PROYECTO FINAL DE ESTADISTICA MATEMATICA

Armando Lara, Dario Quishpe , Jorge Arguello

2024-02-29

Contents

1 Introducción

Para una coperativa de ahorro y crédito es fundamental el análisis de La solvencia financiera de sus socios pues es un factor crítico al evaluar las distintas solicitudes de créditos permitiendo tomar decisiones sobre las mismas.Estás decisiones tienen impacto directo en los diferentes indicadores que son de gran relevancia en la institución para su monitoreo y supervisión. Dentro de estos se encuentran indicadores de mora por productos crediticios, porcentaje de colocación,riesgo crediticio ,etc.Está medida de solvencia nos proporciona una visión de la capacidad que tienen los socios para cumplir con sus obligaciones financieras, ya que en caso de un indicador de solvencia saludable sugiere una menor probabilidad de incumplimiento en el pago de deudas, así como también permite a la cooperativa tomar decisiones de crédito informadas y personalizadas,dando oportunidad de adaptar los términos del crédito u ofrecer productos convenientes según la solvencia individual.

La coperativa de ahorro y credito denominada “X” ,en vista de la importancia de la información otorgada por este indicador a optado por hacer el acompañamiento de tecnicas estadísticas a las decisiones de los analistas sobre los créditos. El presente proyecto intenta poner como un punto de partida el estudio de este indicador en un conjunto de socios.

2 Objetivos

- Generar estimaciones mediante la aplicación de las técnicas de remuestreo para realizar inferencia sobre el indicador de solvencia , con la finalidad de estimar el sesgo, la media,la varianza, un intervalo de confianza, así como realizar un contraste de una hipótesis de acuerdo a valores considerados por el jefe del área según su experiencia.
- Realizar comparaciones entre los valores estimados obtenidos del módelo parámetrico (Montecarlo) y del módelo no paramétrico (Boostrap)
- Obtener conclusiones que permitan tomar decisiones mediante los valores estimados con las técnica implementadas y compararlas con los valores considerados por el jefe del área resultantes de su experiencia.

3 Metodología

3.1 Recolección de los datos

Mediante la colaboración del jefe del área de fábrica de credito de la cooperativa "X" se obtuvo una base de datos de solicitudes de crédito con **660 registros** con corte al mes de **Agosto del 2023**, la cual contiene **35 columnas** con información sobre variables relevantes, dentro de estás las que son de nuestro principal interes son las siguientes

- Total Patrimonio Neto: Esta ofrece la imagen de la salud financiera de una persona, es un resumen de lo que se posee (bienes), menos lo que se debe a otros (pasivos).
- Activos: Un activo es una propiedad o capital propiedad de una persona o compañía que tiene un valor económico

Por cuestiones de confidencialidad de los clientes de dicha cooperativa, se omitió las variables con respecto a información personal , tomando únicamente para nuestro interés en este trabajo sólo las dos variables para generar el indicador .

##Construcción del Indicador de solvencia

Dentro de una cooperativa, es importante la representación de la capacidad financiera mediante un indicador, el patrimonio neto representa la cantidad de flujo con la que está a disposición la cooperativa para hacer frente a situaciones futuras. Por lo tanto, dividir el patrimonio neto por los activos totales proporciona una medida de la capacidad de un socio para cubrir sus obligaciones con sus activos.

Por lo tanto, se construye el indicador de solvencia cómo:

$$I = Patrimonioneto/Activos$$

3.2 Algoritmo Boostrap

En general, para el remuestreo bootstrap, seguiremos estos pasos:

1. Para cada $i = 1, \dots, n$, generar L_i^* a partir de F_n.
2. Obtener $L^* = (L_1^*, \dots, L_n^*)$.
3. Repetir B veces los pasos 1 y 2 para obtener réplicas $L^*(1), \dots, L^*(B)$.
4. Usar estas réplicas bootstrap para aproximar la distribución de remuestreo de R.

Ahora, si consideramos $\hat{\theta} = T(L)$, consideramos F la distribución conocida y definimos el estadístico

$$R(L, F) = \hat{\theta} - \theta$$

Vamos a aproximar el sesgo y la varianza:

$$Sesgo(\hat{\theta}) = E(\hat{\theta} - \theta) = E(R)$$
$$Var(\theta) = Var(\hat{\theta} - \theta)$$

1. Para cada $i = 1, \dots, n$, generar L_i^* a partir de $\hat{F}$ y obtener $L^* = (L_1^*, \dots, L_n^*)$.
2. Calcular $R^* = R(L^*, F) = \hat{\theta}^* - \hat{\theta}$.
3. Repetir B veces los pasos 1 y 2 para obtener las réplicas $R^*(1), \dots, R^*(B)$.
4. Usar las réplicas bootstrap para aproximar las características de interés:

$$Sesgo^*(\hat{\theta}^*) = \frac{1}{B}\sum_{b=1}^{B} R^*(b)$$
$$Var^*(\hat{\theta}^*) = \frac{1}{B}\sum_{b=1}^{B} (R^*(b) - \overline{R^*})^2$$

3.3 Estimación de parámetros mediante el Algoritmo EM

Mediante el algoritmo EM Expectation-Maximization, se procederá a estimar los parámetros del indicador de solvencia. Este algoritmo permitirá identificar los parámetros óptimos de la distribución (en un principio desconocida) del indicador de solvencia. Este algoritmo nos será de gran utilidad, pues una vez estimados los parámetros se los usará para hacer inferencia e identificar su distribución.

3.4 Prueba de distribución de mixturas normales

Una vez que se obtenga los parámetros como la media, la varianza y los pesosse aplicará un test estadístico apropiado para comprobar si la distribución del indicador es una Mixtura de distribuciones normales, dicho test fue realizado por Priscila Guayasamín para la clasificación de cooperativas por segmentos. En el caso en que el test rechace la hipótesis de mixturas de distribuciones normales, se procederá a la transformación de nuestros datos como una logarítmica, Box-Cox, etc.

3.5 Simulación de datos mediante los parámetros obtenidos

Si mediante el test desarrollado por Priscila se obtiene que hay evidencia estadística para aceptar que la distribución del indicador es una mixtura de normales, se procederá a realizar la simulación de nuevos datos para el indicador de solvencia, en esta simulación se incluirá los parámetros obtenidos por el algoritmo EM y con una distribución de mixtura de normales. Esto servirá para realizar intervalos de confianza del indicador así como constrastes de hipótesis.

3.6 Comparación con Montecarlo y Bootstrap

Una vez obtenidas las estimaciones de los parámetros mediante el algoritmo EM y la distribución de mixturas normales, se procede a comparar los resultados utilizando técnicas de Montecarlo y Bootstrap. Esta comparación abarca tanto las estimaciones de los parámetros como sus intervalos de confianza al 95%, así como el sesgo y el contraste de hipótesis para la media (proporciones de rechazo).

Para aplicar Montecarlo en la estimación de parámetros se realiza lo siguiente:

Generación de datos aleatorios: Generamos un conjunto de datos aleatorios para las variables patrimonio neto y activos. En nuestro caso una mixtura de normales, con los parámetros que se obtuvo del algoritmo EM.

Muestras: Realizamos múltiples muestras aleatorias de tamaño `n` de estas variables generado en el paso anterior. Estas muestras deben tomarse con reemplazo, es decir, cada observación puede ser seleccionada más de una vez en la muestra.

Construcción del indicador: Luego, para cada muestra generada de las variables, construimos un indicador y de ellas guardamos sus medias y varianzas obtenidas en un vector.

Cálculo de estimaciones finales: Calculamos la media de los estimadores obtenidos en el indicador de cada iteración, esto se hace promediando las medias y varianzas de las muestras generadas para el indicador anteriores.

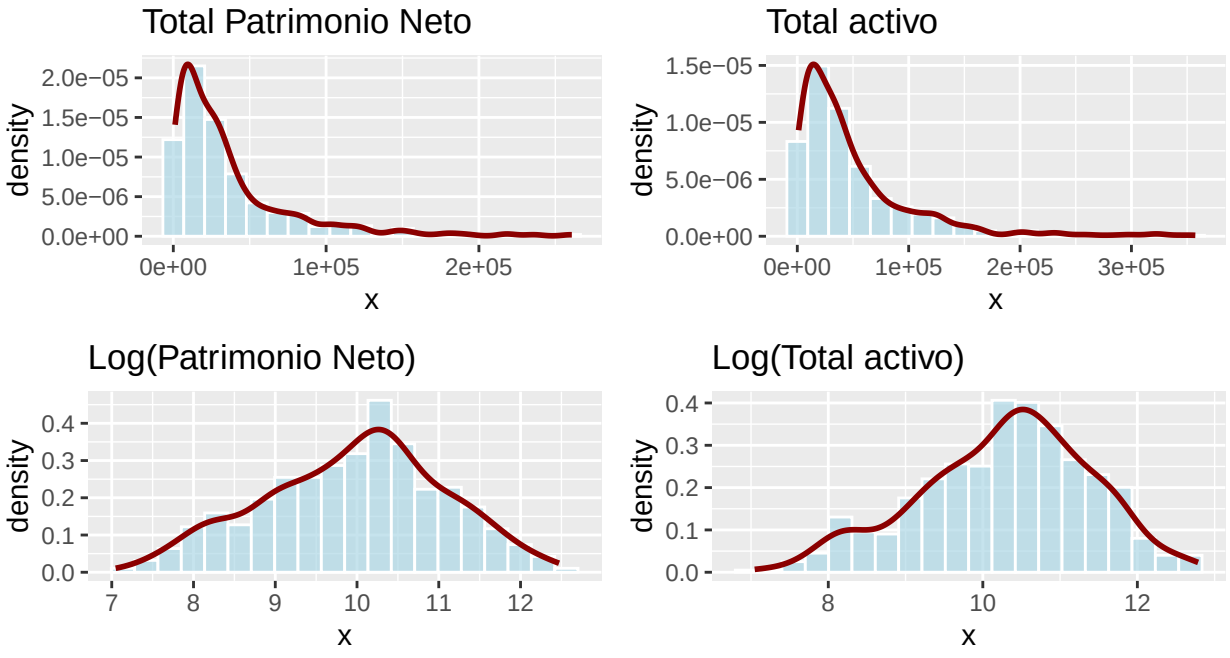
Finalmente, comparamos los resultados obtenidos con cada metología y su respectiva discusión, detallando así las ventajas y desventajas obtenidas con cada una.

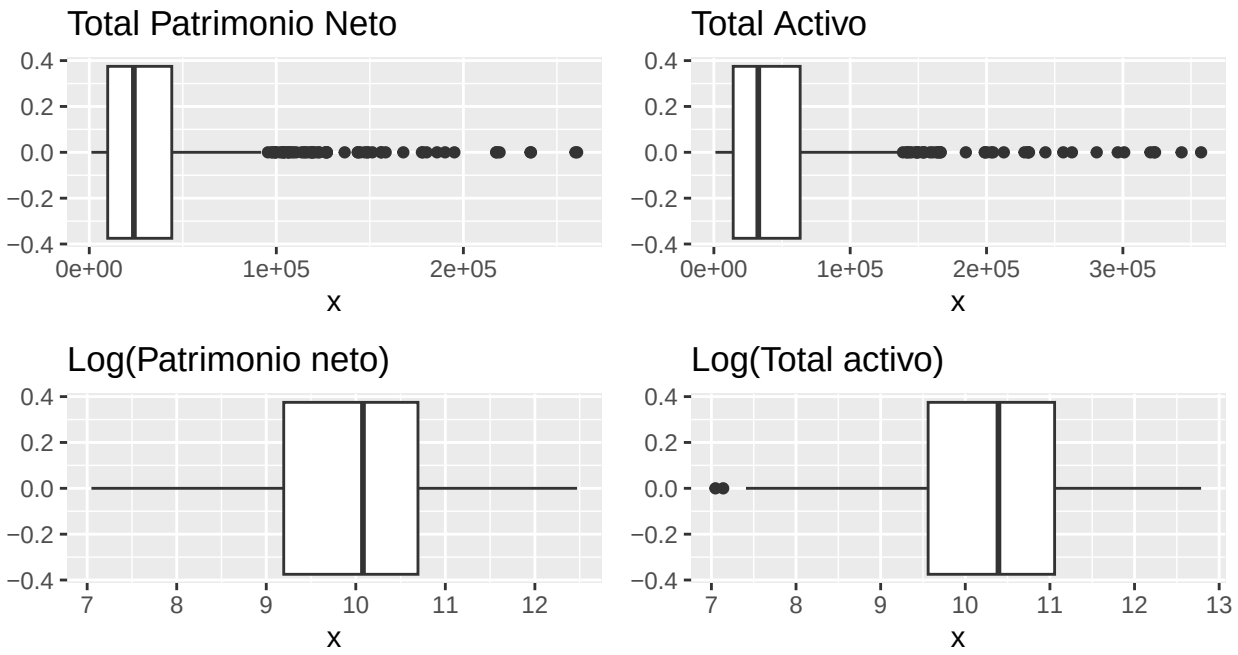
```
test_mixturasnormales<-function(data,mus,sigmapob,lambdapob){
if (!is.data.frame(data) && !is.matrix(data))
stop('data supplied must be either of class \"data frame\" or \"matrix\"')
if (dim(data)[2] < 2 || is.null(dim(data)))
{stop("data dimesion has to be more than 1")}
if (dim(data)[1] < 3) {stop("not enough data for assessing mvn")}
data.name <- deparse(substitute(data))
xp <- as.matrix(data)
p <- dim(xp)[2]
n <- dim(xp)[1]
## getting MLEs...
s.mean <- colMeans(xp)
s.cov <- (n-1)/n*cov(xp)
s.cov.inv <- solve(s.cov) # inverse matrix of S (matrix of sample covariances)
D <- rep(NA,n) # vector of (Xi-mu)'S^-1(Xi-mu)...
for (j in 1:n)
D[j] <- t(xp[j,]-s.mean) %*%(s.cov.inv %*%(xp[j,]-s.mean))
D.or <- sort(D) ## get ordered statistics
Gp <- pchisq(D.or,df=p)
## getting the value of A-D test...
ind <- c(1:n)
an <- (2*ind-1)*(log(Gp[ind])+log(1 - Gp[n+1-ind]))
AD <- -n - sum(an) / n
## getting the p-value...
N <- 1e4
U <- rep(0,N) ## initializing values of the AD test
for (i in 1:N) { ## loop through N reps
dat<-rmvnorm.mixt(1000, mus=mus, Sigmas=sigmapob, props=lambdapob)
mean1 <- colMeans(dat)
cov1 <- (n-1)/n*cov(dat)
```

```r
cov.inv <- solve(cov1) # inverse matrix of S (matrix of sample covariances)
D <- rep(NA,n) # vector of (Xi-mu)'S^-1(Xi-mu)...
for (j in 1:n)
D[j] <- t(data[j,]-mean1)%*%(cov.inv %*%(data[j,]-mean1))
Gp <- pchisq(sort(D),df=p)
## getting the value of A-D test...
an <- (2*ind-1)*(log(Gp[ind])+log(1 - Gp[n+1-ind]))
U[i] <- -n - sum(an) / n
}
p.value <- (sum(U >= AD)+1)/(N+1)
return(p.value)
}
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```






```
## [1] 660   7
```

```
## [1] 660   7
```

```
## number of iterations= 80
```

```
densidad_Originales<-dmvnorm.mixt(A,mus =mu,Sigmas = sigma,props = as.vector(em$lambda))
densidad_Simulados<-dmvnorm.mixt(dat,mus =mu,Sigmas = sigma,props = as.vector(em$lambda))
#plot_ly(x=~BASE1$Alog, y=~BASE1$Blog, z=~densidad_Originales,type = "scatter3d", color=densidad_Origin

#plot_ly(x=~dat[,1], y=~dat[,2], z=~densidad_Simulados,type = "scatter3d", mode="markers",color=densida
#layout(xaxis = list(title = "Eje X"), yaxis = list(title = "Eje Y")) |> show()
```

```
#MONTECARLO
media<-vector(length = 5000)
var<-vector(length = 5000)
sd<-vector(length = 5000)
for(i in 1:5000){
dat<-rmvnorm.mixt(5000,mus = mu,Sigmas = sigma,props = as.vector(em$lambda))
indicador_sim<-dat[,1]-dat[,2]
media[i]<-mean(indicador_sim)
var[i]<-var(indicador_sim)
sd[i]<-sd(indicador_sim)
}
liminfMonte<-quantile(media,0.025)
limisupMonte<-quantile(media,1-0.025)
MediaMonte<-mean(media)
Var_Montemean<-mean(var)
SdMonte<-mean(sd)
liminfMonte
```

```
##       2.5%
## -0.3396859
```

```
limisupMonte
```

```
##      97.5%
## -0.3179281
```

```
MediaMonte
```

```
## [1] -0.3285478
```

```
Var_Montemean
```

```
## [1] 0.1554464
```

```
SdMonte
```

```
## [1] 0.3942394
```

```
#Valores retransformados
LIINF<-exp(liminfMonte)
LIMSUP<-exp(limisupMonte)
MEDIA<-exp(MediaMonte)
VAR<-exp(Var_Montemean)
SD<-exp(SdMonte)
LIINF
```

```
##      2.5%
## 0.7119939
```

```
LIMSUP
```

```
##     97.5%
## 0.7276551
```

```
MEDIA
```

```
## [1] 0.7199685
```

```
VAR
```

```
## [1] 1.168179
```

```
SD
```

```
## [1] 1.483256
```

```r
var(BASE1$IndSolvencia)
```

```
## [1] 0.05156653
```

```r
sd(BASE1$IndSolvencia)
```

```
## [1] 0.2270827
```

```r
mean(BASE1$IndSolvencia)
```

```
## [1] 0.7664082
```

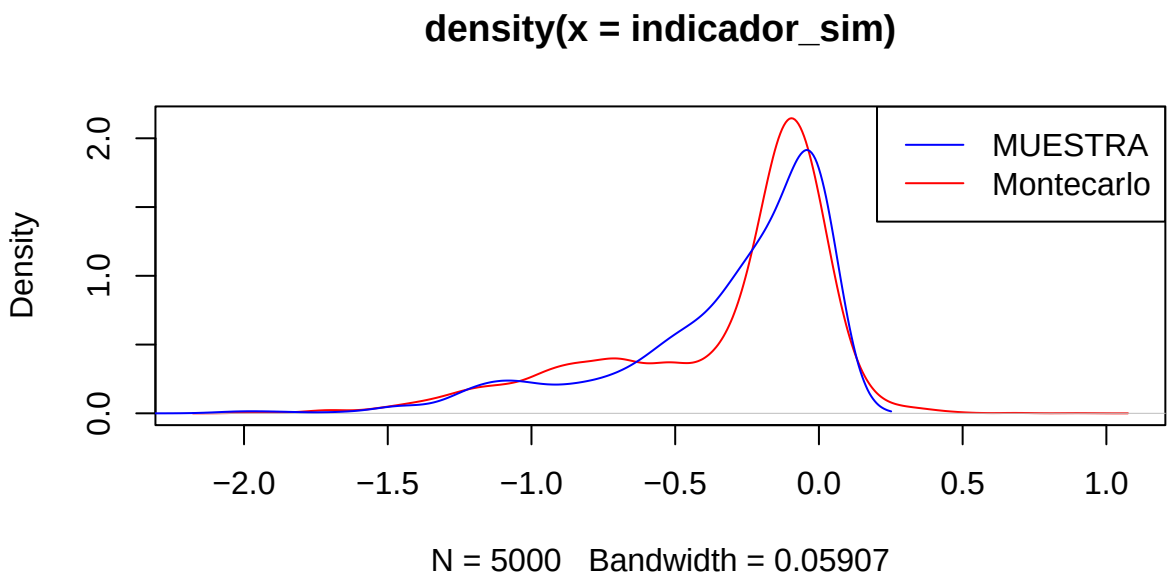


Table 1: **Estimaciones por Montecarlo y Bootstrap (Datos Transformados)**

Método/Medida	Media	Varianza	Sd	LINF	LimSUP
Montecarlo	-0.3285	0.1554	0.3942	-0.3396859	-0.3179281
Bootstrap	-0.328538	0.235	0.221	-0.3589944	-0.2990529

Table 2: **Estimaciones por Montecarlo y Bootstrap (Datos Originales)**

Método	Media	Var	Sd	LINF	LSUP
Montecarlo	0.719	1.1681	1.483255	0.71326	0.72658
Bootstrap	0.766	0.05147959	0.2268183	0.7491227	0.7837504

```r
#SESGO
SesgoMedia<-MEDIA-mean(BASE1$IndSolvencia)
SesgoVar<-VAR-var(BASE1$IndSolvencia)
SesgoSd<-SdMonte-sd(BASE1$IndSolvencia)

SesgoMedia
```

Table 3: **Estimaciones por Montecarlo y Bootstrap (Datos Originales)**

Método	SESGO MEDIA	SESGO Varianza	SESGO Sd
Montecarlo	-0.0464	1.1166	0.1671
Bootstrap	-5.237828e-05	8.694168e-05	0.0002643589

```
## [1] -0.04643974
```

```
SesgoVar
```

```
## [1] 1.116613
```

```
SesgoSd
```

```
## [1] 0.1671568
```

```
#CONTRASTE DE HIPOTESIS datos transformados(log)
#HO:mu=-0.34
numsimu<-1000
pvalue<-numeric(numsimu)
for (i in 1:numsimu){
dat<-rmvnorm.mixt(5000,mus = mu,Sigmas = sigma,props = as.vector(em$lambda))
indicador_sim<-dat[,1]-dat[,2]
ind<-(indicador_sim)
estadistico<-(mean(ind)+0.34)/(sd(ind)/sqrt(5000))
p<-1-pt(abs(estadistico),5000-1)+pt(-abs(estadistico),5000-1)
pvalue[i]<-p
}

cat("\nProporción de rechazos al 1%=",mean(pvalue<0.01),"\n")
```

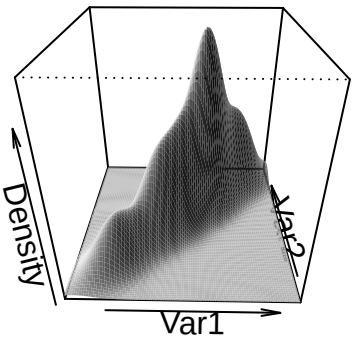
```
##
## Proporción de rechazos al 1%= 0.306
```

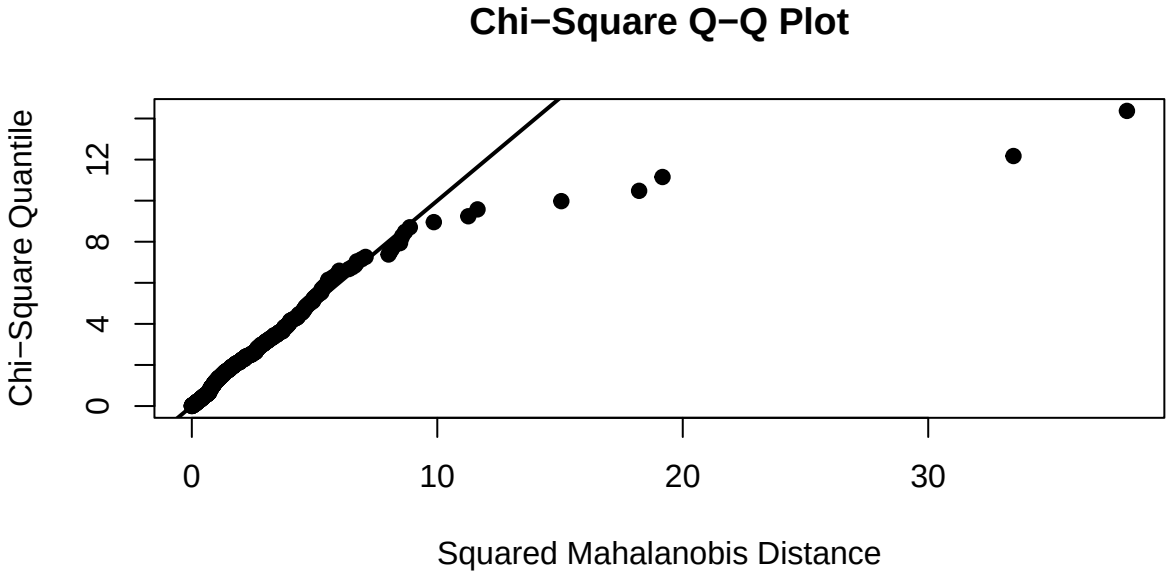
```
cat("\nProporción de rechazos al 5%=",mean(pvalue<0.05),"\n")
```

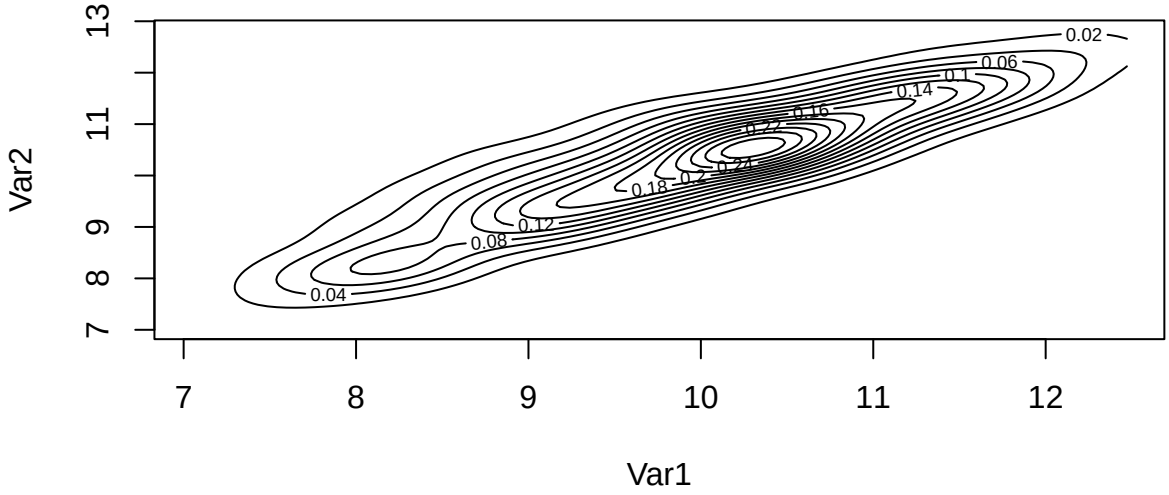
```
##
## Proporción de rechazos al 5%= 0.529
```

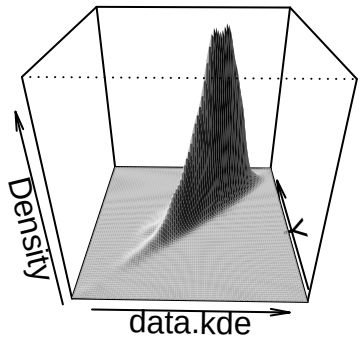
```
cat("\nProporción de rechazos al 10%=",mean(pvalue<0.1),"\n")
```

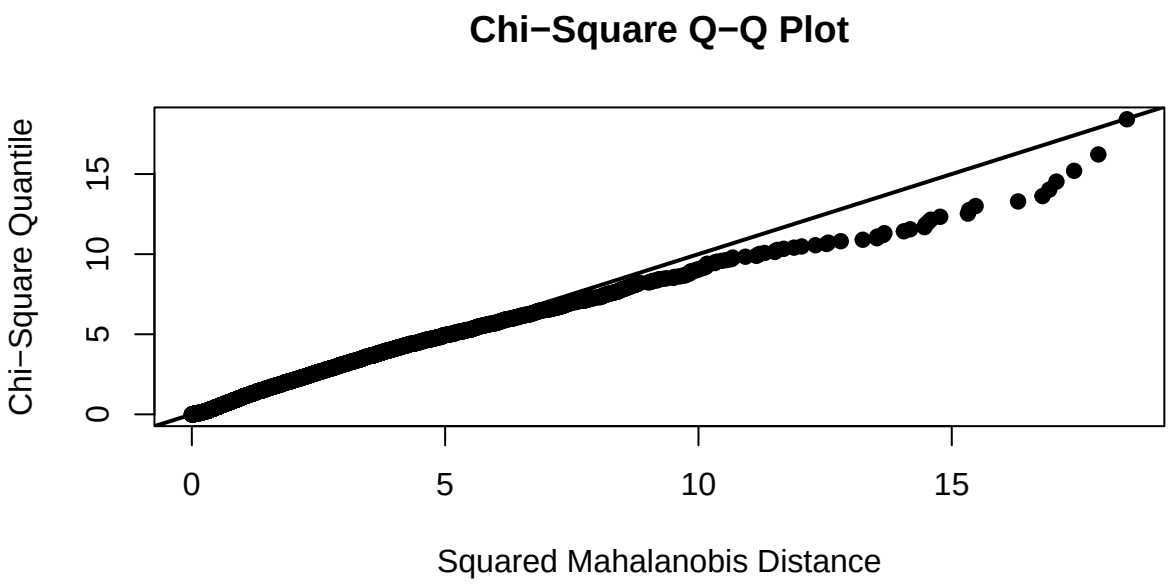
```
##
## Proporción de rechazos al 10%= 0.658
```

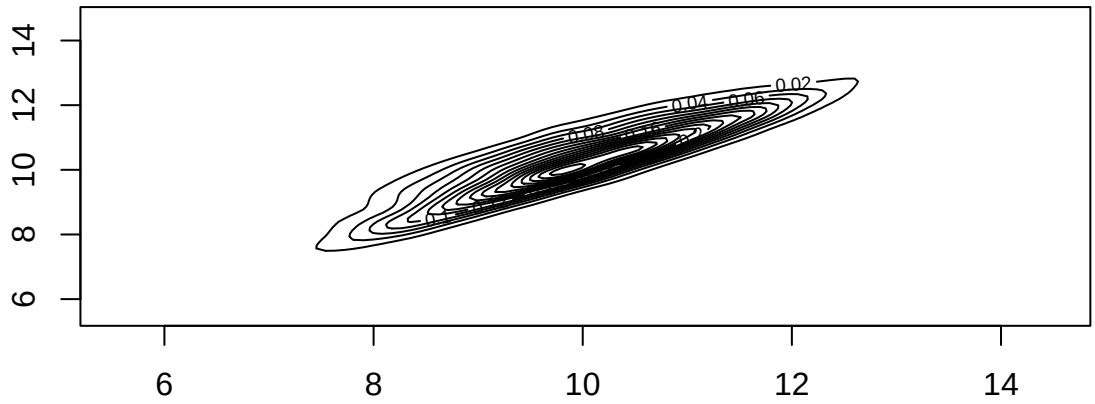

Density
Var2
Var1


Chi−Square Q−Q Plot
Chi−Square Quantile
0
4
8
12
0
10
20
30
Squared Mahalanobis Distance


Var2
13
11
9
8
7
0.02
0.06
0.1
0.14
0.16
0.22
0.2
0.24
0.18
0.12
0.08
0.04
7
8
9
10
11
12
Var1


Density
Y
data.kde


Chi−Square Q−Q Plot
Chi−Square Quantile
0
5
10
15
0
5
10
15
Squared Mahalanobis Distance


6
8
10
12
14
6
8
10
12
14
0.02
0.04
0.06