2020_02_27

Fernando Anorve

3/3/2020

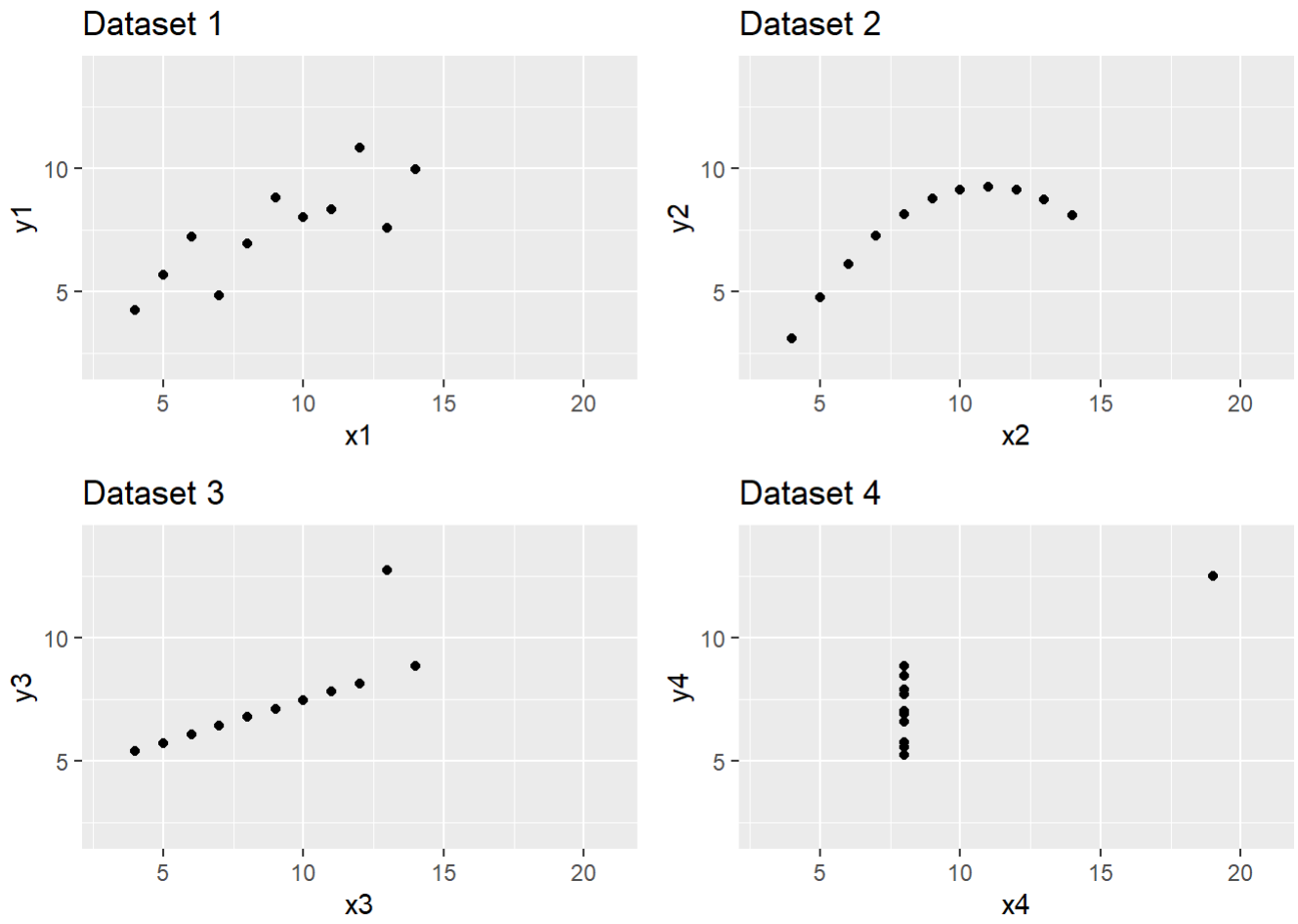
Regresion

Ejemplo 1 - Anscombe dataset

Por que no hay que confiar ciegamente en un coeficiente? No siempre R^2 tiene la razón absoluta.

Para verlo usaremos un conjunto de datos bastante conocido “anscombe”, construido por el estadístico Francis Anscombe en 1975 para demostrar la importancia de graficar datos y buscar comportamientos atípicos antes de construir modelos. Esta colección consta de cuatro conjuntos con los que usualmente se calculan modelos de regresión lineal simple.

```
data("anscombe")

p1 <- qplot(x1, y1, data = anscombe) + ggtitle("Dataset 1") + ylim(c(2,14)) + xlim(c(3,21))
p2 <- qplot(x2, y2, data = anscombe) + ggtitle("Dataset 2") + ylim(c(2,14)) + xlim(c(3,21))
p3 <- qplot(x3, y3, data = anscombe) + ggtitle("Dataset 3") + ylim(c(2,14)) + xlim(c(3,21))
p4 <- qplot(x4, y4, data = anscombe) + ggtitle("Dataset 4") + ylim(c(2,14)) + xlim(c(3,21))

grid.arrange(p1, p2, p3 , p4, nrow = 2)
```



¡Hagan sus apuestas! (no realmente)

- ¿Cual creen que tenga “mejor valor” de R^2?
- ¿Cuál línea creen que tenga una pendiente más pronunciada?

Pero más importante:

- ¿Cuál(es) creen que se ajuste mejor a un modelo lineal?

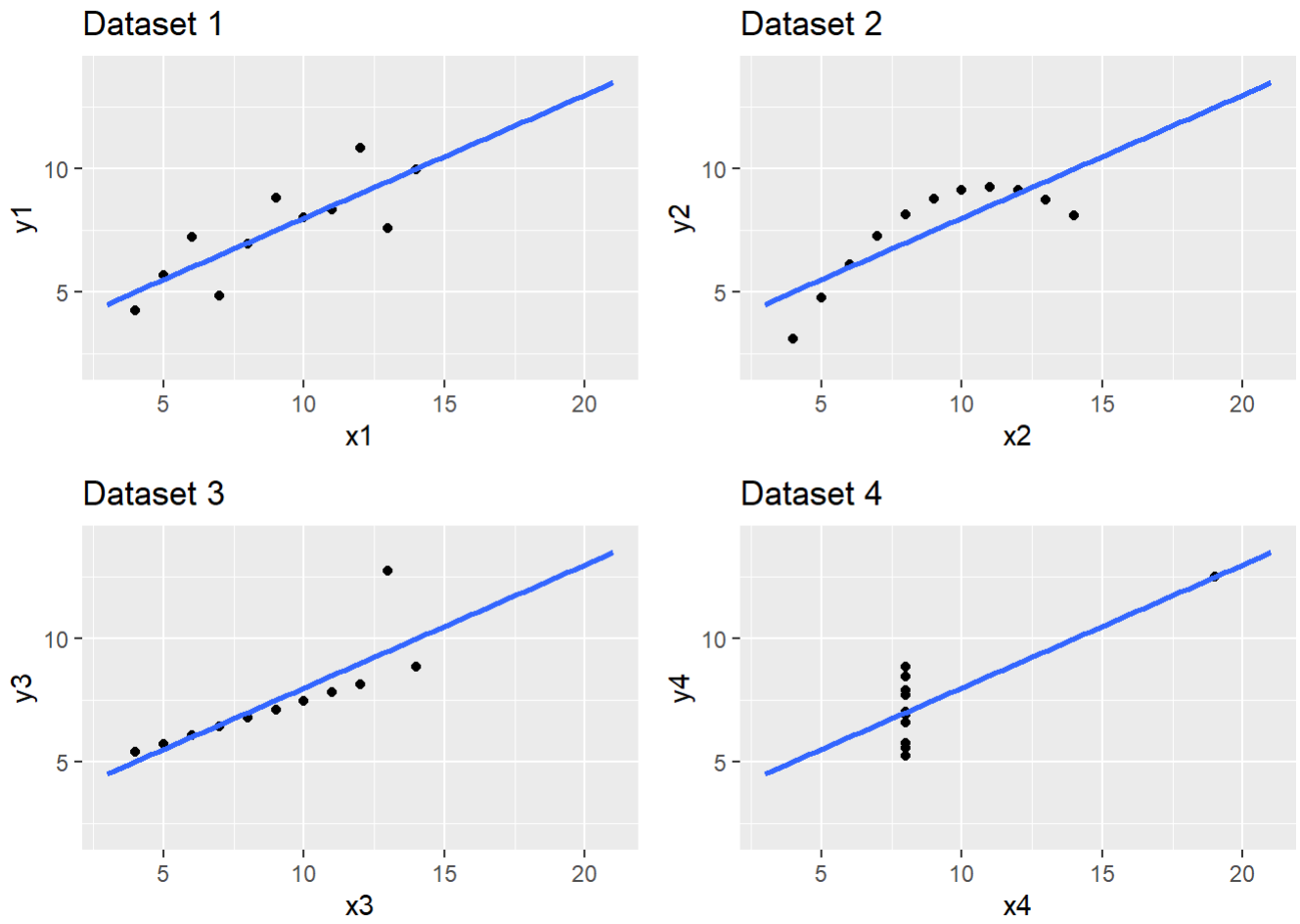
```
lm1 <- lm(y1 ~ x1, data = anscombe)
lm2 <- lm(y2 ~ x2, data = anscombe)
lm3 <- lm(y3 ~ x3, data = anscombe)
lm4 <- lm(y4 ~ x4, data = anscombe)


info <- rbind(lm1$coefficients,lm2$coefficients,lm3$coefficients,lm4$coefficients)
info <- cbind(info , c(summary(lm1)$r.squared , summary(lm2)$r.squared , summary(lm3)$r.squared
, summary(lm4)$r.squared ))
colnames(info)[2:3] = c("(slope)", "r.squared")

round(info, digits = 2)
```

```
##      (Intercept) (slope) r.squared
## [1,]           3     0.5      0.67
## [2,]           3     0.5      0.67
## [3,]           3     0.5      0.67
## [4,]           3     0.5      0.67
```

¡En realidad tienen los mismos coeficientes y el mismo R^2 !

Ya con la linea de ajuste, vemos que no todos quedan igual

```
p1 <- p1 + geom_smooth(method='lm',se=F, fullrange = T)
p2 <- p2 + stat_smooth(method='lm',se=F, fullrange = T)
p3 <- p3 + stat_smooth(method='lm',se=F, fullrange = T)
p4 <- p4 + stat_smooth(method='lm',se=F, fullrange = T)

grid.arrange(p1, p2, p3 , p4, nrow = 2)
```



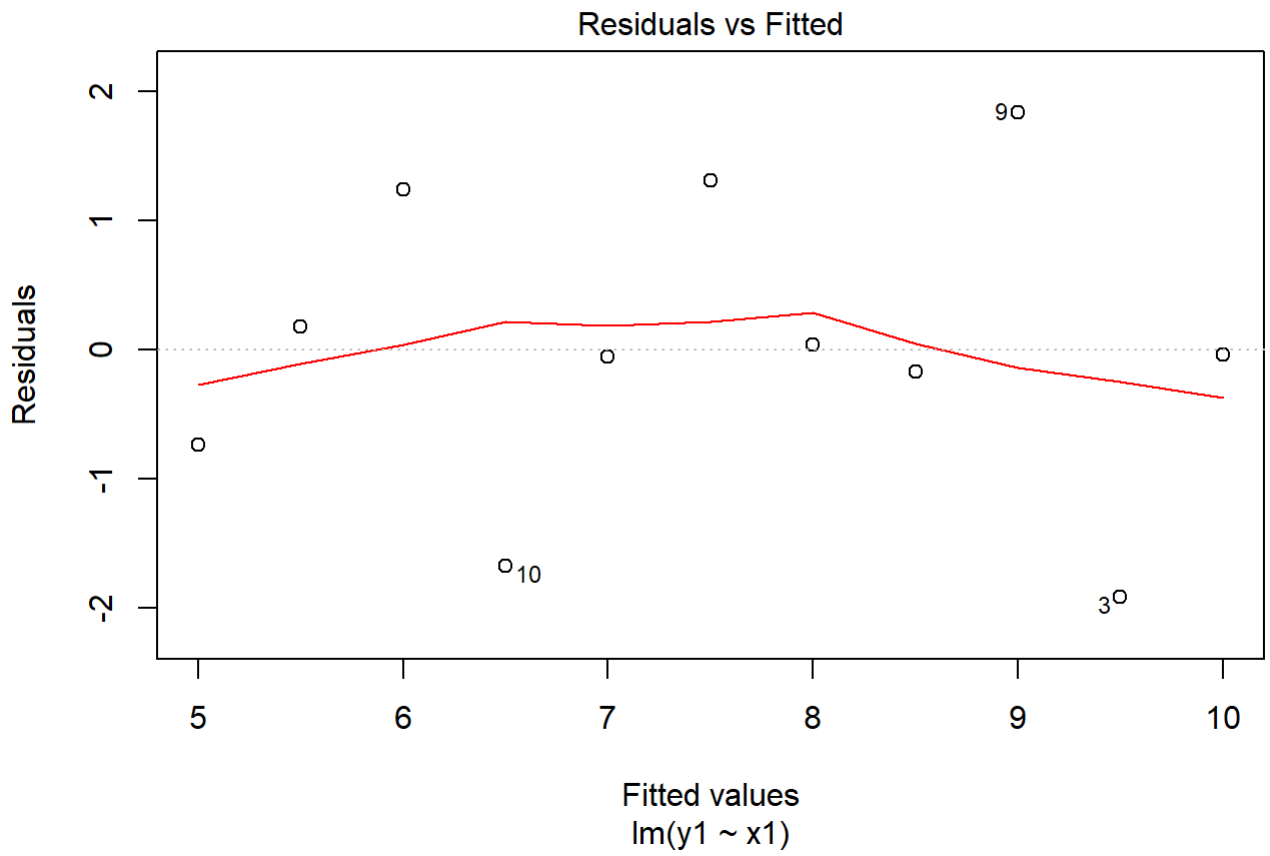
Veamos por ejemplo qué pasa con los primeros tres:

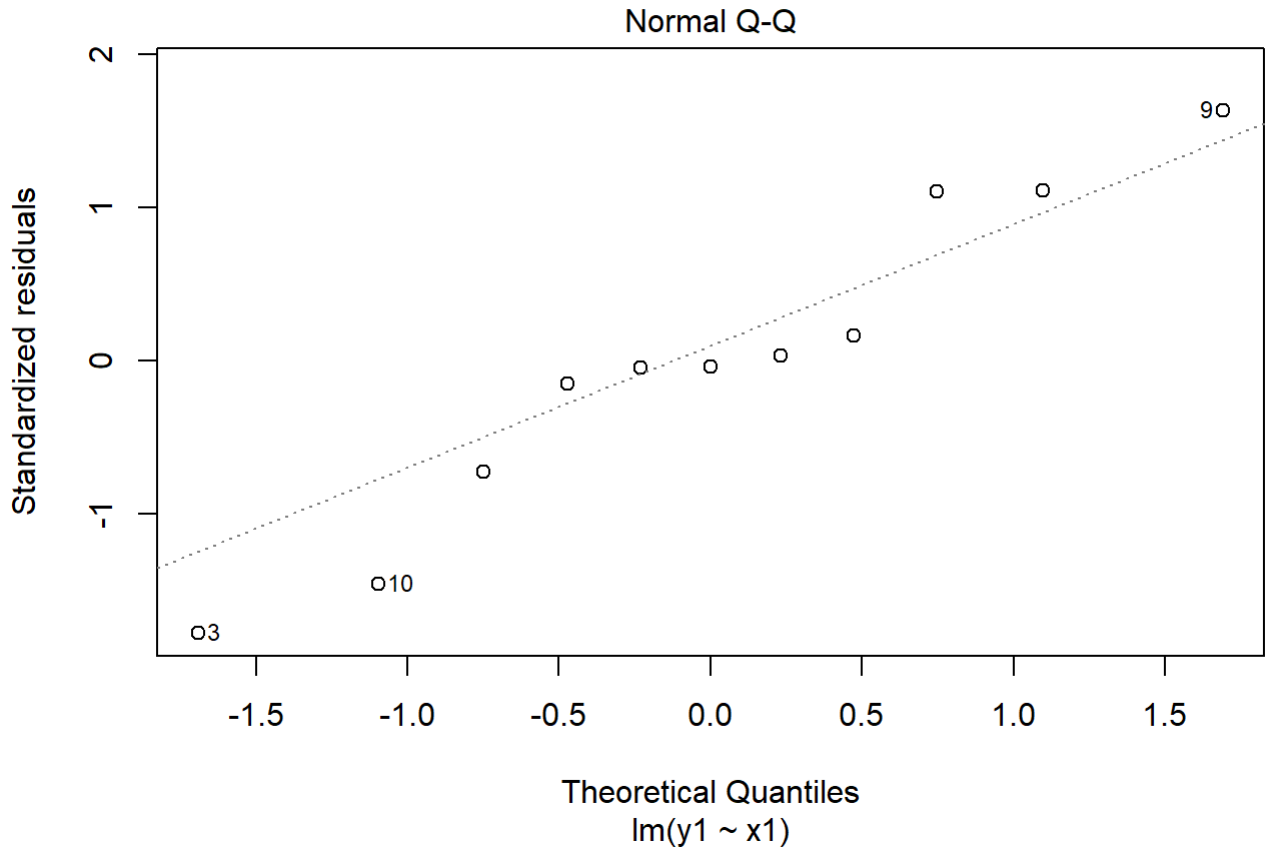
Los plots contienen:

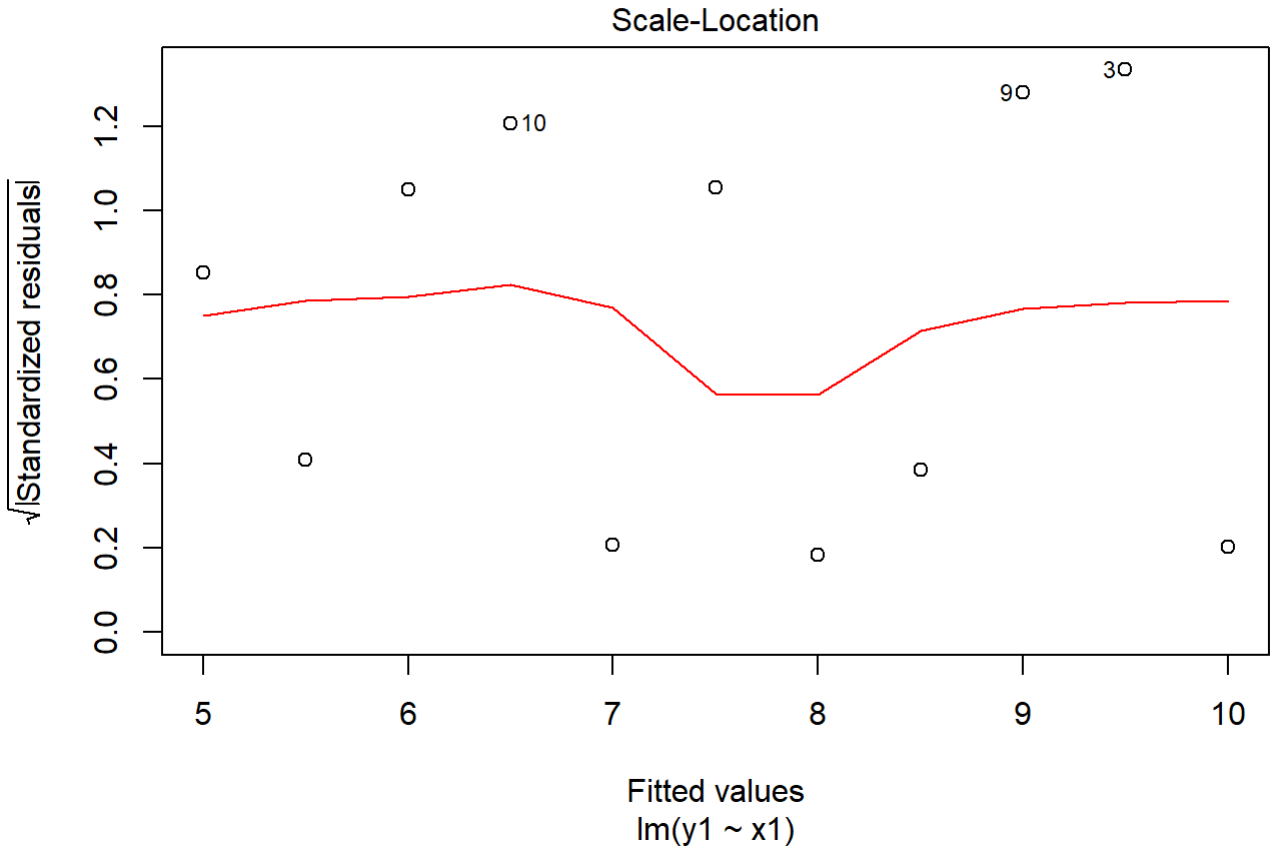
1. Residuals vs Fitted, para observar si los residuales parecen tener patrones lineales (buscamos residuos distribuidos de forma uniforme)
2. Normal Q-Q, para revisar si los residuos parecen seguir una distribución normal (buscamos una tendencia lineal)

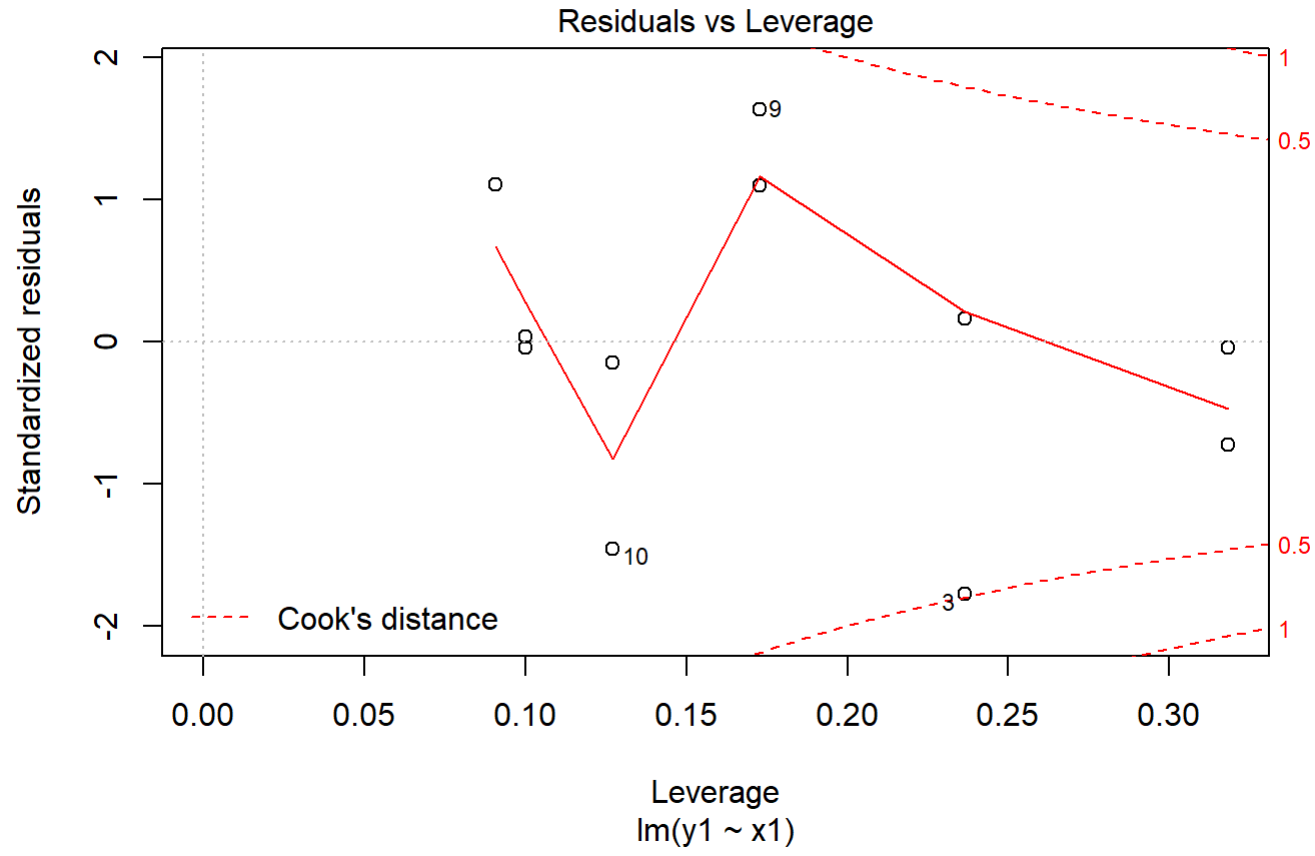
3. Scale-Location, para estudiar la homocedasticidad de las varianzas (buscamos una residuos distribuidos de forma uniforme)
4. Residuals vs Leverage, para buscar posibles observaciones influyentes

```
plot(lm1)
```

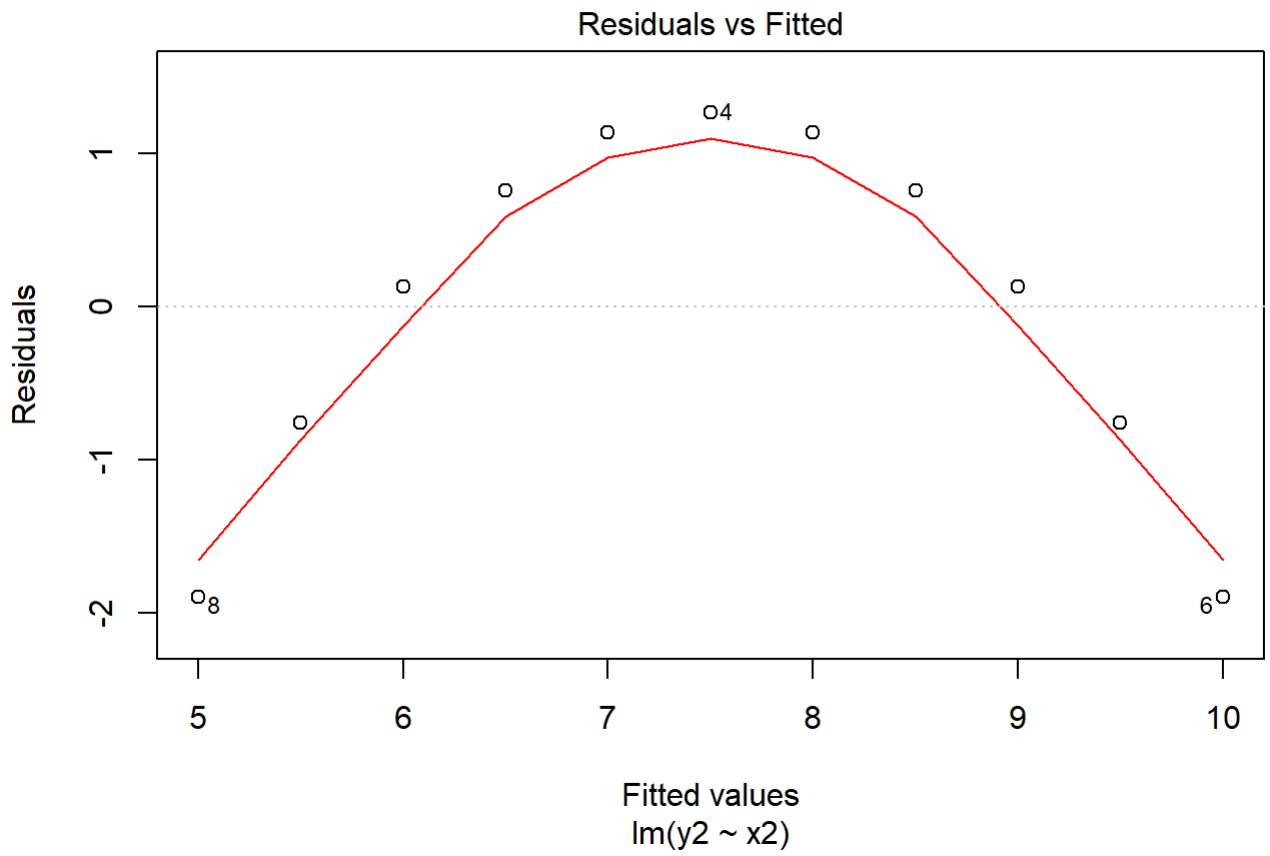

Residuals vs Fitted
Residuals
Fitted values
lm(y1 ~ x1)
9
10
3

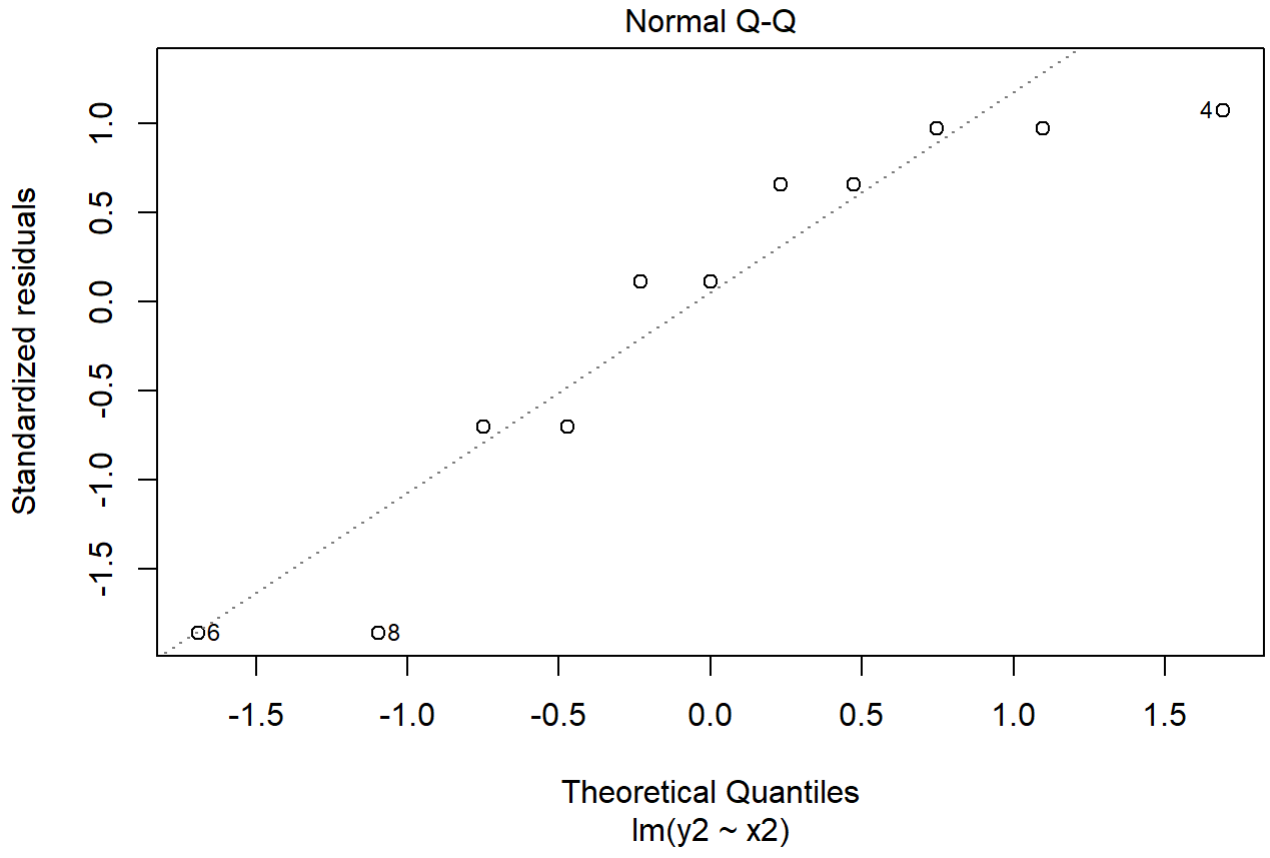

Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(y1 ~ x1)
9
10
3

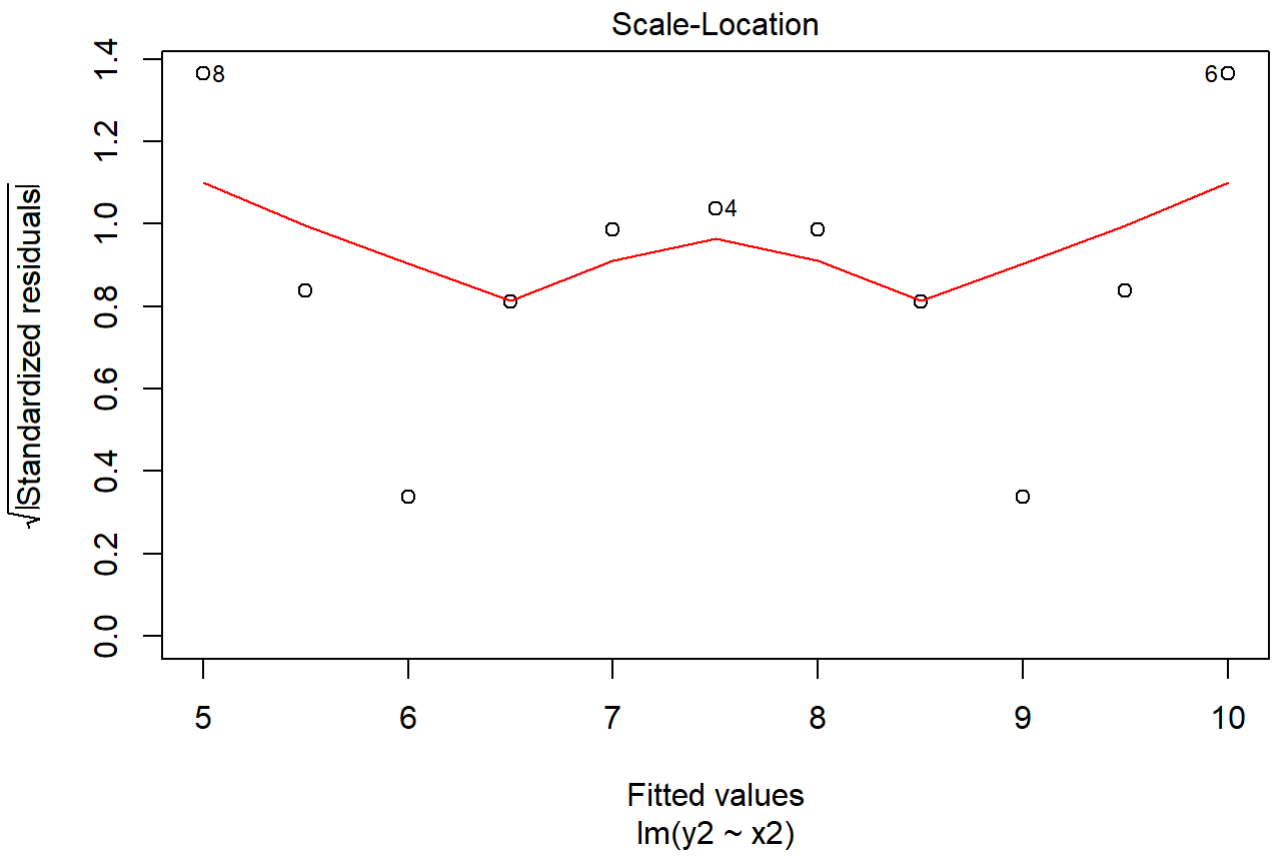

Scale-Location
√|Standardized residuals|
0.0
0.2
0.4
0.6
0.8
1.0
1.2
10
9
3
5
6
7
8
9
10
Fitted values
lm(y1 ~ x1)

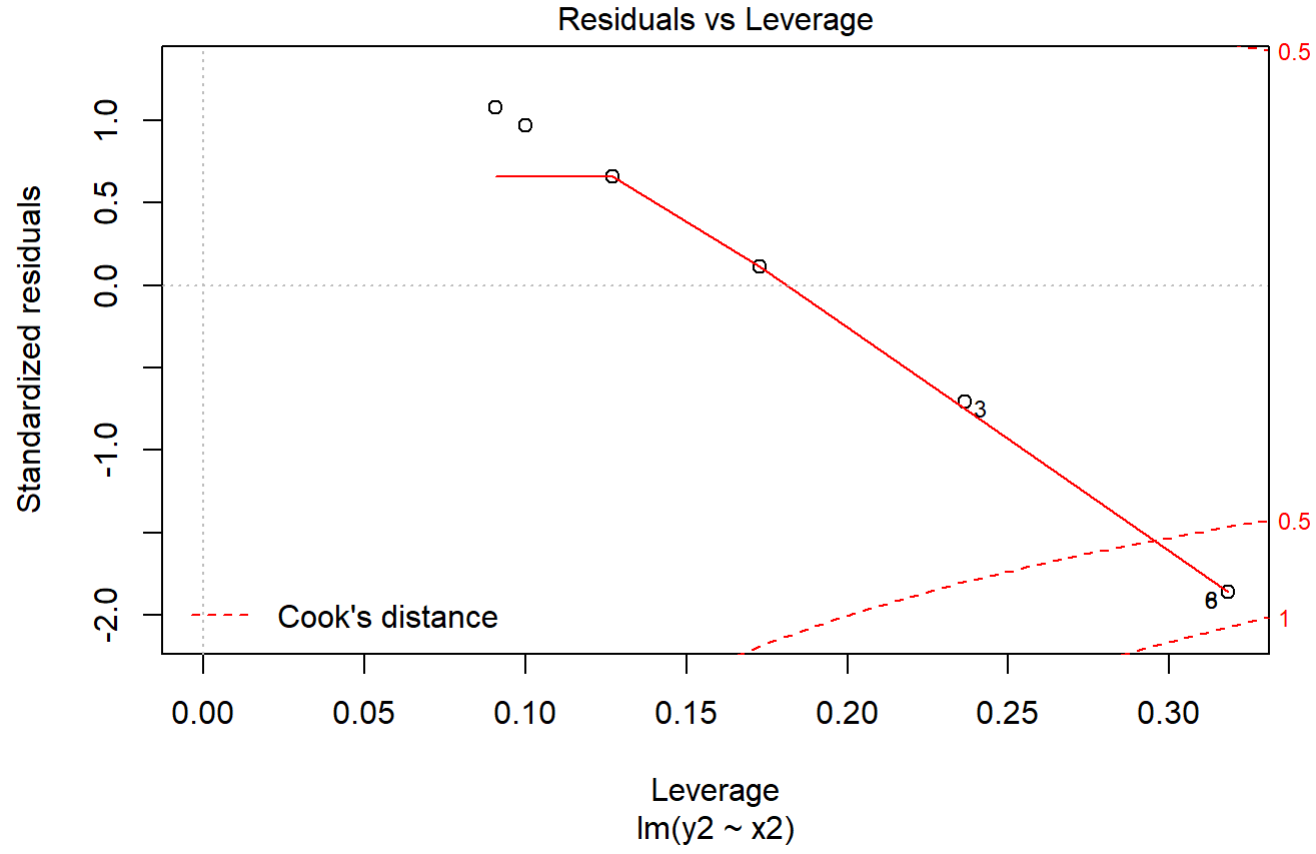

Residuals vs Leverage
Standardized residuals
2
1
0
-1
-2
9
10
3
1
0.5
0.5
1
Cook's distance
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Leverage
lm(y1 ~ x1)

```
plot(lm2)
```


Residuals vs Fitted
Residuals
1
0
-1
-2
4
8
6
5
6
7
8
9
10
Fitted values
lm(y2 ~ x2)

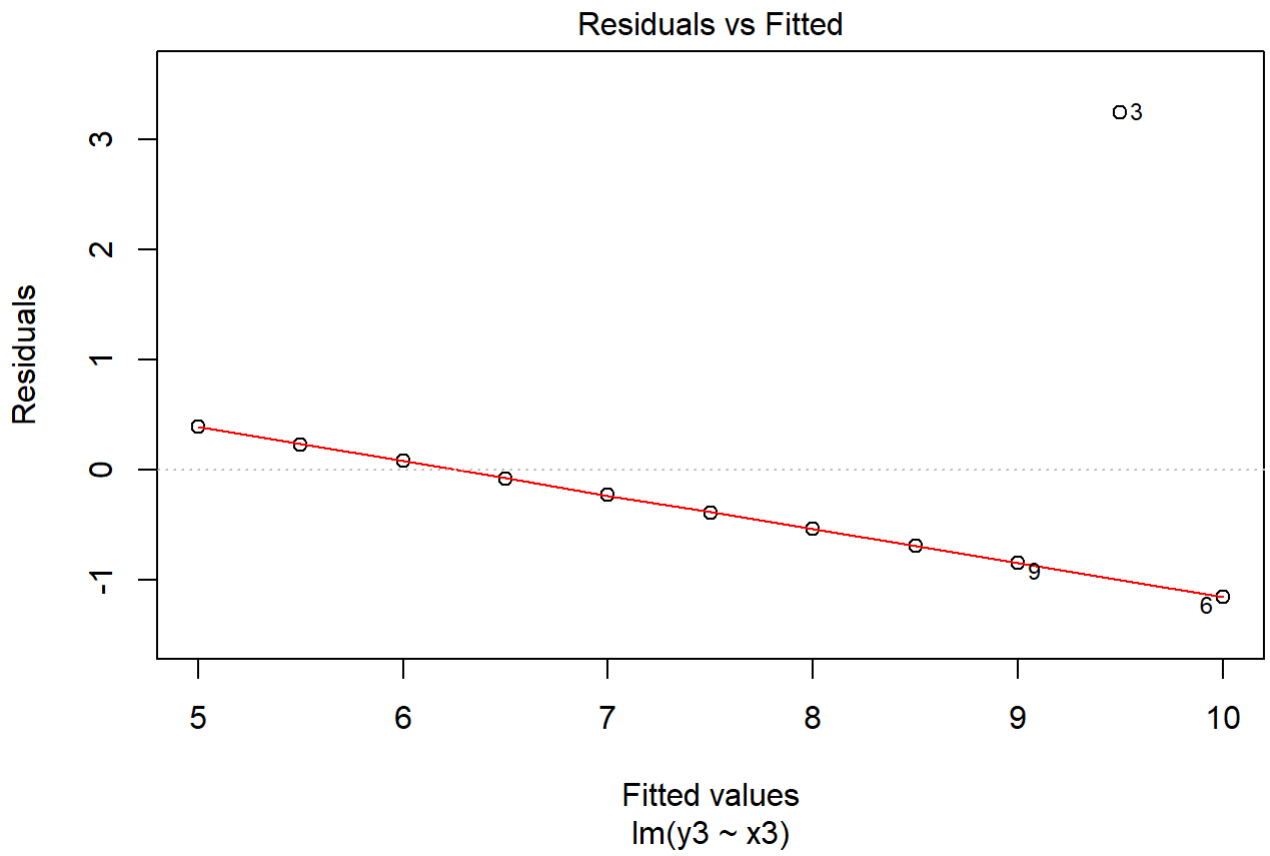

Normal Q-Q
Standardized residuals
1.0
0.5
0.0
-0.5
-1.0
-1.5
4
6
8
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
Theoretical Quantiles
lm(y2 ~ x2)

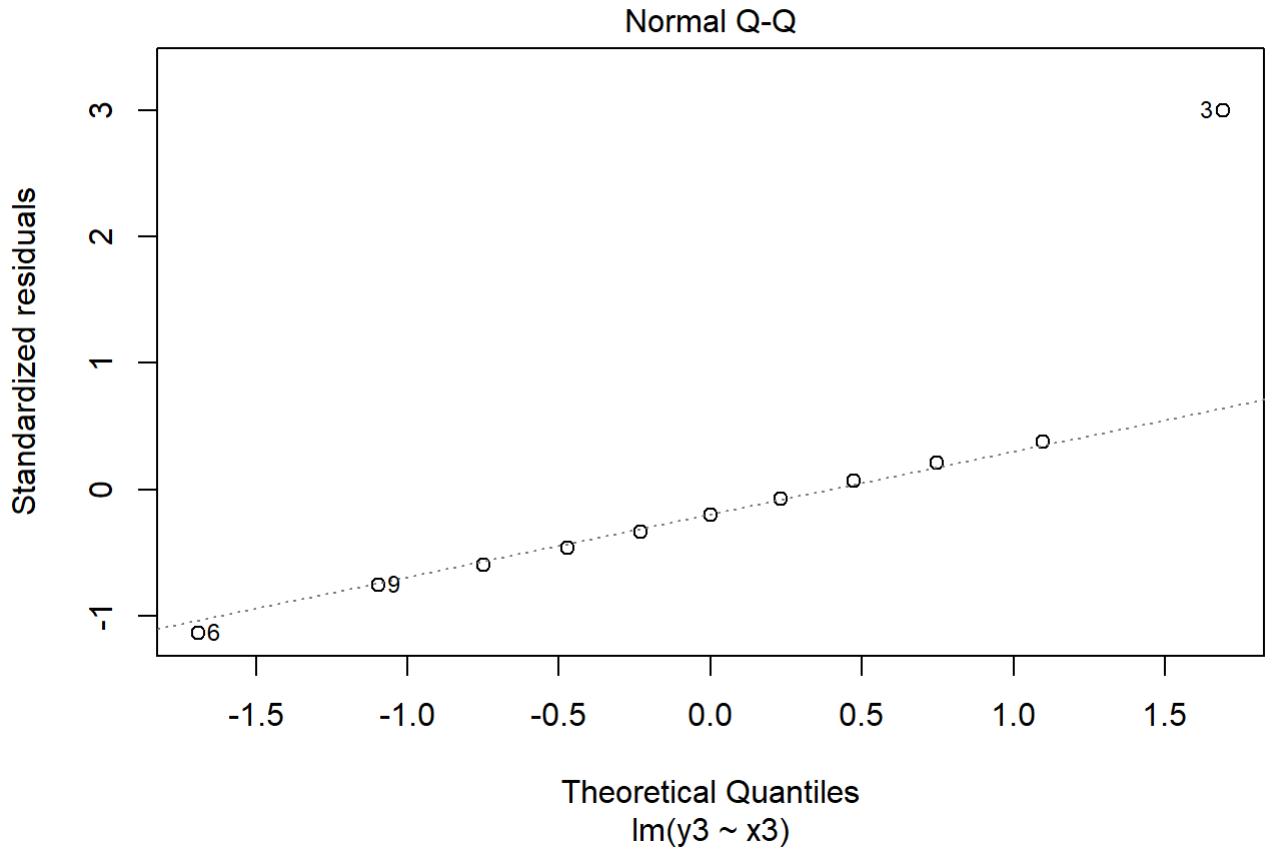

Scale-Location
√|Standardized residuals|
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
8
6
4
5
6
7
8
9
10
Fitted values
lm(y2 ~ x2)

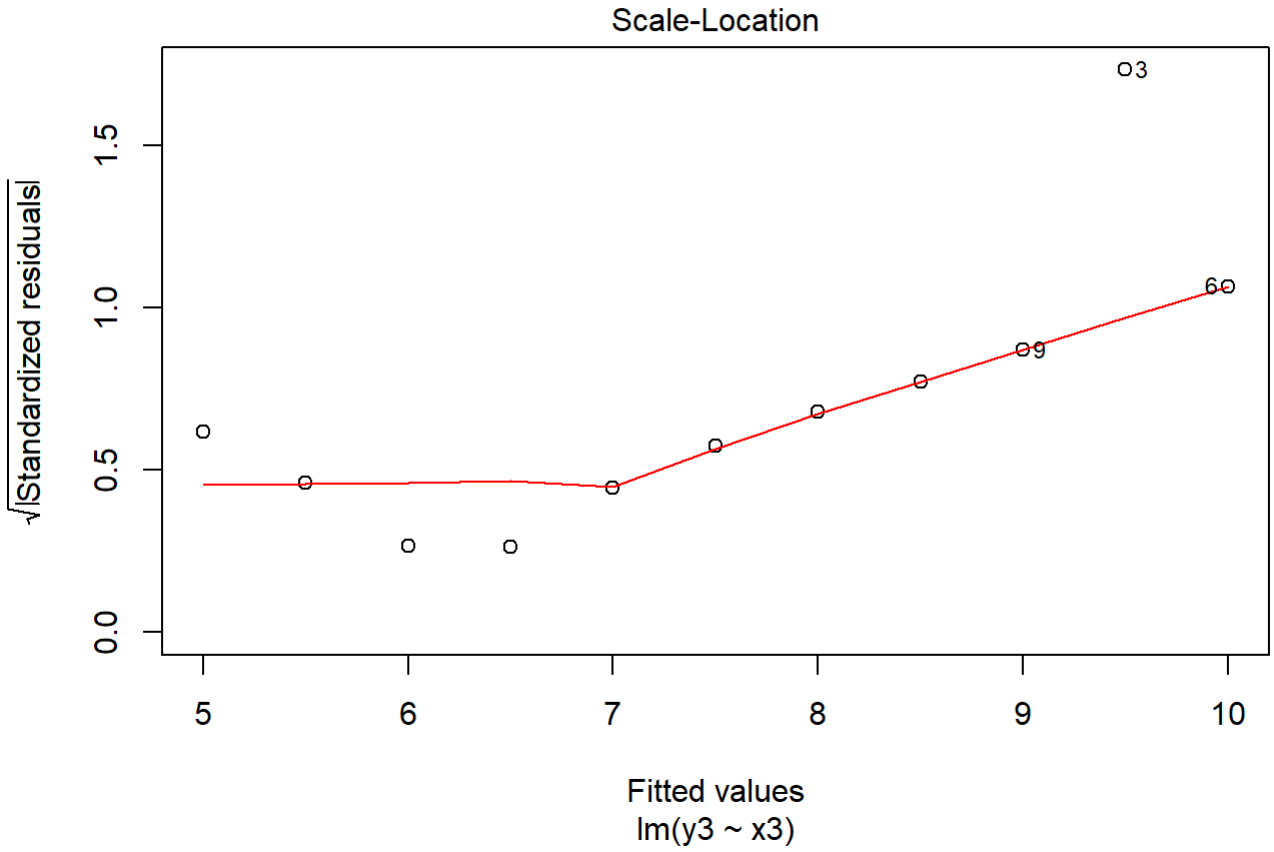

Residuals vs Leverage
Standardized residuals
-2.0
-1.0
0.0
0.5
1.0
0.5
0.5
1
Cook's distance
3
6
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Leverage
lm(y2 ~ x2)

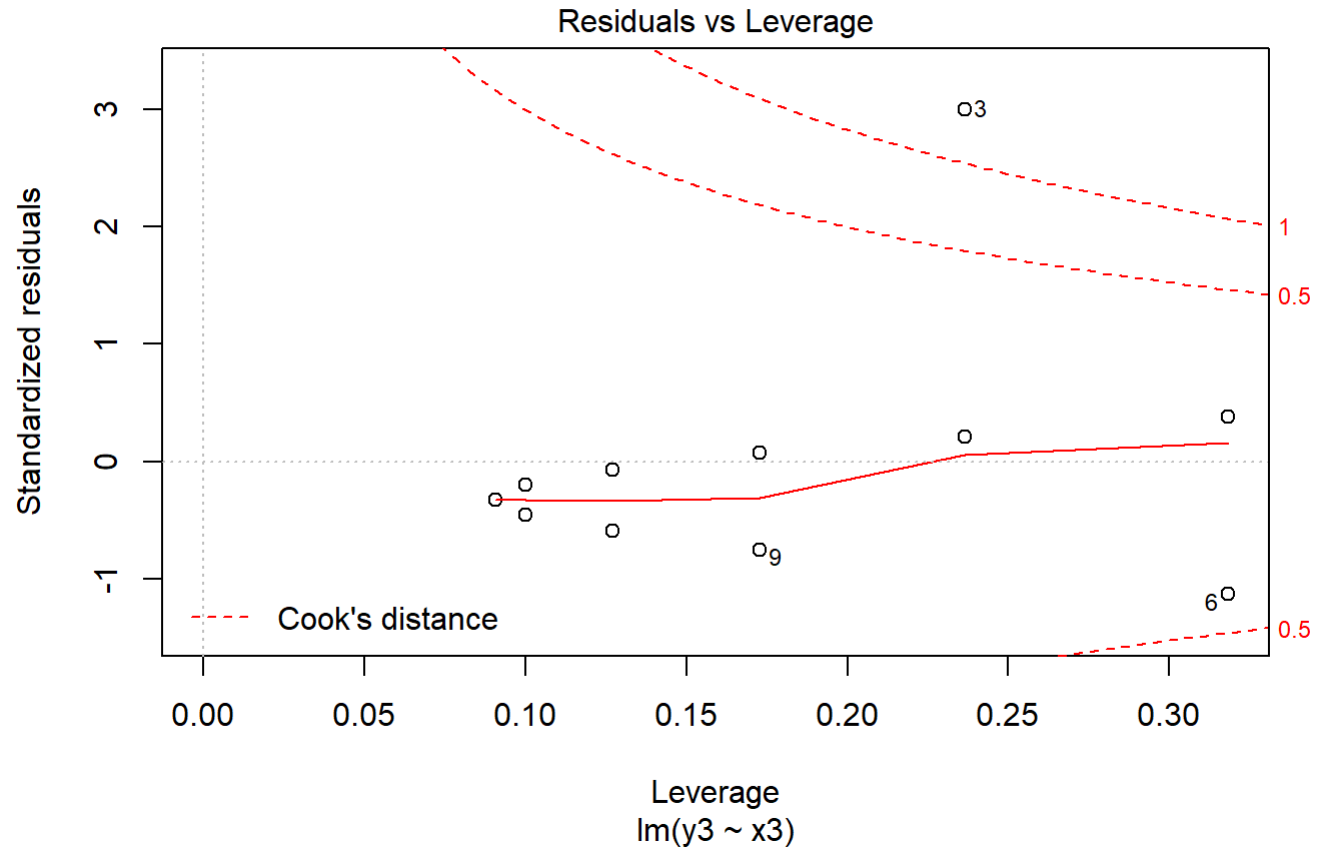
La primera gráfica muestra un comportamiento no lineal

```
plot(lm3)
```


Residuals vs Fitted
Residuals
Fitted values
lm(y3 ~ x3)
3
9
6


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(y3 ~ x3)
3
9
6


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
3
6
9
5
6
7
8
9
10
Fitted values
lm(y3 ~ x3)


Residuals vs Leverage
Standardized residuals
-1
0
1
2
3
3
1
0.5
9
6
0.5
Cook's distance
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Leverage
lm(y3 ~ x3)

La tercera observación se sale de los rangos en la última gráfica

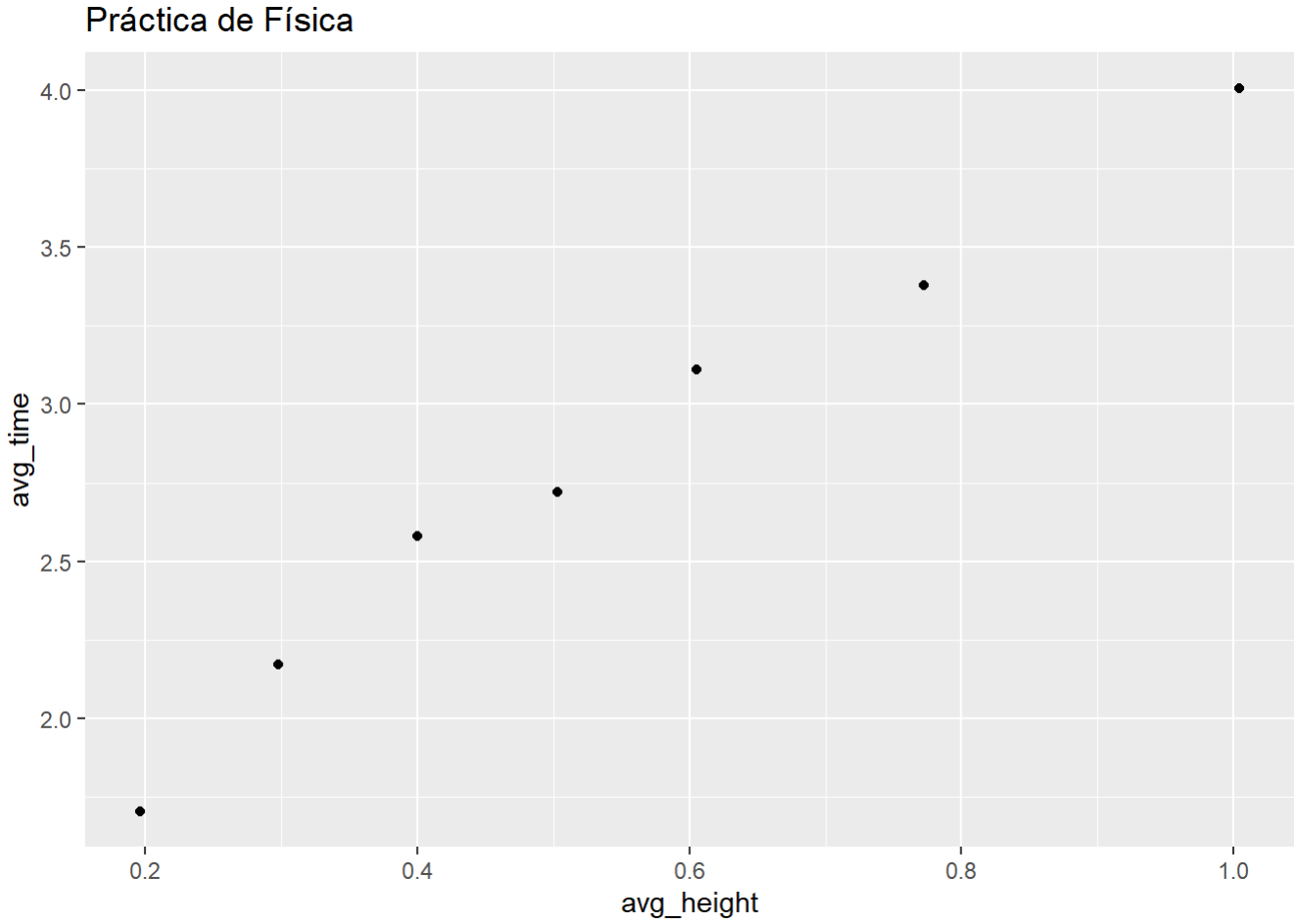
Ejemplo 2 - práctica de física

El archivo *investigation01_en.pdf* contiene un ejemplo práctica de física que sirve como referencia para profesores de preparatoria.

El propósito de esta práctica es tratar de hallar la relación entre el tiempo que necesita una pelota para rebotar seis veces dependiendo de la altura desde la que cae.

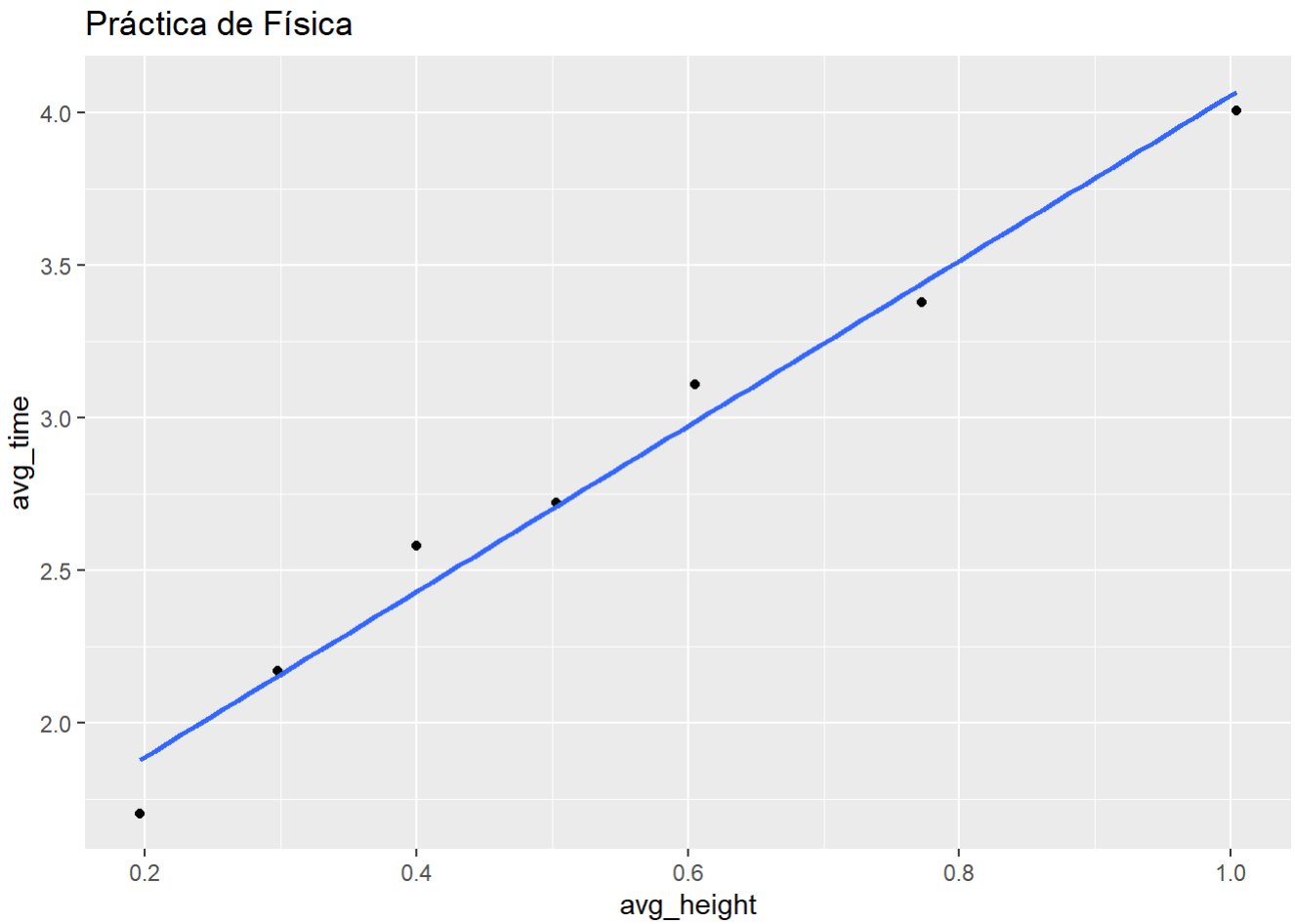
Se obtienen los siguientes resutados (ya promediados)

```
results <- data.frame( avg_height = c(0.196, 0.298, 0.400, 0.503, 0.605, 0.772, 1.004) ,
                       avg_time = c(1.705, 2.170 , 2.580 , 2.722, 3.110, 3.380 , 4.007))

fisica = ggplot(results, aes(x = avg_height , y = avg_time)) + geom_point() + ggtitle("Práctica
de Física")

fisica
```



Un modelo lineal luce bastante conveniente. De hecho, el valor de R^2 es 0.9788

```
fisica + geom_smooth(method='lm',se=F, fullrange = T)
```




```
modelo_fisica = lm(avg_time ~ avg_height, data = results)

summary(modelo_fisica)
```

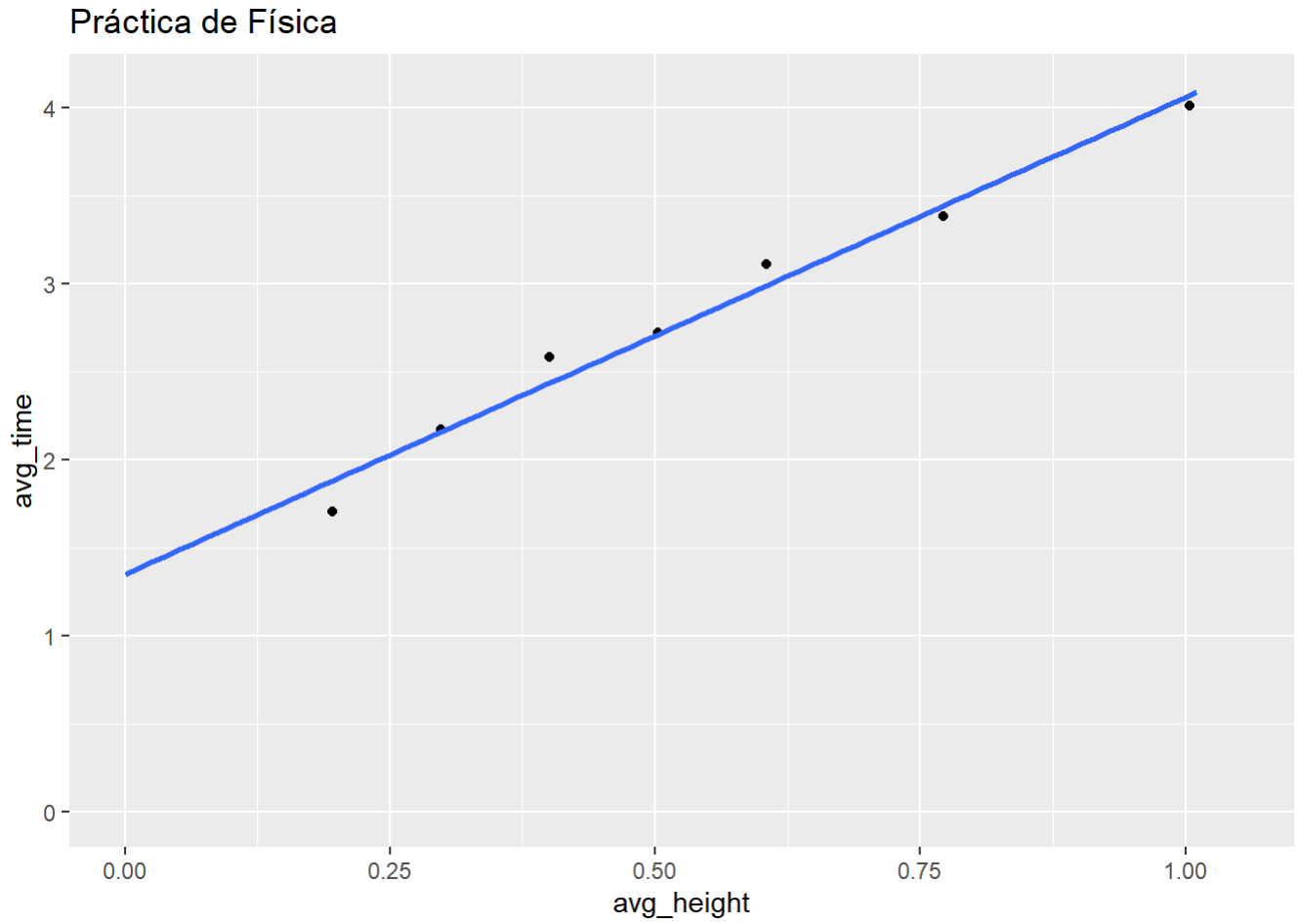
```
##
## Call:
## lm(formula = avg_time ~ avg_height, data = results)
##
## Residuals:
##        1        2        3        4        5        6        7
## -0.17366  0.01479  0.14824  0.01097  0.12242 -0.06037 -0.06239
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.3472     0.1068   12.61 5.57e-05 ***
## avg_height    2.7113     0.1784   15.20 2.24e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1225 on 5 degrees of freedom
## Multiple R-squared:  0.9788, Adjusted R-squared:  0.9746
## F-statistic:   231 on 1 and 5 DF,  p-value: 2.236e-05
```

Pero… ¿tiene sentido en términos físicos?

Si se lanza desde una altura cercana a cero, ¿cuánto tardaría en rebotar seis veces?

```
fisica + ylim(c(0,4.1)) + xlim(c(0,1.05)) + geom_smooth(method='lm',se=F, fullrange = T)
```

```
## Warning: Removed 3 rows containing missing values (geom_smooth).
```

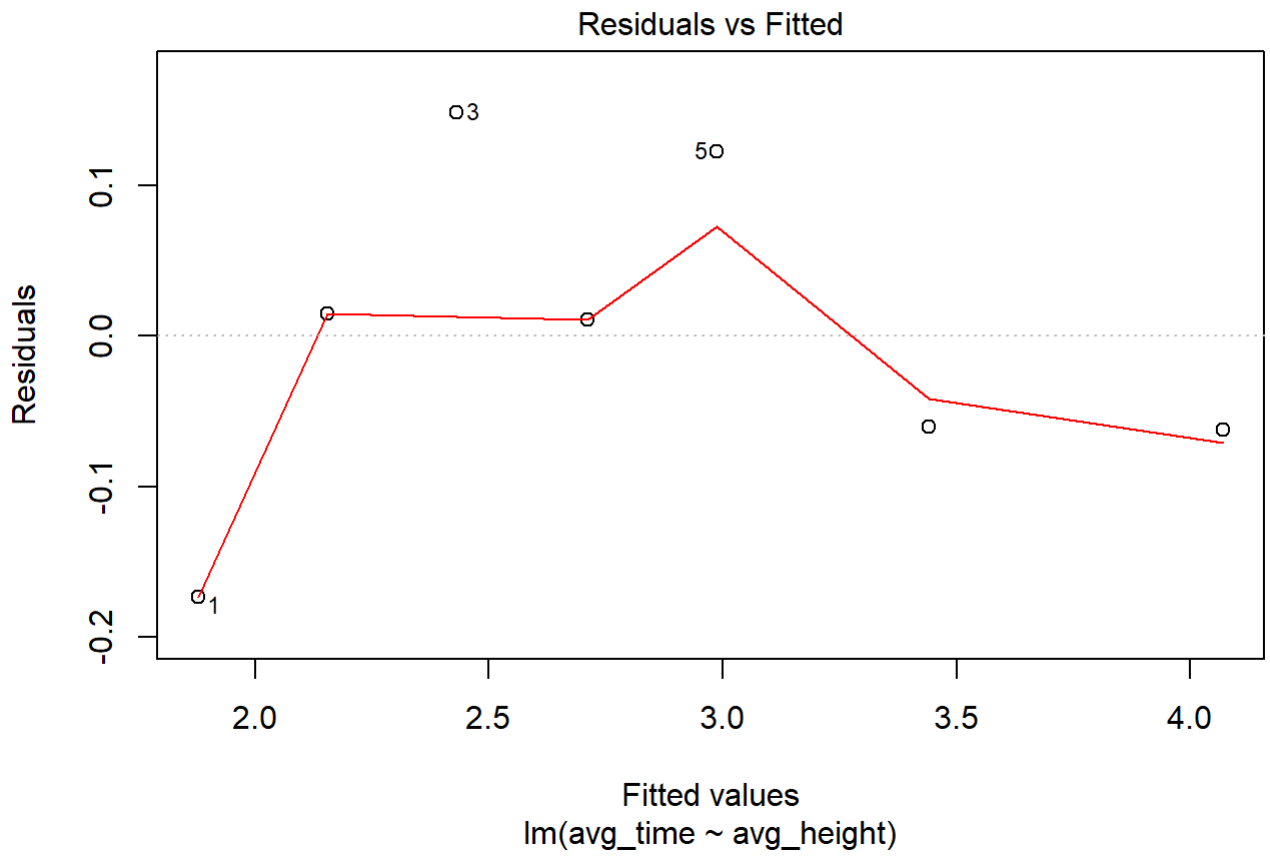


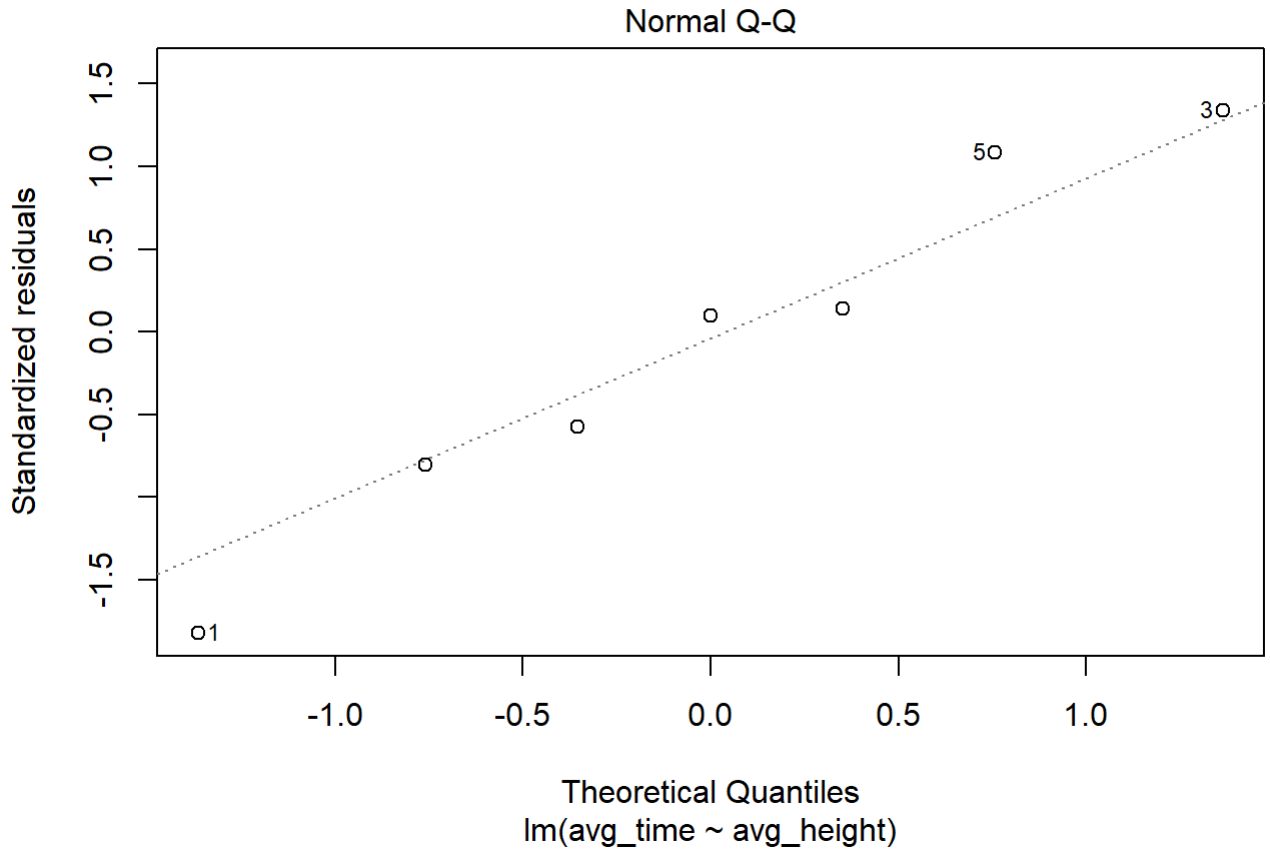
¡Poco menos de un segundo y medio!

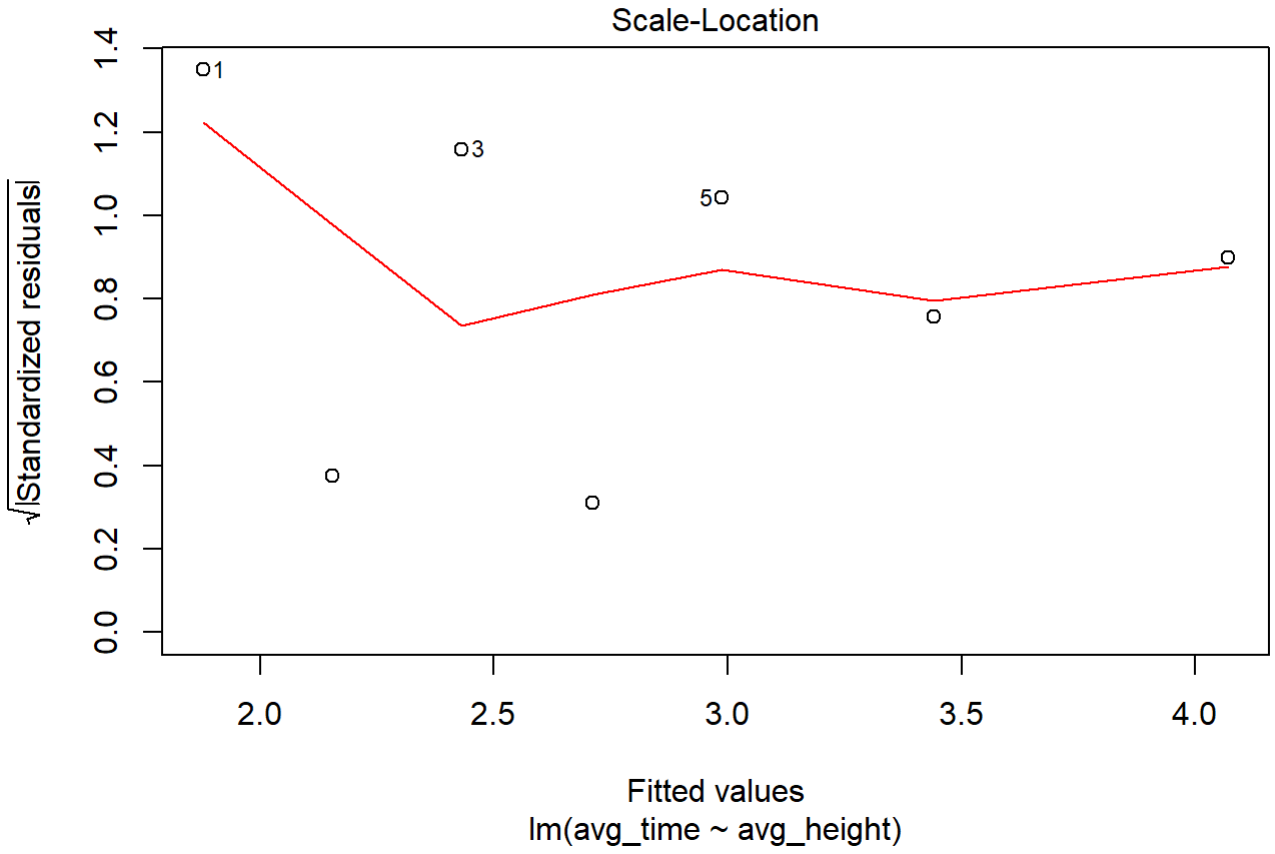
Eso… no tiene mucho sentido

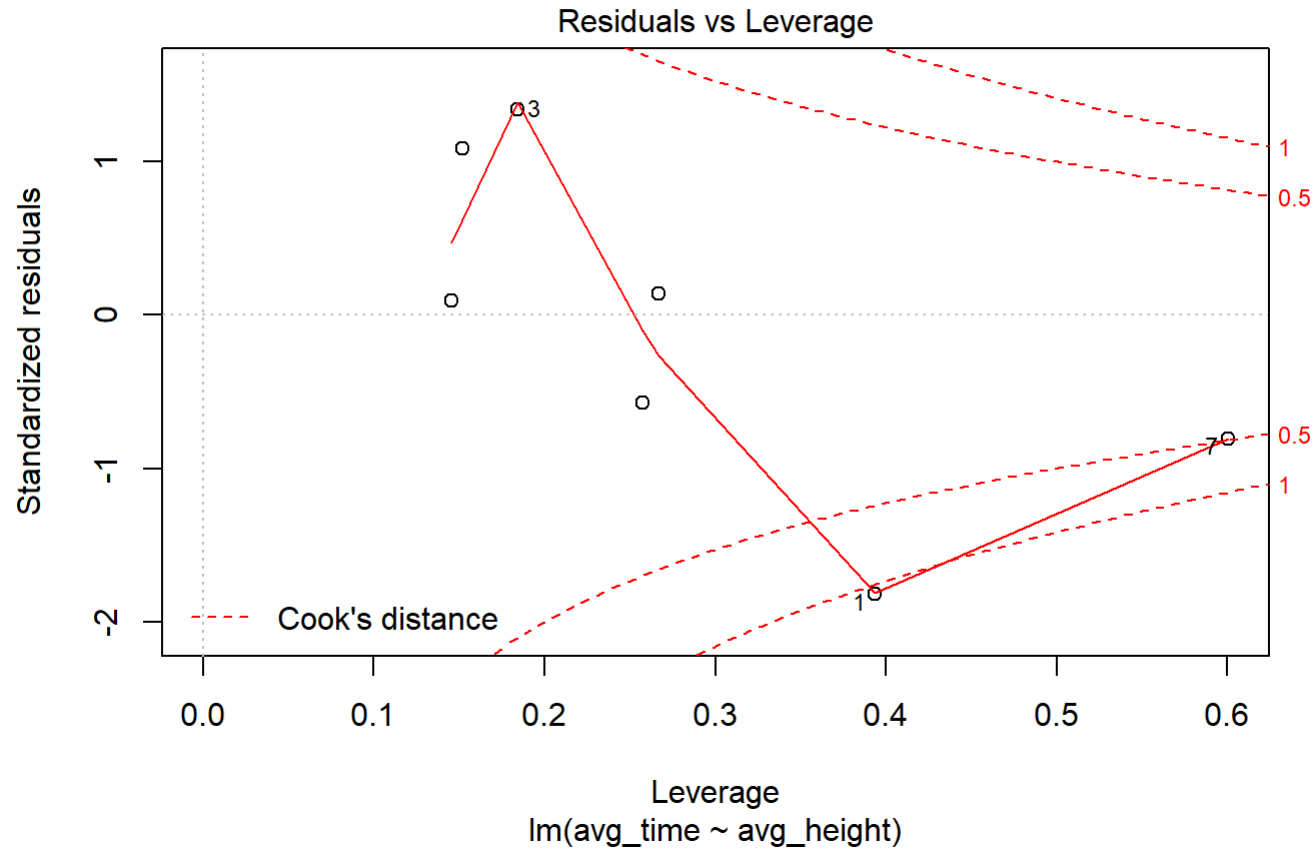
¿Qué dicen las gráficas?

```
plot(modelo_fisica)
```


Residuals vs Fitted
Residuals
0.1
0.0
-0.1
-0.2
3
5
1
2.0
2.5
3.0
3.5
4.0
Fitted values
lm(avg_time ~ avg_height)


Normal Q-Q
Standardized residuals
1.5
1.0
0.5
0.0
-0.5
-1.5
3
5
1
-1.0
-0.5
0.0
0.5
1.0
Theoretical Quantiles
lm(avg_time ~ avg_height)


Scale-Location
√|Standardized residuals|
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1
3
5
2.0
2.5
3.0
3.5
4.0
Fitted values
lm(avg_time ~ avg_height)


Residuals vs Leverage
Standardized residuals
1
0
-1
-2
3
7
1
Cook's distance
1
0.5
0.5
1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Leverage
lm(avg_time ~ avg_height)

No parecen muy convincentes…