Regresión No lineal

Jacobo Hirsch Rodriguez

2024-09-12

Vamos a cargar el Dasated preinstalado en r

```r
# Cargar los datos
data(cars)
head(cars)
```

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

Este dataset cars en R contiene datos de velocidad y distancia de frenado de autos de los años 1920.

#Normalidad de los datos

vamos a realizar las pruebas de normalidad univariada de la velocidad y distanci, pero antes vamos a escribir la hipotesis nula, la alternativa y la regla de decisión para la interpretación de los resultados:

h0: los datos se distribuyen de forma normal

h1: los datos no se distribuyen de forma normal

regla de decisión: si p < 0.05 rechazamos la hipotesis nula

ahora, primero haremos la prueba de shapiro-wilk:

```r
# Prueba de normalidad Shapiro-Wilk para la variable velocidad
shapiro.test(cars$speed)
```

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

```r
# Prueba de normalidad Shapiro-Wilk para la variable distancia
shapiro.test(cars$dist)
```

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

la segunda prueba que haremos es la prueba de Anderson-darling:

```r
library(nortest)

# Prueba de Anderson-Darling para la variable velocidad
ad.test(cars$speed)
```

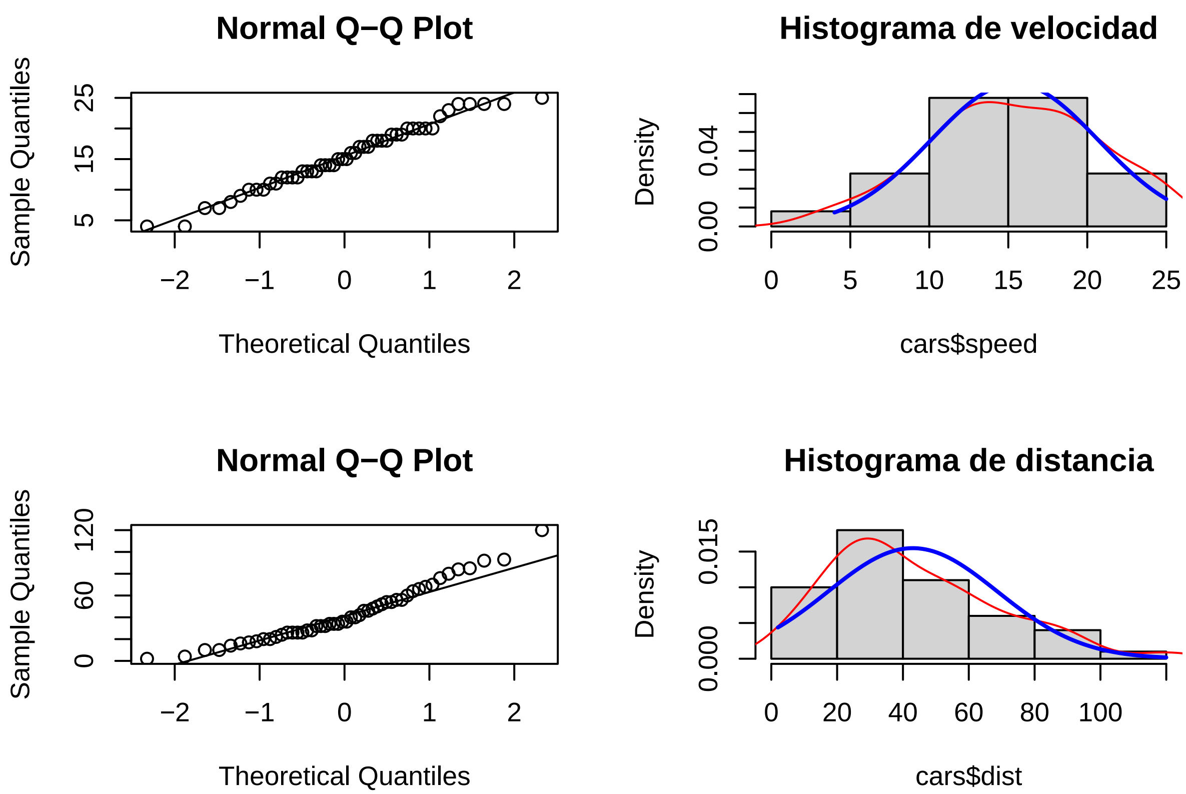
```
##
##  Anderson-Darling normality test
##
## data:  cars$speed
## A = 0.26143, p-value = 0.6927
```

```r
# Prueba de Anderson-Darling para la variable distancia
ad.test(cars$dist)
```

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

Parece ser de acuerdo a ambas pruebas que los datos de Velocidad se distribuyen de forma normal, ya que el valor p resultante de ambas pruebas es bastante superior a 0.05, caso contrario la variable distancia que en la primera prueba el valor es inferior y en la segunda prueba es insigificante la diferencia, especialmente por que la diferencia de la primera prueba es más grande por lo que podria decirse que la variable speed se distribuye normalmente y la variable dist no. Pero vamos a visualizarlas para reforzar las conclusiones.

```r
# Establecemos una disposición para tener dos gráficos por fila
par(mfrow=c(2,2))

# Gráfico para la variable 'speed'
# QQ-Plot para la variable velocidad
qqnorm(cars$speed)
qqline(cars$speed)

# Histograma con la densidad y la curva normal teórica
hist(cars$speed, freq=FALSE, main="Histograma de velocidad")
lines(density(cars$speed), col="red")
curve(dnorm(x, mean=mean(cars$speed), sd=sd(cars$speed)),
      from=min(cars$speed), to=max(cars$speed), add=TRUE, col="blue", lwd=2)

# Gráfico para la variable 'dist'
# QQ-Plot para la variable distancia
qqnorm(cars$dist)
qqline(cars$dist)

# Histograma con la densidad y la curva normal teórica
hist(cars$dist, freq=FALSE, main="Histograma de distancia")
lines(density(cars$dist), col="red")
curve(dnorm(x, mean=mean(cars$dist), sd=sd(cars$dist)),
      from=min(cars$dist), to=max(cars$dist), add=TRUE, col="blue", lwd=2)
```



Podemos confirmar las observaciones anteriores, aunque para la velocidad los datos no son perfectamente normales, parecen ajustarse bien a una distribución normal. El problema de ambas variables parece que esta en los extramos, para la variable distancia parece ser que tiene un sesgo a la derecha, aunque su qqplot nos indica que una cantidad importante de datos se distribuye normalmente, se puede ver este sesgo positivo con los datos por encima de la linea normal.

Ahora que hemos visualizado nuestros datos, vamos a calcular el coeficiente de sesgo y de curtosis:

```r
# Instalar y cargar la librería e1071 si es necesario
# install.packages("e1071")
library(e1071)

# Calcular sesgo y curtosis para la variable velocidad
skewness_speed <- skewness(cars$speed)
kurtosis_speed <- kurtosis(cars$speed)

# Calcular sesgo y curtosis para la variable distancia
skewness_dist <- skewness(cars$dist)
kurtosis_dist <- kurtosis(cars$dist)

# Crear un dataframe con los resultados
resultados_sesgo_curtosis <- data.frame(
  Variable = c("Speed", "Speed", "Dist", "Dist"),
  Metrica = c("Skewness", "Kurtosis", "Skewness", "Kurtosis"),
  Valor = c(skewness_speed, kurtosis_speed, skewness_dist, kurtosis_dist)
)
```

```r
# Mostrar el dataframe
print(resultados_sesgo_curtosis)
```

```
##   Variable  Metrica      Valor
## 1    Speed Skewness -0.1105533
## 2    Speed Kurtosis -0.6730924
## 3     Dist Skewness  0.7591268
## 4     Dist Kurtosis  0.1193971
```

El sesgo negativo pero cercano de velocidad nos indica que tiene una asimetría leve esto significa que los datos de velocidad tienen una cola izquierda un poco más pesada que la derecha, a diferencia de dist que tiene una asimetría considerable a la derecha (positiva). En cambio velocidad tiene colas más ligeras que las de una distribución normal por lo que parece que tiene valores menos extremos en las colas, para dist el valor es cercano a 0, colas no son particularmente pesadas ni ligeras, lo que sugiere que los datos están distribuidos de manera similar a una normal en términos de colas.

#Regresion lineal

Prueba regresión lineal simple entre distancia y velocidad. Usa lm(y~x)

```r
# Realizar la regresión lineal entre distancia y velocidad
modelo_cars <- lm(dist ~ speed, data = cars)

# Mostrar el modelo lineal obtenido
summary(modelo_cars)
```

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

La ecuación obtenida es: dist = -17.58 + 3.93*speed

El coeficiente de la variable speed es 3.93, con un valor p extremadamente pequeño (1.49e-12). Esto indica que la velocidad tiene un efecto altamente significativo sobre la distancia. hablando sobre el r2 0.6511, nos dice que el 65.11% de la variabilidad en la distancia puede ser explicada por la velocidad, aunque esto únicamente podria significar que esto funciona para este conjunto de datos por lo que hace falta analizar los errores para verificar su significancia.

Analiza validez del modelo.

```r
residuos_cars <- residuals(modelo_cars)
```

Residuos con media cero

hipotesis nula : la media de los errores es igual a 0 hiporesis alrernativa : la media de los errores es diferente de 0 regla de decisión con prueba t : Si el valor p es menor que 0.05, se rechaza la hipótesis nula y se concluye que la media de los residuos es significativamente diferente de 0.

```r
prueba_t_cars <- t.test(residuos_cars, mu=0)
print(prueba_t_cars)
```

```
##
##  One Sample t-test
##
## data:  residuos_cars
## t = -2.0629e-16, df = 49, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -4.326  4.326
## sample estimates:
##     mean of x
## -4.440892e-16
```

se obtuvo el valor más alto posible para el p value del t-test, por lo que no hay evidencia suficiente para rechazar la hipotesis nula

Normalidad de los residuos

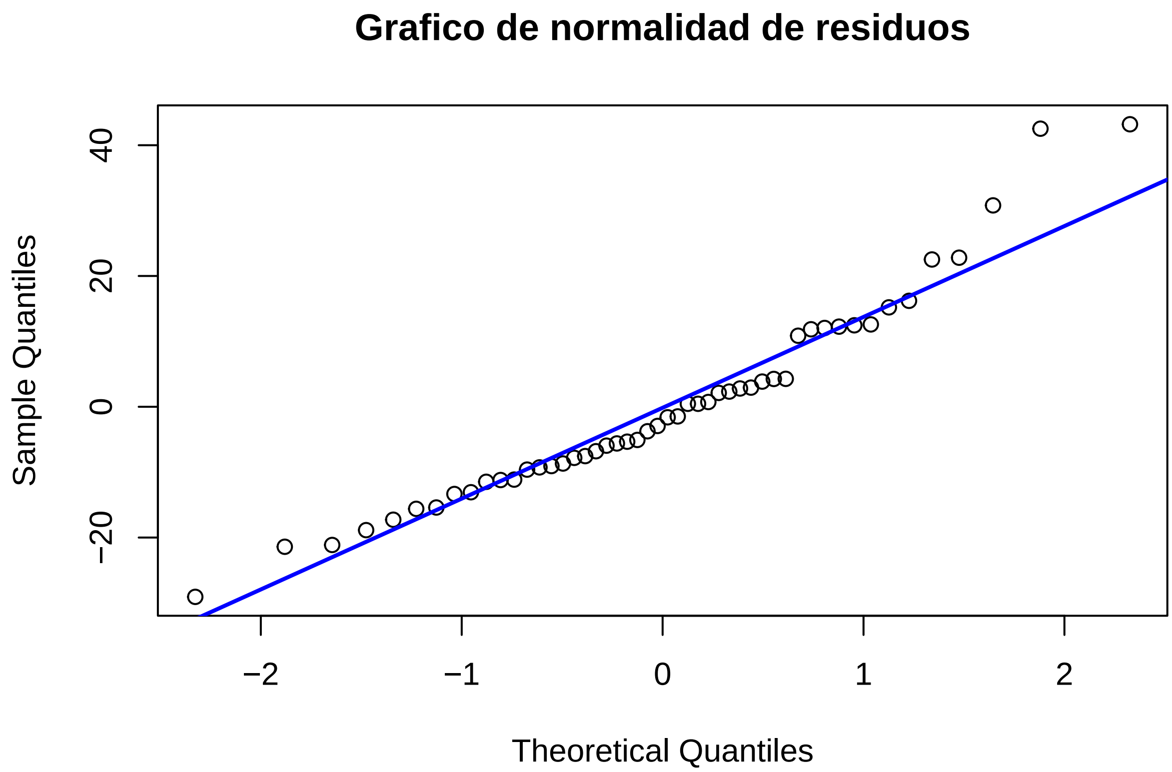
hipotesis nula : los residuos siguen una distribución normal. hipotesis alternativa : los residuos no siguen una distribución normal regla de decisión : Si el valor p es menor a 0.05, se rechaza la hipótesis nula y se concluye que los residuos no siguen una distribución normal

```r
# Prueba de Shapiro-Wilk
shapiro.test(residuos_cars)
```

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

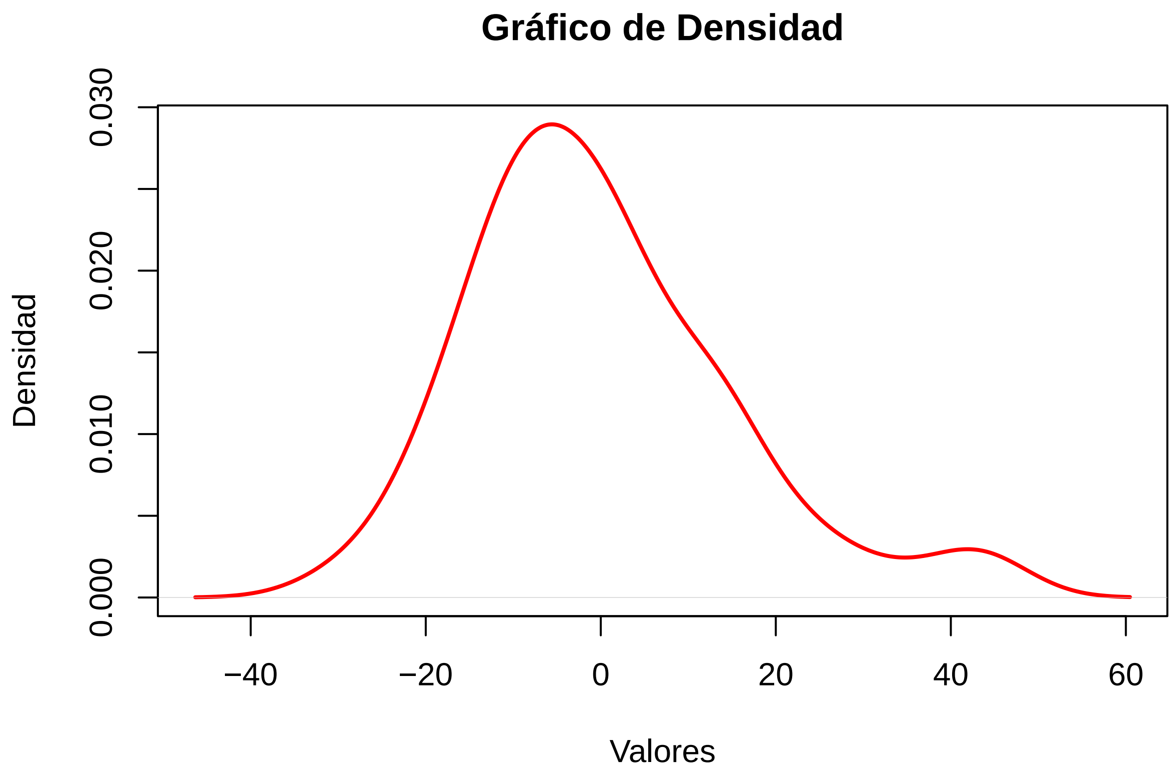
al parecer se rechaza la hipotesis nula, los residuos no parecen seguir una distribución normal, por lo que vamos a reafirmarlo graficando un qqplot y un grafico de densidad

```r
# QQ-Plot de los residuos
qqnorm(residuos_cars, main="Grafico de normalidad de residuos")
qqline(residuos_cars, col = "blue", lwd = 2)
```



ahora hacemos el grafico de densidad:

```
plot(density(residuos_cars),
col = "red",
lwd = 2,
main = "Gráfico de Densidad",
xlab = "Valores",
ylab = "Densidad")
```



con los siguientes gráficos confirmamos que los errores no se distribuyen de forma normal.

Homocedasticidad

hipotesis nula: la varianza de los errores es constante (hay homocedasticidad)

hipotesis alternativa : la varianza de los errores no es constante (hay heterocedasticidad)

regla de decisión: si el valor p es menor o igual que 0.05 entonces rechazamos la hipotesis nula para realizar las pruebas vamos a necesitar descargar un paquete que contiene la prueba que vamos a utilizar

```r
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

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

```r
prueba_white_cars <- bptest(modelo_cars, ~ fitted(modelo_cars) + I(fitted(modelo_cars)^2))
print(prueba_white_cars)
```

```
##
##  studentized Breusch-Pagan test
```

```
##
## data:  modelo_cars
## BP = 3.2157, df = 2, p-value = 0.2003
```

el valor p no es menor a el umbral seleccionado (0.05) por lo que no hay información suficiente para rechazar la hipotesis nula.

independencia

hipotesis nula: los errores no estan correlacionados

hipotesis alternativa : los errores estan correlacionados

regla de decisión: si el valor p de la prueba es menor a 0.05 se rechaza la hipotesis nula

hacemos el modelo durbin watson para verificar independencia entre los residuos

```r
# Realizar la prueba de Durbin-Watson
prueba_dw_cars <- dwtest(modelo_cars)

# Mostrar los resultados de la prueba
print(prueba_dw_cars)
```

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

la hipotesis nula no puede ser rechazada en este escenario tampoco ya que el p value obtenido es mayor $(0.09) > 0.05$

linealidad.

para verificar la linealidad vamos a utilizar una prueba RESET

La prueba RESET de Ramsey (Regression Equation Specification Error Test) es utilizada para detectar posibles errores de especificación en un modelo de regresión lineal. La prueba examina si hay variables omitidas o si la forma funcional del modelo es incorrecta.

hipotesis nula: no hay términos omitidos que indican linealidad

hipotesis alternativa: hay una especificación errónea en el modelo que indica no linealidad

regla de decisión: si el valor p de la prueba es menor a 0.05 se rechaza la hipotesis nula

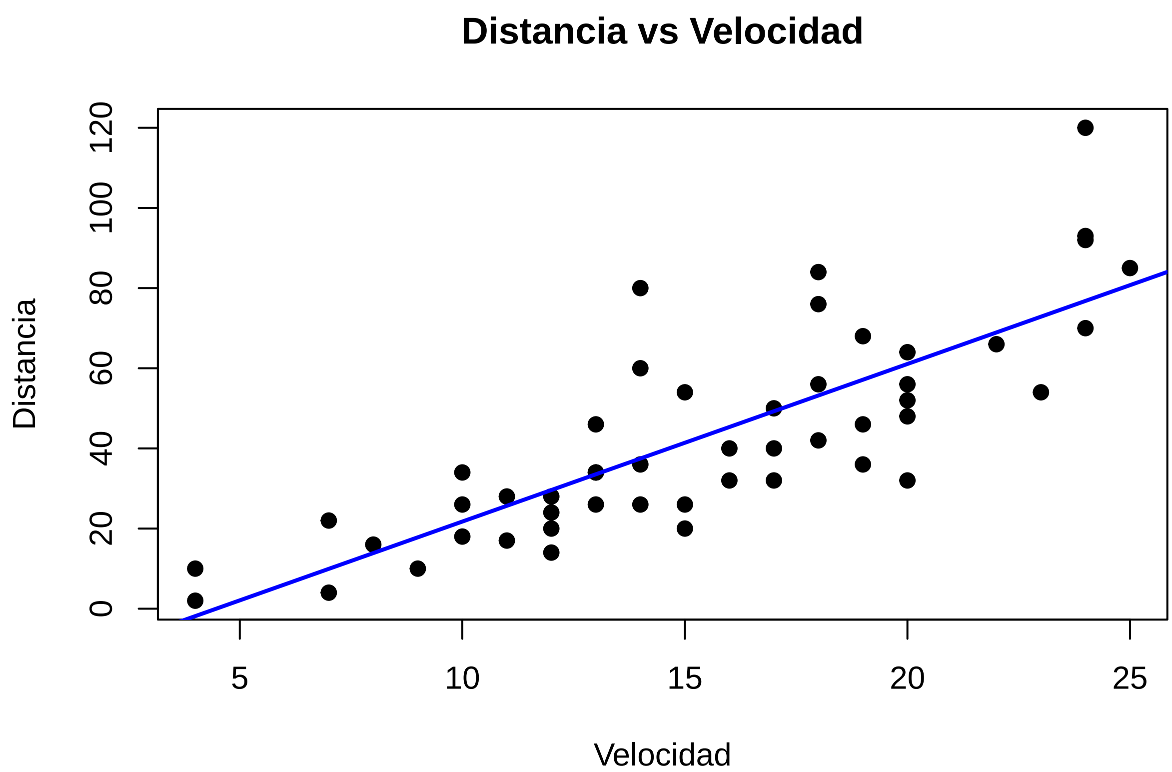
```r
# Realizar la prueba RESET de Ramsey
prueba_reset_cars <- resettest(modelo_cars)

# Mostrar los resultados de la prueba
print(prueba_reset_cars)
```

```
##
##  RESET test
##
## data:  modelo_cars
## RESET = 1.5554, df1 = 2, df2 = 46, p-value = 0.222
```

tampoco se rechaza la hipotesis nula, p-value 0.222 > 0.05

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

```r
# Graficar los datos y la línea del modelo
plot(cars$speed, cars$dist, main="Distancia vs Velocidad", xlab="Velocidad", ylab="Distancia", pch=19)
abline(modelo_cars, col="blue", lwd=2)  # Agregar la línea del modelo
```

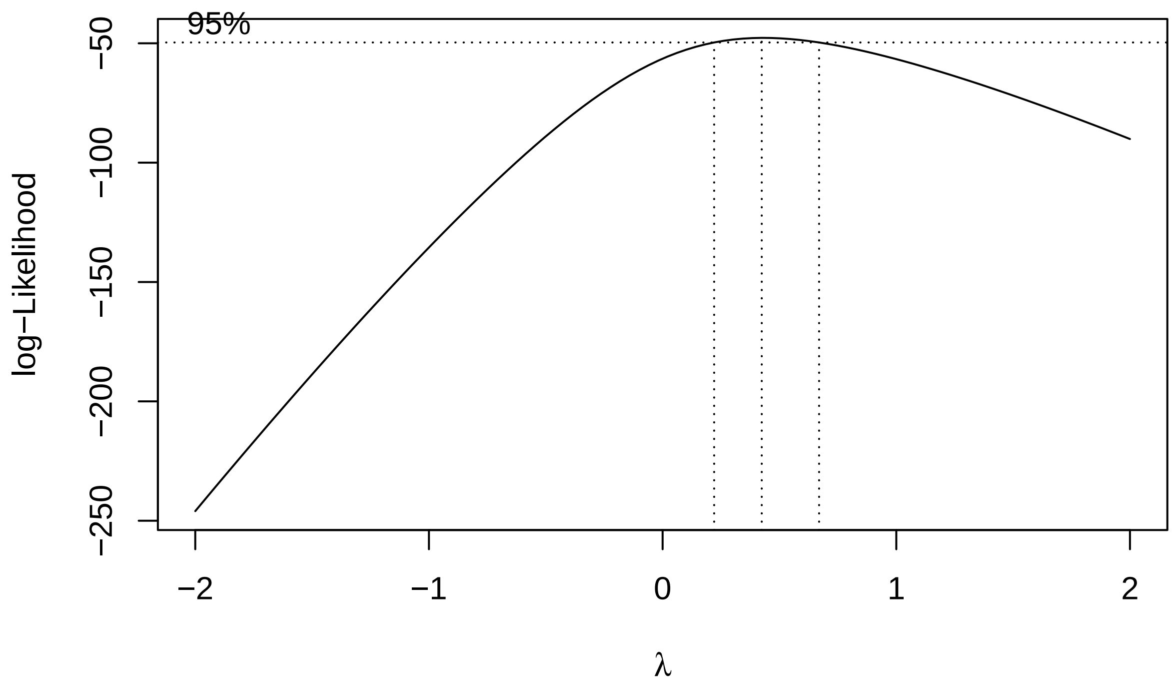


Conclusiones: el modelo parece funcionar en un buen porcentaje (63% de explicación) y la única prueba que no pasó fue la de la normalidad de los residuos. aún asi, aunque parezca (y asi sea) que exista una correlación fuerte entre la velocidad y la distancia con un estadístico F de 89.57 t con un valor p de 1.49e-12, es posible (tal vez9 QUE exista un modelo que nos peuda dar mejores resultados ya que los datos parecen estar muy dispersos a comparación de la linea que nos predice el modelo.

#Regresión no lineal

Con el objetivo de probar un modelo no lineal que explique la relación entre la distancia y la velocidad, haz una transformación con la base de datos car que te garantice normalidad en ambas variables, para este punto nos vamos a concentrar en la variable que tiene menos normalidad, en este caso es la variable dist: Encuentra el valor de en la transformación Box-Cox para el modelo lineal:

```r
library(MASS)

# Aplicar la transformación de Box-Cox en el modelo de regresión lineal
boxcox_model_cars <- boxcox(lm(dist ~ speed, data = cars))
```



vamos a encontrar el valor optimo

```
# Encontrar el valor óptimo de lambda (el valor con el log-likelihood más alto)
lambda_optimo_cars <- boxcox_model_cars$x[which.max(boxcox_model_cars$y)]
print(lambda_optimo_cars)
```

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

Ahora vamos a hacer la transformación exacta:

```
# Aplicar la transformación exacta de Box-Cox con lambda = 0.4242424

dist_transformada_exacta <- (cars$dist^lambda_optimo_cars - 1) / lambda_optimo_cars
```

Ahora la aproximada :

```
# Transformación aproximada (Raíz cuadrada)
dist_transformada_aproximada <- sqrt(cars$dist)
```

vamos revisar el sesgo y la curtosis de las pruebas

```
library(e1071)
# Sesgo y curtosis para comparar la normalidad de las transformaciones
# Original
skewness_original <- skewness(cars$dist)
```

```r
kurtosis_original <- kurtosis(cars$dist)

# Transformación exacta
skewness_exacta <- skewness(dist_transformada_exacta)
kurtosis_exacta <- kurtosis(dist_transformada_exacta)

# Transformación aproximada (Raíz cuadrada)
skewness_aproximada <- skewness(dist_transformada_aproximada)
kurtosis_aproximada <- kurtosis(dist_transformada_aproximada)

# Imprimir resultados de sesgo y curtosis
cat("Sesgo y Curtosis de las Transformaciones:\n")
```

```
## Sesgo y Curtosis de las Transformaciones:
```

```r
cat("Original: Sesgo =", skewness_original, ", Curtosis =", kurtosis_original, "\n")
```

```
## Original: Sesgo = 0.7591268 , Curtosis = 0.1193971
```

```r
cat("Transformación Exacta: Sesgo =", skewness_exacta, ", Curtosis =", kurtosis_exacta, "\n")
```

```
## Transformación Exacta: Sesgo = -0.1701619 , Curtosis = -0.186884
```

```r
cat("Transformación Aproximada (Raíz cuadrada): Sesgo =", skewness_aproximada, ", Curtosis =", kurtosis_
```

```
## Transformación Aproximada (Raíz cuadrada): Sesgo = -0.01902765 , Curtosis = -0.3144682
```

De acuerdo a nuestros resultados, parece ser que la transformacion aproximada nos da mejores resultados, vamos a hacer pruebas de normalidad para ambas

```r
# Prueba de normalidad Shapiro-Wilk para la transformacion exacta
shapiro.test(dist_transformada_exacta)
```

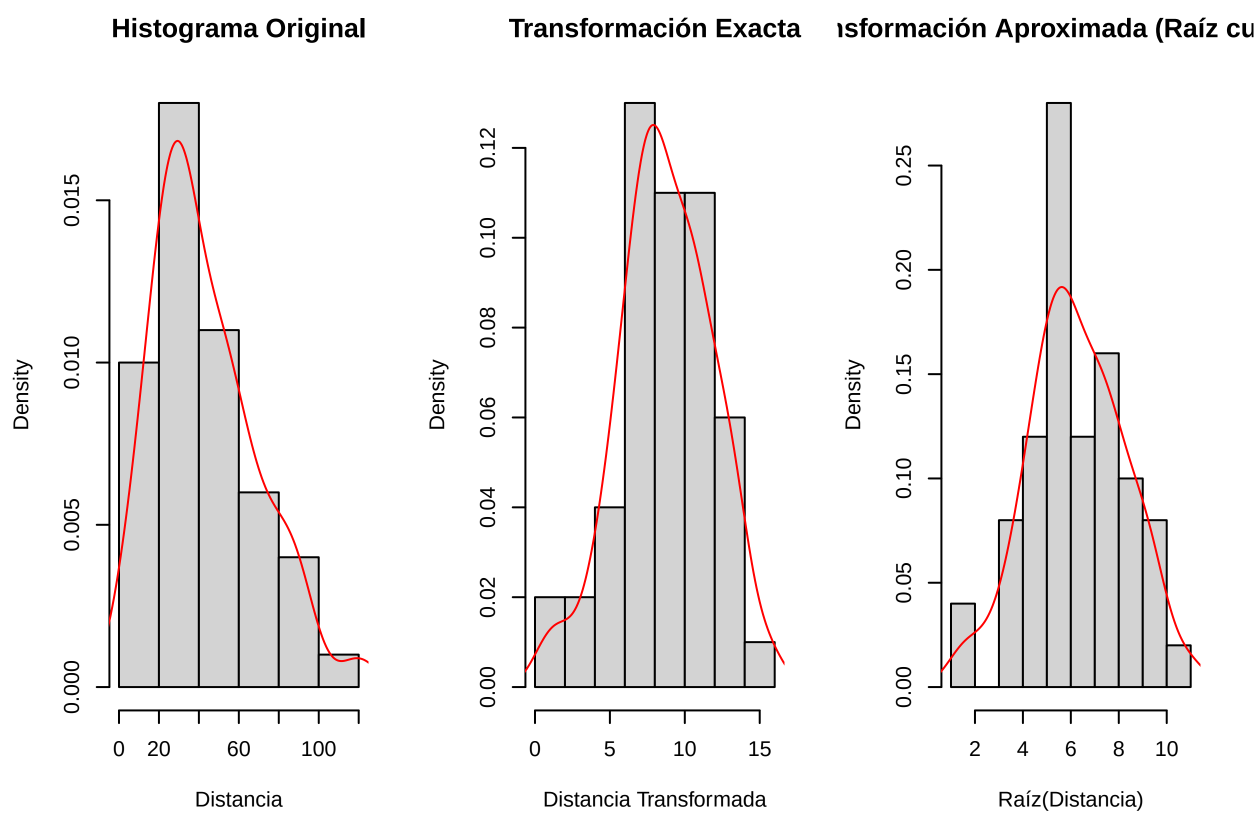
```
##
##  Shapiro-Wilk normality test
##
## data:  dist_transformada_exacta
## W = 0.99168, p-value = 0.9773
```

```r
# Prueba de normalidad Shapiro-Wilk para la transformacion aproximada
shapiro.test(dist_transformada_aproximada)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dist_transformada_aproximada
## W = 0.99347, p-value = 0.9941
```

ambas pruebas nos indican que se distribuyen de forma normal, aunque la transformada aproximada tiene menor curtosis, vamos a visualizarlos

```r
# Comparar las transformaciones con histogramas
par(mfrow = c(1, 3))  # Tres gráficos en una fila

# Histograma de los datos originales
hist(cars$dist, freq=FALSE, main="Histograma Original", xlab="Distancia", col="lightgray")
lines(density(cars$dist), col="red")

# Histograma de la transformación exacta
hist(dist_transformada_exacta, freq=FALSE, main="Transformación Exacta", xlab="Distancia Transformada",
lines(density(dist_transformada_exacta), col="red")

# Histograma de la transformación aproximada (Raíz cuadrada)
hist(dist_transformada_aproximada, freq=FALSE, main="Transformación Aproximada (Raíz cuadrada)", xlab="I
lines(density(dist_transformada_aproximada), col="red")
```



voy a uriliza la transformación de la raíz para mi nuevo modelo

```r
# Realizar la regresión lineal simple usando la transformación exacta
modelo_transformado_aproximada_cars <- lm(dist_transformada_aproximada ~ cars$speed)

# Mostrar el resumen del modelo para ver los coeficientes y la significancia
summary(modelo_transformado_aproximada_cars)
```

```
##
## Call:
```

```
## lm(formula = dist_transformada_aproximada ~ cars$speed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0684 -0.6983 -0.1799  0.5909  3.1534
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.27705    0.48444   2.636   0.0113 *
## cars$speed   0.32241    0.02978  10.825 1.77e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.102 on 48 degrees of freedom
## Multiple R-squared:  0.7094, Adjusted R-squared:  0.7034
## F-statistic: 117.2 on 1 and 48 DF,  p-value: 1.773e-14
```

tenemos como ecuación dist_transformada = 1.27 + 0.32*velocidad

tiene un coeficiente ajustado más elevado que la regresión lineal sin la transformación y el estadistico de f es muy alto, que a a la vez el pvalue correspondiente es super bajo, ambas estadisticas nos indican que existe una relación muy fuerte entre la variable dependiente y la independiente.

Haremos un análisis del modelo

```r
residuos_cars_transformado <- residuals(modelo_transformado_aproximada_cars)
```

Residuos con media cero

hipotesis nula : la media de los errores es igual a 0 hiporesis alrernativa : la media de los errores es diferente de 0 regla de decisión con prueba t : Si el valor p es menor que 0.05, se rechaza la hipótesis nula y se concluye que la media de los residuos es significativamente diferente de 0.

```r
prueba_t_cars_transformada <- t.test(residuos_cars_transformado, mu=0)
print(prueba_t_cars_transformada)
```

```
##
##  One Sample t-test
##
## data:  residuos_cars_transformado
## t = 2.7701e-16, df = 49, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.3100858  0.3100858
## sample estimates:
##    mean of x
## 4.274359e-17
```

el p value nos dio 1, el maximo valor posible por lo que no hay suficientes pruebas para rechazar la hipotesis nula

Normalidad de los residuos para el modelo con la transformada

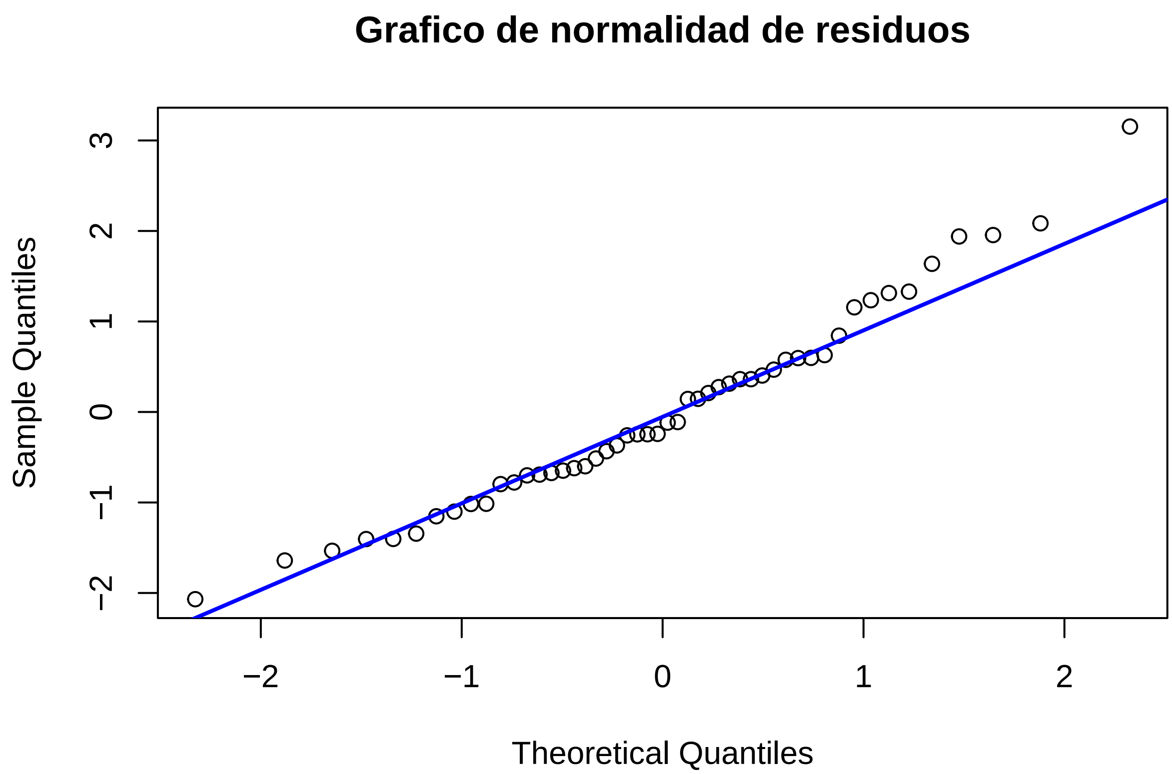
hipotesis nula : los residuos siguen una distribución normal. hipotesis alternativa : los residuos no siguen una distribución normal regla de decisión : Si el valor p es menor a 0.05, se rechaza la hipótesis nula y se concluye que los residuos no siguen una distribución normal

```r
# Prueba de Shapiro-Wilk
shapiro.test(residuos_cars_transformado)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuos_cars_transformado
## W = 0.97332, p-value = 0.3143
```

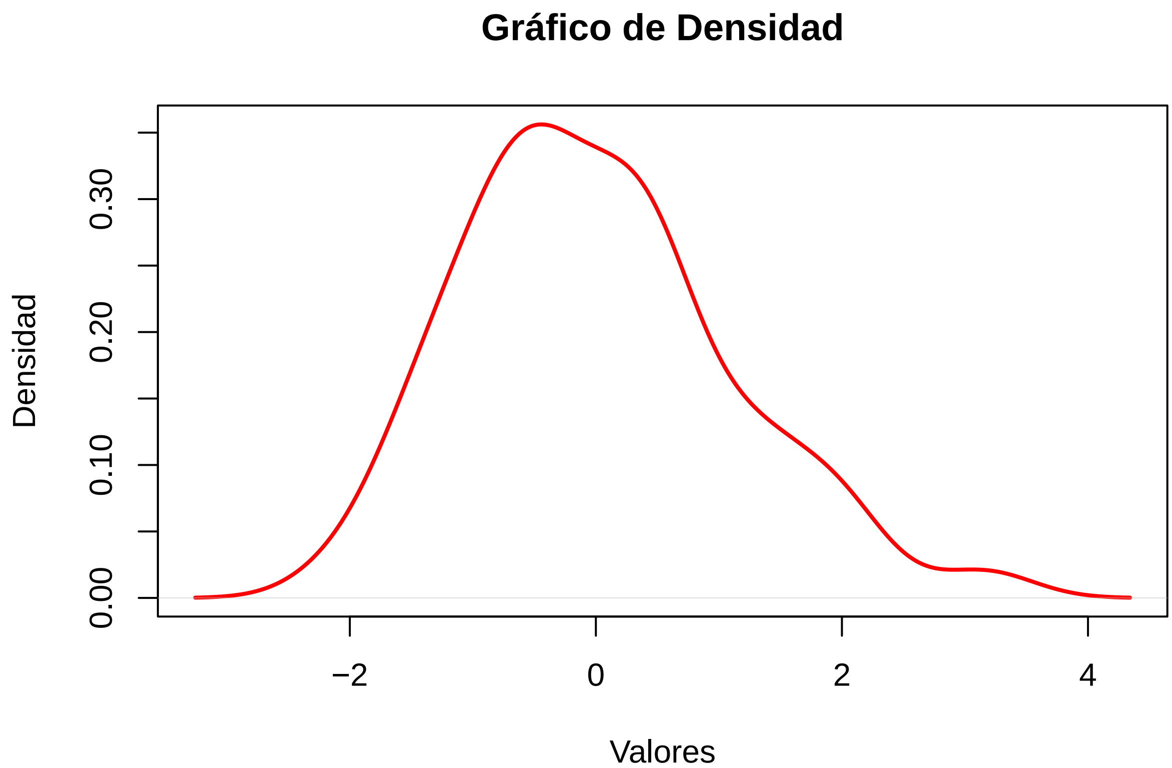
no se rechaza la hipotesis nula, $0.31 > 0.05$

```r
# QQ-Plot de los residuos
qqnorm(residuos_cars_transformado, main="Grafico de normalidad de residuos")
qqline(residuos_cars_transformado, col = "blue", lwd = 2)
```



ahora hacemos el grafico de densidad:

```r
plot(density(residuos_cars_transformado),
col = "red",
lwd = 2,
main = "Gráfico de Densidad",
xlab = "Valores",
ylab = "Densidad")
```



Las graficas parece que muestran una distribución parecida a una normal pero parece estar ligeramente sesgada a la derecha, y al el descenso no es suave como en una distribución normal

independencia

hipotesis nula: los errores no estan correlacionados

hipotesis alternativa : los errores estan correlacionados

regla de decisión: si el valor p de la prueba es menor a 0.05 se rechaza la hipotesis nula

hacemos el modelo durbin watson para verificar independencia entre los residuos

```r
# Realizar la prueba de Durbin-Watson
prueba_dw_cars_transoformada <- dwtest(modelo_transformado_aproximada_cars)

# Mostrar los resultados de la prueba
print(prueba_dw_cars_transoformada)
```

```
##
##  Durbin-Watson test
##
## data:  modelo_transformado_aproximada_cars
## DW = 1.9417, p-value = 0.3609
## alternative hypothesis: true autocorrelation is greater than 0
```

no se rechaza la hipotesis nula

linealidad.

para verificar la linealidad vamos a utilizar una prueba RESET

La prueba RESET de Ramsey (Regression Equation Specification Error Test) es utilizada para detectar posibles errores de especificación en un modelo de regresión lineal. La prueba examina si hay variables omitidas o si la forma funcional del modelo es incorrecta.

hipotesis nula: no hay términos omitidos que indican linealidad

hipotesis alternativa: hay una especificación errónea en el modelo que indica no linealidad

regla de decisión: si el valor p de la prueba es menor a 0.05 se rechaza la hipotesis nula

```
# Realizar la prueba RESET de Ramsey
prueba_reset_cars_transformada <- resettest(modelo_transformado_aproximada_cars)

# Mostrar los resultados de la prueba
print(prueba_reset_cars_transformada)
```

```
##
##  RESET test
##
## data:  modelo_transformado_aproximada_cars
## RESET = 0.47002, df1 = 2, df2 = 46, p-value = 0.628
```

no se rechaza la hipotesis nula

Homocedasticidad

hipotesis nula: la varianza de los errores es constante (hay homocedasticidad)

hipotesis alternativa : la varianza de los errores no es constante (hay heterocedasticidad)

regla de decisión: si el valor p es menor o igual que 0.05 entonces rechazamos la hipotesis nula para realizar las pruebas vamos a necesitar descargar un paquete que contiene la prueba que vamos a utilizar

```
library(lmtest)
```

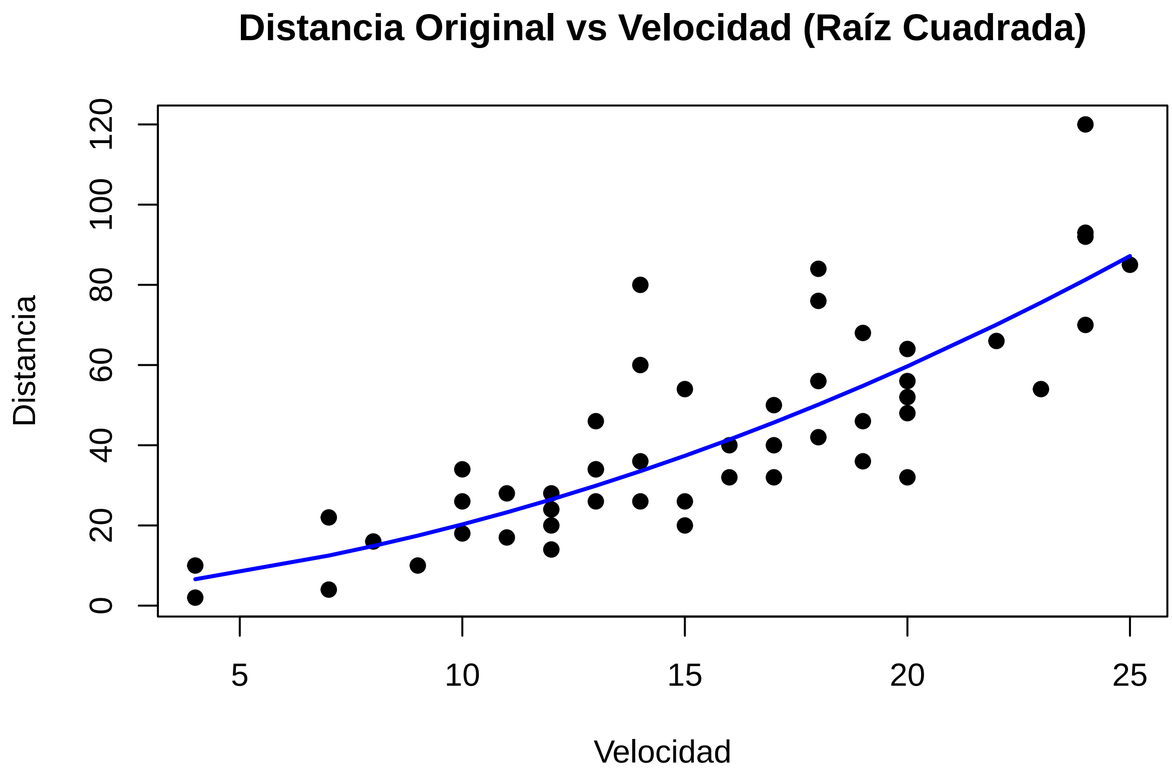
```
prueba_white_cars_transformada <- bptest(modelo_transformado_aproximada_cars, ~ fitted(modelo_transforma
print(prueba_white_cars_transformada)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  modelo_transformado_aproximada_cars
## BP = 0.39944, df = 2, p-value = 0.819
```

no se rechaza la hipotesis nula

vamos a despejar la ecuación

```
# Despejar la distancia original a partir de la predicción del modelo con raíz cuadrada
predicciones_transformadas_aproximada <- predict(modelo_transformado_aproximada_cars)
predicciones_distancia_aproximada <- predicciones_transformadas_aproximada^2

# Graficar los datos originales y la línea del modelo en la escala original
plot(cars$speed, cars$dist, main="Distancia Original vs Velocidad (Raíz Cuadrada)",
     xlab="Velocidad", ylab="Distancia", pch=19)
lines(cars$speed, predicciones_distancia_aproximada, col="blue", lwd=2)
```



Conclusiones:

El mejor modelo fue el que utiliza la transformación, esto se puede determinar por que tiene una explicación sobre el modelo mas elevada y esto lo hace decisivo aunque la significancia de ambos modelos fuera parecido tiene mejores resultados, aparte de que desde mi punto de vista no genera mucha complejidad en el modelo. En la regresión lineal simple parecia que había muchos datos que no estaba tomando en cuenta, a comparación de la transformada que parece que se esta tratando de adaptar a captar más datos.