5Es_A01571214_LautaroCoteja

A01571214 - Lautaro Coteja

2024-08-15

R Markdown

5. Transformaciones

Cargar/Leer Datos

```
# Cargar las bibliotecas necesarias
library(MASS)
library(car)
```

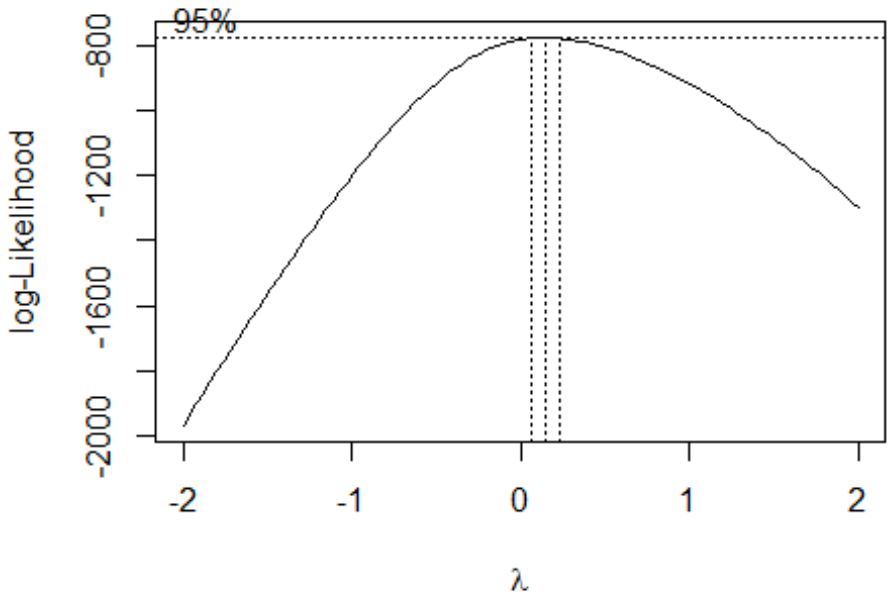
```
## Cargando paquete requerido: carData
```

```
library(e1071)
library(nortest)

# Cargar los datos
data = read.csv("C:/Users/lauta/Downloads/mc-donalds-menu.csv")

# Seleccionar la variable Sodium y eliminar valores nulos
sodium = data$Sodium
sodium = sodium[!is.na(sodium)]

# Eliminar los valores de cero de la variable Sodium para aplicar Box-Cox
sodium_nonzero = sodium[sodium > 0]

summary(sodium_nonzero)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     5.0   115.0   190.0   513.5   905.0  3600.0
```

Box-Cox

```
# Aplicar la transformación de Box-Cox
bc = boxcox(sodium_nonzero ~ 1)
```




```
lambda_bc = bc$x[which.max(bc$y)]

# Aplicar la transformación exacta de Box-Cox
sodium_boxcox = ((sodium_nonzero ^ lambda_bc) - 1) / lambda_bc

# Aplicar la transformación aproximada de Box-Cox (lambda = 1)
sodium_approx = sodium_nonzero + 1

# Ecuaciones de los modelos:
paste("Modelo exacto: (x^", round(lambda_bc, 4), "- 1) /", round(lambda_bc,
4))
```

```
## [1] "Modelo exacto: (x^ 0.1414 - 1) / 0.1414"
```

```
paste("Modelo aproximado: x + 1")
```

```
## [1] "Modelo aproximado: x + 1"
```

Medidas

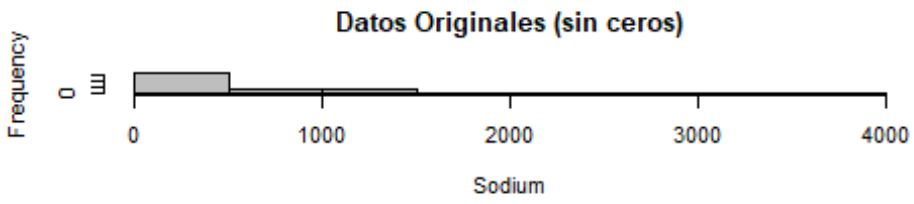
```
# Función para calcular curtosis y sesgo
curtosis = function(x) e1071::kurtosis(x)
sesgo = function(x) e1071::skewness(x)

# Calcular medidas descriptivas para cada transformación
medidas = data.frame(
```

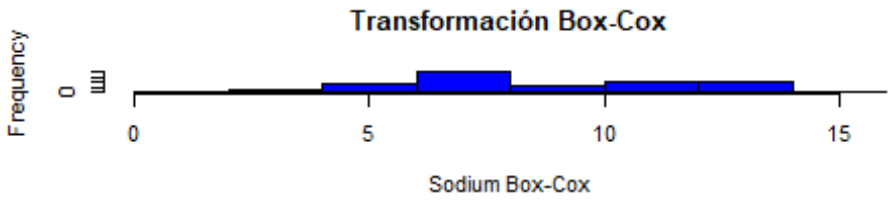
```
  Medida = c("Mínimo", "Máximo", "Media", "Mediana", "Cuartil 1", "Cuartil
3", "Sesgo", "Curtosis"),
  Original = c(min(sodium_nonzero), max(sodium_nonzero),
mean(sodium_nonzero), median(sodium_nonzero),
               quantile(sodium_nonzero, 0.25), quantile(sodium_nonzero,
0.75),
               sesgo(sodium_nonzero), curtosis(sodium_nonzero)),
  BoxCox = c(min(sodium_boxcox), max(sodium_boxcox), mean(sodium_boxcox),
median(sodium_boxcox),
             quantile(sodium_boxcox, 0.25), quantile(sodium_boxcox, 0.75),
             sesgo(sodium_boxcox), curtosis(sodium_boxcox)),
  Aproximado = c(min(sodium_approx), max(sodium_approx), mean(sodium_approx),
median(sodium_approx),
                 quantile(sodium_approx, 0.25), quantile(sodium_approx,
0.75),
                 sesgo(sodium_approx), curtosis(sodium_approx))
)

medidas
```

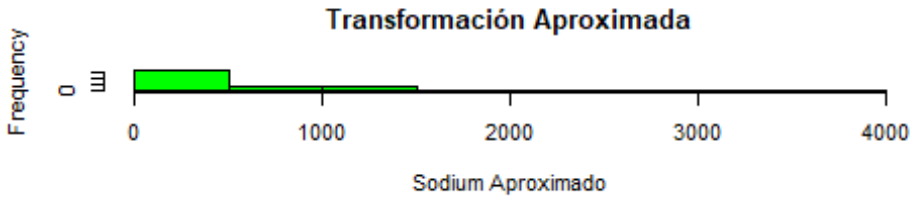
```
##      Medida    Original      BoxCox  Aproximado
## 1    Mínimo    5.000000  1.80731273    6.000000
## 2    Máximo 3600.000000 15.44080695 3601.000000
## 3     Media  513.525896  8.53410400  514.525896
## 4   Mediana  190.000000  7.77955244  191.000000
## 5 Cuartil 1  115.000000  6.76165417  116.000000
## 6 Cuartil 3  905.000000 11.44760310  906.000000
## 7     Sesgo    1.491921 -0.05013937    1.491921
## 8  Curtosis    2.650539 -0.78561423    2.650539
```

Histogramas

```
# Graficar histogramas comparativos
par(mfrow = c(3, 1))
hist(sodium_nonzero, col = "gray", main = "Datos Originales (sin ceros)",
xlab = "Sodium")
hist(sodium_boxcox, col = "blue", main = "Transformación Box-Cox", xlab =
"Sodium Box-Cox")
hist(sodium_approx, col = "green", main = "Transformación Aproximada", xlab =
"Sodium Aproximado")
```







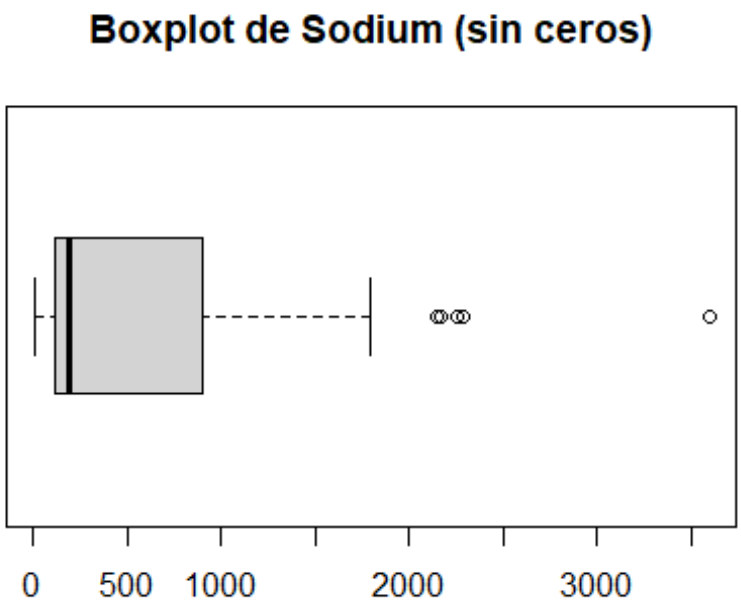
Pruebas de Normalidad

```
# Prueba de Anderson-Darling
ad_original = ad.test(sodium_nonzero)
ad_boxcox = ad.test(sodium_boxcox)
ad_approx = ad.test(sodium_approx)

resultados_normalidad <- data.frame(
  Transformación = c("Original", "Box-Cox", "Aproximada"),
  AD_Statistic = c(ad_original$statistic, ad_boxcox$statistic,
ad_approx$statistic),
  P_Value = c(ad_original$p.value, ad_boxcox$p.value, ad_approx$p.value)
)

resultados_normalidad
```

```
##   Transformación AD_Statistic      P_Value
## 1       Original    20.077141 3.700000e-24
## 2        Box-Cox     4.175196 2.090669e-10
## 3     Aproximada    20.077141 3.700000e-24
```

Deteccion de Errores y Correcion

```
boxplot(sodium_nonzero, main = "Boxplot de Sodium (sin ceros)", horizontal =
TRUE)
```




```
outliers = boxplot.stats(sodium_nonzero)$out
outliers
```

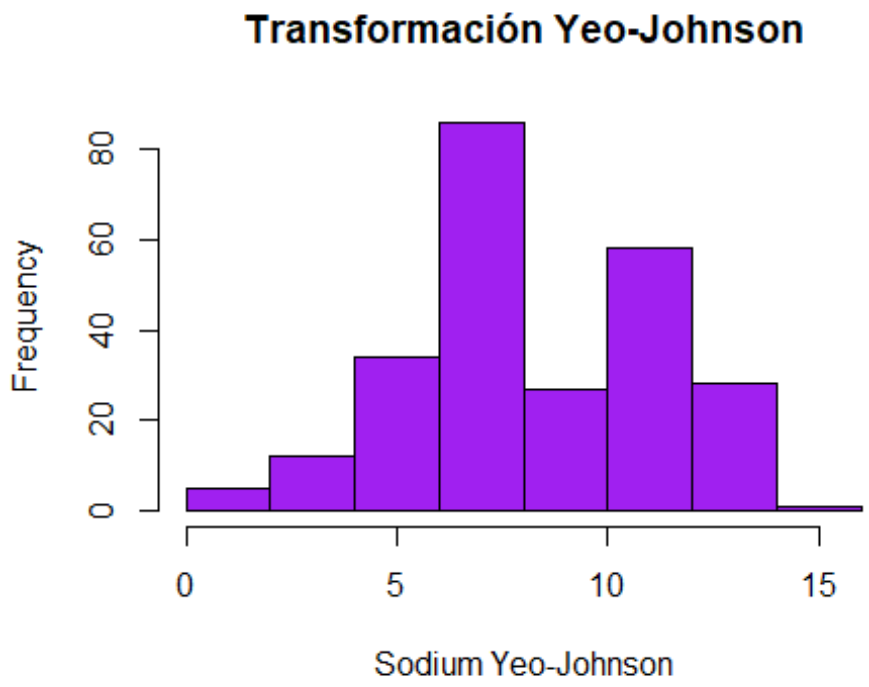
```
## [1] 2150 2260 2170 2290 3600
```

Transformacion Yeo-Johnson

```
# Aplicar la transformación de Yeo-Johnson
lambda_yj = powerTransform(sodium_nonzero, family = "yjPower")$lambda
sodium_yeojohnson = (sodium_nonzero ^ lambda_yj - 1) / lambda_yj

# Ecuación del modelo encontrado:
paste("Modelo Yeo-Johnson: (x^", round(lambda_yj, 4), "- 1) /",
round(lambda_yj, 4))
```

```
## [1] "Modelo Yeo-Johnson: (x^ 0.1325 - 1) / 0.1325"
```

```
# Graficar histograma de Yeo-Johnson
hist(sodium_yeojohnson, col = "purple", main = "Transformación Yeo-Johnson",
xlab = "Sodium Yeo-Johnson")
```



Analisis de Normalidad para Yeo-Johnson

```
# Calcular medidas descriptivas para Yeo-Johnson
medidas_yj = data.frame(
  Medida = c("Mínimo", "Máximo", "Media", "Mediana", "Cuartil 1", "Cuartil
3", "Sesgo", "Curtosis"),
  YeoJohnson = c(min(sodium_yeojohnson), max(sodium_yeojohnson),
mean(sodium_yeojohnson),
                 median(sodium_yeojohnson), quantile(sodium_yeojohnson,
0.25),
                 quantile(sodium_yeojohnson, 0.75), sesgo(sodium_yeojohnson),
                 curtosis(sodium_yeojohnson))
)

medidas_yj
```

```
##      Medida  YeoJohnson
## 1    Mínimo  1.79397422
## 2    Máximo 14.79073797
## 3     Media  8.28067860
## 4   Mediana  7.57942410
## 5 Cuartil 1  6.60571860
## 6 Cuartil 3 11.05546422
## 7     Sesgo -0.07412873
## 8  Curtosis -0.76260886
```

```r
# Prueba de Anderson-Darling para Yeo-Johnson
ad_yj = ad.test(sodium_yeojohnson)

ad_yj_resultados = data.frame(
  Transformación = "Yeo-Johnson",
  AD_Statistic = ad_yj$statistic,
  P_Value = ad_yj$p.value
)

ad_yj_resultados
```

```
##   Transformación AD_Statistic      P_Value
## A    Yeo-Johnson     4.102463 3.135253e-10
```

Conclusion

Mejor Transformacion

En cuanto a la mejor transformacion, esta depende de los datos, y se analiza comparando los valores de las pruebas de normalidad, el sesgo, curtosis, y los histogramas.

Ventajas y Desventajas de Box Cox y Yeo-Johnson

Box-Cox solo se aplica a datos positivos, pero es efectiva para datos que siguen una distribucion positiva y requieren normalizacion. Yeo-Johnson funciona con datos negativos, ceros y positivos, ofreciendo mayor flexibilidad en diferentes situaciones de datos.

Diferencias entre Transformacion y Escalamiento

La transformacion busca cambiar la distribucion de los datos, mientras que el escalamiento no. La transformacion afecta las relaciones entre datos, mientras que el escalamiento conserva la estructura de la distribucion Se debe utilizar la transformacion para normalizar datos y el escalamiento para comparar variables en diferentes escalas

Cuando utilizar cada uno

La transformacion cuando se necesita cumplir con supuestos de normalidad, como en regresion o ANOVA, y el escalamiento cuando se comparan variables con diferentes unidades o rangos de valores.