Actividad1_3

Ricardo Kaleb Flores Alfonso

2024-09-24

En la actividad 1_2, se evaluó la normalidad de dos variables en el conjunto de datos cars con pruebas estadísticas y análisis gráficos. Se usaron las pruebas de Anderson-Darling y Kolmogorov-Smirnov para probar normalidad en las distribuciones de las variables distancia y velocidad. Los resultados mostraron resultados distintos entre las pruebas para la variable distancia, lo que sugiere que mientras algunos datos centrales siguen una distribución normal, los valores extremos se desvían significativamente.

Se usaron gráficos QQPlot y boxplots, asi como el análisis con las medidas de sesgo y curtosis. Estos análisis revelaron que la variable distancia presenta un sesgo positivo, con una mayor cantidad de datos en la cola derecha, y una curtosis que indica la presencia de más datos extremos en comparación con una distribución normal. En contraste, la variable velocidad mostró una distribución más simétrica y menos valores extremos, lo que la aproxima más a una distribución normal.

Este análisis permitió identificar las características clave de los datos y estableció la necesidad de aplicar transformaciones para mejorar la normalidad en el caso de variables como distancia.

0) Se importan las librerias

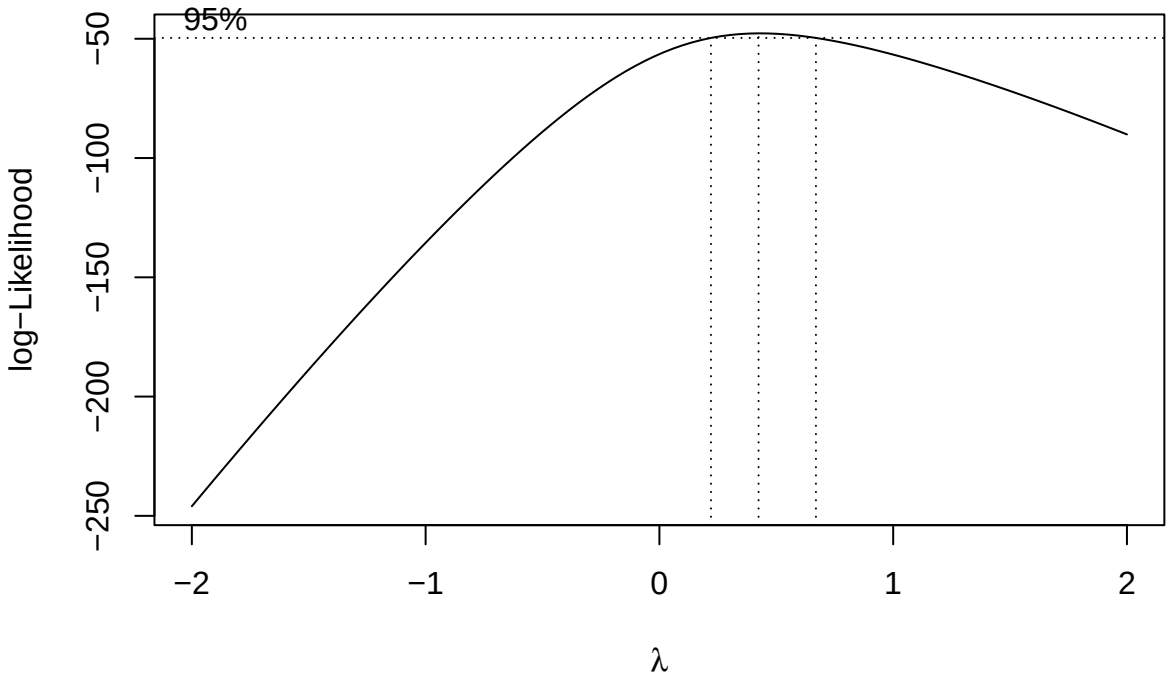
0.1) Se importa la base de datos

```
data(cars)
y <- cars$dist
x <- cars$speed
```

1) Efectua una transformación de los datos que te garantice normalidad en ambas variables

1.1) Encuentra el valor de lambda en la transformación de boxplot y usa la transformación exacta y aproximada

```
bc <- boxcox(lm(y~x))
```




```
l <- bc$x[which.max(bc$y)]
l
```

```
## [1] 0.4242424
```

Se obtiene un valor de $\lambda = 0.4242$ Por lo que se hacen las transformadas

```
dist1 <- sqrt(y+1)
dist2 <- ((y+1)^l-1)/l
speed1 <- sqrt(x+1)
speed2 <- ((x+1)^l-1)/l
```

1.3) Analiza la normalidad de las transformadas

Compara las medidas: Mínimo, máximo, media, mediana, cuartil 1 y cuartil 3, sesgo y curtosis.

```
summary(dist1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.732   5.196   6.083   6.334   7.550  11.000
```

```
print("Kurtosis")
```

```
## [1] "Kurtosis"
```

```
kurtosis(dist1)
```

```
## [1] -0.3647174
```

```r
print("Sesgo")
```

```
## [1] "Sesgo"
```

```r
skewness(dist1)
```

```
## [1] 0.02430858
```

```r
print("Valor de p")
```

```
## [1] "Valor de p"
```

```r
ad.test(dist1)$p.value
```

```
## [1] 0.9699563
```

```r
summary(dist2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.400   7.185   8.549   8.856  10.744  15.673
```

```r
print("Kurtosis")
```

```
## [1] "Kurtosis"
```

```r
kurtosis(dist2)
```

```
## [1] -0.2758653
```

```r
print("Sesgo")
```

```
## [1] "Sesgo"
```

```r
skewness(dist2)
```

```
## [1] -0.1126696
```

```r
print("Valor de p")
```

```
## [1] "Valor de p"
```

```r
ad.test(dist2)$p.value
```

```
## [1] 0.9785659
```

El sesgo (o skewness) mide la asimetría de la distribución de los datos. Los datos de distancia presentaban un sesgo negativo, mientras que después de la transformación, el sesgo se corrigió a estar más cerca del 0, haciendo que los datos fueran aún más simétricos.

Después de aplicar las transformaciones, observamos que los valores de curtosis se acercaron más a 0, lo que sugiere que las distribuciones se volvieron más normales, con menos datos extremos.

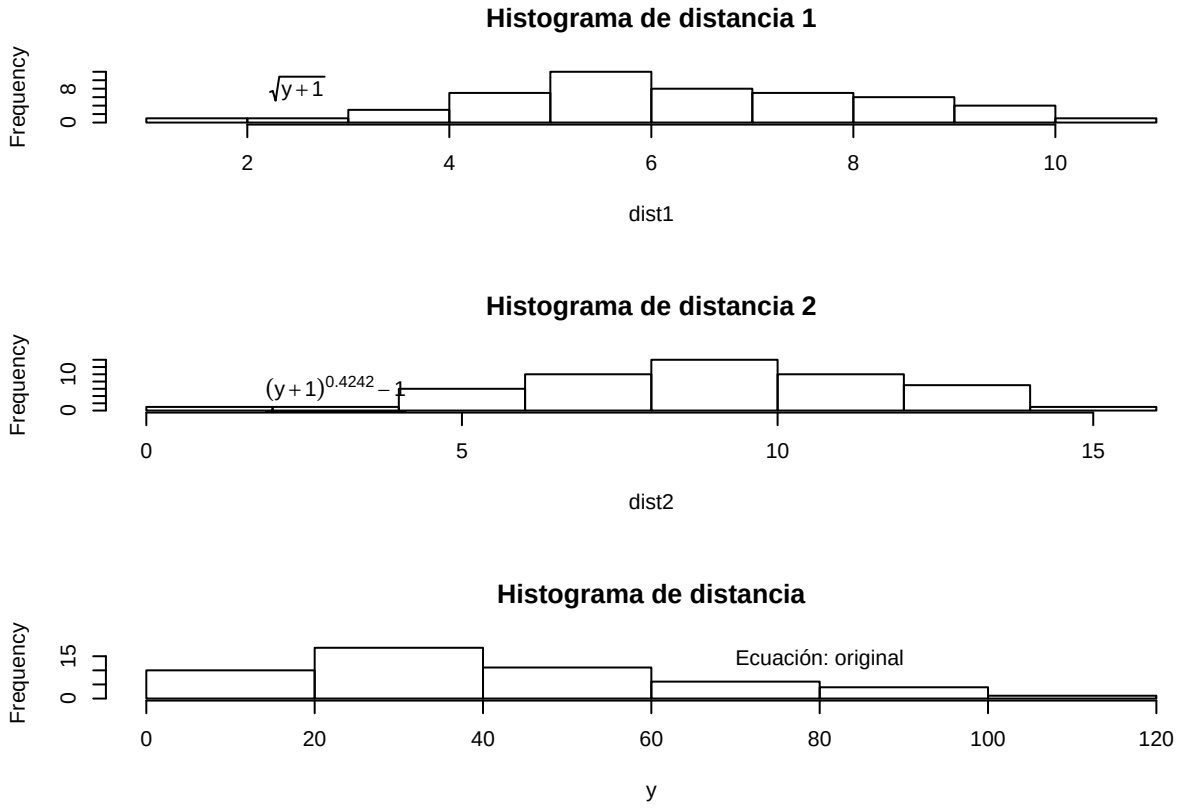
Estas medidas son importantes porque una distribución con un sesgo o una curtosis fuera de la normalidad puede afectar la precisión de los modelos que dependen de la suposición de normalidad, como la regresión lineal.

De igual manera se obtiene un rango menor en el cual estan distribuidos los datos, asi como un valor similar de media y promedio.

Esta transformada pasa normalidad segun el valor de $p > 0.05$

1.3.2) Obten el histograma de los dos modelos, exacto, aproximado y los datos originales

```
par(mfrow=c(3,1))
hist(dist1,col=0,main="Histograma de distancia 1")
text(x=2.5, y=8, expression(dist1=sqrt(y+1)))
hist(dist2,col=0,main="Histograma de distancia 2")
text(x=3, y=1, expression(dist2= frac((y+1)^0.4242-1,0.4242)))
hist(y,col=0,main="Histograma de distancia")
text(x=80, y=14, "Ecuación: original")
```



1.3.3) Realiza pruebas de normalidad para los datos transformados

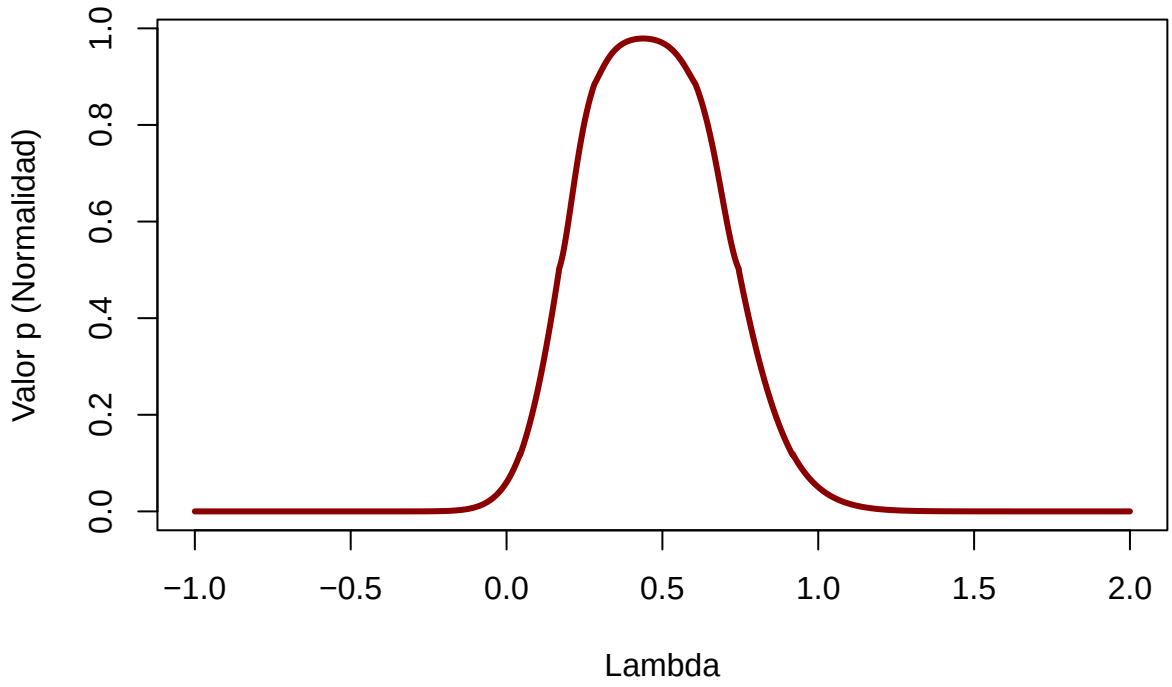
En la prueba de Anderson-Darling, el p-valor indica la probabilidad de que los datos sigan una distribución normal. Un p-valor mayor a 0.05 sugiere que no se puede rechazar la hipótesis nula de que los datos son normales. En cambio, un p-valor menor a 0.05 indica que los datos probablemente no siguen una distribución normal. En este caso los valores de p obtenidos son mayores que 0.05 por lo que los datos transformados siguen una distribución normal.

2) Utiliza la transformación de Yeo Johnson y encuentra el valor de lambda que maximiza el valor p

```
lp <- seq(-1,2,0.001)
nlp <- length(lp)
n=length(y)
D <- matrix(as.numeric(NA),ncol=2,nrow=nlp)
```

```
d <- NA
for (i in 1:nlp){
  d=yeo.johnson(y,lambda=lp[i])
  p=ad.test(d)
  D[i,]=c(lp[i],p$p.value)}

N = as.data.frame(D)
plot(N$V1,N$V2,type="l",col="darkred",lwd=3,xlab="Lambda",ylab="Valor p (Normalidad)")
```

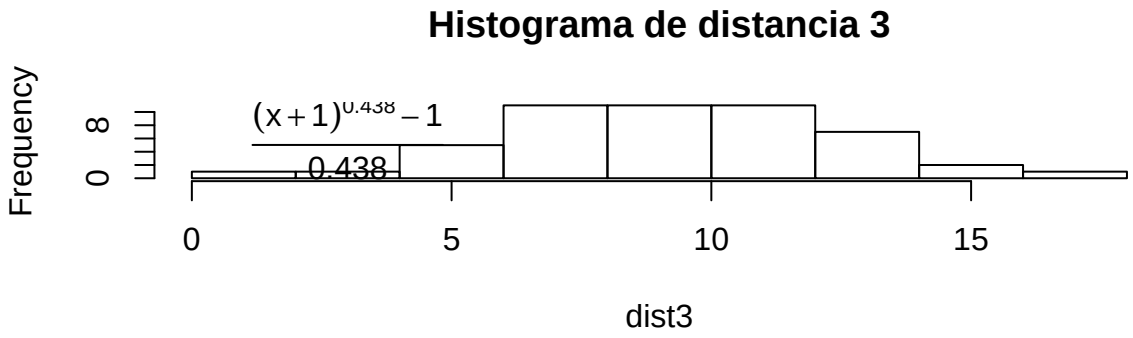


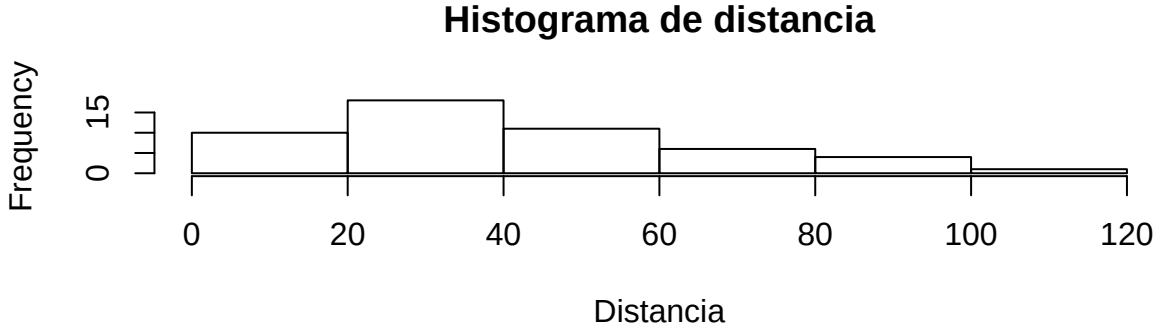

```
G = data.frame(subset(N,N$V2==max(N$V2)))
G
```

```
##         V1        V2
## 1439 0.438 0.9789807
```

2.1) Escribe la ecuación de la transformación encontrada

```
dist3 <- yeo.johnson(y,G$V1)
par(mfrow=c(2,1))
hist(dist3,col=0,main="Histograma de distancia 3")
text(x=3, y=6, expression(speed2= frac((x+1)^0.438-1,0.438)))
hist(y,col=0,main="Histograma de distancia", xlab="Distancia")
```





2.2) Analiza la normalidad de la transformación obtenida

```
summary(dist3)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.411   7.388   8.819   9.155  11.132  16.371
```

```
print("Curtosis")
```

```
## [1] "Curtosis"
```

```
kurtosis(dist3)
```

```
## [1] -0.2964252
```

```
print("Sesgo")
```

```
## [1] "Sesgo"
```

```
skewness(dist3)
```

```
## [1] -0.08710045
```

```
print("Valor p")
```

```
## [1] "Valor p"
```

```
ad.test(dist3)$p.value
```

```
## [1] 0.9789807
```

Después de aplicar la transformación de Yeo-Johnson con $\lambda = 0.438$, se observa que la mediana y la media

tienen valores cercanos, lo que significa que hay una distribución simétrica. El sesgo obtenido es -0.0871, mostrando una minima inclinación hacia la izquierda, pero muy cercana a cero, lo que refuerza la idea de la distribución simétrica. La curtosis obtenida es de -0.2964 lo que nos lleva a tener una distribución que es platicúrtica, es decir, tiene colas más delgadas que una distribución normal, con menos valores extremos. La prueba de Anderson-Darling obtuvo un p-valor de 0.9789, lo cual indica que no hay evidencia suficiente para rechazar la hipótesis de normalidad, confirmando que la transformación es normal.Esta transformación ha sido eficaz para aproximar los datos a una distribución normal, lista para su uso en un modelo de regresión lineal.

3) Concluye sobre las transformaciones realizadas

De acuerdo con los resultados obtenidos, la transformación de Yeo-Johnson con $\lambda = 0.438$ es la mejor opción, ya que maximiza el p-valor de normalidad 0.9789 y presenta mejores valores de simetría y curtosis en comparación con las transformaciones de Box-Cox. De igual manera es una transformación más flexible, lo que contribuye a una mayor economía del modelo.

4) Con la mejor transformación, encuentra la regresión lineal simple

```
model <- lm(dist3 ~ x)
intercept <- model$coefficients[1]
values <- model$coefficients[2]
model
```

```
##
## Call:
## lm(formula = dist3 ~ x)
##
## Coefficients:
## (Intercept)            x
##      1.3390       0.5075
```

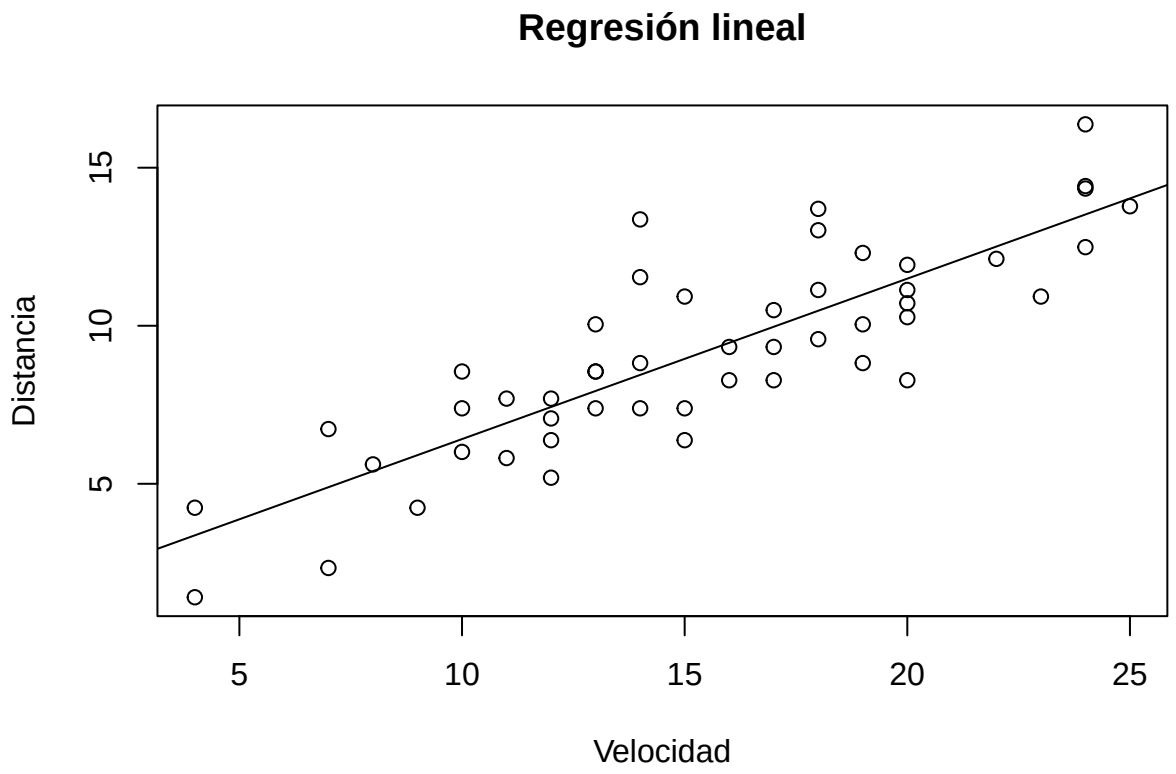
4.1) Escribe el modelo lineal

Modelo Obtenido

$$y = 1.339 + 0.5075x$$

4.2) Grafica los datos y el modelo (ecuación) de transformación elegida vs velocidad.

```
plot(x, dist3 ,ylab = "Distancia" ,xlab = "Velocidad")
abline(model)
title("Regresión lineal")
text(x=8, y=80, "Ecuación: y= -18.39 + 4.15x")
```



4.3) Analiza significancia del modelo ### 4.3.1) Individual, conjunta y correlación

```
summary(model)
```

```
##
## Call:
## lm(formula = dist3 ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2137 -1.0952 -0.3026  0.8604  4.9196
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.33902    0.75845   1.765   0.0838 .
## x            0.50755    0.04663  10.885 1.47e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.726 on 48 degrees of freedom
## Multiple R-squared:  0.7117, Adjusted R-squared:  0.7057
## F-statistic: 118.5 on 1 and 48 DF,  p-value: 1.469e-14
```

T Test

- H_0 := El coeficiente es igual a 0.
- H_A := El coeficiente no es igual a 0.

Significancia de los coeficientes de regresión:

El modelo se ajusta al 70.57% de los datos, de igual manera la prueba de p, nos da un valor debajo de 0.05, por lo que se rechaza la hipotesis nula, demostrando asi que los coeficientes de x son significantes.

La F-statistic del modelo es 118.5, con un p-valor de $1.469e^{-14}$. Esto indica que el modelo es globalmente significativo. Por lo que explica una parte considerable de la variabilidad en distancia.

4.4) Linealidad

- H_0 := La relación entre la variable independiente y dependiente es lineal
- H_A := La relación es no lineal

```
resettest(model)
```

```
##
##  RESET test
##
## data:  model
## RESET = 0.51859, df1 = 2, df2 = 46, p-value = 0.5988
```

Se obtiene un p-value para la prueba de linealidad de 0.5988, por lo que no hay evidencia suficiente para rechazar la hipotesis nula, de esta manera se demuestra que existe una relación lineal entre la variable dependiente e independiente.

4.5) Media de cero de los residuos

T Test

- H_0 := La media de los errores es igual a 0.
- H_A := La media de los errores no es igual a 0.

```
t.test(model$residuals)
```

```
##
##  One Sample t-test
##
## data:  model$residuals
## t = 1.8167e-16, df = 49, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.4854726  0.4854726
## sample estimates:
##    mean of x
## 4.388742e-17
```

Se obtuvo que el resultado de la prueba da como resultado que p = 1. Por lo que no es posible rechazar la hipotesis nula. Por lo tanto se tiene un error promedio de 0 en el modelo.

4.6) Normalidad de residuos

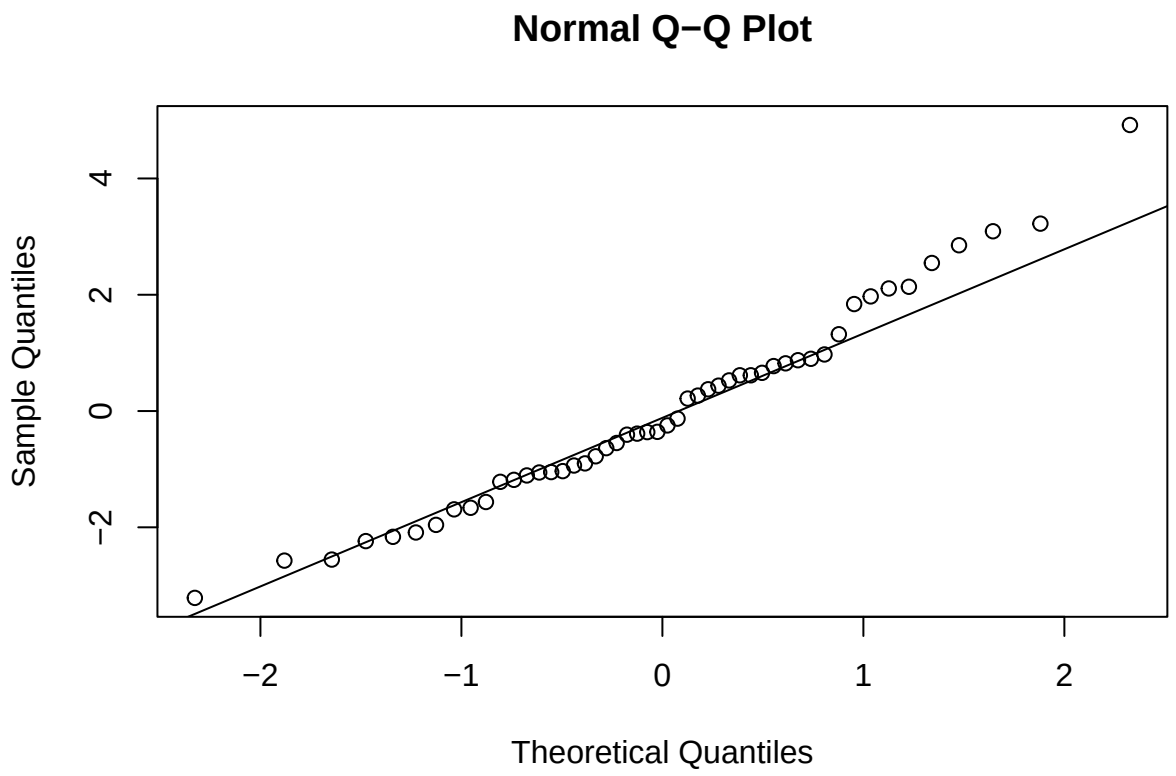
T Test

- H_0 := Los residuos tienen una distribución normal
- H_A := Los residuos no tienen una distribución normal

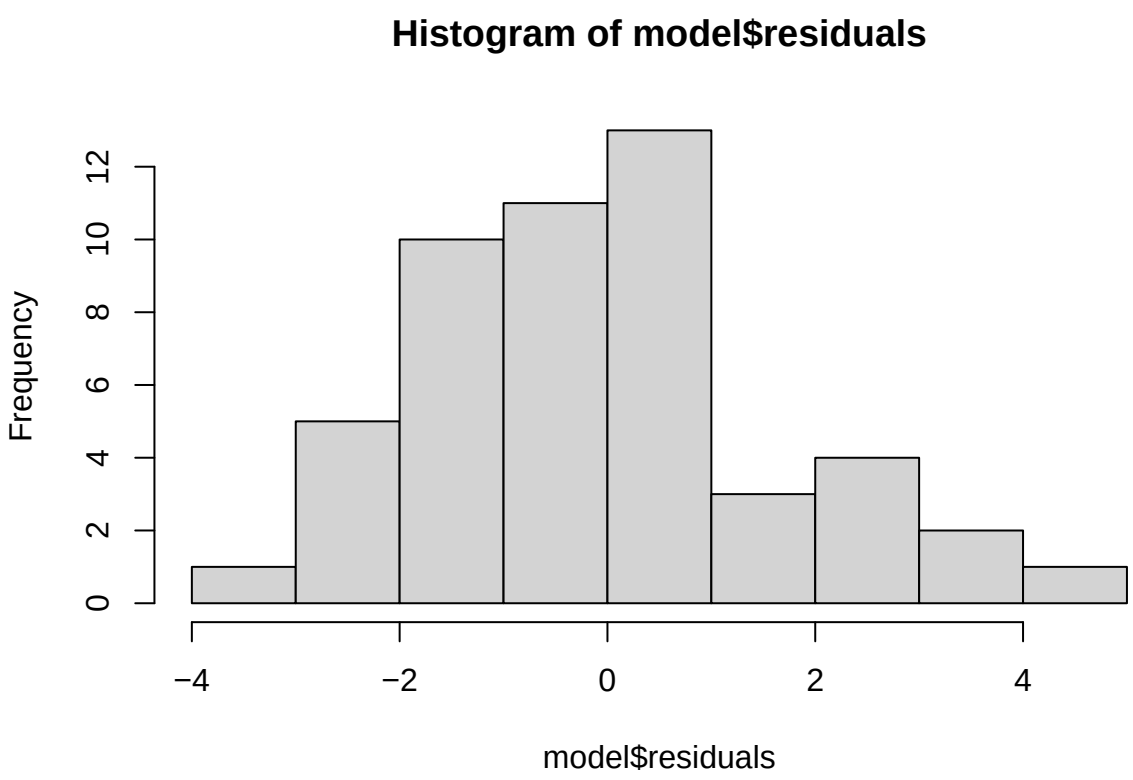
```
shapiro.test(model$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model$residuals
## W = 0.97502, p-value = 0.3656
```

```r
qqnorm(model$residuals)
qqline(model$residuals)
```




```r
hist(model$residuals)
```



Se obtiene un valor de $p = 0.3656$, por lo tanto no existe evidencia para rechazar la hipotesis nula. De esta manera se sabe que los residuos no se desvian de una distribución normal.

4.7) Breusch-Pagan

- H_0 := Los datos tienen homocedasticidad.
- H_A := Los datos no tienen homocedasticidad.

```
bptest(model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 0.011121, df = 1, p-value = 0.916
```

El test de Breusch-Pagan produce un valor de p de 0.916, dado que es mayor que 0.05. Se falla en rechazar la hipotesis nula. Esto demuestra que el modelo tiene varianza constante.

4.8) Independencia

Test de Durbin Watson

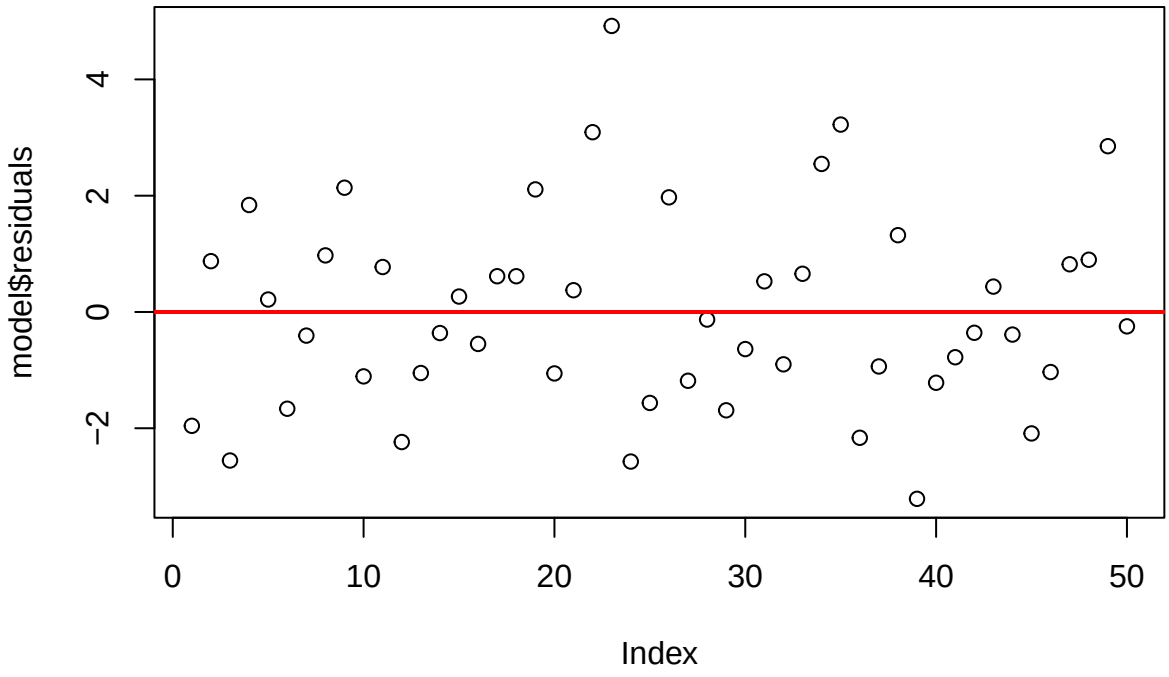
- H_0 := No existe autocorrelación en los datos
- H_A := Existe autocorrelacion en los datos.

```
dwtest(model)
```

```
##
```

```
##  Durbin-Watson test
##
## data:  model
## DW = 1.9539, p-value = 0.3772
## alternative hypothesis: true autocorrelation is greater than 0
```

```
plot(model$residuals)
abline(h=0, col = "red", lwd = 2)
```



Dado que el valor obtenido por el test de durbin watson da un valor de 0.3772 y esto es mayor que 0.05, no existe evidencia para rechazar la hipotesis nula. Esto significa que no existe correlación en los residuos, por lo que estos son independientes.

4.9) Despeja la distancia del modelo obtenido entre la transformación y la velocidad. Obtendras el modelo no lineal que relaciona distancia con velocidad

$$dist3 = 1.339 + 0.5075x$$

$$dist3 = \frac{(y+1)^{0.438} - 1}{0.438}$$

$$\frac{(y+1)^{0.438} - 1}{0.438} = 1.339 + 0.5075x$$

$$y + 1 = (1 + 0.438(1.339 + 0.5075x))^{\frac{1}{0.438}}$$

$$y = (1 + 0.438(1.339 + 0.5075x))^{\frac{1}{0.438}} - 1$$

$

4.10) Grafica los datos y el modelo de la distancia en función de la velocidad

```r
# Gráfica de los datos originales
plot(x, y, xlab = "Velocidad", ylab = "Distancia", main = "Modelo No Lineal: Distancia vs Velocidad")

# Define la ecuación no lineal
modelo_no_lineal <- function(x) {
  (1 + 0.438 * (1.339 + 0.5075 * x))^(1 / 0.438) - 1
}

# Agregar la curva del modelo no lineal
curve(modelo_no_lineal(x), add = TRUE, col = "blue", lwd = 2)

# Título y leyenda
legend("topleft", legend = "Modelo no lineal", col = "blue", lty = 1, cex = 0.8)
```