PROYECTO FINAL DE ESTADISTICA MATEMATICA

Armando Lara, Dario Quishpe , Jorge Arguello

2024-02-28

Contents

1 Introducción **1**

2 Metodología **1**

2.1 Recolección de los datos 1

2.2 Análisis Exploratorio 2

2.3 Estimación de parámetros mediante el Algoritmo EM 2

2.4 Prueba de distribución de mixturas normales 2

2.5 Simulación de datos mediante los parámetros obtenidos 2

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 Metodología

2.1 Recolección de los datos

Se obtuvo una base de datos de solicitudes de crédito de una Cooperativa importante del país, la cual contiene información sobre variables relevantes, incluyendo el patrimonio neto y los activos de los solicitantes. Por cuestiones de confidencialidad de los clientes de dicha cooperativa, se omitió las variables con respecto a información personal de los solicitantes, tomando únicamente para nuestro interés en este trabajo sólo las variables que necesitamos.

##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 una entidad para cubrir sus obligaciones con sus activos.

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

$$I = Patrimonioneto/Activos$$

2.2 Análisis Exploratorio

Se procederá a realizar diversos gráficos y pruebas estadísticas para identificar la distribución de cada variable, incluyendo el indicador de solvencia. El objetivo principal es obtener una distribución que se ajuste a una mixtura de distribuciones normales para nuestro indicador.

2.3 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.

2.4 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.

2.5 Simulación de datos mediante los parámetros obtenidos

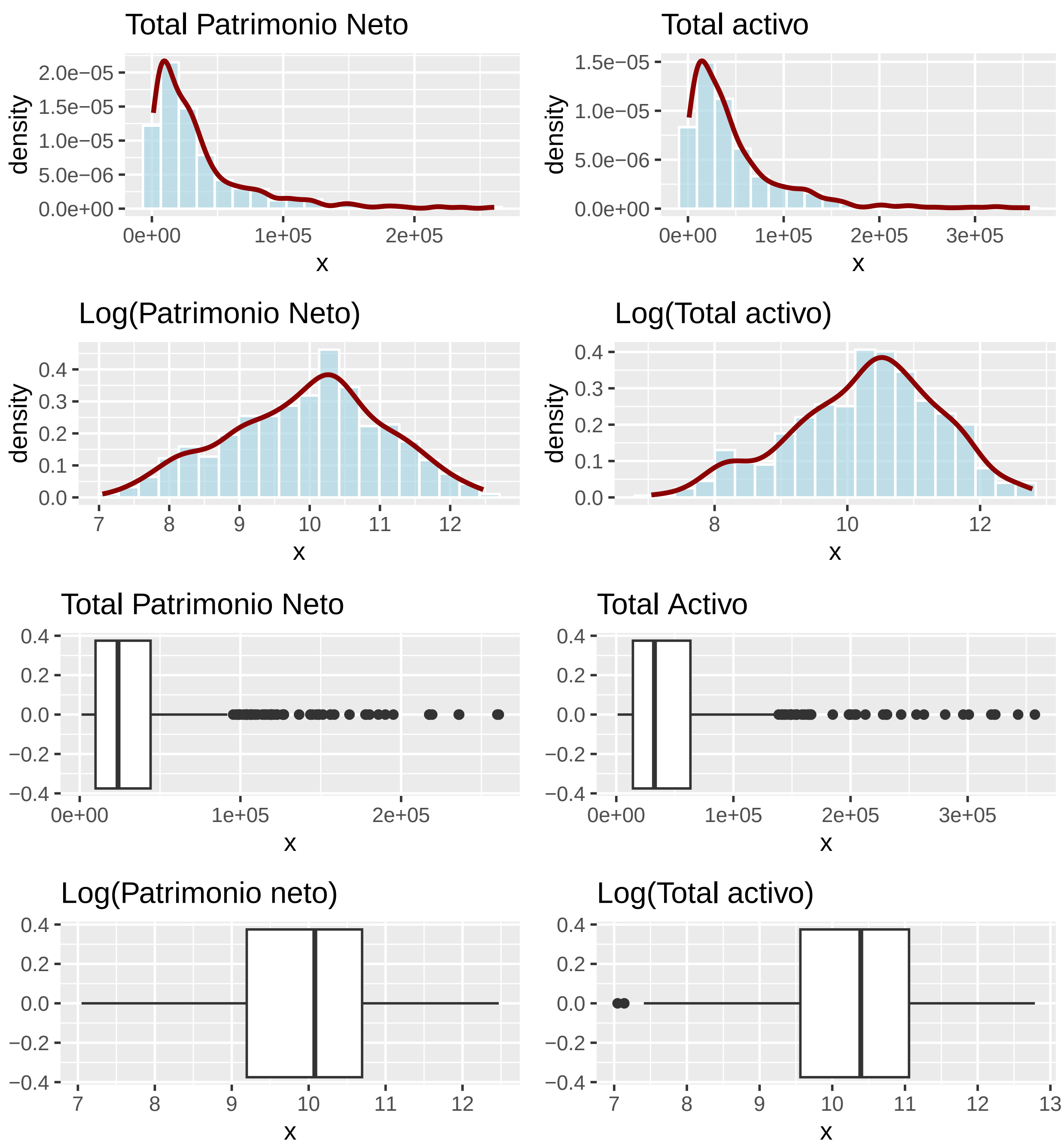
Si mediante el test desarrollado por Priscila se obtiene que hay evidencia estadística para aceptar que la distribución del indicador es una mixtura de normales, se procederá a realizar la simulación de nuevos datos para el indicador de solvencia, en esta simulación se incluirá los parámetros obtenidos 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.

```
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)
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
```

```
#MONTECARLO
media<-vector(length = 5000)
var<-vector(length = 5000)
sd<-vector(length = 5000)
```

```r
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,0.05)
limisupMonte<-quantile(media,1-0.05)
MediaMonte<-mean(media)
Var_Montemean<-mean(var)
SdMonte<-mean(sd)
liminfMonte
```

```
##         5%
## -0.3379069
```

```r
limisupMonte
```

```
##        95%
## -0.3193972
```

```r
MediaMonte
```

```
## [1] -0.3285483
```

```r
Var_Montemean
```

```
## [1] 0.155446
```

```r
SdMonte
```

```
## [1] 0.3942389
```

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

```
##        5%
## 0.7132617
```

```r
LIMSUP
```

```
##       95%
## 0.7265869
```

```r
MEDIA
```

```
## [1] 0.7199681
```

```r
VAR
```

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

```r
SD
```

```
## [1] 1.483255
```

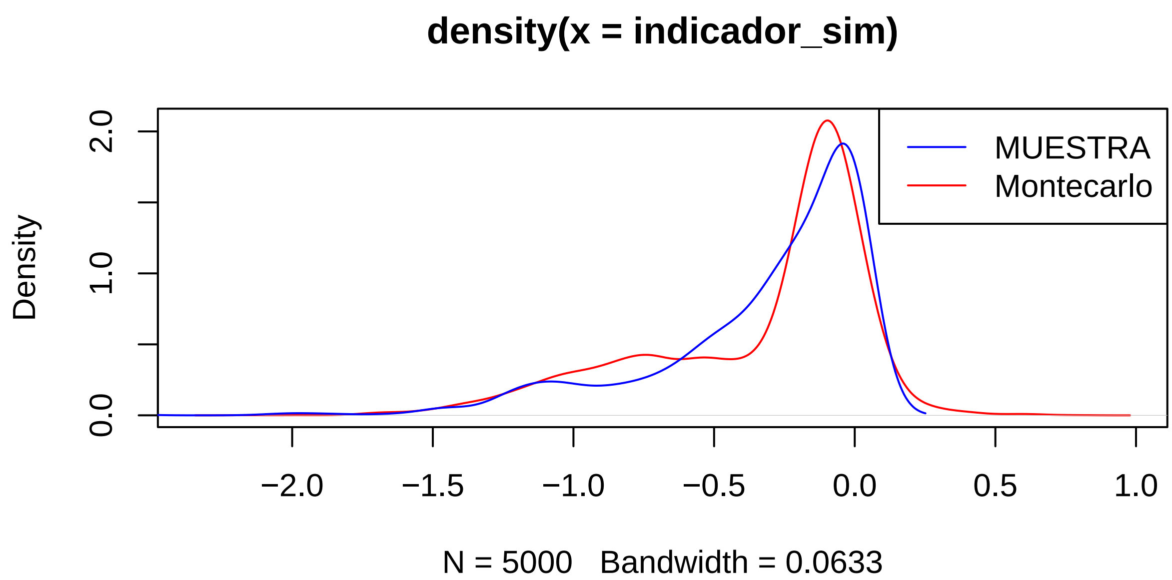


Table 1: **Estimaciones por Montecarlo y Bootstrap (Datos Transformados)**

Método/Medida	Media	Varianza	Sd	LINF	LimSUP
Montecarlo	-0.3285	0.1554	0.3942	-0.33790	-0.31939
Bootstrap	0.225	0.235	0.221	5	4

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	0.225	0.235	0.221	02	1

```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	0.225	0.235	4

```
## [1] -0.04644012
```

```
SesgoVar
```

```
## [1] 1.116612
```

```
SesgoSd
```

```
## [1] 0.1671562
```

```
#CONTRASTE DE HIPOTESIS datos transformados(log)
#H0: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")
```

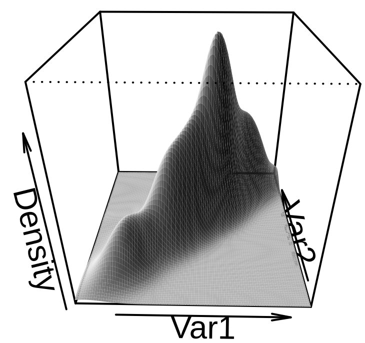
```
##
## Proporción de rechazos al 1%= 0.306
```

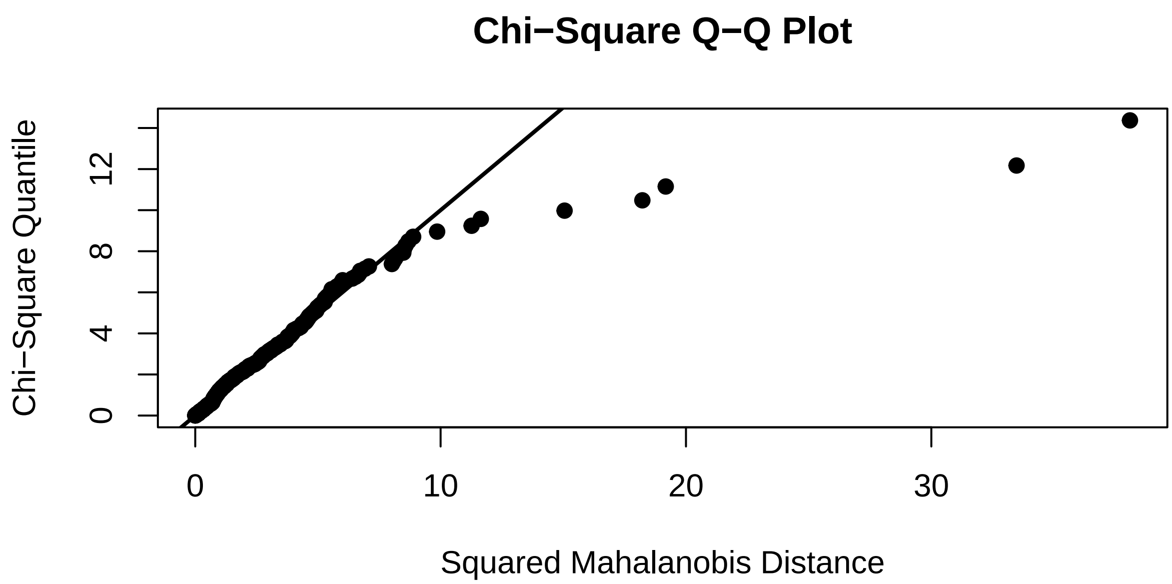
```
cat("\nProporción de rechazos al 5%=",mean(pvalue<0.05),"\n")
```

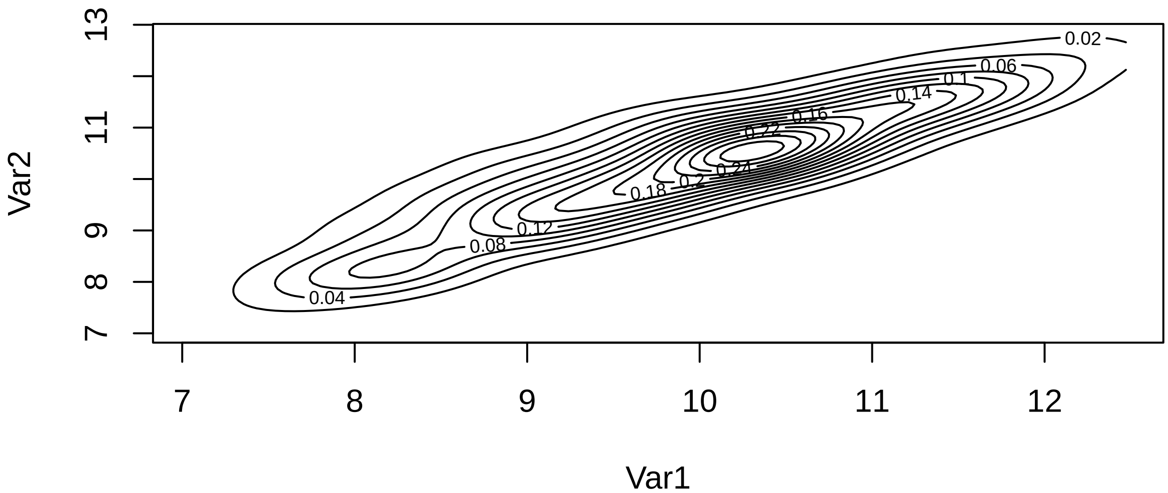
```
##
## Proporción de rechazos al 5%= 0.529
```

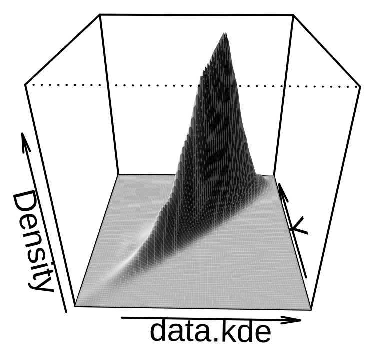
```
cat("\nProporción de rechazos al 10%=",mean(pvalue<0.1),"\n")
```

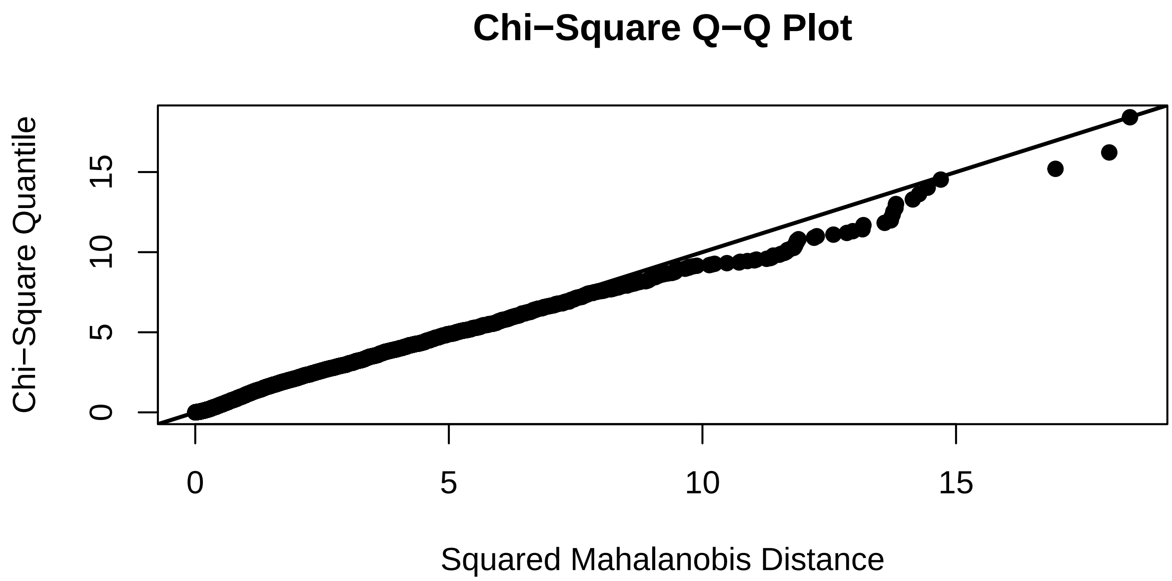
```
##
## Proporción de rechazos al 10%= 0.658
```


Density
Var2
Var1

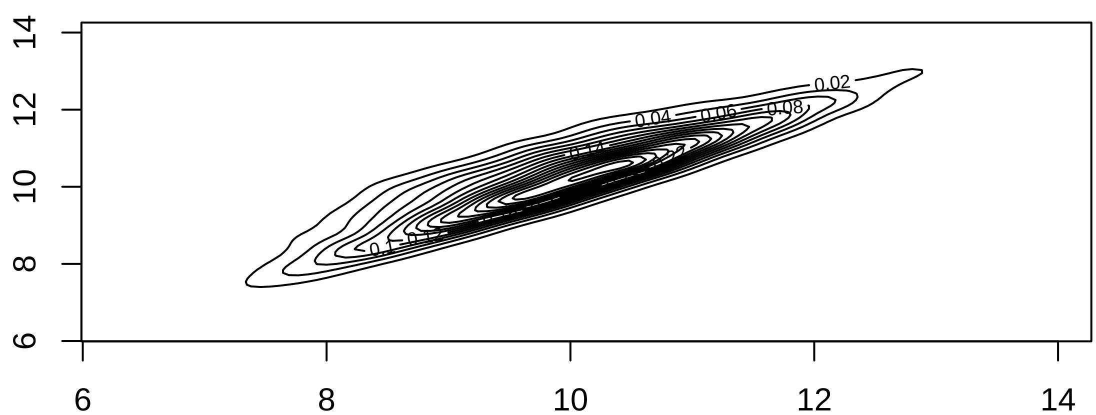

Chi−Square Q−Q Plot
Chi−Square Quantile
0
4
8
12
0
10
20
30
Squared Mahalanobis Distance


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


Density
data.kde
Y



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



0.02
0.04
0.06
0.08
6
8
10
12
14