Dependence Modelling: Exploratory data analysis

H. A. Mohtashami-Borzadaran M. Amini

Ferdowsi University of Mashhad

Motivation

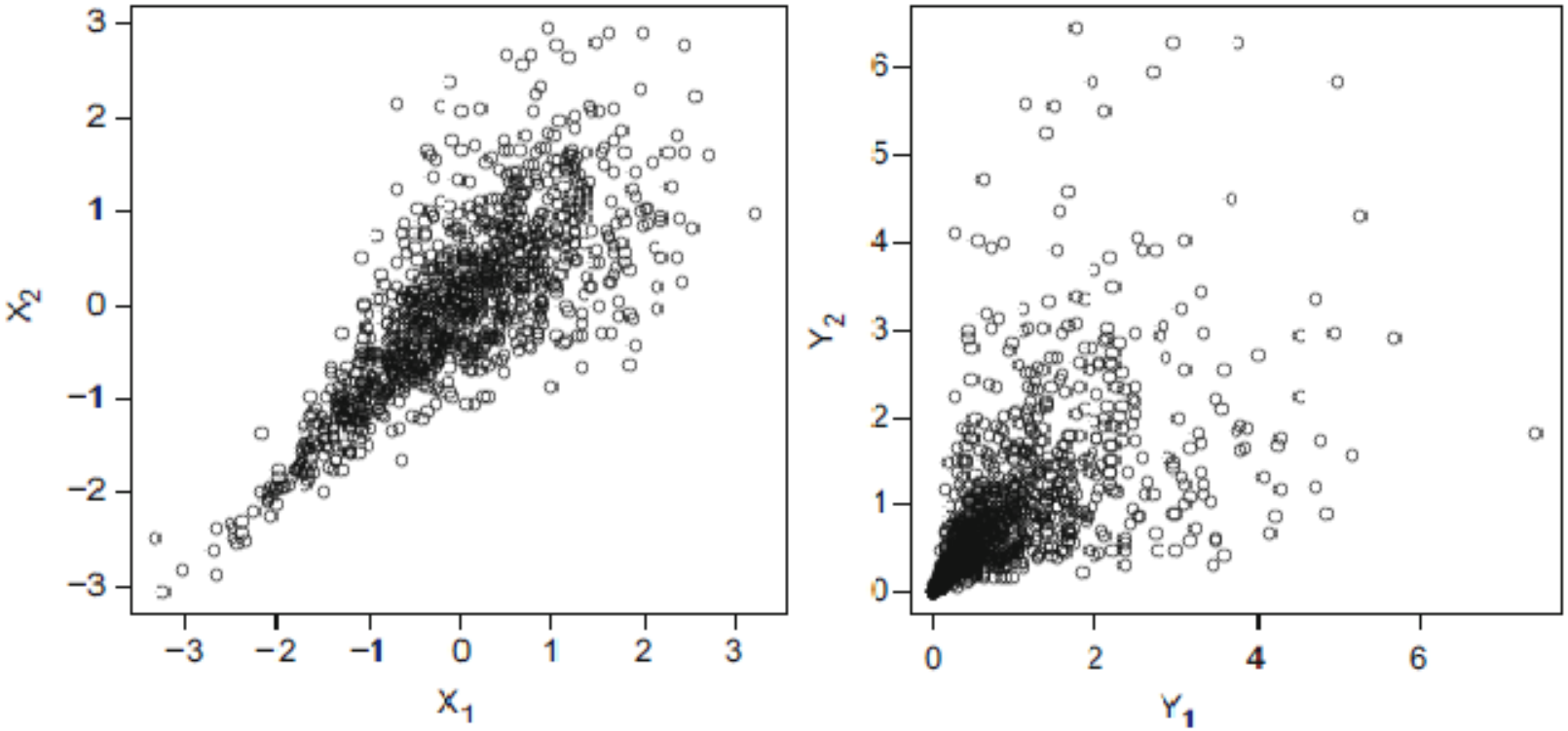


Figure 1: Scatter plots of n = 1000 independent observations.

It is clear you might fit a simple regression model to these data, but there is no dependence among them! So, the simple scatter plot is not a good tool to detect dependence!

How to check for dependence?

The best way would be to do a independence test.

$H_0 : Independence \ \ of \ \ X, Y \quad vs \quad H_1 : dependence \ \ of \ \ X, Y$

```
data(danube, package = "lcopula")
cor.test(danube[,1],danube[,2], method="kendall")
```

```
##
##  Kendall's rank correlation tau
##
## data:  danube[, 1] and danube[, 2]
## z = 21.064, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.5484731
```

Load data

First, let's make fake data:

```
x <- rgamma(350,1,1)
y <- rexp(350,2)
mydata <- cbind(x,y)
head(mydata)
```

```
##                x          y
## [1,] 1.7400731 0.9540654
## [2,] 1.8580755 0.3441064
## [3,] 2.1403457 0.3672433
## [4,] 1.2130525 0.6264123
## [5,] 4.6116069 0.2349267
## [6,] 0.3419637 2.4969196
```

A summary

Having a summary of data is useful in understanding the variation of data:

```
options(digits = 3)
summary(x)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.31    0.73    1.05    1.42    6.91
```

```
summary(y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.003   0.159   0.351   0.479   0.628   2.869
```

```
options(digits = 5)
```

Sklar theorem

Sklar's theorem states that every multivariate cumulative distribution function

$$H(x_1, \ldots, x_d) = \Pr[X_1 \leq x_1, \ldots, X_d \leq x_d]$$

of a random vector

$$(X_1, X_2, \ldots, X_d)$$

can be expressed in terms of its marginals $F_i(x_i) = \Pr[X_i \leq x_i]$ and a copula C. Indeed:

$$H(x_1, \ldots, x_d) = C\left(F_1(x_1), \ldots, F_d(x_d)\right).$$

Probability integral transform

Suppose that a random variable X has a continuous distribution for which the cumulative distribution function (CDF) is $F_X(x)$. Then

$$Y = F_X(X) \sim U(0, 1).$$

Procedure

For example, to find $H(x_1, x_2) = \Pr[X_1 \leq x_1, X_2 \leq x_2]$, we have to find the

Step 1. Find the marginal distributions ($F_1(x_1)$ and $F_2(x_2)$) or use a non-parametric method, and if needed transform the data using the probability integral transform $u_{1i} = F_1(x_{1i})$ and $u_{2i} = F_2(x_{2i})$,

Step 2. Find the copula associated with (u_{1i}, u_{2i}).

What's the plan ?!!!

Do inference on the joint distibution $H(x_1, x_2)$. For Example, $E[X_1|X_2 = x_2]$ and $F^{-1}_{X_1|X_2=x_2}(p)$.

Step 1.

The first step can be performed based on the following methods:

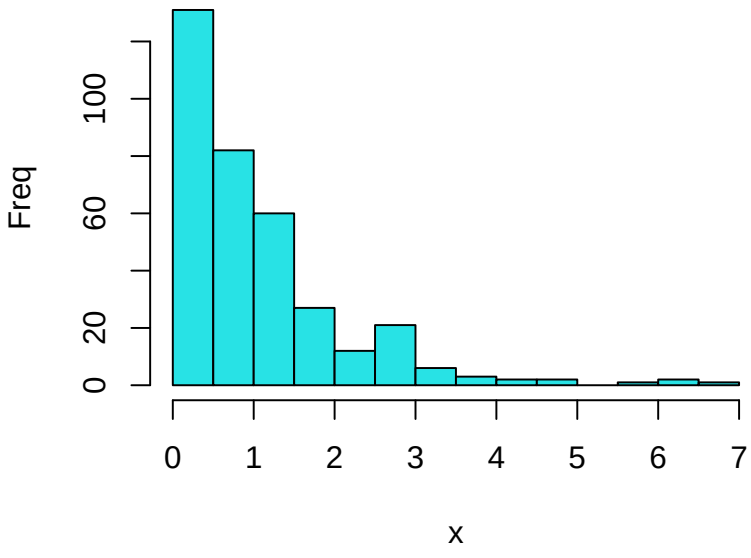
- Parametric method
- Non-parmetric method (Rank estimator)
- Non-parmetric method (Kernel density estimator)

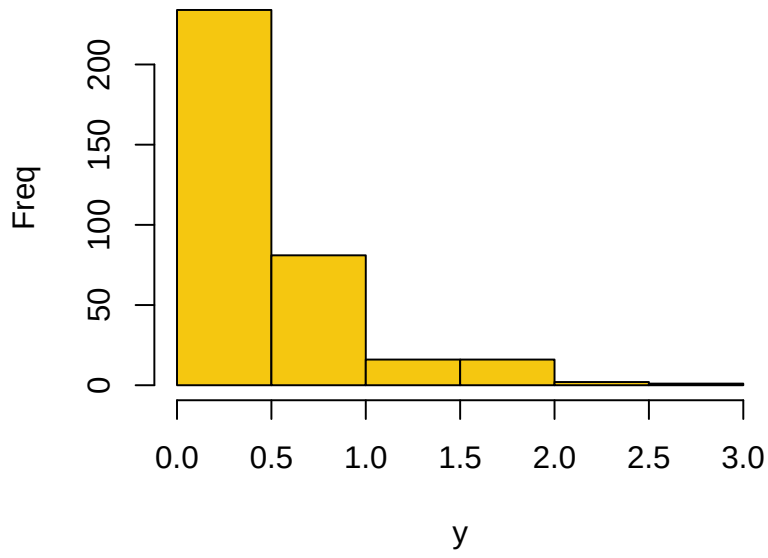
Parametric method

To have a guess of the density function of data, we can use the histogram of data:

```
#par(mfrow=c(1,2))
#hist(x, col = 5, main="", xlab="x", ylab="Freq")
#hist(y, col = 7, main="", xlab="y", ylab="Freq")
#par(mfrow=c(1,1))
```

Histogram


Freq
0
20
60
100
0
1
2
3
4
5
6
7
x


Freq
0
50
100
150
200
0.0
0.5
1.0
1.5
2.0
2.5
3.0
y

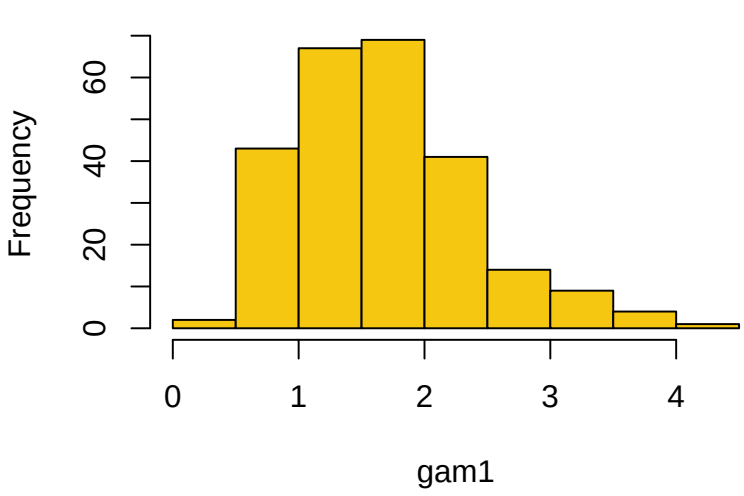
Gamma distribution: A good guess!?

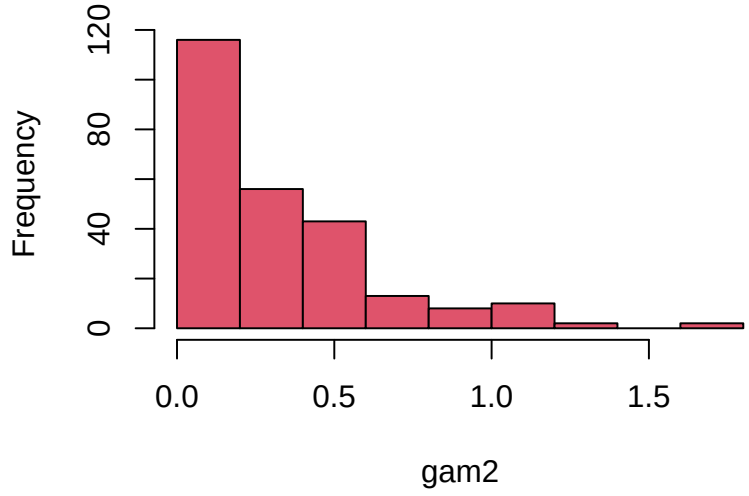
The gamma distribution might be a good candid for both x and y.

```
#gam1 <- rgamma(250, 5 , 3)
#gam2 <- rgamma(250, 1 , 3)
#par(mfrow=c(1,2))
#hist(gam1)
#hist(gam2)
#par(mfrow=c(1,1))
```

Gamma distribution: A good guess!?

The gamma distribution is a good candid for both x and y.





Maximum likelihood estimator

Given set of observations $y = (y_1, ..., y_n)$ distributed from the parametric family $\left\{ f(\cdot\,; \theta) \mid \theta \in \Theta \right\}$ with set of parameters $\Theta = (\theta_1, \theta_2, \ldots, \theta_k)$. The the likelihood function is

$$L_n(\theta) = f_n(y_1, ..., y_n; \Theta).$$

The goal of maximum likelihood estimation is to find the values of the model parameters that maximize the likelihood function over the parameter space, that is

$$\hat{\theta} = \arg\max_{\theta \in \Theta} L_n(\theta).$$

Intuitively, this selects the parameter values that make the observed data most probable.

Maximum likelihood estimator

In practice, it is often convenient to work with the natural logarithm of the likelihood function, called the log-likelihood $l(\theta) = \log\left(L(\theta)\right)$ and for independent observations we have

$$l(\theta) = -\sum_{i=1}^{n} \log f(y_i; \Theta).$$

Since the logarithm is a monotonic function, the maximum of $l(\theta)$ occurs at the same value of θ as does the maximum of $L(\theta)$.

MLE properties

- Consistency: the sequence of MLEs converges in probability to the value being estimated.
- Functional Invariance
- Efficiency, i.e. it achieves the Cramer–Rao lower bound when the sample size tends to infinity.

Model fitting

Now we want to estimate the parameters of gamma distribution (using MLE) based on the data:

```
f<-function(w,alpha,lambda) {
  aa<-w^(alpha-1) * exp(-lambda*w) * lambda^alpha / gamma(alpha)
  return(aa)
}
min.log.lik <- function(alpha,lambda) -sum(log(f(x,alpha,lambda)))
min.log.lik <- Vectorize(min.log.lik)
library(stats4)
mle(min.log.lik,start = list(alpha=0.1,lambda=0.1))@ coef
```

```
##   alpha  lambda
## 0.96065 0.91529
```

Model fitting

Also, we can use the "MASS" package to estimate the parameter of well-known distributions.

```
library(MASS)
options(warn=-1) #warnings turned off
par.x <- fitdistr(x,"gamma")$estimate
par.x
```

```
##   shape    rate
## 0.96065 0.91529
```

```
par.y <- fitdistr(y,"gamma")$estimate
par.y
```

```
##  shape   rate
## 1.2073 2.5207
```

```
options(warn=0) #warnings turned on
```

Goodness of fit

The kolmogorov smirnov GOF test can be performed:

```
ks.test(x,"pgamma", par.x[1], par.x[2])
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  x
## D = 0.0277, p-value = 0.95
## alternative hypothesis: two-sided
```

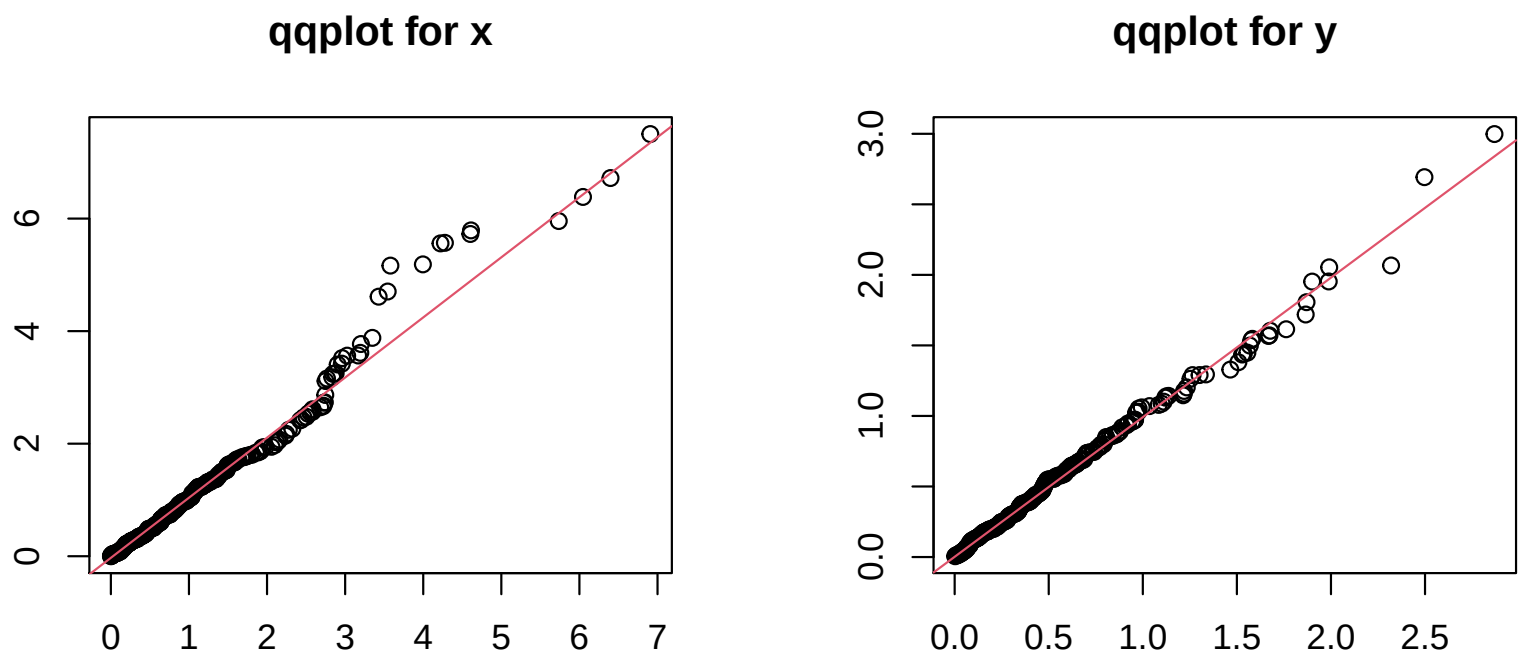
```
ks.test(y,"pgamma", par.y[1], par.y[2])
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  y
## D = 0.0338, p-value = 0.82
## alternative hypothesis: two-sided
```

Goodness of fit

To visually see the GOF test, we use the QQ-plot:

```
par(mfrow=c(1,2))
m1 <- rgamma(length(x), shape = par.x[1], rate=par.x[2])
qqplot(x,m1,main="qqplot for x",xlab = "", ylab = "")
qqline(m1, distribution = function(p) qgamma(p,shape = par.x[1], rate=par.x[2]),probs = c(0.1, 0.6), col =
m2<- rgamma(length(y), shape = par.y[1], rate=par.y[2])
qqplot(y,m2,main="qqplot for y",xlab = "", ylab = "")
qqline(m2, distribution = function(p) qgamma(p,shape = par.y[1], rate=par.y[2]),probs = c(0.1, 0.6), col =
```




```
par(mfrow=c(1,1))
```

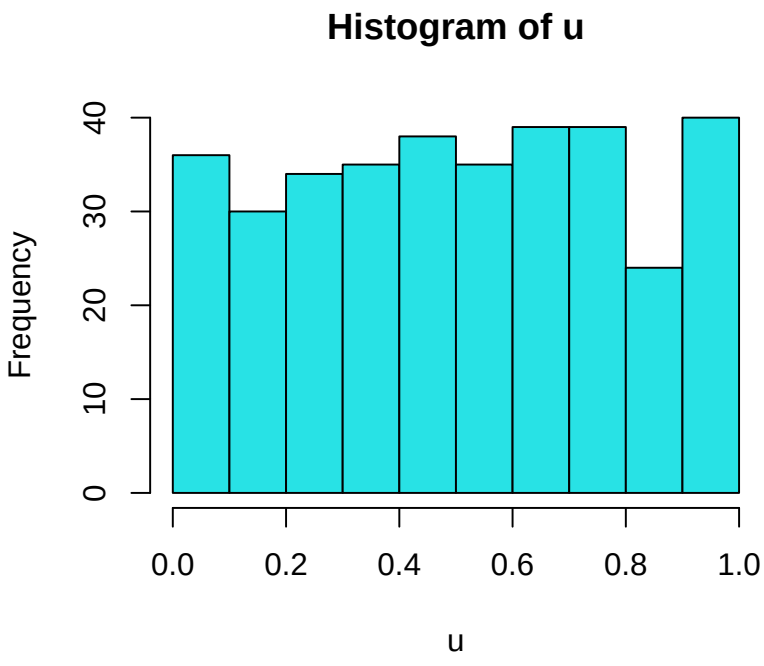
Transform data

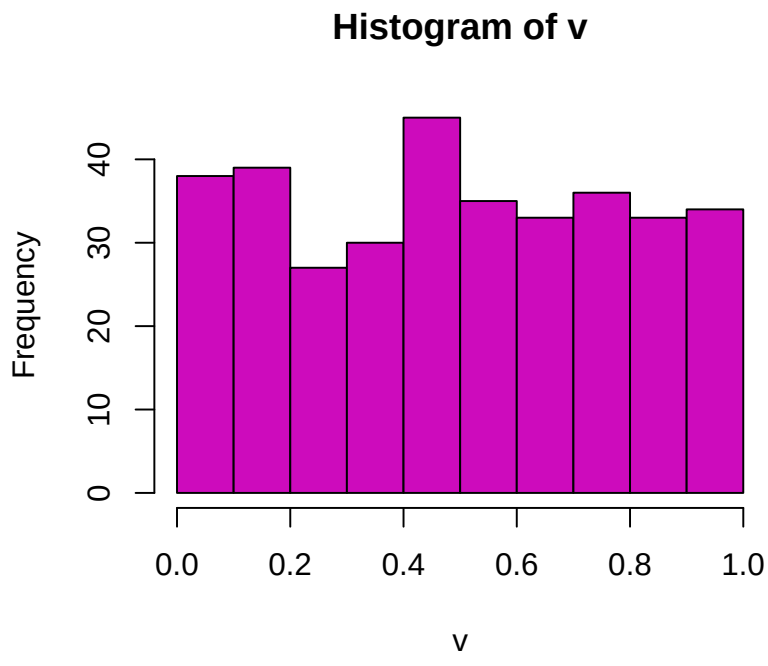
Now, we are going to make the u-scaled data using the fitted distribution:

```
#par(mfrow=c(1,2))
#u <- pgamma(x, shape=par.x[1], rate=par.x[2])
#hist(u)
#v<- pgamma(y, shape=par.y[1], rate=par.y[2])
#hist(v)
#par(mfrow=c(1,1))
```

Transform data

Now, we are going to make the u-scaled data using the fitted distribution:





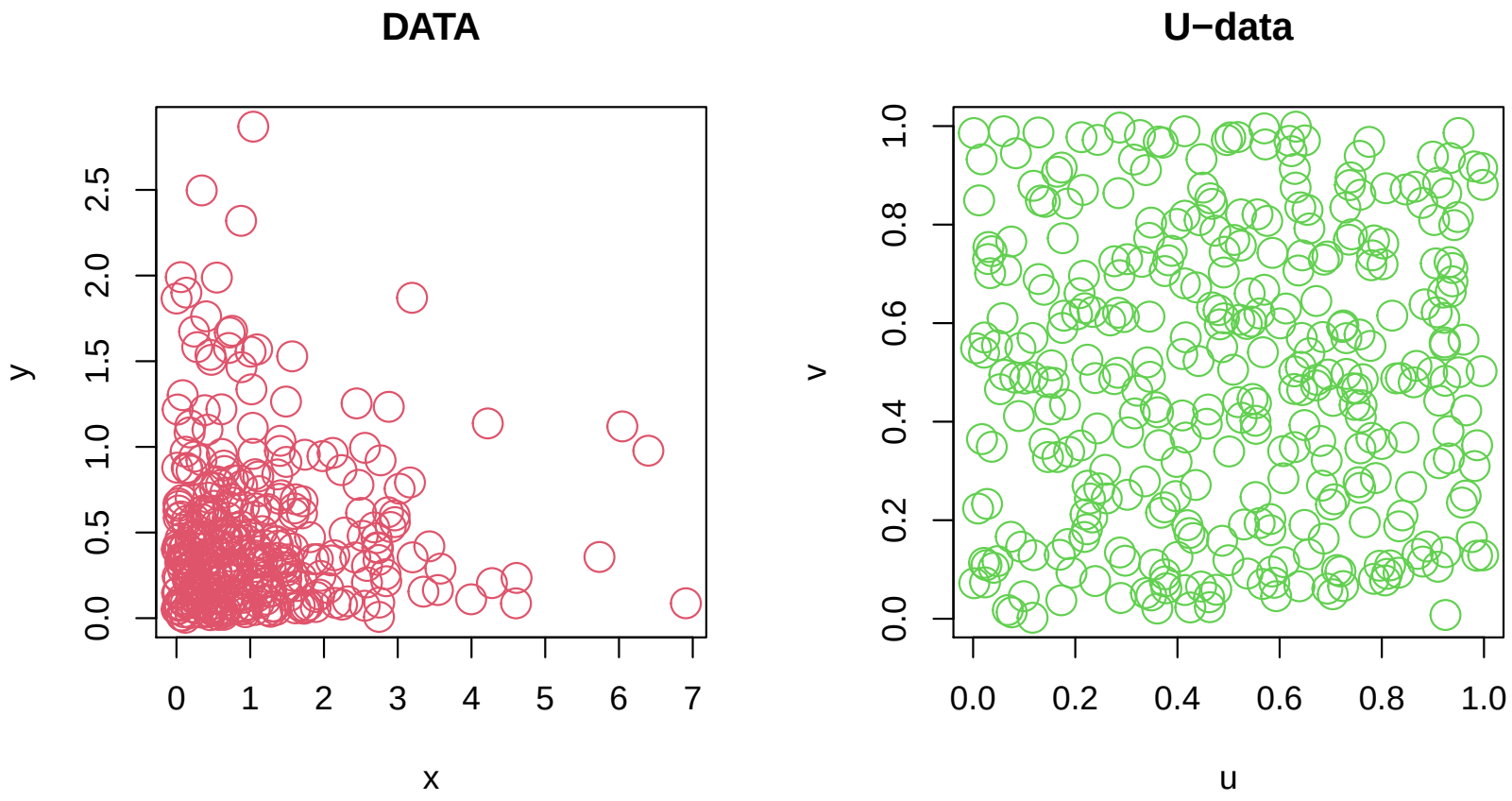
The U-data

In the following you can see the raw data vs the u-scaled data:

```
#par(mfrow=c(1,2))
#plot(x,y,main="DATA",col=2,cex=2)
#plot(u,v,main="U-data",col=3,cex=2)
#par(mfrow=c(1,1))
```

The U-data

In the following you can see the raw data vs the u-scaled data:



Non-parmetric methods

Why Non-parametric method?!!

Noh et al. (2013) point out that modeling the marginals as well as the copula parametrically might cause the resulting fully parametric estimator to be biased and inconsistent if one of the parametric models is misspecified.

Noh, H., Ghouch, A. E., and Bouezmarni, T. (2013), "Copula-based regression estimation and inference," Journal of the American Statistical Association, 108, 676–688.

Non-parmetric method (The Rank estimator)

Definition (Empirical distribution function)

Let $x_1, \ldots, x_n$ be an i.i.d. sample from a distribution function F, then the empirical distribution function is defined as

$$\hat{F}(x) := \frac{1}{n+1} \sum_{i=1}^{n} I_{[x_i \leq x]}, \qquad \textit{for all} \quad x.$$

Division by $n+1$ instead of n is used to avoid boundary problems of the estimator $\hat{F}(x)$.

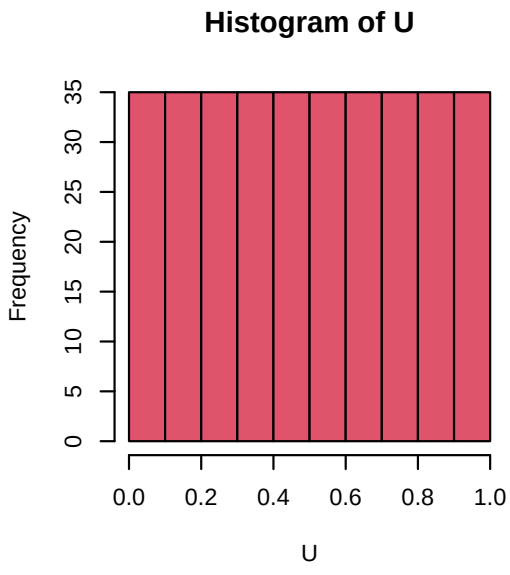
Ranks and empirical distributions:

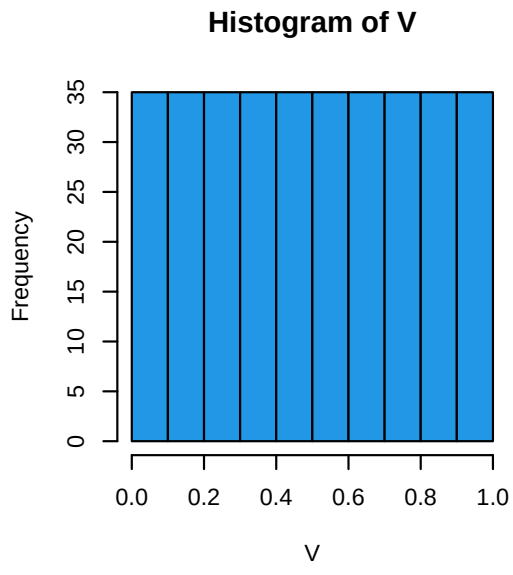
Let R_i be the rank of observation x_i, i.e., $R_i = k$ if the observation x_i is the kth largest observation among the observations $x_1, ..., x_n$. In this case, it follows that $\hat{F}(x_i) = \frac{R_i}{n+1}$ for $i = 1, ..., n$.

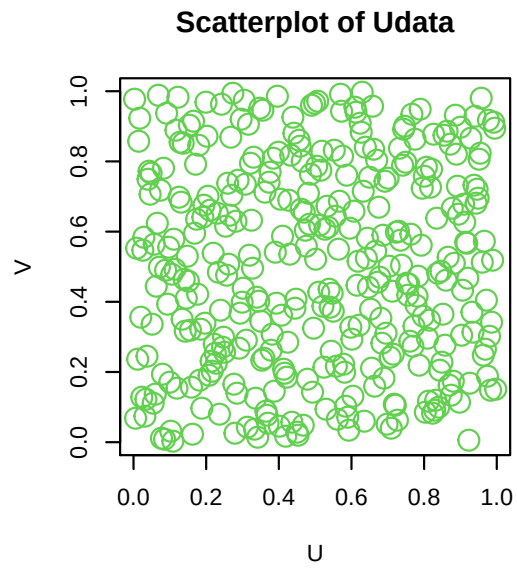
Non-parmetric method (The Rank estimator)

```
#par(mfrow=c(1,3))
#library(copula)
#udata <- pobs(mydata)
#hist(udata[,1],col=2, main="Histogram of U", xlab="U")
#hist(udata[,2],col=4, main="Histogram of V", xlab="V")
#plot(udata,main="Scatterplot of Udata", xlab="U"
#, ylab="V",col=3,cex=2)
#par(mfrow=c(1,1))
```

Non-parmetric method (The Rank estimator)







Non-parmetric method (The Kernel estimator)

We want to use the continuous kernel smoothing estimator, which is, given a sample $\left(x^{(i)}\right)_{i=1,\ldots,n}$, defined as

$$\hat{F}(x) = \frac{1}{n}\sum_{i=1}^{n} K(\frac{x - x^{(i)}}{h}), \quad x \in R,$$

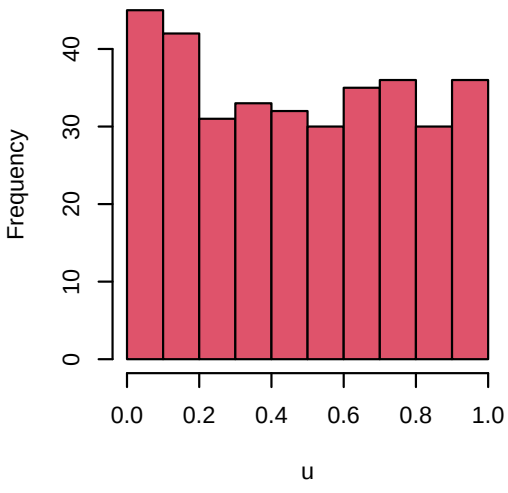
where $K(x) := \int_{-\infty}^{x} k(t)dt$ with $k(.)$ being a symmetric probability density function and $h > 0$ a bandwidth parameter. Usually, we choose $k = \phi$, i.e. a Gaussian kernel, and the plugin bandwidth developed in Duong (2016) which minimizes the asymptotic mean integrated squared error.

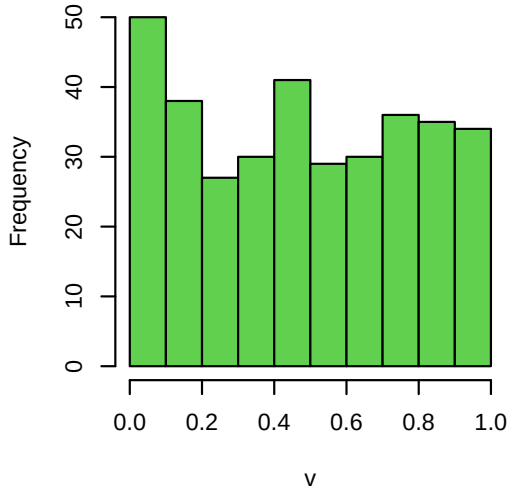
Duong (2016), "Non-parametric smoothed estimation of multivariate cumulative distribution and survival functions, and receiver operating characteristic curves," Journal of the Korean Statistical Society, 45, 33–50.

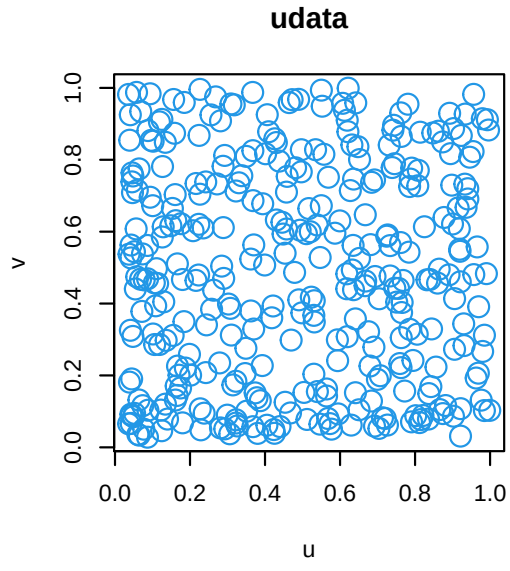
Non-parmetric method (The Kernel estimator)

```
#par(mfrow=c(1,3))
#library(kde1d)
#fit.x <- kde1d(x)
#u<-pkde1d(x, fit.x)
#hist(u,main="",col=2)
#fit.y <- kde1d(y)
#v<-pkde1d(y, fit.y)
#hist(v,main="",col=3)
#plot(cbind(u,v),main="udata",col=4,cex=2)
#par(mfrow=c(1,1))
```

Non-parmetric method (The Kernel estimator)







Another more powerful test of independence

The Cramer-Von Mises test of independence statistic is

$S_n^{\Pi} = \int_{[0,1]^d} n\Big(C_n(u) - \Pi(u)\Big)^2 du.$

$H_0 : Independence \ \ of \ \ X, Y \quad vs \quad H_1 : dependence \ \ of \ \ X, Y$

```
n <- 100; d <- 3;
library(copula)
U <- rCopula(n, frankCopula(2,dim = d))
dist <- indepTestSim(n,p=d, verbose = FALSE)
indepTest(U,d=dist)
```

```
##
## Global Cramer-von Mises statistic: 0.19671 with p-value
## Combined p-values from the Mobius decomposition:
##    0.0004995  from Fisher's rule,
##    0.0004995  from Tippett's rule.
```

See my homepage

To use the copula course materials, go to the web-page

https://hamb8066.github.io/homepage

and click on teaching section. Choose the "Copula Theory and Applications (Msc)".