} - \Sigma) \xrightarrow{d} N_{p^2}(\mathbf{0}, \Omega)$$

a. *Chi-Square Distribution*

Recall that if $Y_1, Y_2, \ldots, Y_n$ are independent normal random variables with mean $\mu_i = 0$ and variance $\sigma^2 = 1$, $Y_i \sim IN\,(0, 1)$, or, employing vector notation $\mathbf{Y} \sim N_n(\mathbf{0}, \mathbf{I})$, then

$$Q = \mathbf{Y}'\mathbf{Y} = \sum_{i=1}^{n} Y_i^2 \sim \chi^2\,(n) \qquad 0 < Q < \infty$$

$Q = \mathbf{Y}'\mathbf{Y}$ has a central χ^2 distribution with n degrees of freedom. Letting $Y_i \sim IN(\mu_i, \sigma^2)$, results in the noncentral chi-square distribution.

Definition 3.4.1 *If the random n-vector* $\mathbf{Y} \sim N_n(\boldsymbol{\mu}, \sigma^2\mathbf{I})$, *then* $\mathbf{Y}'\mathbf{Y}/\sigma^2$ *has a noncentral* χ^2 *distribution with n degrees of freedom and noncentrality parameter* $\gamma = \boldsymbol{\mu}'\boldsymbol{\mu}/\sigma^2$.

For $\boldsymbol{\mu} = \mathbf{0}$, the noncentral chi-square distribution reduces to a central chi-square distribution. For $\mathbf{Y} \sim N_n\,(\boldsymbol{\mu}, \mathbf{I})$, then $\mathbf{Y}'\mathbf{Y} \sim \chi^2(n, \gamma)$ with $\gamma = \boldsymbol{\mu}'\boldsymbol{\mu}$ so that $\gamma = \|\boldsymbol{\mu}\|^2$ is a norm squared. The further $\boldsymbol{\mu}$ is from zero, the larger the noncentrality parameter γ or the norm squared of $\boldsymbol{\mu}$. Because $\mathbf{Y}'\mathbf{Y}$ in Definition 3.4.1 is a special case of the quadratic form $\mathbf{Y}'\mathbf{A}\mathbf{Y}$, with $\mathbf{A} = \mathbf{I}$ and since $\mathbf{I}^2 = \mathbf{I}$, we have the following more general result.

Theorem 3.4.1 *Let* $\mathbf{Y} \sim N_n\left(\boldsymbol{\mu}, \sigma^2\mathbf{I}\right)$ *and* $\mathbf{A}$ *be a symmetric matrix of rank* r. *Then we have* $\mathbf{Y}'\mathbf{A}\mathbf{Y}/\sigma^2 \sim \chi^2(r, \gamma)$, *where* $\gamma = \boldsymbol{\mu}'\mathbf{A}\boldsymbol{\mu}/\sigma^2$, *if and only if* $\mathbf{A} = \mathbf{A}^2$.

Example 3.4.1 *As an example of Theorem 3.4.1, suppose* $\mathbf{Y} \sim N_n(\boldsymbol{\mu}, \sigma^2\mathbf{I}_n)$. *Then*

$$\frac{(n-1)\,s^2}{\sigma^2} = \frac{\sum_{i=1}^{n}\left(Y_i - \overline{Y}\right)^2}{\sigma^2} = \frac{\mathbf{Y}'[\mathbf{I} - \mathbf{1}\left(\mathbf{1}'\mathbf{1}\right)^{-1}\mathbf{1}]\mathbf{Y}}{\sigma^2} = \frac{\mathbf{Y}'\mathbf{A}\mathbf{Y}}{\sigma^2}$$

However, $\mathbf{A}' = \mathbf{A}$ *and* $\mathbf{A}^2 = \mathbf{A}$ *since* $\mathbf{A}$ *is a projection matrix and the* $r(\mathbf{A}) = \mathrm{tr}(\mathbf{A}) = n-1$. *Hence*

$$\frac{(n-1)\,s^2}{\sigma^2} \sim \chi^2\,(n-1,\ \gamma = 0)$$

since $\gamma = E\left(\mathbf{Y}'\right)\mathbf{A}E(\mathbf{Y})/\sigma^2 = 0$. *Thus,* $(n-1)\,s^2 \sim \sigma^2\chi^2(n-1)$.

Theorem 3.4.2 generalizes Theorem 3.4.1 to a vector of dependent variables in a natural manner by setting $\mathbf{Y} = \mathbf{F}\mathbf{X}$ and $\mathbf{F}\mathbf{F}' = \Sigma$.

Theorem 3.4.2 *If* $\mathbf{Y} \sim N_p\,(\boldsymbol{\mu}, \Sigma)$. *Then the quadratic form* $\mathbf{Y}'\mathbf{A}\mathbf{Y} \sim \chi^2(r, \gamma)$, *where* $\gamma = \left(\boldsymbol{\mu}'\mathbf{A}\boldsymbol{\mu}\right)$ *and the* $r(\mathbf{A}) = r$, *if and only if* $\mathbf{A}\Sigma\mathbf{A} = \mathbf{A}$ *or* $\mathbf{A}\Sigma$ *is idempotent.*

Example 3.4.2 *An important application of Theorem 3.4.2 follows:*

Let $\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_n$ be n independent p-vectors from any distribution with mean $\boldsymbol{\mu}$ and nonsingular covariance matrix Σ. Then by the CLT, $\sqrt{n}(\overline{\mathbf{Y}} - \boldsymbol{\mu}) \xrightarrow{d} N_p(\mathbf{0}, \Sigma)$. By Theorem 3.4.2, $T^2 = n(\overline{\mathbf{Y}} - \boldsymbol{\mu})'\Sigma^{-1}\left(\overline{\mathbf{Y}} - \boldsymbol{\mu}\right) = nD^2 \xrightarrow{d} \chi^2(p)$ for $n - p$ large since $\Sigma^{-1}\Sigma\Sigma^{-1} = \Sigma$. The distribution is exactly $\chi^2(p)$ if the sample is from a multivariate normal distribution.

Thus, comparing nD^2 with a χ^2 critical value may be used to evaluate multivariate normality. Furthermore, nD^2 for known Σ may be used to test $H_0 : \boldsymbol{\mu} = \boldsymbol{\mu}_0$ vs. $H_1 : \boldsymbol{\mu} \neq \boldsymbol{\mu}_0$. The critical value of the test with significance level α is represented as

$$\Pr[nD^2 \geq \chi^2_{1-\alpha}(p) \mid H_0] = \alpha$$

where $\chi^2_{1-\alpha}$ is the upper $1 - \alpha$ chi-square critical value. For $\boldsymbol{\mu} \neq \boldsymbol{\mu}_0$, the noncentrality parameter is

$$\gamma = n\left(\boldsymbol{\mu} - \boldsymbol{\mu}_0\right)' \Sigma^{-1}\left(\boldsymbol{\mu} - \boldsymbol{\mu}_0\right)$$

The above result is for a single quadratic form. More generally we have Cochran's Theorem.

Theorem 3.4.3 *If $\mathbf{Y} \sim N_n(\boldsymbol{\mu}, \sigma^2\mathbf{I}_n)$ and $\mathbf{Y}'\mathbf{Y}/\sigma^2 = \sum_{i=1}^n \mathbf{Y}'\mathbf{A}_i\mathbf{Y}$ where $r(\mathbf{A}_i) = r$ and $\sum_{i=1}^n \mathbf{A}_i = \mathbf{I}_n$, then the quadratic forms $\mathbf{Y}'\mathbf{A}_i\mathbf{Y}/\sigma^2 \sim \chi^2(r_i, \gamma_i)$, where $\gamma_i = \boldsymbol{\mu}'\mathbf{A}_i\boldsymbol{\mu}/\sigma^2$ are statistically independent for all i if and only if $\sum_{i=1}^n r_i = n$ and $\sum_i r(\mathbf{A}_i) = r\left(\sum_i \mathbf{A}_i\right)$.*

Example 2.6.3 *Consider a model that relates one dependent variable y to $x_1, x_2, \ldots, x_k$ linearly independent variables by the linear relationship*

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + e$$

where e is a random error. This model is the multiple linear regression model, which, using matrix notation, may be written as

$$\underset{n\times 1}{\mathbf{y}} = \underset{n\times(k+1)}{\mathbf{X}} \underset{(k+1)\times 1}{\boldsymbol{\beta}} + \underset{n\times 1}{\mathbf{e}}$$

Letting $\mathbf{X}$ represent the space spanned by the columns of $\mathbf{X}$, the projection of $\mathbf{y}$ onto $\mathbf{X}$ is

$$\widehat{\mathbf{y}} = \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{y}$$

Assuming $\mathbf{e} = \mathbf{0}$, the system of equations is solved to obtain the best estimate of $\boldsymbol{\beta}$. Then, the best estimate of $\mathbf{y}$ using the linear model is $\mathbf{X}\widehat{\boldsymbol{\beta}}$ where $\widehat{\boldsymbol{\beta}}$ is the solution to the system $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}$ for unknown $\boldsymbol{\beta}$. The least squares estimate $\widehat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{y}$ minimizes the sum of squared errors for the fitted model $\widehat{\mathbf{y}} = \mathbf{X}\widehat{\boldsymbol{\beta}}$. Furthermore,

$$\begin{aligned}
\|(\mathbf{y} - \widehat{\mathbf{y}})\|^2 &= (\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})'(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) \\
&= \mathbf{y}'(\mathbf{I}_n - \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}')\mathbf{y} \\
&= \left\|\mathbf{P}_{V^\perp}\mathbf{y}\right\|^2
\end{aligned}$$

is the squared distance of the projection of $\mathbf{y}$ onto the orthocomplement of $V_r \subseteq V_n$.

Example 3.4.3 *Let* $Y \sim N_4(\mu, \sigma^2 I)$

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} = [\mathbf{A}_1 \; \mathbf{A}_2] \quad \textit{and} \quad \mathbf{y} = \begin{bmatrix} y_{11} \\ y_{12} \\ y_{21} \\ y_{22} \end{bmatrix}$$

where

$$\mathbf{A}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \textit{and} \quad \mathbf{A}_2 \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

In Example 2.6.2, projection matrices of the form

$$\begin{aligned} \mathbf{P}_1 &= \mathbf{A}_1 \left(\mathbf{A}_1'\mathbf{A}_1\right)^- \mathbf{A}_1' \\ \mathbf{P}_2 &= \mathbf{A} \left(\mathbf{A}'\mathbf{A}\right)^- \mathbf{A}' - \mathbf{A}_1 \left(\mathbf{A}_1'\mathbf{A}_1\right)^- \mathbf{A}_1' \\ \mathbf{P}_3 &= \mathbf{I} - \mathbf{A} \left(\mathbf{A}'\mathbf{A}\right)^- \mathbf{A}' \end{aligned}$$

were constructed to project the observation vector $\mathbf{y}$ *onto orthogonal subspaces. The projection matrices were constructed such that* $\mathbf{I} = \mathbf{P}_1 + \mathbf{P}_2 + \mathbf{P}_3$ *where* $\mathbf{P}_i\mathbf{P}_j = \mathbf{0}$ *for* $i \neq j$ *and each* $\mathbf{P}_i$ *is symmetric and idempotent so that the* $r(\mathbf{I}) = \sum_i r(\mathbf{P}_i)$. *Forming an equation of quadratic forms, we have that*

$$\mathbf{y}'\mathbf{y} = \sum_{i=1}^{3} \mathbf{y}'\mathbf{P}_i\mathbf{y}$$

or

$$\|\mathbf{y}\|^2 = \sum_{i=1}^{3} \|\mathbf{P}_i\mathbf{y}\|^2$$

For $\mathbf{P}_1$, $\mathbf{P}_2$, *and* $\mathbf{P}_3$ *given in Example 2.6.3, it is easily verified that*

$$\|\mathbf{P}_1\mathbf{y}\|^2 = \mathbf{y}'\mathbf{P}_1\mathbf{y} = 4^2 y_{..}$$

$$\|\mathbf{P}_2\mathbf{y}\|^2 = \mathbf{y}'\mathbf{P}_2\mathbf{y} = \sum_i 2\,(y_{i.} - y_{..})^2$$

$$\|\mathbf{P}_3\mathbf{y}\|^2 = \mathbf{y}'\mathbf{P}_3\mathbf{y} = \sum_i \sum_j 2\,(y_{ij} - y_{i.})^2$$

Hence, the total sum of squares has the form

$$\mathbf{y}'\mathbf{I}\mathbf{y} = \sum_i \sum_j y_{ij}^2 = 4y_{..}^2 + \sum_i 2(y_{i.} - y_{..})^2 + \sum_i \sum_j (y_{ij} - y_{i.})^2$$

or

$$\sum_i \sum_j (y_{ij} - y_{..})^2 = \sum_i 2(y_{i.} - y_{..})^2 + \sum_i \sum_j (y_{ij} - y_{i.})^2$$

"Total about the Mean" SS = Between SS + Within SS

where the degrees of freedom are the ranks of $r(\mathbf{I} - \mathbf{P}_1) = n - 1$, $r(\mathbf{P}_2) = I - 1$, *and* $r(\mathbf{P}_3) = n - I$ *for* $n = 4$ *and* $I = 2$. *By Theorem 3.4.3, the sum of squares (SS) are independent and may be used to test hypotheses in analysis of variance, by forming ratios of independent chi-square statistics.*

Definition 3.4.2 *If* $\mathbf{Q} = \mathbf{Y}'\mathbf{Y}$ *and the matrix* $\mathbf{Y} \sim N_{n,\,p}(\mathbf{0}, \Sigma \otimes \mathbf{I}_n)$. *Then* $\mathbf{Q}$ *has a central p-dimensional Wishart distribution with n degrees of freedom and covariance matrix* Σ, *written as* $\mathbf{Q} \sim W_p(n, \Sigma)$.

For $E(\mathbf{Y}) = \mathbf{M}$ and $\mathbf{M} \neq \mathbf{0}$, $\mathbf{Q}$ has a noncentral Wishart distribution with noncentrality parameter $\Gamma = \mathbf{M}'\mathbf{M}\Sigma^{-1}$, written as $\mathbf{Q} \sim W_p(n, \Sigma, \Gamma)$. More formally, $\mathbf{Q} \sim W_p(n, \Sigma, \Gamma = \mathbf{M}'\mathbf{M}\Sigma^{-1})$ if and only if $\mathbf{a}'\mathbf{Q}\,\mathbf{a}/\mathbf{a}'\Sigma\mathbf{a} \sim \chi^2(n, \mathbf{a}'\mathbf{M}'\mathbf{M}\mathbf{a}\,/\mathbf{a}'\mathbf{M}\mathbf{a})$ for all nonnull vectors $\mathbf{a}$. In addition $E(\mathbf{Q}) = n\Sigma + \Sigma\Gamma = n\Sigma + \mathbf{M}'\mathbf{M}$ and $E(\mathbf{Y}'\mathbf{A}\mathbf{Y}) = \mathrm{tr}(\mathbf{A})\Sigma + \mathbf{M}'\mathbf{A}\mathbf{M}$ for a symmetric matrix $\mathbf{A}_{n\times n}$. For a comprehensive treatment of the noncentral Wishart distribution, see Muirhead (1982).

If $\mathbf{Q} \sim W_p(n, \Sigma)$, then the distribution of $\mathbf{Q}^{-1}$ is called an inverted Wishart distribution. That is $\mathbf{Q}^{-1} \sim W_p^{-1}(n + p + 1,\ \Sigma^{-1})$ and

$$E(\mathbf{Q}^{-1}) = \Sigma^{-1}/(n - p - 1)$$

for $n - p - 1 > 0$. Or, if $\mathbf{P} \sim W_p^{-1}(n^*, \mathbf{V}^{-1})$ then $E(\mathbf{P}) = \mathbf{V}^{-1}/(n^* - 2p - 2)$.

The Wishart distribution is a multivariate extension of the chi-square distribution and arises in the derivation of the distribution of the sample covariance matrix $\mathbf{S}$. For a random sample of n p-vectors, $\mathbf{Y}_i \sim N_p(\boldsymbol{\mu}, \Sigma)$ for $i = 1, \ldots, n$ and $n \geq p$,

$$(n - 1)\mathbf{S} = \sum_{i=1}^{n}(\mathbf{Y}_i' - \overline{\mathbf{Y}})(\mathbf{Y}_i - \overline{\mathbf{Y}})' \sim W_p(n - 1,\ \Sigma) \tag{3.4.1}$$

or

$$\mathbf{S} \sim W_p[n - 1,\ \Sigma/(n - 1)]$$

In multivariate analysis, the sum of independent Wishart distributions follows the same rules as in the univariate case. Matrix quadratic forms are often used in multivariate mixed models. Also important in multivariate analysis are the ratios of independently distributed Wishart matrices or, more specifically, the determinant and trace of matrix products or ratios which are functions of the eigenvalues of matrices. To construct distributions of roots of Wishart matrices, independence needs to be established. The multivariate extension for Cochran's Theorem follows.

Theorem 3.4.5 *If* $\mathbf{Y}_i \sim IN_p(\boldsymbol{\mu}, \Sigma)$ *for* $i = 1, \ldots, n$ *and* $\mathbf{Y}'\mathbf{Y} = \sum_{i=1}^{k} \mathbf{Y}'\mathbf{P}_i\mathbf{Y}$ *where* $\sum_{i=1}^{k} \mathbf{P}_i = \mathbf{I}_n$, *the forms* $\mathbf{Y}'\mathbf{P}_i\mathbf{Y} \sim W_p(r_i, \Sigma, \Gamma_i)$ *are statistically independent for all* i *if and only if* $\sum_{i=1}^{k} r_i = n$. *If* $r_i < p$, *the Wishart density does not exist.*

b. *Hotelling's T^2 Distribution*

A multivariate extension of Student's t distribution is Hotelling's T^2 distribution.

Definition 3.5.3 *Let* $\mathbf{Y}$ *and* $\mathbf{Q}$ *be independent random variables where* $\mathbf{Y} \sim N_p(\boldsymbol{\mu}, \Sigma)$ *and* $\mathbf{Q} \sim W_p(n, \Sigma)$, *and* $n > p$. *Then Hotelling's* T^2 *(1931) statistic*

$$T^2 = n\mathbf{Y}'\mathbf{Q}^{-1}\mathbf{Y}$$

has a distribution proportional to a noncentral F distribution

$$\frac{n-p+1}{p}\frac{T^2}{n} \sim F\left(p, n-p+1, \gamma\right)$$

where $\gamma = \boldsymbol{\mu}'\Sigma^{-1}\boldsymbol{\mu}$.

Example 3.5.1 *Let* $\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_n$ *be a random sample from a MVN population,* $\mathbf{Y}_i \sim IN_p(\boldsymbol{\mu}, \Sigma)$. *Then* $\overline{\mathbf{Y}} \sim N_p(\boldsymbol{\mu}, \Sigma/n)$ *and* $(n-1)\mathbf{S} \sim W_p(n-1, \Sigma)$, *and* $\overline{\mathbf{Y}}$ *and* $\mathbf{S}$ *are independent. Hence, for testing* $H_0 : \boldsymbol{\mu} = \boldsymbol{\mu}_0$ *vs.* $H_1 : \boldsymbol{\mu} \neq \boldsymbol{\mu}_0$, $T^2 = n(\overline{\mathbf{Y}} - \mu_0)'\mathbf{S}^{-1}\left(\overline{\mathbf{Y}} - \mu_0\right)$ *or*

$$\frac{n-p}{p}\frac{T^2}{n-1} = \frac{n\,(n-p)}{p\,(n-1)}\left(\overline{\mathbf{Y}} - \boldsymbol{\mu}_0\right)'\mathbf{S}^{-1}\left(\overline{\mathbf{Y}} - \boldsymbol{\mu}_0\right) \sim F\left(p, n-p, \gamma\right)$$

where

$$\gamma = n(\boldsymbol{\mu} - \boldsymbol{\mu}_0)'\Sigma^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_0)$$

is the noncentrality parameter. When H_0 *is true, the noncentrality parameter is zero and* T^2 *has a central F distribution.*

Example 3.5.3 *Replacing* **Q** *by* **S** *in Definition 3.5.3, Hotelling's* T^2 *statistic follows an F distribution*

$$(n-p)\,T^2/\,(n-1)\,p \sim F\,(p, n-p, \gamma)$$

For $\gamma = 0$,

$$E(T^2) = (n-1)\,p/\,(n-p-2)$$

$$\mathrm{var}(T^2) = \frac{2p\,(n-1)^2\,(n-2)}{(n-p-2)^2\,(n-p-4)}$$

By Theorem 3.4.2, $T^2 \xrightarrow{d} \chi^2\,(p)$ *as* $n \longrightarrow \infty$. *However, for small values of n, the distribution of* T^2 *is far from chi-square. If* $X^2 \sim \chi^2\,(p)$, *then* $E\left(X^2\right) = p$ *and the* $\mathrm{var}\left(X^2\right) = 2p$. *Thus, if one has a statistic* $T^2 \xrightarrow{d} \chi^2\,(p)$, *a better approximation for small to moderate sample sizes is the statistic*

$$\frac{(n-p)\,T^2}{(n-1)\,p} \dot{\sim} F\,(p, n-p, \gamma)$$

c. *The Beta Distribution*

A distribution closely associated with the F distribution is the beta distribution.

Definition 3.5.4 *Let H and E be independent random variables such that* $H \sim \chi^2(v_h, \gamma)$ *and* $E \sim \chi^2(v_e, \gamma = 0)$. *Then*

$$B = \frac{H}{H+E} \sim \mathrm{beta}\,(v_h/2,\, v_e/2,\, \gamma)$$

has a noncentral (Type I) beta distribution and

$$V = H/E \sim \textit{Inverted beta}\,(v_h/2,\, v_e/2,\, \gamma)$$

has a (Type II) beta or inverted noncentral beta distribution.

To generalize (3.6.3) to the multivariate (linear) regression model, a model is formulated for each of p correlated dependent, response variables

$$\begin{array}{lclclclclcl} \mathbf{y}_1 & = & \beta_{01}\mathbf{1}_n & + & \beta_{11}\mathbf{x}_1 & +\cdots+ & \beta_{k1}\mathbf{x}_k & + & \mathbf{e}_1 \\ \mathbf{y}_2 & = & \beta_{02}\mathbf{1}_n & + & \beta_{12}\mathbf{x}_2 & +\cdots+ & \beta_{k2}\mathbf{x}_k & + & \mathbf{e}_2 \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ \mathbf{y}_p & = & \beta_{0p}\mathbf{1}_n & + & \beta_{1p}\mathbf{x}_p & +\cdots+ & \beta_{kp}\mathbf{x}_k & + & \mathbf{e}_p \end{array} \tag{3.6.15}$$

Each of the vectors $\mathbf{y}_j$, $\mathbf{x}_j$ and $\mathbf{e}_j$, for $j = 1, 2, \ldots, p$ are $n \times 1$ vectors. Hence, we have n observations for each of p variables. To represent (3.6.15) in matrix form, we construct matrices using each variable as a column vector. That is,

$$\begin{aligned} \mathbf{Y}_{n\times p} &= [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_p] \\ \mathbf{X}_{n\times q} &= [\mathbf{1}_n, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k] \\ \mathbf{B}_{q\times p} &= [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_p] \\ &= \begin{bmatrix} \beta_{01} & \beta_{01} & \cdots & \beta_{0p} \\ \beta_{11} & \beta_{11} & \cdots & \beta_{1p} \\ \vdots & \vdots & & \vdots \\ \beta_{k1} & \beta_{k2} & \cdots & \beta_{kp} \end{bmatrix} \\ \mathbf{E}_{n\times p} &= [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_p] \end{aligned} \tag{3.6.16}$$

Then for $q = k + 1$, the matrix linear model for (3.6.15) becomes

$$\begin{aligned}\mathbf{Y}_{n\times p} &= \mathbf{X}_{n\times q}\mathbf{B}_{q\times p} + \mathbf{E}_{n\times p} \\ &= \left[\mathbf{X}\boldsymbol{\beta}_1, \mathbf{X}\boldsymbol{\beta}_2, \ldots, \mathbf{X}\boldsymbol{\beta}_p\right] + \left[\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_p\right]\end{aligned} \tag{3.6.17}$$

Model (3.6.17) is called the multivariate (linear) regression (MR) model , or MLMR model. If the $r(\mathbf{X}) < q = k+1$, so that the design matrix is not of full rank, the model is called the multivariate analysis of variance (MANOVA) model. Partitioning $\mathbf{X}$ into two matrices as in the univariate regression model, $\mathbf{X} = [\mathbf{A}, \mathbf{Z}]$ and $\mathbf{B}' = [\Theta', \Gamma']$, model (3.6.17) becomes the multivariate analysis of covariance (MANCOVA) model.

To represent the MR model as a GLM, the vec $(\cdot)$ operator is employed. Let $\mathbf{y} = \text{vec}(\mathbf{Y})$, $\boldsymbol{\beta} = \text{vec}(\mathbf{B})$ and $\mathbf{e} = \text{vec}(\mathbf{Y})$. Since the design matrix $\mathbf{X}_{n\times q}$ is the same for each of the p dependent variables, the GLM for the MR model is as follows

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_p \end{bmatrix} = \begin{bmatrix} \mathbf{X} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{X} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \\ \vdots \\ \boldsymbol{\beta}_p \end{bmatrix} + \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \vdots \\ \mathbf{e}_p \end{bmatrix}$$

Or, for $N = np$ and $K = pq = p(k+1)$, we have the vector form of the MLMR model

$$\begin{aligned}\mathbf{y}_{N\times 1} &= \left(\mathbf{I}_p \otimes \mathbf{X}\right)_{N\times K} \boldsymbol{\beta}_{K\times 1} + \mathbf{e}_{K\times 1} \\ \text{cov}\,(\mathbf{y}) &= \Sigma \otimes \mathbf{I}_n\end{aligned} \tag{3.6.18}$$

To test hypotheses, we assume that $\mathbf{E}$ in (3.6.17) has a matrix normal distribution, $\mathbf{E} \sim N_{n,\,p}(\mathbf{0}, \Sigma \otimes \mathbf{I}_n)$ or using the row representation that $\mathbf{E}' \sim N_{n,\,p}(\mathbf{0}, \mathbf{I}_n \otimes \Sigma)$. Alternatively, by (3.6.18), $\mathbf{e} \sim N_{np}(\mathbf{0}, \Sigma \otimes \mathbf{I}_n)$. To obtain the ML estimate of $\boldsymbol{\beta}$ given (3.6.18), we associate the covariance structure $\Sigma$with $\Sigma \otimes \mathbf{I}_n$ and apply (3.6.8), even though Σ is unknown. The unknown matrix drops out of the product. To see this, we have by substitution that

$$\begin{aligned}\widehat{\boldsymbol{\beta}}_{ML} &= \left[(\mathbf{I}_p \otimes \mathbf{X})'(\Sigma \otimes \mathbf{I}_n)^{-1}(\mathbf{I}_p \otimes \mathbf{X})\right]^{-1}(\mathbf{I}_p \otimes \mathbf{X})'(\Sigma \otimes \mathbf{I}_n)^{-1}\mathbf{y} \\ &= \left(\Sigma^{-1} \otimes \mathbf{X}'\mathbf{X}\right)^{-1}\left(\Sigma^{-1} \otimes \mathbf{X}'\right)\mathbf{y} \qquad (3.6.19)\end{aligned}$$

However, by property (5) in Theorem 2.4.7, we have that

$$\widehat{\boldsymbol{\beta}}_{ML} = \operatorname{vec}\left[\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y}\right]$$

by letting $\mathbf{A} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'$ and $\mathbf{C}' = \mathbf{I}_p$. Thus,

$$\widehat{\mathbf{B}}_{ML} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y} \qquad (3.6.20)$$

using the matrix form of the model. This is also the OLS estimate of the parameter matrix.

Similarly using (3.6.19), the

$$\begin{aligned}\operatorname{cov}\left(\widehat{\boldsymbol{\beta}}_{ML}\right) &= \left[\left(\mathbf{I}_p \otimes \mathbf{X}\right)' \left(\Sigma \otimes \mathbf{I}_n\right)^{-1} \left(\mathbf{I}_p \otimes \mathbf{X}\right)\right]^{-1} \\ &= \Sigma \otimes \left(\mathbf{X}'\mathbf{X}\right)^{-1}\end{aligned}$$

Finally, the ML estimate of Σ is

$$\widehat{\Sigma} = \mathbf{Y}' \left[\mathbf{I}_n - \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1} \mathbf{X}'\right] \mathbf{Y}/n \tag{3.6.21}$$

or the restricted maximum likelihood (REML) unbiased estimate is $\mathbf{S} = \mathbf{E}/(n-q)$ where $q = r(\mathbf{X})$. Furthermore $\widehat{\boldsymbol{\beta}}_{ML}$ and $\widehat{\Sigma}$ are independent, and $n\,\widehat{\Sigma} \sim W_p(n-q,\ \Sigma)$. Again, the Wishart density only exists if $n \geq p + q$.

In the above discussion, we have assumed that $\mathbf{X}$ has full column rank q. If the $r(\mathbf{X}) = r < q$, then $\widehat{\mathbf{B}}$ is not unique since $(\mathbf{X}'\mathbf{X})^{-1}$ is replaced with a g-inverse. However, $\widehat{\Sigma}$ is still unique since $\left(\mathbf{I}_n - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\right)$ is a unique projection matrix by property (4), Theorem

The null hypothesis tested for the matrix form of the MR model takes the general form

$$H : \mathbf{CBM} = \mathbf{0} \tag{3.6.22}$$

where $\mathbf{C}_{g \times q}$ is a known matrix of full row rank g, $g \leq q$ and $\mathbf{M}_{p \times u}$ is a matrix of full column rank $u \leq p$. Hypothesis (3.6.22) is called the standard multivariate hypothesis. To test (3.6.22) using the vector form of the model, observe that $\text{vec}(\mathbf{C}\widehat{\mathbf{B}}\mathbf{M}) = (\mathbf{M}' \otimes \mathbf{C})\,\text{vec}\,\widehat{\mathbf{B}}$ so that (3.6.22) is equivalent to testing $H : \mathbf{L}\boldsymbol{\beta} = \mathbf{0}$ when $\mathbf{L}$ is a matrix of order $gu \times pq$ of rank $v = gu$. Assuming $\Omega = \Sigma \otimes \mathbf{I}_n$ is known,

$$\widehat{\boldsymbol{\beta}}_{ML} \sim N_{gu}(\boldsymbol{\beta}, \mathbf{L}[(\mathbf{I}_n \otimes \mathbf{X})'\Omega^{-1}(\mathbf{I}_p \otimes \mathbf{X})]^{-1}\mathbf{L}') \tag{3.6.23}$$

Simplifying the structure of the covariance matrix,

$$\text{cov}\left(\widehat{\boldsymbol{\beta}}_{ML}\right) = \left(\mathbf{M}'\Sigma\mathbf{M}\right) \otimes \left(\mathbf{C}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{C}'\right) \tag{3.6.24}$$

For known Σ, the likelihood ratio test of H is to reject H if $X^2 > c_\alpha$ where c_α is chosen such that the $P(X^2 > c_\alpha \mid H) = \alpha$ and $X^2 = \widehat{\boldsymbol{\beta}}'_{ML}[(\mathbf{M}'\Sigma\mathbf{M}) \otimes (\mathbf{C}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{C}')]^{-1}\widehat{\boldsymbol{\beta}}_{ML}$.

However, we can simplify X^2 since

$$
\begin{aligned}
X^2 &= [\mathrm{vec}(\mathbf{C}\widehat{\mathbf{B}}\mathbf{M})]'\left[\left(\mathbf{M}'\Sigma\mathbf{M}\right)^{-1} \otimes \left(\mathbf{C}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\right)\right]\mathrm{vec}(\mathbf{C}\widehat{\mathbf{B}}\mathbf{M}) \\
&= [\mathrm{vec}(\mathbf{C}\widehat{\mathbf{B}}\mathbf{M})]'\,\mathrm{vec}[(\mathbf{C}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{C}']^{-1}(\mathbf{C}\widehat{\mathbf{B}}\mathbf{M})\left(\mathbf{M}'\Sigma\mathbf{M}\right)^{-1} \\
&= \mathrm{tr}[(\mathbf{C}\widehat{\mathbf{B}}\mathbf{M})'[\mathbf{C}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{C}']^{-1}(\mathbf{C}\widehat{\mathbf{B}}\mathbf{A})\left(\mathbf{M}'\Sigma\mathbf{M}\right)^{-1}] \qquad (3.6.25)
\end{aligned}
$$

Thus to test $H : \mathbf{L}\boldsymbol{\beta} = \mathbf{0}$, the hypothesis is rejected if X^2 in (3.6.25) is larger than a chi-square critical value with $v = gu$ degrees of freedom. Again, by finding a consistent estimate of Σ, $X^2 \xrightarrow{d} \chi^2(v = gu)$. Thus an approximate test of H is available if Σ is estimated by $\widehat{\Sigma} \xrightarrow{p} \Sigma$.

Evaluating Normality

The study of robust estimation for location and dispersion of model parameters, the identification of outliers, the analysis of multivariate residuals, and the assessment of the effects of model assumptions on tests of significance and power are as important in multivariate analysis as they are in univariate analysis. However, the problems are much more complex. In multivariate data analysis there is no natural one-dimensional order to the observations, hence we can no longer just investigate the extremes of the distribution to locate outliers or identify data clusters in only one dimension. Clusters can occur in some subspace and outliers may not be extreme in any one dimension. Outliers in multivariate samples effect not only the location and variance of a variable, but also its orientation in the sample as measured by the covariance or correlation with other variables. Residuals formed from fitting a multivariate model to a data set in the presence of extreme outliers may lead to the identification of spurious outliers. Upon replotting the data, they are often removed. Finally, because non-normality can occur in so many ways robustness studies of Type I errors and power are difficult to design and evaluate.

The two most important problems in multivariate data analysis are the detection of outliers and the evaluation of multivariate normality. The process is complex and first begins with the assessment of marginal normality, a variable at a time; see Looney (1995). The evaluation process usually proceeds as follows.

1. Evaluate univariate normality by performing the Shapiro and Wilk (1965) W test a variable at a time when sample sizes are less than or equal to 50. The test is known to show a reasonable sensitivity to nonnormality; see Shapiro et al. (1968). For $50 < n \leq 2000$, Royston's (1982, 1992) approximation is recommended and is implemented

2. Construct normal probability quantile-vs-quantile (Q-Q) plots a variable at a time which compare the cumulative empirical distribution with the expected order values of a normal density to informally assess the lack of linearity and the presence of extreme values; see Wilk and Gnanadesikan (1968) and Looney and Gulledge (1985).

3. If variables are found to be non-normal, transform them to normality using perhaps a Box and Cox (1964) power transformation or some other transformation such as a logit.

4. Locate and correct outliers using graphical techniques or tests of significance as outlined by Barnett and Lewis (1994).

Example 3.7.1 (Generating MVN Distributions) *To illustrate the analysis of multivariate data, several multivariate normal distributions are generated. The data generated are used to demonstrate several of the procedures for evaluating multivariate normality and testing hypotheses about means and covariance matrices.*

By using the properties of the MVN distribution, recall that if $\mathbf{z} \sim N_p(\mathbf{0}, \mathbf{I}_p)$, *then* $\mathbf{y} = \mathbf{zA} + \boldsymbol{\mu} \sim N_p(\boldsymbol{\mu}, \Sigma = \mathbf{A}'\mathbf{A})$. *Hence, to generate a MVN distribution with mean* $\boldsymbol{\mu}$ *and covariance matrix* Σ, *one proceeds as follows.*

1. Specify $\boldsymbol{\mu}$ and Σ.

2. Obtain a Cholesky decomposition for Σ; call it $\mathbf{A}$.

3. Generate a $n \times p$ matrix of $N(0, 1)$ random variables named $\mathbf{Z}$.

4. Transform $\mathbf{Z}$ to $\mathbf{Y}$ using the expression $\mathbf{Y} = \mathbf{ZA} + \mathbf{U}$ where $\mathbf{U}$ is created by repeating the row vector $\mathbf{u}'$ n times producing an $n \times p$ matrix.

Multivariate Analysis Data into R

Data sets are available from the UCI Machine LearningRepository http://archive.ics.uci.edu/ml

For example, the file http://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data contains data on concentrations of 13 different chemicals in wines grown in the same region in Italy that are derived from three different cultivars.

The data set looks like this:

```
1,14.23,1.71,2.43,15.6,127,2.8,3.06,.28,2.29,5.64,1.04,3.92,1065
1,13.2,1.78,2.14,11.2,100,2.65,2.76,.26,1.28,4.38,1.05,3.4,1050
1,13.16,2.36,2.67,18.6,101,2.8,3.24,.3,2.81,5.68,1.03,3.17,1185
1,14.37,1.95,2.5,16.8,113,3.85,3.49,.24,2.18,7.8,.86,3.45,1480
1,13.24,2.59,2.87,21,118,2.8,2.69,.39,1.82,4.32,1.04,2.93,735
...
```

There is one row per wine sample. 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.

When we read the file into R using the read.table() function, we need to use the "sep=" argument in read.table() to tell it that the columns are separated by commas. That is, we can read in the file using the read.table() function as follows:

```
> wine <- read.table("http://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data",
         sep=",")
> wine
    V1    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  5.640000 1.040 3.92 1065
2    1 13.20 1.78 2.14 11.2 100 2.65 2.76 0.26 1.28  4.380000 1.050 3.40 1050
3    1 13.16 2.36 2.67 18.6 101 2.80 3.24 0.30 2.81  5.680000 1.030 3.17 1185
4    1 14.37 1.95 2.50 16.8 113 3.85 3.49 0.24 2.18  7.800000 0.860 3.45 1480
5    1 13.24 2.59 2.87 21.0 118 2.80 2.69 0.39 1.82  4.320000 1.040 2.93  735
...
176  3 13.27 4.28 2.26 20.0 120 1.59 0.69 0.43 1.35 10.200000 0.590 1.56  835
177  3 13.17 2.59 2.37 20.0 120 1.65 0.68 0.53 1.46  9.300000 0.600 1.62  840
178  3 14.13 4.10 2.74 24.5  96 2.05 0.76 0.56 1.35  9.200000 0.610 1.60  560
```

In this case the data on 178 samples of wine has been read into the variable 'wine'.

A Matrix Scatterplot

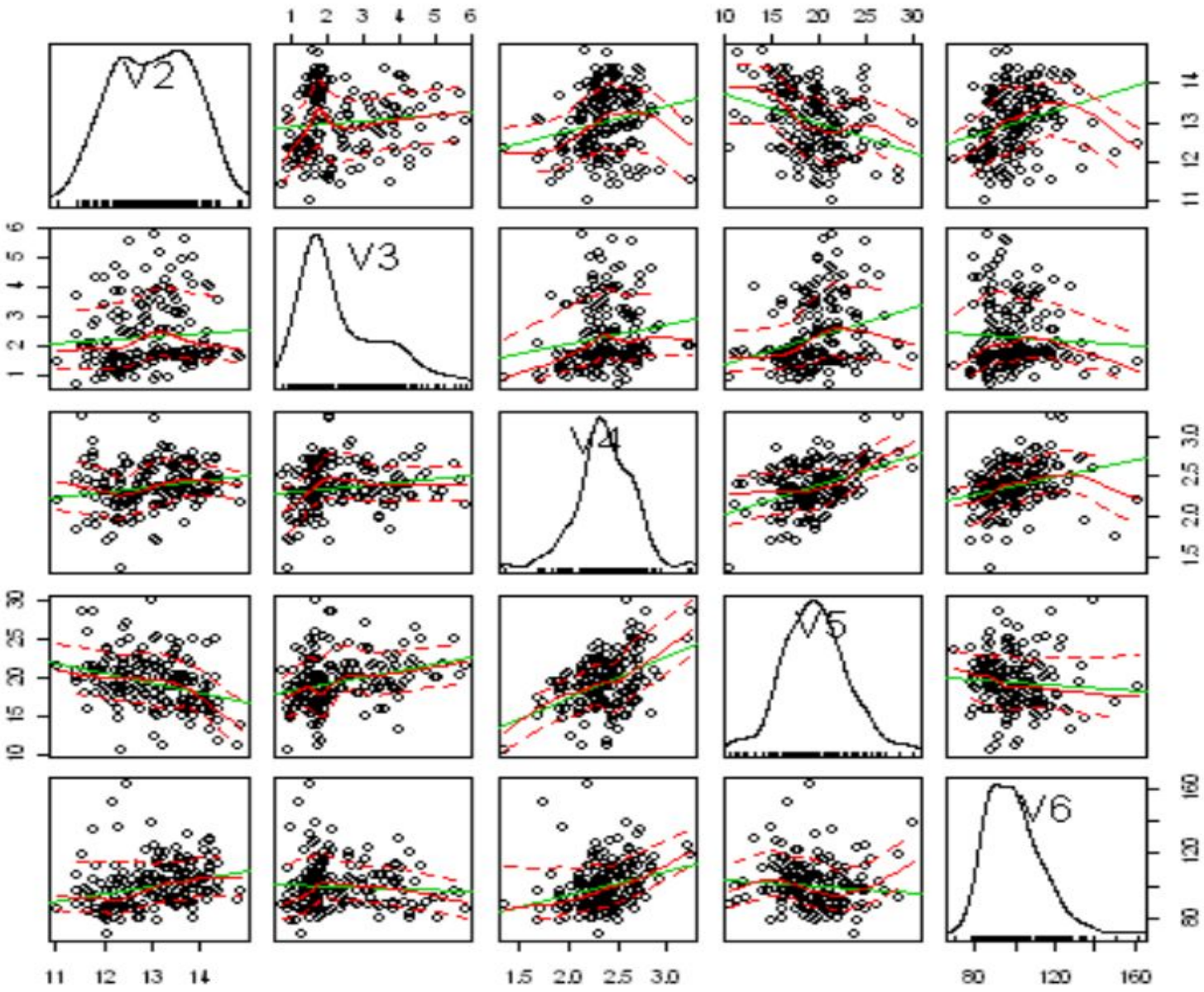
One common way of plotting multivariate data is to make a "matrix scatterplot", showing each pair of variables plotted against each other. We can use the "scatterplotMatrix()" function from the "car" R package to do this. To use this function, we first need to install the "car" R package (for instructions on how to install an R package

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).

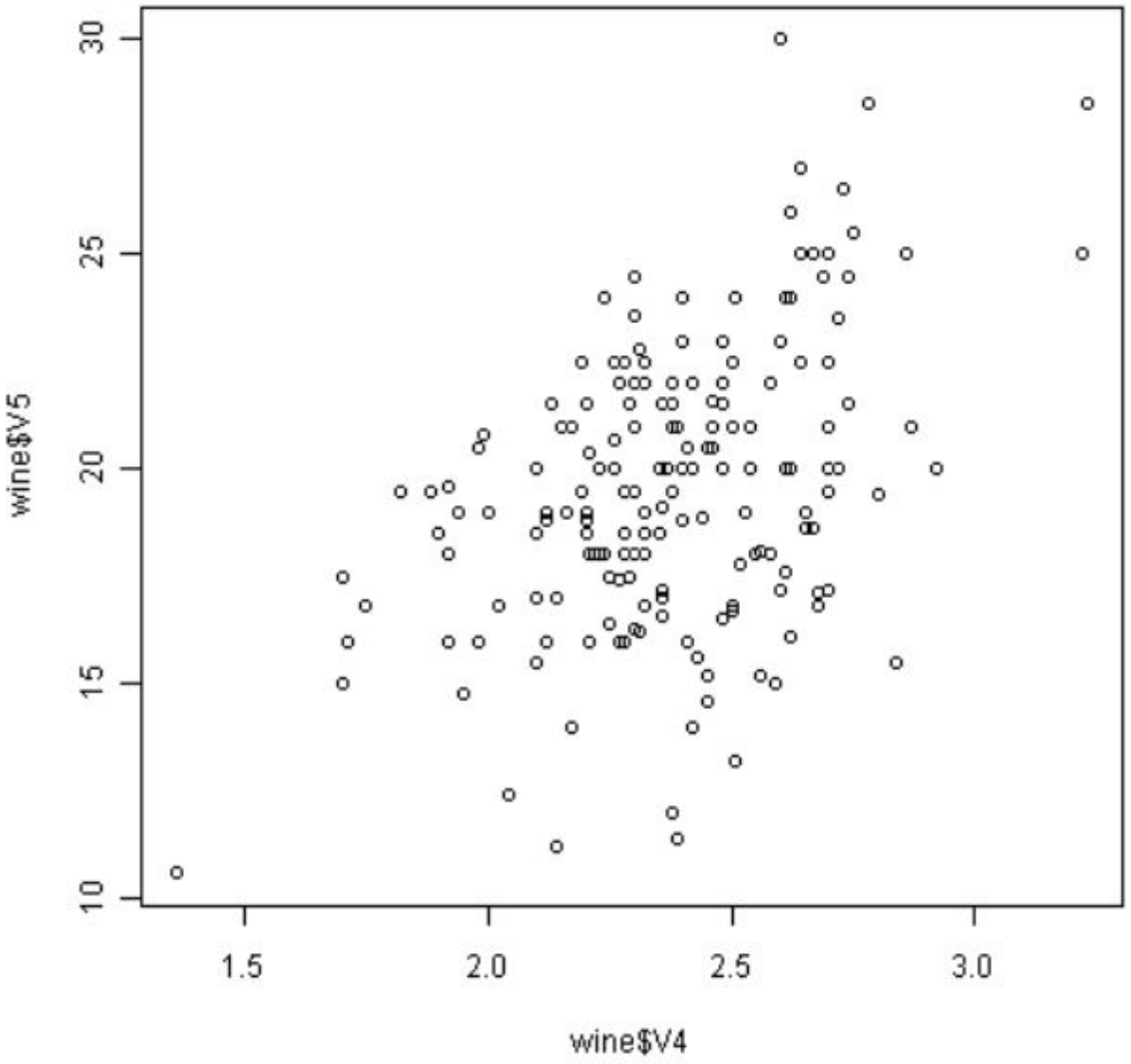
If you see an interesting scatterplot for two variables in the matrix scatterplot, you may want to plot that scatterplot in more detail, with the data points labelled by their group (their cultivar in this case).
For example, in the matrix scatterplot above, the cell in the third column of the fourth row down is a scatterplot of V5 (x-axis) against V4 (y-axis). If you look at this scatterplot, it appears that there may be a positive relationship between V5 and V4.

We may therefore decide to examine the relationship between V5 and V4 more closely, by plotting a scatterplot of these two variable, with the data points labelled by their group (their cultivar). To plot a scatterplot of two variables, we can use the "plot" R function. The V4 and V5 variables are stored in the columns V4 and V5 of the variable "wine", so can be accessed by typing wine$V4 or wine$V5. Therefore, to plot the scatterplot, we type:

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

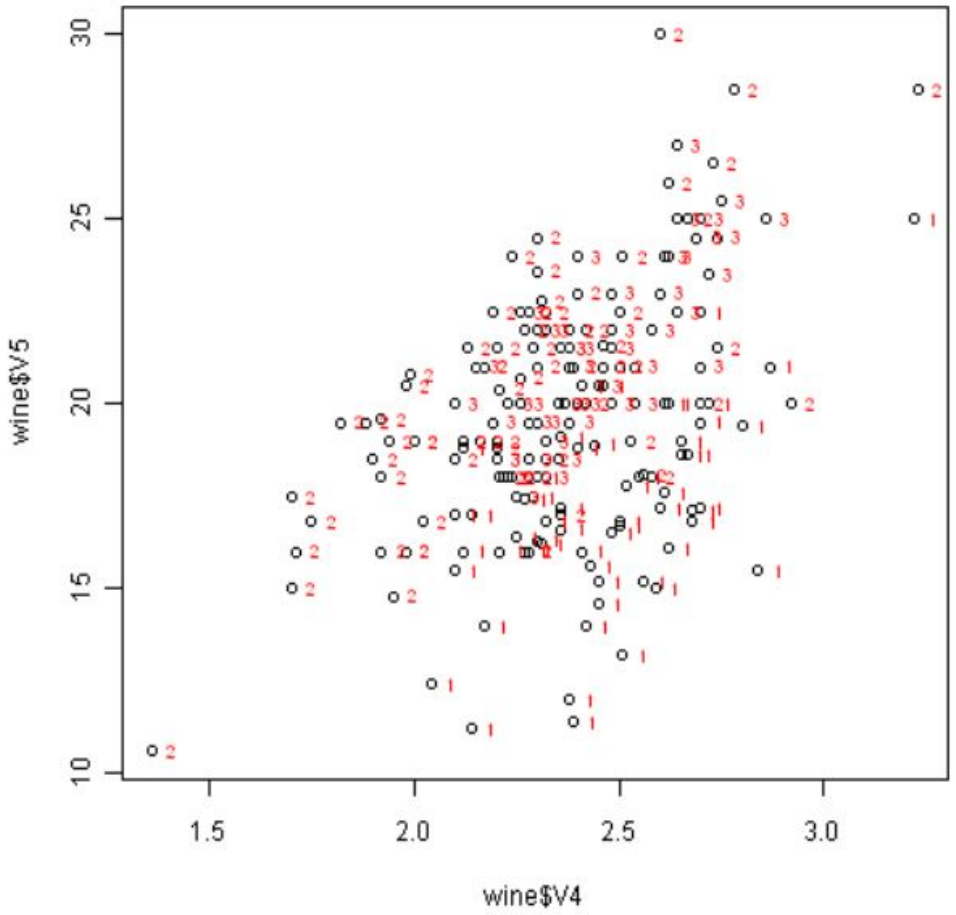


```
> plot(wine$V4, wine$V5)
```



```
> text(wine$V4, wine$V5, wine$V1, cex=0.7, 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:



We can see from the scatterplot of V4 versus V5 that the wines from cultivar 2 seem to have lower values of V4 compared to the wines of cultivar 1.

A Profile Plot

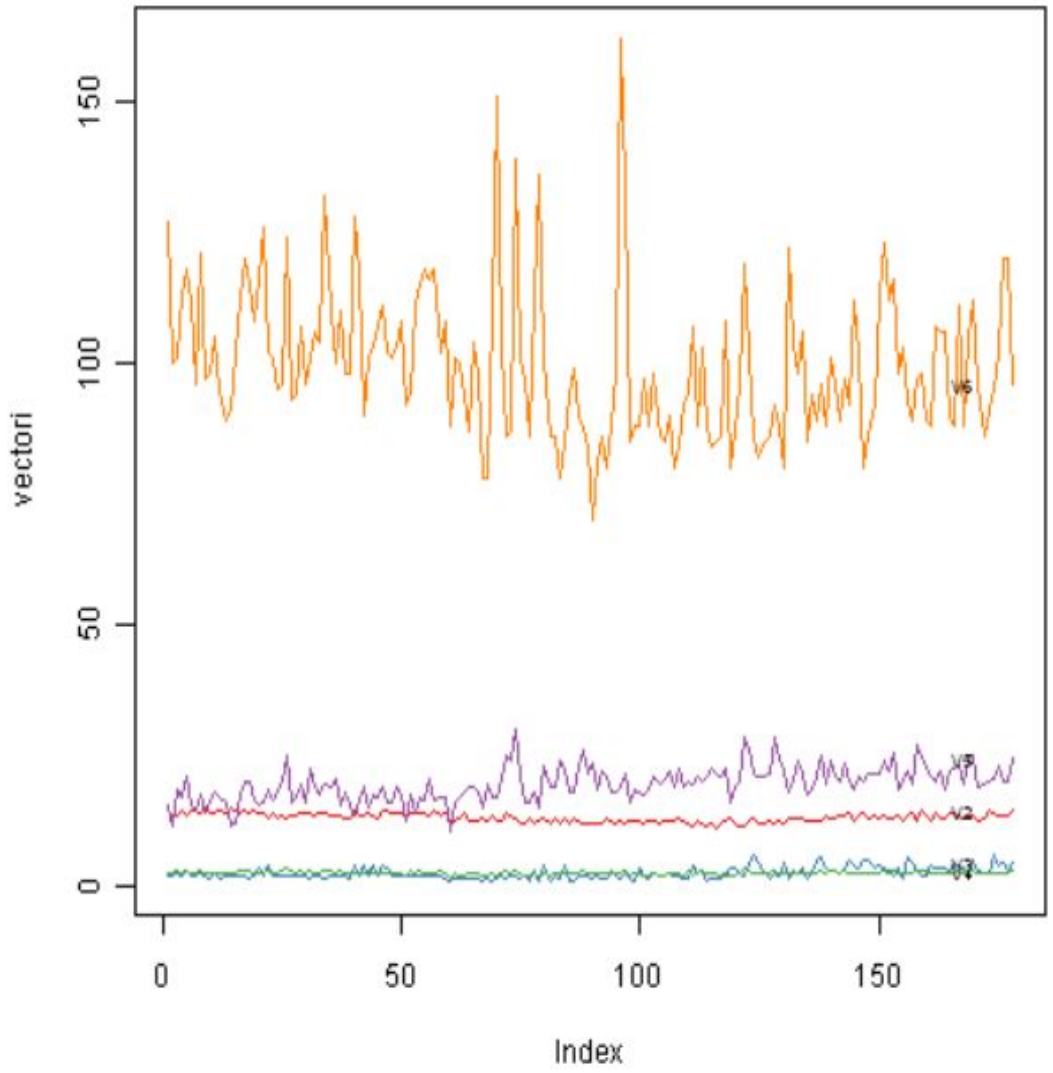
Another type of plot that is useful is a "profile plot", which shows the variation in each of the variables, by plotting the value of each of the variables for each of the samples.

The function "makeProfilePlot()" below can be used to make a profile plot. This function requires the "RColorBrewer"

To use this function, you first need to copy and paste it into R. The arguments to the function are a vector containing the names of the varibles that you want to plot, and a list variable containing the variables themselves.

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:

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



It is clear from the profile plot that the mean and standard deviation for V6 is quite a lot higher than that for the other variables.

Calculating Summary Statistics for Multivariate Data

Another thing that you are likely to want to do is to calculate summary statistics such as the mean and standard deviation for each of the variables in your multivariate data set.

This is easy to do, using the "mean()" and "sd()" functions in R. For example, say we want to calculate the mean and standard deviations of each of the 13 chemical concentrations in the wine samples. These are stored in columns 2-14 of the variable "wine". So we type

```
> sapply(wine[2:14],mean)
        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
```

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

```
> sapply(wine[2:14],sd)
        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
```

We can see here that it would make sense to standardise in order to compare the variables because the variables have very different standard deviations – the standard deviation of V14 is 314.9074743, while the standard deviation of V9 is just 0.1244533. Thus, in order to compare the variables, we need to standardise each variable so that it has a sample variance of 1 and sample mean of 0. We will explain below how to standardise the variables.

Means and Variances Per Group ¶

It is often interesting to calculate the means and standard deviations for just the samples from a particular group, for example, for the wine samples from each cultivar. The cultivar is stored in the column "V1" of the variable "wine".

To extract out the data for just cultivar 2, we can type:

```
> cultivar2wine <- wine[wine$V1=="2",]
```

We can then calculate the mean and standard deviations of the 13 chemicals' concentrations, for just the cultivar 2 samples:

```
> sapply(cultivar2wine[2:14],mean)
    V2         V3         V4         V5         V6         V7         V8
 12.278732   1.932676   2.244789  20.238028  94.549296   2.258873   2.080845
    V9        V10        V11        V12        V13        V14
  0.363662   1.630282   3.086620   1.056282   2.785352 519.507042
> sapply(cultivar2wine[2:14])
    V2          V3          V4          V5          V6          V7          V8
 0.5379642   1.0155687   0.3154673   3.3497704  16.7534975   0.5453611   0.7057008
    V9         V10         V11         V12         V13         V14
 0.1239613   0.6020678   0.9249293   0.2029368   0.4965735 157.2112204
```

To use the function "printMeanAndSdByGroup()", you first need to copy and paste it into R. The arguments of the function are the variables that you want to calculate means and standard deviations for, and the variable containing the group of each sample. For example, to calculate the mean and standard deviation for each of the 13 chemical concentrations, for each of the three different wine cultivars, we type:

```
> printMeanAndSdByGroup(wine[2:14],wine[1])
  [1] "Means:"
    V1       V2       V3       V4       V5       V6       V7        V8       V9      V10      V11
  1  1 13.74475 2.010678 2.455593 17.03729 106.3390 2.840169 2.9823729 0.290000 1.899322 5.528305 1.06
  2  2 12.27873 1.932676 2.244789 20.23803  94.5493 2.258873 2.0808451 0.363662 1.630282 3.086620 1.05
  3  3 13.15375 3.333750 2.437083 21.41667  99.3125 1.678750 0.7814583 0.447500 1.153542 7.396250 0.68
  [1] "Standard deviations:"
    V1        V2        V3        V4       V5       V6        V7        V8         V9       V10
  1  1 0.4621254 0.6885489 0.2271660 2.546322 10.49895 0.3389614 0.3974936 0.07004924 0.4121092 1.2385
  2  2 0.5379642 1.0155687 0.3154673 3.349770 16.75350 0.5453611 0.7057008 0.12396128 0.6020678 0.9249
  3  3 0.5302413 1.0879057 0.1846902 2.258161 10.89047 0.3569709 0.2935041 0.12413959 0.4088359 2.3109
  [1] "Sample sizes:"
    V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14
  1  1 59 59 59 59 59 59 59 59  59  59  59  59  59
  2  2 71 71 71 71 71 71 71 71  71  71  71  71  71
  3  3 48 48 48 48 48 48 48 48  48  48  48  48  48
```

The function "printMeanAndSdByGroup()" also prints out the number of samples in each group. In this case, we see that there are 59 samples of cultivar 1, 71 of cultivar 2, and 48 of cultivar 3.

Between-groups Variance and Within-groups Variance for a Variable

If we want to calculate the within-groups variance for a particular variable (for example, for a particular chemical's concentration), we can use the function "calcWithinGroupsVariance()" below:

You will need to copy and paste this function into R before you can use it. For example, to calculate the within-groups variance of the variable V2 (the concentration of the first chemical), we type:

```
> calcWithinGroupsVariance(wine[2],wine[1])
  [1] 0.2620525
```

Thus, the within-groups variance for V2 is 0.2620525.

We can calculate the between-groups variance for a particular variable (eg. V2) using the function "calcBetweenGroupsVariance()" below:

Once you have copied and pasted this function into R, you can use it to calculate the between-groups variance for a variable such as V2:

```
> calcBetweenGroupsVariance (wine[2],wine[1])
  [1] 35.39742
```

Thus, the between-groups variance of V2 is 35.39742.

We can calculate the "separation" achieved by a variable as its between-groups variance devided by its within-groups variance. Thus, the separation achieved by V2 is calculated as:

```
> 35.39742/0.2620525
  [1] 135.0776
```

If you want to calculate the separations achieved by all of the variables in a multivariate data set, you can use the function "calcSeparations()" below:

For example, to calculate the separations for each of the 13 chemical concentrations, we type:

```
> calcSeparations(wine[2:14],wine[1])
  [1] "variable V2 Vw= 0.262052469153907 Vb= 35.3974249602692 separation= 135.0776242428"
  [1] "variable V3 Vw= 0.887546796746581 Vb= 32.7890184869213 separation= 36.9434249631837"
  [1] "variable V4 Vw= 0.0660721013425184 Vb= 0.879611357248741 separation= 13.312901199991"
  [1] "variable V5 Vw= 8.00681118121156 Vb= 286.41674636309 separation= 35.7716374073093"
  [1] "variable V6 Vw= 180.65777316441 Vb= 2245.50102788939 separation= 12.4295843381499"
  [1] "variable V7 Vw= 0.191270475224227 Vb= 17.9283572942847 separation= 93.7330096203673"
  [1] "variable V8 Vw= 0.274707514337437 Vb= 64.2611950235641 separation= 233.925872681549"
  [1] "variable V9 Vw= 0.0119117022132797 Vb= 0.328470157461624 separation= 27.5754171469659"
  [1] "variable V10 Vw= 0.246172943795542 Vb= 7.45199550777775 separation= 30.2713831702276"
  [1] "variable V11 Vw= 2.28492308133354 Vb= 275.708000822304 separation= 120.664018441003"
  [1] "variable V12 Vw= 0.0244876469432414 Vb= 2.48100991493829 separation= 101.3167953903"
  [1] "variable V13 Vw= 0.160778729560982 Vb= 30.5435083544253 separation= 189.972320578889"
  [1] "variable V14 Vw= 29707.6818705169 Vb= 6176832.32228483 separation= 207.920373902178"
```

Thus, the individual variable which gives the greatest separations between the groups (the wine cultivars) is V8 (separation 233.9). As we will discuss below, the purpose of linear discriminant analysis (LDA) is to find the linear combination of the individual variables that will give the greatest separation between the groups (cultivars here). This hopefully will give a better separation than the best separation achievable by any individual variable (233.9 for V8 here).

Between-groups Covariance and Within-groups Covariance for Two Variables

If you have a multivariate data set with several variables describing sampling units from different groups, such as the wine samples from different cultivars, it is often of interest to calculate the within-groups covariance and between-groups variance for pairs of the variables.

For example, to calculate the within-groups covariance for variables V8 and V11, we type:

```
> calcWithinGroupsCovariance(wine[8],wine[11],wine[1])
[1] 0.2866783
```

For example, to calculate the between-groups covariance for variables V8 and V11, we type:

```
> calcBetweenGroupsCovariance(wine[8],wine[11],wine[1])
[1] -60.41077
```

Thus, for V8 and V11, the between-groups covariance is -60.41 and the within-groups covariance is 0.29. Since the within-groups covariance is positive (0.29), it means V8 and V11 are positively related within groups: for individuals from the same group, individuals with a high value of V8 tend to have a high value of V11, and vice versa. Since the between-groups covariance is negative (-60.41), V8 and V11 are negatively related between groups: groups with a high mean value of V8 tend to have a low mean value of V11, and vice versa.

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 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
```

This tells us that the correlation coefficient is about 0.094, which is a very weak correlation. Furthermore, the P-value for the statistical test of whether the correlation coefficient is significantly different from zero is 0.21. This is much greater than 0.05 (which we can use here as a cutoff for statistical significance), so there is very weak evidence that that the correlation is non-zero.

If you have a lot of variables, you can use "cor.test()" to calculate the correlation coefficient for each pair of variables, but you might be just interested in finding out what are the most highly correlated pairs of variables. For this you can use the function "mosthighlycorrelated()" below.

The function "mosthighlycorrelated()" will print out the linear correlation coefficients for each pair of variables in your data set, in order of the correlation coefficient. This lets you see very easily which pair of variables are most highly correlated.

For example, to calculate correlation coefficients between the concentrations of the 13 chemicals in the wine samples, and to print out the top 10 pairwise correlation coefficients, you can type:

```
> mosthighlycorrelated(wine[2:14], 10)
     First.Variable Second.Variable Correlation
84               V7              V8   0.8645635
150              V8             V13   0.7871939
149              V7             V13   0.6999494
111              V8             V10   0.6526918
157              V2             V14   0.6437200
110              V7             V10   0.6124131
154             V12             V13   0.5654683
132              V3             V12  -0.5612957
118              V2             V11   0.5463642
137              V8             V12   0.5434786
```

This tells us that the pair of variables with the highest linear correlation coefficient are V7 and V8 (correlation = 0.86 approximately).

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 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
```

This tells us that the correlation coefficient is about 0.094, which is a very weak correlation. Furthermore, the P-value for the statistical test of whether the correlation coefficient is significantly different from zero is 0.21. This is much greater than 0.05 (which we can use here as a cutoff for statistical significance), so there is very weak evidence that that the correlation is non-zero.

If you have a lot of variables, you can use "cor.test()" to calculate the correlation coefficient for each pair of variables, but you might be just interested in finding out what are the most highly correlated pairs of variables. For this you can use the function "mosthighlycorrelated()" below.

For example, to calculate correlation coefficients between the concentrations of the 13 chemicals in the wine samples, and to print out the top 10 pairwise correlation coefficients, you can type:

```
> mosthighlycorrelated(wine[2:14], 10)
     First.Variable Second.Variable Correlation
84              V7              V8   0.8645635
150             V8             V13   0.7871939
149             V7             V13   0.6999494
111             V8             V10   0.6526918
157             V2             V14   0.6437200
110             V7             V10   0.6124131
154            V12             V13   0.5654683
132             V3             V12  -0.5612957
118             V2             V11   0.5463642
137             V8             V12   0.5434786
```

This tells us that the pair of variables with the highest linear correlation coefficient are V7 and V8 (correlation = 0.86 approximately).

Standardising Variables

If you want to compare different variables that have different units, are very different variances, it is a good idea to first standardise the variables.

For example, we found above that the concentrations of the 13 chemicals in the wine samples show a wide range of standard deviations, from 0.1244533 for V9 (variance 0.01548862) to 314.9074743 for V14 (variance 99166.72). This is a range of approximately 6,402,554-fold in the variances.

As a result, it is not a good idea to use the unstandardised chemical concentrations as the input for a principal component analysis (PCA, see below) of the wine samples, as if you did that, the first principal component would be dominated by the variables which show the largest variances, such as V14.

Thus, it would be a better idea to first standardise the variables so that they all have variance 1 and mean 0, and to then carry out the principal component analysis on the standardised data. This would allow us to find the principal components that provide the best low-dimensional representation of the variation in the original data, without being overly biased by those variables that show the most variance in the original data.

For example, to standardise the concentrations of the 13 chemicals in the wine samples, we type:

```
> standardisedconcentrations <- as.data.frame(scale(wine[2:14]))
```

We can check that each of the standardised variables stored in "standardisedconcentrations" has a mean of 0 and a standard deviation of 1 by typing:

```
> sapply(standardisedconcentrations,mean)
      V2            V3            V4            V5            V6            V7
-8.591766e-16 -6.776446e-17  8.045176e-16 -7.720494e-17 -4.073935e-17 -1.395560e-17
      V8            V9           V10           V11           V12           V13
6.958263e-17 -1.042186e-16 -1.221369e-16  3.649376e-17  2.093741e-16  3.003459e-16
     V14
-1.034429e-16
> sapply(standardisedconcentrations,sd)
V2  V3  V4  V5  V6  V7  V8  V9 V10 V11 V12 V13 V14
1   1   1   1   1   1   1   1   1   1   1   1   1
```

We see that the means of the standardised variables are all very tiny numbers and so are essentially equal to 0, and the standard deviations of the standardised variables are all equal to 1.

Principal Component Analysis

The purpose of principal component analysis is to find the best low-dimensional representation of the variation in a multivariate data set. For example, in the case of the wine data set, we have 13 chemical concentrations describing wine samples from three different cultivars. We can carry out a principal component analysis to investigate whether we can capture most of the variation between samples using a smaller number of new variables (principal components), where each of these new variables is a linear combination of all or some of the 13 chemical concentrations.

To carry out a principal component analysis (PCA) on a multivariate data set, the first step is often to standardise the variables under study using the "scale()" function (see above). This is necessary if the input variables have very different variances, which is true in this case as the concentrations of the 13 chemicals have very different variances (see above).

Once you have standardised your variables, you can carry out a principal component analysis using the "prcomp()" function in R.

You can get a summary of the principal component analysis results using the “summary()” function on the output of “prcomp()”:

```
> summary(wine.pca)
Importance of components:
                          PC1   PC2   PC3    PC4    PC5    PC6    PC7    PC8    PC9   PC10
Standard deviation     2.169 1.580 1.203 0.9586 0.9237 0.8010 0.7423 0.5903 0.5375 0.5009
Proportion of Variance 0.362 0.192 0.111 0.0707 0.0656 0.0494 0.0424 0.0268 0.0222 0.0193
Cumulative Proportion  0.362 0.554 0.665 0.7360 0.8016 0.8510 0.8934 0.9202 0.9424 0.9617
                          PC11   PC12    PC13
Standard deviation     0.4752 0.4108 0.32152
Proportion of Variance 0.0174 0.0130 0.00795
Cumulative Proportion  0.9791 0.9920 1.00000
```

This gives us the standard deviation of each component, and the proportion of variance explained by each component. The standard deviation of the components is stored in a named element called “sdev” of the output variable made by “prcomp”:

```
> wine.pca$sdev
[1] 2.1692972 1.5801816 1.2025273 0.9586313 0.9237035 0.8010350 0.7423128 0.5903367
[9] 0.5374755 0.5009017 0.4751722 0.4108165 0.3215244
```

The total variance explained by the components is the sum of the variances of the components:

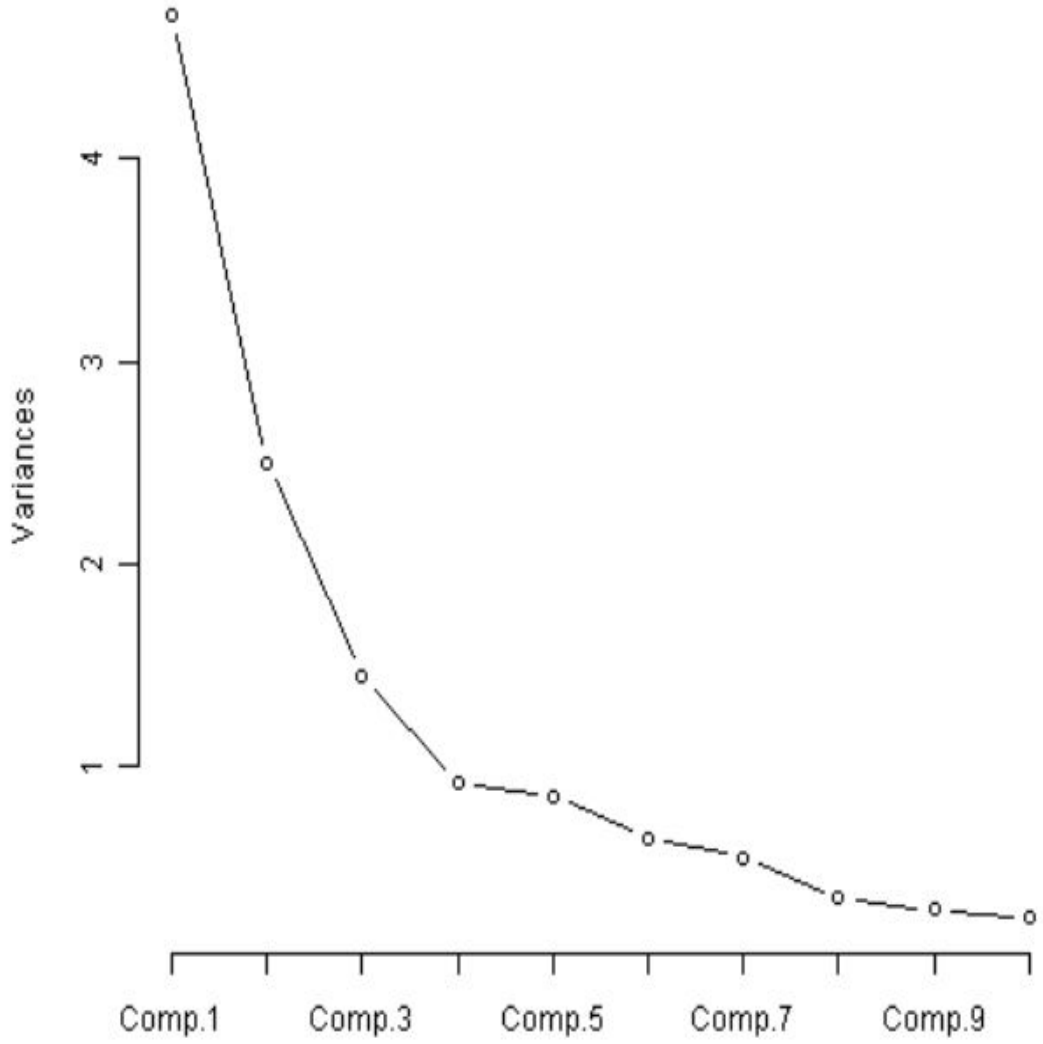
```
> sum((wine.pca$sdev)^2)
  [1] 13
```

In this case, we see that the total variance is 13, which is equal to the number of standardised variables (13 variables). This is because for standardised data, the variance of each standardised variable is 1. The total variance is equal to the sum of the variances of the individual variables, and since the variance of each standardised variable is 1, the total variance should be equal to the number of variables (13 here).

Deciding How Many Principal Components to Retain

In order to decide how many principal components should be retained, it is common to summarise the results of a principal components analysis by making a scree plot, which we can do in R using the "screeplot()" function:

```
> screeplot(wine.pca, type="lines")
```



The most obvious change in slope in the scree plot occurs at component 4, which is the “elbow” of the scree plot. Therefore, it cound be argued based on the basis of the scree plot that the first three components should be retained.

Another way of deciding how many components to retain is to use Kaiser's criterion: that we should only retain principal components for which the variance is above 1 (when principal component analysis was applied to standardised data). We can check this by finding the variance of each of the principal components:

```
> (wine.pca$sdev)^2
  [1] 4.7058503 2.4969737 1.4460720 0.9189739 0.8532282 0.6416570 0.5510283 0.3484974
  [9] 0.2888799 0.2509025 0.2257886 0.1687702 0.1033779
```

We see that the variance is above 1 for principal components 1, 2, and 3 (which have variances 4.71, 2.50, and 1.45, respectively). Therefore, using Kaiser's criterion, we would retain the first three principal components.

A third way to decide how many principal components to retain is to decide to keep the number of components required to explain at least some minimum amount of the total variance. For example, if it is important to explain at least 80% of the variance, we would retain the first five principal components, as we can see from the output of "summary(wine.pca)" that the first five principal components explain 80.2% of the variance (while the first four components explain just 73.6%, so are not sufficient).

Loadings in PCA

Principal Component Analysis (PCA) is a widely used technique for dimensionality reduction and data simplification. One of the core outputs of PCA is the derivation of principal components. These components are linear combinations of the original features or variables in your dataset. The coefficients, or **weights, assigned to** these original variables within these **linear combinations** are termed **loadings**.

Definition: Loadings offer valuable insights into the nature of the principal components. Specifically, they tell us how much each original variable contributes to the respective principal component.

Interpretation: The magnitude of a loading points to the importance or relevance of its corresponding original variable to that component. A larger absolute value suggests that the variable significantly influences the component. Furthermore, the sign of the loading — whether positive or negative — denotes the directionality of this influence. A positive sign implies that the variable and the principal component are positively related, moving in the same direction. Conversely, a negative sign indicates an inverse relationship, where one increases as the other decreases. your dataset's original features or variables

Matrix Representation: In the mathematical underpinnings of PCA, loadings are connected with the eigenvectors of the covariance or correlation matrix of the data. These eigenvectors represent the loadings for the resultant principal components. When visualized in matrix form, each column corresponds to a specific principal component, and each row aligns with an original variable from the dataset.

Orthogonality: An intriguing aspect of PCA is the orthogonality of the principal components, meaning they remain uncorrelated with each other. Consequently, the loadings associated with different principal components are also orthogonal.

Scaling: The choice of scaling before applying PCA is another critical aspect. If PCA performed on a correlation matrix, it means that each variable was standardized to have a mean of 0 and variance of 1 before the analysis. In this scenario, the relationships between variables are evaluated based on their correlations. On the other hand, if PCA is performed on the covariance matrix, the relationships are interpreted based on covariances. In the latter case, variables with larger variances will naturally have more substantial loadings.

Contribution to Variance: A valuable insight from loadings is their contribution to variance. By squaring the value of a loading, one can ascertain the proportion of the variance of a variable captured by a principal component. For instance, if a loading value for a specific variable on the first principal component is 0.8, this indicates that 64% (which is 0.8 squared) of the variance of that variable is encapsulated by the first principal component.

In summary, loadings in PCA provide insights into how the original variables are combined to create each principal component, helping to interpret the nature and meaning of the principal components in the context of the original data.

Loadings for the Principal Components

The loadings for the principal components are stored in a named element "rotation" of the variable returned by "prcomp()". This contains a matrix with the loadings of each principal component, where the first column in the matrix contains the loadings for the first principal component, the second column contains the loadings for the second principal component, and so on.

Therefore, to obtain the loadings for the first principal component in our analysis of the 13 chemical concentrations in wine samples, we type:

```
> wine.pca$rotation[,1]
      V2           V3           V4           V5           V6           V7
-0.144329395  0.245187580  0.002051061  0.239320405 -0.141992042 -0.394660845
      V8           V9          V10          V11          V12          V13
-0.422934297  0.298533103 -0.313429488  0.088616705 -0.296714564 -0.376167411
     V14
-0.286752227
```

This means that the first principal component is a linear combination of the variables: -0.144*Z2 + 0.245*Z3 + 0.002*Z4 + 0.239*Z5 - 0.142*Z6 - 0.395*Z7 - 0.423*Z8 + 0.299*Z9 -0.313*Z10 + 0.089*Z11 - 0.297*Z12 - 0.376*Z13 - 0.287*Z14, where Z2, Z3, Z4...Z14 are the standardised versions of the variables V2, V3, V4...V14 (that each have mean of 0 and variance of 1).

Note that the square of the loadings sum to 1, as this is a constraint used in calculating the loadings:

```
> sum((wine.pca$rotation[,1])^2)
[1] 1
```

To calculate the values of the first principal component, we can define our own function to calculate a principal component given the loadings and the input variables' values:

We can then use the function to calculate the values of the first principal component for each sample in our wine data:

```
> calcpc(standardisedconcentrations, wine.pca$rotation[,1])
  [1] -3.30742097 -2.20324981 -2.50966069 -3.74649719 -1.00607049 -3.04167373 -2.44220051
  [8] -2.05364379 -2.50381135 -2.74588238 -3.46994837 -1.74981688 -2.10751729 -3.44842921
  [15] -4.30065228 -2.29870383 -2.16584568 -1.89362947 -3.53202167 -2.07865856 -3.11561376
  [22] -1.08351361 -2.52809263 -1.64036108 -1.75662066 -0.98729406 -1.77028387 -1.23194878
  [29] -2.18225047 -2.24976267 -2.49318704 -2.66987964 -1.62399801 -1.89733870 -1.40642118
  [36] -1.89847087 -1.38096669 -1.11905070 -1.49796891 -2.52268490 -2.58081526 -0.66660159
  ...
```

In fact, the values of the first principal component are stored in the variable wine.pca$x[,1] that was returned by the "prcomp()" function, so we can compare those values to the ones that we calculated, and they should agree:

```
> wine.pca$x[,1]
  [1] -3.30742097 -2.20324981 -2.50966069 -3.74649719 -1.00607049 -3.04167373 -2.44220051
  [8] -2.05364379 -2.50381135 -2.74588238 -3.46994837 -1.74981688 -2.10751729 -3.44842921
  [15] -4.30065228 -2.29870383 -2.16584568 -1.89362947 -3.53202167 -2.07865856 -3.11561376
  [22] -1.08351361 -2.52809263 -1.64036108 -1.75662066 -0.98729406 -1.77028387 -1.23194878
  [29] -2.18225047 -2.24976267 -2.49318704 -2.66987964 -1.62399801 -1.89733870 -1.40642118
  [36] -1.89847087 -1.38096669 -1.11905070 -1.49796891 -2.52268490 -2.58081526 -0.66660159
  ...
```

We see that they do agree.

The first principal component has highest (in absolute value) loadings for V8 (−0.423), V7 (−0.395), V13 (−0.376), V10 (−0.313), V12 (−0.297), V14 (−0.287), V9 (0.299), V3 (0.245), and V5 (0.239). The loadings for V8, V7, V13, V10, V12 and V14 are negative, while those for V9, V3, and V5 are positive. Therefore, an interpretation of the first principal component is that it represents a contrast between the concentrations of V8, V7, V13, V10, V12, and V14, and the concentrations of V9, V3 and V5.

Similarly, we can obtain the loadings for the second principal component by typing:

```
> wine.pca$rotation[,2]
      V2           V3           V4           V5           V6           V7
 0.483651548  0.224930935  0.316068814 -0.010590502  0.299634003  0.065039512
      V8           V9          V10          V11          V12          V13
-0.003359812  0.028779488  0.039301722  0.529995672 -0.279235148 -0.164496193
     V14
 0.364902832
```

This means that the second principal component is a linear combination of the variables: 0.484*Z2 + 0.225*Z3 + 0.316*Z4 − 0.011*Z5 + 0.300*Z6 + 0.065*Z7 − 0.003*Z8 + 0.029*Z9 + 0.039*Z10 + 0.530*Z11 − 0.279*Z12 − 0.164*Z13 + 0.365*Z14, where Z1, Z2, Z3...Z14 are the standardised versions of variables V2, V3, ... V14 that each have mean 0 and variance 1.

Note that the square of the loadings sum to 1, as above:

```
> sum((wine.pca$rotation[,2])^2)
 [1] 1
```

The second principal component has highest loadings for V11 (0.530), V2 (0.484), V14 (0.365), V4 (0.316), V6 (0.300), V12 (-0.279), and V3 (0.225). The loadings for V11, V2, V14, V4, V6 and V3 are positive, while the loading for V12 is negative. Therefore, an interpretation of the second principal component is that it represents a contrast between the concentrations of V11, V2, V14, V4, V6 and V3, and the concentration of V12. Note that the loadings for V11 (0.530) and V2 (0.484) are the largest, so the contrast is mainly between the concentrations of V11 and V2, and the concentration of V12.

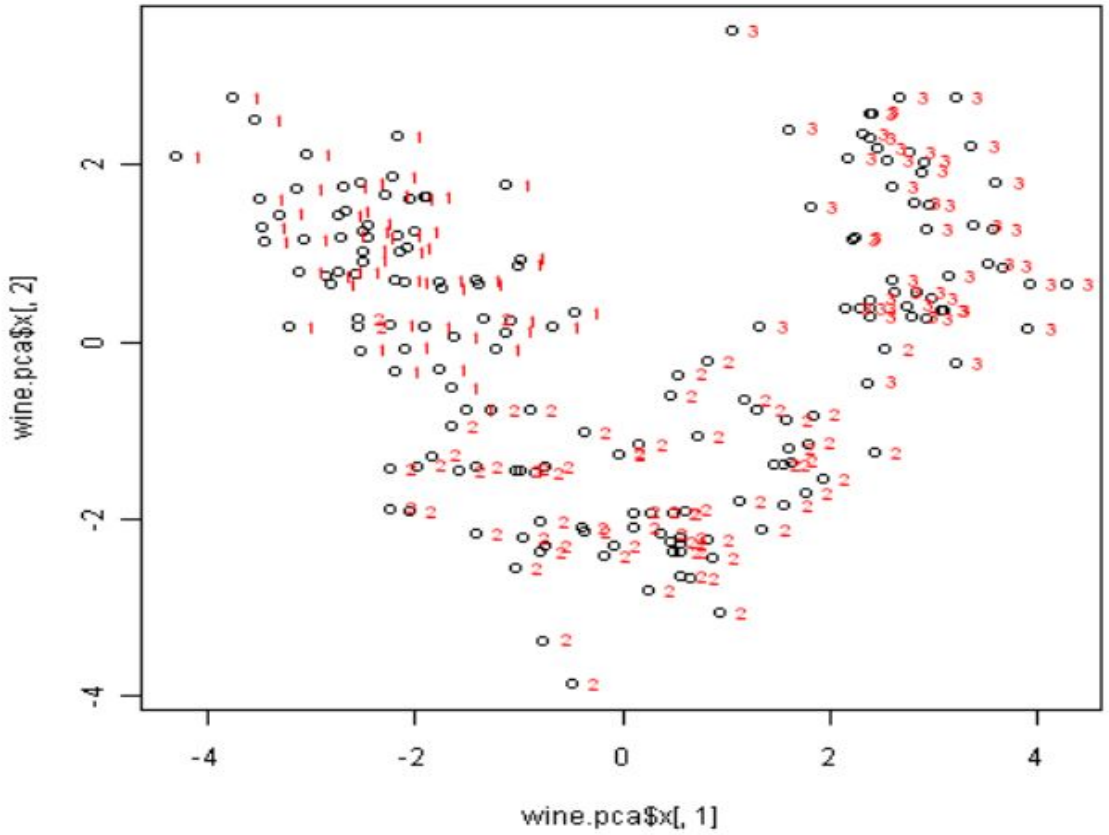
Scatterplots of the Principal Components

The values of the principal components are stored in a named element "x" of the variable returned by "prcomp()". This contains a matrix with the principal components, where the first column in the matrix contains the first principal component, the second column the second component, and so on.

Thus, in our example, "wine.pca$x[,1]" contains the first principal component, and "wine.pca$x[,2]" contains the second principal component.

We can make a scatterplot of the first two principal components, and label the data points with the cultivar that the wine samples come from, by typing:

```
> plot(wine.pca$x[,1],wine.pca$x[,2]) # make a scatterplot
> text(wine.pca$x[,1],wine.pca$x[,2], wine$V1, cex=0.7, pos=4, col="red") # add labels
```



The scatterplot shows the first principal component on the x-axis, and the second principal component on the y-axis. We can see from the scatterplot that wine samples of cultivar 1 have much lower values of the first principal component than wine samples of cultivar 3. Therefore, the first principal component separates wine samples of cultivars 1 from those of cultivar 3.

We can also see that wine samples of cultivar 2 have much higher values of the second principal component than wine samples of cultivars 1 and 3. Therefore, the second principal component separates samples of cultivar 2 from samples of cultivars 1 and 3.

Therefore, the first two principal components are reasonably useful for distinguishing wine samples of the three different cultivars.

Above, we interpreted the first principal component as a contrast between the concentrations of V8, V7, V13, V10, V12, and V14, and the concentrations of V9, V3 and V5. We can check whether this makes sense in terms of the concentrations of these chemicals in the different cultivars, by printing out the means of the standardised concentration variables in each cultivar, using the "printMeanAndSdByGroup()" function (see above):

```
> printMeanAndSdByGroup(standardisedconcentrations,wine[1])
[1] "Means:"
  V1       V2       V3       V4       V5        V6        V7        V8        V9       V10       V11       V12       V13       V14
1  1 0.9166093 -0.2915199 0.3246886 -0.7359212 0.46192317 0.87090552 0.95419225 -0.57735640 0.5388633 0.2028288 0.4575567 0.7691811 1.1711967
2  2 -0.8892116 -0.3613424 -0.4437061 0.2225094 -0.36354162 -0.05790375 0.05163434 0.01452785 0.0688079 -0.8503999 0.4323908 0.2446043 -0.7220731
3  3 0.1886265 0.8928122 0.2572190 0.5754413 -0.03004191 -0.98483874 -1.24923710 0.68817813 -0.7641311 1.0085728 -1.2019916 -1.3072623 -0.3715295
```

Does it make sense that the first principal component can separate cultivar 1 from cultivar 3? In cultivar 1, the mean values of V8 (0.954), V7 (0.871), V13 (0.769), V10 (0.539), V12 (0.458) and V14 (1.171) are very high compared to the mean values of V9 (-0.577), V3 (-0.292) and V5 (-0.736). In cultivar 3, the mean values of V8 (-1.249), V7 (-0.985), V13 (-1.307), V10 (-0.764), V12 (-1.202) and V14 (-0.372) are very low compared to the mean values of V9 (0.688), V3 (0.893) and V5 (0.575). Therefore, it does make sense that principal component 1 is a contrast between the concentrations of V8, V7, V13, V10, V12, and V14, and the concentrations of V9, V3 and V5; and that principal component 1 can separate cultivar 1 from cultivar 3.

Above, we intepreted the second principal component as a contrast between the concentrations of V11, V2, V14, V4, V6 and V3, and the concentration of V12. In the light of the mean values of these variables in the different cultivars, does it make sense that the second principal component can separate cultivar 2 from cultivars 1 and 3? In cultivar 1, the mean values of V11 (0.203), V2 (0.917), V14 (1.171), V4 (0.325), V6 (0.462) and V3 (-0.292) are not very different from the mean value of V12 (0.458). In cultivar 3, the mean values of V11 (1.009), V2 (0.189), V14 (-0.372), V4 (0.257), V6 (-0.030) and V3 (0.893) are also not very different from the mean value of V12 (-1.202). In contrast, in cultivar 2, the mean values of V11 (-0.850), V2 (-0.889), V14 (-0.722), V4 (-0.444), V6 (-0.364) and V3 (-0.361) are much less than the mean value of V12 (0.432). Therefore, it makes sense that principal component is a contrast between the concentrations of V11, V2, V14, V4, V6 and V3, and the concentration of V12; and that principal component 2 can separate cultivar 2 from cultivars 1 and 3.

Discriminant and Classification Analysis

Two Group Discrimination and Classification

For a two group discriminant analysis, we assume that we have two independent samples from two multivariate normal populations with common covariance matrix Σ and unknown means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. Thus, $\mathbf{y}_{ij} \sim IN_p(\boldsymbol{\mu}_i, \Sigma)$ where $i = 1, 2;\ j = 1, 2, \ldots, n_i$ and each observation $\mathbf{y}_{ij}$ is a $p \times 1$ vector of random variables. Fisher's two group linear discriminant function L is the linear combination of the variables that provides for the maximum separation between the groups

$$L = \mathbf{a}'\mathbf{y} = \sum_{j=1}^{p} a_j y_j \tag{7.2.1}$$

a vector $\mathbf{a} \equiv \mathbf{a}_s$ that provides maximum separation between the discriminant scores $L_{ij} = \mathbf{a}_s \cdot \mathbf{y}_{ij}$ for the two group situation is the vector

$$\mathbf{a}_s = \mathbf{S}^{-1}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2) \tag{7.2.2}$$

where $\overline{\mathbf{y}}_i$ is the sample mean for the observations in group i ($i = 1, 2$) and $\mathbf{S}$ is the unbiased estimate of Σ. Evaluating the discriminant scores at the group mean vectors $\overline{\mathbf{y}}_i$, the difference in the mean discriminant scores is exactly Mahalanobis' D^2 statistic

$$\begin{aligned} D^2 &= \overline{L}_1 - \overline{L}_2 = \mathbf{a}_s'\overline{\mathbf{y}}_1 - \mathbf{a}_s'\overline{\mathbf{y}}_2 \\ &= \mathbf{a}_s' \left(\overline{\mathbf{y}}_1 - \overline{\mathbf{y}}_2\right) \\ &= \left(\overline{\mathbf{y}}_1 - \overline{\mathbf{y}}_2\right)' \mathbf{S}^{-1} \left(\overline{\mathbf{y}}_1 - \overline{\mathbf{y}}_2\right) \end{aligned} \tag{7.2.3}$$

Hence, if $T^2 = (n_1 n_2 / n_1 + n_2)\, D^2$ is significant then we have good separation of the sample group centroids for the two populations. It is also easily established that the square of the univariate student t^2 statistic using the mean discriminant scores for two groups with $\mathbf{a} \equiv \mathbf{a}_s$ is equal to T^2 since

$$t^2 = \frac{\left(\overline{L}_1 - \overline{L}_2\right)^2}{s_L^2 \left(\frac{1}{n_1} + \frac{1}{n_2}\right)} = \left(\frac{n_1 n_2}{n_1 + n_2}\right) \left(\overline{\mathbf{y}}_1 - \overline{\mathbf{y}}_2\right)' \mathbf{S}^{-1} \left(\overline{\mathbf{y}}_1 - \overline{\mathbf{y}}_2\right) = T^2 \tag{7.2.4}$$

where s_L^2 is the sample estimate of the common population variance for discriminant scores. Thus, a simple t test on the discriminant scores is equivalent to calculating Hotelling's T^2 statistic for testing for the difference in the mean vectors for two groups.

Letting the discriminant scores $L_{ij} \equiv y$ represent the dependent variable and the dummy independent variables $x_1 = -1$ and $x_2 = 1$ for the two groups, we may fit a regression equation $y = \alpha + \beta x + e$ to the discriminant scores. Testing $H : \beta = 0$ in the regression problem is equivalent to testing $\rho = 0$ using the t^2 statistic

$$t^2 = \frac{(n-2)\,r^2}{1-r^2} \tag{7.2.5}$$

Equating t^2 in (7.2.5) with t^2 in (7.2.4), observe that

$$\begin{aligned} \frac{n_1 n_2 D^2}{n_1 + n_2} &= \frac{(n-2)\,r^2}{1-r^2} \\ D^2 &= \frac{(n_1 - n_2)\,(n-2)\,r^2}{(n_1 n_2)\left(1-r^2\right)} \\ r^2 &= \frac{n_1 n_2 D^2}{(n_1 + n_2)\,(n-2) + n_1 n_2 D^2} \end{aligned} \tag{7.2.6}$$

where $n = n_1 + n_2$. Performing the "pseudoregression" on the "dummy" variables

$$c_1 = \frac{n_2}{n_1 + n_2} \text{ and } c_2 = c_1 - 1 = -\frac{n_1}{n_1 + n_2} \tag{7.2.7}$$

for groups 1 and 2, respectively (Fisher's codes with mean zero), one may regress the original variables on the dummy variables. Then r^2 in (7.2.6) is replaced by R_p^2, the squared multiple correlation or coefficient of determination found by regressing the independent variable c_i on the dependent variables $y_1, y_2, \ldots, y_p$, Fisher (1936). Letting $\mathbf{b}' = [b_1, b_2, \ldots, b_p]$ be the vector of estimated regression coefficients (ignoring the intercept), the vector $\mathbf{b}$ is seen to be proportional to $\mathbf{a}_s$

$$\begin{aligned} \mathbf{b} &= \left[\frac{n_1 n_2}{(n_1 + n_2)(n_1 + n_2 - 2) + n_1 n_2 D_p^2} \right] \mathbf{a}_s \\ \mathbf{a}_s &= \left[\frac{(n_1 + n_2)(n_1 + n_2 - 2)}{n_1 n_2} + D_p^2 \right] \mathbf{b} \end{aligned} \tag{7.2.8}$$

and $R_p^2 = (\overline{\mathbf{y}}_1 - \overline{\mathbf{y}}_2)' \mathbf{b}$.

Linear Discriminant Analysis

The purpose of principal component analysis is to find the best low-dimensional representation of the variation in a multivariate data set. For example, in the wine data set, we have 13 chemical concentrations describing wine samples from three cultivars. By carrying out a principal component analysis, we found that most of the variation in the chemical concentrations between the samples can be captured using the first two principal components, where each of the principal components is a particular linear combination of the 13 chemical concentrations.

The purpose of linear discriminant analysis (LDA) is to find the linear combinations of the original variables (the 13 chemical concentrations here) that gives the best possible separation between the groups (wine cultivars here) in our data set. Linear discriminant analysis is also known as "canonical discriminant analysis", or simply "discriminant analysis".

If we want to separate the wines by cultivar, the wines come from three different cultivars, so the number of groups (G) is 3, and the number of variables is 13 (13 chemicals' concentrations; p = 13). The maximum number of useful discriminant functions that can separate the wines by cultivar is the minimum of G-1 and p, and so in this case it is the minimum of 2 and 13, which is 2. Thus, we can find at most 2 useful discriminant functions to separate the wines by cultivar, using the 13 chemical concentration variables.

You can carry out a linear discriminant analysis using the "lda()" function from the R "MASS" package. To use this function, we first need to install the "MASS" R package

For example, to carry out a linear discriminant analysis using the 13 chemical concentrations in the wine samples, we type:

```
> library("MASS")                                                    # Load the MASS package
> wine.lda <- lda(wine$V1 ~ wine$V2 + wine$V3 + wine$V4 + wine$V5 + wine$V6 + wine$V7 +
                  wine$V8 + wine$V9 + wine$V10 + wine$V11 + wine$V12 + wine$V13 +
                  wine$V14)
```

Loadings for the Discriminant Functions

To get the values of the loadings of the discriminant functions for the wine data, we can type:

```
> wine.lda
  Coefficients of linear discriminants:
              LD1           LD2
  wine$V2  -0.403399781  0.8717930699
  wine$V3   0.165254596  0.3053797325
  wine$V4  -0.369075256  2.3458497486
  wine$V5   0.154797889 -0.1463807654
  wine$V6  -0.002163496 -0.0004627565
  wine$V7   0.618052068 -0.0322128171
  wine$V8  -1.661191235 -0.4919980543
  wine$V9  -1.495818440 -1.6309537953
  wine$V10  0.134092628 -0.3070875776
  wine$V11  0.355055710  0.2532306865
  wine$V12 -0.818036073 -1.5156344987
  wine$V13 -1.157559376  0.0511839665
  wine$V14 -0.002691206  0.0028529846
```

This means that the first discriminant function is a linear combination of the variables: -0.403*V2 + 0.165*V3 - 0.369*V4 + 0.155*V5 - 0.002*V6 + 0.618*V7 - 1.661*V8 - 1.496*V9 + 0.134*V10 + 0.355*V11 - 0.818*V12 - 1.158*V13 - 0.003*V14, where V2, V3, ... V14 are the concentrations of the 14 chemicals found in the wine samples. For convenience, the value for each discriminant function (eg. the first discriminant function) are scaled so that their mean value is zero (see below).

Note that these loadings are calculated so that the within-group variance of each discriminant function for each group (cultivar) is equal to 1, as will be demonstrated below.

These scalings are also stored in the named element "scaling" of the variable returned by the lda() function. This element contains a matrix, in which the first column contains the loadings for the first discriminant function, the second column contains the loadings for the second discriminant function and so on. For example, to extract the loadings for the first discriminant function, we can type:

```
> wine.lda$scaling[,1]
   wine$V2      wine$V3      wine$V4      wine$V5      wine$V6      wine$V7
 -0.403399781  0.165254596 -0.369075256  0.154797889 -0.002163496  0.618052068
   wine$V8      wine$V9     wine$V10     wine$V11     wine$V12     wine$V13
 -1.661191235 -1.495818440  0.134092628  0.355055710 -0.818036073 -1.157559376
  wine$V14
 -0.002691206
```

The function calclda() simply calculates the value of a discriminant function for each sample in the data set, for example, for the first disriminant function, for each sample we calculate the value using the equation -0.403*V2 - 0.165*V3 - 0.369*V4 + 0.155*V5 - 0.002*V6 + 0.618*V7 - 1.661*V8 - 1.496*V9 + 0.134*V10 + 0.355*V11 - 0.818*V12 - 1.158*V13 - 0.003*V14. Furthermore, the "scale()" command is used within the calclda() function in order to standardise the value of a discriminant function (eg. the first discriminant function) so that its mean value (over all the wine samples) is 0.

We can use the function calclda() to calculate the values of the first discriminant function for each sample in our wine data:

```
> calclda(wine[2:14], wine.lda$scaling[,1])
  [1] -4.70024401 -4.30195811 -3.42071952 -4.20575366 -1.50998168 -4.51868934
  [7] -4.52737794 -4.14834781 -3.86082876 -3.36662444 -4.80587907 -3.42807646
  [13] -3.66610246 -5.58824635 -5.50131449 -3.18475189 -3.28936988 -2.99809262
  [19] -5.24640372 -3.13653106 -3.57747791 -1.69077135 -4.83515033 -3.09588961
  [25] -3.32164716 -2.14482223 -3.98242850 -2.68591432 -3.56309464 -3.17301573
  [31] -2.99626797 -3.56866244 -3.38506383 -3.52753750 -2.85190852 -2.79411996
  ...
```

In fact, the values of the first linear discriminant function can be calculated using the "predict()" function in R, so we can compare those to the ones that we calculated, and they should agree:

```
> wine.lda.values <- predict(wine.lda, wine[2:14])
> wine.lda.values$x[,1] # contains the values for the first discriminant function
     1           2           3           4           5           6
-4.70024401 -4.30195811 -3.42071952 -4.20575366 -1.50998168 -4.51868934
     7           8           9          10          11          12
-4.52737794 -4.14834781 -3.86082876 -3.36662444 -4.80587907 -3.42807646
    13          14          15          16          17          18
-3.66610246 -5.58824635 -5.50131449 -3.18475189 -3.28936988 -2.99809262
    19          20          21          22          23          24
-5.24640372 -3.13653106 -3.57747791 -1.69077135 -4.83515033 -3.09588961
    25          26          27          28          29          30
-3.32164716 -2.14482223 -3.98242850 -2.68591432 -3.56309464 -3.17301573
    31          32          33          34          35          36
-2.99626797 -3.56866244 -3.38506383 -3.52753750 -2.85190852 -2.79411996
...
```

We see that they do agree.

It doesn't matter whether the input variables for linear discriminant analysis are standardised or not, unlike for principal components analysis in which it is often necessary to standardise the input variables. However, using standardised variables in linear discriminant analysis makes it easier to interpret the loadings in a linear discriminant function.

In linear discriminant analysis, the standardised version of an input variable is defined so that it has mean zero and within-groups variance of 1. Thus, we can calculate the "group-standardised" variable by subtracting the mean from each value of the variable, and dividing by the within-groups standard deviation. To calculate the group-standardised version of a set of variables, we can use the function "groupStandardise()" below:

For example, we can use the "groupStandardise()" function to calculate the group-standardised versions of the chemical concentrations in wine samples:

```
> groupstandardisedconcentrations <- groupStandardise(wine[2:14], wine[1])
```

We can then use the lda() function to perform linear disriminant analysis on the group-standardised variables:

```
> wine.lda2 <- lda(wine$V1 ~ groupstandardisedconcentrations$V2 + groupstandardisedconcentrations$V3 +
                           groupstandardisedconcentrations$V4 + groupstandardisedconcentrations$V5 +
                           groupstandardisedconcentrations$V6 + groupstandardisedconcentrations$V7 +
                           groupstandardisedconcentrations$V8 + groupstandardisedconcentrations$V9 +
                           groupstandardisedconcentrations$V10 + groupstandardisedconcentrations$V11
                           groupstandardisedconcentrations$V12 + groupstandardisedconcentrations$V13
                           groupstandardisedconcentrations$V14)
> wine.lda2
  Coefficients of linear discriminants:
                                          LD1          LD2
  groupstandardisedconcentrations$V2  -0.20650463  0.446280119
  groupstandardisedconcentrations$V3   0.15568586  0.287697336
  groupstandardisedconcentrations$V4  -0.09486893  0.602988809
  groupstandardisedconcentrations$V5   0.43802089 -0.414203541
  groupstandardisedconcentrations$V6  -0.02907934 -0.006219863
  groupstandardisedconcentrations$V7   0.27030186 -0.014088108
  groupstandardisedconcentrations$V8  -0.87067265 -0.257868714
  groupstandardisedconcentrations$V9  -0.16325474 -0.178003512
  groupstandardisedconcentrations$V10  0.06653116 -0.152364015
  groupstandardisedconcentrations$V11  0.53670086  0.382782544
  groupstandardisedconcentrations$V12 -0.12801061 -0.237174509
  groupstandardisedconcentrations$V13 -0.46414916  0.020523349
  groupstandardisedconcentrations$V14 -0.46385409  0.491738050
```

It makes sense to interpret the loadings calculated using the group-standardised variables rather than the loadings for the original (unstandardised) variables.

In the first discriminant function calculated for the group-standardised variables, the largest loadings (in absolute) value are given to V8 (-0.871), V11 (0.537), V13 (-0.464), V14 (-0.464), and V5 (0.438). The loadings for V8, V13 and V14 are negative, while those for V11 and V5 are positive. Therefore, the discriminant function seems to represent a contrast between the concentrations of V8, V13 and V14, and the concentrations of V11 and V5.

We saw above that the individual variables which gave the greatest separations between the groups were V8 (separation 233.93), V14 (207.92), V13 (189.97), V2 (135.08) and V11 (120.66). These were mostly the same variables that had the largest loadings in the linear discriminant function (loading for V8: -0.871, for V14: -0.464, for V13: -0.464, for V11: 0.537).

We found above that variables V8 and V11 have a negative between-groups covariance (-60.41) and a positive within-groups covariance (0.29). When the between-groups covariance and within-groups covariance for two variables have opposite signs, it indicates that a better separation between groups can be obtained by using a linear combination of those two variables than by using either variable on its own.

Thus, given that the two variables V8 and V11 have between-groups and within-groups covariances of opposite signs, and that these are two of the variables that gave the greatest separations between groups when used individually, it is not surprising that these are the two variables that have the largest loadings in the first discriminant function.

Note that although the loadings for the group-standardised variables are easier to interpret than the loadings for the unstandardised variables, the values of the discriminant function are the same regardless of whether we standardise the input variables or not. For example, for wine data, we can calculate the value of the first discriminant function calculated using the unstandardised and group-standardised variables by typing:

```
> wine.lda.values <- predict(wine.lda, wine[2:14])
> wine.lda.values$x[,1] # values for the first discriminant function, using the unstandardised data
      1           2           3           4           5           6
 -4.70024401 -4.30195811 -3.42071952 -4.20575366 -1.50998168 -4.51868934
      7           8           9          10          11          12
 -4.52737794 -4.14834781 -3.86082876 -3.36662444 -4.80587907 -3.42807646
     13          14          15          16          17          18
 -3.66610246 -5.58824635 -5.50131449 -3.18475189 -3.28936988 -2.99809262
     19          20          21          22          23          24
 -5.24640372 -3.13653106 -3.57747791 -1.69077135 -4.83515033 -3.09588961
 ...
> wine.lda.values2 <- predict(wine.lda2, groupstandardisedconcentrations)
> wine.lda.values2$x[,1] # values for the first discriminant function, using the standardised data
      1           2           3           4           5           6
 -4.70024401 -4.30195811 -3.42071952 -4.20575366 -1.50998168 -4.51868934
      7           8           9          10          11          12
 -4.52737794 -4.14834781 -3.86082876 -3.36662444 -4.80587907 -3.42807646
     13          14          15          16          17          18
 -3.66610246 -5.58824635 -5.50131449 -3.18475189 -3.28936988 -2.99809262
     19          20          21          22          23          24
 -5.24640372 -3.13653106 -3.57747791 -1.69077135 -4.83515033 -3.09588961
 ...
```

We can see that although the loadings are different for the first discriminant functions calculated using unstandardised and group-standardised data, the actual values of the first discriminant function are the same.

Separation Achieved by the Discriminant Functions

To calculate the separation achieved by each discriminant function, we first need to calculate the value of each discriminant function, by substituting the variables' values into the linear combination for the discriminant function (eg. -0.403*V2 - 0.165*V3 - 0.369*V4 + 0.155*V5 - 0.002*V6 + 0.618*V7 - 1.661*V8 - 1.496*V9 + 0.134*V10 + 0.355*V11 - 0.818*V12 - 1.158*V13 - 0.003*V14 for the first discriminant function), and then scaling the values of the discriminant function so that their mean is zero.

As mentioned above, we can do this using the "predict()" function in R. For example, to calculate the value of the discriminant functions for the wine data, we type:

```
> wine.lda.values <- predict(wine.lda, standardisedconcentrations)
```

The returned variable has a named element "x" which is a matrix containing the linear discriminant functions: the first column of x contains the first discriminant function, the second column of x contains the second discriminant function, and so on (if there are more discriminant functions).

We can therefore calculate the separations achieved by the two linear discriminant functions for the wine data by using the "calcSeparations()" function (see above), which calculates the separation as the ratio of the between-groups variance to the within-groups variance:

As mentioned above, the loadings for each discriminant function are calculated in such a way that the within-group variance (Vw) for each group (wine cultivar here) is equal to 1, as we see in the output from calcSeparations() above.

The output from calcSeparations() tells us that the separation achieved by the first (best) discriminant function is 794.7, and the separation achieved by the second (second best) discriminant function is 361.2.

Therefore, the total separation is the sum of these, which is (794.652200566216+361.241041493455=1155.893) 1155.89, rounded to two decimal places. Therefore, the "percentage separation" achieved by the first discriminant function is (794.652200566216*100/1155.893=) 68.75%, and the percentage separation achieved by the second discriminant function is (361.241041493455*100/1155.893=) 31.25%.

The "proportion of trace" that is printed when you type "wine.lda" (the variable returned by the lda() function) is the percentage separation achieved by each discriminant function. For example, for the wine data we get the same values as just calculated (68.75% and 31.25%):

```
> wine.lda
  Proportion of trace:
    LD1    LD2
  0.6875 0.3125
```

Therefore, the first discriminant function does achieve a good separation between the three groups (three cultivars), but the second discriminant function does improve the separation of the groups by quite a large amount, so is it worth using the second discriminant function as well. Therefore, to achieve a good separation of the groups (cultivars), it is necessary to use both of the first two discriminant functions.

We found above that the largest separation achieved for any of the individual variables (individual chemical concentrations) was 233.9 for V8, which is quite a lot less than 794.7, the separation achieved by the first discriminant function. Therefore, the effect of using more than one variable to calculate the discriminant function is that we can find a discriminant function that achieves a far greater separation between groups than achieved by any one variable alone.

The variable returned by the lda() function also has a named element "svd", which contains the ratio of between- and within-group standard deviations for the linear discriminant variables, that is, the square root of the "separation" value that we calculated using calcSeparations() above. When we calculate the square of the value stored in "svd", we should get the same value as found using calcSeparations():

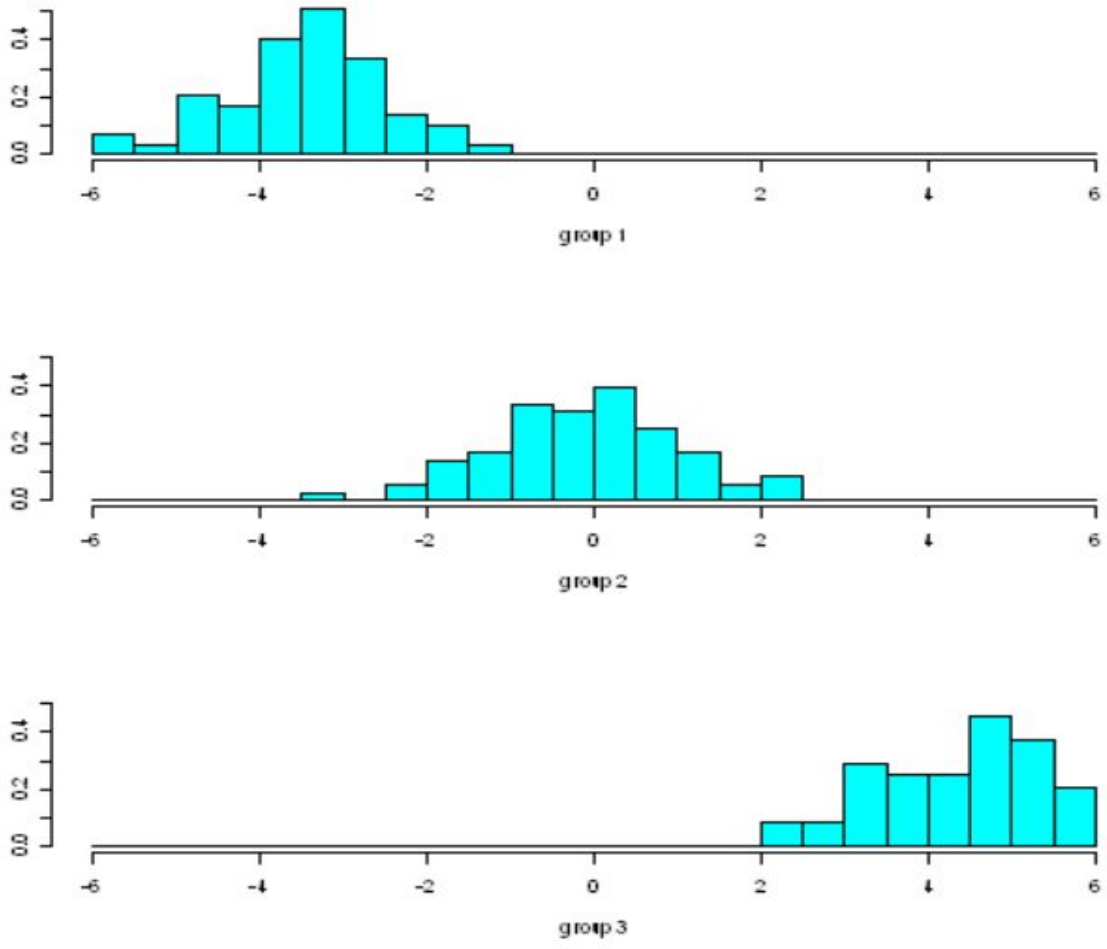
```
> (wine.lda$svd)^2
  [1] 794.6522 361.2410
```

A Stacked Histogram of the LDA Values

A nice way of displaying the results of a linear discriminant analysis (LDA) is to make a stacked histogram of the values of the discriminant function for the samples from different groups (different wine cultivars in our example).

We can do this using the "ldahist()" function in R. For example, to make a stacked histogram of the first discriminant function's values for wine samples of the three different wine cultivars, we type:

```
> ldahist(data = wine.lda.values$x[,1], g=wine$V1)
```



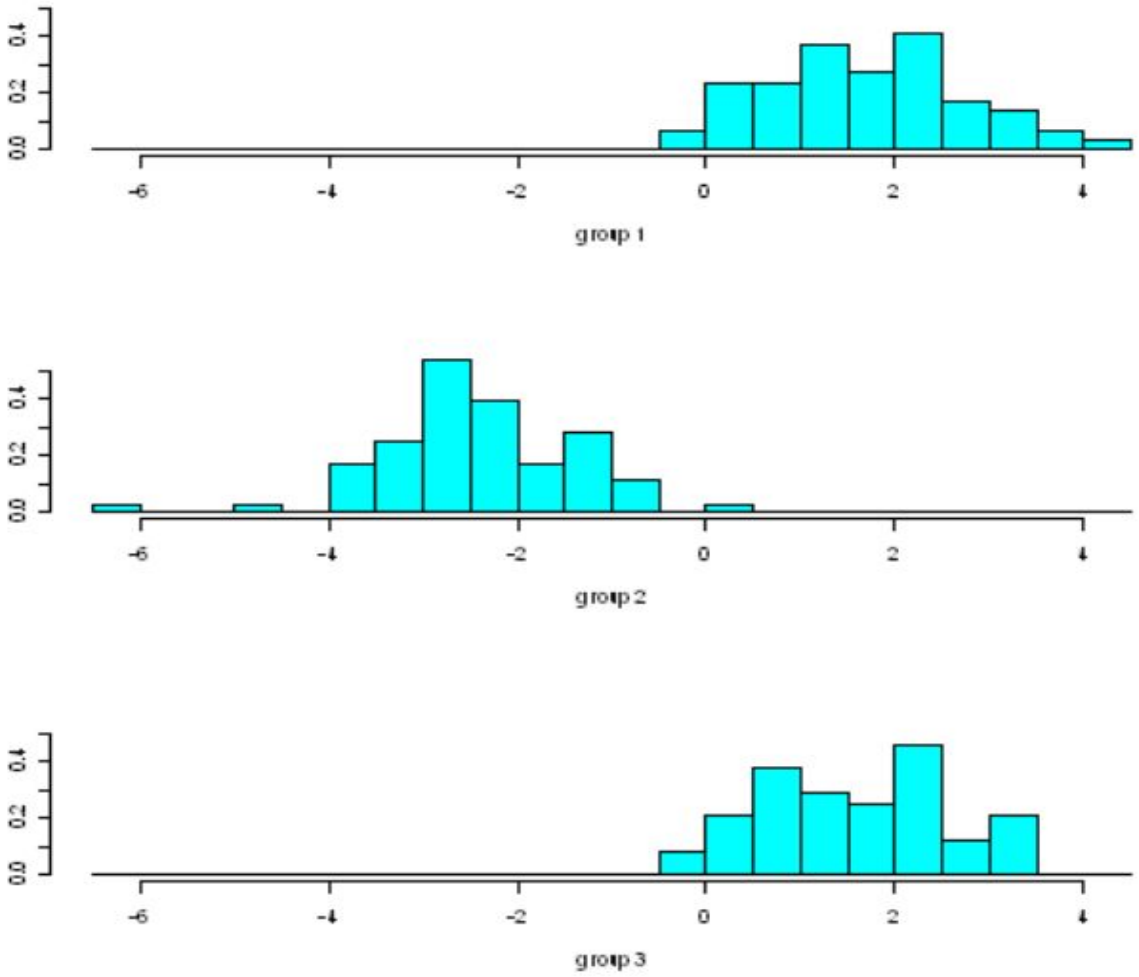
However, the separation achieved by the linear discriminant function on the training set may be an overestimate. To get a more accurate idea of how well the first discriminant function separates the groups, we would need to see a stacked histogram of the values for the three cultivars using some unseen "test set", that is, using a set of data that was not used to calculate the linear discriminant function.

We see that the first discriminant function separates cultivars 1 and 3 very well, but does not separate cultivars 1 and 2, or cultivars 2 and 3, so well.

We therefore investigate whether the second discriminant function separates those cultivars, by making a stacked histogram of the second discriminant function's values:

We therefore investigate whether the second discriminant function separates those cultivars, by making a stacked histogram of the second discriminant function's values:

```
> ldahist(data = wine.lda.values$x[,2], g=wine$V1)
```



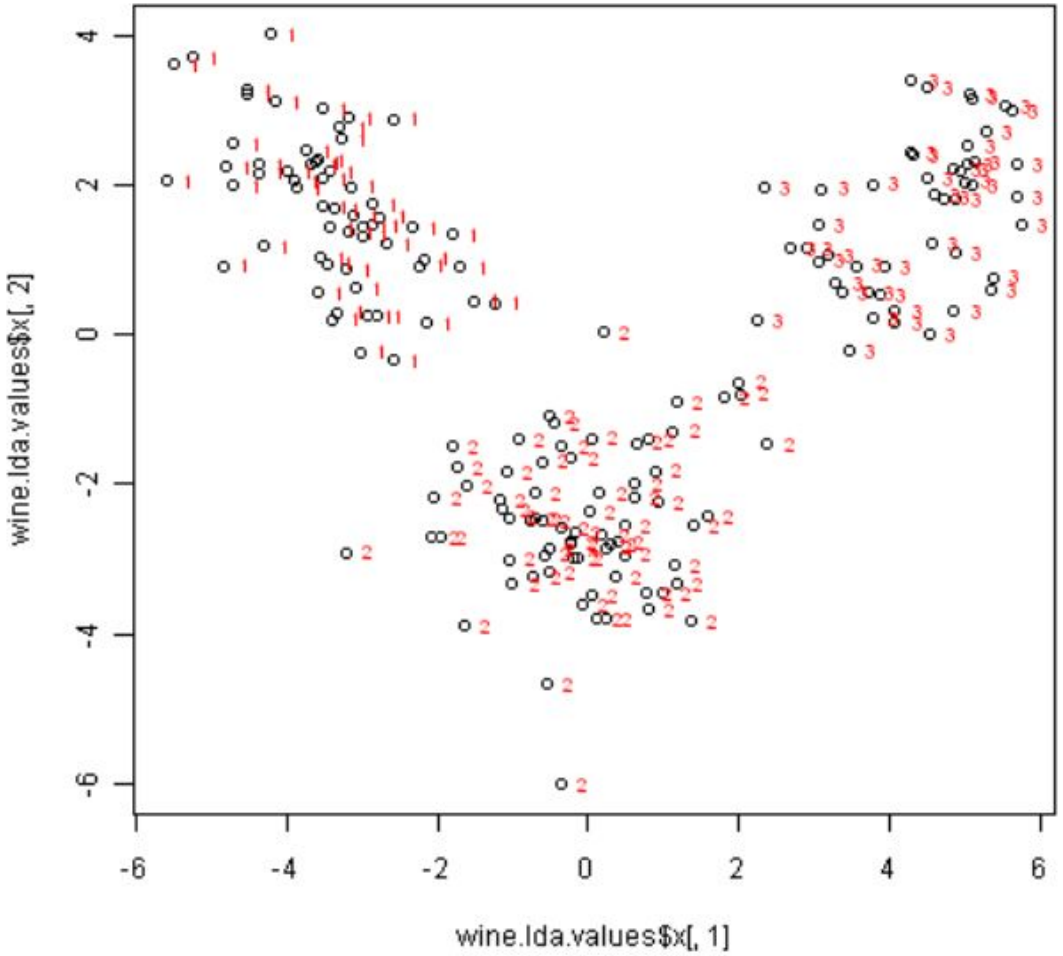
We see that the second discriminant function separates cultivars 1 and 2 quite well, although there is a little overlap in their values. Furthermore, the second discriminant function also separates cultivars 2 and 3 quite well, although again there is a little overlap in their values so it is not perfect.

Thus, we see that two discriminant functions are necessary to separate the cultivars, as was discussed above (see the discussion of percentage separation above).

Scatterplots of the Discriminant Functions

We can obtain a scatterplot of the best two discriminant functions, with the data points labelled by cultivar, by typing:

```
> plot(wine.lda.values$x[,1],wine.lda.values$x[,2]) # make a scatterplot
> text(wine.lda.values$x[,1],wine.lda.values$x[,2],wine$V1,cex=0.7,pos=4,col="red") # add labels
```



From the scatterplot of the first two discriminant functions, we can see that the wines from the three cultivars are well separated in the scatterplot. The first discriminant function (x-axis) separates cultivars 1 and 3 very well, but doesn't not perfectly separate cultivars 1 and 3, or cultivars 2 and 3.

The second discriminant function (y-axis) achieves a fairly good separation of cultivars 1 and 3, and cultivars 2 and 3, although it is not totally perfect.

To achieve a very good separation of the three cultivars, it would be best to use both the first and second discriminant functions together, since the first discriminant function can separate cultivars 1 and 3 very well, and the second discriminant function can separate cultivars 1 and 2, and cultivars 2 and 3, reasonably well.

Allocation Rules and Misclassification Rate

We can calculate the mean values of the discriminant functions for each of the three cultivars using the "printMeanAndSdByGroup()" function (see above):

```
> printMeanAndSdByGroup(wine.lda.values$x,wine[1])
  [1] "Means:"
     V1         LD1       LD2
   1  1 -3.42248851  1.691674
   2  2 -0.07972623 -2.472656
   3  3  4.32473717  1.578120
```

We find that the mean value of the first discriminant function is -3.42248851 for cultivar 1, -0.07972623 for cultivar 2, and 4.32473717 for cultivar 3. The mid-way point between the mean values for cultivars 1 and 2 is (-3.42248851-0.07972623)/2=-1.751107, and the mid-way point between the mean values for cultivars 2 and 3 is (-0.07972623+4.32473717)/2 = 2.122505.

Therefore, we can use the following allocation rule:

- if the first discriminant function is <= -1.751107, predict the sample to be from cultivar 1
- if the first discriminant function is > -1.751107 and <= 2.122505, predict the sample to be from cultivar 2
- if the first discriminant function is > 2.122505, predict the sample to be from cultivar 3

We can examine the accuracy of this allocation rule by using the "calcAllocationRuleAccuracy()" function below:

For example, to calculate the accuracy for the wine data based on the allocation rule for the first discriminant function, we type:

```
> calcAllocationRuleAccuracy(wine.lda.values$x[,1], wine[1], c(-1.751107, 2.122505))
  [1] "Number of samples of group 1 classified as group 1  :  56 (cutoffs: -1.751107 , NA )"
  [1] "Number of samples of group 1 classified as group 2  :  3 (cutoffs: -1.751107 , 2.122505 )"
  [1] "Number of samples of group 1 classified as group 3  :  NA (cutoffs: 2.122505 , NA )"
  [1] "Number of samples of group 2 classified as group 1  :  5 (cutoffs: -1.751107 , NA )"
  [1] "Number of samples of group 2 classified as group 2  :  65 (cutoffs: -1.751107 , 2.122505 )"
  [1] "Number of samples of group 2 classified as group 3  :  1 (cutoffs: 2.122505 , NA )"
  [1] "Number of samples of group 3 classified as group 1  :  NA (cutoffs: -1.751107 , NA )"
  [1] "Number of samples of group 3 classified as group 2  :  NA (cutoffs: -1.751107 , 2.122505 )"
  [1] "Number of samples of group 3 classified as group 3  :  48 (cutoffs: 2.122505 , NA )"
```

This can be displayed in a "confusion matrix":

	Allocated to group 1	Allocated to group 2	Allocated to group 3
Is group 1	56	3	0
Is group 2	5	65	1
Is group 3	0	0	48

There are 3+5+1=9 wine samples that are misclassified, out of (56+3+5+65+1+48=) 178 wine samples: 3 samples from cultivar 1 are predicted to be from cultivar 2, 5 samples from cultivar 2 are predicted to be from cultivar 1, and 1 sample from cultivar 2 is predicted to be from cultivar 3. Therefore, the misclassification rate is 9/178, or 5.1%. The misclassification rate is quite low, and therefore the accuracy of the allocation rule appears to be relatively high.

However, this is probably an underestimate of the misclassification rate, as the allocation rule was based on this data (this is the "training set"). If we calculated the misclassification rate for a separate "test set" consisting of data other than that used to make the allocation rule, we would probably get a higher estimate of the misclassification rate.

Canonical Correlation Analysis

PCA is used to investigate one set of variables with or without covariates. The original variables are replaced by a set of variates called principal components. The components are created to account for maximal variation among the original variables.

A generalization of PCA, developed by Hotelling (1936), is canonical correlation analysis (CCA). The method was developed to investigate relationships between two sets of variables with one or more sets of covariates.

In CCA one is interested in investigating relationships between two sets of variables $\mathbf{Y}' = [Y_1, Y_2, \ldots, Y_p]$ and $\mathbf{X}' = [X_1, X_2, \ldots, X_q]$. In addition one may also have sets of covariates $\mathbf{Z}' = [Z_1, Z_2, \ldots, Z_r]$ and $\mathbf{W}' = [W_1, W_2, \ldots, W_s]$. More will be said about the inclusion of covariates later in this section. The goal of CCA as developed by Hotelling (1936) is to construct two new sets of canonical variates $U = \alpha'\mathbf{Y}$ and $V = \beta'\mathbf{X}$ that are linear combinations of the original variables such that the simple correlation between U and V is maximal, subject to the restriction that each canonical variate U and V has unit variance (to ensure uniqueness, except for sign) and is uncorrelated with other constructed variates within the set.

Canonical correlation analysis may be used to determine whether two sets of variables are independent assuming multivariate normality. As an exploratory tool, it is used as a data reduction method. Given a large number of variables, one may want to locate a few canonical variates in each set to study and reconstruct the intercorrelations among the variables. For example, given a set of ability variables and a set of personality variables, the object of the study may be to determine what sort of personality traits may be associated with various ability domains.

Canonical correlation analysis is used to identify and measure the associations among two sets of variables. Canonical correlation is appropriate in the same situations where multiple regression would be, but where are there are multiple intercorrelated outcome variables. Canonical correlation analysis determines a set of canonical variates, orthogonal linear combinations of the variables within each set that best explain the variability both within and between sets.

Consider, as an example, variables related to exercise and health. On the one hand, you have variables associated with exercise, observations such as the climbing rate on a stair stepper, how fast you can run a certain distance, the amount of weight lifted on a bench press, the number of push-ups per minute, etc. On the other hand, you have variables that attempt to measure overall health, such as blood pressure, cholesterol levels, glucose levels, body mass index, etc. Two types of variables are measured and the relationships between the exercise variables and the health variables are of interest.

As a second example consider variables measured on environmental health and environmental toxins. A number of environmental health variables such as frequencies of sensitive species, species diversity, total biomass, the productivity of the environment, etc. may be measured and a second set of variables on environmental toxins are measured, such as the concentrations of heavy metals, pesticides, dioxin, etc.

For a third example consider a group of sales representatives, on whom we have recorded several sales performance variables along with several measures of intellectual and creative aptitude. We may wish to explore the relationships between the sales performance variables and the aptitude variables.

One approach to studying relationships between the two sets of variables is to use canonical correlation analysis which describes the relationship between two sets of variables. We do not necessarily think of one set of variables as independent and the other as dependent, though that may potentially be another approach.

It is possible to create pairwise scatter plots with variables in the first set (e.g., exercise variables), and variables in the second set (e.g., health variables). But if the dimension of the first set is p and that of the second set is q, there will be pq such scatter plots, it may be difficult, if not impossible, to look at all of these graphs together and interpret the results.

Similarly, you could compute all correlations between variables from the first set (e.g., exercise variables), and variables in the second set (e.g., health variables), however, interpretation is difficult when pq is large.

Canonical Correlation Analysis allows us to summarize the relationships into fewer statistics while preserving the main facets of the relationships. In a way, the motivation for canonical correlation is very similar to principal component analysis. It is another dimension-reduction technique.

a. *Population Model for CCA*

Given two sets of variables $\mathbf{Y}' = [Y_1, Y, \ldots, Y_p]$ and $\mathbf{X}' = [X_1, X_2, \ldots, X_q]$ with associated covariance, matrices Σ_{yy} and Σ_{xx} where $s = \min(p, q)$, one goal of CCA is to identify canonical variates $U = \boldsymbol{\alpha}'\mathbf{Y}$ and $V = \boldsymbol{\beta}'\mathbf{X}$ such that the correlation between U and V is maximal. Letting the population covariance matrix for the joint vector $(Y_1, Y_2, \ldots, Y_p, X_1, X_2, \ldots, X_q) = (\mathbf{Y}', \mathbf{X}')$ be defined as

$$\Sigma = \begin{bmatrix} \Sigma_{yy} & \Sigma_{yx} \\ \Sigma_{xy} & \Sigma_{xx} \end{bmatrix} \tag{8.7.1}$$

the correlation between U and V is

$$\rho_{UV} = \frac{\boldsymbol{\alpha}'\Sigma_{yx}\boldsymbol{\beta}}{\sqrt{(\boldsymbol{\alpha}'\Sigma_{yy}\boldsymbol{\alpha})(\boldsymbol{\beta}'\Sigma_{xx}\boldsymbol{\beta})}} \tag{8.7.2}$$

Because ρ_{UV} involves the canonical variates U and V, it is called a canonical correlation. Thus, the goal of CCA is to find canonical variates $U_i = \boldsymbol{\alpha}_i'\mathbf{Y}$ and $V_i = \boldsymbol{\beta}_i'\mathbf{X}$ for $i = 1, 2, \ldots, s$ such that the set of variates U_i are uncorrelated and the set of variates V_i are uncorrelated, each having unit variance

$$\operatorname{cov}(U_i, U_j) = \operatorname{cov}(V_i, V_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Furthermore, the covariance or correlation between U_i and V_i is ρ_i, for $i = 1, 2, \ldots, s$ and 0, otherwise

$$\begin{aligned} \operatorname{cov}(U_i,\ V_i) &= \operatorname{cov}(U_i,\ V_i) = \rho_i \quad i = 1, 2, \ldots, s \\ \operatorname{cov}(U_i,\ V_i) &= \operatorname{cov}\left(U_i,\ V_j\right) = 0 \quad i \neq j \end{aligned}$$

Thus, if $U_1,\ U_2$ $(p = 2)$ and V_1, V_2 and V_3 $(q = 3)$ are canonical variates, the correlation matrix for $\mathbf{U}' = [U_1,\ U_2]$ and $\mathbf{V}' = [V_1,\ V_2,\ V_3]$ has the form

$$\begin{array}{c} \\ U_1 \\ U_2 \\ \\ V_1 \\ V_2 \\ V_3 \end{array} \begin{array}{c} \begin{array}{cccccc} U_1 & U_2 & & V_1 & V_2 & V_3 \end{array} \\ \begin{bmatrix} 1 & 0 & \vdots & \rho_1 & 0 & 0 \\ 0 & 1 & \vdots & 0 & \rho_2 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \rho_1 & 0 & \vdots & 1 & 0 & 0 \\ 0 & \rho_2 & \vdots & 0 & 1 & 0 \\ 0 & 0 & \vdots & 0 & 0 & 1 \end{bmatrix} \end{array} = \begin{bmatrix} \mathbf{I}_p & \vdots & \Lambda \\ \cdots & \cdots & \cdots \\ \Lambda' & \vdots & \mathbf{I}_q \end{bmatrix}$$

To maximize (8.7.2), we assume that both Σ_{yy} and Σ_{xx} are positive definite and let $\Sigma_{yy}^{1/2}$ and $\Sigma_{xx}^{1/2}$ be the square root matrices of Σ_{yy} and Σ_{xx}, respectively. Without loss of generality we assume $p \leq q$ (if this is not the case, the role of $\mathbf{X}$ and $\mathbf{Y}$ are interchanged)

and let $\widetilde{\alpha} = \Sigma_{yy}^{-1/2}\alpha$ and $\widetilde{\beta} = \Sigma_{xx}^{-1/2}\beta$. Then from (8.7.2) we must find α and β to maximize the

$$\underset{\alpha,\beta}{\operatorname{cov}}\left(\alpha'\mathbf{Y}, \beta'\mathbf{X}\right)^2 = \frac{(\widetilde{\alpha}'\Sigma_{yy}^{-1/2}\Sigma_{yx}\Sigma_{xx}^{-1/2}\widetilde{\beta})^2}{\left(\widetilde{\alpha}'\widetilde{\alpha}\right)\left(\widetilde{\beta}'\widetilde{\beta}\right)}$$

Now, let $\phi = \Sigma_{yy}^{-1/2}\Sigma_{yx}\Sigma_{xx}^{-1/2}$ so that

$$\phi\phi' = \Sigma_{yy}^{-1/2}\Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy}\Sigma_{yy}^{-1/2} \quad \text{and} \quad \phi'\phi = \Sigma_{xx}^{-1/2}\Sigma_{xy}\Sigma_{yy}\Sigma_{yx}\Sigma_{xx}^{-1/2}.$$

Then by the Cauchy-Schwarz inequality (Theorem 2.3.4), $(\widetilde{\alpha}'\phi\widetilde{\beta})^2 \leq \left(\widetilde{\alpha}'\phi\phi'\alpha\right)\left(\widetilde{\beta}'\beta\right)$ and since the roots of $\phi\phi'$ are the same as $\phi'\phi$, the maximum of (8.7.2) may be obtained by solving the eigenequations $\left|\phi\phi' - \rho^2\mathbf{I}\right| = 0$ or $\left|\phi'\phi - \rho^2\mathbf{I}\right| = 0$. In terms of the covariance matrices, we may solve

$$\begin{aligned} \left|\Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy} - \rho^2\Sigma_{yy}\right| &= 0 \\ \left|\Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx} - \rho^2\Sigma_{xx}\right| &= 0 \end{aligned} \tag{8.7.3}$$

for $s = \min(p, q) = p < q$ roots ρ_i^2. Using Theorem 2.6.7, let $\left[\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_p\right] = \mathbf{A}$ be the eigenvectors of the first equation and $\left[\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_q\right] = \mathbf{B}$ be the eigenvectors of the second equation. Then $\mathbf{A}'\Sigma_{yy}\mathbf{A} = \mathbf{I}_p$ and $\mathbf{A}'\Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy}\mathbf{A} = \Lambda$; and $\mathbf{B}'\Sigma_{xx}\mathbf{B} = \mathbf{I}_q$ and $\mathbf{B}'\Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}\mathbf{B} = \Lambda$. The square root of the roots ρ_i^2 are the canonical correlations between the orthogonal canonical variates $U_i = \alpha_i'\mathbf{Y}$ and the orthogonal variates $V_i = \beta_i'\mathbf{X}$

for $i = 1, 2, \ldots, p$. They have been constructed so that the canonical correlations $\rho_1 \geq \rho_2 \geq \ldots \geq \rho_p$ are maximal. The vector of canonical variates are defined as $\mathbf{U} = \mathbf{A}'\mathbf{Y}$ and $\mathbf{V} = \mathbf{B}'\mathbf{X}$. To avoid solving both equations in (8.7.3), the eigenvectors are related as follows

$$\begin{aligned} \boldsymbol{\alpha}_i &= \Sigma_{yy}^{-1}\Sigma_{yx}\boldsymbol{\beta}_i/\rho_i \\ \boldsymbol{\beta}_i &= \Sigma_{xx}^{-1}\Sigma_{xy}\boldsymbol{\alpha}_i/\rho_i \end{aligned} \tag{8.7.4}$$

for $i = 1, 2, \ldots, p$. The vectors $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$ may be standardized to one, $\boldsymbol{\alpha}_i'\Sigma_{yy}\boldsymbol{\alpha}_i = \boldsymbol{\beta}_i'\Sigma_{xx}\boldsymbol{\beta}_i = 1$, so that the variance of U_i and V_i are unity for $i = 1, 2, \ldots, p$.

In applying PCA, a researcher had to establish the scale of measurement for the analysis. This is not the case for CCA since the canonical correlations are invariant to changes in location and scale; the CCA procedure is scale free. Thus, one may replace the covariance matrices in (8.7.3) with population correlation matrices. The eigenvectors of the two are related by the simple rescaling

$$\begin{aligned} \boldsymbol{\xi}_i &= \left(\operatorname{diag}\Sigma_{yy}\right)^{1/2}\boldsymbol{\alpha}_i \\ \boldsymbol{\delta}_i &= \left(\operatorname{diag}\Sigma_{xx}\right)^{1/2}\boldsymbol{\beta}_i \end{aligned} \tag{8.7.5}$$

where $\boldsymbol{\xi}_i$ and $\boldsymbol{\delta}_i$ are the eigenvectors of

$$\begin{aligned} \left|\mathbf{P}_{yy}^{-1/2}\mathbf{P}_{yx}\mathbf{P}_{xx}^{-1}\mathbf{P}_{xy}\mathbf{P}_{yy}^{-1/2} - \rho^2\mathbf{I}\right| &= 0 \\ \left|\mathbf{P}_{xx}^{-1/2}\mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}\mathbf{P}_{yx}\mathbf{P}_{xx}^{-1/2} - \rho^2\mathbf{I}\right| &= 0 \end{aligned} \tag{8.7.6}$$

Letting $\mathbf{C}$ be the eigenvectors of the first equation in (8.7.6) and $\mathbf{D}$ be the eigenvectors of the second, then

$$\begin{aligned} \mathbf{C}'\mathbf{P}_{yy}\mathbf{C} &= \mathbf{I}_p \\ \mathbf{D}'\mathbf{P}_{xx}\mathbf{D} &= \mathbf{I}_q \\ \mathbf{C}'\mathbf{P}_{yx}\mathbf{D} &= \Lambda \end{aligned} \tag{8.7.7}$$

Since $\mathbf{D} = \mathbf{P}_{xx}^{-1}\mathbf{P}_{xy}\mathbf{C}$, the population intercorrelation matrix has the factorization

$$\begin{aligned} \mathbf{P}_{yx} &= \mathbf{P}_{yy}\mathbf{C}\Lambda\mathbf{D}'\mathbf{P}_{xx} \\ &= \sum_{i=1}^{p} \rho_i \mathbf{L}_i \mathbf{M}_i' \end{aligned} \tag{8.7.8}$$

where $\mathbf{L}_i$ and $\mathbf{M}_i$ are column vectors of $\mathbf{P}_{yy}\ \mathbf{C}$ and $\mathbf{P}_{xx}\ \mathbf{D}$, respectively. Thus, by selecting $k \ll s = \min(p, q) = p$ rescaled variates or factors, the correlations explained by k factors are

$$\mathbf{P}_{yx}(k) = \sum_{i=1}^{k} \rho_i \mathbf{L}_i \mathbf{M}_i' \tag{8.7.9}$$

and the residual matrix is

$$\mathbf{P}_{yx} - \mathbf{P}_{yx}(k) = \sum_{i=k+1}^{p} \rho_{i+1} \mathbf{L}_{i+1} \mathbf{M}_{i+1}' \tag{8.7.10}$$

The residual matrix is zero if the rank of $\mathbf{P}_{yx}$ is k.

These observations suggest that one may employ CCA to study and recover the relationship between $\mathbf{Y}$ and $\mathbf{X}$ by studying the structure and rank of $\mathbf{P}_{yx}$, the matrix of intercorrelations. To formalize the relationship, a linear model is formulated for $\mathbf{Y}$ and $\mathbf{X}$

$$\begin{aligned} \underset{p\times 1}{\mathbf{Y}} &= \boldsymbol{\mu}_y + \underset{p\times k}{\Lambda_y}\ \underset{k\times 1}{\mathbf{f}} + \underset{p\times 1}{\mathbf{e}_y} \\ \underset{q\times 1}{\mathbf{X}} &= \boldsymbol{\mu}_x + \underset{q\times k}{\Lambda_x}\ \underset{k\times 1}{\mathbf{f}} + \underset{q\times 1}{\mathbf{e}_x} \end{aligned} \tag{8.7.11}$$

where $\mathbf{f}$ is a vector of unobserved hypothetical factors, $\mathbf{e}_x$ and $\mathbf{e}_y$ are random error vectors, and Λ_y and Λ_x are matrices of regression weights of rank $k \leq p$. Furthermore, we assume that the elements of $\mathbf{f}$ are uncorrelated and have unit variance; and that $\mathbf{e}_y$, $\mathbf{e}_x$ and $\mathbf{f}$ are mutually uncorrelated with zero means

$$\begin{aligned} E(\mathbf{e}_x) = E(\mathbf{e}_y) &= E(\mathbf{f}) = \mathbf{0} \\ \operatorname{cov}(\mathbf{f}) &= \mathbf{I} \\ \operatorname{cov}(\mathbf{f}, \mathbf{e}_y) = \operatorname{cov}(\mathbf{f}, \mathbf{e}_x) &= \operatorname{cov}(\mathbf{e}_x, \mathbf{e}_y) = \mathbf{0} \end{aligned} \tag{8.7.12}$$

Then, standardizing $\mathbf{Y}$ and $\mathbf{X}$ to variables $\mathbf{Z}_y$ and $\mathbf{Z}_x$, model (8.7.11) and our assumptions imply that the population correlation matrices have the structure

$$\begin{aligned} \mathbf{P}_{yy} &= \Lambda_y \Lambda_y' + \Sigma_1 \\ \mathbf{P}_{xx} &= \Lambda_x \Lambda_x' + \Sigma_2 \\ \mathbf{P}_{yx} &= \Lambda_y \Lambda_x' \quad \text{and} \quad \mathbf{P}_{xy} = \Lambda_x \Lambda_y' \end{aligned} \tag{8.7.13}$$

where the cov $(\mathbf{e}_y) = \Sigma_1$ and the cov $(\mathbf{e}_x) = \Sigma_2$. Furthermore, by Theorem 2.5.2, the rank of $\mathbf{P}_{yx} = r\left(\Lambda_y \Lambda_x'\right) \leq k$ where k is the number of factors common to $\mathbf{Y}$ and $\mathbf{X}$. We may further assume that the rank of $\Lambda_y \Lambda_x'$ is k. For if it is less than k model (8.7.11) may be reformulated to have the number of common factors equal to the rank of $\mathbf{P}_{yx}$.

In expression (8.7.8), the matrices $\mathbf{L} = \mathbf{P}_{yy}\mathbf{C}$ and $\mathbf{M} = \mathbf{P}_{xx}\mathbf{D}$ represent the correlations between the p variables and the canonical variates since

$$\begin{aligned}
\rho_{\mathbf{YU}} = \operatorname{cov}(\mathbf{Y},\ \mathbf{U}) &= \left(\operatorname{diag} \Sigma_{yy}\right)^{-1/2} \Sigma_{yy}\mathbf{A} \\
&= \left(\operatorname{diag} \Sigma_{yy}\right)^{-1/2} \Sigma_{yy} \left(\operatorname{diag} \Sigma_{yy}\right)^{-1/2} \mathbf{C} \\
&= \mathbf{P}_{yy}\mathbf{C} \\
\rho_{\mathbf{XV}} = \operatorname{cov}(\mathbf{X},\ \mathbf{V}) &= (\operatorname{diag} \Sigma_{xx})^{-1/2} \Sigma_{xx}\mathbf{B} \\
&= (\operatorname{diag} \Sigma_{xx})^{-1/2} \Sigma_{yy} (\operatorname{diag} \Sigma_{xx})^{-1/2} \mathbf{D} \\
&= \mathbf{P}_{xx}\mathbf{D}
\end{aligned} \tag{8.7.14}$$

Thus, the vectors $\mathbf{L}_1, \mathbf{L}_2, \ldots, \mathbf{L}_k$ and $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_k$ fully recover $\mathbf{P}_{yx}$ for some $k \leq p$ where $k = r\left(\mathbf{P}_{yx}\right)$ so that $\rho_{k+1} = \ldots = \rho_p = 0$.

It is often the case that the first few canonical correlations $\rho_1, \rho_1, \rho_2, \ldots, \rho_k$ are large while the other $\rho_{k+1}, \rho_{k+2}, \ldots, \rho_p$ are small, then most of the intercorrelation is explained by the first k components or factors. An index of how much the correlation structure is explained by the first k canonical variates is

$$\rho_k^2 = \sum_{i=1}^{k} \rho_i^2 / \sum_{i=1}^{p} \rho_i^2 = \sum_{i=1}^{k} \rho_i^2 / \operatorname{tr}(\mathbf{P}_{yy}^{-1/2}\mathbf{P}_{yx}\mathbf{P}_{xx}^{-1}\mathbf{P}_{xy}\mathbf{P}_{yy}^{-1/2}) \tag{8.7.15}$$

If the $r\left(\mathbf{P}_{yx}\right) = k$, then $\rho_k^2 = 1$. Each ρ_i^2 is a measure of the proportion of variance of U_i explained by V_i.

Examples of canonical correlation analysis

Example 1. A researcher has collected data on three psychological variables, four academic variables (standardized test scores) and gender for 600 college freshman. She is interested in how the set of psychological variables relates to the academic variables and gender. In particular, the researcher is interested in how many dimensions (canonical variables) are necessary to understand the association between the two sets of variables.

Example 2. A researcher is interested in exploring associations among factors from two multidimensional personality tests, the MMPI and the NEO. She is interested in what dimensions are common between the tests and how much variance is shared between them. She is specifically interested in finding whether the neuroticism dimension from the NEO can account for a substantial amount of shared variance between the two tests.

Description of the data

For our analysis example, we are going to expand example 1 about investigating the associations between psychological measures and academic achievement measures.

We have a data file, **mmreg.dta**, with 600 observations on eight variables. The psychological variables are **`locus_of_control`**, **`self_concept`** and **`motivation`**. The academic variables are standardized tests in reading (**`read`**), writing (**`write`**), math (**`math`**) and science (**`science`**). Additionally, the variable **`female`** is a zero-one indicator variable with the one indicating a female student.

```r
mm <- read.csv("https://stats.idre.ucla.edu/stat/data/mmreg.csv")
colnames(mm) <- c("Control", "Concept", "Motivation", "Read", "Write", "Math",
    "Science", "Sex")
summary(mm)
```

```
##     Control           Concept          Motivation         Read
##  Min.   :-2.2300   Min.   :-2.6200   Min.   :0.000   Min.   :28.3
##  1st Qu.:-0.3725   1st Qu.:-0.3000   1st Qu.:0.330   1st Qu.:44.2
##  Median : 0.2100   Median : 0.0300   Median :0.670   Median :52.1
##  Mean   : 0.0965   Mean   : 0.0049   Mean   :0.661   Mean   :51.9
##  3rd Qu.: 0.5100   3rd Qu.: 0.4400   3rd Qu.:1.000   3rd Qu.:60.1
##  Max.   : 1.3600   Max.   : 1.1900   Max.   :1.000   Max.   :76.0
##      Write           Math         Science          Sex
##  Min.   :25.5   Min.   :31.8   Min.   :26.0   Min.   :0.000
##  1st Qu.:44.3   1st Qu.:44.5   1st Qu.:44.4   1st Qu.:0.000
##  Median :54.1   Median :51.3   Median :52.6   Median :1.000
##  Mean   :52.4   Mean   :51.9   Mean   :51.8   Mean   :0.545
##  3rd Qu.:59.9   3rd Qu.:58.4   3rd Qu.:58.6   3rd Qu.:1.000
##  Max.   :67.1   Max.   :75.5   Max.   :74.2   Max.   :1.000
```

Canonical correlation analysis

Below we use the **canon** command to conduct a canonical correlation analysis. It requires two sets of variables enclosed with a pair of parentheses. We specify our psychological variables as the first set of variables and our academic variables plus gender as the second set. For convenience, the variables in the first set are called "u" variables and the variables in the second set are called "v" variables.

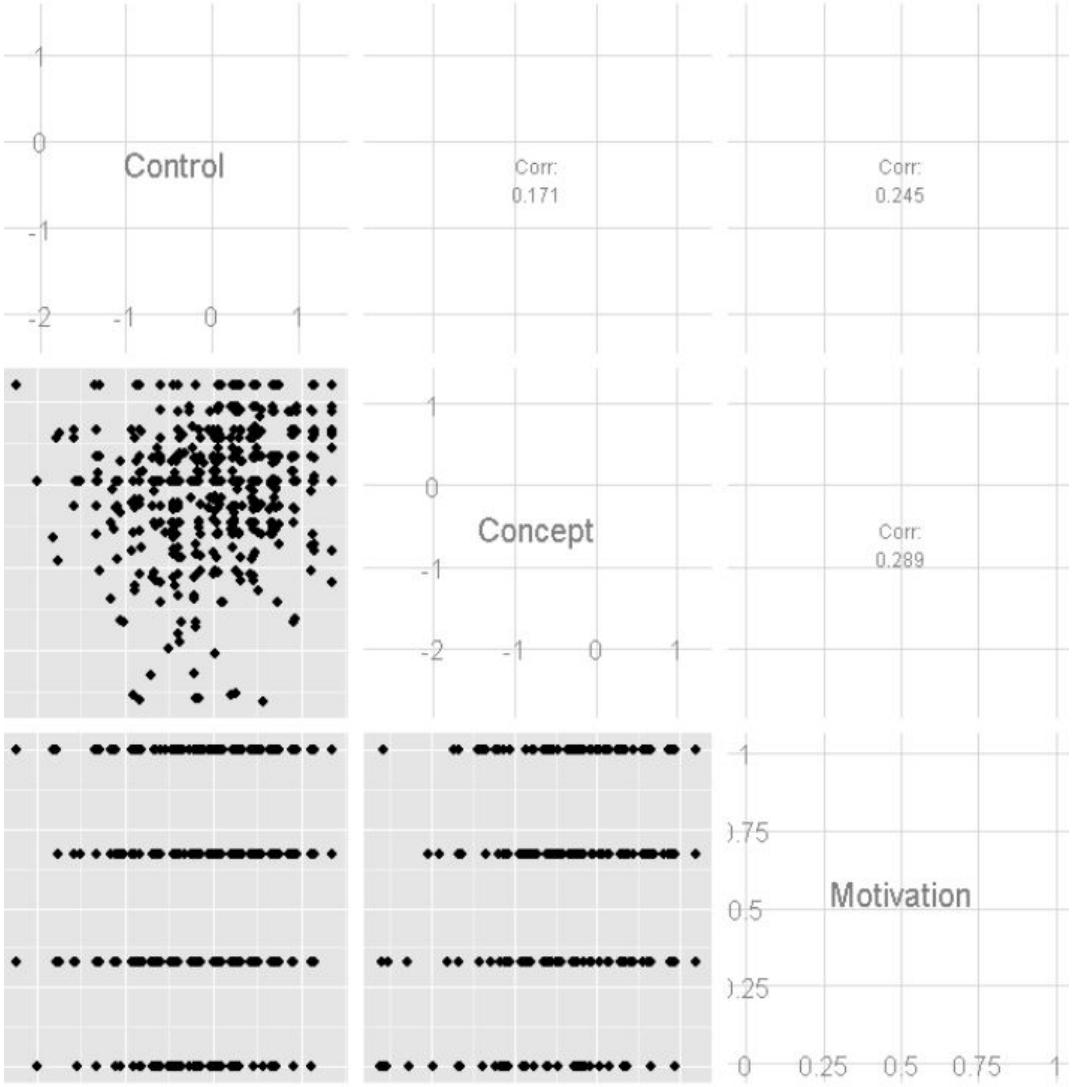
Let's look at the data.

```
xtabs(~Sex, data = mm)
```

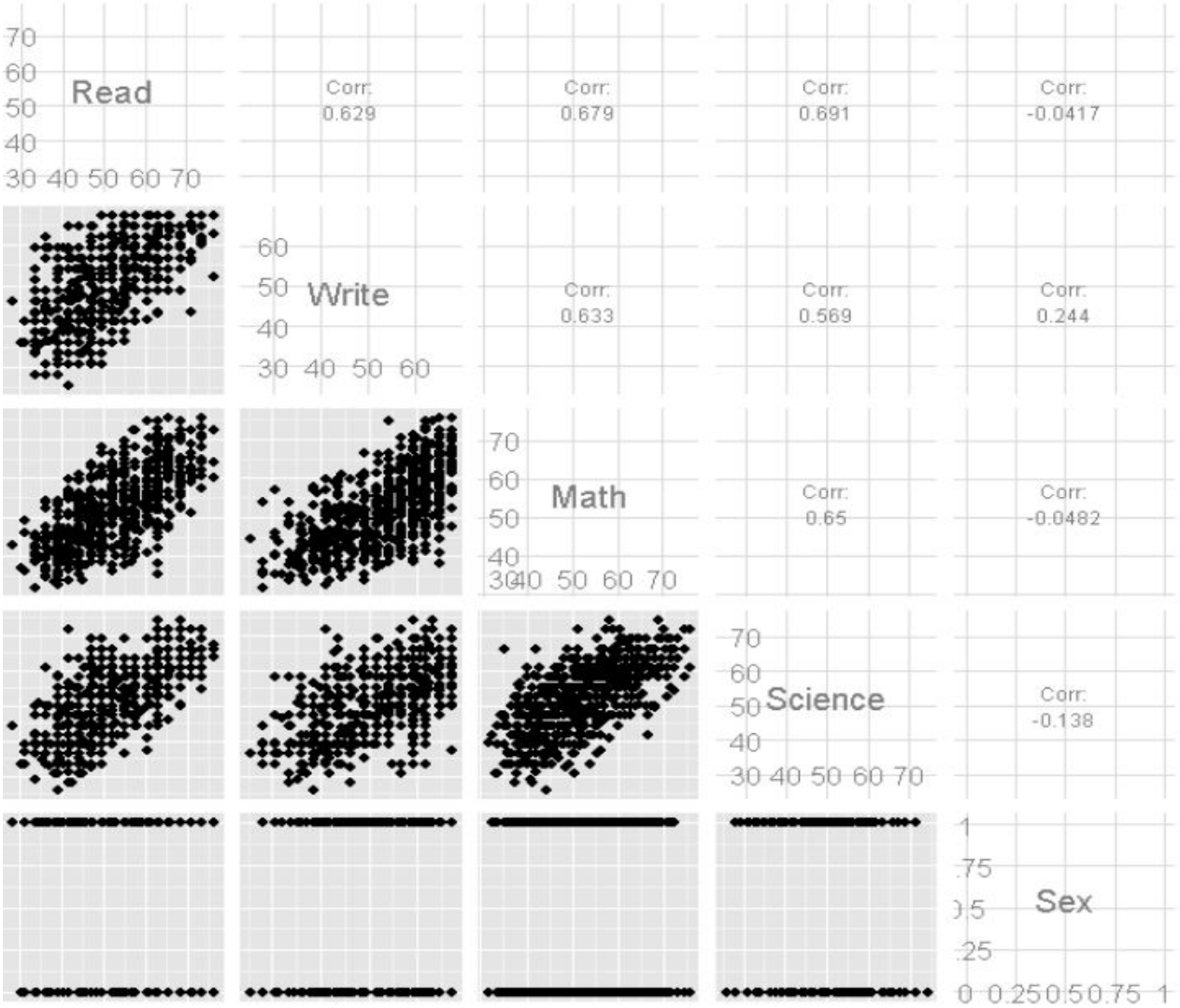
```
## Sex
##   0   1
## 273 327
```

```
psych <- mm[, 1:3]
acad <- mm[, 4:8]
```

```r
ggpairs(psych)
```




```r
ggpairs(acad)
```



```r
# correlations
matcor(psych, acad)
```

```
## $Xcor
##            Control Concept Motivation
## Control     1.0000  0.1712     0.2451
## Concept     0.1712  1.0000     0.2886
## Motivation  0.2451  0.2886     1.0000
##
## $Ycor
##             Read  Write     Math Science      Sex
## Read     1.00000 0.6286  0.67928  0.6907 -0.04174
## Write    0.62859 1.0000  0.63267  0.5691  0.24433
## Math     0.67928 0.6327  1.00000  0.6495 -0.04822
## Science  0.69069 0.5691  0.64953  1.0000 -0.13819
## Sex     -0.04174 0.2443 -0.04822 -0.1382  1.00000
##
```

```
## $XYcor
##            Control  Concept Motivation     Read   Write     Math  Science
## Control     1.0000  0.17119     0.2451  0.37357 0.35888  0.33727  0.32463
## Concept     0.1712  1.00000     0.2886  0.06066 0.01945  0.05360  0.06983
## Motivation  0.2451  0.28857     1.0000  0.21061 0.25425  0.19501  0.11567
## Read        0.3736  0.06066     0.2106  1.00000 0.62859  0.67928  0.69069
## Write       0.3589  0.01945     0.2542  0.62859 1.00000  0.63267  0.56915
## Math        0.3373  0.05360     0.1950  0.67928 0.63267  1.00000  0.64953
## Science     0.3246  0.06983     0.1157  0.69069 0.56915  0.64953  1.00000
## Sex         0.1134 -0.12595     0.0981 -0.04174 0.24433 -0.04822 -0.13819
##                 Sex
## Control     0.11341
## Concept    -0.12595
## Motivation  0.09810
## Read       -0.04174
## Write       0.24433
## Math       -0.04822
## Science    -0.13819
## Sex         1.00000
```

R Canonical Correlation Analysis

```
cc1 <- cc(psych, acad)

# display the canonical correlations
cc1$cor
```

```
## [1] 0.4641 0.1675 0.1040
```

```
# raw canonical coefficients
cc1[3:4]
```

```
## $xcoef
##                [,1]    [,2]    [,3]
## Control     -1.2538 -0.6215 -0.6617
## Concept      0.3513 -1.1877  0.8267
## Motivation -1.2624  2.0273  2.0002
##
## $ycoef
##               [,1]       [,2]       [,3]
## Read     -0.044621 -0.004910  0.021381
## Write    -0.035877  0.042071  0.091307
## Math     -0.023417  0.004229  0.009398
## Science -0.005025 -0.085162 -0.109835
## Sex      -0.632119  1.084642 -1.794647
```

The raw canonical coefficients are interpreted in a manner analogous to interpreting regression coefficients i.e., for the variable **read**, a one unit increase in reading leads to a .0446 decrease in the first canonical variate of set 2 when all of the other variables are held constant. Here is another example: being female leads to a .6321 decrease in the dimension 1 for the academic set with the other predictors held constant.

Next, we'll use **comput** to compute the loadings of the variables on the canonical dimensions (variates). These loadings are correlations between variables and the canonical variates.

```
# compute canonical loadings
cc2 <- comput(psych, acad, cc1)


# display canonical loadings
cc2[3:6]
```

```
## $corr.X.xscores
##                  [,1]    [,2]    [,3]
## Control     -0.90405 -0.3897 -0.1756
## Concept     -0.02084 -0.7087  0.7052
## Motivation -0.56715  0.3509  0.7451
##
## $corr.Y.xscores
##              [,1]      [,2]      [,3]
## Read     -0.3900 -0.06011  0.01408
## Write    -0.4068  0.01086  0.02647
## Math     -0.3545 -0.04991  0.01537
## Science -0.3056 -0.11337 -0.02395
## Sex      -0.1690  0.12646 -0.05651
##
## $corr.X.yscores
##                   [,1]      [,2]      [,3]
## Control     -0.419555 -0.06528 -0.01826
## Concept     -0.009673 -0.11872  0.07333
## Motivation -0.263207  0.05878  0.07749
##
## $corr.Y.yscores
##              [,1]      [,2]    [,3]
## Read     -0.8404 -0.35883  0.1354
## Write    -0.8765  0.06484  0.2546
## Math     -0.7639 -0.29795  0.1478
## Science -0.6584 -0.67680 -0.2304
## Sex      -0.3641  0.75493 -0.5434
```

The above correlations are between observed variables and canonical variables which are known as the canonical loadings. These canonical variates are actually a type of latent variable.

In general, the number of canonical dimensions is equal to the number of variables in the smaller set; however, the number of significant dimensions may be even smaller. Canonical dimensions, also known as canonical variates, are latent variables that are analogous to factors obtained in factor analysis. For this particular model there are three canonical dimensions of which only the first two are statistically significant. For statistical test we use R package **"CCP"**.

```
# tests of canonical dimensions
rho <- cc1$cor
## Define number of observations, number of variables in first set, and number of variables in the second set.
n <- dim(psych)[1]
p <- length(psych)
q <- length(acad)

## Calculate p-values using the F-approximations of different test statistics:
p.asym(rho, n, p, q, tstat = "Wilks")
```

```
## Wilks' Lambda, using F-approximation (Rao's F):
##            stat approx df1  df2 p.value
## 1 to 3:  0.754  11.72  15 1635 0.00000
## 2 to 3:  0.961   2.94   8 1186 0.00291
## 3 to 3:  0.989   2.16   3  594 0.09109
```

```
p.asym(rho, n, p, q, tstat = "Hotelling")
```

```
##  Hotelling-Lawley Trace, using F-approximation:
##             stat approx df1  df2 p.value
## 1 to 3:  0.3143  12.38  15 1772 0.00000
## 2 to 3:  0.0398   2.95   8 1778 0.00281
## 3 to 3:  0.0109   2.17   3 1784 0.09001
```

```
p.asym(rho, n, p, q, tstat = "Pillai")
```

```
##  Pillai-Bartlett Trace, using F-approximation:
##             stat approx df1  df2 p.value
## 1 to 3:  0.2542  11.00  15 1782 0.00000
## 2 to 3:  0.0389   2.93   8 1788 0.00293
## 3 to 3:  0.0108   2.16   3 1794 0.09044
```

As shown in the table above, the first test of the canonical dimensions tests whether all three dimensions are significant (they are, F = 11.72), the next test tests whether dimensions 2 and 3 combined are significant (they are, F = 2.94). Finally, the last test tests whether dimension 3, by itself, is significant (it is not). Therefore dimensions 1 and 2 must each be significant while dimension three is not.

When the variables in the model have very different standard deviations, the standardized coefficients allow for easier comparisons among the variables. Next, we'll compute the standardized canonical coefficients.

```
# standardized psych canonical coefficients diagonal matrix of psych sd's
s1 <- diag(sqrt(diag(cov(psych))))
s1 %*% cc1$xcoef
```

```
##        [,1]   [,2]   [,3]
## [1,] -0.8404 -0.4166 -0.4435
## [2,]  0.2479 -0.8379  0.5833
## [3,] -0.4327  0.6948  0.6855
```

```r
# standardized acad canonical coefficients diagonal matrix of acad sd's
s2 <- diag(sqrt(diag(cov(acad))))
s2 %*% cc1$ycoef
```

```
##         [,1]    [,2]    [,3]
## [1,] -0.45080 -0.04961  0.21601
## [2,] -0.34896  0.40921  0.88810
## [3,] -0.22047  0.03982  0.08848
## [4,] -0.04878 -0.82660 -1.06608
## [5,] -0.31504  0.54057 -0.89443
```

The standardized canonical coefficients are interpreted in a manner analogous to interpreting standardized regression coefficients. For example, consider the variable **read**, a one standard deviation increase in reading leads to a 0.45 standard deviation decrease in the score on the first canonical variate for set 2 when the other variables in the model are held constant.

Sample Write-Up of the Analysis

There is a lot of variation in the write-ups of canonical correlation analyses. The write-up below is fairly minimal, including only the tests of dimensionality and the standardized coefficients.

Table 1: Tests of Canonical Dimensions

Dimension	Canonical Corr.	Mult. F	df1	df2	p
1	0.46	11.72	15	1634.7	0.0000
2	0.17	2.94	8	1186	0.0029
3	0.10	2.16	3	594	0.0911

Table 2: Standardized Canonical Coefficients

	Dimension 1	Dimension 2
Psychological Variables		
locus of control	-0.84	-0.42
self-concept	0.25	-0.84
motivation	-0.43	0.69
Academic Variables plus Gender		
reading	-0.45	-0.05
writing	-0.35	0.41
math	-0.22	0.04
science	-0.05	-0.83
gender (female=1)	-0.32	0.54

Tests of dimensionality for the canonical correlation analysis, as shown in Table 1, indicate that two of the three canonical dimensions are statistically significant at the .05 level. Dimension 1 had a canonical correlation of 0.46 between the sets of variables, while for dimension 2 the canonical correlation was much lower at 0.17.

Table 2 presents the standardized canonical coefficients for the first two dimensions across both sets of variables. For the psychological variables, the first canonical dimension is most strongly influenced by locus of control (-.84) and for the second dimension self-concept (-.84) and motivation (.69). For the academic variables plus gender, the first dimension was comprised of reading (-.45), writing (-.35) and gender (-.32). For the second dimension writing (.41), science (-.83) and gender (.54) were the dominating variables.

- Multivatiate normal distribution assumptions are required for both sets of variables.
- Canonical correlation analysis is not recommended for small samples.

Exploratory Factor Analysis

Exploratory Factor Analysis (EFA) is a powerful statistical method used in data analysis for uncovering the underlying structure of a relatively large set of variables. It is particularly valuable in situations where the relationships between variables are not entirely known or when data analysts seek to identify underlying latent factors that explain observed patterns in data.

At its core, EFA helps in simplifying complex data sets by reducing a large number of variables into a smaller set of underlying factors, without significant loss of information. This technique is instrumental in various fields, including psychology, marketing, finance, and social sciences, where it aids in identifying patterns and relationships that are not immediately apparent.

The importance of EFA lies in its ability to provide insights into the underlying mechanisms or constructs that influence data. For example, in psychology, EFA can be used to identify underlying personality traits from a set of observed behaviors. In customer satisfaction surveys, it helps in pinpointing key factors that drive consumer perceptions and decisions.

Moreover, EFA is crucial for enhancing the validity and reliability of research findings. By identifying the underlying factor structure, it ensures that subsequent analyses, like regression or hypothesis testing, are based on relevant and concise data constructs. This not only streamlines the data analysis process but also contributes to more accurate and interpretable results.

- The fundamental idea behind EFA is that there are latent factors that cannot be directly measured but are represented by the observed variables.
- Mathematically, EFA models the observed variables as linear combinations of potential factors plus error terms. This model is represented as: $X = LF + E$, where X is the matrix of observed variables, L is the matrix of loadings (which shows the relationship between variables and factors), F is the matrix of factors, and E is the error term.
- Factor loadings, which are part of the output of EFA, indicate the degree to which each variable is associated with each factor. High loadings suggest that the variable has a strong association with the factor.
- The process involves extracting factors from the data and then rotating them to achieve a more interpretable structure. Common rotation methods include Varimax and Oblimin.
- EFA differs from Confirmatory Factor Analysis (CFA) in its purpose and application. While EFA is exploratory in nature, used when the structure of the data is unknown, CFA is confirmatory, used to test hypotheses or theories about the structure of the data.

Exploratory Factor Analysis

Exploratory factor analysis (EFA) is a causal modeling technique that attempts to "explain" correlations among a set of observed (manifest) variables through the linear combination of a few unknown number of latent (unobserved) random factors. The procedure was originated by the psychologist Charles Spearman in the early 1900's to model human intelligence. Spearman developed the technique to try to understand the causal relationship between the latent human trait intelligence and test scores (grades) obtained in several disciplines.

Because both PCA and EFA usually begin with an analysis of the variation of a set of variables as characterized by the correlation or covariance matrix and because both may be used to characterize the variation by a few hypothetical constructs they are often confused. In EFA one may explain all the covariances or correlations with a few common factors that are unobservable or latent. In PCA one needs all components to account for all the covariances or correlations. While a few factors may account for all the intercorrelations (covariances), the same number of factors will not explain as much of the total variance as the same number of principal components. Thus, PCA is concerned with explaining the variance in the variables while EFA is concerned with explaining the covariances. In EFA, the correlation or covariance matrix is partitioned into two parts; that due to the common factors and that due to the unique factors. Any correlations (covariances) not explained by

the common factors are associated with the mutually uncorrelated unique (residual) factors. In PCA there is no residual variance, all variance is explained by the components. Another important difference between the two procedures lies in the direction of analysis. In EFA one sets forth a causal model and uses the data to validate the model. This has led to some controversy among statisticians since the power to reject model fit depends on the number of unknown factors and sample size. In PCA there is no causal model since all components are linear combinations of observed data, observed with no error.

a. Population Model for EFA

In EFA we begin with a random observation vector $\mathbf{Y}'_{1\times p} = [Y_1, Y_2, \ldots, Y_p]$ with mean $\boldsymbol{\mu}$ and covariance matrix Σ. Each vector of p-variables is assumed to have the linear structure

$$\underset{p\times 1}{\mathbf{Y}} - \underset{p\times 1}{\boldsymbol{\mu}} = \underset{p\times k}{\Lambda}\ \underset{k\times 1}{\mathbf{f}} + \underset{p\times 1}{\mathbf{e}} \tag{8.9.1}$$

where $\Lambda = [\lambda_{ij}]$ is a matrix of regression weights or "loadings" with rank $k \ll p$, $\mathbf{f}$ is a vector of random, unobserved latent factors and $\mathbf{e}$ is a vector of random errors. It is further assumed that $E(\mathbf{Y}) = \boldsymbol{\mu}$, $E(\mathbf{f}) = E(\mathbf{e}) = \mathbf{0}$, $\text{cov}(\mathbf{f}) = \mathbf{I}$, $\text{cov}(\mathbf{Y}) = \Sigma$, $\text{cov}(\mathbf{e}) = \Psi$ is a diagonal matrix with elements $\psi_i > 0$, and that the $\text{cov}(\mathbf{f}, \mathbf{e}) = \mathbf{0}$. These assumptions imply that the covariance matrix Σ for each observation $\mathbf{Y}$ has the structure

$$\begin{aligned} \Sigma &= \text{cov}(\mathbf{Y}) = \text{cov}(\Lambda\ \mathbf{f} + \mathbf{e}) \\ &= \Lambda\ \text{cov}(\mathbf{f})\ \Lambda' + \text{cov}(\mathbf{e}) \\ &= \Lambda\Lambda' + \Psi \end{aligned} \tag{8.9.2}$$

where the covariance matrix Ψ is diagonal, $\Psi = \operatorname{diag}\left[\psi_1, \psi_2, \ldots, \psi_p\right]$. Thus, all the covariances σ_{ij} for $i \neq j$ in Σ depend only on the λ_{ij} through the product $\Lambda\Lambda'$, while the variances σ_{ii} depend on λ_{ij}^2 and ψ_i.

One may also think of (8.9.1) as a linear regression model where

$$\begin{aligned} E(\mathbf{Y} \mid \mathbf{f}) &= \boldsymbol{\mu} + \Lambda\mathbf{f} \\ \operatorname{cov}(\mathbf{Y} \mid \mathbf{f}) &= \Psi = \operatorname{diag}\left[\psi_1, \psi_2, \ldots, \psi_p\right] \end{aligned} \tag{8.9.3}$$

Because the covariance matrix of $\mathbf{Y}$ given $\mathbf{f}$ is diagonal, $\mathbf{f}$ accounts for all the intercorrelations, and hence all linear relationships among the elements of $\mathbf{Y}$. Once the factors $\mathbf{f}$ are partialed out there remains no correlation among the elements of $\mathbf{Y}$. Furthermore, given (8.9.3), the covariance matrix

$$\begin{aligned} \Sigma &= E(\mathbf{Y} - \boldsymbol{\mu})(\mathbf{Y} - \boldsymbol{\mu})' = E(\Lambda\mathbf{f} + \mathbf{e})(\Lambda\mathbf{f} + \mathbf{e})' \\ &= \Lambda E\left(\mathbf{f}\mathbf{f}'\right)\Lambda' + \left(\mathbf{e}\mathbf{e}'\right) \\ &= \Lambda\Lambda' + \Psi \end{aligned} \tag{8.9.4}$$

so that the $\operatorname{cov}(\mathbf{Y} \mid \mathbf{f}) = \Sigma - \Lambda\Lambda' = \Psi$. Letting $\Psi = \operatorname{diag}\left(\Sigma - \Lambda\Lambda'\right)$, the partial correlation matrix becomes $\Psi^{-1/2}\left(\Sigma - \Lambda\Lambda'\right)\Psi^{-1/2}$.

From (8.9.1) or (8.9.3), we observe that the EFA model makes no distributional assumptions regarding $\mathbf{Y}$. The model parameters are the mean vector $\boldsymbol{\mu}$ and the parameter matrices Λ and Ψ. Later we will make distribution assumptions regarding $\mathbf{Y}$ to test hypotheses of model fit and to obtain maximum likelihood estimates of the model parameters.

Expanding (8.9.1) into a system of linear equations, the model equations became

$$\begin{array}{rcccccccc} Y_1 - \mu_1 & = & \lambda_{11} f_1 & + & \lambda_{12} f_2 & + \ldots + & \lambda_{1k} f_k & + & e_1 \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ Y_i - \mu_i & = & \lambda_{i1} f_1 & + & \lambda_{i2} f_2 & + \ldots + & \lambda_{ik} f_k & + & e_i \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ Y_p - \mu_p & = & \lambda_{p1} f_1 & + & \lambda_{p2} f_2 & + \ldots + & \lambda_{pk} f_k & + & e_p \end{array} \tag{8.9.5}$$

where the $\text{cov}(e_i, e_j) = 0$ for $i \neq j$, and the λ_{ij} are regression coefficients. For psychologists, the observations Y_i are usually test scores, the λ_{ij} are termed factor loadings, and the residuals e_i are called unique or specific factors. Thus, each test score or observation about the mean is represented as two parts, a common part and a unique part

$$Y_i - \mu_i = c_i + e_i \tag{8.9.6}$$

EFA is used to investigate the unobserved common parts

$$c_i = \lambda_{i1} f_1 + \lambda_{i2} f_2 + \ldots + \lambda_{ik} f_k \tag{8.9.7}$$

whereas PCA is used to study the observed observations Y_i. The unique factor may be regarded as a sum of an error of measurement and a specific factor where the specific factor is associated with a particular test; however, since these do not contribute to any correlation or covariance among test scores, they are omitted in our discussion (see Harmon, 1976).

Corresponding to the structure of Σ given in (8.9.2), Σ is also partitioned into two parts, the common part and the unique part. The variance of a random observation Y_i is

$$\begin{aligned} \sigma_{ii} \equiv \sigma_i^2 &= \lambda_{i1}^2 + \lambda_{i2}^2 + \ldots + \lambda_{ik}^2 + \psi_i \\ \text{var}\,(Y_i) &= \text{var}\,(c_i) + \text{var}\,(e_i) \\ &= h_i^2 + \psi_i \end{aligned} \tag{8.9.8}$$

The variance of the common part of Y_i is represented by h_i^2 and is called the common variance or communality of the response and the var $(e_i) = \psi_i$, the ith diagonal element of Ψ, is termed the unique, or specific variance or the uniqueness of Y_i. The uniqueness is that part of the total variance not accounted for by the common factors, while the communality is that portion of the variance attributed to the common factors.

The covariance between Y_i and Y_j, for $i \neq j$, is

$$\sigma_{ij} = \lambda_{i1}\lambda_{j1} + \lambda_{i2}\lambda_{j2} + \ldots + \lambda_{ik}\lambda_{jk} \tag{8.9.9}$$

where λ_{ij} is the covariance between Y_i and the jth common factor f_j. Using matrix notation, the

$$\begin{aligned} \text{cov}\,(\mathbf{Y}, \mathbf{f}) &= \text{cov}\,(\Lambda\mathbf{f} + \mathbf{e}, \mathbf{f}) \\ &= \Lambda\,\text{cov}\,(\mathbf{f}, \mathbf{f})\,\Lambda' + \text{cov}\,(\mathbf{e}, \mathbf{f}) \\ &= \Lambda\Lambda' \end{aligned} \tag{8.9.10}$$

since the cov $(\mathbf{e}, \mathbf{f}) = \mathbf{0}$. If standardized variables are analyzed, Σ is a correlation matrix so that (8.9.2) becomes

$$\mathbf{P}_\rho = \Lambda\Lambda' + \Psi \tag{8.9.11}$$

and the loadings λ_{ij} become correlations. Then (8.9.9) becomes

$$\rho_{ij} = \lambda_{i1}\lambda_{j1} + \ldots + \lambda_{ik}\lambda_{jk} \tag{8.9.12}$$

where the diagonal elements of $\mathbf{P}_\rho$ are 1 so that (8.9.8) is written as

$$\begin{aligned}\psi_i &= 1 - h_i^2 \\ &= 1 - \left(\lambda_{i1}^2 + \lambda_{i2}^2 + \ldots + \lambda_{ik}^2\right)\end{aligned} \tag{8.9.13}$$

The quantities $h_i^2 = \sum_{j=1}^{k} \lambda_{ik}^2$, the inner product of the ith row of Λ with itself, are communalities. They represent the common variance of a variable accounted for by a factor solution.

In our discussion of the EFA model we have assumed that the common factors are orthogonal or uncorrelated as in PCA since the cov $(\mathbf{f}) = \mathbf{I}$. A more general model permits the common factors to be correlated so that the cov $(\mathbf{f}) = \Phi$ where $\Phi \neq \mathbf{I}$. Then Σ has the structure

$$\Sigma = \Lambda \Phi \Lambda' + \Psi \tag{8.9.14}$$

and the resulting common factors are said to be oblique.

A basic property of the EFA model with the structure given by (8.9.14) or (8.9.2) is that the model is scale free under affine transformations of the form $\mathbf{Z} = \mathbf{AY} + \mathbf{b}$ where $\mathbf{A}$ is nonsingular. To see this, observe that the

$$\begin{aligned}\text{cov}\,(\mathbf{Z}) &= \mathbf{A}\,\text{cov}\,(\mathbf{Y})\,\mathbf{A}' \\ &= \mathbf{A}\left(\Lambda \Phi \Lambda'\right)\mathbf{A}' + \mathbf{A}\Psi\mathbf{A}' \\ &= (\mathbf{A}\Lambda)\,\Phi\,(\mathbf{A}\Lambda)' + \mathbf{A}\Psi\mathbf{A}' \\ &= \Lambda_* \Phi \Lambda_*' + \Psi_* \\ &= \Sigma_*\end{aligned}$$

Thus, an affine transformation of the original variables is also applied to Λ and Ψ. This scale free property of the EFA model allows one to analyze either Σ or $\mathbf{P}_\rho$. Even though the model is scale free, some early procedures used to estimate model parameters are not scale free and should be avoided, Harris (1964).

We'll work with a built-in R dataset containing personality assessment data. This will serve as a hands-on example to showcase how EFA can help discover the underlying patterns in personality traits.

1) Packages & Sample Dataset
2) Data Exploration & Preparation
3) Correlation Matrix Computation
4) Factor Number Estimation
5) Exploratory Factor Analysis
6) Reliability Check of Item Groups
7) Visualisation of Results

Next, we need to load the respective libraries.

```
library(psych) # load libraries
library(nFactors)
library(reshape2)
library(ggplot2)
library(dplyr)
library(GPArotation)
library(corrplot)
library(Hmisc)
```

Once all libraries are set, we can load our sample data. We will use a built-in dataset called bfi, which includes 25 personality related self-report items taken from the International Personality Item Pool as part of the Synthetic Aperture Personality Assessment (SAPA) web based personality assessment project.

The columns in the original `bfi` dataset have predefined names that relate directly to specific personality traits. However, to simulate a real-world scenario where you work with an unstructured dataset, I will rename the columns to have a more generic format: Q1, Q2, … Q25.

```
names(mydata) <- paste0("Q", 1:25) # rename columns

head(mydata) # print mydata
#       Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19 Q20 Q21 Q22 Q23 Q24 Q25
# 61617  2  4  3  4  4  2  3  3  4   4   3   3   3   4   4   3   4   2   2   3   3   6   3   4   3
# 61618  2  4  5  2  5  5  4  4  3   4   1   1   6   4   3   3   3   3   5   5   4   2   4   3   3
# 61620  5  4  5  4  4  4  5  4  2   5   2   4   4   4   5   4   5   4   2   3   4   2   5   5   2
# 61621  4  4  6  5  5  4  4  3  5   5   5   3   4   4   4   2   5   2   4   1   3   3   4   3   5
# 61622  2  3  3  4  5  4  4  5  3   2   2   2   5   4   5   2   3   4   4   3   3   3   4   3   3
# 61623  6  6  5  6  5  6  6  6  1   3   2   1   6   5   6   3   5   2   2   3   4   3   5   6   1
```

Data Exploration & Preparation

```
describe(mydata)
#     vars    n mean   sd median trimmed  mad min max range  skew kurtosis   se
# Q1     1 2784 2.41 1.41      2    2.23 1.48   1   6     5  0.83    -0.31 0.03
# Q2     2 2773 4.80 1.17      5    4.98 1.48   1   6     5 -1.12     1.05 0.02
# Q3     3 2774 4.60 1.30      5    4.79 1.48   1   6     5 -1.00     0.44 0.02
# Q4     4 2781 4.70 1.48      5    4.93 1.48   1   6     5 -1.03     0.04 0.03
# Q5     5 2784 4.56 1.26      5    4.71 1.48   1   6     5 -0.85     0.16 0.02
# Q6     6 2779 4.50 1.24      5    4.64 1.48   1   6     5 -0.85     0.30 0.02
# Q7     7 2776 4.37 1.32      5    4.50 1.48   1   6     5 -0.74    -0.14 0.03
# Q8     8 2780 4.30 1.29      5    4.42 1.48   1   6     5 -0.69    -0.13 0.02
# Q9     9 2774 2.55 1.38      2    2.41 1.48   1   6     5  0.60    -0.62 0.03
# Q10   10 2784 3.30 1.63      3    3.25 1.48   1   6     5  0.07    -1.22 0.03
# Q11   11 2777 2.97 1.63      3    2.86 1.48   1   6     5  0.37    -1.09 0.03
# Q12   12 2784 3.14 1.61      3    3.06 1.48   1   6     5  0.22    -1.15 0.03
# Q13   13 2775 4.00 1.35      4    4.07 1.48   1   6     5 -0.47    -0.47 0.03
# Q14   14 2791 4.42 1.46      5    4.59 1.48   1   6     5 -0.82    -0.30 0.03
# Q15   15 2779 4.42 1.33      5    4.56 1.48   1   6     5 -0.78    -0.09 0.03
# Q16   16 2778 2.93 1.57      3    2.82 1.48   1   6     5  0.37    -1.01 0.03
# Q17   17 2779 3.51 1.53      4    3.51 1.48   1   6     5 -0.08    -1.05 0.03
# Q18   18 2789 3.22 1.60      3    3.16 1.48   1   6     5  0.15    -1.18 0.03
# Q19   19 2764 3.19 1.57      3    3.12 1.48   1   6     5  0.20    -1.09 0.03
# Q20   20 2771 2.97 1.62      3    2.85 1.48   1   6     5  0.37    -1.06 0.03
# Q21   21 2778 4.82 1.13      5    4.96 1.48   1   6     5 -0.90     0.43 0.02
# Q22   22 2800 2.71 1.57      2    2.56 1.48   1   6     5  0.59    -0.81 0.03
# Q23   23 2772 4.44 1.22      5    4.56 1.48   1   6     5 -0.77     0.30 0.02
# Q24   24 2786 4.89 1.22      5    5.10 1.48   1   6     5 -1.22     1.08 0.02
# Q25   25 2780 2.49 1.33      2    2.34 1.48   1   6     5  0.74    -0.24 0.03
```

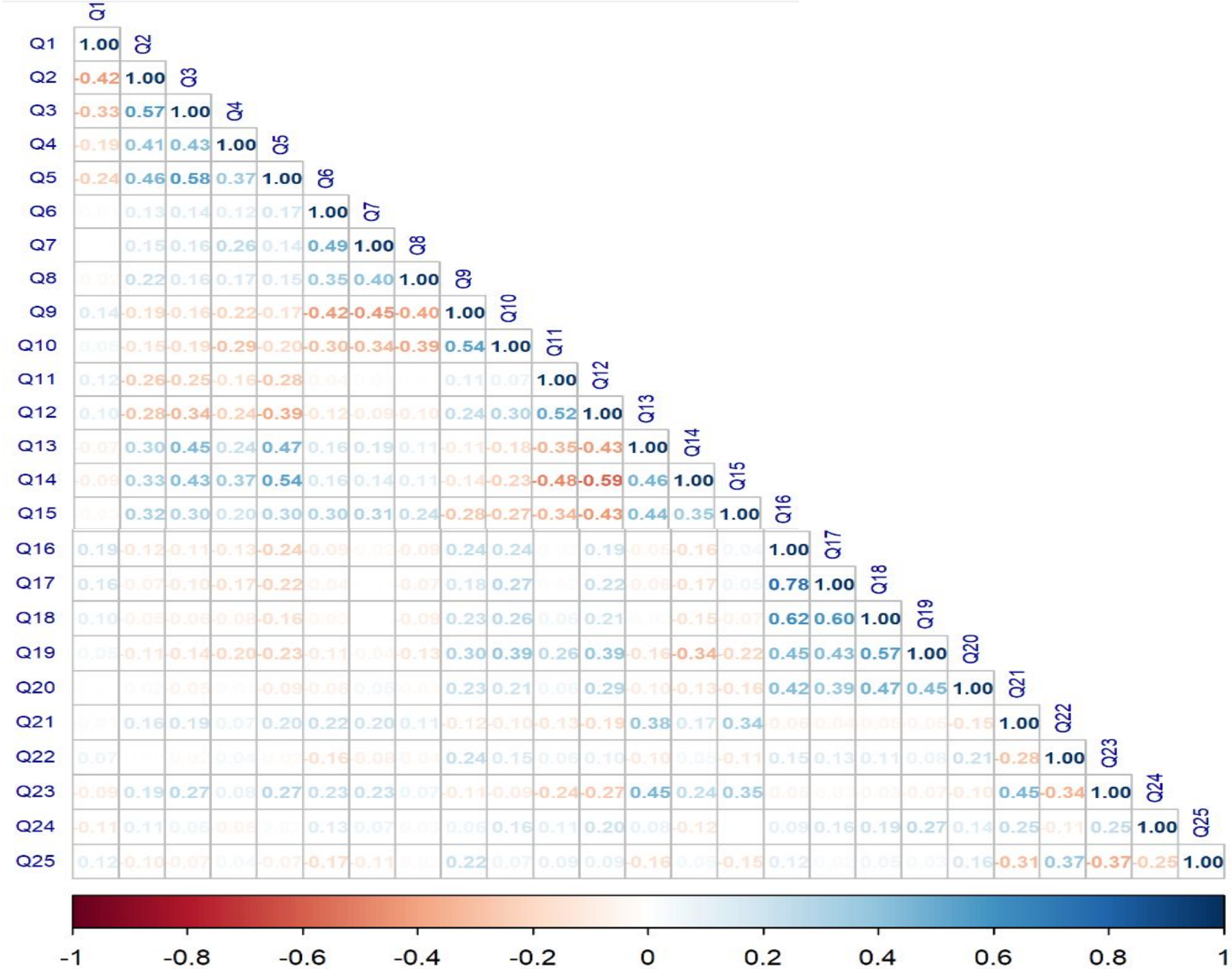
Correlation Matrix Computation

First, we will use the polychoric() function of the psych package for the computation. After that, we will retrieve the correlation coefficients to visualize the results using the corrplot() function of the corrplot package.

```
polychoric_corr <- polychoric(mydata_nm)$rho # # compute correlations

corrplot(polychoric_corr, # visualize correlations
         type = "lower",
         method = "number",
         tl.col = "darkblue",
         number.cex = 0.6,
         tl.cex = 0.6,
         col = COL2('RdBu'))
```

As seen, the `rho` element storing the correlations was retrieved and parsed to the `corrplot()` function. In the function, certain visual elements, like matrix type `type`, were also specified. If you are interested different visual designs,


Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19 Q20 Q21 Q22 Q23 Q24 Q25
-1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1

The subsequent questions have either a strong positive or a strong negative correlation. Also, the question group Q2-Q5 considerably correlate with the question group Q12-Q15. These significant associations are indicators of hidden potential factors in the dataset.

Let's take this analysis to the next level by computing the KMO statistic!

```
KMO(polychoric_corr) # check factorability
# Kaiser-Meyer-Olkin factor adequacy
# Call: KMO(r = polychoric_corr)
# Overall MSA =  0.86
# MSA for each item =
# Q1   Q2   Q3   Q4   Q5   Q6   Q7   Q8   Q9  Q10  Q11  Q12  Q13  Q14  Q15  Q16  Q17  Q18  Q19  Q20  Q21  Q22  Q
# 0.77 0.84 0.88 0.88 0.91 0.86 0.80 0.86 0.84 0.87 0.84 0.89 0.90 0.88 0.90 0.78 0.78 0.87 0.89 0.86 0.88 0.80
# Q24  Q25
# 0.78 0.79
```

The KMO statistic assesses the adequacy of the data for factor analysis. A result that yields values exceeding 0.8 for each variable, and 0.86 for the complete dataset, suggests that the data is suitable for this analysis.

Factor Number Estimation

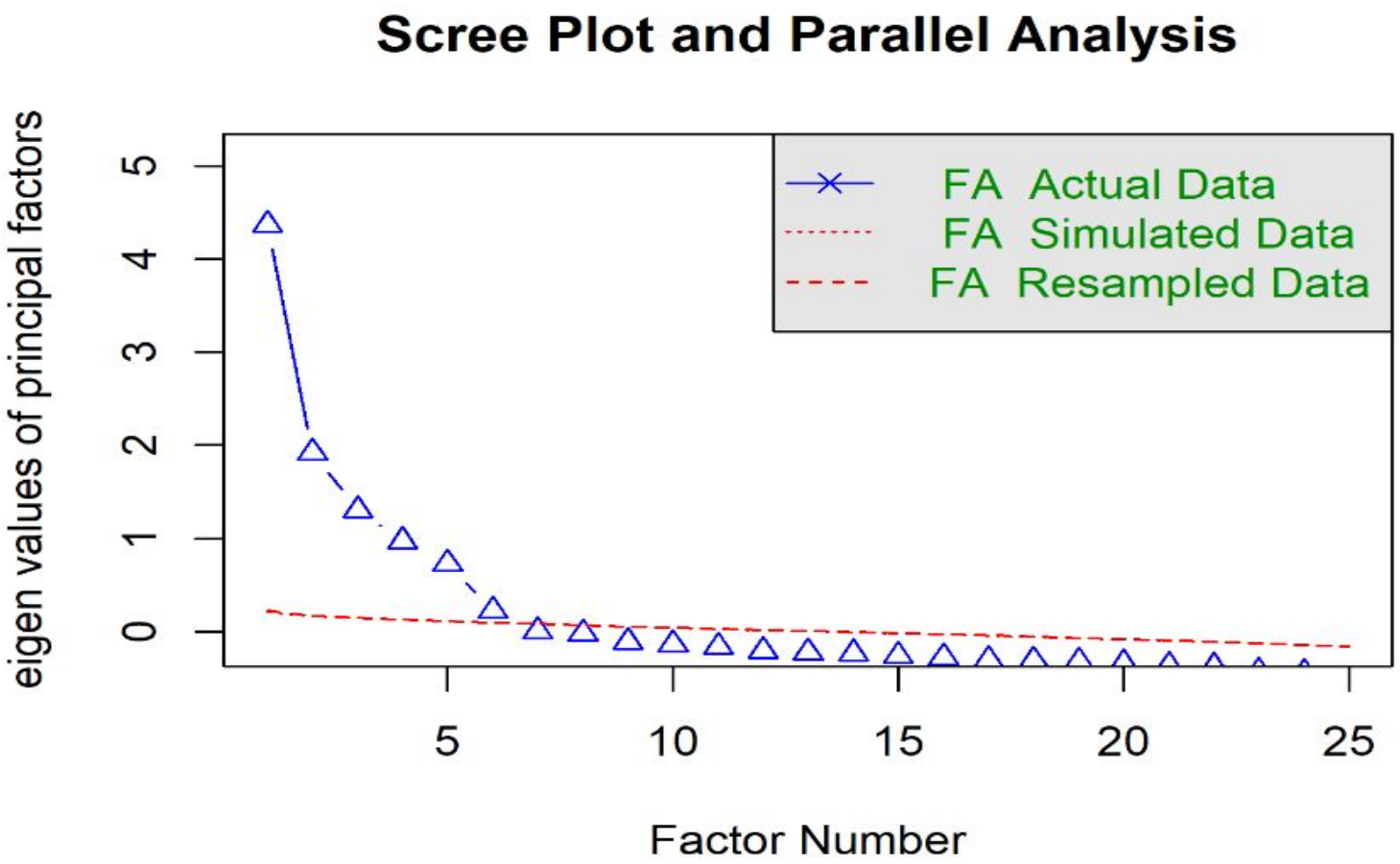
There are different factor selection methods that can be considered. In this tutorial, we will use relatively sophisticated methods called parallel analysis (PA) and minimum average partial (MAP) test.

We will use the fa.parallel() function of the psych package to return a PA solution with a graphical representation.

```
set.seed(1234) # set seed for reproducability

fa.parallel(mydata_nm, # perform parallel analysis
            fa = "fa",
            fm = "pa",
            show.legend = TRUE,
            main = "Scree Plot and Parallel Analysis")
# Parallel analysis suggests that the number of factors =  6  and the number of components =  NA
```

As you can see, the analysis suggests six factors. Since we requested only the factor computation by specifying `fa = "fa"`, there is no suggestion regarding the principal component number.



Here, we observe the visual representation of the six factor solution. The simulated data eigenvalues and resampled eigenvalues overlap, which makes the corresponding lines be seen as a single line. This overlap is a result of using a polychoric matrix, see the function documentation.

Exploratory Factor Analysis

Let's start with the six factor solution. All we need is to use the fa() function parsing the specified correlation matrix, rotation and extraction method, and the number of factors.

```
efa_promax6 <- fa(polychoric_corr, # perform EFA with 6 factors
                        nfactors=6,
                        rotate="promax",
                        fm="pa")
```

As seen, we input our polychoric correlation matrix `polychoric_corr` and number of factors `6`, and set the rotation to `"promax"` and the extraction method to principal axis factoring `"pa"`.

Let's print the result showing factor loadings above 0.40, which is an arbitrary yet common threshold. You can also print the result without applying any threshold; it's just for the sake of interpretation.

```
print(efa_promax6, # print efa_promax6
      cut = 0.4)

# Factor Analysis using method =  pa
# Call: fa(r = polychoric_corr, nfactors = 6, rotate = "promax", fm = "pa")
# Standardized loadings (pattern matrix) based upon correlation matrix
#     PA1   PA2   PA3   PA4   PA5   PA6   h2   u2 com
# Q1                         -0.69       0.40 0.60 1.3
# Q2                          0.70       0.60 0.40 1.2
# Q3                          0.53       0.59 0.41 1.9
# Q4                                     0.35 0.65 3.1
# Q5                                     0.55 0.45 3.5
```

```
# Q6                0.63                     0.41 0.59 1.2
# Q7                0.79                     0.55 0.45 1.1
# Q8                0.65                     0.35 0.65 1.1
# Q9               -0.72               0.48 0.66 0.34 1.9
# Q10              -0.59                     0.48 0.52 1.6
# Q11 -0.73                                  0.44 0.56 1.4
# Q12 -0.78                                  0.62 0.38 1.1
# Q13  0.55                                  0.54 0.46 2.1
# Q14  0.70                                  0.63 0.37 1.4
# Q15  0.54                                  0.46 0.54 1.9
# Q16        0.94                            0.76 0.24 1.2
# Q17        0.93                            0.71 0.29 1.2
# Q18        0.76                            0.59 0.41 1.0
# Q19        0.49                            0.55 0.45 2.2
# Q20        0.51                            0.40 0.60 1.8
# Q21                    -0.49               0.42 0.58 2.2
# Q22                     0.54               0.34 0.66 1.2
# Q23                    -0.56               0.54 0.46 2.1
# Q24                    -0.41               0.35 0.65 3.1
# Q25                     0.65               0.43 0.57 1.1
#
#                        PA1  PA2  PA3  PA4  PA5  PA6
# SS loadings           2.95 2.92 2.34 1.76 1.75 1.01
# Proportion Var        0.12 0.12 0.09 0.07 0.07 0.04
# Cumulative Var        0.12 0.23 0.33 0.40 0.47 0.51
# Proportion Explained  0.23 0.23 0.18 0.14 0.14 0.08
```

```
# Cumulative Proportion 0.23 0.46 0.65 0.78 0.92 1.00
#
# With factor correlations of
#       PA1   PA2   PA3   PA4   PA5   PA6
# PA1  1.00 -0.38  0.43 -0.11  0.31  0.12
# PA2 -0.38  1.00 -0.22  0.04 -0.18  0.25
# PA3  0.43 -0.22  1.00 -0.19  0.31  0.25
# PA4 -0.11  0.04 -0.19  1.00 -0.02 -0.07
# PA5  0.31 -0.18  0.31 -0.02  1.00  0.34
# PA6  0.12  0.25  0.25 -0.07  0.34  1.00
#
# Mean item complexity =  1.7
# Test of the hypothesis that 6 factors are sufficient.
#
# df null model =  300  with the objective function =  9.59
# df of  the model are 165  and the objective function was  0.57
#
# The root mean square of the residuals (RMSR) is  0.02
# The df corrected root mean square of the residuals is  0.03
#
# Fit based upon off diagonal values = 0.99
# Measures of factor score adequacy
#                                                   PA1  PA2  PA3  PA4  PA5  PA6
# Correlation of (regression) scores with factors   0.93 0.95 0.91 0.87 0.89 0.83
# Multiple R square of scores with factors          0.87 0.90 0.83 0.75 0.78 0.69
# Minimum correlation of possible factor scores     0.74 0.80 0.67 0.51 0.57 0.39
```

As seen, the factor analysis object provides an extensive output. Eventhough, usually, the factor loadings are the main interest in EFA.

Based on the loadings table, certain questions load sufficiently on certain factors, except for Q9. This implies that some groups of questions may measure a similar concept. For example, Q1, Q2, and Q3 might all pertain to a specific personal trait.

However, it is important to note that the choice of the factor extraction and rotation methods have impact on the outcome. Therefore, it is advised to try out multiple extraction and rotation methods to get the most interpretable result, ideally, where each variable loads highly onto one and only one factor, also called simple structure.

Additionally, you can check the explained variance statistics per factor in the second table. For instance, factors explain a total of 51% (`0.51`) of variance in the dataset. However, we are less interested in the maximum explained variance in EFA, unlike in PCA).

You can observe the correlations of factors in the third table, where you can see some moderate-level correlations. For instance, the correlation between `PA1` and `PA2` it is `-0.38`. This implies that these two personal traits are inversely associated.

Random Samples and the Expected Values of the Sample Mean and Covariance Matrix

In order to study the sampling variability of statistics such as $\bar{\mathbf{x}}$ and $\mathbf{S}_n$ with the ultimate aim of making inferences, we need to make assumptions about the variables whose observed values constitute the data set $\mathbf{X}$.

Suppose, then, that the data have not yet been observed, but we *intend* to collect n sets of measurements on p variables. Before the measurements are made, their values cannot, in general, be predicted exactly. Consequently, we treat them as random variables. In this context, let the (j, k)-th entry in the data matrix be the random variable X_{jk}. Each set of measurements $\mathbf{X}_j$ on p variables is a random vector, and we have the random matrix

$$\underset{(n\times p)}{\mathbf{X}} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1p} \\ X_{21} & X_{22} & \cdots & X_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{np} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1' \\ \mathbf{X}_2' \\ \vdots \\ \mathbf{X}_n' \end{bmatrix} \tag{3-8}$$

A *random sample* can now be defined.

If the row vectors $\mathbf{X}_1', \mathbf{X}_2', \ldots, \mathbf{X}_n'$ in (3-8) represent *independent* observations from a *common* joint distribution with density function $f(\mathbf{x}) = f(x_1, x_2, \ldots, x_p)$, then $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ are said to form a *random sample* from $f(\mathbf{x})$. Mathematically, $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ form a random sample if their joint density function is given by the product $f(\mathbf{x}_1)f(\mathbf{x}_2)\cdots f(\mathbf{x}_n)$, where $f(\mathbf{x}_j) = f(x_{j1}, x_{j2}, \ldots, x_{jp})$ is the density function for the jth row vector.

Two points connected with the definition of random sample merit special attention:

1. The measurements of the p variables in a *single* trial, such as $\mathbf{X}_j' = [X_{j1}, X_{j2}, \ldots, X_{jp}]$, will usually be correlated. Indeed, we expect this to be the case. The measurements from *different* trials must, however, be independent.

2. The independence of measurements from trial to trial may not hold when the variables are likely to drift over time, as with sets of p stock prices or p economic indicators. Violations of the tentative assumption of independence can have a serious impact on the quality of statistical inferences.

Certain conclusions can be reached concerning the sampling distributions of $\overline{\mathbf{X}}$ and $\mathbf{S}_n$ without making further assumptions regarding the form of the underlying joint distribution of the variables. In particular, we can see how $\overline{\mathbf{X}}$ and $\mathbf{S}_n$ fare as point estimators of the corresponding population mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

Result 3.1. Let $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ be a random sample from a joint distribution that has mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Then $\overline{\mathbf{X}}$ is an *unbiased* estimator of $\boldsymbol{\mu}$, and its covariance matrix is

$$\frac{1}{n}\boldsymbol{\Sigma}$$

That is,

$$E(\overline{\mathbf{X}}) = \boldsymbol{\mu} \qquad \text{(population mean vector)}$$

$$\operatorname{Cov}(\overline{\mathbf{X}}) = \frac{1}{n}\boldsymbol{\Sigma} \qquad \left(\begin{array}{c}\text{population variance–covariance matrix}\\ \text{divided by sample size}\end{array}\right) \tag{3-9}$$

For the covariance matrix $\mathbf{S}_n$,

$$E(\mathbf{S}_n) = \frac{n-1}{n}\boldsymbol{\Sigma} = \boldsymbol{\Sigma} - \frac{1}{n}\boldsymbol{\Sigma}$$

Thus,

$$E\left(\frac{n}{n-1}\mathbf{S}_n\right) = \boldsymbol{\Sigma} \tag{3-10}$$

so $[n/(n-1)]\mathbf{S}_n$ is an *unbiased* estimator of $\boldsymbol{\Sigma}$, while $\mathbf{S}_n$ is a *biased* estimator with (bias) $= E(\mathbf{S}_n) - \boldsymbol{\Sigma} = -(1/n)\boldsymbol{\Sigma}$.

(Unbiased) Sample Variance–Covariance Matrix

$$\mathbf{S} = \left(\frac{n}{n-1}\right)\mathbf{S}_n = \frac{1}{n-1}\sum_{j=1}^{n}(\mathbf{X}_j - \bar{\mathbf{X}})(\mathbf{X}_j - \bar{\mathbf{X}})' \qquad (3\text{-}11)$$

Here S, without a subscript, has (i, k)th entry $(n-1)^{-1}\sum_{j=1}^{n}(X_{ji} - \bar{X}_i)(X_{jk} - \bar{X}_k)$.

Generalized Variance

With a single variable, the sample variance is often used to describe the amount of variation in the measurements on that variable. When p variables are observed on each unit, the variation is described by the sample variance–covariance matrix

$$\mathbf{S} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1p} \\ s_{12} & s_{22} & \cdots & s_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ s_{1p} & s_{2p} & \cdots & s_{pp} \end{bmatrix} = \left\{ s_{ik} = \frac{1}{n-1}\sum_{j=1}^{n}(x_{ji} - \bar{x}_i)(x_{jk} - \bar{x}_k) \right\}$$

The sample covariance matrix contains p variances and $\frac{1}{2}p(p-1)$ potentially different covariances. Sometimes it is desirable to assign a *single* numerical value for the variation expressed by **S**. One choice for a value is the determinant of **S**, which reduces to the usual sample variance of a single characteristic when $p = 1$. This determinant[2] is called the *generalized sample variance:*

$$\text{Generalized sample variance} = |\mathbf{S}| \qquad (3\text{-}12)$$

Sample Mean, Covariance, and Correlation as Matrix Operations

We have it that $\bar{x}_i = (x_{1i} \cdot 1 + x_{2i} \cdot 1 + \cdots + x_{ni} \cdot 1)/n = \mathbf{y}_i'\mathbf{1}/n$. Therefore,

$$\bar{\mathbf{x}} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_p \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{y}_1'\mathbf{1}}{n} \\ \dfrac{\mathbf{y}_2'\mathbf{1}}{n} \\ \vdots \\ \dfrac{\mathbf{y}_p'\mathbf{1}}{n} \end{bmatrix} = \frac{1}{n} \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{p1} & x_{p2} & \cdots & x_{pn} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

or

$$\bar{\mathbf{x}} = \frac{1}{n}\mathbf{X}'\mathbf{1} \tag{3-24}$$

That is, $\bar{\mathbf{x}}$ is calculated from the transposed data matrix by postmultiplying by the vector $\mathbf{1}$ and then multiplying the result by the constant $1/n$.

Next, we create an $n \times p$ *matrix of means* by transposing both sides of (3-24) and premultiplying by $\mathbf{1}$; that is,

$$\mathbf{1}\bar{\mathbf{x}}' = \frac{1}{n}\mathbf{1}\mathbf{1}'\mathbf{X} = \begin{bmatrix} \bar{x}_1 & \bar{x}_2 & \cdots & \bar{x}_p \\ \bar{x}_1 & \bar{x}_2 & \cdots & \bar{x}_p \\ \vdots & \vdots & \ddots & \vdots \\ \bar{x}_1 & \bar{x}_2 & \cdots & \bar{x}_p \end{bmatrix} \tag{3-25}$$

Subtracting this result from $\mathbf{X}$ produces the $n \times p$ *matrix of deviations* (residuals)

$$\mathbf{X} - \frac{1}{n}\mathbf{1}\mathbf{1}'\mathbf{X} = \begin{bmatrix} x_{11} - \bar{x}_1 & x_{12} - \bar{x}_2 & \cdots & x_{1p} - \bar{x}_p \\ x_{21} - \bar{x}_1 & x_{22} - \bar{x}_2 & \cdots & x_{2p} - \bar{x}_p \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} - \bar{x}_1 & x_{n2} - \bar{x}_2 & \cdots & x_{np} - \bar{x}_p \end{bmatrix} \tag{3-26}$$

Subtracting this result from $\mathbf{X}$ produces the $n \times p$ *matrix of deviations* (residuals)

$$\mathbf{X} - \frac{1}{n}\mathbf{1}\mathbf{1}'\mathbf{X} = \begin{bmatrix} x_{11} - \bar{x}_1 & x_{12} - \bar{x}_2 & \cdots & x_{1p} - \bar{x}_p \\ x_{21} - \bar{x}_1 & x_{22} - \bar{x}_2 & \cdots & x_{2p} - \bar{x}_p \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} - \bar{x}_1 & x_{n2} - \bar{x}_2 & \cdots & x_{np} - \bar{x}_p \end{bmatrix} \tag{3-26}$$

Now, the matrix $(n-1)\mathbf{S}$ representing sums of squares and cross products is just the transpose of the matrix (3-26) times the matrix itself, or

$$(n-1)\mathbf{S} = \begin{bmatrix} x_{11} - \bar{x}_1 & x_{21} - \bar{x}_1 & \cdots & x_{n1} - \bar{x}_1 \\ x_{12} - \bar{x}_2 & x_{22} - \bar{x}_2 & \cdots & x_{n2} - \bar{x}_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{1p} - \bar{x}_p & x_{2p} - \bar{x}_p & \cdots & x_{np} - \bar{x}_p \end{bmatrix}$$

$$\times \begin{bmatrix} x_{11} - \bar{x}_1 & x_{12} - \bar{x}_2 & \cdots & x_{1p} - \bar{x}_p \\ x_{21} - \bar{x}_1 & x_{22} - \bar{x}_2 & \cdots & x_{2p} - \bar{x}_p \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} - \bar{x}_1 & x_{n2} - \bar{x}_2 & \cdots & x_{np} - \bar{x}_p \end{bmatrix}$$

$$= \left(\mathbf{X} - \frac{1}{n}\mathbf{1}\mathbf{1}'\mathbf{X}\right)'\left(\mathbf{X} - \frac{1}{n}\mathbf{1}\mathbf{1}'\mathbf{X}\right) = \mathbf{X}'\left(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}'\right)\mathbf{X}$$

since

$$\left(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}'\right)'\left(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}'\right) = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}' - \frac{1}{n}\mathbf{1}\mathbf{1}' + \frac{1}{n^2}\mathbf{1}\mathbf{1}'\mathbf{1}\mathbf{1}' = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}'$$

To summarize, the matrix expressions relating $\bar{\mathbf{x}}$ and $\mathbf{S}$ to the data set $\mathbf{X}$ are

$$\bar{\mathbf{x}} = \frac{1}{n}\mathbf{X}'\mathbf{1}$$

$$\mathbf{S} = \frac{1}{n-1}\mathbf{X}'\left(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}'\right)\mathbf{X} \tag{3-27}$$

The result for $\mathbf{S}_n$ is similar, except that $1/n$ replaces $1/(n-1)$ as the first factor.

The relations in (3-27) show clearly how matrix operations on the data matrix $\mathbf{X}$ lead to $\bar{\mathbf{x}}$ and $\mathbf{S}$.

Once $\mathbf{S}$ is computed, it can be related to the sample correlation matrix $\mathbf{R}$. The resulting expression can also be "inverted" to relate $\mathbf{R}$ to $\mathbf{S}$. We first define the $p \times p$ *sample standard deviation matrix* $\mathbf{D}^{1/2}$ and compute its inverse, $(\mathbf{D}^{1/2})^{-1} = \mathbf{D}^{-1/2}$. Let

$$\underset{(p\times p)}{\mathbf{D}^{1/2}} = \begin{bmatrix} \sqrt{s_{11}} & 0 & \cdots & 0 \\ 0 & \sqrt{s_{22}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{s_{pp}} \end{bmatrix} \tag{3-28}$$

Then

$$\underset{(p\times p)}{\mathbf{D}^{-1/2}} = \begin{bmatrix} \frac{1}{\sqrt{s_{11}}} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sqrt{s_{22}}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sqrt{s_{pp}}} \end{bmatrix}$$

Since

$$\mathbf{S} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ s_{1p} & s_{2p} & \cdots & s_{pp} \end{bmatrix}$$

and

$$\mathbf{R} = \begin{bmatrix} \frac{s_{11}}{\sqrt{s_{11}}\sqrt{s_{11}}} & \frac{s_{12}}{\sqrt{s_{11}}\sqrt{s_{22}}} & \cdots & \frac{s_{1p}}{\sqrt{s_{11}}\sqrt{s_{pp}}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{s_{1p}}{\sqrt{s_{11}}\sqrt{s_{pp}}} & \frac{s_{2p}}{\sqrt{s_{22}}\sqrt{s_{pp}}} & \cdots & \frac{s_{pp}}{\sqrt{s_{pp}}\sqrt{s_{pp}}} \end{bmatrix} = \begin{bmatrix} 1 & r_{12} & \cdots & r_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ r_{1p} & r_{2p} & \cdots & 1 \end{bmatrix}$$

we have

$$\mathbf{R} = \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{-1/2} \tag{3-29}$$

Postmultiplying and premultiplying both sides of (3-29) by $\mathbf{D}^{1/2}$ and noting that $\mathbf{D}^{-1/2}\mathbf{D}^{1/2} = \mathbf{D}^{1/2}\mathbf{D}^{-1/2} = \mathbf{I}$ gives

$$\mathbf{S} = \mathbf{D}^{1/2}\mathbf{R}\mathbf{D}^{1/2} \tag{3-30}$$

That is, $\mathbf{R}$ can be obtained from the information in $\mathbf{S}$, whereas $\mathbf{S}$ can be obtained from $\mathbf{D}^{1/2}$ and $\mathbf{R}$. Equations (3-29) and (3-30) are sample analogs of (2-36) and (2-37).

Definition 2.1 *An additive tariff structure is defined by the property that*

$$P_{i_1 i_2 \ldots i_K} = \mu + \sum_{k=1}^{K} \alpha_{k,i_k}, \tag{2.1}$$

where μ *and* α_{k,i_k}, $k = 1, \ldots, K$, $i_k = 1 \ldots, I_k$, *are real numbers.*

We call μ and α_{k,i_k} the parameters of the tariff. The parameter μ might be the average premium (averaged over all cells) or it might refer to the premium of a reference cell, for instance of $cell_{1,1,,\ldots,1}$. In the latter case $\mu = P_{1,1,..,1}$ and hence $\alpha_{k,1} = 0$ for $k = 1, \ldots, K$. This shows that the tariff contains $I_\bullet - K + 1$ free parameters where $I_\bullet = \sum_k I_k$. Here and in the following a dot in the index means summation over the corresponding index.

Definition 2.2 *A multiplicative tariff structure is defined by the property that*

$$P_{i_1 i_2 \ldots i_K} = \mu \cdot \prod_{k=1}^{K} \alpha_{k,i_k}, \tag{2.2}$$

where μ *and* α_{k,i_k}, $k = 1, \ldots, K$, $i_k = 1 \ldots, I_k$, *are real numbers.*

Again, the parameter μ might be the average premium or it might refer to the premium of $cell_{1,1,,\ldots,1}$, in which case $\mu = P_{1,1,..,1}$ and $\alpha_{k,1} = 1$ for $k = 1, \ldots, K$. This shows that the tariff contains $I_\bullet - K + 1$ free parameters.

For simplicity and for didactical reasons, in the following, we will only consider the case of two rating factors. However all results can be extended in an obvious way to any number of rating factors. We will denote the two rating factors by A and B, the corresponding levels by $i = 1, \ldots, I$ and $j = 1, \ldots, J$, the observable random variables by X_{ij}, the associated exposure measures by w_{ij} and the parameters of the tariff by μ, ψ_i and φ_j. The additive tariff is then given by

$$P_{ij} = \mu + \psi_i + \varphi_j$$

and the multiplicative one by

$$P_{ij} = \mu \cdot \psi_i \cdot \varphi_j.$$

Model-Assumptions 3.1 (classical additive model) *The observable random variables* X_{ij}, $i = 1, 2, \ldots, I$ *and* $j = 1, 2, \ldots, J$, *satisfy*

$$X_{ij} = \mu_0 + \psi_i + \varphi_j + \varepsilon_{ij},$$

where ε_{ij} *are independent random variables with*

$$E[\varepsilon_{ij}] = 0, \tag{3.1}$$

$$\mathrm{Var}\,(\varepsilon_{ij}) = \frac{\sigma^2}{w_{ij}}, \tag{3.2}$$

where μ_0, ψ_i, φ_j *are real numbers.*

The aim is to estimate for each $cell_{ij}$ the corresponding premium

$$P_{ij} = E\left[X_{ij}\right] = \mu_0 + \psi_i + \varphi_j. \tag{3.3}$$

Remark:

- The parameters μ_0, ψ_i, φ_j are determined only up to an additive constant. They are uniquely defined if we fix μ_0. In the following μ_0 is referred to as the average premium over the portfolio. This is equivalent to the side constraint

$$\sum_{i=1}^{I} w_{i\bullet}\psi_i = \sum_{j=1}^{J} w_{\bullet j}\varphi_j = 0. \tag{3.4}$$

Model-Assumptions 5.1 (classical multiplicative model) *The observable random variables X_{ij}, $i = 1, 2, \ldots, I$ and $j = 1, 2, \ldots, J$, satisfy*

$$E[X_{ij}] = \mu_0 \cdot \psi_i \cdot \varphi_j, \tag{5.1}$$

where ψ_i and φ_j are real numbers.

The aim is again to estimate for each $cell_{ij}$ the corresponding premium

$$P_{ij} = E[X_{ij}] = \mu_0 \cdot \psi_i \cdot \varphi_j.$$

Remark:

- The tariff parameters μ_0, ψ_i and φ_j are determined only up to a constant factor, i.e. if we multiply the ψ_i with a constant factor c_1 and the φ_j with a constant factor c_2 and then replace μ_0 by $\mu_0^* = \mu_0/c_1c_2$, we get the same tariff.
- To make the tariff parameters uniquely defined we have to fix μ_0. In the following, μ_0 is referred to as the average premium, i.e.

$$\mu_0 = E\left[\overline{X}_{\bullet\bullet}\right], \tag{5.2}$$

Decomposition models

The general mathematical representation of the decomposition approach is:

$$Y_t = f(S_t, T_t, E_t) \tag{3.1}$$

where Y_t is the time series value (actual data) at period t,
S_t is the seasonal component (or index) at period t,
T_t is the trend-cycle component at period t, and
E_t is the irregular (or remainder) component at period t.

The exact functional form of (3.1) depends on the decomposition method actually used. A common approach is to assume equation (3.1) has the additive form

$$Y_t = S_t + T_t + E_t.$$

That is, the seasonal, trend-cycle and irregular components are simply added together to give the observed series.

Alternatively, the multiplicative decomposition has the form

$$Y_t = S_t \times T_t \times E_t.$$

That is, the seasonal, trend-cycle and irregular components are multiplied together to give the observed series.

An additive model is appropriate if the magnitude of the seasonal fluctuations does not vary with the level of the series. But if the seasonal fluctuations increase and decrease proportionally with increases and decreases in the level of the series, then a multiplicative model is appropriate. Multiplicative decomposition is more prevalent with economic series because most seasonal economic series do have seasonal variation which increases with the level of the series. (See, for example, the electricity production data in Figure 2-10.)

Rather than choosing either an additive or multiplicative decomposition, we could use a transformation as discussed in Section 2/5/1. Very often the transformed series can be modeled additively, when the original data are not additive. Logarithms, in particular, turn a multiplicative relationship into an additive relationship, since if

$$Y_t = S_t \times T_t \times E_t,$$

$$\text{then} \qquad \log Y_t = \log S_t + \log T_t + \log E_t.$$

Act

So we can fit a multiplicative relationship by fitting an additive relationship to the logarithms of the data. Other transformations allow a decomposition which is somewhere between the additive and multiplicative forms.

A further decomposition method is *pseudo-additive decomposition* which takes the form

$$Y_t = T_t(S_t + E_t - 1).$$

This type of decomposition is useful in series where there is one month (or quarter) that is much higher or lower than all the other months

By averaging these estimates of the monthly effects for each month, we obtain a single estimate of the effect for each month. If the period of the time series is a whole number of years, the number of monthly effects averaged for each month is one less than the number of years of record. At this stage, the twelve monthly additive components should have an average value close to, but not usually exactly equal to, zero. It is usual to adjust them by subtracting this mean so that they do average zero. If the monthly effect is multiplicative, the estimate is given by division; i.e., $\hat{s}_t = x_t/\hat{m}_t$. It is usual to adjust monthly multiplicative factors so that they average unity. The procedure generalises, using the same principle, to any seasonal frequency.

2.1 Factor analytic models

A model concerning the evaluation of several treatments or genotypes in several environments is given by:

$$Y_{ij} = \mu + g_i + e_j + ge_{ij} + \varepsilon_{ij},$$

where: μ, g, e, ge and ε are the fixed constant, genotype, environment, genotype x environment interaction and within environment error effects, respectively. The μ and e effects can be regarded as fixed and the others as random. A model referring to random genotype effects in each environment can be written as:

$$Y_{ij} = \mu + g_{ij} + e_j + \varepsilon_{ij}.$$

In the context of MET data, the factor analysis approach can be used to provide a class of structures for the variance-covariance matrix of g_{ij} (G). The model is postulated in terms of the unobservable genotype effects in different environments:

$$g_{ij} = \sum_{r=1}^{k} \lambda_{jr} f_{ir} + \delta_{ij} \ ,$$

where:

g_{ij} : effect of genotype i in environment j;

λ_{jr} : loading for factor r in environment j;

f_{ir} : score for genotype i in factor r;

δ_{ij} : error representing the lack of fit of the model.

In a compact way, the model is:

$$g_s = (\Lambda \otimes I_g) f + \delta ,$$

where:

$\Lambda^{(s\,x\,k)} = [\lambda_1 \ldots\ldots \lambda_k]$;

$f^{(gk\,x1)} = (f_1^{'}, f_2^{'} \ldots\ldots f_k^{'})'$.

The joint distribution of f *and* δ is given by

$$\begin{pmatrix} f \\ \delta \end{pmatrix} \sim N\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} I_k \otimes I_g & 0 \\ 0 & \Psi \otimes I_g \end{pmatrix}\right],$$

where:

$\Psi = \mathrm{diag}(\psi_1 \ldots \psi_p)$;

ψ_i : specific variance for the *i*th trial.

The variance matrix for genotype effects on environments is given by

$$\mathrm{var}(g_s) = (\Lambda \otimes I_g)\,\mathrm{var}(f)\,(\Lambda' \otimes I_g) + \mathrm{var}(\delta) = (\Lambda\Lambda' + \Psi) \otimes I_g .$$

The model for genotype effects in each environment leads to a model for G in which:

$\sigma_{g_{jj'}} = \sum_{r=1}^{k} \lambda_{jr}^2 + \psi_j$: genotype variance in environment *j*;

$\sigma_{g_{jj'}} = \sum_{r=1}^{k} \lambda_{jr}\lambda_{j'r}$: genotype covariance between environments *j* and *j*';

$\rho_{g_{jj'}} = \sum_{r=1}^{k} \lambda_{jr}\lambda_{j'r} / [(\sum_{r=1}^{k} \lambda_{jr}^2 + \psi_j)(\sum_{r=1}^{k} \lambda_{j'r}^2 + \psi_{j'})]^{1/2}$: genotype correlation between environments *j* and *j*'

The equation for g_s has the form of a (random) regression on k environmental covariates $\lambda_1 \ldots \lambda_k$ in which all regressions pass through the origin. It may be more appropriate to allow a separate (non-zero) intercept for each genotype. This is equivalent to the model with genotype main effects, g, and a k-factor analytic model for g x e interaction. Then, the expression for g_s turns to

$$g_s = (1_s \otimes I_g)g + ge = (1_s \otimes I_g)g + (\Lambda \otimes I_g)f + \delta.$$

Vector g has mean zero and variance $\sigma_g^2 I$ or $\sigma_g^2 A$, where A is a genetic relationship matrix. The model can be written as

$$g_s = (\sigma_g 1_s \otimes I_g)f_0 + (\Lambda \otimes I_g)f + \delta = (\Lambda_g \otimes I_g)f_g + \delta,$$

where:

$\Lambda_g^{s(k+1)} = [\sigma_g 1_s \ \Lambda]; \quad f_0 = g/\sigma_g; \quad f_g' = (f_0' f')$.

Thus the model with genotype main effects and a k-factor analytic model for g x e interactions is a special case of a (k+1)-factor analytic genotype effects in each environment, in which the loadings in the first set are constrained to be equal.

The feature that distinguishes equations for g, from standard random multivariate regression problems is that both the covariates and the regression coefficients are unknown and therefore must be estimated from the data. The model is

then a multiplicative model of environment and genotypes coefficients (known as loadings and factorial scores, respectively).

Multivariate Data Visualization with R

A method for visualizing data with numerous variables is called multivariate data visualization with R. In this method, graphs and charts are made to show how the various factors relate to one another. The programming language R, which is frequently used for data visualization, provides a number of tools for the visualization of multivariate data.

- **Scatterplots:** In multivariate data visualization, scatterplots can be used to visualize the relationship between multiple variables by coloring the points or adding shapes or sizes.
- **Line Graphs:** In multivariate data visualization, Line graphs can be used to compare the trends of two variables or to show the trends of several variables across time.
- **Bar Charts:** In multivariate data visualization, By stacking or grouping the bars, bar charts can be used to compare the values of several category variables.
- **Heatmaps:** Heatmaps are a type of plot used to display the relationship between multiple variables. Heatmaps use colors to represent the values of the variables, and rows and columns represent the different variables.

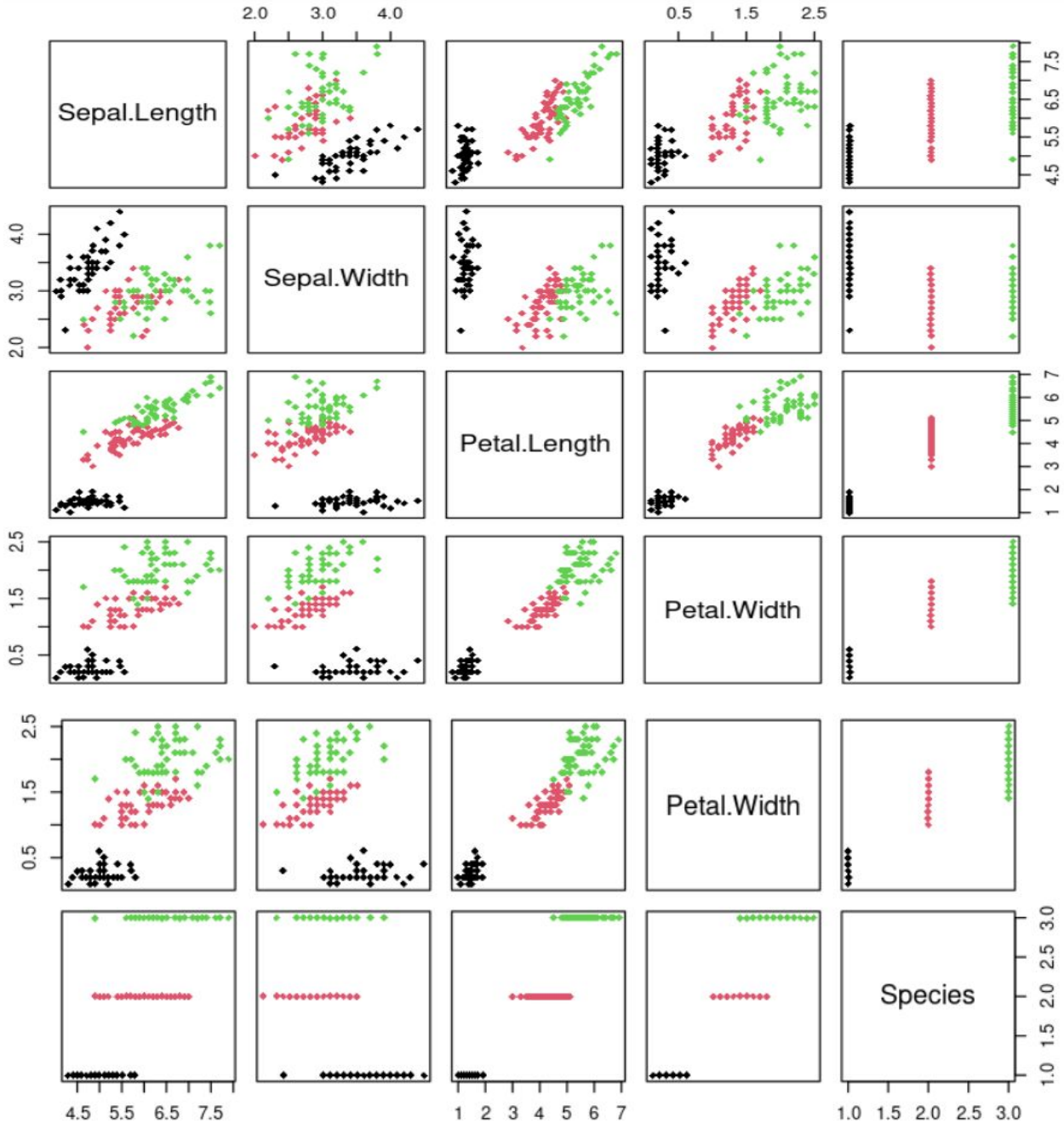
- **Boxplots:** In multivariate data visualization, the distributions of a continuous variable over various categories can be compared using boxplots.
- **Correlation Matrix:** In multivariate data visualization, to see the connections between variables, a heatmap or other graphical display can be used to visualize a correlation matrix.

Overall, Multivariate data visualization's primary objective is to find patterns and relationships in the data and effectively communicate these findings.

Scatterplot Matrix

It is possible to see the pairwise correlations between various variables using a scatterplot matrix. Using the "pairs()" function in R Programming Language, this may be generated. Here is an illustration utilizing the integrated iris dataset.

```
library(datasets)
data(iris)
pairs(iris[, 1:5],
      col = as.integer(iris$Species),
      pch = 18)
```



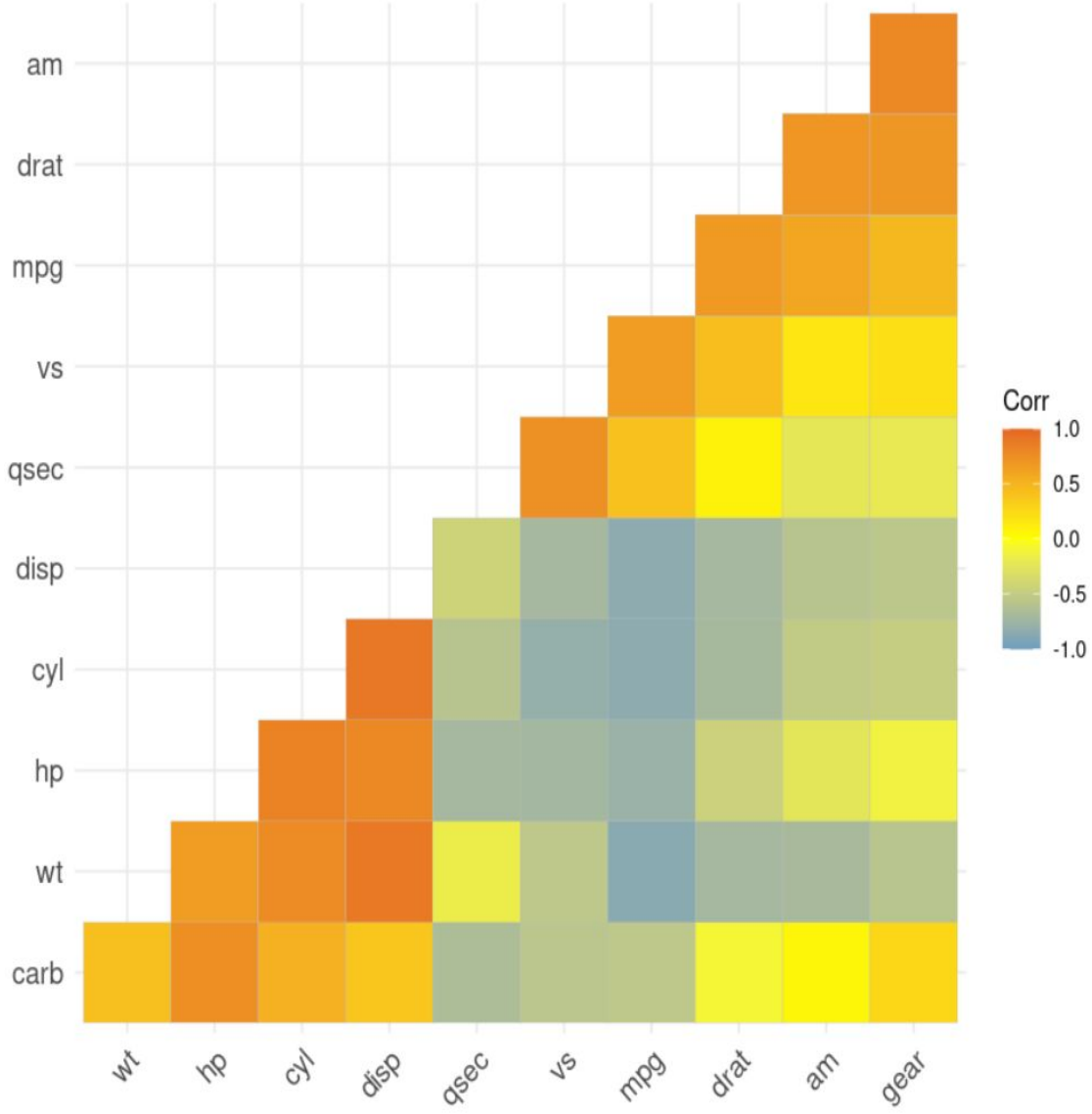
Scatter plot matrix for multivariate data visualizations

Correlation Matrix Heatmap

The pairwise correlations between several variables are shown visually using a correlation matrix heatmap. The “ggcorrplot()” function from the “ggcorrplot” package in R can be used to construct this. Here is an illustration utilizing the integrated “mtcars” dataset:

R

```r
library(ggcorrplot)
data(mtcars)
corr_matrix <- cor(mtcars)
ggcorrplot(corr_matrix, hc.order = TRUE,
           type = "lower",
           colors = c("#6D9EC1",
                      "yellow", "#E46726"))
```

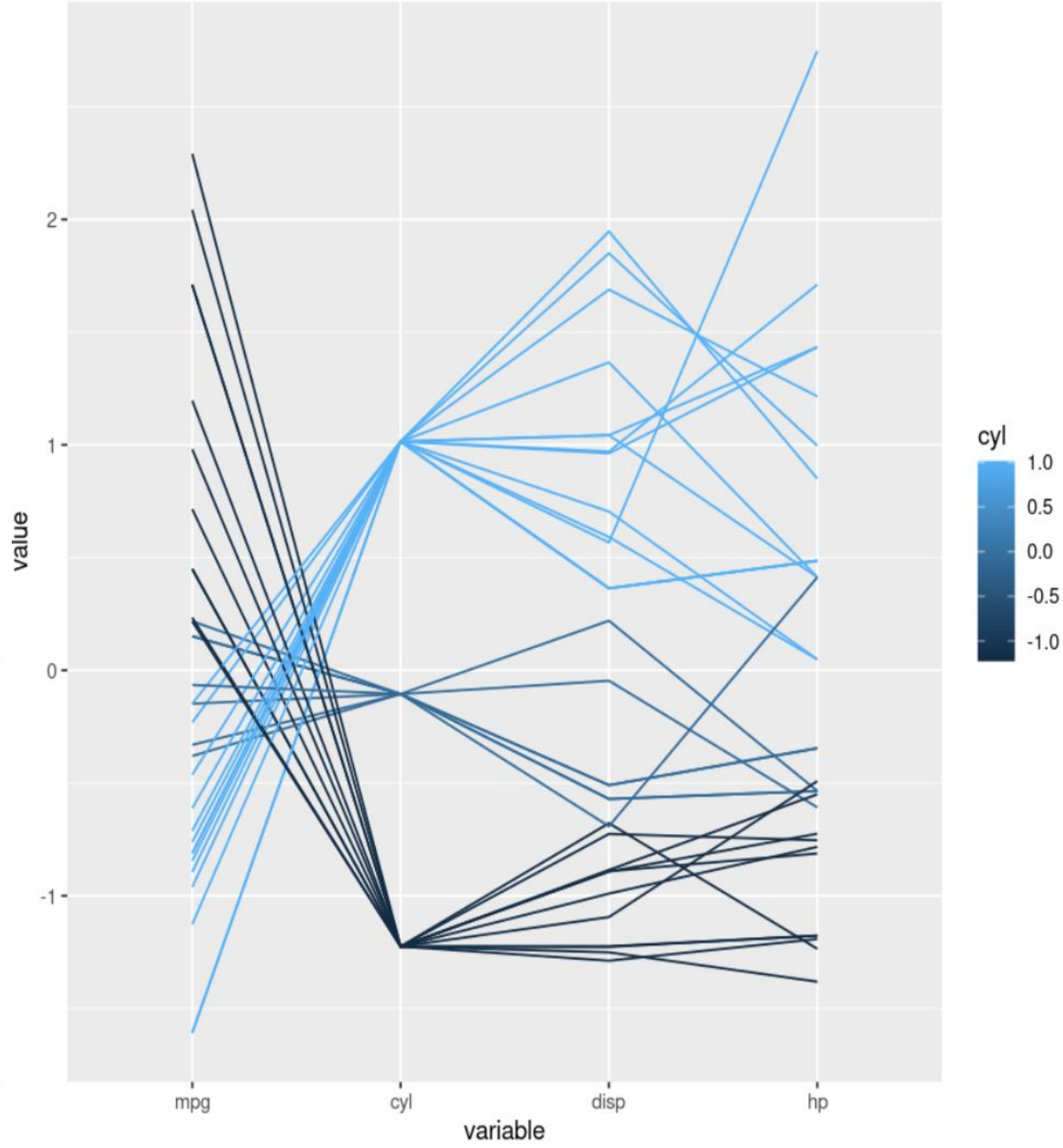


Parallel Coordinate Plot

To see the link between several variables and spot patterns or outliers, utilize a parallel coordinate graphic. Using the “ggparcoord()” function from the “GGally” package in R, this may be generated. Here is an illustration utilizing the integrated “mtcars” dataset:

R

```r
library(GGally)
data(mtcars)
ggparcoord(mtcars, columns = 1:4,
           groupColumn = "cyl",
           alphaLines = 1.0)
```

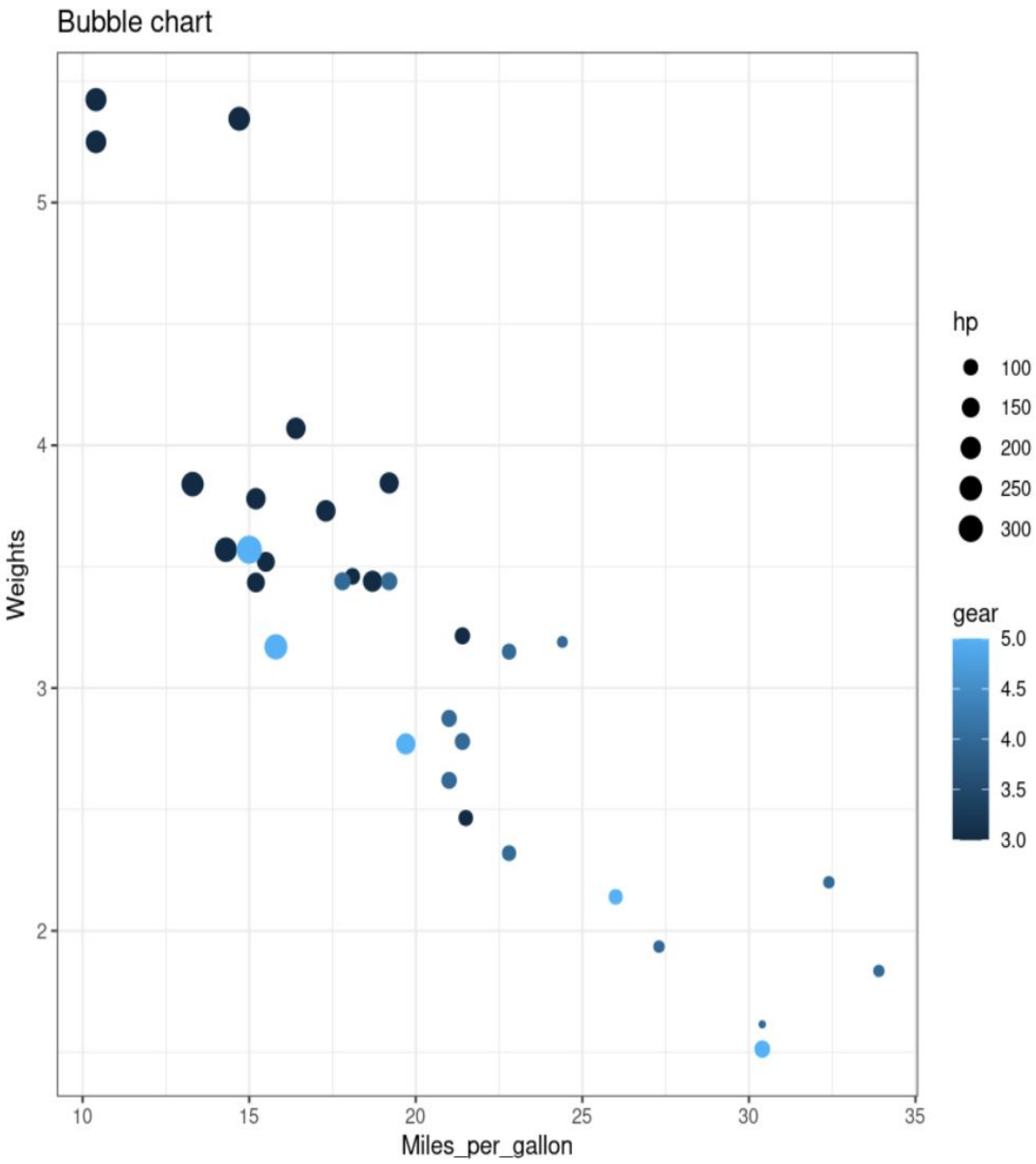


Bubble Chart

The size of the points in a bubble chart, a type of scatterplot, represents a third variable. By doing so, it is possible to simultaneously visualize the relationship between two continuous variables and a categorical variable. A bubble chart can be made in R using the ggplot2 package.

R

```r
library(ggplot2)
ggplot(data = mtcars, aes(x = mpg,
                          y = wt, size = hp,
                          color = gear)) +
  geom_point() +
  scale_size(range = c(1, 5)) +
  labs(title = "Bubble chart",
       x = "Miles_per_gallon",
       y = "Weights") +
  theme_bw()
```



Bubble Chart

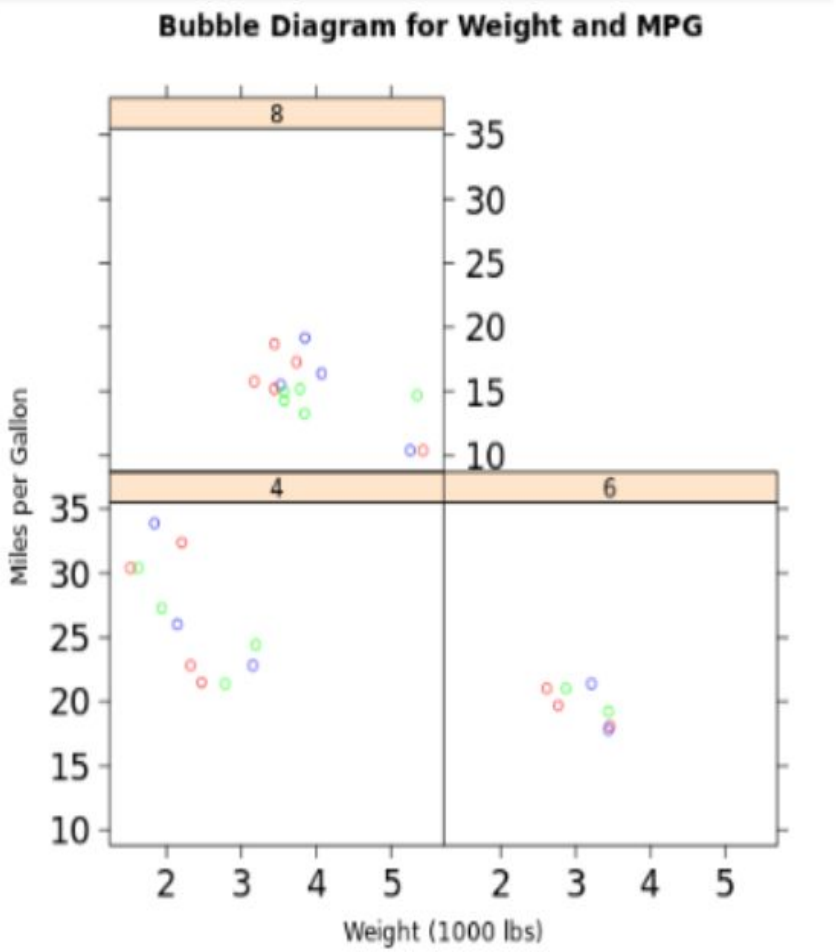
For developing multivariate visualizations in R, the Lattice package is an effective resource. It offers a high-level interface for developing a variety of visualizations and is built on top of the grid graphics system. This is also most widely used for Multivariate data visualization. A few illustrations of multivariate visualizations made with the Lattice program are provided below:

R

```r
# Load the Lattice package
library(lattice)

# Load the mtcars dataset
data(mtcars)

# Create a bubble chart
xyplot(mpg ~ wt | factor(cyl), data=mtcars,
       main="Bubble Diagram for Weight and MPG",
       xlab="Weight (1000 lbs)", ylab="Miles per Gallon",
       alpha=0.8, col=rainbow(nlevels(factor(mtcars$cyl))),
       scales=list(cex=1.5, x=list(alternating=1)),
       par.settings=list(superpose.symbol=list(pch=16),
                         layout.heights=list(bottom.padding=4)))
```



Bubble Chart for multivariate data visualizations using Lattice

Conclusion

The main distinctions that make Lattice better and more popular for multivariate data visualization include the trellis graphics system, flexibility, ease of use, extensive collection of graph kinds, and support for conditional graphics as clearly shown.

Lattice's popularity for multivariate data visualization can also be attributed to its open-source nature, a rich documentation, active user community, compatibility with other R packages, and capacity for handling big datasets. Overall, We may produce visualizations that clearly illustrate the relationships and patterns in our data by mapping variables to various plot aesthetic features.

In multivariate (linear) regression (MR) models, one is not interested in predicting only one dependent variable but rather several dependent random variables $y_1, y_2, \ldots, y_p$. Two possible extensions with regard to the set of independent variables for MR models are (1) the design matrix $\mathbf{X}$ of independent variables is the same for each dependent variable or (2) each dependent variable is related to a different set of independent variables so that p design matrices are permitted. Clearly, situation (1) is more restrictive than (2) and (1) is a special case of (2).

In MR models, the rows of $\mathbf{Y}$ or $\mathbf{E}$, are assumed to be distributed independent MVN so that vec $(\mathbf{E}) \sim N_{np}(\mathbf{0}, \Sigma \otimes \mathbf{I}_n)$. Fitting a model of the form $E(\mathbf{Y}) = \mathbf{XB}$ to the data matrix $\mathbf{Y}$ under MVN, the maximum likelihood (ML) estimate of $\mathbf{B}$ is given in (3.6.20). This ML estimate is identical to the unique best linear unbiased estimator (BLUE) obtained using the multivariate ordinary least squares criterion that the Euclidean matrix norm squared, $\operatorname{tr}\left[(\mathbf{Y} - \mathbf{XB})'(\mathbf{Y} - \mathbf{XB})\right] = \|\mathbf{Y} - \mathbf{XB}\|^2$ is minimized over all parameter matrices $\mathbf{B}$ for fixed $\mathbf{X}$, Seber (1984).

For the MR model $\mathbf{Y} = \mathbf{XB} + \mathbf{E}$, the parameter matrix $\mathbf{B}$ is

$$\mathbf{B} = \begin{bmatrix} \boldsymbol{\beta}_0' \\ --- \\ \mathbf{B}_1 \end{bmatrix} = \begin{bmatrix} \beta_{01} & \beta_{02} & \cdots & \beta_{0p} \\ --- & --- & --- & --- \\ \beta_{11} & \beta_{12} & \cdots & \beta_{1p} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ \beta_{k1} & \beta_{k2} & \cdots & \beta_{kp} \end{bmatrix} \tag{4.2.1}$$

where $q = k + 1$ and is the number of independent variables associated with each dependent variable. The vector $\boldsymbol{\beta}_0'$ contains intercepts while the matrix $\mathbf{B}_1$ contains coefficients associated with independent variables. The matrix $\mathbf{B}$ in (4.2.1) is called the raw score form of the parameter matrix since the elements y_{ij} in $\mathbf{Y}$ have the general form

$$y_{ij} = \beta_{0j} + \beta_{1j} x_{i1} + \ldots + \beta_{kj} x_{ik} + e_{ij} \tag{4.2.2}$$

for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, p$.

To obtain the deviation form of the MR model, the means $\overline{x}_j = \sum_{i=1}^{n} x_{ij}/n$, $j = 1, 2, \ldots, k$ are calculated and the deviation scores $d_{ij} = x_{ij} - \overline{x}_j$, are formed. Then, (4.2.2) becomes

$$y_{ij} = \beta_{0j} + \sum_{h=1}^{k} \beta_{hj}\,\overline{x}_h + \sum_{h=1}^{k} \beta_{hj}\,(x_{ih} - \overline{x}_h) + e_{ij} \tag{4.2.3}$$

Letting

$$\begin{aligned} \alpha_{0j} &= \beta_{0j} + \sum_{h=1}^{k} \beta_{hj}\overline{x}_h \quad j = 1, 2, \ldots, p \\ \boldsymbol{\alpha}_0' &= [\alpha_{01}, \alpha_{02}, \ldots, \alpha_{0p}] \\ \mathbf{B}_1 &= [\beta_{hj}] \quad h = 1, 2, \ldots, k \text{ and } j = 1, 2, \ldots, p \\ \mathbf{X}_d &= [d_{ij}] \quad i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, p \end{aligned} \tag{4.2.4}$$

the matrix representation of (4.2.3) is

$$\underset{n \times p}{\mathbf{Y}} = [\mathbf{1}_n\ \mathbf{X}_d] \begin{bmatrix} \boldsymbol{\alpha}_0' \\ \mathbf{B}_1 \end{bmatrix} + \mathbf{E} \tag{4.2.5}$$

where $\mathbf{1}_n$ is a vector of n $1's$. Applying (3.6.21),

$$\widehat{\mathbf{B}} = \begin{bmatrix} \widehat{\boldsymbol{\alpha}}_0' \\ \widehat{\mathbf{B}}_1 \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{y}}' \\ (\mathbf{X}_d'\mathbf{X}_d)^{-1}\,\mathbf{X}_d'\mathbf{Y} \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{y}}' \\ (\mathbf{X}_d'\mathbf{X}_d)^{-1}\,\mathbf{X}_d\mathbf{Y}_d \end{bmatrix} \tag{4.2.6}$$

where $\mathbf{Y}_d = [y_{ij} - \overline{y}_j]$, and $\overline{y}_j$ is the mean of the jth dependent variable. The matrix $\mathbf{Y}$ may be replaced by $\mathbf{Y}_d$ since the $\sum_j d_{ij} = 0$ for $i = 1, 2, \ldots, n$. This establishes the equivalence of the raw and deviation forms of the MR model since $\widehat{\beta}_{0j} = \overline{y}_j - \sum_{h=1}^{k} \beta_{hj}\overline{x}_j$. Letting the matrix $\mathbf{S}$ be the partitioned sample covariance matrices for the dependent and independent variables

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{yy} & \mathbf{S}_{yx} \\ \mathbf{S}_{xy} & \mathbf{S}_{xx} \end{bmatrix} \tag{4.2.7}$$

and

$$\widehat{\mathbf{B}}_1 = \left(\frac{\mathbf{X}_d'\mathbf{X}_d}{n-1}\right)^{-1}\left(\frac{\mathbf{X}_d'\mathbf{Y}_d}{n-1}\right) = \mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}$$

Because the independent variables are considered to be fixed variates, the matrix $\mathbf{S}_{xx}$ does not provide an estimate of the population covariance matrix. Another form of the MR regression model used in applications is the standard score form of the model. For this form, all dependent and independent variables are standardized to have mean zero and variance one. Replacing the matrix $\mathbf{Y}_d$ with standard scores represented by $\mathbf{Y}_z$ and the matrix $\mathbf{X}_d$ with the standard score matrix $\mathbf{Z}$, the MR model becomes

$$\mathbf{Y}_z = \mathbf{Z}\mathbf{B}_z + \mathbf{E} \tag{4.2.8}$$

and

$$\widehat{\mathbf{B}}_z = \mathbf{R}_{xy}^{-1}\mathbf{R}_{xy} \text{ or } \widehat{\mathbf{B}}_z' = \mathbf{R}_{yx}\mathbf{R}_{xx}^{-1} \tag{4.2.9}$$

where $\mathbf{R}_{xx}$ is a correlation matrix of the fixed $x's$ and $\mathbf{R}_{yx}$ is the sample intercorrelation matrix of the fixed $\mathbf{x}$ and random $\mathbf{y}$ variables. The coefficients in $\widehat{\mathbf{B}}_z$ are called standardized or standard score coefficients. Using the relationships that

$$\begin{aligned} \mathbf{R}_{xx} &= (\operatorname{diag} \mathbf{S}_{xx})^{1/2}\, \mathbf{S}_{xx}\, (\operatorname{diag} \mathbf{S}_{xx})^{1/2} \\ \mathbf{R}_{xy} &= \left(\operatorname{diag} \mathbf{S}_{xy}\right)^{1/2} \mathbf{S}_{xy} \left(\operatorname{diag} \mathbf{S}_{xy}\right)^{1/2} \end{aligned} \tag{4.2.10}$$

$\widehat{\mathbf{B}}_1$ is easily obtained from $\widehat{\mathbf{B}}_z$.

Many regression packages allow the researcher to obtain both raw and standardized coefficients to evaluate the importance of independent variables and their effect on the dependent variables in the model. Because the units of measurement for each independent variable in a MR regression model are often very different, the sheer size of the coefficients may reflect the unit of measurement and not the importance of the variable in the model. The standardized form of the model converts the variables to a scale free metric that often facilitates the direct comparison of the coefficients. As in multiple regression, the magnitude of the coefficients are affected by both the presence of large intercorrelations among the independent variables and the spacing and range of measurements for each of the independent variables. If the spacing is well planned and not arbitrary and the intercorrelations of the independent variables are low so as not to adversely effect the magnitude of the coefficients when variables are added or removed from the model, the standardized coefficients may be used to evaluate the relative simultaneous change in the set Y for a unit change in each X_i when holding the other variables constant.

Having fit a MR model of the form $\mathbf{Y} = \mathbf{XB} + \mathbf{E}$ in (3.6.17), one usually tests hypotheses regarding the elements of $\mathbf{B}$. The most common test is the test of no linear relationship between the two sets of variables or the overall regression test

$$H_1 : \mathbf{B}_1 = \mathbf{0} \tag{4.2.11}$$

Selecting $\mathbf{C}_{k\times q} = [\mathbf{0}, \mathbf{I}_k]$ of full row rank k and $\mathbf{M}_{p\times p} = \mathbf{I}_p$, the test that $\mathbf{B}_1 = \mathbf{0}$ is easily derived from the general matrix form of the hypothesis, $\mathbf{CBM} = \mathbf{0}$. Using (3.6.26) and partitioning $\mathbf{X} = [\mathbf{1}\ \mathbf{X}_2]$ where $\mathbf{Q} = \mathbf{I} - \mathbf{1}\left(\mathbf{1}'\mathbf{1}\right)^{-1}\mathbf{1}'$ then

$$\widehat{\mathbf{B}} = \begin{bmatrix} \widehat{\boldsymbol{\beta}}_0' \\ \widehat{\mathbf{B}}_1 \end{bmatrix} = \begin{bmatrix} \left(\mathbf{1}'\mathbf{1}\right)^{-1}\mathbf{1}'\mathbf{Y} - \left(\mathbf{1}'\mathbf{1}\right)^{-1}\mathbf{1}'\mathbf{X}_2\left(\mathbf{X}_2'\mathbf{Q}\mathbf{X}_2\right)^{-1}\mathbf{X}_2'\mathbf{Q}\mathbf{Y} \\ \left(\mathbf{X}_2'\mathbf{Q}\mathbf{X}_2\right)^{-1}\mathbf{X}_2'\mathbf{Q}\mathbf{Y} \end{bmatrix} \tag{4.2.12}$$

and $\widehat{\mathbf{B}}_1 = \mathbf{X}_2\left(\mathbf{X}_2'\mathbf{Q}\mathbf{X}_2\right)^{-1}\mathbf{X}_2'\mathbf{Q}\mathbf{Y} = \left(\mathbf{X}_d'\mathbf{X}_d\right)^{-1}\mathbf{X}_d'\mathbf{Y}_d$ since $\mathbf{Q}$ is idempotent and

$$\begin{aligned} \mathbf{H} &= \widehat{\mathbf{B}}_1'\left(\mathbf{X}_d'\mathbf{X}_d\right)\widehat{\mathbf{B}}_1 \\ \mathbf{E} &= \mathbf{Y}'\mathbf{Y} - \overline{\mathbf{y}}\,\overline{\mathbf{y}}' - \widehat{\mathbf{B}}_1'\left(\mathbf{X}_d'\mathbf{X}_d\right)\widehat{\mathbf{B}}_1 = \mathbf{Y}_d'\mathbf{Y}_d - \widehat{\mathbf{B}}_1'\left(\mathbf{X}_d'\mathbf{X}_d\right)\widehat{\mathbf{B}}_1 \end{aligned} \tag{4.2.13}$$

so that $\mathbf{E} + \mathbf{H} = \mathbf{T} = \mathbf{Y}'\mathbf{Y} - n\overline{\mathbf{y}}\,\overline{\mathbf{y}}' = \mathbf{Y}_d'\mathbf{Y}_d$ is the total sum of squares and cross products matrix, about the mean. The MANOVA table for testing $\mathbf{B}_1 = \mathbf{0}$ is given in Table 4.2.1.

To test $H_1 : \mathbf{B}_1 = \mathbf{0}$, Wilks' Λ criterion from (3.5.2), is

$$\Lambda = \frac{|\mathbf{E}|}{|\mathbf{E}+\mathbf{H}|} = \prod_{i=1}^{s} (1+\lambda_i)^{-1}$$

TABLE 4.2.1. MANOVA Table for Testing $\mathbf{B}_1 = \mathbf{0}$

Source	df	SSCP	E(MSCP)
$\boldsymbol{\beta}_0$	1	$n\bar{\mathbf{y}}\,\bar{\mathbf{y}}'$	$\Sigma + n\boldsymbol{\beta}_0\boldsymbol{\beta}_0'$
$\mathbf{B}_1 \mid \boldsymbol{\beta}_0$	k	$\mathbf{H} = \widehat{\mathbf{B}}_1'(\mathbf{X}_d'\mathbf{X}_d)\widehat{\mathbf{B}}_1$	$\Sigma + \frac{\mathbf{B}_1'(\mathbf{X}_d'\mathbf{X}_d)\mathbf{B}_1}{k}$
Residual	$n-k-1$	$\mathbf{E} = \mathbf{Y}_d'\mathbf{Y}_d - \widehat{\mathbf{B}}_1'(\mathbf{X}_d'\mathbf{X}_d)\widehat{\mathbf{B}}_1$	Σ
Total	n	$\mathbf{Y}'\mathbf{Y}$	

where λ_i are the roots of $|\mathbf{H}-\lambda\mathbf{E}| = 0$, $s = \min(v_h, p) = \min(k, p)$, $v_h = k$ and $v_e = n-q = n-k-1$. An alternative form for Λ is to employ sample covariance matrices. Then $\mathbf{H} = \mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}$ and $\mathbf{E} = \mathbf{S}_{yy} - \mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}$ so that $|\mathbf{H}-\lambda\mathbf{E}| = 0$ becomes $\left|\mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy} - \lambda(\mathbf{S}_{yy} - \mathbf{S}_{xx}^{-1}\mathbf{S}_{xy})\right| = 0$. From the relationship among the roots in Theorem 2.6.8, $|\mathbf{H} - \theta(\mathbf{H}+\mathbf{E})| = \left|\mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy} - \theta\,\mathbf{S}_{yy}\right| = 0$ so that

$$\Lambda = \prod_{i=1}^{s} (1+\lambda_i)^{-1} = \prod_{i=1}^{s} (1-\theta_i) = \frac{\left|\mathbf{S}_{yy} - \mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}\right|}{\left|\mathbf{S}_{yy}\right|}$$

Finally, letting $\mathbf{S}$ be defined as in (4.2.7) and using Theorem 2.5.6 (6), the Λ criterion for testing $H_1 : \mathbf{B}_1 = \mathbf{0}$ becomes

$$\begin{aligned}\Lambda &= \frac{|\mathbf{E}|}{|\mathbf{E}+\mathbf{H}|} = \prod_{i=1}^{s} (1+\lambda_i)^{-1} \\ &= \frac{|\mathbf{S}|}{|\mathbf{S}_{xx}|\ |\mathbf{S}_{yy}|} = \prod_{i=1}^{s} (1-\theta_i)\end{aligned} \tag{4.2.14}$$

Using (3.5.3), one may relate Λ to an F distribution. Comparing (4.2.14) with the expression for W for testing independence in (3.8.32), we see that testing $H_1 : \mathbf{B}_1 = \mathbf{0}$ is equivalent to testing $\Sigma_{xy} = \mathbf{0}$ or that the set X and Y are independent under joint multivariate normality. We shall see in Chapter 8 that the quantities $r_i^2 = \theta_i = \lambda_i / (1+\lambda_i)$ are then sample canonical correlations. For the other test criteria, $M = (|p-k|-1)/2$ and $N = (n-k-p-2)/2$ in Theorem 3.5.1. To test additional hypotheses regarding the elements of $\mathbf{B}$ other matrices $\mathbf{C}$ and $\mathbf{M}$ are selected. For example, for $\mathbf{C} = \mathbf{I}_q$ and $\mathbf{M} = \mathbf{I}_p$, one may test that all coefficients are zero, $H_o : \mathbf{B} = \mathbf{0}$. To test that any single row of $\mathbf{B}_1$ is zero, a row of $\mathbf{C} = [\mathbf{0}, \mathbf{I}_k]$ would be used with $\mathbf{M} = \mathbf{I}_p$. Failure to reject $H_i : \mathbf{c}_i'\mathbf{B} = \mathbf{0}'$ may suggest removing the variable from the MR model.

Assumptions about MANOVA

- **Observation Independence:** Each participant or observation should be independent of one another. For example, one student's performance should not influence another's.
- **Multivariate Normality:** The combined dependent variables should be approximately normally distributed for each group of the independent variable.
- **Homogeneity of Variance-Covariance Matrices:** The variance-covariance matrix of the dependent variables should be similar for all groups. This means that the spread and relationship between variables should be consistent across groups.
- **Absence of Multicollinearity:** The dependent variables should not be too highly correlated. If two variables are very similar, it doesn't add value to have both.
- **Linear Relationships:** There should be a linear relationship between each pair of dependent variables for each group of the independent variable.

Multivariate Analysis of Variance (MANOVA) has several advantages over univariate methods:

- **Multiple Outcome Variables:** Unlike univariate methods like ANOVA, MANOVA allows you to examine the effect of one or more independent variables on two or more dependent variables simultaneously. This allows for a more comprehensive view of the data.
- **Protection against Type I Errors:** When multiple ANOVAs are performed separately for each dependent variable, the likelihood of making a Type I error (falsely rejecting the null hypothesis) increases. By considering multiple dependent variables simultaneously, MANOVA helps control this error rate.
- **Interactions among Dependent Variables:** MANOVA allows for the examination of potential interactions between dependent variables, providing insights into how they may collectively be influenced by the independent variables.

- **Detection of Multivariate Effects:** There may be instances where the effect of an independent variable isn't significant on any individual dependent variable but is significant when all dependent variables are considered together. MANOVA can help detect such multivariate effects, which could be missed by separate ANOVAs.
- **Economical:** MANOVA is more economical in terms of sample size requirements as it incorporates multiple dependent variables into a single analysis, unlike separate ANOVAs which require larger sample sizes to maintain power.

While Multivariate Analysis of Variance (MANOVA) offers many advantages, there are also several potential challenges and limitations to consider:

- **Assumptions:** MANOVA makes several assumptions, such as multivariate normality, homogeneity of covariance matrices, and absence of multicollinearity among dependent variables. If these assumptions are violated, the results of the MANOVA may be invalid. Checking and meeting these assumptions can be more complex and challenging than for univariate tests like ANOVA.

- **Interpretation Difficulty:** MANOVA results can be difficult to interpret, especially if there are interactions between factors or if there is more than one dependent variable. The relationships between multiple dependent variables can be complex, and understanding the multivariate test statistics (such as Wilks' Lambda or Pillai's Trace) requires a solid understanding of multivariate statistics.
- **High Computational Requirements:** Because it involves multiple dependent variables, the computations for MANOVA are more complex and demanding than for ANOVA. This may be a limitation in situations where computational resources are limited.
- **Requires Large Sample Sizes:** MANOVA typically requires larger sample sizes than univariate ANOVA to achieve the same statistical power. This is because the more dependent variables you have, the more challenging it becomes to detect an effect.
- **Risk of Overfitting:** With multiple dependent variables, there's a greater risk of overfitting the model, especially if there's a high degree of correlation between dependent variables.
- **Less Robust to Missing Data:** Like most statistical methods, MANOVA is less robust when dealing with missing data. Depending on the amount and nature of the missing data, it may impact the validity of the results.

To perform MANOVA test in R let's take **iris** data set.

R

```r
# R program to illustrate
# MANOVA test

# Import required library
library(dplyr)

# Taking iris data set
myData = iris

# Show a random sample
set.seed(1234)
dplyr::sample_n(myData, 10)
```

```
 Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1          5.5         2.5          4.0         1.3 versicolor
2          5.6         2.5          3.9         1.1 versicolor
3          6.0         2.9          4.5         1.5 versicolor
4          6.4         3.2          5.3         2.3  virginica
5          4.3         3.0          1.1         0.1     setosa
6          7.2         3.2          6.0         1.8  virginica
7          5.9         3.0          4.2         1.5 versicolor
8          4.6         3.1          1.5         0.2     setosa
9          7.9         3.8          6.4         2.0  virginica
10         5.1         3.4          1.5         0.2     setosa
```

To know if there is any important difference, in sepal and petal length, between the different species then perform MANOVA test. Hence the function **manova()** can be used as follows.

```r
# Taking two dependent variable
sepal = iris$Sepal.Length
petal = iris$Petal.Length

# MANOVA test
result = manova(cbind(Sepal.Length, Petal.Length) ~ Species,
                                                  data = iris)
summary(result)
```

Output:

```
            Df Pillai approx F num Df den Df    Pr(>F)
Species      2 0.9885   71.829      4    294 < 2.2e-16 ***
Residuals  147
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

From the output above, it can be seen that the two variables are highly significantly different among Species.

Cluster Analysis

Cluster analysis, also known as clustering, is a statistical technique used in machine learning and data mining that involves the grouping of objects or points in such a way that objects in the same group, also known as a cluster, are more similar to each other than to those in other groups. It is a main task of exploratory data analysis and is used in various fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

Cluster analysis is a multi-step process and the specific steps can vary somewhat depending on the specific technique being used. However, the general methodology is typically similar and can be outlined as follows:

- **Data preparation**: The data you plan to cluster must be gathered, cleaned, and preprocessed. This can involve dealing with missing or erroneous data, transforming data into a usable format, normalizing data so that different scales can be compared, and reducing dimensionality if the data has a high number of variables.

- **Feature selection**: This step involves deciding which variables or features will be used for clustering. The selected features should be relevant to the clustering task. Irrelevant or redundant features can distort the structure of the data and lead to poor clustering results.
- **Choice of clustering algorithm**: Different clustering algorithms are suitable for different types of data and different clustering tasks. Some algorithms, like K-means, work best with spherical clusters of similar size, while others, like DBSCAN, can handle clusters of different shapes and sizes.
- **Parameter setting**: Most clustering algorithms have parameters that need to be set before the algorithm can run. For example, the K-means algorithm requires the number of clusters to be specified in advance. These parameters can have a big impact on the clustering results, so they need to be chosen carefully.
- **Clustering**: Run the clustering algorithm on your data. This will typically involve an iterative process where the algorithm continually adjusts the clusters until it finds the best fit for the data.
- **Cluster validation**: After the clusters have been formed, it's important to validate the results to ensure they make sense. This can involve statistical testing, comparison to known classes, or domain-specific validation

Cluster analysis is a versatile process with various types that can be used depending on the specific needs of a task. Here are some common types of cluster analysis:

Partitioning Clustering

This type of clustering divides data into a set of mutually exclusive clusters. The most well-known method in this category is the K-means clustering algorithm, where 'K' refers to the pre-specified number of clusters. These methods typically start with a random partitioning of data and refine it through an iterative process.

Hierarchical Clustering

This type of clustering creates a tree of clusters. Hierarchical clustering, not only clusters the data, but also builds a hierarchy of clusters, like a binary tree structure. It comes in two flavors:

- **Agglomerative (Bottom-Up)**: Each data point starts in its own cluster and pairs of clusters are merged as one moves up the hierarchy.
- **Divisive (Top-Down)**: All data points start in one cluster, and splits are performed recursively as one moves down the hierarchy.

Density-Based Clustering

These types of algorithms look for areas in the feature space where there are high densities of observations. The most famous of these is DBSCAN (Density-Based Spatial Clustering of Applications with Noise). It works by defining a neighborhood around a data point and if there are a minimum number of points within this neighborhood then a cluster is started.

Grid-Based Clustering

These types of algorithms quantize the space into a finite number of cells forming a grid structure and perform all clustering operations on this obtained grid structure. The primary advantage of these algorithms is its fast processing time, which is typically dependent on the number of cells in each dimension in the quantized space.

Model-Based Clustering

These algorithms hypothesize a model for each cluster and find the best fit of data to a given model. Examples of these are Gaussian Mixture Models and Expectation-Maximization algorithms. The advantage here is the model provides a probabilistic framework for estimating the characteristics of the process generating the data.

Subspace Clustering or Biclustering

While in standard clustering, an object belongs to exactly one cluster, in subspace clustering, an object can belong to more than one cluster and each cluster is associated with a subset of the dimensions. This type of clustering is particularly useful for high-dimensional data where each dimension represents a feature of the data.

K-means Clustering:

The main objective in K-means is to minimize the within-cluster variance, which is typically measured by Euclidean distance. The formula for Euclidean distance between two points x and y in n dimensional space is:

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$
$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

Where n is the number of dimensions, x_i is the i-th coordinate of point x, and y_i is the i-th coordinate of point y.

The objective function for K-means, which needs to be minimized, is the sum of the Euclidean distances from each data point to the center of the cluster (centroid) it was assigned to. Here's how the formula looks like:

$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

Where ||x_i - v_j|| is the Euclidean distance from data point i to the centroid of cluster j, and w_ij equals 1 if point i belongs to cluster j and 0 otherwise.

Hierarchical Clustering:

In hierarchical clustering, we use different types of linkage methods to find the distance between clusters, which can be single linkage (minimum distance), complete linkage (maximum distance), average linkage, and centroid linkage. Here are the formulas for single and complete linkage methods:

Single Linkage: $d(S,T) = \min \{d(s,t) : s \in S, t \in T\}$

Complete Linkage: $d(S,T) = \max \{d(s,t) : s \in S, t \in T\}$

Where S and T are two different clusters, s and t are any two points in clusters

S and T respectively, and $d(s,t)$ is the distance between points s and t.

Advantages of Cluster Analysis:

1. It can help identify patterns and relationships within a dataset that may not be immediately obvious.

2. It can be used for exploratory data analysis and can help with feature selection.

3. It can be used to reduce the dimensionality of the data.

4. It can be used for anomaly detection and outlier identification.

5. It can be used for market segmentation and customer profiling.

Disadvantages of Cluster Analysis:

1. It can be sensitive to the choice of initial conditions and the number of clusters.

2. It can be sensitive to the presence of noise or outliers in the data.

3. It can be difficult to interpret the results of the analysis if the clusters are not well-defined.

4. It can be computationally expensive for large datasets.

5. The results of the analysis can be affected by the choice of clustering algorithm used.

Discriminant analysis is used to evaluate group separation and to develop rules for assigning observations to groups. Cluster analysis is concerned with group identification. The goal of cluster analysis is to partition a set of observations into a distinct number of unknown groups or clusters in such a manner that all observations within a group are similar, while observations in different groups are not similar. If data are represented as an $n \times p$ matrix $\mathbf{Y} = [y_{ij}]$ where

$$\underset{n \times p}{\mathbf{Y}} = \begin{bmatrix} \mathbf{y}_1' \\ \mathbf{y}_i' \\ \vdots \\ \mathbf{y}_n' \end{bmatrix}$$

the goal of cluster analysis is to develop a classification scheme that will partition the rows of $\mathbf{Y}$ into k distinct groups (clusters). The rows of the matrix usually represent items or objects. To uncover the groupings in the data, a measure of nearness, also called a proximity measure needs to be defined. Two natural measures of nearness are the degree of distance or "dissimilarity" and the degree of association or "similarity" between groups. The choice of the proximity measure depends on the subject matter, scale of measurement (nominal, ordinal, interval, ratio), and type of variables (continuous, categorical) being analyzed. In many applications of cluster analysis, one begins with a proximity matrix rather than a data matrix. Given the proximity matrix of order $(n \times n)$ say, the entries may represent dissimilarities $[d_{rs}]$ or similarities $[s_{rs}]$ between the rth and s^{th} objects. Cluster analysis is a tool for classifying objects into groups and is not concerned with the geometric representation of the objects in a low-dimensional space. To explore the dimensionality of the space, one may use multidimensional scaling.

Proximity measures are used to represent the nearness of two objects. If a proximity measure represents similarity, the value of the measure increases as two objects become more similar. Alternatively, if the proximity measure represents dissimilarity the value of the measure decreases in value as two objects become more alike. Letting $\mathbf{y}_r$ and $\mathbf{y}_s$ represent two objects in a p-variate space, an example of a dissimilarity measure is the Euclidean distance between $\mathbf{y}_r$ and $\mathbf{y}_s$. As a measure of similarity, one may use the proportion of the elements in the two vectors that match. More formally, one needs to establish a set of mathematical axioms to create dissimilarity and similarity measures.

a. Dissimilarity Measures

Given two objects $\mathbf{y}_r$ and $\mathbf{y}_s$ in a p-dimensional space, a dissimilarity measure satisfies the following conditions.

$$
\begin{aligned}
&1. \quad d_{rs} \geq 0 \text{ for all objects } \mathbf{y}_r \text{ and } \mathbf{y}_s \\
&2. \quad d_{rs} = 0 \text{ if and only if } \mathbf{y}_r = \mathbf{y}_s \\
&3. \quad d_{rs} = d_{sr}
\end{aligned}
\tag{9.2.1}
$$

Condition (3) implies that the measure is symmetric so that the dissimilarity measure that compares $\mathbf{y}_r$ (object r) with $\mathbf{y}_s$ (object s) is the same as the comparison for object s versus object r. Condition (2) requires the measure to be zero whenever object r equals object s,

the objects are identical only if $d_{rs} = 0$ and under no other situation. Finally, (1) implies that the measure is never negative. A dissimilarity measure that satisfies conditions (1) to (3) is said to be a semi-metric measure.

For continuous (interval, ratio scale) variables, the most common dissimilarity measure is the Euclidean distance between two objects. Given an $(n \times p)$ matrix $\mathbf{Y}$ with $(1 \times p)$ row vectors $\mathbf{y}_i'$, the square of the Euclidean distance between two rows $\mathbf{y}_r$ and $\mathbf{y}_s$ is defined as

$$d_{rs}^2 = (\mathbf{y}_r - \mathbf{y}_s)' (\mathbf{y}_r - \mathbf{y}_s) = \|\mathbf{y}_r - \mathbf{y}_s\|^2 \tag{9.2.2}$$

The $(n \times n)$ data matrix $\mathbf{D} = [d_{rs}]$ is called the Euclidean distance matrix. Because a change in the units of measurement may cause a variable to dominate the ranking of distances, the Euclidean distance matrix is most effective for variables that are commensurate.

When variables are not commensurate, one may weight the squared differences by $s_j^2 = \sum_{i=1}^{n} (y_{ij} - \overline{y}_{.j})^2 / (n-1)$, $j = 1, 2, ..., p$ where s_j^2 and $\overline{y}_{.j}$ represent estimates of the mean and variance of variable j.

$$d_{rs}^2 = (\mathbf{y}_r - \mathbf{y}_s)' (\text{diag } \mathbf{S})^{-1} (\mathbf{y}_r - \mathbf{y}_s) \tag{9.2.3}$$

This process eliminates the dependence of the analysis on the units of measurement. However, it often causes the within-cluster distances to become larger than the between-cluster differences thus tending to mask clusters, Hartigan (1975, p. 59). This problem also occurs when using the square of Mahalanobis distances as a proximity measure in the metric of the covariance matrix $\mathbf{S}$ defined as follows.

$$d_{rs}^2 = (\mathbf{y}_r - \mathbf{y}_s)' \mathbf{S}^{-1} (\mathbf{y}_r - \mathbf{y}_s) \tag{9.2.4}$$

Mahalanobis distances, whether or not the variables are commensurate, tend to mask clusters even more since correlations tend to be reduced even further, Hartigan (1975, p. 63). There is no satisfactory solution to the inconsistent unit of measurement problem.

Because Euclidean distance is a special case of the Minkowski metric (L_p-norm), the dissimilarity measures may be represented as

$$d_{rs} = \left(\sum_{j=1}^{p} \left| y_{rj} - y_{sj} \right|^{\lambda} \right)^{1/\lambda} \tag{9.2.5}$$

Varying λ in (9.2.5) changes the weight assigned to larger and smaller distances. For $\lambda = 1$, we have city-block distances (L_1-norm) and for $\lambda = 2$, we have Euclidean distances. City-block distances tend to have low sensitivity to outliers.

In defining dissimilarity measures, we did not include the triangular inequality condition that $d_{rs} \leq d_{rq} + d_{qs}$ for all points r, s, and q. With this condition, the dissimilarity measures form a metric. While all Minkowski distances form a metric, the requirement is a sufficient condition but not a necessary condition for a proximity measure. For example, we may cluster using the matrix of squared Euclidean distances which do not form a metric or use the norm $d_{rs} = \|\mathbf{y}_r - \mathbf{y}_s\|$, which does form a metric. Replacing the triangular inequality with the condition that $d_{rs} \leq \max\left(d_{rq},\ d_{qs}\right)$ for all points r, s, and q, the semi-metric becomes an ultrametric. It is called an ultrametric since this condition is stronger than the triangular inequality requirement for a metric.

FIGURE 9.2.1. 2×2 Contingency Table, Binary Variables

		Row s		Total
		1	0	
Row r	1	a	b	$a+b$
	0	c	d	$c+d$
Total		$a+c$	$b+d$	$p=a+b+c+d$

To reduce the size of squared distances (or distances), they are often divided by the number of variables p. Because all dissimilarities are divided by the same number p, this does not effect the clustering results. In addition, the squared Euclidean distances are often mean centered by subtracting from each variable its corresponding mean. Then $\mathbf{Y}_d = [y_{ij} - \overline{y}_{.j}]$ is analyzed. Variable centering does not effect the Euclidean distances since removing the means from the p variables does not change the Euclidean distance between any two objects.

Two other dissimilarity measures that have been proposed when all y_{ij} are positive are the following.

$$\text{Canberra Metric} \qquad d_{rs} = \sum_{j=1}^{p} \left\{ \frac{|\, y_{rj} - y_{sj} \,|}{(y_{rj} + y_{sj})} \right\} \tag{9.2.6}$$

$$\text{Czekanowski Coefficient} \qquad d_{rs} = \frac{\sum_{j=1}^{p} |\, y_{rj} - y_{sj} \,|}{\sum_{j=1}^{p} (y_{rj} + y_{sj})} \tag{9.2.7}$$

Both of these measures are modified L_1-norms and are used when data are skewed and/or contain outliers. A scaled L_1-norm that may be used for data that includes both positive and negative values is Gower's metric

$$\text{Gower Metric} \qquad d_{rs} = \sum_{j=1}^{p} \left| y_{rj} - y_{sj} \right| / R_j \tag{9.2.8}$$

where R_j represents the range of variable j.

b. *Similarity Measures*

Given two objects $\mathbf{y}_r$ and $\mathbf{y}_s$ in a p-dimensional space, a similarity measure satisfies the following conditions.

$$
\begin{aligned}
&1. \quad 0 \leq s_{rs} \leq 1 \text{ for all objects } \mathbf{y}_r \text{ and } \mathbf{y}_s \\
&2. \quad s_{rs} = 1 \text{ if and only if } \mathbf{y}_r = \mathbf{y}_s \\
&3. \quad s_{rs} = s_{sr}
\end{aligned}
\tag{9.2.10}
$$

Condition (3) again implies the measure is symmetric while conditions (1) and (2) ensure that its always positive and identically one only if objects r and s are identical.

Given a similarity measure that satisfies (9.2.10), one may always create a dissimilarity measure using the relationship that $d_{rs} = 1 - s_{rs}$ or some other decreasing function, however, the new measure may not form a metric. Conversely, given a dissimilarity measure d_{rs} one may want to construct a similarity measure as $s_{rs} = 1/(1 + d_{rs})$. Because d_{rs} is unbounded, $0 < s_{rs} \leq 1$ so that s_{rs} never attains the value of zero. Thus, one may not create s_{rs} from d_{rs}. While one may use either similarity measures or dissimilarity measures to cluster the rows of a data matrix,

A common measure of similarity suggested by some authors is to use the Pearson product moment correlation between objects $\mathbf{y}_r$ and $\mathbf{y}_s$ $r, s = 1, 2, \ldots, n$ defined as

$$q_{rs} = \frac{\sum_{j=1}^{p} (y_{rj} - \overline{y}_{r.})(y_{sj} - \overline{y}_{s.})}{\left[\sum_{j=1}^{p} (y_{rj} - \overline{y}_{r.})^2 \sum_{j=1}^{p} (y_{sj} - \overline{y}_{s.})^2\right]} \tag{9.2.11}$$

where $\overline{y}_{r.} = \sum_j y_{rj}/p$ and $\overline{y}_{s.} = \sum_j y_{sj}/p$. The symbol q_{rs} is being used since we are calculating correlations using the rows of the data matrix. However, because $-1 \leq q_{rs} \leq 1$, it does not satisfy condition (1) in (9.2.10). To correct this situation, one may instead use the quantities $|q_{rs}|$ or $1 - q_{rs}^2$. It has also been suggested to add one to q_{rs} and divide by 2, $q_{rs}^* = (q_{rs} + 1)/2$. Or, equivalently the quantity $1 - q_{rs}$. This still presents a problem since $q_{rs}^* = 1$ does not imply that $\mathbf{y}_r = \mathbf{y}_s$ only that the elements in each vector are perfectly linearly related.

If the matrix $\mathbf{Y}$ is standardized so that $\widetilde{y}_{ij} = (y_{rj} - \overline{y}_{i.})/s_i$ where $s_i = \sum_{j=1}^{p} \widetilde{y}_{sj}^2/(p-1)$ for $i = 1, 2, \ldots, n$ objects, then one may relate q_{rs} to squared Euclidean distances as follows

$$\begin{aligned} d_{rs}^2 &= \sum_{j=1}^{p} (\widetilde{y}_{rj} - \widetilde{y}_{sj})^2 \\ &= \sum_{j=1}^{p} \widetilde{y}_{rj}^2 + \sum_{j=1}^{p} \widetilde{y}_{sj}^2 - 2\sum_{j=1}^{p} \widetilde{y}_{rj}\widetilde{y}_{sj} \\ &= 2(1 - q_{rs}) \end{aligned} \tag{9.2.12}$$

so that $d_{rs} = \sqrt{2(1 - q_{rs})}$ becomes a metric. One may then use the quantities d_{rs}, $1 - q_{rs}$ or d_{rs}^2 to cluster rows.

Similarity measures for binary variables are important in cluster analysis. To construct similarity measures for binary data we again consider the entries in the 2×2 table given in Table 9.2.1. We consider how to weight matches and mismatches because a $(1-1)$ match may be more important than a $(0-0)$ match since the former implies the presence while the later implies the absence of an attribute. One may even want to discard $(0-0)$ matches completely. To provide for the differential weighting of matches and mismatches and the treatment of $(0-0)$ matches, Anderberg (1973, p. 89) using Table 9.2.1 summarizes several matching counting schemes as shown in Table 9.2.1

TABLE 9.2.1. Matching Schemes

Similarity Measure	Weighting of Matches and Mismatches	Coefficient Name	Dissimilarity= (1-Similarity)
1. $\frac{a+d}{p}$	Equal Weight to Matched Pairs With $(0-0)$ Matches	Simple Matching	Metric
2. $\frac{2(a+d)}{2(a+d)+b+c}$	Double Weight to Matched Pairs With $(0-0)$ Matches	Double Matching	Metric
3. $\frac{a+d}{a+d+2(b+c)}$	Double Weight to Unmatched Pairs With $(0-0)$ Matches	Rogers-Tanimoto	Metric
4. $\frac{a}{p}$	Equal Weight to Matches With No$(0-0)$ Matches in Numerator	Russell-Rao	Semi-metric
5. $\frac{a}{a+b+c}$	Equal Weight to Matches With No $(0-0)$ Matches in Denominator	Jaccard	Metric
6. $\frac{2a}{2a+b+c}$	Double Weight to $(1-1)$ Matches With No $(0-0)$ Matches in Numerator or Denominator	Czekanowski-Sørensen-Dice	Metric
7. $\frac{a}{a+2(b+c)}$	Double Weight to Unmatched Pairs With No $(0-0)$ Matches in Numerator or Denominator	Unnamed	Metric
8. $\frac{a}{a+c}$	Ratio of Matches to Mismatches No $(0-0)$ Matches Included	Kulezynski	Semi-metric
9. $\frac{a+d}{b+c}$	Ratio of Matches to Mismatches With $(0-0)$ Matches Included	Unnamed	Semi-metric

c. *Clustering Variables*

Proximity measures for clustering rows of $\mathbf{Y}_{n\times p}$ may also be used to cluster rows of $\mathbf{Y}'_{p\times n}$, or variables. Replacing p with n in Table 9.2.1, all similarity (dissimilarity) measures in Table 9.3.1 may be used to cluster variables. When clustering variables, one is likely to standardize the variables and use some measure of association for clustering. When working with variables, standardization is usually a natural step and a measure of association a_{ij} is used to cluster variables. To convert a_{ij} to a dissimilarity measure, the function $d_{ij} = \sqrt{1 - a_{ij}}$ is often used.

Agresti (1981) provides an overview of several measures of association for nominal and ordinal variables. For binary data, we may use Table 9.2.1 to calculate the Pearson product moment correlation coefficient r_{ij} and the cosine of the angle between variables i and j as follows, Anderberg (1973, pp. 84-85)

$$r_{ij} = \frac{ad - bc}{[(a+b)(c+d)(a+c)(b+d)]^{1/2}} \tag{9.2.15}$$

$$c_{ij} = \left[\left(\frac{a}{a+b}\right)\left(\frac{a}{a+c}\right)\right]^{1/2} \tag{9.2.16}$$

Then, from (9.2.15) and (9.2.16) dissimilarity measures that may be used to cluster variables are $d_{ij}^2 = (1 - r_{ij})$, $d_{ij} = \sqrt{1 - r_{ij}}$, $d_{ij}^2 = (1 - c_{ij})$, or $d_{ij} = \sqrt{1 - c_{ij}}$. Formula (9.2.15) is the familiar phi coefficient. The coefficient c_{ij} is also called the Ochiai coefficient. These measures work most effectively when all $r_{ij} \geq 0$ or all $c_{ij} \geq 0$. In general the entries in Table 9.2.1 are preferred.

Cluster Analysis

To initiate a cluster analysis one constructs a proximity matrix. The proximity matrix represents the strength of the relationship between pairs of rows in $\mathbf{Y}'_{p\times n}$ or the data matrix $\mathbf{Y}_{n\times p}$. Algorithms designed to perform cluster analysis are usually divided into two broad classes called hierarchical and nonhierarchical clustering methods. Generally speaking, hierarchical methods generate a sequence of cluster solutions beginning with clusters containing a single object and combines objects until all objects form a single cluster; such methods are called agglomerative hierarchical methods. Other hierarchical methods begin with a single cluster and split objects successively to form clusters with single objects; these methods are called diversive hierarchical methods. In both the agglomerative and diversive processes, a tree diagram, or dendogram, is created as a map of the process. The agglomerative hierarchical procedures fall into three broad categories: Linkage, Centroid, and Error Variance methods. Among these procedures, only linkage algorithms may be used to cluster either objects (items) or variables. The other two methods can be used to cluster only objects. Nonhierarchical methods may only be used to cluster items. In this section we review several commonly used agglomerative hierarchical methods: the single linkage (nearest neighbor), complete linkage (farthest neighbor), average linkage (average distance), centroid, and Ward's method. The nonhierarchical k-means method and some procedures for determining the number of clusters are also reviewed.

a. Agglomerative Hierarchical Clustering Methods

Agglomerative hierarchical clustering methods use the elements of a proximity matrix to generate a tree diagram or dendogram as shown in Figure 9.3.1. To begin the process, we start with $n = 5$ clusters which are the branches of the tree. Combining item 1 with 2 reduces the number of clusters by one, from 5 to 4. Joining items 3 and 4, results in 3 clusters. Next, joining item 5 with the cluster (3, 4), results in 2 clusters. Finally, all items are combined to form a single cluster, the root of the tree. As we move from left to right, groups of items are successively combined to form clusters. Once an item or group of items are combined they are never separated. Figure 9.3.1 represents the agglomerative process as we move from left to right. The label at the top of the diagram denotes the number of clusters at each step of the process. Figure 9.3.1 also represents a divisive hierarchical process which begins with the root (a single cluster) and moves from right to left to create the branches (items).

More generally, given a proximity matrix $\mathbf{D}_{n \times n} = [d_{rs}]$, the steps for the agglomerative hierarchical clustering algorithm are as follows.

1. Begin with n clusters, each containing only a single object.
2. Search the dissimilarity matrix $\mathbf{D}$ for the most similar pair. Let the pair chosen be associated with element d_{rs} so that object r and s are selected.
3. Combine objects r and s into a new cluster (rs) employing some criterion and reduce the number of clusters by 1 by deleting the row and column for objects r and s. Calculate the dissimilarities between the cluster (rs) and all remaining clusters, using the criterion, and add the row and column to the new dissimilarity matrix.
4. Repeat steps 2 and 3, $(n - 1)$ times until all objects form a single cluster. At each step, identify the merged clusters and the value of the dissimilarity at which the clusters are merged.

By changing the criterion in Step 3 above, we obtain several agglomerative hierarchical clustering methods. While we indicated that divisive methods may also be used to cluster objects, the procedure requires a significant number of calculations. For example, to locate the first split in Figure 9.3.1 would require investigating $15 = 2^{n-1} - 1$ partitions for the $n = 5$ objects using some distance criterion. In general divisive methods use large computer resources and splitting criteria are sometimes difficult to establish.

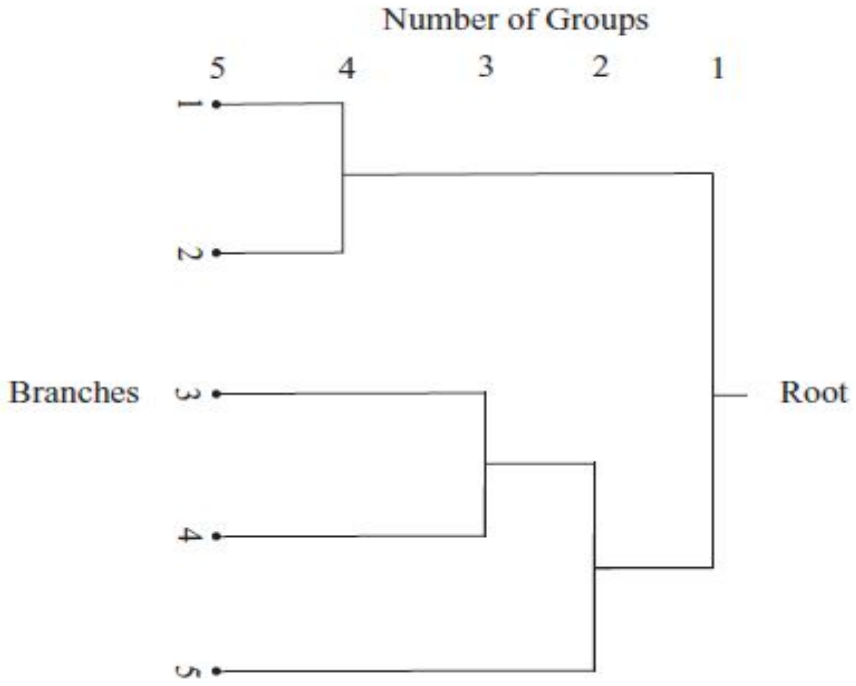


FIGURE 9.3.1. Dendogram for Hierarchical Cluster

(a1) Single Link (Nearest-Neighbor) Method

To implement the single link method, one combines objects in clusters using the minimum dissimilarity between clusters. Letting r represent any element in cluster R, $r \in R$, and s be any element in cluster S, $r \in S$, from the clusters in Step 3 of the agglomerative clustering algorithm, distances between R and S are calculated using the rule

$$d_{(R)(S)} = \min \{d_{rs} \mid r \in R \text{ and } s \in S\} \tag{9.3.1}$$

At each step of the process, a dendogram is created representing the ordered distances where objects are joined.

Example 9.3.1 *To illustrate rule (9.3.1), consider the hypothetical dissimilarity matrix*

$$\mathbf{D} = [d_{rs}] = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & \boxed{2} & 4 & 7 & 9 \\ 2 & 0 & 8 & 9 & 8 \\ 4 & 8 & 0 & 3 & 7 \\ 7 & 9 & 3 & 0 & 5 \\ 9 & 8 & 7 & 5 & 0 \end{bmatrix} \end{array}$$

Scanning the matrix **D**, *the most similar objects are represented by* $d_{rs} = d_{12} = 2$ *so that objects* 1 and 2 *are jointed to form the cluster* (12). *Following Step 3, we must calculate the minimum values using criterion (9.3.1) as follows.*

$$d_{(12)(3)} = \min\{d_{13}, d_{23}\} = \min\{4, 8\} = 4$$
$$d_{(12)(4)} = \min\{d_{14}, d_{24}\} = \min\{7, 9\} = 7$$
$$d_{(12)(5)} = \min\{d_{15}, d_{25}\} = \min\{9, 8\} = 8$$

Deleting the rows in **D** *corresponding to objects* 1 and 2, *and adding the row and column for the cluster* (12), *the new dissimilarity matrix becomes*

$$\mathbf{D}_1 = \begin{array}{c} \\ (12) \\ 3 \\ 4 \\ 8 \end{array} \begin{array}{c} \begin{array}{cccc} (12) & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 4 & 7 & 8 \\ 4 & 0 & \boxed{3} & 7 \\ 7 & 3 & 0 & 5 \\ 8 & 7 & 5 & 0 \end{bmatrix} \end{array}$$

The most similar object in $\mathbf{D}_1$ *is* $d_{34} = 3$, *so that we combine element* 3 and 4 *to form the cluster* (34). *Again, we calculate the values*

$$d_{(34)(12)} = \min\left\{d_{(3)(12)}, d_{4(12)}\right\} = \min\{4, 7\} = 4$$
$$d_{(34)(5)} = \min\{d_{35}, d_{45}\} = \min\{7, 5\} = 5$$

so that the new dissimilarity matrix is

$$\mathbf{D}_2 = \begin{array}{c} \\ (12) \\ (34) \\ (8) \end{array} \begin{array}{c} \begin{array}{ccc} (12) & (34) & (5) \end{array} \\ \begin{bmatrix} 0 & \boxed{4} & 8 \\ 4 & 0 & 5 \\ 8 & 5 & 0 \end{bmatrix} \end{array}$$

The most similar value in $\mathbf{D}_2$ *is the value* 4 *at which* (34) *is combined with* (12). *Then*

$$d_{((12)(34))5} = \min\left\{d_{(12)(5)}, d_{(34)5}\right\} = \{8, 5\} = 5$$

so that object 5 *is combined with clusters* (12) and (34) *to form the single cluster (1 2 3 4 5) at the value* 5. *The dendogram for nearest dissimilarities follows.*
The groupings and ordered dissimilarities at which clusters are joined are included in the tree diagram at the left margin.

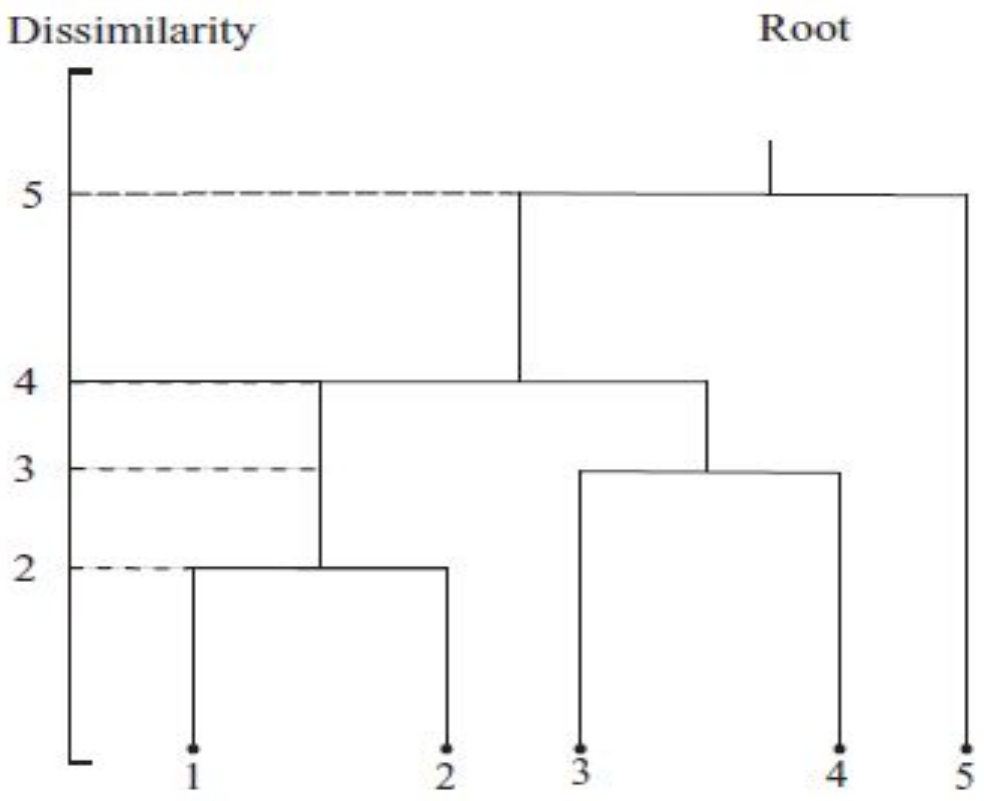


FIGURE 9.3.2. Dendogram for Single Link Example

(a2) Complete Link (Farthest-Neighbor) Method

In the single link method, dissimilarities were replaced using minimum values. For the complete link procedure, maximum values are calculated instead. Letting $r \in R$ and $s \in S$, where R and S are two clusters, distances between clusters R and S are calculated using the rule

$$d_{(R)(S)} = \max \{d_{rs} \mid r \in R \text{ and } s \in S\} \tag{9.3.2}$$

Example 9.3.2 *To illustrate rule (9.3.2.), we consider the same dissimilarity matrix discussed in Example 9.3.1.*

$$\mathbf{D} = [d_{rs}] = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & \boxed{2} & 4 & 7 & 9 \\ 2 & 0 & 8 & 9 & 8 \\ 4 & 8 & 0 & 3 & 7 \\ 7 & 9 & 3 & 0 & 5 \\ 9 & 8 & 7 & 5 & 0 \end{bmatrix} \end{array}$$

From $\mathbf{D}$ *we again see that* $d_{rs} = 2$ *represents the most similar objects. Using (9.3.2.), we replace minimum values with maximum values*

$$d_{(12)(3)} = \max\{d_{13}, d_{23}\} = d_{23} = 8$$
$$d_{(12)(4)} = \max\{d_{14}, d_{24}\} = d_{24} = 9$$
$$d_{(12)(5)} = \max\{d_{15}, d_{25}\} = d_{15} = 9$$

so that the new dissimilarity matrix is

$$\mathbf{D}_1 = \begin{array}{c} \\ (1\,2) \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccc} (1\,2) & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 8 & 9 & 9 \\ 8 & 0 & \boxed{3} & 7 \\ 9 & 3 & 0 & 5 \\ 9 & 7 & 5 & 0 \end{bmatrix} \end{array}$$

The most similar object in $\mathbf{D}_1$ *is* $d_{34} = 3$ *so that* 3 *is combined with* 4. *Using the maximum rule,*

$$d_{(34)(12)} = \max\left\{d_{(3)(12)}, d_{4(12)}\right\} = d_{4(12)} = 9$$
$$d_{(34)(5)} = \max\left\{d_{35}, d_{45}\right\} = d_{35} = 7$$

so that the modified dissimilarities matrix is

$$\mathbf{D}_2 = \begin{array}{c} \\ (12) \\ (34) \\ (5) \end{array} \begin{array}{c} \begin{array}{ccc} (12) & (34) & (5) \end{array} \\ \begin{bmatrix} 0 & 9 & 9 \\ 9 & 0 & 7 \\ 9 & 7 & 0 \end{bmatrix} \end{array}$$

In $\mathbf{D}_2$, (34) *is combined with* 5. *And finally, all elements are combined at the value* 9. *The dendogram for the process follows*

Comparing Figure 9.3.3. with Figure 9.3.2, observe that they do not yield the same result. Objects 3 and 4 *are joined to object 5 by the complete link process while in the single link process they were joined to objects* 1 and 2. *While the two methods do not result in the same dendogram, the tree diagrams are not changed by monotonic transformations of the proximity measures.*

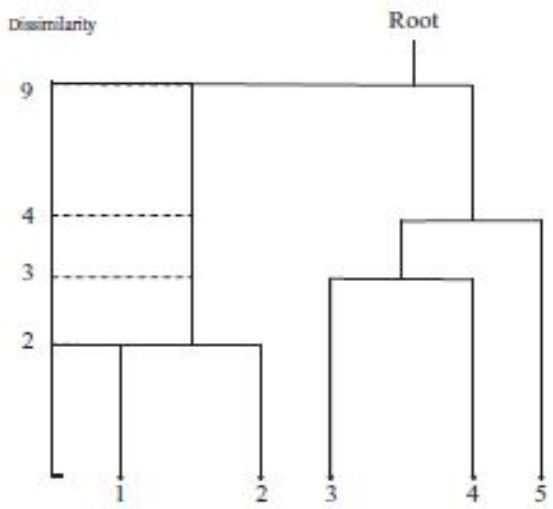


FIGURE 9.3.3. Dendogram for Complete Link Example

(a3) Average Link Method

When comparing two clusters of objects R and S, the single link and complete link methods of combining clusters depended only upon a single pair of objects within each cluster. Instead of using a minimum or maximum measure, the average link method calculates the distance between two clusters using the average of the dissimilarities in each cluster

$$d_{(R)(S)} = \frac{\sum_r \sum_s d_{rs}}{n_R n_S} \tag{9.3.3}$$

where $r \in R$, $s \in S$, and n_R and n_S represent the number of objects in each cluster. Hence, the dissimilarities in Step 3 are replaced by an average of $n_R n_S$ dissimilarities between all pairs of elements $r \in R$ and $s \in S$.

(a4) Centroid Method

In the average link method, the distance between two clusters is defined as an average of dissimilarity measures. Alternatively, suppose cluster R contains n_R elements and cluster S contains n_S elements. Then, the centroids for the two item clusters are

$$\bar{\mathbf{y}}_r = \frac{\sum_r \mathbf{y}_r}{n_R} = \begin{bmatrix} \bar{y}_{r1} \\ \bar{y}_{r2} \\ \vdots \\ \bar{y}_{rp} \end{bmatrix} \quad \text{and} \quad \bar{\mathbf{y}}_s = \frac{\sum_s \mathbf{y}_s}{n_S} = \begin{bmatrix} \bar{y}_{s1} \\ \bar{y}_{s2} \\ \vdots \\ \bar{y}_{sp} \end{bmatrix} \tag{9.3.4}$$

and the square of the Euclidean distance between the two clusters is $d_{rs}^2 = \left\|\overline{\mathbf{y}}_r - \overline{\mathbf{y}}_s\right\|^2$. For the centroid agglomerative process, one begins with any dissimilarity matrix $\mathbf{D}$ (in SAS, the distances are squared unless one uses the NOSQUARE option). Then, the two most similar clusters are combined using the weighted average of the two clusters. Letting T represent the new cluster, the centroid of T is

$$\overline{\mathbf{y}}_t = \left(n_R\overline{\mathbf{y}}_r + n_S\overline{\mathbf{y}}_s\right) / \left(n_R + n_S\right) \tag{9.3.5}$$

The centroid method is called the median method if an unweighted average of the centroids is used, $\overline{\mathbf{y}}_t = \left(\overline{\mathbf{y}}_r + \overline{\mathbf{y}}_s\right)/2$. The median method is preferred when $n_R >> n_S$ or $n_S \gg n_R$.

Letting the dissimilarity matrix $\mathbf{D} = \left[d_{rs}^2\right]$ where $d_{rs}^2 = \|\mathbf{y}_r - \mathbf{y}_s\|^2$, suppose the elements $r \in R$ and $s \in S$ are combined into a cluster T where

$$\overline{\mathbf{y}}_t = \left(n_R\overline{\mathbf{y}}_s + n_S\overline{\mathbf{y}}_s\right) / \left(n_R + n_S\right).$$

Then to calculate the square of the Euclidean distance between cluster T and the centroid $\overline{\mathbf{y}}_u$ of a third cluster U, the following formula may be used

$$d_{tu}^2 = \left(\frac{n_R}{n_R + n_S}\right) d_{rs}^2 + \left(\frac{n_S}{n_R + n_S}\right) d_{ru}^2 - \left(\frac{n_R n_S}{n_R + n_S}\right) d_{rs}^2 \tag{9.3.6}$$

This is a special case of a general algorithm for updating proximity measures for the single link, complete link, average link, centroid and median methods developed by Williams and Lance (1977).

b. Nonhierarchical Clustering Methods

In hierarchical clustering the number of clusters is unknown. The process is initiated with a dissimilarity proximity matrix and once an object is assigned to a cluster it is never reallocated. Linkage methods may be used to cluster either items or variables. In nonhierarchical clustering, methods are only applied to cluster items. The process is initiated using the raw data matrix **Y** and not a dissimilarity matrix, **D**. One has to know a priori the number of clusters k which are either cluster centroids or seeds, and observations are reassigned using some criterion with reallocation terminating based upon some stopping rule.

Nonhierarchical clustering methods usually follow the following steps.

1. Select k p-dimensional centroids or seeds (clusters).
2. Assign each observation to the nearest centroid using some L_p-norm, usually the Euclidean distance.
3. Reassign each observation to one of the k clusters based upon some criterion.
4. Stop if there is no reallocation of observations or if reassignment meets some convergence criterion; otherwise, return to Step 2.

In implementing a nonhierarchical clustering method one may vary the method for selecting the k clusters, employ different L_p-norms, and vary the criterion for reallocating observations to clusters to attain cluster stability.

To initiate a nonhierarchical cluster method, one first selects k centroids or seeds. The k seeds may be the first k observations, the first k observations at some defined level of separation, k random seeds, k initial seeds that may be replaced based upon some replacement algorithm, and other variations. Once the seeds are selected, each of the observations are evaluated for assignment or reassignment based upon some convergence criterion. While numerous criteria use multivariate statistics that involve the determinant and trace of the within-cluster and between-cluster variability,

The basic steps are as follows.

1. Select k seeds.
2. Assign each of the $n - k$ observations to the nearest seed and recalculate the cluster centroid (mean, median, or other depending on the L_p-norm).
3. Repeat Step 2 until all observations are assigned or until changes in cluster centroids become small (no reassignments are made in cluster membership).

In Step 2 of the k-means process, the seed may or may not be updated. Two tests may be made for seed replacement. An observation may replace one of a pair of seeds if the distance between the seeds is less than the distance between an observation and the nearest seed. The former seed becomes an observation in the recalculation of the centroid. If an observation fails this test, one may invoke a second test. The observation replaces the nearest seed if the smallest Euclidean distance from the observation to all seeds other than the nearest one is greater than the shortest distance from the nearest seed to all seeds. For one pass of the data, all observations are associated with k clusters. This process is repeated until all changes in cluster seeds become small based upon a convergence criterion.

When clustering items, hierarchical and nonhierarchical clustering methods may be combined to facilitate the identification of clusters. As a first step, one may use a hierarchical procedure to identify the seeds and number of clusters, these may be input into the nonhierarchical procedure to refine the cluster solution.

c. *Number of Clusters*

Given a cluster analysis solution, one needs to evaluate various indices to determine the number of clusters k. For some value of k, one wants to determine whether the clusters are sufficiently separated so as to illustrate minimal overlap. Thus, evaluating a cluster solution for k clusters is similar to trying to determine the dimension of the space in a PCA or the number of factors in an EFA. In those situations, recall that one constructed informal SCREE plots and under MVN, performed some test of dimensionality.

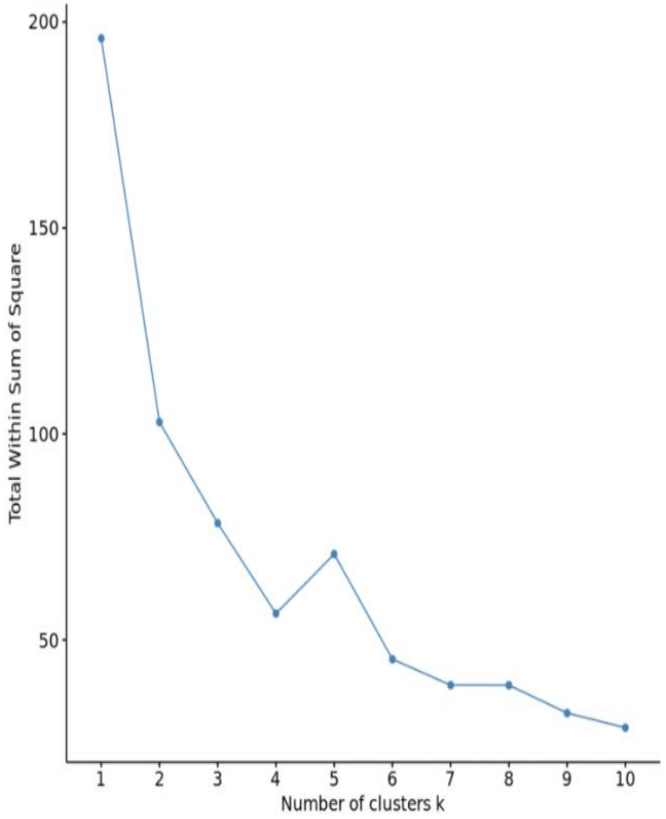
This is also the situation in cluster analysis. At each step of the clustering process a cluster is formed by joining two observations, by joining an observation and a previous cluster, or by joining two previously formed clusters. At each step of the joining process the number of clusters decrease from $k = n$ to $k = 1$. The "distance" between two clusters joined beginning with n clusters and ending with one is the Euclidean distance or dissimilarity measure between clusters R and S. For example, for the single, complete, and average link methods it is the minimum, maximum or average Euclidean distance, respectively. For Ward's method, it is the between cluster sum of squares as given in (9.3.8). As the number of clusters decreases from n to 1, the value for the "distance" measure should increase since it should be largest when two dissimilar clusters are joined. A shape elbow in the plot of distances versus the number of clusters may be an indication of the number of clusters. If one could construct an index of separation or a "pseudo" test statistic and plot its values as clusters are joined, a large change in the index when plotted against the number of clusters may help to locate a range of values or even one value for the number of clusters. There is no exact procedure for determining the number of clusters since even with random data one may find spurious clusters.

```
#scale each variable to have a mean of 0 and sd of 1
df <- scale(df)

#view first six rows of dataset
head(df)

               Murder   Assault   UrbanPop        Rape
Alabama    1.24256408 0.7828393 -0.5209066 -0.003416473
Alaska     0.50786248 1.1068225 -1.2117642  2.484202941
Arizona    0.07163341 1.4788032  0.9989801  1.042878388
Arkansas   0.23234938 0.2308680 -1.0735927 -0.184916602
California 0.27826823 1.2628144  1.7589234  2.067820292
Colorado   0.02571456 0.3988593  0.8608085  1.864967207
```

Optimal Number of Clusters



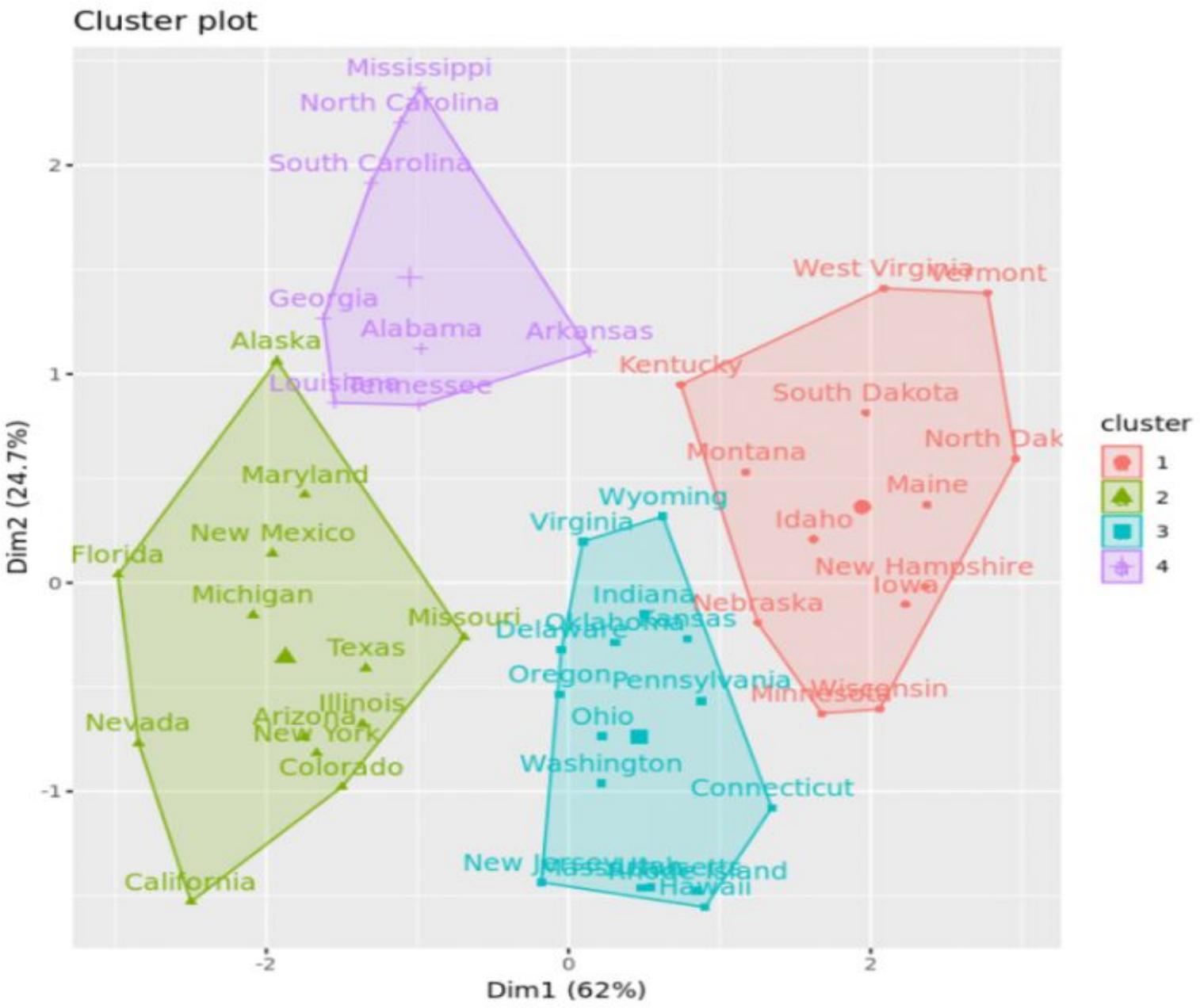
Typically when we create this type of plot we look for an “elbow” where the sum of squares begins to “bend” or level off. This is typically the optimal number of clusters.

- **16** states were assigned to the first cluster
- **13** states were assigned to the second cluster
- **13** states were assigned to the third cluster
- **8** states were assigned to the fourth cluster

```
Cluster means:
      Murder    Assault   UrbanPop       Rape
1 -0.4894375 -0.3826001  0.5758298 -0.26165379
2 -0.9615407 -1.1066010 -0.9301069 -0.96676331
3  0.6950701  1.0394414  0.7226370  1.27693964
4  1.4118898  0.8743346 -0.8145211  0.01927104
```

```
Within cluster sum of squares by cluster:
[1] 16.212213 11.952463 19.922437  8.316061
 (between_SS / total_SS =  71.2 %)
```

```
Clustering vector:
       Alabama         Alaska        Arizona       Arkansas     California
             4              3              3              4              3
   Connecticut       Delaware        Florida        Georgia         Hawaii
             1              1              3              4              1
      Illinois        Indiana           Iowa         Kansas       Kentucky
             3              1              2              1              2
         Maine       Maryland  Massachusetts       Michigan      Minnesota
             2              3              1              3              2
      Missouri        Montana       Nebraska         Nevada  New Hampshire
             3              2              2              3              2
    New Mexico       New York North Carolina   North Dakota           Ohio
             3              3              4              2              1
        Oregon   Pennsylvania   Rhode Island South Carolina   South Dakota
             1              1              1              4              2
         Texas           Utah        Vermont       Virginia     Washington
             3              1              2              1              1
     Wisconsin        Wyoming
             2              1
```


Cluster plot
Mississippi
North Carolina
South Carolina
Georgia
Alabama
Arkansas
Alaska
Louisiana
Tennessee
West Virginia
Vermont
Kentucky
South Dakota
North Dak
Montana
Maine
Idaho
New Hampshire
Iowa
Nebraska
Maryland
New Mexico
Florida
Michigan
Missouri
Texas
Nevada
Arizona
Illinois
New York
Colorado
California
Wyoming
Virginia
Indiana
Oklahoma
Kansas
Delaware
Oregon
Pennsylvania
Ohio
Washington
Connecticut
Minnesota
Wisconsin
New Jersey
Massachusetts
Rhode Island
Hawaii
Dim2 (24.7%)
Dim1 (62%)
2
1
0
-1
-2
0
2
cluster
1
2
3
4

In regression analysis the coefficient of determination, R^2, is a measure of the total variance in the dependent variable accounted for by the independent variables. In an ANOVA study involving the analysis of trends, R^2 or more correctly $\widehat{\eta}^2$, is defined as the ratio of between sum of squares to the total sum of squares and is a measure of the total variation in the dependent variable accounted for by the group means. Thus, in cluster analysis we may construct an index of R^2 as the number of clusters change. For n clusters, the total sum of squares is $T = \sum_{i=1}^{n} \|\mathbf{y}_i - \overline{\mathbf{y}}\|^2$ and the within cluster sum of squares for cluster k (C_k) is $SSE_k = \sum_i \left\|\mathbf{y}_i - \overline{\mathbf{y}}_k\right\|^2$. Then, R^2 for k clusters is defined as

$$R_k^2 = \frac{T - \sum_k SSE_k}{T} \tag{9.3.10}$$

For n clusters, each $SSE_k = 0$ so that $R^2 = 1$. As the number of clusters decrease from n to 1 they should become more widely separated. A large decrease in R_k^2 would represent a distinct join. Alternatively, joining clusters R and S one might compute the incremental change in R^2 or

$$SR^2 = R_k^2 - R_{k-1}^2 \tag{9.3.11}$$

called the semipartial R^2 index. The statistic SR^2 compares the ratio of $SSE_t - (SSE_r + SSE_s)$ where clusters C_R and C_S are jointed to form C_T to the total sum of squares $T = \sum_{i=1}^{n} \|\mathbf{y}_i - \overline{\mathbf{y}}\|^2$. The larger the increment, the larger the "loss of homogeneity" or the more separated the clusters.

The goal of cluster analysis is to find a small number of homogeneous clusters. For $n = 1$ cluster, the pooled within variance for all variables is the average of the variances for each variable so that $s^2 = \sum_{i=1}^{n} \|\mathbf{y}_i - \overline{\mathbf{y}}\|^2 / p\,(n-1)$ and the root mean square standard deviation is $s = \text{RMSSTD}$. For any new cluster C_k with n_k observations,

$$[\text{RMSSTD}]^2 = \sum_{i=1}^{n_k} \left\|\mathbf{y}_i - \overline{\mathbf{y}}_k\right\|^2 / p\,(n_k - 1) \tag{9.3.12}$$

is the pooled variance of all variables forming a cluster at a given step. Large values of the pooled variance suggest that the clusters are not homogeneous. Thus, a shape decrease to near zero for some $k < n$ suggest the formation of homogeneous clusters.

Under multivariate normality and independence of the n p-vectors for $\Sigma = \sigma^2\mathbf{I}$, one may test that the centroids of k clusters show significant separation using an ANOVA F statistic. One may also evaluate whether two means are significantly separated at any level of the clustering hierarchy as one proceeds through a cluster analysis using a t statistic.

Multidimensional Scaling

Nonhierarchical clustering methods use the data matrix $\mathbf{Y}_{n\times p}$ and a dissimilarity matrix $\mathbf{D}_{n\times n}$ to cluster items. Alternatively, hierarchical clustering techniques use only a dissimilarity matrix. Multidimensional scaling (MDS) techniques seek to find a low dimensional coordinate system to represent n objects using only a proximity matrix usually without observing the n $(p \times 1)$ observation vectors where the dimension of the space is $k \ll p$. Given perceptions or judgements regarding n objects, one tries to construct a low dimensional space to represent the judgements.

In classical, metric MDS problems, we assume there exists a data matrix $\mathbf{Y}_{n\times p}$ such that the distances between any two objects r and s is $\delta_{rs} = \|\mathbf{y}_r - \mathbf{y}_s\|$. Because $\mathbf{Y}$ is not usually observed, we try to find a set of observations $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n]'$ where $\mathbf{z}_i$ is in a k-dimensional space such that the distances $d_{rs} = \|\mathbf{z}_r - \mathbf{z}_s\|$ where $d_{rs} \approx \delta_{rs}$. Thus, a model for the distances may take the form

$$d_{rs} = \alpha + \beta\delta_{rs} + e_{rs} \text{ for all } r, s \tag{9.5.1}$$

where e_{rs} is random measurement error. Generating a scatter plot of the pairs (d_{rs}, δ_{rs}), one may obtain a set of estimated distances d_{rs} using the relationship

$$\widehat{d}_{rs} = \widehat{\alpha} + \widehat{\beta}\delta_{rs} \tag{9.5.2}$$

The fitted values $\widehat{d}_{rs}$ are not distances, but simply numbers fitted to distances and referred to as disparities. Ordering the proximities δ_{rs} so that

$$\delta_{r_1 s_1} < \delta_{r_2 s_2} < \ldots < \delta_{r_m s_m} \tag{9.5.3}$$

where $m = n(n-1)/2$ and $r_i < s_i$ are distinct integers, then by (9.5.2) if

$$\delta_{rs} < \delta_{u\,\mathrm{v}} \Longrightarrow \widehat{d}_{rs} \leq \widehat{d}_{u\,\mathrm{v}} \quad \text{for all } r < s,\, u < \mathrm{v} \tag{9.5.4}$$

This is called the weak monotonicity constraint.

In most applications of multidimensional scaling, exact proximity measures δ_{rs} are not available; however, one may be able to order the proximities as in (9.5.3) where the ordered quantities are perceived judgments regarding n objects. In addition, we may not be able to specify the relationship between δ_{rs} and d_{rs} as in (9.5.1), but only that

$$d_{rs} = f(\delta_{rs}) + e_{rs} \quad \text{for all } r, s \tag{9.5.5}$$

where the function f is an unknown monotonic increasing function with property (9.5.4). Thus, only the rank order of the distances d_{rs} is used to construct δ_{rs}. Because only the rank order or ordinal relationship of the distances d_{rs} is used in finding the configuration to reconstruct δ_{rs}, the scaling process is termed nonmetric. This has nothing to do with the properties of the space which remains a metric space.

An Introduction to Copulas.

- Let $(X_1, X_2)^{(n)} = (X_{i_1}, X_{i_2})_{i=1}^n$, $(Y_1, Y_2)^{(n)} = (Y_{i_1}, Y_{i_2})_{i=1}^n$ be two given data set.

Goal: Comparing the dependence between between X_1 and X_2 with Y_1 and Y_2.

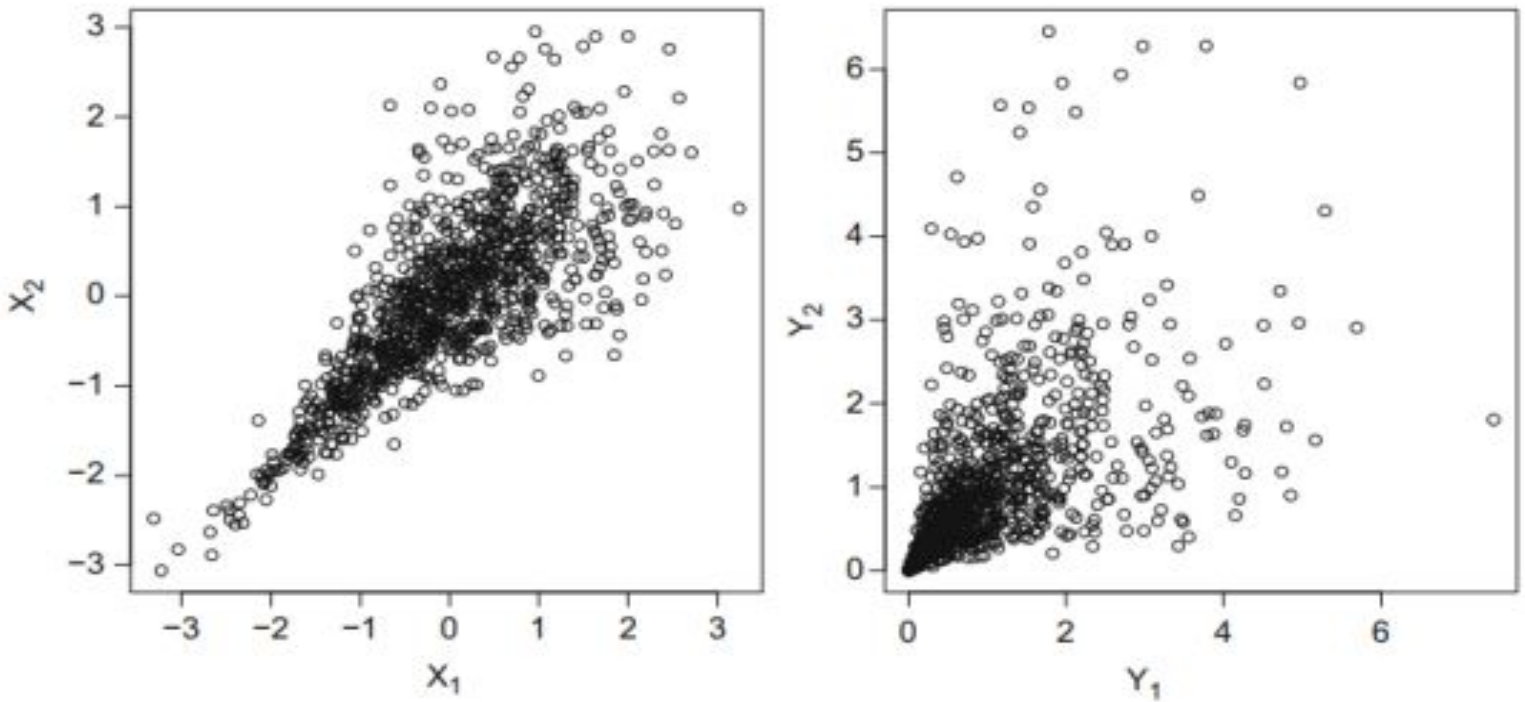


Figure: $n = 1000$ independent observations of (X_1, X_2) and (Y_1, Y_2).

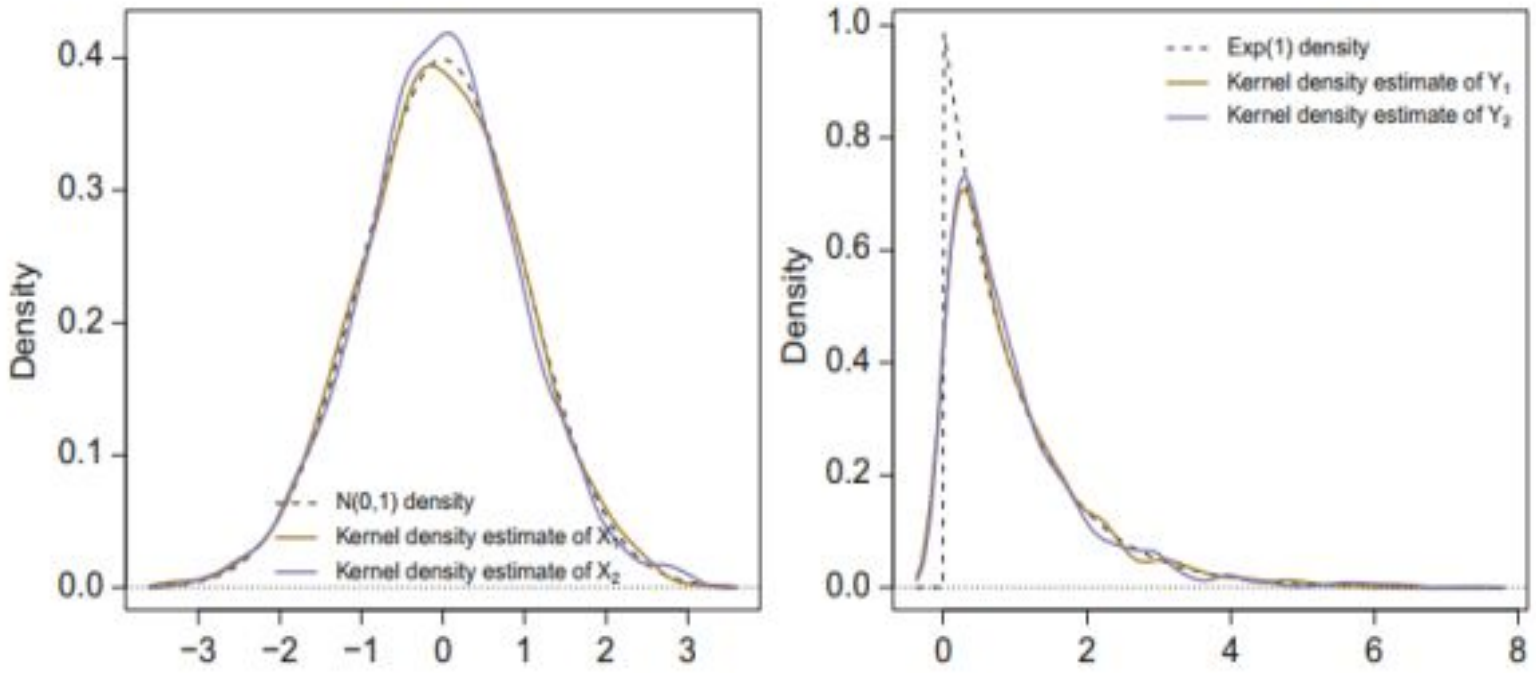


Figure: Kernel density estimates of the densities of (X_1, X_2) and (Y_1, Y_2).

- X_1 and X_2 are likely to follow **standard normal distribution**.
- Y_1 and Y_2 are likely to follow **standard exponential distribution**.

Transform to the uniform distributions.

Lemma: Let X be continuous random variable with distribution function F, then $F(X)$ is a standard uniform random variable, i.e., $F(X) \sim U(0,1)$

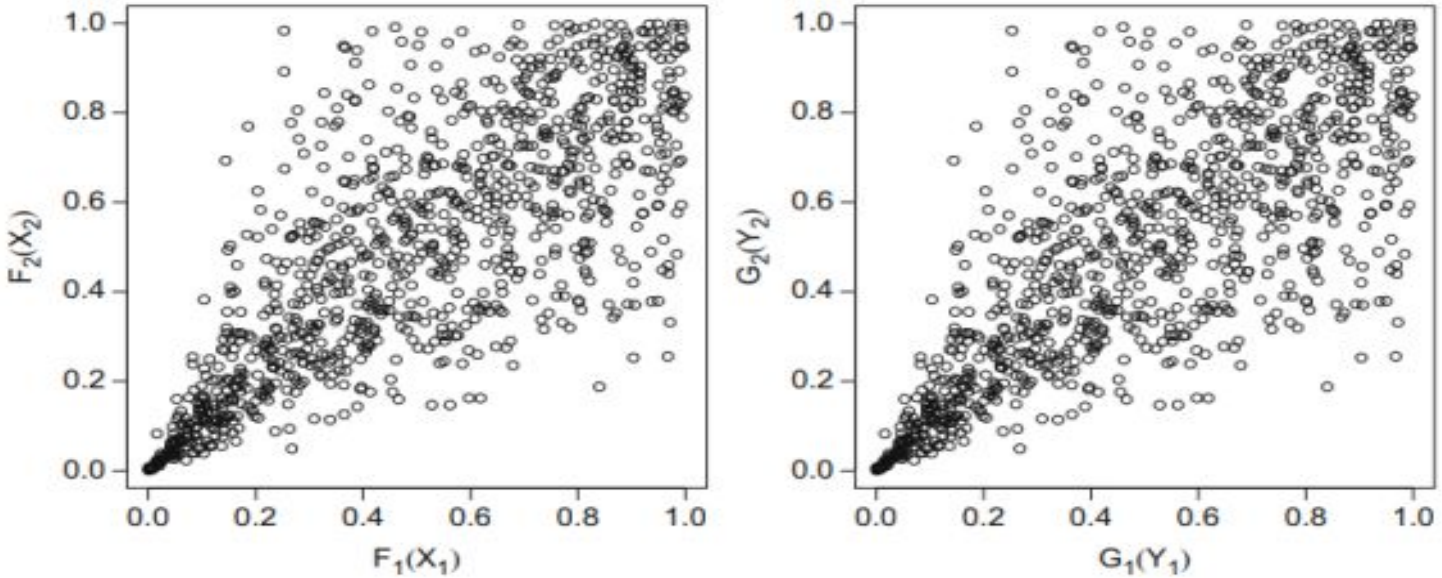


Figure: Scatter plots of $(F(X_1), F(X_2))$ and $(G(Y_1), G(Y_2))$

- The distribution of $(F(X_1), F(X_2))$ and $(G(Y_1), G(Y_2))$ seem to be identical.
- Two given data sets is indistinguishable in terms of **dependence** and only differ in terms of the underlying **marginal distribution function**.

Is there any tool to measure the dependence between random variables?

Duc

- A distribution function is a function $\boldsymbol{F}$ with domain $\overline{\mathbb{R}}$ such that
  - $\boldsymbol{F}$ is nondecreasing
  - $\boldsymbol{F(-\infty) = 0}$ and $\boldsymbol{F(\infty) = 1}$

- A joint distribution function is a function $\boldsymbol{H}$ whose domain $\overline{\mathbb{R}}^{\mathbf{2}}$ such that
  - $\boldsymbol{H}$ is 2 - increasing
  - $\boldsymbol{H(x, -\infty) = 0 = H(-\infty, y)}$ and $\boldsymbol{H(\infty, \infty) = 1}$.

- The margins $\boldsymbol{F}$ and $\boldsymbol{G}$ of $\boldsymbol{H}$ are given by and .
  - $\boldsymbol{F(x) = H(x, \infty)}$
  - $\boldsymbol{G(y) = H(\infty, y)}$

Sklar's Theorem

Let H be a joint distribution function with margins F and G. Then there exists a **copula** C such that for all x, y in $\overline{\mathbb{R}}$,

$$H(x, y) = C(F(x), G(y)). \tag{1}$$

If F and G are continuous, then C is **unique**; otherwise, C is **uniquely determined** on $RanF \times RanG$.

Conversely, if C is a copula and F, G are distributions function, then the function H defined by (1) is a joint distribution function with margins F and G.

Sklar's Theorem

Let X and Y be random variables with distribution functions F and G, respectively, and joint distribution function H. Then there exists a copula C satisfying. Then there exists a **copula** C such that for all x, y in $\overline{\mathbb{R}}$,

$$H(x, y) = C(F(x), G(y)). \tag{2}$$

If F and G are continuous, then C is **unique**; otherwise, C is **uniquely determined** on $RanF \times RanG$.

Lemma: Let X be continuous random variable with distribution function F, then $F(X)$ is a standard uniform random variable, i.e., $F(X) \sim U(0,1)$.

- For any random variable X with distribution function F, the *quantile function* $F^{\leftarrow}$ is defined by

$$F^{\leftarrow}(y) = \inf\{x \in \mathbb{R} : F(x) \geq y\}, \qquad y \in [0,1],$$

with the convention that $\inf \emptyset = \infty$.

Lemma: Let $X \sim U(0,1)$ and F be any distribution function. Then $F^{\leftarrow}(X)$ is a random variable with distribution function F.

- Copula is a multivariate distribution function with standard uniform univariate margins, that is, $\boldsymbol{U(0,1)}$ margins.

- For any $\boldsymbol{(u,v)} \in \boldsymbol{RanF} \times \boldsymbol{RanG}$, (2) can be rewritten as follows

$$\boldsymbol{H(F^{\leftarrow}(u), G^{\leftarrow}(v)) = C(u,v).}$$

- A random vector has a continuous joint distribution function if and only if it has continuous margins.

- If one can estimate the copula and the margin from a given data, then the multivariate is provided.

Example.

Let X, Y be random variables with joint distribution H

$$H(x,y) = \begin{cases} \dfrac{(x+1)(e^y-1)}{x+2e^y-1}, & (x,y) \in [-1,1] \times [0,\infty] \\ 1-e^{-y}, & (x,y) \in (1,\infty] \times [0,\infty] \\ 0, & \text{else where.} \end{cases}$$

with margins F, G given by

$$F(x) = \begin{cases} 0, & x < -1 \\ (x+1)/2, & x \in [-1,1] \\ 1, & x > 1 \end{cases}, \quad G(y) = \begin{cases} 0, & y < 0 \\ 1-e^{-y}, & y \geq 0 \end{cases}$$

Obviously $\boldsymbol{F}, \boldsymbol{G}$ are continuous functions, $\boldsymbol{RanF} = \boldsymbol{RanG} = \mathbb{I}$ and the quantile functions of $\boldsymbol{F}, \boldsymbol{G}$ are

$$F^{\leftarrow}(x) = 2x - 1, \qquad G^{\leftarrow}(y) = -\ln(1 - v).$$

Because $\boldsymbol{RanF} = \boldsymbol{RanG} = \mathbb{I}$, one can find the copula

$$C(u, v) = \frac{uv}{u + v - uv}.$$

Hence,

$$H(x, y) = \frac{F(x)G(y)}{F(x) + G(y) - F(x)G(y)}$$

Using a normal copula –

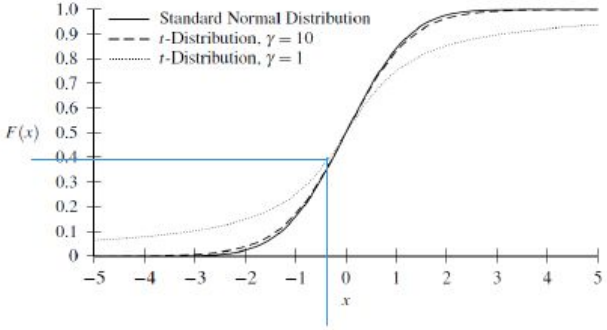
- Assume you know two data sets are linked by a normal copula, ρ=0.7
- You have two assets, each with a *t*-distribution for returns
- Asset x: mean = 0.04, scale* = 0.15, df = 8
- Asset y: mean = 0.07, scale* = 0.17, df = 5
- You want the probability that both assets produce a loss (i.e. return less than zero)

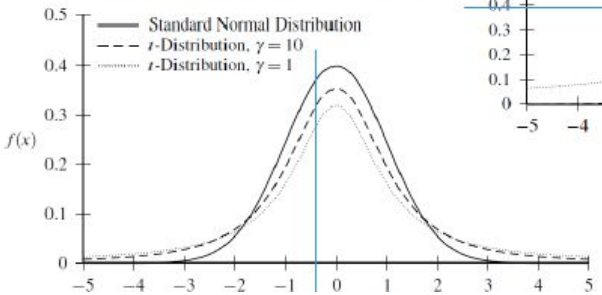
- Step 1: calculate the probabilities that *x* and *y* (that is, $F_x(x)$ and $F_y(y)$ are less than zero (that is $F_x(0)$ and $F_y(0)$)
- Step 2: Convert these probabilities to standard normal variables, so $\Phi^{-1}[F_x(0)]$ and $\Phi^{-1}[F_y(0)]$
- Step 3: Use these as inputs to a standard bivariate normal distribution, so $\Phi_\rho\{\Phi^{-1}[F_x(0)],\Phi^{-1}[F_y(0)]\}$

- Step 1: $F_x(0)$ – probability that X is less than 0 – is 0.3982
  $F_y(0)$ – probability that Y is less than 0 – is 0.3488

Asset x: mean = 0.04, scale* = 0.15, df = 8

$(0 - 0.4)/0.15 = -0.2667$,
$t_8(-0.2667) = 0.3982$

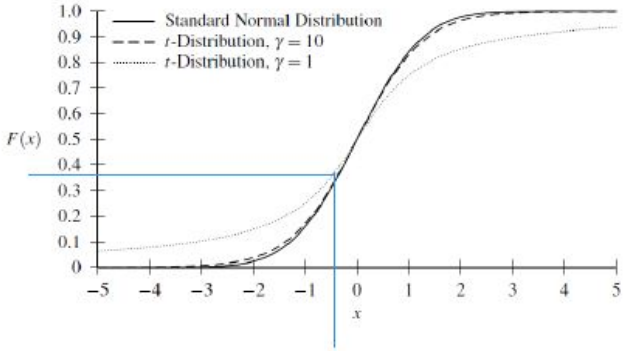


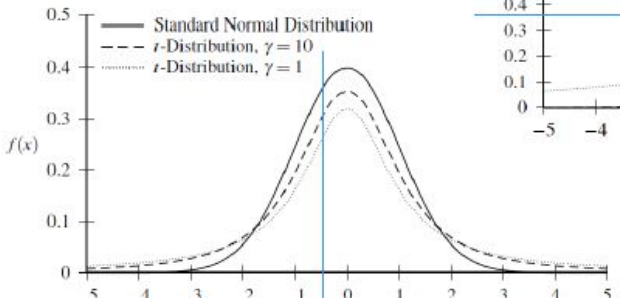


- Step 1: $F_x(0)$ – probability that X is less than 0 – is 0.3982
  $F_y(0)$ – probability that Y is less than 0 – is 0.3488

Asset y: mean = 0.07, scale* = 0.17, df = 5

$(0 - 0.7)/0.17 = -0.4118$
$t_5(-0.4118) = 0.3488$





- Step 1: $F_x(0)$ – probability that X is less than 0 – is 0.3982
  $F_y(0)$ – probability that Y is less than 0 – is 0.3488
- Step 2: $\Phi^{-1}[F_x(0)] = \Phi^{-1}[0.3982] = -0.2579$
  $\Phi^{-1}[F_y(0)] = \Phi^{-1}[0.3488] = -0.3886$
- Step 3: $\Phi_\rho\{\Phi^{-1}[F_x(0)],\Phi^{-1}[F_y(0)]\} =$
  $\Phi_{0.7}\{-0.2579,-0.3886\} =$
  0.2522

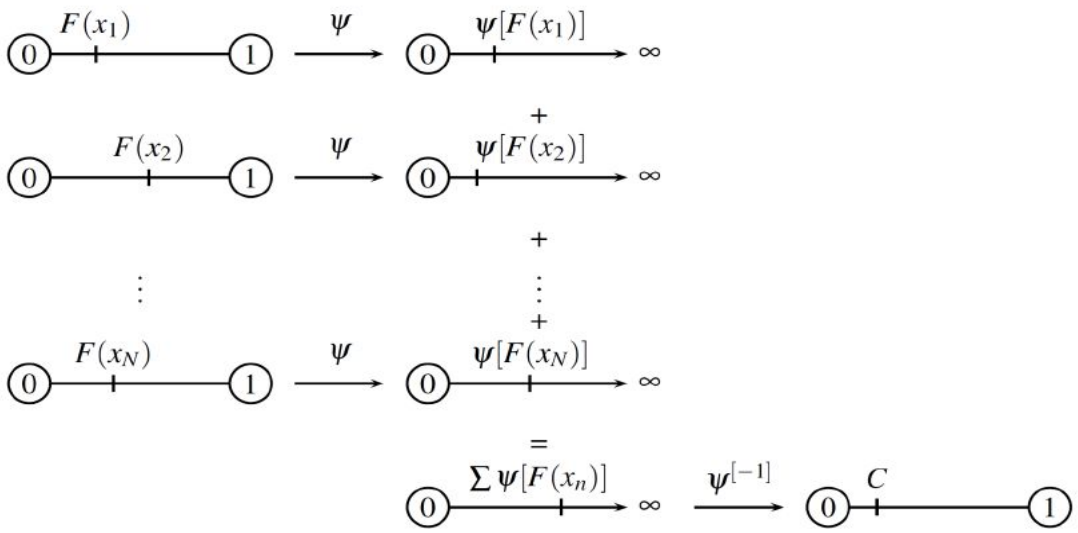
- Taking the t-distributed data
- Stretching each of the two items so that it has a standard normal distribution
- Treating the result as a standard bivariate normal distribution

- The “standard” correlation coefficient (Pearson’s rho) depends on the shape of the distribution…
- …so stretching and reshaping marginal distribution will give different answers for the Pearson’s rho…
- …even if the order of observations is unchanged
- For a normal copula, possible to compromise… …and use rho based on “stretched” data
- Better alternative is to use a different measure – a *rank* correlation coefficient

Archimedean Copulas

- As we've seen, the inputs to bivariate copulas are $F(x)$ and $F(y)$
- For multivariate copulas, just need to extend to $F(x_1)$, $F(x_2)$…$F(x_N)$
- Each of these can take a value from 0 to 1
- Archimedean copulas offer an easy way to combine $F(x_1)$, $F(x_2)$…$F(x_N)$ into a joint probability
- These convert distribution functions into functions that can be added together
- Denoted ψ ("psi")
- Takes each of to $F(x_1)$, $F(x_2)$…$F(x_N)$…
- …and generates a number that lies between 0 and ∞
- When added together, result also between 0 and ∞
- …so using *inverse* generator gives number between 0 and 1…
- …which is the joint probability, $C[F(x_1), F(x_2)…F(x_N)]$

Using an Archimedean copula



- Example: Gumbel copula
- Generator function = $[-\ln F(x)]^{\alpha}$
- So apply that formula to each $F(x)$, sum them, apply inverse generator function to the total
- But what is α?
- $\alpha=1/(1-\tau)$...
- ...where τ is Kendall's tau, a measure of rank correlation

How to choose a copula

- Standard goodness-of-fit tests (e.g. χ^2)
- Maximum likelihood (using copula density function)
- But can also think about appropriate shape
  - Jointly fat tails? t-copula
  - Complex correlation structure? Normal or t-copula
  - Different levels of tail dependence? Archimedean copula
- Beware – joint tails have by definition few observations…
- …and there is also the curse of dimensionality…