

# PROYECTO FINAL DE ESTADISTICA MATEMATICA

Armando Lara, Dario Quishpe , Jorge Arguello

2024-03-01

## Índice

<b>1</b>	<b>Introducción</b>	<b>1</b>
<b>2</b>	<b>Objetivos</b>	<b>1</b>
<b>3</b>	<b>Metodología</b>	<b>1</b>
3.1	Recolección de los datos . . . . .	1
3.2	Construcción del Indicador de solvencia . . . . .	2
3.3	Algoritmo Bootstrap . . . . .	2
3.4	Estimación de parámetros mediante el Algoritmo EM . . . . .	3
3.5	Prueba de distribución de mixturas normales . . . . .	3
3.6	Simulación de datos mediante los parámetros obtenidos . . . . .	3
3.7	Comparación con Montecarlo y Bootstrap . . . . .	3
<b>4</b>	<b>Experimentación</b>	<b>4</b>
<b>5</b>	<b>Conclusiones y recomendaciones</b>	<b>8</b>
<b>6</b>	<b>Anexos</b>	<b>8</b>
<b>7</b>	<b>Referencias</b>	<b>17</b>

## 1 Introducción

Para una cooperativa de ahorro y crédito, es fundamental el análisis de la solvencia financiera de sus socios, ya que es un factor crítico al evaluar las distintas solicitudes de créditos, lo que permite tomar decisiones sobre las mismas. Estas decisiones tienen un impacto directo en diversos indicadores de gran relevancia para la institución, que son utilizados en su monitoreo y supervisión. Entre estos indicadores se encuentran la morosidad por productos crediticios, el porcentaje de colocación, el riesgo crediticio, entre otros. Esta medida de solvencia nos proporciona una visión de la capacidad que tienen los socios para cumplir con sus obligaciones financieras. En caso de contar con un indicador de solvencia saludable, sugiere una menor probabilidad de incumplimiento en el pago de deudas. Además, permite a la cooperativa tomar decisiones

de crédito informadas y personalizadas, dando la oportunidad de adaptar los términos del crédito o ofrecer productos convenientes según la solvencia individual.

La cooperativa de ahorro y crédito denominada “X”, en vista de la importancia de la información otorgada por este indicador, ha optado por incorporar técnicas estadísticas en el proceso de toma de decisiones de los analistas respecto a los créditos. El presente proyecto intenta establecer como 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 modelo paramétrico (Montecarlo) y del modelo no paramétrico (Bootstrap).
- Obtener conclusiones que permitan tomar decisiones mediante los valores estimados con las técnicas 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 fabricación de crédito 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 estas, las que son de nuestro principal interés son las siguientes:

- Total Patrimonio Neto: Esta variable ofrece una imagen de la salud financiera de una persona, ya que 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 perteneciente a una persona o compañía que tiene un valor económico.

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

### 3.2 Construcción del Indicador de solvencia

Dentro de una cooperativa, es crucial contar con una representación de la capacidad financiera mediante un indicador. El patrimonio neto refleja la cantidad de flujo disponible para la cooperativa, lo cual le permite hacer frente a situaciones futuras. Por ende, dividir el patrimonio neto entre los activos totales proporciona una medida de la capacidad de un socio para cubrir sus obligaciones con sus activos.

Por consiguiente, el indicador de solvencia se construye de la siguiente manera:

$$I = \text{Patrimonio neto} / \text{Activos}$$

### 3.3 Algoritmo Bootstrap

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

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

Vamos a aproximar el sesgo y la varianza:

$$\begin{aligned} \text{Sesgo}(\hat{\theta}) &= E(\hat{\theta} - \theta) = E(R) \\ \text{Var}(\theta) &= \text{Var}(\hat{\theta} - \theta) \end{aligned}$$

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:

$$\begin{aligned} \text{Sesgo}^*(\hat{\theta}^*) &= \frac{1}{B} \sum_{b=1}^B R^*(b) \\ \text{Var}^*(\hat{\theta}^*) &= \frac{1}{B} \sum_{b=1}^B (R^*(b) - \bar{R}^*)^2 \end{aligned}$$

### 3.4 Estimación de parámetros mediante el Algoritmo EM

Mediante el algoritmo EM (Expectation-Maximization), se procederá a estimar los parámetros del indicador de solvencia de la subpoblación. Este algoritmo permitirá identificar los parámetros óptimos de la distribución, inicialmente desconocida, del indicador de solvencia. Este algoritmo será de gran utilidad, ya que una vez estimados los parámetros, se utilizarán para hacer inferencias e identificar su distribución.

### 3.5 Prueba de distribución de mixturas normales

Una vez obtenidos los parámetros, como la media, la varianza y los pesos, se aplicará un test estadístico apropiado para comprobar si la distribución del indicador es una mezcla de distribuciones normales. Este test fue realizado por Priscila Guayasamín para la clasificación de cooperativas por segmentos [1]. En el caso de que el test rechace la hipótesis de mezcla de distribuciones normales, se procederá a realizar una transformación de nuestros datos para adaptarlos lo mejor posible al desarrollo del proyecto.

### 3.6 Simulación de datos mediante los parámetros obtenidos

Si, mediante el test adaptado en [1], no se encuentra suficiente evidencia estadística para rechazar el hecho de que la distribución del indicador es una mezcla de normales, se procederá a realizar la simulación de nuevos datos obtenidos de dichas simulaciones con el fin de poder construir el indicador de solvencia. En esta simulación se incluirán los parámetros obtenidos por el algoritmo EM y se utilizará una distribución de mezcla de normales. Esto servirá para realizar estimaciones de intervalos de confianza del indicador, así como contrastes de hipótesis.

### 3.7 Comparación con Montecarlo y Bootstrap

Una vez obtenidas las estimaciones de los parámetros mediante el algoritmo EM y la distribución de mezclas 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, utilizamos una mezcla de normales con los parámetros obtenidos del algoritmo EM.

**Muestras:** Realizamos múltiples muestras aleatorias de tamaño  $n$  de estas variables generadas en el paso anterior. Estas muestras se toman con reemplazo, lo que significa que cada observación puede ser seleccionada más de una vez en la muestra.

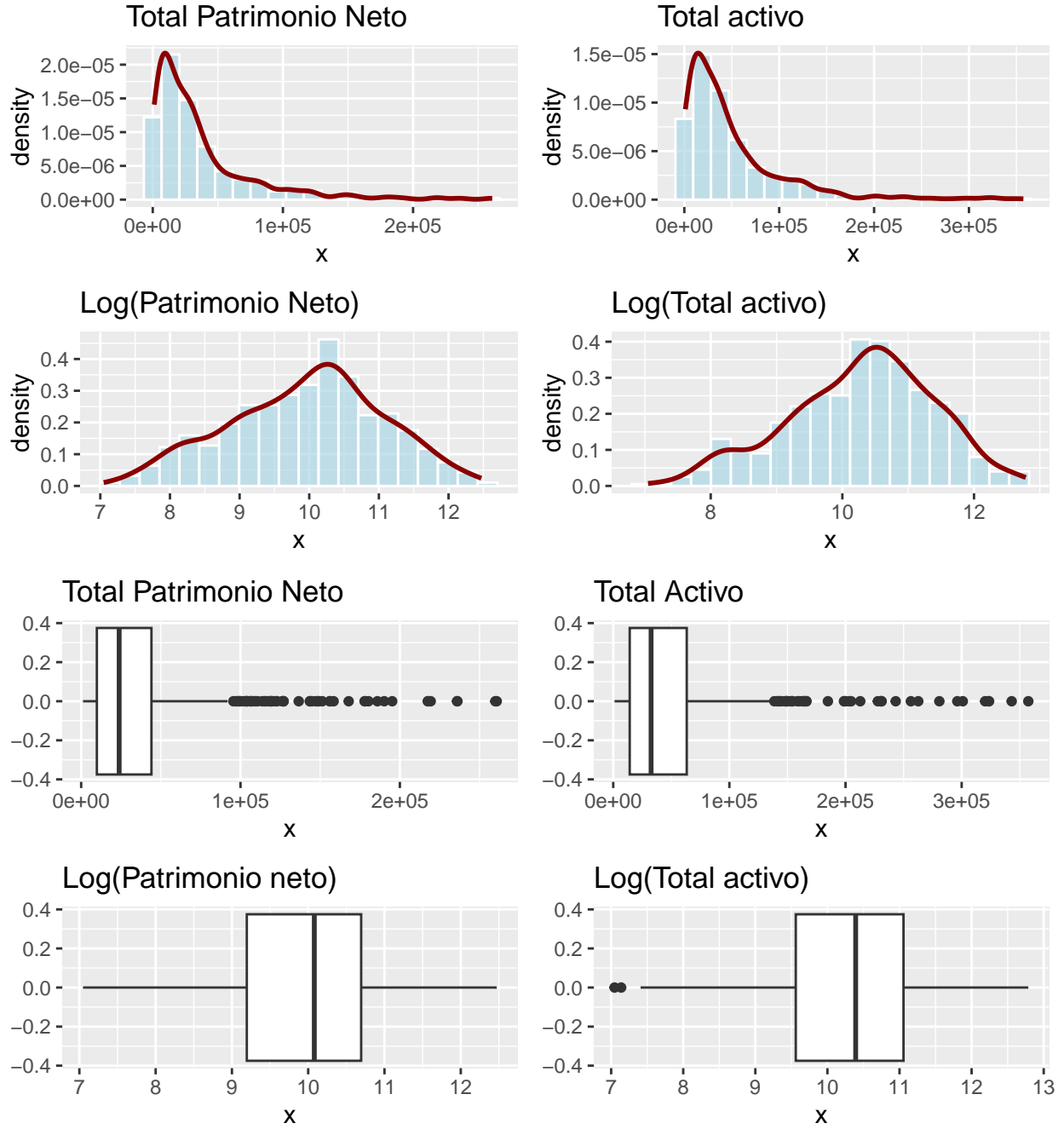
**Construcción del indicador:** Luego, para cada muestra generada de las variables, construimos un indicador y guardamos las medias y varianzas obtenidas en un vector.

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

Finalmente, comparamos los resultados obtenidos con cada metodología y discutimos sus respectivas ventajas y desventajas.

## 4 Experimentación

En primer lugar, se requiere ajustar el indicador de solvencia muestral a una distribución multivariante. Para ello, se realiza el test AD modificado en [1]. Obtenemos un **valor-p** para los datos originales prácticamente de cero ( $5e+7$ ), lo que nos lleva a rechazar la hipótesis nula, es decir, los datos no provienen de una distribución de mezclas normales. En vista de esto, se generan varias transformaciones sobre las variables necesarias para generar el indicador, y se llegó a la conclusión de que la transformación más conveniente y óptima para nuestros datos es el **Logaritmo**, como se ve a continuación:



Realizando esta transformación, obtenemos un **valor-p** de 0.4748, lo cual indica que no hay suficiente evidencia estadística para rechazar la hipótesis nula. Por tanto, se concluye que los datos provienen de una distribución de mezclas normales.

Con este resultado, procedemos a ejecutar el algoritmo EM para encontrar los parámetros de la distribución normal multivariante. Las medias y matrices de varianza y covarianza de la subpoblación son las siguientes:

$$\mu_1 = [9.850069; 10.47885]$$

$$\mu_2 = [10.032998; 10.12234]$$

$$\sum_1 = \begin{bmatrix} 1.025812 & 0.8741010 \\ 0.874101 & 0.8984864 \end{bmatrix}$$

$$\sum_2 = \begin{bmatrix} 1.352622 & 1.3783079 \\ 1.378308 & 1.4139173 \end{bmatrix}$$

y los pesos correspondientes son

$$\lambda_1 = 0.4434142; \lambda_2 = 0.5565858$$

Los resultados de este ajuste los podemos ver a continuación:

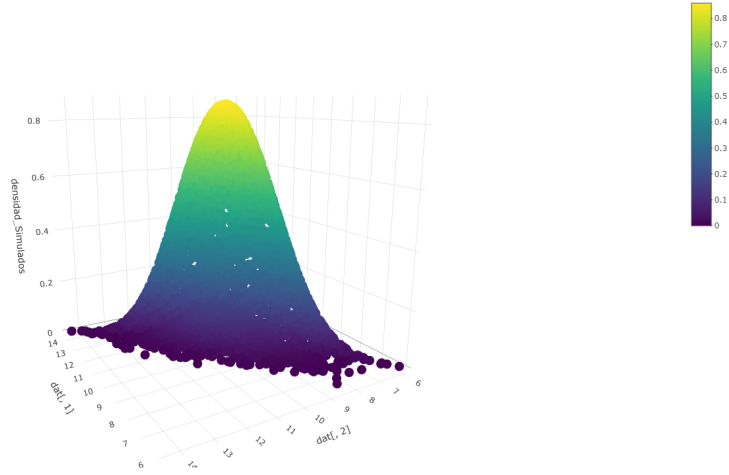


Figura 1: Densidad Simulados

También podemos observar la gráfica correspondiente a los datos originales transformados.

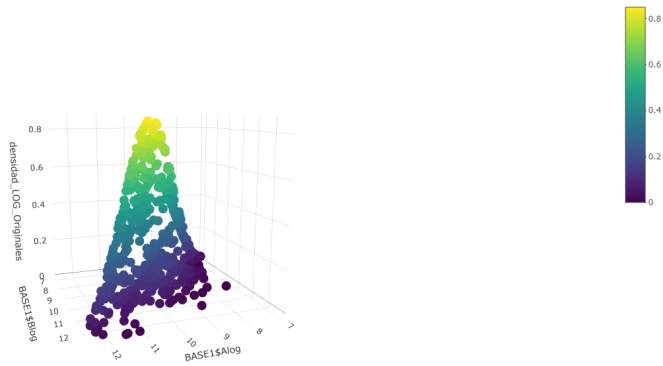
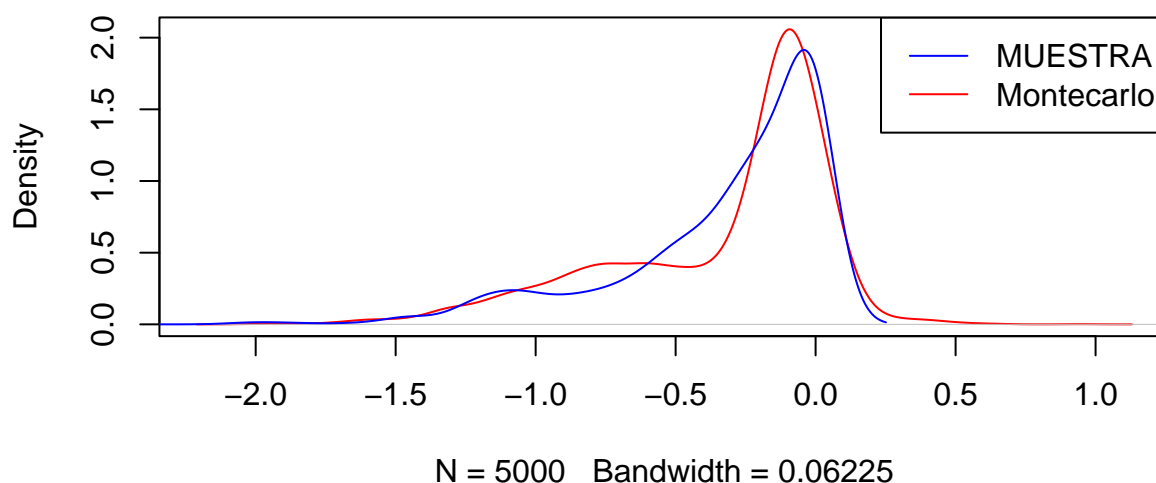


Figura 2: Densidad Originales transformados.

Del procedimiento anterior, encontramos una distribución exacta (mezcla de normales multivariante) que describe la relación entre el patrimonio neto transformado y los activos transformados. Ahora, mediante Montecarlo, simulamos los datos necesarios para la construcción del indicador de solvencia.

## Distribución del indicador de Solvencia



Finalmente presentamos los resultados obtenidos de las simulaciones realizadas

**Cuadro 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

Para comprender los resultados, retransformamos los valores estimados.

**Cuadro 2: Estimaciones por Montecarlo y Bootstrap (Datos Retransformados)**

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

Podemos observar que el método por Montecarlo es más angosto que el bootstrap, pero en cambio, este último presenta sesgos bastante pequeños, como se muestra a continuación.

**Cuadro 3: Estimaciones por Montecarlo y Bootstrap (Datos Retransformados)**

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

Para la prueba de hipótesis, el jefe, según su experiencia, asegura que el índice de solvencia en promedio es igual al 72%. Por lo tanto, nuestra prueba de hipótesis se formula de la siguiente manera:

$$H_0 : \theta = 0.72$$

$$H_a : \theta \neq 0.72$$

Donde  $\theta$  representa la media poblacional de nuestro estimador.

```
##
## Proporción de rechazos al 1%= 0.011

##
## Proporción de rechazos al 5%= 0.042

##
## Proporción de rechazos al 10%= 0.092
```

Esta salida proporciona información sobre la tasa de rechazo bajo diferentes niveles de significancia, lo cual es útil para evaluar el poder estadístico del procedimiento.

## 5 Conclusiones y recomendaciones

- En los resultados, se observó que los intervalos de confianza de los parámetros obtenidos mediante Montecarlo son más cortos que los generados por Bootstrap. Sin embargo, presentan un sesgo mayor, sobre todo en la varianza. Esto implica que, si bien las estimaciones de Montecarlo son más estrechas, pueden estar sujetas a una mayor variabilidad y posiblemente a un mayor grado de error.
- Los intervalos de confianza con Montecarlo podrían brindar una sensación de mayor certeza en las estimaciones de solvencia, lo que podría generar confianza entre los socios y la Cooperativa "X". Sin embargo, es crucial tener en cuenta el sesgo y la variabilidad asociados con estas estimaciones.
- Mediante los resultados obtenidos con Montecarlo, se estima que en promedio, un socio de la cooperativa tiene un indicador de solvencia de 0.72. Entonces, se podría decir que los socios tienen una base financiera sólida en la cooperativa. Esto podría implicar que tienen una capacidad relativamente buena para cumplir con sus obligaciones financieras y participar en actividades de inversión y toma de decisiones dentro de la cooperativa. Sin embargo, considerando que el sesgo en la variabilidad es más alto en comparación con Bootstrap, las estimaciones tienden a sobrevalorar la solvencia financiera, lo que podría llevar a decisiones financieras erróneas si se basan únicamente en estas estimaciones sesgadas.
- En este proyecto se han omitido factores externos de la cooperativa, como la infraestructura, la edad de los socios, la condición financiera actual de la empresa, etc. Por lo tanto, sería recomendable considerar estos factores para obtener resultados más realistas sobre las inferencias que se puedan obtener del indicador.
- El aspecto de la potencia computacional también es un factor importante en la realización del algoritmo EM, pues ejecutarlo llevó aproximadamente 5 minutos con una computadora de gama media y solo se contaba con 660 datos, lo cual es bastante aceptable, pero puede ser un problema con bases de datos con muchos más registros. En ese caso, se recomienda utilizar un ordenador con una mejor capacidad de procesamiento.

## 6 Anexos



```

#TEST AD MODIFICADO
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)
}

```

```

#GRAFICAS DE CAJAS Y DENSIDADES
G1<-ggplot(data.frame(x = BASE1$TOTAL_PATRIMONIO_NETO), aes(x)) +
  geom_histogram(aes(y = ..density..), bins = 20, color = "white",
    fill = "lightblue", alpha = 0.7) +geom_density(color = "darkred", size = 1) +
  labs(title = "Total Patrimonio Neto")

G2<-ggplot(data.frame(x = BASE1$TOTAL_ACTIVIVO), aes(x)) +
  geom_histogram(aes(y = ..density..), bins = 20, color = "white",
    fill = "lightblue", alpha = 0.7) +geom_density(color = "darkred", size = 1) +

```

```

labs(title = "Total activo")

G3<-ggplot(data.frame(x = BASE1$Alog), aes(x)) +
  geom_histogram(aes(y = ..density..), bins = 20, color = "white",
    fill = "lightblue", alpha = 0.7) +geom_density(color = "darkred", size = 1) +
  labs(title = "Log(Patrimonio Neto)")

G4<-ggplot(data.frame(x = BASE1$Blog), aes(x)) +
  geom_histogram(aes(y = ..density..), bins = 20, color = "white",
    fill = "lightblue", alpha = 0.7) +
  geom_density(color = "darkred", size = 1) + labs(title = "Log(Total activo)")

grid.arrange(G1,G2,G3,G4)
G5<-ggplot(data.frame(x = BASE1$TOTAL_PATRIMONIO_NETO), aes(x)) +
  geom_boxplot() + labs(title = "Total Patrimonio Neto")
G6<-ggplot(data.frame(x = BASE1$TOTAL_ACTIVIVO), aes(x)) +
  geom_boxplot() + labs(title = "Total Activo")
G7<-ggplot(data.frame(x = BASE1$Alog), aes(x)) +
  geom_boxplot() + labs(title = "Log(Patrimonio neto)")
G8<-ggplot(data.frame(x = BASE1$Blog), aes(x)) +
  geom_boxplot() + labs(title = "Log(Total activo)")
grid.arrange(G5,G6,G7,G8)

```

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

```

#RESULTADOS MONTECARLO
#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)
}

#Estimaciones con variables Transformadas
liminfMonte<-quantile(media,0.025)
limisupMonte<-quantile(media,1-0.025)
MediaMonte<-mean(media)
Var_Montemean<-mean(var)
SdMonte<-mean(sd)
liminfMonte
limisupMonte
MediaMonte
Var_Montemean
SdMonte

#Estimaciones con variables retransformadas
LIINF<-exp(liminfMonte)
LIMSUP<-exp(limisupMonte)
MEDIA<-exp(MediaMonte)
VAR<-exp(Var_Montemean)

```

```
SD<-exp(SdMonte)
LIINF
LIMSUP
MEDIA
VAR
SD
```

```
#CALCULO DEL SESGO DE VARIABLES RETRANSFORMADAS
```

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

```
SesgoMedia
```

```
SesgoVar
```

```
SesgoSd
```

```
#CONTRASTE DE HIPOTESIS datos transformados(log)
```

```
#H0:mu=-0.3285041
```

```
#Ha:mu=-0.3285041
```

```
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)+0.3285041)/(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")
```

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

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

```
#BOOTSRAPO NO PARAMETRICO
```

```
library(readxl)
```

```
library(tidyverse)
```

```
library(dplyr)
```

```
library(boot)
```

```
#Cargar la base
```

```
load('BASE_SOLICITUD_DE_CREDITO.RData')
```

```
#Crear indicador
```

```
Datos1<-BASE1 %>% mutate(Ind_efectivo=TOTAL_PATRIMONIO_NETO/TOTAL_ACTIVOS) |>
```

```
  select(TOTAL_ACTIVOS,
         TOTAL_PATRIMONIO_NETO,Ind_efectivo)
```

```

#Función bootstrap no paramétrica

bootstrap <- function(N_muestras,datos_exp,alpha,valor){
  n <- length(datos_exp)
  media <- numeric(n)
  varianza <- numeric(n)
  desviacion <- numeric(n)
  valor_p <- numeric(N_muestras)
  for (i in 1:N_muestras){
    #remuestras
    muestra <- sample(datos_exp,replace = TRUE)
    #estadísticos
    media[i] <- mean(muestra)
    #media
    varianza[i] <- var(muestra)
    # desviación
    desviacion[i] <- sd(muestra)
    #media
    x_barra_boot <- mean(muestra)
    # desviación
    cuasi_dt_boot <- sd(muestra)
    #Contraste de hipotesis Media
    #Ho: mu=valor
    t <- (x_barra_boot-valor)/(cuasi_dt_boot/sqrt(n))
    p.valor <- 1-pt(abs(t),n-1)+pt(-abs(t),n-1)
    valor_p[i] <- p.valor
  }
  #Intervalos al 95% de confianza media
  ic_inf_media <- quantile(media, prob = 0.025)
  ic_sup_media <- quantile(media, prob = 0.975)
  intervalo_confianza_media <- c(ic_inf_media, ic_sup_media)
  #Intervalos al 95% de confianza varianza
  ic_inf_varianza <- quantile(varianza, prob = 0.025)
  ic_sup_varianza <- quantile(varianza, prob = 0.975)
  intervalo_confianza_varianza <- c(ic_inf_varianza, ic_sup_varianza)
  #Intervalos al 95% de confianza desviacion
  ic_inf_desviacion <- quantile(desviacion, prob = 0.025)
  ic_sup_desviacion <- quantile(desviacion, prob = 0.975)
  intervalo_confianza_desviacion <- c(ic_inf_desviacion, ic_sup_desviacion)

  #Sesgo del estadístico
  sesgo_media <- mean(datos_exp)-sum(media)/N_muestras
  #Sesgo del estadístico
  sesgo_varianza <- var(datos_exp)-sum(varianza)/N_muestras
  #Sesgo del estadístico
  sesgo_desviacion <- sd(datos_exp)-sum(desviacion)/N_muestras

  return(list(Muestras_media = media,Muestras_varianza = varianza,
             Muestras_desviacion = desviacion,
             media_estimada =sum(media)/N_muestras,
             varianza_estimada =sum(varianza)/N_muestras,
             desv_estimada =sum(desviacion)/N_muestras,
             IC_media = intervalo_confianza_media,

```

```

        IC_varianza = intervalo_confianza_varianza,
        IC_desv = intervalo_confianza_desviacion,
        Sesgo_mean=sesgo_media,Sesgo_var=sesgo_varianza,
        Sesgo_desv=sesgo_desviacion,prop_rechazo_media=mean(valor_p<0.05)))
}

##### CON LOS DATOS ORIGINALES #####

##### PARA LA MEDIA #####
remuestreo <- bootstrap(50000,Datos1$Ind_efectivo,0.05,0.75)
#Media estimada
remuestreo$media_estimada
#Intervalo
remuestreo$IC_media
#sesgo
remuestreo$Sesgo_mean
#contraste
remuestreo$prop_rechazo_media

##### PARA LA VARIANZA #####
#varianza estimada
remuestreo$varianza_estimada
#Intervalo
remuestreo$IC_varianza
#sesgo
remuestreo$Sesgo_var

##### PARA LA DESVIACIÓN #####
#Desviacion estimada
remuestreo$desv_estimada
#Intervalo
remuestreo$IC_desv
#sesgo
remuestreo$Sesgo_desv

##### CON LOS DATOS TRANSFORMADOS #####
Datos1$Ind_efectivo <- log(Datos1$Ind_efectivo)
remuestreo <- bootstrap(50000,Datos1$Ind_efectivo,0.05,0.75)

##### PARA LA MEDIA #####
#Media estimada
remuestreo$media_estimada
#Intervalo
remuestreo$IC_media
#sesgo
remuestreo$Sesgo_mean
#contraste
remuestreo$prop_rechazo_media

##### PARA LA VARIANZA #####
#Varianza estimada
remuestreo$varianza_estimada

```

```

#Intervalo
remuestreo$IC_varianza
#sesgo
remuestreo$Sesgo_var

##### PARA LA DESVIACIÓN #####
#Desviacion estimada
remuestreo$desv_estimada
#Intervalo
remuestreo$IC_desv
#sesgo
remuestreo$Sesgo_desv

#Densidad de simulados

x <- Datos1$Ind_efectivo
h <- bw.SJ(x)
dens <- density(x)
npden <- density(x, bw = h)
# npden <- density(x, bw = "SJ")
# h <- npden$bw

# plot(npden)
#hist(x, freq = FALSE, breaks = "FD", main = "Kernel density estimation",
#      xlab = paste("Bandwidth =", formatC(h)), border = "darkgray",
#      xlim = c(-3,0.5))
#lines(dens, lwd = 2)

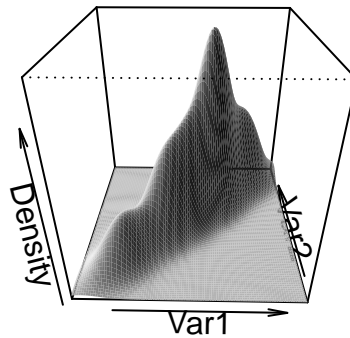
```

## TEST Y GRAFICAS EXTRA PARA LOS VALORES ORIGINALES TRANSFORMADOS

```

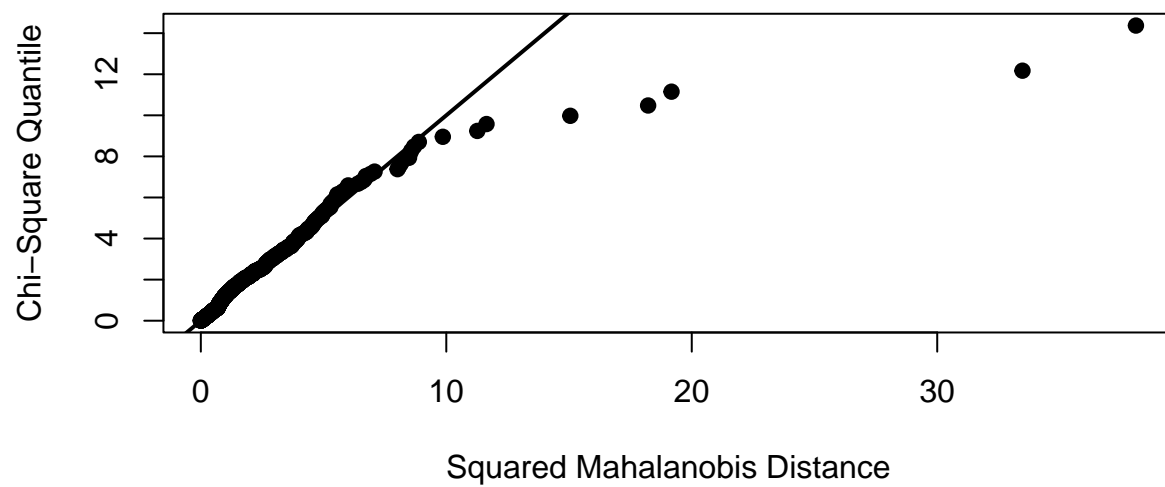
#Test extra -mixture de normales Valores originales con logaritmo
N<-mvn(data, mvnTest = "mardia",multivariatePlot = "persp",tol =0.005)

```

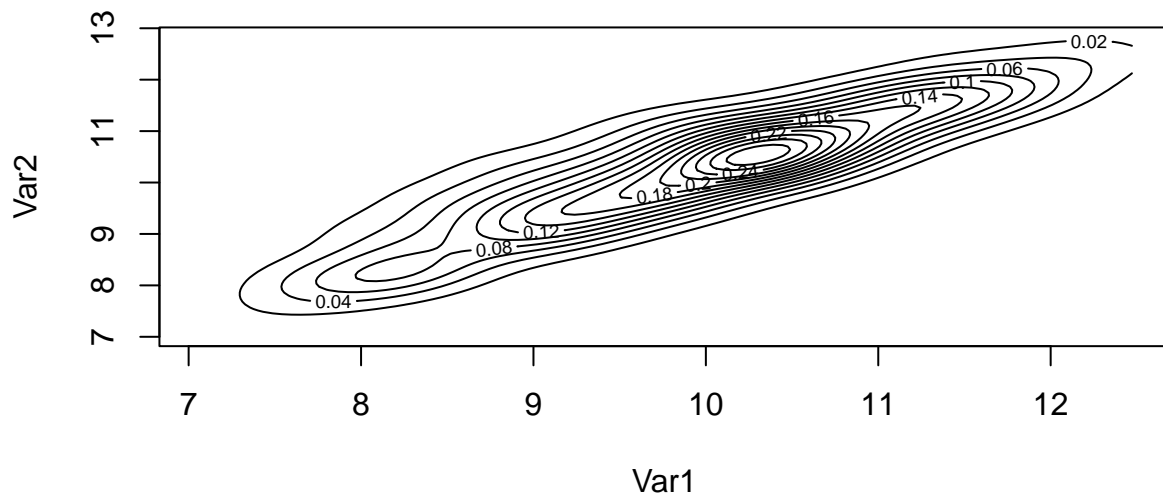


```
M<-mvn(data, mvnTest = "mardia",multivariatePlot = "qq",tol =0.005 )
```

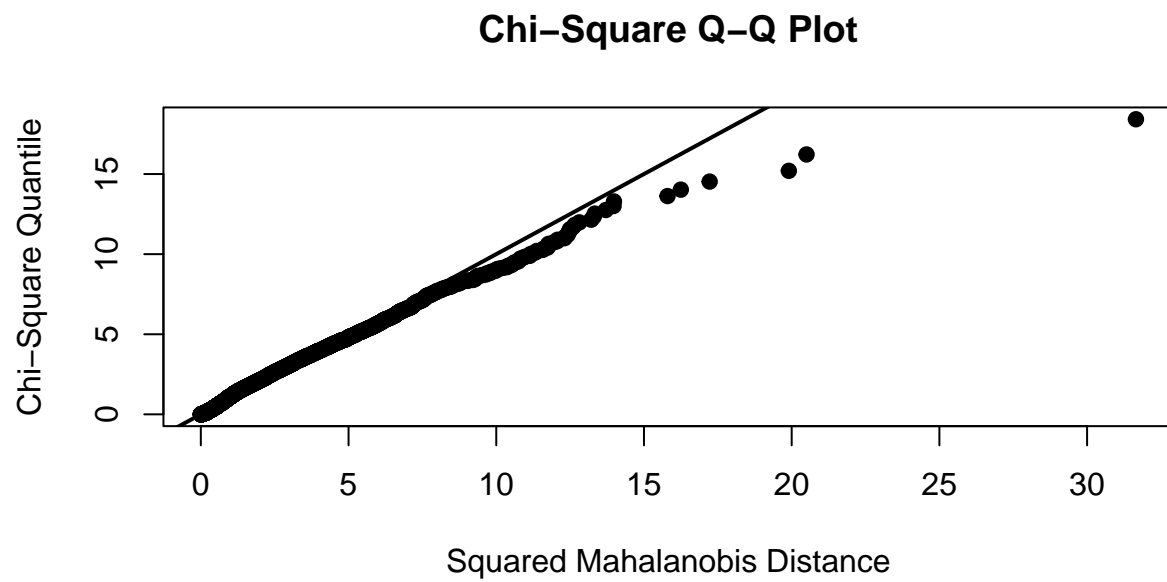
**Chi-Square Q-Q Plot**



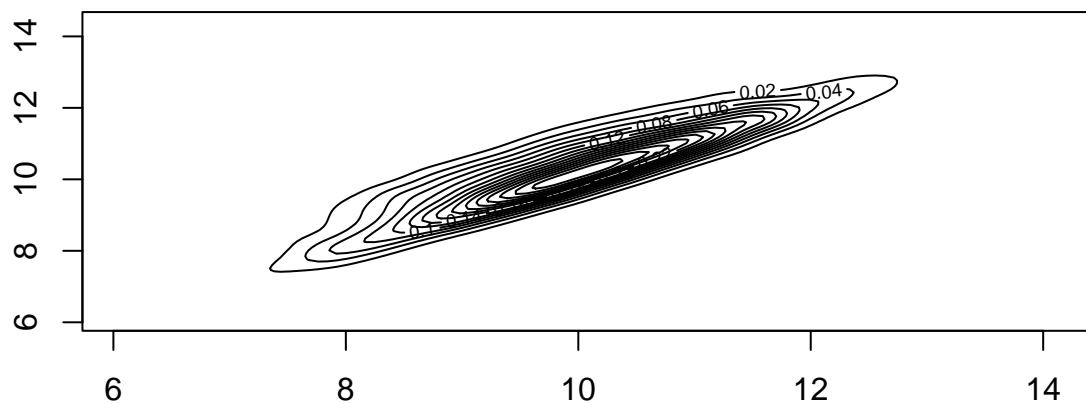
```
O<-mvn(data, mvnTest = "mardia",multivariatePlot = "contour",tol =0.005 )
```







```
Q<-mvn(data, mvnTest = "mardia", showOutliers = TRUE,multivariatePlot = "contour",tol =0.005 )
```



## 7 Referencias

- [1] Guayasamín, P. (17 de septiembre de 2023). Simulación Montecarlo y Bootstrap: Cooperativas deterioradas en el indicador de liquidez diciembre 2022 - enero 2023.
- [2] Fernández-Casal, R., Cao, R., & Costa, J. (2023). Técnicas de Simulación y Remuestreo (github). Recuperado de [<https://rubenfcasal.github.io/simbook/>]

[3] Cao, R., & Fernández-Casal, R. (2021). Técnicas de Remuestreo (github). Recuperado de [[https://rubenfcasal.github.io/book\\_remuestreo/](https://rubenfcasal.github.io/book_remuestreo/)]