Práctica Tema 6

Álvaro Miranda García

2023-03-27

```
y_cuentas = c(110,2,6,98,40,94,31,5,8,10)
x_distancia = c(1.1,100.2,90.3,5.4,57.5,6.6,34.7,65.8,57.9,86.1)
```

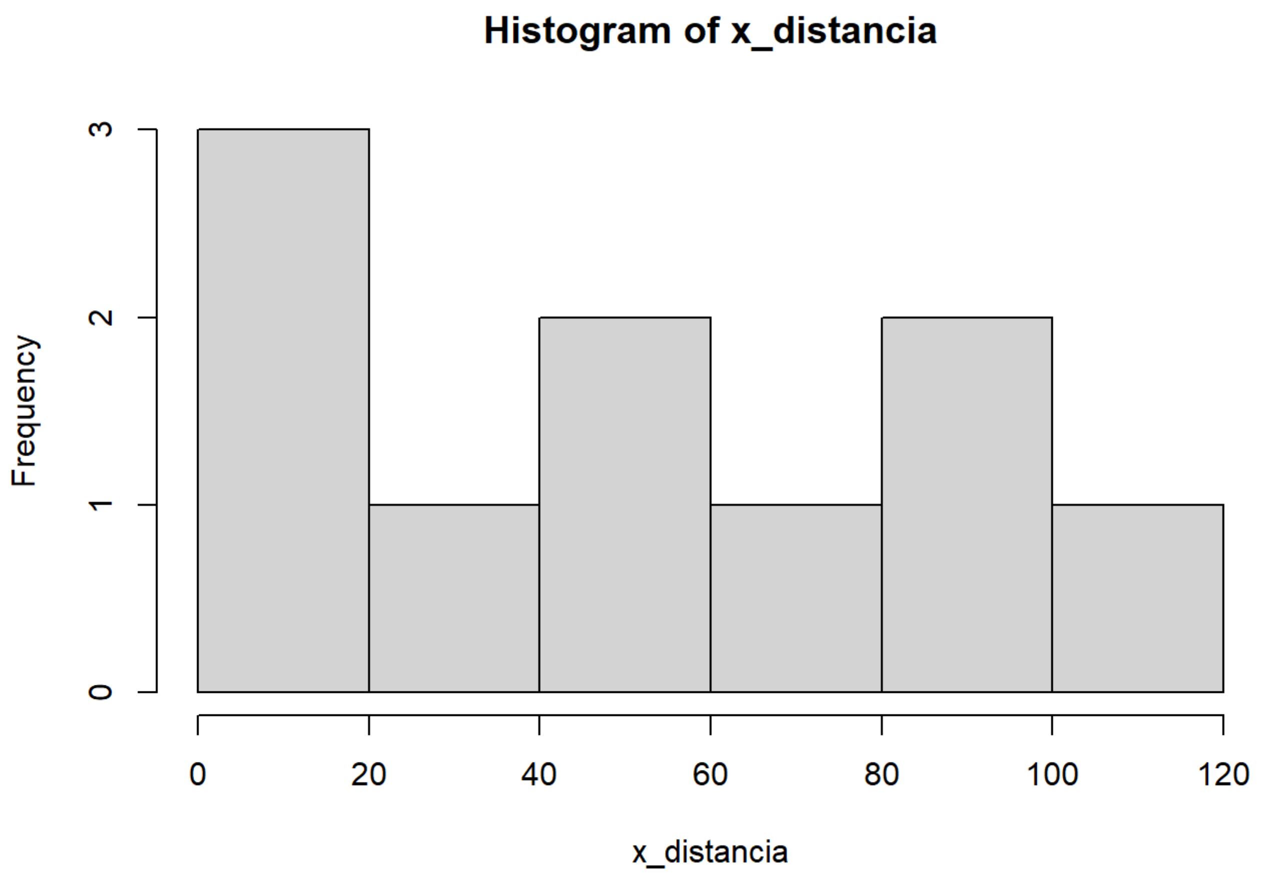
#2

```
modelo <- lm(y_cuentas ~ x_distancia)
anova(modelo)
```

```
## Analysis of Variance Table
##
## Response: y_cuentas
##             Df  Sum Sq Mean Sq F value    Pr(>F)
## x_distancia  1 14552.3   14552  47.399 0.0001265 ***
## Residuals    8  2456.1     307
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

#3

```
hist(x_distancia)
```



```
shapiro.test(x_distancia)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  x_distancia
## W = 0.90687, p-value = 0.2602
```

#4

```
xy <- y_cuentas * x_distancia
```

#5

```
x_cuadrado <- x_distancia^2
```

#6

```
tabla_datos <- data.frame(y_cuentas, x_distancia, xy, x_cuadrado)
```

#7

```
library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 4.2.3
```

```
kable(tabla_datos)
```

y_cuentas	x_distancia	xy	x_cuadrado
110	1.1	121.0	1.21
2	100.2	200.4	10040.04
6	90.3	541.8	8154.09
98	5.4	529.2	29.16
40	57.5	2300.0	3306.25
94	6.6	620.4	43.56
31	34.7	1075.7	1204.09
5	65.8	329.0	4329.64
8	57.9	463.2	3352.41
10	86.1	861.0	7413.21

#8

```
sumatorio <- rowSums(tabla_datos)
```

#9

```
tabla_datos <- rbind(tabla_datos, sumatorio)
```

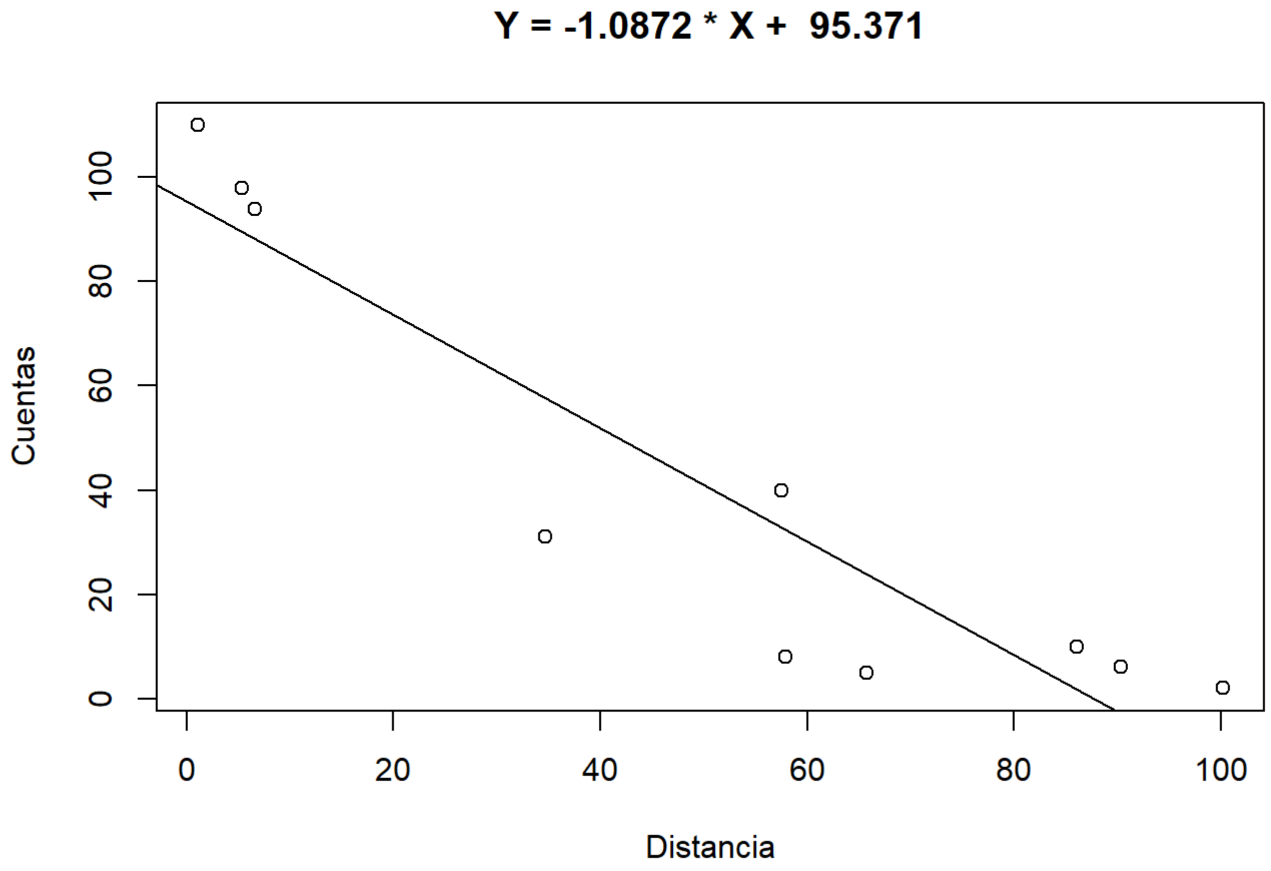
```
## Warning in rbind(deparse.level, ...): number of columns of result, 4, is not a
## multiple of vector length 10 of arg 2
```

#10

```
modelo <- lm(y_cuentas ~ x_distancia)
```

#11

```
plot(x_distancia, y_cuentas, main = paste("Y =", round(modelo$coefficients[2],4),"* X + ", ro
und(modelo$coefficients[1],4)), xlab = "Distancia", ylab = "Cuentas")
abline(modelo)
```



#12

```
residuos <- resid(modelo)
residuos_estandarizados <- rstandard(modelo)
residuos_estudentizados <- rstudent(modelo)
```

#13

```
predict(modelo, newdata = data.frame(x_distancia = 6.6))
```

```
##        1
## 88.19523
```

#14

```
set.seed(12345)
entrenamiento <- sample(1:nrow(tabla_datos), 0.7*nrow(tabla_datos))
validacion <- setdiff(1:nrow(tabla_datos), entrenamiento)
```

#15

```
modelo_entrenamiento <- lm(y_cuentas ~ x_distancia, data = tabla_datos, subset = entrenamient
o)
```

#16

```
summary(modelo_entrenamiento)
```

```
## 
## Call:
## lm(formula = y_cuentas ~ x_distancia, data = tabla_datos, subset = entrenamiento)
## 
## Residuals:
##         3        10         8         2         5        11         9 
##  -6.14754  -2.05697  -6.61921 -10.36103  28.55978   0.07382  -3.44885 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 10.200249   5.877396   1.736    0.143    
## x_distancia  0.021565   0.001503  14.345 2.97e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 14.29 on 5 degrees of freedom
## Multiple R-squared:  0.9763, Adjusted R-squared:  0.9715 
## F-statistic: 205.8 on 1 and 5 DF,  p-value: 2.967e-05
```

#17

Los asteriscos significan que el coeficiente de regresión es significativo estadísticamente para un nivel de significación del 5%.

#18

Los grados de libertad se calculan restando el número de observaciones menos el número de parámetros estimados. En este caso, es 8.

#19

```
#Varianza explicada:
ssr = sum(residuos^2)

#Varianza no explicada:
sse = sum((y_cuentas - predict(modelo))^2)
```

#20

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.2.3
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
cv <- trainControl(method = "cv", number = 10)
modelo_cross <- train(y_cuentas ~ x_distancia, data = tabla_datos, method = "lm", trControl = 
cv)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

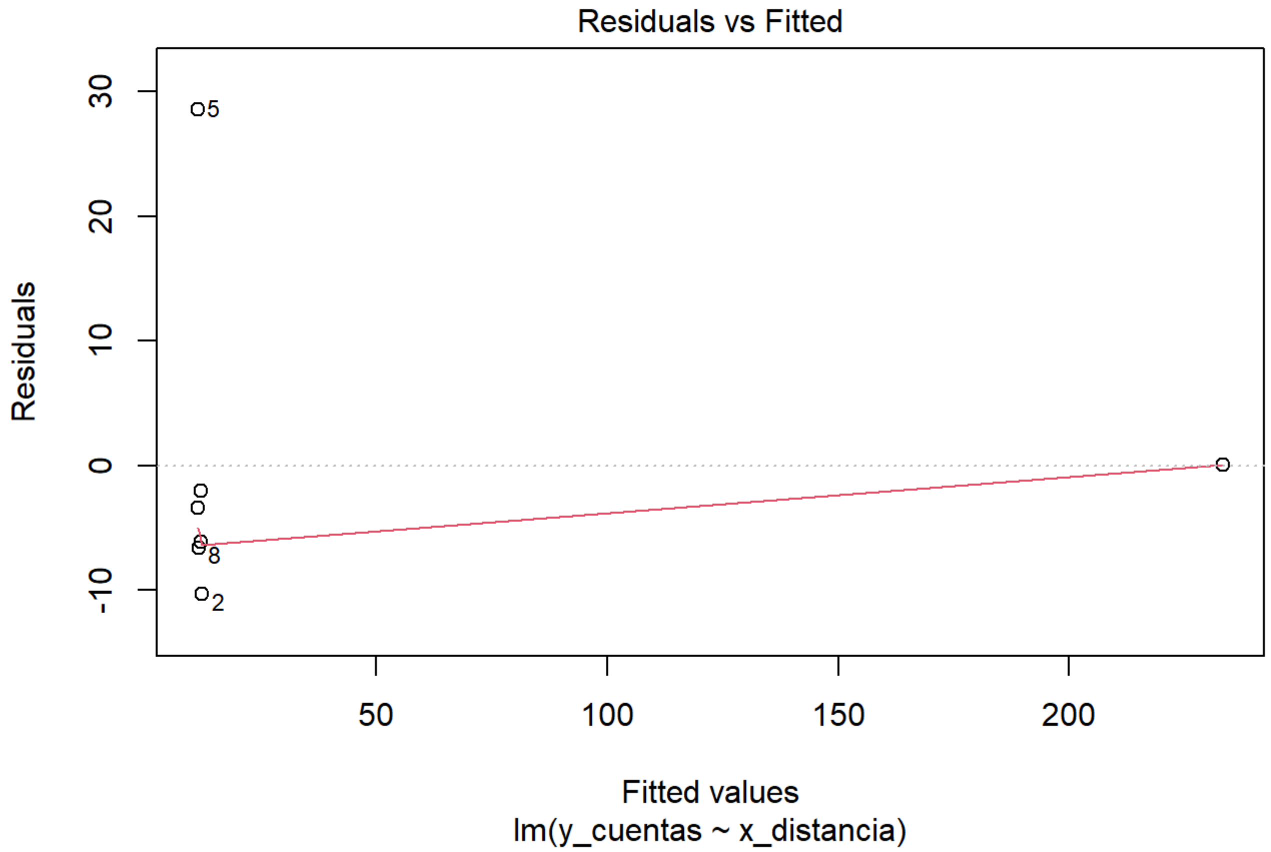
#21

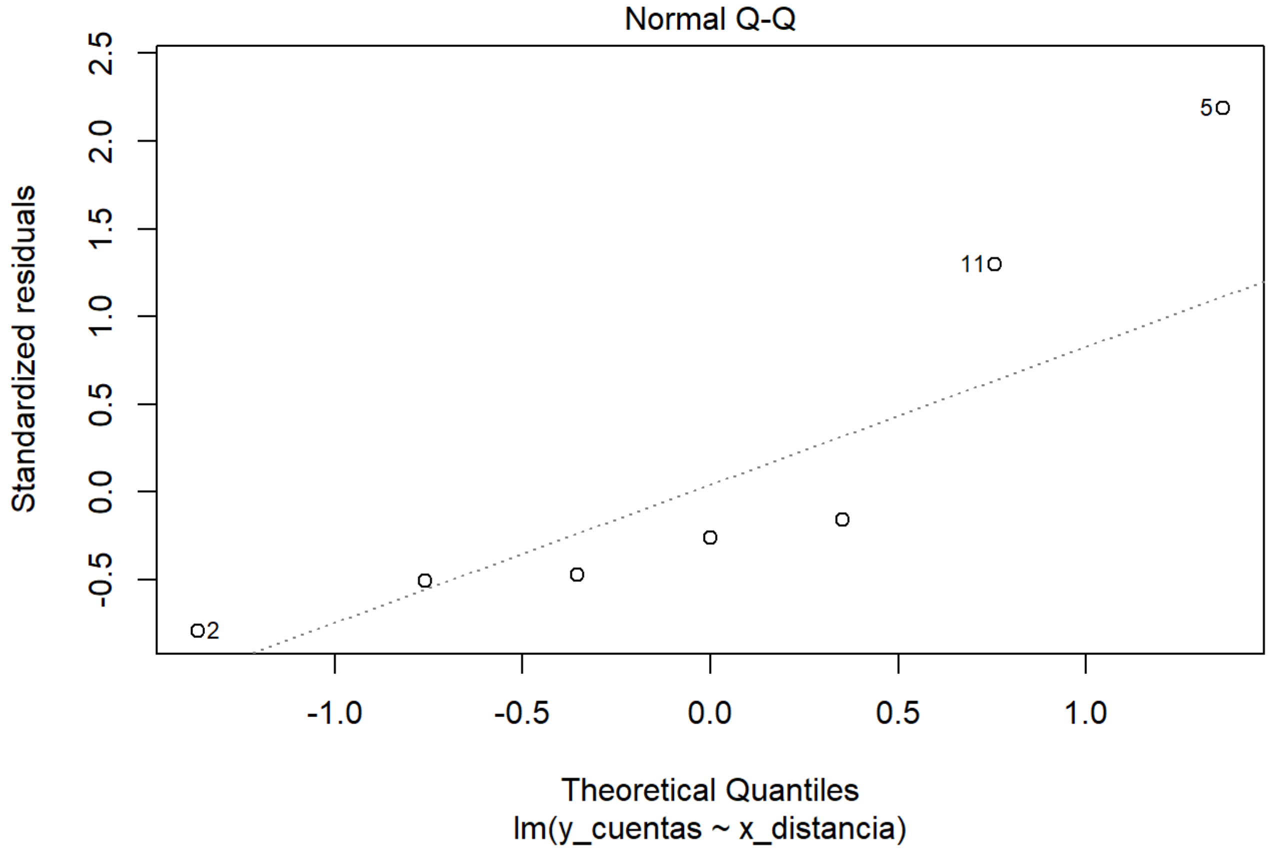
```
influence.measures(modelo_entrenamiento)
```

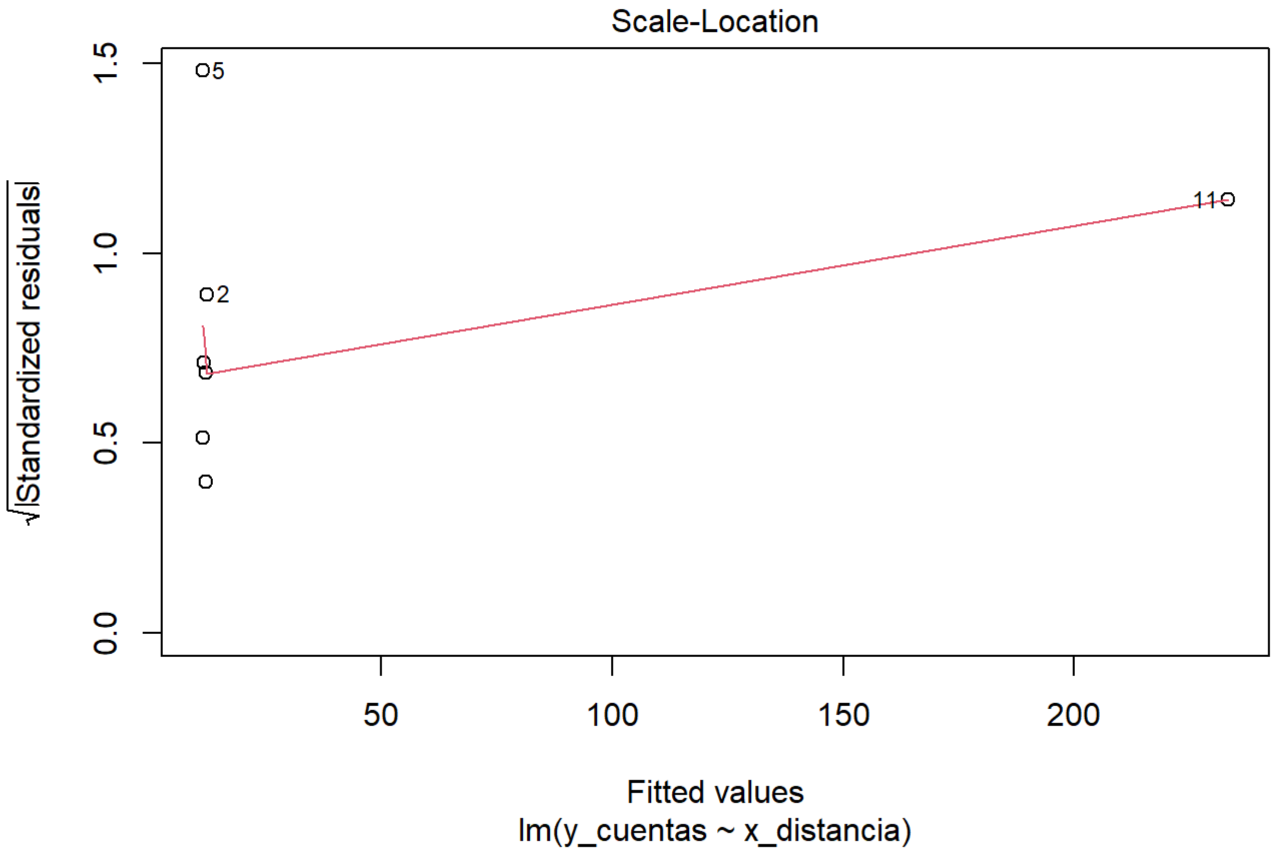
```
## Influence measures of
##   lm(formula = y_cuentas ~ x_distancia, data = tabla_datos, subset = entrenamiento) :
## 
##     dfb.1_ dfb.x_ds    dffit    cov.r   cook.d   hat inf
## 3  -0.1924   0.0722  -0.1925 1.71e+00 2.21e-02 0.166    
## 10 -0.0631   0.0237  -0.0632 1.86e+00 2.48e-03 0.166    
## 8  -0.2087   0.0794  -0.2087 1.69e+00 2.58e-02 0.167    
## 2  -0.3387   0.1263  -0.3388 1.43e+00 6.27e-02 0.166    
## 5   4.3686  -1.6694   4.3690 3.06e-03 4.82e-01 0.167   *
## 11 -6.4902 332.6935 359.3506 4.34e+04 5.34e+04 1.000   *
## 9  -0.1068   0.0408  -0.1068 1.82e+00 7.03e-03 0.167    
```

#22

```
plot(modelo_entrenamiento)
```

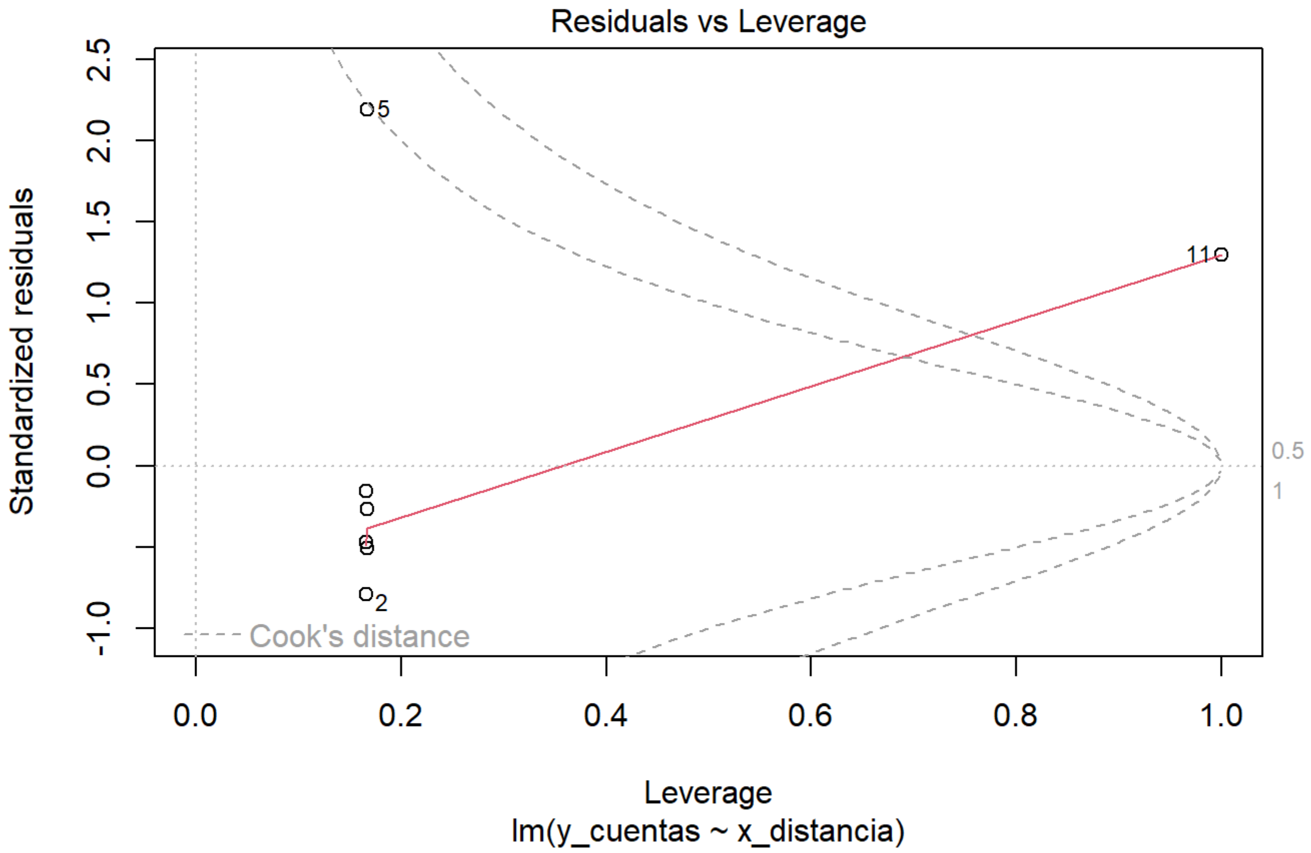

Residuals vs Fitted
Residuals
30
20
10
0
-10
5
8
2
50
100
150
200
Fitted values
lm(y_cuentas ~ x_distancia)


Normal Q-Q
Standardized residuals
2.5
2.0
1.5
1.0
0.5
0.0
-0.5
5
11
2
-1.0
-0.5
0.0
0.5
1.0
Theoretical Quantiles
lm(y_cuentas ~ x_distancia)



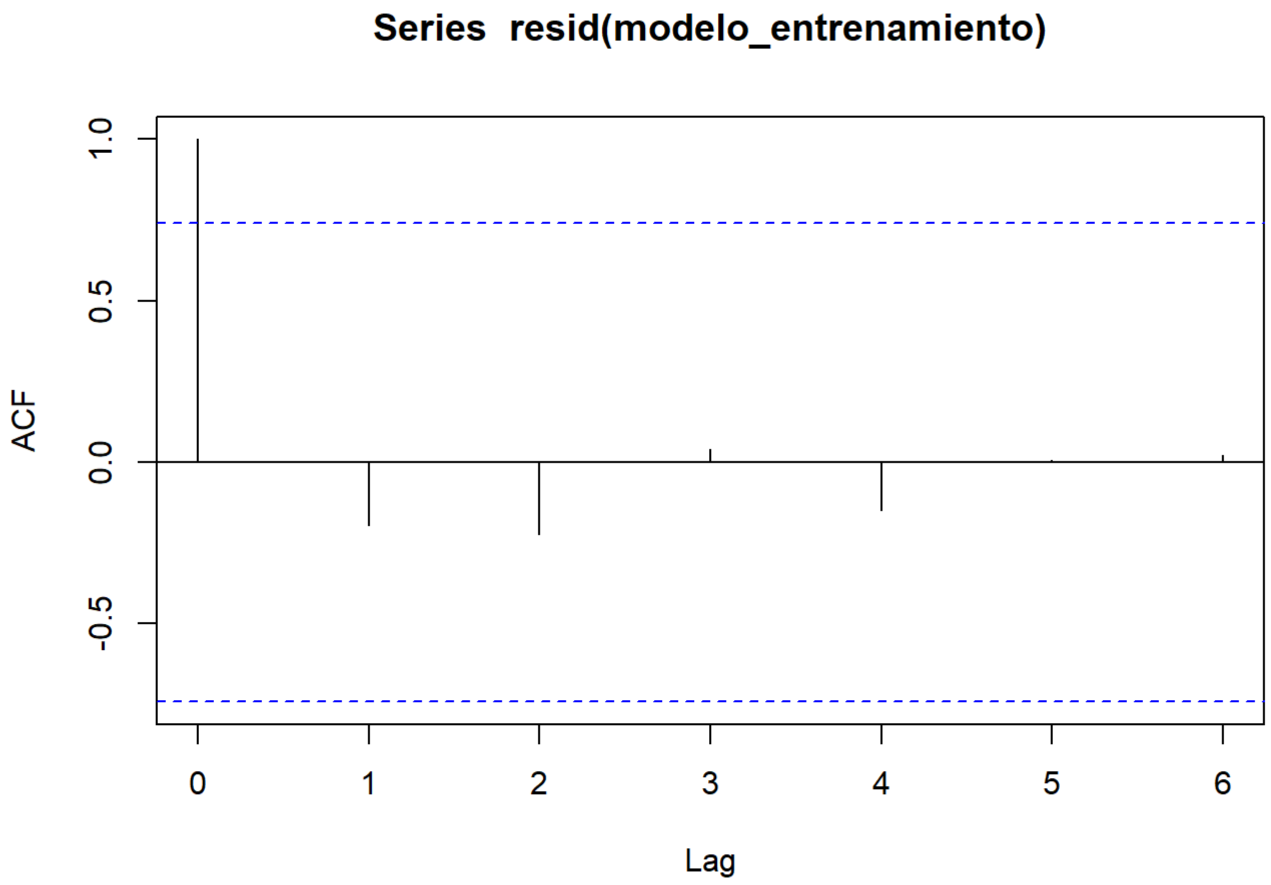
```
## Warning in sqrt(crit * p * (1 - hh)/hh): Se han producido NaNs

## Warning in sqrt(crit * p * (1 - hh)/hh): Se han producido NaNs
```



#23

```
acf(resid(modelo_entrenamiento))
```


Series resid(modelo_entrenamiento)
ACF
1.0
0.5
0.0
-0.5
0
1
2
3
4
5
6
Lag