13. Regresión no lineal

Eliezer Cavazos

2024-09-10

Parte 1: Análisis de normalidad

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

```
oData = cars
```

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

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

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

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

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

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

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

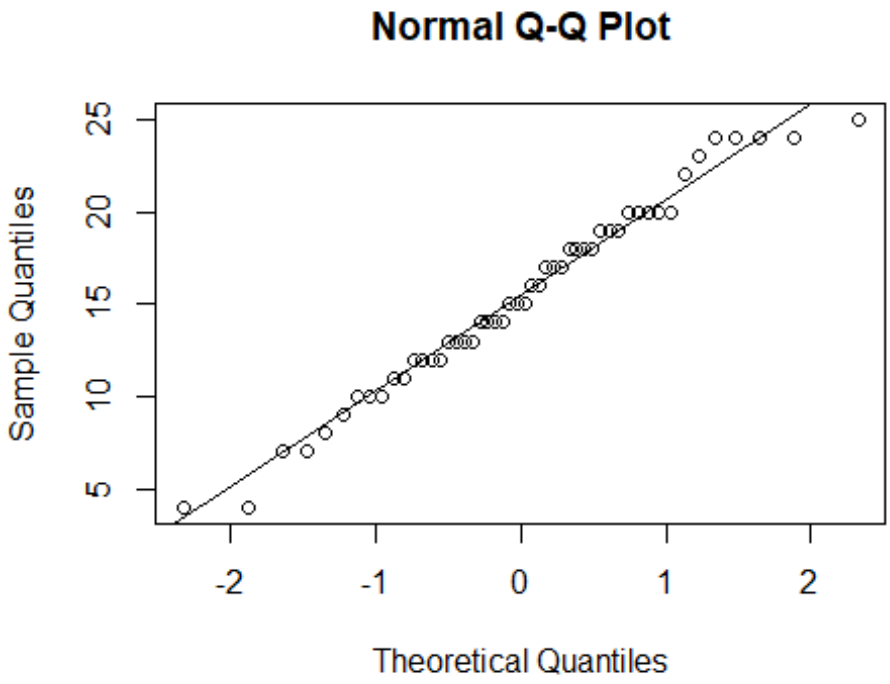
1.2.2 Realiza el histograma y su distribución teórica de probabilidad (sugerencia, adapta el código:

hist(datos,freq=FALSE) lines(density(datos),col="red")
curve(dnorm(x,mean=mean(datos),sd=sd(datos)), from=min(datos), to=max(datos), add=TRUE, col="blue",lwd=2)

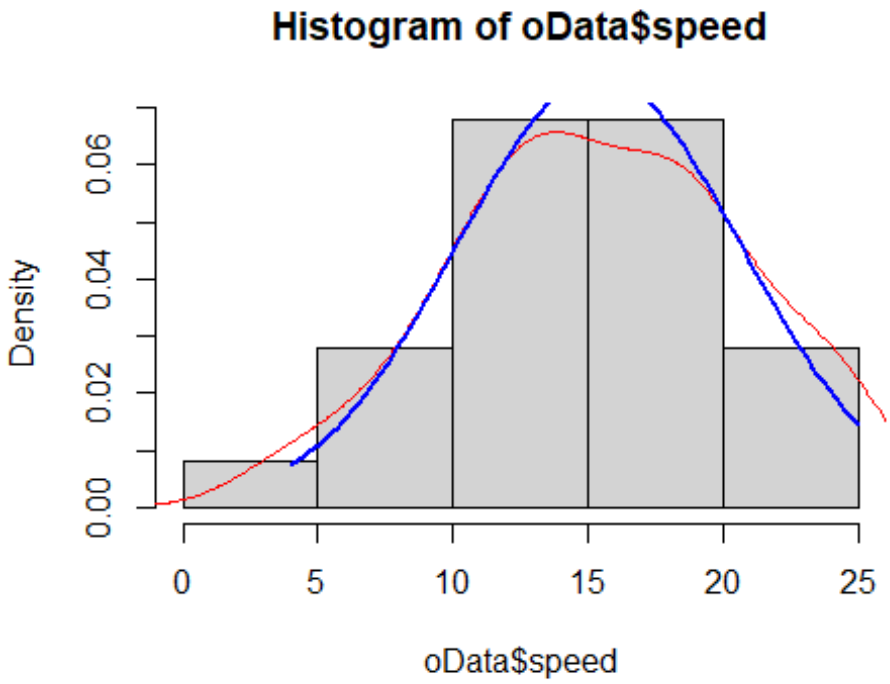
Se te sugiere usar par(mfrow=c(1,2)) para graficar el QQ plot y el histograma de una variable en un mismo espacio.

Velocidad:

```
qqnorm(oData$speed)
qqline(oData$speed)
```

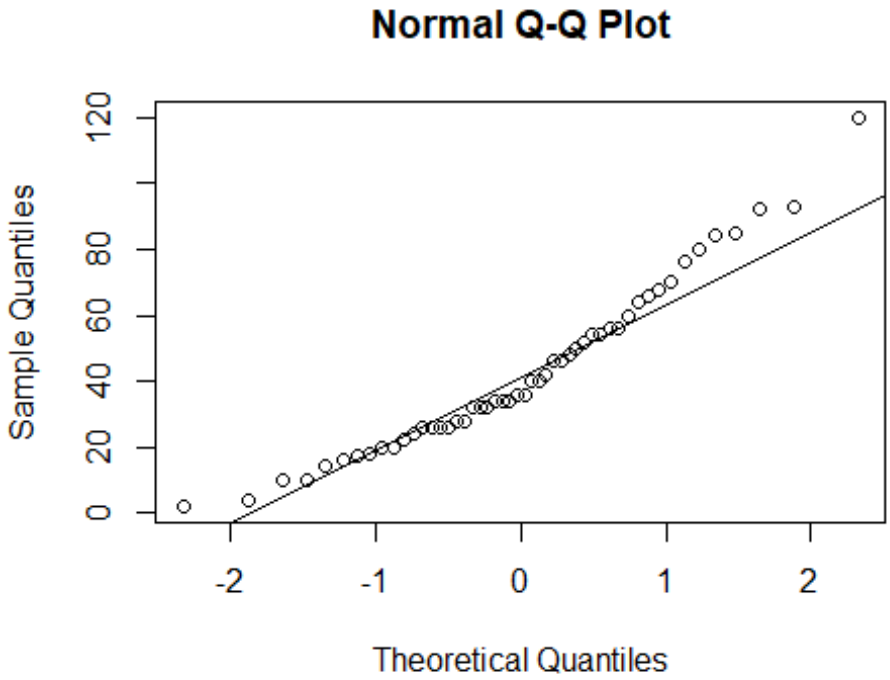



```r
hist(oData$speed, freq=FALSE)
lines(density(oData$speed), col="red")
curve(dnorm(x, mean=mean(oData$speed), sd=sd(oData$speed)),
from=min(oData$speed), to=max(oData$speed), add=TRUE, col="blue", lwd=2)
```

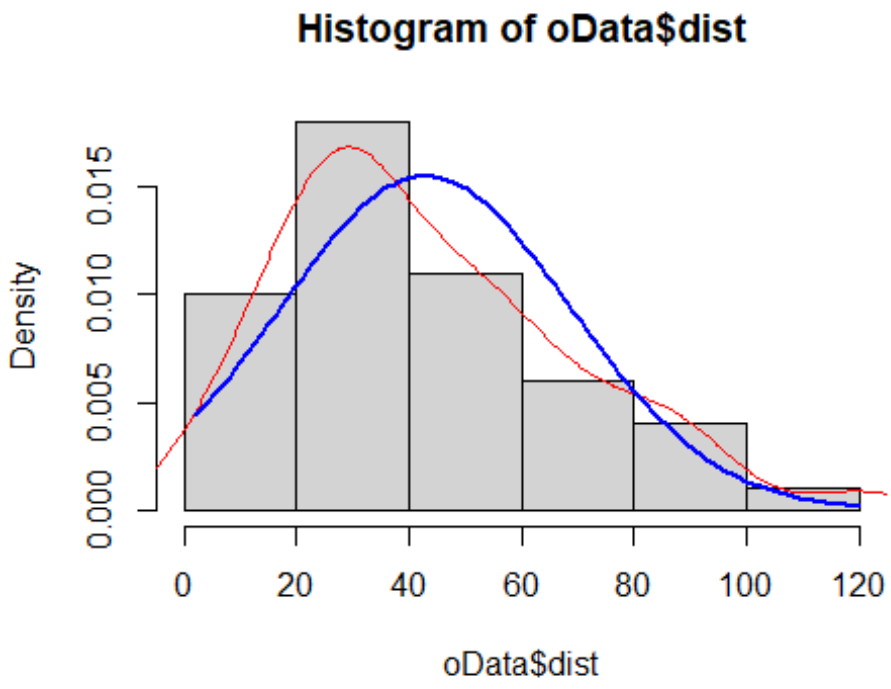


Distancia:

```
qqnorm(oData$dist)
qqline(oData$dist)
```



```r
hist(oData$dist, freq=FALSE)
lines(density(oData$dist), col="red")
curve(dnorm(x, mean=mean(oData$dist), sd=sd(oData$dist)),
from=min(oData$dist), to=max(oData$dist), add=TRUE, col="blue", lwd=2)
```



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

```r
library(e1071)
print("Velocidad: ")
```

```
## [1] "Velocidad: "
```

```r
skewness(oData$speed)
```

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

```r
kurtosis(oData$speed)
```

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

```r
print("Distancia: ")
```

```
## [1] "Distancia: "
```

```r
skewness(oData$dist)
```

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

```r
kurtosis(oData$dist)
```

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

2. 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.

Los resultado obtenidos me dicen que el valor de Velocidad tiene un menor sesgo que distancia, donde el Sesgo de Velocidad es un sesgo a la izquierda, mientras el sesgo de distancia es un sesgo a la derecha, ademas la Distancia contiene más datos atipicos que Velocidad

Parte 2: Regresión lineal

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

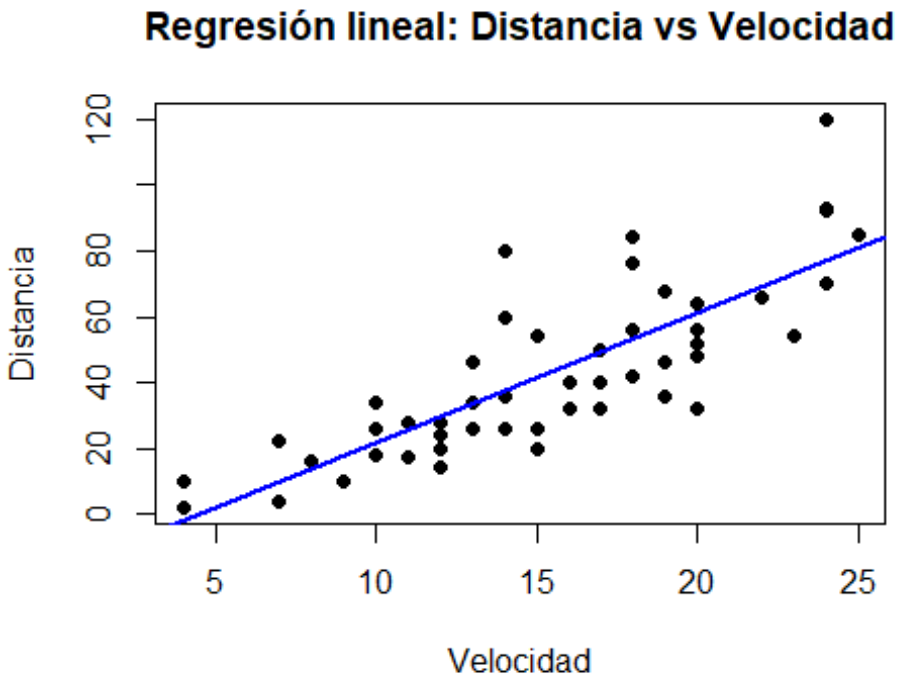
1.1 Escribe el modelo lineal obtenido.

```
oModelo = lm(dist ~ speed, data=oData)
summary(oModelo)
```

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

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

```
plot(oData$speed, oData$dist, main="Regresión lineal: Distancia vs
Velocidad",
     xlab="Velocidad", ylab="Distancia", pch=19)
abline(oModelo, col="blue", lwd=2)
```



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

3. Analiza validez del modelo.

3.1 Residuos con media cero

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

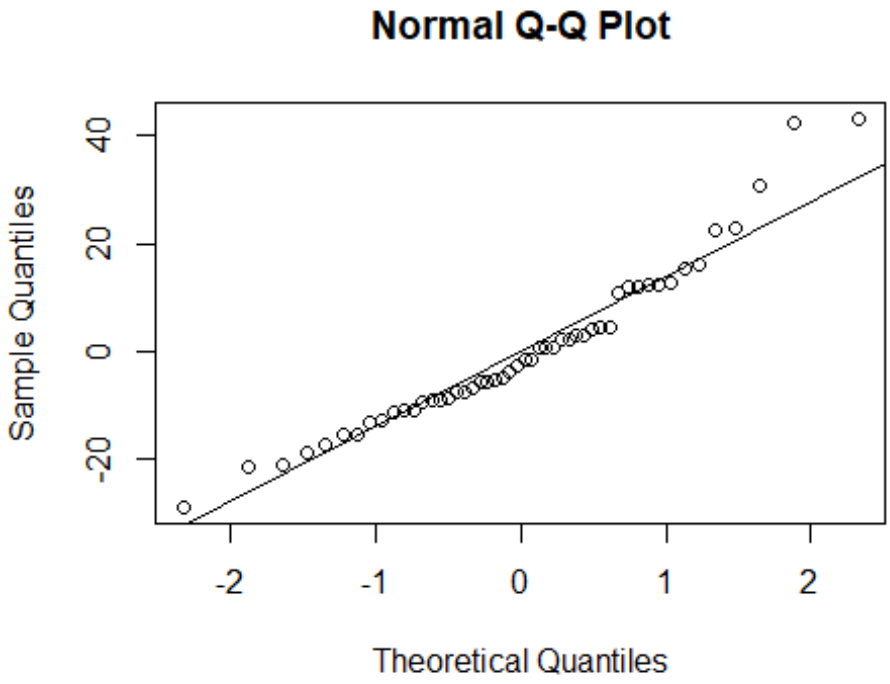
```
##
##  One Sample t-test
##
## data:  oModelo$residuals
## t = 1.0315e-16, df = 49, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -4.326  4.326
## sample estimates:
##    mean of x
## 2.220446e-16
```

3.2 Normalidad de los residuos

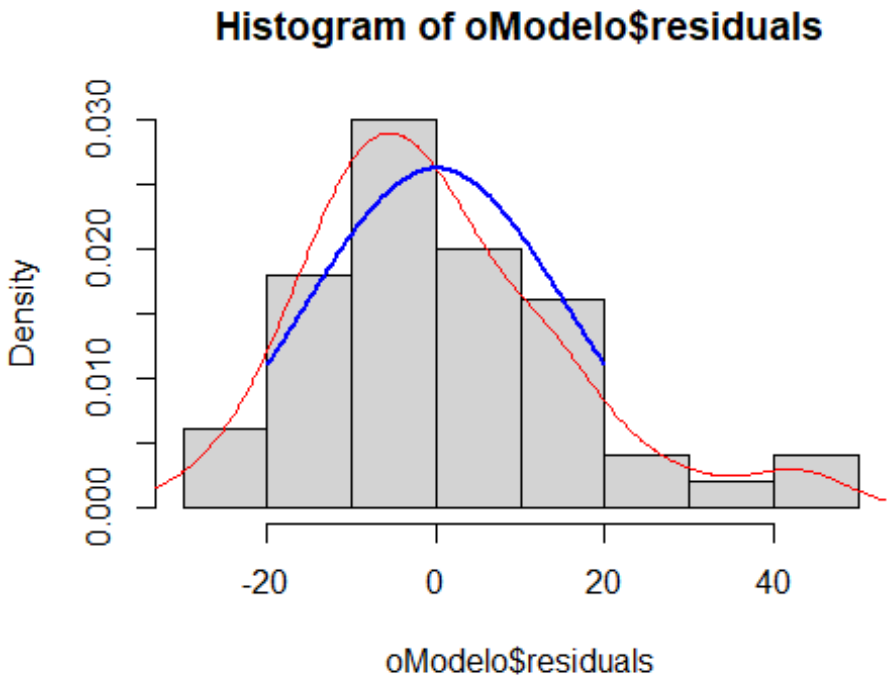
```
library(nortest)
ad.test(oModelo$residuals)
```

```
##
##  Anderson-Darling normality test
```

```
##
## data:  oModelo$residuals
## A = 0.79406, p-value = 0.0369

qqnorm(oModelo$residuals)
qqline(oModelo$residuals)
```




```
hist(oModelo$residuals,freq=FALSE)
lines(density(oModelo$residual),col="red")
curve(dnorm(x,mean=mean(oModelo$residuals),sd=sd(oModelo$residuals)), from=-
20, to=20, add=TRUE, col="blue",lwd=2)
```



3.3 Homocedasticidad, independencia y linealidad.

Homocedasticidad:

```
library(lmtest)
```

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

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

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

```
bptest(oModelo)
```

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

```
bgtest(oModelo)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
```

```
## data:  oModelo
## LM test = 1.2908, df = 1, p-value = 0.2559
```

Independencia:

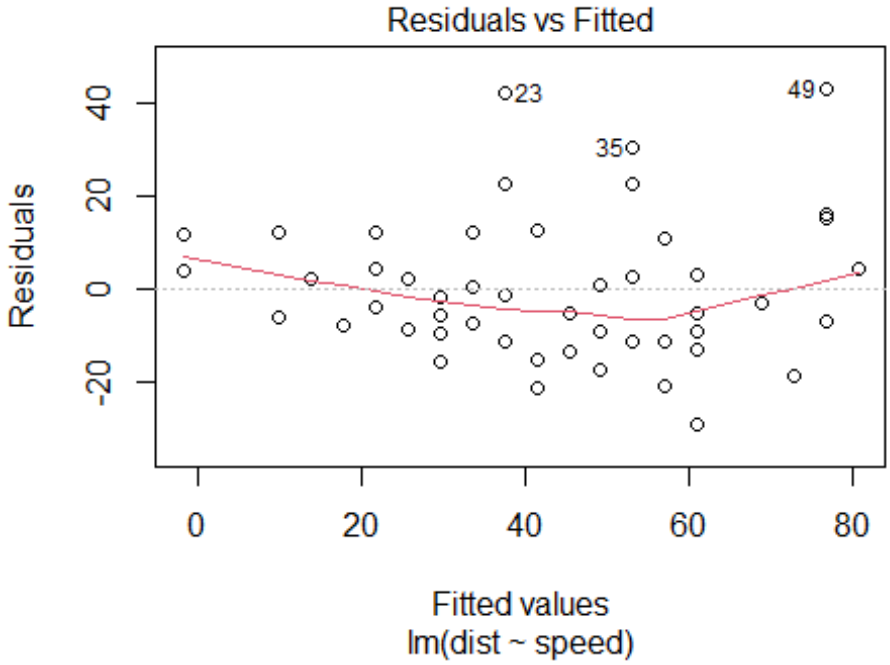
```
dwtest(oModelo)

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

gqtest(oModelo)

##
##  Goldfeld-Quandt test
##
## data:  oModelo
## GQ = 1.5512, df1 = 23, df2 = 23, p-value = 0.1498
## alternative hypothesis: variance increases from segment 1 to 2
```

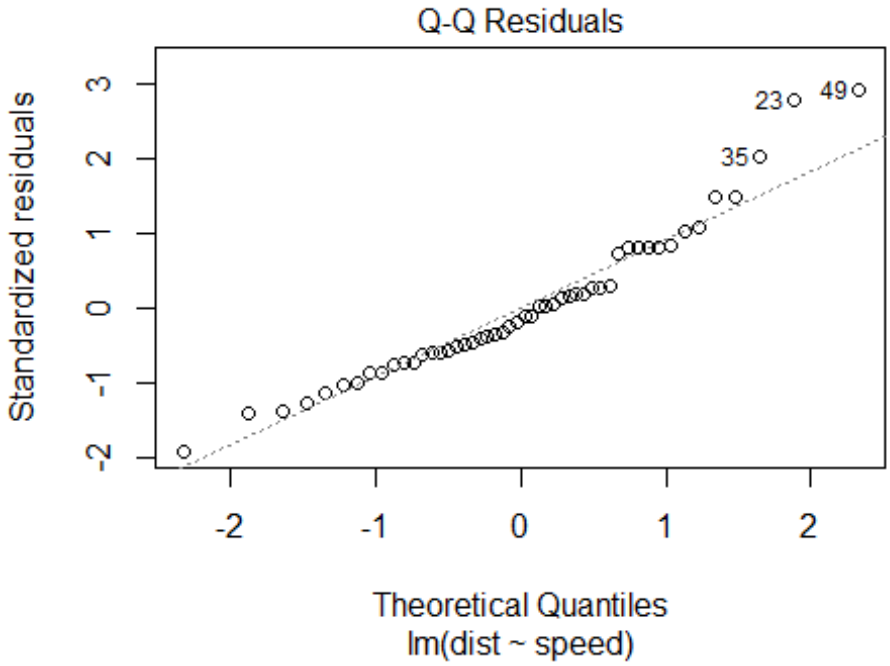
Linealidad:

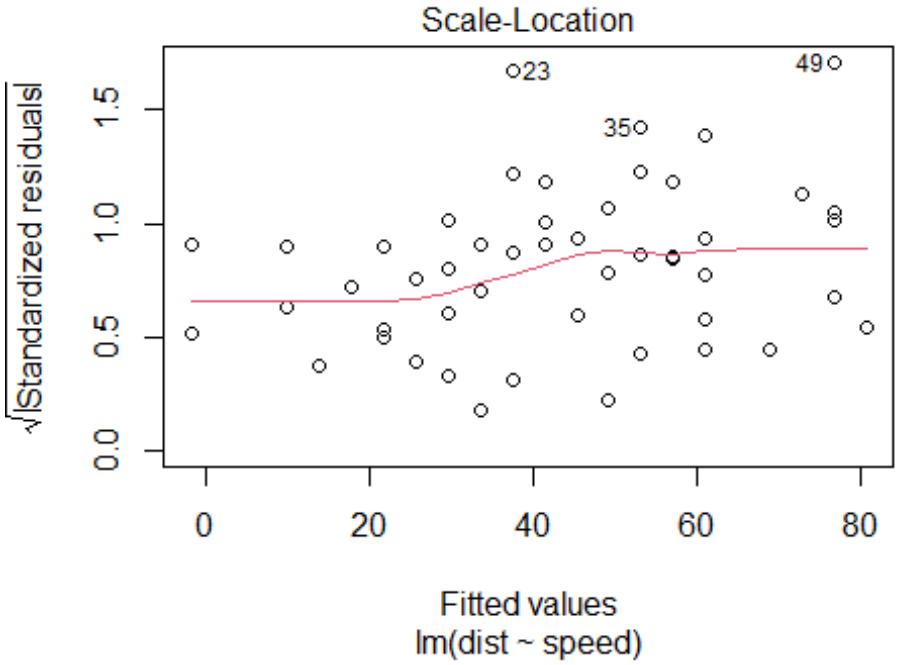
```
resettest(oModelo)

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

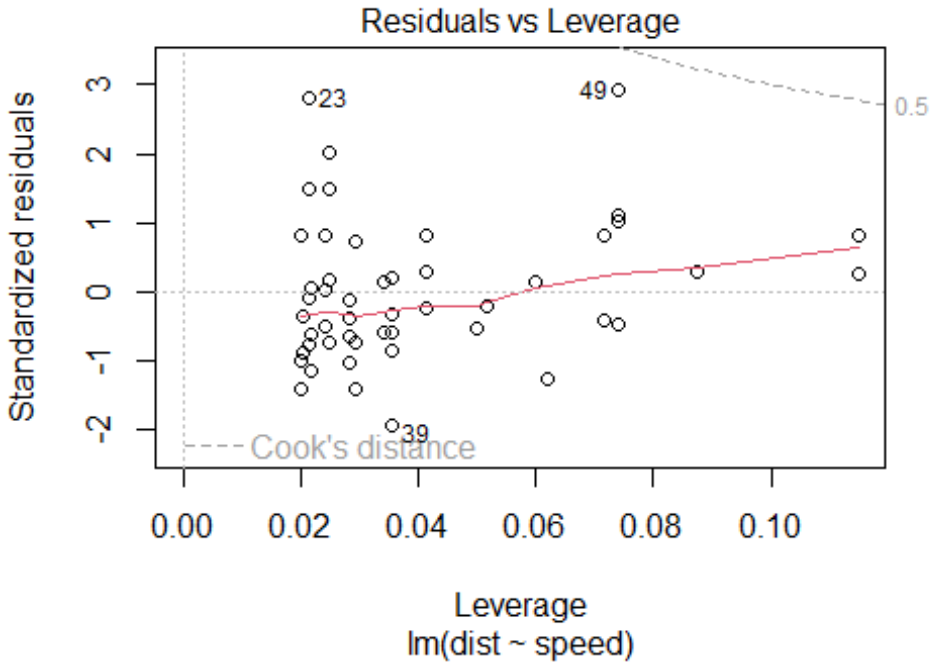
3.4 Usa plot(Modelo) para los gráficos y añade pruebas de hipótesis.

```
plot(oModelo)
```


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
-2
-1
0
1
2
3

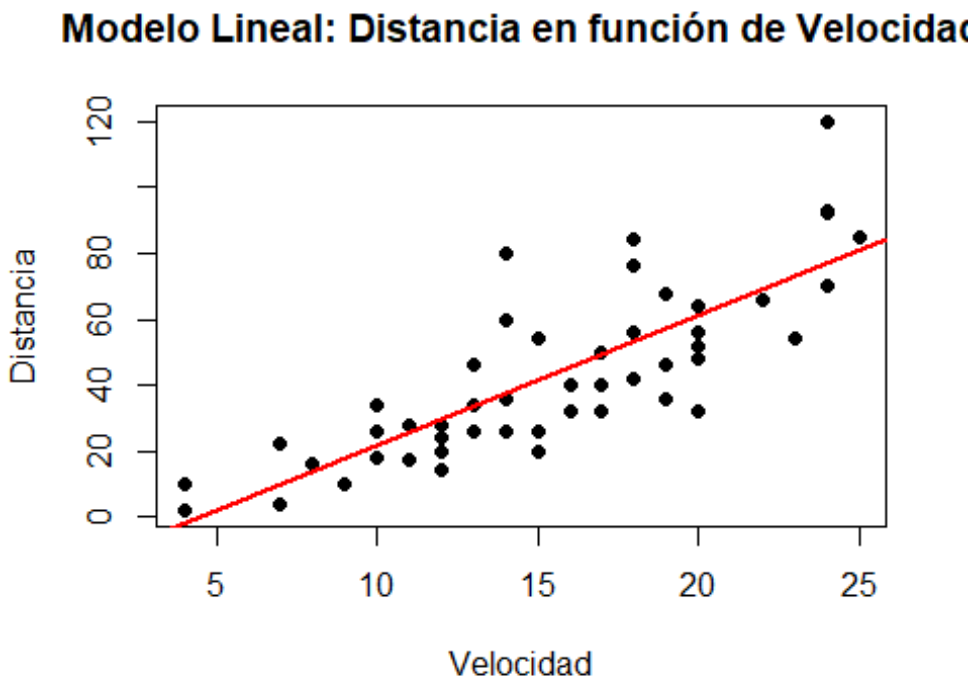




4. Grafica los datos y el modelo de la distancia en función de la velocidad.

```
# Graficar los datos y el modelo de la distancia en función de la velocidad
plot(oData$speed, oData$dist, main="Modelo Lineal: Distancia en función de
Velocidad",
```

```
     xlab="Velocidad", ylab="Distancia", pch=19)
abline(oModelo, col="red", lwd=2)
```



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

El modelo podria ser mejor si se limpiaran los datos atipicos, ya que podemos saber que si el valor de velocidad si es significativo, pero al tener muchos datos atipicos puede arruinar la calidad del modelo.

Parte 3: Regresión no lineal

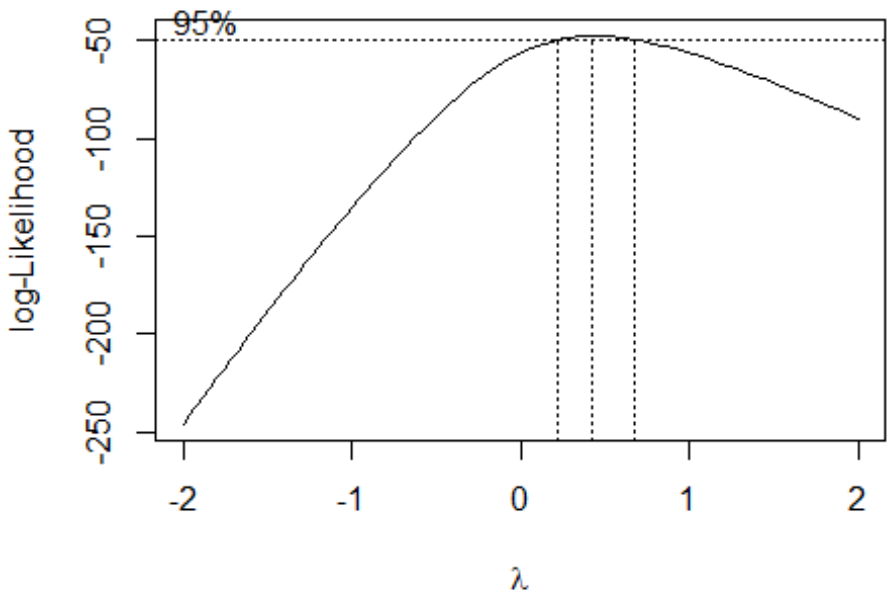
1. Con el objetivo de probar un modelo no lineal que explique la relación entre la distancia y la velocidad, haz una transformación con la base de datos car que te garantice normalidad en ambas variables (ojo: concentrate solo en la variable que tiene más alejamiento de normalidad).

1.1 Encuentra el valor de λ en la transformación Box-Cox para el modelo lineal:

$Y = \beta_0 + \beta_1$ 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: Utiliza: boxcox(lm(Distancia~Velocidad)) si la variable con más alejamiento de normalidad es la distancia Utiliza: boxcox(lm(Velocidad~Distancia)) si la variable con más alejamiento de

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

```
library(MASS)

# Aplicar la transformación Box-Cox
oBoxCoxModel = boxcox(lm(dist ~ speed, data=oData), plotit=TRUE)
```




```
oLambda <- oBoxCoxModel$x[which.max(oBoxCoxModel$y)]
oLambda

## [1] 0.4242424
```

1.2 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.

Ecu1 = $\sqrt{x+1}$ Ecu2 = $\frac{(x+1)^{0.4242424}-1}{0.4242424}$

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

Compara las medidas: sesgo y curtosis.

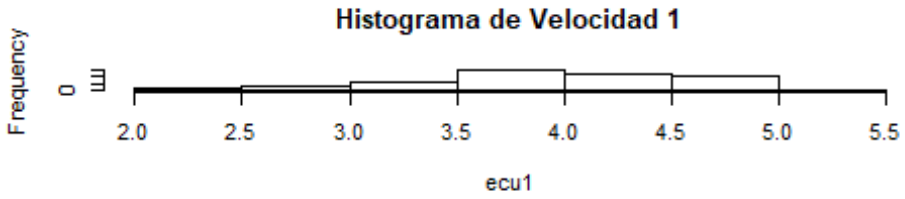
```
library(nortest)
library(e1071)
```

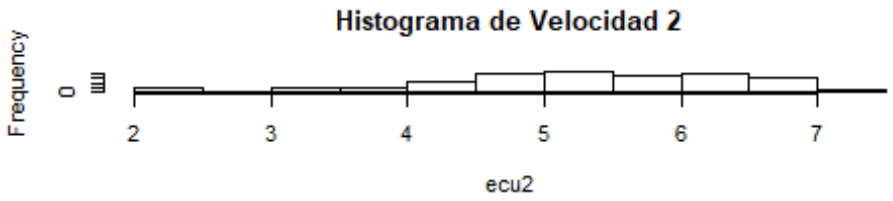
```r
oSpeed = oData$speed

ecu1=sqrt(oSpeed+1)
ecu2=((oSpeed+1)^oLambda-1)/oLambda

D0=ad.test(oSpeed)
D1=ad.test(ecu1)
D2=ad.test(ecu2)

m0=round(c(as.numeric(summary(oSpeed)),kurtosis(oSpeed),skewness(oSpeed),D0$p
.value),3)
m1=round(c(as.numeric(summary(ecu1)),kurtosis(ecu1),skewness(ecu1),D1$p.value
),3)
m2=round(c(as.numeric(summary(ecu2)),kurtosis(ecu2),skewness(ecu2),D2$p.value
),3)

m<-as.data.frame(rbind(m0,m1,m2))
row.names(m)=c("Original","Primer modelo","Segundo Modelo")
names(m)=c("Minimo","Q1","Mediana","Media","Q3","Máximo","Curtosis","Sesgo","
Valor p")

m
```

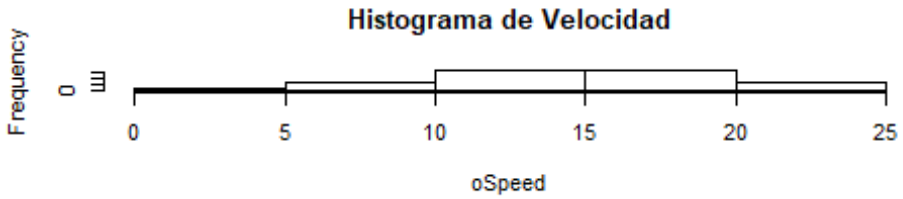
```
##                Minimo     Q1 Mediana  Media     Q3 Máximo Curtosis  Sesgo
## Original        4.000 12.000  15.000 15.400 19.000 25.000   -0.673 -0.111
## Primer modelo   2.236  3.606   4.000  3.992  4.472  5.099   -0.127 -0.529
## Segundo Modelo  2.309  4.641   5.285  5.256  6.044  7.033    0.008 -0.601
##                Valor p
## Original         0.693
## Primer modelo    0.335
## Segundo Modelo   0.249
```

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

```r
par(mfrow=c(3,1))
hist(ecu1,col=0,main="Histograma de Velocidad 1")
hist(ecu2,col=0,main="Histograma de Velocidad 2")
hist(oSpeed,col=0,main="Histograma de Velocidad")
```







Realiza algunas pruebas de normalidad para los datos transformados.

```
print("Transformacion 1")
```

```
## [1] "Transformacion 1"
```

```
ad.test(ecu1)
```

```
##
##  Anderson-Darling normality test
##
## data:  ecu1
## A = 0.40802, p-value = 0.3352
```

```
print("Transformacion 2")
```

```
## [1] "Transformacion 2"
```

```
ad.test(ecu2)
```

```
##
##  Anderson-Darling normality test
##
## data:  ecu2
## A = 0.46106, p-value = 0.2493
```

```
print("Original")
```

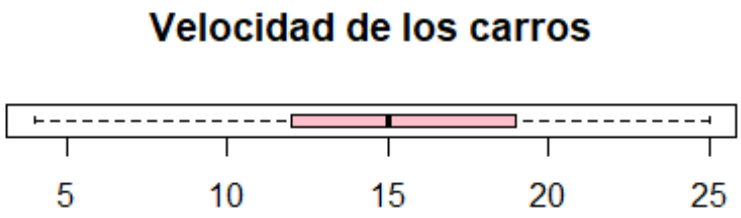
```
## [1] "Original"
```

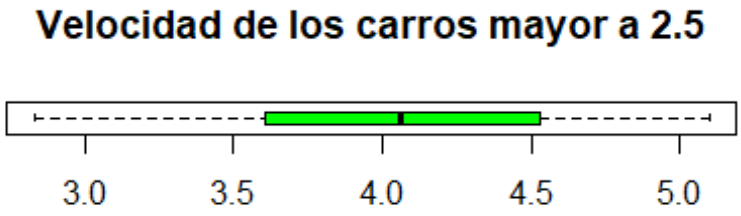
```r
ad.test(oSpeed)
```

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

1.4 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 $ $ para tus nuevos datos.

```r
M2=subset(ecu1, ecu1> 2.5)
par(mfrow=c(2,1))
boxplot(oData$speed, horizontal = TRUE,col="pink", main="Velocidad de los carros")
boxplot(M2, horizontal = TRUE,col="green", main="Velocidad de los carros mayor a 2.5")
```





2. 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.

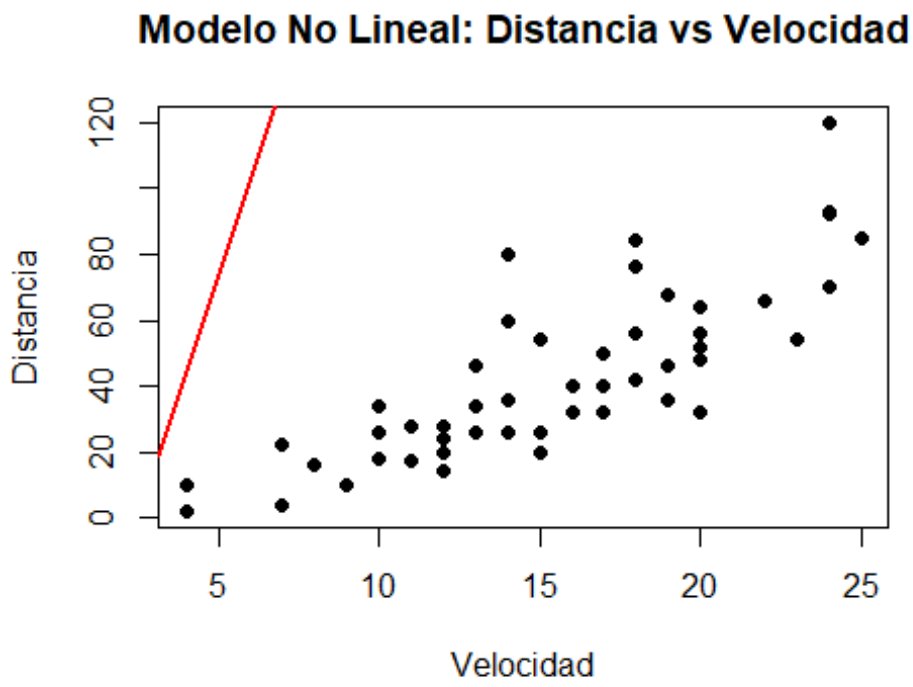
3. Con la mejor transformación (punto 2), realiza la regresión lineal simple entre la mejor transformación (exacta o aproximada) y la variable velocidad:

Escribe el modelo lineal para la transformación.

```
oModeloTransformacion = lm( dist ~ ecu1 ,data=oData) # Mejor Transformacion
```

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

```
plot(oData$speed, oData$dist, main="Modelo No Lineal: Distancia vs
Velocidad",
     xlab="Velocidad", ylab="Distancia", pch=19)
abline(oModeloTransformacion, col="red", lwd=2)
```



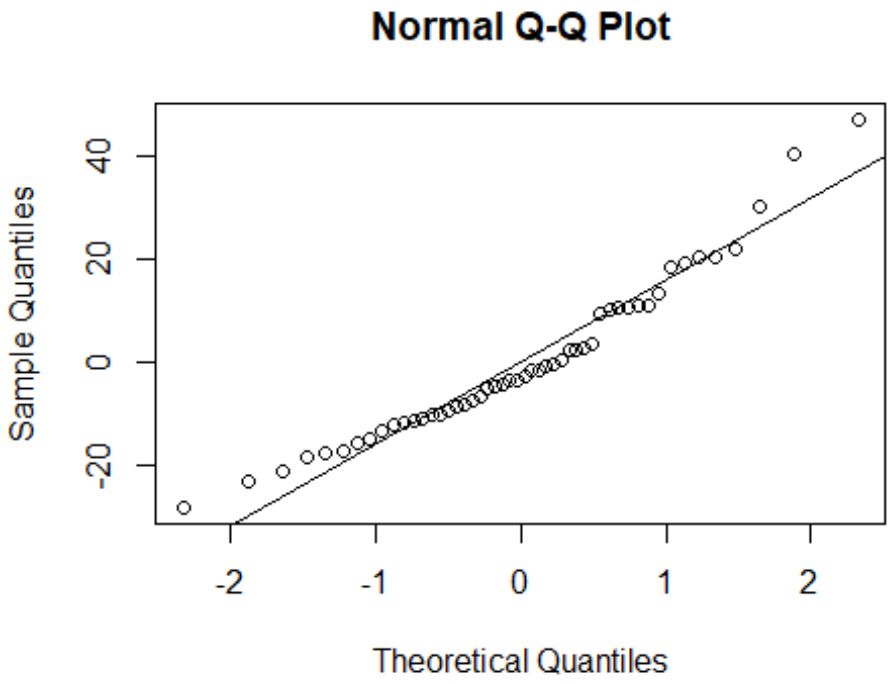
Analiza significancia del modelo (individual, conjunta y coeficiente de correlación)

```
summary(oModeloTransformacion)

##
## Call:
## lm(formula = dist ~ ecu1, data = oData)
```

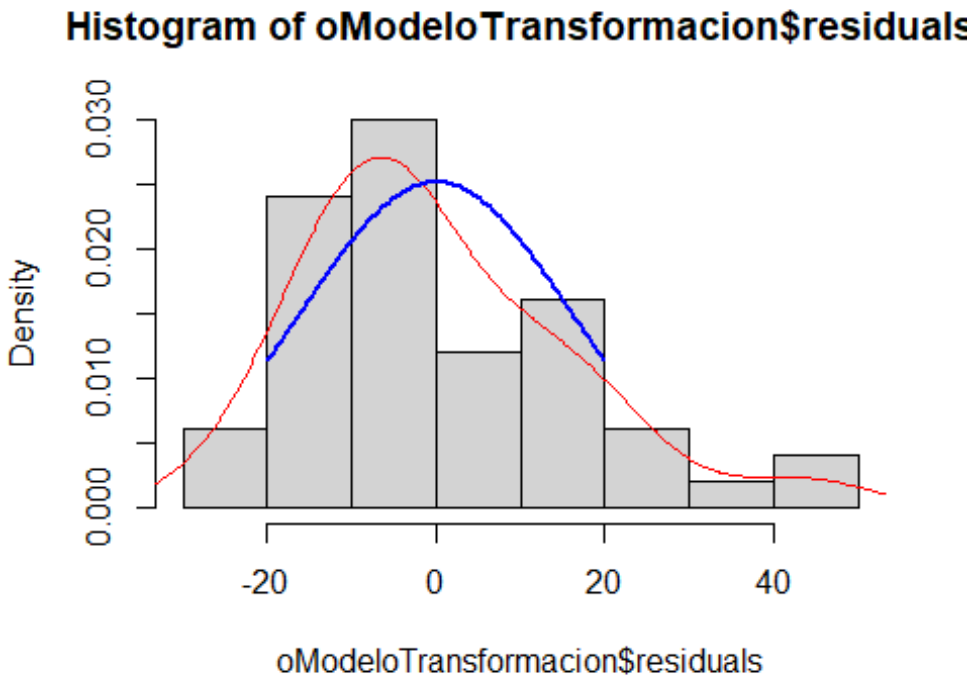
```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.377 -10.956  -3.076  10.624  47.334
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -74.536     13.429  -5.550 1.21e-06 ***
## ecu1          29.440      3.316   8.878 1.07e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.02 on 48 degrees of freedom
## Multiple R-squared:  0.6215, Adjusted R-squared:  0.6136
## F-statistic: 78.82 on 1 and 48 DF,  p-value: 1.072e-11
```

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.

```
t.test(oModeloTransformacion$residuals)

##
##  One Sample t-test
##
## data:  oModeloTransformacion$residuals
## t = -1.8492e-18, df = 49, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -4.505532  4.505532
## sample estimates:
##     mean of x
## -4.145989e-18

ad.test(oModeloTransformacion$residuals)

##
##  Anderson-Darling normality test
##
## data:  oModeloTransformacion$residuals
## A = 0.78117, p-value = 0.03975

qqnorm(oModeloTransformacion$residuals)
qqline(oModeloTransformacion$residuals)
```

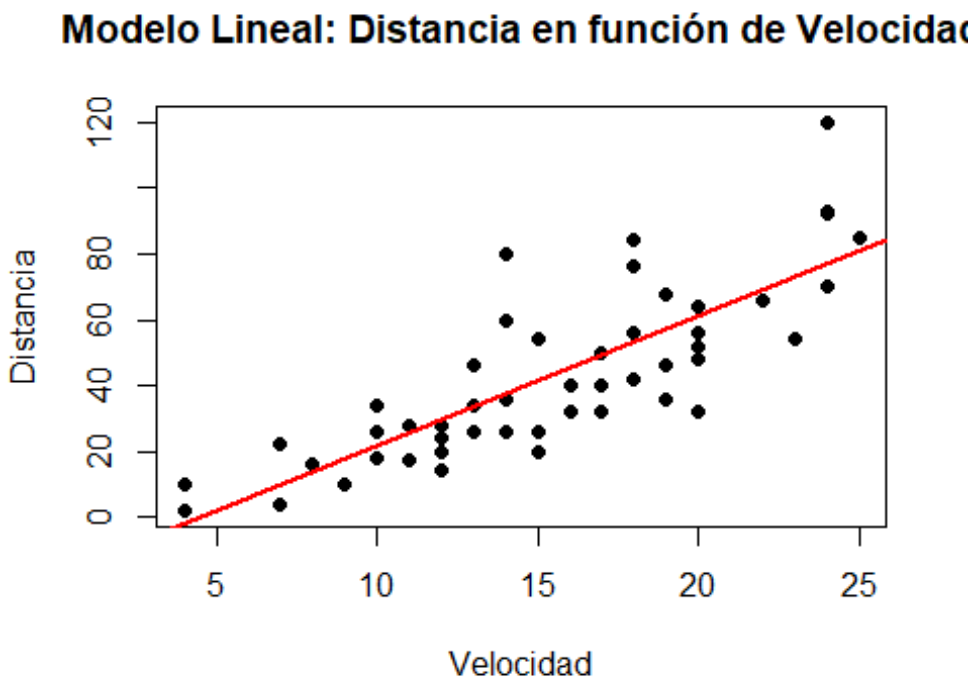



```
hist(oModeloTransformacion$residuals,freq=FALSE)
lines(density(oModeloTransformacion$residual),col="red")
curve(dnorm(x,mean=mean(oModeloTransformacion$residuals),sd=sd(oModeloTransfo
rmacion$residuals)), from=-
20, to=20, add=TRUE, col="blue",lwd=2)
```



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.

```
# Graficar los datos y el modelo de la distancia en función de la velocidad
plot(oData$speed, oData$dist, main="Modelo Lineal: Distancia en función de
Velocidad",
     xlab="Velocidad", ylab="Distancia", pch=19)
abline(oModelo, col="red", lwd=2)
```



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

El nuevo modelo si normalizo los datos, bajo un poco la significancia de los datos sobre distancia, pero creo yo que tiene que ver porque se quitaron datos eso hizo que bajara la significancia aunque fuera muy poco

Parte 4: Conclusión

1. 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 que identifique es el nuevo modelo con los datos transformados, ya que normalizo los datos e hizo que el modelo fuera de mejor calidad.

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

Algunos problemas del nuevo modelo son los datos atipicos usando qqplot se puede identificar que se tiene datos atipicos en la mitad de los datos, solo que no encontre como poder limpiarlos para mejorar la transformacion.