###############################################################

# Example 7.2 #

# Fit Copula Parameters and Simulate Credit Portfolio Losses #

###############################################################

# 1. Load data

library(copula)

t.orig <- as.data.frame(read.csv("Chap7sector5.csv",

header = TRUE, sep = ";", dec="."))

sector.5 <- as.matrix(read.csv("Chap7sector5head.csv",

header = TRUE, sep = ";", dec="."))

SIMcustomer.5 <- as.data.frame(read.csv("Chap7copula5.csv",

header = TRUE, sep = ";", dec="."))

# 2. Set parameters

set.seed(1234567)

t.d <-t.orig[,2:6]

mat.cor.td <- cor(t.d, method="kendall")

cor.td <- mat.cor.td[upper.tri(mat.cor.td,diag=FALSE)]

n.row.td <- nrow(t.d)

n.col.td <- ncol(t.d)

nSimI <- 10000

# 3. Fit data with Student-T copula

t.cop1 <- tCopula(cor.td,dim=n.col.td, dispstr="un",

df=5,df.fixed=TRUE)

aa <- rep(0, n.col.td\*(n.col.td-1)/2)

u <- apply(t.d, 2, rank) / (n.row.td + 1)

fit.tau <- fitCopula(t.cop1, u, method="itau")

param.estimate <-attributes(fit.tau)$estimate

fitted.t.copula <- tCopula(param.estimate, dim=n.col.td,

dispstr="un", df=5, df.fixed=TRUE)

# 4. Simulate random generations

r.t.copula <- rcopula(fitted.t.copula,nSimI)

# 5 Merge simulation with SECTOR

rand.copula <-cbind(sector.5,t(as.matrix(r.t.copula)))

# 6. Merge database simulation and customer (by row)

mm1 <- as.matrix(SIMcustomer.5$SECTOR)

colnames(mm1) <- c("SECTOR")

rand.copula.1 <- merge(mm1, rand.copula,

by.x = "SECTOR", by.y = "SECTOR")

rand.copula.pd <-as.matrix(rand.copula.1

[,2: ncol(rand.copula.1)] )

# 7 Simulate defaults

cr.customer <- 200

sim.default0cr <- matrix(NaN, nrow=cr.customer,ncol=nSimI)

for (i in 1:cr.customer)

for (j in 1:nSimI)

{

{

verif <- SIMcustomer.5[i,4]- rand.copula.pd[i,j]

if(verif>=0){sim.default0cr[i,j]<-(1-SIMcustomer.5[i,5])}

else{sim.default0cr[i,j]<- 1}

}

}

percen <- 0.999

ptf.value0cr <- t(SIMcustomer.5[1:cr.customer,3])%\*%sim.default0cr

loss0cr <- sum(SIMcustomer.5[1:cr.customer,3])-ptf.value0cr

q0cr.999 <- quantile(loss0cr, percen)

UL0cr.999 <- q0cr.999 - mean(loss0cr)

sortloss0cr <- as.matrix(sort(loss0cr, decreasing=FALSE))

ES0cr.999 <- mean(sortloss0cr[percen\*nSimI:nSimI,])

# Figure Loss

hist(loss0cr, breaks = 50, col="lightblue",

main="",axes = TRUE,ylab="Frequency",xlab="Loss", cex.main = 2, font.main= 2, col.main= "black",cex.lab = 2.0, font.lab = 2.0, col.lab = "black", cex.axis=2)#,axes = TRUE # 8