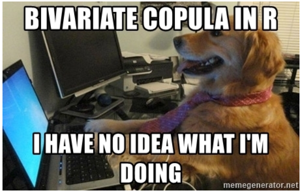
Simulación

Dependencia a través de cópula

Jorge de la Vega Góngora

Departamento de Estadística,
Instituto Tecnológico Autónomo de México

Semanas 16



2.4 Dependencia

Introducción

- En sesiones anteriores hemos visto cómo generar muestras aleatorias de varias familias de distribuciones de probabilidad.
- Una característica de estas muestras es que son *aleatorias*, es decir, son *independientes e idénticamente distribuidas* (iid).
- Sin embargo, en algunas aplicaciones la dependencia de las variables es importante y requiere ser modelada. Con MCMC podemos generar muestras dependientes, pero no tenemos control sobre esa dependencia.
- La **dependencia** es un concepto mucho más complejo que la independencia y mucho más difícil de modelar adecuadamente.
- Posibles aplicaciones:
  - dos variables de pérdida en seguros
  - agregación de exposición a riesgos
  - p rendimientos de activos en finanzas
  - velocidad máxima de viento en p localidades distintas
  - p respuestas ordinales en una prueba psicológica
  - conteo de tumores en clusters familiares
  - incumplimiento de un portafolio de créditos en un mismo sector.

Agregación de riesgos en la crisis de 2007

"Una de las lecciones más importantes aprendida durante la crisis financiera global en 2007 fue que la tecnología de información de los bancos (IT) y arquitectura de datos fue inadecuada para soportar la amplia gestión de los riesgos financieros. Muchos bancos no tenían la capacidad para agregar las exposiciones al riesgo e identificar concentraciones rápida y adecuadamente al nivel de grupo de bancos, a través de las líneas de negocio y entre entidades legales. Algunos bancos fueron incapaces de manejar sus riesgos propiamente debido a la débil agregación de datos de riesgo y prácticas de reporteo. Esto tuvo severas consecuencias para los bancos mismos y para la estabilidad del sistema financiero como un todo."

BCBS Enero 2013

Lo anterior enfatiza la importancia de la dependencia en finanzas:

- La probabilidad de que un evento puede estar relacionado a la realización de otro evento
- En bancos y en aseguradoras esto surge principalmente entre y a través de:
  - riesgos de crédito (activo y contraparte)
  - riesgos de inversión (activo)
  - riesgo operacional
  - riesgo de subidentificación

Dependencia I

- Imaginen que quieren generar un par de variables aleatorias X y Y, con distribución conjunta $F(X, Y)$ y con marginales F_X y F_Y, y quieren que tenga una estructura de dependencia dada. Esto da origen a algunas preguntas:
    - ¿La conjunta determina a las marginales de manera única? Sí
    - ¿Las marginales determinan a la conjunta de manera única? No
    - ¿Las marginales y correlación determinan de manera única a la conjunta? No en general, sólo en el caso de distribuciones normales o elípticas.
- ¿Qué debemos entender por dependencia (estocástica)? La definición usual es en términos de la función de distribución conjunta, sus condicionales y sus marginales:

$$F(X, Y) = F(X|Y)F_Y(Y) = F(Y|X)F_X(X)$$

- Otra manera es pensar en la correlación. ¿Es la dependencia entre variables su correlación? No, la dependencia es un concepto mucho más complicado.
- Hay muchas maneras de medir/modelar la dependencia.
- El coeficiente de correlación de Pearson es una manera, pero no siempre es la mejor.
- Las cópulas pueden ser mejores, si se aplican eficientemente (es decir, si se tiene un buen número de observaciones, etc.)

Dependencia: ejemplos

- Lanzamiento de un dado dos veces. X_1 es el resultado del primer lanzamiento y X_2 el del segundo lanzamiento.
- Si los lanzamientos son independientes, entonces el conocimiento de X_1 no da información sobre X_2:

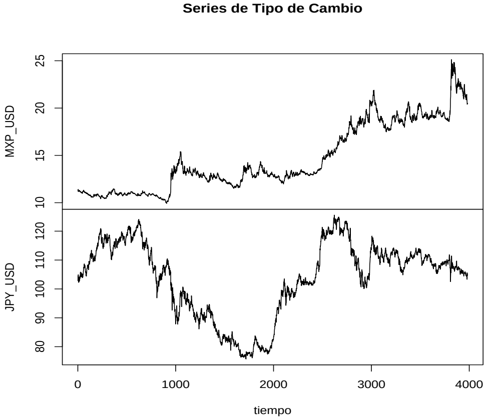
$$P(X_1 < x_1, X_2 < x_2) = F(x_1)F(x_2)$$

- Ahora supongan que Y_1 es el resultado menor de los dos lanzamientos, y Y_2 es el resultado mayor. En este caso, la información del primer lanzamiento afecta al segundo, por ejemplo si $Y_1 = 6$, necesariamente $Y_2 = 6$. Si $Y_1 = 5$, entonces $Y_2 = 5$ o 6 con probabilidades iguales a $1/2$.
  En este ejemplo, claramente las marginales no nos dan información de manera separada sobre la conjunta. En el caso de que Y_1 es el mínimo y Y_2 es el máximo,

$$P(Y_1 \leq y_1, Y_2 \leq y_2) = 2F(y_1)[F(y_2) - F(y_1)]$$

Ejemplo

```r
mxpusd <- Quandl("FED/RXI_N_B_MX", start_date = "2005-01-01")
jpyusd <- Quandl("FED/RXI_N_B_JA", start_date = "2005-01-01")
n <- length(mxpusd[,1])
plot.ts(cbind(MXP_USD = mxpusd[n:1,2], JPY_USD = jpyusd[n:1,2]),
                 main = "Series de Tipo de Cambio", xlab = "tiempo")
```



Ejemplo I

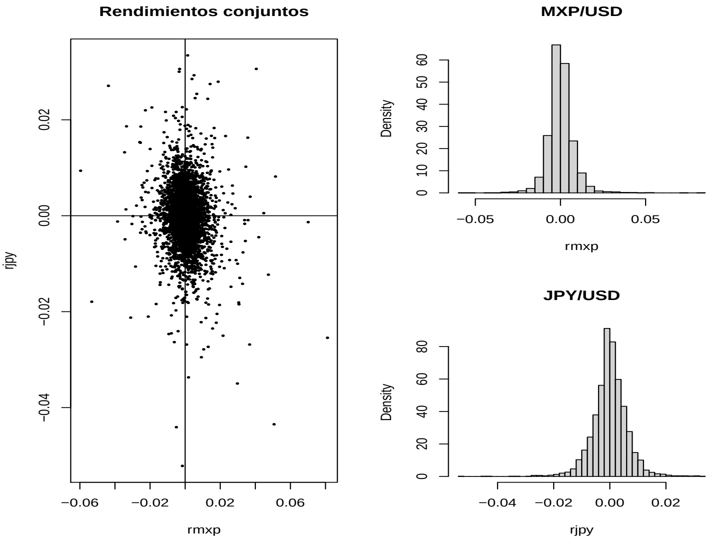
Supongamos que las variables aleatorias son los rendimientos del tipo de cambio en el periodo considerado.

```
rmxp <- diff(log(mxpusd[n:1,2])) #diferencia del log de los rendimientos
rjpy <- diff(log(jpyusd[n:1,2]))
layout(matrix(c(1,2,1,3), nrow = 2, byrow = T))
plot(rmxp, rjpy, pch = 16, cex = 0.5, main = 'Rendimientos conjuntos')
abline(h = 0)
abline(v = 0)
cor(rmxp,rjpy)

[1] -0.1098542

hist(rmxp, breaks = 50, main = "MXP/USD", prob = T)
hist(rjpy, breaks = 50, main = "JPY/USD", prob = T)
```

Ejemplo II


Rendimientos conjuntos
rjpy
0.02
0.00
−0.02
−0.04
−0.06
−0.02
0.02
0.06
rmxp
MXP/USD
Density
0
10
20
30
40
50
60
−0.05
0.00
0.05
rmxp
JPY/USD
Density
0
20
40
60
80
−0.04
−0.02
0.00
0.02
rjpy

Problemas de correlación como medida de dependencia

- La correlación es un concepto de **dependencia lineal** y por lo tanto no captura dependencias no lineales.
- Aplica naturalmente en distribuciones elípticas (multivariadas normal, t, doble exponencial, uniforme).
- Recordar que si $X_1 \perp\!\!\!\perp X_2 \implies \rho(X_1, X_2) = 0$, pero la conversa no es cierta en general. Sólo es cierta en la distribución normal.
- Por otra parte, si $\rho(X_1, X_2) = \pm 1 \implies X_2 = \alpha \pm \beta X_2$.
- Muy importante: **¡Correlación no es causalidad!**
- La correlación es una medida limitada de dependencia, y en finanzas su estimación falta de robustez y/o estabilidad en el tiempo.

Problema:
Obtener una muestra de un vector normal multivariado de orden 3: $\mathbf{X} = (X_1, X_2, X_3) \sim \mathcal{N}_3\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ donde $\boldsymbol{\Sigma}$ es una matriz de covarianzas que induce una estructura de dependencia entre las variables componentes: Por ejemplo $\boldsymbol{\mu} = (1, 2, 3)'$ y $\boldsymbol{\Sigma} = \begin{pmatrix} 2.5 & 0.75 & 0.175 \\ 075 & 0.7 & 0.135 \\ 0.175 & 0.135 & 0.43 \end{pmatrix}$.

Solución.

- En este caso, la correlación es suficiente. Este problema se puede resolver estandarizando. Para estandarizar, necesitamos descomponer la matriz de covarianzas en su "raíz cuadrada": tenemos que encontrar $\mathbf{B} \ni \mathbf{B}\mathbf{B}' = \boldsymbol{\Sigma}$.

$$\mathbf{Z} \sim \mathcal{N}_p\left(\mathbf{0}, \mathbf{I}_p\right) \implies \mathbf{X} = \boldsymbol{\mu} + \mathbf{B}\mathbf{Z} \sim \mathcal{N}_p\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

- Entonces la estructura de dependencia en esta distribución está contenida en $\mathbf{B}$. A partir de normales independientes podemos encontrar las variables que nos interesan.

Ejemplo 2 II

- El siguiente script muestra como obtener $\mathbf{B}$ con diagonalización:

```
Sigma <- matrix(c(2.5, 0.75, 0.175,
                  0.75, 0.7, 0.135,
                  0.175, 0.135, 0.43),byrow=T,nrow=3)
e <- eigen(Sigma)
v <- e$vectors
B <- v %*% diag(sqrt(e$values)) %*% t(v)
B
```

```
           [,1]       [,2]       [,3]
[1,] 1.54657939 0.32165372 0.06805199
[2,] 0.32165372 0.76821450 0.07990852
[3,] 0.06805199 0.07990852 0.64728939
```

```
# Ahora generamos una muestra aleatoria de X:
Z <- matrix(rnorm(300),nrow=100,ncol=3,byrow=T)
X <- c(1,2,3) + Z %*% B
head(X,4)
```

```
         [,1]       [,2]     [,3]
[1,] 1.467286  1.8547934 3.706927
[2,] 1.129679  3.3062568 1.812088
[3,] 1.786511 -0.6309437 1.459998
[4,] 3.757713  3.4896506 3.541380
```

$\square$

Esta solución no es generalizable a otras distribuciones que no sean elípticas.

Problemas de correlación como medida de dependencia

Algunos problemas adicionales de la correlación como medida de dependencia son los siguientes:

- **La correlación no es invariante bajo transformación de variables**. Por ejemplo:

$$\mathit{cor}(X_1, X_2) \neq \mathit{cor}(\log(X_1), \log(X_2))$$

- **Valores factibles para la correlación dependen de las distribuciones marginales**. Por ejemplo si X_1 y X_2 son lognormales, entonces ciertos valores de la correlación son imposibles, eg: ($\log(X_1) \sim \mathcal{N}(0,1)$, $\log(X_1) \sim \mathcal{N}\left(0,\sqrt{2}\right)$ y $\mathit{cor}(\log(X_1), \log(X_2)) = 0.7$ no existe).
- **Una dependencia lineal perfecta no implica una correlación de 1**.
- En general, **Correlación 0 no implica independencia**.
- Entonces necesitamos otras formas de medir dependencia que tome en cuenta su estructura, y ésta se definirá a través de la función cópula.

Datasaurius I

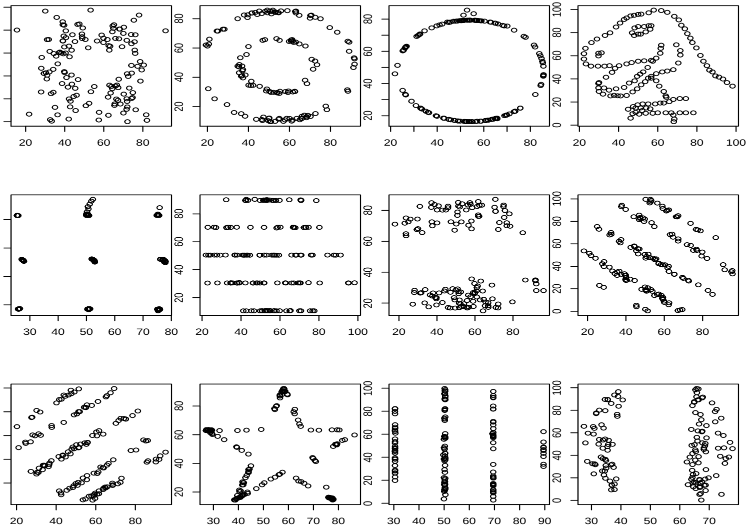
Ver: Same Stats, Different Graphs: Generating Datasets with Varied Appearance and Identical Statistics through Simulated Annealing
En este conjunto de datos todos los pares de puntos tienen la misma correlación (difieren a dos cifras decimales). También las marginales son diferentes en cada caso

```
temp <- tempfile()
download.file("file:///home/jvega/Dropbox/Academia/ITAM/2020-II/Sim_S20-II/data/The Datasaurus Dozen.zip",temp)
datos <- read.csv(unz(temp, "The Datasaurus Dozen/DatasaurusDozen-wide.tsv"),
sep="\t",header=T,skip=1)
unlink(temp)
par(pty="s");par(mfrow=c(3,4));par(mar=c(1,1,1,1))
rho <- NULL
for(i in 0:11) rho[i+1] <- cor(datos[,2*i+1],datos[,2*(i+1)])
rho
```

```
 [1] -0.06412835 -0.06858639 -0.06834336 -0.06447185 -0.06034144 -0.06171484
 [7] -0.06850422 -0.06897974 -0.06860921 -0.06296110 -0.06944557 -0.06657523
```

```
for(i in 0:11) plot(datos[,2*i+1],datos[,2*(i+1)],xlab="",ylab="")
```

Datasaurius II



Cópulas

Definición de cópula

Cópulas

Una cópula es una función de distribución conjunta $C[0,1]^n \rightarrow [0,1]$ 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, Y) con distribución conjunta F y marginales $F_X(x)$ y $F_Y(y)$.
- Definan

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

Entonces C es una cópula:

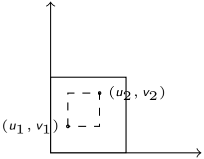
$$\begin{aligned} C(u,v) &= F(F_X^{-1}(u), F_Y^{-1}(v)) \\ &= P(X \leq F_X^{-1}(u), Y \leq F_Y^{-1}(v)) \\ &= P(F_X(X) \leq u, F_Y(Y) \leq v) \\ &= P(U \leq u, V \leq v) \end{aligned}$$

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

Cópulas y sus propiedades

- 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.
- Las propiedades anteriores caracterizan a las cópulas.

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.
- Para un vector aleatorio $\mathbf{X} = (X_1, X_2)$, noten que como C es una distribución y haciendo $u = F_1(x_1)$ y $v = F_2(x_2)$:

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

  Esto es, $F_{\mathbf{X}}(x_1, x_2) = C(F_1(x_1), F_2(x_2))$. De este modo, $F_{\mathbf{X}}$ se "descompone" en dos partes: una cópula C que contiene información de las dependencias de $\mathbf{X}$, y las marginales.
- 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 $F_1, \ldots, F_p$. Entonces $\exists$ una cópula $C : [0,1]^p \to [0,1] \ni \forall \mathbf{x} \in \mathbb{R}^p$

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

Además, si las marginales son continuas, entonces C es única. Conversamente, si C es una cópula y $F_1, \ldots, F_p$ son funciones de distribución, entonces F como se definió arriba, es una distribución con marginales $F_1, \ldots, F_p$.

Abe Sklar

Uso de las cópulas

- En resumen, usamos las cópulas para especificar una distribución conjunta en un proceso de dos etapas:
  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.

En las secciones y ejemplos siguientes utilizaremos los siguientes paquetes de `R`: `copula` y `mvtnorm`.

Cópula de independencia. I

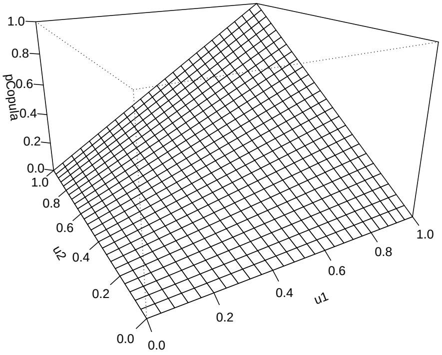
Definamos $C : [0,1]^2 \to [0,1]$ como $C(u_1, u_2) = u_1 u_2$. Entonces, Si F_1 y F_2 son distribuciones,

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

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

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

Cópula de independencia. II



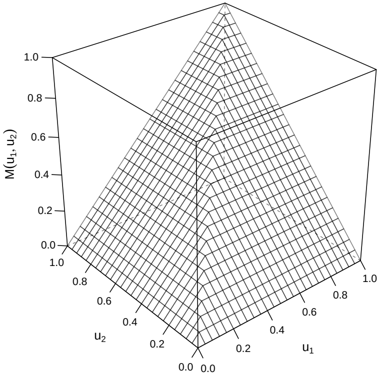
Cópula de co-monotonicidad I

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

$$C^M(u_1, u_2) = P(U \le u_1, U \le u_2) = P(U \le \min\{u_1, u_2\}) = \min\{u_1, u_2\}$$

```
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)
```

Cópula de co-monotonicidad II



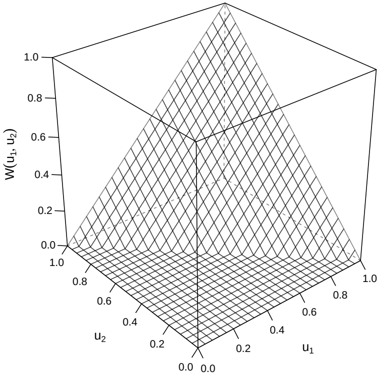
Cópula de contra-monotonicidad I

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

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

```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)
```

Cópula de contra-monotonicidad II



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, $\forall u_1, \ldots, u_n \in [0,1]$:

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

Cópula Gaussiana I

Si $Z \sim \mathcal{N}_2\left(\mathbf{0}, \mathbf{\Sigma}\right)$ 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

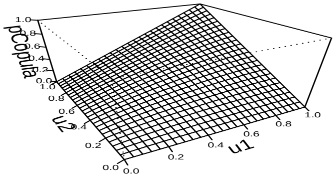
$$C(u_1, u_2) = \Phi_{\mathbf{\Sigma}}(\Phi^{-1}(u_1), \Phi^{-1}(u_2))$$

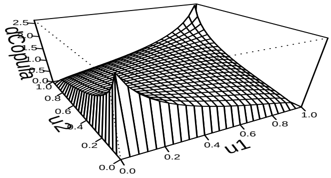
es una cópula.
La fórmula explícita de la cópula es:

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

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

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


pCopula
u1
u2
0.0
0.2
0.4
0.6
0.8
1.0


dCopula
u1
u2
0.0
0.5
1.0
1.5
2.0
2.5
0.2
0.4
0.6
0.8

Método para construir variables dependientes (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. Por ejemplo, para el caso de la cópula gaussiana:

1. Generar $(Z_1, Z_2) = \mathbf{Z} \sim \mathcal{N}_2\left(\mathbf{0}, \mathbf{\Sigma}(\rho)\right)$. 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 $(X_1, X_2) = \mathbf{X} \sim (F^{-1}(u_1), G^{-1}(u_2))$ [1]

X_1 y X_2 son 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 sean invariantes ante transformaciones no lineales que veremos más adelante.

[1] Notar que la cópula normal se usa en este punto, ya que $C(u_1, u_2) = \Phi(\Phi^{-1}(u_1), \Phi^{-1}(u_2)) = \Phi(\Phi^{-1}(\Phi(Z_1)), \Phi^{-1}(\Phi(Z_2)) = \Phi(Z_1, Z_2) = \mathcal{N}_2\left(\mathbf{0}, \mathbf{\Sigma}(\rho)\right)$.

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 a las mismas fuerzas del mercado. Sin embargo, si los subyacentes son de diferentes mercados (hipotecario y de tasas de interés) 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
```

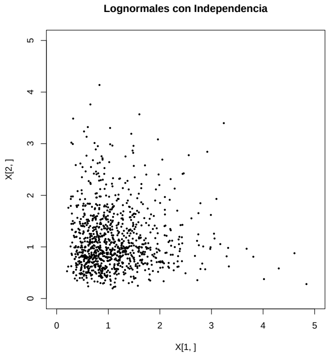
```
#Genera las muestras de dos normales
z <- matrix(rnorm(2*n),nrow=2)
#Transforma para escalar las variables
Y <- sigma*diag(2) %*% z
X <- exp(Y) #X tiene distribución lognormal
dim(X)
```

```
[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.

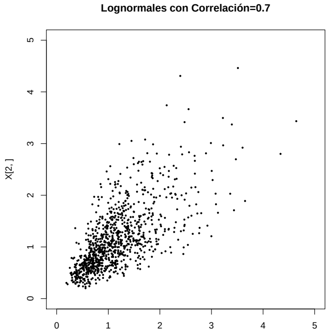
Podemos introducir dependencia entre las variables a través de un coeficiente de correlación en las normales bivariadas. En este caso se puede observar que valores mayores o menores de las dos variables tienden a estar más asociados que en el caso anterior.

```
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.

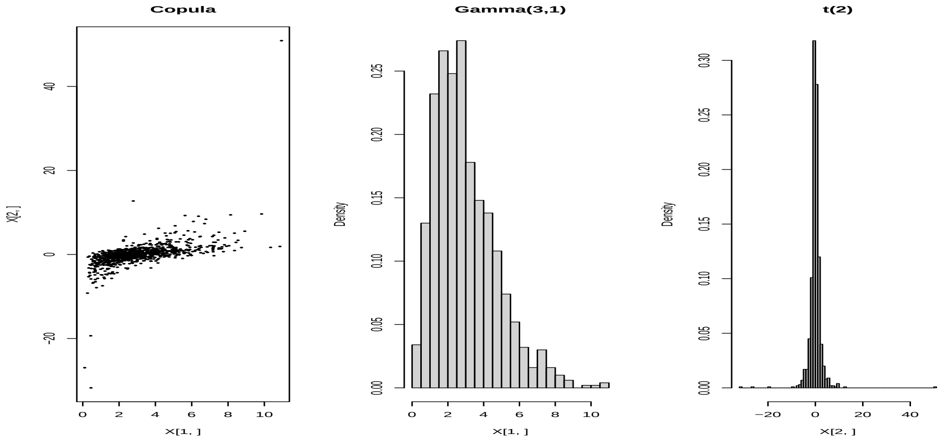
Generación de variables aleatorias usando cópula gaussiana I

```
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)); cor(X[1,],X[2,])

[1] 0.5327403

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)
```

Generación de variables aleatorias usando cópula gaussiana II



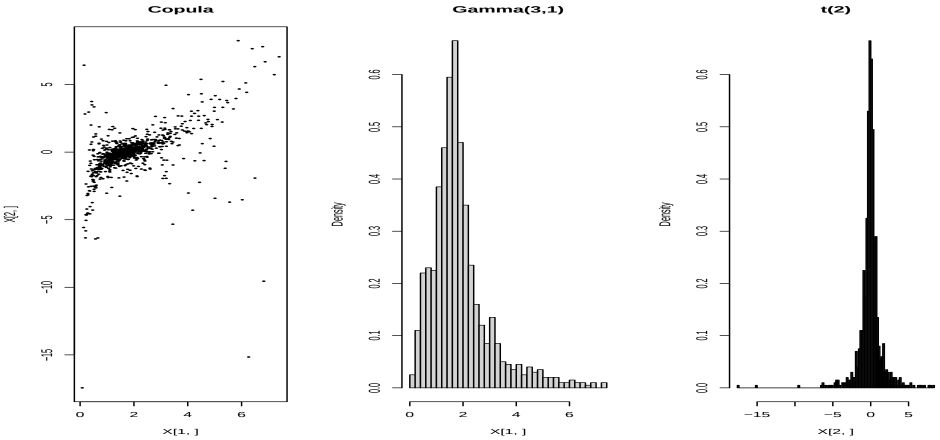
Generación de variables aleatorias usando cópula t I

¿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)); cor(X[1,],X[2,])

[1] 0.4727318

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)
```


Copula
X[2,]
X[1,]
5
0
-5
-10
-15
0
2
4
6
Gamma(3,1)
Density
X[1,]
0.0
0.1
0.2
0.3
0.4
0.5
0.6
t(2)
Density
X[2,]
−15
−5
0
5

Cópula t.

Si sustituimos a $Z = (Z_1, Z_2)$ por una variable bivariada $T_{(\mathbf{\Sigma},n)} = (t_1, t_2)$ donde n 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.

Familia Arquimediana de Cópulas

Cópulas Arquimedianas

Una cópula Arquimediana con generador ϕ tiene la forma:

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

donde ϕ satisface:

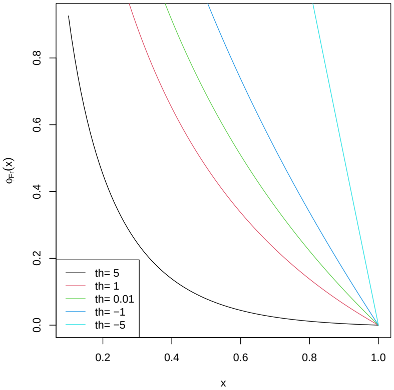
1. $\phi : [0, 1] \to [0, \infty]$ es continua y estrictamente decreciente.
2. $\phi(0) = \infty$
3. $\phi(1) = 0$

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:

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



Cópula de Frank I

Verificando las propiedades de cópula arquimediana:

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

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

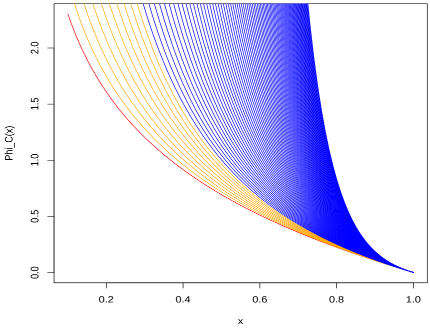
Cópula de Clayton

```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:

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



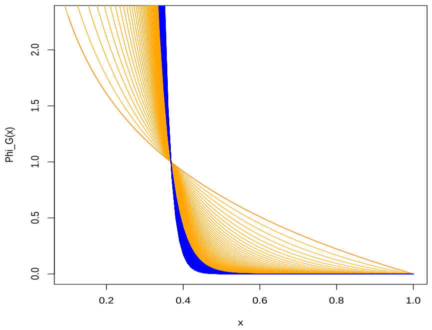
Cópula de Gumbel

```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:

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



Medidas de dependencia

Rango de una variable aleatoria

- Se mencionó que la correlación de Pearson

$$\rho(X, Y) = \frac{\mathrm{cov}\,(X, Y)}{\sigma_X \sigma_Y}$$

  es una medida de dependencia limitada, ya que depende tanto de la distribución conjunta como de las marginales.
- 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:

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

- 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.

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 sistemáticamente P4 y P5 simultáneamente.

τ de Kendall

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. 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)
```

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:

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

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

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

Ejemplo

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 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
```

Medidas de dependencia en términos de cópulas

- Las medidas de correlación aún toman valores entre -1 y 1.
- $\rho_\tau = \rho_S = 0$ para variables independientes (la conversa no se cumple)
- $\rho_\tau = \rho_S = 1$ cuando X y Y son comonótonas.
- $\rho_\tau = \rho_S = -1$ cuando X y Y son contramonótonas.
- En términos de cópulas, se puede ver que:

$$\rho_\tau = 4\int_0^1\int_0^1 C(u,v)dC(u,v) - 1$$

y

$$\rho_S = 12\int_0^1\int_0^1 (C(u,v) - uv)dudv$$

- Aunque en la práctica estas fórmulas no son muy útiles, salvo en el caso normal.

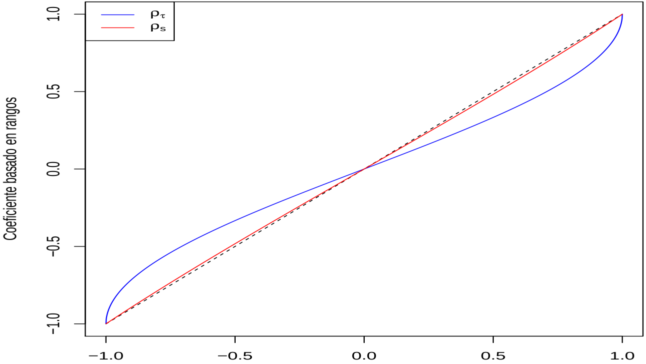
Relación de medidas de dependencia en el caso normal

En el caso de la normal bivariada, hay una relación biyectiva entre las medidas no paramétricas y el coeficiente de correlación:

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

y

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



Aplicaciones

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.

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:

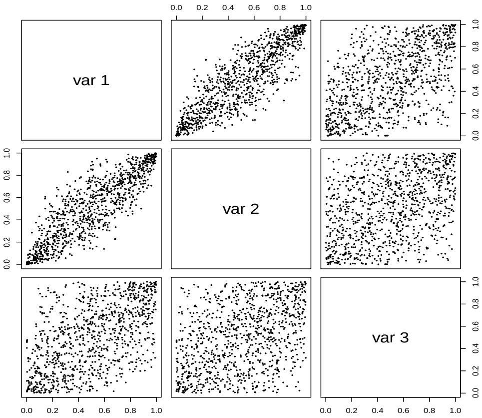
```
library(copula)
# Define el objeto cópula a generar, en este caso es una normal con correlaciones
# dadas
# El argumento dispstr se refiere a la estructura a la matriz de covarianza que caracteriza a
# la cópula. "un" es para indicar que no tiene estructura.
# ver detalle en https://www.jstatsoft.org/index.php/jss/article/view/v021i04/v21i04.pdf

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



Ejemplo IV

```
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
```

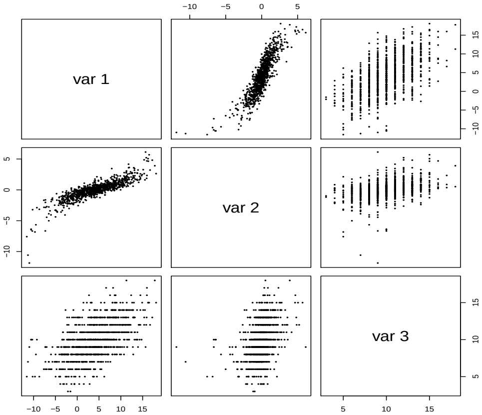
Ahora procedemos al paso 4, para generar nuestro vector y hacemos los histogramas para ver si tienen el comportamiento deseado, y se muestra un ejemplo de los valores generados:

```
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)
```

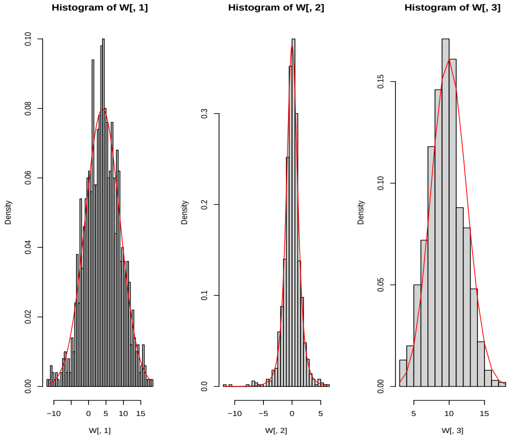
Ejemplo V



Ejemplo VI

```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")
```

Ejemplo VII



Podemos corroborar que alcanzamos la medida de dependencia requerida

Ejemplo VIII

```r
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
```

□

Simulación de cópulas bivariadas

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):

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

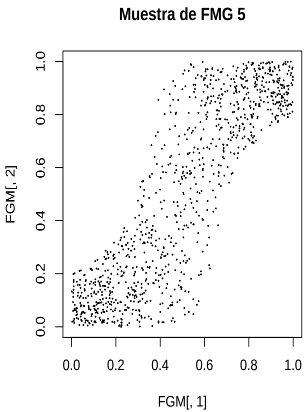
Evaluando $C_{V|U}(u, v) = v(1 + \theta(1-2u)) + \theta(1-2u)v$. Para esta cópula el algoritmo propuesto es de la forma siguiente (hay que hacer el ejercicio de inversión):

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



Paquete `copula`

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).

Cópulas de Valores Extremos

Sea $A : [0,1] \to [1/2,1]$ una función convexa que satisface la siguiente condición para $w \in [0,1]$: $\max\{w, 1-w\} \leq A(x) \leq 1$. 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.
- $A(w) = \max\{w, 1-w\}$ 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.

Clases definidas en el paquete copula I

Hay básicamente dos clases de objetos:

`copula` para definir cópulas:

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

donde las funciones F y F_i son dadas.

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

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

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
```

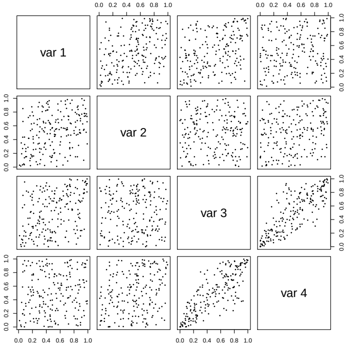
```
Normal copula, dim. d = 4
Dimension:  4
Parameters:
  rho.1   = 0.4
  rho.2   = 0.5
  rho.3   = 0.2
  rho.4   = 0.0
  rho.5   = 0.3
  rho.6   = 0.8
dispstr:  un
```

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

```
          [,1]       [,2]       [,3]      [,4]
[1,] 1.0000000 0.35972684 0.38090721 0.1184919
[2,] 0.3597268 1.00000000 0.00595825 0.2695009
[3,] 0.3809072 0.00595825 1.00000000 0.8157029
[4,] 0.1184919 0.26950087 0.81570291 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:

```r
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
```

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

```
           [,1]       [,2]       [,3]
[1,] 0.08814359 0.08825308 0.25674456
[2,] 0.76893760 0.82148435 0.54899528
[3,] 0.58092339 0.19711718 0.09534241
[4,] 0.38986680 0.72322997 0.58192259
[5,] 0.11925949 0.09054355 0.43658931
```

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)
```

Clase copula IV

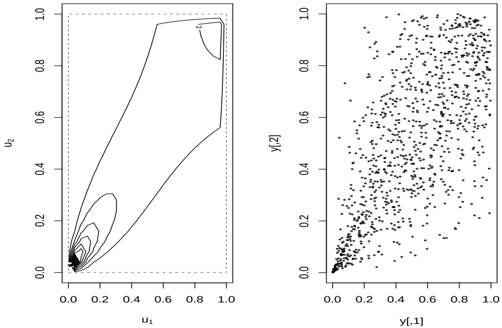


Tabla de cópulas Arquimedianas I

Familia	Espacio paramétral θ	$\psi(t)$	$\psi^{-1}(t)$	$C(u, v)$
Clayton	$\theta \geq 0$	$\frac{t^{-\theta}-1}{\theta}$	$(1+\theta t)^{-1/\theta}$	$(u^{-\theta}+v^{-\theta}-1)^{-1/\theta}$
Frank	$\theta \geq 0$	$-\log\frac{e^{-\theta t}-1}{e^{-\theta}-1}$	$-\theta^{-1}\log(1+e^{-t}(e^{-\theta}-1))$	$-\frac{1}{\theta}\log\left(1+\frac{(e^{-\theta u}-1)e^{-\theta v}-1)}{e^{-\theta}-1}\right)$
Gumbel	$\theta \geq 1$	$(-\log t)^{\theta}$	$e^{-t^{1/\theta}}$	$\exp\left[-((-\log u)^{\theta}+(-\log v)^{\theta})^{1/\theta}\right]$

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.

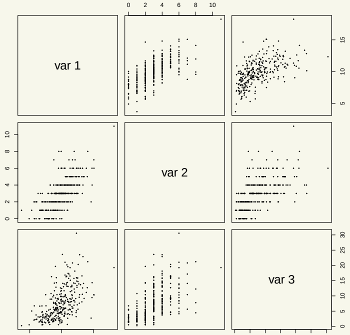
  `margins`: Especifica los nombres de las distribuciones marginales a usar.

  `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)
```



Para la distribución conjunta creada a partir de la cópula

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

          [,1] [,2]      [,3]
[1,] 11.615491    3 21.357093
[2,]  9.401512    3  3.898501
[3,] 10.187742    3  5.421881
[4,]  7.911254    1  1.672087
[5,] 12.516360    3 23.015955

dMvdc(u,micopula) # puntos de la densidad

[1] 0.0004190256 0.0039733596 0.0061947430 0.0056085227 0.0001736552

pMvdc(u,micopula) # puntos de la distribución

[1] 0.58960707 0.18218564 0.30290270 0.01986233 0.62244752
```

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,] 0.83877484 0.7343751 0.8100057
[2,] 0.25521356 0.2354689 0.1400723
[3,] 0.69032156 0.4436316 0.6774955
[4,] 0.07489659 0.1660202 0.2956587
[5,] 0.87940794 0.8044847 0.7613466
```

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

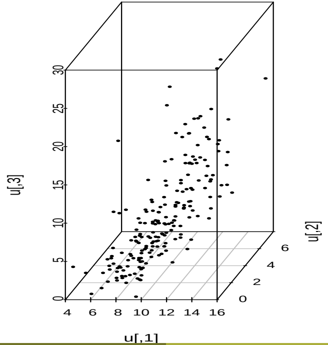
```
[1] 3.884945 3.277010 1.467077 3.075991 4.219991
```

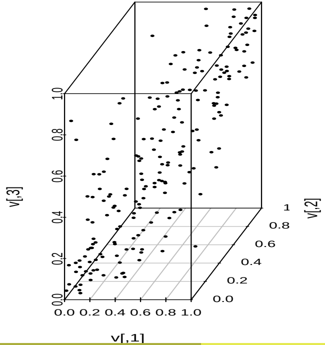
```r
pCopula(u,copula.Frank5) # puntos de la distribución
```

```
[1] 0.62463977 0.05876621 0.37303774 0.02966460 0.65066642
```

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)
```





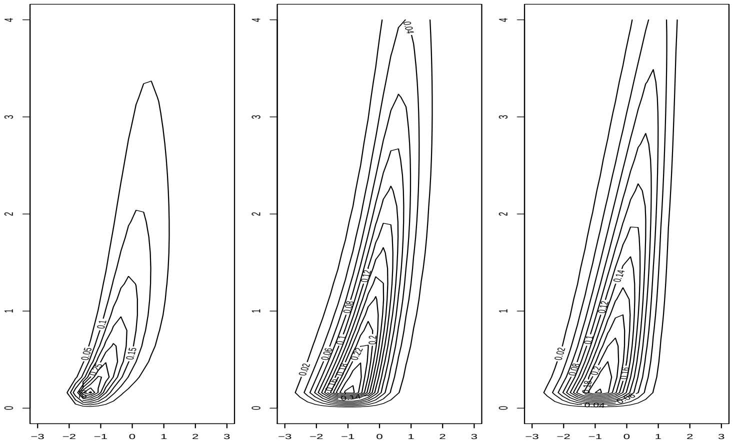
El siguiente código grafica los contornos de densidad de distribuciones bivariadas definidas con las tres cópulas de Clayton, Frank y Gumbel con marginales normales.

```
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))
```

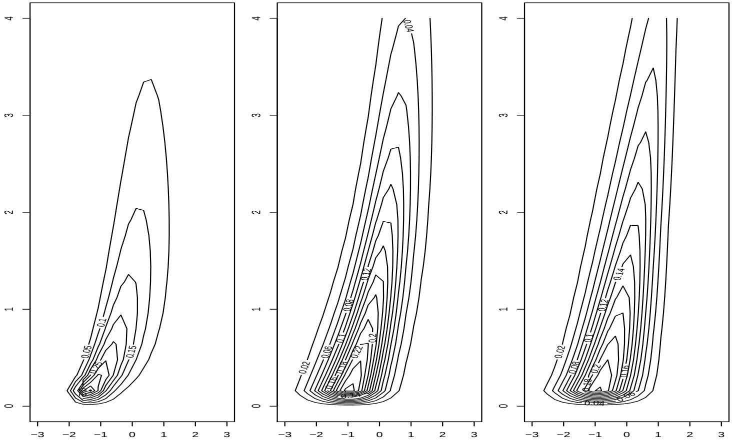
Contornos de funciones de densidad/distribución II



Contornos de funciones de densidad/distribución III

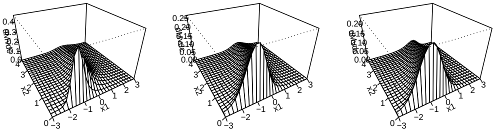
```r
par(mfrow=c(1,3), mar=c(2,2,1,1), oma=c(1,1,0,0), mgp=c(2,1,0))
contour(miMvd1, dMvdc,xlim=c(-3,3), ylim=c(0,4))
contour(miMvd2, dMvdc,xlim=c(-3,3), ylim=c(0,4))
contour(miMvd3, dMvdc,xlim=c(-3,3), ylim=c(0,4))
```

Contornos de funciones de densidad/distribución IV



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))
```



Estimación de cópulas

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
```

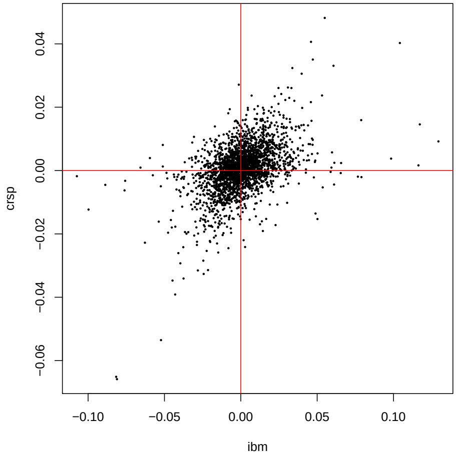
```
  year month day        ge       ibm     mobil      crsp
1 1989     1   3 -0.016760  0.000000 -0.002747 -0.007619
2 1989     1   4  0.017045  0.005128  0.005510  0.013016
3 1989     1   5 -0.002793 -0.002041  0.005479  0.002815
4 1989     1   6  0.000000 -0.006135  0.002725  0.003064
5 1989     1   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).

```r
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

```r
# 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(copula, data = data, method = "ml", start = ..3, lower = ..4,
    upper = ..5, optim.method = "L-BFGS-B")
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:

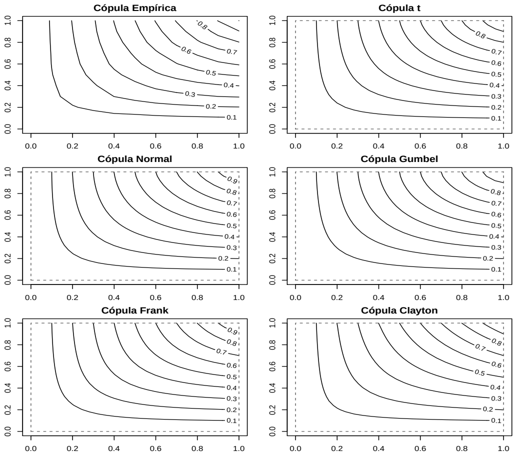
```
#Ajusta copula normal
        fnorm <- fitCopula(data = d1, copula = normalCopula(-0.3, dim = 2),
                           method = "ml", optim.method = "BFGS", start = 0.5)
#Ajusta Gumbel
        fgumbel <- fitCopula(data=d1,copula=gumbelCopula(3,dim=2),
        method="ml",optim.method="BFGS",start=2)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

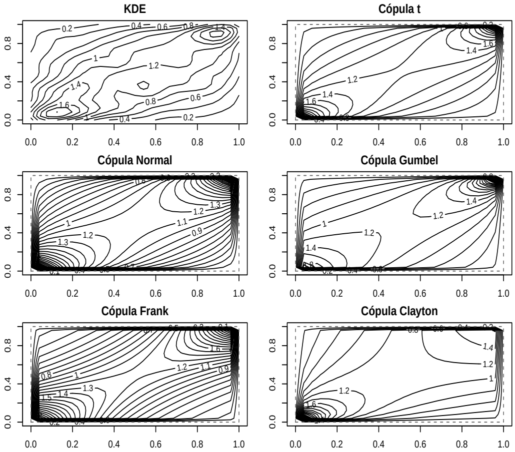
```r
u <- d1
dem <- pempiricalCopula(u[,1],u[,2])
par(mfrow=c(3,2),mar=c(2,2,2,2))
contour(dem$x,dem$y,dem$z,main="Cópula Empírica")
contour(tCopula(fit1@estimate[1],df=round(fit1@estimate[2],0)),pCopula,main="Cópula t")
contour(normalCopula(fnorm@estimate),pCopula,main="Cópula Normal")
contour(gumbelCopula(fgumbel@estimate),pCopula,main="Cópula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


KDE
Cópula t
Cópula Normal
Cópula Gumbel
Cópula Frank
Cópula Clayton
0.0
0.2
0.4
0.6
0.8
1.0

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:

```
#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
```