Assignment #4 Fitting Copulas to Bivariate Return Data
Scott Morgan

Solve textbook Problems 4-6 at pages 210-212 (Ruppert). Each part of the assignment is worth 5 points, except Problem 6-d which is worth 10 points.

```
library(MASS) # for fitdistr() and kde2d() functions
library(copula) # for copula functions 3
library(fGarch) # for standardized t density
netRtns =
read.csv("C:/Users/Scott/Desktop/451/Module4/IBM_SP500_04_14_daily_netRtns.csv",
header = T)
ibm = netRtns[,2]
sp500 = netRtns[,3]

#univariate estimates
est.ibm = as.numeric(fitdistr(ibm,"t")$estimate) #mu, StdDev,DF
est.sp500 = as.numeric( fitdistr(sp500,"t")$estimate) #mu, StdDev,DF
est.ibm[2] = est.ibm[2] * sqrt( est.ibm[3] / (est.ibm[3]-2) ) #Scale matrix of est.ibm
//equation 7.17 // Equals Sigma
est.sp500[2] = est.sp500[2] * sqrt(est.sp500[3] / (est.sp500[3]-2)) #Equals Sigma
```

Problem 4

How did you complete line 12 of the code? What was the computed value of* omega*? (5 pts)

Per page 201 of Ruppert & Matteson (2015), equation 8.27 tells us:

$$\rho_T(Y_i, Y_j) = \frac{2}{\pi}\arcsin(\boldsymbol{\Omega}_{ij})$$

By moving the equation around algebraically we are able to solve for Omega or the estimate of the Pearson's correlation given Kendall's Tau.

$(\boldsymbol{\Omega}_{ij}) = \sin(\frac{\pi}{2})\rho_T(Y_i, Y_j)$ where $\rho_T(Y_i, Y_j) = Kendall\ Tau's\ correlation$ (page 201)

Below is the code used to compute line 12:

```
cor_tau = cor(ibm, sp500, method = "kendall")
omega = sin((pi/2)*cor_tau)
omega
#[1] 0.7018346
```

Problem 5

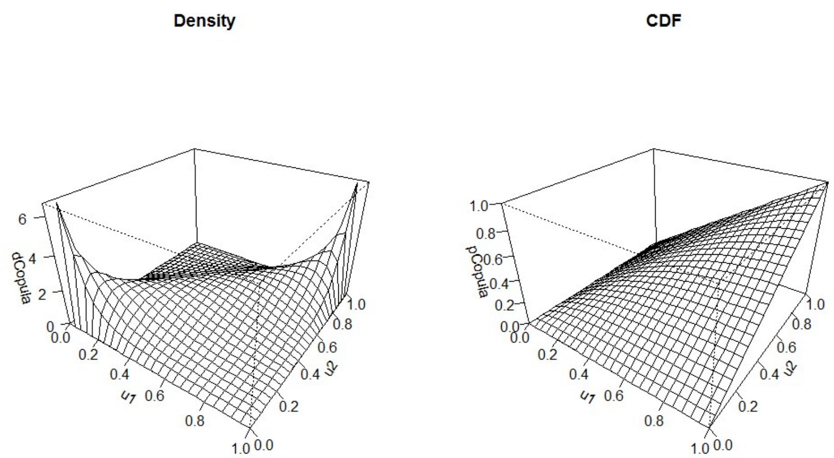
```
#Define t-copula using omega
cop_t_dim2 = tCopula(omega, dim = 2, dispstr = "un", df = 4)


#Parametric
data1 = cbind(pstd(ibm, est.ibm[1], est.ibm[2], est.ibm[3]), #parametric
              pstd(sp500, est.sp500[1], est.sp500[2], est.sp500[3]))

#Empirical Distribution
n = nrow(netRtns) ; n
data2 = cbind(rank(ibm)/(n+1), rank(sp500)/(n+1)) #Equation 8.26 page 200

#Fit copulas to the uniform-transformed data
ft1 = fitCopula(cop_t_dim2, data1, method="ml", start=c(omega,4))
ft2 = fitCopula(cop_t_dim2, data2, method="ml", start=c(omega,4)) #(8.5.2 Marg.ranks)
```

(a) Explain the difference between methods used to obtain the two estimates ft1 and ft2. (5 pts)

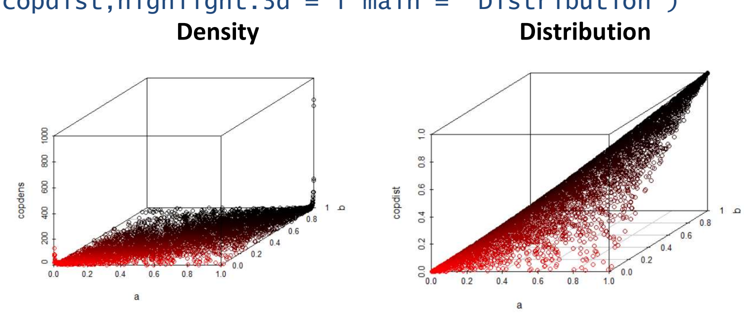
ft1 is a parametric approach using t-distributions while ***ft2*** is a non-parametric approach estimated by CDFs. Both fits are by pseudo-maximum likelihood. The non-parametric approach, in comparison to the parametric approach used for ***ft1***, assumes no models for the marginals but rather empirical distribution functions are employed to retrieve the pseudo-uniform variables (Pfaff, 2016).

The copula function can also be visualized. The below displays the density of the bivariate t-copula that is estimated. The t-copula adds tail correlation to the normal copula but maintains overall correlation by adding some negative correlation in the middle of the distribution. Following this are 3D graphs of the density and distribution functions where I use a random sample of 5000. The distributions are relatively the same between the two extra visuals.

```
#Additional Visuals Part 1
par(mfrow=c(1,2))
persp(cop_t_dim2, dCopula, main="Density", xlab="u1", ylab="u2", theta=35)
persp(cop_t_dim2, pCopula, main="CDF",     xlab="u1", ylab="u2", theta=35)
```




```
#Additional Visuals Part 2

library(scatterplot3d)
mycop=cop_t_dim2

u<-rCopula(5000,mycop)
a<-u[,1]
b<-u[,2]
copdens<-dCopula(u,mycop)
copdist<-pCopula(u,mycop)
scatterplot3d(a,b,copdens,highlight.3d = T, main = 'Density')
scatterplot3d(a,b,copdist,highlight.3d = T main = 'Distribution')
```



(b) Do the two estimates seem significantly different (in a practical sense)? (5 pts)

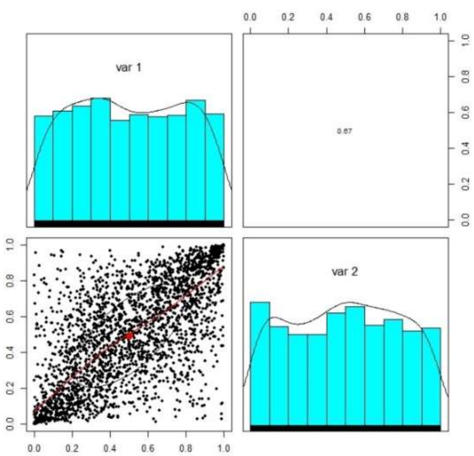
The two estimates do not vary significantly in a practical sense. The correlation estimates are 0.7022 and 0.7031; the degrees of freedom estimates are 2.9834 and 3.0222 for ***ft1*** and ***ft2***, respectively. The correlation estimates are also similar to Kendall's Tau, which was 0.7018.

```
#Mean estimates of the marginals. Correlation and degrees of freedom
ft1@estimate
#[1] 0.7021719 2.9834079
ft2@estimate
#[1] 0.7031347 3.0221711
```

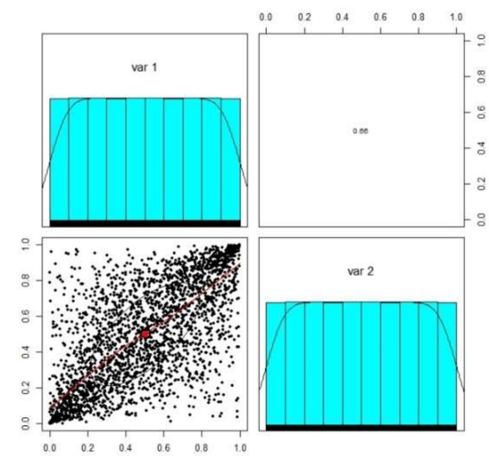
The additional charts below show the two sets of marginal uniform distributions and variable correlation to graphically illustrate their similarities.

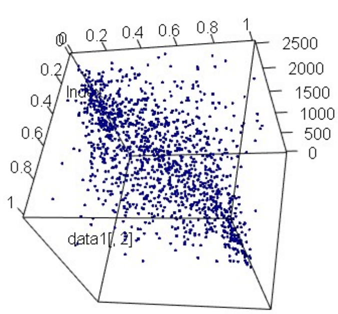
```
library(psych)
pairs.panels(data1)
pairs.panels(data2)

library(rgl)
plot3d(data1[,1],data1[,2],pch=20,col='navyblue')
plot3d(data2[,1],data2[,2],pch=20,col='navyblue')
```

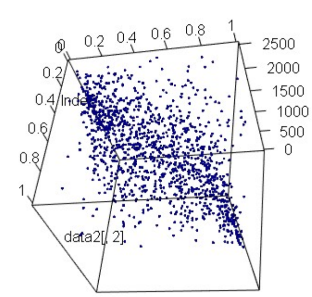
data1



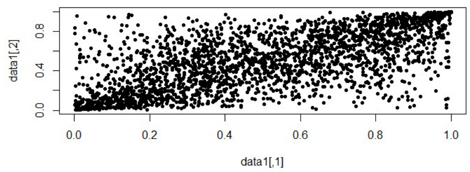
data2

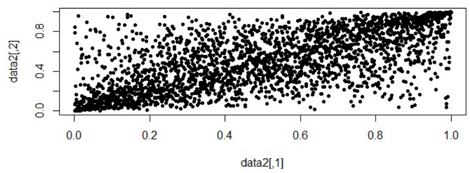






```
#Alternatively:
par(mfrow=c(2,1))
plot(data1, pch=20)
plot(data2, pch=20)
par(mfrow=c(1,1))
```





Problem 6

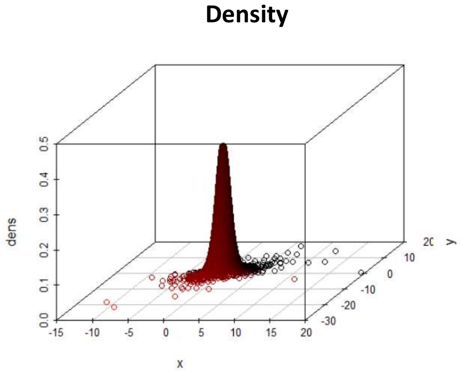
```
#Define meta-t distribution #8.7.3 Calibrating Meta-Gaussian and Meta-t-Distributions
mvdc_t_t = mvdc( cop_t_dim2, c("std","std"), list(
 list(mean=est.ibm[1],sd=est.ibm[2],nu=est.ibm[3]),
 list(mean=est.sp500[1],sd=est.sp500[2],nu=est.sp500[3])))


#Fitting meta t-distribution.
# L-BFGS-B allows box constr. (each variable can be given a lower and/or upper bound)
start = c(est.ibm, est.sp500, ft1@estimate)
objFn = function(param) -loglikMvdc(param,cbind(ibm,sp500),mvdc_t_t)
tic = proc.time()
ft_cop = 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
```

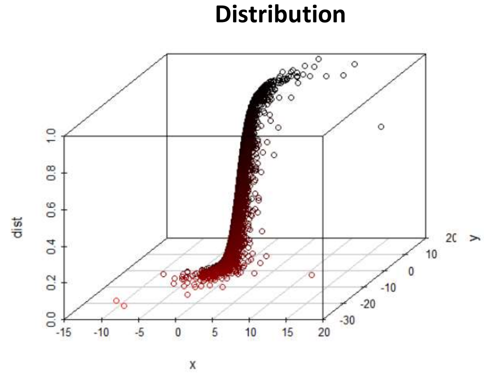
(a) What are the estimates of the copula parameters in fit cop? (5 pts)

The estimates of the copula parameters in the fit copula (***ft_cop***) are 0.7042 (rho) and 2.969 (nu). Also, we can visualize the density and cumulative distributions from the multivariate distribution. This is done by generating 5000 random numbers from ***mvdc_t_t.***

```
rho = ft_cop$par[7]
nu = ft_cop$par[8]

rho
#[1] 0.7042161

nu
#[1] 2.96935

par(mfrow=c(1,2))
library(scatterplot3d)
mymvd=mvdc_t_t
r<-rMvdc(5000,mvdc_t_t)
dens<-dMvdc(r,mymvd)
dist<-pMvdc(r,mymvd)
x<-r[,1]
y<-r[,2]
scatterplot3d(x,y,dens,highlight.3d = T, main = 'Density')
```

```
scatterplot3d(x,y,dist,highlight.3d = T,main = 'Distribution')
```





(b) What are the estimates of the parameters in the univariate marginal distributions? (5 pts)

For the univariate marginal distributions, the correlation estimates are 1.4282 and 1.9682; the degrees of freedom estimates are 3.2544 and 2.2498 for IBM and the S&P 500, respectively.

```
est.ibm[2]#Sigma
#1.42822953

est.ibm[3]#Correlation
#3.254383

est.sp500[2]#Sigma
# 1.968172

est.sp500[3]#Correlation
# 2.249776

#Another way of listing parameters.
start
#[1] 0.05015879 1.42822953 3.25438269 0.07918415 1.96817162 2.24977588 0.70217192
#[8] 2.98340790
```

(c) Was the estimation method maximum likelihood, semiparametric pseudo maximum likelihood, or parametric pseudo-maximum likelihood? (5 pts)

The estimation method for the entire fit (***ft_cop)***, which are the copula parameters and that of the univariate marginal distribution, is parametric pseudo-maximum likelihood. Pseudo-maximum likelihood estimation is a two-step procedure where first the univariate marginal distribution functions are estimated and then used for maximizing the likelihood (Pfaff, 2016). By estimating the parameters in the univariate marginal distributions and copula separately, high-dimensional optimization is avoided (Ruppert & Matteson, 2015).

If the parameters for ***ft2***, which was the non-parametric approach, were used as starting values for the MLE instead then the estimation method would be considered semiparametric pseudo maximum likelihood.

(d) Estimate the coefficient of lower tail dependence for this copula. (10 pts)

Tail dependence measures association between the extreme values of two random variables and depends only on their copula (Ruppert & Matteson, 2015). The coefficient of lower tail dependence is denoted by λ_l and is defined by equations 8.20 and 8.21:

$$\lambda_\ell := \lim_{q \downarrow 0} \frac{C_Y(q,q)}{q}$$

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

Therefore, the coefficient of tail lower tail dependence for this copula is 0.4535.

```
# The coefficient of tail dependence:
lambda_l = 2 * pt( - sqrt( (nu+1)*(1-rho) / (1+rho) ), nu+1 )
lambda_l
#[1] 0.4535334
```

REFERENCES

Pfaff, B. (2016). *Financial risk modelling and portfolio optimization with R.* Chichester: West Sussex, United Kingdom.

Ruppert, D., & Matteson, D. S. (2015). *Statistics and data analysis for financial engineering: With R examples*. New York: Springer.