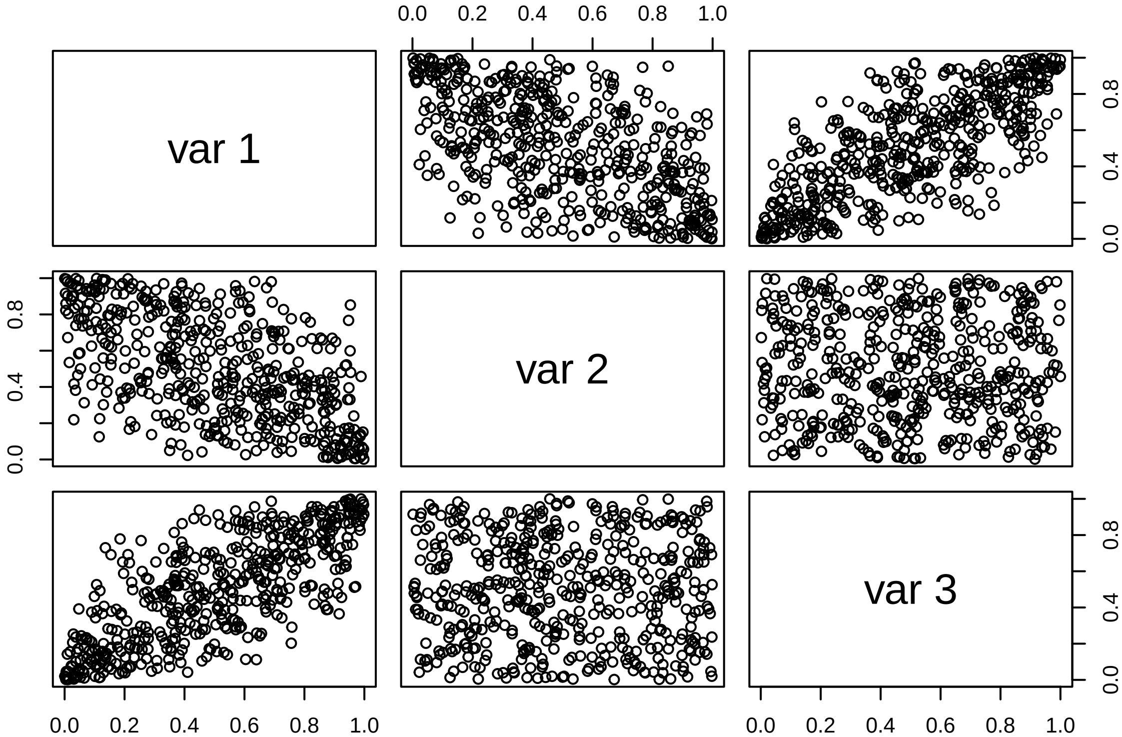
Copula-Example

Simulation from a tri-dimensional normal copula with

$$(\rho_{12}, \rho_{13}, \rho_{23}) = (-0.6, 0.75, 0)$$

```
library(copula)
cop_n_dim3 = normalCopula(dim = 3, param = c(-0.6,0.75,0), dispstr = "un")
set.seed(5640)
rand_n_cop = rCopula(n = 500, copula = cop_n_dim3)
pairs(rand_n_cop)
```




```
cor(rand_n_cop)
```

```
##            [,1]       [,2]       [,3]
## [1,]  1.0000000 -0.5750497  0.7561955
## [2,] -0.5750497  1.0000000 -0.0268701
## [3,]  0.7561955 -0.0268701  1.0000000
```

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

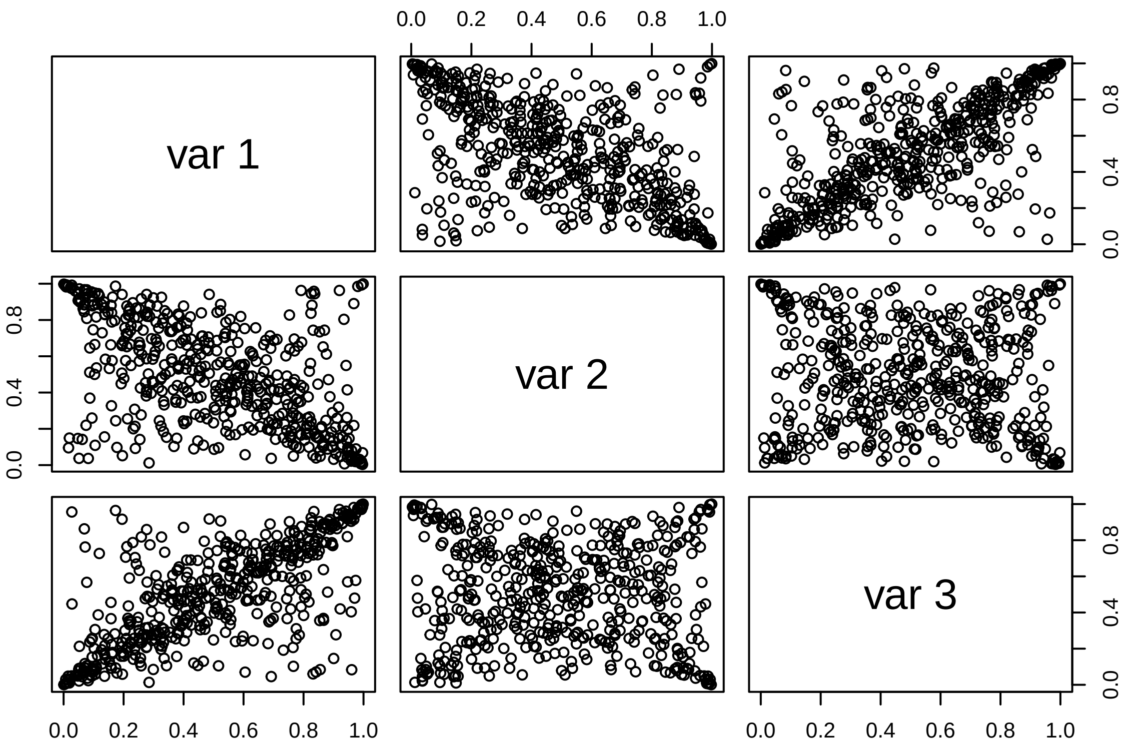
```
##
##  Pearson's product-moment correlation
##
## data:  rand_n_cop[, 1] and rand_n_cop[, 3]
## t = 25.789, df = 498, p-value < 2.2e-16
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7159826 0.7914069
## sample estimates:
##       cor
## 0.7561955
```

Simulation from a tri-dimensional t copula withwith

$$(\rho_{12}, \rho_{13}, \rho_{23}) = (-0.6, 0.75, 0)$$

```
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)
```

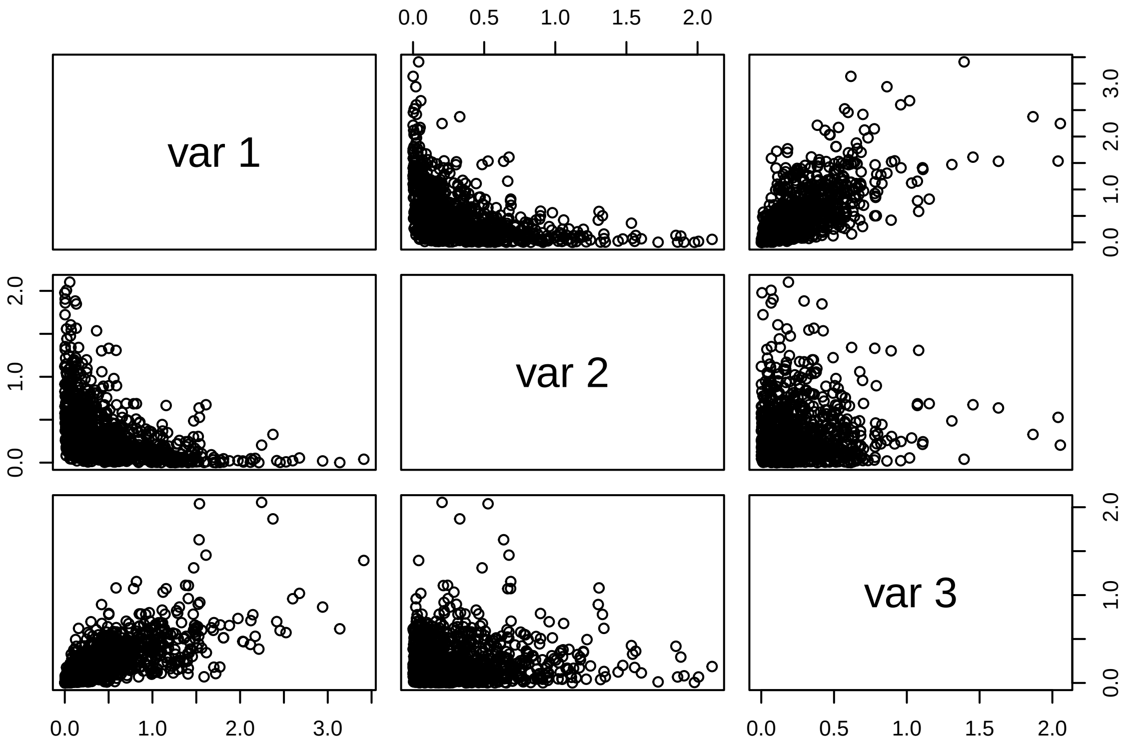
```
##            [,1]        [,2]        [,3]
## [1,]  1.0000000 -0.54999514  0.70707296
## [2,] -0.5499951  1.00000000 -0.06538499
## [3,]  0.7070730 -0.06538499  1.00000000
```

Simulation of a data set from a distribution whose copula is normal and

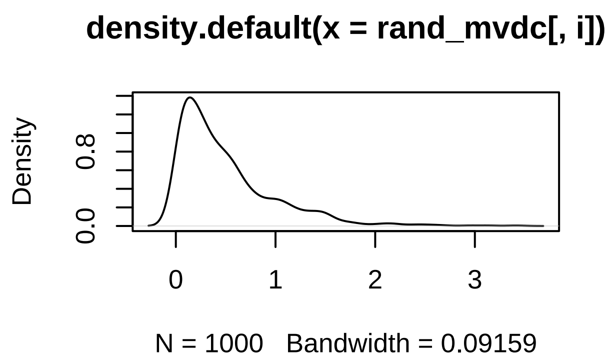
the marginals are exponentionl distribution with rate

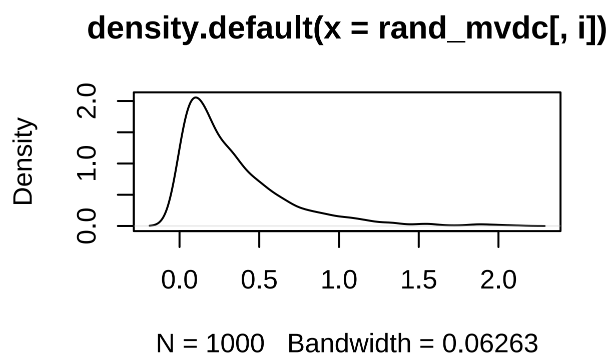
$$\lambda = 3.$$

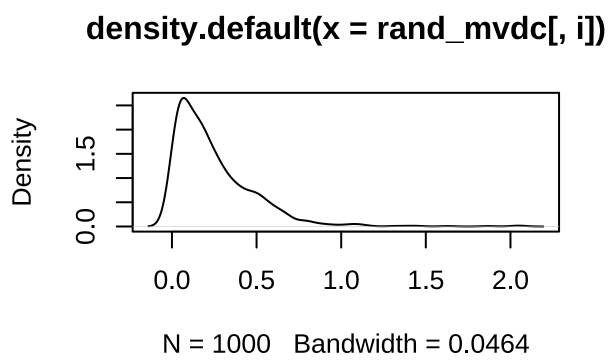
```r
cop_normal_dim3 = normalCopula(dim = 3, param = c(-0.6,0.75,0), dispstr = "un")
mvdc_normal = mvdc(copula = cop_normal_dim3, margins = rep("exp",3),
                   paramMargins = list(list(rate=2), list(rate=3),
                                       list(rate=4)))
set.seed(5640)
rand_mvdc = rMvdc(n = 1000, mvdc = mvdc_normal)
pairs(rand_mvdc)
```




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







R lab - 8.11.2 - Fitting Copula Models to Bivariate Return Data

Data download 9/9/2014

Data download 9/9/2014

```
library(quantmod)
library(xts)
getSymbols(c("IBM", "GE"), from="2021-05-01",to="2021-05-31")
```

```
## [1] "IBM" "GE"
```

```
IBM.GE = cbind(IBM[,6],GE[,6]) ; tail(IBM.GE)
```

```
##            IBM.Adjusted GE.Adjusted
## 2021-05-21     143.0926    105.6773
## 2021-05-24     143.0728    105.2780
## 2021-05-25     142.1534    104.7987
## 2021-05-26     141.7481    107.0352
## 2021-05-27     142.1831    114.6236
## 2021-05-28     142.1040    112.3071
```

```
netReturns = ((diff(IBM.GE)/lag(IBM.GE)*100)[-1,]) ; tail(netReturns)
```

```
##            IBM.Adjusted GE.Adjusted
## 2021-05-21   0.59771559   1.3016856
## 2021-05-24  -0.01382042  -0.3779202
## 2021-05-25  -0.64262236  -0.4552434
## 2021-05-26  -0.28512787   2.1341459
```

```
## 2021-05-27   0.30688032   7.0895571
## 2021-05-28  -0.05563320  -2.0209072
```

```
colnames(netReturns) = c("IBM", "GE") ; colnames(netReturns)
```

```
## [1] "IBM" "GE"
```

```
head(netReturns) ; tail(netReturns)
```

```
##                   IBM         GE
## 2021-05-04  0.6908447 -2.4535292
## 2021-05-05 -0.3636329  0.6859771
## 2021-05-06  2.2035598  0.0000000
## 2021-05-07 -0.8993057  0.6813035
## 2021-05-10  0.4881067  0.3759377
## 2021-05-11 -1.3340661 -1.9475671
```

```
##                    IBM         GE
## 2021-05-21  0.59771559  1.3016856
## 2021-05-24 -0.01382042 -0.3779202
## 2021-05-25 -0.64262236 -0.4552434
## 2021-05-26 -0.28512787  2.1341459
## 2021-05-27  0.30688032  7.0895571
## 2021-05-28 -0.05563320 -2.0209072
```

```
write.zoo(netReturns,"IBM_GE_04_14_daily_netRtns.csv", index.name="Date", sep=",")
```

```
library(MASS)     #  for fitdistr() and kde2d() functions
library(copula)   #  for copula functions
library(fGarch)   #  for standardized t density
netRtns = read.csv("IBM_GE_04_14_daily_netRtns.csv", header = T)
ibm = netRtns[,2]
ge = netRtns[,3]
est.ibm = as.numeric( fitdistr(ibm,"t")$estimate )
est.ibm[2] = est.ibm[2] * sqrt( est.ibm[3] / (est.ibm[3]-2) )
est.ge = as.numeric( fitdistr(ge,"t")$estimate )
est.ge[2] = est.ge[2] * sqrt(est.ge[3] / (est.ge[3]-2))
cor_tau = cor(ibm, ge, method = "kendall")
omega = 0.5 ######### need to get correct value
cop_t_dim2 = tCopula(omega, dim = 2, dispstr = "un", df = 4)
data1 = cbind(pstd(ibm, est.ibm[1], est.ibm[2], est.ibm[3]),
              pstd(ge, est.ge[1], est.ge[2], est.ge[3]))
n = nrow(netRtns) ; n
```

```
## [1] 19
```

```
data2 = cbind(rank(ibm)/(n+1), rank(ge)/(n+1))
ft1 = fitCopula(cop_t_dim2, data1, method="ml", start=c(omega,4) )
ft2 = fitCopula(cop_t_dim2, data2, method="ml", start=c(omega,4) )
ft1
```

```
## Call: fitCopula(copula, data = data, method = "ml", start = ..2)
## Fit based on "maximum likelihood" and 19 2-dimensional observations.
## Copula: tCopula
##    rho.1       df
##   0.3109 154.4462
## The maximized loglikelihood is 0.9442
## Convergence problems: code is 1 see ?optim.
```

```r
ft2
```

```
## Call: fitCopula(copula, data = data, method = "ml", start = ..2)
## Fit based on "maximum likelihood" and 19 2-dimensional observations.
## Copula: tCopula
## rho.1    df
## 0.366 5.147
## The maximized loglikelihood is 0.9582
## Optimization converged
```