Regresion no lineal

2024-09-10

```r
library(MASS)
library(nortest)
library(e1071)
library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

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

```r
M = cars
head(M)
```

```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

```r
shapiro.test(M$speed)
```

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

```r
shapiro.test(M$dist)
```

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

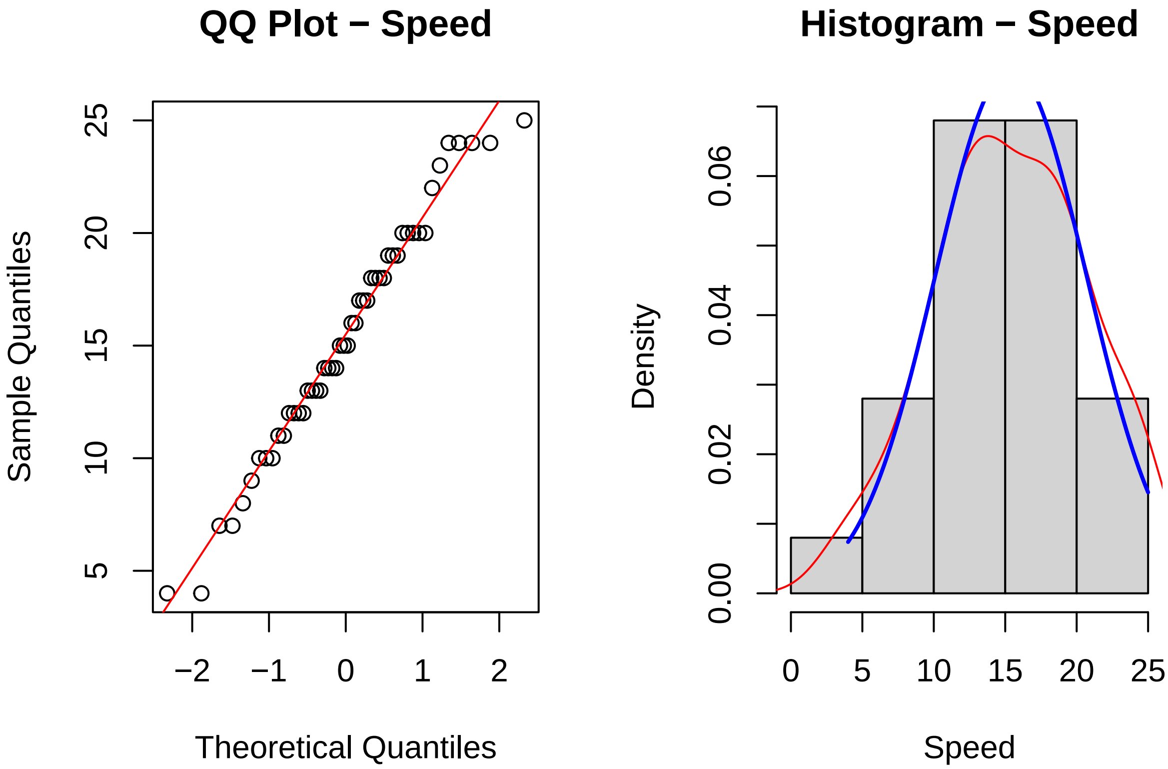
```r
ad.test(M$speed)
```

```
##
##  Anderson-Darling normality test
##
## data:  M$speed
## A = 0.26143, p-value = 0.6927
```

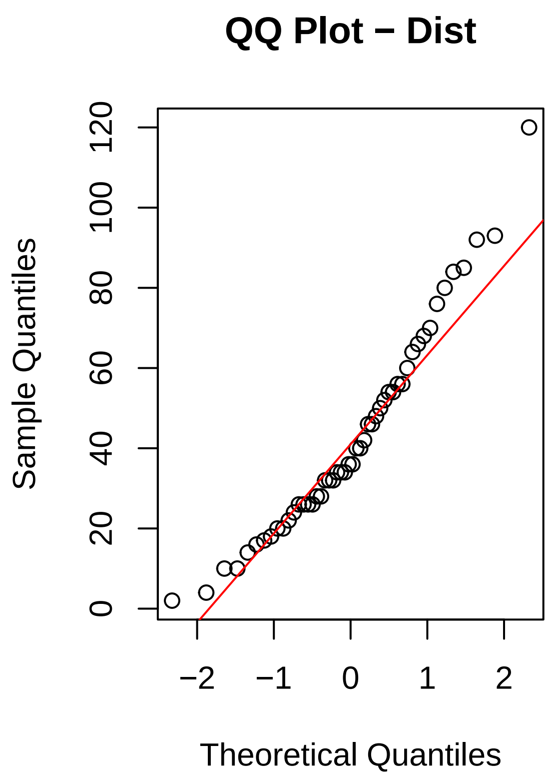
```r
ad.test(M$dist)
```

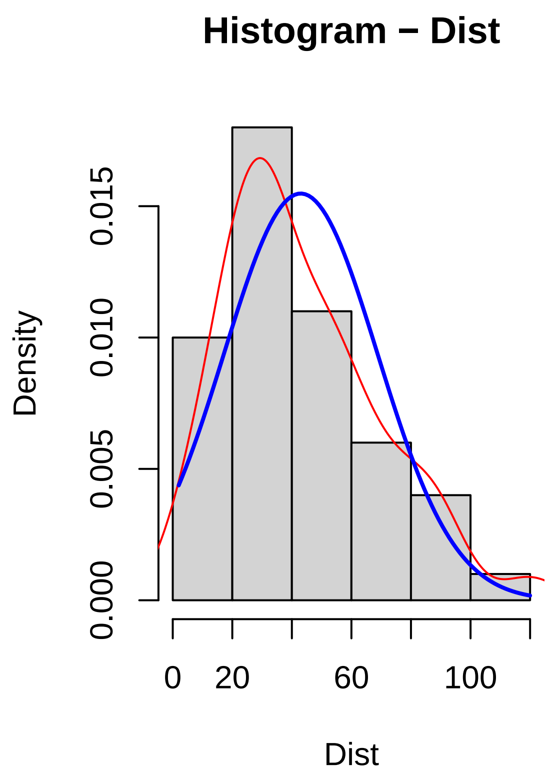
```
##
##  Anderson-Darling normality test
##
## data:  M$dist
## A = 0.74067, p-value = 0.05021
```

```r
par(mfrow = c(1, 2))

qqnorm(M$speed, main = "QQ Plot - Speed")
qqline(M$speed, col = "red")

hist(M$speed, freq = FALSE, main = "Histogram - Speed", xlab = "Speed", ylab = "Density", col = "lightg
lines(density(M$speed), col = "red")
curve(dnorm(x, mean = mean(M$speed), sd = sd(M$speed)), from = min(M$speed), to = max(M$speed), add = T
```




```r
par(mfrow = c(1, 2))

qqnorm(M$dist, main = "QQ Plot - Dist")
qqline(M$dist, col = "red")

hist(M$dist, freq = FALSE, main = "Histogram - Dist", xlab = "Dist", ylab = "Density", col = "lightgray
lines(density(M$dist), col = "red")
curve(dnorm(x, mean = mean(M$dist), sd = sd(M$dist)), from = min(M$dist), to = max(M$dist), add = TRUE,
```






```
skew_speed <- skewness(M$speed)
skew_dist <- skewness(M$dist)

kurtosis_speed <- kurtosis(M$speed)
kurtosis_dist <- kurtosis(M$dist)

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

```
## Sesgo de Speed: -0.1105533
```

```
cat("Curtosis de Speed:", kurtosis_speed, "\n")
```

```
## Curtosis de Speed: -0.6730924
```

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

```
## Sesgo de Dist: 0.7591268
```

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

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

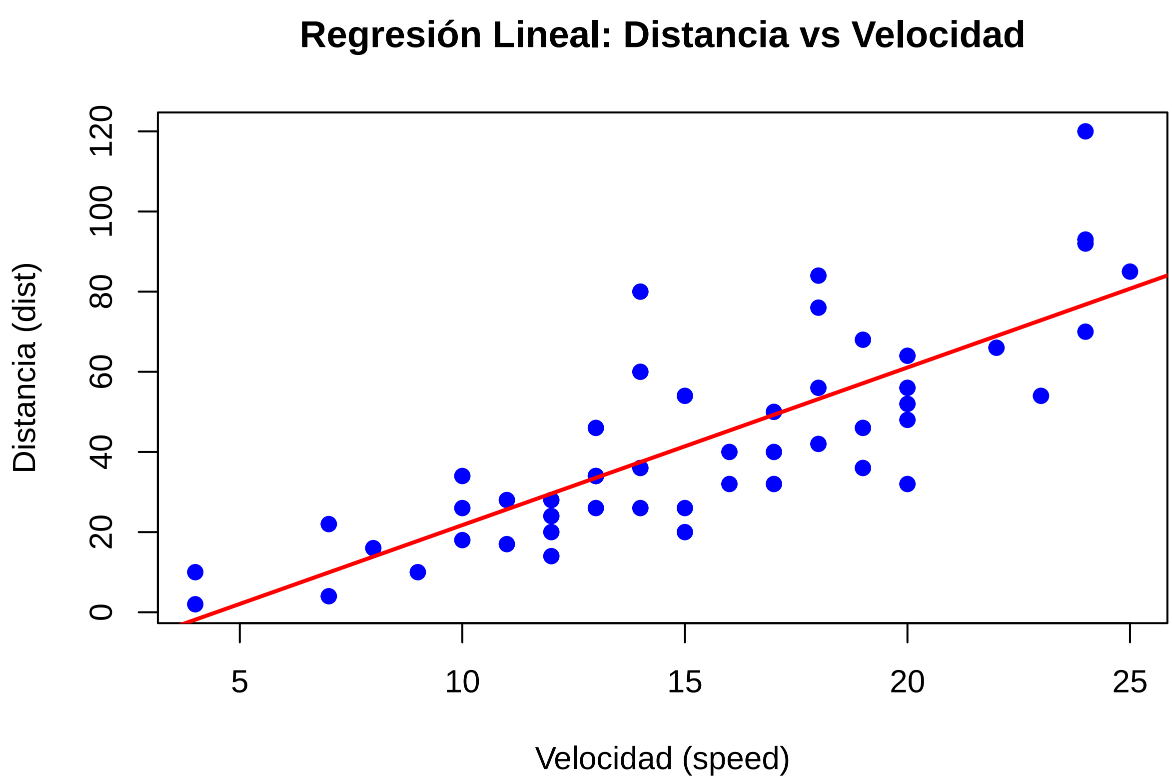
Como podemos ver speed sigue una distribuciin aproximadamente normal, tiene ligeras desviaciones pero con lsa graficas podemos asumir normalidad, mientras tanto la variable dist muestra un alejamiento mas claro de la normalidad.

Este alejamiento se puede dar por muchas cosas, puede ser el tamaño de los valores o tambien de factores que no fueron capturados en el conjunto de los datos, ya que en el contexto del problema, la variable distancia puede ser afectada de diferente manera.

```
modelo <- lm(dist ~ speed, data = M)

summary(modelo)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = M)
##
## 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 *
## 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
```

```r
coeficientes <- coef(modelo)
slope <- coeficientes["speed"]
intercept <- coeficientes["(Intercept)"]

plot(M$speed, M$dist, main = "Regresión Lineal: Distancia vs Velocidad",
     xlab = "Velocidad (speed)", ylab = "Distancia (dist)", pch = 19, col = "blue")

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

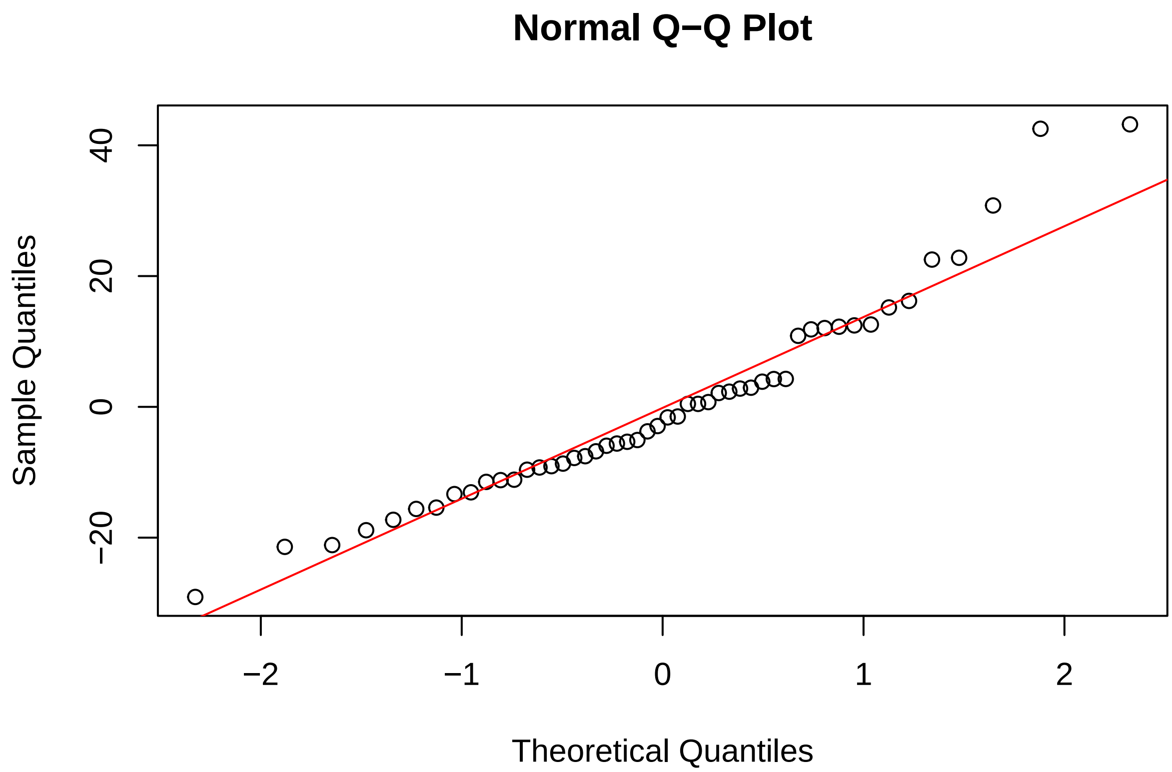



```r
residuos <- residuals(modelo)
```

```r
cat("Media de los residuos:", mean(residuos), "\n")
```

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

```r
qqnorm(residuos)
qqline(residuos, col = "red")
```



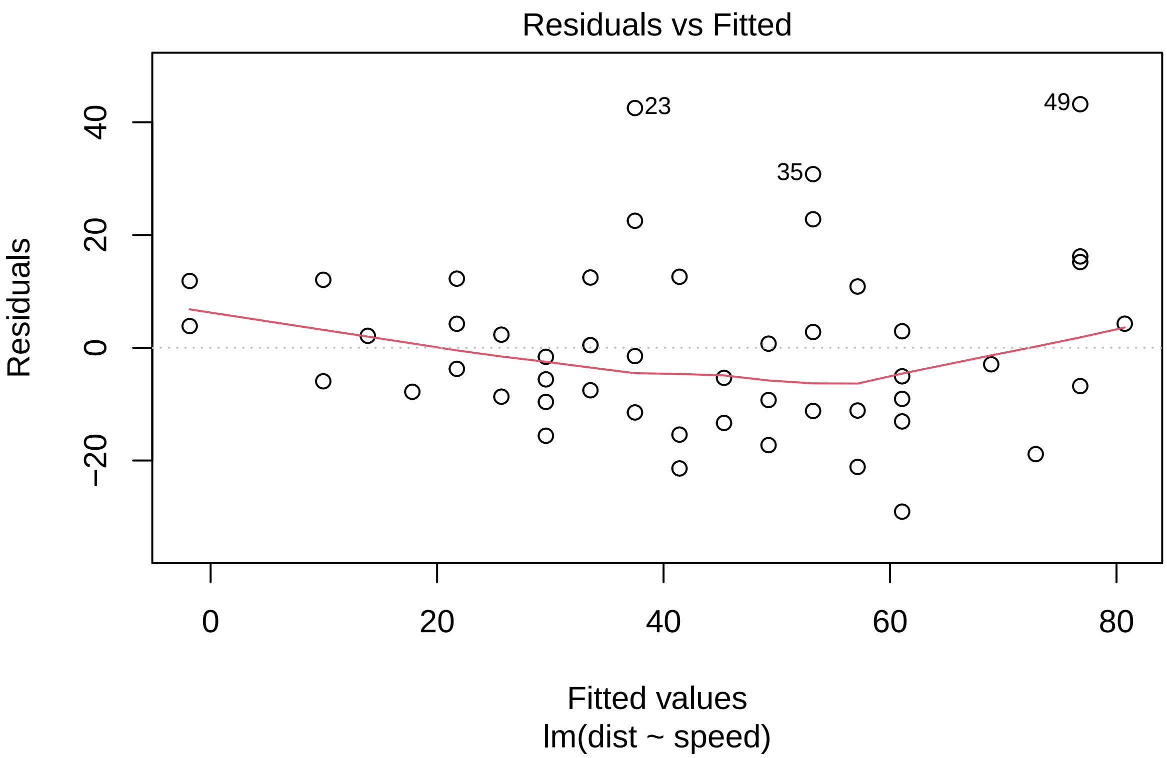

```r
shapiro.test(residuos)
```

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

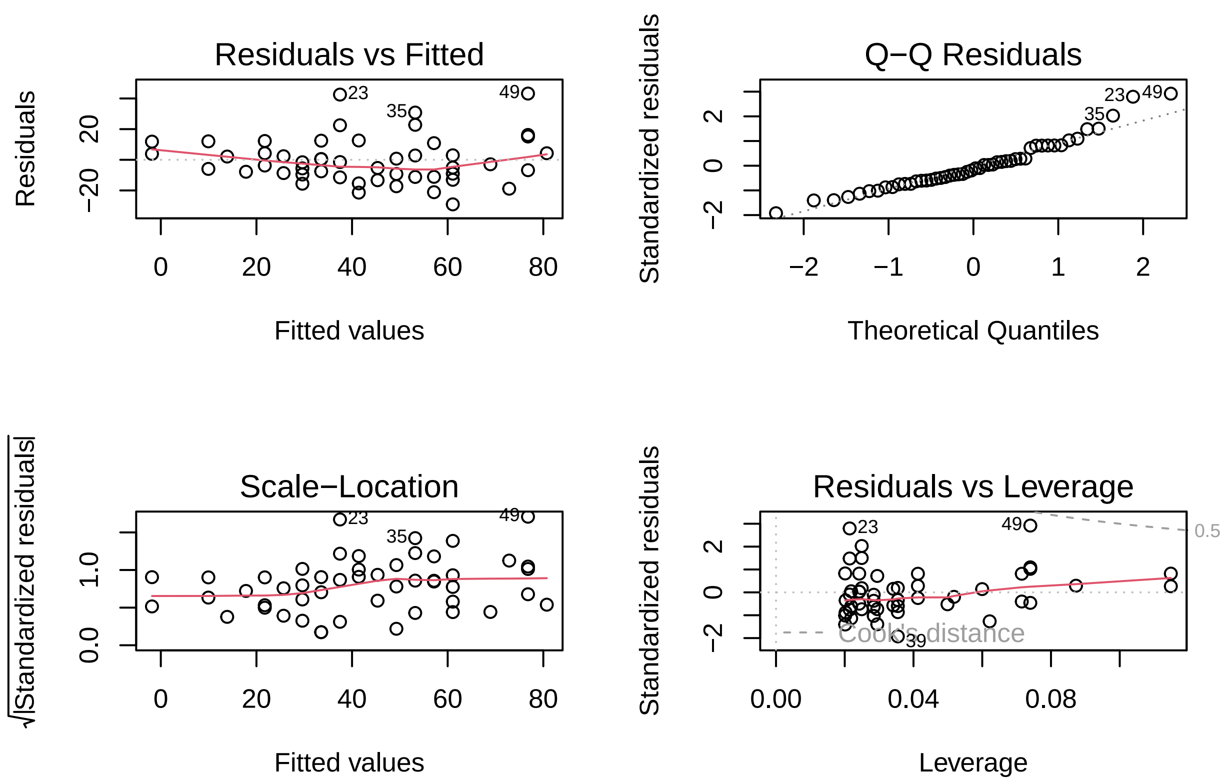
```r
dwtest(modelo)
```

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

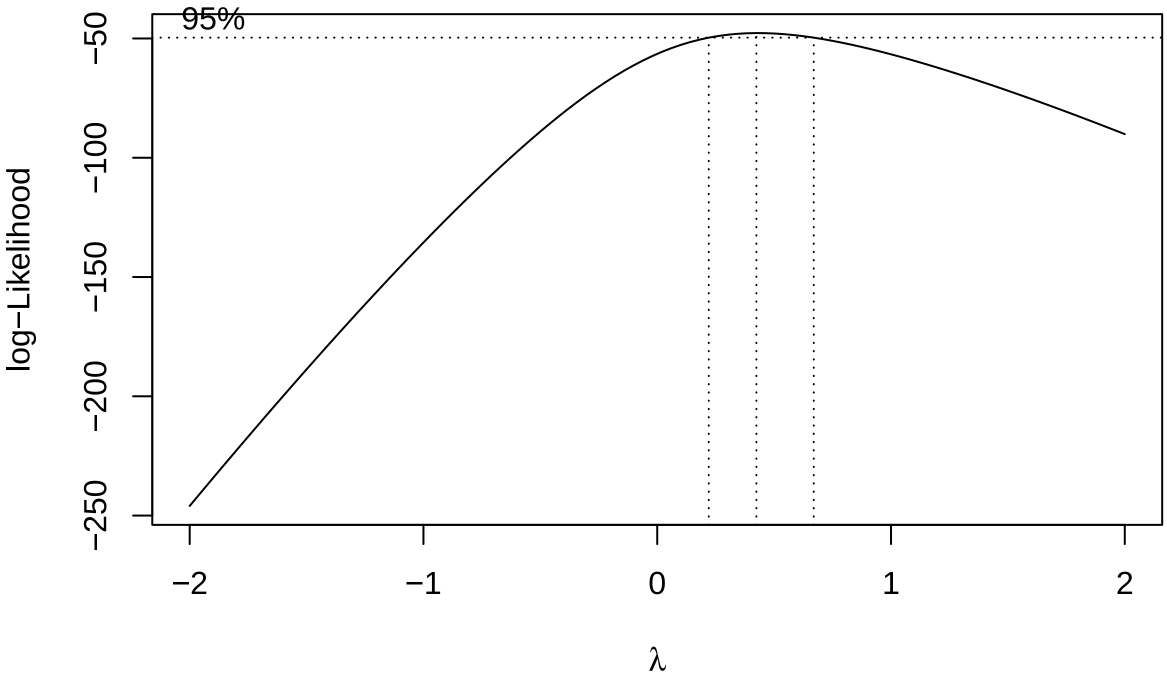
```r
plot(modelo, which = 1)

plot(modelo, which = 1)
```




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



```r
boxcox_result <- boxcox(modelo)
```




```r
lambda_optimo <- boxcox_result$x[which.max(boxcox_result$y)]
cat("El valor óptimo de lambda para la transformación Box-Cox es:", lambda_optimo, "\n")
```

```
## El valor óptimo de lambda para la transformación Box-Cox es: 0.4242424
```

$$Y(\lambda) = \begin{cases} \frac{Y^\lambda - 1}{\lambda}, & \text{si } \lambda \neq 0 \\ \ln(Y), & \text{si } \lambda = 0 \end{cases}$$

```r
M$dist_boxcox_exact <- (M$dist^lambda_optimo - 1) / lambda_optimo
M$dist_boxcox_approx <- log(M$dist)

cat("Medidas originales:\n")
```

```
## Medidas originales:
```

```r
cat("Sesgo:", skewness(M$dist), "Curtosis:", kurtosis(M$dist), "\n")
```

```
## Sesgo: 0.7591268 Curtosis: 0.1193971
```

```r
cat("Transformación exacta (Box-Cox):\n")
```

```
## Transformación exacta (Box-Cox):
```

```r
cat("Sesgo:", skewness(M$dist_boxcox_exact), "Curtosis:", kurtosis(M$dist_boxcox_exact), "\n")
```

```
## Sesgo: -0.1701619 Curtosis: -0.186884
```

```r
cat("Transformación aproximada (Log):\n")
```

```
## Transformación aproximada (Log):
```

```r
cat("Sesgo:", skewness(M$dist_boxcox_approx), "Curtosis:", kurtosis(M$dist_boxcox_approx), "\n")
```

```
## Sesgo: -1.302538 Curtosis: 2.543008
```

```r
shapiro.test(M$dist)
```

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

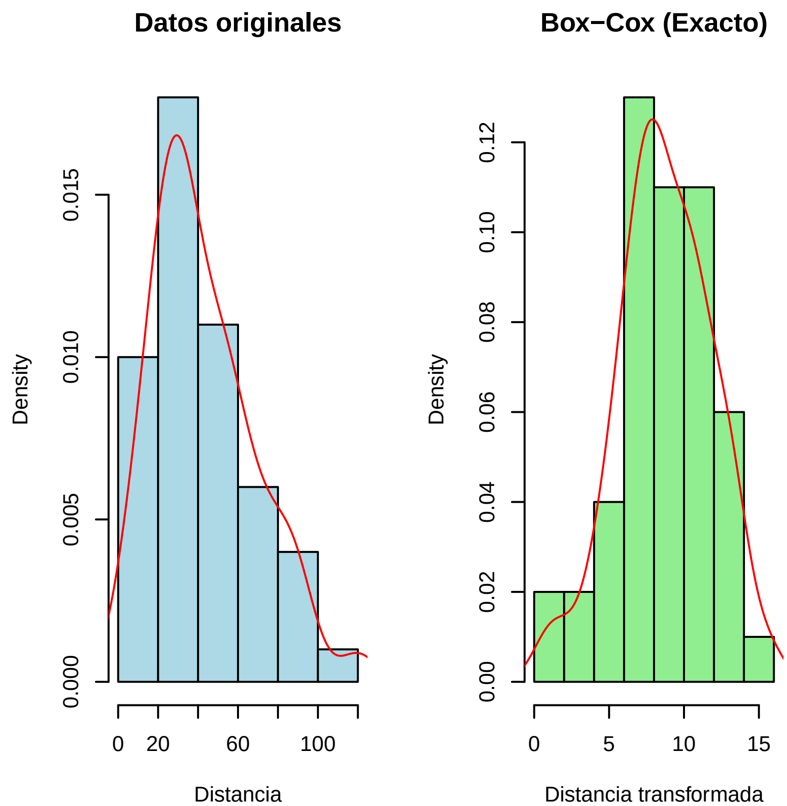
```r
shapiro.test(M$dist_boxcox_exact)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  M$dist_boxcox_exact
## W = 0.99168, p-value = 0.9773
```

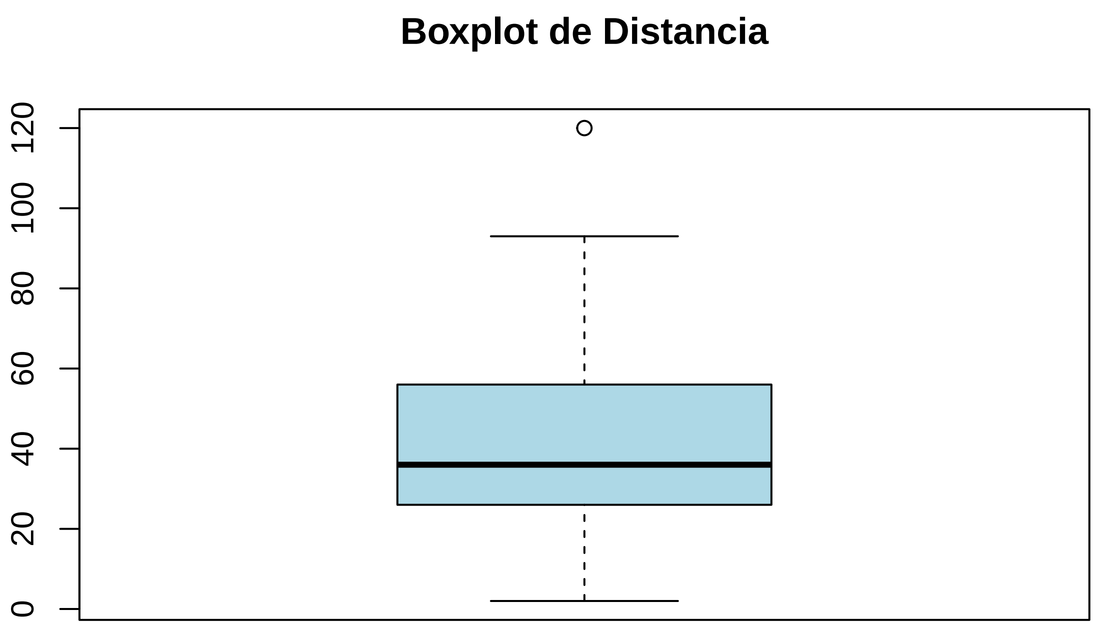
```r
shapiro.test(M$dist_boxcox_approx)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  M$dist_boxcox_approx
## W = 0.91024, p-value = 0.001066
```

```r
par(mfrow = c(1, 3))
hist(M$dist, main = "Datos originales", xlab = "Distancia", col = "lightblue", freq = FALSE)
lines(density(M$dist), col = "red")

hist(M$dist_boxcox_exact, main = "Box-Cox (Exacto)", xlab = "Distancia transformada", col = "lightgreen
lines(density(M$dist_boxcox_exact), col = "red")

hist(M$dist_boxcox_approx, main = "Log (Aprox)", xlab = "Distancia transformada", col = "lightcoral", f
lines(density(M$dist_boxcox_approx), col = "red")
```

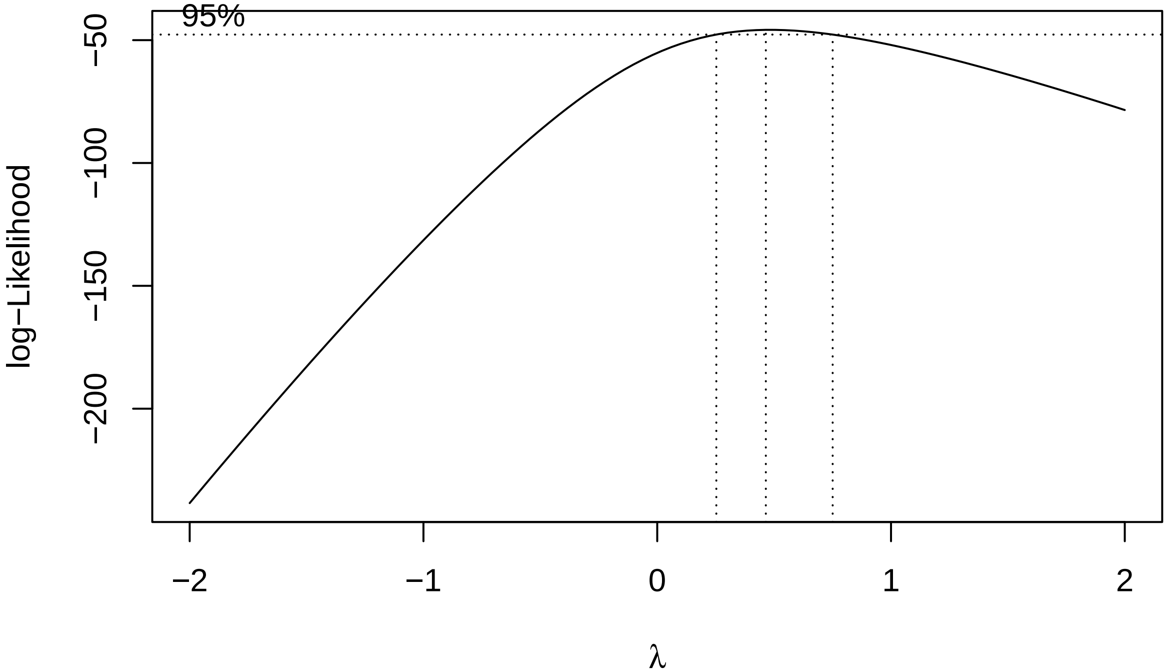



```r
par(mfrow = c(1, 1))
boxplot(M$dist, main = "Boxplot de Distancia", col = "lightblue")
```

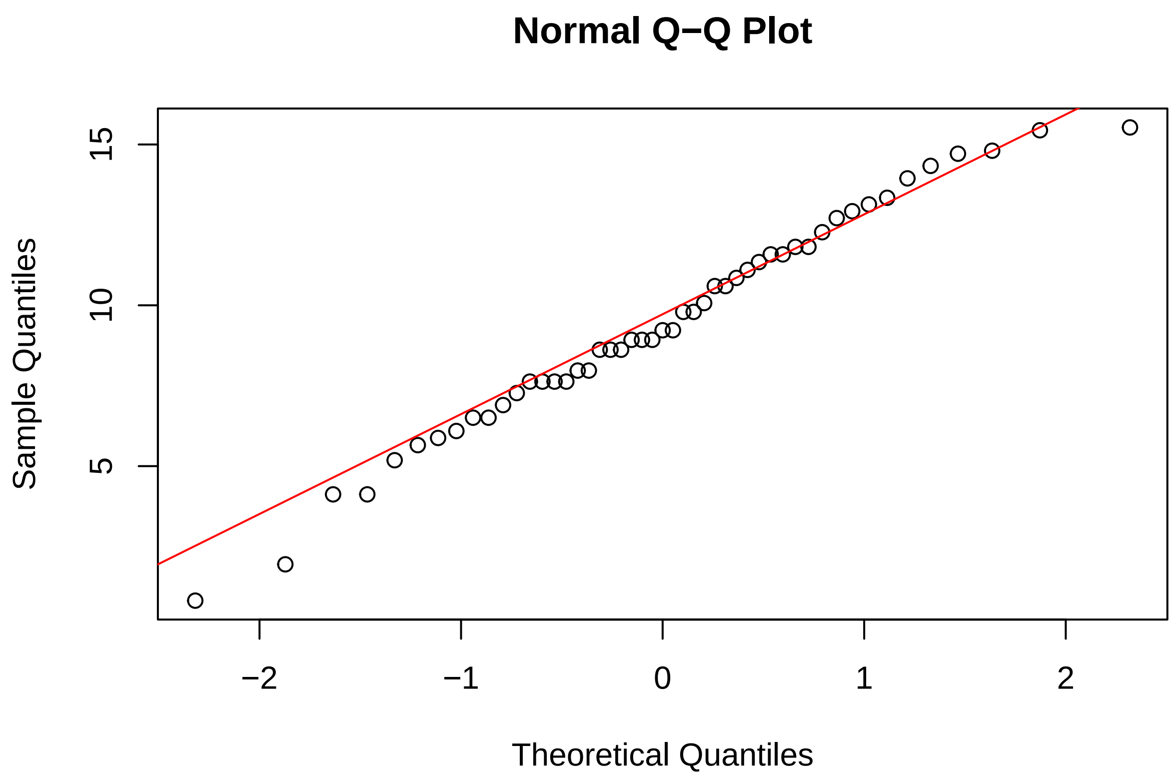



```r
outliers <- boxplot.stats(M$dist)$out

M_clean <- M[!M$dist %in% outliers, ]

M_clean <- M_clean[M_clean$dist != 0, ]
```

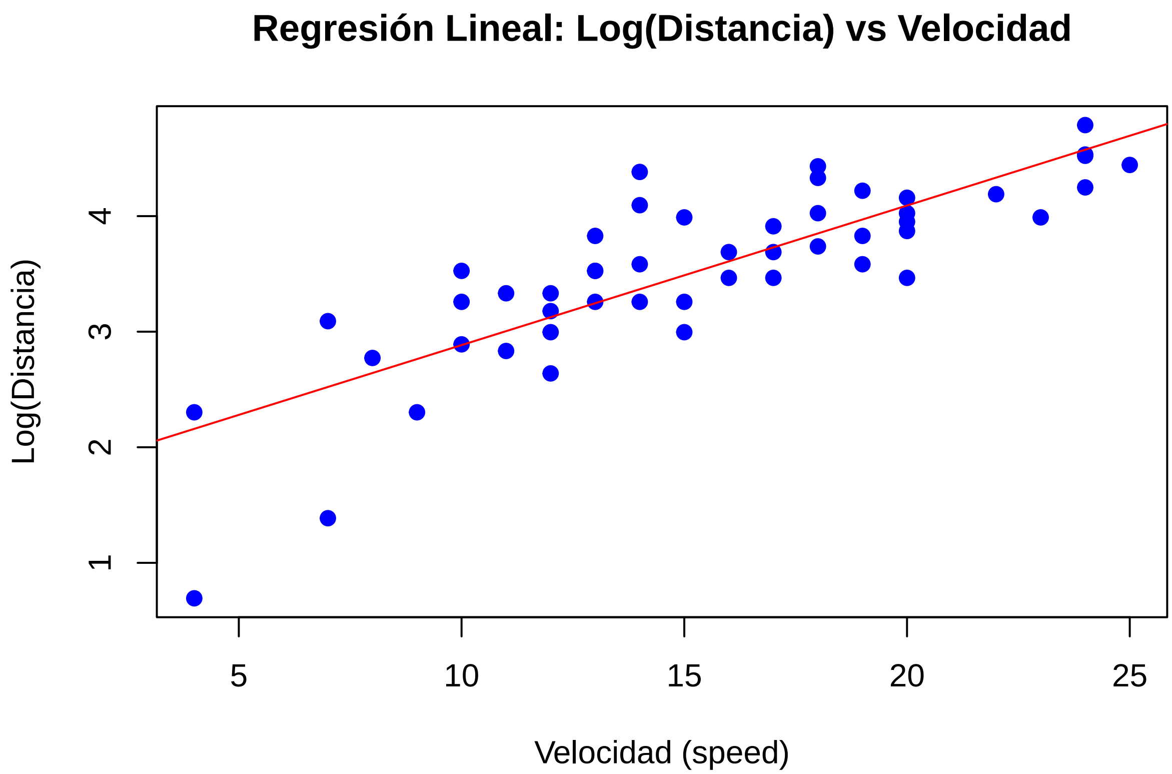
```r
modelo_clean <- lm(dist ~ speed, data = M_clean)

boxcox_result_clean <- boxcox(modelo_clean)
```



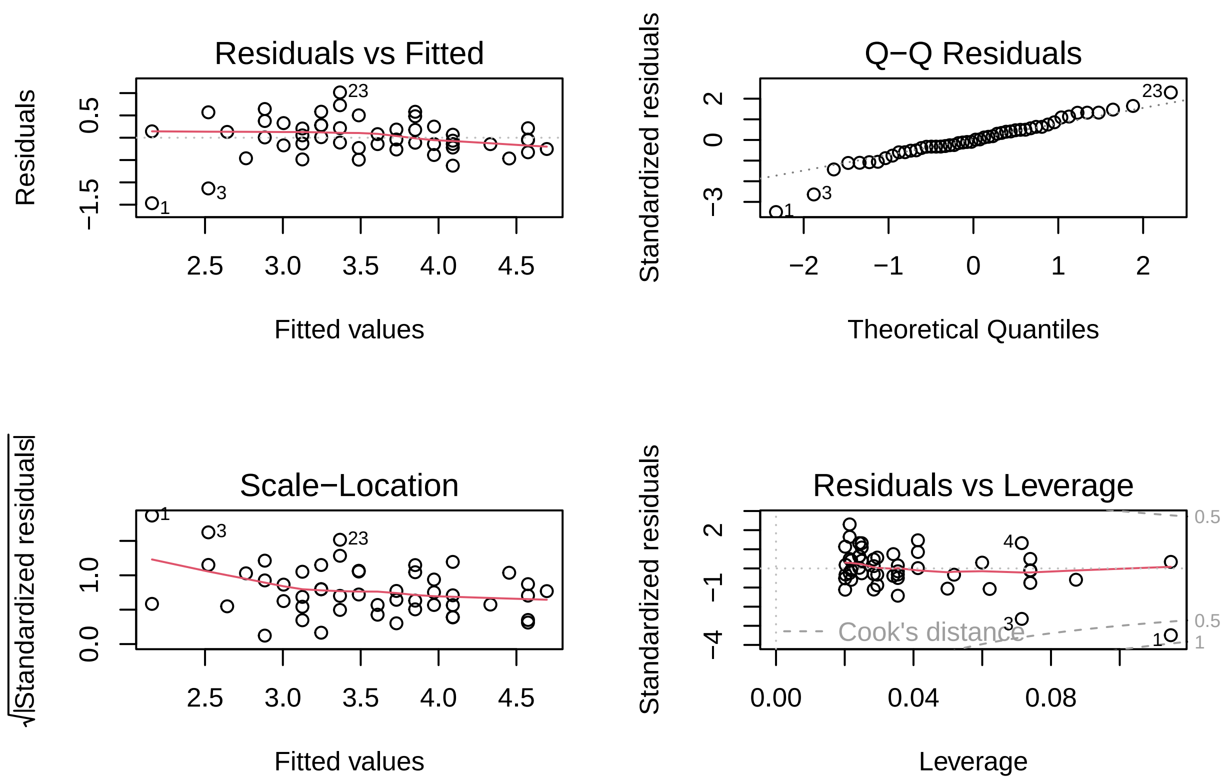

```r
lambda_optimo_clean <- boxcox_result_clean$x[which.max(boxcox_result_clean$y)]

M_clean$dist_boxcox_exact <- (M_clean$dist^lambda_optimo_clean - 1) / lambda_optimo_clean

qqnorm(M_clean$dist_boxcox_exact)
qqline(M_clean$dist_boxcox_exact, col = "red")
```




```r
M$log_dist <- log(M$dist)

modelo2 <- lm(log_dist ~ speed, data = M)

plot(M$speed, M$log_dist, main = "Regresión Lineal: Log(Distancia) vs Velocidad",
     xlab = "Velocidad (speed)", ylab = "Log(Distancia)", pch = 19, col = "blue")
abline(modelo2, col = "red")
```




```
par(mfrow = c(2, 2))
plot(modelo2)
```



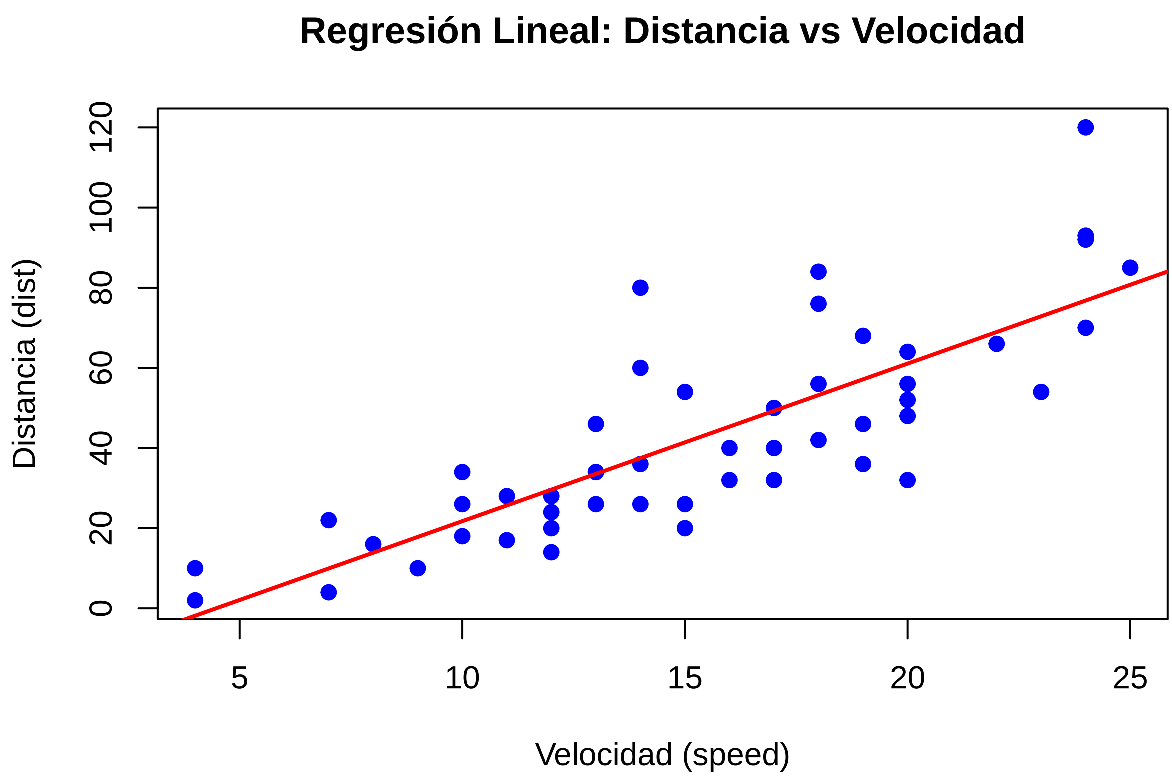
```r
shapiro.test(residuals(modelo2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(modelo2)
## W = 0.95734, p-value = 0.06879
```

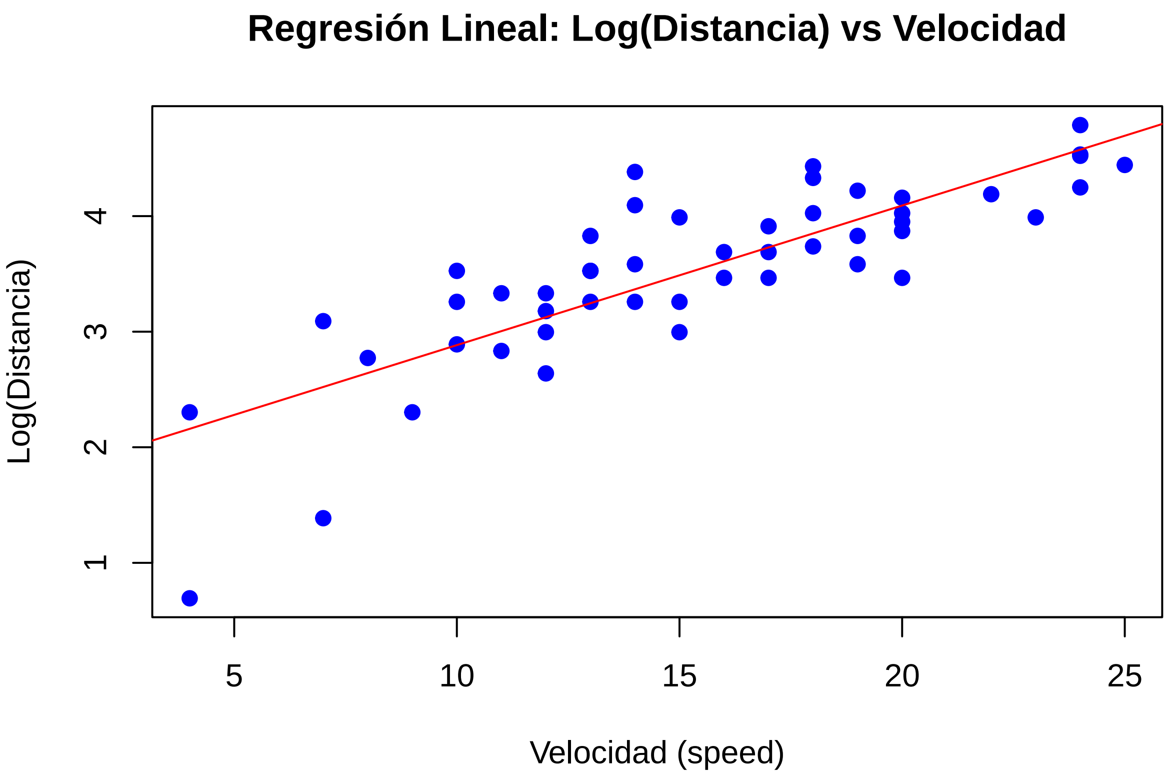
```r
dwtest(modelo2)
```

```
##
##  Durbin-Watson test
##
## data:  modelo2
## DW = 1.8247, p-value = 0.2194
## alternative hypothesis: true autocorrelation is greater than 0
```

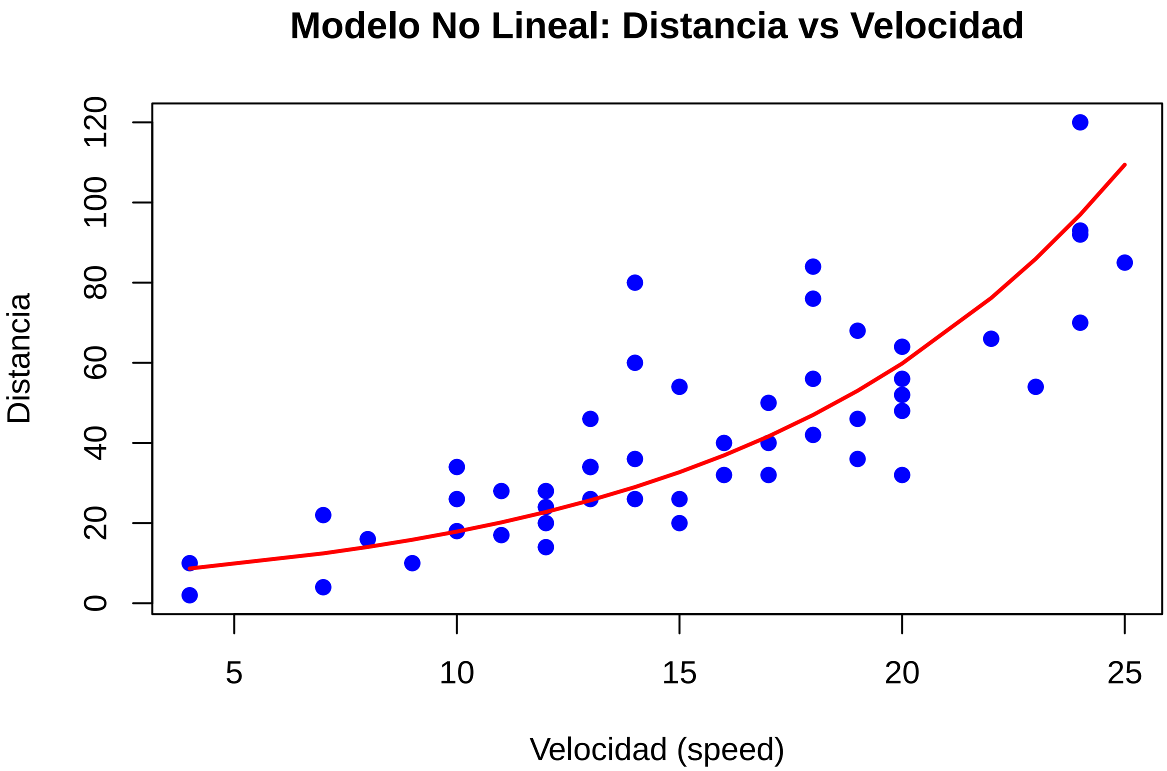
```r
plot(M$speed, M$dist, main = "Regresión Lineal: Distancia vs Velocidad",
     xlab = "Velocidad (speed)", ylab = "Distancia (dist)", pch = 19, col = "blue")

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




```r
plot(M$speed, M$log_dist, main = "Regresión Lineal: Log(Distancia) vs Velocidad",
     xlab = "Velocidad (speed)", ylab = "Log(Distancia)", pch = 19, col = "blue")
abline(modelo2, col = "red")
```




```r
plot(M$speed, M$dist, main = "Modelo No Lineal: Distancia vs Velocidad",
     xlab = "Velocidad (speed)", ylab = "Distancia", pch = 19, col = "blue")

predicted_dist <- exp(predict(modelo2))

lines(M$speed, predicted_dist, col = "red", lwd = 2)
```



El modelo 2 es mejor, ya que este proporciona un mejor ajuste y mejora la validez del modelo en terminos de normalidad de los residuos y homocedasticidad.