Act13 Regresión no Lineal

Andrés Villarreal González

2024-09-10

Regresión No Lineal

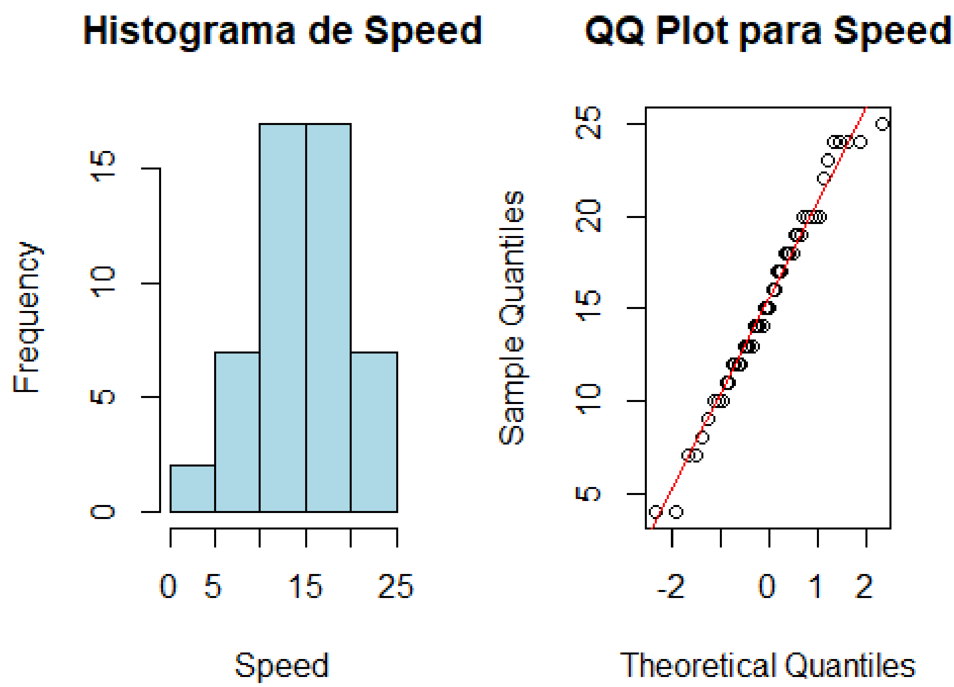
Parte 1 Análisis de Normalidad

Prueba normalidad univariada de la velocidad y distancia (prueba con dos de las pruebas vistas en clase)

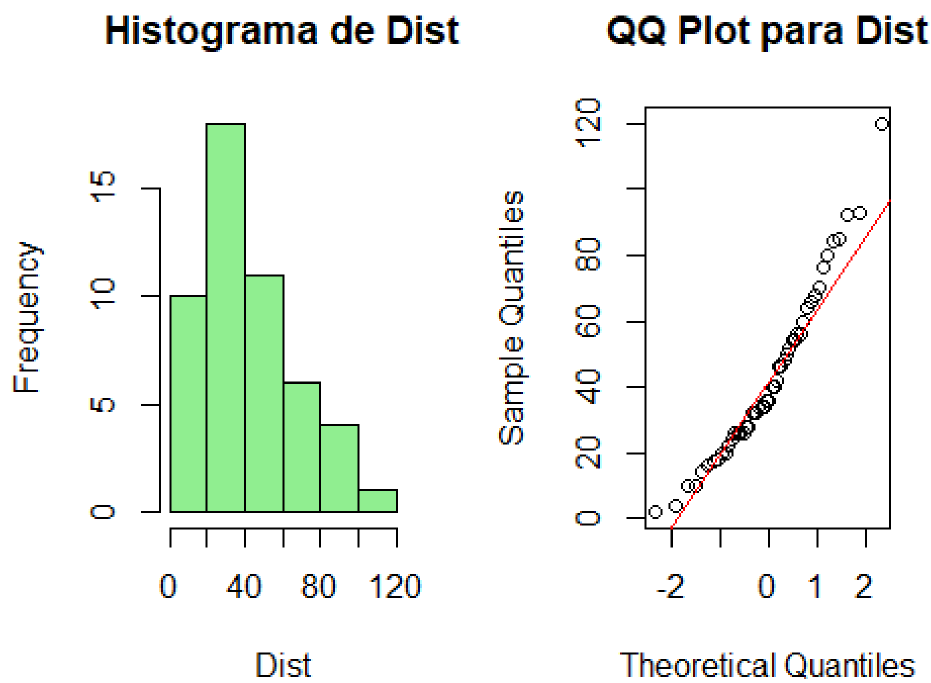
```
data = cars

shapiro.test(data$speed)

##
##  Shapiro-Wilk normality test
##
## data:  data$speed
## W = 0.97765, p-value = 0.4576

shapiro.test(data$dist)

##
##  Shapiro-Wilk normality test
##
## data:  data$dist
## W = 0.95144, p-value = 0.0391
```

Histogramas y QQPlot

```
# Configurar espacio para dos gráficos en una fila
par(mfrow = c(1, 2))

# Gráficos para speed
# Histograma
hist(data$speed, main = "Histograma de Speed", xlab = "Speed", col =
"lightblue", border = "black")

# QQ plot
qqnorm(data$speed, main = "QQ Plot para Speed")
qqline(data$speed, col = "red")
```




```r
# Configurar nuevamente para los gráficos de dist
par(mfrow = c(1, 2))

# Gráficos para dist
# Histograma
hist(data$dist, main = "Histograma de Dist", xlab = "Dist", col =
"lightgreen", border = "black")

# QQ plot
qqnorm(data$dist, main = "QQ Plot para Dist")
qqline(data$dist, col = "red")
```



Sesgo y Curtosis

```
library(e1071)
# Calcular sesgo para speed y dist
skew_speed <- skewness(data$speed)
skew_dist <- skewness(data$dist)

# Calcular curtosis para speed y dist
kurt_speed <- kurtosis(data$speed)
kurt_dist <- kurtosis(data$dist)

# Mostrar los resultados
cat("Sesgo y Curtosis de las variables:\n")

## Sesgo y Curtosis de las variables:

cat("Sesgo de Speed:", skew_speed, "\n")

## Sesgo de Speed: -0.1105533

cat("Curtosis de Speed:", kurt_speed, "\n\n")

## Curtosis de Speed: -0.6730924

cat("Sesgo de Dist:", skew_dist, "\n")

## Sesgo de Dist: 0.7591268

cat("Curtosis de Dist:", kurt_dist, "\n")
```

```
## Curtosis de Dist: 0.1193971
```

Regresión Lineal

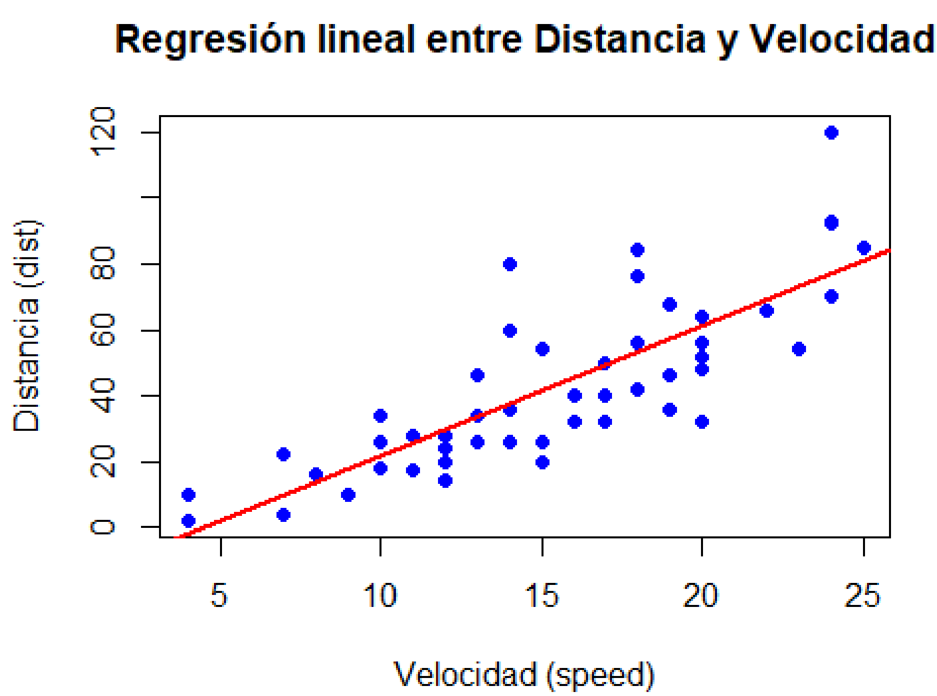
```r
model <- lm(data$dist ~ data$speed, data=data)
```

Coeficientes del modelo

```r
# Obtener los coeficientes del modelo
coeficientes <- coef(model)
cat("El modelo lineal es: dist =", coeficientes[1], "+", coeficientes[2],
"* speed\n")

## El modelo lineal es: dist = -17.57909 + 3.932409 * speed
```

Grafica los datos y el modelo (ecuación) que obtuviste.

```r
# Graficar los datos y la línea de regresión
plot(data$speed, data$dist, main = "Regresión lineal entre Distancia y
Velocidad",
     xlab = "Velocidad (speed)", ylab = "Distancia (dist)", pch = 19, col
= "blue")

# Añadir la línea de regresión al gráfico
abline(model, col = "red", lwd = 2)
```



Analiza significancia del modelo: individual, conjunta y coeficiente de determinación. Usa summary(Modelo)

```r
summary(model)
```

```
##
## Call:
## lm(formula = data$dist ~ data$speed, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## data$speed    3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

El intercepto de -17.5791 indica que, según el modelo, cuando la velocidad es 0, la distancia predicha es negativa, lo cual no tiene un significado físico claro. Esto sugiere que el modelo no es adecuado para valores muy bajos de velocidad, lo que puede ser esperado debido a la naturaleza de los datos y su posible límite práctico en condiciones reales.

El coeficiente de speed es 3.9324, lo que significa que por cada aumento de una unidad en la velocidad, la distancia de frenado se incrementa en aproximadamente 3.93 unidades. Este coeficiente es estadísticamente significativo ($p < 0.001$), lo que indica que hay una relación fuerte y positiva entre la velocidad y la distancia de frenado.

Analiza validez del modelo.

Residuos con media cero

$H_0: \mu = 0$ $H_1: \mu \neq 0$

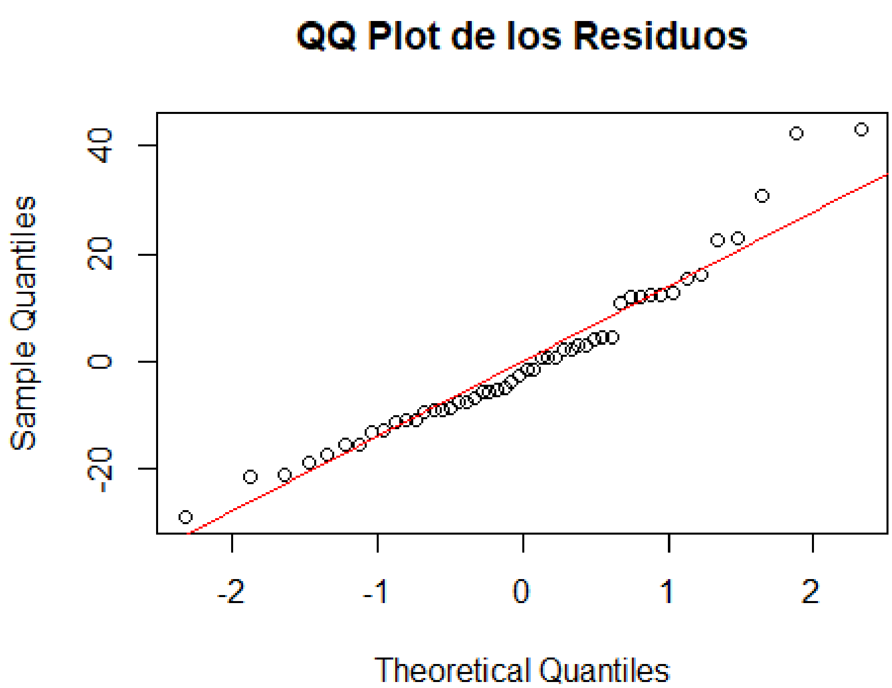
```r
# Obtener los residuos del modelo
residuos <- residuals(model)

# Verificar si la media de los residuos es aproximadamente cero
mean_residuos <- mean(residuos)
cat("Media de los residuos:", mean_residuos, "\n")

## Media de los residuos: 2.220446e-16
```

Este resultado confirma que el modelo ajustado cumple con el supuesto de que los residuos tienen una media cercana a cero, lo cual es esperado en un modelo de regresión lineal bien ajustado.

Normalidad de los residuos

H_0: Los datos provienen de una población normal H_1: Los datos no provienen de una población normal

```
# Prueba de normalidad de los residuos
shapiro.test(residuos)

##
##  Shapiro-Wilk normality test
##
## data:  residuos
## W = 0.94509, p-value = 0.02152

# QQ plot de los residuos
qqnorm(residuos, main = "QQ Plot de los Residuos")
qqline(residuos, col = "red")
```



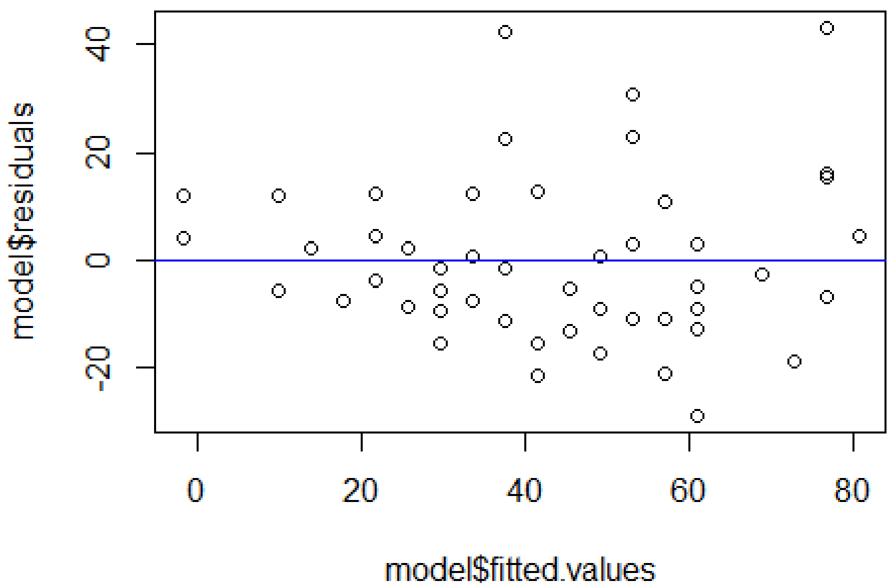
En este caso, el valor p (0.02152) es menor que 0.05, lo que significa que rechazamos la hipótesis nula de que los residuos siguen una distribución normal.

Homocedasticidad, independencia y linealidad.

H_0: La varianza de los errores es constante (homocedasticidad) H_1: La varianza de los errores no es constante (heterocedasticidad)

H_0: Los errores no están correlacionados H_1: Los errores están correlacionados

```r
plot(model$fitted.values,model$residuals)
abline(h=0, col="blue")
```




```r
library(lmtest)

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

dwtest(model)

##
##  Durbin-Watson test
##
## data:  model
## DW = 1.6762, p-value = 0.09522
## alternative hypothesis: true autocorrelation is greater than 0

bgtest(model)

##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
```

```
## data:  model
## LM test = 1.2908, df = 1, p-value = 0.2559
```

```r
library(lmtest)
bptest(model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 3.2149, df = 1, p-value = 0.07297
```

```r
gqtest(model)
```

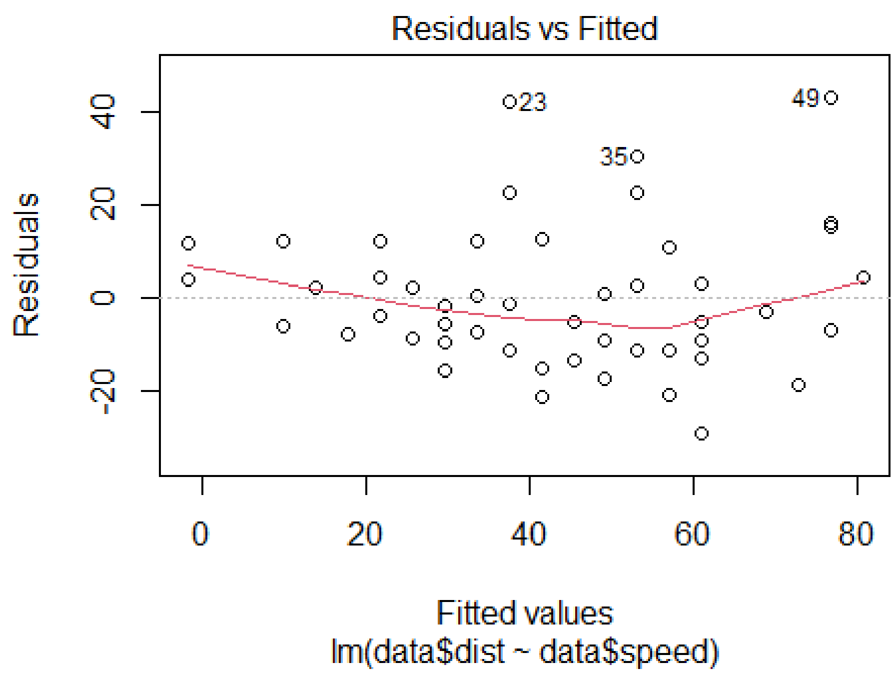
```
##
##  Goldfeld-Quandt test
##
## data:  model
## GQ = 1.5512, df1 = 23, df2 = 23, p-value = 0.1498
## alternative hypothesis: variance increases from segment 1 to 2
```

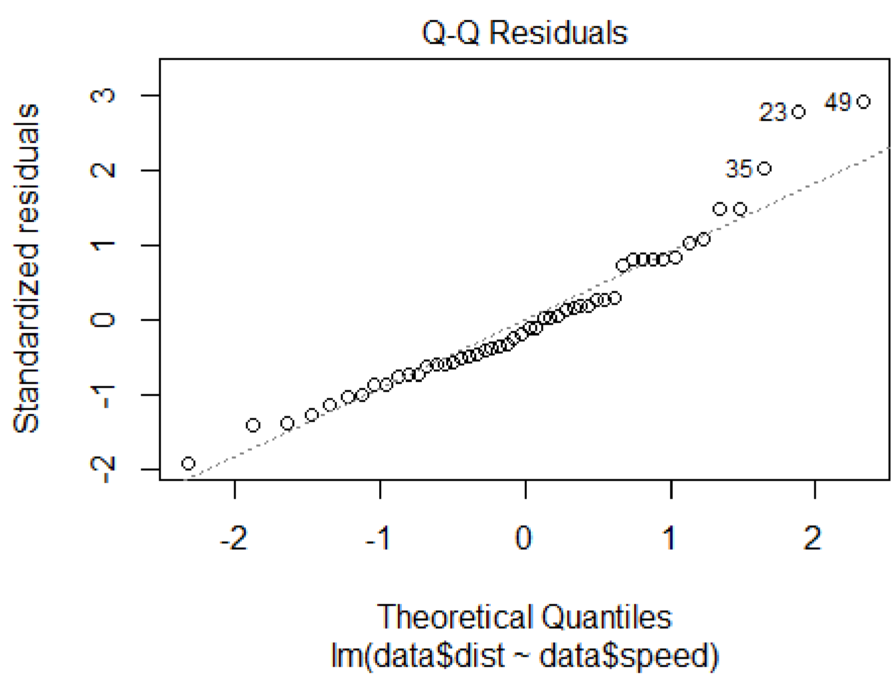
No se observa autocorrelación significativa en los residuos. Esto es favorable para el modelo, ya que la autocorrelación podría indicar una violación del supuesto de independencia de los residuos. El test Breusch-Godfrey refuerza el resultado del test de Durbin-Watson, sugiriendo que no hay autocorrelación significativa en los residuos.

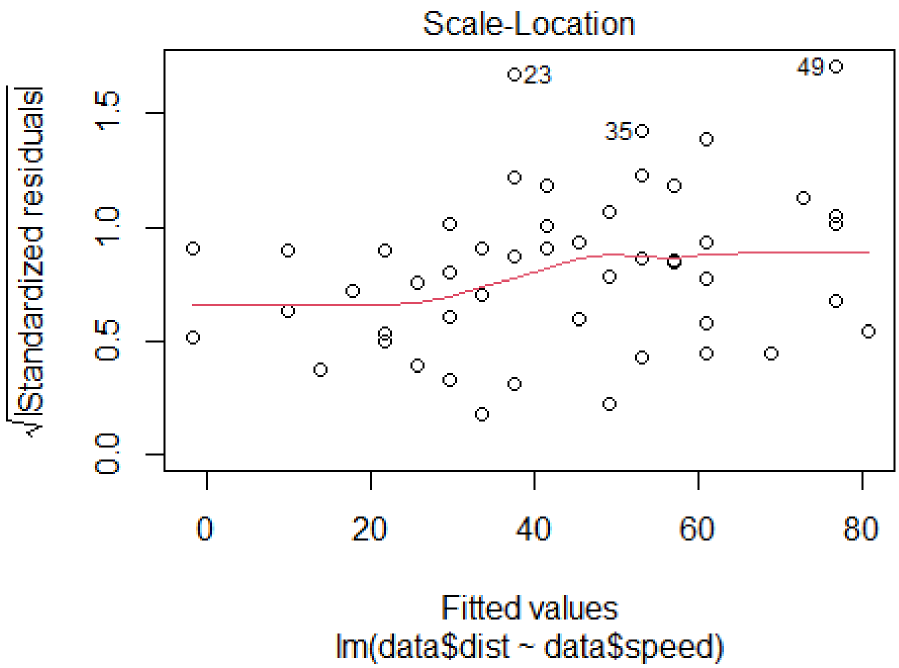
No se rechaza la hipótesis de homocedasticidad, pero el valor p cercano a 0.05 indica que podría haber alguna variabilidad en los residuos que requiera más atención o análisis. No hay evidencia de heterocedasticidad significativa. Los residuos parecen tener una varianza constante, lo que refuerza la validez del modelo bajo el supuesto de homocedasticidad.

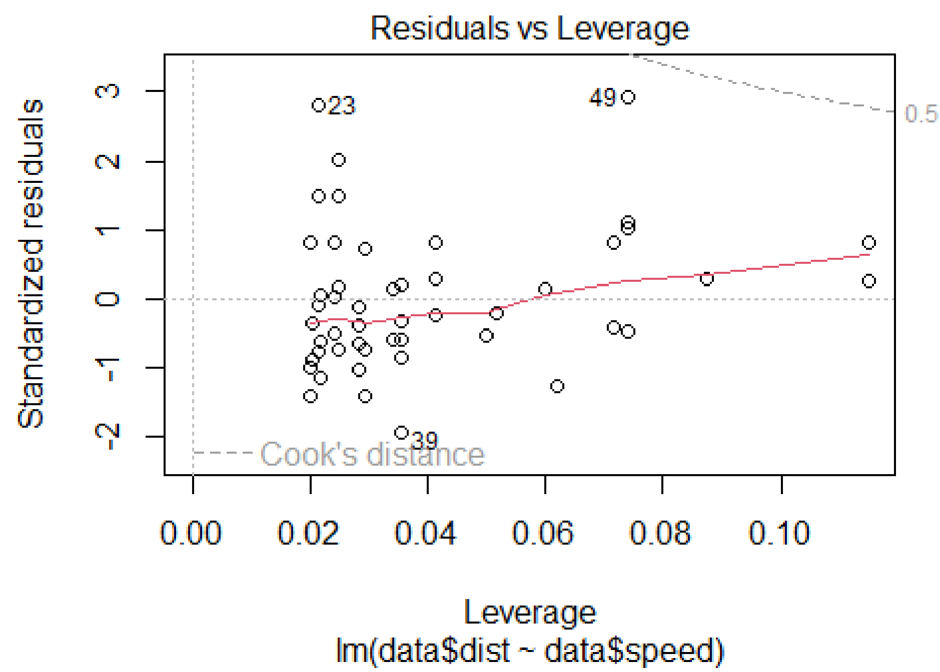
Usa plot(Modelo) para los gráficos y añade pruebas de hipótesis.

```r
plot(model)
```


Residuals vs Fitted
Residuals
Fitted values
lm(data$dist ~ data$speed)
23
35
49
-20
0
20
40
0
20
40
60
80


Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(data$dist ~ data$speed)
23
35
49
-2
-1
0
1
2
3
-2
-1
0
1
2

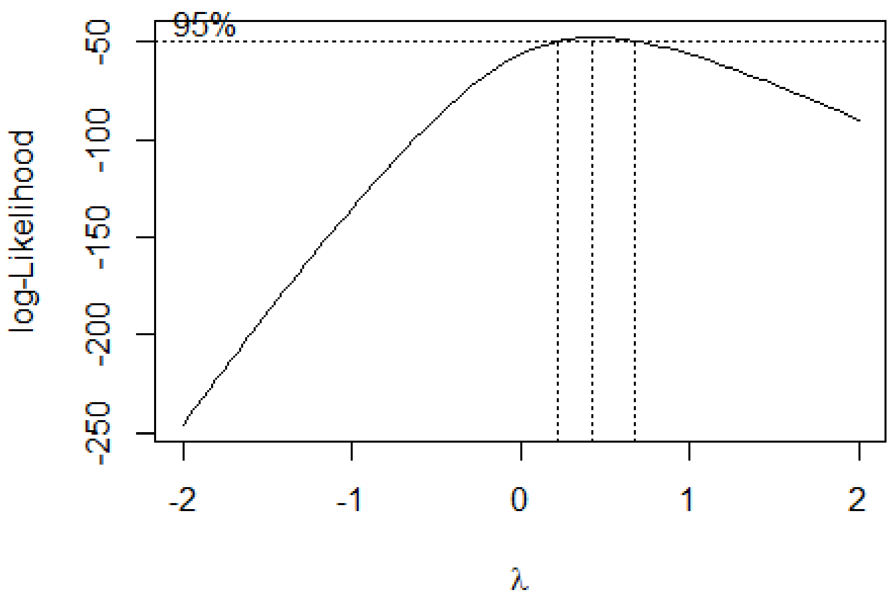




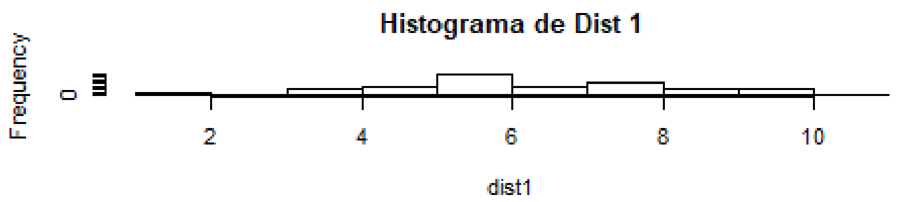
Regresion No Lineal

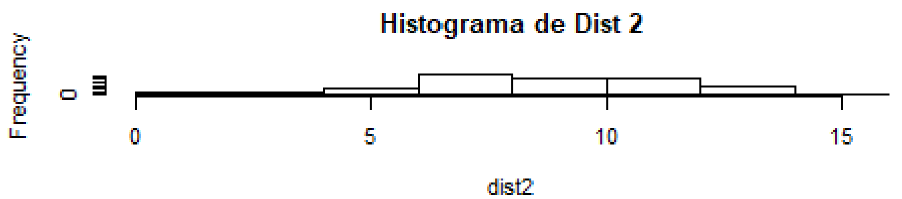
```
library(MASS)
x <- cars$speed
```

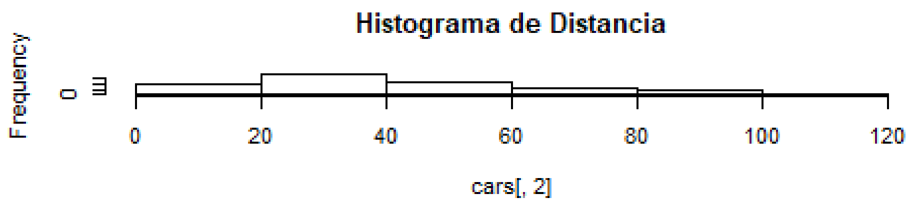
```r
y <- cars$dist
bc <- boxcox(lm(y~x))
```




```r
lambda <- bc$x[which.max(bc$y)]
cat("El valor de λ que maximiza la función de verosimilitud es:", lambda)

## El valor de λ que maximiza la función de verosimilitud es: 0.4242424

dist1=sqrt(cars[ ,2])
dist2=((cars[ ,2])^lambda-1)/lambda
par(mfrow=c(3,1))
hist(dist1,col=0,main="Histograma de Dist 1")
hist(dist2,col=0,main="Histograma de Dist 2")
hist(cars[ ,2],col=0,main="Histograma de Distancia")
```







En los histogramas con los datos transformados se pueden observar como hay mas normalidad en los datos dado que el valor p de estos aumentaron hasta 0.28 y 0.19 respectivamente

Transformación aproximada

```
library(e1071)
library(nortest)
cat("Transformacion aproximada: \n")

## Transformacion aproximada:

summary(dist1)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.414   5.099   6.000   6.242   7.483  10.954

cat("Curtosis:", kurtosis(dist1), "\n")

## Curtosis: -0.3144682

cat("Sesgo:", skewness(dist1), "\n")

## Sesgo: -0.01902765

D=ad.test(dist1)
cat("Valor p:",D$p.value," \n")

## Valor p: 0.9731952
```

Transformación exacta

```r
library(e1071)
library(nortest)
cat("Transformacion aproximada: \n")

## Transformacion aproximada:

summary(dist2)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.8058  7.0331  8.4233  8.7116 10.6458 15.6093

cat("Curtosis:", kurtosis(dist2), "\n")

## Curtosis: -0.186884

cat("Sesgo:", skewness(dist2), "\n")

## Sesgo: -0.1701619

D=ad.test(dist2)
cat("Valor p:",D$p.value," \n")

## Valor p: 0.9717478
```

Se puede observar viendo los histogramas que la mejor transformacion es la exacta.

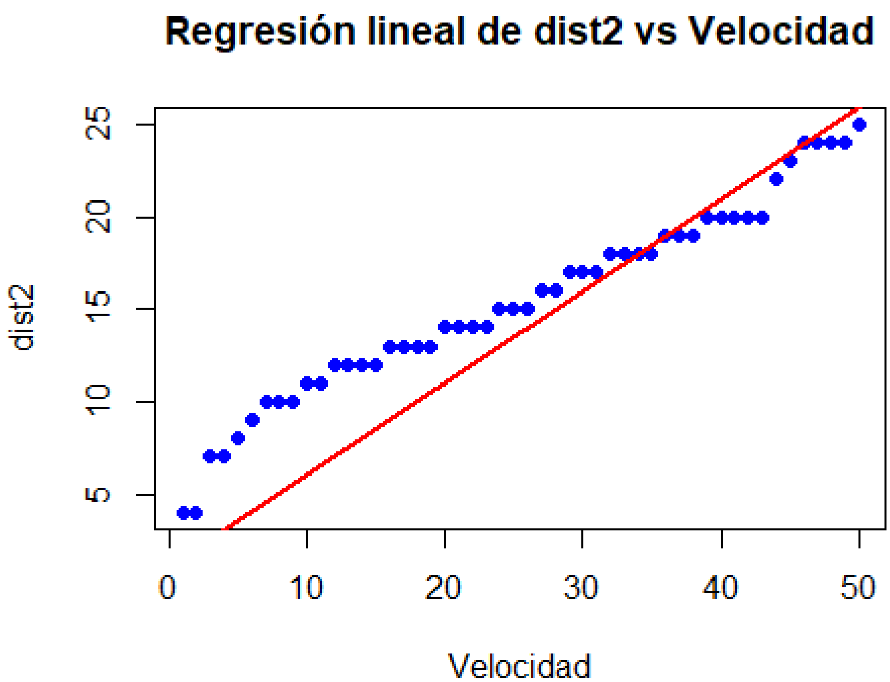
Con la mejor transformación (punto 2), realiza la regresión lineal simple entre la mejor transformación (exacta o aproximada) y la variable velocidad:

```r
# Realizar la regresión lineal simple entre dist2 (transformación) y speed (velocidad)
modelo_transformado <- lm(dist2 ~ speed, data = cars)

# Mostrar los coeficientes del modelo
summary(modelo_transformado)

##
## Call:
## lm(formula = dist2 ~ speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0926 -1.0444 -0.3055  0.7999  4.7520
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.08227    0.73856   1.465    0.149
## speed        0.49541    0.04541  10.910 1.35e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.681 on 48 degrees of freedom
```

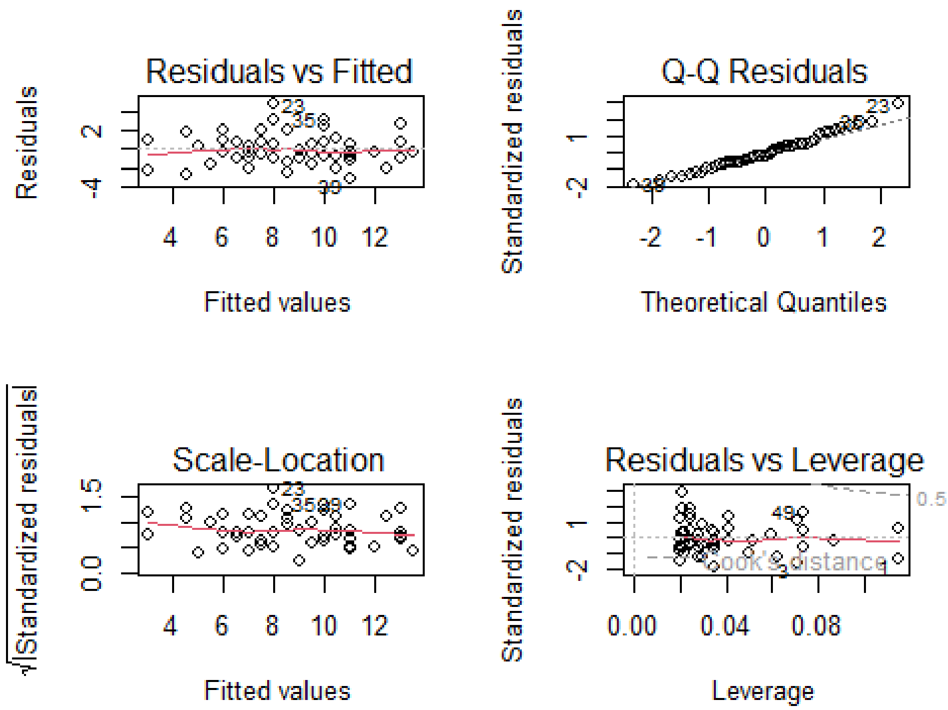
```
## Multiple R-squared:  0.7126, Adjusted R-squared:  0.7066
## F-statistic:   119 on 1 and 48 DF,  p-value: 1.354e-14
```

Gráfica del modelo lineal

```r
plot(cars$speed, cars$dist2, main = "Regresión lineal de dist2 vs
Velocidad",
     xlab = "Velocidad", ylab = "dist2", pch = 19, col = "blue")

abline(modelo_transformado, col = "red", lwd = 2)
```




```r
par(mfrow = c(2, 2))
plot(modelo_transformado)
```




```r
shapiro.test(residuals(modelo_transformado))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(modelo_transformado)
## W = 0.97668, p-value = 0.4218
```

Como el valor p es mayor que 0.05, no se rechaza la hipótesis nula de que los residuos siguen una distribución normal. Esto sugiere que los residuos del modelo lineal transformado son aproximadamente normales, lo cual es un buen indicio de la validez del modelo.

```r
library(lmtest)

bptest(modelo_transformado)
```

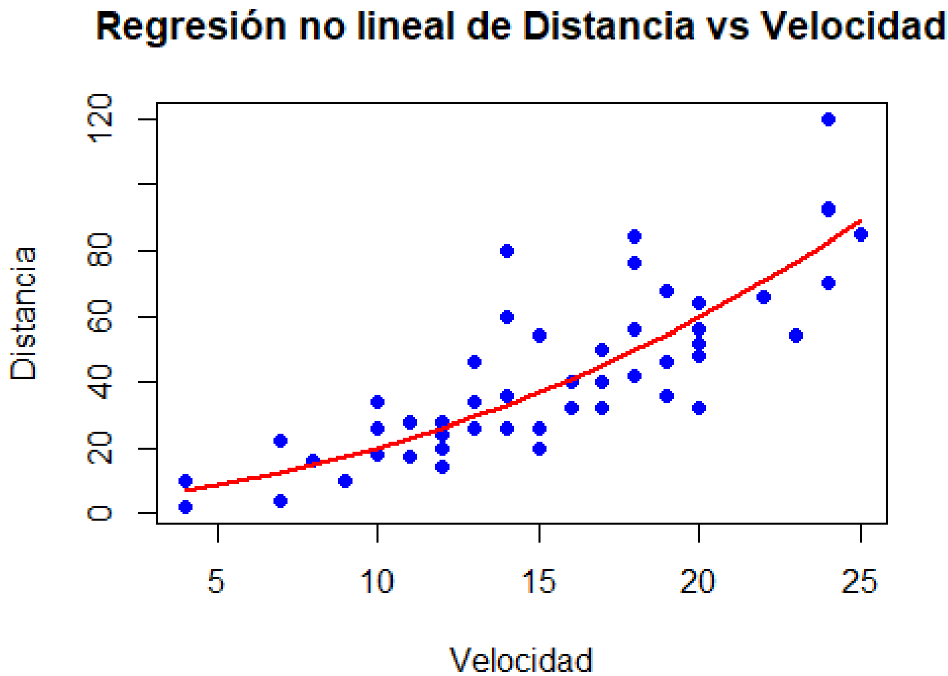
```
##
##  studentized Breusch-Pagan test
##
## data:  modelo_transformado
## BP = 0.13933, df = 1, p-value = 0.709
```

El valor p es mucho mayor que 0.05, por lo que no se rechaza la hipótesis nula de homocedasticidad, es decir, no hay evidencia de que los residuos presenten varianza no constante. Esto indica que el modelo cumple con el supuesto de homocedasticidad.

```r
library(car)

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

## Loading required package: carData

durbinWatsonTest(modelo_transformado)

##  lag Autocorrelation D-W Statistic p-value
##    1    0.0005469587      1.960621   0.786
##  Alternative hypothesis: rho != 0
```

El valor de Durbin-Watson está muy cercano a 2, y el valor p es mayor que 0.05, por lo que no se rechaza la hipótesis nula de que los residuos son independientes. Esto indica que no hay evidencia de autocorrelación en los residuos.

```r
predicciones_transformadas <- predict(modelo_transformado, newdata =
cars)

lambda <- lambda

predicciones_no_lineales <- (lambda * predicciones_transformadas +
1)^(1/lambda)

plot(cars$speed, cars$dist, main = "Regresión no lineal de Distancia vs
Velocidad",
     xlab = "Velocidad", ylab = "Distancia", pch = 19, col = "blue")

lines(cars$speed, predicciones_no_lineales, col = "red", lwd = 2)
```



El gráfico de la Regresión no lineal de Distancia vs Velocidad muestra una curva que ajusta bien los datos originales. El comportamiento no lineal es adecuado para describir la relación entre la distancia y la velocidad, con una clara tendencia creciente a medida que la velocidad aumenta.

Conclusión

El modelo no lineal es el mejor modelo para describir la relación entre la distancia y la velocidad, ya que: Ajusta mejor la relación subyacente en los datos originales sin necesidad de transformaciones. Refleja la tendencia no lineal esperada en este tipo de problema (distancia de frenado vs velocidad). Aunque el modelo lineal sobre los datos transformados es válido, el modelo no lineal es más representativo y adecuado para la interpretación directa en el contexto de los datos originales.