Assignment 2

Arvid Gramer and Anton Linnér

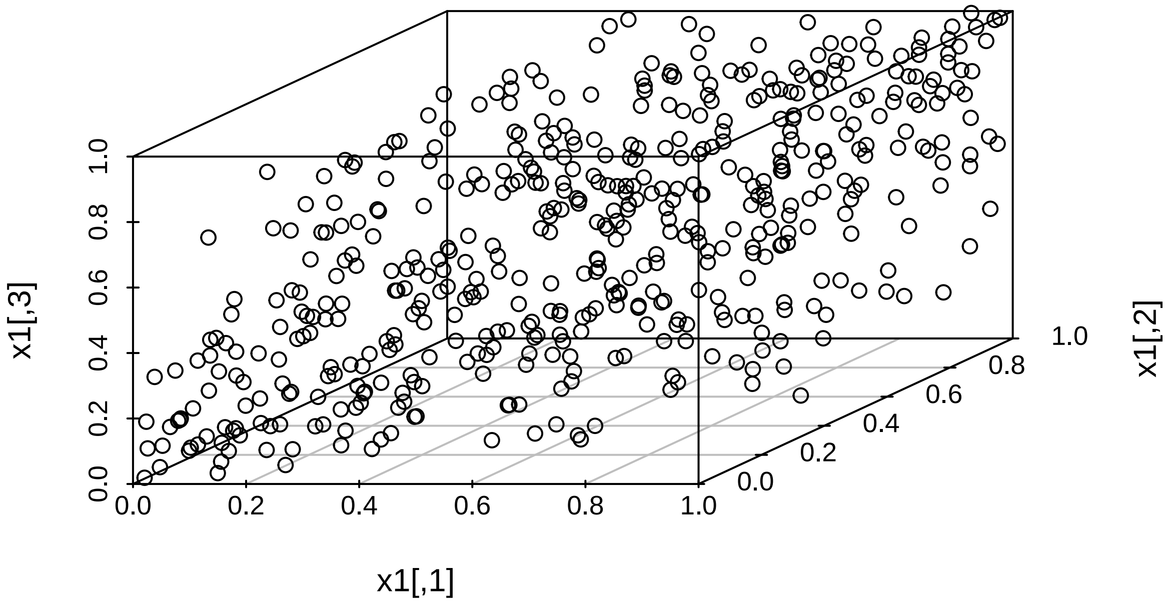
2023-09-28

Exercise 2

Exercise 2.1

We can see that Spearmans Rho and normal correlation are somewhat similar, but Kendalls Tau is lower than both. We also see that the theoretical values are very close to the empirical values for all methods. And lastly we see that when we rank the observations, Spearmans rho gives the same value as when we didn't rank the values.

```
myCop.norm <- ellipCopula(family="normal", dim=3, dispstr="ex", param = 0.4)
x1 <- rCopula(500, myCop.norm)
scatterplot3d(x1)
```




```
#Correlation
(rho1 = cor(x1[,1], x1[,2]))
```

```
## [1] 0.3803658
```

```
(rho2 = cor(x1[,1], x1[,3]))
```

```
## [1] 0.4256553
```

```r
(rho3 = cor(x1[,2], x1[,3]))
```

```
## [1] 0.4044758
```

```r
#Kendall
(cor(x1[,1], x1[,2], method="kendall"))
```

```
## [1] 0.2598958
```

```r
(cor(x1[,1], x1[,3], method="kendall"))
```

```
## [1] 0.2879519
```

```r
(cor(x1[,2], x1[,3], method="kendall"))
```

```
## [1] 0.2764088
```

```r
#Theoretical
(t1_theo = 2/pi*asin(0.4))
```

```
## [1] 0.2619798
```

```r
#Spearman
(cor(x1[,1], x1[,2], method="spearman"))
```

```
## [1] 0.3800591
```

```r
(cor(x1[,1], x1[,3], method="spearman"))
```

```
## [1] 0.4240455
```

```r
(cor(x1[,2], x1[,3], method="spearman"))
```

```
## [1] 0.4072218
```

```r
#Theoretical
rho_s_theo = 6/pi*asin(0.2)

#Rho
(rho_Ss1 = cor(rank(x1[,1]), rank(x1[,2])))
```

```
## [1] 0.3800591
```

```r
(rho_Ss2 = cor(rank(x1[,1]), rank(x1[,3])))
```

```
## [1] 0.4240455
```

```r
(rho_Ss3 = cor(rank(x1[,2]), rank(x1[,3])))
```

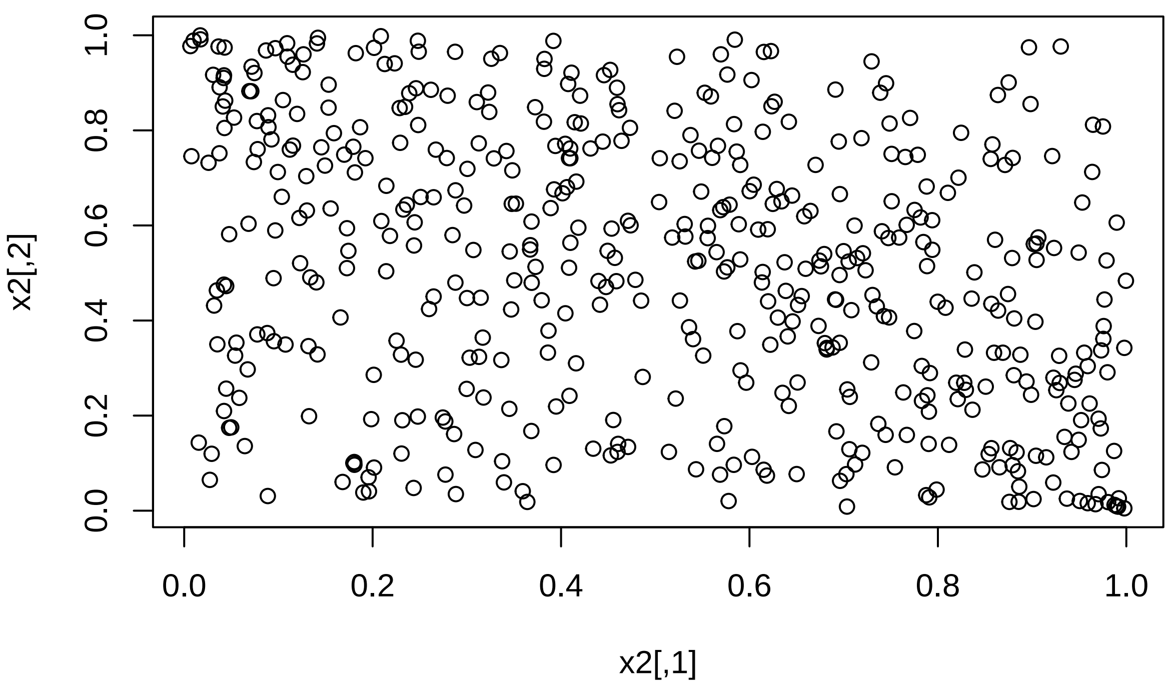
```
## [1] 0.4072218
```

Exercise 2.2

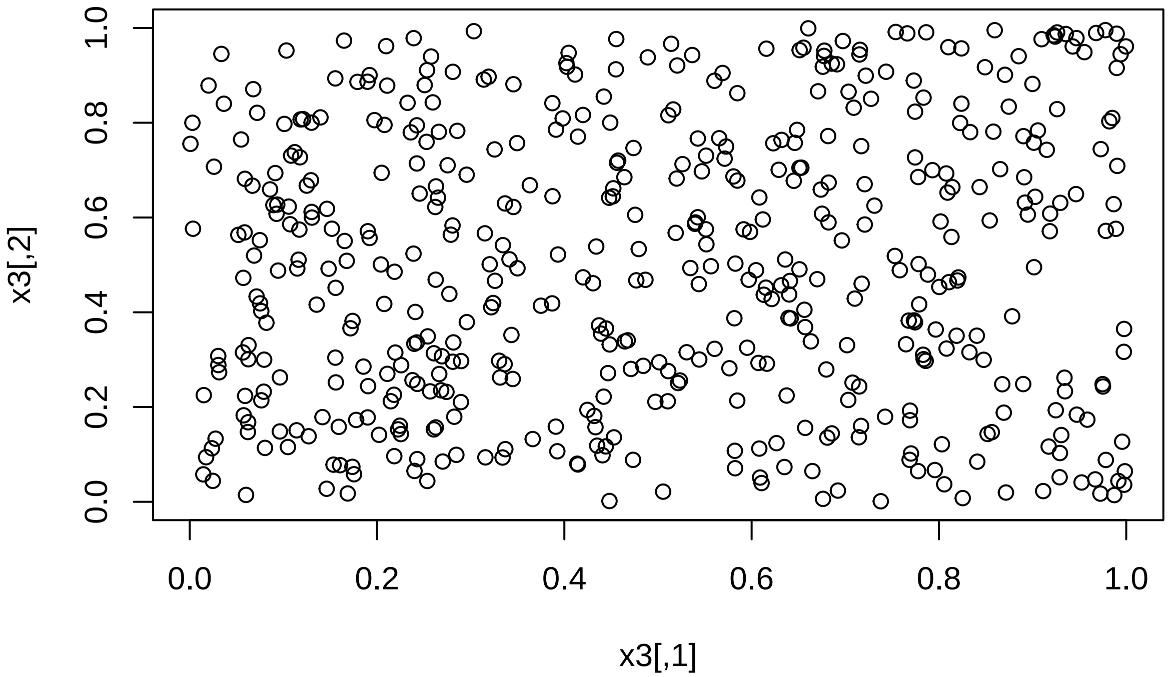
```
myCop.t1 <- ellipCopula(family="t", dim=2, dispstr="ex", param = -0.4, df = 8)
myCop.t2 <- ellipCopula(family="t", dim=2, dispstr="ex", param = 0, df = 8)
myCop.t3 <- ellipCopula(family="t", dim=2, dispstr="ex", param = 0.4, df = 8)

x2 = rCopula(500, myCop.t1)
x3 = rCopula(500, myCop.t2)
x4 = rCopula(500, myCop.t3)
```

As in previously when Gaussian copula was used, the methods seem to give results that are in accordance with the theoretical results.

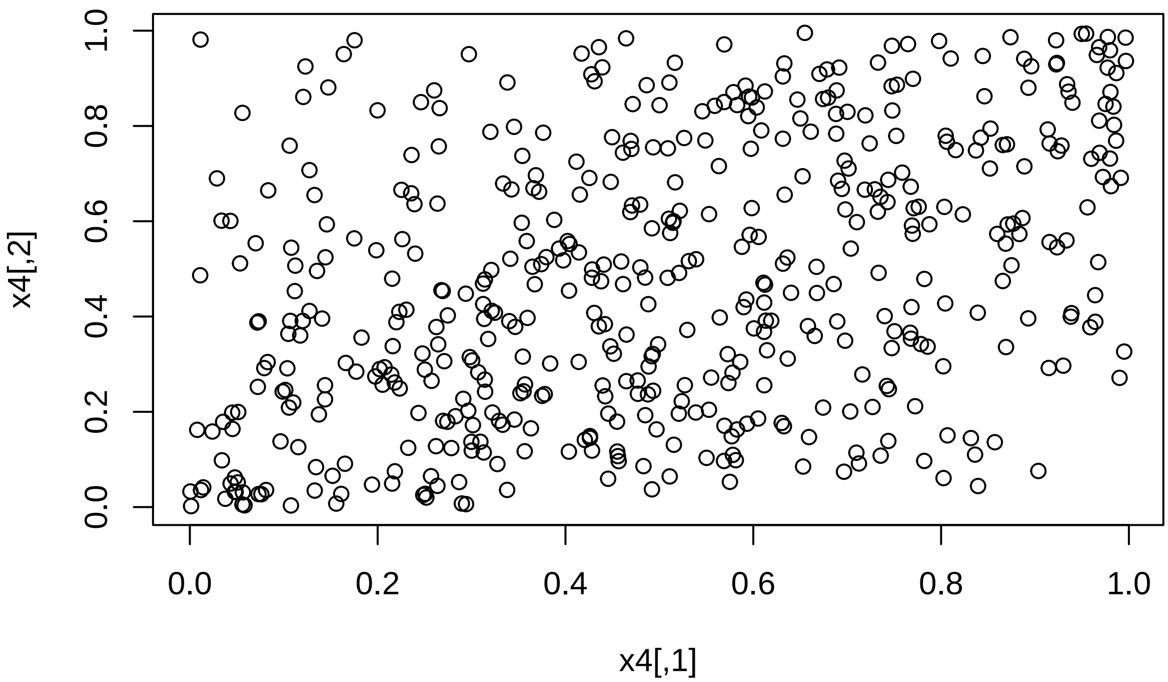
```
plot(x2)
```




```
plot(x3)
```




```
plot(x4)
```




```
(rho1 = cor(x2[,1], x2[,2]))
```

```
## [1] -0.3255582
```

```
(rho2 = cor(x3[,1], x3[,2]))
```

```
## [1] 0.08466172
```

```
(rho3 = cor(x4[,1], x4[,2]))
```

```
## [1] 0.4104306
```

```
#Kendall

#Empirical
(cor(x2[,1], x2[,2], method="kendall"))
```

```
## [1] -0.2304128
```

```
(cor(x3[,1], x3[,2], method="kendall"))
```

```
## [1] 0.05882164
```

```r
(cor(x4[,1], x4[,2], method="kendall"))
```

```
## [1] 0.2802886
```

```r
#Theoretical for the three cases
(t1_theo = 2/pi*asin(-0.4))
```

```
## [1] -0.2619798
```

```r
(t1_theo = 2/pi*asin(0))
```

```
## [1] 0
```

```r
(t1_theo = 2/pi*asin(0.4))
```

```
## [1] 0.2619798
```

```r
#Spearman
#Empirical
(cor(x2[,1], x2[,2], method="spearman"))
```

```
## [1] -0.3291115
```

```r
(cor(x3[,1], x3[,2], method="spearman"))
```

```
## [1] 0.08433154
```

```r
(cor(x4[,1], x4[,2], method="spearman"))
```

```
## [1] 0.4074197
```

```r
#Theoretical
(rho_s_theo = 6/pi*asin(-0.2))
```

```
## [1] -0.3845653
```

```r
(rho_s_theo = 6/pi*asin(0))
```

```
## [1] 0
```

```r
(rho_s_theo = 6/pi*asin(0.2))
```

```
## [1] 0.3845653
```

Exercise 2.3

By looking at the appendix and solving for the expression for the τ using $\theta = 3$ and k $= 1$. Using this, we get a value for τ thats about 0.3, which is in agreement with what is calculated below.

```
frank.cop <- frankCopula(dim = 3, param=3)
x5 = rCopula(500, frank.cop)

(rho1_x5 = cor(x5[,1], x5[,2]))
```

```
## [1] 0.4798229
```

```
(rho2_x5 = cor(x5[,1], x5[,3]))
```

```
## [1] 0.4858532
```

```
(rho3_x5 = cor(x5[,2], x5[,3]))
```

```
## [1] 0.4227796
```

```
(t1_x5 = cor(x5[,1], x5[,2], method = "kendall"))
```

```
## [1] 0.3278397
```

```
(t2_x5 = cor(x5[,1], x5[,3], method = "kendall"))
```

```
## [1] 0.3331303
```

```
(t3_x5 = cor(x5[,2], x5[,3], method = "kendall"))
```

```
## [1] 0.2868617
```

```
(rho_s1 = cor(x5[,1], x5[,2], method = "kendall"))
```

```
## [1] 0.3278397
```

```
(rho_s2 = cor(x5[,1], x5[,3], method = "kendall"))
```

```
## [1] 0.3331303
```

```
(rho_s3 =cor(x5[,2], x5[,3], method = "kendall"))
```

```
## [1] 0.2868617
```

Exercise 2.4

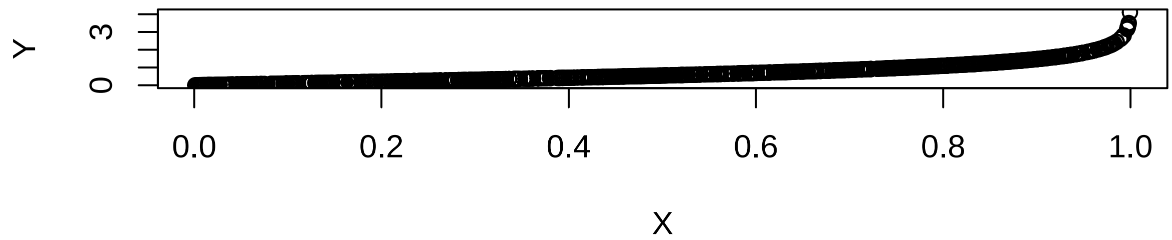
Below is the code for the function:

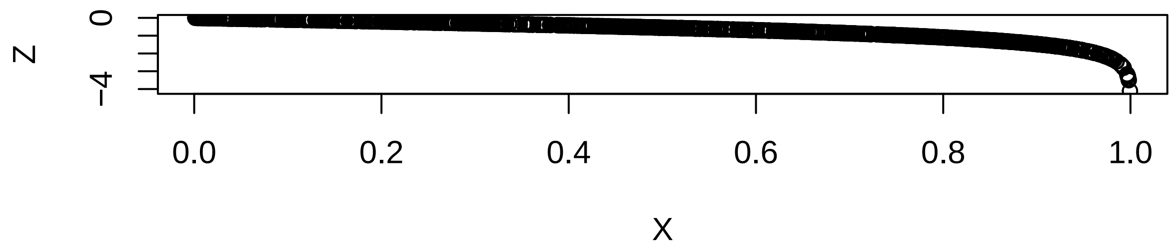
```r
gammaRand <- function(n,theta){
   B= 1/theta
   M=runif(n)
   E = rexp(2*n)
   dim(E) = c(2,n)
   E=t(E)
   U = E/M
   sample = B/(U+B)

}
gammaRand(500, 0.5)
gammaRand(500, 1)
gammaRand(500, 5)
gammaRand(500, 50)
```

Exercise 2.5

The reason T(x) is increasing is due to the fact that arctanh is increasing in this interval (between -1 and 1). We get -1 and 1 as dependace for Kendalls Tau and Spearman, but only -0.914 and + 0.914 for normal correlation. Since Z and Y are created directly from X, it is reasonable that they should have perfect dependance. How ever, since there is a non-linear transform, normal Rho wont be quite as good when trying to measure this depndance.

```r
X = runif(1000)
Y= atanh(X)
Z=-Y
par(mfrow=c(2,1))
plot(X,Y)
plot(X,Z)
```






```
cor(X,Y, method="kendall")
```

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

```
cor(X,Z, method = "kendall")
```

```
## [1] -1
```

```
cor(X,Y, method= "spearman")
```

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

```
cor(X,Z, method = "spearman")
```

```
## [1] -1
```

```
cor(X,Y)
```

```
## [1] 0.9067501
```

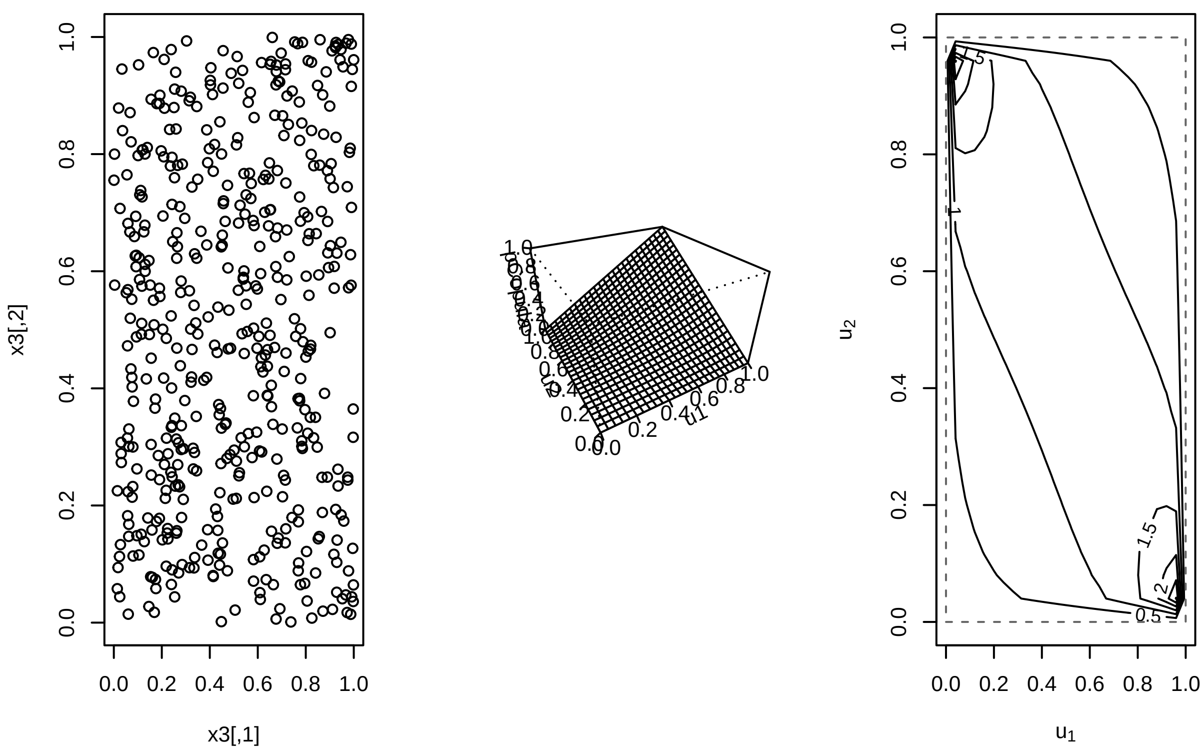
```
cor(X,Z)
```

```
## [1] -0.9067501
```

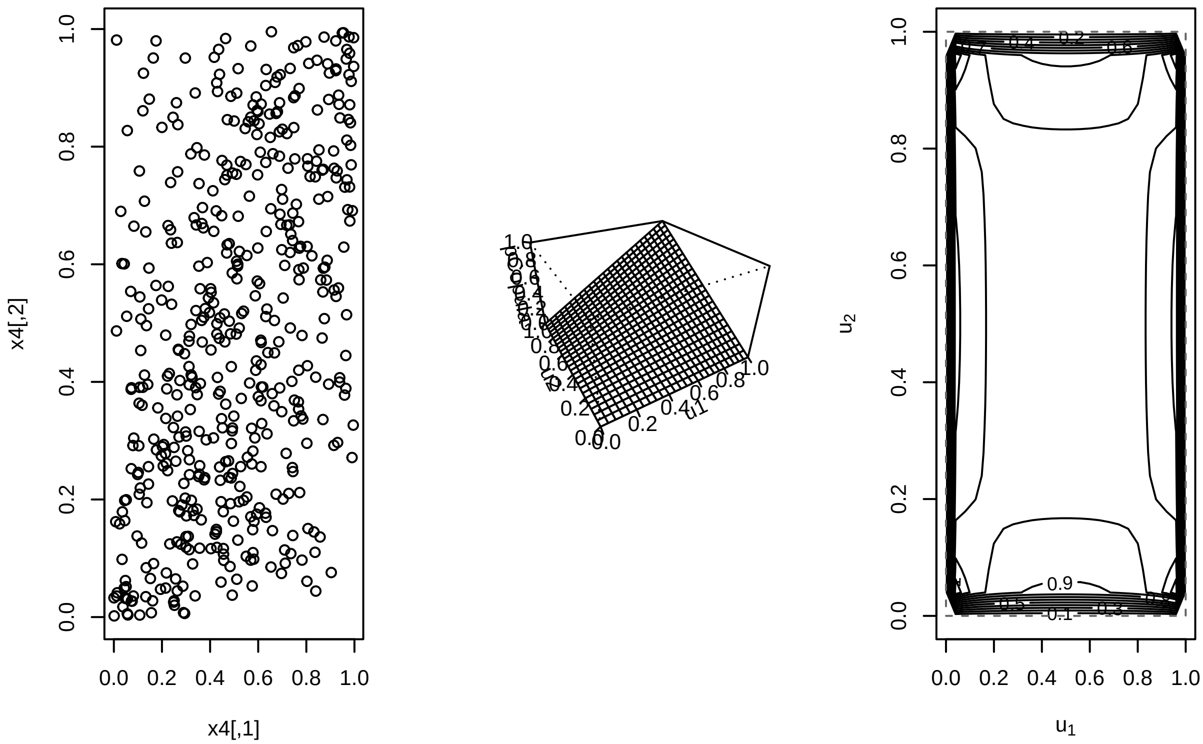
Exercise 3

The plots for the t-copula. For the first plots (scatter, cdf, pdf) we can see that the plots show more dependance, which is reasonable since this is how they were created.

```
par(mfrow= c(1,3)) #mfrow = c(1,2) is one row two cols.


plot(x3)
persp(myCop.t1, pCopula)
contour(myCop.t1, dCopula)
```




```
plot(x4)
persp(myCop.t2, pCopula)
contour(myCop.t2, dCopula)
```



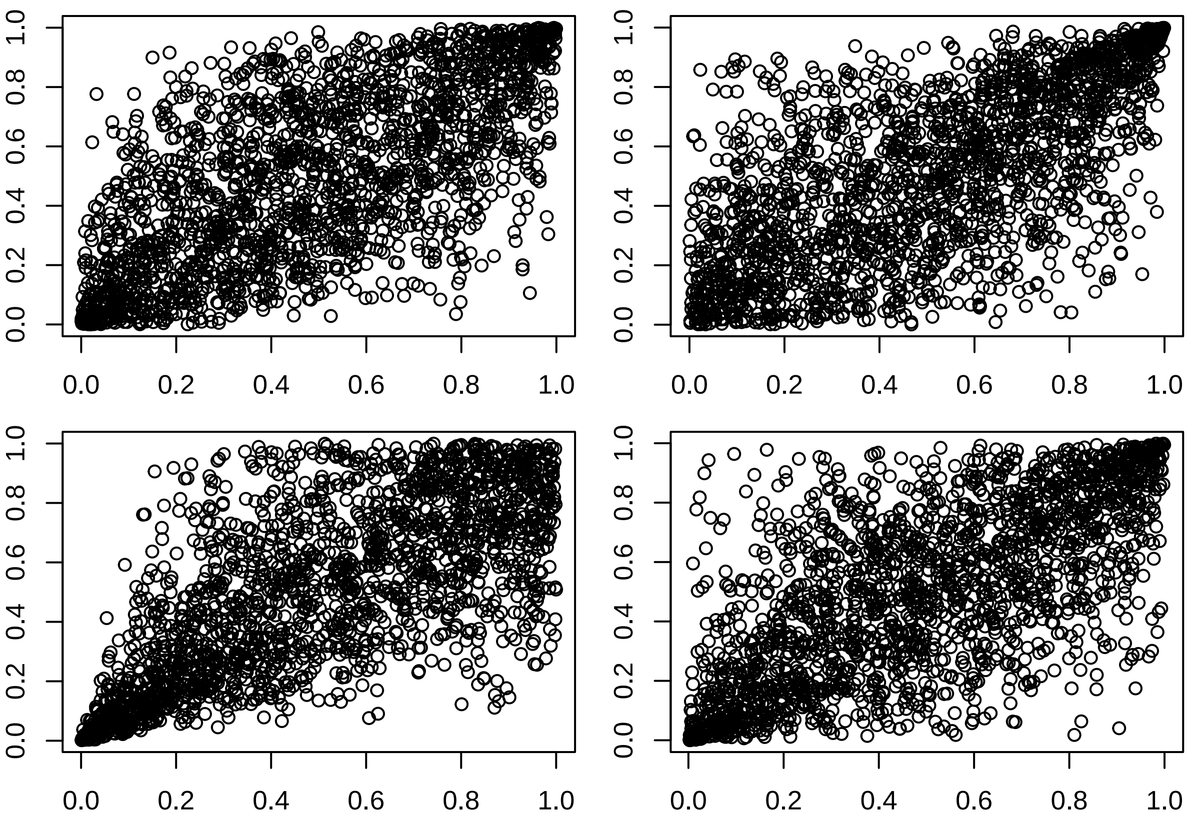
Exercise 4

Scatterplots

The plots are, in order, Gaussian, Gumble, Clayton and t-copula. We can see that there is a somewhat similar dependance in all of the plots. We can see in plot 3 (Clayton), it seems to have quite a strong lower-tail dependance. The t-copula seems to have stronger tail dependance than the Gaussian, which might be expected. The Gumble copula seems to have slightly stronger upper tail dependance.

```r
# Start with simulating samples:
gauss.cop <- ellipCopula(family="normal", dim=2, dispstr="ex", param = 0.7) # Is this gaussian?
gumbel.cop <- archmCopula("gumbel",2 )
clayton.cop <- claytonCopula(2.2, dim = 2)
myCop.t <- ellipCopula(family="t", dim=2, dispstr="ex", param = 0.71, df = 4)
x6= rCopula(2000, gauss.cop)
x7= rCopula(2000, gumbel.cop)
x8= rCopula(2000, clayton.cop)
x9= rCopula(2000, myCop.t)


par(mfrow= c(2,2), mar = c(2,2,1,1), oma=c(1,1,0,0), mgp = c(2,1,0))
plot(x6)
plot(x7)
```

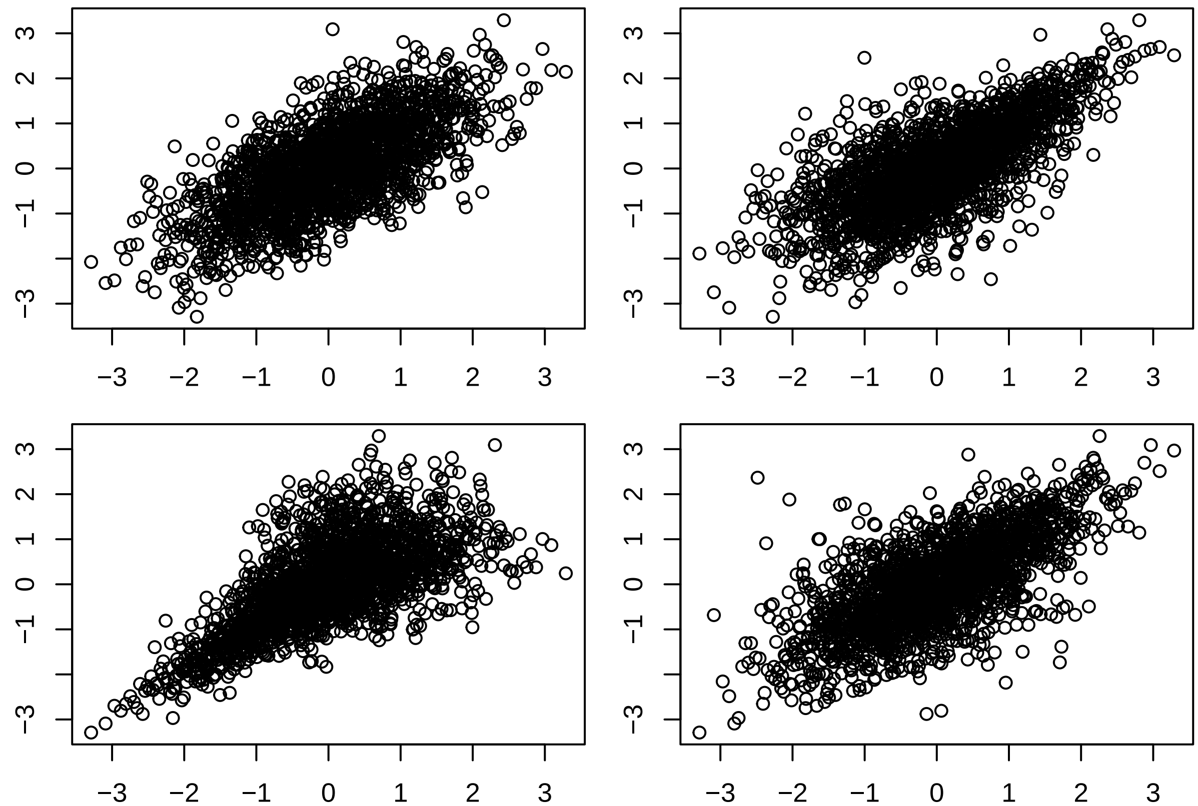
```
plot(x8)
plot(x9)
```



Quantile transformation

What we can see in these plots is that we still have the same behaviour which was described in the previous section, only slightly clearer.

```
par(mfrow= c(2,2), mar = c(2,2,1,1), oma=c(1,1,0,0), mgp = c(2,1,0))
x6= rCopula(2000, gauss.cop)
x7= rCopula(2000, gumbel.cop)
x8= rCopula(2000, clayton.cop)
x9= rCopula(2000, myCop.t)

x6[,1] = rank(x6[,1])/length(x6[,1])
x6[,2] = rank(x6[,2])/length(x6[,2])

x6[,1] <- qnorm(x6[,1])
x6[,2] <- qnorm(x6[,2])



x7[,1] = rank(x7[,1])/length(x7[,1])
x7[,2] = rank(x7[,2])/length(x7[,2])

x7[,1] <- qnorm(x7[,1])
x7[,2] <- qnorm(x7[,2])


x8[,1] = rank(x8[,1])/length(x8[,1])
x8[,2] = rank(x8[,2])/length(x8[,2])

x8[,1] <- qnorm(x8[,1])
x8[,2] <- qnorm(x8[,2])

x9[,1] = rank(x9[,1])/length(x9[,1])
x9[,2] = rank(x9[,2])/length(x9[,2])

x9[,1] <- qnorm(x9[,1])
x9[,2] <- qnorm(x9[,2])
plot(x6)
plot(x7)
plot(x8)
plot(x9)
```



Exercise 5

As we can see, when we use the Gumbel copula with either two exponential margins or two standard normal margins, we get a similar Spearman Rho (around 0.65 to 0.75).

```
cop_dist1 <- mvdc(copula=ellipCopula(family="norm", dim = 2, param=0.75), margins=c("norm", "exp"), para
cop_dist2 <- mvdc(copula=archmCopula("gumbel",dim=2, 2), margins=c("exp", "exp"), paramMargins = list(li

x10 = rMvdc(500, cop_dist1)
x11 = rMvdc(500, cop_dist2)
# FIrst
cor(x10[,1],x10[,2], method = "spearman")
```

```
## [1] 0.7107968
```

```
cor(x10[,1],x10[,2])
```

```
## [1] 0.645677
```

```
#Second
cor(x11[,1],x11[,2])
```

```
## [1] 0.812511
```

```r
cor(x11[,1],x11[,2], method = "spearman")
```

```
## [1] 0.71049
```

```r
#third compared to 2nd
cop_dist3 <- mvdc(copula=archmCopula("gumbel",dim=2, 2), margins=c("norm", "norm"), paramMargins = list
x12 = rMvdc(500, cop_dist3)
cor(x12[,1],x12[,2])
```

```
## [1] 0.6610874
```

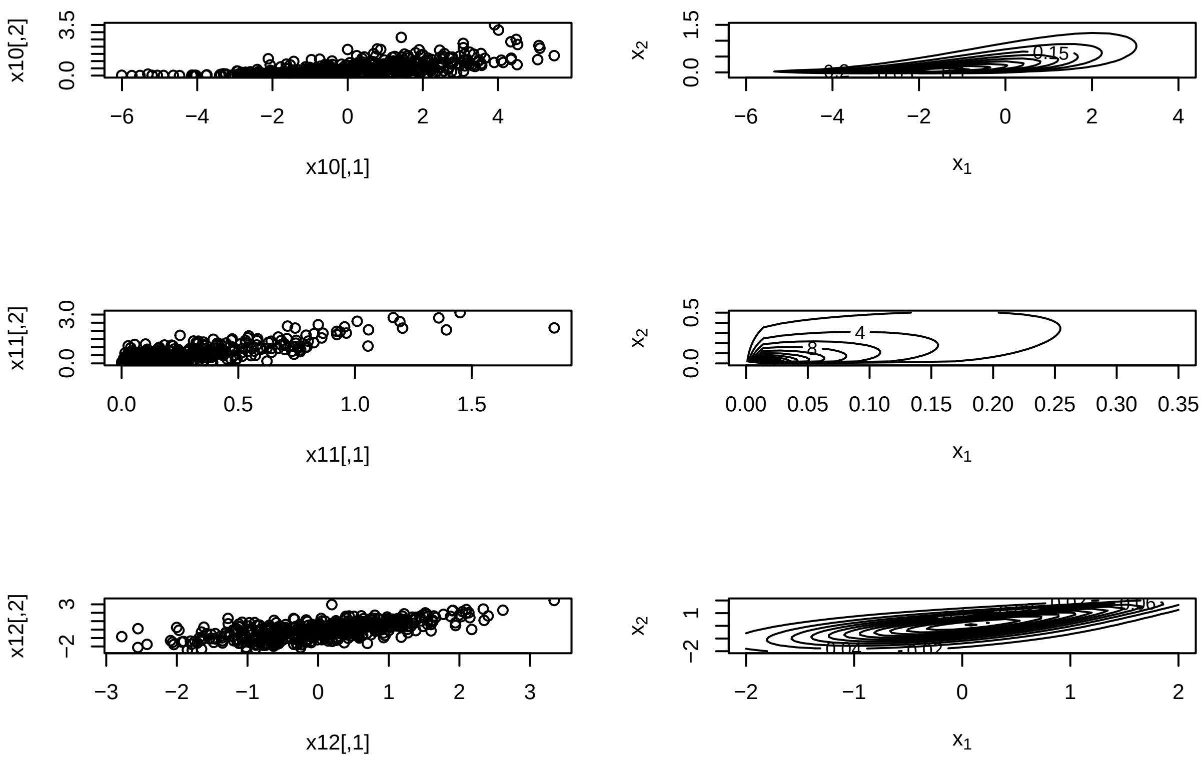
```r
cor(x12[,1],x12[,2], method = "spearman")
```

```
## [1] 0.6386756
```

Exercise 6

Plots

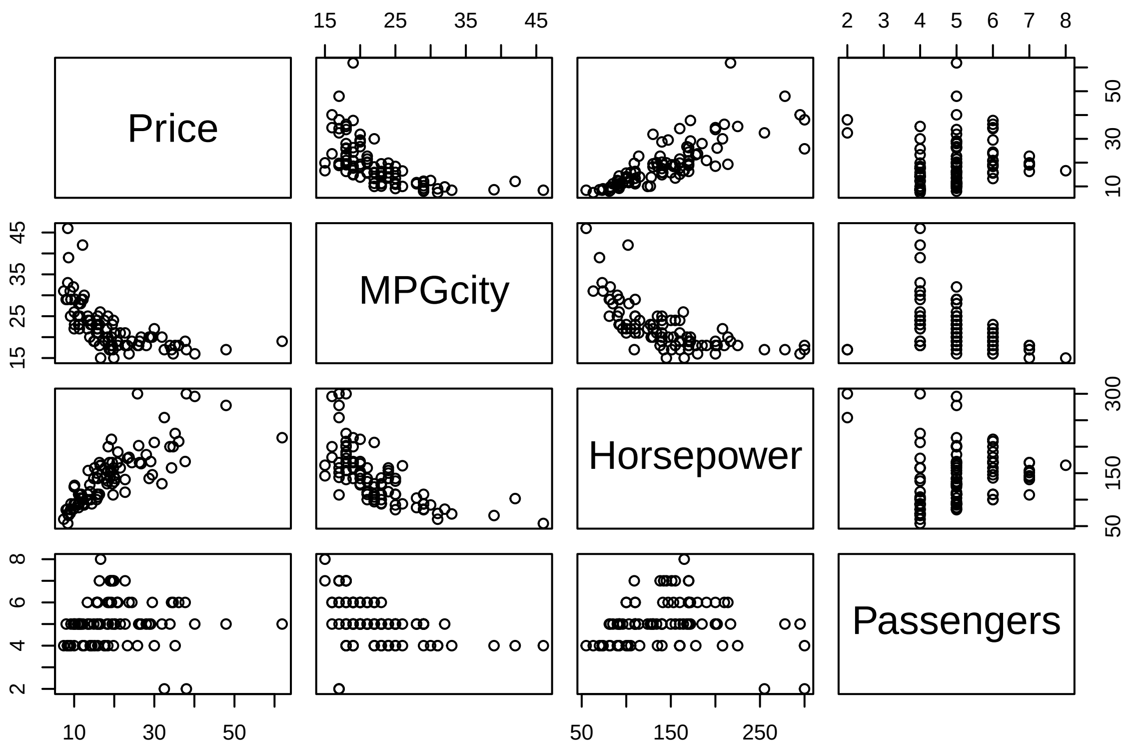
```
#Basically what is left to do is to find good xlim and ylim for the plots, and add plots for the x10, x
par(mfrow= c(3,2))
plot(x10)
contour(cop_dist1, dMvdc, xlim=c(-6, 4), ylim=c(-0.1, 1.5))
plot(x11)
contour(cop_dist2, dMvdc, xlim = c(-0,0.35), ylim=c(-0,0.5))
plot(x12)
contour(cop_dist3, dMvdc, c(-2,2), ylim=c(-2,2))
```



Exercise 7

We can see that the correlation of price and horsepower is high and positive, so it does seen to be the case that vehicles with higher horsepower are more costly. Simliarly, MPGcity and price have a negative correlation.

```
#Load data. Also, change path to where the textfile is.
car_data <- read.csv("C:/Users/anton/Desktop/University/FMSN65  MASM33 - Quantitative Risk Management U

#Pairwise scatterplot of the data
pairs(car_data)
```




```
cor(car_data, method = "spearman")
```

```
##                 Price    MPGcity Horsepower Passengers
## Price       1.0000000 -0.7860976  0.8583904  0.2662086
## MPGcity    -0.7860976  1.0000000 -0.7893071 -0.4942519
## Horsepower  0.8583904 -0.7893071  1.0000000  0.2441089
## Passengers  0.2662086 -0.4942519  0.2441089  1.0000000
```

```
cor.test(car_data[,1],car_data[,3], method = "spearman")
```

```
## Warning in cor.test.default(car_data[, 1], car_data[, 3], method = "spearman"):
## Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  car_data[, 1] and car_data[, 3]
## S = 18982, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.8583904
```

```
cor.test(car_data[,1],car_data[,2], method = "spearman")
```

```
## Warning in cor.test.default(car_data[, 1], car_data[, 2], method = "spearman"):
## Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  car_data[, 1] and car_data[, 2]
## S = 239416, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.7860976
```

Exercise 8

We see that the probabilites of exceeding both thresholds is around 6-8 %, when using the 95th-percentile and 1-2 % for the 99th-percentile. We can also see that the ratios don't change when we change the threshold. We also see that Gumble copula gives the highest value for expected loss at 0.395, and the t-copula gives the lowest expected value at around 0.354.
We can also see from the tail-dependance plots that the Gauss copula shows quite poor dependance. But the Gumble and t-copula shows some upper tail dependance. This makes it more reasonable to use these two copulas if what you are interested in is in the tails of a distribution. An example might be when dealing with stocks.

```
#Calculating theta nad rho from tau= 0.5 gives rho
Gauss <- mvdc( ellipCopula(family="normal", dim=2, dispstr="ex", param = 0.7), margins=c("lnorm", "lnorm

t_cop = mvdc( ellipCopula(family="t", dim=2, dispstr="ex", param = 0.7, df = 2), margins=c("lnorm", "ln

gumb = mvdc( archmCopula("gumbel" , 2), margins=c("lnorm", "lnorm"), paramMargins = list(list(mean=0, s



#The threshould is:
(u = qlnorm(0.95,0,1))
```

```
## [1] 5.180252
```

```
(prob_gauss = 1- pMvdc(c(u,u), Gauss))
```

```
## [1] 0.0804007
```

```
(prob_t = 1- pMvdc(c(u,u), t_cop))
```

```
## [1] 0.07303974
```

```
(prob_gumb =1- pMvdc(c(u,u), gumb))
```

```
## [1] 0.06997115
```

```
(r1 = prob_gauss/prob_t)
```

```
## [1] 1.10078
```

```
(r2 = prob_gauss/prob_gumb)
```

```
## [1] 1.149055
```

```
(r3 = prob_t/prob_gumb)
```

```
## [1] 1.043855
```

```r
#The 99th percentile is:
(u = qlnorm(0.99,0,1))
```

```
## [1] 10.24047
```

```r
#The probabilites of exceeding both thresholds:
(prob_gauss.a = 1- pMvdc(c(u,u), Gauss))
```

```
## [1] 0.0173316
```

```r
(prob_t.b = 1- pMvdc(c(u,u), t_cop))
```

```
## [1] 0.01476622
```

```r
(prob_gumb.c =1- pMvdc(c(u,u), gumb))
```

```
## [1] 0.01411279
```

```r
(r1a = prob_gauss/prob_t)
```

```
## [1] 1.10078
```

```r
(r2b = prob_gauss/prob_gumb)
```

```
## [1] 1.149055
```

```r
(r3c = prob_t/prob_gumb)
```

```
## [1] 1.043855
```

```r
#t-copula
loss = rMvdc(10000, t_cop)
loss[which(loss[,1] <= u & loss[,2] <= u), ] <- 0
loss = rowSums(loss)
(E.t = mean(loss))
```

```
## [1] 0.3086138
```

```r
#Gauss
loss = rMvdc(10000, Gauss)
loss[which(loss[,1] <= u & loss[,2] <= u), ] <- 0
loss = rowSums(loss)
(E.Gauss = mean(loss))
```

```
## [1] 0.3674443
```

```
#Gumble
loss = rMvdc(10000, gumb)
loss[which(loss[,1] <= u & loss[,2] <= u), ] <- 0
loss = rowSums(loss)
(E.gumb = mean(loss))
```

```
## [1] 0.3880207
```

```
par(mfrow= c(1,3))
u = qlnorm(0.95,0,1)

loss = rMvdc(100000, Gauss)
tail.gauss = loss[apply(loss > u, 1, all), ]
plot(tail.gauss,ylim=c(0,100), xlim=c(0,100))

loss = rMvdc(100000, gumb)
tail.gumb = loss[apply(loss > u, 1, all), ]
plot(tail.gumb,ylim=c(0,100), xlim=c(0,100))

loss = rMvdc(100000, t_cop)
tail.t_cop = loss[apply(loss > u, 1, all), ]
plot(tail.t_cop,ylim=c(0,100), xlim=c(0,100))
```