# Conditional Copula

## Example of a Conditional Copula

Consider the two-dimensional random vector $(X, Y)$ having joint conditional distribution

$$f(x, y|z) = f_X(x)f_Y(y)c(F_X(x), F_Y(y) \mid z)$$

where each marginal is a mixture of univariate normal random variables.

$$f_X(x) = f_Y(x) = \sum_{r=1}^{R} \pi_r \phi(x \mid \mu_r, \sigma_r)$$

We chose the following settings and parameter values: $f_X = f_Y$, $R = 6$, $\mu_{1:6} = (-9, -5.4, -1.8, 1.8, 5.4, 9)$, and $\sigma_r = 1/\sqrt{10}$, $\pi_r = 1/R$.

and the copula is

$$C(u, v \mid x) = u^{1-\alpha(x)}v^{1-\beta(x)}[u^{-\theta\alpha(x)} + v^{-\theta\beta(x)} - 1]^{-1/\theta}$$

where

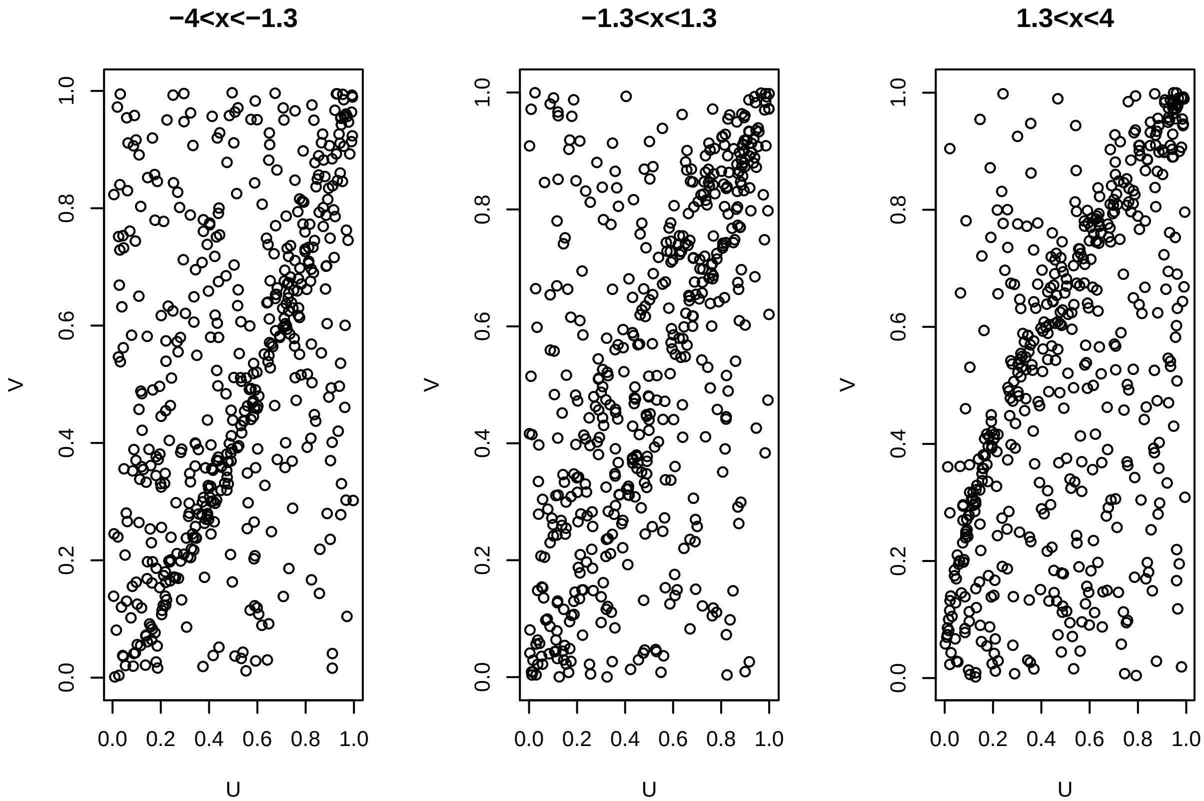
- $\theta = 20$
- $$\alpha(x) = \frac{2}{3} - \frac{1}{4} \times \frac{1}{\exp(x) + 1}$$
- $$\beta(x) = \frac{1}{2} + \frac{1}{4} \times \frac{\exp(x)}{\exp(x) + 1}$$

Draw a sample when $x \in (-4, 4)$

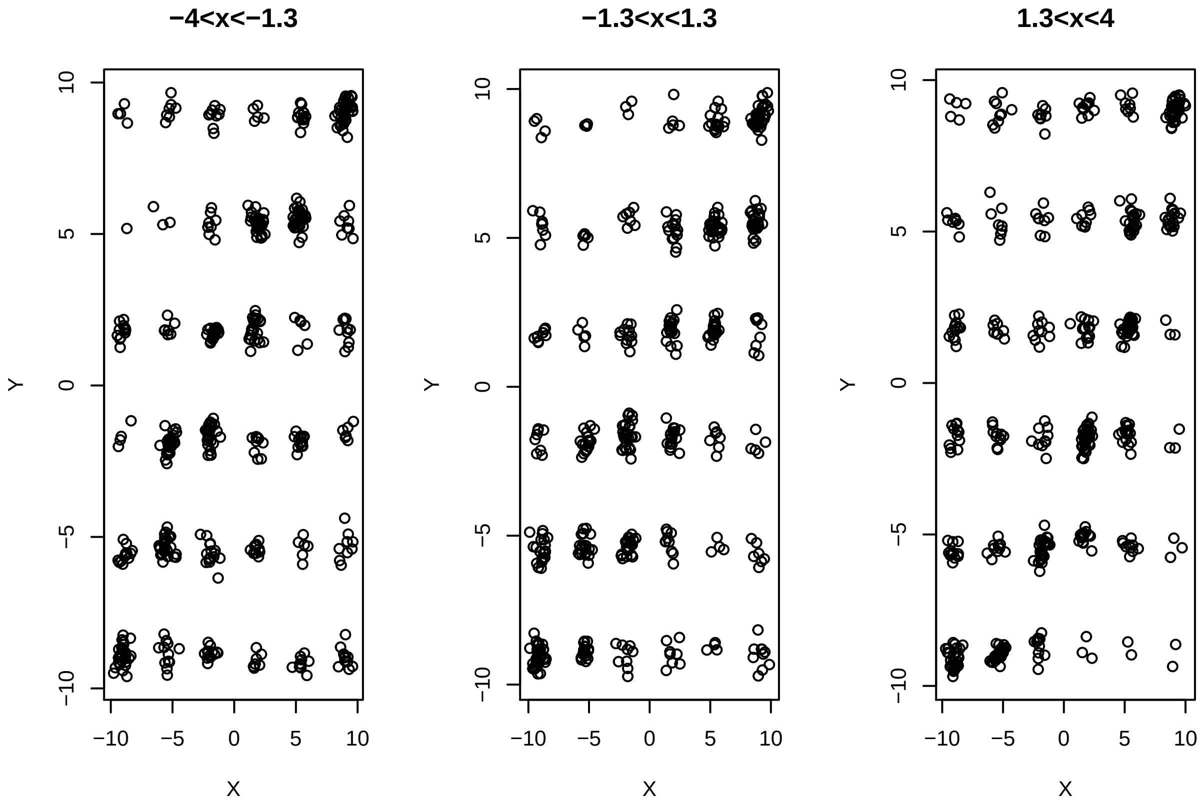
```
set.seed(100)

R=6
mu=c(-9, -5.4, -1.8, 1.8, 5.4, 9)
sigmar=rep(1/sqrt(10), R)
pi=rep(1/R, R)
theta=40

# evaluate the function at the point x, where the components
# of the mixture have weights w, means stored in u, and std deviations
# stored in s - all must have the same length.
F = function(x,w,u,s) sum( w*pnorm(x,mean=u,sd=s) )

# provide an initial bracket for the quantile. default is c(-1000,1000).
F_inv = function(p,w,u,s,br=c(-1000,1000))
{
  G = function(x) F(x,w,u,s) - p
  return( uniroot(G,br)$root )
```

```r
}


l=5
#function alpha(x)
a<-function(x){
  2/3-1/4*(exp(l*x)/(exp(l*x)+1))
}
#function beta(x)
b<-function(x){
  1/2+1/4*(exp(l*x)/(exp(l*x)+1))
}

n=1500
V=runif(n)
W=runif(n)

X=seq(-4,4, length.out = n)

C <- function(u) {
  (1-beta)*u^(1-alpha)*v^(-beta)*(u^(-theta*alpha)+v^(-theta*beta)-1)^(-1/theta)+
    beta*u^(1-alpha)*v^(-beta*(1+theta))*(u^(-theta*alpha)+v^(-theta*beta)-1)^(-1/theta-1)-w
}

U=rep(0,n)
for (i in 1:n) {
  v=V[i]
  w=W[i]
  alpha=a(X[i])
  beta=b(X[i])
  U[i]=uniroot(C, lower = 0, upper = 1)$root
}
```

```r
par(mfrow=c(1,3))
plot(U[1:(1/3*n)], V[1:(1/3*n)], xlab = 'U', ylab='V', main = '-4<x<-1.3')
plot(U[(1/3*n):(2/3*n)], V[(1/3*n):(2/3*n)],xlab = 'U', ylab='V', main = '-1.3<x<1.3')
plot(U[(2/3*n):n], V[(2/3*n):n], xlab = 'U', ylab='V', main = '1.3<x<4')
```




```
X1=sapply(V, F_inv, w=pi, u=mu, s=sigmar)
X2=sapply(U, F_inv, w=pi, u=mu, s=sigmar)

par(mfrow=c(1,3))
plot(X1[1:(1/3*n)], X2[1:(1/3*n)],  xlab = 'X', ylab='Y', main ='-4<x<-1.3')
plot(X1[(1/3*n):(2/3*n)], X2[(1/3*n):(2/3*n)], xlab = 'X', ylab='Y', main =  '-1.3<x<1.3')
plot(X1[(2/3*n):n], X2[(2/3*n):n], xlab = 'X', ylab='Y', main = '1.3<x<4')
```




## Another Example of Conditional Copula

Notation:

- $\Phi$ is the cdf of the standard normal distribution.
- $\phi$ is the pdf of standard normal distribution.
- $\phi_2$ is the pdf of bivariate normal.

Then they have

$$p(u_1, u_2 \mid \mu, \sigma) = \frac{\phi_2(\Phi^{-1}(u_1), \Phi^{-1}(u_2) \mid \boldsymbol{\mu}, \sigma I_2)}{\phi(\Phi^{-1}(u_1))\phi(\Phi^{-1}(u_2))}$$

**A mixture of $p(\boldsymbol{u})$ can be used to estimate any arbitrarily continuous density on $(0,1)^2$.**

$$\tilde{c}(u_1, u_2 \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{j=1}^{J} \pi_j \frac{\phi_2(\Phi^{-1}(u_1), \Phi^{-1}(u_2) \mid \boldsymbol{\mu}_j, \sigma_j I_2)}{\phi(\Phi^{-1}(u_1))\phi(\Phi^{-1}(u_2))}$$

where $\pi_j$s are mixing proportions.

**When $\pi_j$ and $\mu_j$ are conditioned on covariates:**

- $$\pi_j(x_i) = \frac{\exp(\beta_j^T x_i)}{\sum_{j=1}^{J} \exp(\beta_j^T x_i)}$$
- $$\boldsymbol{\mu}_j = \begin{bmatrix} \lambda_{j1}^T x_i \\ \lambda_{j2}^T x_i \end{bmatrix}$$

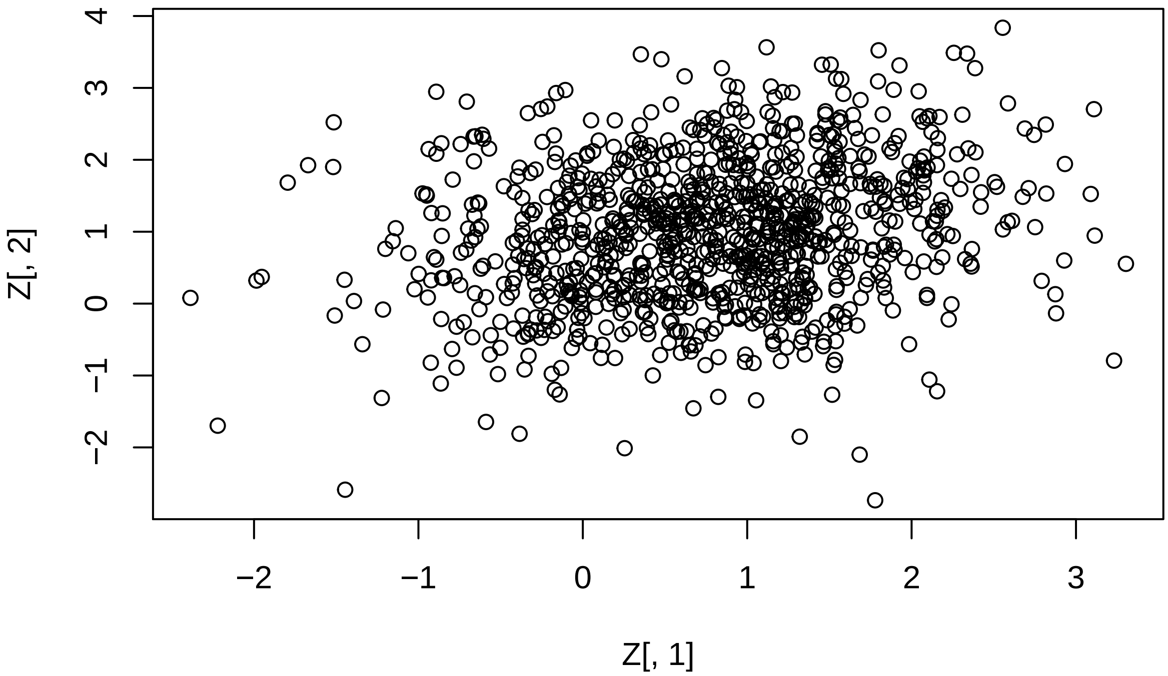
We give an example of a mixture with two components

- $$\beta = \begin{bmatrix} 0.1 \\ 0.3 \end{bmatrix}$$
- $$\lambda_{11} = \begin{bmatrix} 0.1 \\ 0.3 \end{bmatrix}$$
$$\lambda_{12} = \begin{bmatrix} 0.1 \\ 0.4 \end{bmatrix}$$
$$\lambda_{21} = \begin{bmatrix} 0.3 \\ 0.1 \end{bmatrix}$$
$$\lambda_{22} = \begin{bmatrix} 0.4 \\ 0.1 \end{bmatrix}$$
- $\sigma_1 = 0.5$ and $\sigma_2 = 1$.

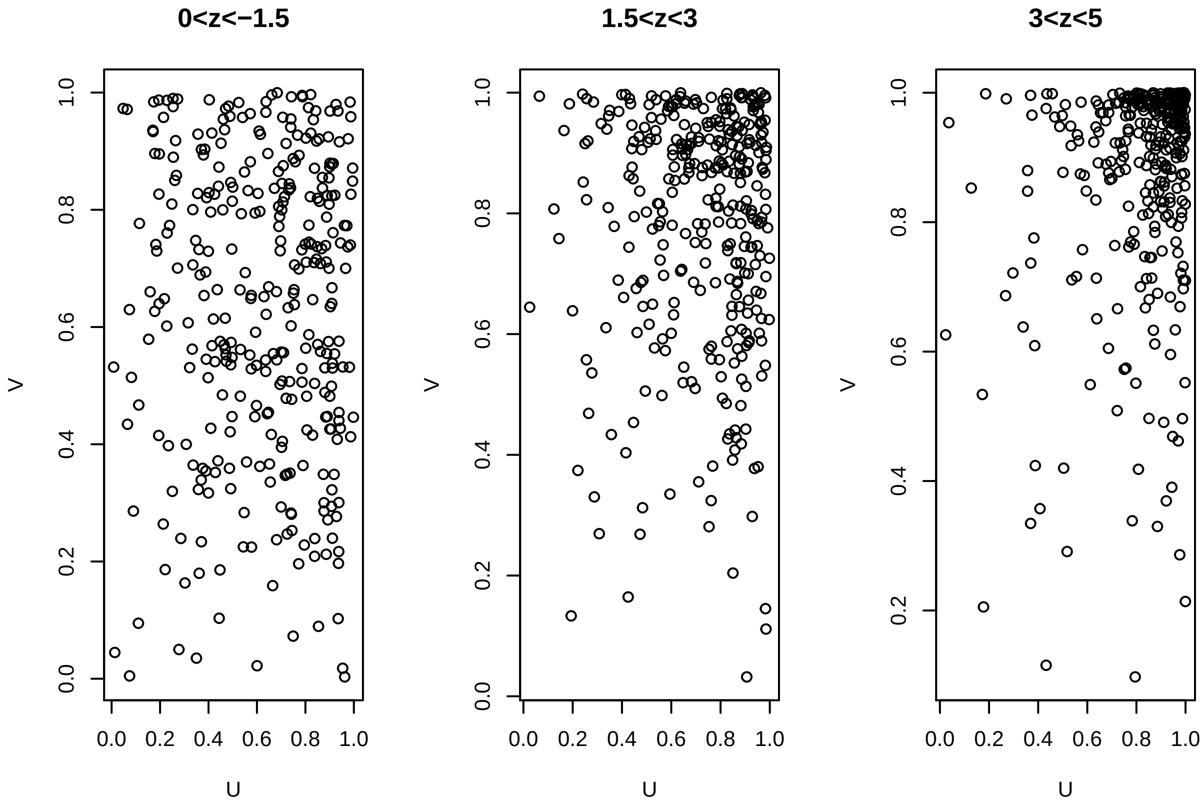
Draw a random sample from above model when $X \in (0, 5)$.

```
set.seed(100)

n=1000
X=seq(0,5, length.out = n)
X=cbind(rep(1, n), X)
delta=c(0.1, 0.3)

pi=cbind(exp(X%*%delta)/(exp(X%*%delta)+1), 1/(exp(X%*%delta)+1))
#Generate "latent" variable indicating which component
h=t(apply(pi, 1,  rmultinom, n=1, size=1))


#The first bivariate normal component
beta11=c(0.1,0.3)
beta12=c(0.1,0.4)
sigma1=0.5
I=matrix(c(1,0,0,1), nrow = 2)
mu1=cbind((X%*%beta11)[,1], (X%*%beta12)[,1])

#Sample from the first component
Z1=t(apply(mu1, 1, rmvnorm, n=1, sigma=sigma1*I))



#The second bivariate normal component
beta21=c(0.3, 0.1)
beta22=c(0.4, 0.1)
sigma2=1
I=matrix(c(1,0,0,1), nrow = 2)
#mean vector
mu2=cbind((X%*%beta21)[,1], (X%*%beta22)[,1])
#sample from the second component
Z2=t(apply(mu2, 1, rmvnorm, n=1, sigma=sigma2*I))
```

```r
#use h to select from Z1 and Z2
Z=matrix(rep(0, 2*n), nrow = n)
for (i in 1:n) {
Z[i,]=Z1[i,]*h[i,1]+Z2[i,]*h[i,2]
}

plot(Z[,1], Z[,2])
```




```r
dat=data.frame(Z[,1], Z[,2], X[,2])
colnames(dat)=c("Y1", "Y2", "X")
write.table(dat, "dat")

#Get copula data
u=t(apply(Z, 1, pnorm))
U=u[,1]
V=u[,2]

par(mfrow=c(1,3))
plot(U[1:(1/3*n)], V[1:(1/3*n)], xlab = 'U', ylab='V', main = '0<z<-1.5')
plot(U[(1/3*n):(2/3*n)], V[(1/3*n):(2/3*n)],xlab = 'U', ylab='V', main = '1.5<z<3')
plot(U[(2/3*n):n], V[(2/3*n):n], xlab = 'U', ylab='V', main = '3<z<5')
```




```r
#Set up marginal distribution
R=6
mu=c(-9, -5.4, -1.8, 1.8, 5.4, 9)
sigmar=rep(1/sqrt(10), R)
pi=rep(1/R, R)
theta=40

# evaluate the function at the point x, where the components
# of the mixture have weights w, means stored in u, and std deviations
# stored in s - all must have the same length.
F = function(x,w,u,s) sum( w*pnorm(x,mean=u,sd=s) )

#Marginal quantile function
# provide an initial bracket for the quantile. default is c(-1000,1000).
F_inv = function(p,w,u,s,br=c(-1000,1000))
{
  G = function(x) F(x,w,u,s) - p
  return( uniroot(G,br)$root )
}


#Get the orginal data
X1=sapply(V, F_inv, w=pi, u=mu, s=sigmar)
X2=sapply(U, F_inv, w=pi, u=mu, s=sigmar)

par(mfrow=c(1,3))
plot(X1[1:(1/3*n)], X2[1:(1/3*n)],  xlab = 'X', ylab='Y', main = '0<z<-1.5')
plot(X1[(1/3*n):(2/3*n)], X2[(1/3*n):(2/3*n)], xlab = 'X', ylab='Y', main = '1.5<z<3')
```

```
plot(X1[(2/3*n):n], X2[(2/3*n):n], xlab = 'X', ylab='Y', main = '3<z<5')
```