13_Regresion_no_lineal_fer

Fernanda Pérez

2024-09-10

Parte 1: Análisis de normalidad

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

```
data(cars)
head(cars)
```

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

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

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

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

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

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

```r
ks.test(cars$speed, "pnorm", mean=mean(cars$speed), sd=sd(cars$speed))
```

```
## Warning in ks.test.default(cars$speed, "pnorm", mean = mean(cars$speed), : ties
## should not be present for the one-sample Kolmogorov-Smirnov test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  cars$speed
## D = 0.068539, p-value = 0.9729
## alternative hypothesis: two-sided
```

```r
ks.test(cars$dist, "pnorm", mean=mean(cars$dist), sd=sd(cars$dist))
```

```
## Warning in ks.test.default(cars$dist, "pnorm", mean = mean(cars$dist), sd =
## sd(cars$dist)): ties should not be present for the one-sample
## Kolmogorov-Smirnov test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  cars$dist
## D = 0.12675, p-value = 0.3979
## alternative hypothesis: two-sided
```

Se hizo la prueba de normalidad de Shapiro-Wilk y Kolmogorov-Smirnov para nuestras varaibles: speed y dist.

Por shapiro-wilk:

Al tener un p-value mayor a 0.05 en speed sabemos que la velocidad de los autos la podemos considerar como normalmente distribuida.

Y con dist como el valor p es menor a 0.05 no sigue una distribución normal.

Por kolmogorov-smirnov:

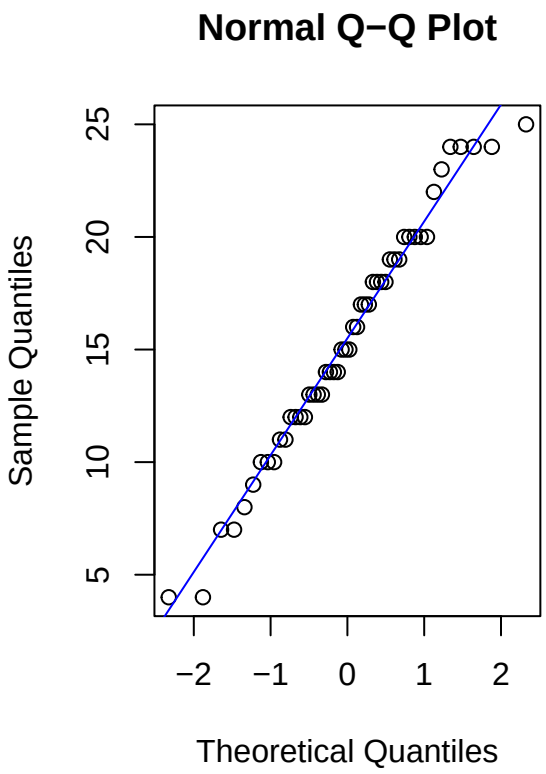
para speed el valor p es muy alto, sigue una distribución normal.

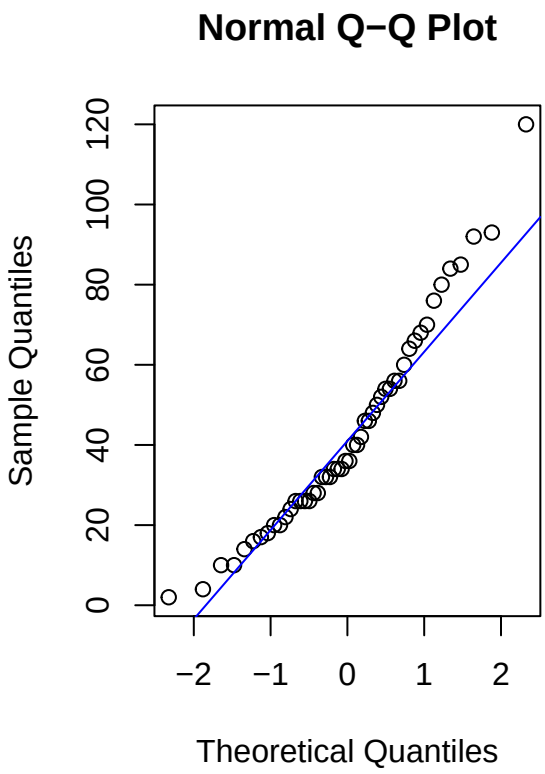
para dist al obtener un valor mayor a 0.05 nos dice que no podemos rechazar la hipotesis nula de normatividad.

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

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

```r
par(mfrow = c(1, 2))
qqnorm(cars$speed)
qqline(cars$speed, col = "blue")
qqnorm(cars$dist)
qqline(cars$dist, col = "blue")
```





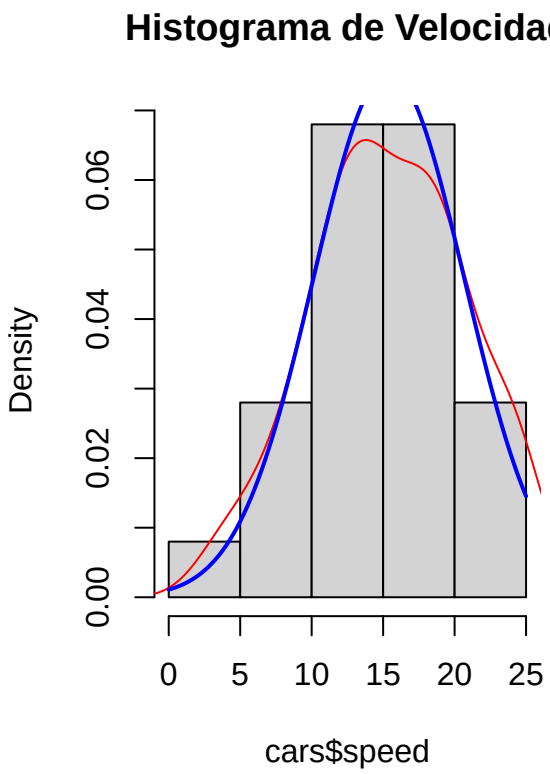
En la grafica de velocidad los puntos si siguen la línea teórica por lo que sigue un comportamient de distribución normal.

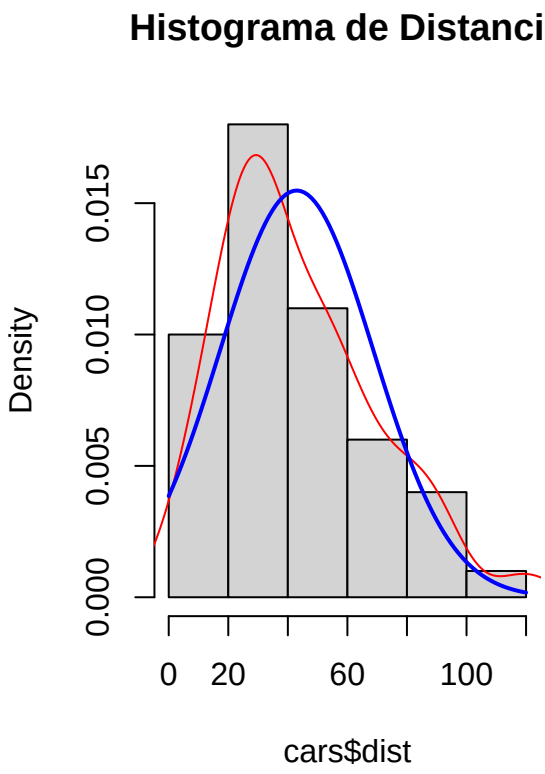
Y en el grafico de distancia la mayoria de los puntos siguen la línea teorica , pero en los valores extremos vemos un alejamiento.

Realiza el histograma y su distribución teórica de probabilidad

```
par(mfrow = c(1, 2))
hist(cars$speed, freq = FALSE, main = "Histograma de Velocidad", col = "lightgray")
lines(density(cars$speed), col = "red")
curve(dnorm(x, mean=mean(cars$speed), sd=sd(cars$speed)), add = TRUE, col = "blue", lwd = 2)

hist(cars$dist, freq = FALSE, main = "Histograma de Distancia", col = "lightgray")
lines(density(cars$dist), col = "red")
curve(dnorm(x, mean=mean(cars$dist), sd=sd(cars$dist)), add = TRUE, col = "blue", lwd = 2)
```





En el histograma de velocidad las curvas teorica y empirica se ajustan bien, dandonos una idea de ser distribución normal.

Y en el histograma de distancia como las curvas empírica y teorica no estan en sintonia sugiere no ser una distribución normal.

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

```
if(!require(e1071)) {
  install.packages("e1071", dependencies = TRUE)
}
```

```
## Cargando paquete requerido: e1071
```

```
library(e1071)

skewness(cars$speed)
```

```
## [1] -0.1105533
```

```
kurtosis(cars$speed)
```

```
## [1] -0.6730924
```

```r
skewness(cars$dist)
```

```
## [1] 0.7591268
```

```r
kurtosis(cars$dist)
```

```
## [1] 0.1193971
```

speed: sesgo cercano a 0, distribución simétrica curtosis: es más plana que la normal

distancia: sesgo: distribución sesgada a la derecha curtosis: cercana a 0, distribución con colas similares a la normal

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.

Para la variable speed de acuerdo a los análisis realizados vemos una distribución normal ya que eso nos indican las pruebas y las gráficas.

Y por otro lado la variable dist vemos como se aleja de la normalidad ya que presenta un sesgo a la derecha y vemos que posibles valores atípicos en las colas, de acuerdo a los análisis y a las gráficas.

Parte 2: Regresión lineal

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

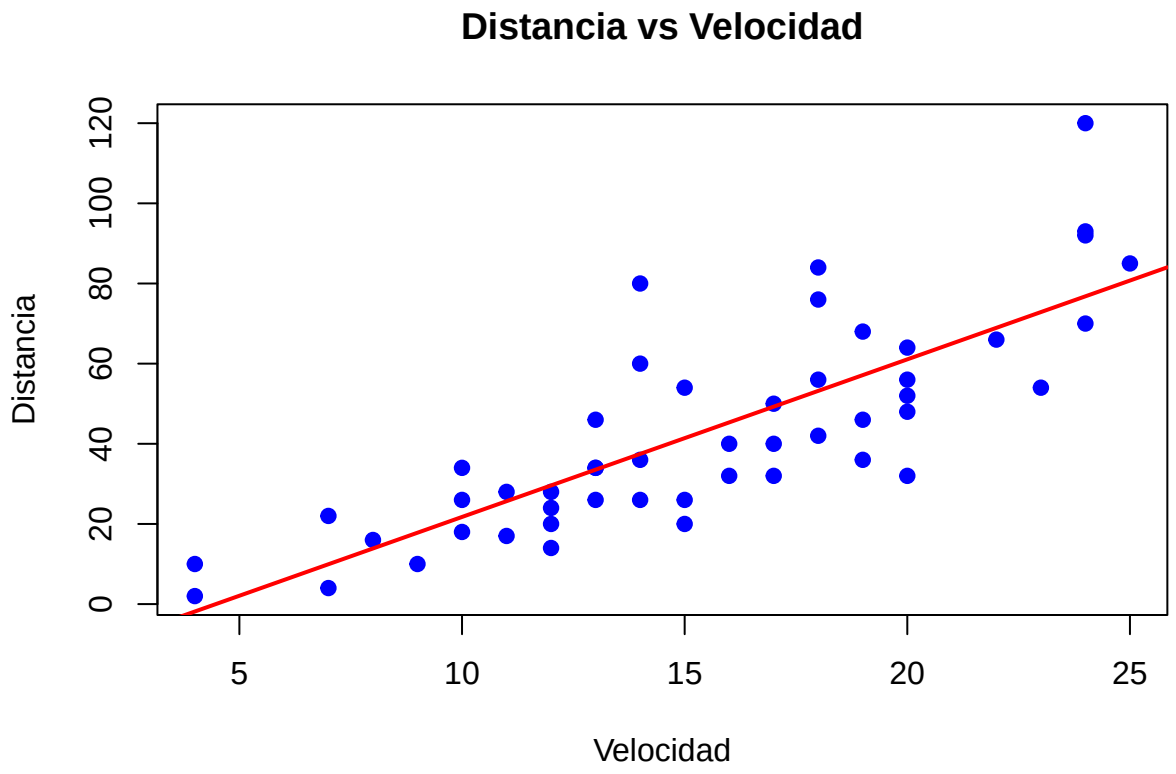
Escribe el modelo lineal obtenido.

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

```
##
## 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
```

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

```
plot(cars$speed, cars$dist, main = "Distancia vs Velocidad",
     xlab = "Velocidad", ylab = "Distancia", pch = 19, col = "blue")
abline(modelo, col = "red", lwd = 2)
```



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

Analiza validez del modelo.

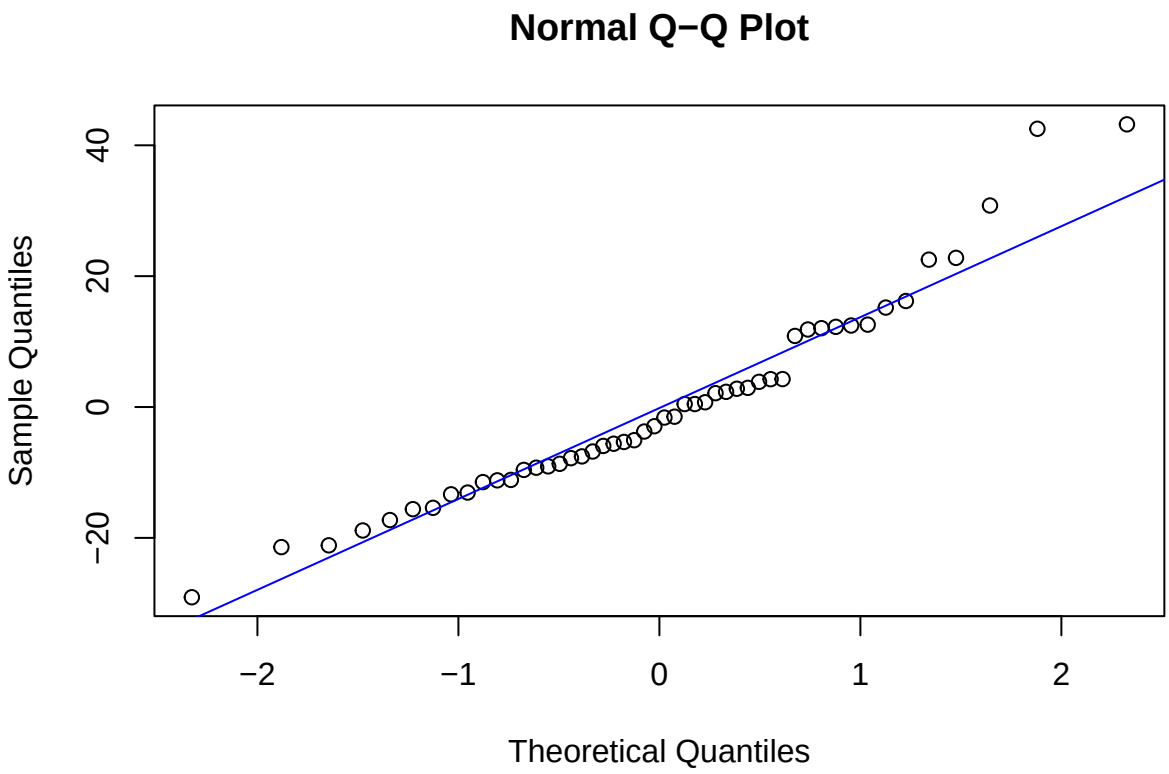
Residuos con media cero

```
mean(residuals(modelo))
```

```
## [1] 2.220446e-16
```

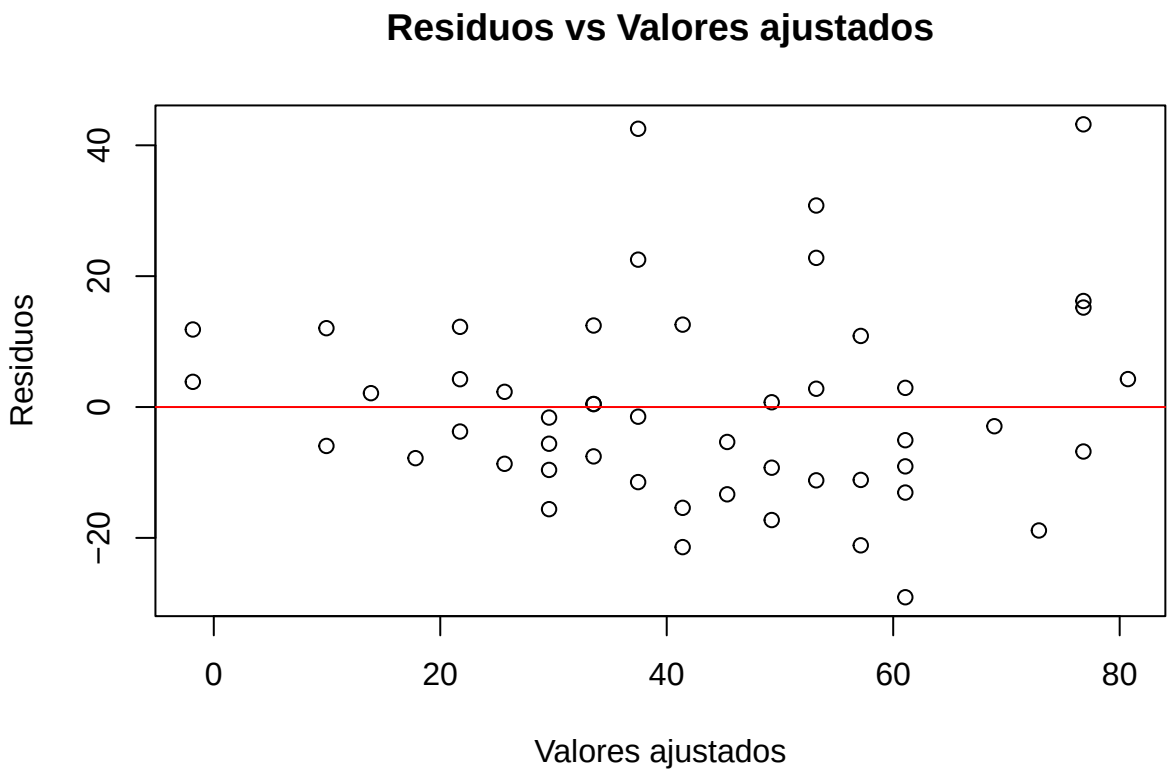
Normalidad de los residuos

```
qqnorm(residuals(modelo))
qqline(residuals(modelo), col = "blue")
```



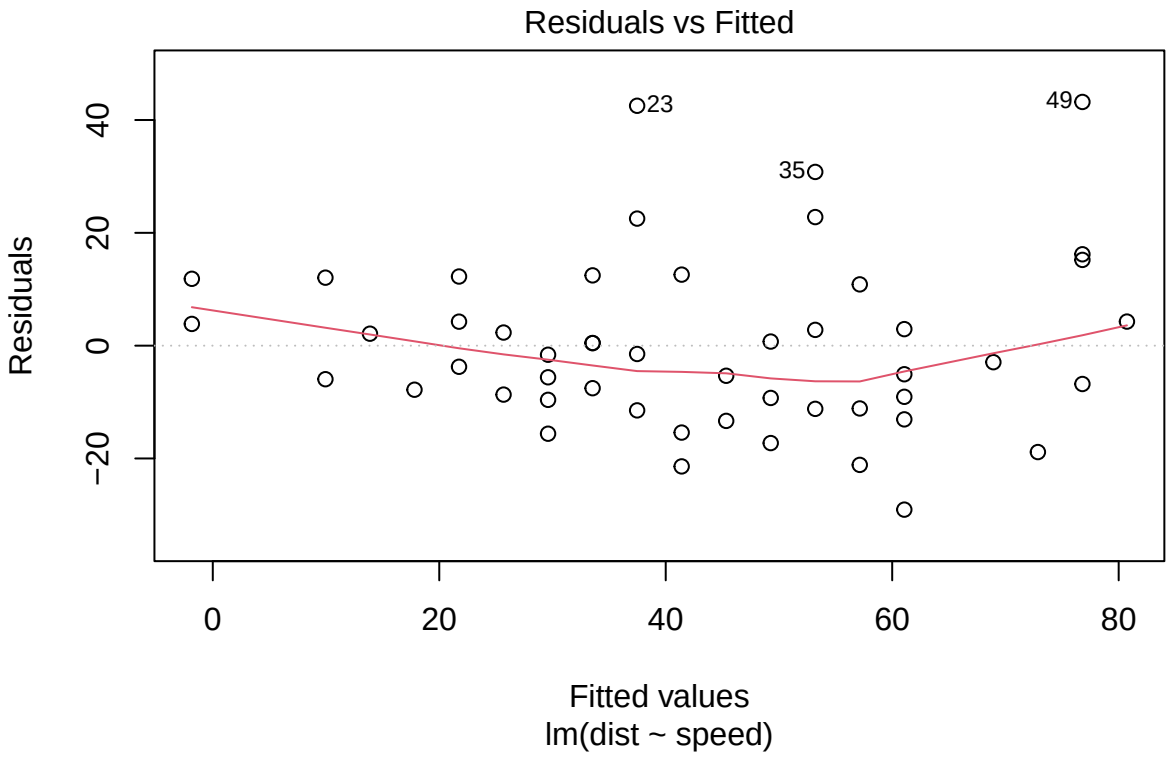
Homocedasticidad, independencia y linealidad.

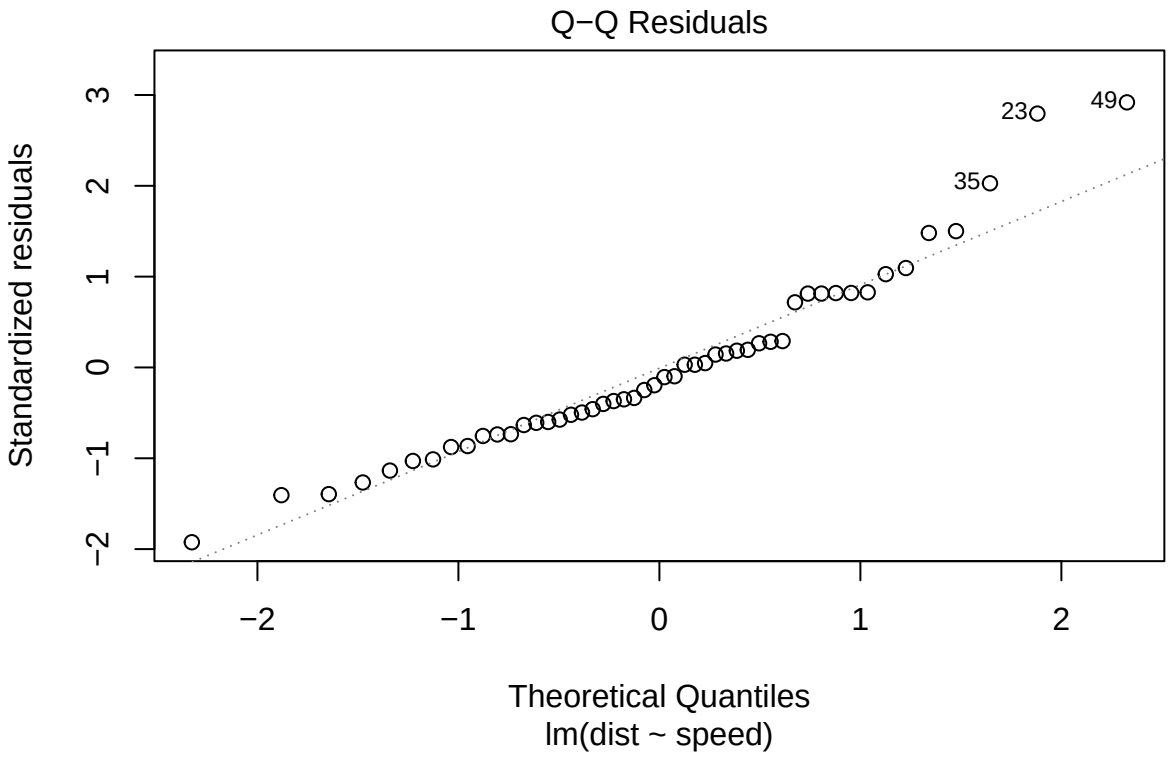
```
plot(fitted(modelo), residuals(modelo),
     main = "Residuos vs Valores ajustados",
     xlab = "Valores ajustados", ylab = "Residuos")
abline(h = 0, col = "red")
```

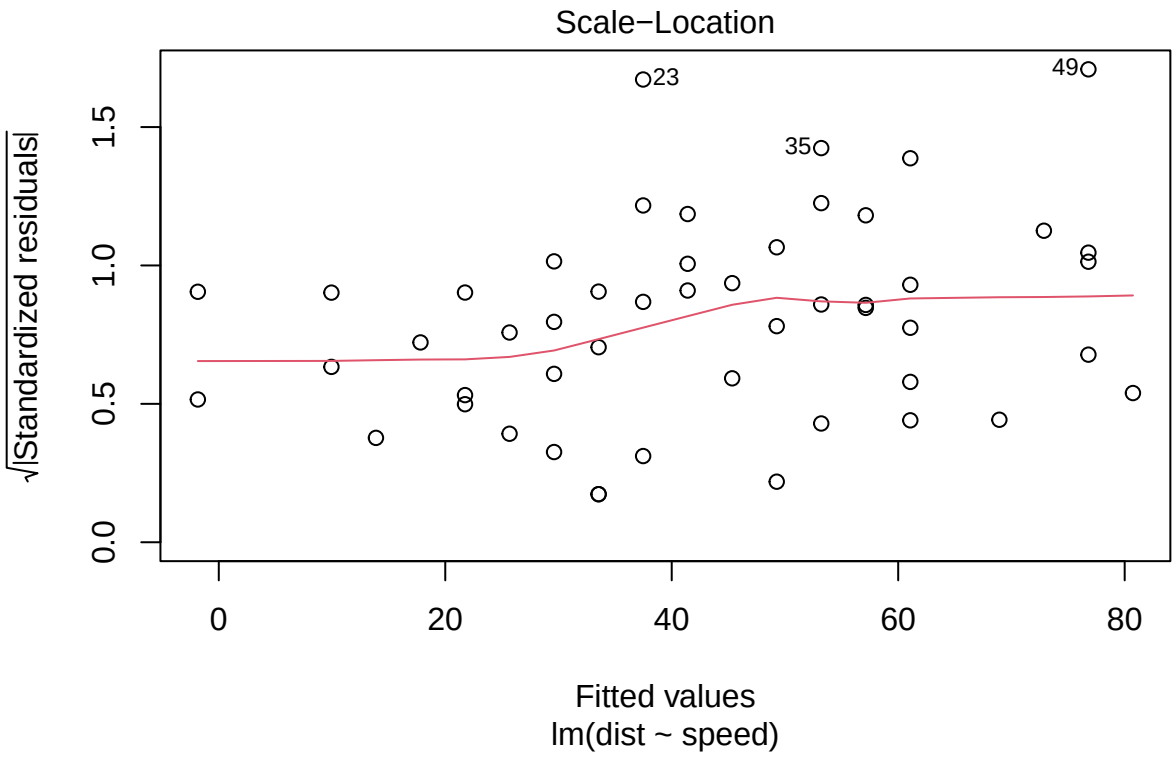


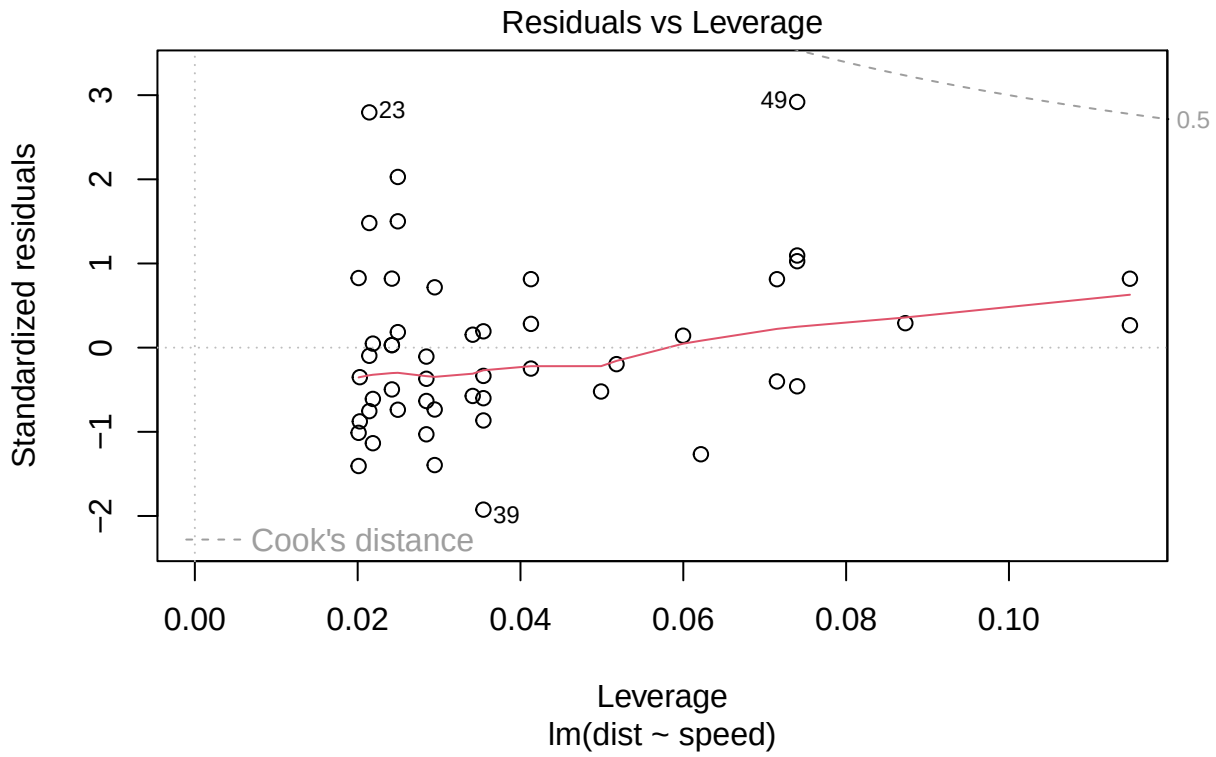
Usa plot(Modelo) para los gráficos y añade pruebas de hipótesis.

```
plot(modelo)
```


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


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


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
23
35
49
0
20
40
60
80
Fitted values
lm(dist ~ speed)

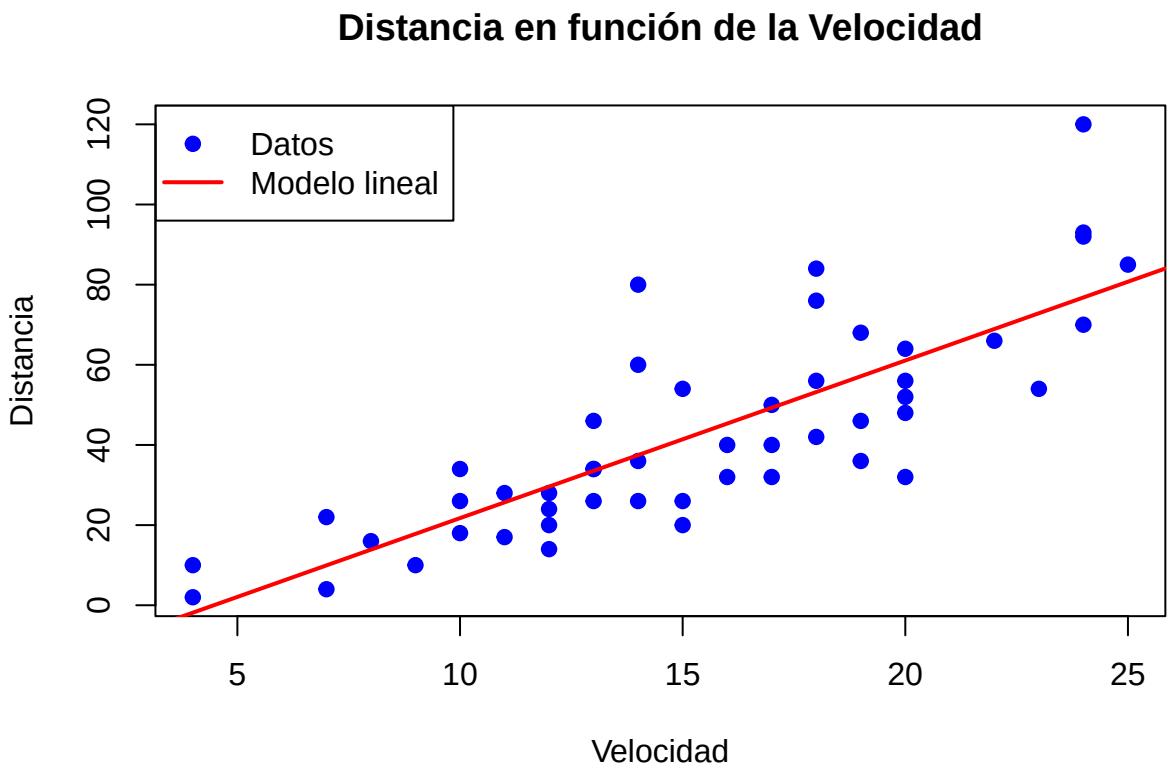


Grafica los datos y el modelo de la distancia en función de la velocidad.

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

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

legend("topleft", legend = c("Datos", "Modelo lineal"),
       col = c("blue", "red"), pch = c(19, NA), lty = c(NA, 1), lwd = c(NA, 2))
```



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

Dado el p-value del modelo es muy bajo, indica que el modelo es altamente significativo.Lo cual significa queexiste una relación significativa entre la velocidad y la distancia.

El R^2 del modelo siendo 0.6511 explica que el 65.11% de la variación en la distancia puede ser explicada por la velocidad.

La media de los resuduos es cercana a 0 lo cual es un buen indicador de la validez del modelo.

El modelo es significativo y si logra explicar una gran parte de la variación en la distancia de frenado.

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).

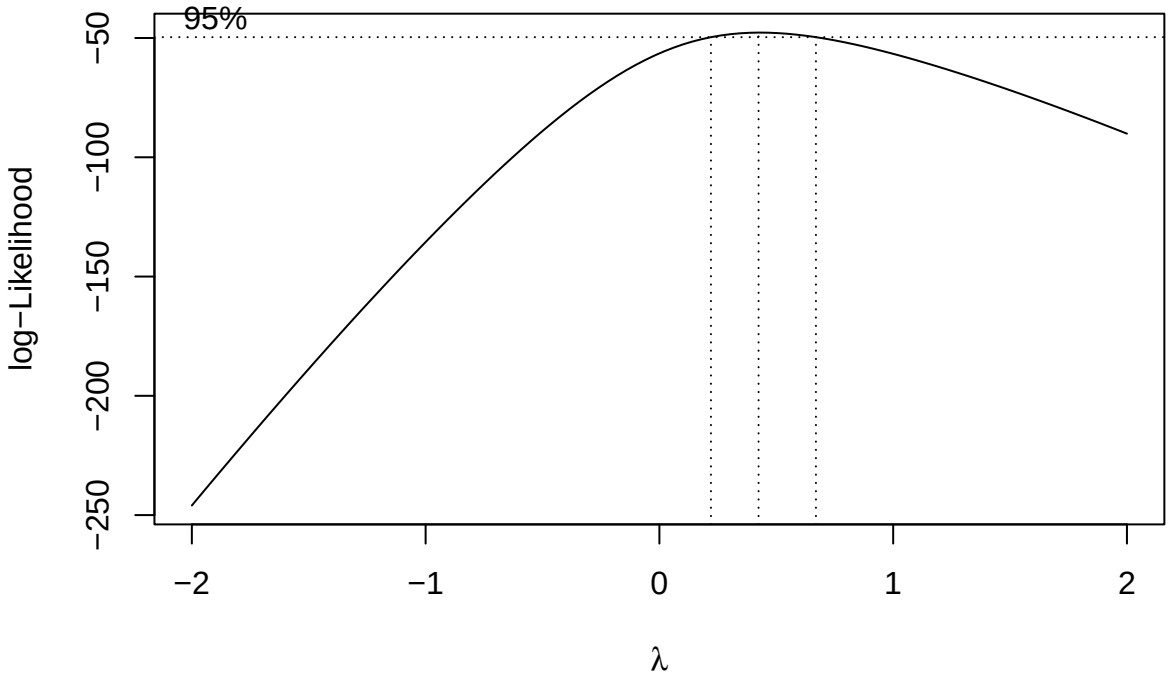
Encuentra el valor de lambda en la transformación Box-Cox para el modelo lineal: Y=B0 + B1X donde Y sea la distancia y X la velocidad. Aprovecha que el comando de boxcox en R te da la oportunidad de trabajar con el modelo líneal:

```
if(!require(MASS)) {
  install.packages("MASS", dependencies = TRUE)
}
```

```
## Cargando paquete requerido: MASS
```

```
library(MASS)
```

```
boxcox_model <- boxcox(lm(dist ~ speed, data = cars),
                       lambda = seq(-2, 2, by = 0.1))
```



```
lambda_optimo <- boxcox_model$x[which.max(boxcox_model$y)]
lambda_optimo
```

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

Utiliza: boxcox(lm(Distancia~Velocidad)) si la variable con más alejamiento de normalidad es la distancia.

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

```
cars$dist_transformado <- (cars$dist^lambda_optimo - 1) / lambda_optimo
```

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.

```
cars$dist_log <- log(cars$dist)
```

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

Compara las medidas: sesgo y curtosis.

```
library(e1071)

skewness(cars$dist_transformado)
```

```
## [1] -0.1701619
```

```
kurtosis(cars$dist_transformado)
```

```
## [1] -0.186884
```

```
skewness(cars$dist_log)
```

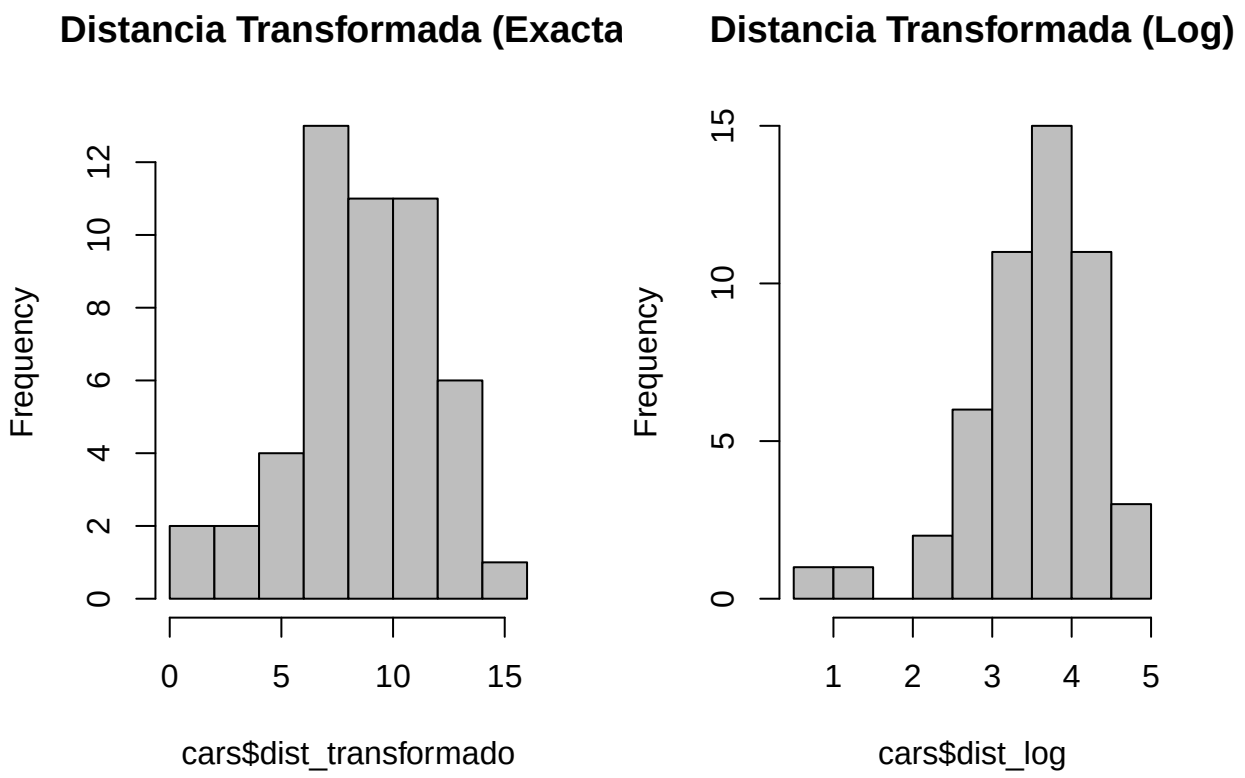
```
## [1] -1.302538
```

```
kurtosis(cars$dist_log)
```

```
## [1] 2.543008
```

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

```r
par(mfrow = c(1, 2))
hist(cars$dist_transformado, main = "Distancia Transformada (Exacta)", col = "gray")
hist(cars$dist_log, main = "Distancia Transformada (Log)", col = "gray")
```



Realiza algunas pruebas de normalidad para los datos transformados.

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

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

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

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

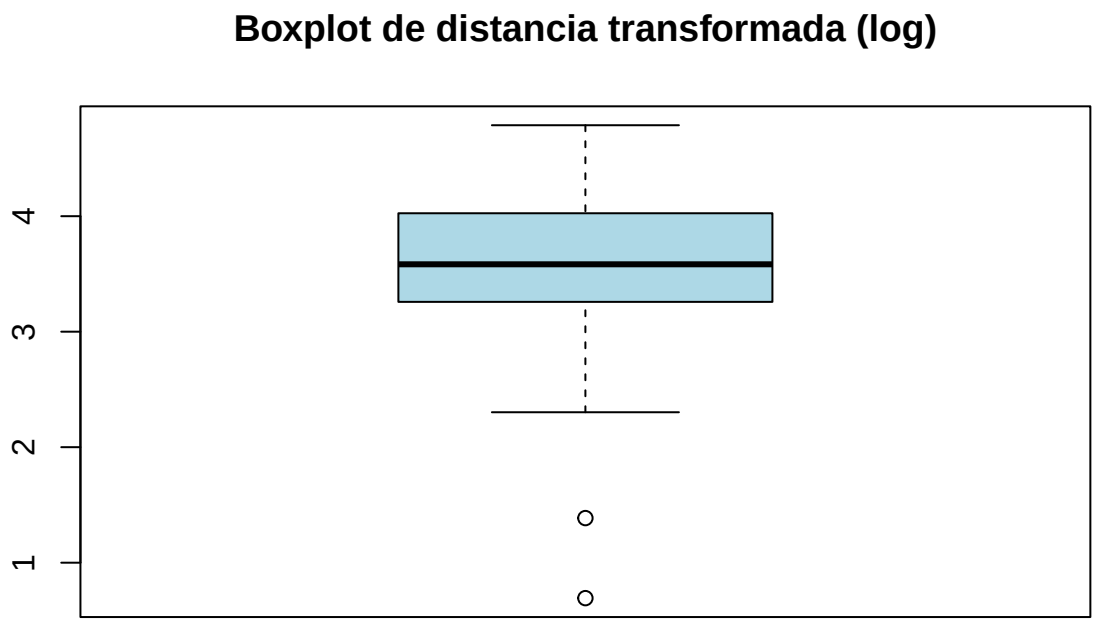
Detecta anomalías y corrige tu base de datos tranformado (datos atípicos, ceros anámalos, etc): solo en caso de no tener normalidad en las transformaciones. En caso de corrección de los datos por anomalías, vuelve a buscar la lambda para tus nuevos datos.

Al analisar nuestros resultados vemos que la transformación logaritmica no produce normalidad, indicandonos que hay datos atípicos y posiblemente anomalías en los datos transformados con el logaritmo.

Por lo cual se decidió revisar con el boxplot si hay posibles outliers.

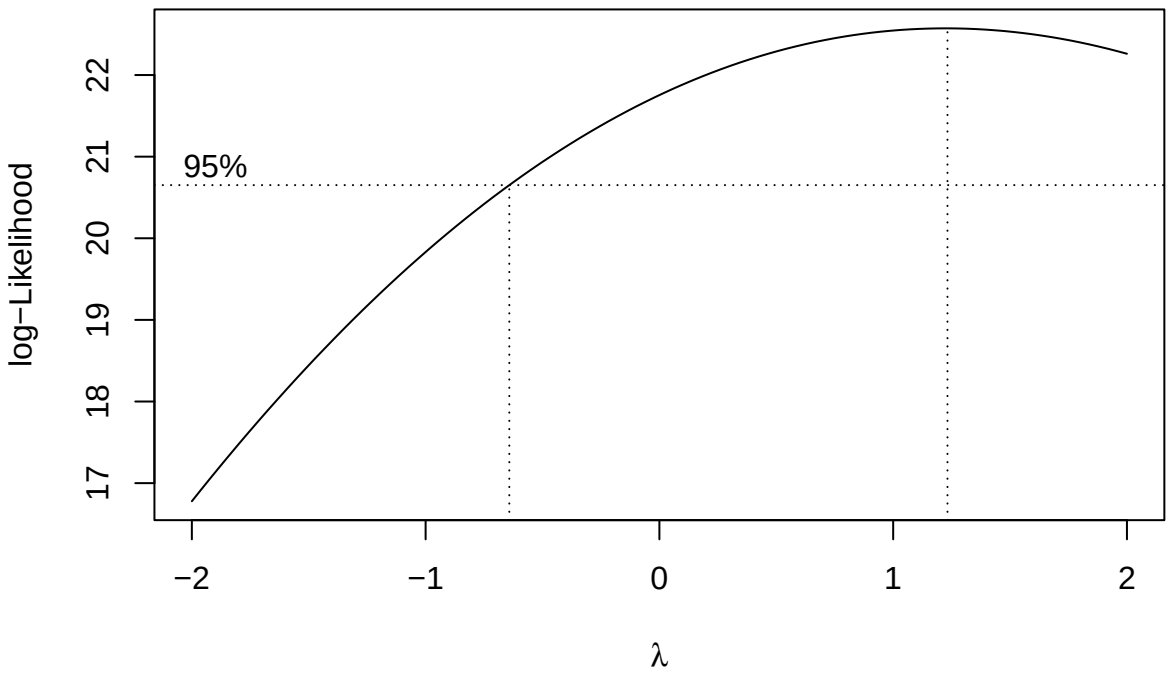
Despues se corrigieron y eliminaron valores atipicos , para proceder a recalcular lambda despues de la correción.

```
boxplot(cars$dist_log, main = "Boxplot de distancia transformada (log)", col = "lightblue")
```




```
lim_inf <- quantile(cars$dist_log, 0.05)
lim_sup <- quantile(cars$dist_log, 0.95)
cars_filtrado <- subset(cars, dist_log > lim_inf & dist_log < lim_sup)
```

```
boxcox_model_filtrado <- boxcox(lm(dist_log ~ speed, data = cars_filtrado),
                                lambda = seq(-2, 2, by = 0.1))
```




```
lambda_optimo_nuevo <- boxcox_model_filtrado$x[which.max(boxcox_model_filtrado$y)]
lambda_optimo_nuevo
```

```
## [1] 1.232323
```

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.

Despues de realizar las 2 transformaciones vemos que en Transformación exacta (Box-Cox con lambda óptimo) los datos nos muestran una distribución nomrla y con una mejor aproximación a la normalidad,le da validez al modelo.

Y con la Transformación aproximada (logarítmica) en este caso no nos garantiza normalidad y hasta vemos uan distribución sesgada con colas más largas, cosa que no queremos.

Por lo cual la transformación exacta de Box-Cox es la mejor opción en este caso.

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

Escribe el modelo lineal para la transformación.

```r
modelo_boxcox <- lm(dist_transformado ~ speed, data = cars)

summary(modelo_boxcox)
```

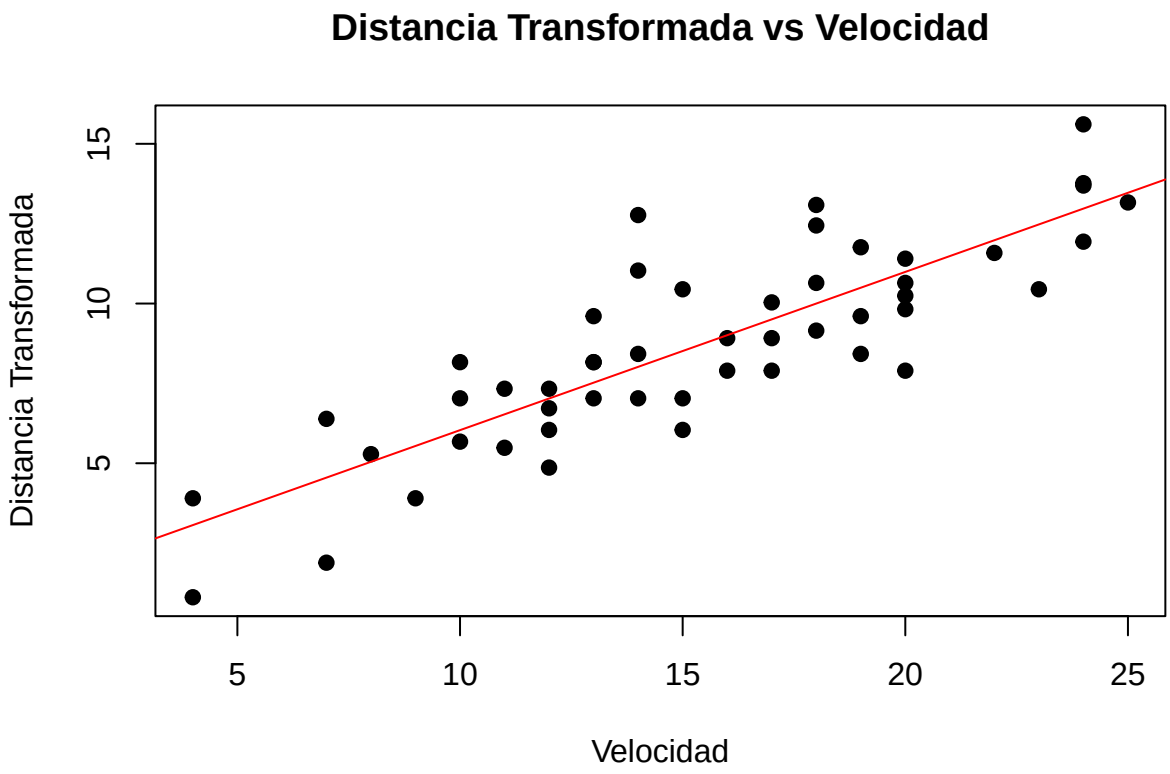
```
##
## Call:
## lm(formula = dist_transformado ~ speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0926 -1.0444 -0.3055  0.7999  4.7520
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.08227    0.73856   1.465    0.149
## speed        0.49541    0.04541  10.910 1.35e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.681 on 48 degrees of freedom
## Multiple R-squared:  0.7126, Adjusted R-squared:  0.7066
## F-statistic:   119 on 1 and 48 DF,  p-value: 1.354e-14
```

la velocidad tiene un efecto significativo en la distancia transformada. por si p-value.

vemos que tiene un R^2 de 0.7066 lo que significa que aproximadamente el 70.66% de la variación en la distancia transformada puede ser explicada por la velocidad. Indicandonos que el modelo captura una gran parte de la variación en los datos.

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

```r
plot(cars$speed, cars$dist_transformado, main = "Distancia Transformada vs Velocidad",
     xlab = "Velocidad", ylab = "Distancia Transformada", pch = 19)
abline(modelo_boxcox, col = "red")
```

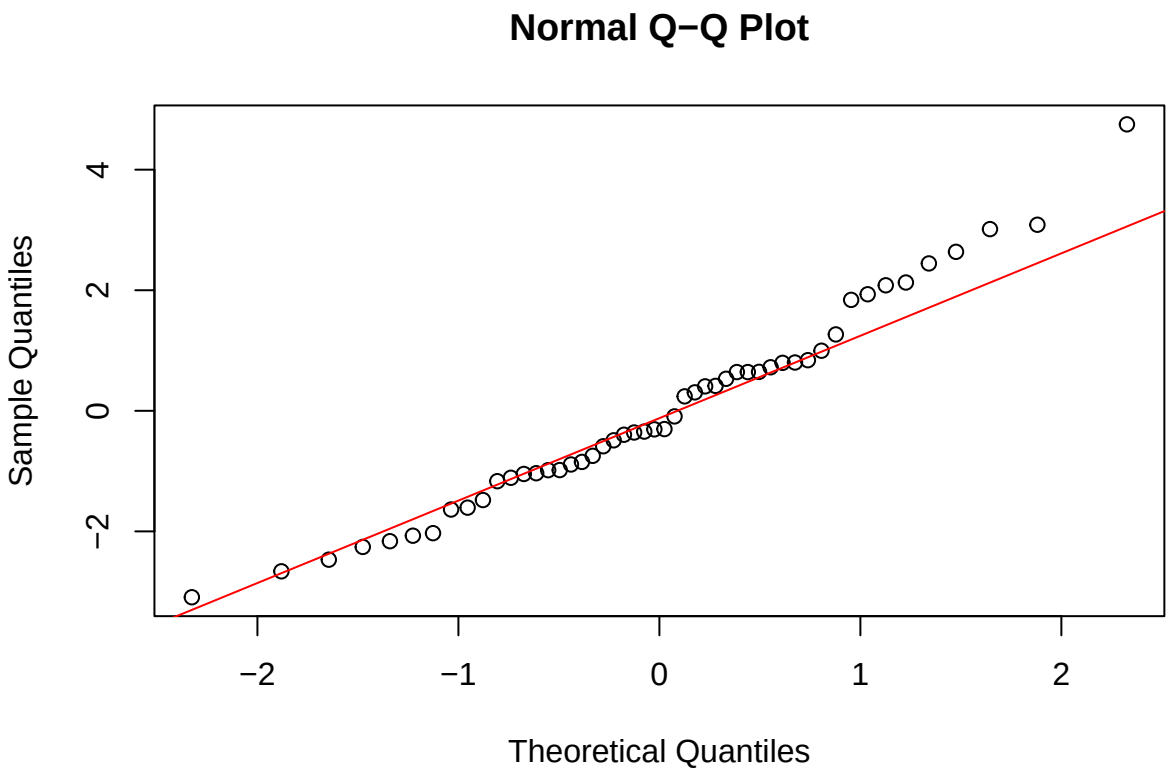


Analiza significancia del modelo (individual, conjunta y coeficiente de correlación)

vemos que tiene un R^2 de 0.7066 lo que significa que aproximadamente el 70.66% de la variación en la distancia transformada puede ser explicada por la velocidad. Indicandonos que el modelo captura una gran parte de la variación en los datos.

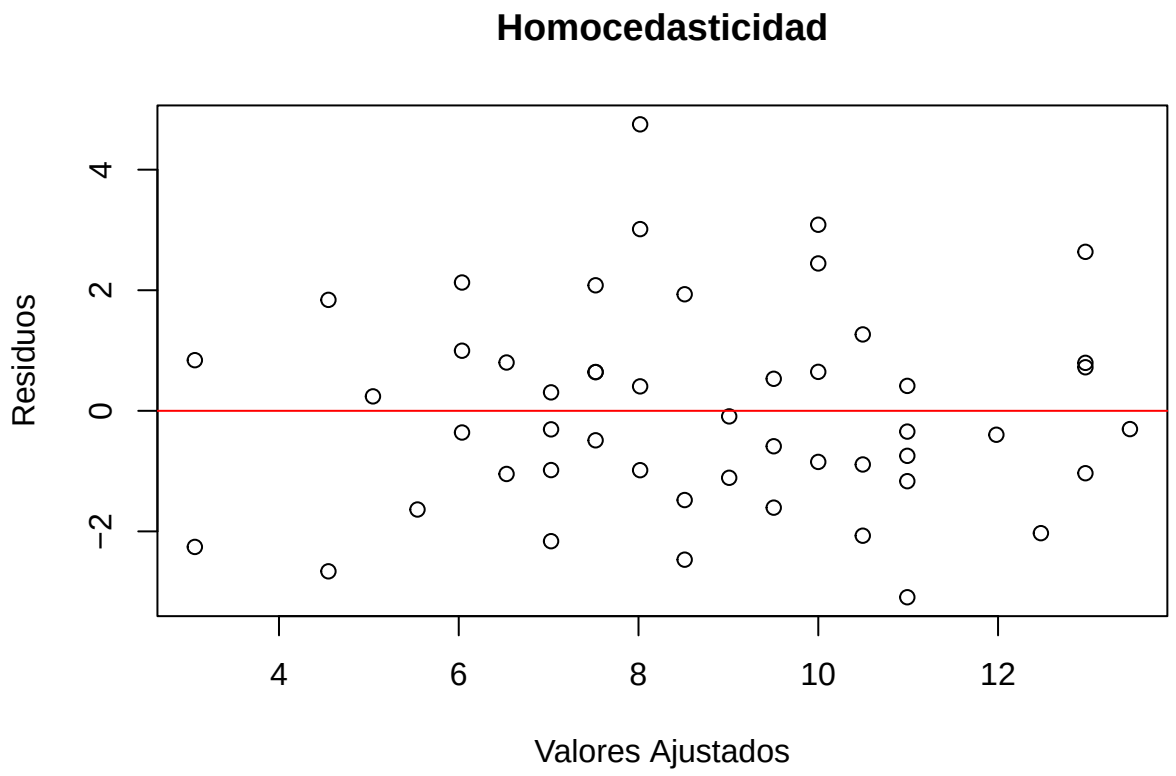
Analiza validez del modelo: normalidad de los residuos, homocedasticidad e independencia. Indica si hay candidatos a datos atípicos o influyentes en la regresión. Usa plot(Modelo) para los gráficos y añade pruebas de hipótesis.

```
qqnorm(residuals(modelo_boxcox))
qqline(residuals(modelo_boxcox), col = "red")
```



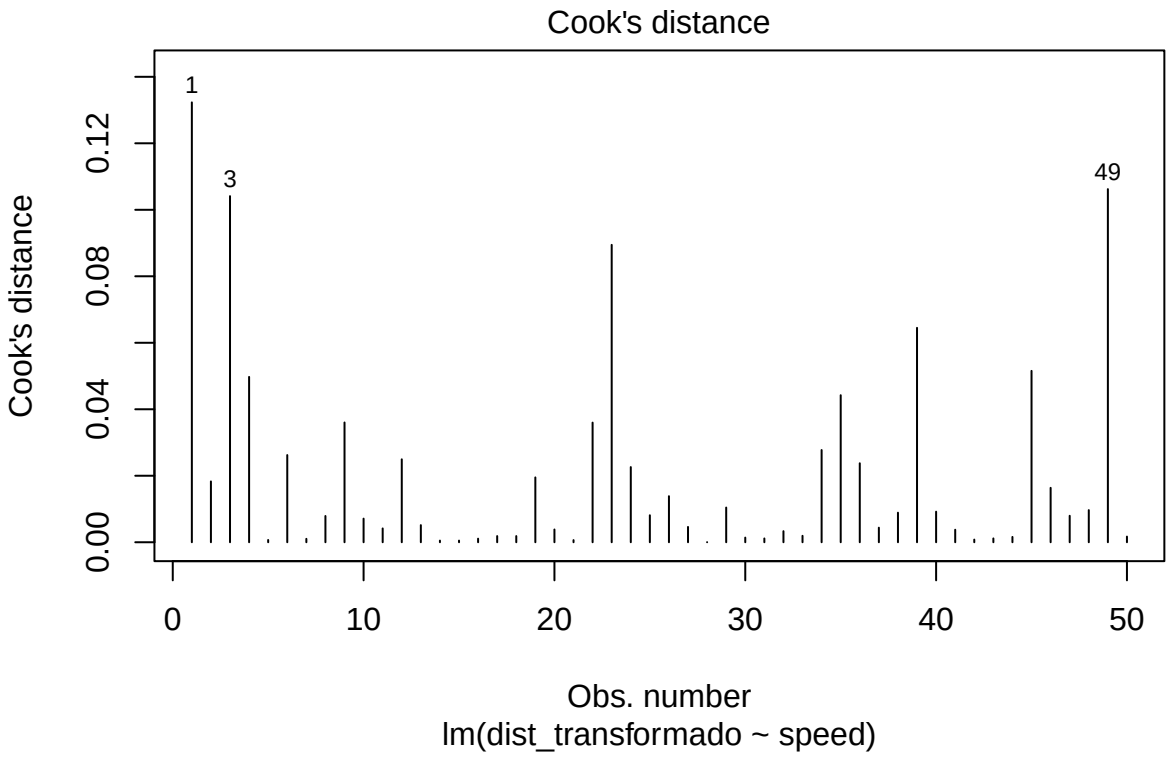
En este q-q plot la gran mayoria de los puntos si siguen la línea teórica de normalidad, en los extremos vemos desvios, sin embargo se mantiene dentro de un rango aceptable.

```
plot(fitted(modelo_boxcox), residuals(modelo_boxcox), main = "Homocedasticidad",
     xlab = "Valores Ajustados", ylab = "Residuos")
abline(h = 0, col = "red")
```

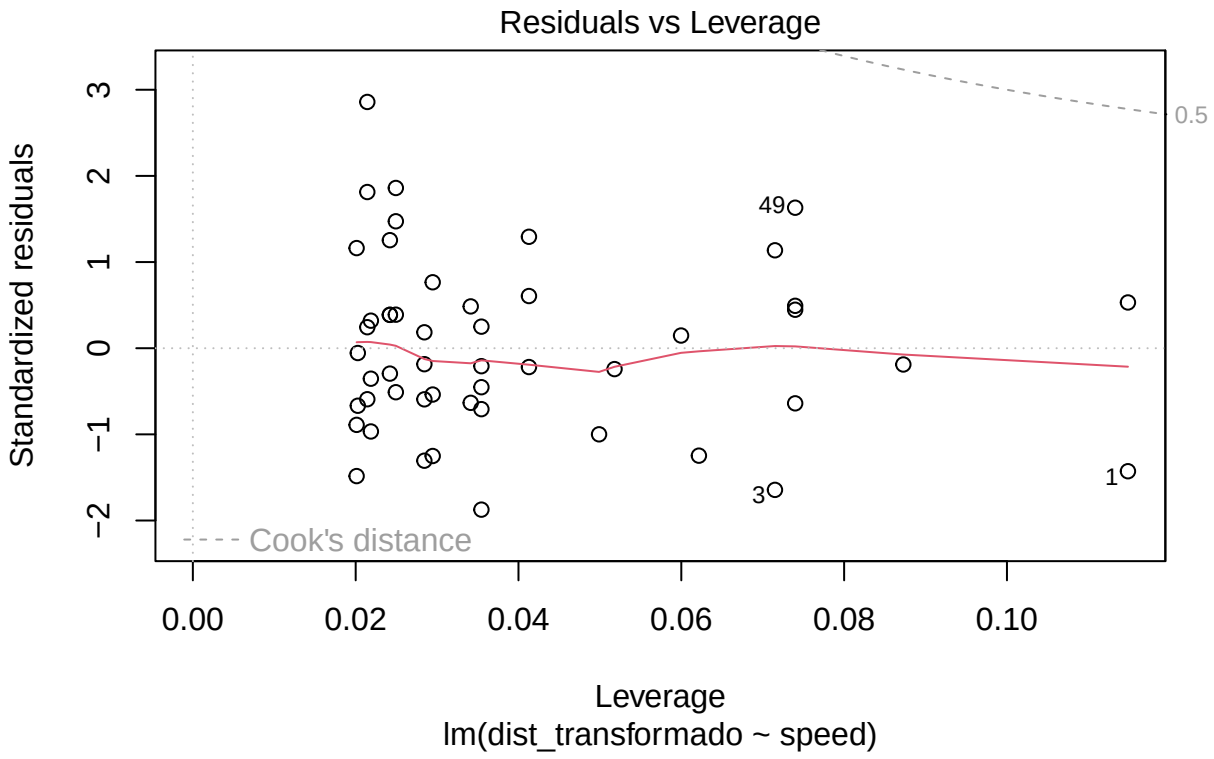


Vemos que los residuos se distribuyen de manera uniforme alrededor del cero.

```
plot(modelo_boxcox, which = 4)
```




```
plot(modelo_boxcox, which = 5)
```



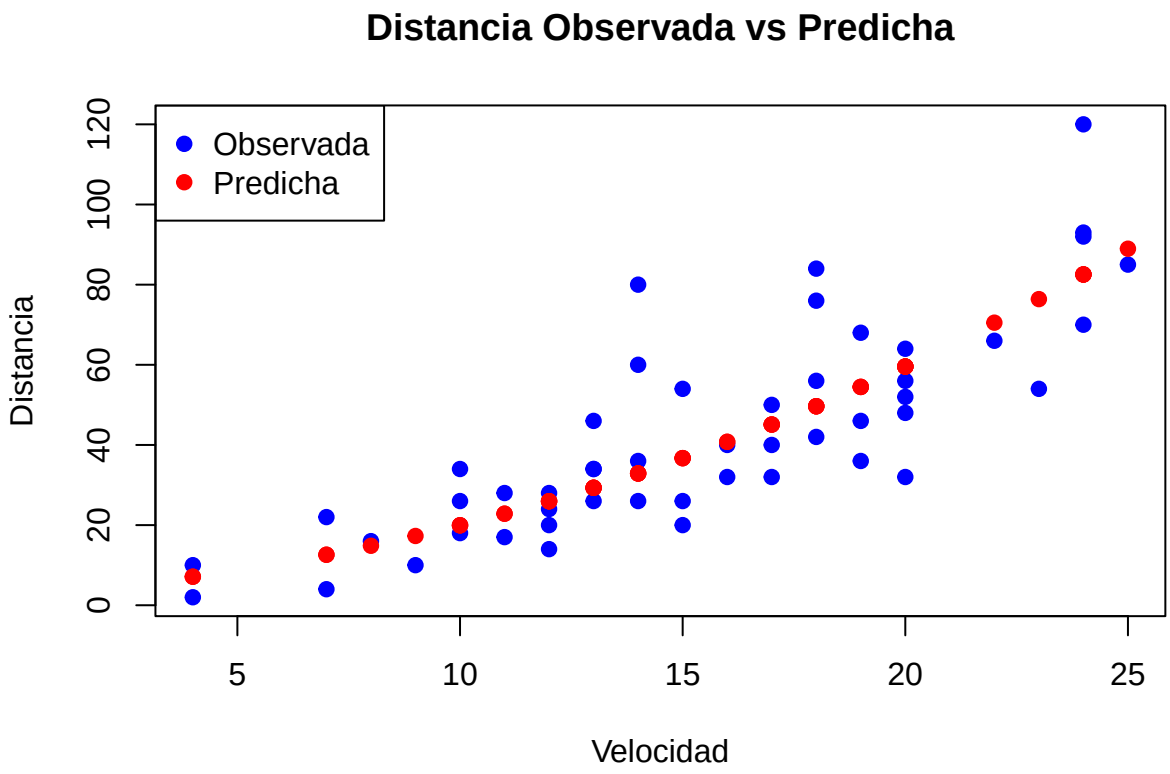
Despeja la distancia del modelo lineal obtenido entre la transformación y la velocidad. Obtendrás el modelo no lineal que relaciona la distancia con la velocidad directamente (y no con su transformación).

```
predicciones <- predict(modelo_boxcox, type = "response")

distancia_original_estimada <- (predicciones * lambda_optimo + 1)^(1/lambda_optimo)
```

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

```
plot(cars$speed, cars$dist, main = "Distancia Observada vs Predicha",
     xlab = "Velocidad", ylab = "Distancia", pch = 19, col = "blue")
points(cars$speed, distancia_original_estimada, pch = 19, col = "red")
legend("topleft", legend = c("Observada", "Predicha"),
       col = c("blue", "red"), pch = 19)
```



el modelo predice de manera correcta la distancia.

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

Vemos que en el q-q plot la gran mayoria de los puntos si siguen la línea teórica de normalidad , tenemos un R^2 alto, es decir es significativo y tiene un buen ajuste, en homocedasticidad lols residuos se distribuyen de manera uniforme.

Al análisar el modelo nos damos cuenta de que es idóneo y válidos para lograr explicar la relación entre la velocidad y distancia.

Parte 4: Conclusión

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.

Vemos que de los dos modeloa analizados el modelo con la transformación exacta de Box-Cox es el mejor para describir la relación entre la distancia y la velocidad. Ya que presenta un mejor ajuste y un R^2 másl ato, sus residuos siguen una distribución normal y ceunta con homocedasticidad y muestra buenas predicciones.

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

El el gráfico de residuos vs leverage no se ven puntos altamente influyentes, sin embargo algunos puntos se desvían de los valores ajustados, especialmente en los valores más altos de velocidad. Vemos algunos

problemas en el ajuste de valores extremos ya que se ve un poc de dispersión en esas zonas.