First steps of multivariate data analysis

January 22, 2024

Let's Have Some Coffee

We reproduce the coffee example from Carmona, page 60 ff.

This vignette is the first excursion away from univariate data. Here we consider samples of simultaneous observations of at least two numerical variables. The data we consider are daily prices of Brazilian and Colombian coffee. For each day of the period from 1986-01-10 to 1999-01-01 the data contain the daily log-returns.

To analyze the coffee prices we first load the two datasets (Brazilian and Columbian coffee):

```
R> load("coffee.RData")
```

Then we show the first values of the data

```
R> head(BCofLRet)
```

```
[1]  0.00000000  0.00000000  0.00000000  0.00000000 -0.01227009  0.00000000
```

```
R> head(CCofLRet)
```

```
[1]  0.00000000  0.00000000  0.00000000  0.00000000 -0.01801851  0.00000000
```

and plot the relationship of both log-returns in a scatterplot.

```
R> plot(BCofLRet, CCofLRet)
```

The resulting scatterplot (Figure 1) shows that some prices do not change from one day to the next day and so their log-returns are equal to zero. This effect indicates that the cumulative distribution functions have jumps at 0.0.

In fact,

```
R> table(BCofLRet != 0, CCofLRet != 0)
```

```
       FALSE TRUE
  FALSE  1726   85
  TRUE    192 1383
```

shows the frequencies of the change/no-change combinations (indicating that it is in fact rather tricky to "correctly" model the data). For the further analysis, we simply restrict ourselves to days where both daily log-returns are different from zero:

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

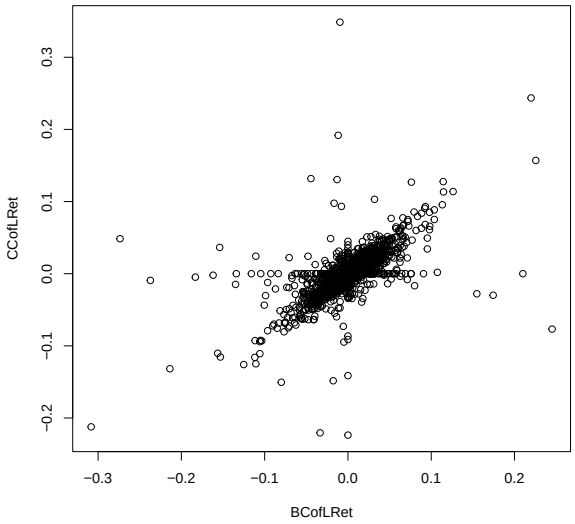

Figure 1: Scatterplot of the daily log-returns of Brazilian and Columbian coffee.

The proportion of days retained is

```
R> mean(NZ)

[1] 0.4084465
```

From now on we use the new bivariate sample, but we should keep in mind that if we want to compute statistics of actual log-returns, we need to put the zeros back.

In a next step, we are interested whether the joint distribution of `BLRet` and `CLRet` is normal. One way (using formal statistical inference) to do this is the function `mshapiro.test` form R package **mvnormtest**. Here, we will follow a more intuitive way: we compare graphically the the joint empirical distribution of `BLRet` and `CLRet` with the empirical distribution of a bivariate sample which has the same means and variance-covariance matrix. We set up a bivariate data matrix

```
R> Rets <- cbind(BLRet, CLRet)
R> N <- nrow(Rets)
```

and estimate the parameters

```
R> Mu <- colMeans(Rets)
R> Mu

        BLRet         CLRet
-0.0008518472 -0.0003550291

R> Sigma <- var(Rets)
R> Sigma

             BLRet        CLRet
BLRet 0.0014690263 0.0008822318
CLRet 0.0008822318 0.0011124274
```

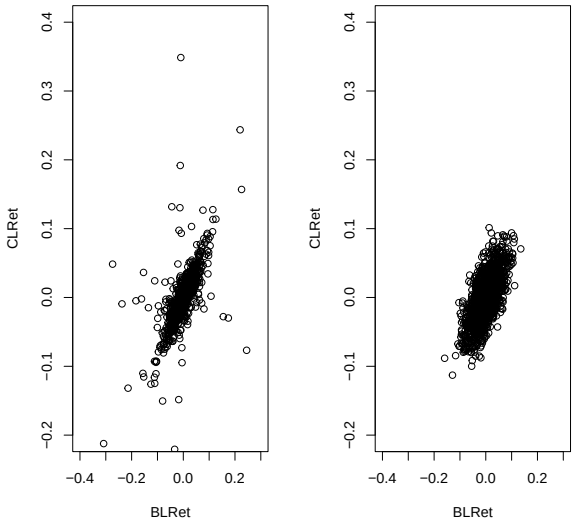

Figure 2: Scatterplot of daily log returns and simulated distribution

(note that `var` automatically computes the variance-covariance matrix of a given data matrix taking the columns as variables, but `mean` does not compute the variable means as column means).

We use the mean vector `Mu`, the variance-covariance matrix `Sigma` and the sample size `N` to generate a sample from a bivariate Gaussian distribution. This is done with the command

```
R> CNsim <- MASS::mvrnorm(N, mu = Mu, Sigma = Sigma)
```

Then we create scatterplots of the observed and simulated data, see Figure 2:

```
R> op <- par(mfrow =c(1,2))
R> plot(Rets,  xlim =c(-.4,.3), ylim =c(-.2,.4))
R> plot(CNsim, xlim =c(-.4,.3), ylim =c(-.2,.4))
R> par(op)
```

The comparison of these plots shows rather different pictures, suggesting that the data does not come from a bivariate normal distribution.

If we wanted a more formal test of agreement with normality (or not). R package `mvnormtest` provides the (multivariate) Shapiro-Wilk test via function `mshapiro.test`. (Note that this requires using `t` to transpose the data matrix.)

```
R> library("mvnormtest")
R> MS <- mshapiro.test(t(Rets))
R> MS

        Shapiro-Wilk normality test

data:  Z
W = 0.63866, p-value < 2.2e-16
```

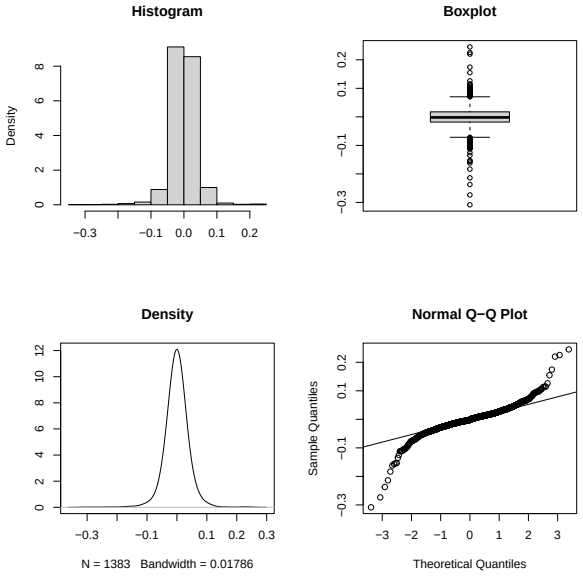

Figure 3: Exploratory data analysis of the Brazilian coffee daily log returns, after removing of the zeros.

Since the p-value 4.5922150526554e-47 is below 0.05 we can reject the null hypothesis of an underlying bivariate normal distribution.

There are many possible reasons why the joint distribution of `BLRet` and `CLRet` is not normal. In general, it is because at least one of the variables is not normal. Therefore, we check if neither the marginal distribution of `BLRet` nor of `CLRet` is normal.

We thus create some plots of the marginal distributions. The commands

```
R> op <- par(mfrow = c(2, 2))
R> hist(BLRet, probability=TRUE, main="Histogram", xlab="")
R> boxplot(BLRet, main ="Boxplot")
R> iqd <- IQR(BLRet)
R> plot(density(BLRet, width = 2 * iqd),  ylab = "",type = "l", main = "Density")
R> qqnorm(BLRet)
R> qqline(BLRet)
R> par(op)
```

produce four graphical representations (a histogram, a normal Q-Q plot, a box-plot and a kernel density estimate) of the `BLRet` data, see Figure 3.

In the above, `par` is used to divide the plotting area into a 2×2 grid. The next commands produce the histogram, the boxplot and the plot of kernel density estimate. Finally, the last two commands yield the normal Q-Q plot. To be able to conveniently reproduce such plots, it makes sense to integrate them into a function:

```
R> eda.shape <- function(x)
+ {
+     op <- par(mfrow = c(2, 2))
+     on.exit(par(op))
+     hist(x, probability = TRUE, main = "Histogram", xlab = "")
```

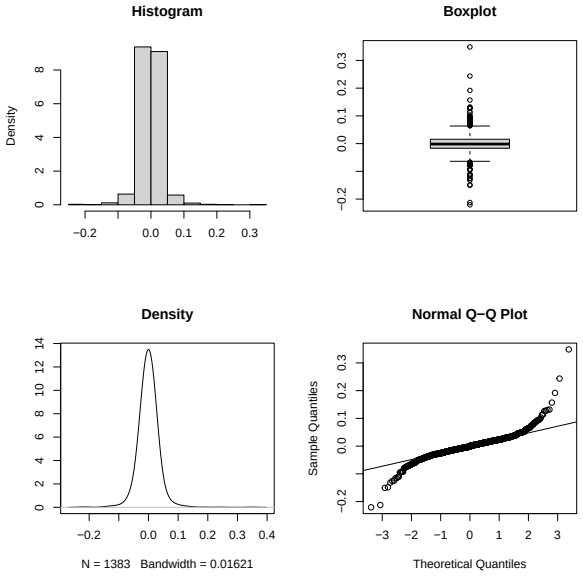

Figure 4: Exploratory data analysis of the daily log-returns of the Columbian coffee prices, after removing of the zeros.

```
+     boxplot(x, main = "Boxplot")
+     iqd <- IQR(x)
+     plot(density(x, width = 2 * iqd), ylab = "", type = "l",
+         main = "Density")
+     qqnorm(x)
+     qqline(x)
+ }
```

Using this function we create the plots for the daily log-returns of the Columbian coffee prices, see Figure 4:

```
R> eda.shape(CLRet)
```

The presence of tails which are heavier than normal are suggested by the boxplots, which show a very large number of observations outside the box. However, when it comes to the tails, the clearest diagnostic can be performed with Q-Q plots. The departures from the Q-Q lines (`qqline`) are a clear indication that the tails are much heavier than the tails of the normal distributions with the same means and variances. Alternatively, other families of distributions, like Pareto distributions can be used.

As in the case of the S&P 500 index—described in the "Pareto vignette"—we use the function `GNG_fit()` to fit a generalized Pareto distribution to our data.

```
R> library(mistr)
R> B.est <- GNG_fit(BLRet, start = c(break1 = -0.06, break2 = 0.06, mean = 0,
+                                    sd = 0.0265, shape1 = 0.15, shape2 = 0.15))
```

Further, we check graphically the quality of the fit as illustrated in Figure 5:

```
R> plot(B.est)
```

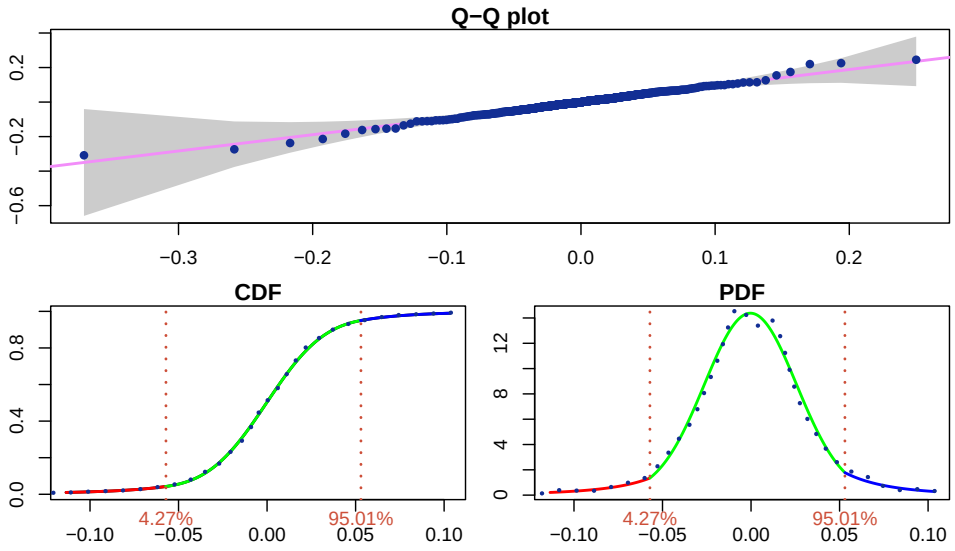

Figure 5: Estimation of a generalized Pareto distribution for the daily log-returns of the Brazilian coffee.

The analysis of the heavy tail nature also has be performed for the log-returns of the Colombian Coffee prices.

```
R> C.est <- GNG_fit(CLRet, start = c(break1 = -0.07, break2 = 0.07, mean = 0,
+                                    sd = 0.0315, shape1 = 0.15, shape2 = 0.15))
```

Monte Carlo Simulation of the Coffee Returns

Considering a portfolio including both Brazilian as well as Colombian Coffee futures contracts, a typical task is to analyze the risk of such portfolio.

Therefore, we proceed with a Monte Carlo simulation of samples of log-returns. With the following commands we generate two samples (`BLRet.sim` and `CLRet.sim`) of the same sizes as the original data.

```
R> BLRet.sim <- r(distribution(B.est), length(BLRet))
R> CLRet.sim <- r(distribution(C.est), length(CLRet))
```

We use Q-Q plots to compare the fitted distributions to the corresponding empirical data (Figure 6):

```
R> op <- par(mfrow = c(2,1))
R> plot(B.est, which = "qq")
R> plot(C.est, which = "qq")
R> par(op)
```

We note that the simulated values capture the marginal distributions with great precision and hence be used for statistics involving the log-returns separately.

However, they can not be used for joint statistics since they do not capture the dependencies between the log-returns, which is clearly indicated by plotting them together in a scatterplot, as it is done in Figure 7:

```
R> op <- par(mfrow = c(1, 2))
R> plot(BLRet, CLRet, xlim=c(-.4,.3), ylim=c(-.3,.4))
```

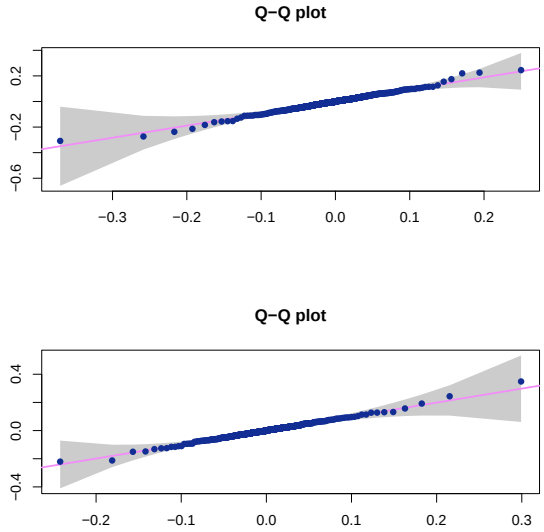

Figure 6: Q-Q plot of the fitted distribution against the empirical log-returns of Brazilian (upper plot) and Colombian (lower plot) coffee prices.

```
R> plot(BLRet.sim, CLRet.sim, xlim=c(-.4,.3), ylim=c(-.3,.4))
R> par(op)
```

Thus, additional effort is needed to better understand the dependencies between the two log-return variables and to include their effects in Monte Carlo Simulations.

In a first step, we discuss two famous measures of dependence, *Kendall's* τ and *Spearman's* ρ.

Kendall's τ measures 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 and is formally given by

$$\tau(X,Y) = \frac{1}{\binom{n}{2}} \sum_{1 \le i \le j \le n} \text{sign}((x_i - x_j)(y_i - y_j)) \tag{1}$$

Spearman's ρ is defined as the ordinary Pearson product-moment correlation of the ranks of the data, and computed as

$$\rho(X,Y) = \frac{12}{n(n^2-1)} \sum_{i=1}^{n} \left(\text{rank}(x)_i - \frac{n+1}{2}\right)\left(\text{rank}(y)_i - \frac{n+1}{2}\right) \tag{2}$$

(in case of no ties).

In R we can estimate these correlation coefficients with the command `cor.test` from the **stats** package.

```
R> cor.test(BLRet, CLRet, method ="kendall")

        Kendall's rank correlation tau

data:  BLRet and CLRet
z = 38.337, p-value < 2.2e-16
alternative hypothesis: true tau is not equal to 0
```

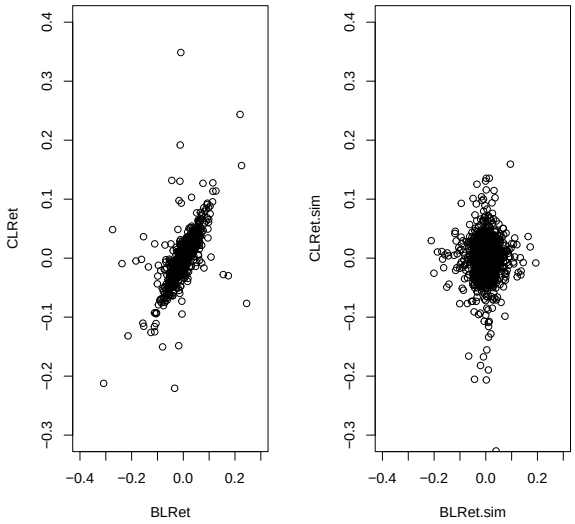

Figure 7: Scatterplot of the empirical and simulated values.

```
sample estimates:
     tau
0.688215

R> cor.test(BLRet, CLRet, method ="spearman")

        Spearman's rank correlation rho

data:  BLRet and CLRet
S = 72455928, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.8356541
```

where `method="kendall"` estimates Kendall's τ and `method="spearman"` is used for Spearman's ρ.

Copulas and Random Simulations

Before we introduce copulas, we remember that if X is a random variable with continuous cumulative distribution function F_X and quantile function F_X^{-1}, then $F_X^{-1}(X)$ has a uniform distribution on $[0, 1]$. We can therefore transform the coffee log-returns and create a bivariate sample in the unit square in such a way that both marginal distributions are uniform.

Indeed this can be done by evaluating each marginal cdf exactly at the sample points. We use our fitted distribution functions of the log-returns as proxies for the theoretical distributions,

```
R> U <- p(distribution(B.est), BLRet)
R> V <- p(distribution(C.est), CLRet)
```

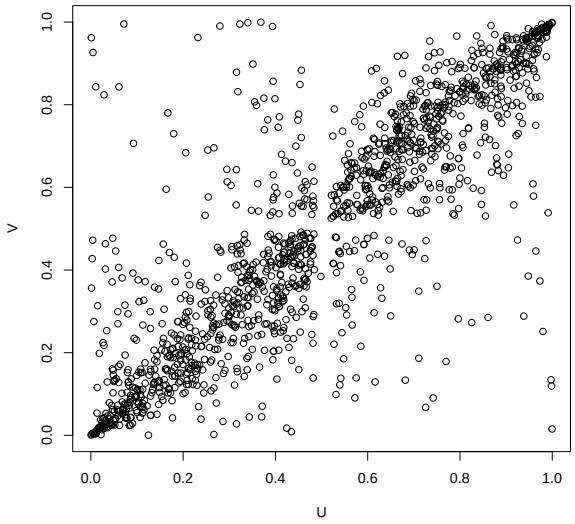

Figure 8: Dependency between the coffee log-returns after removing the effects of the marginal distributions

and plot them, as illustrated in Figure 8.

```
R> plot(U,V)
```

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, v_i) are paired.

This motivates to the following abstract definition of capturing the dependence between several random variables

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

If $U_1, U_2, \ldots, U_n$ are $U(0,1)$, then the function C from $[0,1] \times \cdots \times [0,1]$ into $[0,1]$ defined by

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

is a copula.

Because of the lack of data in the tails of the marginal distributions, copulas are best estimated by parametric methods. We choose a function `BiCopSelect` from the package **VineCopula** which selects an appropriate bivariate copula family for given bivariate copula data using one of a range of methods and estimates the corresponding parameters by maximum likelihood estimation.

```
R> library(VineCopula)
R> cop <- BiCopSelect(U, V, familyset = NA, selectioncrit = "BIC")
R> cop

Bivariate copula: t (par = 0.9, par2 = 2, tau = 0.71)
```

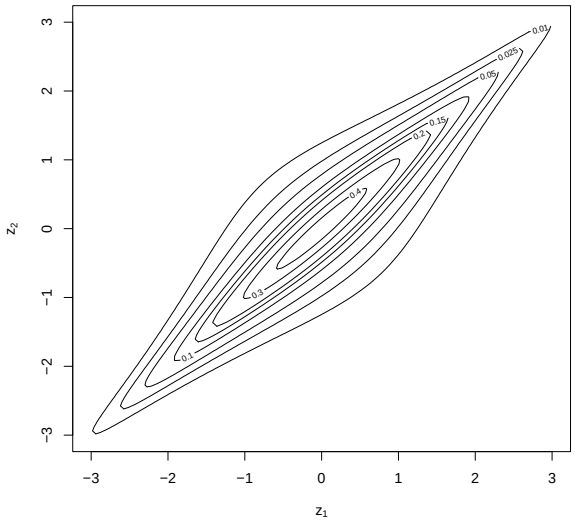

Figure 9: Contour plot of the fitted t-copula.

One can use the `familyset` argument to specify the copula families to use (e.g., `familyset = 4` uses the Gumbel copula only).

Function `BiCopSelect` returns the object `cop` of class `BiCop` and the resulting contour plot is illustrated in Figure 9.

```
R> plot(cop, type = "contour")
```

Monte Carlo Simulation with Copulas

Now, we consider the problem of drawing random samples from a bivariate distribution. The following commands produce a bivariate sample from the estimated joint distribution of the coffee log-returns.

```
R> SD <- BiCopSim(N, obj = cop)
R> Xsim <- q(distribution(B.est), SD[,1])
R> Ysim <- q(distribution(C.est), SD[,2])
```

Function `BiCopSim` produces a bivariate sample from the copula object `cop`, which needs be of class `BiCop`. According to the definition of a copula, `SD[,1]` and `SD[,2]`, respectively, are uniformly distributed random samples over the unit interval.

The other two commands produce samples `Xsim` and `Ysim` from distributions given by the `GNG` objects `B.est` and `C.est`, respectively.

Like in Figure 7 we produce a scatterplot of the simulated values `Xsim` and `Ysim`. Compared to figure 7, we see, that this plot captures rather well the characteristics of the empirical sample and the differences to Figure 7 are striking, see Figure 10.

```
R> op <- par(mfrow = c(1, 2))
R> plot(BLRet, CLRet, xlim=c(-.4,.3), ylim=c(-.3,.4))
R> plot(Xsim, Ysim, xlim=c(-.4,.3), ylim=c(-.3,.4))
R> par(op)
```

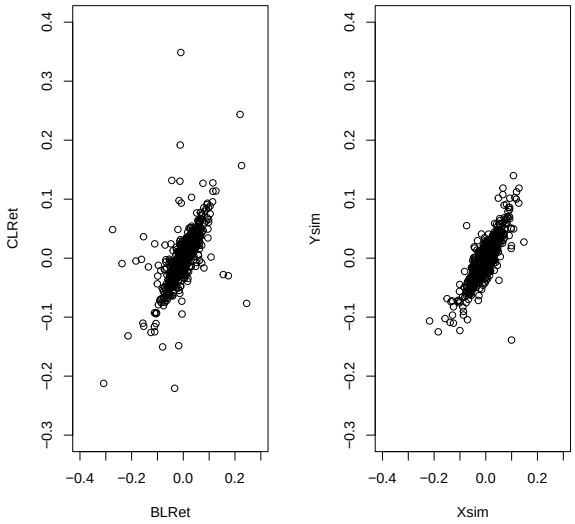

Figure 10: Scatterplot of empirical log returns and MC sample produced with dependence as captured by the fitted copula.

As a further evidence we want to check our fit with numerical measures. This can be done with

```
R> cor(Xsim, Ysim, method ="kendall")
```

```
[1] 0.6975743
```

for Kendall's τ and with

```
R> cor(Xsim, Ysim, method ="spearman")
```

```
[1] 0.858306
```

for Spearman's ρ.