tarea_12_A01742161

Rogelio Lizárraga

2024-09-10

El objetivo es encontrar el mejor modelo que relacione la velocidad de los automóviles y las distancias necesarias para detenerse en autos de modelos existentes en 1920 (base de datos car). La ecuación encontrada no sólo deberá ser el mejor modelo obtenido sino también deberá ser el más económico en terminos de la complejidad del modelo.

Parte 1: Análisis de normalidad

```r
data(cars)
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

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

```r
library(nortest)
library(e1071)
shapiro.test(cars$speed)
```

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

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

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

```r
jarque.bera.test(cars$speed)
```

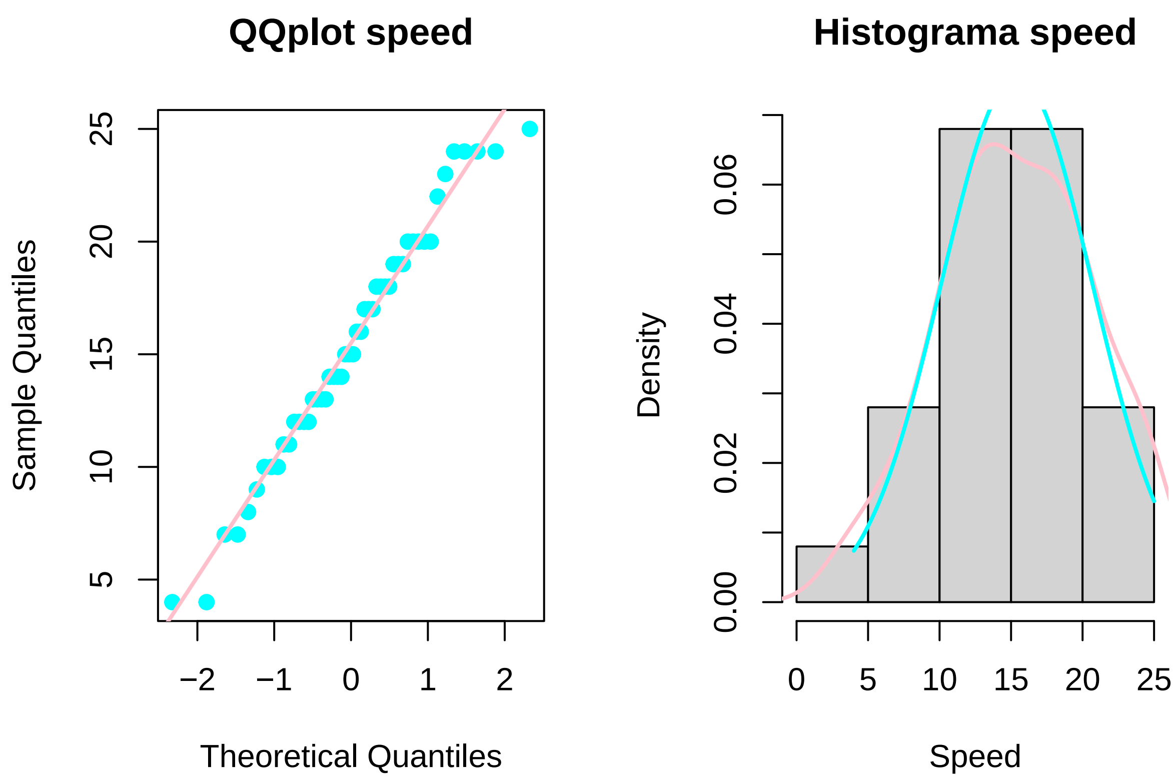
```
##
##  Jarque Bera Test
##
## data:  cars$speed
## X-squared = 0.80217, df = 2, p-value = 0.6696
```

```r
jarque.bera.test(cars$dist)
```

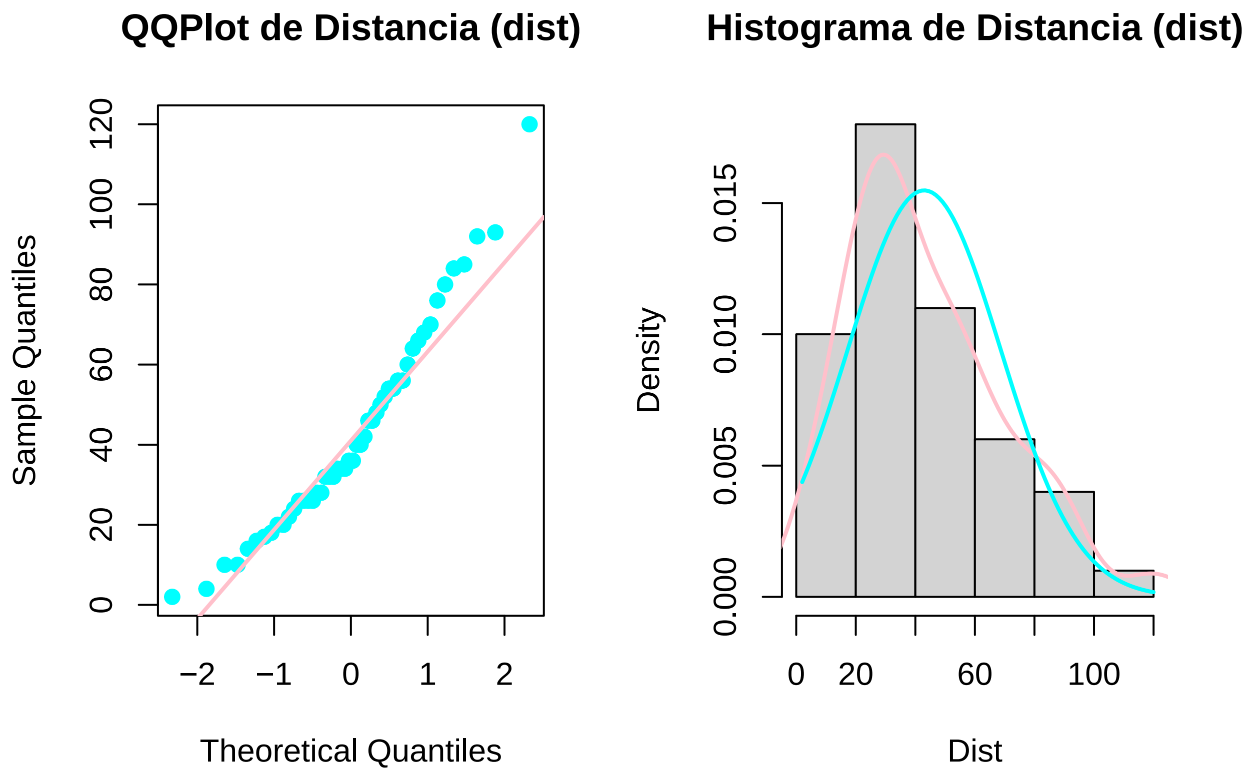
```
##
##  Jarque Bera Test
##
## data:  cars$dist
## X-squared = 5.2305, df = 2, p-value = 0.07315
```

Realiza gráficos que te ayuden a identificar posibles alejamientos de normalidad:

```r
# Establecer el espacio de los gráficos para dos columnas
par(mfrow = c(1, 2))

# Gráficos para la variable velocidad (speed)
# QQPlot para velocidad
qqnorm(cars$speed, main = "QQplot speed", pch = 19, col = "cyan")
qqline(cars$speed, col = "pink", lwd = 2)

# Histograma y densidad para velocidad
hist(cars$speed, freq = FALSE, main = "Histograma speed",
     xlab = "Speed", col = "lightgray", border = "black")
lines(density(cars$speed), col = "pink", lwd = 2)
curve(dnorm(x, mean = mean(cars$speed), sd = sd(cars$speed)),
      from = min(cars$speed), to = max(cars$speed), add = TRUE, col = "cyan", lwd = 2)
```




```r
par(mfrow = c(1, 2))
qqnorm(cars$dist, main = "QQPlot de Distancia (dist)", pch = 19, col = "cyan")
qqline(cars$dist, col = "pink", lwd = 2)

# Histograma y densidad para distancia
hist(cars$dist, freq = FALSE, main = "Histograma de Distancia (dist)",
     xlab = "Dist", col = "lightgray", border = "black")
lines(density(cars$dist), col = "pink", lwd = 2)
curve(dnorm(x, mean = mean(cars$dist), sd = sd(cars$dist)),
      from = min(cars$dist), to = max(cars$dist), add = TRUE, col = "cyan", lwd = 2)
```



Calcula el coeficiente de sesgo y el coeficiente de curtosis (sugerencia: usar la librería e1071, usar: skeness y kurtosis) para cada variable.

```
library(e1071)
cat('sesgo speed:', skewness(cars$speed))
```

```
## sesgo speed: -0.1105533
```

```
cat('\ncurtosis speed:', kurtosis(cars$speed))
```

```
##
## curtosis speed: -0.6730924
```

```
cat ('\nsesgo distancia:', skewness(cars$dist))
```

```
##
## sesgo distancia: 0.7591268
```

```
cat ('\ncurtosis distancia:', kurtosis(cars$dist))
```

```
##
## curtosis distancia: 0.1193971
```

Comenta cada gráfico y resultado que hayas obtenido. Emite una conclusión final sobre la normalidad de los datos. Argumenta basándote en todos los análisis realizados en esta parte. Incluye posibles motivos de alejamiento de normalidad.

Como podemos observar, los valores p de Jarque-Bera en distancia y velocidad > 0.05 por lo que la hipótesis inicial no se rechaza y podemos concluir que los datos se distribuyen como una normal. Sin embargo, con shapiro-wilk el valor p de velocidad > 0.05, mientras que el valor p de distanca < 0.05, por lo que la hipótesis se rechaza en distancia y no podemos decir que los datos de distancia se distribuyen como una normal, pero sí los de velocidad. Por otro lado, observamos una cola derecha pesada en distancia, así como una asimetría. Además, tenemos un sesgo de 0.759 (bastante alto) y una curtosis de 0.119 (es adecuada y aceptable), por lo que dist no es normal.

Finalmente, observamos que speed tiene una curtosis de -0.67 (negativamente alta) y un sesgo de -0.11 (bastante bajo) y su distribución es simétrica, por lo que es normal.

La distancia no es normal por su variabilidad y distribución no adecuada para considerarse normal, lo cual puede ser causado por diversos factores, como puede ser el frenado de los carros, o bien, mediciones erróneas.

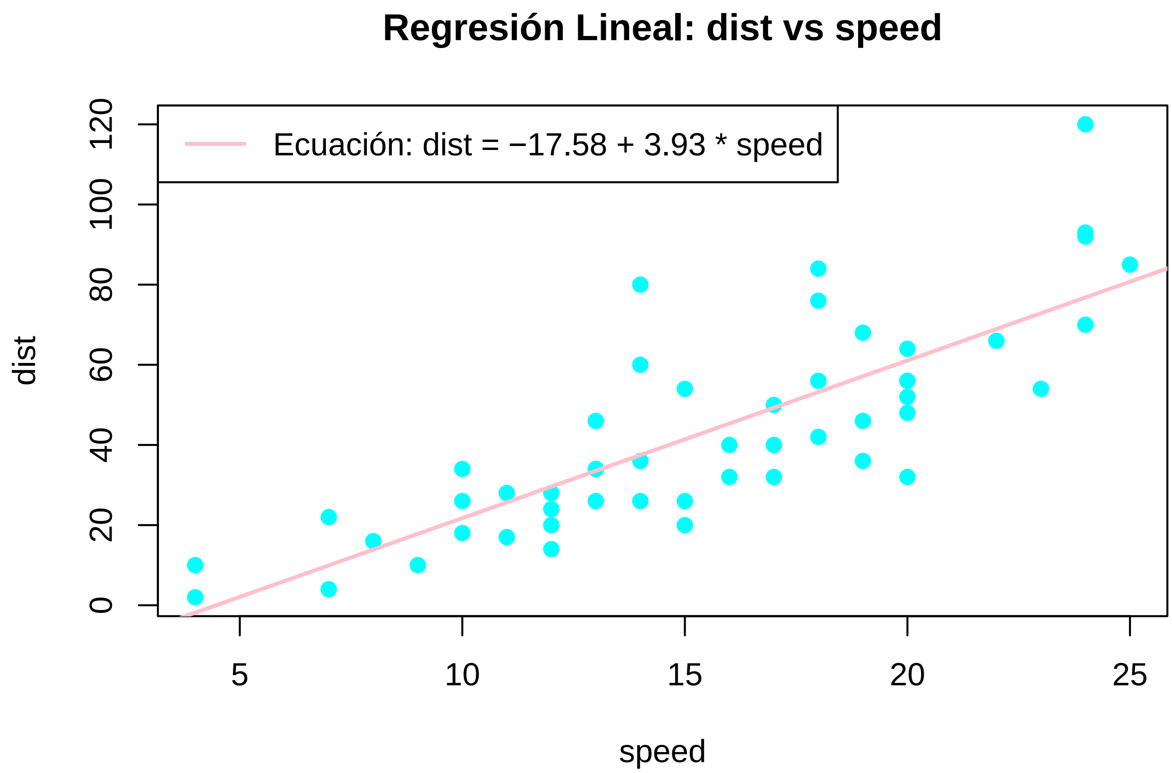
Parte 2: regresión lineal

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

```
modelo1 <- lm(dist ~ speed, data = cars)
summary(modelo1)
```

```
##
## 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
```

```
plot(cars$speed, cars$dist,
     main = "Regresión Lineal: dist vs speed",
     xlab = "speed",
     ylab = "dist",
     pch = 19, col = "cyan")
abline(modelo1, col = "pink", lwd = 2)
legend("topleft", legend = paste("Ecuación: dist =", round(coef(modelo1)[1], 2), "+", round(coef(modelo
       col = "pink", lwd = 2)
```



Como podemos observar, la ecuación del modelo es: -17.591 + 3.9324*speed. Obtenemos un coeficiente de determinación de 0.6511, por lo que el modelo explica el 65.11% de la variabilidad de la distancia. Vemos que β_0 y $\beta_1 < 0.05$, por lo que son estadísticamente significativos. De la misma manera, nuestro valor p del modelo es < 0.05, por lo que es estadísticamente significativo.

Supuestos del modelo

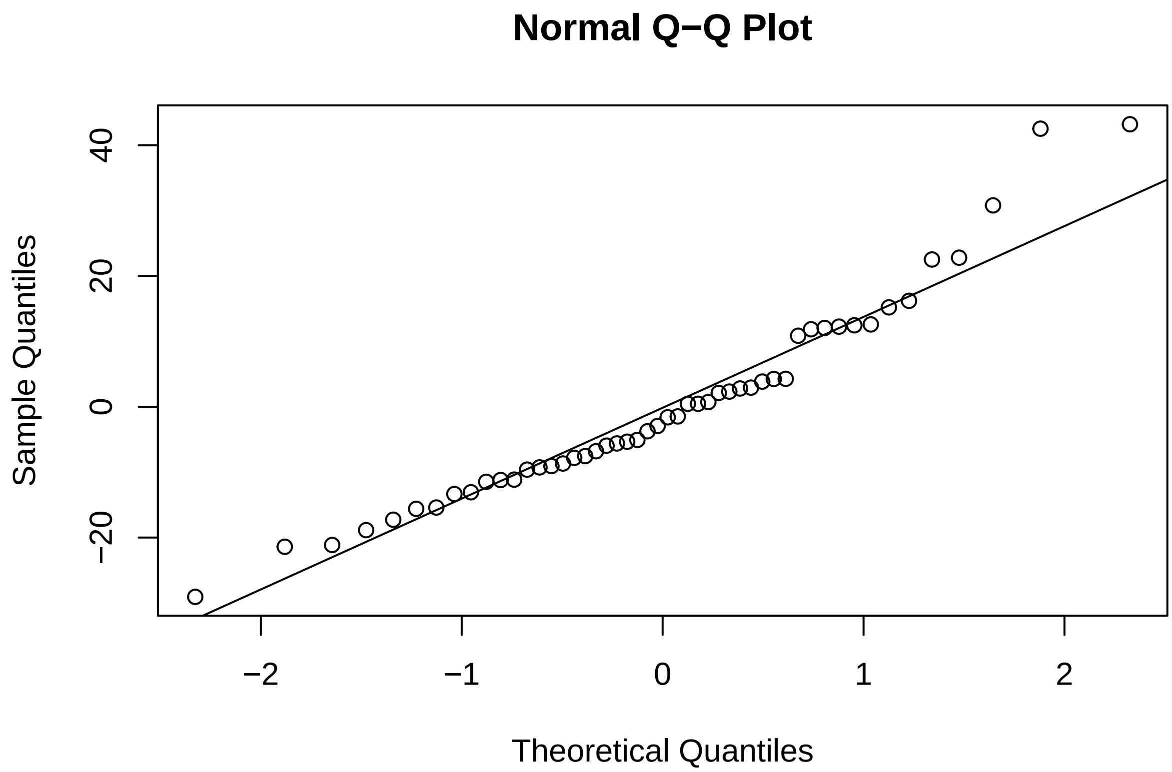
Normalidad

H_0 : Los residuos siguen una distribución normal. H_1 : Los residuos NO siguen una distribución normal.

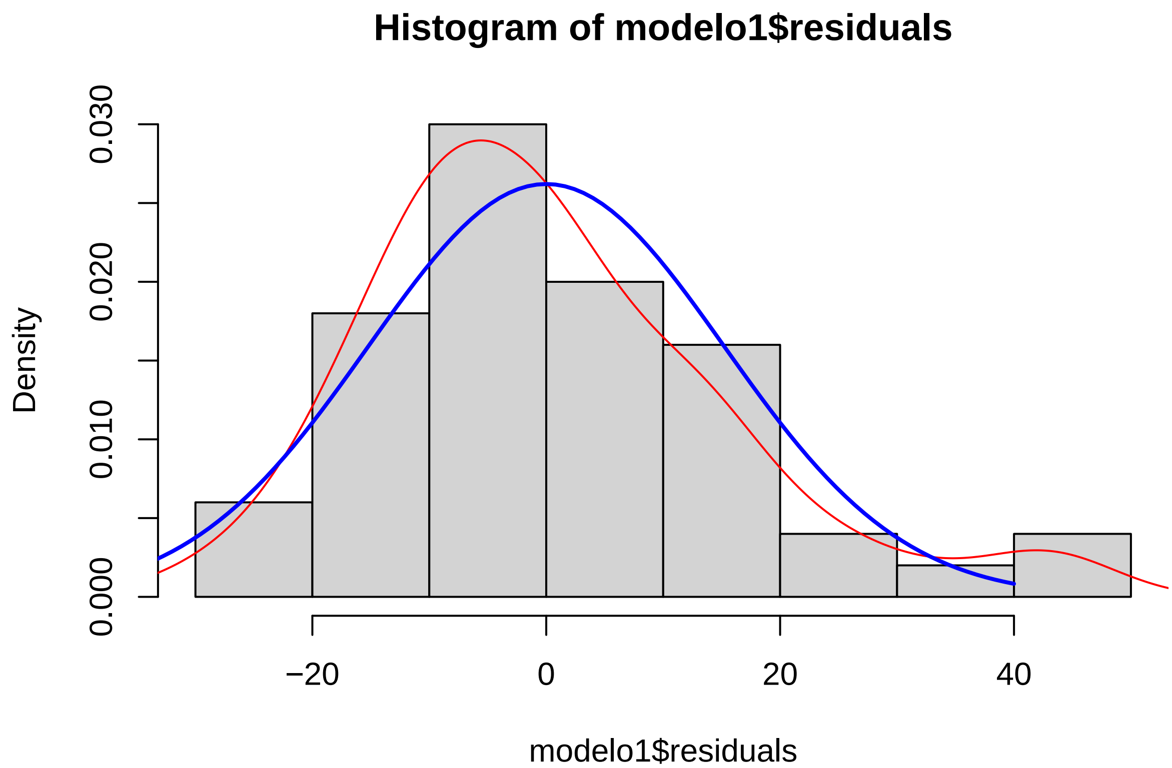
```r
library(nortest)
shapiro.test(residuals(modelo1))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(modelo1)
## W = 0.94509, p-value = 0.02152
```

```r
qqnorm(modelo1$residuals)
qqline(modelo1$residuals)
```




```
hist(modelo1$residuals,freq=FALSE)
lines(density(modelo1$residuals),col="red")
curve(dnorm(x,mean=mean(modelo1$residuals),sd=sd(modelo1$residuals)), from=-
40, to=40, add=TRUE, col="blue",lwd=2)
```



Como podemos observar, el valor p < 0.05, por lo que H_0 se rechaza y los residuos NO siguen una distribución normal.

Verificación de media cero

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

```
t.test(modelo1$residuals)
```

```
##
##  One Sample t-test
##
## data:  modelo1$residuals
## t = 1.0315e-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
## 2.220446e-16
```

Como tenemos un valor p ≈ 1, H_0 no se rechaza, por lo que los residuos tienen media cero.

Homocedasticidad

H_0 : La varianza de los errores es constante (Hay homocedasticidad). H_1: La varianza de los errores NO es constannte (Hay heterocedasticidad).

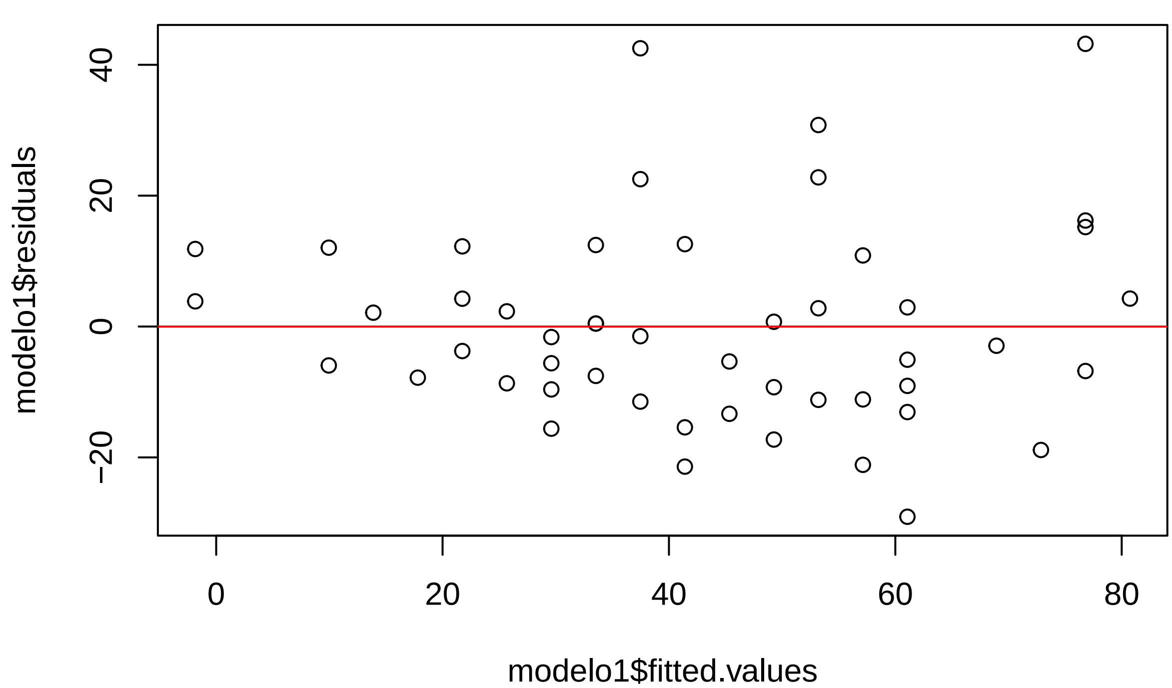
```
library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

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

```
bptest(modelo1)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  modelo1
## BP = 3.2149, df = 1, p-value = 0.07297
```

```
plot(modelo1$fitted.values,modelo1$residuals)
abline(h=0, col= 'red')
```



Como el valor p > 0.05, No rechaza H_0, por lo que la varianza de los errores es constante (hay homocedasticidad). Además, esto se puede observar en el gráfico, pues la varianza fluctúa dentro de un rango de variabilidad sin importar el punto del gráfico.

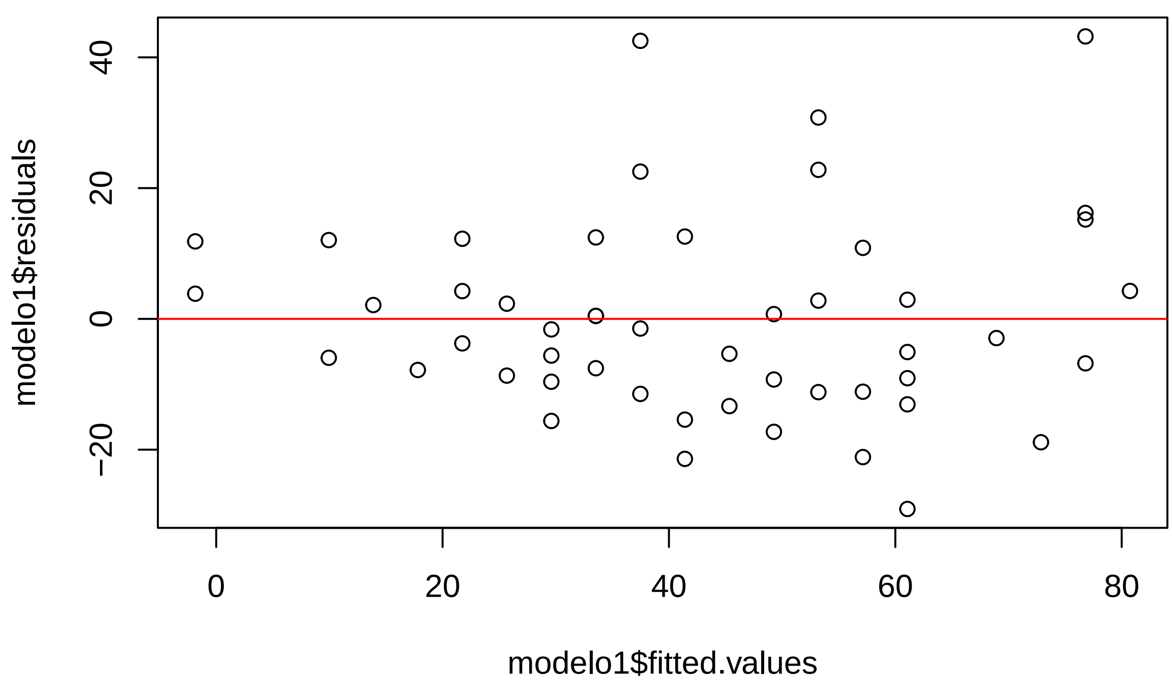
Independencia

H_0: La autocorrelación de los residuos es 0 (hay independencia). H_1: La autocorrelación de los residuos $\neq 0$ (no hay independencia).

```
dwtest(modelo1)
```

```
##
##  Durbin-Watson test
##
## data:  modelo1
## DW = 1.6762, p-value = 0.09522
## alternative hypothesis: true autocorrelation is greater than 0
```

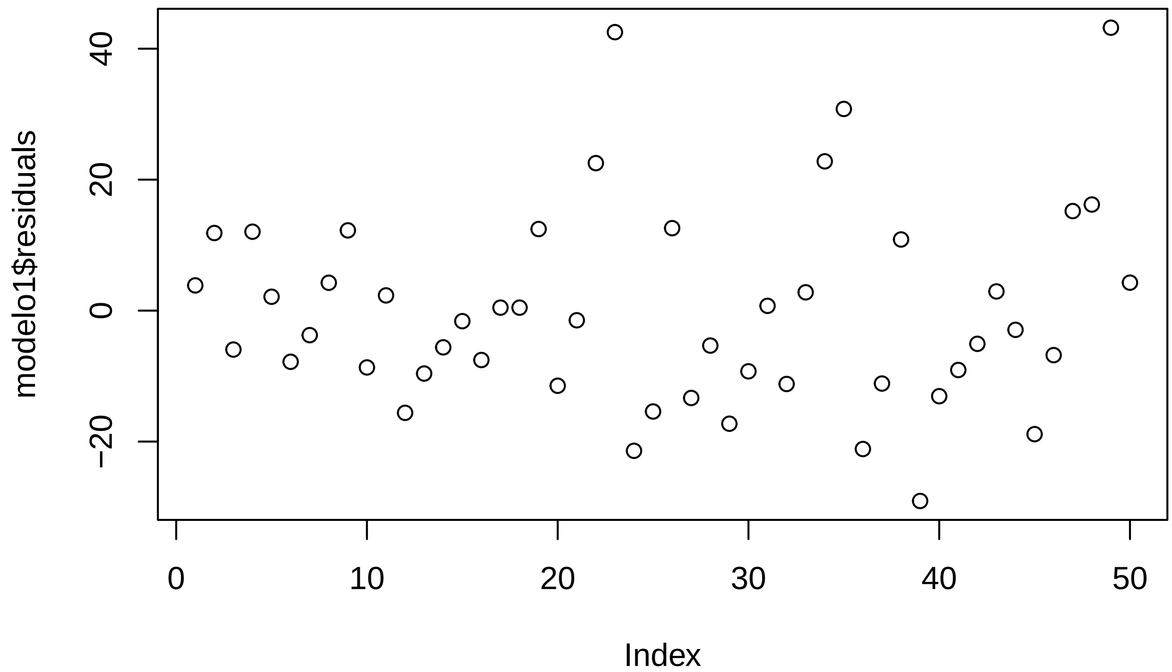
```
plot(modelo1$fitted.values,modelo1$residuals)
abline(h=0, col= 'red')
```



Como el valor p > 0.05, no se rechaza H_0, por lo que sí hay independencia en los residuos. Además, no se observa un patrón en la gráfica, o una dependencia entre los residuos.

Linealidad

```
plot(modelo1$residuals)
```



Observamos que los residuos se distribuyen aleatoriamente en torno a cero, por lo que la relación lineal es adecuada.

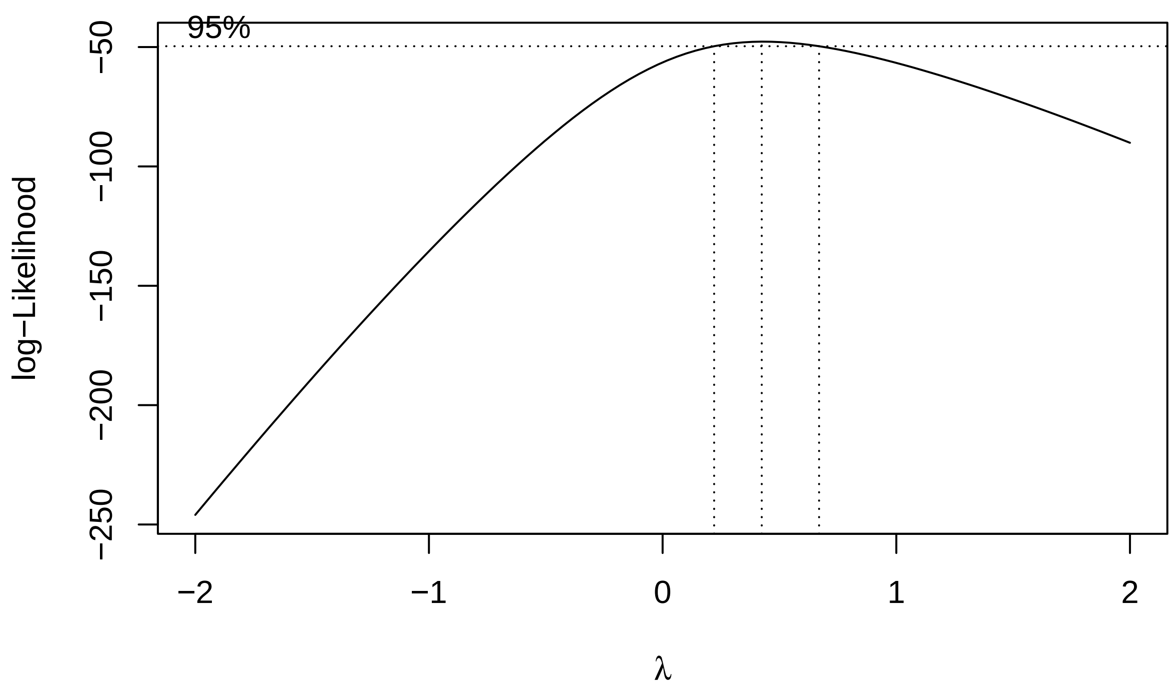
Conclusión

El modelo tiene un coeficiente de determinación moderado y se ajusta adecuadamente. Sin embargo, debido a que el modelo no cumple con normalidad, este modelo no es adecuado, por lo que compararemos con otro modelo de regresión.

Parte 3: 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 (ojo: concentrate solo en la variable que tiene más alejamiento de normalidad). Encuentra el valor de en la transformación Box-Cox para el modelo lineal: donde Y sea la distancia y X la velocidad. Aprovecha que el comando de boxcox en R te da la oportunidad de trabajar con el modelo líneal. La transformación se hará sobre la variable que usas como dependiente en el comando lm(y~x). Posteriormente, define la transformación exacta y el aproximada de acuerdo con el valor de que encontraste en la transformación de Box y Cox. Escribe las ecuaciones de las dos transformaciones encontradas.

```r
library(MASS)
boxcox_result <- boxcox(modelo1, lambda = seq(-2, 2, by = 0.1))
```




```r
cat('Lambda óptimo:', boxcox_result$x[which.max(boxcox_result$y)])
```

```
## Lambda óptimo: 0.4242424
```

```r
X_original <- cars$dist
X1 <- sqrt(X_original)

lambda <- 0.4242424
X2 <- ((X_original)^lambda - 1) / lambda

sesgo_original <- skewness(X_original)
curtosis_original <- kurtosis(X_original)

sesgo_X1 <- skewness(X1)
curtosis_X1 <- kurtosis(X1)

sesgo_X2 <- skewness(X2)
curtosis_X2 <- kurtosis(X2)
cat("Datos originales:\n")
```

```
## Datos originales:
```

```r
cat("  - Sesgo:", sesgo_original, "\n")
```

```
##   - Sesgo: 0.7591268
```

```r
cat("  - Curtosis:", curtosis_original, "\n\n")
```

```
##   - Curtosis: 0.1193971
```

```r
cat("Transformación con el modelo aproximado (X^0.5):\n")
```

```
## Transformación con el modelo aproximado (X^0.5):
```

```r
cat("  - Sesgo:", sesgo_X1, "\n")
```

```
##   - Sesgo: -0.01902765
```

```r
cat("  - Curtosis:", curtosis_X1, "\n\n")
```

```
##   - Curtosis: -0.3144682
```

```r
cat("Transformación con el modelo exacto ((X^lambda - 1) / lambda):\n")
```

```
## Transformación con el modelo exacto ((X^lambda - 1) / lambda):
```

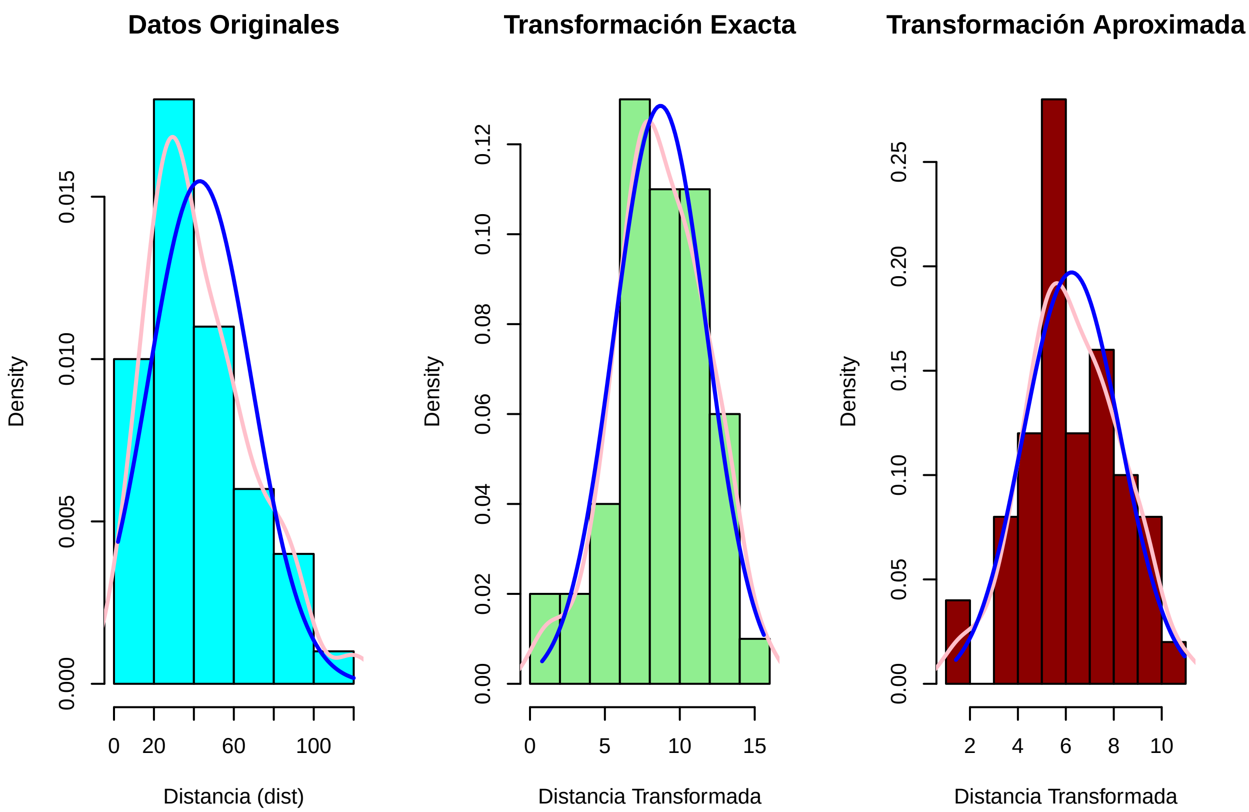
```r
cat("  - Sesgo:", sesgo_X2, "\n")
```

```
##   - Sesgo: -0.1701619
```

```
##   - Sesgo: -0.1126697
cat("  - Curtosis:", curtosis_X2, "\n")
```

```
##   - Curtosis: -0.186884
```

```
X_exacta <- X2
X_aproximada <- X1
par(mfrow = c(1, 3))
hist(X_original, freq = FALSE, main = "Datos Originales",
     xlab = "Distancia (dist)", col = "cyan", border = "black")
lines(density(X_original), col = "pink", lwd = 2)
curve(dnorm(x, mean = mean(X_original), sd = sd(X_original)),
      from = min(X_original), to = max(X_original), add = TRUE, col = "blue", lwd = 2)

# Histograma de la transformación exacta de Box-Cox
hist(X_exacta, freq = FALSE, main = "Transformación Exacta",
     xlab = "Distancia Transformada", col = "lightgreen", border = "black")
lines(density(X_exacta), col = "pink", lwd = 2)
curve(dnorm(x, mean = mean(X_exacta), sd = sd(X_exacta)),
      from = min(X_exacta), to = max(X_exacta), add = TRUE, col = "blue", lwd = 2)

# Histograma de la transformación aproximada de Box-Cox
hist(X_aproximada, freq = FALSE, main = "Transformación Aproximada",
     xlab = "Distancia Transformada", col = "darkred", border = "black")
lines(density(X_aproximada), col = "pink", lwd = 2)
curve(dnorm(x, mean = mean(X_aproximada), sd = sd(X_aproximada)),
      from = min(X_aproximada), to = max(X_aproximada), add = TRUE, col = "blue", lwd = 2)
```



Ecuación modelo aproximado :

$$y = \sqrt(x)$$

Ecuación modelo exacto :

$$y = (x^\lambda - 1)/\lambda$$

Como podemos observar, los datos originales tienen un sesgo de 0.75 (bastante alto) y una curtosis de (0.119), por lo que la distribución no es normal, debido al sesgo. Por otro lado, la transformación con el modelo aproximado tiene un sesgo de -0.019 (bastante bajo) y una curtosis de -0.31 (ligeramente alta), por lo que la distribución se puede considerar normal. Vemos que la transformación exacta da mejores valores para la curtosis y peores para el sesgo, pues el sesgo aumenta a -0.17 (relativamente bajo) y la curtosis disminuye bastante a -0.18 (relativamente baja), por lo que este modelo también da buenos resultados.

Realiza algunas pruebas de normalidad para los datos transformados.

```
cat('Datos originales: \n')
```

```
## Datos originales:
```

```
shapiro.test(cars$dist)
```

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

```r
jarque.bera.test(cars$dist)
```

```
##
##  Jarque Bera Test
##
## data:  cars$dist
## X-squared = 5.2305, df = 2, p-value = 0.07315
```

```r
cat('\n Transformación del modelo exacto: \n')
```

```
##
##  Transformación del modelo exacto:
```

```r
shapiro.test(X_exacta)
```

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

```r
jarque.bera.test(X_exacta)
```

```
##
##  Jarque Bera Test
##
## data:  X_exacta
## X-squared = 0.26684, df = 2, p-value = 0.8751
```

```r
cat('\n Transformación del modelo aproximado: \n')
```

```
##
##  Transformación del modelo aproximado:
```

```r
shapiro.test(X_aproximada)
```

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

```r
jarque.bera.test(X_aproximada)
```

```
##
##  Jarque Bera Test
##
## data:  X_aproximada
## X-squared = 0.089682, df = 2, p-value = 0.9561
```

Como ya se sabía, no hay normalidad para los datos originales, pues se rechaza la hipótesis inicial para shapiro-wilk, por lo que la distribución no es normal. Sin embargo, vemos que las transformaciones del modelo exacto y aproximado cumplen la normalidad bastante bien, pues los valores $p > 0.05$ y muy cercanos a 0.9, por lo que la hipótesis inicial no se rechaza y los datos se distribuyen de manera normal.

Detecta anomalías y corrige tu base de datos tranformado (datos atípicos, ceros anámalos, etc): solo en caso de no tener normalidad en las transformaciones. En caso de corrección de los datos por anomalías, vuelve a buscar la λ para tus nuevos datos.

Debido a que las transformaciones se distribuyen como una normal, lo anterior no es necesario, por lo que no se harán correcciones de datos atípicos y otro tipo de cambios en el conjunto de datos.

Concluye sobre las dos transformaciones realizadas: Define la mejor transformación de los datos de acuerdo a las características de las dos transformaciones encontradas (exacta o aproximada). Toman en cuenta la normalidad de los datos y la economía del modelo.

La mejor transformación es la aproximada, pues si bien aumenta la curtosis comparado con el modelo exacto, este aumento es muy poco, pero sí disminuye bastante el sesgo, por lo que logra encontrar un sesgo menor que el modelo exacto, el cual es más difícil de solucionar para la mayoría del análisis estadístico, por lo que nos quedaremos con el modelo aproximado

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

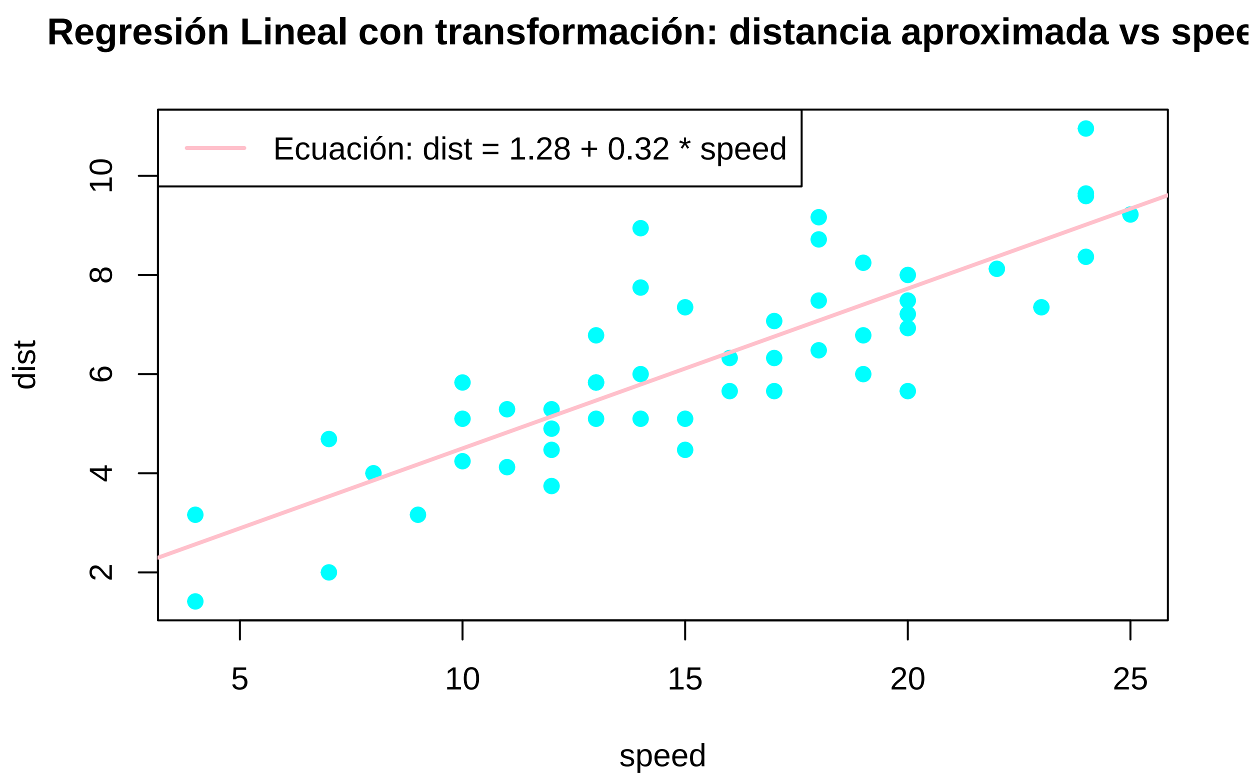
Prueba regresión lineal simple entre distancia y velocidad. Usa lm(y~x). Escribe el modelo lineal para la transformación.

```
Modelo2 <- lm(X_aproximada ~ speed, data = cars)
summary(Modelo2)
```

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

```
plot(cars$speed, X_aproximada,
     main = "Regresión Lineal con transformación: distancia aproximada vs speed",
     xlab = "speed",
     ylab = "dist",
     pch = 19, col = "cyan")
abline(Modelo2, col = "pink", lwd = 2)
legend("topleft", legend = paste("Ecuación: dist =", round(coef(Modelo2)[1], 2), "+", round(coef(Modelo2
       col = "pink", lwd = 2)
```




```
cat("\nCoeficiente de Correlación:",cor(X_aproximada, cars$speed),"\n")
```

```
##
## Coeficiente de Correlación: 0.8422666
```

Como podemos observar, la ecuación del modelo es: 1.28 + 0.32*speed. Obtenemos un coeficiente de determinación de 0.7094, por lo que el modelo explica el 70.94% de la variabilidad de la distancia. Vemos que β_0 y $\beta_1 < 0.05$, por lo que los coeficientes son estadísticamente significativos. De la misma manera, nuestro valor p del modelo es < 0.05, por lo que es estadísticamente significativo. Finalmente, tenemos un coeficiente de correlación del 84.22%, por lo que la velocidad y la distancia con el modelo aproximado están correlacionadas.

El modelo lineal de nuestra ecuación es : $\hat{dist}_{aprox} = 1.28 + 0.32 \cdot speed$

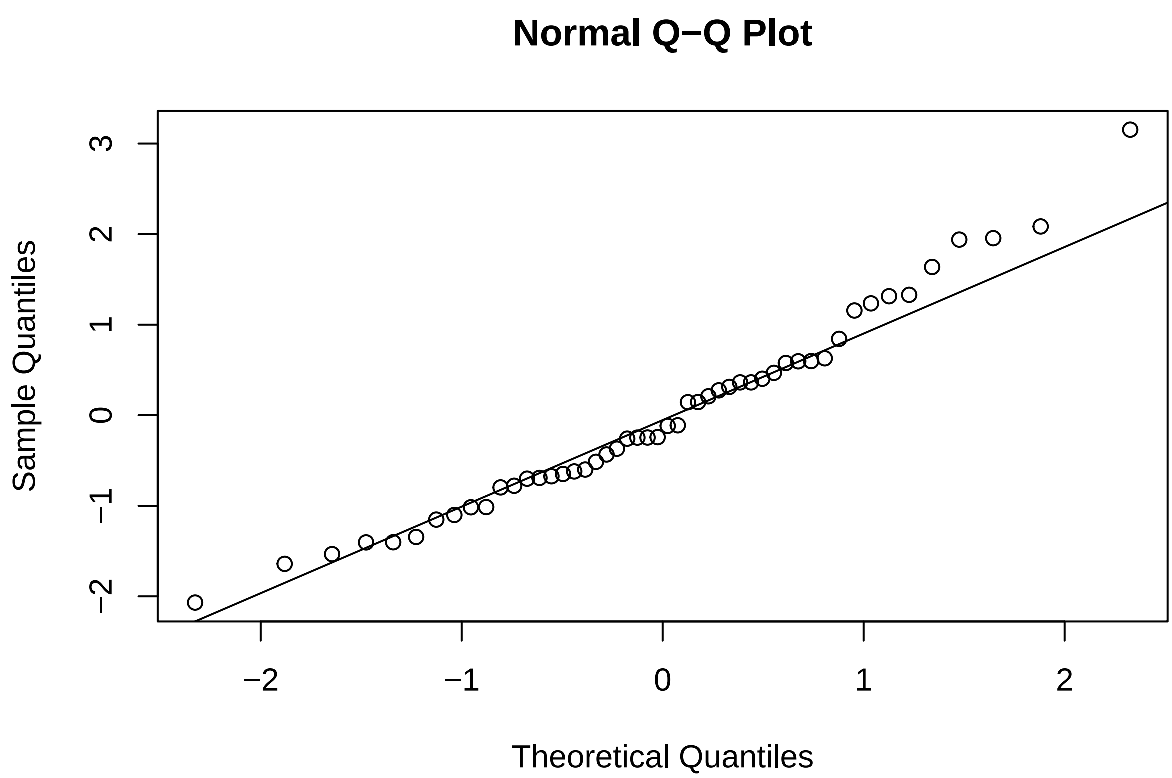
Normalidad

H_0 : Los residuos siguen una distribución normal. H_1 : Los residuos NO siguen una distribución normal.

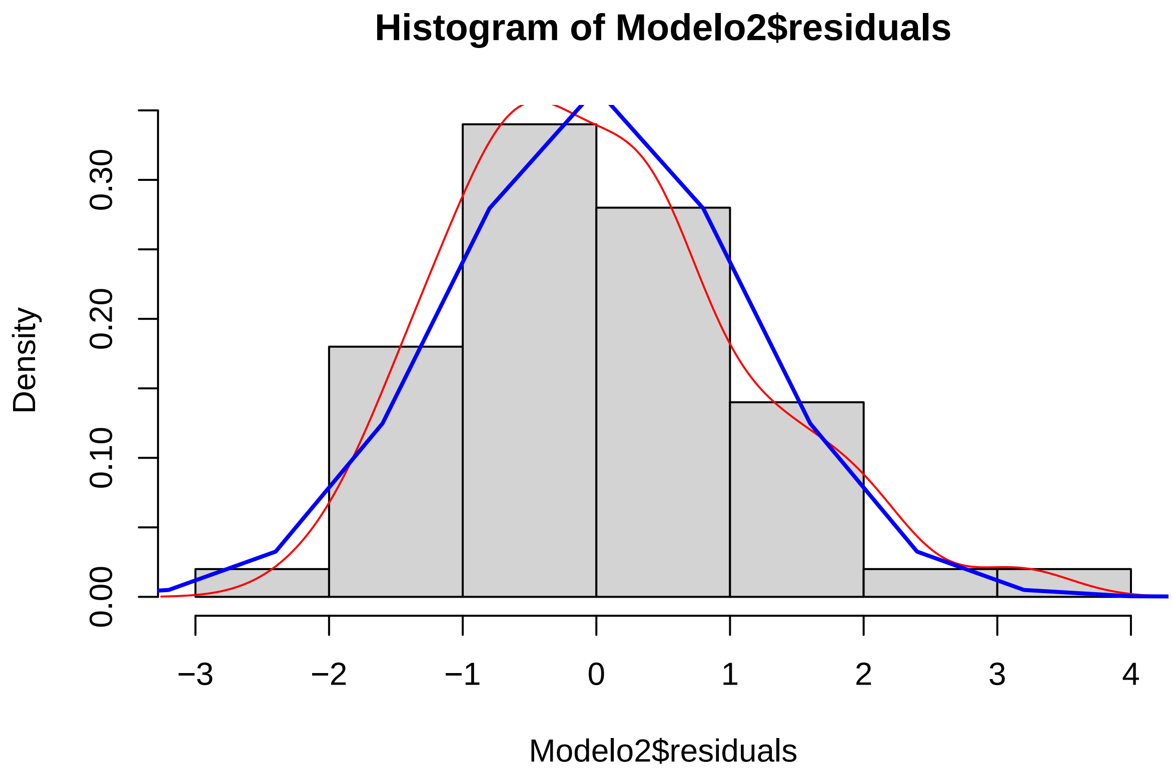
```
library(nortest)
shapiro.test(residuals(Modelo2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(Modelo2)
## W = 0.97332, p-value = 0.3143
```

```
qqnorm(Modelo2$residuals)
qqline(Modelo2$residuals)
```




```
hist(Modelo2$residuals,freq=FALSE)
lines(density(Modelo2$residuals),col="red")
curve(dnorm(x,mean=mean(Modelo2$residuals),sd=sd(Modelo2$residuals)), from=-
40, to=40, add=TRUE, col="blue",lwd=2)
```



Como podemos observar, el valor p > 0.05, por lo que H_0 no se rechaza y los residuos siguen una distribución normal.

Verificación de media cero

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

```r
t.test(Modelo2$residuals)
```

```
##
##  One Sample t-test
##
## data:  Modelo2$residuals
## t = 3.9571e-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
## 6.105969e-17
```

Como tenemos un valor p ≈ 1, H_0 no se rechaza, por lo que los residuos tienen media cero.

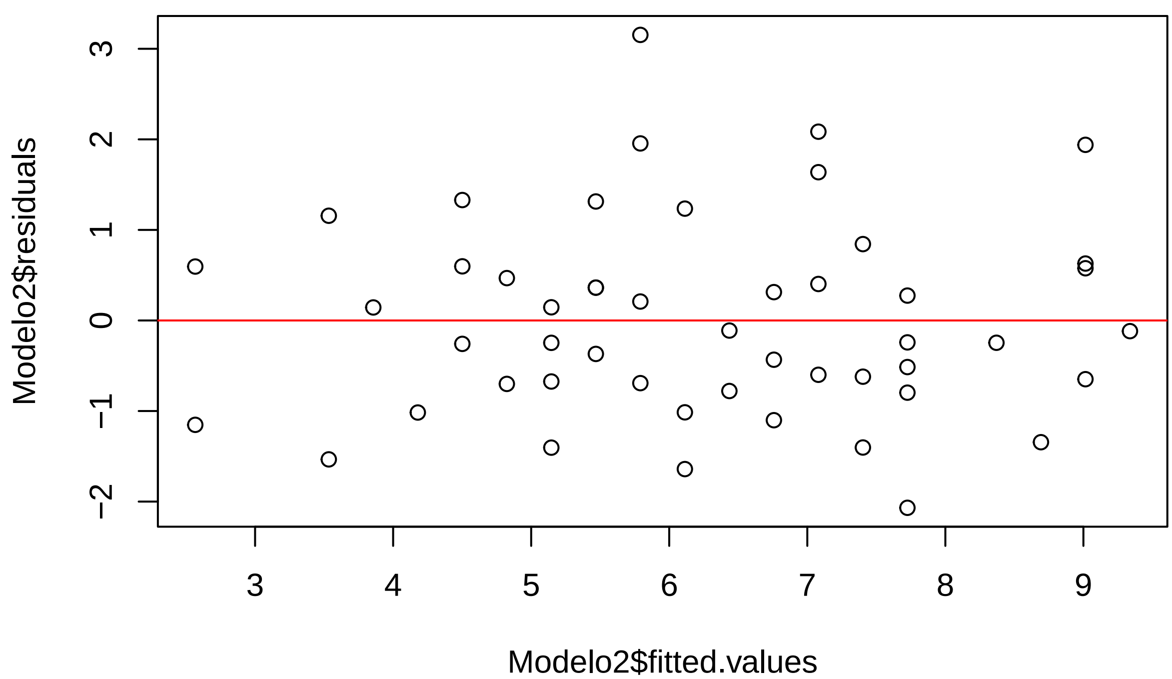
Homocedasticidad

H_0 : La varianza de los errores es constante (Hay homocedasticidad). H_1: La varianza de los errores NO es constannte (Hay heterocedasticidad).

```
library(lmtest)
bptest(Modelo2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Modelo2
## BP = 0.011192, df = 1, p-value = 0.9157
```

```
plot(Modelo2$fitted.values,Modelo2$residuals)
abline(h=0, col= 'red')
```



Como el valor p > 0.05, No rechaza H_0, por lo que la varianza de los errores es constante (hay homocedasticidae). Además, esto se puede observar en el gráfico, pues la varianza fluctúa dentro de un rango de variabilidad sin importar el punto del gráfico.

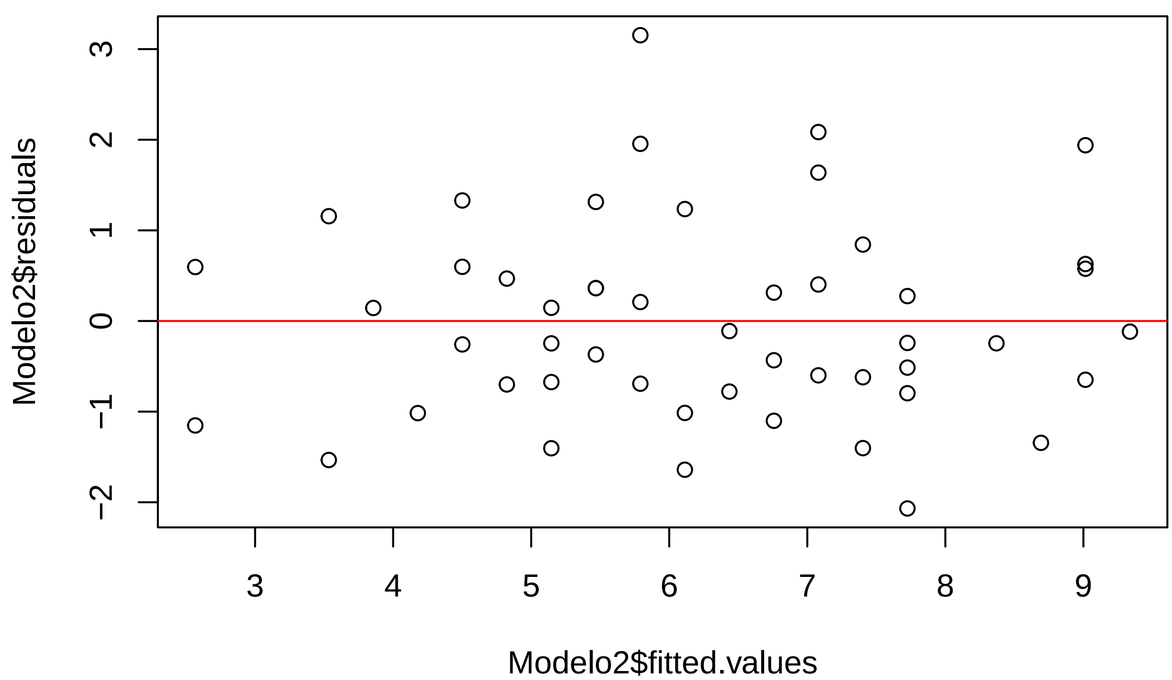
Independencia

H_0: La autocorrelación de los residuos es 0 (hay independencia). H_1: La autocorrelación de los residuos $\neq 0$ (no hay independencia).

```r
dwtest(Modelo2)
```

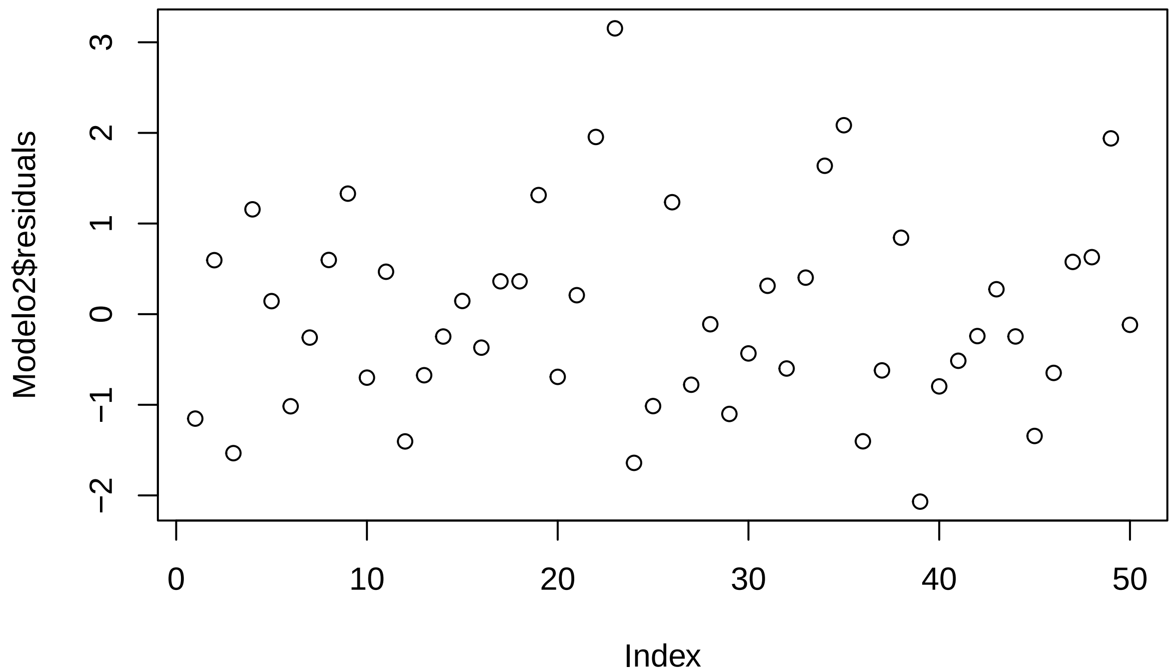
```
##
##  Durbin-Watson test
##
## data:  Modelo2
## DW = 1.9417, p-value = 0.3609
## alternative hypothesis: true autocorrelation is greater than 0
```

```r
plot(Modelo2$fitted.values,Modelo2$residuals)
abline(h=0, col= 'red')
```



Como el valor p > 0.03, no se rechaza H_0, por lo que sí hay independencia en los residuos. Además, no se observa un patrón en la gráfica, o una dependencia entre los residuos. ### Linealidad

```r
plot(Modelo2$residuals)
```



Observamos que los residuos se distribuyen aleatoriamente en torno a cero, por lo que la relación lineal es adecuada.

Conclusión

El modelo tiene un coeficiente de determinación bastante fuerte y se ajusta adecuadamente. Además, cumple todos los supuestos, por lo que este modelo es adecuado.

Despeja la distancia del modelo lineal obtenido entre la transformación y la velocidad. Obtendrás el modelo no lineal que relaciona la distancia con la velocidad directamente (y no con su transformación).

```r
intercepto <- coef(Modelo2)[1]
pendiente <- coef(Modelo2)[2]

cat("Modelo lineal obtenido:\n")
```

```
## Modelo lineal obtenido:
```

```r
cat("dist^0.5=", round(intercepto, 4), "+", round(pendiente, 4), "* speed\n")
```

```
## dist^0.5= 1.2771 + 0.3224 * speed
```

```r
cat("\nEl modelo no lineal que relaciona la distancia con la velocidad es:\n")
```

```
##
## El modelo no lineal que relaciona la distancia con la velocidad es:
```

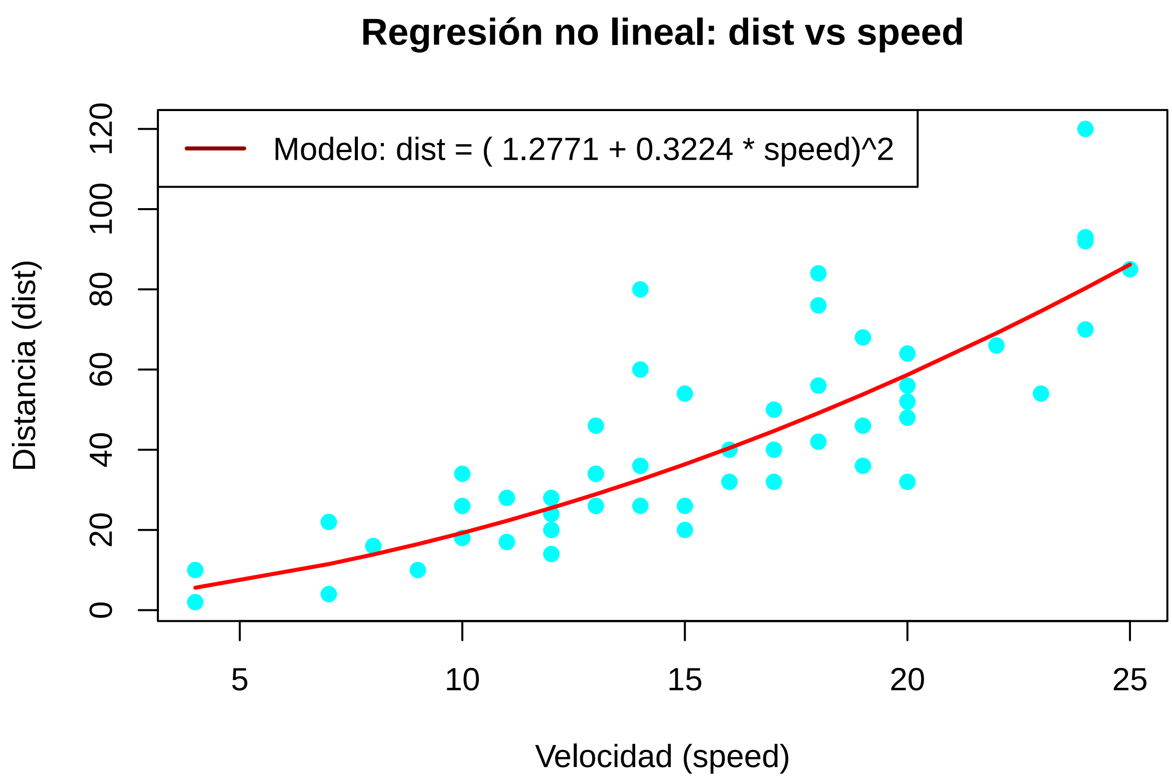
```r
cat("dist = (", round(intercepto, 4), "+", round(pendiente, 4), "* speed)^2 \n")
```

```
## dist = ( 1.2771 + 0.3224 * speed)^2
```

Como podemos observar, el modelo lineal obtenido es tal que $dist^{0.5} = 1.2771 + 0.3224 * speed$. Entonces, el modelo no lineal de distancia vs velocidad: $dist = (1.2771 + 0.3224 \cdot speed)^2$.

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

```r
intercepto <- coef(Modelo2)[1]
pendiente <- coef(Modelo2)[2]
dist_predicha <- (intercepto + pendiente * cars$speed)^2 - 1

plot(cars$speed, X_original,
     main = "Regresión no lineal: dist vs speed",
     xlab = "Velocidad (speed)",
     ylab = "Distancia (dist)",
     pch = 19, col = "cyan")
lines(cars$speed, dist_predicha, col = "red", lwd = 2)
legend("topleft", legend = paste("Modelo: dist = (", round(intercepto, 4), "+",
                                 round(pendiente, 4), "* speed)^2 "),
       col = "darkred", lwd = 2)
```



Comenta sobre la idoneidad del modelo en función de su significancia y validez.

Como se había comentado previamente, el modelo y sus coeficientes son estadísticamente significativos, por lo que la velocidad es predictor relevante para la distancia. Además hay una buena correlación positiva del 0.8422 y un coeficiente de determinación del 70.94, por lo que el modelo explica el 70.94% de la variabilidad de la distancia, lo cual es bueno. Finalmente, el modelo cumple con todos los supuestos, por lo que es adecuado y se ajusta de manera razonable a la variabilidadd de la distancia.

Parte 4

Define cuál de los dos modelos analizados (Punto 1 o Punto 2) es el mejor modelo para describir la relación entre la distancia y la velocidad.

El mejor modelo es del segundo punto, pues cumple con todos los supuestos y obtiene una fuerte correlación y un mejor coeficiente de determinación, logrando explicar el 70.94% de la variabilidad de la distancia, por lo que la transformación de box-cox con el modelo aproximado fue exitosa.

Comenta sobre posibles problemas del modelo elegido (datos atípicos, alejamiento de los supuestos, dificultad de cálculo o interpretación)

Ciertos problemas del modelo del punto 2, puede ser la interpretación, pues el modelo no lineal resultante y las transformaciones de box-cox pueden complicar la interpretación de los resultados obtenidos. De la misma manera, puede llegar a ser sensible a los cambios realizados en los datos, pues las transformaciones nuevamente realizadas puede que no cumplan todos los supuestos, principalmente la de normalidad, la cual puede llegar a ser la más díficil de validar.