Actividad 13

José Carlos Sánchez Gómez

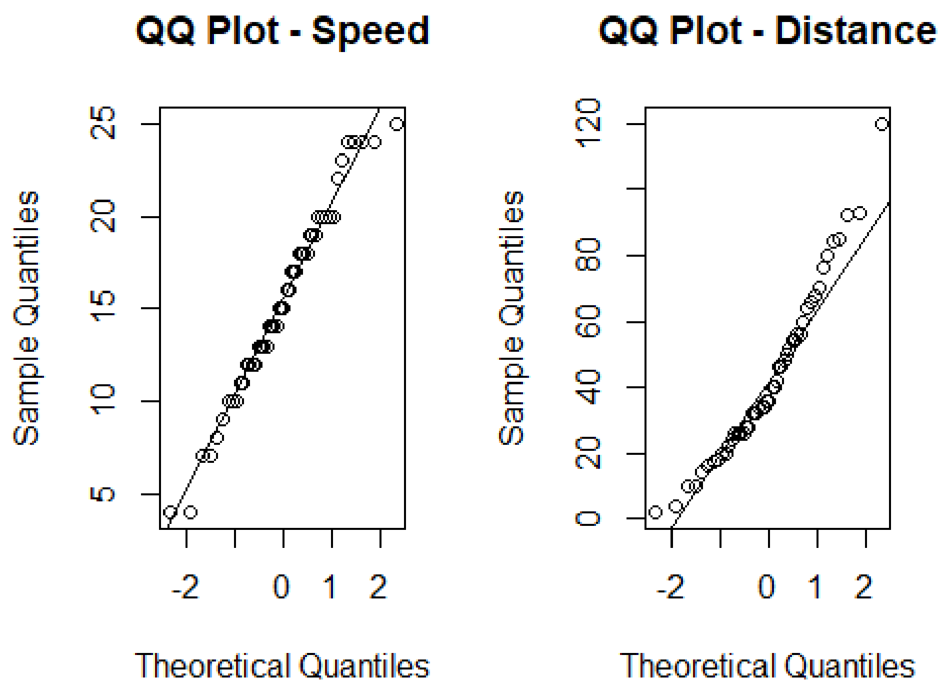
2024-09-12

Analisis de Normalidad

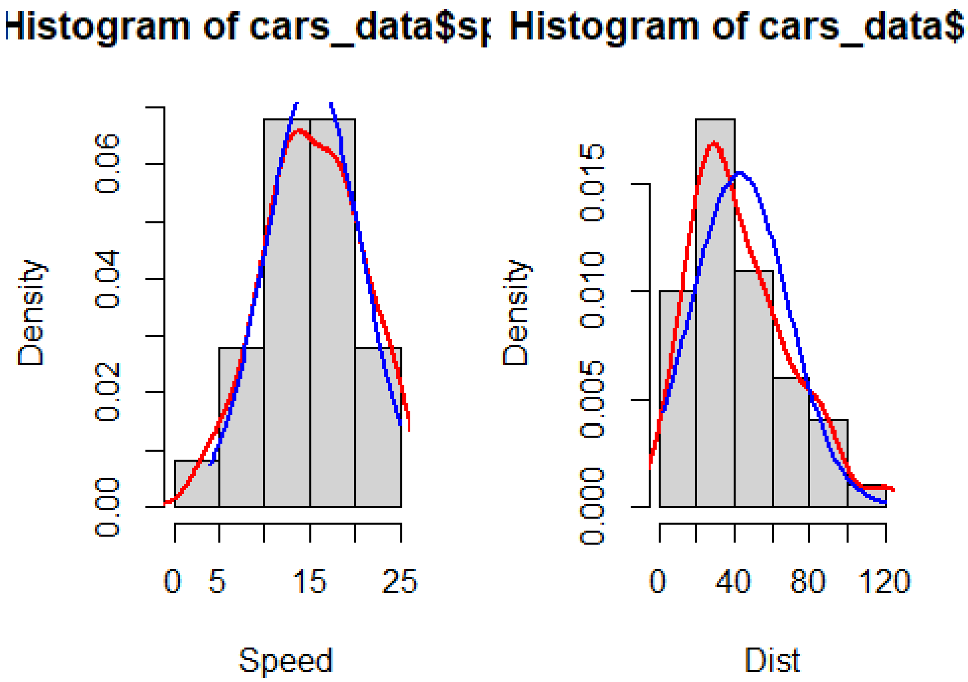
```
cars_data = cars
```

Pruebas de normalidad

```
shapiro.test(cars_data$speed)

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

shapiro.test(cars_data$dist)

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

library(nortest)
ad.test(cars_data$speed)

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

ad.test(cars_data$dist)

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

par(mfrow=c(1, 2))
qqnorm(cars$speed, main="QQ Plot - Speed")
qqline(cars$speed)

qqnorm(cars$dist, main="QQ Plot - Distance")
qqline(cars$dist)
```




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

hist(cars_data$speed, freq=FALSE, xlab="Speed", col="lightgray",
border="black")

lines(density(cars_data$speed), col="red", lwd=2)

curve(dnorm(x, mean=mean(cars_data$speed), sd=sd(cars_data$speed)),
      from=min(cars_data$speed), to=max(cars_data$speed),
      add=TRUE, col="blue", lwd=2)

hist(cars_data$dist, freq=FALSE, xlab="Dist", col="lightgray",
border="black")

# Añadir la línea de densidad (estimada a partir de los datos)
lines(density(cars_data$dist), col="red", lwd=2)

# Añadir la curva de la distribución normal teórica
curve(dnorm(x, mean=mean(cars_data$dist), sd=sd(cars_data$dist)),
      from=min(cars_data$dist), to=max(cars_data$dist),
      add=TRUE, col="blue", lwd=2)
```



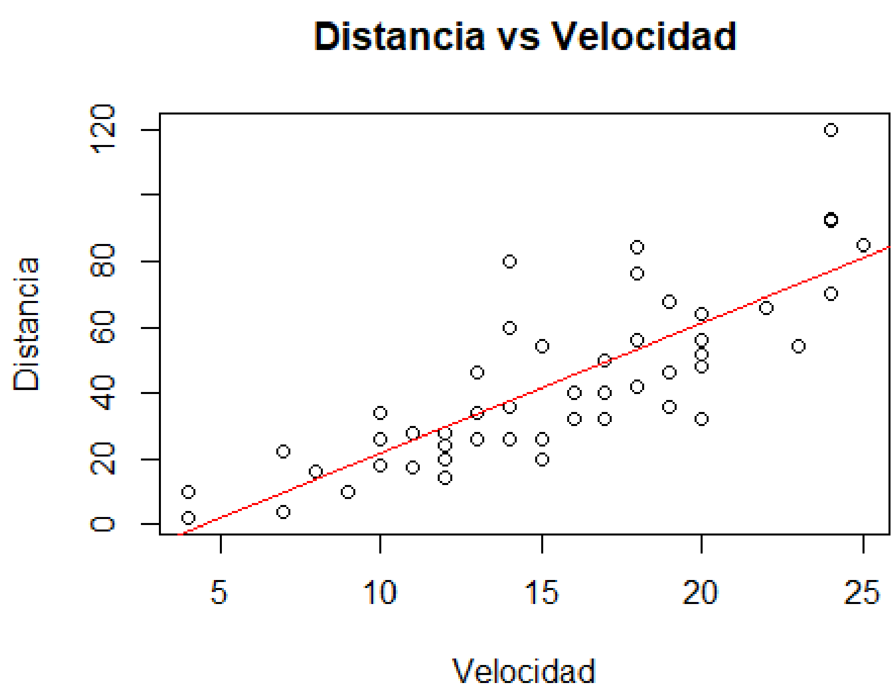

```
library(e1071)
cat("Curtosis de speed:", kurtosis(cars_data$speed), "\n")

## Curtosis de speed: -0.6730924

cat("Sesgo de speed:", skewness(cars_data$speed), "\n", "\n")

## Sesgo de speed: -0.1105533
##

cat("Curtosis de dist:", kurtosis(cars_data$dist), "\n")

## Curtosis de dist: 0.1193971

cat("Sesgo de dist:", skewness(cars_data$dist), "\n", "\n")

## Sesgo de dist: 0.7591268
##
```

Regresión Lineal

Prueba de regresion lineal entre distancia y velocidad

```
Modelo1 = lm(dist ~ speed, data = cars_data)
summary(Modelo1)

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

```
##
## 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_data$speed, cars_data$dist, main="Distancia vs Velocidad",
xlab="Velocidad", ylab="Distancia")
abline(Modelo1, col="red")
```



Analisis de significancia

```
summary(Modelo1)

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

```
## 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 significancia del intercepto, como la de la velocidad son significativas, teniendo la velocidad una mayor relación con la distancia (demostrado por su valor p). La significancia en conjunto de igual manera es significativa, pues el valor p es muy bajo. El coeficiente de determinación nos dice que el 65 % de la variabilidad en la distancia, puede ser explicada por la velocidad.

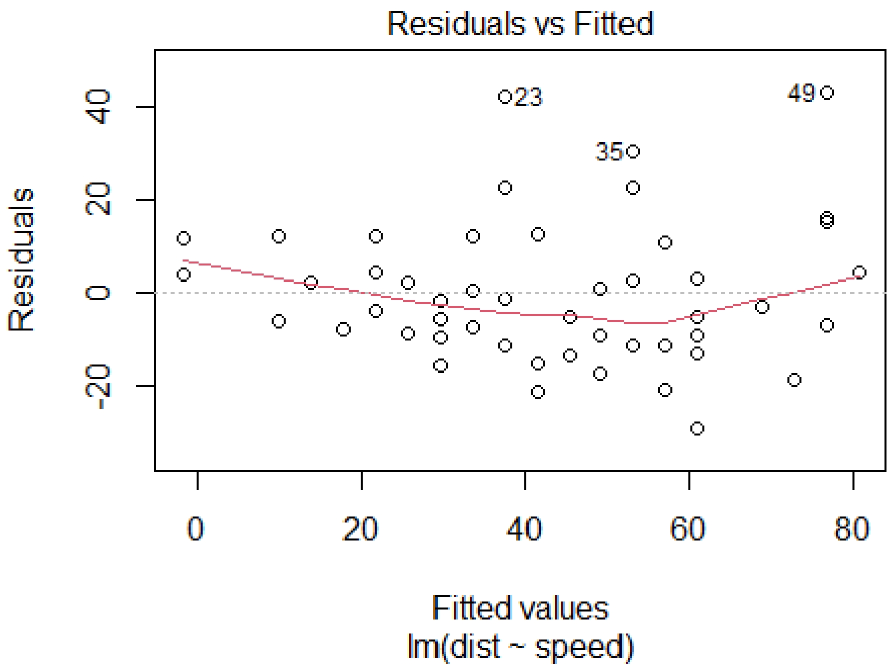
Validez del modelo

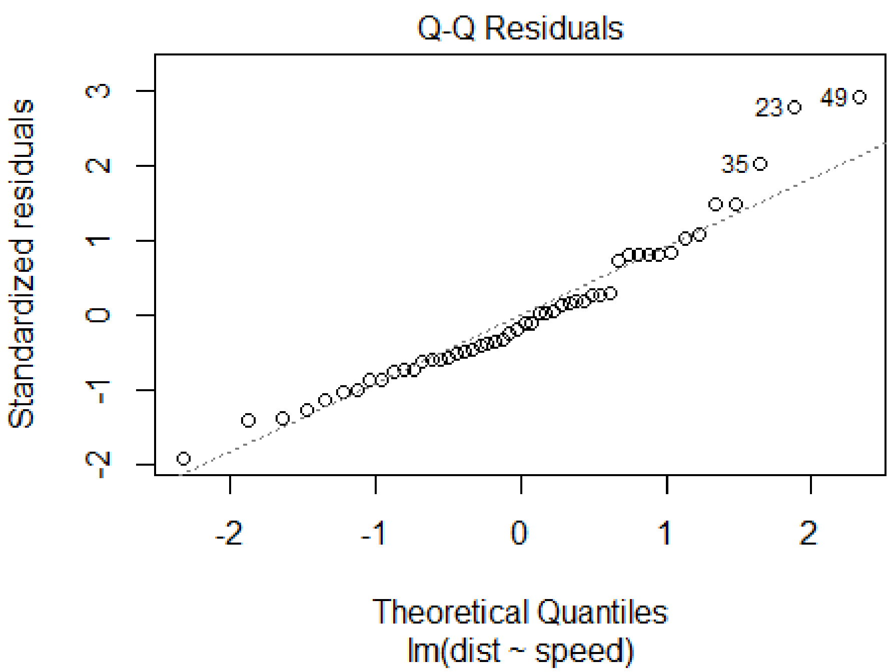
Normalidad de los residuos

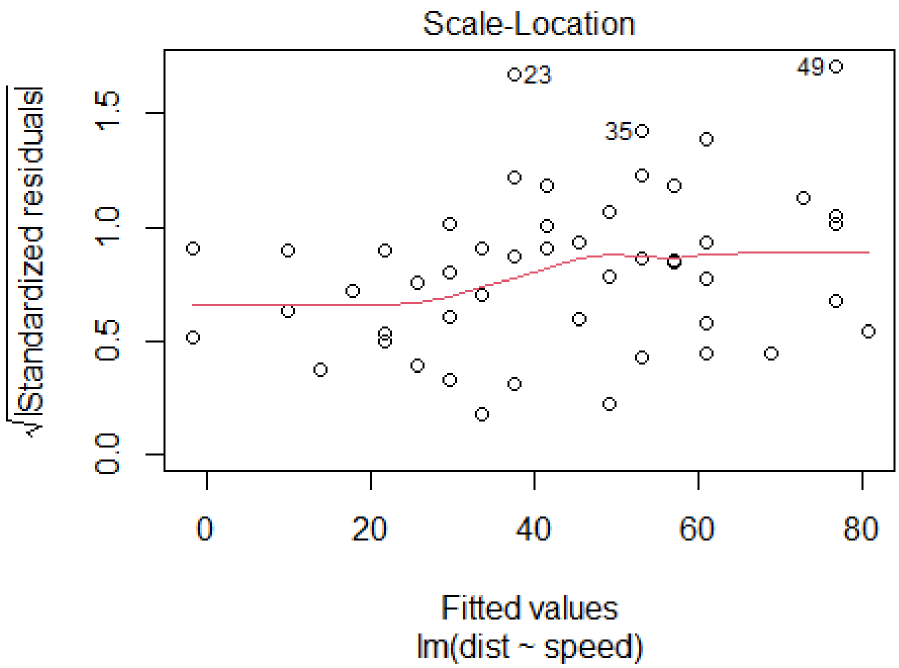
```
ad.test(Modelo1$residuals)
```

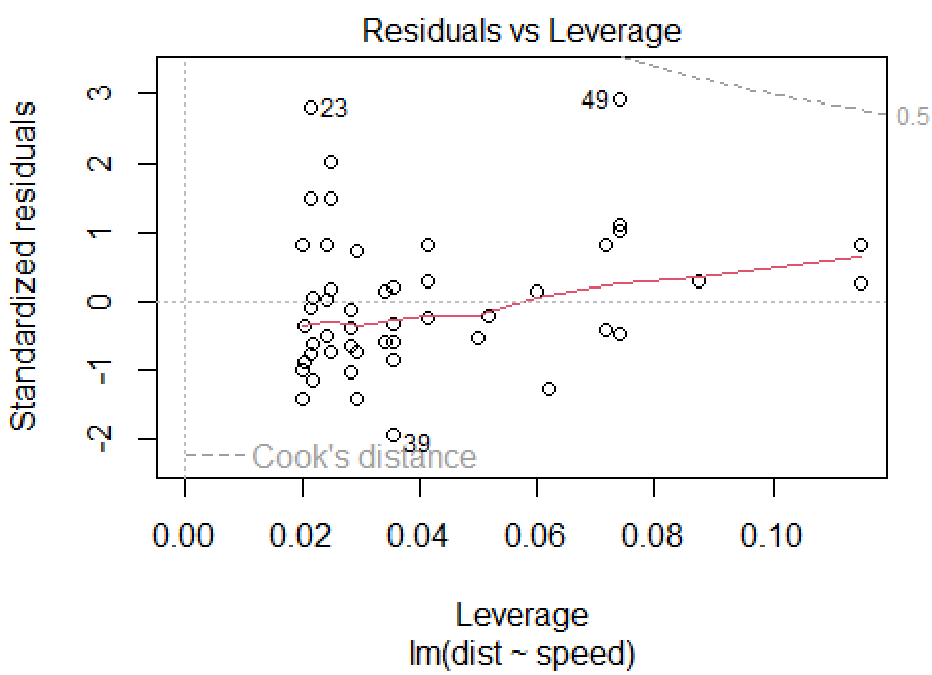
```
## 
##  Anderson-Darling normality test
## 
## data:  Modelo1$residuals
## A = 0.79406, p-value = 0.0369
```

```
plot(Modelo1)
```


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


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





Podemos decir que no existe una normalidad en los residuos, pues los valores en las colas de la gráfica, se despegan a los datos de la qqline, especialmente los valores mayores. Además de esto se rechaza h_0 ya que el valor p de los residuos es menor a

0.05 (alfa que se esta utilizando), por lo que podemos decir que los datos no provienen de una población normal.

Media de los residuos

```
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
```

Podemos decir que la media de los residuos no es igual a 0. Se rechaza de igual manera h_0 puesto que el valor p de la media no es igual a cero. Por lo tanto se acepta h_1

Homocedasticidad

```
library(lmtest)

## Cargando paquete requerido: zoo

##
## Adjuntando el paquete: '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
```

Se acepta h_0 ya que el valor p propuesto por la prueba de Breusch-Pagan, es mayor a 0.05. Lo que nos dice que la varianza de los errores es constante.

Independencia

```
library(lmtest)
dwtest(Modelo1)

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

```
## data:  Modelo1
## DW = 1.6762, p-value = 0.09522
## alternative hypothesis: true autocorrelation is greater than 0
```

Se acepta h_0 puesto que el valor p propuesto por la prueba de Durbin-Watson, supera a 0.05 que es alfa. Esto nos dice que los errores no están relacionados.

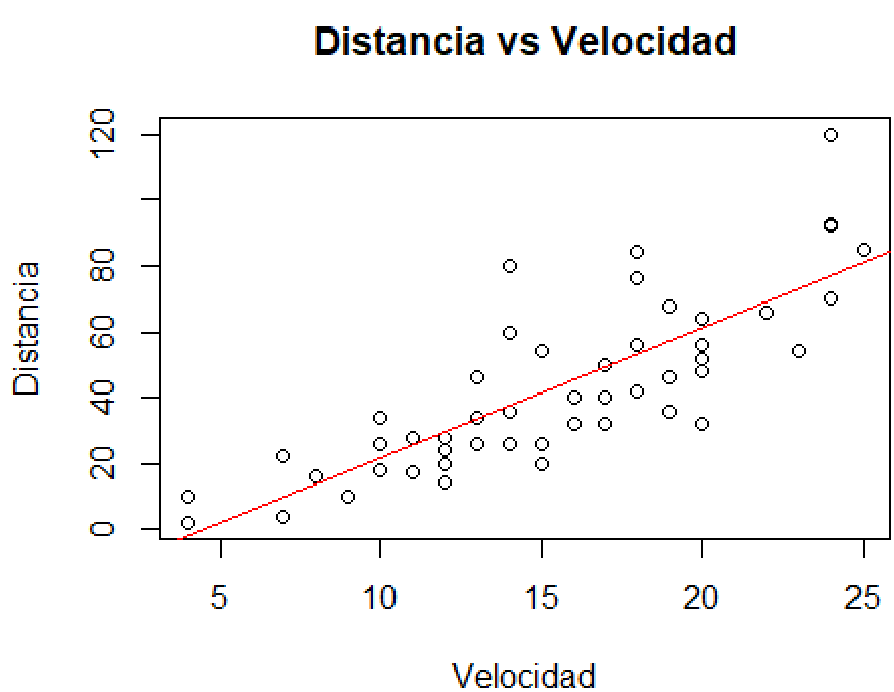
Linealidad

```
library(lmtest)
resettest(Modelo1)

##
##  RESET test
##
## data:  Modelo1
## RESET = 1.5554, df1 = 2, df2 = 46, p-value = 0.222
```

Se acepta h_0 puesto que el valor p propuesto por la prueba de RESET, supera a 0.05 que es alfa. Esto nos dice que no hay términos omitidos que indiquen linealidad.

Grafica de datos y modelo de distancia en funcion de velocidad

```
plot(cars_data$speed, cars_data$dist, main="Distancia vs Velocidad",
xlab="Velocidad", ylab="Distancia")
abline(Modelo1, col="red")
```



Modelo propuesto: $y = b_0 + b_1 * speed$

```r
b0 = Modelo1$coefficients[1]
b1 = Modelo1$coefficients[2]

cat("B0: ", b0, "\n")

## B0:  -17.57909

cat("B1: ", b1, "\n")

## B1:  3.932409
```

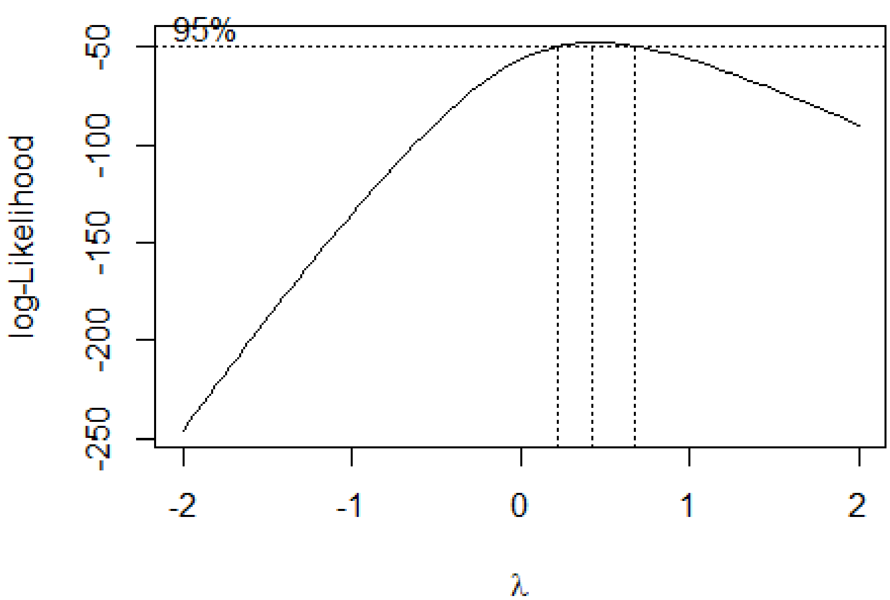
Modelo con valores: $y = -17.5709 + 3.9324x$

El modelo propuesto, a pesar de que podría ser mejor, arroja buenos resultados en las pruebas de significancia y validación. La velocidad es un factor importante (dado por su valor p), y el modelo explica un 65% de la variabilidad de la distancia. Apesar de que los valores no provengan de una población normal y la media de los residuos no sea igual a 0, con las pruebas podemos decir que hay homocedasticidad, linealidad e independencia en los datos.

Regresión No Lineal

Obtención de lambda

```r
library(MASS)
boxcox_value = boxcox(lm(dist ~ speed, data = cars_data))
```



```
lambda_opt = boxcox_value$x[which.max(boxcox_value$y)]

cat("Mejor lambda: ", lambda_opt)

## Mejor lambda:  0.4242424
```

Dado que nuestra lambda es de 0.4242, la función recomendada para la aproximación es $\sqrt(x)$, y para la exacta será $\frac{x^{\lambda}-1}{\lambda}$, que quedaria como $\frac{x^{(0.4242)}-1}{0.4242}$.

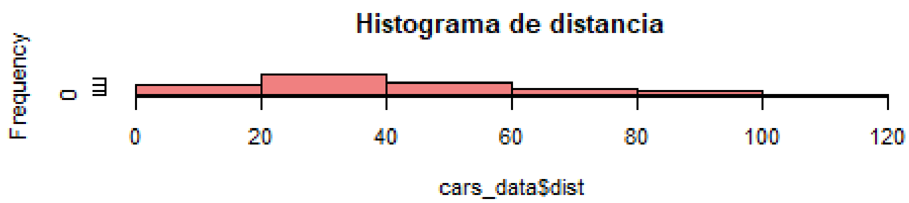
```
min(cars_data$dist)

## [1] 2
```

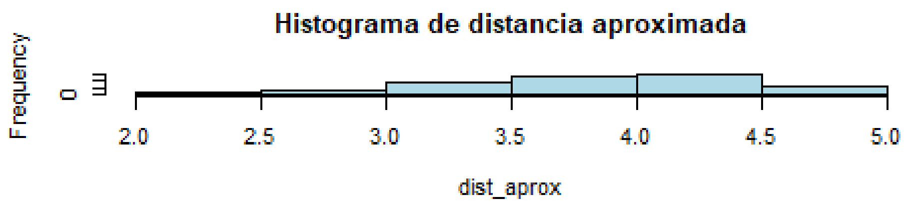
Obtención de sesgo y curtosis

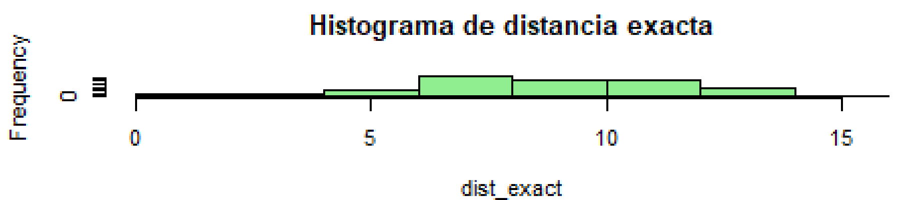
```
library(e1071)
library(nortest)
dist = cars_data$speed

dist_aprox = sqrt(cars_data$speed)
dist_exact = (cars_data$dist^lambda_opt - 1) / lambda_opt

# resumen de los datos normales
dist_kurtosis = kurtosis(dist)
dist_sesgo = skewness(dist)

datos = data.frame(
  Estadistico = c("Curtosis", "Sesgo"),
  Orginal = c(dist_kurtosis, dist_sesgo),
  "Modelo Aproximado" = c(kurtosis(dist_aprox), skewness(dist_aprox)),
  "Modelo Exacto" = c(kurtosis(dist_exact), skewness(dist_exact))
)
datos

##   Estadistico    Orginal Modelo.Aproximado Modelo.Exacto
## 1    Curtosis -0.6730924       -0.03564543    -0.1868840
## 2       Sesgo -0.1105533       -0.57277746    -0.1701619
```

Histogramas con valores transformados

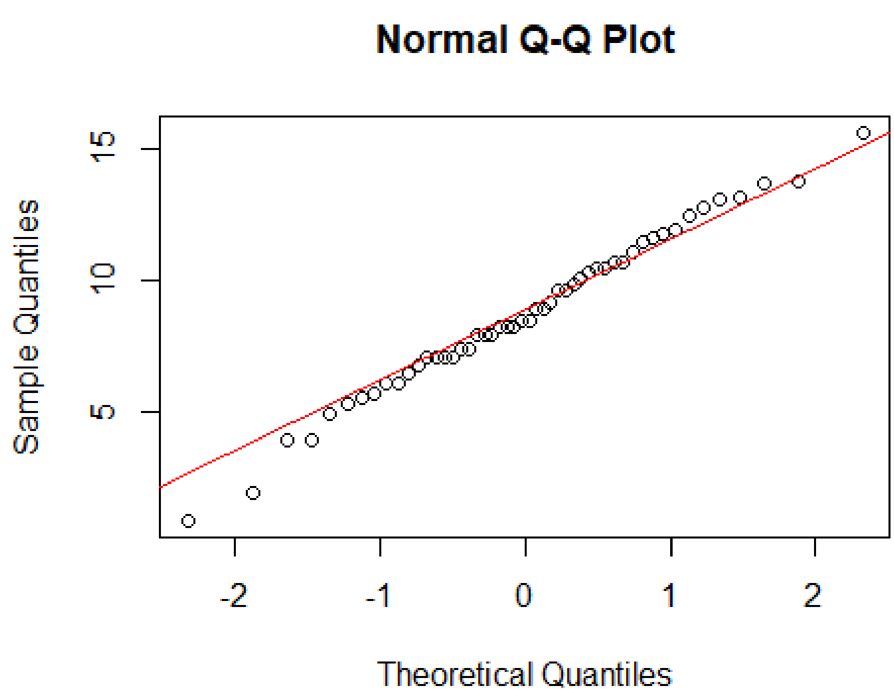
```
par(mfrow=c(3,1))

hist(cars_data$dist, col = "lightcoral", main="Histograma de distancia")
hist(dist_aprox, col = "lightblue", main = "Histograma de distancia
aproximada")
hist(dist_exact, col = "lightgreen", main = "Histograma de distancia
exacta")
```



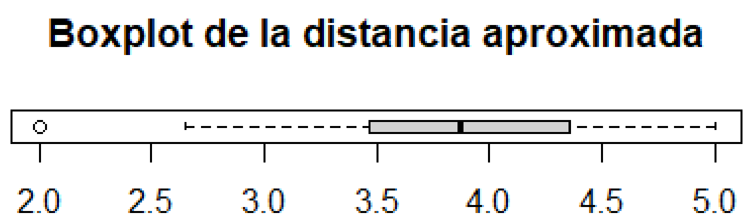


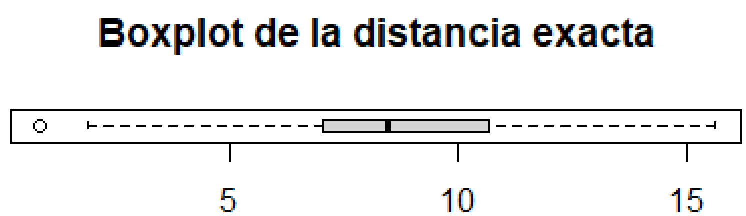



```r
qqnorm(dist_exact)
qqline(dist_exact, col = "red")
```



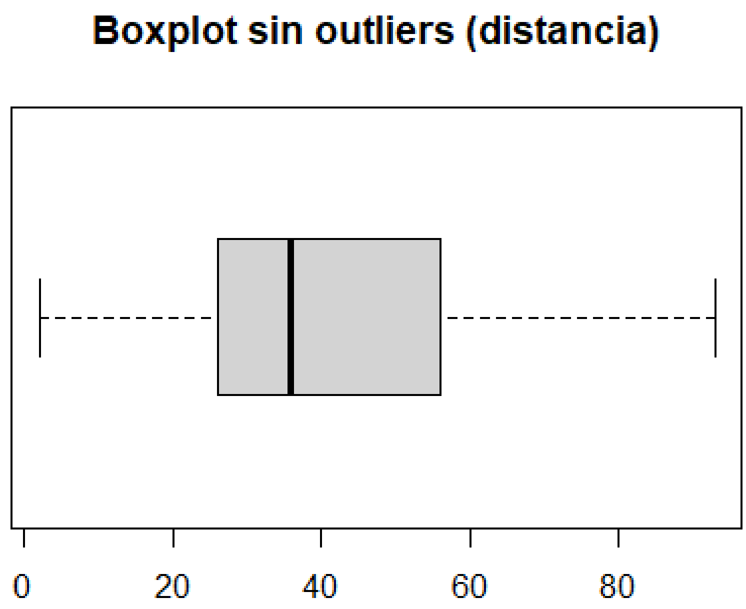
```r
# Boxplot para la transformación exacta y aproximada
par(mfrow = c(2, 1))
boxplot(dist_aprox, main = "Boxplot de la distancia aproximada",
horizontal = TRUE)
boxplot(dist_exact, main = "Boxplot de la distancia exacta", horizontal =
TRUE)
```





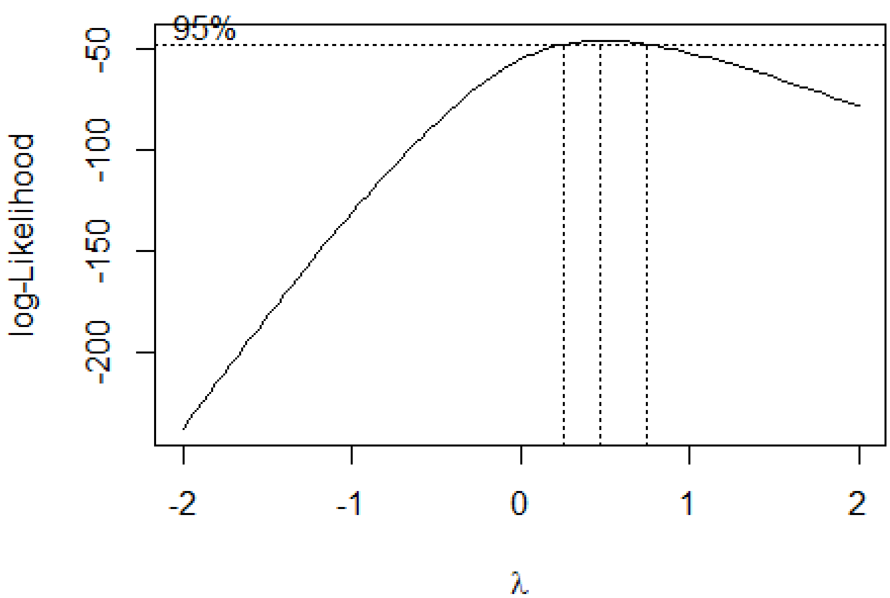
Procederemos a borrar los valores atipicos del modelo pues estos no representan de buena manera la distancia recorrida a la velocidad, pues estos datos son cuando el automovil apenas esta arrancando.

```r
Q1 <- quantile(cars_data$dist, 0.25)
Q3 <- quantile(cars_data$dist, 0.75)
IQR_value <- Q3 - Q1

# Definir límites inferiores y superiores
lower_bound <- Q1 - 1.5 * IQR_value
upper_bound <- Q3 + 1.5 * IQR_value

# Eliminar los outliers
filtered_data = cars_data[cars_data$dist >= lower_bound & cars_data$dist
<= upper_bound, ]

# Graficar el nuevo boxplot sin outliers
boxplot(filtered_data$dist, main = "Boxplot sin outliers (distancia)",
horizontal = TRUE)
```



Sacaremos la lambda de nuevo para otro modelo

```
library(MASS)
boxcox_value = boxcox(lm(dist ~ speed, data = filtered_data))
```



```r
lambda_opt = boxcox_value$x[which.max(boxcox_value$y)]

cat("Mejor lambda: ", lambda_opt)

## Mejor lambda:  0.4646465
```

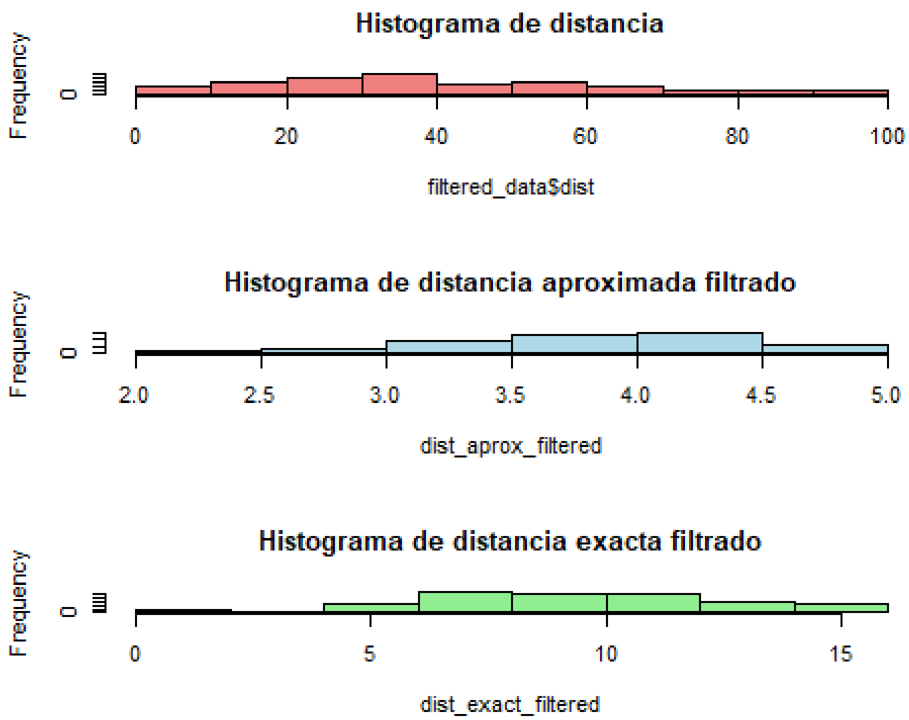
Dado que nuestra lambda es de 0.4646, la función recomendada para la aproximación es $\log(x)$, y para la exacta será $\frac{x^{\lambda}-1}{\lambda}$, que quedaria como $\frac{x^{(0.4646)}-1}{0.4646}$.

```r
min(filtered_data$dist)

## [1] 2
```

Obtención de sesgo y curtosis

```r
library(e1071)
library(nortest)
dist = filtered_data$dist
dist_aprox_filtered = sqrt(filtered_data$speed)
dist_exact_filtered = (filtered_data$dist^lambda_opt - 1) / lambda_opt

# resumen de los datos normales
dist_kurtosis = kurtosis(dist)
dist_sesgo = skewness(dist)

datos = data.frame(
  Estadistico = c("Curtosis", "Sesgo"),
  Orginal = c(dist_kurtosis, dist_sesgo),
  "Modelo Aproximado" = c(kurtosis(dist_aprox_filtered),
skewness(dist_aprox_filtered)),
  "Modelo Exacto" = c(kurtosis(dist_exact_filtered),
skewness(dist_exact_filtered)),
  "Modelo Exacto Anterior" = c(kurtosis(dist_exact),
skewness(dist_exact))
)
datos

##   Estadistico    Orginal Modelo.Aproximado Modelo.Exacto
Modelo.Exacto.Anterior
## 1    Curtosis -0.6409882       0.004643848    -0.3566406            -
0.1868840
## 2       Sesgo  0.5002547      -0.582083098    -0.2381772            -
0.1701619

par(mfrow=c(3,1))

hist(filtered_data$dist, col = "lightcoral", main="Histograma de
distancia")
hist(dist_aprox_filtered, col = "lightblue", main = "Histograma de
distancia aproximada filtrado")
```

```r
hist(dist_exact_filtered, col = "lightgreen", main = "Histograma de distancia exacta filtrado")
```



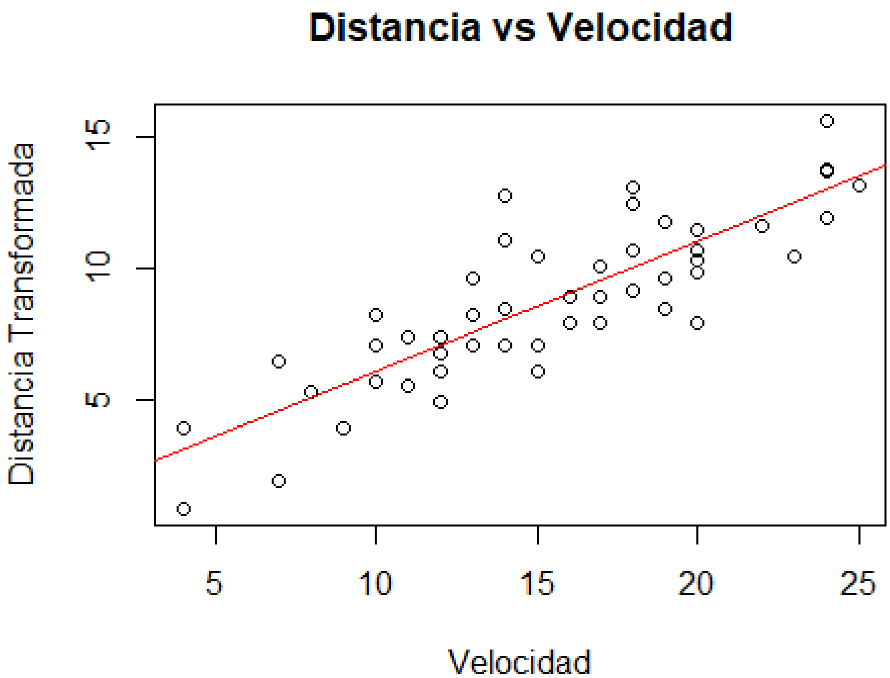
Tras ver los resultados de las gráficas de los datos, así como los valores de sesgo y kurtosis de los datos normales, como los transformados. Podemos concluir que la mejor transformación es la exacta sin datos filtrados por los cuantilos, ya que de todas las transformaciones, esta es la que se acerca más a la normal, además de ser la que tiene un sesgo y una curtosis más acercada al cero.

Regresion Lineal con datos transformados

```r
Modelo2 = lm(dist_exact ~ cars_data$speed)
summary(Modelo2)

## 
## Call:
## lm(formula = dist_exact ~ cars_data$speed)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -3.0926 -1.0444 -0.3055  0.7999  4.7520 
## 
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)    
## (Intercept)      1.08227    0.73856   1.465    0.149    
## cars_data$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
```

```
plot(cars_data$speed, dist_exact, main="Distancia vs Velocidad",
xlab="Velocidad", ylab="Distancia Transformada")
abline(Modelo2, col="red")
```



Analisis de significancia

```
summary(Modelo2)
```

```
##
## Call:
## lm(formula = dist_exact ~ cars_data$speed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0926 -1.0444 -0.3055  0.7999  4.7520
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.08227    0.73856   1.465    0.149
## cars_data$speed  0.49541    0.04541  10.910 1.35e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

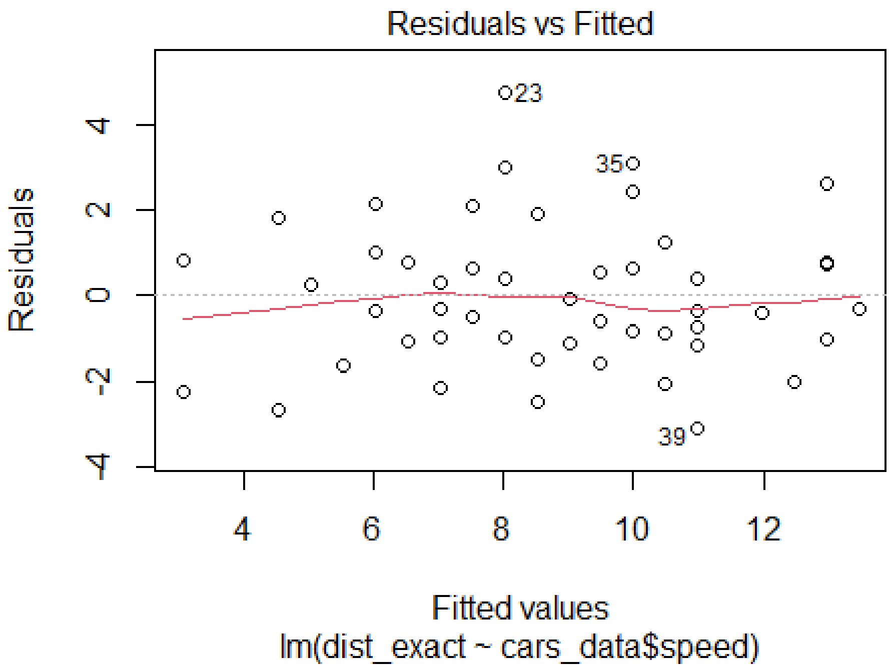
```
## Residual standard error: 1.681 on 48 degrees of freedom
## Multiple R-squared:  0.7126, Adjusted R-squared:  0.7066
## F-statistic:   119 on 1 and 48 DF,  p-value: 1.354e-14
```

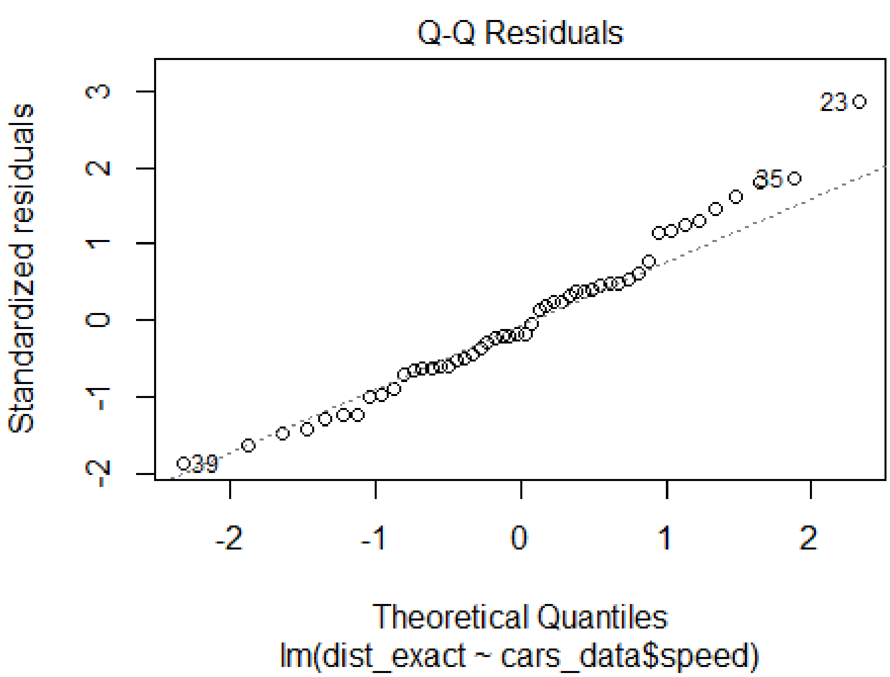
La significancia del intercepto, como la de la velocidad son significativas, teniendo la velocidad una mayor relación con la distancia (demostrado por su valor p). La significancia en conjunto de igual manera es significativa, pues el valor p es muy bajo. El coeficiente de determinación nos dice que el 71 % de la variabilidad en la distancia, puede ser explicada por la velocidad.

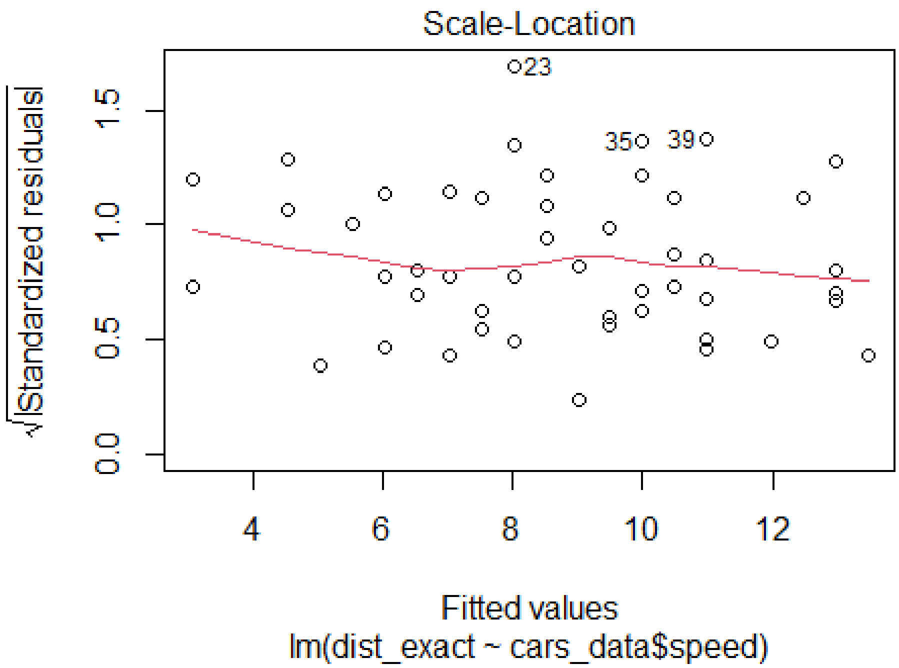
Validez del modelo

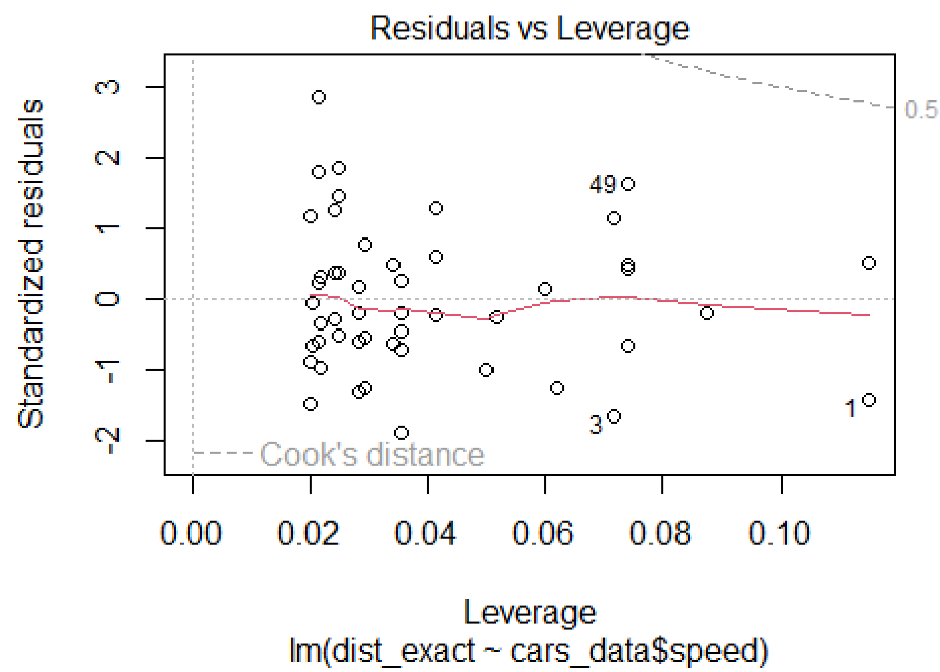
Normalidad de los residuos

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

##
##  Anderson-Darling normality test
##
## data:  Modelo2$residuals
## A = 0.34822, p-value = 0.4636

plot(Modelo2)
```


Residuals vs Fitted
Residuals
Fitted values
lm(dist_exact ~ cars_data$speed)
23
35
39


Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(dist_exact ~ cars_data$speed)
23
35
39





Se acepta h_0 ya que el valor p de los residuos es mayor a 0.05 (alfa que se esta utilizando), por lo que podemos decir que los datos provienen de una población normal.

Media de los residuos

```
t.test(Modelo2$residuals)

##
##  One Sample t-test
##
## data:  Modelo2$residuals
## t = -2.6429e-16, df = 49, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.4727457  0.4727457
## sample estimates:
##     mean of x
## -6.217357e-17
```

Podemos decir que la media de los residuos no es igual a 0. Se rechaza de igual manera h_0 puesto que el valor p de la media no es igual a cero. Por lo tanto se acepta h_1

Homocedasticidad

```
library(lmtest)
bptest(Modelo2)

##
##  studentized Breusch-Pagan test
##
## data:  Modelo2
## BP = 0.13933, df = 1, p-value = 0.709
```

Se acepta h_0 ya que el valor p propuesto por la prueba de Breusch-Pagan, es mayor a 0.05. Lo que nos dice que la varianza de los errores es constante.

Independencia

```
library(lmtest)
dwtest(Modelo2)

##
##  Durbin-Watson test
##
## data:  Modelo2
## DW = 1.9606, p-value = 0.3864
## alternative hypothesis: true autocorrelation is greater than 0
```

Se acepta h_0 puesto que el valor p propuesto por la prueba de Durbin-Watson, supera a 0.05 que es alfa. Esto nos dice que los errores no están relacionados.

Linealidad

```
library(lmtest)
resettest(Modelo2)
```

```
##
##  RESET test
##
## data:  Modelo2
## RESET = 0.68493, df1 = 2, df2 = 46, p-value = 0.5092
```

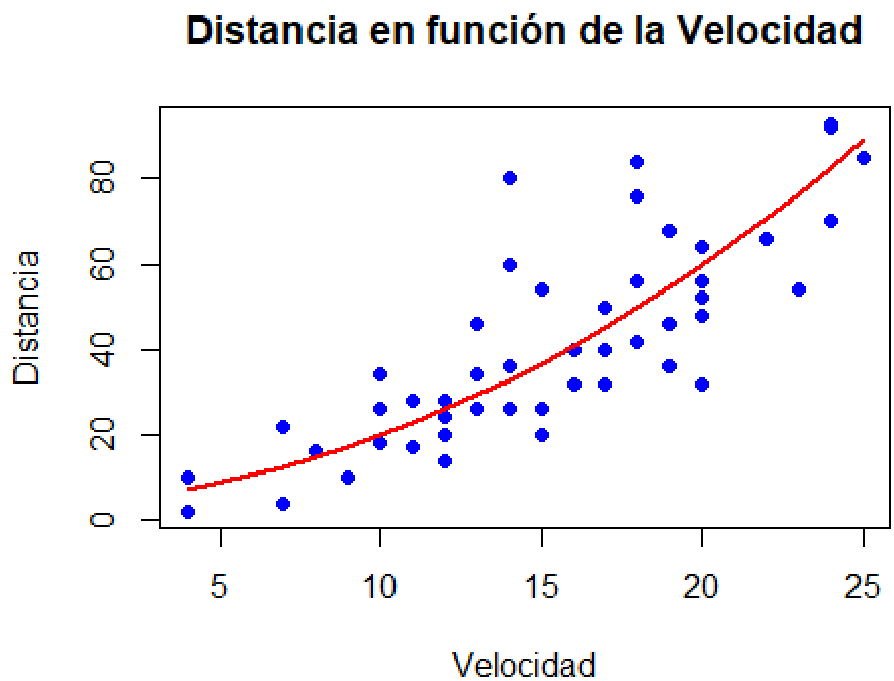
Se acepta h_0 puesto que el valor p propuesto por la prueba de RESET, supera a 0.05 que es alfa. Esto nos dice que no hay términos omitidos que indiquen linealidad.

Despejando a distancia del modelo lineal

Modelo despejado:

$$y_d = \left(\lambda(b_0 + b_1 * velocidad + 1)\right)^{\frac{1}{\lambda}}$$

```
linear_model = lm(dist_exact ~ cars_data$speed)
l = 0.4242 # lambda de la primera transformación
b0 = linear_model$coefficients[1]
b1 = linear_model$coefficients[2]
b0

## (Intercept)
##    1.082275

b1

## cars_data$speed
##       0.4954078

# Distancia despejada del modelo
distancia_no_lineal = function(velocidad){
  return(((l * (b0 + b1 * velocidad)) + 1)^(1 / l))
}

velocidades = cars_data$speed
distancias_predicted = distancia_no_lineal(velocidades)

plot(filtered_data$speed, filtered_data$dist,
     main = "Distancia en función de la Velocidad",
     xlab = "Velocidad", ylab = "Distancia",
     pch = 16, col = "blue")

# Añadir la curva del modelo no lineal
lines(velocidades, distancias_predicted, col = "red", lwd = 2)
```



Modelo propuesto:

$$y_d = \left(0.4242(1.0822 + 0.4954 * velocidad + 1)\right)^{\frac{1}{0.4242}}$$

El modelo no linear, mejor el modelo linear en varios aspectos. Además de que se la gráfica generada se incorpora mejor a los datos que la linear, da mejores resultados en significancia y validación.

Conclusiones

Tras ver todos los datos, modelos generados, y gráficas creadas; podemos concluir que el mejor modelo es el no linear, pues además de ser el que tiene un mejor coeficiente de determinación, predice de manera más certera el comportamiento de la distancia con función a la velocidad. El modelo podría mejorar si se contara con una mayor cantidad de datos, además de hacer una mejor limpieza de outliers, ya que a pesar de que se redujeron los outliers de valores bajos, cuando la distancia era muy significativa, estos valores se despegaban mucho de la normal, haciendo que el modelo no pudiera tener un mejor rendimiento.