

## MULTIVARIATE DATA EXPLORATION

This chapter contains our first excursion away from the simple problems of univariate samples and univariate distribution estimation. We consider samples of simultaneous observations of several numerical variables. We generalize some of the exploratory data analysis tools used in the univariate case. In particular, we discuss histograms and kernel density estimators. Then we review the properties of the most important multivariate distribution of all, the normal or Gaussian distribution. For jointly normal random variables, dependence can be completely captured by the classical Pearson correlation coefficient. In general however, the situation can be quite different. We review the classical measures of dependence, and emphasize how inappropriate some of them can become in cases of significant departure from the Gaussian hypothesis. In such situations, quantifying dependence requires new ideas, and we introduce the concept of copula as a solution to this problem. We show how copulas can be estimated, and how one can use them for Monte Carlo computations and random scenarios generation. We illustrate all these concepts with an example of coffee futures prices. The last section deals with principal component analysis, a classical technique from multivariate data analysis, which is best known for its use in dimension reduction. We demonstrate its usefulness on data from the fixed income markets.

---

### 2.1 MULTIVARIATE DATA AND FIRST MEASURE OF DEPENDENCE

We begin the chapter with an excursion into the world of multivariate data, where dependencies between variables are important, and where analyzing variables separately would cause significant features of the data to be missed. We try to illustrate this point by means of several numerical examples, but we shall focus most of our discussion on the specific example of the daily closing prices of futures contracts on Brazilian and Colombian coffee which we describe in full detail in Subsection 2.2.4 below. We reproduce the first seven rows to show how the data look like after computing the daily log-returns.

	[ , 1]	[ , 2]
[1, ]	-0.0232	-0.0146
[2, ]	-0.0118	-0.0074
[3, ]	-0.0079	-0.0074
[4, ]	0.0275	0.0258
[5, ]	-0.0355	-0.0370
[6, ]	0.0000	0.0000
[7, ]	0.0000	-0.0038

Each row corresponds to a given day, the log return on the Brazilian contract being given in the first column of that row, the log return of the Colombian contract being given in the second one. As we shall see in Subsection 2.2.4, the original data came with time stamps, but as we already explained in the previous chapter, the latter are irrelevant in the static analysis of the marginal distributions. Indeed, for that purpose, the dependence of the measurements upon time does not play any role, and we could shuffle the rows of the data set without affecting the results of this static analysis.

The data set described above is an example of *bivariate* data. We consider examples of multivariate data sets in higher dimensions later in the chapter, but in the present situation, the data can be abstracted in the form of a bivariate sample:

$$(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_n, y_n),$$

which is to be understood as a set of realizations of  $n$  independent couples

$$(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)$$

of random variables with the same joint probability distribution. The goal of this chapter is the analysis of the statistical properties of this joint distribution, and in particular of the dependencies between the components  $X$  and  $Y$  of these couples. Recall from Chapter 1 that if  $X$  and  $Y$  are real valued random variables, then their joint distribution is characterized by their joint cdf which is defined by:

$$(x, y) \mapsto F_{(X,Y)}(x, y) = \mathbb{P}\{X \leq x, Y \leq y\}. \quad (2.1)$$

This joint distribution has a density  $f_{(X,Y)}(x, y)$  if the joint cdf can be written as an indefinite (double) integral:

$$F_{(X,Y)}(x, y) = \int_{-\infty}^x \int_{-\infty}^y f_{(X,Y)}(x', y') dx' dy',$$

in which case the density is given by the (second partial) derivative:

$$f_{(X,Y)}(x, y) = \frac{\partial^2 F_{(X,Y)}(x, y)}{\partial x \partial y}.$$

Setting  $y = +\infty$  in (2.1) leads to a simple expression for the marginal density  $f_X(x)$  of  $X$ . It reads:

$$f_X(x) = \int_{-\infty}^{+\infty} f_{(X,Y)}(x, y') dy'$$

and similarly

$$f_Y(y) = \int_{-\infty}^{+\infty} f_{(X,Y)}(x', y) dx'.$$

### 2.1.1 Density Estimation

The notions of histogram and empirical cdf used in the previous chapter can be generalized to the multivariate setting. Let us discuss the bivariate case for the sake of definiteness. Indeed, one can divide the domain of the couples  $(x_i, y_i)$  into plaquettes or rectangular bins, and create a surface plot by forming cylinders above these plaquettes, the height of each cylinder being proportional to the number of couples  $(x_i, y_i)$  falling into the base. If the lack of smoothness of the one-dimensional histograms was a shortcoming, this lack of smoothness is even worse in higher dimensions. The case of the empirical cdf is even worse: the higher the dimension, the more difficult it becomes to compute it, and use it in a reliable manner. The main drawback of both the histogram and the empirical cdf is the difficulty in adjusting to the larger and larger proportions of the space without data points. However, they can still be used effectively in regions with high concentrations of points. As we shall see later in this chapter, this is indeed the case in several of the *S-Plus* objects used to code multivariate distributions.

#### *The Kernel Estimator*

The clumsiness of the multivariate forms of the histogram is one of the main reasons for the extreme popularity of kernel density estimates in high dimension. Given a sample  $(x_1, y_1), \dots, (x_n, y_n)$  from a distribution with (unknown) density  $f(x, y)$ , the formal kernel density estimator of  $f$  is the function  $\hat{f}_b$  defined by:

$$\hat{f}_b(x, y) = \frac{1}{nb^2} \sum_{i=1}^n K\left(\frac{1}{b}[(x, y) - (x_i, y_i)]\right) \quad (2.2)$$

where the function  $K$  is a given non-negative function of the couple  $(x, y)$  which integrates to one (i.e. a probability density function) which we call the kernel, and  $b > 0$  is a positive number which we call the bandwidth. The interpretation of formula (2.2) is exactly the same as in the univariate case. If  $(x, y)$  is in a region with many data points  $(x_i, y_i)$ , then the sum in the right hand side of (2.2) will contain many terms significantly different from 0 and the resulting density estimate  $\hat{f}_b(x, y)$  will be large. On the other hand, if  $(x, y)$  is in a region with few or no data points  $(x_i, y_i)$ , then the sum in the right hand side of (2.2) will contain only very small numbers and the resulting density estimate  $\hat{f}_b(x, y)$  will be very small. This intuitive explanation of the behavior of the kernel estimator is exactly what is expected from

any density estimator. Notice that the size of the bandwidth  $b$  regulates the extent to which this statement is true by changing how much the points  $(x_i, y_i)$  will contribute to the sum.

### *S-Plus Implementation*

There is no function for multivariate histogram or kernel density estimation in the commercial distribution of *S-Plus*, so we added to our library the function `kdest` which takes a bivariate sample as argument, and produces an *S* object (a data frame to be specific) containing a column for the values of the density estimator, and two columns for the values of the coordinates of the points of the grid on which the estimate is computed. To be specific, if  $X$  and  $Y$  are numeric vectors with equal lengths, the command:

```
> DENS <- kdest(X, Y)
```

produces a data frame with three columns. Selecting these three columns and using one of the 3-D surface plot commands will produce a surface plot of the values of the kernel density estimate over a regular grid of  $256 \times 256$  points covering the range of the bivariate vector  $(X, Y)$ , i.e. at points of the  $(x, y)$ -plane for which  $x$  grows from  $x_{\min}$  to  $x_{\max}$  and  $y$  grows from  $y_{\min}$  to  $y_{\max}$  in  $n = 256$  regular increments. The size of the grid and the default values of the bandwidth parameters can be specified by the user. We illustrate the results of the (bivariate) kernel density estimation with a couple of examples.

- The first example concerns part of a data set which we will study thoroughly in the next chapter. The surface plot of Figure 2.1 is the result of running the command `kdest` on two data vectors  $X$  and  $Y$  derived from the values of indexes computed from the share values and the capitalizations of ENRON and DUKE over the period ranging from January 4, 1993 to December 31, 1993. The well-separated bumps show clearly that the observations  $(x_i, y_i)$  can be divided into several subsets which can be discriminated from each other on the basis of the values of the two variables. This situation is very much sought after in pattern recognition applications where the goal is to subdivide the population into well-defined, and hopefully well separated, clusters which can be identified by their local means, for example.

- Our second example concerns, once more, the daily closing values of the S&P 500 index. The goal is to estimate the joint probability density of the log-return computed on a period of 5 days starting on a given day, and the log-return computed on a period of 15 days ending the same day. The scatterplot of these two variables is given in the left pane of Figure 2.2. From a central blob of points two sparse clouds extend in the direction of the negative  $x$ -axis and the positive  $y$ -axis. The most interesting feature of this scatterplot, however, is the following: the large positive values of the 5 days log-returns follow large negative values of the 15 days log-returns. Anticipating the discussion of the correlation coefficient introduced in the next subsection, we suspect there being a negative correlation between the two returns: indeed computing the correlation between these two variables gives a value approximately equal to

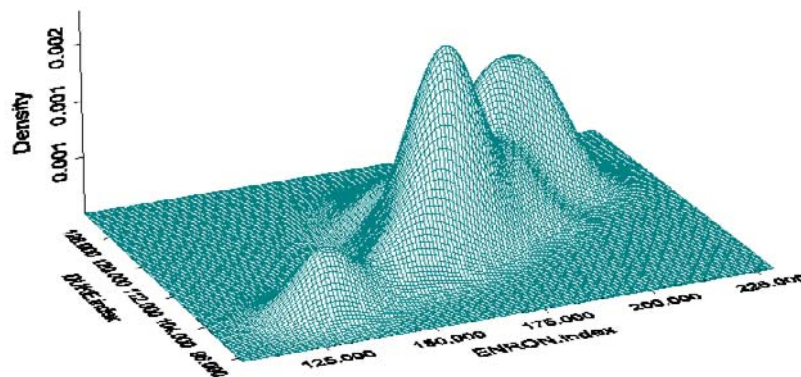
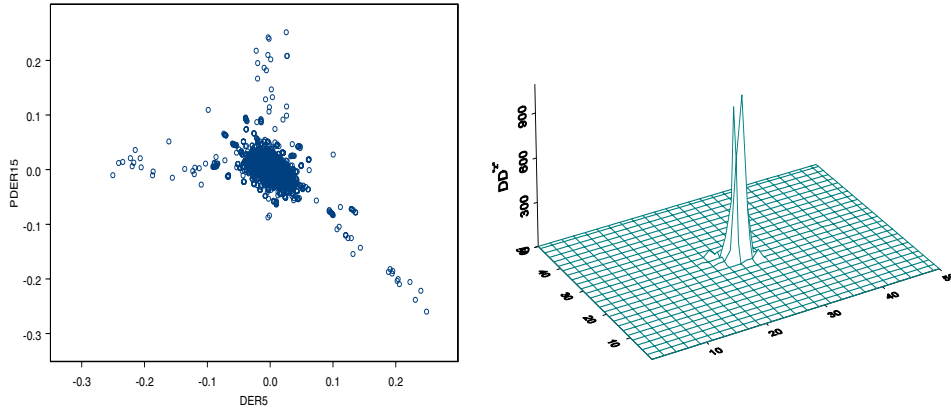


Fig. 2.1. Kernel density estimate for the utility data

— .5. The density estimate reproduced in the right pane shows that the central blob of points appearing in the scatterplot is in fact formed by two separate narrow bumps. But this density estimate fails to reproduce the trail of points in the right part of the scatterplot. As we explained earlier, we believe that these points are responsible for the significant negative correlation, and we do not like seeing them ignored by the kernel density estimator. The problem is very delicate. A smaller bandwidth restores the presence of these points, but the surface would be so rough that the density estimate would be less instructive than the scatterplot itself. On the other hand a larger bandwidth gives a smoother surface, but the latter becomes unimodal, wiping out the signs of possible separate bumps near the center of the distribution. We chose the bandwidth to reach a compromise between these extremes, but as we already explained, we lost the trail of days responsible for the negative correlation. Unfortunately, the serious difficulties experienced in the analysis of this example are typical of many of the real-life applications in which one would like to use density estimation.

### 2.1.2 The Correlation Coefficient

Motivated by the previous discussion of the evidence of a possible linear dependence between variables, we introduce the correlation coefficient between two random variables. This theoretical concept and its empirical counterpart are designed to capture this type of linear dependence. It is the most widely-used measure of dependence between two random variables. It is called the Pearson correlation coefficient. For



**Fig. 2.2.** Scatterplot (left) and kernel density estimate (right) for the 5 days & 15 days S&P log-returns

random variables  $X$  and  $Y$  it is defined as:

$$\rho_P\{X, Y\} = \frac{\text{cov}\{X, Y\}}{\sigma_X \sigma_Y} \quad (2.3)$$

where the covariance  $\text{cov}\{X, Y\}$  is defined by:

$$\text{cov}\{X, Y\} = \mathbb{E}\{(X - \mathbb{E}\{X\})(Y - \mathbb{E}\{Y\})\} = \mathbb{E}\{XY\} - \mathbb{E}\{X\}\mathbb{E}\{Y\} \quad (2.4)$$

and where  $\sigma_X$  and  $\sigma_Y$  denote as usual the standard deviations of  $X$  and  $Y$ , respectively, i.e.

$$\sigma_X = \sqrt{\mathbb{E}\{(X - \mathbb{E}\{X\})^2\}} = \sqrt{\mathbb{E}\{X^2\} - \mathbb{E}\{X\}^2} \quad (2.5)$$

and similarly for  $\sigma_Y$ . If  $X$  and  $Y$  have a joint density  $f(x, y)$  then the definition of the covariance can be rewritten in terms of a double integral as:

$$\text{cov}\{X, Y\} = \int \int xyf(x, y) dx dy - \left( \int xf_X(x) dx \right) \left( \int yf_Y(y) dy \right).$$

Because of its frequent use, the subscript P is often dropped from the notation, and the Pearson correlation coefficient is commonly denoted by  $\rho$ . The empirical analog of this measure of dependence is defined for samples  $x_1, \dots, x_n$  and  $y_1, \dots, y_n$ . By analogy with formula (2.3) it is defined as:

$$\hat{\rho}\{X, Y\} = \frac{\widehat{\text{cov}\{X, Y\}}}{\hat{\sigma}_X \hat{\sigma}_Y} \quad (2.6)$$

and it is called the empirical correlation between the samples. Here, the empirical covariance  $\widehat{\text{cov}\{X, Y\}}$  is defined by:

$$\widehat{\text{cov}\{X, Y\}} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) = \frac{1}{n} \sum_{i=1}^n x_i y_i - \bar{x} \bar{y} \quad (2.7)$$

where we used the notations  $\bar{x}$  and  $\bar{y}$  for the sample means of  $x$  and  $y$  defined by:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{and} \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i, \quad (2.8)$$

and where the sample standard deviations  $\hat{\sigma}_X$  and  $\hat{\sigma}_Y$  are defined by:

$$\hat{\sigma}_X = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2 - \bar{x}^2} \quad (2.9)$$

and similarly for  $\hat{\sigma}_Y$ . Some of the properties of these correlation coefficients are well known. Others are less so. We review them in order to emphasize the usefulness of the correlation coefficient, and at the same time to stress its limitations.

### *Properties of the Correlation Coefficient*

The most immediate properties of the correlation coefficient are:

- The real numbers  $\rho$  and  $\hat{\rho}$  are always between  $-1$  and  $+1$
- $\rho = 0$  when the random variables  $X$  and  $Y$  are independent
- $\rho = 1$  when  $Y$  is a linear function of  $X$ .

These simple properties have lead to the following usage of the sample correlation coefficient  $\hat{\rho}$ . The samples are regarded as independent when  $\hat{\rho}$  is small, while the samples are regarded as strongly dependent when  $\hat{\rho}$  is close to  $1$  or  $-1$ . We shall see below that this practice is okay when the samples come from a multivariate normal distribution, but it can be very misleading for other distributions.

The properties listed in the three bullets above are well known. Their intuitive content is the main reason for the enormous popularity of the correlation coefficient as a measure of dependence.

What is often overlooked is the fact that the Pearson correlation coefficient is only a measure of linear dependence between two random variables. In fact,  $\rho$  measures the relative reduction of the response variation by a linear regression. Indeed, anticipating our upcoming discussion on least squares linear regression, we can use the following general formula

$$\rho\{X, Y\} = \frac{\sigma^2\{Y\} - \min_{\beta_0, \beta_1} \mathbb{E}\{|Y - \beta_0 - \beta_1 X|^2\}}{\sigma^2\{Y\}}$$

to justify this claim. The numerator of the right hand side is the difference between the variation in the variable  $Y$ , and the smallest possible remaining variation after removing a linear function  $\beta_0 + \beta_1 X$ , of  $X$ . This formula gives the slope of the least squares regression line of  $Y$  against  $X$  in terms of  $\rho_P$ .

Finally, we close this section with a very surprising property of the Pearson correlation coefficient. Strangely enough, this property is little known despite its important practical implications, especially in the world of financial models. If the marginal distributions of  $X$  and  $Y$  are given, but no information is given on the nature of their dependence or lack thereof, the possible values of the correlation coefficient  $\rho$  are limited to an interval  $[\rho_{min}, \rho_{max}]$ . However, contrary to popular belief, this interval is not always the whole interval  $[-1, +1]$ . There are cases for which this interval is much smaller, even for frequently-used distributions. See for example Problems 2.3 and 2.7 at the end of this chapter, where the case of lognormal random variables is analyzed in detail.

---

## 2.2 THE MULTIVARIATE NORMAL DISTRIBUTION

We start our analysis of multivariate statistical distributions with the case of the well-known normal family. All the reasons we gave for the popularity of the univariate normal distribution still hold in the multivariate case. Moreover, the possible competition from other distribution families vanishes. Indeed, the normal family is essentially the only one for which explicit analytic computations are possible. We first give an abstract definition and concentrate on the interpretation of the consequences of such a definition. Even though most of the explicit computations done in the book will be limited to the bivariate case, we start with the general definition of the multivariate normal distribution because of its widespread use in portfolio theory where realistic situations involve very large numbers of instruments. Because of this general setup, the discussion which follows is of rather abstract nature, and a quick look at the contents of Appendix 1 at the end of the chapter may help with some of the mathematics.

One says that  $k$  real valued random variables  $Z_1, \dots, Z_k$  are jointly normal, or that their distribution is a multivariate normal distribution, if the joint density of  $Z_1, \dots, Z_k$  is given by:

$$f_{(Z_1, \dots, Z_k)}(z_1, \dots, z_k) = \frac{1}{\sqrt{(2\pi)^k \det(\Sigma)}} \exp \left( -\frac{1}{2} [\mathbf{z} - \mu]^t \Sigma^{-1} [\mathbf{z} - \mu] \right) \quad (2.10)$$

for some  $k \times k$  invertible matrix  $\Sigma$  and a  $k$ -dimensional vector  $\mu$ . In this formula we used the notation  $\mathbf{Z}$  for the  $k$ -dimensional vector whose components are the  $Z_i$ 's. The above definition is usually encapsulated in the notation:

$$\mathbf{Z} \sim N_k(\mu, \Sigma)$$

to signify that the random vector has the  $k$ -variate normal distribution with mean vector  $\mu$  and variance/covariance matrix  $\Sigma$ . This terminology is consistent with the standard practice of probability calculus with random vectors and matrices, which we recall in Appendix 1 at the end of the chapter.  $\mu$  is the  $k \times 1$  vector of means



$\mu_i = \mathbb{E}\{Z_i\}$  and  $\Sigma$  is the variance/covariance matrix whose entries are  $\Sigma_{i,j} = \text{cov}\{Z_i, Z_j\}$ . Using the convention introduced in the appendix, this reads:

$$\mathbb{E}\{\mathbf{Z}\} = \mu \quad \text{and} \quad \Sigma_{\mathbf{Z}} = \Sigma.$$

According to its definition (2.17), the entries of the covariance matrix  $\Sigma_{\mathbf{Z}}$  are the covariances  $\text{cov}\{Z_i, Z_j\}$ , and consequently, the knowledge of all the marginal (bivariate) distributions of the couples  $(Z_i, Z_j)$  is enough to determine the entire joint distribution. This particular property is specific to the multivariate normal distribution. It does not hold for general distributions. Moreover, the matrix calculus developed for random vectors in Appendix 1 implies that:

$$\mathbf{Z} \sim N_k(\mu, \Sigma) \quad \text{when} \quad \mathbf{Z} = \mu + \Sigma^{1/2} \mathbf{X} \quad \text{and} \quad \mathbf{X} \sim N_k(0, \mathbf{I}_k) \quad (2.11)$$

where  $\mathbf{I}_k$  denotes the  $k \times k$  identity matrix, and  $\Sigma^{1/2}$  denotes the square root of the symmetric nonnegative-definite matrix  $\Sigma$ . See Problem 2.8 at the end of the chapter for details on the definition and the first properties of this square root matrix. In other words, starting from a random vector  $\mathbf{X}$  with independent  $N(0, 1)$  components (see the following remark) we can get to a vector  $\mathbf{Z}$  with the most general multivariate normal distribution just by linear operations: multiplying by a matrix and adding a vector. This simple fact is basic for the contents of the following subsection.

### Important Remark about Independence

If the random variables  $Z_i$  are independent, then obviously all the covariances  $\text{cov}\{Z_i, Z_j\}$  are zero when  $i \neq j$ , and the variance/covariance matrix  $\Sigma_{\mathbf{Z}}$  is diagonal. The converse is not true in general, *even when the random variable  $Z_i$ 's are normal* ! See for example Problem 2.5 for a counter-example. But *the converse is true when the  $Z_i$ 's are jointly normal* ! This striking fact highlights what a difference it makes to assume that the marginal distributions are normal, versus assuming that the joint distribution is normal. The proof of this fact goes as follow: if  $\Sigma_{\mathbf{Z}}$  is diagonal, then:

$$\frac{1}{2}[\mathbf{z} - \mu]^t \Sigma^{-1} [\mathbf{z} - \mu] = \frac{(z_1 - \mu_1)^2}{2\sigma_1^2} + \frac{(z_2 - \mu_2)^2}{2\sigma_2^2} + \dots + \frac{(z_k - \mu_k)^2}{2\sigma_k^2},$$

if we denote by  $\sigma_1^2, \sigma_2^2, \dots, \sigma_k^2$  the elements which appear on the diagonal of  $\Sigma_{\mathbf{Z}}$ . So using the definition (2.10) of the multivariate normal distribution, this implies that:

$$f_{(Z_1, Z_2, \dots, Z_k)}(z_1, z_2, \dots, z_k) = f_{Z_1}(z_1) f_{Z_2}(z_2) \cdots f_{Z_k}(z_k),$$

which in turn implies that the joint cdf is the product of the marginal cdf's, proving the desired independence property. So the conclusion is that:

*For jointly normal random variables, independence is equivalent to the variance/covariance matrix being diagonal !*

#### 2.2.1 Simulation of Random Samples

We now show how one can use formula (2.11) to generate random samples from a multivariate normal distribution. To that end, we assume that we are given a  $k \times 1$

vector  $\mu$  of means, and a  $k \times k$  variance/covariance matrix  $\Sigma$ , and that we want to generate a sample of size  $N$  from the distribution  $N_k(\mu, \Sigma)$ . We proceed as follows:

1. We create a  $k \times N$ -matrix whose columns are all identical, any column being a copy of the mean vector  $\mu$ ;
2. We generate a sample of size  $Nk$  from the standard normal distribution and reshape this  $(N \times k) \times 1$  vector into a  $k \times N$ -matrix;
3. We compute a square root for the variance/covariance matrix, then we multiply each column of the random matrix constructed in Step 2, by the square root of the variance/covariance matrix;
4. We add the matrix of means constructed in Step 1 to the random matrix constructed in Step 3 above.

Notice that Step 3 (which is the most involved) is not needed when  $\Sigma = \mathbf{I}_k$ . The details of this random generation algorithm are given in Problem 2.8 at the end of the chapter. There, we show how to compute the square root of a covariance matrix and we develop the code for a *home grown* function capable of generating samples from a multivariate normal distribution. Writing this code is just for the sake of illustration, since S-Plus provides the function `rmvnorm` whose use we illustrate in Subsection 2.2.3 below.

### 2.2.2 The Bivariate Case

In the bivariate case we have:

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \quad \text{and} \quad \Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$$

and consequently, the joint density can be written as:

$$f_{(Z_1, Z_2)}(z_1, z_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left(-\frac{1}{1-\rho^2} \left[ \frac{(z_1 - \mu_1)^2}{2\sigma_1^2} - \rho \frac{(z_1 - \mu_1)(z_2 - \mu_2)}{\sigma_1\sigma_2} + \frac{(z_2 - \mu_2)^2}{2\sigma_2^2} \right]\right).$$

This formula shows that, if we know the marginal distributions of  $Z_1$  and  $Z_2$ , in other words, if we know  $\mu_1$ ,  $\sigma_1$ ,  $\mu_2$  and  $\sigma_2$ , then the joint distribution is entirely determined by the correlation coefficient  $\rho$ . Also, we clearly see from this formula that when  $\rho = 0$  we have:

$$\begin{aligned} f_{(Z_1, Z_2)}(z_1, z_2) &= \frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\frac{(z_1 - \mu_1)^2}{2\sigma_1^2} - \frac{(z_2 - \mu_2)^2}{2\sigma_2^2}\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{(z_1 - \mu_1)^2}{2\sigma_1^2}\right) \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{(z_2 - \mu_2)^2}{2\sigma_2^2}\right) \\ &= f_{Z_1}(z_1)f_{Z_2}(z_2) \end{aligned}$$

which shows the independence of  $Z_1$  and  $Z_2$ . So we recover the fact that if  $Z_1$  and  $Z_2$  are jointly Gaussian, their independence is equivalent to their being uncorrelated. As we already pointed out, this fact is not true in general, not even when  $Z_1$  and  $Z_2$  are (separately) Gaussian. See Problem 2.5 for a counterexample.

### 2.2.3 A Simulation Example

For the sake of illustration, we consider the case of the distribution of a couple of (slightly correlated) normal random variables  $X$  and  $Y$ , and we generate one sample of size  $n = 128$  from the joint distribution of  $(X, Y)$ . We use the S-Plus command:

```
> TSAMPLE<-rmvnorm(n=128,mean=rep(0,2),sd=rep(1,2),rho=.18)
> TDENS <- kdest(TSAMPLE[,1],TSAMPLE[,2])
```

The function `rmvnorm` is the multivariate analog of `rnom`. It is designed to generate multivariate normal samples. We chose the vector  $[0, 0]$  for the mean by using the command `rep(0, 2)` which creates a vector by repeating the number zero twice, and we used the command `rep(1, 2)` to specify that both components have standard deviations equal to one. Finally, we decided on the correlation coefficient  $\rho = .18$  by setting the parameter `rho`. We could have given the entire variance/covariance matrix by specifying the parameter `cov` instead of giving the vector of standard deviations and the correlation coefficient separately. See the help file for details. The second command computes a kernel density estimator (with the default kernel and bandwidth choices) and plots the resulting *surface*. The output

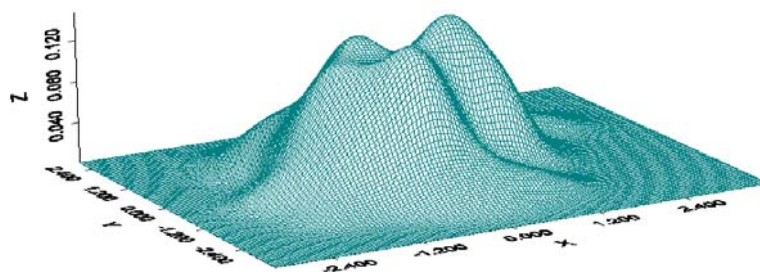


Fig. 2.3. Kernel Density Estimator for a (Bivariate) Normal Sample

is given in Figure 2.3. We see that the unimodality of the density is violated by the estimate which seems to indicate the presence of several bumps. Increasing the bandwidth would resolve this problem, at the cost of a looser fit, by somewhat flattening

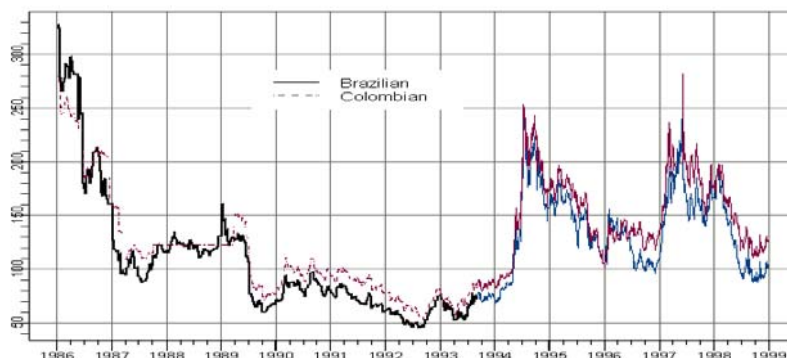
the central bump. The poor quality of the estimation is to be blamed on the small size of the sample: in general, the higher the dimension, the larger is the sample size needed to get reasonable density estimates.

### 2.2.4 Let's Have Some Coffee

We use Paul Erdős' famous quote:

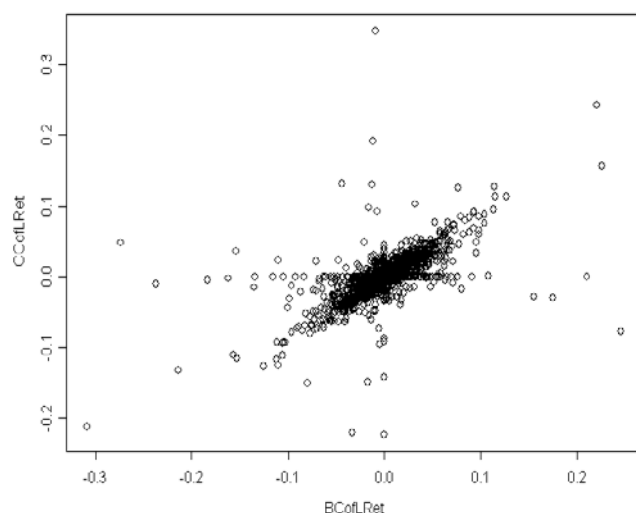
*A mathematician is a machine that turns caffeine into theorems*

as a justification for our interest in the price of coffee. As explained in the abstract at the beginning of the chapter, we chose to illustrate the analysis of multivariate distributions with a simple example of two quantities which are obviously correlated. We use samples of log-returns of Brazilian and Colombian coffee spot prices. The original data are plotted in Figure 2.4, which shows the daily spot prices of coffee in Brazil and Colombia between January 9, 1986 and January 1, 1999. We do not



**Fig. 2.4.** Sequential plot of the daily prices of coffee in Brazil and Colombia from January 9, 1986 to January 1, 1999.

give the *S-Plus* commands used to produce Figure 2.4 as they involve time series objects which we will consider only in Part III of the book. The data of interest to us in this section are contained in the *S* objects *BCofLRet* and *CCofLRet*. They are the two columns of a data matrix given at the beginning of the first section of this chapter. For each day of the period starting January 10, 1986 and ending January 1, 1999, we computed the logarithms of the daily returns from the nearest futures contract active on that day. The scatterplot of these two variables is given in Figure 2.5. A close look at this scatterplot shows that many points are on the vertical axis and on the horizontal axis. This means that quite often, the price does not change from one day to the next, forcing the log returns to vanish on these days. The presence



**Fig. 2.5.** Scatterplot of the daily log-returns of the coffee futures contracts in Brazil and Colombia from January 10, 1986 to January 1, 1999.

of so many of these zeroes indicates that the probability distributions are singular, in the sense that the cumulative distribution functions have jumps at 0. These jumps can be a hindrance to the analysis, so we choose to remove them by removing the zeroes from the data samples.

```
> NZ <- (BCofLRet != 0 & CCofLRet != 0)
> BLRet <- BCofLRet[NZ]
> CLRet <- CCofLRet[NZ]
```

The vector `NZ` created by the first command is a boolean vector with the same length as `BCofLRet` and `CCofLRet`. It is true (equal to `T`) when both the daily Brazilian and Colombian log-returns are non-zero. The next two commands show the power of the sub-scripting capabilities of the `S` language. `BLRet` and `CLRet` are the vectors obtained by keeping the entries of `BCofLRet` and `CCofLRet` whose indices are those for which the value of `NZ` is `T`. The scatterplot of `BLRet` and `CLRet` is reproduced in the left pane of Figure 2.6.

We shall work with this new bivariate sample from now on, but we should keep in mind that, if we want to compute statistics of the actual log-returns, we need to put the zeroes back. The command

```
> PNZ <- mean(NZ)
> PNZ
[1] 0.4084465
```

gives the proportion of T's in the vector NZ, and it should be viewed as an estimate of the probability not to have a zero in the data. This probability could be used to add a random number of zeroes should we need to create random samples from the original distribution using samples from the modified distribution.

### 2.2.5 Is the Joint Distribution Normal?

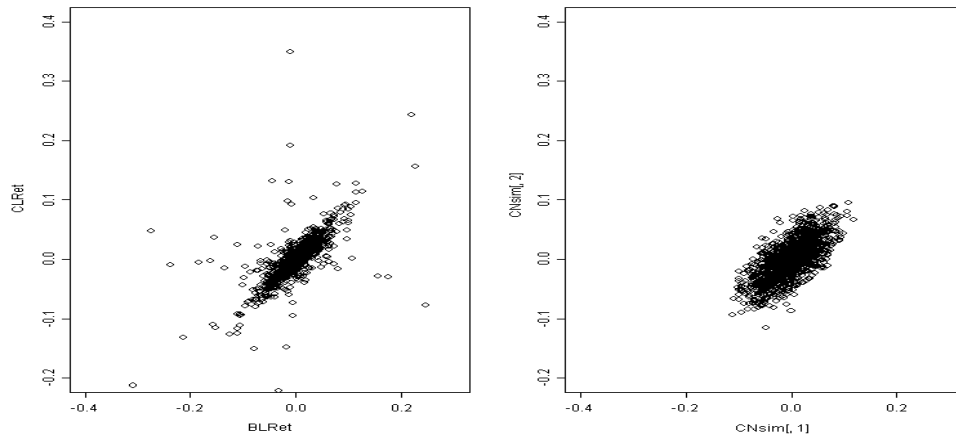
The first question we address is the normality of the joint distribution of BLRet and CLRet. Our first step is quite mundane: we compare graphically the joint (empirical) distribution of BLRet and CLRet to the distribution of a bivariate normal sample which has the same five parameters (i.e. means, standard deviations and correlation coefficient). In order to do so, we first compute these parameters. We use the commands:

```
> XLIM <- c(-.4, .3)
> YLIM <- c(-.2, .4)
> Mu <- c(mean(BLRet), mean(CLRet))
> Sigma <- var(cbind(BLRet, CLRet))
> N <- length(BLRet)
```

We defined the vectors XLIM and YLIM as the limits of the ranges of BLRet and CLRet, respectively. We use these values each time we want to make sure that the scatterplot of BLRet and CLRet on one hand and the scatterplot of the simulated data on the other are on the same scale. As defined, Mu is the mean vector since it is defined as the vector of the means. Next we use the S-Plus function cbind to bind the columns BLRet and CLRet into one single matrix, then applying the function var to this data matrix produces the variance/covariance matrix of the columns. So Sigma is the variance/covariance matrix, and N is the sample size. We use the S-Plus function rmvnorm to generate the desired bivariate sample  $(x_1, y_1), \dots, (x_N, y_N)$  of size  $N$  from the bivariate Gaussian distribution with mean Mu and variance/covariance matrix Sigma.

```
> CNSim <- rmvnorm(N, mean=Mu, cov=Sigma)
> par(mfrow=c(2,1))
> plot(BLRet, CLRet, xlim=XLIM, ylim=YLIM)
> plot(CNSim, xlim=XLIM, ylim=YLIM)
> par(mfrow=c(1,1))
```

Notice that we could just as well have used as well the home-grown function vnorm developed in Problem 2.8 at the end of the chapter. The results are given in Figure 2.6. Both scatterplots comprise an ellipsoidal cloud of points around the origin. Clearly, this cloud seems to be thinner for the coffee data. However, even if we were to consider that the bulk of the distribution had been reproduced in a reasonable manner, the presence of isolated points in the empirical coffee data is a distinctive feature which has not been reproduced by the simulation. This is a clear indication that the joint distribution of BLRet and CLRet is not normal. There are many reasons why



**Fig. 2.6.** Comparison of the empirical scatterplot of the coffee log-returns after the removal of the zeroes (left) and of the scatterplot of the sample of the same size simulated from a jointly Gaussian distribution with the same mean and covariance structure (right).

this could be the case. In general, it is because at least one of the variables, BLRet or CLRet, is not normal. But these variables could be normally distributed even when the joint distribution is not normal. We now check that this is not the case by showing that the marginal distributions of BLRet and CLRet are not normal either.

---

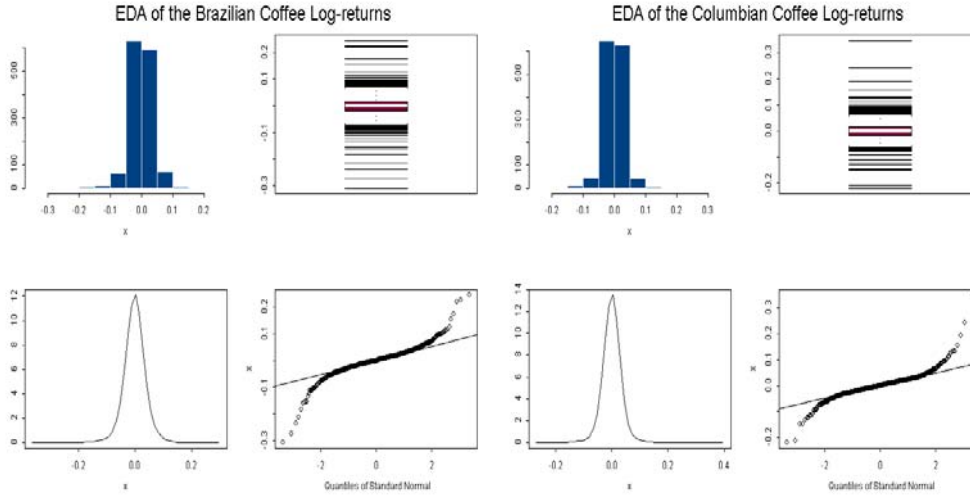
## 2.3 MARGINALS AND MORE MEASURES OF DEPENDENCE

Trying to fit a multivariate distribution to a multivariate sample, as we did when trying to fit a bivariate normal distribution to the sample of BLRet and CLRet, may be trying to tackle all the difficulties at once, and may be overwhelming. So instead, we try the *divide and conquer* approach: we first consider the estimation of the univariate marginal distributions separately, and only after this has been done, do we consider the issue of dependence. In the case of a bivariate sample from a normal distribution, this would amount to first estimating the means and the variances of the two variables separately, and then estimating the correlation coefficient. Since we are interested in more general distributions, possibly with marginals having heavy tails, we may not be able to use the correlation coefficient as a way to quantify dependence. To this end, we review the most commonly used statistics measuring the dependence of two samples, and we prepare for the concept of copula which will be introduced and analyzed in the next section.

As before, we try to sprinkle the presentation of the mathematical concepts with numerical examples, and we still use the example of the coffee data for that.

### 2.3.1 Estimation of the Coffee Log-Return Distributions

We use the graphical tools introduced in Chapter 1, as encapsulated in the `S-Plus` function `eda.shape` defined in appendix at the end of the book, as part of the introduction to `S-Plus`. The results of the applications of the function `eda.shape` to `BLRet` and `CLRet` are reproduced in Figure 2.7. In both cases one sees clearly that they differ significantly from the results of Figure 7.27 obtained from a normal sample, in the introductory session to `S-Plus` reproduced in appendix to this book. The histograms and kernel density estimates vouch for a unimodal distri-



**Fig. 2.7.** Exploratory Data Analysis of the Brazilian (left) and Colombian (right) coffee daily log-returns for the period from January 9, 1986 to January 1, 1999 after removal of the zeroes.

bution, possibly with extended tails on both sides. The presence of tails which are heavier than normal is confirmed by the boxplots, which show a very large number of observations outside the box. However, when it comes to the tails, the clearest diagnostic is given by the Q-Q plots. The departures from the Q-Q lines are a clear indication that the tails are much heavier than the tails of the normal distributions with the same means and variances, so fitting of a generalized Pareto distribution may be appropriate. Since the analysis of the marginal distribution of the daily log returns of the Colombian coffee is essentially the same, we only report the analysis of the Brazilian coffee.

**Remark.** Similar results would have been obtained with the original log return samples (prior to the removal of the zeroes) since the zeroes affect only the center of the distribution and not the tails. The main difference would appear in the density plots. Indeed, these density plots would seem *absent* and this requires an explanation.

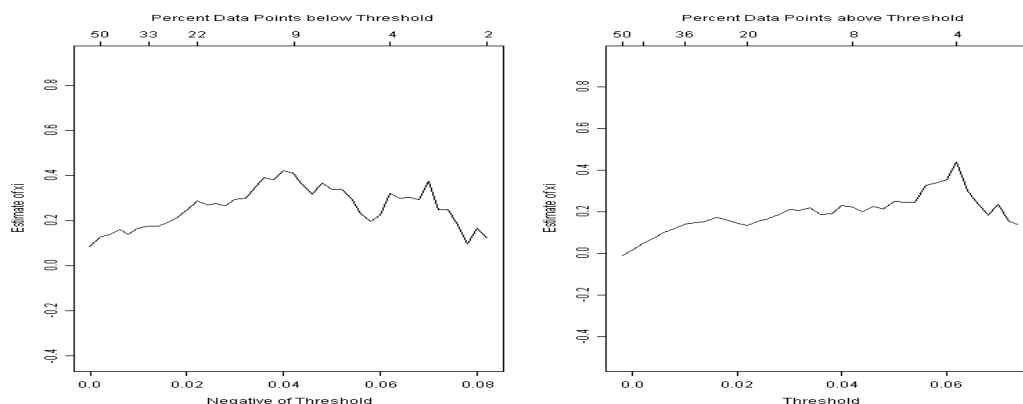


A closer look at the command used in the code of the function `eda.shape` reminds us that the plot of the density estimate is restricted to the inter-quantile interval. If we compute this interval for the data at hand, we see that the lower and the upper quantile are essentially 0. This explains why no graph would be produced by the density estimator.

As in the case of our analysis of the S&P 500 daily log-returns, we use the function `gpd.tail` to fit a generalized Pareto distribution to both `BLRet` and `CLRet`. This function requires information on the locations of the tails in the form of two parameters telling the program where these tails should start. We explained how to choose these thresholds from a Q-Q plot of the data. We emphasize now the use of the function `shape.plot` as an alternative tool. In practice we recommend a combination of the two approaches to pick values for these thresholds. The commands

```
> par(mfrow=c(1,2))
> shape.plot(BLRet,tail="lower")
> shape.plot(BLRet,tail="upper")
> par(mfrow=c(1,1))
```

produce the results given in Figure 2.8. From these plots we decide that the estima-

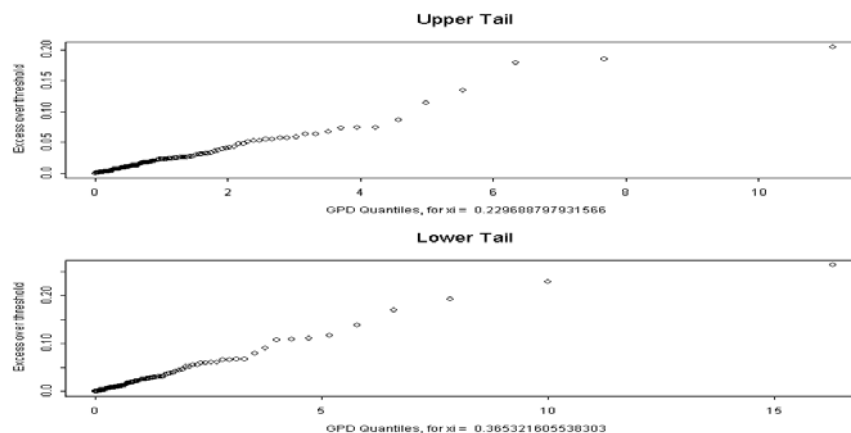


**Fig. 2.8.** Maximum likelihood estimates of the shape parameters as functions of the thresholds for the left tail (left pane) and right tail (right pane) of the distribution of the daily log returns of the Brazilian coffee.

tion of the upper tail could be done to the right of the value .04, while the estimation of the lower tail could be done to the left of the value  $-.045$ . Among other things, this will guarantee that each tail contains a bit more than 8% of the data, namely more than 115 points, which is not unreasonable. With this choice we proceed to the actual estimate of the GPD with the command:

```
> B.est <- gpd.tail(BLRet, upper = 0.04, lower = -0.045)
```

The function `gpd.tail` creates Q-Q plots (reproduced in Figure 2.9) of excesses over lower and upper thresholds against the quantiles of a GPD. As we have mentioned several times, if the left parts of these plots are approximately linear, the estimation of the tail is expected to be good. These empirical facts can be justified by mathematical results which are beyond the scope of this book. Also, these mathematical results require the independence of the successive entries in the data set. In other words, our analysis ignores the serial dependence between the successive log returns. This is usually not a great mistake, and also we have rigorous results to back up this claim. We check graphically the quality of the fit with the plots of the tails on



**Fig. 2.9.** Estimation of the distribution of the daily log returns of the Brazilian coffee by a generalized Pareto distribution.

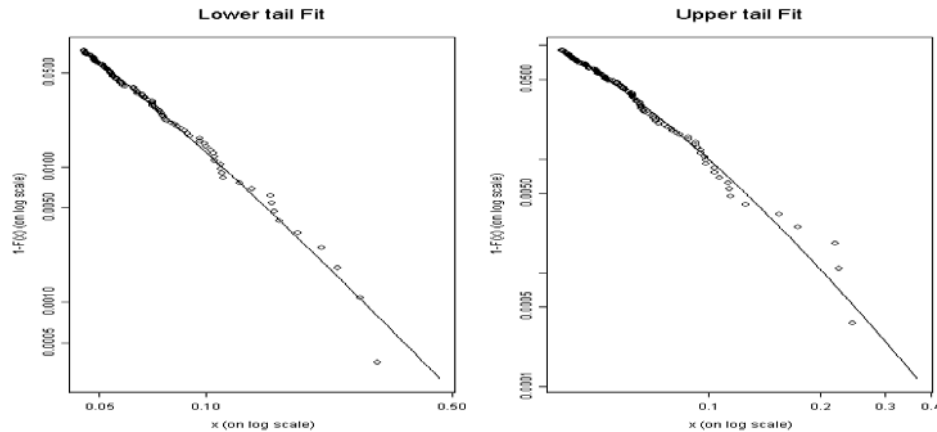
a logarithmic scale.

```
> par(mfrow=c(1,2))
> tailplot(B.est,tail="lower")
> title("Lower tail Fit")
> tailplot(B.est,tail="upper")
> title("Upper tail Fit")
> par(mfrow=c(1,1))
```

The results are given in Figure 2.10. Given the point patterns, the fit looks very good. We perform the analysis of the heavy tail nature of the distribution of the daily log-returns of the Colombian coffee in exactly the same way.

### *First Monte Carlo Simulations*

Motivated by the desire to perform a simulation analysis of the risk associated with various coffee portfolios containing both Brazilian and Colombian futures contracts,



**Fig. 2.10.** Goodness of the fits for the left tail (left pane) and the right tail (right pane).

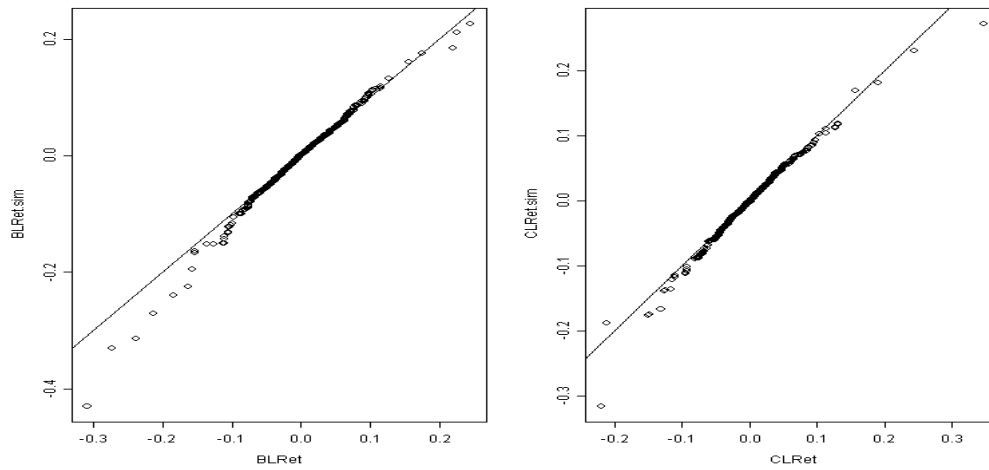
we proceed to the Monte Carlo simulation of samples of log-returns using the tools developed in the previous chapter. The commands:

```
> BLRet.sim <- gpd.2q(runif(length(BLRet)), B.est)
> CLRet.sim <- gpd.2q(runif(length(CLRet)), C.est)
```

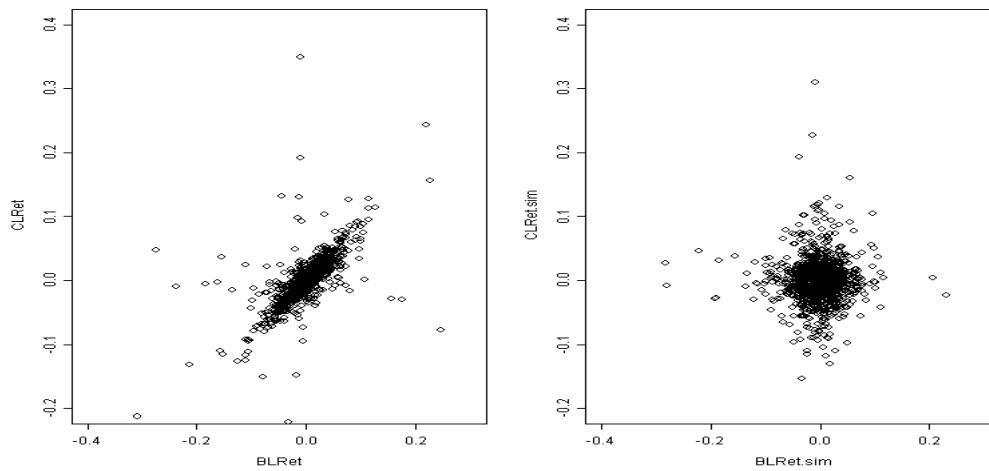
generate samples `BLRet.sim` and `CLRet.sim` of the same sizes as the original data, and from the GPD's fitted to the data. To make sure that these samples have the right distributions, we check that their Q-Q plots against the empirical data are concentrated along the main diagonal. This is clear from Figure 2.11 which was produced with the S-Plus commands:

```
> qqplot(BLRet, BLRet.sim)
> abline(0, 1)
> qqplot(CLRet, CLRet.sim)
> abline(0, 1)
```

So it is clear that the distributions of the simulated samples are as close as we can hope for from the empirical distributions of the Brazilian and Colombian coffee log-returns. Since they capture the marginal distributions with great precision, these simulated samples can be used for the computations of statistics involving the log-returns separately. However, they cannot be used for the computations of joint statistics since they do not capture the dependencies between the two log-returns. Indeed, the simulated samples are statistically independent. This is clearly illustrated by plotting them together in a scatterplot as in Figure 2.12. We need to work harder to understand better the dependencies between the two log-return variables, and to be able to include their effects in Monte Carlo simulations.



**Fig. 2.11.** Empirical Q-Q plot of the Monte Carlo sample against the empirical coffee log-return sample in the case of the Brazilian futures (left pane) and the Colombian futures prices (right pane).



**Fig. 2.12.** Scatterplot of the Colombian coffee log-returns against the Brazilian ones (left pane), and scatterplot of the Monte Carlo samples (right pane).

### 2.3.2 More Measures of Dependence

Because of the limitations of the correlation coefficient  $\rho$  as a measure of the dependence between two random variables, other measures of dependence have been

proposed and used throughout the years. They mostly rely on sample order statistics. For the sake of completeness, we shall quote two of the most commonly used: the Kendall's  $\tau$  and the Spearman's  $\rho$ . Given two random variables  $X$  and  $Y$ , their Kendall's correlation coefficient  $\rho_K(X, Y)$  is defined as:

$$\rho_K(X, Y) = \mathbb{P}\{(X_1 - X_2)(Y_1 - Y_2) > 0\} - \mathbb{P}\{(X_1 - X_2)(Y_1 - Y_2) < 0\} \quad (2.12)$$

provided  $(X_1, Y_1)$  and  $(X_2, Y_2)$  are independent random couples with the same joint distribution as  $(X, Y)$ . Even though the notation  $\rho_K$  should be used for consistency, we shall often use the notation  $\tau(X, Y)$  because this correlation coefficient is usually called Kendall's tau.

The dependence captured by Kendall's tau is better understood on sample data. Given samples  $x_1, \dots, x_n$  and  $y_1, \dots, y_n$ , the empirical estimate of the Kendall correlation coefficient is given by:

$$\hat{\rho}_K(X, Y) = \frac{1}{\binom{n}{2}} \sum_{1 \leq i < j \leq n} \text{sign}((x_i - x_j)(y_i - y_j))$$

which shows clearly that what is measured here is merely the relative frequency with which a change in one of the variables is accompanied by a change in the same direction of the other variable. Indeed, the `sign` appearing in the right hand side is equal to one when  $x_i - x_j$  has the same sign as  $y_i - y_j$ , whether this sign is plus or minus, independently of the actual sizes of these numbers. Computing this coefficient with `S-Plus` can be done in the following way:

```
> cor.test(BLRet, CLRet, method="k")$estimate
      tau
0.71358
```

The Spearman rho of  $X$  and  $Y$  is defined by:

$$\rho_S(X, Y) = \rho\{F_X(X), F_Y(Y)\}, \quad (2.13)$$

and its empirical estimate from sample data is defined as:

$$\hat{\rho}_S(X, Y) = \frac{12}{n(n^2 - 1)} \sum_{i=1}^n \left( \text{rank}(x_i) - \frac{n+1}{2} \right) \left( \text{rank}(y_i) - \frac{n+1}{2} \right).$$

The value of this correlation coefficient depends upon the relative rankings of the  $x_i$  and the  $y_j$ . However, the interpretation of the definition is better understood from the theoretical definition (2.13). Indeed, this definition says that the Spearman's correlation coefficient between  $X$  and  $Y$  is exactly the Pearson's correlation coefficient between the uniformly distributed random variables  $F_X(X)$  and  $F_Y(Y)$ . This shows that Spearman's coefficient attempts to remove the relative sizes of the values of  $X$  among themselves, similarly for the relative values of  $Y$ , and then to capture what is left of the dependence between the transformed variables. We shall come back to this approach to dependence below. Spearman's rho is computed in `S-Plus` with the command:

```
> cor.test(BLRET, CLRET, method="s")$estimate
rho
0.8551515
```

---

## 2.4 COPULAS AND RANDOM SIMULATIONS

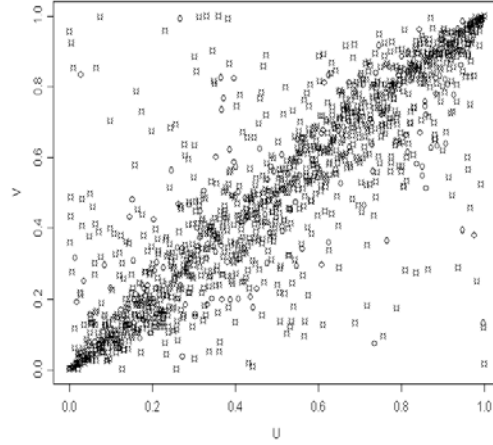
The first part of this section elaborates on the rationale behind the introduction of Spearman's correlation coefficient. As a warm up to the introduction of the abstract concept of copula, we consider first the practical implication of the first of the two fundamental facts of the theory of random generation as presented in Section 1.6. Because of Fact 1, which reads:

$$X \text{ r.v. with cdf } F_X(\cdot) \implies F_X(X) \text{ uniform on } [0, 1],$$

we can transform the original coffee log-return data and create a bivariate sample in the unit square in such a way that both marginal point distributions are uniform. Indeed, the above theoretical result says that this can be done by evaluating each marginal cdf exactly at the sample points. In some sense, this wipes out the dependence of the point cloud pattern, seen for example in the left pane of Figure 2.6, upon the marginal distributions, leaving only the intrinsic dependence between the variables. We use our estimates of the distribution functions of the coffee log-returns as proxies for the theoretical distributions.

```
> U <- gpd.2p(BLRet, B.est)
> V <- gpd.2p(CLRet, C.est)
> plot(U, V)
> EMPCOP <- empirical.copula(U, V)
```

The first two commands use the function `gpd.2p` from the `FinMetrics` module, to compute the estimate of the cdf of the GPD, identified by the object of class `gpd`, at the points given in its first argument. Figure 2.13 shows the result of the above `plot` command. As expected all the data points are in the unit square. Moreover, the first coordinates of the points seem to be uniformly distributed on the unit interval of the horizontal axis, as they should be according to Fact 1, which was recalled above. Similarly for the second coordinates. The fact that the marginal distributions are now uniform is a sign that the influences of the original marginal distributions have been removed from the data. The only remaining feature is the way the numbers  $u_i$  and  $v_i$  are paired, and we claim that the dependence between the two log-returns is captured by the way these couplings are done. The dense point concentration around the second diagonal of the unit square is a consequence of this pairing, and it should be viewed as a graphical representation of the intrinsic dependence between the two random variates.



**Fig. 2.13.** Dependence between the coffee log-returns after removing the effects of the marginal distributions.

### 2.4.1 Copulas

The above discussion motivates the following abstract definition for capturing the dependence between several random variates. The analysis of the coffee data will be continued in Subsection 2.4.4.

**Definition 1.** A copula is the joint distribution of uniformly distributed random variables.

If  $U_1, \dots, U_n$  are  $U(0, 1)$ , then the function  $C$  from  $[0, 1] \times \dots \times [0, 1]$  into  $[0, 1]$  defined by:

$$C(u_1, \dots, u_n) = \mathbb{P}\{U_1 \leq u_1, \dots, U_n \leq u_n\}$$

is a copula. Moreover, if  $X_1, \dots, X_n$  are r.v.'s with cdf's  $F_{X_1}, \dots, F_{X_n}$  respectively, then the copula of the uniform random variables

$$U_1 = F_{X_1}(X_1), \dots, U_n = F_{X_n}(X_n)$$

is called the copula of  $(X_1, \dots, X_n)$ . Copulas are typically used in the bivariate case  $n = 2$ , or for  $n$  very large. The case  $n$  large is of crucial importance for the analysis of the risk of large portfolios of financial instruments whose distributions are not well accounted for by normal distributions. Unfortunately, except for the normal and the Student copulas no other single copula is available in high dimension for analysis. However, despite this limitation, we present the first properties of copulas in the general case. This choice is justified by the crucial importance of risk-management applications. On the other hand, a complete analysis is possible in the case  $n = 2$ ,

that offers a complete description of all the possible forms of dependence between two random variables. We take that up in Subsection 2.4.3 below.

### *First Properties of Copulas*

It is straightforward to check that:

- $C$  does not change if one replaces any of the  $X_i$ 's by a non-decreasing functions of itself;
- The joint cdf can be recovered from the copula and the marginal cdf's via the formula:

$$F_{(X_1, \dots, X_n)}(x_1, \dots, x_n) = C(F_{X_1}(x_1), \dots, F_{X_n}(x_n));$$

- $C$  is unique if  $F_{(X_1, \dots, X_n)}(x_1, \dots, x_n)$  is continuous (no jumps).

Moreover:

- $C(u_1, \dots, u_n)$  is non-decreasing in each variable  $u_i$  since it is a cdf;
- $C(1, \dots, 1, u_i, 1, \dots, 1) = u_i$  for all  $i$  since the marginal distributions of a copula are uniformly distributed;
- if  $u_i \leq v_i$  for all  $i$ , then:

$$\sum_{1 \leq i_1 \leq 2} \dots \sum_{1 \leq i_n \leq 2} (-1)^{i_1 + \dots + i_n} C(u_{i_1}, \dots, u_{i_n}) \geq 0.$$

This last inequality is very technical. It is reproduced here only for the sake of completeness. It essentially formalizes mathematically the fact that a copula is a multivariate cdf, and as such, it has to satisfy some positivity and monotonicity properties. It is instructive to visualize the meaning of this condition in the bivariate case  $n = 2$  for which one can easily check that it holds true on a picture!

### 2.4.2 First Examples of Copula Families

The following are examples of copulas.

#### ◇ *Independent copula*

$$C_{ind}(u_1, \dots, u_n) = u_1 \dots u_n.$$

This is the copula of independent random variables.

#### ◇ *Gaussian copula* For each $\rho \in [-1, 1]$ the function defined by:

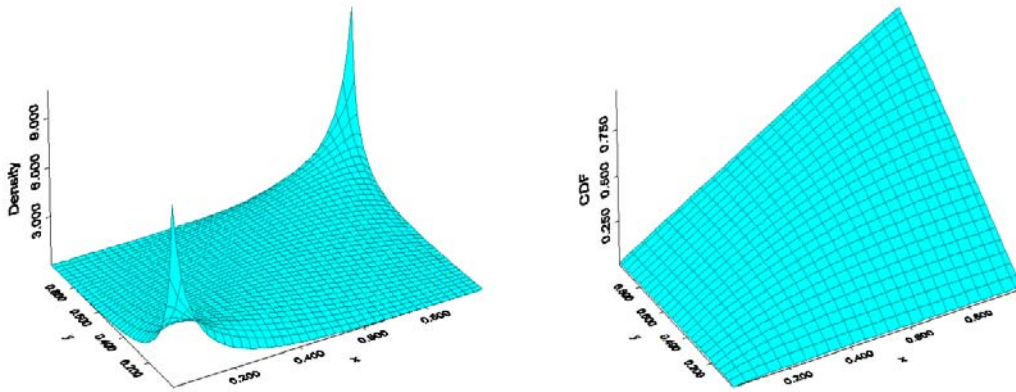
$$C_{Gauss, \rho}(u_1, u_2) = \frac{1}{2\pi\sqrt{1-\rho^2}} \int_{-\infty}^{\Phi^{-1}(u_1)} \int_{-\infty}^{\Phi^{-1}(u_2)} e^{-[s^2 - 2\rho st + t^2]/2(1-\rho^2)} ds dt$$

is a copula called the Gaussian copula with parameter  $\rho$ . This is the copula of random variables which, whether or not their marginal distributions are Gaussian, depend upon each other as jointly Gaussian random variables do. The family of normal copulas is parameterized by the parameter  $\rho \in [-1, +1]$ . Figure 2.14 gives



the surface plot of this copula when  $\rho = .7$ , i.e. the plot of the graph of the function  $(u, v) \mapsto C_{Gauss,.7}(u, v)$ , together with the plot of its density. The fact that the marginals of a copula are uniform is clearly seen on this plot. Indeed, the facts  $C(u, 1) = u$  and  $C(1, v) = v$  force edges of the surface to be linear (coinciding with the second diagonal) and to meet at height 1 above the point  $(1, 1)$ .

Varying the parameter  $\rho$  is a way of varying the degree of dependence between the two random variables. Notice that two normal random variables  $X$  and  $Y$  may not be *jointly normal* if their copula is not in the normal family. They are jointly normal when the copula is from the normal family, in which case the parameter  $\rho$  has a simple interpretation since it is the correlation coefficient of  $X$  and  $Y$ . This interpretation is not valid in general. Indeed, if  $X$  and  $Y$  have Cauchy marginal distributions, their (Pearson) correlation coefficient does not exist since Cauchy random variables do not have means or variances . . . , but nevertheless, it is quite possible for their copula to be in the Gaussian family, i.e. to be equal to  $C_{Gauss,\rho}$  for some  $\rho \in [-1, +1]$ . However, in this case, the parameter  $\rho$  cannot have the interpretation of correlation coefficient. Notice that, even though we only gave the formula in the bivariate case,



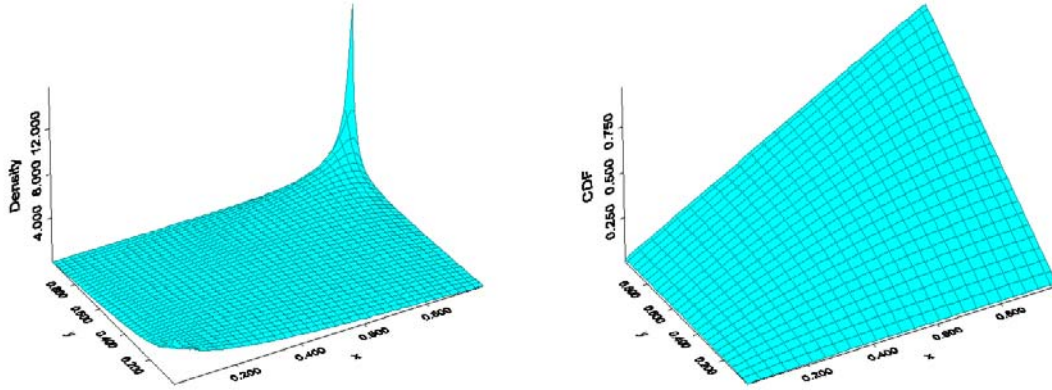
**Fig. 2.14.** Surface plot of the normal copula with parameter  $\rho = .7$  (right), and of its density (left).

it is quite easy to define the Gaussian copula for any number of dimensions, the parameter now being a correlation matrix instead of a mere scalar.

◇ For each  $\beta \in [0, 1]$  the function

$$C_{Gumbel,\beta}(u_1, u_2) = e^{-[(-\log u_1)^{1/\beta} + (-\log u_2)^{1/\beta}]}$$

is a copula called the **Gumbel** (or **logistic**) copula with parameter  $\beta$ . The Gumbel



**Fig. 2.15.** Surface plot of the Gumbel copula with parameter  $\beta = 1.5$  (right), and of its density (left).

copula family is parameterized by the parameter  $\beta$ , however the latter does not have as nice an interpretation as the parameter  $\rho$  of the Gaussian copula family. Figure 2.15 gives the surface plot of this copula when  $\beta = 1.5$  together with the plot of its density. As before, varying the parameter is a way of varying the *strength* of the dependence between two random variables. This family appears naturally in the analysis of extreme events, as it is quite often the case that the copula of random variables with heavy tail distributions is of this family.

A complete list of the parametric copula families supported by the library EVANESCE and the module FinMetrics is given in Appendix 2, where the reader will also find the defining formulae together with some of the most important properties of these families.

### 2.4.3 Copulas and General Bivariate Distributions

The goal of this subsection is to show how copulas and univariate cdf's come together to characterize ALL the models of bivariate statistical distributions.

All the copulas which we consider in this book have a density. In other words, the copulas  $C(u, v)$  will be differentiable, and the function:

$$c(u, v) = \frac{\partial^2}{\partial u \partial v} C(u, v)$$

will be the density of the copula. Notice that since we are dealing with bivariate distributions, we need a second order derivative in order to get a density from its cdf. Instead of limiting ourselves to distributions with uniform marginals, we can apply

this remark to a general bivariate distribution as well. This leads to some interesting formulae.

Let us denote by  $F_{(X,Y)}$  the joint cdf of a couple  $(X, Y)$  of random variables, and let us denote by  $C_{(X,Y)}$  their copula. For the sake of simplicity we momentarily drop the subscript  $(X, Y)$  from the notation. According to our definition, we have:

$$F(x, y) = C(F_X(x), F_Y(y)) \quad (2.14)$$

if we denote by  $F_X$  and  $F_Y$  the cdf's of  $X$  and  $Y$  respectively. We can compute the joint density  $f_{(X,Y)}$  of  $X$  and  $Y$  by taking partial derivatives on both sides of (2.14). We get:

$$\begin{aligned} f(x, y) &= \frac{\partial^2}{\partial x \partial y} F(x, y) \\ &= \frac{\partial^2}{\partial u \partial v} C(F_X(x), F_Y(y)) \frac{\partial F_X(x)}{\partial x} \frac{\partial F_Y(y)}{\partial y} \end{aligned}$$

which gives the following formula for the joint density of  $X$  and  $Y$ :

$$f(x, y) = c(F_X(x), F_Y(y)) f_X(x) f_Y(y) \quad (2.15)$$

in terms of the density of their copula, their marginal cdf's and their marginal densities. Obviously we used the formulae

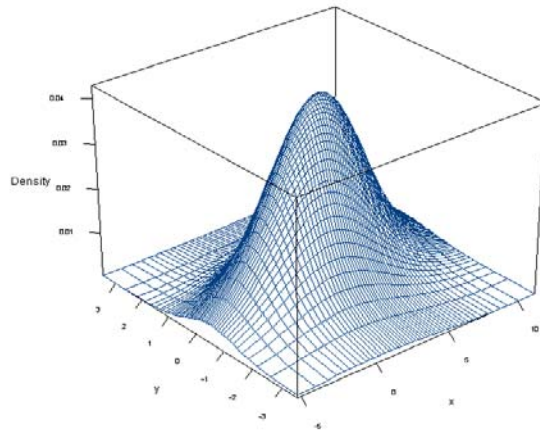
$$f_X(x) = \frac{d}{dx} F_X(x) \quad \text{and} \quad f_Y(y) = \frac{d}{dy} F_Y(y)$$

giving the densities of  $X$  and  $Y$  in terms of their respective cdf's. Formula (2.15) has the following interesting consequence. Contrary to what can be done with the correlation coefficient (see Problem 2.3 and Problem 2.7 for the striking example of the lognormal distributions), it is *always* possible to specify a bivariate distribution by specifying:

- the marginal distributions
- a copula

without having to worry about the existence of the distribution. Moreover, as formulae (2.14) and (2.15) show, formulae for the components can be used to get formulae for the cdf and the density of the bivariate distribution. Figure 2.16 shows an example where we computed the density of a joint distribution specified by the Gumbel copula with parameter 1.4, and with the normal distribution  $N(3, 4)$  and the Student  $t$ -distribution  $T(3)$  as marginals. A bivariate distribution can be created with the command `bivd`, and the function `persp.dbivd` can be used to produce a 3-D surface plot of the density of a bivariate distribution. The plot of Figure 2.16 was obtained with the *S-Plus* commands:

```
> BIV1 <- bivd(gumbel.copula(1.4), "norm", "t", c(3,4), 3)
> persp.dbivd(BIV1)
```



**Fig. 2.16.** Surface plot of the density of the bivariate distribution with Gumbel copula with parameter 1.4 and marginal distributions  $N(3, 4)$  and  $T(3)$ .

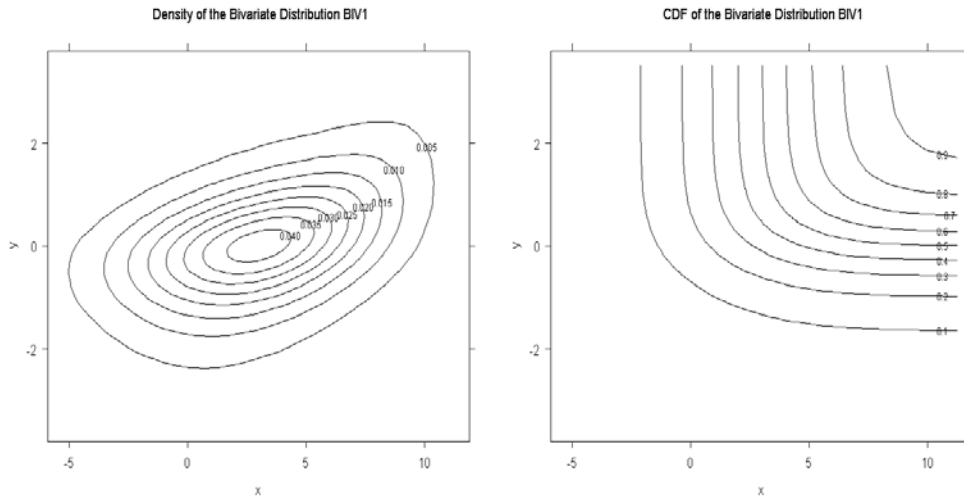
The function `persp.pbivd` produces a surface plot of the cdf of the bivariate distribution, but these plots are not very instructive, especially for copulas, as we can see from Figure 2.14 and Figure 2.15. Indeed, all these copula surface plots show a *tent* tied at the level 0 on the segments going from the origin  $(0, 0)$  to the points  $(0, 1)$  and  $(1, 0)$ , and it is also tied at the point  $(1, 1)$  where its value is always 1, and it is linear on the two coordinate planes. These properties are mere re-statements of the first properties of copulas given in Subsection 2.4.1. These constraints are common to all the copula surface plots, so it is extremely difficult to differentiate between them from the plots of their cdf's. For this reason, one very often uses contour plots to get a sense of the shape of the distribution and possibly to compare several copulas, or more general bivariate distributions. The commands:

```
> par(mfrow=c(1,2))
> contour.dbivd(BIV1)
> title("Density of the Bivariate Distribution BIV1")
> contour.pbivd(BIV1)
> title("CDF of the Bivariate Distribution BIV1")
> par(mfrow=c(1,1))
```

were used to produce the plots of Figure 2.17.

#### 2.4.4 Fitting Copulas

Because of the serious difficulties resulting from the lack of data in the tails of the marginal distributions, copulas are best estimated by parametric methods. In order



**Fig. 2.17.** Contour plot of the density (left) and the cdf (right) of the bivariate distribution with Gumbel copula with parameter 1.4 and marginal distributions  $N(3, 4)$  and  $T(3)$ .

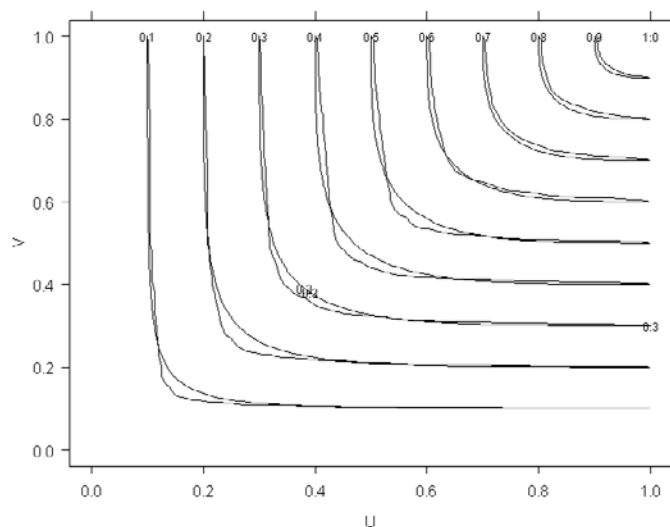
to do so, we choose a family of copulas (see Appendix 2 for a list of the parametric families supported by the library `EVANESCE`) and the module `FinMetrics`, and we estimate the parameter(s) of the family in question by a maximum likelihood standard estimation procedure. The function `fit.copula` which we use below returns an object of class `copula.family` and creates a contour plot of level sets of the empirical copula and of the fitted copula. Since it is so difficult to compare copulas using only their graphs, in order to visualize the goodness of the fit, we chose to put a selected ensemble of level sets for the two surfaces on the same plot. Differences in these level curves can easily be interpreted as differences between the two surfaces. The fitting procedure can be implemented in the case of the coffee log-return data by the following commands.

```
> FAM <- "gumbel"
> ESTC <- fit.copula(EMPCOP, FAM)
```

Recall that `EMPCOP` was the `S-Plus` object constructed as the empirical copula of the Brazilian and Colombian coffee daily log-returns. The results are shown in Figure 2.18. The level sets of the Gumbel copula fitted to the data are very close to the level sets of the empirical copula. This graphical check shows that the fit is very good.

### 2.4.5 Monte Carlo Simulations with Copulas

We learned in Chapter 1 how to generate random samples from a univariate distribution, and we introduced and tested a set of tools to do just that, even when the



**Fig. 2.18.** Contour plot of the empirical copula with the contours of the fitted Gumbel copula superimposed.

distribution in question had to be estimated from data, and even when the distribution was suspected to have heavy tails. But as we saw earlier (recall Figure 2.12) having separate univariate random samples is not enough if we want to have a realistic rendering of how the variables in a bivariate sample relate to each other.

In this subsection, we consider the problem of the generation of random samples from a bivariate distribution, which we assume to be given by its marginal distributions and a copula. Let us imagine that we have a tool capable of producing bivariate samples from a copula. We shall not enter into the details of the construction of such a tool, we shall merely indicate that it can be built by aggregation of one dimensional random generators for the various conditional distributions. The gory details of such a construction are too technical for this book, so we shall leave them aside. Armed with such a weapon, it is very simple to generate samples from all the distributions having this specific copula as their own copula. Indeed, the first components of a random sample from a copula form a univariate sample uniformly distributed on  $[0, 1]$ . So transforming this sample by computing the quantile function of the first marginal distribution will turn this uniform sample into a sample from the first marginal distribution. This is an instance of our favorite method for generating random samples. Similarly, transforming the second components (which also form a uniform sample, by definition of a copula) by computing the quantile function of the second marginal distribution will give a random sample from the second marginal distribution. Now, by the very definition of the copula, these two univariate samples have not only the

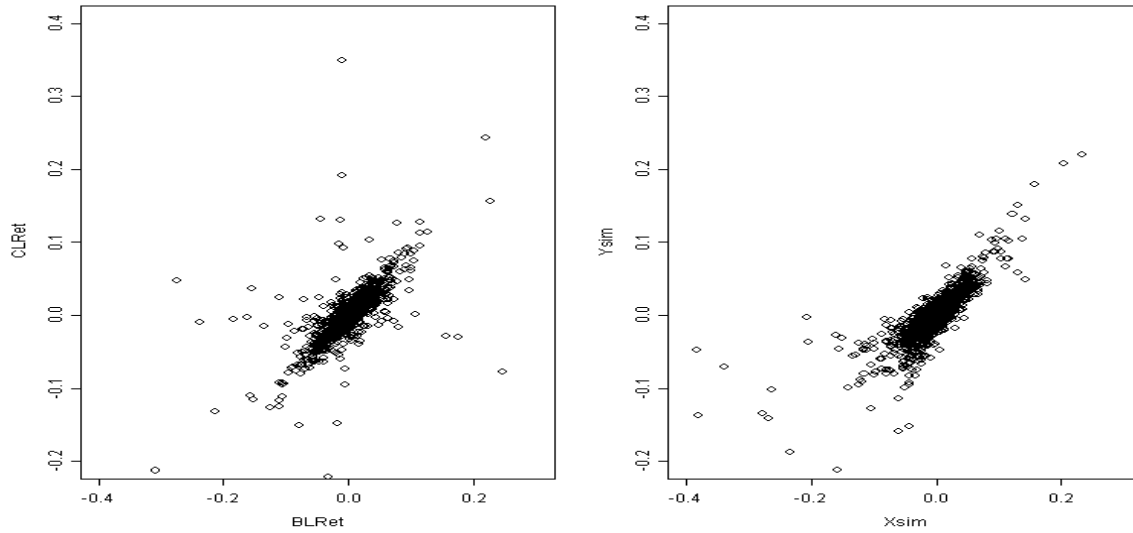
right marginals, but they also have the right copula! So put together, they form a bivariate sample from the bivariate distribution we started with.

We implement this strategy on our example of the coffee log returns. The following set of commands produces a bivariate sample of same size as the data, from our estimation of the joint distribution of the coffee log-returns. Remember that this estimate is comprised of the estimates of the marginal distributions of the two random quantities together with the parametric estimate of their copula.

```
> N <- length(BLRet)
> SD <- rcopula(ESTC,N)
> Xsim <- gpd.2q(SD$x, B.est)
> Ysim <- gpd.2q(SD$y, C.est)
```

We review the main steps of the simulation before commenting on the plots. The function `rcopula` produces bivariate samples from the copula whose information is encapsulated in the argument `ESTC`, which needs to be an object of class `copula.object`. As usual, we extract the two columns of the  $N \times 2$  matrix `SD` by means of the dollar signs `$` followed by the lower case `x` for the first column, and by the lower case `y` for the second column. By definition of a copula, `SD$x` and `SD$y` are random samples uniformly distributed over the unit interval. Consequently, the third and fourth commands result in samples `Xsim` and `Ysim` from the GPD's given by the `gpd` objects `BEST` and `CEST`. This is because we compute quantile functions on uniform samples. We already used this trick several times to generate random samples from a given distribution. But the situation is quite different from those earlier in that the uniform samples are paired by the copula used to generate them. So the copula of the resulting bivariate sample is the copula we started from. The loop is closed, and we have produced a bivariate sample with the desired distribution. In the same way we produced Figure 2.12, we can place the scatterplot of the simulated samples `Xsim` and `Ysim` to the right of the scatterplot of the original samples `BLRet` and `CLRet`. The result is reproduced in Figure 2.19. The differences with Figure 2.12 are striking. This plot shows that our model and the ensuing simulations capture rather well the characteristics of the point distribution in the plane. As further evidence, the numerical measures of dependence which we introduced earlier confirm that the results are very satisfactory. This is clear from the comparison of the values of the Kendall's tau and Spearman's rho statistics computed for the empirical copula (directly from the data) and from the fitted copula. We reproduce the commands and the results:

```
> print(ESTC)
Gumbel copula family; Extreme value copula.
Parameters :
  delta = 2.98875657681924
> Kendalls.tau(EMPCOP)
[1] 0.6881483
> Kendalls.tau(ESTC, tol = 1e-2)
[1] 0.6654127
```



**Fig. 2.19.** Scatterplot of the Colombian coffee log-returns against the Brazilian ones (left pane), and scatterplot of the Monte Carlo samples produced with the dependence as captured by the fitted copula (right pane).

```
> Spearman.rho(EMPCOP)
[1] 0.8356747
> Spearman.rho(ESTC)
[1] 0.8477659
```

#### 2.4.6 A Risk Management Example

Even though the practical example treated in this subsection is limited in scope, it should not be underestimated. Indeed, the risk measures which we compute provide values which are orders of magnitude different from the values obtained from the theory of the Gaussian distribution. Relying on the normal theory leads to overly optimistic figures ... and possibly to very bad surprises. Fitting heavy tail distributions and using copulas give more conservative (and presumably more realistic) quantifications of the risk carried by most portfolios.

We consider a simple example of risk management using the tools developed in this chapter. The goal is to quantify the risk exposure of a portfolio. For the sake of simplicity, we consider only a single period analysis, and we assume that the portfolio is comprised of only two instruments. The initial value of the portfolio is:

$$V_0 = n_1 S_1 + n_2 S_2$$



where  $n_1$  and  $n_2$  are the numbers of units of the two instruments, which are valued at  $S_1$  and  $S_2$  at the beginning of the period. Let us denote by  $S'_1$  and  $S'_2$  their values at the end of the period. Then the new value of the portfolio is:

$$\begin{aligned} V &= n_1 S'_1 + n_2 S'_2 \\ &= n_1 S_1 e^X + n_2 S_2 e^Y, \end{aligned}$$

if we denote by  $X = \log(S'_1/S_1)$  and  $Y = \log(S'_2/S_2)$  the log returns on the individual instruments. Notice that (just to keep in line with the example discussed in this chapter) we could consider the portfolio to be composed of futures contracts of Brazilian and Colombian coffee, in which case the random log-returns  $X$  and  $Y$  are just the quantities studied so far. In any case, the log-return of the portfolio is

$$R = \log\left(\frac{V}{V_0}\right) = \log\left(\frac{n_1 S_1}{n_1 S_1 + n_2 S_2} e^X + \frac{n_2 S_2}{n_1 S_1 + n_2 S_2} e^Y\right) = \log\left(\lambda_1 e^X + \lambda_2 e^Y\right).$$

if we use the notation  $\lambda_1$  and  $\lambda_2$  for the fractions of the portfolio invested in the two instruments.

#### **Value-at-Risk** $VaR_q$

We now compute the value at risk of this portfolio. According to the discussion of Chapter 1, for a given level  $q$ ,  $VaR_q$  was defined in relation to the capital needed to cover losses occurring with frequency  $q$ , and more precisely,  $VaR_q$  was defined as the 100 $q$ -th percentile of the loss distribution. i.e the solution  $r$  of the equation:

$$q = \mathbb{P}\{-R \geq r\} = \mathbb{P}\{R \leq -r\} = F_R(-r).$$

In order to solve for  $r$  in the above equation, we need to be able to compute the cdf of the log-return  $R$ . The latter can be expressed analytically as:

$$\begin{aligned} \mathbb{P}\{-R \geq r\} &= \mathbb{P}\{\log(\lambda_1 e^X + \lambda_2 e^Y) \leq -r\} \\ &= \int \int_{\{(x,y); \lambda_1 e^x + \lambda_2 e^y \leq e^{-r}\}} f_{(X,Y)}(x,y) dx dy \\ &= \int_{-\infty}^{-r - \log \lambda_1} dx \int_{-\infty}^{\log(e^{-r}/\lambda_2 - \lambda_1/\lambda_2 e^x)} c(F_X(x), F_Y(y)) f_X(x) f_Y(y) dy \\ &= \int_0^{F_X(-r - \log \lambda_1)} du \int_0^{F_Y(\log(e^{-r}/\lambda_2 - \lambda_1/\lambda_2 e^{F_X^{-1}(u)}))} dv c(u,v) \\ &= \int_0^{F_X(-r - \log \lambda_1)} du \left. \frac{\partial}{\partial u} C(u,v) \right|_{v=F_Y(\log(e^{-r}/\lambda_2 - \lambda_1/\lambda_2 e^{F_X^{-1}(u)}))} \end{aligned}$$

where we used several substitutions to change variables in simple and double integrals. Despite all these efforts, and despite the fact that we managed to reduce

the computation to the evaluation of a single integral, this computation cannot be pushed further in this generality. Even when we know more about the copula and the marginal distributions, this integral can very rarely be computed explicitly. We need to use numerical routines to compute this integral. In fact, we need to run these routines quite a lot of times to solve the equation giving the desired value of  $r$ .

The function `VaR.exp.portf` was written to compute the value at risk following this strategy. But the user should be aware that this numerical procedure does not converge all the time, and the results can be disappointing. A typical call to this function looks like:

```
> VaR.exp.portf(0.01, range=c(0.016,0.08), copula=ESTC,
  x.est=B.est, y.est=C.est, lambda1=0.5, lambda2=0.5)
```

We give it only for the sake of completeness.

### ***Expected Shortfall $\mathbb{E}\{\Theta_q\}$***

We now compute the other measure of risk which we introduced in Chapter 1. The analytic technicalities of the computation of the expected shortfall  $\mathbb{E}\{\Theta_q\}$  (recall the definition given in Chapter 1) are even more daunting than for the computation of the value at risk  $VaR_q$ . Just to give a flavor of these technical difficulties, we initialize the process by:

$$\begin{aligned}\mathbb{E}\{\Theta_q\} &= \mathbb{E}\{-R \mid -R > VaR_q\} \\ &= \frac{1}{q} \int_{-\infty}^{-VaR_q} -r F_R(dr) \\ &= \int_{-\infty}^{-r - \log \lambda_1} dx f_X(x) \int_{-\infty}^{\log(e^{-r}/\lambda_2 - \lambda_1/\lambda_2 e^x)} \log(\lambda_1 e^X + \lambda_2 e^Y) \\ &\quad c(F_X(x), F_Y(y)) f_Y(y) dy\end{aligned}$$

where we have used the same notation as before. Unfortunately, it seems much more difficult to reduce this double integral to a single one, and it seems hopeless to try to derive reasonable approximations of the analytic expression of the expected shortfall which can be evaluated by tractable computations. Following this road, we ended up in a cul-de-sac.

Fortunately, random simulation of large samples from the joint distribution of  $(X, Y)$  and Monte Carlo computations can come to the rescue and save the day.

### ***Use of Monte Carlo Computations***

We illustrate the use of Monte Carlo techniques by computing the  $VaR_q$  and the expected shortfall  $\mathbb{E}\{\Theta_q\}$  of a portfolio of Brazilian and Colombian coffee futures contracts. We solve the problem by simulation using historical data on the daily log-returns of the two assets. The strategy consists of generating a large sample from the

joint distribution of  $X$  and  $Y$  as estimated from the historical data, and computing for each couple  $(x_i, y_i)$  in the sample, the value of  $R$ . Our estimate of the value at risk is simply given by the empirical quantile of the set of values of  $R$  thus obtained. We can now restrict ourselves to the values of  $R$  smaller than the negative of the  $VaR_q$  just computed, and the negative of the average of these  $R$ 's gives the expected shortfall. This is implemented in the function `VaR.exp.sim` whose use we illustrate in the commands below.

```
> VaR.exp.sim(n=10000, Q=0.01, copula=ESTC, x.est=B.est,
              y.est=C.est, lambda1=0.7, lambda2=0.3)[1]
Simulation size
10000

> VaR.exp.sim(n=10000, Q=0.01, copula=ESTC, x.est=B.est,
              y.est=C.est, lambda1=0.7, lambda2=0.3)[2]
VaR Q=0.01
0.09290721

> VaR.exp.sim(n=10000, Q=0.01, copula=ESTC, x.est=B.est,
              y.est=C.est, lambda1=0.7, lambda2=0.3)[3]
ES Q=0.01
0.1460994
```

which produce the value at risk and the expected shortfall over a one-period horizon of a unit portfolio with 70% of Brazilian coffee and 30% of Colombian coffee. Notice that the function `VaR.exp.sim` returns a vector with three components. The first one is the number of Monte Carlo samples used in the computation, the second is the estimated value of the VaR while the third one is the estimated value of the expected shortfall.

### *Comparison with the Results of the Normal Model*

For the sake of comparison, we compute the same value at risk under the assumption that the joint distribution of the coffee log-returns is normal. This assumption is implicit in most of the VaR computations done in everyday practice. Our goal here is to show how different the results are.

```
> Port <- c(.7, .3)
> MuP <- sum(Port*Mu)
> MuP
[1] -0.0007028017
> SigP <- sqrt(t(Port) %**% Sigma %**% Port)
> SigP
      [,1]
[1,] 0.03450331
> qnorm(p=.01, mean=MuP, sd=SigP)
[1] -0.08096951
```

For the given portfolio `Port`, we compute the mean `MuP` and the standard deviation `SigP` of the portfolio return, and we compute the one percentile of the corresponding normal distribution. We learn that only one percent of the time will the return be less than 8% while the above computation was telling us that it should be expected to be less than 9.2% with the same frequency. One cent on the dollar is not much, but for a large portfolio, things add up!

---

## 2.5 PRINCIPAL COMPONENT ANALYSIS

Dimension reduction without significant loss of information is one of the main challenges of data analysis. The internet age has seen an exponential growth in the research efforts devoted to the design of efficient codes and compression algorithms. Whether the data are comprised of video streams, images, and/or speech signals, or financial data, finding a basis in which these data can be expressed with a small (or at least a smaller) number of coefficients is of crucial importance. Other important domains of applications are cursed by the high dimensionality of the data. Artificial intelligence applications, especially those involving machine learning and data mining, have the same dimension reduction problems. Pattern recognition problems are closer to the hearts of traditional statisticians. Indeed, regression and statistical classification problems have forced statisticians to face the curse of dimensionality, and to design systematic procedures to encapsulate the important features of high dimensional observations in a small number of variables. Principal component analysis as presented in this chapter, offers an optimal solution to these dimension reduction issues.

### 2.5.1 Identification of the Principal Components of a Data Set

Principal component analysis (PCA, for short) is a data analysis technique designed for numerical data (as opposed to categorical data). The typical situation that we consider is where the data come in the form of a matrix  $[x_{i,j}]_{i=1,\dots,N,j=1,\dots,M}$  of real numbers, the entry  $x_{i,j}$  representing the value of the  $i$ -th observation of the  $j$ -th variable. As usual, we follow the convention of labeling the columns of the data matrix by the variables measured, and the rows by the individuals of the population under investigation. Examples are plentiful in most data analysis applications. We give below detailed analyses of several examples from the financial arena.

As we mentioned above, the  $N$  members of the population can be identified with the  $N$  rows of the data matrix, each one corresponding to an  $M$ -dimensional (row) vector of numbers giving the values of the variables measured on this individual. It is often desirable (especially when  $M$  is large) to reduce the complexity of the descriptions of the individuals and to replace the  $M$  descriptive variables by a smaller number of variables, while at the same time, losing as little information as possible. Let us consider a simple (and presumably naive) illustration of this idea. Imagine

momentarily that all the variables measured are scores of the same nature (for examples they are all lengths expressed in the same unit, or they are all prices expressed in the same currency, ...) so that it would be conceivable to try to characterize each individual by the mean, and a few other numerical statistics computed on all the individual scores. The mean:

$$\overline{x_{i.}} = \frac{x_{i1} + x_{i2} + \cdots + x_{iM}}{M}$$

can be viewed as a linear combination of the individual scores with coefficients  $1/M, 1/M, \dots, 1/M$ . Principal component analysis, is an attempt to describe the individual features in the population in terms of  $M$  linear combinations of the original features, as captured by the variables originally measured on the  $N$  individuals. The coefficients used in the example of the mean are all non-negative and sum up to one. Even though this convention is very attractive because of the probabilistic interpretation which can be given to the coefficients, we shall use another convention for the linear combinations. We shall allow the coefficients to be of any sign (positive as well as negative) and we normalize them so that the sum of their squares is equal to 1. So if we were to use the mean, we would use the normalized linear combination (NLC, for short) given by:

$$\widetilde{x_{i.}} = \frac{1}{\sqrt{M}}x_{i1} + \frac{1}{\sqrt{M}}x_{i2} + \cdots + \frac{1}{\sqrt{M}}x_{iM}.$$

The goal of principal component analysis is to search for the main sources of variation in the  $M$ -dimensional row vectors by identifying  $M$  linearly independent and orthogonal NLC's in such a way that a small number of them capture most of the variation in the data. This is accomplished by identifying the eigenvectors and eigenvalues of the covariance matrix  $C_x$  of the  $M$  column variables. This covariance matrix is defined by:

$$C_x[j, j'] = \frac{1}{N} \sum_{i=1}^N (x_{ij} - \overline{x_{.j}})(x_{ij'} - \overline{x_{.j'}}), \quad j, j' = 1, \dots, M,$$

where we used the standard notation:

$$\overline{x_{.j}} = \frac{x_{1j} + x_{2j} + \cdots + x_{Nj}}{N}$$

for the mean of the  $j$ -th variable over the population of  $N$  individuals. It is easy to check that the matrix  $C_x$  is symmetric (hence diagonalizable) and non-negative definite (which implies that all the eigenvalues are non-negative). One usually orders the eigenvalues in decreasing order, say:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M \geq 0.$$

The corresponding eigenvectors are called the loadings. In practice we choose  $c_1$  to be a normalized eigenvector corresponding to the eigenvalue  $\lambda_1$ ,  $c_2$  to be a normalized eigenvector corresponding to the eigenvalue  $\lambda_2$ , ..., and finally  $c_M$  to be a

normalized eigenvector corresponding to the eigenvalue  $\lambda_M$ , and we make sure that all the vectors  $c_j$  are orthogonal to each other. This is automatically true when the eigenvalues  $\lambda_j$  are simple. See the discussion below for the general case. Recall that we say a vector is normalized if the sum of the squares of its components is equal to 1. If we denote by  $C$  the  $M \times M$  matrix formed by the  $M$  column vectors containing the components of the vectors  $c_1, c_2, \dots, c_M$  in this order, this matrix is orthogonal (since it is a matrix transforming one orthonormal basis into another) and it satisfies:

$$C_x = C^t D C$$

where we use the notation  $^t$  to denote the transpose of a matrix or a vector, and where  $D$  is the  $M \times M$  diagonal matrix with  $\lambda_1, \lambda_2, \dots, \lambda_M$  on the diagonal. Notice the obvious lack of uniqueness of the above decomposition. In particular, if  $c_j$  is a normalized eigenvector associated to the eigenvalue  $\lambda_j$ , so is  $-c_j$ ! This is something one should keep in mind when plotting the eigenvectors  $c_j$ , and when trying to find an intuitive interpretation for the features of the plots. However, this sign flip is easy to handle, and fortunately, it is the only form of non uniqueness when the eigenvalues are simple (i.e. nondegenerate). The ratio:

$$\frac{\lambda_j}{\sum_{j'=1}^M \lambda_{j'}}$$

of a given eigenvalue to the trace of  $C_x$  (i.e. the sum of its eigenvalues) has the interpretation of the proportion of the variation explained by the corresponding eigenvector, i.e. the loading  $c_j$ . In order to justify this statement, we appeal to the Raleigh-Ritz variational principle from linear algebra. Indeed, according to this principle, the eigenvalues and their corresponding eigenvectors can be characterized recursively in the following way. The largest eigenvalue  $\lambda_1$  appears as the maximum:

$$\lambda_1 = \max_{x \in \mathbb{R}^M, \|x\|=1} x^t C_x x$$

while the corresponding eigenvector  $c_1$  appears as the argument of this maximization problem:

$$c_1 = \arg \max_{x \in \mathbb{R}^M, \|x\|=1} x^t C_x x.$$

If we recall the fact that  $x^t C_x x$  represents the quadratic variation (empirical variance) of the NLC's  $x^t x_1, x^t x_2, \dots, x^t x_N$ ,  $\lambda_1$  can be interpreted as the maximal quadratic variation when we consider all the possible (normalized) linear combinations of the  $M$  original measured variables. Similarly, the corresponding (normalized) eigenvector has precisely the interpretation of this NLC which realizes the maximum variation.

As we have already pointed out, the first loading is uniquely determined up to a sign change if the eigenvalue  $\lambda_1$  is simple. If this is not the case, and if we denote by  $m_1$  the multiplicity of the eigenvalue  $\lambda_1$ , we can choose any orthonormal set  $\{c_1, \dots, c_{m_1}\}$  in the eigenspace of  $\lambda_1$  and repeat the eigenvalue  $\lambda_1$ ,  $m_1$  times in the

list of eigenvalues (and on the diagonal of  $D$  as well). This lack of uniqueness is not a mathematical difficulty, it is merely annoying. Fortunately, it seldom happens in practice ! We shall assume that all the eigenvalues are simple (i.e. non-degenerate) for the remainder of our discussion. If they were not, we would have to repeat them according to their multiplicities.

Next, still according to the Raleigh-Ritz variation principle, the second eigenvalue  $\lambda_2$  appears as the maximum:

$$\lambda_2 = \max_{x \in \mathbb{R}^M, \|x\|=1, x \perp c_1} x^t C_x x$$

while the corresponding eigenvector  $c_2$  appears as the argument of this maximization problem:

$$c_2 = \arg \max_{x \in \mathbb{R}^M, \|x\|=1, x \perp c_1} x^t C_x x.$$

The interpretation of this statement is the following: if we avoid any overlap with the loading already identified (i.e. if we restrict ourselves to NLC's  $x$  which are orthogonal to  $c_1$ ), then the maximum quadratic variation will be  $\lambda_2$  and any NLC realizing this maximum variation can be used for  $c_2$ . We can go on and identify in this way all the eigenvalues  $\lambda_j$  (having to possibly repeat them according to their multiplicities) and the loadings  $c_j$ 's.

Armed with a new basis of  $\mathbb{R}^M$ , the next step is to rewrite the data observations (i.e. the  $N$  rows of the data matrix) in this new basis. This is done by multiplying the data matrix by the *change of basis* matrix (i.e. the matrix whose columns are the eigenvectors identified earlier). The result is a new  $N \times M$  matrix whose columns are called *principal components*. Their relative importance is given by the proportion of the variance explained by the loadings, and for that reason, one typically considers only the first few principal components, the remaining ones being ignored and/or treated as noise.

### 2.5.2 PCA with S-Plus

The principal component analysis of a data set is performed in S-Plus with the function `princomp`, which returns an object of class `princomp` that can be printed and plotted with generic methods. Illustrations of the calls to this function and of the interpretation of the results are given in the next subsections in which we discuss several financial applications of the PCA.

### 2.5.3 Effective Dimension of the Space of Yield Curves

Our first application concerns the markets of fixed income securities which we will introduce in Section 3.8. The term structure of interest rates is conveniently captured by the daily changes in the yield curve. The dimension of the space of all possible yield curves is presumably very large, potentially infinite if we work in the idealized world of continuous-time finance. However, it is quite sensible to try to approximate

these curves by functions from a class chosen in a parsimonious way. Without any a priori choice of the type of functions to be used to approximate the yield curve, PCA can be used to extract, one by one, the components responsible for the variations in the data.

### *PCA of the Yield Curve*

For the purposes of illustration, we use data on the US yield curve as provided by the Bank of International Settlements (BIS, for short). These data are the result of a nonparametric processing (smoothing spline regression, to be specific) of the raw data. The details will be given in Section 4.4 of Chapter 4, but for the time being, we shall ignore the possible effects of this pre-processing of the raw data. The data are imported into an S-object named `us.bis.yield` which gives, for each of the 1352 successive trading days following January 3rd 1995, the yields on the US Treasury bonds for times to maturity

$$x = 0, 1, 2, 3, 4, 5, 5.5, 6.5, 7.5, 8.5, 9.5 \text{ months.}$$

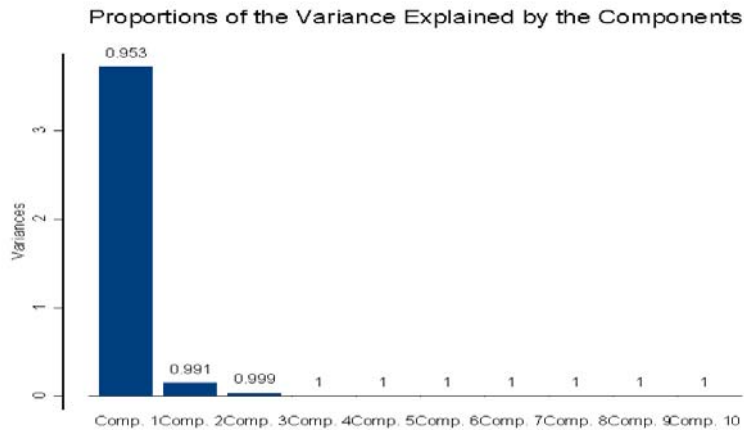
We run a PCA on these data with the following S-Plus commands:

```
> dim(us.bis.yield)
[1] 1352 11
> us.bis.yield.pca <- princomp(us.bis.yield)
> plot(us.bis.yield.pca)
[1] 0.700000 1.900000 3.100000 4.300000 5.500000
[6] 6.700000 7.900000 9.099999 10.299999 11.499999
> title("Proportions of the Variance Explained
        by the Components")
```

The results are reproduced in Figure 2.20 which gives the proportions of the variation explained by the various components. The first three eigenvectors of the covariance matrix (the so-called loadings) explain 99.9% of the total variation in the data. This suggests that the effective dimension of the space of yield curves could be three. In other words, any of the yield curves from this period can be approximated by a linear combination of the first three loadings, the relative error being very small. In order to better understand the far reaching implications of this statement we plot the first four loadings.

```
> X <- c(0,1,2,3,4,5,5.5,6.5,7.5,8.5,9.5)
> par(mfrow=c(2,2))
> plot(X,us.bis.yield.pca$loadings[,1],ylim=c(-.7,.7))
> lines(X,us.bis.yield.pca$loadings[,1])
> plot(X,us.bis.yield.pca$loadings[,2],ylim=c(-.7,.7))
> lines(X,us.bis.yield.pca$loadings[,2])
> plot(X,us.bis.yield.pca$loadings[,3],ylim=c(-.7,.7))
> lines(X,us.bis.yield.pca$loadings[,3])
> plot(X,us.bis.yield.pca$loadings[,4],ylim=c(-.7,.7))
```



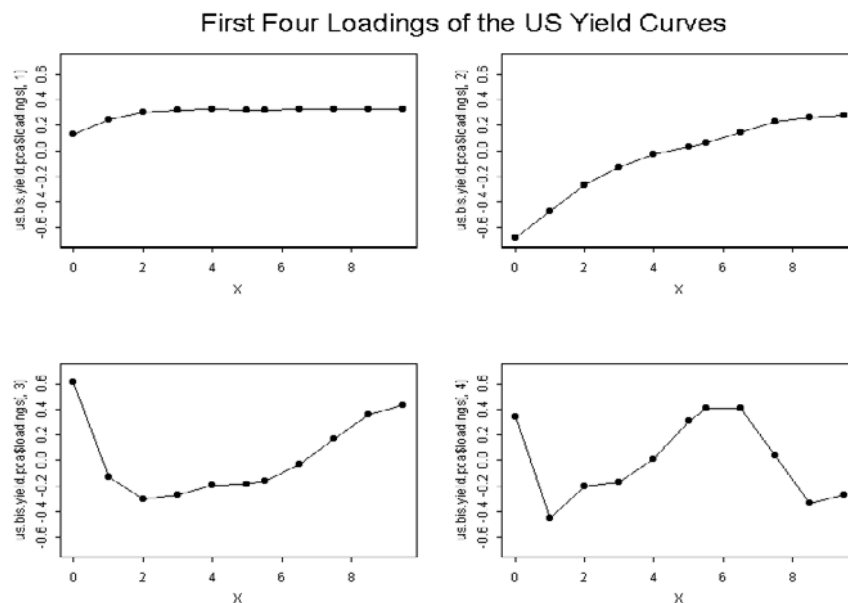


**Fig. 2.20.** Proportions of the variance explained by the components of the PCA of the daily changes in the US yield curve.

```
> lines(X,us.bis.yield.pca$loadings[,4])
> par(mfrow=c(1,1))
> title("First Four Loadings of the US Yield Curves")
```

The results are reproduced in Figure 2.21. The first loading is essentially flat, so a component on this loading will essentially represent the average yield over the maturities, and the effect of this most-important component on the actual yield curve is a parallel shift. Because of the monotone and increasing nature of the second loading, the second component measures the upward trend (if the component is positive, and the downward trend otherwise) in the yield. This second factor is interpreted as the tilt of the yield curve. The shape of the third loading suggests that the third component captures the curvature of the yield curve. Finally, the shape of the fourth loading does not seem to have an obvious interpretation. It is mostly noise (remember that most of the variations in the yield curve are explained by the first three components). These features are very typical, and they should be expected in most PCA's of the term structure of interest rates.

The fact that the first three components capture so much of the yield curve may seem strange when compared to the fact that some estimation methods, which we discuss later in the book, use parametric families with more than three parameters! There is no contradiction there. Indeed, for the sake of illustration, we limited the analysis of this section to the first part of the yield curve. Restricting ourselves to short maturities makes it easier to capture all the features of the yield curve in a small number of functions with a clear interpretation.



**Fig. 2.21.** From left to right and top to bottom, sequential plots of the first four US yield loadings.

### 2.5.4 Swap Rate Curves

Swap contracts have been traded publicly since 1981. As of today, they are the most popular fixed income derivatives. Because of this popularity, the swap markets are extremely liquid, and as a consequence, they can be used to hedge interest-rate risk of fixed income portfolios at a low cost. The estimation of the term-structure of swap rates is important in this respect and the PCA which we present below is the first step toward a better understanding of this term structure.

#### *Swap Contracts and Swap Rates*

As implied by its name, a swap contract obligates two parties to exchange (or swap) some specified cash flows at agreed upon times. The most common swap contracts are interest rate swaps. In such a contract, one party, say counter-party A, agrees to make interest payments determined by an instrument  $P_A$  (say, a 10 year US Treasury bond rate), while the other party, say counter-party B, agrees to make interest payments determined by another instrument  $P_B$  (say, the London Interbank Offer Rate – LIBOR for short). Even though there are many variants of swap contracts, in a typical contract, the principal on which counter-party A makes interest payments is equal to the principal on which counterparty B makes interest payments. Also, the

payment schedules are identical and periodic, the payment frequency being quarterly, semi-annually, . . . .

It is not difficult to infer from the above discussion that a swap contract is equivalent to a portfolio of forward contracts, but we shall not use this feature here. In this section, we shall restrict ourselves to the so-called plain vanilla contracts involving a fixed interest rate and the 3 or 6 months LIBOR rate.

We will not attempt to derive here a formula for the price of a swap contract, neither will we try to define rigorously the notion of swap rate. These derivations are beyond the scope of this book. See the Notes & Complements at the end of the chapter for references to appropriate sources. We shall use only the intuitive idea of the swap rate being a rate at which both parties will agree to enter into the swap contract.

### *PCA of the Swap Rates*

Our second application of principal component analysis concerns the term structure of swap rates as given by the swap rate curves. As before, we denote by  $M$  the dimension of the vectors. We use data downloaded from `Data Stream`. It is quite likely that the raw data have been processed, but we are not quite sure what kind of manipulation is performed by `Data Stream`, so for the purposes of this illustration, we shall ignore the possible effects of the pre-processing of the data. In this example, the day  $t$  labels the rows of the data matrix. The latter has  $M = 15$  columns, containing the swap rates with maturities  $T$  conveniently labeled by the times to maturity  $x = T - t$ , which have the values 1, 2, . . . , 10, 12, 15, 20, 25, 30 years in the present situation. We collected these data for each day  $t$  of the period from May 1998 to March 2000, and we rearranged the numerical values in a matrix  $R = [r_{i,j}]_{i=1,\dots,N, j=1,\dots,M}$ . Here, the index  $j$  stands for the time to maturity, while the index  $i$  codes the day the curve is observed.

The data is contained in the S object `swap`. The PCA is performed in `S-Plus` with the command:

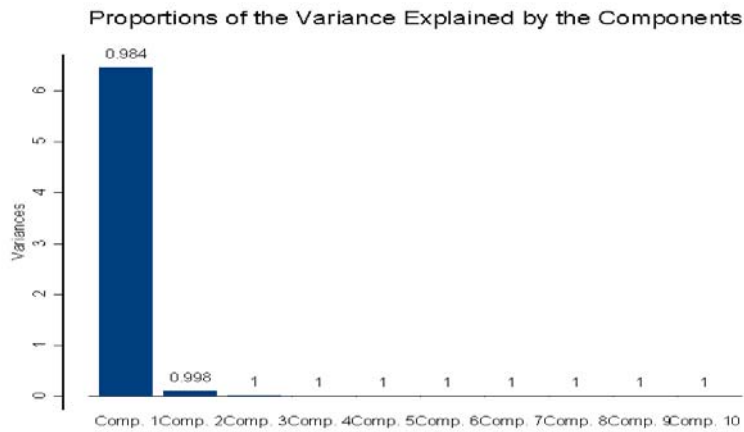
```
> dim(swap)
[1] 496 15
> swap.pca <- princomp(swap)
> plot(swap.pca)
[1] 0.700000 1.900000 3.100000 4.300000 5.500000
[6] 6.700000 7.900000 9.099999 10.299999 11.499999
> title("Proportions of the Variance Explained by
        the Components")
> YEARS <- c(1,2,3,4,5,6,7,8,9,10,12,15,20,25,30)
> par(mfrow=c(2,2))
> plot(YEARS, swap.pca$loadings[,1], ylim=c(-.6, .6))
> lines(YEARS, swap.pca$loadings[,1])
> plot(YEARS, swap.pca$loadings[,2], ylim=c(-.6, .6))
> lines(YEARS, swap.pca$loadings[,2])
> plot(YEARS, swap.pca$loadings[,3], ylim=c(-.6, .6))
```

```

> lines(YEARS, swap.pca$loadings[,3])
> plot(YEARS, swap.pca$loadings[,4], ylim=c(-.6, .6))
> lines(YEARS, swap.pca$loadings[,4])
> par(mfrow=c(1,1))
> title("First Four Loadings of the Swap Rates")

```

Figure 2.22 gives the proportions of the variation explained by the various components, while Figure 2.23 gives the plots of the first four eigenvectors.



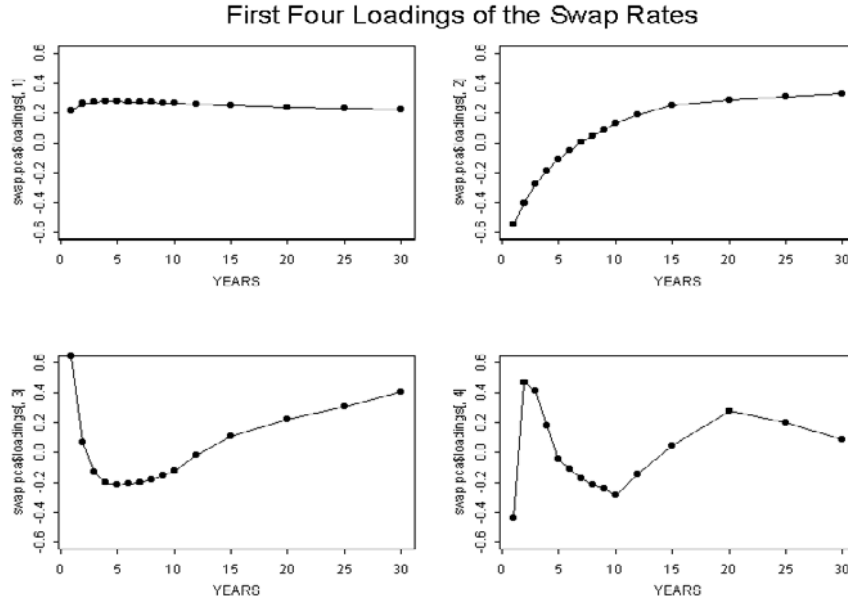
**Fig. 2.22.** Proportions of the variance explained by the components of the PCA of the daily changes in the swap rates for the period from May 1998 to March 2000.

Looking at Figure 2.23 one sees that the remarks made above, for the interpretation of the results in terms of a parallel shift, a tilt and a curvature component, do apply to the present situation as well.

Since such an overwhelming proportion of the variation is explained by one single component, it is often recommended to remove the effect of this component from the data, (here, that would amount to subtracting the overall mean rate level) and to perform the PCA on the transformed data (here, the fluctuations around the mean rate level).

## APPENDIX 1: CALCULUS WITH RANDOM VECTORS AND MATRICES

The nature and technical constructs of this chapter justify our spending some time discussing the properties of random vectors (as opposed to random variables) and reviewing the fundamental results of the calculus of probability with random vectors. Their definition is very natural: a random vector is a vector whose entries are random



**Fig. 2.23.** From left to right and top to bottom, sequential plots of the eigenvectors (loadings) corresponding to the 4 largest eigenvalues. Notice that we changed the scale of the horizontal axis to reflect the actual times to maturity.

variables. With this definition in hand, it is easy to define the notion of expectation. The expectation of a random vector is the (deterministic) vector whose entries are the expectations of the entries of the original random vector. In other words,

$$\text{if } \mathbf{X} = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix}, \quad \text{then } \mathbb{E}\{\mathbf{X}\} = \begin{bmatrix} \mathbb{E}\{X_1\} \\ \vdots \\ \mathbb{E}\{X_n\} \end{bmatrix}.$$

Notice that, if  $\mathbf{B}$  is an  $n$ -dimensional (deterministic) vector then:

$$\mathbb{E}\{\mathbf{X} + \mathbf{B}\} = \mathbb{E}\{\mathbf{X}\} + \mathbf{B}. \quad (2.16)$$

Indeed:

$$\mathbb{E}\{\mathbf{X} + \mathbf{B}\} = \begin{bmatrix} \mathbb{E}\{X_1 + b_1\} \\ \vdots \\ \mathbb{E}\{X_n + b_n\} \end{bmatrix} = \begin{bmatrix} \mathbb{E}\{X_1\} + b_1 \\ \vdots \\ \mathbb{E}\{X_n\} + b_n \end{bmatrix} = \mathbb{E}\{\mathbf{X}\} + \mathbf{B}$$

where we used the notation  $b_i$  for the components of the vector  $\mathbf{B}$ . The notion of variance (or more generally of second moment) appears somehow less natural at

first. We define the variance/covariance matrix of a random vector to be the (deterministic) matrix whose entries are the variances and covariances of the entries of the original random vector. More precisely, if  $\mathbf{X}$  is a random vector as above, then its variance/covariance matrix is the matrix  $\Sigma_{\mathbf{X}}$  defined by:

$$\Sigma_{\mathbf{X}} = \begin{bmatrix} \sigma_1^2 & \text{cov}(X_1, X_2) & \cdots & \text{cov}(X_1, X_n) \\ \text{cov}(X_2, X_1) & \sigma_2^2 & \cdots & \text{cov}(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(X_n, X_1) & \text{cov}(X_n, X_2) & \cdots & \sigma_n^2 \end{bmatrix}, \quad (2.17)$$

In other words, on the  $i$ -th row and the  $j$ -th column of  $\Sigma_{\mathbf{X}}$  we find the covariance of  $X_i$  and  $X_j$ . For the purposes of this appendix, we limit ourselves to random variables of order 2 (i.e. for which the first two moments exist) so that all the variances and covariances make perfectly good mathematical sense. Note that this is not the case for many generalized Pareto distributions, and especially for our good old friend the Cauchy distribution.

The best way to look at the variance/covariance matrix of a random vector is the following. Using the notation  $\mathbf{Z}^t$  for the transpose of the vector or matrix  $\mathbf{Z}$ , we notice that:

$$\begin{aligned} [\mathbf{X} - \mathbb{E}\{\mathbf{X}\}][\mathbf{X} - \mathbb{E}\{\mathbf{X}\}]^t &= \begin{bmatrix} X_1 - \mu_1 \\ \vdots \\ X_n - \mu_n \end{bmatrix} [X_1 - \mu_1, \dots, X_n - \mu_n] \\ &= \begin{bmatrix} (X_1 - \mu_1)^2 & (X_1 - \mu_1)(X_2 - \mu_2) & \cdots & (X_1 - \mu_1)(X_n - \mu_n) \\ (X_2 - \mu_2)(X_1 - \mu_1) & (X_2 - \mu_2)^2 & \cdots & (X_2 - \mu_2)(X_n - \mu_n) \\ \vdots & \vdots & \ddots & \vdots \\ (X_n - \mu_n)(X_1 - \mu_1) & (X_n - \mu_n)(X_2 - \mu_2) & \cdots & (X_n - \mu_n)^2 \end{bmatrix} \end{aligned}$$

if we use the notation  $\mu_j = \mathbb{E}\{X_j\}$  to shorten the typesetting of the formula. The variance/covariance matrix  $\Sigma_{\mathbf{X}}$  is nothing more than the expectation of this random matrix, since the expectation of a random matrix is defined as the (deterministic) matrix whose entries are the expectations of the entries of the original random matrix. Consequently we have proven that, for any random vector  $\mathbf{X}$  of order 2, the variance/covariance matrix  $\Sigma_{\mathbf{X}}$  is given by the formula:

$$\Sigma_{\mathbf{X}} = \mathbb{E}\{[\mathbf{X} - \mathbb{E}\{\mathbf{X}\}][\mathbf{X} - \mathbb{E}\{\mathbf{X}\}]^t\}. \quad (2.18)$$

Notice that, if the components of a random vector are independent, then the variance/covariance matrix of this random vector is diagonal since all the entries off the diagonal must vanish due to the independence assumption.

### Some Useful Formulae

If  $\mathbf{X}$  is an  $n$ -dimensional random vector, if  $\mathbf{A}$  is an  $m \times n$  deterministic matrix and  $\mathbf{B}$  is an  $m$ -dimensional deterministic vector, then:

$$\mathbb{E}\{\mathbf{A}\mathbf{X} + \mathbf{B}\} = \mathbf{A}\mathbb{E}\{\mathbf{X}\} + \mathbf{B} \quad (2.19)$$

as can be checked by computing the various components of the  $m$ -dimensional vectors on both sides of the equality sign. Notice that formula (2.16) is merely a particular case of (2.19) when  $m = n$  and  $\mathbf{A}$  is the identity matrix. In fact, formula (2.19) remains true when  $\mathbf{X}$  is an  $n \times p$  random matrix and  $\mathbf{B}$  is an  $m \times p$  deterministic matrix. By transposition one gets that

$$\mathbb{E}\{\mathbf{X}\mathbf{A} + \mathbf{B}\} = \mathbb{E}\{\mathbf{X}\}\mathbf{A} + \mathbf{B} \quad (2.20)$$

holds whenever  $\mathbf{X}$  is an  $n \times p$  random matrix and  $\mathbf{A}$  and  $\mathbf{B}$  are deterministic matrices with dimensions  $p \times m$  and  $n \times m$  respectively. Using (2.19) and (2.20) we get:

$$\Sigma_{\mathbf{A}\mathbf{X}+\mathbf{B}} = \mathbf{A}\Sigma_{\mathbf{X}}\mathbf{A}^t \quad (2.21)$$

A proof of this formula goes as follows:

$$\begin{aligned} \Sigma_{\mathbf{A}\mathbf{X}+\mathbf{B}} &= \mathbb{E}\{[\mathbf{A}\mathbf{X} + \mathbf{B} - \mathbb{E}\{\mathbf{A}\mathbf{X} + \mathbf{B}\}][\mathbf{A}\mathbf{X} + \mathbf{B} - \mathbb{E}\{\mathbf{A}\mathbf{X} + \mathbf{B}\}]^t\} \\ &= \mathbb{E}\{[\mathbf{A}(\mathbf{X} - \mathbb{E}\{\mathbf{X}\})][\mathbf{A}(\mathbf{X} - \mathbb{E}\{\mathbf{X}\})]^t\} \\ &= \mathbb{E}\{\mathbf{A}[\mathbf{X} - \mathbb{E}\{\mathbf{X}\}][\mathbf{X} - \mathbb{E}\{\mathbf{X}\}]^t\mathbf{A}^t\} \\ &= \mathbf{A}\mathbb{E}\{[\mathbf{X} - \mathbb{E}\{\mathbf{X}\}][\mathbf{X} - \mathbb{E}\{\mathbf{X}\}]^t\}\mathbf{A}^t \\ &= \mathbf{A}\Sigma_{\mathbf{X}}\mathbf{A}^t \end{aligned}$$

Similar formulae can be proven for the variance/covariance matrix of expressions of the form  $\mathbf{A}\mathbf{X}\mathbf{B} + \mathbf{C}$  when  $\mathbf{X}$  is a random vector or a random matrix and when  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$  are deterministic matrices or vectors whose dimensions make the product meaningful.

**Warning:** Remember to be very cautious with the order in a product of matrices. Just because one can change the order in the product of numbers, does not mean that it is a good idea to do the same thing with a product of matrices, as the results are in general (very) different !!!

---

## APPENDIX 2: FAMILIES OF COPULAS

There are many parametric families of copulas, and new ones are created every month. For the sake of completeness, we review those implemented in the `S-Plus` library `EVANESCE`. They can be organized in two main classes.

◇ **Extreme value copulas** are copulas of the form

$$C(x, y) = \exp \left[ \log(xy) A \left( \frac{\log(x)}{\log(xy)} \right) \right],$$

where  $A : [0, 1] \rightarrow [0.5, 1]$ , is a convex function, and  $\max(t, 1 - t) \leq A(t) \leq 1$  for all  $t \in [0, 1]$ .

◊ **Archimedean copulas** are copulas of the form

$$C(x, y) = \phi^{-1} \left[ (\phi(x) + \phi(y)) A \left( \frac{\phi(x)}{\phi(x) + \phi(y)} \right) \right],$$

where  $\phi(t)$  is a valid Archimedean generator (convex and decreasing on  $(0, 1)$ ), and  $A$  is as above.

We now list the most commonly used parametric families of copulas:

• **Bivariate Normal**, "normal"

> normal.copula(delta)

$$C(x, y) = \Phi_{\delta}(\Phi^{-1}(x), \Phi^{-1}(y)),$$

$0 \leq \delta \leq 1$ , where  $\Phi^{-1}$  is the quantile function of the standard normal distribution, and  $\Phi_{\delta}$  is the cdf of the standard bivariate normal with correlation  $\delta$

• **Frank Copula**, "frank"

> frank.copula(delta)

$$C(x, y) = -\delta^{-1} \log \left( \frac{\eta - (1 - e^{-\delta x})(1 - e^{-\delta y})}{\eta} \right)$$

$0 \leq \delta < \infty$ , and  $\eta = 1 - e^{-\delta}$ .

• **Kimeldorf and Sampson copula**, "kimeldorf.sampson"

> kimeldorf.sampson.copula(delta)

$$C(x, y) = (x^{-\delta} + y^{-\delta} - 1)^{-1/\delta},$$

$0 \leq \delta < \infty$ .

• **Gumbel copula**, "gumbel"

> gumbel.copula(delta)

$$C(x, y) = \exp \left( - [(-\log x)^{\delta} + (-\log y)^{\delta}]^{1/\delta} \right),$$

$1 \leq \delta < \infty$ . This is an extreme value copula with the dependence function

$$A(t) = (t^{\delta} + (1 - t)^{\delta})^{1/\delta}.$$

• **Galambos**, "galambos"

> galambos.copula(delta)

$$C(x, y) = xy \exp \left( [(-\log x)^{-\delta} + (-\log y)^{-\delta}]^{-1/\delta} \right),$$

$0 \leq \delta < \infty$ . This is an extreme value copula with the dependence function

$$A(t) = 1 - (t^{-\delta} + (1 - t)^{-\delta})^{-1/\delta}.$$



• **Hüsler and Reiss**, "husler.reiss"

> husler.reiss.copula(delta)

$$C(x, y) = \exp \left( -\tilde{x} \Phi \left[ \frac{1}{\delta} + \frac{1}{2} \delta \log \left( \frac{\tilde{x}}{\tilde{y}} \right) \right] - \tilde{y} \Phi \left[ \frac{1}{\delta} + \frac{1}{2} \delta \log \left( \frac{\tilde{y}}{\tilde{x}} \right) \right] \right)$$

$0 \leq \delta < \infty$ ,  $\tilde{x} = -\log x$ ,  $\tilde{y} = -\log y$ , and  $\Phi$  is the cdf of the standard normal distribution. This is an extreme value copula with the dependence function

$$A(t) = t \Phi \left[ \delta^{-1} + \frac{1}{2} \delta \log \left( \frac{t}{1-t} \right) \right] + (1-t) \Phi \left[ \delta^{-1} - \frac{1}{2} \delta \log \left( \frac{t}{1-t} \right) \right]$$

• **Twan**, "twan"

> twan.copula(alpha, beta, r)

This is an extreme value copula with the dependence function

$$A(t) = 1 - \beta + (\beta - \alpha) + \{\alpha^r t^r + \beta^r (1-t)^r\}^{1/r},$$

$0 \leq \alpha, \beta \leq 1$ ,  $1 \leq r < \infty$ .

• **BB1**, "bb1"

> bb1.copula(theta, delta)

$$C(x, y) = \left( 1 + [(x^{-\theta} - 1)^\delta + (y^{-\theta} - 1)^\delta]^{1/\delta} \right)^{-1/\theta}$$

$\theta > 0, \delta \geq 1$ . This is an Archimedean copula with the Archimedean generator  $\phi(t) = (t^{-\theta} - 1)^\delta$ .

• **BB2**, "bb2"

> bb2.copula(theta, delta)

$$C(x, y) = \left[ 1 + \delta^{-1} \log \left( e^{\delta(x^{-\theta})} + e^{\delta(y^{-\theta})} - 1 \right) \right]^{1/\theta},$$

$\theta > 0, \delta > 0$ . This is an Archimedean copula with the Archimedean generator  $\phi(t) = e^{\delta(t^{-\theta} - 1)} - 1$ .

• **BB3**, "bb3"

> bb3.copula(theta, delta)

$$C(x, y) = \exp \left( - \left[ \delta^{-1} \log \left( e^{\delta \tilde{x}^\theta} + e^{\delta \tilde{y}^\theta} - 1 \right) \right]^{1/\theta} \right),$$

$\theta \geq 1, \delta > 0$ ,  $\tilde{x} = -\log x$ ,  $\tilde{y} = -\log y$ . This is an Archimedean copula with the Archimedean generator  $\phi(t) = \exp \{ \delta (-\log t)^\theta \} - 1$ .

• **BB4**, "bb4"

> bb4.copula(theta, delta)

$$C(x, y) = \left( x^{-\theta} + y^{-\theta} - 1 - \left[ (x^{-\theta} - 1)^{-\delta} + (y^{-\theta} - 1)^{-\delta} \right]^{-\frac{1}{\delta}} \right)^{-\frac{1}{\theta}}$$

$\theta \geq 0, \delta > 0$ . This is an Archimedean copula with the Archimedean generator  $\phi(t) = t^{-\theta} - 1$  and the dependence function  $A(t) = 1 - (t^{-\delta} + (1-t)^{-\delta})^{-1/\delta}$  (same as for B7 family).

• **BB5**, "bb5"

```
> bb5.copula(theta, delta)
```

$$C(x, y) = \exp \left( - \left[ \tilde{x}^\theta + \tilde{y}^\theta - (\tilde{x}^{-\theta\delta} + \tilde{y}^{-\theta\delta})^{-1/\delta} \right]^{1/\theta} \right),$$

$\delta > 0, \theta \geq 1, \tilde{x} = -\log x, \tilde{y} = -\log y$ . This is an extreme value copula with the dependence function

$$A(t) = \left[ t^\theta + (1-t)^\theta - (t^{-\delta\theta} + (1-t)^{-\delta\theta})^{-1/\delta} \right]^{1/\theta}.$$

• **BB6**, "bb6"

```
> bb6.copula(theta, delta)
```

This is an Archimedean copula with the generator  $\phi(t) = [-\log(1 - (1-t)^\theta)]^\delta$   $\theta \geq 1, \delta \geq 1$ .

• **BB7**, "bb7"

```
> bb7.copula(theta, delta)
```

This is an Archimedean copula with the generator

$$\phi(t) = (1 - (1-t)^\theta)^{-\delta} - 1, \quad \theta \geq 1, \delta > 0.$$

• **B1Mix**, "normal.mix"

```
> normal.mix.copula(p, delta1, delta2)
```

$$C(x, y) = p C_{\delta_1}^{(B1)}(x, y) + (1-p) C_{\delta_2}^{(B1)}(x, y),$$

$0 \leq p, \delta_1, \delta_2 \leq 1$ , where  $C_{\delta}^{(B1)}(x, y)$  is a bivariate normal copula (family "normal").

## PROBLEMS

Ⓔ **Problem 2.1** This problem is based on the data contained in the script file `utilities.asc`.

Opening it in S-Plus, and running it as a script creates a matrix with two columns. Each row corresponds to a given day. The first column gives the log of the weekly return on an index based on Southern Electric stock value and capitalization, (we'll call that variable  $X$ ), and the second column gives, on the same day, the same quantity for Duke Energy (we'll call that variable  $Y$ ), another large utility company.

1. Compute the means and the standard deviations of  $X$  and  $Y$ , and compute their correlation coefficients.

2. We first assume that  $X$  and  $Y$  are samples from a jointly Gaussian distribution with parameters computed in question 1. Compute the  $q$ -percentile with  $q = 2\%$  of the variables  $X + Y$  and  $X - Y$ .

3. Fit a generalized Pareto distribution (GPD) to  $X$  and  $Y$  separately, and fit a copula of the Gumbel family to the empirical copula of the data.

4. Generate a sample of size  $N$  (where  $N$  is the number of rows of the data matrix) from the joint distribution estimated in question 3.

4.1. Use this sample to compute the same statistics as in question 1 (i.e. means and standard deviations of the columns, as well as their correlation coefficients), and compare the results to the numerical values obtained in question 1.

4.2. Compute, still for this simulated sample, the two percentiles considered in question 2, compare the results, and comment.

- (E) Problem 2.2** This problem is based on the data contained in the script file `SPfutures.asc` which creates a matrix `SPFUT` with two columns, each row corresponding to a given day. The first column gives, for each day, the log return of a futures contract which matures three weeks later, (we'll call that variable  $X$ ), and the second column gives, on the same day, the log return of a futures contract which matures one week later (we'll call that variable  $Y$ ). Question 2 is not required for the rest of the problem. In other words, you can answer questions 3 and 4 even if you did not get question 2.

1. Compute the means and the standard deviations of  $X$  and  $Y$ , and compute their correlation coefficients.

2. We first assume that  $X$  and  $Y$  are samples from a jointly Gaussian distribution with parameters computed in part 1. For each value  $\alpha = 25\%$ ,  $\alpha = 50\%$  and  $\alpha = 75\%$  of the parameter  $\alpha$ , compute the  $q$ -percentile with  $q = 2\%$  of the variable  $\alpha X + (1 - \alpha)Y$ .

3. Fit a generalized Pareto distribution (GPD) to  $X$  and  $Y$  separately, and fit a copula of the Gumbel family to the empirical copula of the data.

4. Generate a sample of size  $N$  (where  $N$  is the number of rows of the data matrix) from the joint distribution estimated in question 3.

4.1. Use this sample to compute the same statistics as in question 1 (i.e. means and standard deviations of the columns, as well as their correlation coefficients) and compare to the numerical values obtained in question 1.

4.2. Compute, still for this simulated sample, the three percentiles considered in question 2, and compare the results.

- (S) Problem 2.3** 1. Construct a vector of 100 increasing and regularly spaced numbers starting from .1 and ending at 20. Call it `SIG2`. Construct a vector of 21 increasing and regularly spaced numbers starting from  $-1.0$  and ending at  $1.0$ . Call it `RHO`.

2. For each entry  $\sigma^2$  of `SIG2` and for each entry  $\rho$  of `RHO`:

- Generate a sample of size  $N = 500$  from the distribution of a bivariate normal vector  $Z = (X, Y)$ , where  $X \sim N(0, 1)$ , and  $Y \sim N(0, \sigma^2)$ , and the correlation coefficient of  $X$  and  $Y$  is  $\rho$  (the `S` object you create to hold the values of the sample of  $Z$ 's should be a  $500 \times 2$  matrix);

- Create a  $500 \times 2$  matrix, call it `EXPZ`, with the exponentials of the entries of  $Z$  (the distributions of these columns are lognormal as defined in Problem 2.7);

- Compute the correlation coefficient, call it  $\tilde{\rho}$ , of the two columns of `EXPZ`

3. Produce a scatterplot of all the points  $(\sigma^2, \tilde{\rho})$  so obtained. Comment.

- (T) Problem 2.4** This elementary exercise is intended to give an example showing that lack of correlation does not necessarily mean independence!

Let us assume that  $X \sim N(0, 1)$  and let us define the random variable  $Y$  by:

$$Y = \frac{1}{\sqrt{1 - 2/\pi}}(|X| - \sqrt{2/\pi})$$

1. Compute  $\mathbb{E}\{X\}$
2. Show that  $Y$  has mean zero, variance 1, and that it is uncorrelated with  $X$ .

Ⓣ **Problem 2.5** The purpose of this problem is to show that lack of correlation does not imply independence, even when the two random variables are Gaussian !!!

We assume that  $X$ ,  $\epsilon_1$  and  $\epsilon_2$  are independent random variables, that  $X \sim N(0, 1)$ , and that  $\mathbb{P}\{\epsilon_i = -1\} = \mathbb{P}\{\epsilon_i = +1\} = 1/2$  for  $i = 1, 2$ . We define the random variable  $X_1$  and  $X_2$  by:

$$X_1 = \epsilon_1 X, \quad \text{and} \quad X_2 = \epsilon_2 X.$$

1. Prove that  $X_1 \sim N(0, 1)$ ,  $X_2 \sim N(0, 1)$  and that  $\rho\{X_1, X_2\} = 0$ .
2. Show that  $X_1$  and  $X_2$  are not independent.

Ⓣ **Problem 2.6** The goal of this problem is to prove rigorously a couple of useful formulae for normal random variables.

1. Show that, if  $Z \sim N(0, 1)$ , if  $\sigma > 0$ , and if  $f$  is ANY function, then we have:

$$\mathbb{E}\{f(Z)e^{\sigma Z}\} = e^{\sigma^2/2} \mathbb{E}\{f(Z + \sigma)\},$$

and use this formula to recover the well known fact

$$\mathbb{E}\{e^X\} = e^{\mu + \sigma^2/2}$$

whenever  $X \sim N(\mu, \sigma^2)$ .

2. We now assume that  $X$  and  $Y$  are jointly-normal mean-zero random variables and that  $h$  is ANY function. Prove that:

$$\mathbb{E}\{e^X h(Y)\} = \mathbb{E}\{e^X\} \mathbb{E}\{h(Y + \text{cov}\{X, Y\})\}.$$

Ⓣ **Problem 2.7** The goal of this problem is to prove rigorously the theoretical result illustrated by the simulations of Problem 2.3.

1. Compute the density of a random variable  $X$  whose logarithm  $\log X$  is  $N(\mu, \sigma^2)$ . Such a random variable is usually called a lognormal random variable with mean  $\mu$  and variance  $\sigma^2$ .

Throughout the rest of the problem we assume that  $X$  is a lognormal random variable with parameters 0 and 1 (i.e.  $X$  is the exponential of an  $N(0, 1)$  random variable) and that  $Y$  is a lognormal random variable with parameters 0 and  $\sigma^2$  (i.e.  $Y$  is the exponential of an  $N(0, \sigma^2)$  random variable). Moreover, we use the notation  $\rho_{\min}$  and  $\rho_{\max}$  introduced in the last paragraph of Subsection 2.1.2.

2. Show that  $\rho_{\min} = (e^{-\sigma} - 1)/\sqrt{(e - 1)(e^{\sigma^2} - 1)}$ .
3. Show that  $\rho_{\max} = (e^{\sigma} - 1)/\sqrt{(e - 1)(e^{\sigma^2} - 1)}$ .
4. Check that  $\lim_{\sigma \rightarrow \infty} \rho_{\min} = \lim_{\sigma \rightarrow \infty} \rho_{\max} = 0$ .

Ⓢ Ⓣ **Problem 2.8** The first question concerns the computation in S-Plus of the square root of a symmetric nonnegative-definite square matrix.

1. Write an S-function, call it `msqrt`, with argument  $A$  which:

- checks that  $A$  is a square matrix and exits if not;
- checks that  $A$  is symmetric and exits if not;
- diagonalizes the matrix by computing the eigenvalues and the matrix of eigenvectors (hint: check out the help file of the function `eigen` if you are not sure how to proceed);
- checks that all the eigenvalues are nonnegative and exits, if not;

- returns a symmetric matrix of the same size as  $A$ , with the same eigenvectors, the eigenvalue corresponding to a given eigenvector being the square root of the corresponding eigenvalue of  $A$ .

The matrix returned by such a function `msqrt` is called the square root of the matrix  $A$  and it will be denoted by  $A^{1/2}$ .

The second question concerns the generation of normal random vectors in `S-Plus`. In other words, we write an `S-Plus` function to play the role of the function `rnorm` in the case of multidimensional random vectors. Such a function does exist in the `S-Plus` distribution. It is called `mvrnorm`. The goal of this second question is to understand how such a generation method works.

2. Write an `S-function`, call it `vnorm`, with arguments `Mu`, `Sigma` and `N` which:

- checks that `Mu` is a vector, exits if not, and otherwise reads its dimension, say  $L$ ;
- checks that `Sigma` is an  $L \times L$  symmetric matrix with nonnegative eigenvalues and exits, if not;
- creates a numeric array with  $N$  rows and  $L$  columns and fills it with independent random numbers with the standard normal distribution  $N(0, 1)$ ;
- treats each row of this array as a vector, and multiplies it by the square root of the matrix `Sigma` (as computed in question **L.1** above) and adds the vector `Mu` to the result;
- returns the random array modified in this way.

The array produced by the function `vnorm` is a sample of size  $N$  of  $L$ -dimensional random vectors (arranged as rows of the matrix outputted by `vnorm`) with the normal distribution with mean `Mu` and variance/covariance matrix `Sigma`. Indeed, this function implements the following simple fact reviewed during the lectures:

If  $X$  is an  $L$ -dimensional normal vector with mean 0 and covariance matrix given by the  $L \times L$  identity matrix (i.e. if all the  $L$  entries of  $X$  are independent  $N(0, 1)$  random variables), then:

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

is an  $L$ -dimensional normal vector with mean  $\mu$  and variance/covariance matrix  $\Sigma$ .

---

## NOTES & COMPLEMENTS

This chapter concentrated on multivariate distributions and on dependence between random variables. The discussion of the various correlation coefficients is modeled after the standard treatments which can be found in most multivariate statistics books. The originality of this chapter lies in the statistical analysis of the notion of dependence by way of copulas. The latter are especially important when the marginal distributions have heavy tails, which is the case in most financial applications as we saw in the first chapter. The recent renewal of interest in the notion of copula prompted a rash of books on the subject. We shall mention for example the monograph [65] of R.B. Nelsen, or the book of D. Drouot Mari and S. Kotz [29]. We refer the interested reader to their bibliographies for further references on the various notions of dependence and copulas.

The `S-Plus` methods used in this chapter to estimate copulas and generate random samples from multivariate distributions identified by their copulas were originally developed for the library `EVANESCE` [17] developed by J. Morrisson and the author. As explained in the

Notes & Complements of Chapter 1 this library has been included in the `S+FinMetrics` module of the commercial `S-Plus` distribution.

To the best of my knowledge, the first attempt to apply principal component analysis to the yield curve is due to Litterman and Scheinkmann [57]. Rebonato's book [70], especially the short second chapter, and the book [3] by Anderson, Breeden, Deacon, Derry, and Murphy, are good sources of information on the statistical properties of the yield curve. Discussions of interest rate swap contracts and their derivatives can also be found in these books. The reader interested in a statistical discussion of the fixed income markets with developments in stochastic analysis including pricing and hedging of fixed income derivatives, can also consult the monograph [21]. An application of PCA to variable selection in a regression model is given in Problem 4.17 of Chapter 4.

The decomposition of a data set into its principal components is known in signal analysis as the Karhunen-Loève decomposition, and the orthonormal basis of principal components is called the Karhunen-Loève basis. This basis was identified as optimal for compression purposes. Indeed, once a signal is decomposed on this basis, most of the coefficients are zero or small enough to be discarded without significantly affecting the information contained in the signal. Not surprisingly, the optimality criterion is based on a form of the entropy of the set of coefficients. PCA is most useful for checking that data do contain features which are suspected to be present. For this reason, some authors suggest to remove by regression the gross features identified by a first PCA run, and to then run PCA on the residuals. PCA has been successfully used in many applications, especially in signal and image analysis. For example, the Notes & Complements to Chapter 5 contain references to studies using PCA in brain imaging.

Statistical Analysis of Financial Data in S-Plus

Carmona, R.

2004, XVI, 455 p. 138 illus., Hardcover

ISBN: 978-0-387-20286-0