Problem Background

Fitting Copula Models

In this lab we are going to fit copula models to a bivariate data set of daily returns on IBM and S&P500 Index.

First, we need to fit a model with the univariate marginal t-distributions and a t-copula.

This model will have three degrees of freedom parameters:

- IBM tail index
- S&P 500 Index tail index
- Joint tail index (the copula)

```
net.returns <- read.csv(paste0(data.dir, "IBM_SP500_04_14_daily_netRtns.csv"),
                        header = T)

ibm <- net.returns[, "IBM"]
sp500 <- net.returns[, "SP500"]

suppressWarnings({
  est.ibm = as.numeric( fitdistr( ibm, "t" )$estimate )
  est.sp500 = as.numeric( fitdistr( sp500, "t" )$estimate )
})

est.ibm[2] = est.ibm[2] * sqrt( est.ibm[3] / (est.ibm[3]-2) )
est.sp500[2] = est.sp500[2] * sqrt(est.sp500[3] / (est.sp500[3]-2) )
```

The univariate estimates will be used as starting values when we estimate the *meta-t* distribution is fit by maximum liklihood. Before we do that, we need to compute an estimate of the correlation coefficent in the *t*-copula.

Problem 1

Using Kendall's tau, compute omega, which is the estimate of the Pearson correlation from Kendall's tau.

From **8.27** we have Kendall's tau, ρ_τ, =

$\rho_\tau(Y_i, Y_j) = \frac{2}{\pi} arcsin(\Omega_{i,j})$.

Inverting, we derive that:

$\Omega_{i,j} = sin[\frac{\pi}{2}\rho_\tau(Y_i, Y_j)]$

```
cor_tau = cor(ibm, sp500, method = "kendall")
omega = sin((pi/2) * cor_tau)
```

$\Omega = 0.701835$

The *t*-copula using omega as the correlation parameter and 4 as the degrees of freedom:

```
cop_t_dim2 <- tCopula(omega, dim = 2, dispstr = "un", df = 4)
```

```
t-copula, dim. d = 2
Dimension:  2
Parameters:
  rho.1    = 0.7018346
  df       = 4.0000000
```

Now fit copulas to the uniformed-transformed data:

```
n = nrow(net.returns)

data1 = cbind( pstd( ibm, mean=est.ibm[1], sd=est.ibm[2], nu=est.ibm[3] ),
               pstd( sp500, mean=est.sp500[1], sd=est.sp500[2], nu=est.sp500[3] ) )

data2 = cbind( rank(ibm)/(n+1), rank(sp500)/(n+1) )

ft1 = fitCopula(cop_t_dim2, data=data1, method="ml", start = c(omega,4))
ft2 = fitCopula(cop_t_dim2, data=data2, method="ml", start = c(omega,4))
```

Problem 2

Explain the difference between methods used to obtain the two estimates *ft1* and *ft2*.

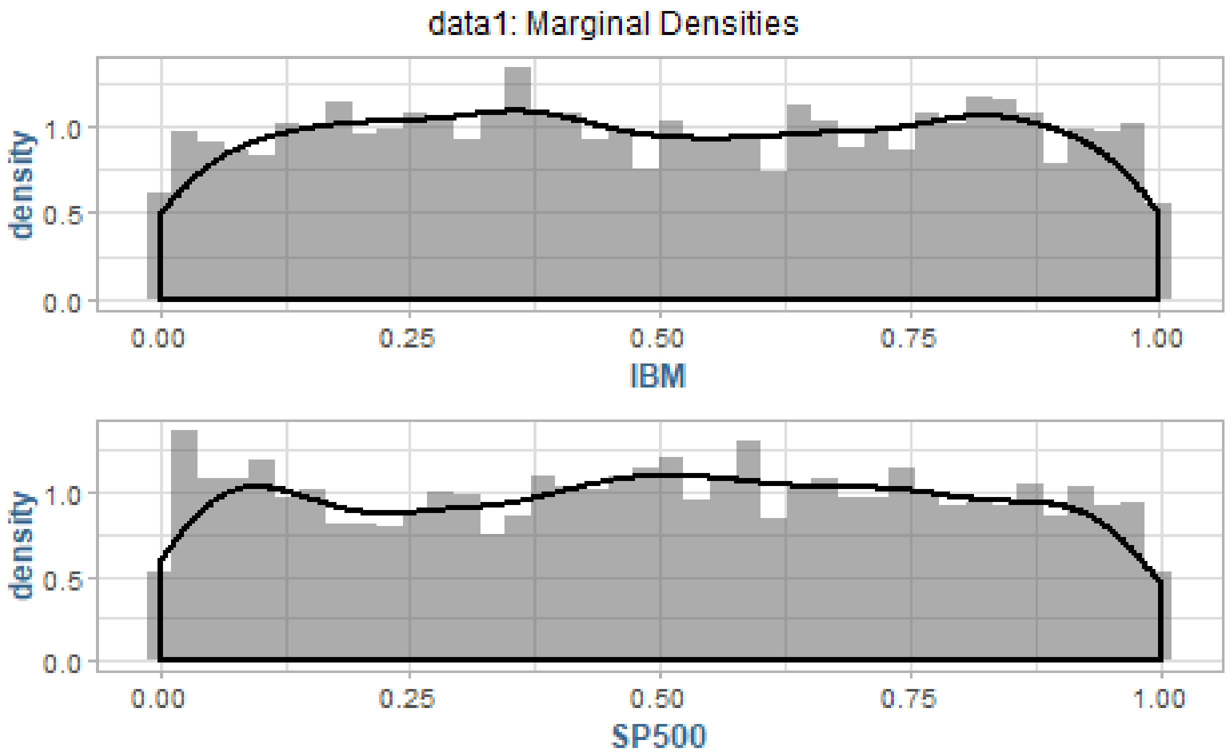
Here we are fitting the copulas to the uniform-transformed data.

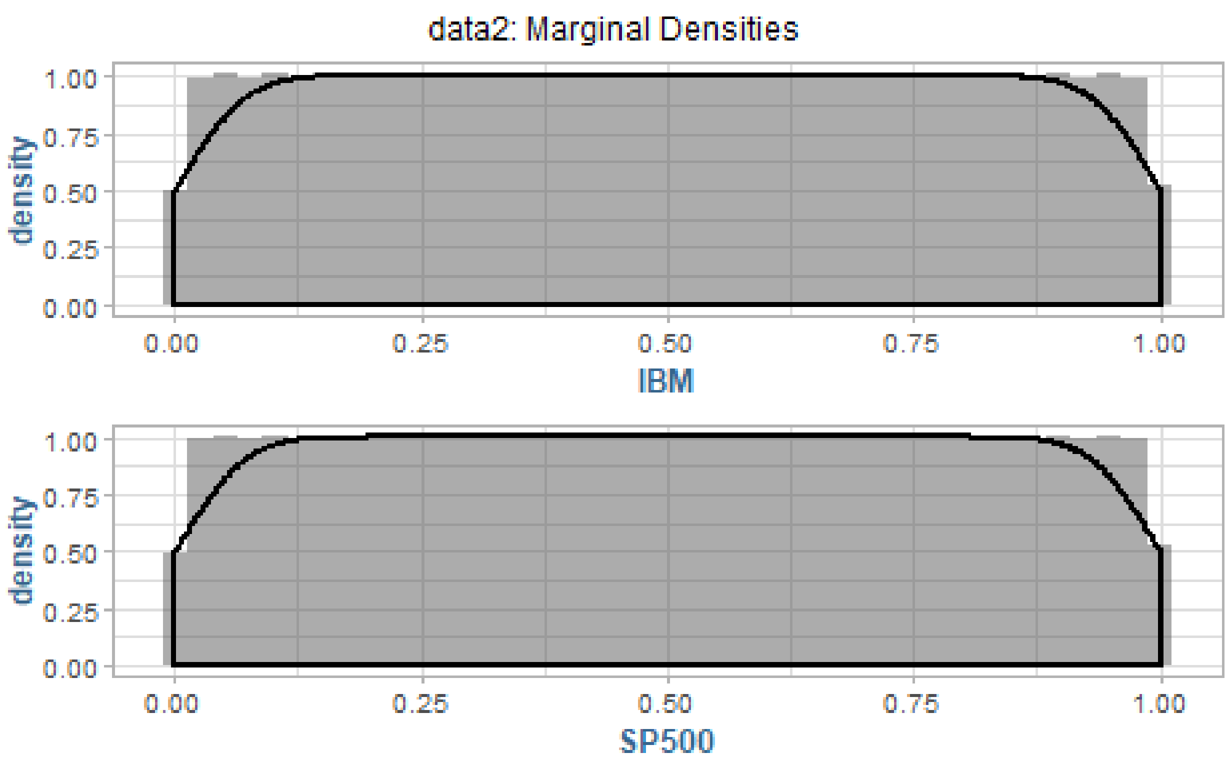
data1 is computed using the probability density function for the standard t-distrubution, since we previously estimated the paramters to fit a t-distribution to the respective returns using the **fitdistr** function above.

data2 is computed using the percentile rank method.

Both of these methods are calculating the cumulative density of the marginal values in IBM and SP500, respectively.

Below we will have a quick visual inspection of the densities of the two methods.


data1: Marginal Densities
density
IBM
density
SP500
0.00
0.25
0.50
0.75
1.00
0.0
0.5
1.0


data2: Marginal Densities
density
IBM
density
SP500
0.00
0.25
0.50
0.75
1.00

Do the two estimates seem significantly different (*in a practical sense*)?.

```r
summary(ft1)
```

```
Call: fitCopula(copula, data = data, method = "ml", start = ..2)
Fit based on "maximum likelihood" and 2516 2-dimensional observations.
t-copula, dim. d = 2
      Estimate Std. Error
rho.1   0.7022      0.012
df      2.9834      0.269
The maximized loglikelihood is 967.2
Optimization converged
Number of loglikelihood evaluations:
function gradient
      40        9
```

```r
summary(ft2)
```

```
Call: fitCopula(copula, data = data, method = "ml", start = ..2)
Fit based on "maximum likelihood" and 2516 2-dimensional observations.
t-copula, dim. d = 2
      Estimate Std. Error
rho.1   0.7031      0.012
df      3.0222      0.278
The maximized loglikelihood is 964.6
Optimization converged
Number of loglikelihood evaluations:
function gradient
      38        9
```

These two estimates are fairly close to each other for practical purposes.

However, I think method one would be a more robust estimte due to the estimates coming from the fitted t-distribution.

Problem 3

Next, we will define a meta-*t*-distribution by specifying its *t*-copula and its univariate marginal distributions.

```
mvdc_t_t = mvdc( cop_t_dim2, c("std","std"), list(
           list(mean=est.ibm[1],sd=est.ibm[2],nu=est.ibm[3]),
           list(mean=est.sp500[1],sd=est.sp500[2],nu=est.sp500[3])))
```

```
mvdc_t_t
```

```
Multivariate Distribution Copula based ("mvdc")
 @ copula:
t-copula, dim. d = 2
Dimension:  2
Parameters:
  rho.1   = 0.7018346
  df      = 4.0000000
 @ margins:
[1] "std" "std"
   with 2 (not identical)  margins; with parameters (@ paramMargins)
List of 2
 $ :List of 3
  ..$ mean: num 0.05015879
  ..$ sd  : num 1.42823
  ..$ nu  : num 3.254383
 $ :List of 3
  ..$ mean: num 0.07918415
  ..$ sd  : num 1.968172
  ..$ nu  : num 2.249776
```

Now we fit the meta *t*-distribution.

```
start = c(est.ibm, est.sp500, ft1@estimate)
objFn = function(param) -loglikMvdc( param, cbind(ibm,sp500), mvdc_t_t)
tic = proc.time()
ft = optim(start, objFn, method="L-BFGS-B",
           lower = c(-.1,0.001,2.2, -0.1,0.001,2.2,  0.2,2.5),
           upper = c( .1,   10, 15,  0.1,   10, 15,  0.9, 15) )
toc = proc.time()
total_time = toc - tic ; total_time[3]/60
```

```
elapsed
 0.4285
```

What are the estimates of the copula parameters?

For $C_{t(\Omega,v)}$

- $\Omega = 0.704216$
- $\nu = 2.969349$

What are the estimates of the parameters in the univariate marginal distributions?

Table 1: Marginal t-distribution estimates

Symbol	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\nu}$
IBM	0.065047	1.379828	3.357926
SP500	0.074221	1.807512	2.334159

Was the estimation method maximum likelihood, semiparametric pseudo-maximum likelihood, or parametric pseudo-maximum likelihood?

Here, since we are estimating the parameters of the marginal distributions with the parameters of the t-copula at the same time, this is a maximum likelihood estimate.

Estimate the coefficent of lower tail dependence for this copula.

From **8.21** we have:

$\lambda_\ell = 2F_{t,\nu+1}\left\{ -\sqrt{\frac{(\nu+1)(1-\rho)}{1+\rho}} \right\}$

Where,

- $\nu = 0.704216$
- $\rho = 2.969349$

```
threshold <- .01

# Method 1: book formula

x <- -sqrt( ( nu+1 ) * ( 1-rho ) / ( 1+rho ) )
lambda <- 2 * pt( x, nu + 1)

# Method 2: package function
lambda2 <- fitLambda(data1, method = c("t"), p = min( 100 / nrow(data1), 0.1),
           lower.tail = TRUE, verbose = F)
```

```r
# assert similarity
stopifnot((lambda - lambda2$Lambda[1,2]) < threshold)
```

$\lambda_\ell = 0.453534$