R Notebook

Code

Hide

```
#Loading the data

library(quantmod)
getSymbols("EA",src="yahoo",from=as.Date("2018-02-06"),to=as.Date("2023-02-06"))
```

```
[1] "EA"
```

Hide

```
head(EA)
```

```
           EA.Open EA.High EA.Low EA.Close EA.Volume EA.Adjusted
2018-02-06  118.86  123.35 117.76   123.13   4652300    121.4598
2018-02-07  122.86  125.00 122.18   123.05   4066900    121.3809
2018-02-08  123.00  123.00 116.52   116.54   5478900    114.9592
2018-02-09  117.96  122.14 114.67   120.64   5945100    119.0036
2018-02-12  121.78  124.16 121.53   122.22   3695100    120.5622
2018-02-13  120.85  123.13 120.58   122.28   2388700    120.6214
```

Hide

```
tail(EA)
```

```
           EA.Open EA.High EA.Low EA.Close EA.Volume EA.Adjusted
2023-01-27  129.14  130.57 128.79   128.87   1786200    128.6496
2023-01-30  128.92  129.47 128.11   128.99   2446900    128.7694
2023-01-31  129.19  129.99 128.38   128.68   3067700    128.4599
2023-02-01  116.78  117.22 112.58   116.76  14492300    116.5603
2023-02-02  117.50  117.52 114.10   115.99   6355600    115.7916
2023-02-03  115.15  115.54 113.78   113.92   4393500    113.7252
```

Hide

```
getSymbols("ATVI",src="yahoo",from=as.Date("2018-02-06"),to=as.Date("2023-02-06"))
```

```
[1] "ATVI"
```

Hide

```
head(ATVI)
```

```
           ATVI.Open ATVI.High ATVI.Low ATVI.Close ATVI.Volume ATVI.Adjusted
2018-02-06     66.00     69.84    65.72      69.70    10524300      67.60927
2018-02-07     69.62     70.86    69.43      69.46     6255200      67.37649
2018-02-08     69.63     69.79    65.76      65.83    11179300      63.85536
2018-02-09     66.99     67.78    63.32      67.08    18582300      65.06787
2018-02-12     67.16     69.20    67.16      68.32     8315700      66.27067
2018-02-13     67.96     68.21    67.18      68.03     5373600      65.98937
```

Hide

```
tail(ATVI)
```

```
           ATVI.Open ATVI.High ATVI.Low ATVI.Close ATVI.Volume ATVI.Adjusted
2023-01-27     75.50     76.76    75.22      76.61     4382700         76.61
2023-01-30     76.63     77.08    75.84      75.96     4247400         75.96
2023-01-31     76.13     77.00    75.85      76.57     4118000         76.57
2023-02-01     76.00     76.82    75.58      76.70     4575400         76.70
2023-02-02     76.50     77.39    76.07      77.11     4696100         77.11
2023-02-03     76.64     76.78    75.03      75.24     5781000         75.24
```

Hide

```
#EA - Electronic Arts
#ATVI - Activision Blizzard

#EA Log Return Calculation
EA_log_return_1<-diff(log(EA[,6]))
EA_log_return<-as.numeric(EA_log_return_1[-1])
head(EA_log_return)
```

```
[1] -0.0006498333 -0.0543563003  0.0345762205  0.0130120086  0.0004906305
[6]  0.0121114982
```

Hide

```
#Mean and sd EA Log Return Calculation
mean_EA<-mean(EA_log_return)
mean_EA
```

```
[1] -5.234602e-05
```

Hide

```
sd_EA<- sd(EA_log_return)
sd_EA
```

```
[1] 0.01994566
```

Hide

```
#EA Log Return Calculation
ATVI_log_return_1<-diff(log(ATVI[,6]))
ATVI_log_return<-as.numeric(ATVI_log_return_1[-1])
head(ATVI_log_return)
```

```
[1] -0.003448960 -0.053675557  0.018810337  0.018316492 -0.004253687
[6]  0.023533879
```

Hide

```
#Mean and sd EA Log Return Calculation
mean_ATVI<-mean(ATVI_log_return)
mean_ATVI
```

```
[1] 8.507391e-05
```

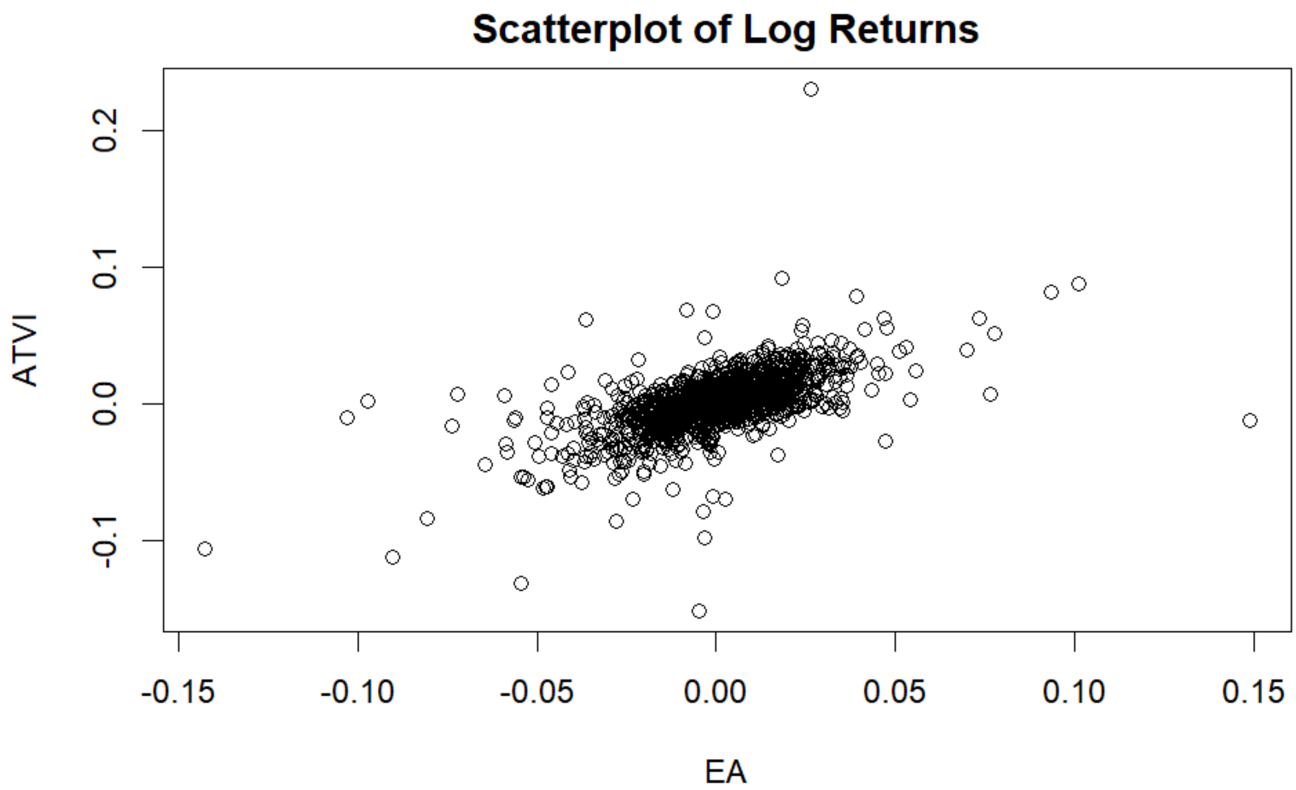
Hide

```
sd_ATVI<- sd(ATVI_log_return)
sd_ATVI
```

```
[1] 0.02164873
```

Hide

```
#1-Scatterplot of the two log returns
plot(EA_log_return, ATVI_log_return, main="Scatterplot of Log Returns", xlab="EA", ylab="ATV
I")
```



Hide

```
# After generating the scatterplot between the two sets of log returns, due to their positive
correlation, it can indicate that they have a linear relationship (many points clustered in t
he middle in an upward form). Other techniques could also be applied to better determine thei
r relationship.
```

Hide

```
#2-Maximum likelihood method to fit a bivariate t distribution. v- tail index parameter, for
v estimate vector mean and correlation matrix

library(mnormt)
library(MASS)

df <- seq(2.25, 6, 0.01)
n <- length(df)
loglik <- rep(0, n)

dat <- cbind(ATVI_log_return, EA_log_return)

for(i in 1:n) {
  fit <- cov.trob(dat, nu = df[i])
  loglik[i] <- sum(log(dmt(dat, mean = fit$center, S = fit$cov, df = df[i])))
}

aic_t <- -max(2 * loglik) + 2 * (8 + 10 + 1) + 64000
z1 <- (2 * loglik > 2 * max(loglik) - qchisq(0.95, 1))
z1
```

```
  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [73] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [85]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [97]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[109]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[121]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[133]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[145]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[157]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[169]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[181]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[193] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[205] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[217] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[229] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[241] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[253] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[265] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[277] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[289] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[301] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[313] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[325] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[337] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[349] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[361] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[373] FALSE FALSE FALSE FALSE
```

Hide

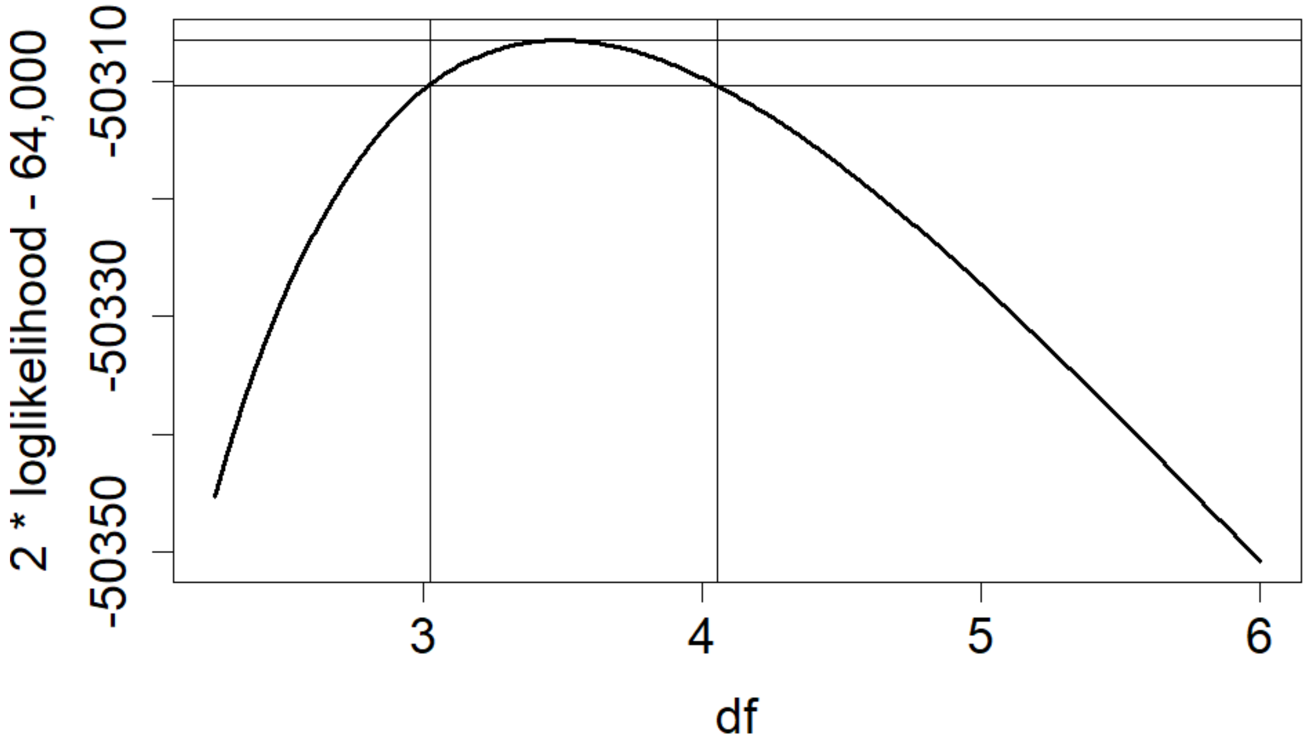
```
plot(df, 2 * loglik - 64000, type = "l", cex.axis = 1.5, cex.lab = 1.5, ylab = "2 * loglikeli
hood - 64,000", lwd = 2)
abline(h = 2 * max(loglik) - qchisq(0.95, 1) - 64000)
```

Hide

```
abline(h = 2 * max(loglik) - 64000)
abline(v=(df[78]+df[79])/2)
```

Hide

```
abline(v=(df[181]+df[182])/2)
```



Hide

```
best_index <- which.max(loglik)

# best degree of freedom
best_df <- df[best_index]
best_df
```

```
[1] 3.49
```

Hide

```
# best fit using the best df
bestfit <- cov.trob(dat,nu=best_df,cor=TRUE)
bestfit
```

```
$cov
                ATVI_log_return EA_log_return
ATVI_log_return    0.0001958947  0.0001258700
EA_log_return      0.0001258700  0.0001845843

$center
ATVI_log_return   EA_log_return
   0.0007788840    0.0006633221

$n.obs
[1] 1257

$cor
                ATVI_log_return EA_log_return
ATVI_log_return       1.0000000     0.6619324
EA_log_return         0.6619324     1.0000000

$call
cov.trob(x = dat, cor = TRUE, nu = best_df)

$iter
[1] 4
```

Hide

```
#correlation matrix
bestfit$cor
```

```
                ATVI_log_return EA_log_return
ATVI_log_return       1.0000000     0.6619324
EA_log_return         0.6619324     1.0000000
```

Hide

```
#Vector means for both series
bestfit$center
```

```
ATVI_log_return   EA_log_return
   0.0007788840    0.0006633221
```

Hide

```
#3- fitdistr  to fit univariate t distributions to the two log-return series. The means, scal
e parameters, and degrees of freedom


# Fit t-distribution to ATVI log-returns
fit_ATVI <- fitdistr(ATVI_log_return, "t")
```

```
Warning: NaNs producedWarning: NaNs producedWarning: NaNs producedWarning: NaNs producedWarni
ng: NaNs producedWarning: NaNs producedWarning: NaNs producedWarning: NaNs producedWarning: N
aNs produced
```

Hide

```
cat("ATVI log-return:\n")
```

```
ATVI log-return:
```

Hide

```
cat("Mean:", fit_ATVI$estimate[1], "\n")
```

```
Mean: 0.0007543853
```

Hide

```
cat("Scale parameter:", fit_ATVI$estimate[2], "\n")
```

```
Scale parameter: 0.01324006
```

Hide

```
cat("Degrees of freedom:", fit_ATVI$estimate[3], "\n\n")
```

```
Degrees of freedom: 3.053332
```

Hide

```
# Fit t-distribution to EA log-returns
fit_EA <- fitdistr(EA_log_return, "t")
```

```
Warning: NaNs producedWarning: NaNs producedWarning: NaNs producedWarning: NaNs producedWarni
ng: NaNs producedWarning: NaNs producedWarning: NaNs producedWarning: NaNs producedWarning: N
aNs producedWarning: NaNs producedWarning: NaNs produced
```

Hide

```
cat("EA log-return:\n")
```

```
EA log-return:
```

Hide

```
cat("Mean:", fit_EA$estimate[1], "\n")
```

```
Mean: 0.0004849973
```

Hide

```
cat("Scale parameter:", fit_EA$estimate[2], "\n")
```

```
Scale parameter: 0.01406657
```

Hide

```
cat("Degrees of freedom:", fit_EA$estimate[3], "\n")
```

```
Degrees of freedom: 4.018489
```

Hide

```
# Now convert estimated scale parameters to estimated standard deviations

cat("Standard deviation:", fit_ATVI$estimate[2] * sqrt((fit_ATVI$estimate[3] )/(fit_ATVI$esti
mate[3]-2)), "\n")
```

```
Standard deviation: 0.0225421
```

Hide

```
cat("Standard deviation:", fit_EA$estimate[2] * sqrt((fit_EA$estimate[3])/(fit_EA$estimate[3]
-2)), "\n")
```

```
Standard deviation: 0.01984752
```

Hide

```
#4- calculate the sample Kendall’s Tau rank correlation ˆρτ and the sample Pearson correlatio
n ˆρ for the pair of log-return series. Then compare ˆρ with the estimate sin(π/2 × ρˆτ ) and
discuss briefly

#Kendall
ρτ<-cor(ATVI_log_return, EA_log_return, method = "kendall") # kendall
ρτ
```

```
[1] 0.4676636
```

Hide

```
#Pearson
ρ<-cor(ATVI_log_return, EA_log_return, method = "pearson") #pearson
ρ
```

```
[1] 0.5981101
```

Hide

```
# Pearson estimation based on Kendall
omega<-sin(ρτ*pi/2)
omega
```

```
[1] 0.6702935
```

Hide

```
# The estimate sin(π/2 × ρˆτ) is greater than the sample Pearson correlation ˆρ, suggesting t
hat the relationship between ATVI and EA log returns may not be linear. This is consistent wi
th our earlier observations that the log returns do not follow a normal distribution and may
have a non-linear relationship even though they have a strong and positive correlation, but t
hey may have some normality between them
```

Hide

```
#5-t-Copula (connection between two series) generation
library(copula)
library(fGarch)

# t-copula values
cop_t_dim2<-tCopula(omega, dim = 2, dispstr = "un", df=best_df)
cop_t_dim2
```

```
t-copula, dim. d = 2
Dimension:  2
Parameters:
  rho.1   = 0.6702935
  df      = 3.4900000
```

Hide

```
# ATVI data percentiles
ATVI_data<-pstd(ATVI_log_return,fit_ATVI$estimate[1], fit_ATVI$estimate[2] * sqrt((fit_ATVI$e
stimate[3] )/(fit_ATVI$estimate[3]-2)), fit_ATVI$estimate[3])

# EA data percentiles
EA_data<-pstd(EA_log_return,fit_EA$estimate[1], fit_EA$estimate[2] * sqrt((fit_EA$estimate[3]
)/(fit_EA$estimate[3]-2)), fit_EA$estimate[3])

#fit the copulas to the uniform -transformed data

data1<- cbind(ATVI_data, EA_data)

#n=nrow(dat);n
#data2<- cbind(rank(ATVI_log_return)/(n+1),rank(EA_log_return)/(n+1))

# fit the t-copula on data1
fit1<- fitCopula(cop_t_dim2, data1, method="ml", start=c(omega, best_df) )

# Estimated values for the correlation and for the tail-index parameter, along with the stand
ard errors of these estimates
summary(fit1)
```

```
Call: fitCopula(cop_t_dim2, data = data1, ... = pairlist(method = "ml", start = c(omega,
    best_df)))
Fit based on "maximum likelihood" and 1257 2-dimensional observations.
t-copula, dim. d = 2
      Estimate Std. Error
rho.1   0.6687      0.016
df      4.5858      0.703
The maximized loglikelihood is 384
Optimization converged
Number of loglikelihood evaluations:
function gradient
      19        6
```

Hide

```
# from the summary we can conclude that the bivariate t-copula has successfully been fit to t
he data, capturing a positive dependence between the two assets and accounting for tail depen
dence. To check how good this estimation it is important to also look at other copulas summar
ies.
```

Hide

```
#6-  fit a normal (Gaussian) copula, a Clayton copula, and a Joe Copula to your data using ma
ximum likelihood

#Gaussian copula
fnorm<- fitCopula(copula = normalCopula(dim=2), data=data1, method="ml")
summary(fnorm)
```

```
Call: fitCopula(normalCopula(dim = 2), data = data1, ... = pairlist(method = "ml"))
Fit based on "maximum likelihood" and 1257 2-dimensional observations.
Normal copula, dim. d = 2
      Estimate Std. Error
rho.1   0.6497      0.014
The maximized loglikelihood is 344
Optimization converged
Number of loglikelihood evaluations:
function gradient
       8        8
```

Hide

```
#Clayton copula
fclayton<- fitCopula(copula = claytonCopula(1,dim=2), data=data1, method="ml")
summary(fclayton)
```

```
Call: fitCopula(claytonCopula(1, dim = 2), data = data1, ... = pairlist(method = "ml"))
Fit based on "maximum likelihood" and 1257 2-dimensional observations.
Clayton copula, dim. d = 2
      Estimate Std. Error
alpha    1.757      0.072
The maximized loglikelihood is 233.4
Optimization converged
Number of loglikelihood evaluations:
function gradient
       3        3
```

Hide

```
#Joe copula
fjoe<- fitCopula(copula = joeCopula(2,dim=2), data=data1, method="ml")
summary(fjoe)
```

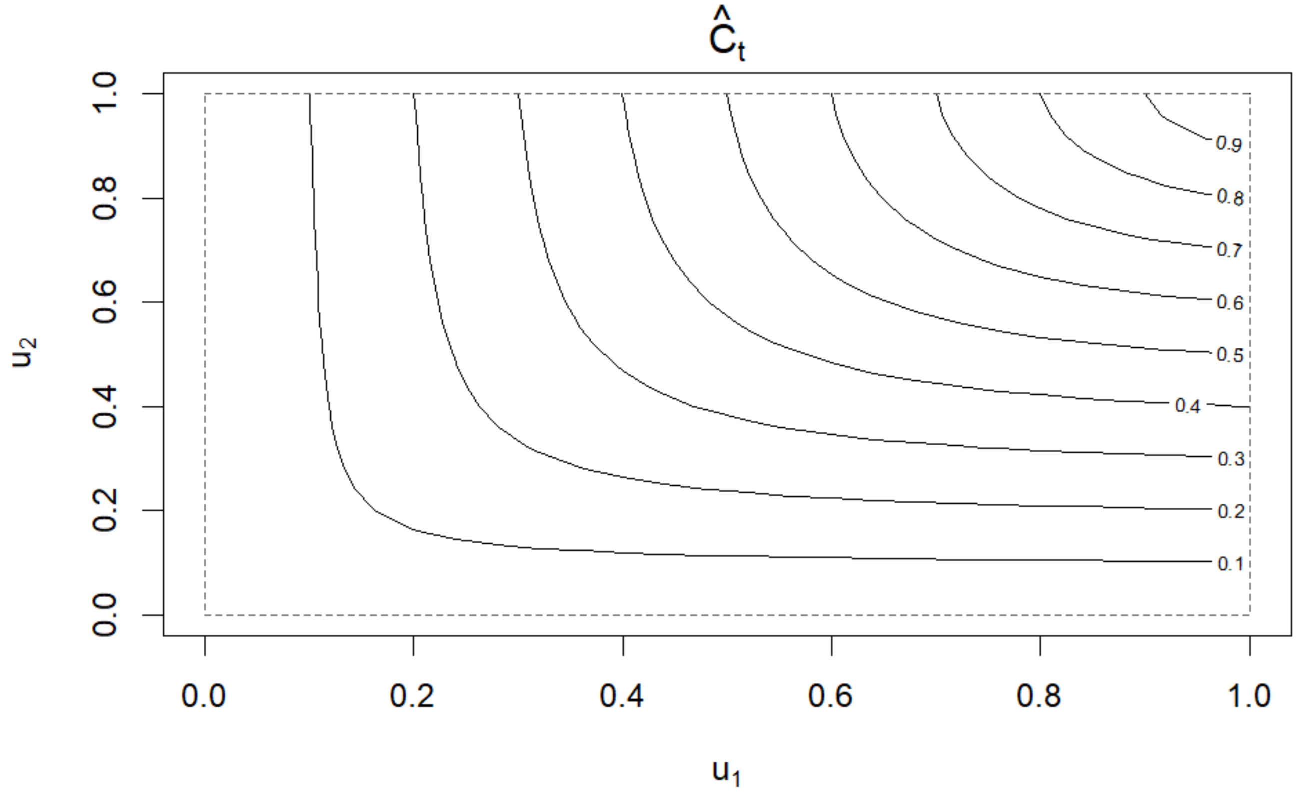
```
Call: fitCopula(joeCopula(2, dim = 2), data = data1, ... = pairlist(method = "ml"))
Fit based on "maximum likelihood" and 1257 2-dimensional observations.
Joe copula, dim. d = 2
      Estimate Std. Error
alpha     2.09       0.06
The maximized loglikelihood is 276.8
Optimization converged
Number of loglikelihood evaluations:
function gradient
       7        7
```

Hide

```
#7- Use contour to create diagrams of the four fitted copulas (t, Normal, Clayton, and Joe)
```

Hide

```
#tCopula
contour(tCopula(param=0.6702994, dim=2, df=round(3.49)), pCopula,main=expression(hat(C)[t]))
```



Hide

```
#NormalCopula
contour(normalCopula(param=0.6497, dim=2), pCopula, main=expression(hat(C)[Gauss]))
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

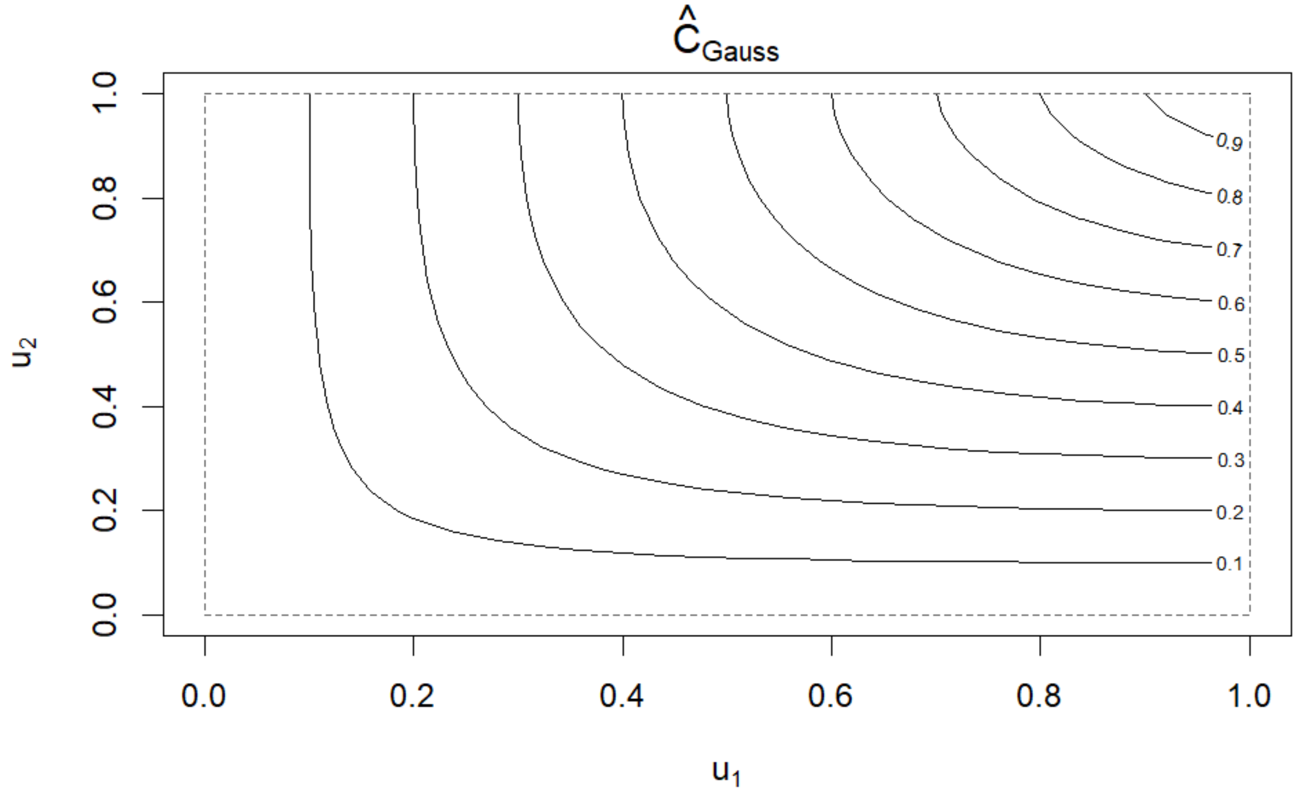
```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```

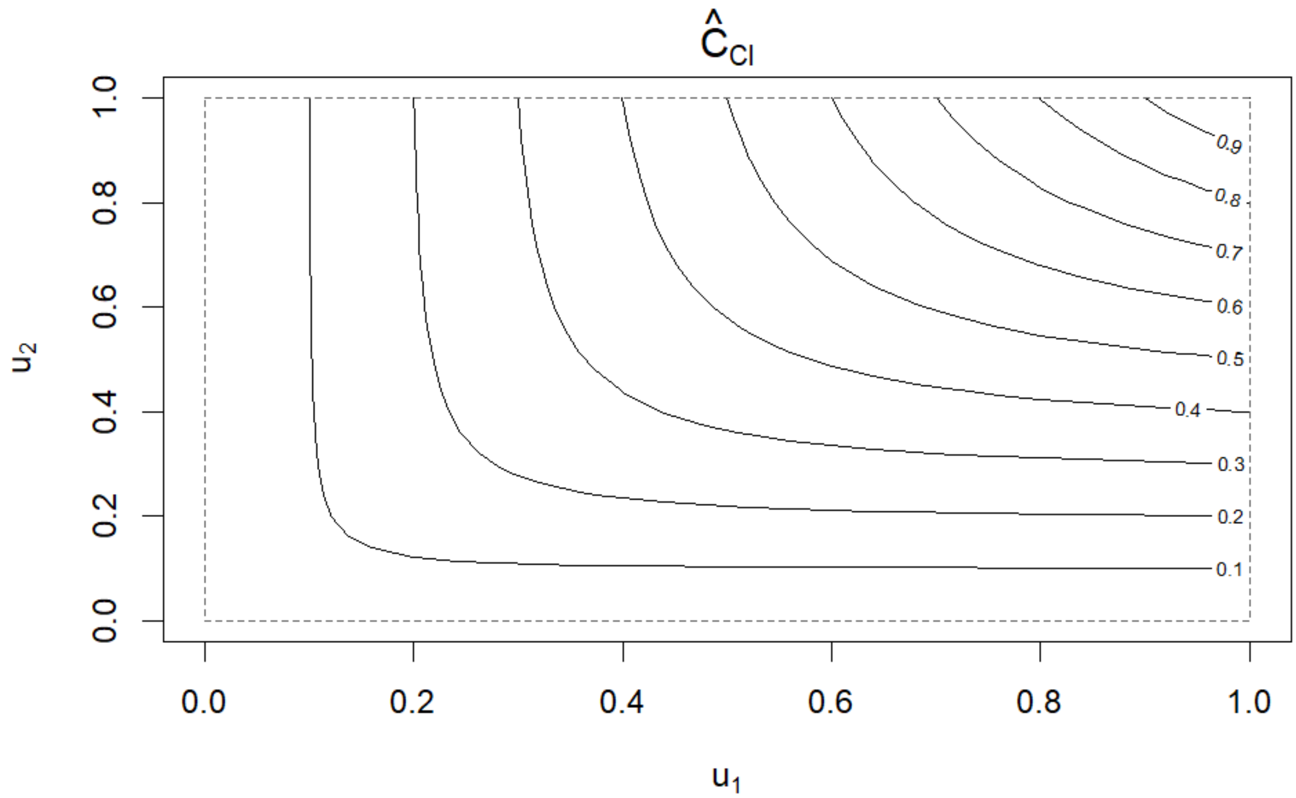
```
Warning: Approximating +/-Inf by +/-1000
```

```
Warning: Approximating +/-Inf by +/-1000
```



Hide

```
#claytonCopula
contour(claytonCopula(param=1.757, dim=2), pCopula,main=expression(hat(C)[Cl]))
```



Hide

```
#joeCopula
contour(joeCopula(param=2.09, dim=2), pCopula,main=expression(hat(C)[Joe]))
```