Métodos estadísticos multivariados aplicados al análisis de riesgos financieros

Jorge de la Vega Góngora

Departamento de Estadística
Instituto Tecnológico Autónomo de México

9 de octubre de 2023

Introducción

Introducción

- El objetivo de este curso es ofrecer algunas herramientas estadísticas multivariadas que son útiles para el análisis de riesgos junto con algunas de sus aplicaciones.
- En particular, una forma de medir y cuantificar la dependencia de las variables de mercado se puede medir a través de *cópulas* y de correlaciones.
- Las cópulas son un marco de referencia para definir definir distribuciones multivariadas y para modelar datos multivariados.
- A partir de ellas, se pueden modelar o simular el comportamiento de rendimientos, factores de riesgo y en general de cualquier incertidumbre.

Día 1: Cópulas y dependencia

Dependencia

- **X**, **Y** $\in \mathbb{R}^d$ vectores aleatorios son independientes si

$$P(X \in A, Y \in B) = P(\mathbf{X} \in A)P(\mathbf{Y} \in B) \qquad A, B \in \mathcal{F}$$

  Equivalentemente, en términos de probabilidad condicional

$$P(\mathbf{X} \in A | \mathbf{Y} \in B) = P(\mathbf{X} \in A)$$

- Dependencia total se da cuando existe una función biyectiva $G : \mathbb{R}^d \to \mathbb{R}^d$ tal que $\mathbf{Y} = G(\mathbf{X})$ c.s.
- El caso intermedio: no independiente y no completamente dependiente, se puede cuantificar la dependencia a través de una *cópula*.

Definición de cópula

Cópulas

Una cópula es una función de distribución conjunta $C[0,1]^n \to [0,1]$ cuyas distribuciones marginales son todas $\mathcal{U}\,(0,1)$.

Dada la definición, podemos construir una cópula de la siguiente manera:

- Consideren dos variables aleatorias (X_1, X_2) con distribución conjunta F y marginales $F_1(x_1)$ y $F_2(x_2)$.
- Definan

$$C(u,v) = F(F_1^{-1}(u), F_2^{-1}(v)) \quad \forall\, u,v \in [0,1]$$

Entonces C es una cópula:

$$\begin{aligned} C(u,v) &= F(F_1^{-1}(u), F_2^{-1}(v)) \\ &= P(X \le F_1^{-1}(u), Y \le F_2^{-1}(v)) \\ &= P(F_1(X) \le u, F_2(Y) \le v) \\ &= P(U \le u, V \le v) \end{aligned}$$

Y claramente las marginales de $C(u,v)$ son uniformes.

Cópulas

- Sabemos que $F(X) \sim \mathcal{U}\,(0,1)$ para cualquier v.a. X. Entonces, por definición, la función $C(F_1(x_1), F_2(x_2))$ es una cópula.
- Noten que como C es una distribución y haciendo $u = F_1(x_1)$ y $v = F_2(x_2)$:

$$\begin{aligned} C(F_1(x_1), F_2(x_2)) = C(u,v) &= P(U \leq u, V \leq v) \\ &= P(F_1(X_1) \leq u, F_2(X_2) \leq v) \\ &= P(X_1 \leq F_1^{-1}(u), X_2 \leq F_2^{-1}(v)) \\ &= F_X(F_1^{-1}(u), F_2^{-1}(v)) = F_X(x_1, x_2) \end{aligned}$$

  Esto es, $F_X(x_1, x_2) = C(F_1(x_1), F_2(x_2))$. De este modo, F_X se "descompone" en dos partes: una cópula C que contiene información de las dependencias de $X = (X_1, X_2)$, y las marginales.
- Entonces una cópula caracteriza la dependencia, y sólo la dependencia entre los componentes de una distribución multivariada.
- Lo anterior se puede resumir en el teorema de Sklar (1959).

Teorema de Sklar (1959)

Teorema de Sklar

- Sea F una función de distribución conjunta con marginales $F_1, \dots, F_p$. Entonces $\exists$ una cópula $C : [0,1]^p \to [0,1] \ni \forall \mathbf{x} \in \mathbb{R}^p$

$$F(x_1, \dots, x_p) = C(F_1(x_1), \dots, F_p(x_p))$$

  Además, si las marginales son continuas, entonces C es única.

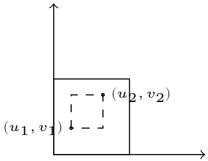
- Conversamente, si C es una cópula y $F_1, \dots, F_p$ son funciones de distribución, entonces F como se definió arriba, es una distribución con marginales $F_1, \dots, F_p$.

Abe Sklar

Cópulas y sus propiedades I

- Consideremos por simplicidad $n = 2$. Las propiedades de la cópula son las propiedades usuales de cualquier función de distribución conjunta.
  - $C(u, 1) = u$ y $C(1, v) = v \; \forall u, v \in [0, 1]$.
  - $C(u, 0) = C(0, v) = 0 \; \forall u, v \in [0, 1]$.
  - El área (volumen si $n > 2$) de un cuadrado (cubo) en el cuadro unitario (hipercubo) es positiva: si $(u_1, v_1), (u_2, v_2) \in [0, 1]^2$ y $u_2 \geq u_1$, $v_2 \geq v_1$ entonces

$$C(u_2, v_2) - C(u_1, v_2) - C(u_2, v_1) + C(u_1, v_1) \geq 0$$



  - Una cópula es invariante bajo transformaciones estrictamente crecientes de las distribuciones marginales: si g_i es estrictamente creciente y $Y_i = g_i(X_i)$, entonces **Y** y **X** tienen la misma cópula.
- Las propiedades anteriores caracterizan a las cópulas.

Cópulas y sus propiedades II

Otras propiedades importantes a mencionar son las siguientes:

- La densidad de un vector $\mathbf{X} = (X_1, \ldots, X_n)$ se puede expresar en términos de la cópula como

$$f_{\mathbf{X}}(x_1, \ldots, x_n) = c_{\mathbf{X}}(F_{X_1}, \ldots, F_{X_n}) f_{X_1}(x_1) \cdots f_{X_n}(x_n)$$

  en donde $c_{\mathbf{X}}(x_1, \ldots, x_n) = \frac{\partial^n}{\partial x_1 \cdots \partial x_n} C_{\mathbf{X}}(x_1, \ldots, x_n)$ y f_{X_i} es la densidad marginal de X_i.

Uso de las cópulas

- En resumen, usamos las cópulas para especificar en un proceso de dos etapas, una distribución conjunta:
  1. Especificamos el tipo de distribuciones marginales que se desea conjuntar
  2. Especificamos la distribución cópula.
- Como las cópulas sólo especifican la estructura de la dependencia, diferentes cópulas producen diferentes distribuciones conjuntas cuando se aplican a las mismas marginales.
- Es importante notar que para un par de marginales, se pueden usar infinitas cópulas para crear una distribución conjunta, así que es necesario 'ajustar' la que más se apegue al comportamiento de dependencia entre las variables.

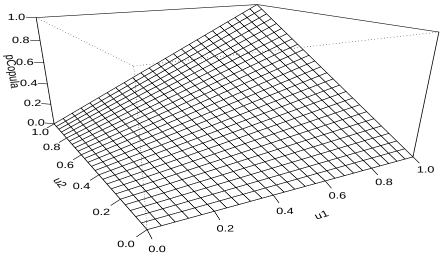
Ejemplo: Cópula de independencia.

```r
library(copula)
persp(indepCopula(), pCopula, theta = -30, phi = 20)
```

Definamos $C : [0,1]^2 \to [0,1]$ como $C(u_1, u_2) = u_1 u_2$. Entonces, Si F_1 y F_2 son distribuciones,

$$C(F_1(x_1), F_2(x_2)) = F_1(x_1)F_2(x_2) = F(x_1, x_2)$$

Entonces $X_1 \perp\!\!\!\perp X_2$.

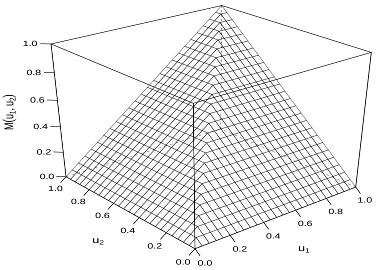


Cópula de co-monotonicidad I

```r
n.grid <- 26
u <- seq(0,1,length.out=n.grid)
grid <- expand.grid("u[1]"= u,"u[2]"= u)
M <- function(u) apply(u,1,min) #Cota superior M
x.M <- cbind(grid,"M(u[1],u[2])" = M(grid)) #Evalua M en el grid
wireframe2(x.M)
```

Si $U \sim \mathcal{U}(0,1)$ y $\mathbf{U} = (U, U)$ (dos copias de U, así que las variables son completamente dependientes)

$$
\begin{aligned}
C^M(u_1, u_2) &= P(U \leq u_1, U \leq u_2) \\
&= P(U \leq \text{mín}\{u_1, u_2\}) \\
&= \text{mín}\{u_1, u_2\}
\end{aligned}
$$

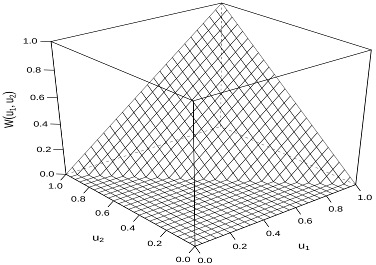


Cópula de contra-monotonicidad

```r
n.grid <- 26
u <- seq(0,1,length.out=n.grid)
grid <- expand.grid("u[1]"= u,"u[2]"= u)
W <- function(u) pmax(0,rowSums(u)-1) #cota inferior W
x.W <- cbind(grid,"W(u[1],u[2])" = W(grid)) #Evalua W en el grid
wireframe2(x.W)
```

Si $\mathbf{U} = (U, 1-U)$

$$
\begin{aligned}
C^{CM}(u_1,u_2) &= P(U \leq u_1, 1-U \leq u_2) \\
&= P(1-u_2 \leq U \leq u_1) \\
&= \text{máx}\{u_1+u_2-1, 0\}
\end{aligned}
$$



Cotas para cópulas

Un resultado importante es que las cópulas de co- y contra-monotonicidad son los extremos que cualquier cópula puede tomar (es decir, son cotas mínima y máxima):

Cotas inferior y superior de Fréchet-Hoeffing

Dada una cópula C, $\forall u_1, \dots, u_n \in [0,1]$:

$$max\{u_1 + \cdots + u_n + 1 - n, 0\} \leq C(u_1, \dots, u_n) \leq min\{u_1, \dots, u_n\}$$

Nota: no es posible tener una cópula de contra-monotonicidad en más de dos dimensiones, pero la cota inferior es alcanzada sólo puntualmente, pero no es en sí misma una cópula para $n \geq 3$

Cópula Gaussiana I

Si $Z \sim \mathcal{N}_2\left(\mathbf{0}, \Sigma\right)$ con marginales $Z_1 \sim \mathcal{N}\left(0, \sigma^2\right)$ y $Z_2 \sim \mathcal{N}\left(0, \sigma^2\right)$ y $\rho(Z_1, Z_2) = \rho$, entonces

$$C(u_1, u_2) = \Phi_{\sum}(\Phi^{-1}(u_1), \Phi^{-1}(u_2))$$

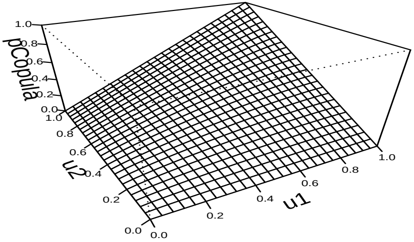
es una cópula.
La fórmula explícita de la cópula es:

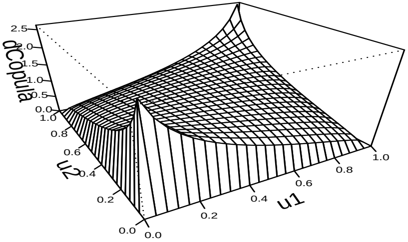
$$C_\rho(u_1, u_2) = \int_{-\infty}^{x_1} \int_{-\infty}^{x_2} \frac{1}{2\pi\sqrt{1-\rho^2}} exp\left\{\frac{s^2 + t^2 - 2\rho st}{2(1-\rho^2)}\right\} ds\, dt$$

con $x_i = \phi^{-1}(u_i)$.

```r
norm.cop <- normalCopula(0.4)
par(mar = c(4, 4, .1, .1),mfrow=c(1,2))
persp(norm.cop,pCopula,cex.axis=0.5)
persp(norm.cop,dCopula,cex.axis=0.5)
```

Cópula Gaussiana II





Familia Arquimediana de Cópulas

Las copulas arquimedianas son más útiles cuando la dependencia entre pares de variables tienen dependencias similares, debido a que cumplen con la propiedad de intercambiabilidad (cualquier permutación de las variables da la misma cópula).

Cópulas Arquimedianas

Una cópula Arquimediana con función generadora ϕ tiene la forma:

$$C(u_1, \ldots, u_p) = \phi^{-1}(\phi(u_1) + \cdots + \phi(u_1))$$

donde la función generadora ϕ satisface:

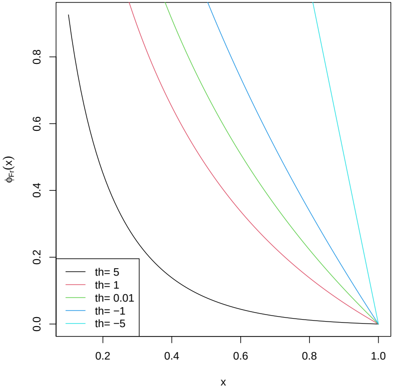
1. $\phi : [0,1] \to [0,\infty]$ es continua, convexa y estrictamente decreciente.
2. $\phi(0) = \infty$
3. $\phi(1) = 0$

Diferentes generadores generan diferentes cópulas. Algunas de las más comunes son las que siguen a continuación.

Cópula de Frank I

La cópula de Frank tiene función generadora:

$$\phi^{Fr}(u) = -\log\left\{\frac{e^{-\theta u}-1}{e^{-\theta}-1}\right\}, \theta \in \mathbb{R}$$



Cópula de Frank I

Verificando las propiedades de cópula arquimediana:

- $\phi^{Fr}(0) = -\log\left\{\frac{1-1}{e^{-\theta}-1}\right\} = -log(0) = \infty$
- $\phi^{Fr}(1) = -\log\left\{\frac{e^{-\theta}-1}{e^{-\theta}-1}\right\} = -\log(1) = 0$
- Si $y = -log\left\{\frac{e^{-\theta u}-1}{e^{-\theta}-1}\right\}$, entonces

$$
\begin{aligned}
e^{-y} &= \frac{e^{-\theta u}-1}{e^{-\theta}-1} \\
(e^{-\theta}-1)e^{-y}+1 &= e^{-\theta u} \\
u &= -\frac{1}{\theta}\log\left\{(e^{-\theta}-1)e^{-y}+1\right\}
\end{aligned}
$$

- Por lo tanto: $\phi Fr^{-1}(y) = -\frac{1}{\theta}\log\left\{(e^{-\theta}-1)e^{-y}+1\right\}$

Cópula de Frank II

Con ambas funciones $\phi^{Fr^{-1}}$ y ϕ^{Fr}, hay que resolver la ecuación:

$$\phi^{Fr^{-1}}[\phi^{Fr}(u_1) + \phi^{Fr}(u_2)]$$

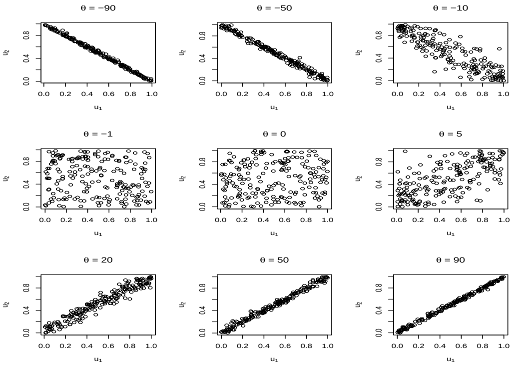
La cópula que se obtiene (en el caso bidimensional) es la siguiente:

$$C^{Fr}(u_1, u_2) = -\frac{1}{\theta} \log \left(1 + \frac{(e^{-\theta u_1} - 1)(e^{-\theta u_2} - 1)}{e^{-\theta} - 1} \right)$$

A continuación se muestra una simulación para distintos valores de θ

```
library(copula)
set.seed(1)
par(mfrow = c(3,3))
theta <- c(-90, -50, -10, -1, 0 , 5, 20, 50, 90)
for(i in 1:9){
U <- rCopula(n = 200,
copula = archmCopula(family = "frank", param = theta[i]))
plot(U,
     xlab = expression(u[1]), ylab = expression(u[2]),
         main = eval(substitute(expression(paste(theta, " = ", j)), list(j = as.character(theta[i])))))
}

parameter at boundary ==> returning indepCopula()
```

Cópula de Frank III

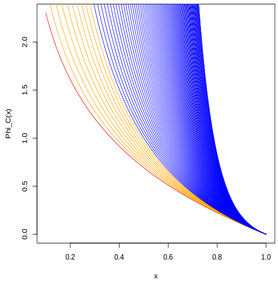


Cópula de Clayton I

```r
Phi_C <- function(u, theta = 0.0001)(u^(-theta)-1)/theta
unitario <- seq(0, 1, by = 0.1)

curve(Phi_C, from = 0.1, to = 1, col = "red")
for(theta in seq(0, 10, length = 100)){
curve(Phi_C(x, theta = theta), from = 0.001, to = 1, add = T,
col = ifelse(theta > 1, "blue", "orange"), ylab = expression(Phi[C](x)))
}
```

La cópula de Clayton tiene función generadora: $\phi^C(u) = \frac{u^{-\theta}-1}{\theta}, \theta > 0$ y de aquí se obtiene la ecuación de la cópula:

$$C^C(u_1, u_2) = (u_1^{-\theta} + u_2^{-\theta} - 1)^{-1/\theta}$$

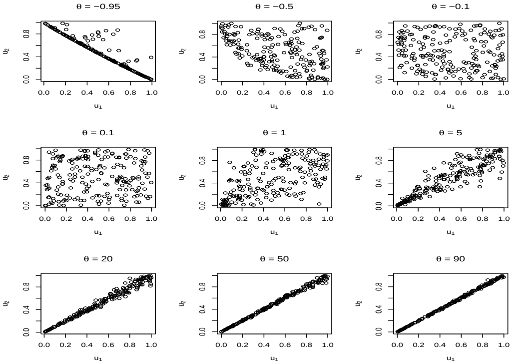


Cópula de Clayton II

Simulando ahora respecto a esta cópula

```
library(copula)
set.seed(1)
par(mfrow = c(3,3))
theta <- c(-.95, -0.5, -0.1, 0.1, 1 , 5, 20, 50, 90)
for(i in 1:9){
U <- rCopula(n = 200,
copula = archmCopula(family = "clayton", param = theta[i]))
plot(U,
        xlab = expression(u[1]), ylab = expression(u[2]),
        main = eval(substitute(expression(paste(theta, " = ", j)), list(j = as.character(theta[i])))))
}
```

Cópula de Clayton III

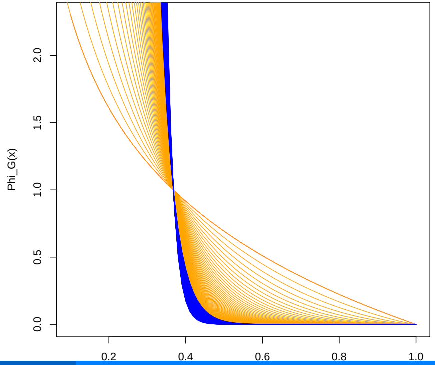


Cópula de Gumbel I

```r
Phi_G <- function(u,theta=1)(-log(u))^theta

curve(Phi_G, from = 0.1, to = 1, col = "red")
for(theta in seq(1, 20, length = 100)){
        curve(Phi_G(x, theta = theta), from = 0.001,to=1,add=T,
        col = ifelse(theta > 10,"blue","orange"))
}
```

La función generadora de la cópula de Gumbel tiene la forma: $\phi^G(u) = (-\log u)^\theta, \theta \geq 1$ y la ecuación de la cópula queda de la siguiente manera:

$$C^G(u_1, u_2) = \exp\{-[(-\log u_1)^\theta + (-\log u_2)^\theta]\}$$

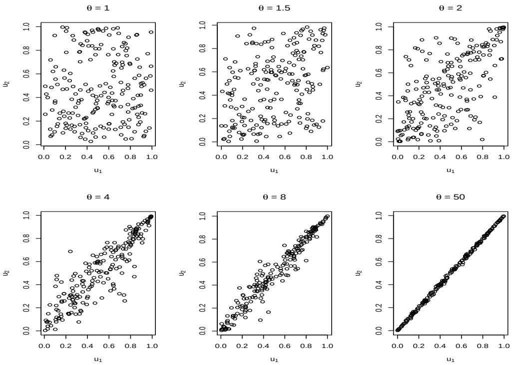


Cópula de Gumbel II

Simulando muestras de la cópula de Gumbel:

```
library(copula)
set.seed(1)
par(mfrow = c(2,3))
theta <- c(1, 1.5, 2, 4, 8, 50)
for(i in 1:6){
U <- rCopula(n = 200,
                         copula = archmCopula(family = "gumbel", param = theta[i]))
plot(U,
xlab = expression(u[1]), ylab = expression(u[2]),
main = eval(substitute(expression(paste(theta, " = ", j)), list(j = as.character(theta[i])))))
}

parameter at boundary ==> returning indepCopula()
```

Cópula de Gumbel III



Cópulas de Valores Extremos

Sea $A : [0,1] \to [1/2, 1]$ una función convexa que satisface la siguiente condición para $w \in [0,1]$: $\text{máx}\{w, 1-w\} \leq A(x) \leq 1$. Las familia de cópulas de valores extremos (Pickands, 1981) se define a partir de la función A como:

$$C(u,v) = \exp\left[\log(uv) A\left(\frac{\log(v)}{\log(uv)}\right)\right]$$

Algunos ejemplos de casos particulares:

- $A(w) = 1$ es la cópula de independencia.
- $A(w) = \text{máx}\{w, 1-w\}$ es la cópula comonotónica.
- $A(w) = \left[w^\theta + (1-w)^\theta\right]^{1/\theta}$, $\theta \geq 1$ es la cópula de Gumbel.
- $A(w) = 1 - \left[w^{-\theta} + (1-w)^{-\theta}\right]^{-1/\theta}$, $\theta \geq 0$ es la cópula de Galambos.

Hay muchísimas otras que se han desarrollado para aplicaciones en seguros y finanzas.

Día 2: Simulación de cópulas

Método para construir variables dependientes via simulación

- Se puede dar un método general para construir variables dependientes con distribuciones generales: $\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim \begin{pmatrix} F \\ G \end{pmatrix}$, utilizando una cópula dada. Aquí se muestra para dos dimensiones pero funciona para cualquier dimensión.
- Por ejemplo, para el caso de la cópula gaussiana, el método es el siguiente:
  1. Generar $(Z_1, Z_2) = \mathbf{Z} \sim \mathcal{N}_2(\mathbf{0}, \Sigma(\rho))$. Entonces Z_1 y Z_2 están relacionadas, $cor(Z_1, Z_2) = \rho$.
  2. Obtener $(u_1, u_2) = \mathbf{u} \sim (\Phi(Z_1), \Phi(Z_2))$.
  3. Obtener $(X_1, X_2) = \mathbf{X} \sim (F^{-1}(u_1), G^{-1}(u_2))$

  Bajo este esquema,las normales dependientes permiten generar X_1 y X_2 dependientes. Sin embargo, $cor(X_1, X_2) \neq \rho$, ya que se aplicaron transformaciones no lineales. Entonces, es necesario introducir nuevas medidas de dependencia que sean invariantes ante transformaciones no lineales que veremos más adelante.

Ejemplo práctico

- Queremos estudiar el comportamiento conjunto de los precios de dos instrumentos en el mercado financiero. Cada precio puede tener su propio comportamiento, originado por las diferentes fuentes de variación.
- Por ejemplo, si ambos instrumentos son derivados sobre el mismo subyacente, puede ser que su comportamiento se deba las mismas fuerzas del mercado. Sin embargo, si los subyacentes son de diferentes mercados (hipotecario y de tasas de interés, por ejemplo) entonces pueden responder a diferentes fuentes de variación. Supongamos:
  - Los precios se comportan como dos lognormales de manera independiente.
  - Se desea simular el comportamiento que han tenido en los pasados 1,000 días.
  - Para generar lognormales, generamos 1,000 pares de variables normales independientes, y las exponenciamos.
  - Asumimos una variabilidad de los precios conocida de $\sigma = 0.5$.

Ejemplo: Primer caso: independencia

La siguiente gráfica muestra el comportamiento conjunto de estas dos variables independientes.

```
set.seed(1)
n <- 1000
sigma <- 0.5 # asumida
#Matriz de covarianzas
Sigma <- sigma^2 * diag(2)
Sigma
```

```
     [,1] [,2]
[1,] 0.25 0.00
[2,] 0.00 0.25
```

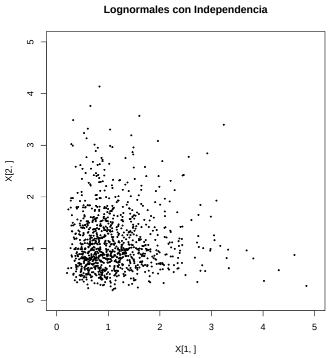
```
#Genera las muestras de dos normales
z <- matrix(rnorm(2*n),nrow=2)
#Transforma para escalar las variables
Y <- sigma*diag(2) %*% z
X <- exp(Y) #X tiene distribución lognormal
dim(X)
```

```
[1]    2 1000
```

```
X[1:2,1:3]
```

```
         [,1]      [,2]      [,3]
[1,] 0.731084 0.6584845 1.1791029
[2,] 1.096169 2.2202957 0.6634948
```

```
par(pty="s") #gráfico cuadrado
plot(X[1,], X[2,], xlim=c(0,5), ylim=c(0,5), pch=16,
cex=0.5, main="Lognormales con Independencia")
```



Segundo caso: Introducción de dependencia lineal.

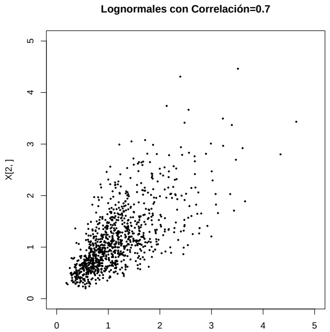
Podemos introducir dependencia entre las variables a través de un coeficiente de correlación en las normales bivariadas. En este caso se puede observar que valores mayores o menores de las dos variables tienden a estar más asociados que en el caso anterior.

```
rho = 0.7 #correlación
Sigma <- sigma^2*matrix(c(1,rho,rho,1),nrow=2)
Sigma
```

```
      [,1]  [,2]
[1,] 0.250 0.175
[2,] 0.175 0.250
```

```
# Obten la matriz raíz cuadrada B de la matriz
# definida positiva Sigma
e <- eigen(Sigma)
v <- e$vectors
B <- v %*% diag(sqrt(e$values)) %*% t(v)
z <- matrix(rnorm(2*n),nrow=2)
Y <-  B %*% z #Transforma para escalar las variables
X <- exp(Y)
```

```
par(pty="s") #haz el gráfico cuadrado
plot(X[1,], X[2,], xlim=c(0,5), ylim=c(0,5), pch=16, cex=0.5,
main="Lognormales con Correlación=0.7")
```

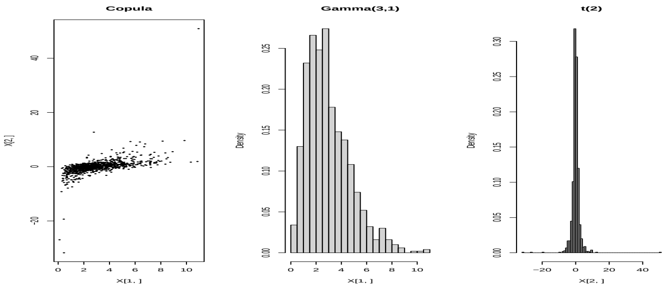


Caso más general: diferentes familias de distribuciones

- En el ejemplo anterior se puede incluir diferentes lognormales, haciendo la transformación de y a X de la manera apropiada.
- Pero ¿si las distribuciones de los precios de los activos no es la misma? Por ejemplo, uno de los activos puede provenir del mercado cambiario, presentado mayor volatilidad y con colas pesadas en sus variaciones. En este caso, más que la distribución lognormal, podría ser más importante una distribución de colas pesadas, como la distribución t.
- Una siguiente extensión es aplicar la cópula gaussiana. Partimos de los mismos supuestos de correlación, pero ahora supondremos que un precio es $\mathcal{G}\left(3,1\right)$ y el otro es $t_{(2)}$. Ahora tenemos que obtener uniformes a partir de Z y luego tomar las inversas de las distribuciones marginales.

Generación de variables aleatorias usando cópula gaussiana I

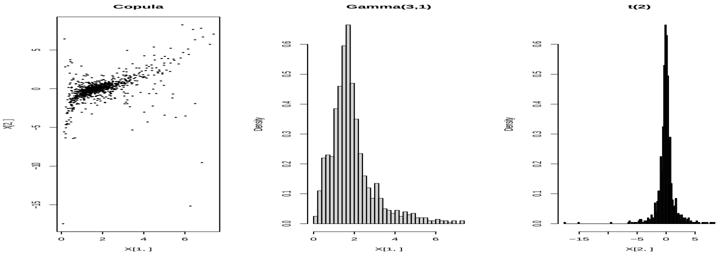
```
set.seed(3); z <- matrix(rnorm(2*n), nrow = 2)
Y <-  B %*% z #Transforma para escalar las variables
U <- pnorm(Y, sd = 0.5); X <- rbind(qgamma(U[1,], 3, 1), qt(U[2,], 2))
par(mfcol = c(1, 3)); plot(X[1,], X[2,], pch = 16,cex = 0.5, main = "Copula")
hist(X[1,], main = "Gamma(3,1)",breaks = 30, prob = T)
hist(X[2,], main = "t(2)", breaks = 100,prob = T); cor(X[1,],X[2,])
```




```
[1] 0.5327403
```

¿Qué pasa si hacemos el mismo ejercicio pero con una cópula diferente? Consideremos ahora la cópula t que también se puede parametrizar con la correlación y con los grados de libertad.

```
library(mvtnorm) #para generar t multivariada
set.seed(3); TT <- t(rmvt(n = 1000, sigma = Sigma, df = 1)) #genera una dist. t(1)
U  <- pt(TT, df = 1) #distribución t(1) para uniformes
X <- rbind(qgamma(U[1,],2,1), qt(U[2,],2))
par(mfcol = c(1,3)); plot(X[1,], X[2,], pch = 16, cex = 0.5, main = "Copula")
hist(X[1,], main = "Gamma(3,1)", breaks = 30, prob = T)
hist(X[2,], main = "t(2)", breaks = 100, prob = T);      cor(X[1,], X[2,])
```




```
[1] 0.4727318
```

Método para construir variables dependientes

Cópula t.

Si sustituimos a $Z = (Z_1, Z_2)$ por una variable bivariada $T_{(\Sigma,g)} = (t_1, t_2)$ donde g son los grados de libertad, podemos construir la cópula C^* del mismo modo que en el ejercicio previo.

- ¿Cuál es la diferencia entre usar una cópula Gaussiana y una cópula t? Ambas tienen la misma correlación.
- la diferencia está en la estructura de la dependencia, lo cual:
  - comprueba que la estructura de la dependencia es mucho más que la simple covarianza, y
  - se requiere poder estimar una cópula específica a las estructuras de dependencia, y por eso tiene sentido considerar *familias de cópulas* para diferentes estructuras de dependencia:
    - Cópulas elípticas: Gaussianas, Student.
    - Cópulas Arquimedianas: Frank, Clayton, Gumbel
    - Cópulas de valores extremos.

Simulación de cópulas bivariadas

- El siguiente método para generar muestras de cópulas bivariadas fue propuesto por Nelsen (2006). La función de distribución condicional de V, dado $U = u$, se puede escribir como:

$$\begin{aligned} C_{V|U}(u,v) = P(V \leq v|U = u) &= \frac{C(u,v)}{C(u,1)} \\ &= \lim_{\epsilon\to 0} \frac{C(u+\epsilon, v) - C(u,v)}{\epsilon} \\ &= \frac{\partial C(u,v)}{\partial u} \end{aligned}$$

- Basado en lo anterior, un algoritmo general para extraer muestras de una cópula $C(u,v)$ es el siguiente:

Muestreo de observaciones de una cópula bivariada

1. Extrae dos variadas uniformes independientes u_1 y v_2.
2. Hacer $u_2 = C^{-1}_{V|U=u_1}(u_1, v_2)$
3. El vector (u_1, u_2) es el generado de la cópula C.

Ejemplo I

En muchas aplicaciones prácticas se utiliza la cópula general de Farlie-Gumbel-Morgensten (FMG):

$$C_\theta(u,v) = uv + \theta u(1-u)v(1-v)$$

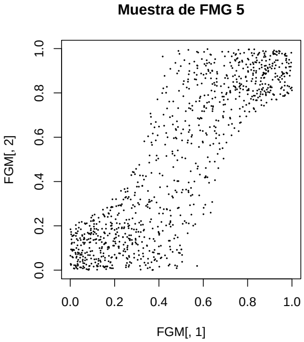
Evaluando $C_{V|U}(u,v) = v(1+\theta(1-2u)) + \theta(1-2u)v$. Para esta cópula el algoritmo propuesto es de la forma siguiente (hay que hacer el ejercicio de inversión):

1. extrae dos uniformes independientes (u_1, v_2)
2. Define: $u_2 = \frac{2v_2}{A+B}$, donde $A = 1+\theta(1-2u_1)$, $B = \sqrt{A^2 - 4(A-1)v_2}$

Por ejemplo, para $\theta = 5$:

```
theta <- 5
c1 <- cbind(runif(1000),runif(1000)) # genera u y v
A <- 1+theta*(1-2*c1[,1])
B <- sqrt(A^2-4*(A-1)*c1[,2])
FGM <- cbind(c1[,1],2*c1[,2]/(A+B))
par(pty="s"); plot(FGM[,1],FGM[,2],pch=16,cex=0.3,main=paste("Muestra de FMG",theta))
```

Ejemplo II



Características del paquete `copula`

- El paquete `copula` desarrollada por Jun Yan (2006) provee una plataforma para la modelación multivariada de cópulas.
- Incluye, para las cópulas elípticas (normal, t), y arquimedianas (Clayton, Frank, Gumbel), los siguientes métodos
  - Evaluación de densidad/distribución
  - Generación de muestras de la cópula
  - Visualización
  - Ajuste de modelos basados en cópulas, utilizando máxima verosimilitud.
  - La cópula de valores extremos sólo está implementada para el caso bivariado.
- Otro paquete importante es el paquete `fCopulae` de Tobias Setz, como parte de los paquetes financieros de `Rmetrics`, que complementa los modelos de cópula para incluir las cópulas de Valor Extremo (se ve en la siguiente lámina), y la cópula empírica (con la definición usual de distribución empírica).

Clases definidas en el paquete `copula` I

Hay básicamente dos clases de objetos:

`copula` para definir cópulas:

$$C(u_1, \dots, u_p) = F(F_1^{-1}(u_1), \dots, F_p^{-1}(u_p))$$

donde las funciones F y F_i son dadas.

`mvdc` para definir distribuciones multivariadas a partir de cópulas:

$$F(x_1, \dots, x_p) = C(F_1(x_1), \dots, F_p(x_p))$$

donde C y F_i son dadas.

Clase `copula`

La clase `copula` considera las subclases:

- `ellipCopula`, incluye `normalCopula` y `tCopula`. Como se vió en clase, basta con definir la matriz de correlaciones como parámetro de las dos familias, y en el caso de la t también se requieren los grados de libertad (`df`).
- La matriz de correlaciones define la estructura de dependencia, y se pueden usar las siguientes configuraciones:

  `ar1:` especificación para dependencias autorregresivas, por ejemplo, con $p = 3$:

  $$\begin{pmatrix} 1 & \rho_1 & \rho_1^2 \\ \rho_1 & 1 & \rho_1 \\ \rho_1^2 & \rho_1 & 1 \end{pmatrix}$$

  `ex:` exchangeable: todas las variables tienen la misma correlación

  $$\begin{pmatrix} 1 & \rho_1 & \rho_1 \\ \rho_1 & 1 & \rho_1 \\ \rho_1 & \rho_1 & 1 \end{pmatrix}$$

  `toep` Toeplitz: matriz con estructuras diagonales constantes:

  $$\begin{pmatrix} 1 & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_1 \\ \rho_2 & \rho_1 & 1 \end{pmatrix}$$

  `un` unstructured: todas las correlaciones son diferentes.

  $$\begin{pmatrix} 1 & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_3 \\ \rho_2 & \rho_3 & 1 \end{pmatrix}$$

- `archmCopula`, que incluye `claytonCopula`, `frankCopula` y `gumbelCopula`.

Ejemplo 1

Para generar una cópula gaussiana de dimensión 4 de tipo no estructurado, se requiere definir 6 valores de correlaciones:

```
library(copula)
copula.normal4 <- ellipCopula(family = "normal", dim = 4, dispstr = "un",
param = c(0.4,0.5,0.2,0,0.3,0.8))
copula.normal4 #objeto de clase normalCopula
```

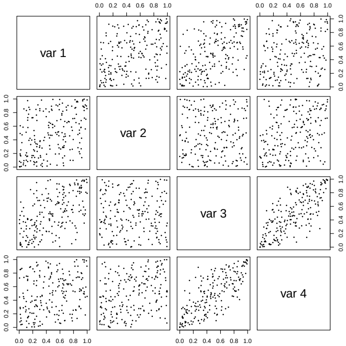
```
Normal copula, dim. d = 4
Dimension:  4
Parameters:
  rho.1   = 0.4
  rho.2   = 0.5
  rho.3   = 0.2
  rho.4   = 0.0
  rho.5   = 0.3
  rho.6   = 0.8
dispstr:  un
```

```
u <- rCopula(200,copula.normal4) #genera observaciones de la cópula construida
cor(u)
```

```
          [,1]      [,2]      [,3]      [,4]
[1,] 1.0000000 0.3999356 0.5583336 0.2587525
[2,] 0.3999356 1.0000000 0.0091090 0.3456752
[3,] 0.5583336 0.0091090 1.0000000 0.7516383
[4,] 0.2587525 0.3456752 0.7516383 1.0000000
```

Una gráfica de la función anterior en pares:

```
pairs(u,pch=16, cex=0.5)
```



Ejemplo 2

Para generar una cópula t de dimensión 3 de tipo Toeplitz, se requiere definir 2 valores de correlaciones y los grados de libertad:

```
micopula.t3 <- ellipCopula(family = "t", dim = 3, dispstr = "toep",
param = c(0.8,0.5), df = 8)
micopula.t3 #objeto de clase tCopula

t-copula, dim. d = 3
Dimension:  3
Parameters:
  rho.1    = 0.8
  rho.2    = 0.5
  df       = 8.0
dispstr:  toep

rCopula(5,micopula.t3) #genera cinco observaciones de la cópula construida

          [,1]       [,2]      [,3]
[1,] 0.7561208 0.83477990 0.3862287
[2,] 0.1214745 0.15426384 0.3950312
[3,] 0.1372568 0.06109462 0.1017845
[4,] 0.5601066 0.55269459 0.5608740
[5,] 0.4282554 0.43444576 0.6929481
```

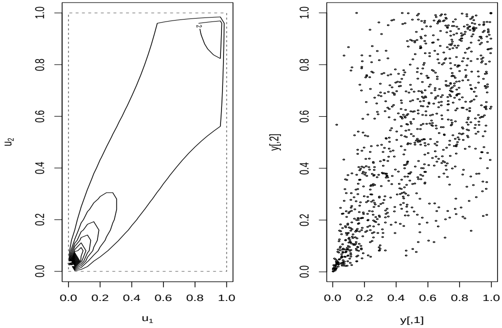
Ejemplo 3

Para generar una cópula tipo Clayton bidimensional con parámetro $\theta = 2$:

```
clayton2 <- archmCopula(family = "clayton", dim = 2, param = 2)
clayton2 #el programa llama alpha al parámetro

Clayton copula, dim. d = 2
Dimension:  2
Parameters:
  alpha   = 2

# Generemos una muestra de ésta cópula:
y <- rCopula(1000,clayton2)
```

Para comparar con las curvas de nivel de la cópula de Clayton, podemos generar una muestra aleatoria y ver su distribución conjunta. Notemos que la cópula de Clayton tiene una alta de concentración de probabilidad en el origen. Esto puede ser de utilidad para correlacionar pequeñas pérdidas.

```
par(mfrow=c(1,2))
contour(clayton2,dCopula) #gráfica de curvas de nivel
plot(y,cex=0.3)
```



u2
u1
y[,2]
y[,1]

Clase `mvdc`

- Esta clase de objetos está diseñada para construir distribuciones multivariadas con maginales dadas usando cópulas.
- Este es el caso que hicimos en la última clase de cópulas, dadas las marginales y una estructura de dependencia, construir una distribución conjunta usando, por ejemplo, la cópula Gaussiana.
- Esta clase tiene tres componentes:

  `copula`: Especifica la cópula C a usar para 'pegar' las marginales.

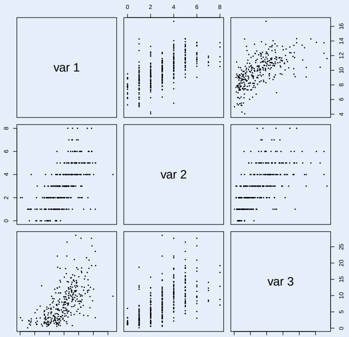
  `margins`: Especifica los nombres de las distribuciones marginales a usar.

  `paramMargins`: una lista de listas, con los parámetros de las distribuciones marginales.

Ejemplo 4

Generar una distribución conjunta con una cópula Frank con parámetro $\theta = 5$ y con marginales $\mathcal{N}\,(10,4)$, $\mathcal{P}\,(3)$ y $\mathcal{G}\,(2,4)$.

```
copula.Frank5 <- archmCopula(family = "frank", dim = 3, param = 5)
micopula <- mvdc(copula = copula.Frank5, margins = c("norm","pois","gamma"),
paramMargins = list(list(mean=10,sd=2), list(lambda=3), list(shape=2,scale=4)))
u <- rMvdc(300,micopula) #muestra aleatoria
par(mar=c(1,1,1,1));pairs(u,pch=16,cex=0.5)
```



Funciones de distribución y densidad para cópulas I

Para la distribución conjunta creada a partir de la cópula

```
u <- rMvdc(5,micopula)
u # puntos del dominio

          [,1] [,2]      [,3]
[1,] 12.217118    5 23.825533
[2,] 10.959193    6  5.464355
[3,]  9.983242    3  4.927729
[4,]  9.062085    2  2.944202
[5,]  7.397641    1  2.956918

dMvdc(u,micopula) # puntos de la densidad

[1] 0.0004065068 0.0003107008 0.0055394034 0.0062570787 0.0049991691

pMvdc(u,micopula) # puntos de la distribución

[1] 0.81171345 0.36484532 0.26675940 0.10375430 0.03015304
```

Para la cópula dada:

Funciones de distribución y densidad para cópulas II

```r
u <- rCopula(5,copula.Frank5)
u # puntos del dominio
```

```
          [,1]      [,2]      [,3]
[1,] 0.7872706 0.8028639 0.8070379
[2,] 0.8792277 0.8008430 0.9959447
[3,] 0.3188677 0.6734840 0.6816883
[4,] 0.4410392 0.2512305 0.4597243
[5,] 0.1723916 0.5241862 0.2036805
```

```r
dCopula(u,copula.Frank5) # puntos de la densidad
```

```
[1] 4.3678780 5.6718894 0.6929495 1.7070607 1.3589326
```

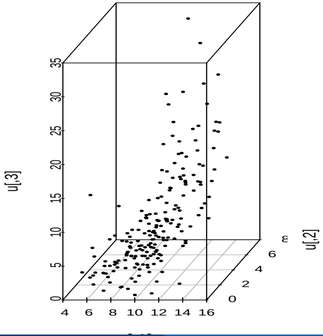
```r
pCopula(u,copula.Frank5) # puntos de la distribución
```

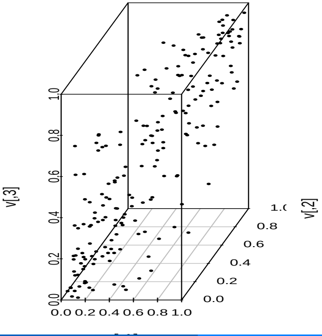
```
[1] 0.63703314 0.74680497 0.28060158 0.17364825 0.08518855
```

Gráficas

La siguiente gráfica muestra la realización de una muestra de la distribución conjunta solicitada:

```
library(scatterplot3d)
par(mfrow=c(1,2),mar=c(1,2,1,1),oma=c(0,0,1,1),mgp=c(2,1,0))
u <- rMvdc(200,micopula)
scatterplot3d(u,cex.symbols = 0.5, pch=16)
v <- rCopula(200,copula.Frank5)
scatterplot3d(v,cex.symbols = 0.5, pch=16)
```





Contornos de funciones de densidad/distribución I

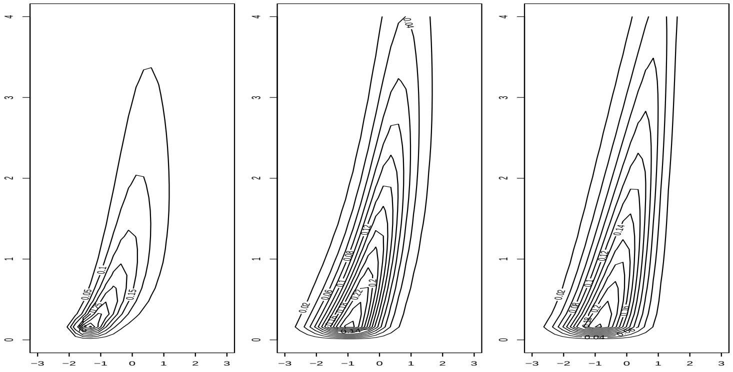
El siguiente código grafica los contornos de densidad de distribuciones bivariadas definidas con las tres cópulas de Clayton, Frank y Gumbel con marginales normales.

```
miMvd1 <- mvdc(copula = archmCopula(family="clayton", param=2), margins = c("norm", "gamma"),
paramMargins = list(list(mean=0,sd=1), list(shape=1,scale=2)))
miMvd2 <- mvdc(copula = archmCopula(family="frank", param=5.763), margins = c("norm", "gamma"),
paramMargins = list(list(mean=0,sd=1), list(shape=1,scale=2)))
miMvd3 <- mvdc(copula = archmCopula(family="gumbel", param=2), margins = c("norm", "gamma"),
paramMargins = list(list(mean=0,sd=1), list(shape=1,scale=2)))
```

Los parámetros han sido escogidos para dar una τ de Kendall para las tres distribuciones igual a 0.5.

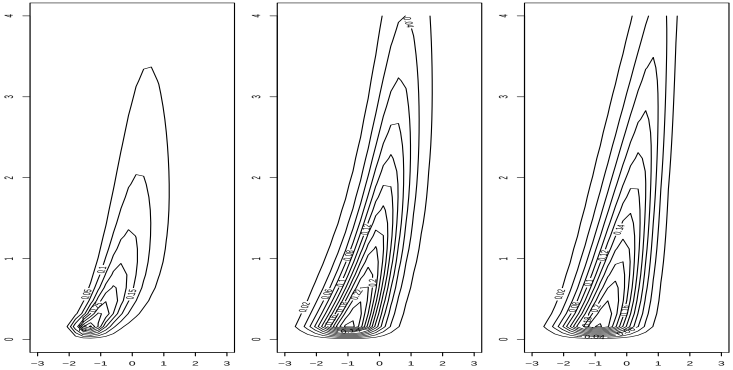
```
par(mfrow=c(1,3), mar=c(2,2,1,1), oma=c(1,1,0,0), mgp=c(2,1,0))
contour(miMvd1, dMvdc, xlim = c(-3,3), ylim = c(0,4))
contour(miMvd2, dMvdc, xlim = c(-3,3), ylim = c(0,4))
contour(miMvd3, dMvdc, xlim = c(-3,3), ylim = c(0,4))
```

Contornos de funciones de densidad/distribución II



```
par(mfrow=c(1,3), mar=c(2,2,1,1), oma=c(1,1,0,0), mgp=c(2,1,0))
contour(miMvd1, dMvdc,xlim=c(-3,3), ylim=c(0,4))
contour(miMvd2, dMvdc,xlim=c(-3,3), ylim=c(0,4))
contour(miMvd3, dMvdc,xlim=c(-3,3), ylim=c(0,4))
```

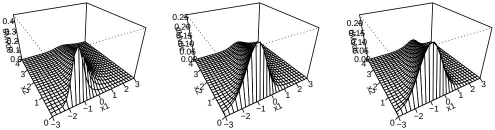
Contornos de funciones de densidad/distribución III



persp

La función `persp` es similar:

```
par(mfrow=c(1,3), mar=c(2,2,1,1), oma=c(1,1,0,0), mgp=c(2,1,0))
persp(miMvd1, dMvdc,xlim=c(-3,3), ylim=c(0,4))
persp(miMvd2, dMvdc,xlim=c(-3,3), ylim=c(0,4))
persp(miMvd3, dMvdc,xlim=c(-3,3), ylim=c(0,4))
```



Día 3: Medidas de correlación

Medidas de dependencia I

- Hemos visto que las cópulas nos permiten modelar la dependencia entre variables aleatorias. Podemos decir que dos vectores aleatorios tienen la misma dependencia cuando tienen la misma cópula.
- Los coeficientes de correlación están definidos entre dos variables aleatorias reales, y se consideran como una medida del grado de dependencia.
- Hay tres coeficientes de correlación que son ampliamente usados. Todos satisfacen la condición $\rho(X,Y) \in [-1,1]$:
  1. correlación lineal o correlación de Pearson ρ_L
  2. correlación de rangos de Spearman ρ_S
  3. correlación de rangos de Kendall ρ_τ.
- A continuación analizaremos cada uno, pero daremos las propiedades deseables de la correlación.

Propiedades deseables de las medidas de dependencia

Sea $\delta(\cdot, \cdot)$ una medida de dependencia que asigna un número real a cualquier par de variables aleatorias reales X y Y. Idealmente, se desean las siguientes propiedades:

P1. $\delta(X, Y) = \delta(Y, X)$ (simetría)

P2. $-1 \leq \delta(X, Y) \leq 1$ (normalización)

P3. $\delta(X, Y) = 1 \iff X, Y$ comonótonas, $\delta(X, Y) = -1 \iff X, Y$ contramonótonas

P4. Para $T : \mathbb{R} \to \mathbb{R}$ estríctamente monótona en el rango de X:

$$\delta(T(X), Y) = \begin{cases} \delta(X, Y) & T \quad \text{creciente} \\ -\delta(X, Y) & T \quad \text{decreciente} \end{cases}$$

P5. $\delta(X, Y) = 0 \iff X \perp\!\!\!\perp Y$

La correlación lineal cumple sólo P1 y P2. La correlación de rangos cumple P3 y P4 si X y Y son continuas. No hay medida de dependencia que cumpla P4 y P5 simultáneamente.

Correlación de Pearson

- La correlación de Pearson

$$\rho_L(X,Y) = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}$$

es una medida de dependencia limitada, ya que depende tanto de una cópula como de las marginales, lo que se puede ver a través de la fórmula de Höffding:

$$\begin{aligned} \text{cov}(X,Y) &= \int_{\mathbb{R}^2} (F(u,v) - F_1(u)F_2(v))du\,dv \\ &= \int_{[0,1]^2} [C(u,v) - uv]dF_1^{-1}(u)dF_2^{-1}(v) \end{aligned}$$

donde $C(u,v) = F(F_1^{-1}(u), F_2^{-1}(v))$.

Correlaciones basadas en rangos

- La *correlación basada en rangos* son medidas escalares que dependen sólo de la cópula de la distribución bivariada y no de las marginales.

Rango

El rango de una variable es la posición de sus valores ordenados de menor a mayor:

$$\text{rango}(x) = \sum_{j=1}^{n} I(X_j \leq X_i)$$

- Los rangos no cambian por transformaciones estríctamente monótonas.
- Hay dos principales variedades de correlación basada en rangos: τ de Kendall y ρ_S de Spearman.

Correlación de Spearman

La correlación de Spearman es la correlación de Pearson sobre los valores evaluados en los rangos inducidos por las distribuciones marginales de los datos:

$$\rho_S(X, Y) = \rho_L(F_1(X), F_2(Y))$$

donde F_k es la función de distribución de X_k para $k = 1, 2$.

ρ_S de Spearman II

- Cuando las variables aleatorias tienen distribuciones continuas, la correlación de Spearman se puede expresar en términos de la cópula:

$$\begin{aligned}\rho_S(X_1, X_2) &= 12 \int_{\mathbb{R}^2} [F(u,v) - F_1(u)F_2(v)] dF_1(u)\, dF_2(v) \\ &= 12 \int_{[0,1]^2} [C(u,v) - uv] du\, dv \\ &= 12 \int_{[0,1]^2} C(u,v) du\, dv\end{aligned}$$

- Podemos ver que la correlación de Spearman es la correlación de la cópula de (X, Y).
- La ρ_S muestral está dada por:

$$\hat{\rho}_S(X,Y) = \frac{12}{n(n^2-1)} \sum_{i=1}^{n} \left(\text{rango}(X_i) - \frac{n+1}{2}\right) \left(\text{rango}(Y_i) - \frac{n+1}{2}\right)$$

τ de Kendall

Sea (X, Y) un vector aleatorio y (X^*, Y^*) una vector con la misma distribución e independiente de (X, Y) (es una copia de (X, Y)). Entonces (X, Y) y (X^*, Y^*) son pares *concordantes* (*discordantes*) si $(X - X^*)(Y - Y^*) > 0$ ($(X - X^*)(Y - Y^*) < 0$).
La τ de Kendall es la diferencia de probabilidades de par concordante y de par discordante:

$$\begin{aligned} \rho_\tau(X, Y) &= P((X - X^*)(Y - Y^*) > 0) - P((X - X^*)(Y - Y^*) < 0) \\ &= E(sgn\{(X - X^*)(Y - Y^*)\}) \end{aligned}$$

La τ de Kendall muestral está dada por:

$$\hat{\rho}_\tau = \binom{n}{2}^{-1} \sum_{1 \leq i \leq j \leq n} sgn\{(X_i - X_j^*)(Y_i - Y_j^*)\} = \frac{C - D}{C + D} = \frac{C - D}{\binom{n}{2}}$$

donde C son los pares concordantes y D son los pares discordantes.

Ejemplo 1

1. Para la muestra de parejas $(2,3), (3,4), (1,5), (5,2), (4,8), (9,6), (6,8), (4,3), (2,1), (10,10)$ calcular la τ de Kendall.

```
muestra <- cbind(c(2,3,1,5,4,9,6,4,2,10),c(3,4,5,2,8,6,8,3,1,10))
cor(muestra, method = "kendall")
```

```
          [,1]      [,2]
[1,] 1.0000000 0.3953488
[2,] 0.3953488 1.0000000
```

```
# La correlación de Pearson usual
cor(muestra)
```

```
          [,1]      [,2]
[1,] 1.0000000 0.6440133
[2,] 0.6440133 1.0000000
```

2. Consideren X =rango de calificación del examen 1 y Y =rango de calificación del examen 2. Calcular la τ de Kendall de manera manual.

```
X <- c(1,2,3,4,5,6,7)
Y <- c(1,3,6,2,7,4,5)
```

Ejemplo 2

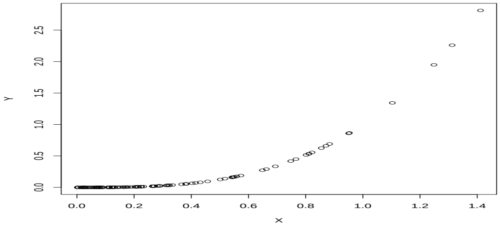
- Para la muestra de parejas $(2,3),(3,4),(1,5),(5,2),(4,8),(9,6),(6,8),(4,3),(2,1),(10,10)$ calcular la ρ_S de Spearman

```
muestra <- cbind(c(2,3,1,5,4,9,6,4,2,10),c(3,4,5,2,8,6,8,3,1,10))
cor(muestra, method = "spearman")

          [,1]      [,2]
[1,] 1.0000000 0.5613497
[2,] 0.5613497 1.0000000

#Para efectos comparativos
cor(muestra, method = "kendall")

          [,1]      [,2]
[1,] 1.0000000 0.3953488
[2,] 0.3953488 1.0000000

# La correlación de Pearson usual
cor(muestra)

          [,1]      [,2]
[1,] 1.0000000 0.6440133
[2,] 0.6440133 1.0000000
```

Ejemplo 3

- Supongamos que $Y = X^2$ para una variable aleatoria $X \sim \exp(3)$. Supongamos que tenemos una muestra de X. ¿Cuanto valen las correlaciones?

```
X <- rexp(100, rate = 3)
Y <- X^3
plot(X,Y)
```




```
cor(X,Y)
[1] 0.8494689
cor(X,Y, method = "spearman")
[1] 1
cor(X,Y, method = "kendall")
[1] 1
```

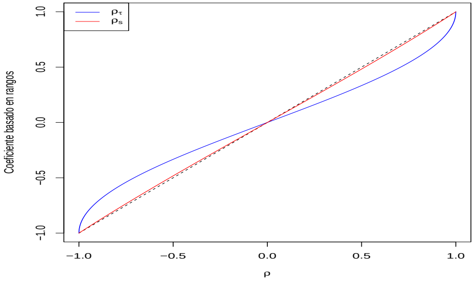
Relación de medidas de dependencia en el caso normal

- En el caso de distribuciones elípticas, hay una relación biyectiva entre las medidas no paramétricas y el coeficiente de correlación de Pearson:

$$\rho_\tau = \frac{2}{\pi} arcsin(\rho_L)$$

y

$$\rho_s = \frac{6}{\pi} arcsin(\rho_L/2)$$



Ejemplo I

1. Generar 1,000 observaciones de un vector aleatorio $W = (X, Y, Z)$ donde $X \sim N(4, 25)$, $Y \sim t_4$ y $Z \sim Binom(25, 0.4)$, considerando las siguientes restricciones: $\tau(X,Y) = 0.7$, $\tau(X,Z) = 0.3$ y $\tau(Y,Z) = 0.4$.

Solución.

La primera alternativa es usar una cópula Gaussiana para construir nuestra muestra, incorporando la restricción de la dependencia. El procedimiento que se utilizará es el siguiente:

1. Transformar las τ de Kendall a la correlación para construir la matriz de correlaciones necesaria Σ. Para transformar los valores usamos la fórmula conocida para la distribución normal: $\rho = sin(\frac{\pi * \tau}{2})$. Entonces la matriz Σ tiene la forma

$$\Sigma = \begin{pmatrix} 1 & 0.8910065 & 0.4539905 \\ 0.8910065 & 1 & 0.5877853 \\ 0.4539905 & 0.5877853 & 1 \end{pmatrix}$$

2. Obtener un vector $Z \sim N(0, \Sigma)$
3. Obtener un vector $U \sim (\Phi(Z_1), \Phi(Z_2), \Phi(Z_3))$, donde $\Phi(x)$ es la función de distribución normal estándar
4. Obtener un vector $W \sim (F_1^{-1}(U_1), F_2^{-1}(U_2), F_3^{-1}(U_3))$, donde las F_i son las distribuciones deseadas. Entonces W tiene la composición deseada.

Ejemplo II

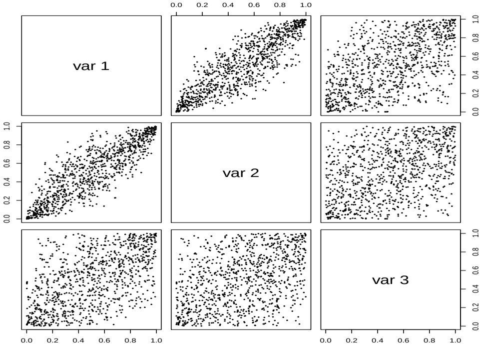
Justo los pasos 2 y 3 son los que corresponden a obtener una muestra aleatoria de una cópula, en este caso una cópula gaussiana. Entonces podemos escribir un código ligeramente más sencillo:

```
library(copula)
# Define el objeto cópula a generar, en este caso es una normal con correlaciones
# dadas
# El argumento dispstr se refiere a la estructura a la matriz de covarianza que caracteriza a
# la cópula. "un" es para indicar que no tiene estructura.
# ver detalle en https://www.jstatsoft.org/index.php/jss/article/view/v021i04/v21i04.pdf

copula_normal_3 <- normalCopula(c(sin(0.7*pi/2), sin(0.4*pi/2), sin(0.3*pi/2)),
                                dim = 3, dispstr = "un")
set.seed(100) #fija una semilla
U <- rCopula(1000, copula_normal_3) #Genera la muestra aleatoria
```

Con el código previo, se realizan los primeros tres pasos de la simulación. Antes de continuar, podemos ver cómo se ven las muestras generadas por pares, y podemos ver si tenemos la condición establecida como restricción sobre las covarianzas. La matriz de tau de Kendall está dada por:

```
pairs(U, pch=16, cex=0.5)
```

Ejemplo III



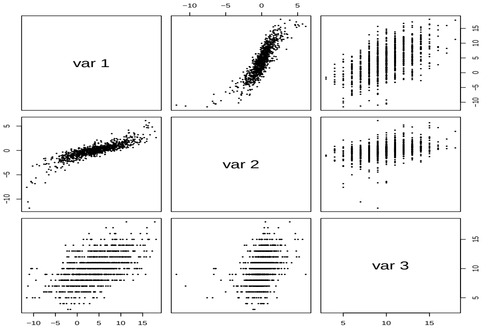

```r
round(cor(U, method = "kendall"), 2)

     [,1] [,2] [,3]
[1,] 1.00 0.69 0.39
[2,] 0.69 1.00 0.28
[3,] 0.39 0.28 1.00
```

Ejemplo IV

Ahora procedemos al paso 4, para generar nuestro vector y hacemos los histogramas para ver si tienen el comportamiento deseado, y se muestra un ejemplo de los valores generados:

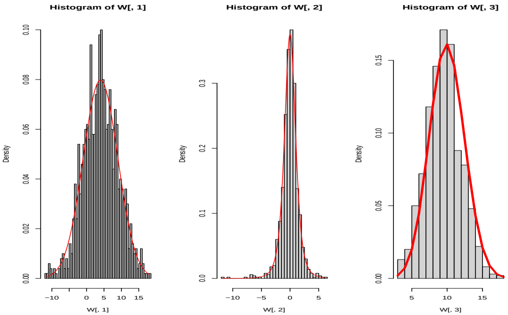
```
W <- cbind(qnorm(U[,1], mean = 4, sd = 5),
qt(U[,2], 4),
qbinom(p = U[,3], size = 25, prob = 0.4))
head(W)

         [,1]        [,2] [,3]
[1,] 2.180664 -0.16821757    9
[2,] 8.355016  0.65959400   11
[3,] 2.275934  0.17518414    8
[4,] 2.902724 -0.09615615   10
[5,] 5.235607  0.58978484   10
[6,] 3.620093 -0.27198890   11

pairs(W, pch = 16, cex = 0.5)
```

Ejemplo V




```r
#Grafica los histogramas y agrega densidades con las distribuciones deseadas para ver
#la aproximación
par(mfrow = c(1,3))
hist(W[,1], prob = T,breaks = 50); points(sort(W[,1]),dnorm(sort(W[,1]), 4, 5), type = "l", col = "red")
hist(W[,2], prob = T, breaks = 50); points(sort(W[,2]), dt(sort(W[,2]),4), type = "l", col = "red")
hist(W[,3], prob = T);
points(sort(W[,3]),dbinom(sort(W[,3]), size = 25, prob = 0.4),type = "l", col = "red", lwd = 3)
```

Ejemplo VI



Podemos corroborar que alcanzamos la medida de dependencia requerida

```
cor(W, method = "kendall")

          [,1]      [,2]      [,3]
[1,] 1.0000000 0.6884324 0.4120571
[2,] 0.6884324 1.0000000 0.3006960
[3,] 0.4120571 0.3006960 1.0000000
```

Día 4: Aplicación

Introducción

- Un problema importante es: dado un conjunto de datos, elegir una cópula para ajustarlos.
- El problema tiene dificultades técnicas y trampas con las que hay que ser muy cuidadoso.
- El problema es que la estimación de cópulas implica usualmente que cada marginal debe ser evaluada y conectada a una distribución multivariada estimada.
- A continuación veremos un ejemplo de estimación.
- Usaremos varios paquetes de `R` para realizar el ejercicio:
  - paquetes `Ecdat` para tomar algunos datos
  - `copula`, que es el paquete principal de donde se obtienen la mayoría de las características.
  - `fGarch` para el uso de la densidad t estandarizada
  - `MASS` para el uso de las funciones *fitdistr, kde2d*
  - `fCopulae` para funciones adicionales de copulas: *pempiricalCopula, ellipticalCopulafit*.

Datos I

- El paquete `Ecdat` es un conjunto de datos Econométricos. Los datos `CRSPday` contiene una base de datos de rendimientos diarios de acciones del Center for Research in Security Prices (CRSP), del 3 de enero de 1969 al 31 de diciembre de 1998.
- Nos vamos a fijar en las dos variables `ibm`, que es el rendimiento de IBM y `crsp` que es un índice ponderado de rendimientos construído por el CRSP.

Datos II

```r
library(Ecdat) # fuente de datos
library(copula)
library(fGarch)
library(MASS) # usa las funciones fitdistr y kde2d
library(fCopulae)  # funciones adicionales de copula (pempiricalCopula y ellipticalCopulaFit)

data(CRSPday,package="Ecdat")
head(as.data.frame(CRSPday)) # muestra la estructura de los datos
```

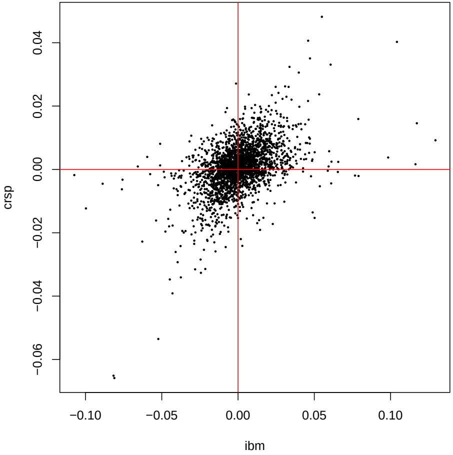
```
  year month day        ge       ibm     mobil      crsp
1 1989     1   3 -0.016760  0.000000 -0.002747 -0.007619
2 1989     1   4  0.017045  0.005128  0.005510  0.013016
3 1989     1   5 -0.002793 -0.002041  0.005479  0.002815
4 1989     1   6  0.000000 -0.006135  0.002725  0.003064
5 1989     1   9  0.000000  0.004115  0.005435  0.001633
6 1989     1  10 -0.005602 -0.007172  0.008108 -0.001991
```

```r
ibm <- CRSPday[,5]; crsp <- CRSPday[,7]
n <- length(ibm); n #número de observaciones
```

```
[1] 2528
```

```r
par(pty = "s"); plot(ibm, crsp, cex = 0.4, pch = 16)
abline(h = 0, v = 0, col="red")
```

Datos III



Marginales propuestas

- A continuación se ajustará una distribución t a cada una de las variables marginales. Los valores que se guardan corresponden a los valores estimados de las distribuciones t marginales. Cada distribución marginal puede ajustar diferentes grados de libertad.
- La función `fitdistr` del paquete `MASS` estima las características de una función de distribución usando máxima verosimilitud (en el caso de la t, su media, escala, y grados de libertad).

```
ibm <- CRSPday[,5]
crsp <- CRSPday[,7]
est.ibm <- as.numeric(fitdistr(ibm,"t")$estimate) #parámetros t: media, escala, gl
est.crsp <- as.numeric(fitdistr(crsp,"t")$estimate)
#Convierte los parámetros de escala a desviaciones estándar en el caso de la t
est.ibm[2] <- est.ibm[2]*sqrt(est.ibm[3]/(est.ibm[3]-2))
est.crsp[2] <- est.crsp[2]*sqrt(est.crsp[3]/(est.crsp[3]-2))
#Grados de libertad para cada caso
est.ibm[3]

[1] 4.276156

est.crsp[3]

[1] 3.473982
```

Ajuste de cópula específica I

- Como un ejercicio inicial, supongamos que se quiere ajustar una cópula específica, por ejemplo, una cópula t.
- Para estimar una t-cópula por máxima verosimilitud, se requiere una estimación de la correlación y un valor inicial adecuado.
- Se usarán las densidades t estimadas como valores iniciales. Se define la cópula t con 2 grados de libertad

```
(tau <- cor(ibm,crsp,method = "kendall"))

[1] 0.3308049

(omega <- 2/pi*asin(tau))

[1] 0.2146404

copula2 <- tCopula(omega,dim=2,dispstr = "un",df = 2)
```

Ahora hay que ajustar la copula a los datos uniformes transformados:

Ajuste de cópula específica II

```
# La función pstd es la distribución estándar t
# por el método de máxima verosimilitud
d1 <- cbind(fGarch::pstd(ibm, mean = est.ibm[1], sd = est.ibm[2], nu = est.ibm[3]),
fGarch::pstd(crsp, mean = est.crsp[1], sd = est.crsp[2], nu = est.crsp[3]))

fit1 <- fitCopula(copula2, method = "ml", optim.method = "L-BFGS-B", data = d1,
start = c(omega,5), lower = c(0,2.5), upper = c(0.5,15))
fit1


Call: fitCopula(copula2, data = d1, ... = pairlist(method = "ml", optim.method = "L-BFGS-B", start = c(omega,
    5), lower = c(0, 2.5), upper = c(0.5, 15)))
Fit based on "maximum likelihood" and 2528 2-dimensional observations.
Copula: tCopula
 rho.1     df
0.4937 9.8537
The maximized loglikelihood is 362
Optimization converged
```

Cópulas alternas I

Para efectos de comparación, consideremos ahora el ajuste de otras cópulas a los datos:

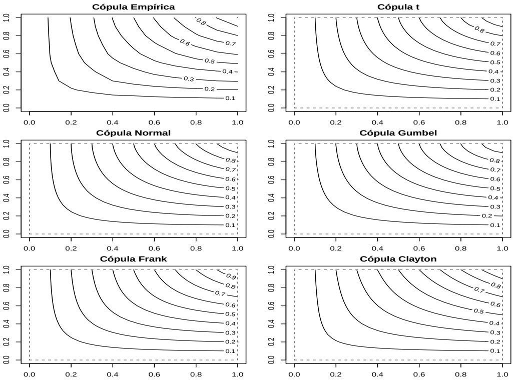
```
#Ajusta copula normal
fnorm <- fitCopula(data = d1, copula = normalCopula(-0.3, dim = 2),
method = "ml", optim.method = "BFGS", start = 0.5)
#Ajusta Gumbel
fgumbel <- fitCopula(data=d1,copula=gumbelCopula(3,dim=2),
method="ml",optim.method="BFGS",start=2)
#Ajusta Frank
ffrank <- fitCopula(data=d1,copula=frankCopula(3,dim=2),
method="ml",optim.method="BFGS",start=2)
#Ajusta Clayton
fclayton <- fitCopula(data=d1,copula=claytonCopula(3,dim=2),
method="ml",optim.method="BFGS",start=2)
```

Las copulas estimadas se compararán contra la cópula empírica y se verá si hay alguna estimación que quede cerca a la cópula que se obtiene de los datos.

Comparación gráfica I

```r
u <- d1
dem <- pempiricalCopula(u[,1],u[,2])
par(mfrow=c(3,2),mar=c(2,2,2,2))
contour(dem$x,dem$y,dem$z,main="Cópula Empírica")
contour(tCopula(fit1@estimate[1],df=round(fit1@estimate[2],0)),pCopula,main="Cópula t")
contour(normalCopula(fnorm@estimate),pCopula,main="Cópula Normal")
contour(gumbelCopula(fgumbel@estimate),pCopula,main="Cópula Gumbel")
contour(frankCopula(ffrank@estimate),pCopula,main="Cópula Frank")
contour(claytonCopula(fclayton@estimate),pCopula,main="Cópula Clayton")
```

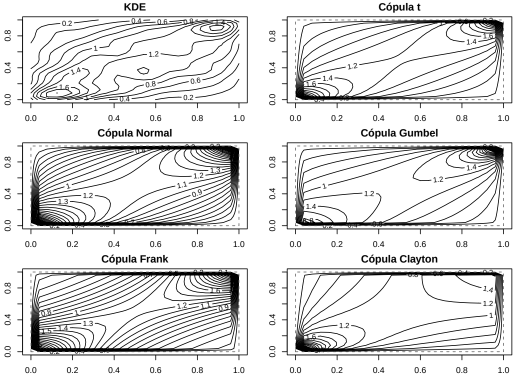
Comparación gráfica II



En el siguiente conjunto de gráficas se comparará la estimación de la densidad bivariada entre las diferentes estimaciones paramétricas

```
par(mfrow=c(3,2),mar=c(2,2,2,2))
contour(kde2d(u[,1],u[,2]),main="KDE")
contour(tCopula(fit1@estimate[1],df=fit1@estimate[2]),dCopula,
main="Cópula t",nlevels=25)
contour(normalCopula(fnorm@estimate),dCopula,main="Cópula Normal",nlevels=25)
contour(gumbelCopula(fgumbel@estimate),dCopula,main="Cópula Gumbel",nlevels=25)
contour(frankCopula(ffrank@estimate),dCopula,main="Cópula Frank",nlevels=25)
contour(claytonCopula(fclayton@estimate),dCopula,main="Cópula Clayton",nlevels=25)
```

Comparación de estimación con distribuciones paramétricas II



Evaluación a través de AIC

Por último, podemos comparar los AIC. Recuerden que el criterio de información de Akaike se utiliza para comparar modelos. Aquel con el mayor valor absoluto nos dice cuál es el mejor modelo:

```r
#AIC Normal
2*length(fnorm@estimate)-2*fnorm@loglik

[1] -692.3688

#AIC Gumbel
2*length(fgumbel@estimate)-2*fgumbel@loglik

[1] -624.4514

#AIC frank
2*length(ffrank@estimate)-2*ffrank@loglik

[1] -648.5734

#AIC Clayton
2*length(fclayton@estimate)-2*fclayton@loglik

[1] -584.2204

#AIC t
2*length(fit1@estimate)-2*fit1@loglik

[1] -719.9693
```

Indices de dependencia de colas I

- Las cópulas en general no difieren mucho en el grado de asociación que proveen, sino más bien en la parte de las distribuciones donde las asociaciones son más fuertes. Como se quiere poner énfasis en la correlación de pérdidas grandes en carteras, se pone usualmente el énfasis en la cola de la distribución.
- El coeficiente de correlación lineal basado en la covarianza de dos variables (Pearson) no se preserva en la cópula: dos pares de variables correlacionadas con la misma cópula pueden tener diferentes correlaciones. Pero la τ de Kendall es una constante de la copula.

Dependencia de colas: Índice fuerte I

- Un índice (fuerte) de dependencia de colas (izquierda) sugerido por Harry Joe (1990) se calcula como:

$$\lambda_l = \lim_{u \to 0} P(F_X(x) \leq u | F_y(y) \leq u) = \lim_{u \to 0} \frac{C(u,u)}{u}$$

  donde C es una función cópula. El comportamiento de λ_l nos puede decir cómo se comporta la dependencia en la cola. Si $\lambda_l = 0$, entonces se puede concluir que en las colas, X y Y se comportan como si fueran independientes.

- Se puede graficar la función λ_l como función de u, para tratar de visualizar su comportamiento límite. Podemos definir las siguientes **funciones de concentración**, por la izquierda y por la derecha, que miden qué tanto se concentran en las regiones cerca de (1,1) y (0,0):

$$L(u) = \frac{C(u,u)}{u}$$

  para la cola inferior. Esta función se acerca a 1 conforme $u \to 1$, por lo que tiene más sentido ver sus valores para $u < 1$. Si los valores de $L(u)$ se mantienen cerca de 1, podemos considerar que los datos tienen mayor dependencia de colas.

Dependencia de colas: Índice fuerte II

- Para la cola superior, tenemos que

$$\lambda_u = \lim_{u \to 1} \frac{C^*(1-u, 1-u)}{1-u}$$

y la función de concentración derecha la podemos definir como:

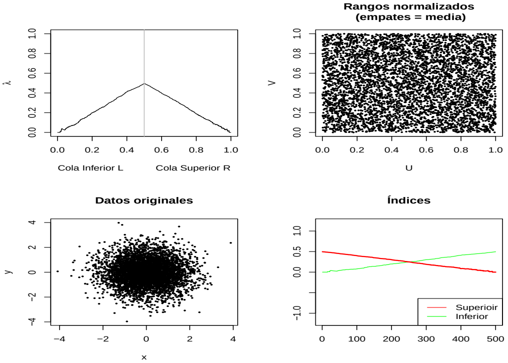
$$R(u) = \frac{P(X > F_X^{-1}(u), Y > F_Y^{-1}(u))}{P(X > F_X^{-1}(u))} = \frac{C^*(1-u, 1-u)}{1-u}$$

donde C^* es la cópula de supervivencia asociada a C. Esta función se acerca al 0 conforme $u \to 0$, por lo que es mejor ver su comportamiento para $u > 0$.

- Las funciones $L(u)$ y $R(u)$ se pueden estimar a través de la cópula empírica y graficarse de manera combinada en una función LR que sea L para $u \leq \frac{1}{2}$ y R para $u \geq \frac{1}{2}$.
- Para efectos de bondad de ajuste, podemos comparar con alguna cópula paramétrica, por ejemplo, la cópula gaussiana o gumbel por ejemplo.

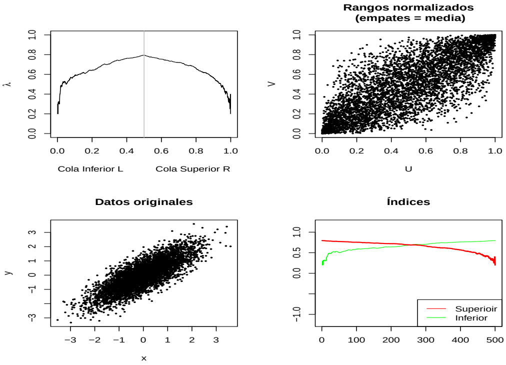
Ejemplo I

- El siguiente ejemplo elabora la gráfica LR para datos normales e independientes. Podemos ver que conforme $u \to 0$, la linea negra se acerca a 0, indicando que hay independencia en la cola izquierda. Lo mismo se observa en el lado de R, conforme $u \to 1$, $\lambda_u = 0$, indicando independencia de las dos variables aleatorias.



Ejemplo II

- Como otro ejemplo, considerando ahora datos normales con correlación de 0.8. Podemos ver que los límites son más inestables conforme las funciones de concentración se acercan a 0 y a 1 por la izquierda y por la derecha, respectivamente.



Índice débil I

- Un índice alternativo se puede calcular de la siguiente manera.Supóngase que X y Y tienen la misma distribución. Si las variables son independientes, se tiene que

$$P(X \leq x, Y \leq y) = P(X \leq x)P(Y \leq y) = F(x)^2$$

dado que tienen la misma distribución, Por otro lado, si las variabes son co-monotónicas (es decir, son dos copias exactas de la misma variable), entonces son exactamenete idénticas porque tienen la misma distribución y en ese caso:

$$P(X \leq x, Y \leq y) = P(X \leq x) = F(x)^1$$

- Entonces podemos tomar $\alpha \in \{1, 2\}$ como una medida de dependencia. El parámetro $\alpha = 2$ se interpreta como independencia, y $\alpha = 1$ como dependencia fuerte positiva perfecta.

Índice débil II

- Para tener un parámetro en $[0,1]$ e indicando fuerza en la dependencia que se incrementa con el índice, consideramos la transformación $\xi = \frac{2}{\alpha} - 1$. Para este caso se define el índice como el límite de la función

$$R_2(u) = 2\frac{\log(P(X > F_X^{-1}(u))}{\log(P(X > F_X^{-1}(u), Y > F_Y^{-1}(u))} - 1$$

que se interpreta como un índice de dependencia de cola débil. Definimos

$$L_2(u) = 2\frac{\log(P(X < F_X^{-1}(u)))}{\log(P(X < F_X^{-1}(u), Y < F_Y^{-1}(u)))} - 1$$

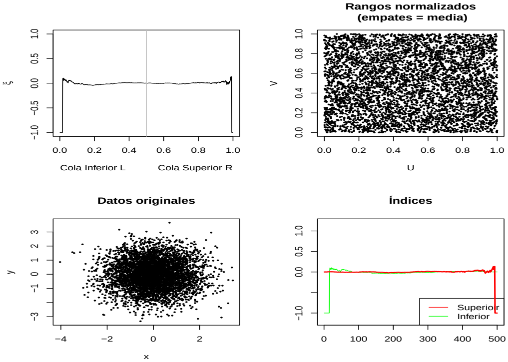
y $R_2(u)$ como arriba.

Ejemplo: índice débil I

- Haciendo el ejercicio para el índice débil en el caso de independencia. En este caso se espera que el índice ξ esté alrededor de 0, porque $\alpha \approx 2$

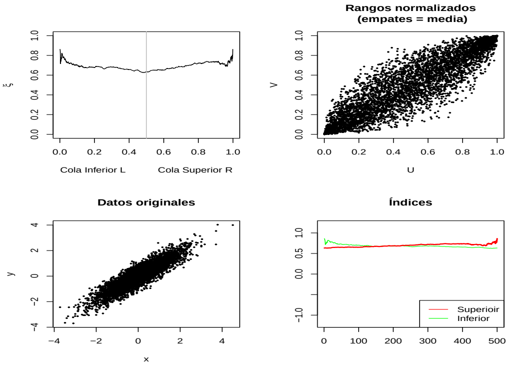
```
par(mfrow = c(2,2))
n <- 5000
x <- rnorm(n); y <- rnorm(n)


# Aplicando el ejercicio:
U <- rank(x)/(n+1); V <- rank(y)/(n+1)
u <- seq(0.001,0.5,by = 0.001)
L <- Vectorize(L2emp)(u)
R <- Vectorize(R2emp)(rev(u))


plot(c(u, u + 0.5 - u[1]), c(L, R), type = "l", ylim = c(-1,1),
xlab = "Cola Inferior L           Cola Superior R",
ylab = expression(xi))
abline(v = 0.5, col = "grey")


plot(U, V, main = "Rangos normalizados\n (empates = media)", pch = 16, cex = 0.5)
plot(x, y, main = "Datos originales", pch = 16, cex = 0.5)
plot(1:length(L), L, ylim = c(-1.2, 1.2), main = "Índices", ylab = "", xlab = "", type = "l", col = "green")
legend("bottomright", legend = c("Superioir", "Inferior"), lty = c(1,1), col = c("red", "green"))
lines(R, col = "red", lwd = 2)
```

Ejemplo: índice débil II



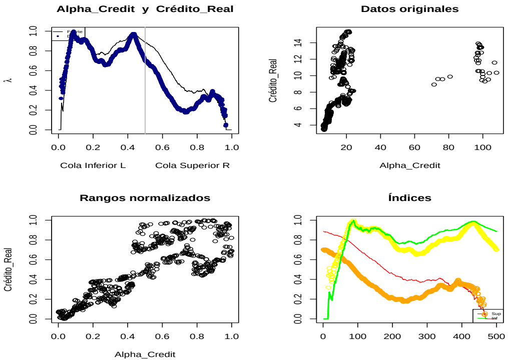
Ejemplo: índice débil III

- En el caso en el que los datos son dependientes, con $\rho = 0.9$, se puede ver que ξ está más cerca del valor 1 que de 0. Conform aumnenta la dependencia, el índice se debe acercar más al valor unitario.



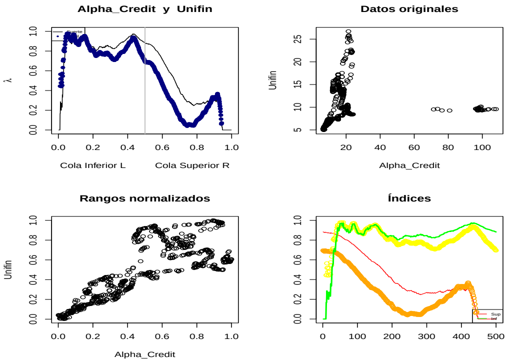
Ejemplo con datos reales I

```
[1] "Alpha_Credit 1" "Crédito_Real 2"
```



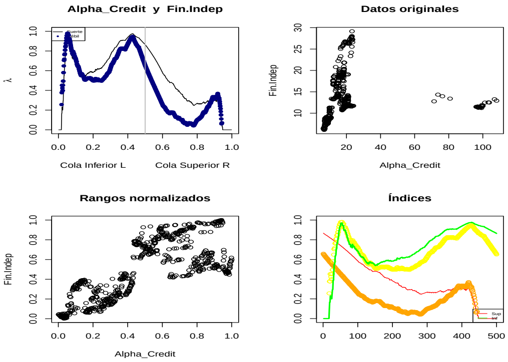

```
[1] "Alpha_Credit 1" "Unifin 3"
```

Ejemplo con datos reales II



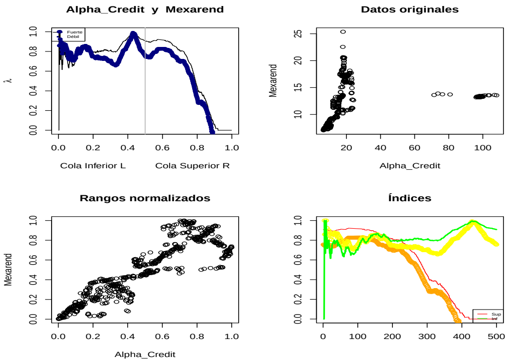

```
[1] "Alpha_Credit 1" "Fin.Indep 4"
```

Ejemplo con datos reales III

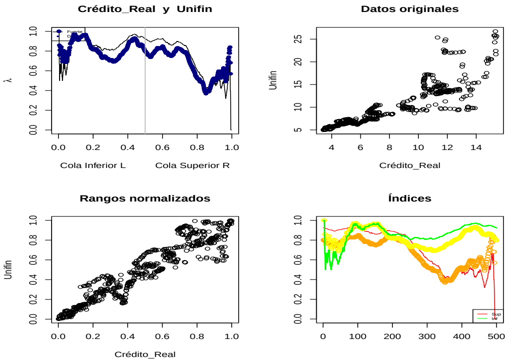



```
[1] "Alpha_Credit 1" "Mexarend 5"
```

Ejemplo con datos reales IV

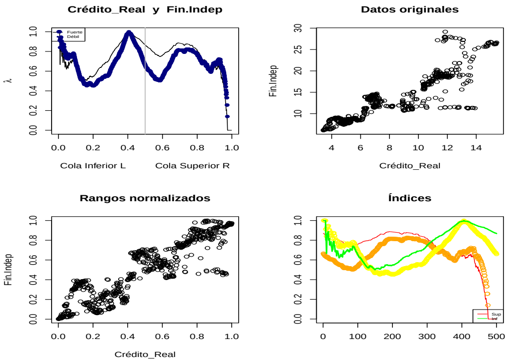



```
[1] "Crédito_Real 2" "Unifin 3"
```

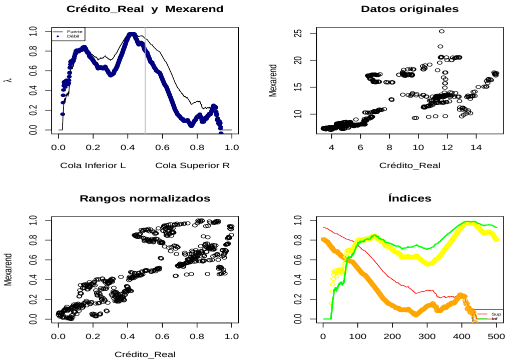



```
[1] "Crédito_Real 2" "Fin.Indep 4"
```

Ejemplo con datos reales VI

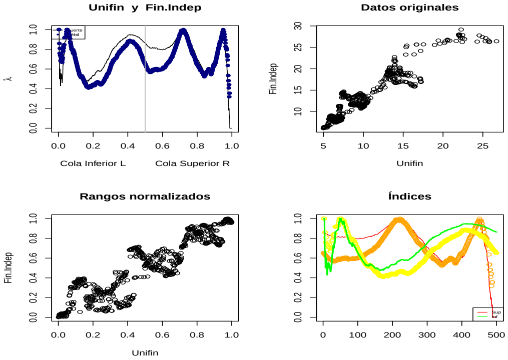



```
[1] "Crédito_Real 2" "Mexarend 5"
```



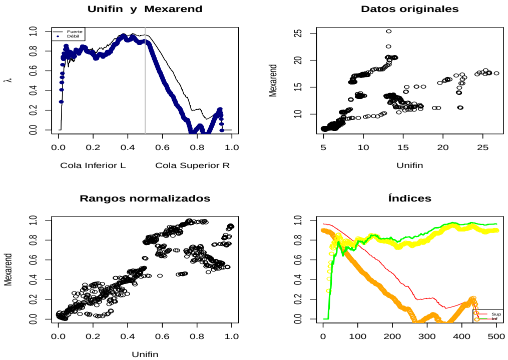

```
[1] "Unifin 3"    "Fin.Indep 4"
```

Ejemplo con datos reales VIII



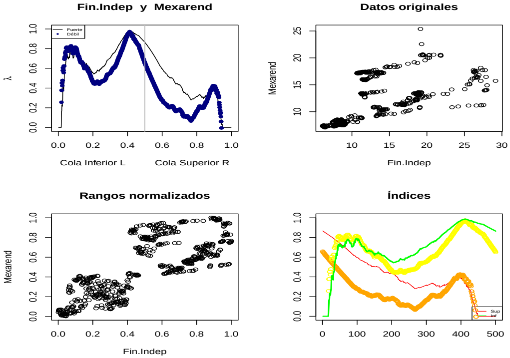

```
[1] "Unifin 3"   "Mexarend 5"
```

Ejemplo con datos reales IX




```
[1] "Fin.Indep 4" "Mexarend 5"
```

Ejemplo con datos reales X



Referencias

Bibliografía

- Nelsen, Roger B. **An Introduction to Copulas** 2nd. Ed. Springer 2006.
- Trivedi, Pravin K. & Zimmer, David M. **Copula Modeling. An Introduction for Practitioners** Foundations and Trends in Econoemtrics, Ed. Now 2005.
- Hofert, Marius et al. **Elements of Copula Modeling with R** Springer, Use R! 2018.
- Joe, Harry. **Dependence Modeling with Copulas**. Monographs on Statistics and Applied Probability 134. CRC Press 2015.
- Drouet Mari, Dominique & Kotz, Samuel.**Correlation and Dependence**. Imperial College Press, 2001.
- Czado, Claudia. **Analyzing Dependent Data with Vine Copulas: A practical Guide with R**. Springer, 2019.
- Ruppert, David. **Statistics and Data Analysis for Financial Engineering**. Springer 2010.
- Kurowicka, Dorota & Joe, Harry, Eds. **Dependence Modeling. Vine Copula Handbook** World Scientific, 2011.
- Salmon, Felix. *The Formula that killed Wall Street*. Wired, Feb 23, 2009. Disponible en https://www.wired.com/2009/02/wp-quant/.