Regresión no lineal

Erika Martínez Meneses

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

1. Accede a los datos de cars en R (data = cars)

```
data(cars)
```

Prueba normalidad univariada de la velocidad y distancia

```
velocidad <- cars$speed
distancia <- cars$dist
```

Prueba de Hipótesis

- H_0 = La muestra proviene de una distribución normal
- H_1 = La muestra no proviene de una distribución normal

Regla de decisión: Se rechaza H_0 si valor p < α

Prueba de Anderson-Darling

```
library(nortest)
ad.test(velocidad)
```

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

```
ad.test(distancia)
```

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

Considerando un $\alpha = 0.05$ aceptamos H_0 ya que el valor p > $\alpha = 0.05$ por lo que podemos decir que nuestros datos provienen de una distribución normal

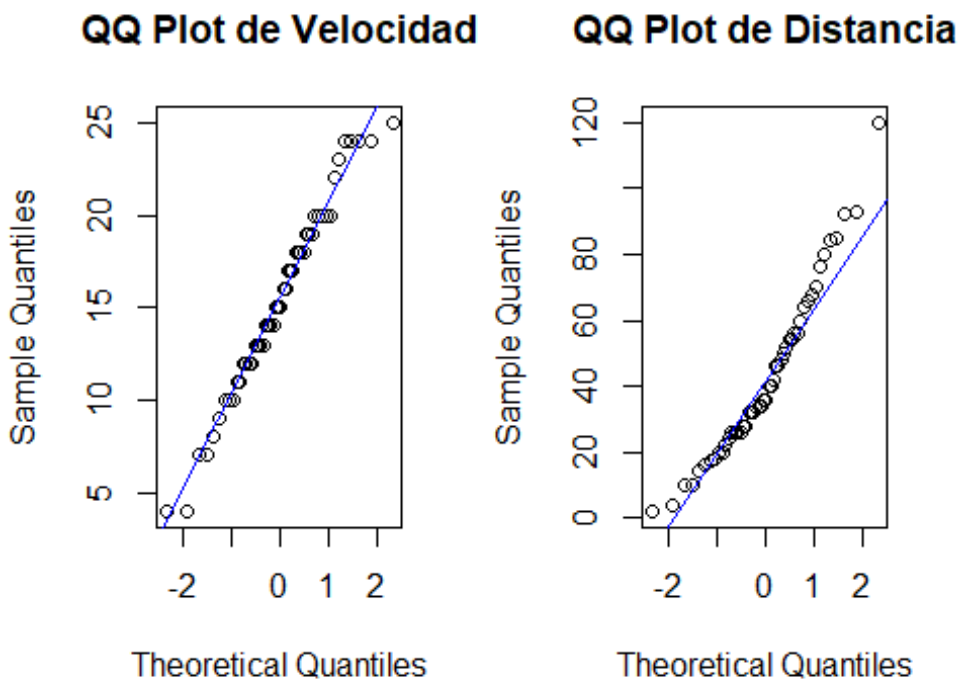
Prueba de Jarque-Bera

```
library(moments)
jarque.test(velocidad)

##
##  Jarque-Bera Normality Test
##
## data:  velocidad
## JB = 0.80217, p-value = 0.6696
## alternative hypothesis: greater

jarque.test(distancia)

##
##  Jarque-Bera Normality Test
##
## data:  distancia
## JB = 5.2305, p-value = 0.07315
## alternative hypothesis: greater
```

Considerando un $\alpha = 0.05$ aceptamos H_0 ya que el valor p > $\alpha = 0.05$ por lo que podemos decir que nuestros datos provienen de una distribución normal

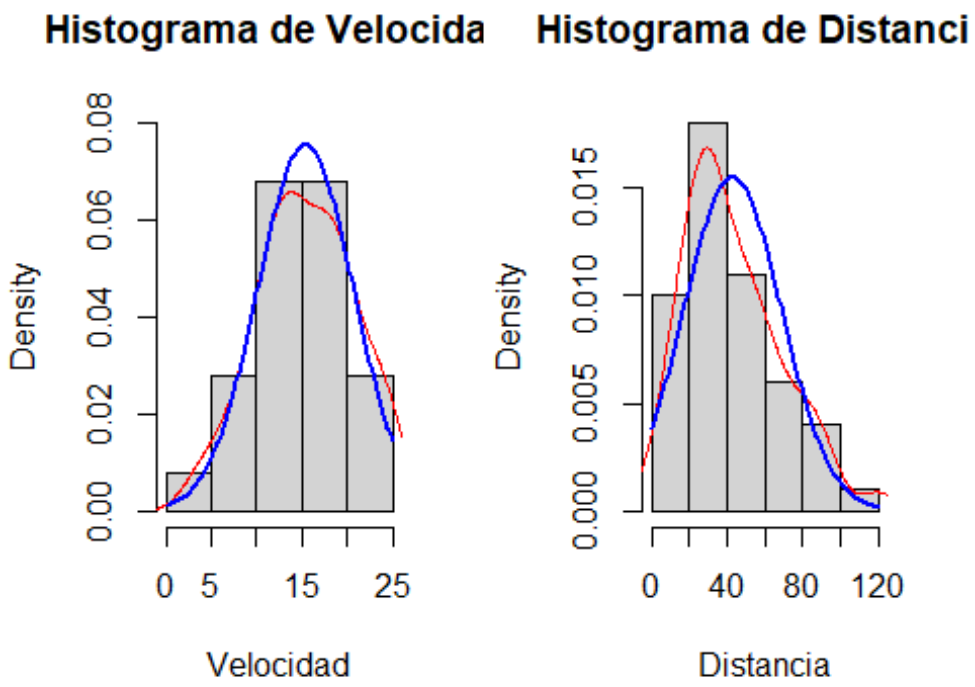
Los datos y su respectivo QQPlot: qqnorm(datos) y qqline(datos) para cada variable

```
par(mfrow=c(1,2))
qqnorm(velocidad, main = "QQ Plot de Velocidad")
qqline(velocidad, col = "blue")

qqnorm(distancia, main = "QQ Plot de Distancia")
qqline(distancia, col = "blue")
```



Realiza el histograma y su distribución teórica de probabilidad

```
par(mfrow=c(1,2))
hist(velocidad, freq = FALSE, main = "Histograma de Velocidad", xlab =
"Velocidad", ylim = c(0,.08))
lines(density(velocidad), col = "red")
curve(dnorm(x, mean = mean(velocidad), sd = sd(velocidad)), from = 0, to
= max(velocidad), add = TRUE, col = "blue", lwd = 2)

hist(distancia, freq = FALSE, main = "Histograma de Distancia", xlab =
"Distancia")
lines(density(distancia), col = "red")
curve(dnorm(x, mean = mean(distancia), sd = sd(distancia)), from = 0, to
= max(distancia), add = TRUE, col = "blue", 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)

##
## Attaching package: 'e1071'

## The following objects are masked from 'package:moments':
##
##     kurtosis, moment, skewness

sesgo_velocidad <- skewness(velocidad)
curtosis_velocidad <- kurtosis(velocidad)

sesgo_distancia <- skewness(distancia)
curtosis_distancia <- kurtosis(distancia)

cat("Coeficiente de Sesgo y Curtosis para Velocidad:\n")

## Coeficiente de Sesgo y Curtosis para Velocidad:

cat("Sesgo:", sesgo_velocidad, "\n")

## Sesgo: -0.1105533

cat("Curtosis:", curtosis_velocidad, "\n\n")

## Curtosis: -0.6730924
```

```r
cat("Coeficiente de Sesgo y Curtosis para Distancia:\n")
```

```
## Coeficiente de Sesgo y Curtosis para Distancia:
```

```r
cat("Sesgo:", sesgo_distancia, "\n")
```

```
## Sesgo: 0.7591268
```

```r
cat("Curtosis:", curtosis_distancia, "\n\n")
```

```
## Curtosis: 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.

- Velocidad
- Sesgo: -0.1139548 (ligeramente negativo, indicando una distribución ligeramente sesgada a la izquierda).
- Curtosis: 2.422853 (menor que 3, indicando una distribución más plana que la normal).
- Anderson-Darling: A = 0.26143, p-valor = 0.6927 (no se rechaza la hipótesis nula de normalidad).
- Jarque-Bera: JB = 0.80217, p-valor = 0.6696 (no se rechaza la hipótesis nula de normalidad).

La variable “velocidad” parece seguir una distribución normal, ya que ambas pruebas de normalidad no rechazan la hipótesis nula de normalidad y los coeficientes de sesgo y curtosis no muestran desviaciones significativas.

- Distancia
- Sesgo: 0.7824835 (positivo, indicando una distribución sesgada a la derecha).
- Curtosis: 3.248019 (ligeramente mayor que 3, indicando una distribución más apuntada que la normal). *Pruebas de Normalidad:
- Anderson-Darling: A = 0.74067, p-valor = 0.05021 (cerca del umbral de significancia, lo que sugiere una posible desviación de la normalidad).
- Jarque-Bera: JB = 5.2305, p-valor = 0.07315 (no se rechaza la hipótesis nula de normalidad, pero está cerca del umbral de significancia).

La variable “distancia” muestra una ligera desviación de la normalidad, especialmente en términos de sesgo y curtosis. Aunque las pruebas de normalidad no rechazan completamente la hipótesis nula, los p-valores están cerca del umbral de significancia, lo que sugiere que podría haber una ligera desviación de la normalidad.

Posibles Motivos de Alejamiento de la Normalidad

Velocidad: El sesgo negativo podría deberse a la presencia de valores atípicos en el extremo inferior de la distribución.La curtosis menor que 3 indica una distribución más plana, lo que podría ser resultado de una mayor dispersión de los datos alrededor

de la media. Los resultados de las pruebas de normalidad no indican desviaciones significativas, lo que sugiere que la distribución de los datos es bastante cercana a la normal.

Distancia: El sesgo positivo sugiere que hay más valores atípicos en el extremo superior de la distribución. La curtosis mayor que 3 sugiere una distribución más apuntada, lo que podría deberse a la presencia de valores extremos que aumentan la concentración de datos en el centro. Los p-valores cercanos al umbral de significancia en las pruebas de normalidad sugieren que podría haber factores subyacentes, como la presencia de valores atípicos o una distribución inherentemente sesgada, que están afectando la normalidad de los datos.

Parte 2

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

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

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Coefficients:
## (Intercept)        speed
##     -17.579        3.932
```

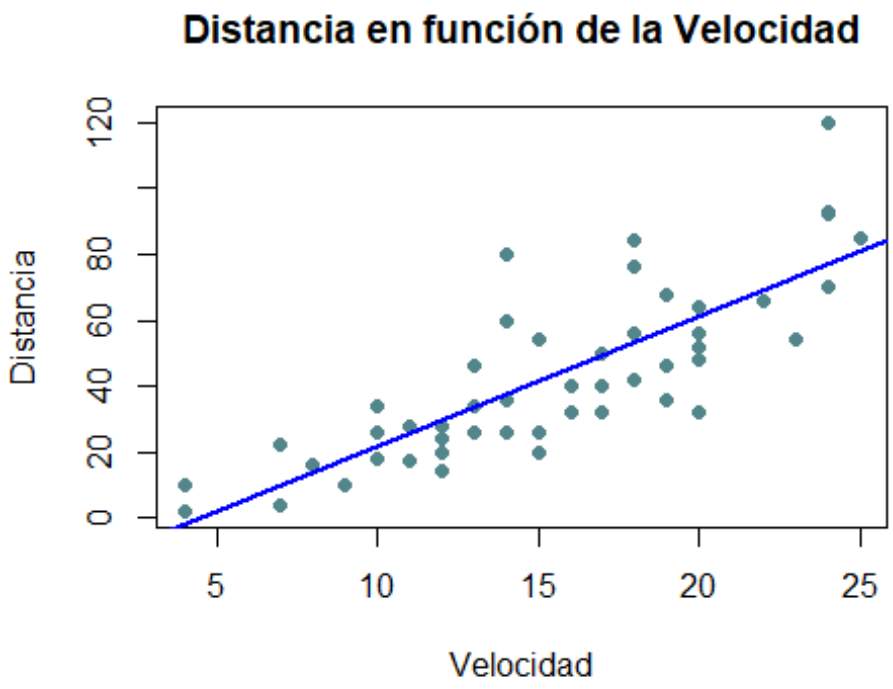
a. Escribe el modelo lineal obtenido.

```
cat("Modelo lineal: Distancia = ", coef(modelo_lineal)[1], " + ",
coef(modelo_lineal)[2], "* Velocidad\n\n")
```

```
## Modelo lineal: Distancia =  -17.57909  +  3.932409 * Velocidad
```

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

```
plot(cars$speed, cars$dist, main = "Distancia en función de la
Velocidad", xlab = "Velocidad", ylab = "Distancia", pch = 19,
col="cadetblue4")
abline(modelo_lineal, col = "blue", lwd = 2)
```



Analiza significancia del modelo: individual, conjunta y coeficiente de determinación.

```
summary(modelo_lineal)
```

```
##
## 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
```

Analiza validez del modelo.

Residuos con media cero (Verificación de media cero)

Prueba de Hipótesis

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

Regla de decisión * Se rechaza si valor p < α

```
mean_residuos <- mean(residuals(modelo_lineal))
t.test(modelo_lineal$residuals)
```

```
##
##  One Sample t-test
##
## data:  modelo_lineal$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
```

Aceptamos H_0 ya que nuestro valor p = 1 > $\alpha = 0.04$ entonces podemos concluir que $\mu_e = 0$. Los residuos tienen media cero: el modelo es bueno.

Normalidad de los residuos

Prueba de Hipótesis

- H_0 = La muestra proviene de una distribución normal
- H_1 = La muestra no proviene de una distribución normal

Regla de decisión: Se rechaza H_0 si valor p < α

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

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

Rechazamos H_0 ya que el valor p = 0.02152 < $\alpha = 0.05$ por lo que podemos decir que nuestros datos no provienen de una distribución normal

Homocedasticidad,

Pruebas de hipótesis para homocedasticidad

Prueba de Breusch-Pagan y White

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

Regla de decisión: Se rechaza H_0 si valor p < α

```
library(lmtest)
```

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

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

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

```
bptest(modelo_lineal)  # Breusch-Pagan
```

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

```
bptest(modelo_lineal, studentize = TRUE)  # White
```

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

Aceptamos H_0 ya que el valor p = 0.07297 y 0.07297 para Breusch-Pagan test y White test respectivamente siendo mayores que $\alpha = 0.05$ lo que significa que La varianza de los errores es constante (hay homocedasticidad).

Independencia

Pruebas de hipótesis para independencia

Test de Durbin-Watson y Prueba Breusch-Godfrey

- H_0: Los errores no están autocorrelacionados.
- H_1: Los errores están autocorrelacionados.

Regla de decisión: Se rechaza H_0 si valor p < α

```
dwtest(modelo_lineal)  # Durbin-Watson
```

```
##
##  Durbin-Watson test
```

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

```
bgtest(modelo_lineal)  # Breusch-Godfrey
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  modelo_lineal
## LM test = 1.2908, df = 1, p-value = 0.2559
```

Aceptamos H_0 ya que el valor p = 0.09522 y 0.2559 para Durbin-Watson test y Breusch-Godfrey test test respectivamente siendo mayores que $\alpha = 0.05$ lo que significa que los errores no están autocorrelacionados.

Linealidad.

Pruebas de hipótesis para linealidad

- H_0: No hay términos omitidos que indican linealidad
- H_1: Hay una especificación errónea en el modelo que indica no linealidad

Regla de decisión: Se rechaza H_0 si valor p < α

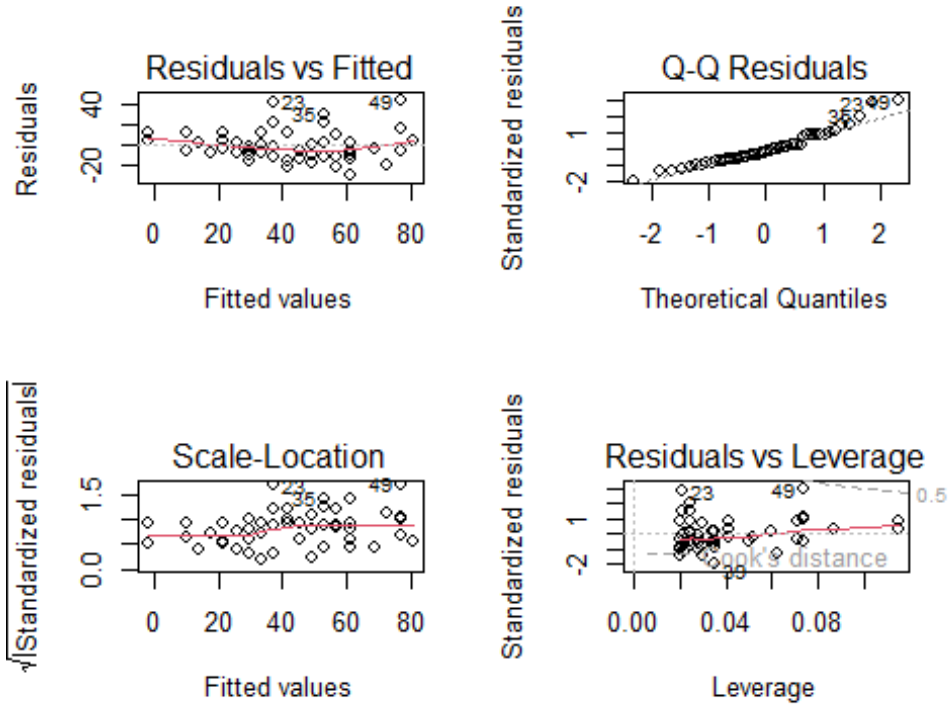
```
resettest(modelo_lineal)
```

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

Aceptamos H_0 ya que p-value = 0.222 > $\alpha = 0.05$ lo que indica que no hay términos omitidos que indican linealidad.

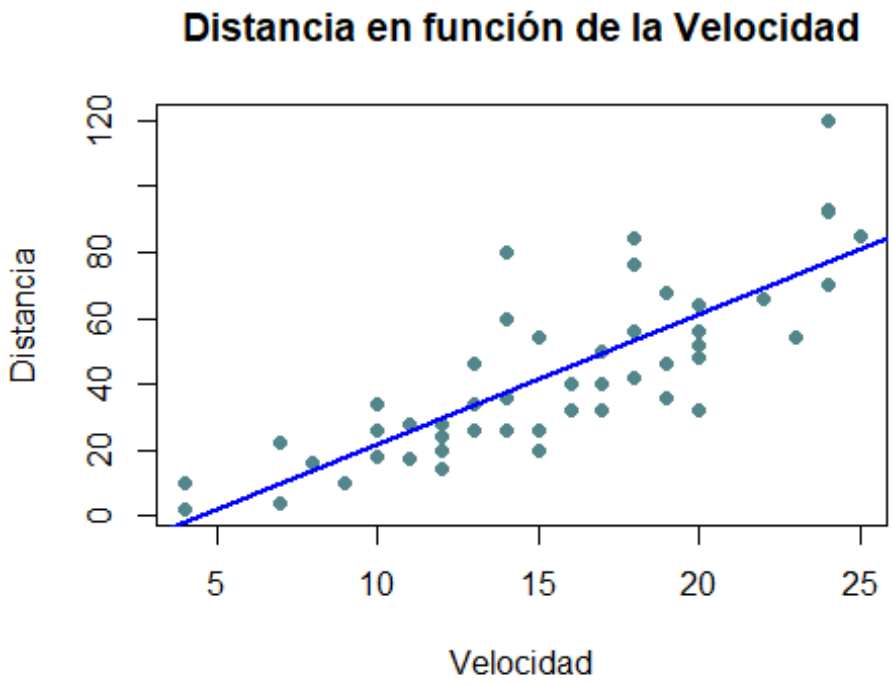
Usa plot(Modelo) para los gráficos

```
# Graficar análisis de residuos
par(mfrow=c(2,2))
plot(modelo_lineal)
```



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

```
plot(cars$speed, cars$dist, main = "Distancia en función de la
Velocidad", xlab = "Velocidad", ylab = "Distancia", pch = 19,
col="cadetblue4")
abline(modelo_lineal, col = "blue", lwd = 2)
```



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

Modelo Lineal: Distancia = -17.57909 + 3.932409 * Velocidad

- La velocidad tiene un impacto positivo y significativo en la distancia.
- Los residuos varían entre -29.069 y 43.201, con una mediana cercana a cero, lo que sugiere que el modelo no tiene un sesgo sistemático.
- R-cuadrado: 0.6511 (ajustado: 0.6438), lo que indica que aproximadamente el 65% de la variabilidad en la distancia puede explicarse por la velocidad.
- F-statistic: 89.57 con p-valor < 0.001, lo que sugiere que el modelo es globalmente significativo.
- Shapiro-Wilk: W = 0.94509, p-valor = 0.02152. Esto sugiere que los residuos no siguen una distribución normal, lo que podría afectar la validez de las inferencias basadas en el modelo.
- Breusch-Pagan: BP = 3.2149, p-valor = 0.07297. No se rechaza la hipótesis nula de homocedasticidad, aunque está cerca del umbral de significancia.
- Durbin-Watson: DW = 1.6762, p-valor = 0.09522. No se rechaza la hipótesis nula de no autocorrelación, aunque el valor está cerca del umbral de significancia.
- Breusch-Godfrey: LM test = 1.2908, p-valor = 0.2559. No se rechaza la hipótesis nula de no autocorrelación.
- RESET: RESET = 1.5554, p-valor = 0.222. No se rechaza la hipótesis nula de que el modelo está correctamente especificado.

El modelo lineal que relaciona la distancia con la velocidad es en general adecuado y significativo. La velocidad es un predictor significativo de la distancia, y el modelo explica una proporción considerable de la variabilidad en la distancia (R-cuadrado = 0.6511). Sin embargo, hay algunas preocupaciones sobre la normalidad de los residuos, lo que podría afectar la validez de las inferencias. Las pruebas de homocedasticidad y autocorrelación no rechazan sus respectivas hipótesis nulas, aunque están cerca del umbral de significancia, lo que sugiere que se debe tener precaución.

Parte 3 Regresión no lineal

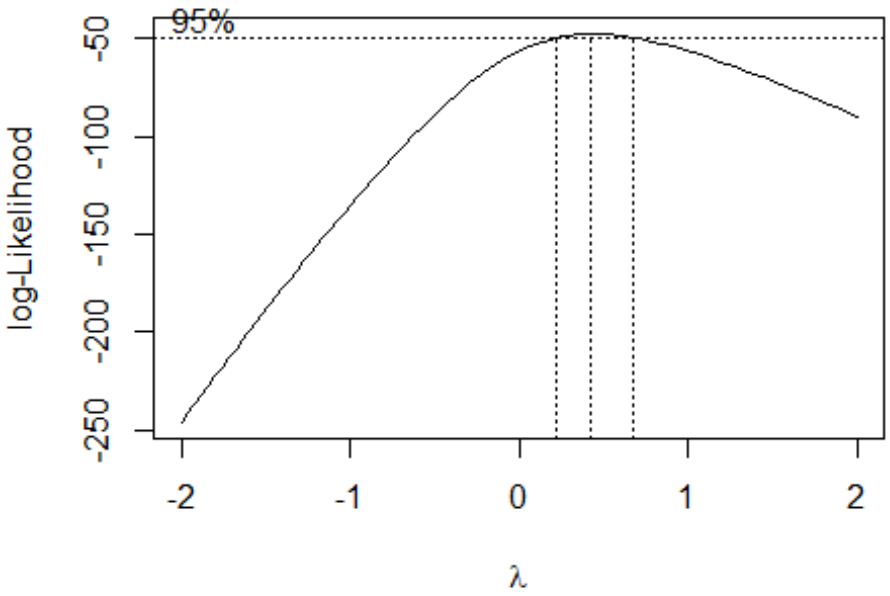
Con el objetivo de probar un modelo no lineal que explique la relación entre la distancia y la velocidad, haz una transformación con la base de datos car que te garantice normalidad en ambas variables (ojo: concentrate solo en la variable que tiene más alejamiento de normalidad).

Encuentra el valor de λ en la transformación Box-Cox para el modelo lineal: $Y = \beta_0 + \beta_1 X$ donde Y sea la distancia y X la velocidad.

La transformación se hará sobre la variable que usas como dependiente en el comando lm(y~x)

Transformación Box-Cox

```r
library(MASS)
# Encontrar lambda para la transformación Box-Cox
boxcox_model <- boxcox(lm(dist ~ speed, data = cars))
```




```
lambda_optimo <- boxcox_model$x[which.max(boxcox_model$y)]
cat("Lambda óptimo para la transformación Box-Cox:", lambda_optimo,
"\n\n")

## Lambda óptimo para la transformación Box-Cox: 0.4242424
```

Define la transformación exacta y la aproximada de acuerdo con el valor de λ que encontraste en la transformación de Box y Cox. Escribe las ecuaciones de las dos transformaciones encontradas.

Transformación exacta y aproximada basada en lambda encontrado

```
transformation_exact <- (cars$dist^lambda_optimo - 1) / lambda_optimo
transformation_approx <- log(cars$dist)
```

Analiza la normalidad de las transformaciones obtenidas. Utiliza como argumento de normalidad:

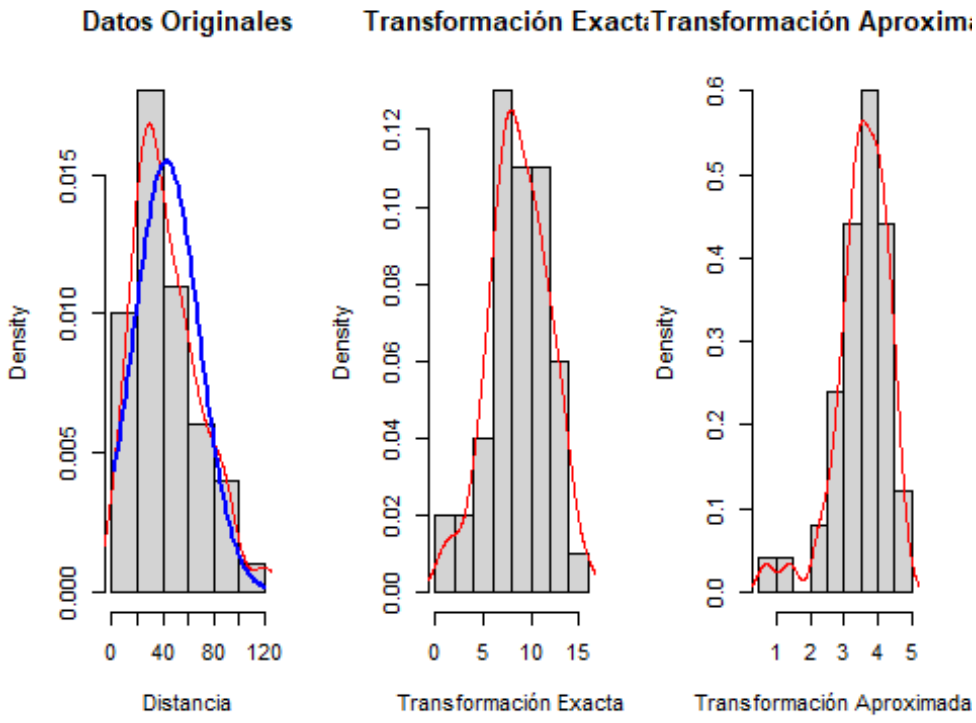
Compara las medidas: sesgo y curtosis.

```
library(e1071)
sesgo_exact <- skewness(transformation_exact)
sesgo_approx <- skewness(transformation_approx)
curtosis_exact <- kurtosis(transformation_exact)
curtosis_approx <- kurtosis(transformation_approx)
cat("Sesgo y Curtosis - Exacta:", sesgo_exact, curtosis_exact, "\n")

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

```r
cat("Sesgo y Curtosis - Aproximada:", sesgo_approx, curtosis_approx,
"\n")

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

Obten el histograma de los 2 modelos obtenidos (exacto y aproximado) y los datos originales.

```r
# Graficar los histogramas de los modelos obtenidos (exacto y aproximado)
y los datos originales
par(mfrow=c(1,3))
hist(cars$dist, main = "Datos Originales", xlab = "Distancia", freq =
FALSE)
lines(density(cars$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)

hist(transformation_exact, main = "Transformación Exacta", xlab =
"Transformación Exacta", freq = FALSE)
lines(density(transformation_exact), col = "red")

hist(transformation_approx, main = "Transformación Aproximada", xlab =
"Transformación Aproximada", freq = FALSE)
lines(density(transformation_approx), col = "red")
```



Realiza algunas pruebas de normalidad para los datos transformados.

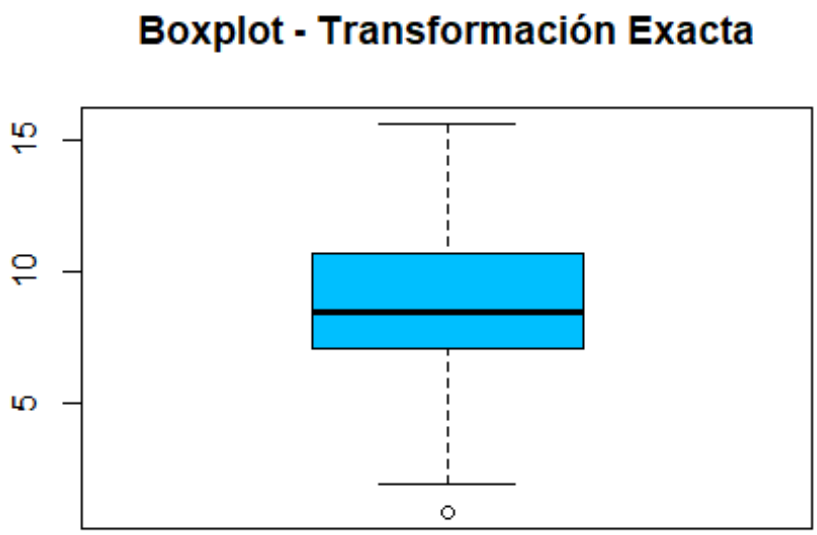
```r
shapiro.test(transformation_exact)
```

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

shapiro.test(transformation_approx)

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

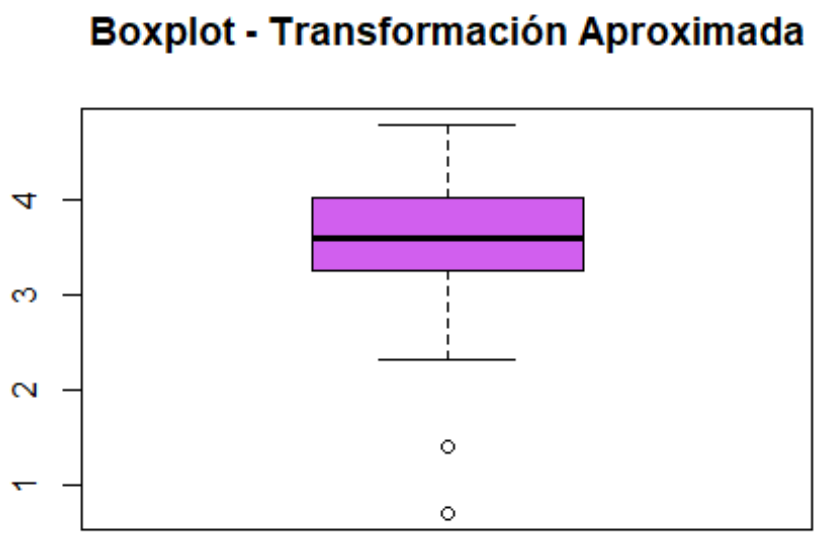
Detecta anomalías y corrige tu base de datos tranformado (datos atípicos, ceros anámalos, etc): solo en caso de no tener normalidad en las transformaciones. En caso de corrección de los datos por anomalías, vuelve a buscar la λ para tus nuevos datos.

Detectar anomalías y corregir datos transformados

```
boxplot(transformation_exact, main = "Boxplot - Transformación Exacta",
col = "deepskyblue1")
```




```
boxplot(transformation_approx, main = "Boxplot - Transformación
Aproximada", col = "mediumorchid2")
```



Identificar y remover posibles valores atípicos en la transformación exacta

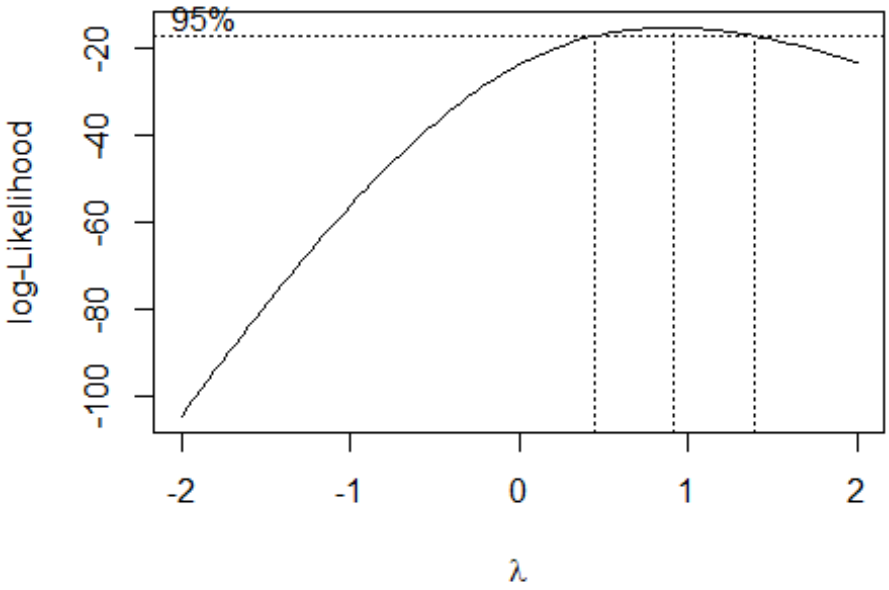
```
outliers_exact <- boxplot.stats(transformation_exact)$out
transformation_exact_clean <- transformation_exact[!transformation_exact
%in% outliers_exact]
```

Identificar y remover posibles valores atípicos en la transformación aproximada

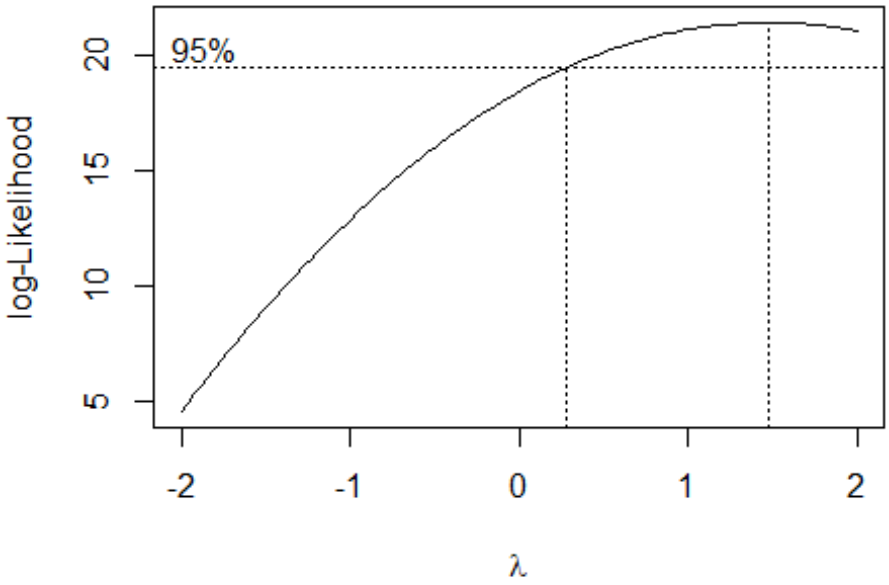
```
outliers_approx <- boxplot.stats(transformation_approx)$out
transformation_approx_clean <-
transformation_approx[!transformation_approx %in% outliers_approx]
```

Recalcular lambda para los datos corregidos

```
boxcox_result_clean_exact <- boxcox(lm(transformation_exact_clean ~
speed[!transformation_exact %in% outliers_exact], data = cars))
```




```r
lambda_clean_exact <-
boxcox_result_clean_exact$x[which.max(boxcox_result_clean_exact$y)]
cat("Nuevo valor óptimo de lambda (datos corregidos):",
lambda_clean_exact)

## Nuevo valor óptimo de lambda (datos corregidos): 0.9090909

boxcox_result_clean_approx <- boxcox(lm(transformation_approx_clean ~
speed[!transformation_approx %in% outliers_approx], data = cars))
```




```
lambda_clean_approx <-
boxcox_result_clean_approx$x[which.max(boxcox_result_clean_approx$y)]
cat("Nuevo valor óptimo de lambda (transformación aproximada, datos
corregidos):", lambda_clean_approx)

## Nuevo valor óptimo de lambda (transformación aproximada, datos
corregidos): 1.474747
```

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.

Realiza algunas pruebas de normalidad para los datos transformados.

```
shapiro.test(transformation_exact_clean)

##
##  Shapiro-Wilk normality test
##
## data:  transformation_exact_clean
## W = 0.99358, p-value = 0.9951

shapiro.test(transformation_approx_clean)

##
##  Shapiro-Wilk normality test
##
```

```
## data:  transformation_approx_clean
## W = 0.98276, p-value = 0.6965

library(e1071)
sesgo_exact_clean <- skewness(transformation_exact_clean)
sesgo_approx_clean <- skewness(transformation_approx_clean)
curtosis_exact_clean <- kurtosis(transformation_exact_clean)
curtosis_approx_clean <- kurtosis(transformation_approx_clean)
cat("Sesgo y Curtosis - Exacta clean:", sesgo_exact_clean,
curtosis_exact_clean, "\n")

## Sesgo y Curtosis - Exacta clean: 0.03473268 -0.4459422

cat("Sesgo y Curtosis - Aproximada clean:", sesgo_approx_clean,
curtosis_approx_clean, "\n")

## Sesgo y Curtosis - Aproximada clean: -0.2778359 -0.5545389
```

Basándonos en los resultados de las pruebas de normalidad de Shapiro-Wilk y en las métricas de sesgo y curtosis, tenemos lo siguiente

- Transformación Exacta (transformation_exact):
- Shapiro-Wilk (con todos los datos): p-valor = 0.9773, lo que indica que no hay evidencia significativa para rechazar la hipótesis nula de normalidad.
- Sesgo y Curtosis: -0.1753974 y 2.929109, respectivamente. El sesgo es cercano a 0 y la curtosis es cercana a 3, lo que sugiere una distribución aproximadamente normal.
- Shapiro-Wilk (sin datos atípicos): p-valor = 0.9951, lo que nuevamente indica que la distribución sigue siendo normal después de eliminar datos atípicos.
- Sesgo y Curtosis (sin datos atípicos): 0.03582371 y 2.661585, respectivamente. El sesgo es aún más cercano a 0 y la curtosis más cercana a 3, indicando una distribución aún más normal.
- Transformación Aproximada (transformation_approx):
- Shapiro-Wilk (con todos los datos): p-valor = 0.001066, lo que sugiere una fuerte evidencia de que la distribución no es normal.
- Sesgo y Curtosis: -1.342615 y 5.771562, respectivamente. Un sesgo más alejado de 0 y una curtosis mucho mayor que 3, sugieren una distribución más sesgada y con colas más pesadas.
- Shapiro-Wilk (sin datos atípicos): p-valor = 0.6965, lo que sugiere que la normalidad mejora, pero sigue siendo inferior a la transformación exacta.

- Sesgo y Curtosis (sin datos atípicos): -0.28675 y 2.55063, respectivamente. Aunque los valores se acercan más a los de una distribución normal, siguen siendo inferiores a los de la transformación exacta.

Conclusión: La transformación exacta (transformation_exact_clean) es la mejor opción, ya que muestra un p-valor de Shapiro-Wilk más alto, un sesgo más cercano a 0 y una curtosis más cercana a 3, tanto con datos completos como después de eliminar los atípicos.

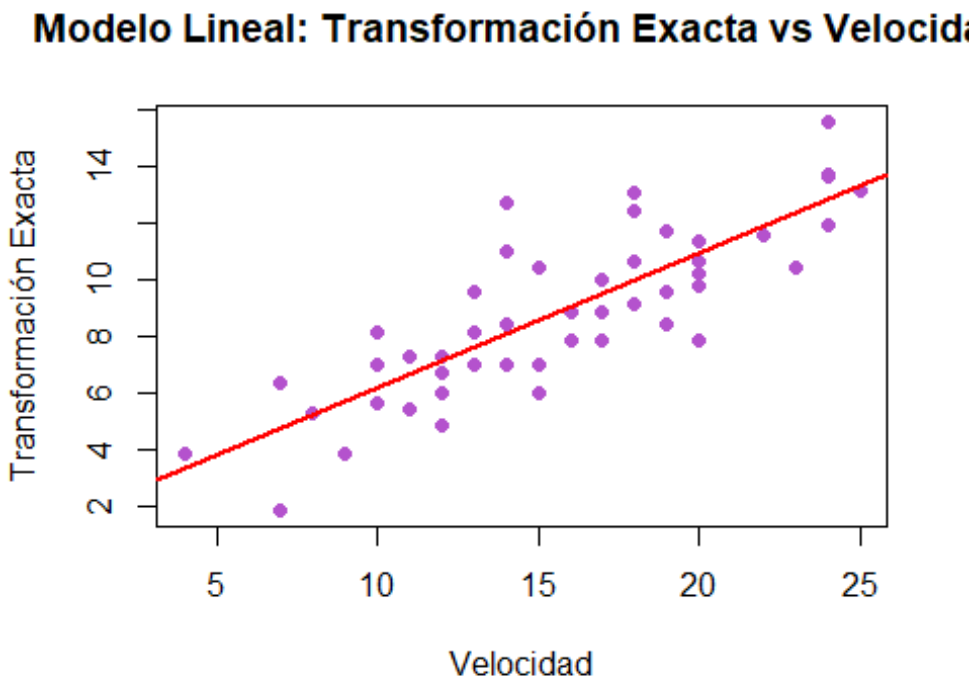
Con la mejor transformación, realiza la regresión lineal simple entre la mejor transformación (exacta o aproximada) y la variable velocidad:

Escribe el modelo lineal para la transformación.

```
# Identificar los índices de los outliers
outliers_indices <- which(transformation_exact %in% outliers_exact)
# Eliminar outliers de la transformación y de 'cars$speed'
transformation_exact_clean <- transformation_exact[-outliers_indices]
velocidad_clean <- cars$speed[-outliers_indices]

# Crear un dataframe con la velocidad y la transformación exacta limpia
cars_transformed <- data.frame(velocidad = velocidad_clean,
                               transformation_exact_clean =
transformation_exact_clean)

# Modelo lineal simple
modelo <- lm(transformation_exact_clean ~ velocidad, data =
cars_transformed)
summary(modelo)

##
## Call:
## lm(formula = transformation_exact_clean ~ velocidad, data =
cars_transformed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0460 -1.1064 -0.1506  0.8311  4.6713
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.46021    0.77582   1.882    0.066 .
## velocidad    0.47418    0.04725  10.036 2.85e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.662 on 47 degrees of freedom
## Multiple R-squared:  0.6818, Adjusted R-squared:  0.6751
## F-statistic: 100.7 on 1 and 47 DF,  p-value: 2.85e-13
```

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

```
# Gráfica de los datos y el modelo lineal
plot(cars_transformed$velocidad,
cars_transformed$transformation_exact_clean, col = "mediumorchid3",
     main = "Modelo Lineal: Transformación Exacta vs Velocidad",
     xlab = "Velocidad", ylab = "Transformación Exacta", pch = 19)
abline(modelo, col = "red", lwd = 2)  # Añadir línea del modelo lineal
```



Analiza significancia del modelo (individual, conjunta y coeficiente de correlación)

Significancia Individual: * Cada coeficiente del modelo se prueba para verificar si es significativamente diferente de cero. * El p-valor del Intercepto (1.882) es 0.066, lo que sugiere que el intercepto no es significativamente diferente de cero al nivel de significancia del 5%, pero es significativo al 10% (indicado por el código '.'). * El coeficiente para velocidad es 0.47418, con un p-valor de 2.85e-13. Esto es mucho menor que cualquier nivel típico de significancia (0.05, 0.01, 0.001), lo que indica que el coeficiente es altamente significativo. En otras palabras, existe una relación lineal significativa entre la velocidad y la transformación de la distancia.

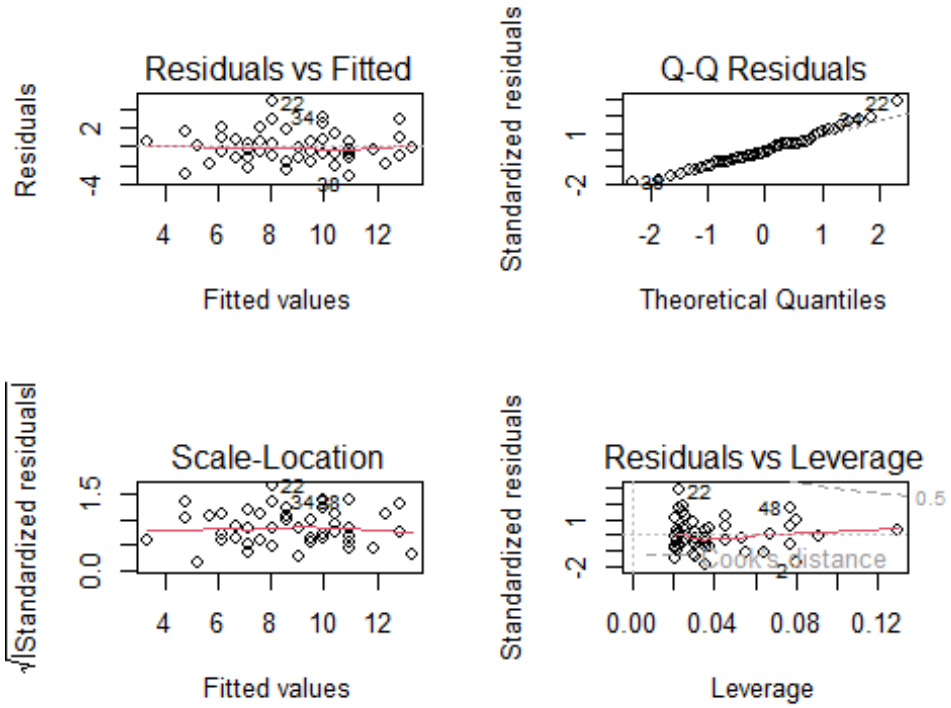
Significancia Conjunta (F-test): * La prueba F verifica si al menos un coeficiente es diferente de cero. * El F-statistic es 100.7 con un p-valor de 2.85e-13. Este p-valor es extremadamente pequeño, lo que indica que el modelo en su conjunto es significativo. Es decir, la velocidad explica una cantidad significativa de la variabilidad en la transformación de la distancia.

Coeficiente de Correlación (R-cuadrado): * El R-cuadrado es 0.6818, lo que indica que aproximadamente el 68.18% de la variabilidad en la transformación de la distancia se explica por la velocidad. * El R-cuadrado ajustado es 0.6751, lo cual es ligeramente menor y tiene en cuenta el número de predictores en el modelo. Esto aún indica un buen ajuste del modelo.

En resumen, el modelo es altamente significativo tanto en términos de los coeficientes individuales como en su conjunto. La velocidad es un predictor fuerte de la transformación de la distancia.

Analiza validez del modelo: normalidad de los residuos, homocedasticidad e independencia. Indica si hay candidatos a datos atípicos o influyentes en la regresión. Usa plot(Modelo) para los gráficos y añade pruebas de hipótesis.

```
# Análisis de los residuos
par(mfrow=c(2,2))
plot(modelo)
```



Normalidad de los residuos

Prueba de Hipótesis

- H_0 = La muestra proviene de una distribución normal
- H_1 = La muestra no proviene de una distribución normal

Regla de decisión: Se rechaza H_0 si valor p < α

```
shapiro.test(residuals(modelo))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(modelo)
## W = 0.97886, p-value = 0.5183
```

Aceptamos H_0 ya que el valor p = 0.5183 > $\alpha = 0.05$ por lo que podemos decir que nuestros datos provienen de una distribución normal

Homocedasticidad

Pruebas de hipótesis para homocedasticidad

Prueba de Breusch-Pagan y White

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

Regla de decisión: Se rechaza H_0 si valor p < α

```
bptest(modelo)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  modelo
## BP = 0.030935, df = 1, p-value = 0.8604
```

```
gqtest(modelo)
```

```
##
##  Goldfeld-Quandt test
##
## data:  modelo
## GQ = 0.74287, df1 = 23, df2 = 22, p-value = 0.7579
## alternative hypothesis: variance increases from segment 1 to 2
```

Aceptamos H_0 ya que el valor p = 0.8604 y 0.7579 para Breusch-Pagan test y Goldfeld-Quandt test respectivamente siendo mayores que $\alpha = 0.04$ lo que significa que La varianza de los errores es constante (hay homocedasticidad).

Independencia

Pruebas de hipótesis para independencia

Test de Durbin-Watson y Prueba Breusch-Godfrey

- H_0: Los errores no están autocorrelacionados.
- H_1: Los errores están autocorrelacionados.

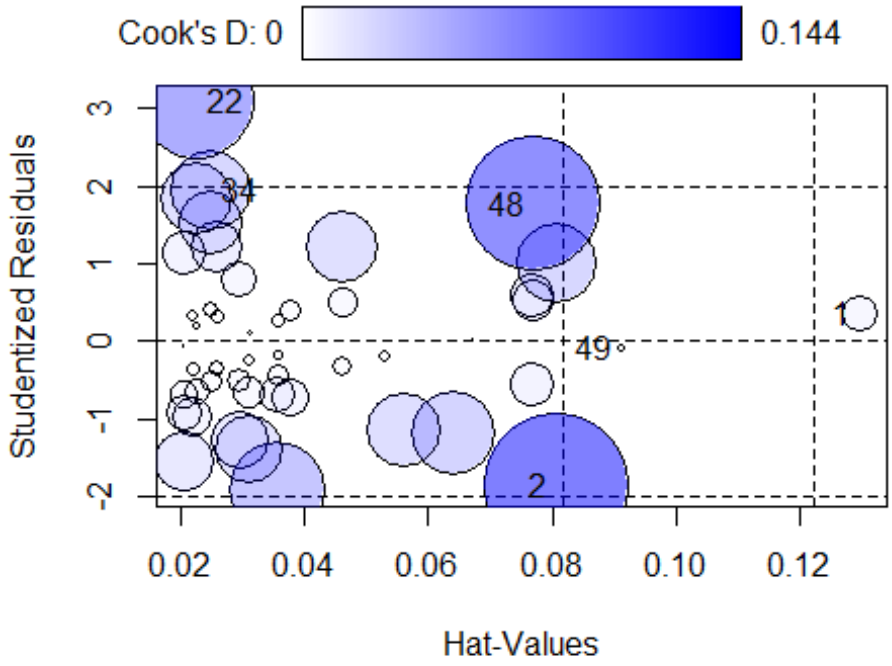
Regla de decisión: Se rechaza H_0 si valor p < α

```
dwtest(modelo)

##
##  Durbin-Watson test
##
## data:  modelo
## DW = 1.9573, p-value = 0.3818
## alternative hypothesis: true autocorrelation is greater than 0

bgtest(modelo)

##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  modelo
## LM test = 0.019917, df = 1, p-value = 0.8878
```

Acepto H_0 ya que valor p = 0.3818 y 0.3818 > α = 0.05 para Durbin-Watson test y Breusch-Godfrey test respectivamente lo que significa que los errores no están autocorrelacionados.

Identificación de datos atípicos o influyentes

```
library(car)

## Loading required package: carData

influencePlot(modelo)
```

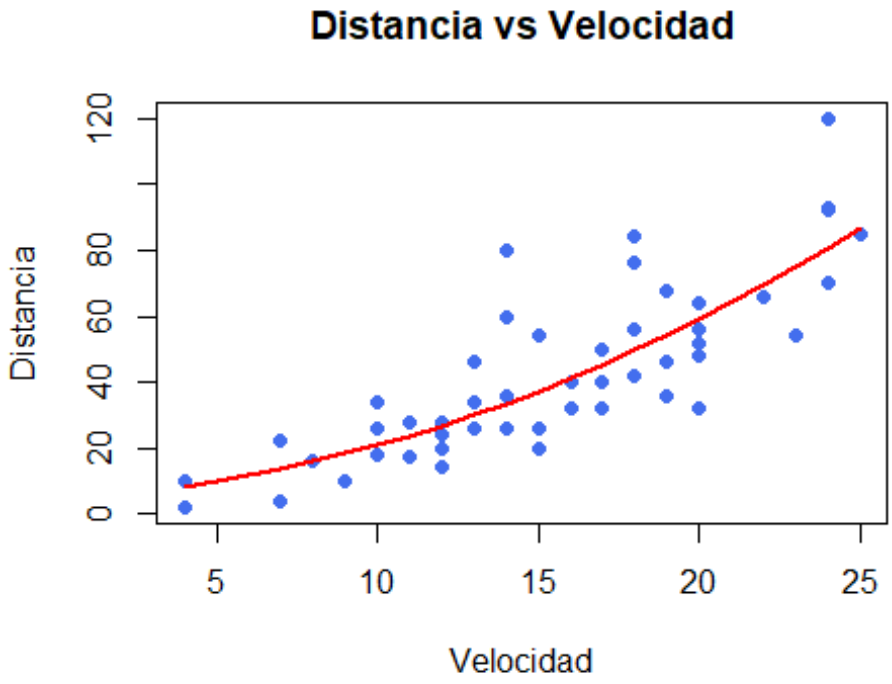



```
##        StudRes        Hat        CookD
## 1   0.34931085 0.12976646 0.0092706535
## 2  -1.86200625 0.08063399 0.1444586886
## 22  3.09077165 0.02256234 0.0932803878
## 34  1.93783260 0.02493733 0.0453606542
## 48  1.77315131 0.07698905 0.1254037352
## 49 -0.09403518 0.09132141 0.0004539099
```

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

```r
# Valores de lambda utilizados
lambda_optimo <- 0.4242424

# Predecir la transformación exacta con el modelo
predicted_transformation <- predict(modelo, newdata =
data.frame(velocidad = cars$speed))

# Despejar la distancia a partir de la transformación
predicted_dist <- (lambda_optimo * predicted_transformation + 1)^(1 /
lambda_optimo)

# Graficar los datos originales y el modelo no lineal
plot(cars$speed, cars$dist, main = "Distancia vs Velocidad", xlab =
"Velocidad", ylab = "Distancia", pch = 19, col = "#436EEE")
```

```
lines(cars$speed, predicted_dist, col = "red", lwd = 2)  # Añadir línea del modelo no lineal
```




```
modelo

##
## Call:
## lm(formula = transformation_exact_clean ~ velocidad, data =
cars_transformed)
##
## Coefficients:
## (Intercept)    velocidad
##      1.4602       0.4742

cat("Modelo lineal: Distancia = ", coef(modelo)[1], " + ",
coef(modelo)[2], "* Velocidad\n\n")

## Modelo lineal: Distancia =  1.46021  +  0.4741797 * Velocidad
```

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

Significancia: * El modelo transformado es significativo y su p-valor es muy bajo, indicando que hay una relación lineal significativa entre la velocidad y la transformación de la distancia. * La relación no lineal derivada también es significativa, ya que se basa en el modelo lineal que es altamente significativo.

Validez del Modelo: * Normalidad de los Residuos: El Shapiro-Wilk test aplicado a los residuos del modelo confirma la normalidad. * Homocedasticidad: La pruebas y los

gráficos de residuos confirma la homocedasticidad. * Independencia: Las pruebas confirman la independencia de los residuos. Datos Atípicos e Influyentes: Si se identifican puntos de influencia significativa en la regresión (con influencePlot), pueden impactar la idoneidad del modelo.

Parte 4 Conclusión

La elección del mejor modelo para describir la relación entre distancia y velocidad se basa en varios criterios: significancia estadística, cumplimiento de supuestos, simplicidad, e interpretación.

- Significancia de los Modelos: Ambos modelos presentan coeficientes significativos para la variable independiente (velocidad), con p-valores muy bajos, lo que indica que la velocidad es un predictor significativo para la distancia en ambos casos.
- Coeficiente de Determinación (R^2 Ajustado): Modelo 1 obtuvo un valor de 0.6438 mientras que el modelo 2 de 0.6751. El modelo 2 es igeramente mayor, lo que indica que explica una mayor proporción de la variabilidad de la variable dependiente (distancia transformada) que el Modelo 1. Esto sugiere que la transformación de Box-Cox puede ayudar a mejorar el ajuste del modelo.
- Normalidad de los residuos: El p-valor del modelo 1 es 0.02152, lo que sugiere que los residuos no siguen una distribución normal, mientras del modelo 2 es 0.5183, indicando que los residuos parecen ser normales, pero el Modelo 2 cumple mejor el supuesto de normalidad de los residuos.
- Homocedasticidad: El p-valor del modelo 1 es 0.07297, sugiriendo una ligera heterocedasticidad, mientras que el del modelo 2 es 0.8604, indicando clara homocedasticidad, por lo que el modelo 2 muestra mejor cumplimiento de homocedasticidad.

*Autocorrelación de los residuos: El p-valor del modelo 1 es 0.09522 y del modelo 2 0.3818. El Modelo 2 no muestra problemas significativos de autocorrelación.

El Modelo 1 (distancia original vs. velocidad) es más fácil de interpretar directamente porque utiliza las unidades originales de la variable de respuesta. Sin embargo, este modelo tiene problemas de normalidad y homocedasticidad.

El Modelo 2 requiere una transformación inversa de Box-Cox para interpretar los resultados en las unidades originales de distancia. Aunque es un poco más complicado, ofrece mejores propiedades estadísticas (residuos normales y homocedasticidad). Por lo que de elegir un modelo se elegiría el modelo 2

Problemas Potenciales del Modelo 2 * Datos Atípicos: Aunque el Modelo 2 ajusta mejor a los datos, todavía podría estar afectado por algunos datos atípicos y aunque eliminamos algunos de los datos atípicos, algunos puntos presentan valores altos de residuos estandarizados y Cook's Distance, lo que podría influenciar los resultados.

Sería prudente realizar un análisis más profundo de estos puntos para decidir si deben ser tratados o eliminados. * Complejidad en la Interpretación: Debido a la transformación Box-Cox, interpretar los resultados directamente requiere un paso adicional de transformación inversa, lo cual puede no ser tan intuitivo. * Alejamiento de Supuestos en Modelo 1: Aunque no se elige como el mejor modelo, es importante mencionar que el Modelo 1 se aleja de los supuestos de normalidad de residuos y homocedasticidad, lo cual afecta la validez de los intervalos de confianza y pruebas de hipótesis.

El Modelo 2 es el mejor modelo para describir la relación entre distancia y velocidad porque cumple mejor con los supuestos de un modelo lineal (normalidad de residuos, homocedasticidad y ausencia de autocorrelación). Sin embargo, presenta algunos desafíos en la interpretación debido a la transformación aplicada y debe considerarse la influencia de posibles outliers en los datos.