

- 1.Portfolio Variance and Multivariate t-Model Fit (MLE for  $v$ )
- 2.Bivariate Samples with Different Dependencies
- 3.T-Copula Model and Dependence Analysis
- 4.Multivariate Normal Copula and Marginal Distributions
- 5.Simulate random variates

## 3.2\_Multivariate-Return-Modelling&Copulas-2

This is sourced from STAD70 course practice questions and sample R code taught by professor Sotos. If you have any questions/concerns/comments feel free to email me: [cristal.wang111@gmail.com](mailto:cristal.wang111@gmail.com) (mailto:cristal.wang111@gmail.com).

### 1.Portfolio Variance and Multivariate t-Model Fit (MLE for $v$ )

Loading csv dataset `berndtInvest` from SDAFE website  
(<https://people.orie.cornell.edu/davidr/SDAFE2/index.html>)

```
berndtInvest = read.csv("berndtInvest.csv")
Berndt = as.matrix(berndtInvest[, 2:5])
cov(Berndt)
```

```
##           CITCRP      CONED      CONTIL      DATGEN
## CITCRP 0.006556442 0.0010933247 0.006052187 0.0055091027
## CONED  0.001093325 0.0025272604 0.000795463 0.0006124665
## CONTIL 0.006052187 0.0007954630 0.022710259 0.0050502504
## DATGEN 0.005509103 0.0006124665 0.005050250 0.0162664367
```

```
cor(Berndt)
```

```
##           CITCRP      CONED      CONTIL      DATGEN
## CITCRP 1.0000000 0.2685901 0.4959835 0.5334584
## CONED  0.2685901 1.0000000 0.1049987 0.0955237
## CONTIL 0.4959835 0.1049987 1.0000000 0.2627578
## DATGEN 0.5334584 0.0955237 0.2627578 1.0000000
```

# 1.1.Portfolio Variance

Suppose the four variables being used are denoted by  $X_1, \dots, X_4$ . Use the sample covariance matrix to estimate the variance of  $0.5X_1 + 0.3X_2 + 0.2X_3$ .

- $t(a)$  : is the transpose of a vector or matrix  $a$  ;  $a \% \% b$  : is the matrix product of  $a$  and  $b$  .

```
COV = var(Berndt)    # variance-covariance matrix of returns
w = c(.5, .3, .2, 0) # vector of weights

t(w) \% \% COV \% \% w    # variance of linear combination of returns (portfolio)
```

```
##           [,1]
## [1,] 0.004408865
```

## 1.2.Multivariate t-Model Fit (unknown Degrees of Freedom)

### 1.2.1.Estimating DF( $\nu$ ) via Profile Likelihood

Fit a multivariate-t model to the data using the function `cov.trob` in the `MASS` package. This function computes the MLE of the mean and covariance matrix with a fixed value of  $\nu$  . To find the MLE of  $\nu$  , the following code computes the profile log-likelihood for  $\nu$  .

```
library(MASS) # needed for cov.trob
library(mnormt) # needed for dmt
df = seq(2.5, 8, 0.01)
n = length(df)
loglik_profile = rep(0, n)
for(i in 1:n)
{
  fit = cov.trob(Berndt, nu = df[i])
  mu = as.vector(fit$center)
  sigma = matrix(fit$cov, nrow = 4)
  loglik_profile[i] = sum(log(dmt(Berndt, mean = fit$center, S= fit$cov, df = df[i])))
}
```

### 1.2.2.MLE and 90% CI for Degrees of Freedom ( $\nu$ )

Using the results produced by the code above, find the MLE of  $\nu$  and a 90% profile likelihood confidence interval for  $\nu$  . Include your R code with your work. Also, plot the profile log-likelihood and indicate the MLE and the confidence interval on the plot.

- Adds horizontal line at 90% confidence threshold level
- Threshold based on chi-squared distribution: `qchisq(0.9, 1)`
  - Gives cutoff for 90% CI with 1 degree of freedom

#### Theoretical Foundation

- **Likelihood ratio test statistic** follows asymptotic chi-squared distribution:

$$\Lambda(\theta) = 2(\ell(\hat{\theta}) - \ell(\theta)) \sim \chi_1^2, \text{ where } MLE(\theta) = \hat{\theta}$$

1. Set threshold  $\chi_{1,\alpha}^2$  (e.g., 2.71 for 90% CI)
2. Likelihood-based a = 90% CI includes all  $\theta$  satisfying:

$$\Lambda(\theta) = 2[\ell(\hat{\theta}) - \ell(\theta)] \leq \chi_{1,0.9}^2 \iff \ell(\theta) \geq \ell(\hat{\theta}) - \frac{1}{2}\chi_{1,0.9}^2$$

```

CI_threshold = 0.9 # here is 90%
plot(df, 2 * loglik_profile, type = "l", cex.axis = 1.5, # plot 2*Log(Likelihood)
      cex.lab = 1.5, ylab = "2 * log(likelihood)", lwd = 2)
abline(h = 2 * max(loglik_profile) )

### plot MLE
max_ind = which.max(loglik_profile) # Find index of maximum Log(Likelihood)
points( df[max_ind], 2*loglik_profile[max_ind], pch=15, cex=2) # plot maximum
text( df[max_ind], 2*loglik_profile[max_ind], pos=1, labels = "MLE")

abline(h = 2 * max(loglik_profile) - qchisq(CI_threshold, 1 ) ) # plot 90% CI Level

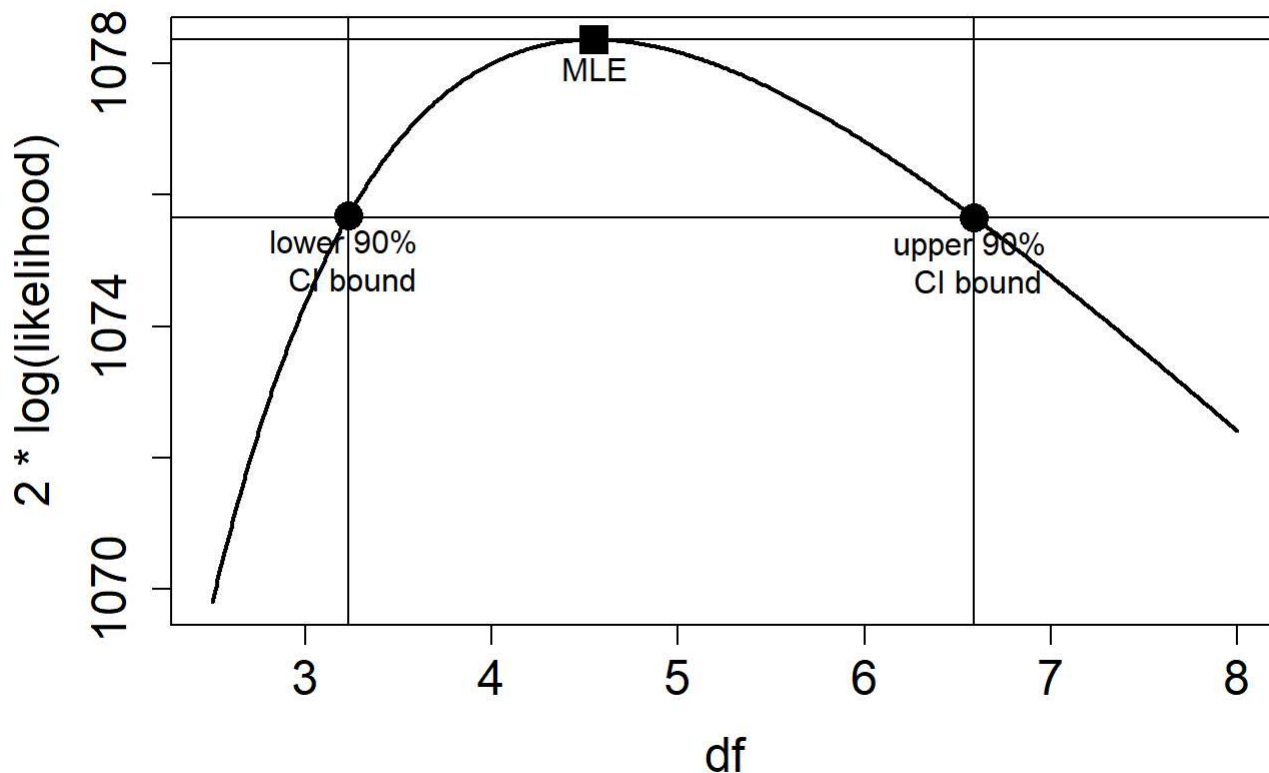
### find indexes of lower & upper bounds of CI_threshold (90%) CI
lower_ind = head( which( loglik_profile > max(loglik_profile) - qchisq(CI_threshold, 1 )/2
), 1 )
upper_ind = tail( which( loglik_profile > max(loglik_profile) - qchisq(CI_threshold, 1 )/2
), 1 )

abline(v = df[lower_ind] ); abline(v = df[upper_ind] )

### plot lower bound
points( df[lower_ind], 2*loglik_profile[lower_ind], pch=16, cex=2)
text( df[lower_ind], 2*loglik_profile[lower_ind], pos=1, labels = "lower 90% \n CI bound")

### plot upper bound
points( df[upper_ind], 2*loglik_profile[upper_ind], pch=16, cex=2)
text( df[upper_ind], 2*loglik_profile[upper_ind], pos=1, labels = "upper 90% \n CI bound")

```



```
print( rbind( c( "lower", "max", "upper"), df[ c(lower_ind, max_ind, upper_ind)] ))
```

```
##      [,1]  [,2]  [,3]
## [1,] "lower" "max" "upper"
## [2,] "3.23"  "4.55" "6.59"
```

## 2. Bivariate Samples with Different Dependencies

### 2.1. Generating and Plotting Bivariate Samples

The following code generates and plots four bivariate samples. Each sample has univariate marginals that are standard  $t(df = 3)$ -distributions. However, the dependencies are different.

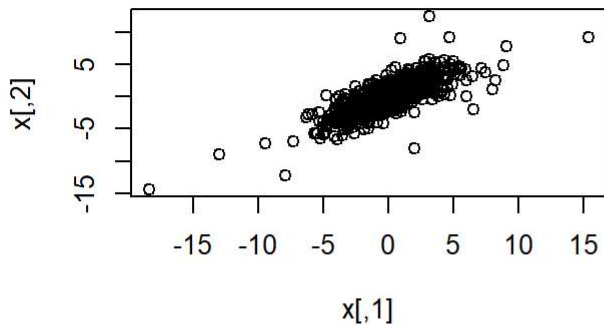
```
library(MASS) # need for mvrnorm
par(mfrow=c(2,2))
N = 2500
nu = 3
set.seed(5640)
cov=matrix(c(1, 0.8, 0.8, 1), nrow = 2)
x= mvrnorm(N, mu = c(0, 0), Sigma = cov)
w = sqrt(nu / rchisq(N, df = nu))
x = x * cbind(w, w)
plot(x, main = "(a) Correlated with Tail Dependence")

set.seed(5640)
cov=matrix(c(1, 0.8, 0.8, 1),nrow = 2)
x= mvrnorm(N, mu = c(0, 0), Sigma = cov)
w1 = sqrt(nu / rchisq(N, df = nu))
w2 = sqrt(nu / rchisq(N, df = nu))
x = x * cbind(w1, w2)
plot(x, main = "(b) Correlated without Tail Dependence")

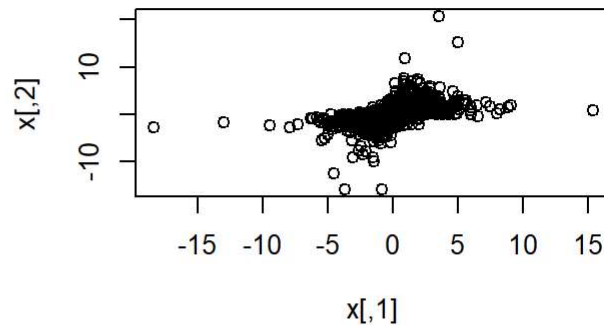
set.seed(5640)
cov=matrix(c(1, 0, 0, 1), nrow = 2)
x= mvrnorm(N, mu = c(0, 0), Sigma = cov)
w1 = sqrt(nu / rchisq(N, df = nu))
w2 = sqrt(nu / rchisq(N, df = nu))
x = x * cbind(w1, w2)
plot(x, main = "(c) Independent")

set.seed(5640)
cov=matrix(c(1, 0, 0, 1), nrow = 2)
x= mvrnorm(N, mu = c(0, 0), Sigma = cov)
w = sqrt(nu / rchisq(N, df = nu))
x = x * cbind(w, w)
plot(x, main = "(d) Uncorrelated with Tail Dependence")
```

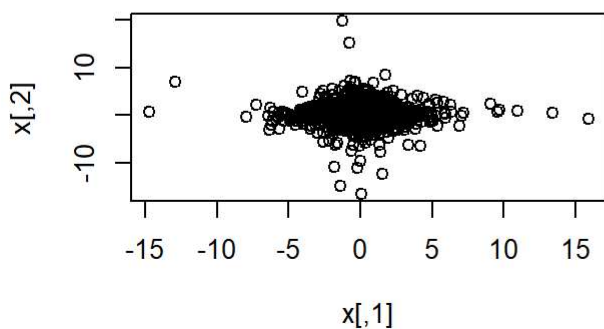
(a) Correlated with Tail Dependence



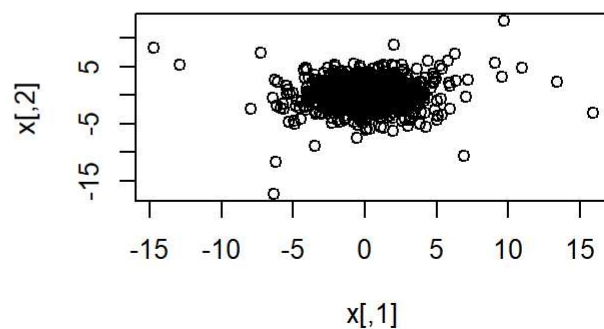
(b) Correlated without Tail Dependence



(c) Independent



(d) Uncorrelated with Tail Dependence



Note the use of these R commands:

- `set.seed` to set the seed of the random number generator,
- `mvnrm` to generate multivariate normally distributed vectors,
- `rchisq` to generate  $\chi^2$ -distributed random numbers,
- `cbind` to bind together vectors as the columns of a matrix, and
- `matrix` to create a matrix from a vector.

In R, `a * b` is elementwise multiplication of same-size matrices `a` and `b`, and `a %*% b` is matrix multiplication of conforming matrices `a` and `b`.

## (1).Independent Variates

Which sample has independent variates? Explain your answer.

- Samples (a) and (b) are (positively) correlated, because they have arise from (randomly scaled) correlated Normals (with  $\rho = 0.8$ ).
- Sample (d) exhibits *tail-dependence*, because the uncorrelated/independent Normals are scaled by the *same* random ( $\chi^2$ -based) factor.
- Only sample (c) is independent, because the two independent Normal variates are scaled by independent random factors.

## (2).Correlated Variates without Tail Dependence

Which sample has variates that are correlated but do not have tail dependence? Explain your answer.

- Sample (b) has correlated variates (because it is based on correlated Normals), but does not have tail-dependence, because its variates were scaled by independent random factors. The plot shows that extreme values seem to occur independently in the two dimensions (i.e., extreme values are either horizontally or vertically extreme, but not both.)

### (3).Uncorrelated Variates with Tail Dependence

Which sample has variates that are uncorrelated but with tail dependence? Explain your answer.

- Sample (d) has uncorrelated variates (based on uncorrelated Normals) but with tail-dependence, because they are scaled by the same random factor. The plot shows that extreme values tend to occur simultaneously in both dimensions.

## 3.T-Copula Model and Dependence Analysis

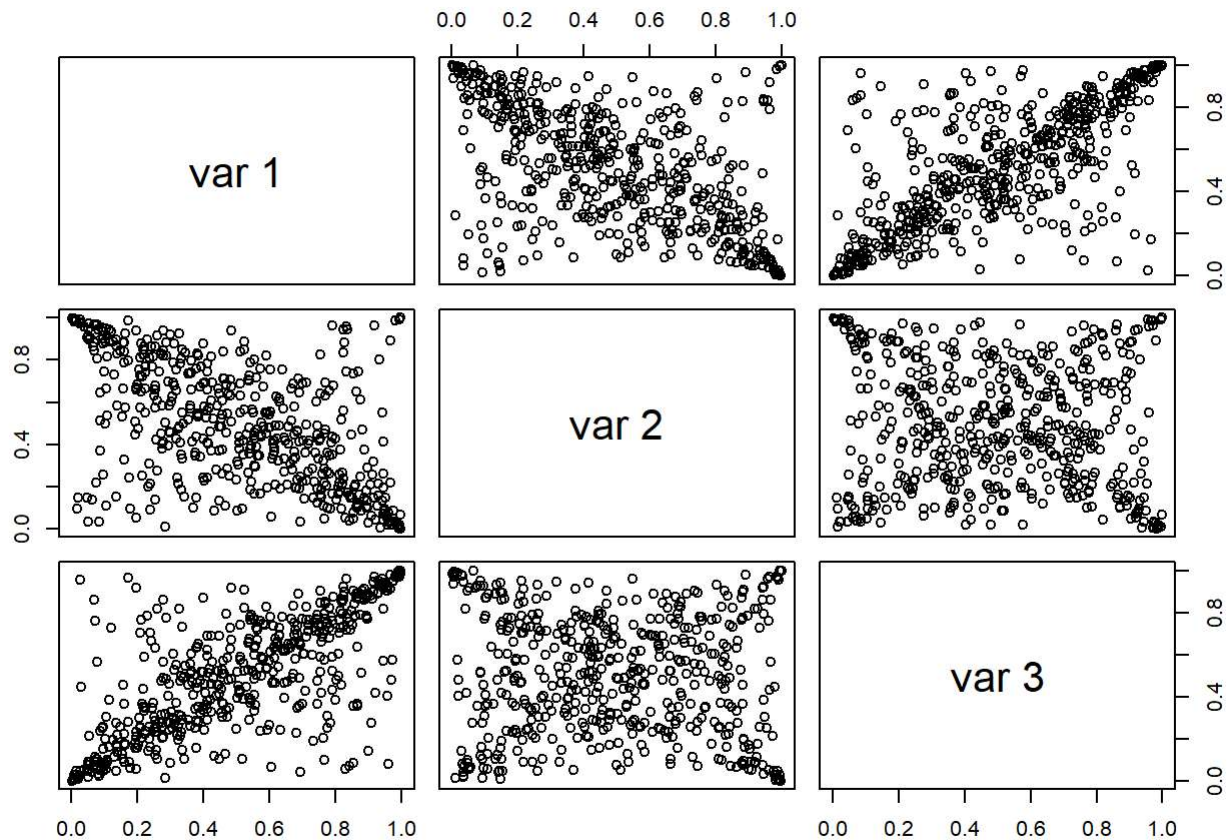
### 3.1.Sampling from a t-copula

Sample t-copula model with degrees of freedom parameter be 1, sample size be 500. The correlation matrix is unstructured, meaning that it is an arbitrary correlation matrix. Here, the correlation matrix is

$$\begin{bmatrix} 1 & -0.6 & 0.75 \\ -0.6 & 1 & 0 \\ 0.75 & 0 & 1 \end{bmatrix}$$

- The R code generates data from a copula. `library(copula)` loads the copula package.
- `cop_t_dim3 = tCopula(dim = 3, param = c(-0.6,0.75,0), dispstr = "un", df = 1)` define a copula object. At this point, nothing is done with the copula object — it is simply defined.
  - The `param` vector specifies the pairwise linear correlations between latent t-distributed variables, These are the correlations in the underlying t-distribution from which the copula is derived.
- However, the copula object is used in `rand_t_cop = rCopula(n = 500, copula = cop_t_dim3)` to generate a random sample from the specified copula model.
  - The values in `rand_t_cop` (the result of `rCopula`) are not t-distributed anymore — they are uniform on  $[0, 1]$ , because copulas work in the copula space where marginals (t) are transformed to uniforms.
- The remaining lines create a scatter plot matrix of the sample and print its sample Pearson correlation matrix.

```
library(copula)
cop_t_dim3 = tCopula(dim = 3, param = c(-0.6,0.75,0), dispstr = "un", df = 1)
set.seed(5640)
rand_t_cop = rCopula(n = 500, copula = cop_t_dim3)
pairs(rand_t_cop)
```



```
cor(rand_t_cop)
```

```
##           [,1]      [,2]      [,3]
## [1,]  1.0000000 -0.54999514  0.70707296
## [2,] -0.5499951  1.00000000 -0.06538499
## [3,]  0.7070730 -0.06538499  1.00000000
```

### 3.1.1.Examination of the Scatterplot Matrix

Examine the scatterplot matrix and answer the questions below.

#### (1).Components Independence

Components 2 and 3 are uncorrelated. Do they appear independent? Why or why not?

No. Components 2 and 3 would be **uniformly scattered over the unit square if they were independent**. Clearly, the scatter is not uniform, so they do not appear independent.

#### (2).Signs of Tail Dependence

Do you see signs of tail dependence? If so, where?

Yes. The non-uniformity mentioned in (1) is that there are more data in the corners, which shows that extreme values tend to occur together, although because of the zero correlation, a positive extreme value of one component is equally likely to be paired with a positive or negative extreme value of the other component.



### (3).Effects of Dependence on the Plots

What are the effects of dependence upon the plots?

The effect of tail dependence is **the tendency of extreme values to pair**.

The negative correlation of components 1 and 2 shows in the concentration of the data along the diagonal from upper left to lower right. Positive extreme values in one component tend to pair with negative extreme values of the other component.

The positive correlation of components 2 and 3 shows in the concentration of the data along the diagonal from lower left to upper right. Positive extreme values in one component tend to pair with positive extreme values of the other component.

### (4).Discrepancy Between Copula and Sample Correlations

The nonzero correlations in the copula do not have the same values as the corresponding sample correlations. Do you think this is just due to random variation or is something else going on? If there is another cause besides random variation, what might that be?

- To help answer this question, you can get confidence intervals for the Pearson correlation: For example, `cor.test(rand_t_cop[,1],rand_t_cop[,3])` : will give a confidence interval (95 percent by default) for the correlation (Pearson by default) between components 1 and 3. Does this confidence interval include 0.75?

The output is below, and the confidence interval is (0.6603, 0.7484), which does not quite include 0.75. This is not surprising. 0.75 is the correlation between the t-distributed random variables that define the copula and need not be the same as the uniformly-distributed variables in the copula itself.

```
cor.test(rand_t_cop[,1],rand_t_cop[,3])
```

```
##
## Pearson's product-moment correlation
##
## data:  rand_t_cop[, 1] and rand_t_cop[, 3]
## t = 22.314, df = 498, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6603249 0.7483624
## sample estimates:
##          cor
## 0.707073
```

## 4.Multivariate Normal Copula and Marginal Distributions

Consider the R code below, using the `copula` library in R.

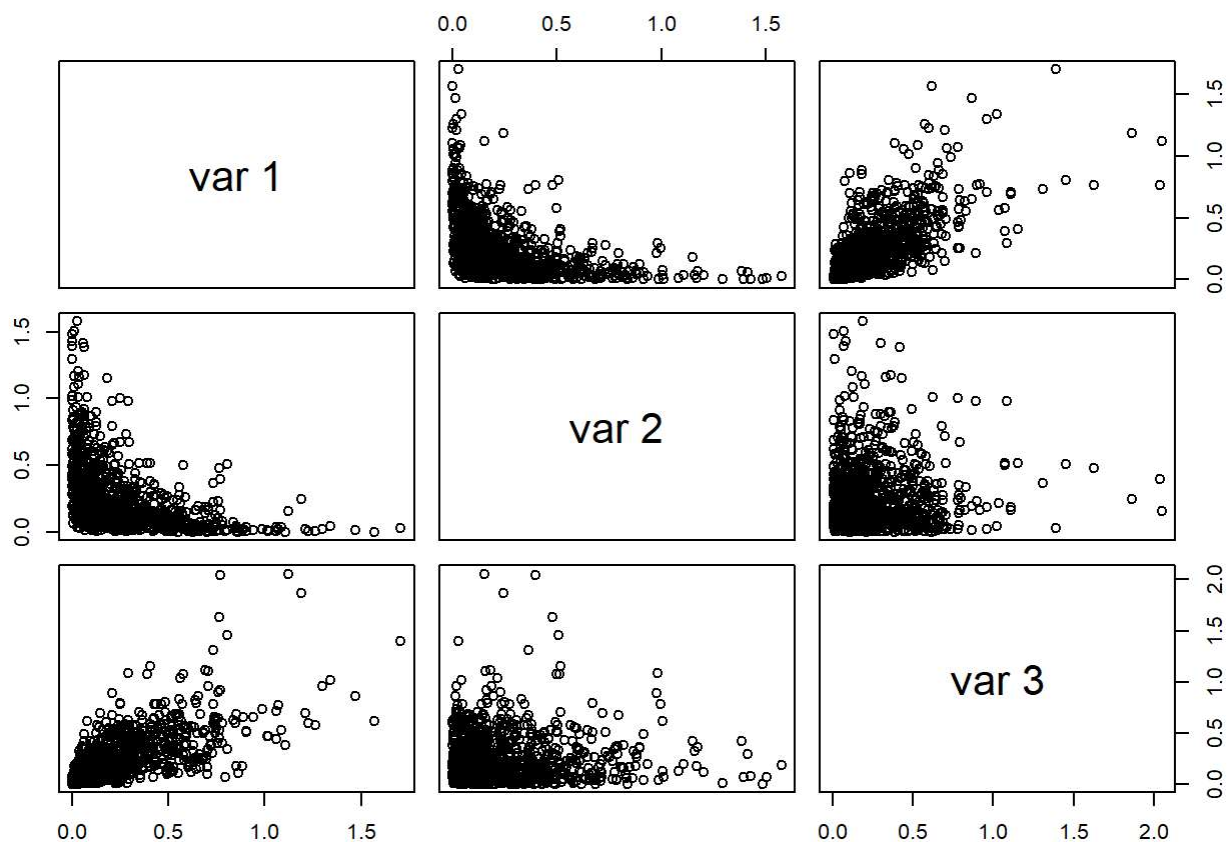
- The function `normalCopula` below defines a multivariate Normal (Gaussian) copula, with the given correlations.

- The function `mvdc` specifies a multivariate distribution by specifying its (Normal) copula and its marginal distributions.
- The function `rMvdc` generates 1000 multivariate (3D in this case) random variates from the previous.

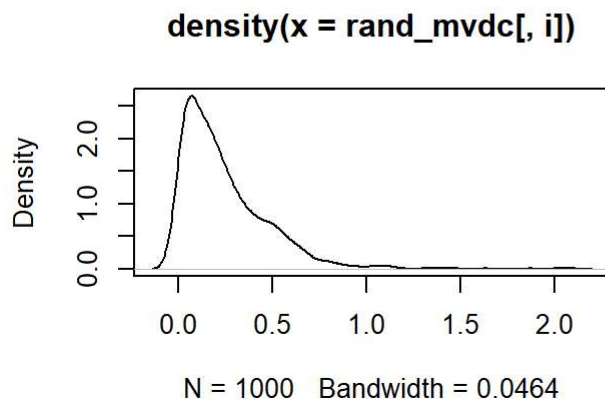
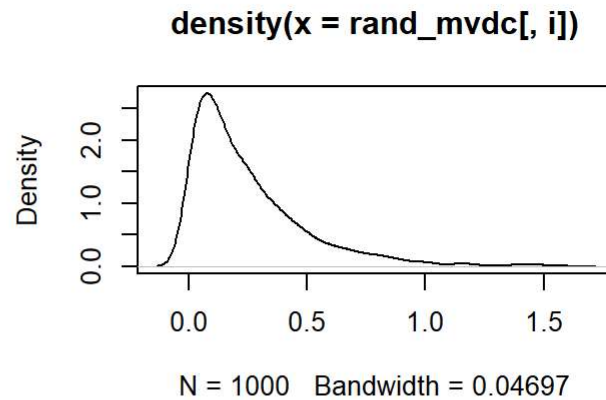
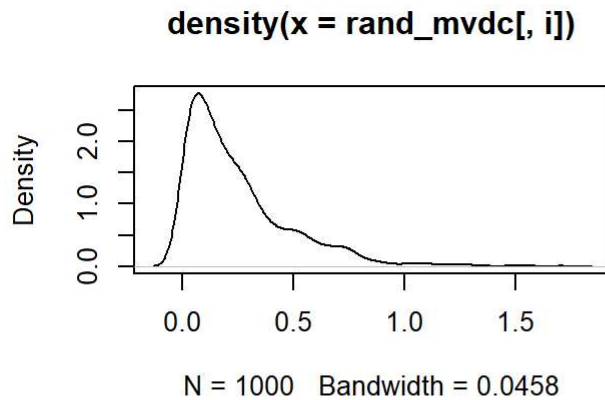
The remaining lines create a scatterplot matrix and kernel estimates of the marginal densities for each component.

```
library(copula);set.seed(5640)
cop_normal_dim3 = normalCopula(dim = 3, param = c(-0.6,0.75,0), dispstr = "un")
mvdc_normal = mvdc(copula = cop_normal_dim3, margins = rep("exp",3),
                   paramMargins = list(list(rate=4), list(rate=4),list(rate=4)))
rand_mvdc = rMvdc(n = 1000, mvdc = mvdc_normal)

pairs(rand_mvdc)
```



```
par(mfrow = c(2,2)); for(i in 1:3) plot(density(rand_mvdc[,i]))
```



## (1).Marginal Distributions and Expected Values

What are the marginal distributions of the three components in rand mvdc? What are their expected values?

The marginal distributions for all 3 components are Exponential with rate  $\lambda = 4$ , which means that their mean value is  $1/\lambda = 1/4$ .

## (2).Independence Between Components

Are the 2nd and 3rd components independent? Why or why not?

- The 2nd and 3rd components are (*marginally*) independent, since they jointly (i.e., var2 and var3 combined, without var1) follow 2D Normal/Gaussian, and their correlation is 0.
- Their correlation is given by the `copula::p2P` function, which specifies the correlation *matrix* based on the vector of distinct correlations;

for this problem we had:

```
p2P( c( -0.6, +0.75, 0 ) )
```

```
##      [,1] [,2] [,3]
## [1,]  1.00 -0.6 0.75
## [2,] -0.60  1.0 0.00
## [3,]  0.75  0.0 1.00
```

which means that

- var1 & var2 are negatively correlated,
- var1 & var 3 are positively correlated, and
- var2 & var3 are un-correlated

Note that these relations are reflected in the Exponential random variates that were generated.

Also note that the *conditional* distribution var2 and var3 *given* var1 would NOT be independent (that's why we have to specify that the "marginal" distribution of var2 & var3 is independent.)

## 5. Simulate random variates

We can simulate random variates from the Burr distribution using the inverse CDF method, which for continuous distributions is the quantile function, i.e. the same as the VaR formula.

In other words, we first generate

$$U \sim \text{Uniform}(0, 1)$$

and then calculate inverse CDF to take

$$X = F_X^{-1}(U) = \sigma(U^{-1/k} - 1)^{1/c}$$

```
set.seed(12345)
U = runif(10000)
X = ( U ^{-0.5} - 1 )
hist( log(X) ) # hist(X) does not look good b/c of heavy tails, use log scale instead
```

