

Actividad Integradora

Andrés Villarreal González

2024-08-20

Actividad Integradora

```
M = read.csv('food_data_g.csv')
```

Seleccionamos nuestra variable asignada

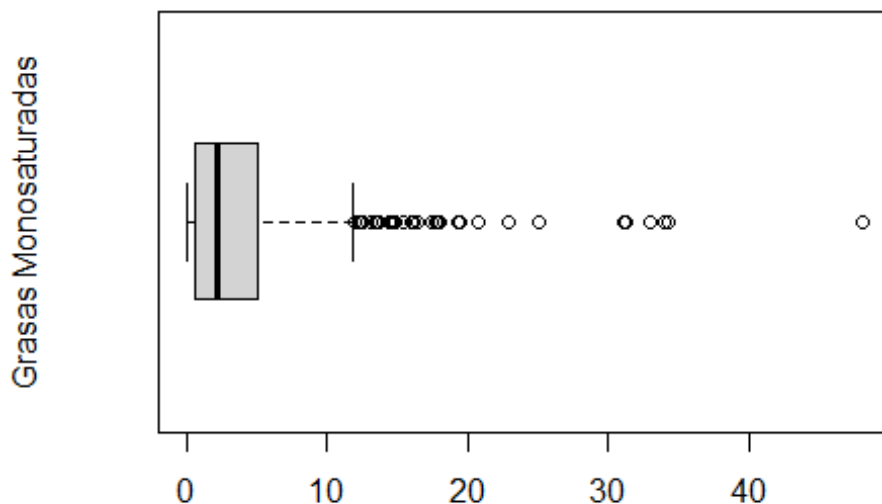
```
mof <- M$Monounsaturated.Fats
```

Analizar datos atípicos

Graficar el diagrama de caja y bigote

```
boxplot(mof, horizontal=TRUE, ylim=c(min(mof), max(mof)),  
        main = "Boxplot de Grasas Monosaturadas", ylab = "Grasas  
Monosaturadas")
```

Boxplot de Grasas Monosaturadas



Calcula las principales medidas que te ayuden a identificar datos atípicos

Cálculo del primer cuartil (q1) y tercer cuartil (q3) para Grasas Monosaturadas

```
q1_mof <- quantile(mof, 0.25)
```

```

q3_mof <- quantile(mof, 0.75)
iqr_mof <- q3_mof - q1_mof # Rango intercuartílico para Grasas
Monosaturadas
mean_mof <- mean(mof) # Media
sd_mof <- sd(mof) # Desviación estándar

cat("Primer cuartil:", q1_mof, "\n")

## Primer cuartil: 0.6

cat("Tercer cuartil:", q3_mof, "\n")

## Tercer cuartil: 5.15

cat("Rango intercuartílico:", iqr_mof, "\n")

## Rango intercuartílico: 4.55

cat("Media:", mean_mof, "\n")

## Media: 4.001564

cat("Desviación estándar:", sd_mof, "\n")

## Desviación estándar: 5.540608

```

Identifica la cota de 1.5 rangos intercuartílicos para datos atípicos

```

# Cotas de 1.5 IQR
limite_inferior <- q1_mof - 1.5 * iqr_mof
limite_superior <- q3_mof + 1.5 * iqr_mof

cat("Limite inferior:", limite_inferior, "\n")

## Limite inferior: -6.225

cat("Limite superior:", limite_superior, "\n")

## Limite superior: 11.975

# Identificar los datos atípicos
datos_atipicos <- mof[mof < limite_inferior | mof > limite_superior]

# Contar los datos atípicos
cantidad_atipicos <- length(datos_atipicos)

cat("Cantidad de datos atípicos:", cantidad_atipicos, "\n")

## Cantidad de datos atípicos: 40

```

Identifica la cota de 3 desviaciones estándar alrededor de la media

```

# Calcular los límites superior e inferior
limite_inferior <- mean_mof - 3 * sd_mof
limite_superior <- mean_mof + 3 * sd_mof

```

```

cat("Limite inferior:", limite_inferior, "\n")
## Limite inferior: -12.62026

cat("Limite superior:", limite_superior, "\n")
## Limite superior: 20.62339

# Identificar los datos atípicos
datos_atipicos <- mof[mof < limite_inferior | mof > limite_superior]

# Contar Los datos atípicos
cantidad_atipicos <- length(datos_atipicos)

cat("Cantidad de datos atípicos:", cantidad_atipicos, "\n")
## Cantidad de datos atípicos: 11

```

Identifica la cota de 3 rangos intercuartílicos para datos extremos

```

# Cotas de 1.5 IQR
limite_inferior <- q1_mof - 3 * iqr_mof
limite_superior <- q3_mof + 3 * iqr_mof

cat("Limite inferior:", limite_inferior, "\n")
## Limite inferior: -13.05

cat("Limite superior:", limite_superior, "\n")
## Limite superior: 18.8

# Identificar los datos atípicos
datos_atipicos <- mof[mof < limite_inferior | mof > limite_superior]

# Contar Los datos atípicos
cantidad_atipicos <- length(datos_atipicos)

cat("Cantidad de datos atípicos extremos:", cantidad_atipicos, "\n")
## Cantidad de datos atípicos extremos: 13

```

Analizar normalidad

Realiza pruebas de normalidad univariada para la variable

H_0 : Los datos siguen una distribución normal H_1 : Los datos no siguen una distribución normal

```

library(nortest)
library(tseries)

```

```
## Warning: package 'tseries' was built under R version 4.3.3

## Registered S3 method overwritten by 'quantmod':
##   method      from
##   as.zoo.data.frame zoo

suppressWarnings(library(tseries))
suppressWarnings(library(nortest))

# Anderson Darling
ad_test<-ad.test(mof)

# Jarque Bera
jb_test_result <- jarque.bera.test(mof)

cat("Valor-p Anderson Darling:", ad_test$p.value, "\n")

## Valor-p Anderson Darling: 3.7e-24

cat("Valor-p Jarque Bera:", jb_test_result$p.value, "\n")

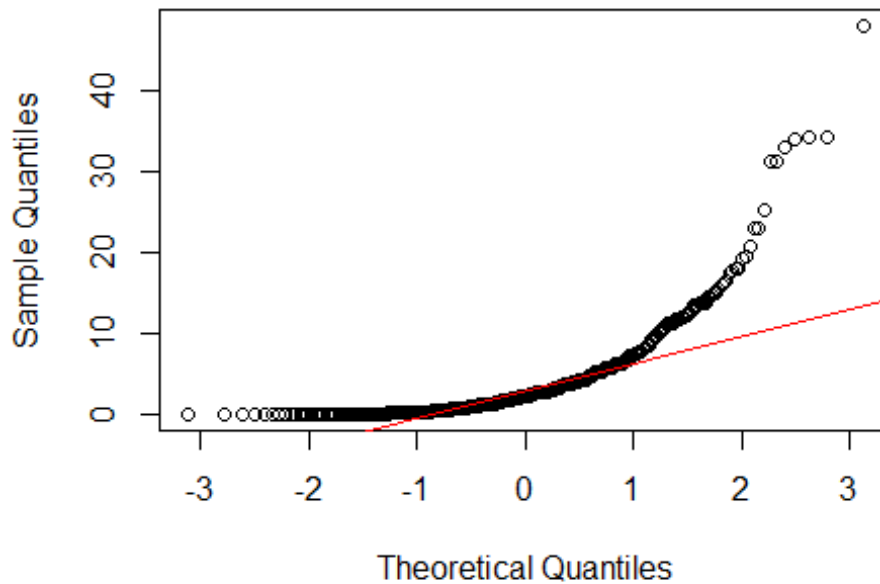
## Valor-p Jarque Bera: 0
```

Debido a que ambos valores p son menores a 0.05, podemos llegar a la conclusión de que los datos no siguen una distribución normal.

Grafica los datos y su respectivo QQPlot

```
# QQPlot para Grasas Monosaturadas
qqnorm(mof, main="QQPlot de Grasas Monosaturadas")
qqline(mof, col="red")
```

QQPlot de Grasas Monosaturadas



Viendo la gráfica podemos observar claramente que los datos no siguen una distribución normal

Calcula el coeficiente de sesgo y el coeficiente de curtosis

```
library(moments)
```

```
# Calcular sesgo y curtosis para Calorías
```

```
sesgo_mof <- skewness(mof)
```

```
curtosis_mof <- kurtosis(mof)
```

```
# Mostrar los resultados
```

```
cat("Sesgo de Grasas Monosaturadas:", sesgo_mof, "\n")
```

```
## Sesgo de Grasas Monosaturadas: 3.207982
```

```
cat("Curtosis de Grasas Monosaturadas", curtosis_mof, "\n")
```

```
## Curtosis de Grasas Monosaturadas 17.66712
```

Compara las medidas de media, mediana y rango medio de cada variable

```
mediana_mof <- median(mof)
```

```
rango_medio_mof <- (min(mof) + max(mof)) / 2
```

```
cat("Media:", mean_mof, "\n")
```

```
## Media: 4.001564
```

```
cat("Mediana:", mediana_mof, "\n")
```

```
## Mediana: 2.2
```

```
cat("Rango Medio:", rango_medio_mof, "\n")
```

```
## Rango Medio: 24
```

Realiza el gráfico de densidad empírica y teórica suponiendo normalidad en la variable

Supongamos que mof ya está definido

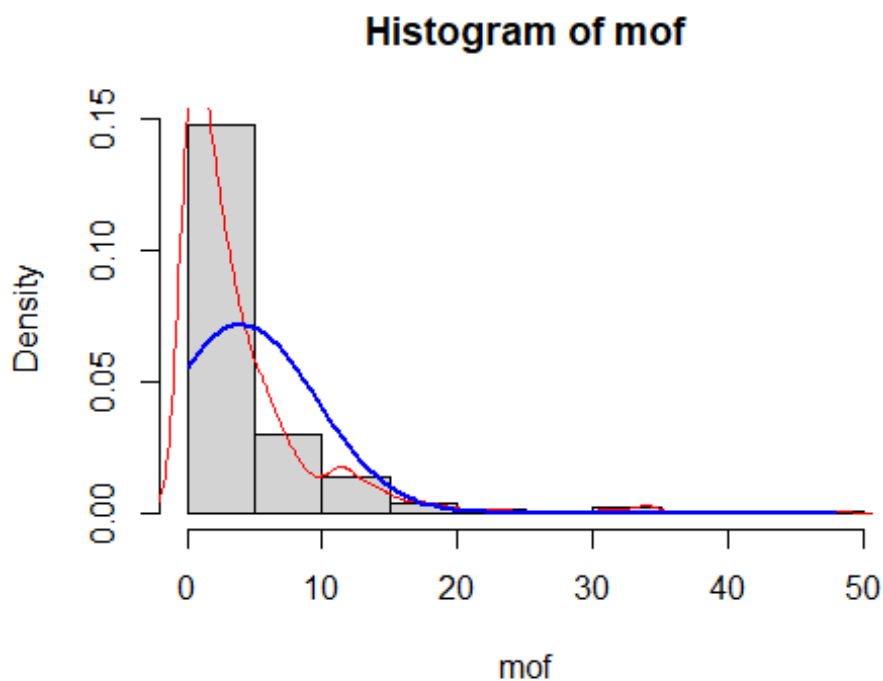
```
hist(mof, freq = FALSE)
```

Densidad estimada de Los datos

```
lines(density(mof), col = "red")
```

Agregar La curva de distribución normal con La media y desviación estándar de Los datos

```
curve(dnorm(x, mean = mean(mof), sd = sd(mof)),  
      from = min(mof), to = max(mof),  
      add = TRUE, col = "blue", lwd = 2)
```



Despues de observar las pruebas de normalidad, ambas gráficas, curtosis y sesgo se puede observar claramente que los datos no siguen una distribución normal.

Transformación a Normalidad

Yeo-Johnson ya que se tienen valores negativos

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.3

## Loading required package: carData

# Encontrar Lambda usando powerTransform
result_yeojohnson <- powerTransform(mof ~ 1, family = "yjPower")

# Obtener La Lambda
lambda <- result_yeojohnson$lambda

# Aplicar la transformación Yeo-Johnson
transformed_data <- ifelse(mof >= 0,
                          ((mof + 1)^lambda - 1) / lambda,
                          -((( -mof + 1)^(2 - lambda) - 1) / (2 -
lambda))))
```

Escribe las ecuaciones de los modelos de transformación encontrados

Para valores positivos ($y \geq 0$):

$$y_{transformed} = \frac{(y + 1)^{-0.251} - 1}{-0.251}$$

Para valores negativos ($y < 0$):

$$y_{transformed} = -\frac{(-y + 1)^{2.251} - 1}{2.251}$$

Analiza normalidad de la transformación obtenida

Medidas descriptivas para datos originales

```
original_stats <- data.frame(
  Minimo = min(mof, na.rm = TRUE),
  Maximo = max(mof, na.rm = TRUE),
  Media = mean(mof, na.rm = TRUE),
  Mediana = median(mof, na.rm = TRUE),
  Cuartil_1 = quantile(mof, 0.25, na.rm = TRUE),
  Cuartil_3 = quantile(mof, 0.75, na.rm = TRUE),
  Sesgo = skewness(mof, na.rm = TRUE),
  Curtosis = kurtosis(mof, na.rm = TRUE)
)
```

Medidas descriptivas para datos transformados (Modelo Exacto)

```
exact_transformed_stats <- data.frame(
  Minimo = min(transformed_data, na.rm = TRUE),
  Maximo = max(transformed_data, na.rm = TRUE),
  Media = mean(transformed_data, na.rm = TRUE),
  Mediana = median(transformed_data, na.rm = TRUE),
  Cuartil_1 = quantile(transformed_data, 0.25, na.rm = TRUE),
  Cuartil_3 = quantile(transformed_data, 0.75, na.rm = TRUE),
  Sesgo = skewness(transformed_data, na.rm = TRUE),
  Curtosis = kurtosis(transformed_data, na.rm = TRUE)
```

)

```
print(original_stats)
```

```
##      Minimo Maximo      Media Mediana Cuartil_1 Cuartil_3      Sesgo  
Curtosis  
## 25%      0      48 4.001564      2.2      0.6      5.15 3.207982  
17.66712
```

```
print(exact_transformed_stats)
```

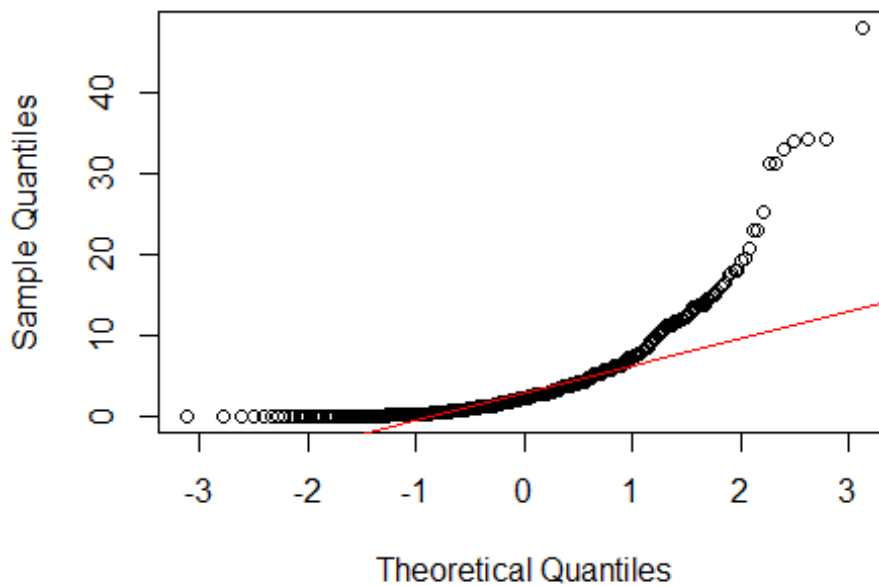
```
##      Minimo      Maximo      Media      Mediana Cuartil_1 Cuartil_3      Sesgo  
Curtosis  
## 25%      0 2.483194 0.9746611 1.008622 0.4433168 1.458438 0.06929913  
2.015156
```

Gráficas de densidad y QQPlot con datos originales y transformados

Datos originales

```
qqnorm(mof, main="QQPlot de Grasas Monosaturadas (datos originales)")  
qqline(mof, col="red")
```

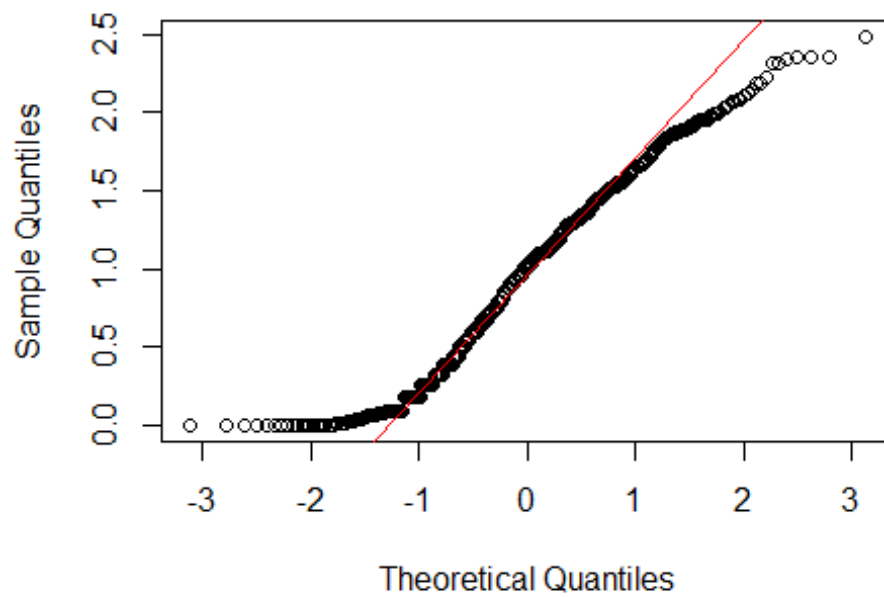
QQPlot de Grasas Monosaturadas (datos originale



Datos transformados

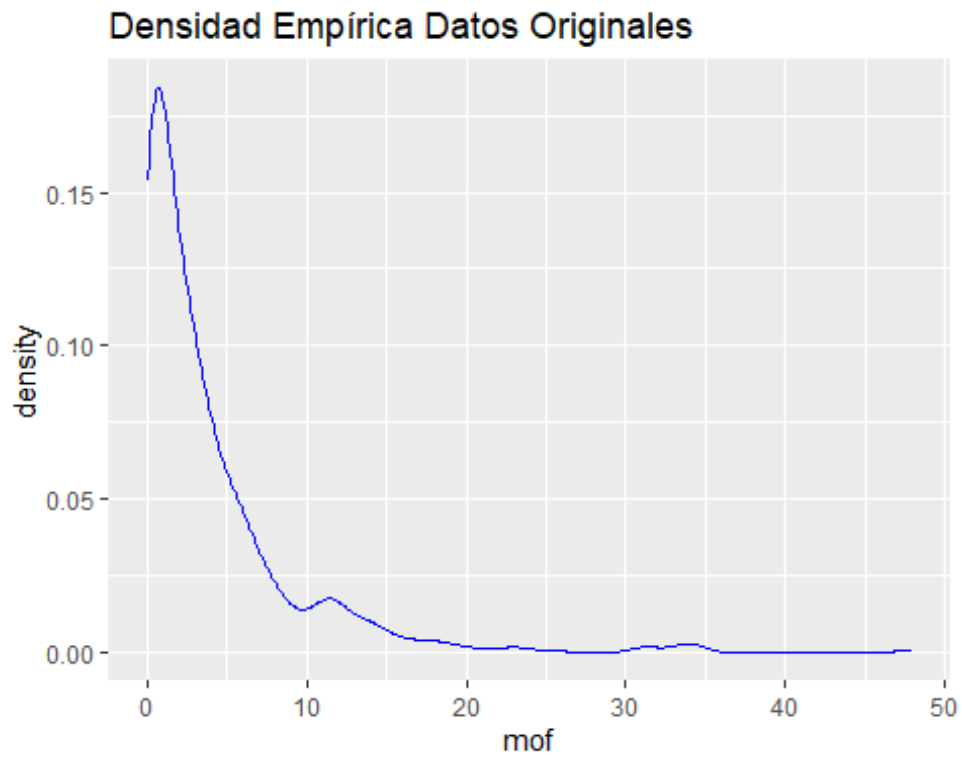
```
qqnorm(transformed_data, main="QQPlot de Grasas Monosaturadas (datos  
transformados)")  
qqline(transformed_data, col="red")
```


Q-QPlot de Grasas Monosaturadas (datos transforma



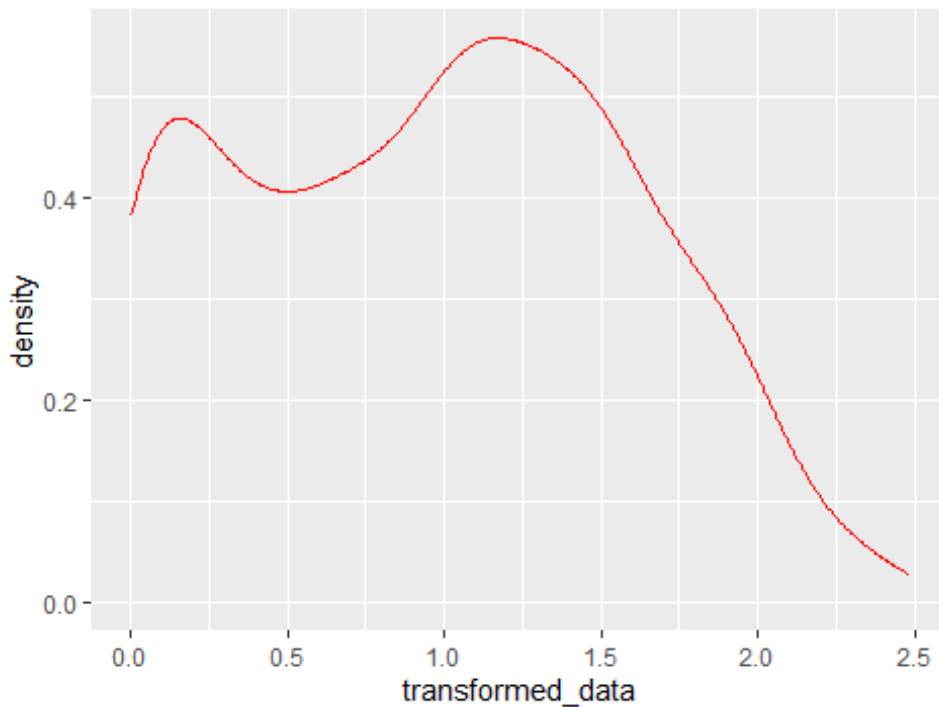
```
library(ggplot2)

# Datos originales
ggplot() +
  geom_density(aes(x = mof), color = "blue") +
  ggtitle("Densidad Empírica Datos Originales")
```



```
# Datos transformados (Modelo Exacto)
ggplot() +
  geom_density(aes(x = transformed_data), color = "red") +
  ggtitle("Densidad Empírica Datos Transformados (Modelo Exacto)")
```

Densidad Empírica Datos Transformados (Modelo Exa



```
library(nortest)
library(tseries)

# Pruebas de normalidad para datos originales
ad_test_original <- ad.test(mof)
jb_test_original <- jarque.bera.test(mof)

# Pruebas de normalidad para datos transformados (Modelo Exacto)
ad_test_exact <- ad.test(transformed_data)
jb_test_exact <- jarque.bera.test(transformed_data)

cat("Datos originales","\n")
## Datos originales
cat("Valor-p Anderson Darling:", ad_test_original$p.value,"\n")
## Valor-p Anderson Darling: 3.7e-24
cat("Valor-p Jarque Bera:", jb_test_original$p.value,"\n")
## Valor-p Jarque Bera: 0
cat("Datos Transformados","\n")
## Datos Transformados
```

```
cat("Valor-p Anderson Darling:", ad_test_exact$p.value, "\n")  
## Valor-p Anderson Darling: 1.377104e-11  
cat("Valor-p Jarque Bera:", jb_test_exact$p.value, "\n")  
## Valor-p Jarque Bera: 1.171828e-05
```

Despues de realizar la transformación y comparar con los datos originales se pudo concluir lo siguiente: La transformacion ha mejorado la normalidad en comparación con los datos originales, pero aún presentan desviaciones significativas. Esto sugiere que, aunque la transformacion Yeo-Johnson es util, no logró una normalización perfecta.

La transformacion ha reducido el sesgo y la curtosis, acercando los datos a una forma más normal, pero no completamente.

Visualmente, la forma de la densidad transformada es más cercana a la normalidad que la de los datos originales.

Los valores-p siguen siendo bajos, indicando que aunque hay una mejora, los datos transformados aún no cumplen con los criterios de normalidad.