

# LOKTA-VOLTERRA Reporte

Diego Guerrero

11/29/2021

## Proyecto final Ecuaciones Diferenciales Ordinarias: Ajustando el modelo de Lotka - Volterra a datos de una serie de tiempo con el paquete `gauseR`.

*Resumewn:* ## DESCRIPCIÓN DEL ESCENARIO ### Introducción ## Ajustando el modelo matemático a los datos ### Parte 1: Crecimiento logístico de una especie

Se utilizará el dataset **Paramecium caudatum** incluido en el paquete de `gauseR`. Se trata de una especie protista unicelular en el phylum Ciliphora, esta especie es muy abundante en ambientes marinos, salobres y agua dulce. Gause ha crecido esta especie en un monocultivo, y la información se puede encontrar en el paquete `gauseR` bajo el nombre de `gause_1934_science_f02_03`. Este archivo contiene datos de ambas especies **Paramecium caudatum** y **Paramecium aurelia** en un monocultivo y mezclados.

```
library(gauseR)
require(gauseR)
library(ggplot2)

data("gause_1934_science_f02_03")
#Exploramos nuestros datos
View(gause_1934_science_f02_03)
summary(gause_1934_science_f02_03)
```

### Visualizando los datos de monocultivo

```
##          Paper      Figure      Day      Species1
##  gause_1934_science:63  2-3:63  Min.   : 2.00  Paramecium caudatum:63
##                                     1st Qu.: 7.00
##                                     Median :12.00
##                                     Mean   :12.29
##                                     3rd Qu.:17.50
##                                     Max.   :24.00
##  Volume_Species1      Species2 Volume_Species2  Treatment
##  Min.   : 0.00  Paramecium aurelia:63  Min.   : 0.00  Mixture:23
##  1st Qu.: 0.00                                     1st Qu.: 0.00  Pa      :17
##  Median : 48.79                                     Median : 63.21  Pc      :23
##  Mean   : 75.61                                     Mean   : 77.33
##  3rd Qu.:150.33                                     3rd Qu.:137.74
##  Max.   :222.35                                     Max.   :238.68
```

```
Paramecium_caudatum <- gause_1934_science_f02_03[gause_1934_science_f02_03$Treatment == "Pc",] #Creamos
```

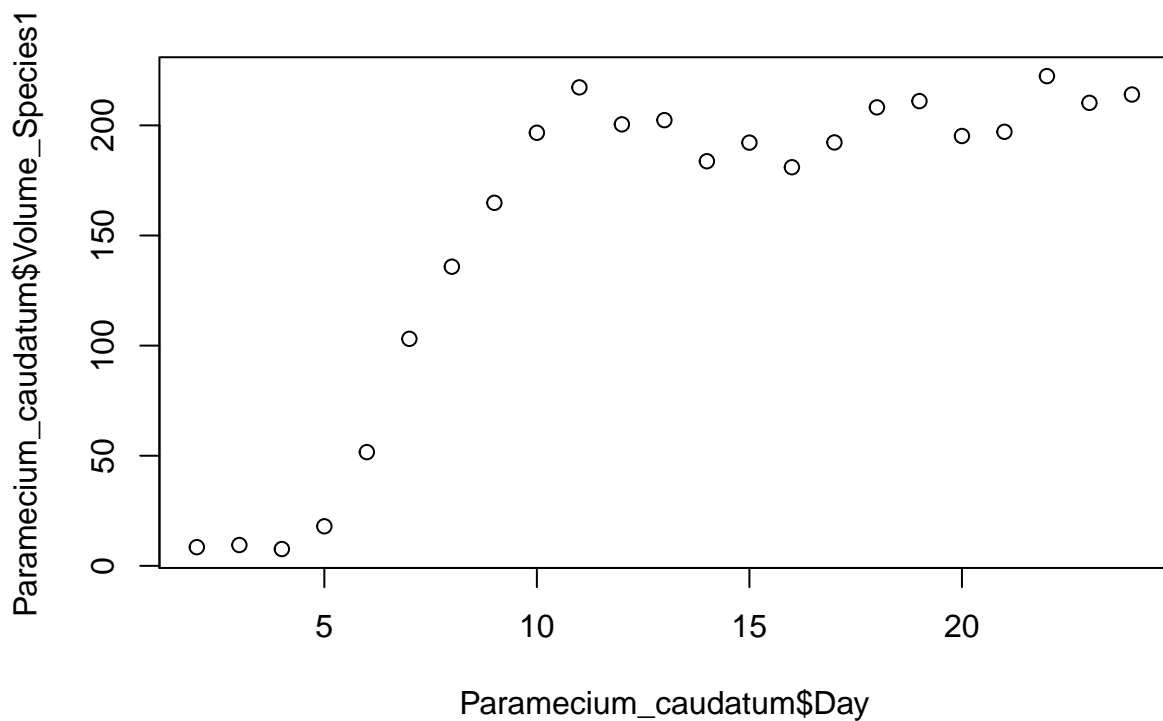
```
View(Paramecium_caudatum) #Checamos dataset, debería contener sólo datos de Paraceum caudatum.
names(Paramecium_caudatum) #Contiene columnas "Day" y "Volume_Species1"

## [1] "Paper"          "Figure"          "Day"             "Species1"
## [5] "Volume_Species1" "Species2"         "Volume_Species2" "Treatment"

#Graficamos las columnas, volumen... (y) vs Day(x)
class(Paramecium_caudatum)

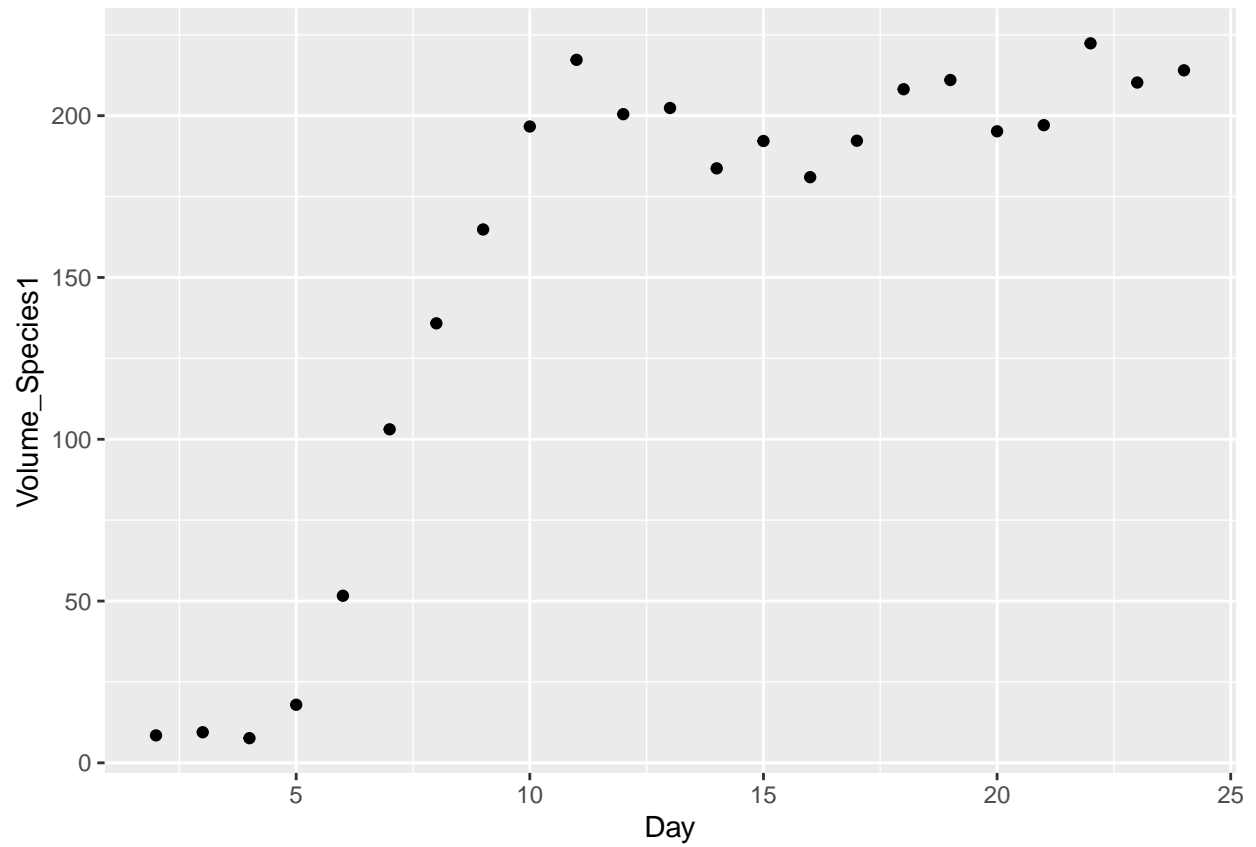
## [1] "data.frame"

plot(Paramecium_caudatum$Volume_Species1~Paramecium_caudatum$Day) #Graficamos sin ggplot2
```

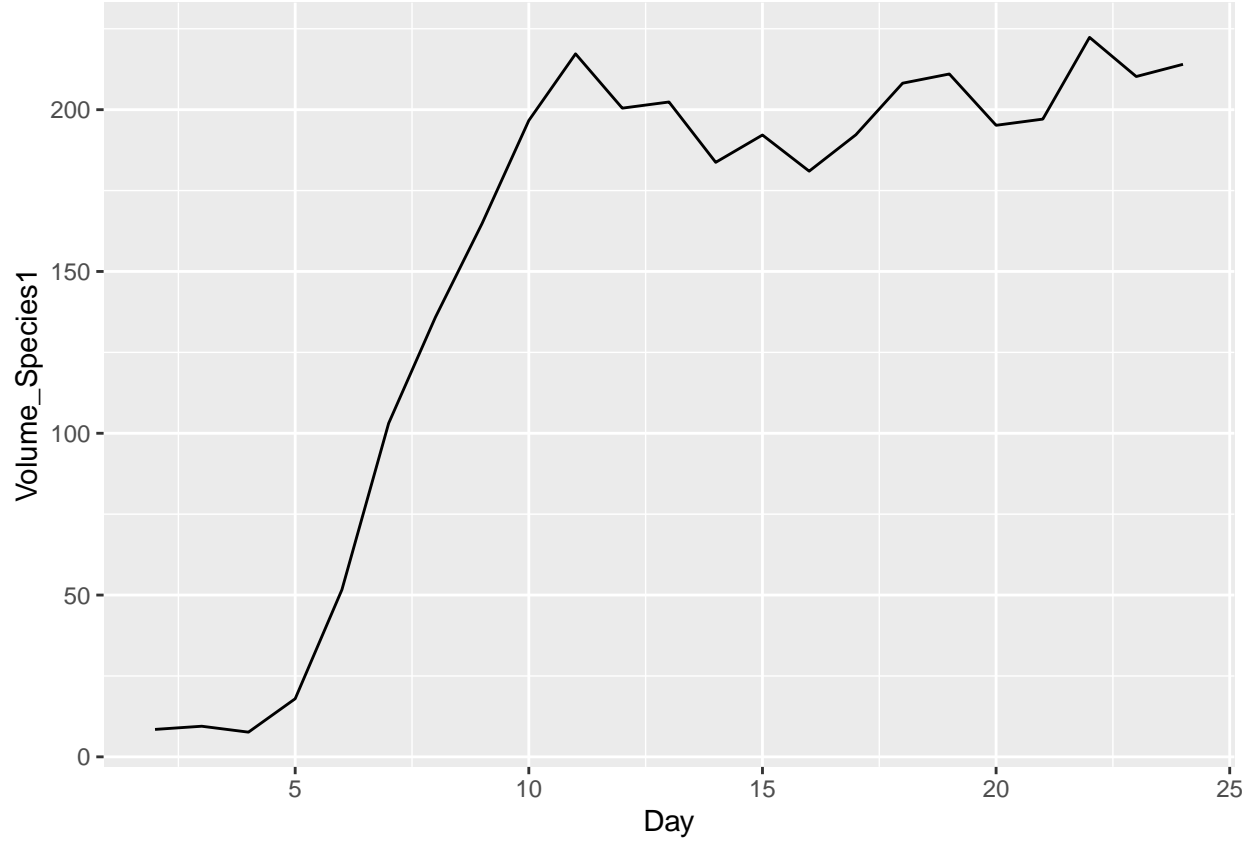


```
#Para visualizar mejor los datos se utilizará ggplot2.

ggplot(Paramecium_caudatum, aes(Day, Volume_Species1))+
  geom_point() #graficamos con puntos por medio de ggplot2
```



```
ggplot(Paramecium-caudatum, aes(Day, Volume_Species1))+  
  geom_path() #graficamos con linea para ver mejor la tendencia de los datos
```



se puede observar que los datos presentan un crecimiento logístico. Considerando la ecuación siguiente:

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right) = N(r - sN)$$

Donde  $r$  es la tasa de crecimiento intrínseca (la tasa neta en la cual los nuevos individuos son introducidos a la población cuando la población está esparcida desvanecidamente),  $s$  es la dependencia del parámetro de densidad (el cual refleja como el tamaño de la población afecta la tasa en general), y  $K = r/s$  es la capacidad de carga (máximo tamaño de población) (Lehman *et al.*, 2019).

Taza de crecimiento per-cápita:

$$\frac{dN}{Ndt} = (r - sN)$$

(4)

El lado derecho de la ecuación es una simple ecuación lineal, esto supone una ventaja cuando se analizan datos empíricos, ya que los parámetros se pueden estimar por medio de una regresión ordinaria de mínimos cuadrados de las tasas de crecimiento contra la abundancia de las especies. Por lo tanto, si calculamos la tasa de crecimiento per-cápita usando los datos, podemos encontrar una regresión lineal que se ajuste a esos datos. Para encontrar los datos sobre el crecimiento per-cápita, usaremos las ecuaciones diferenciales:

$$dN = N_i - N_{i-1}$$

y:

$$dt = t_i - t_{i-1}$$

Si ajustamos una regresión lineal a los datos, ¿Qué representará el intercepto en y? ¿Cómo encontrarás el parámetro  $s$  usando las propiedades de la línea? ¿Qué representará el intercepto en x?

*Todos los cálculos se pueden realizar en R*

**Cálculando la tasa de crecimiento per-cápita (dN/dt)** Se usaron las diferencias retardadas y se graficaron las contra la abundancia de N.

*¿Qué son las diferencias retardadas?*

*#Calculando abundancia retardada en tiempo usando el comando get\_lag de gauseR*

?get\_lag

```
lagged_data <- get_lag(x = Paramecium_caudatum$Volume_Species1, time = Paramecium_caudatum$Day)
```

*# Visualizamos los datos*

```
summary(lagged_data)
```

```
##           x           laggedx           xmid           dt
## Min.      : 7.633   Min.      : 7.633   Min.      : 8.548   Min.      :1
## 1st Qu.:119.452   1st Qu.:111.266   1st Qu.:127.172   1st Qu.:1
## Median :192.254   Median :192.213   Median :190.496   Median :1
## Mean    :153.168   Mean    :150.402   Mean    :155.073   Mean    :1
## 3rd Qu.:205.275   3rd Qu.:201.896   3rd Qu.:205.989   3rd Qu.:1
## Max.    :222.350   Max.    :222.350   Max.    :216.294   Max.    :1
##          NA's      :1          NA's      :1          NA's      :1
##           time        laggedtime        treatment
## Min.      : 3.00   Min.      : 2.00   Min.      :1
## 1st Qu.: 8.25   1st Qu.: 7.25   1st Qu.:1
## Median :13.50   Median :12.50   Median :1
## Mean    :13.50   Mean    :12.50   Mean    :1
## 3rd Qu.:18.75   3rd Qu.:17.75   3rd Qu.:1
## Max.    :24.00   Max.    :23.00   Max.    :1
##          NA's      :1          NA's      :1
```

```
lagged_data
```

```
##           x           laggedx           xmid dt time laggedtime treatment
## 1      8.486438              NA              NA NA    NA              NA          1
## 2      9.463023      8.486438      8.974730  1    3          2          1
## 3      7.633013      9.463023      8.548018  1    4          3          1
## 4     17.956706      7.633013     12.794859  1    5          4          1
## 5     51.637301     17.956706     34.797004  1    6          5          1
## 6    103.078996     51.637301     77.358149  1    7          6          1
## 7    135.825026    103.078996    119.452011  1    8          7          1
## 8    164.837139    135.825026    150.331083  1    9          8          1
## 9    196.652950    164.837139    180.745045  1   10          9          1
## 10   217.252521    196.652950    206.952736  1   11         10          1
## 11   200.466558    217.252521    208.859540  1   12         11          1
## 12   202.371914    200.466558    201.419236  1   13         12          1
## 13   183.721166    202.371914    193.046540  1   14         13          1
## 14   192.171380    183.721166    187.946273  1   15         14          1
## 15   180.994262    192.171380    186.582821  1   16         15          1
## 16   192.253970    180.994262    186.624116  1   17         16          1
## 17   208.177814    192.253970    200.215892  1   18         17          1
## 18   211.022082    208.177814    209.599948  1   19         18          1
```

```
## 19 195.172134 211.022082 203.097108 1 20 19 1
## 20 197.078938 195.172134 196.125536 1 21 20 1
## 21 222.349890 197.078938 209.714414 1 22 21 1
## 22 210.238206 222.349890 216.294048 1 23 22 1
## 23 214.017040 210.238206 212.127623 1 24 23 1

# Calculamos el crecimiento per cápita (dNNdt) y lo guardamos en una columna en el archivo de los datos.

Paramecium_caudatum$dNNdt <- percap_growth(x = lagged_data$x, laggedx = lagged_data$laggedx, dt = lagged_data$dt)

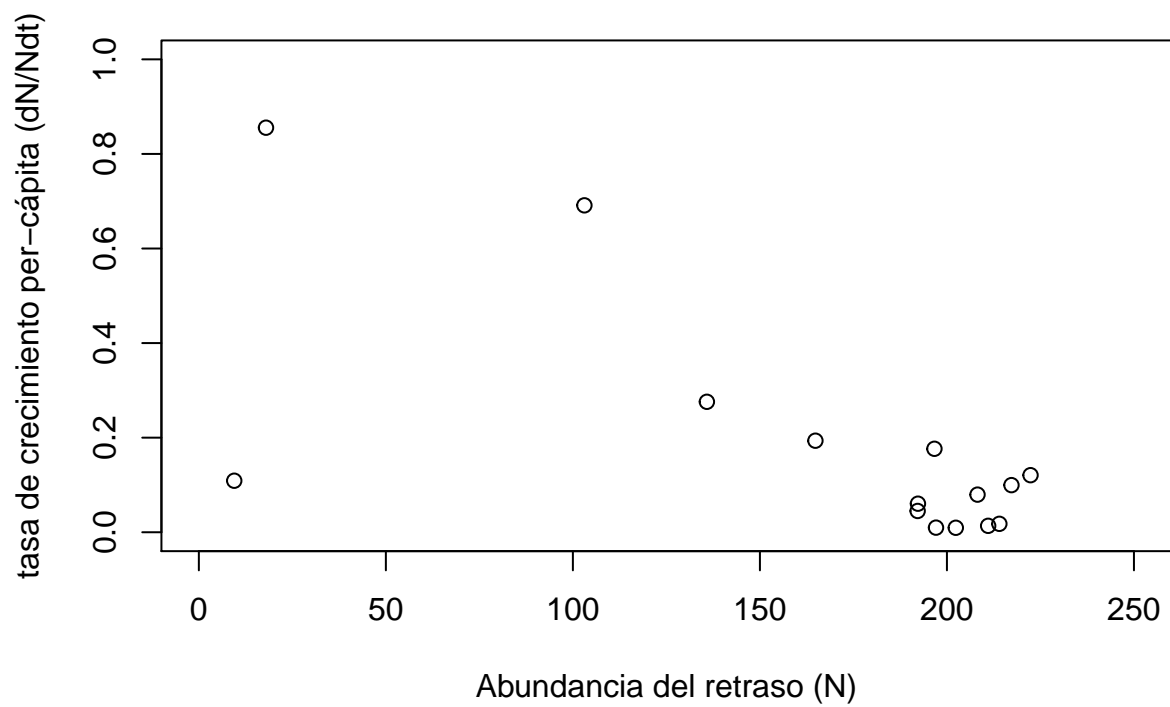
# Visualizamos

head(Paramecium_caudatum) # Va está la nueva columna dNNdt

##          Paper Figure Day          Species1 Volume_Species1
## 1 gauge_1934_science  2-3    2 Paramecium caudatum      8.486438
## 2 gauge_1934_science  2-3    3 Paramecium caudatum      9.463023
## 3 gauge_1934_science  2-3    4 Paramecium caudatum      7.633013
## 4 gauge_1934_science  2-3    5 Paramecium caudatum     17.956706
## 5 gauge_1934_science  2-3    6 Paramecium caudatum     51.637301
## 6 gauge_1934_science  2-3    7 Paramecium caudatum    103.078996
##          Species2 Volume_Species2 Treatment      dNNdt
## 1 Paramecium aurelia          0      Pc      NA
## 2 Paramecium aurelia          0      Pc  0.1089225
## 3 Paramecium aurelia          0      Pc -0.2149093
## 4 Paramecium aurelia          0      Pc  0.8554810
## 5 Paramecium aurelia          0      Pc  1.0562807
## 6 Paramecium aurelia          0      Pc  0.6912513

# Graficamos la relación

plot(Paramecium_caudatum$dNNdt ~ Paramecium_caudatum$Volume_Species1, xlab = "Abundancia del retraso (N")
```

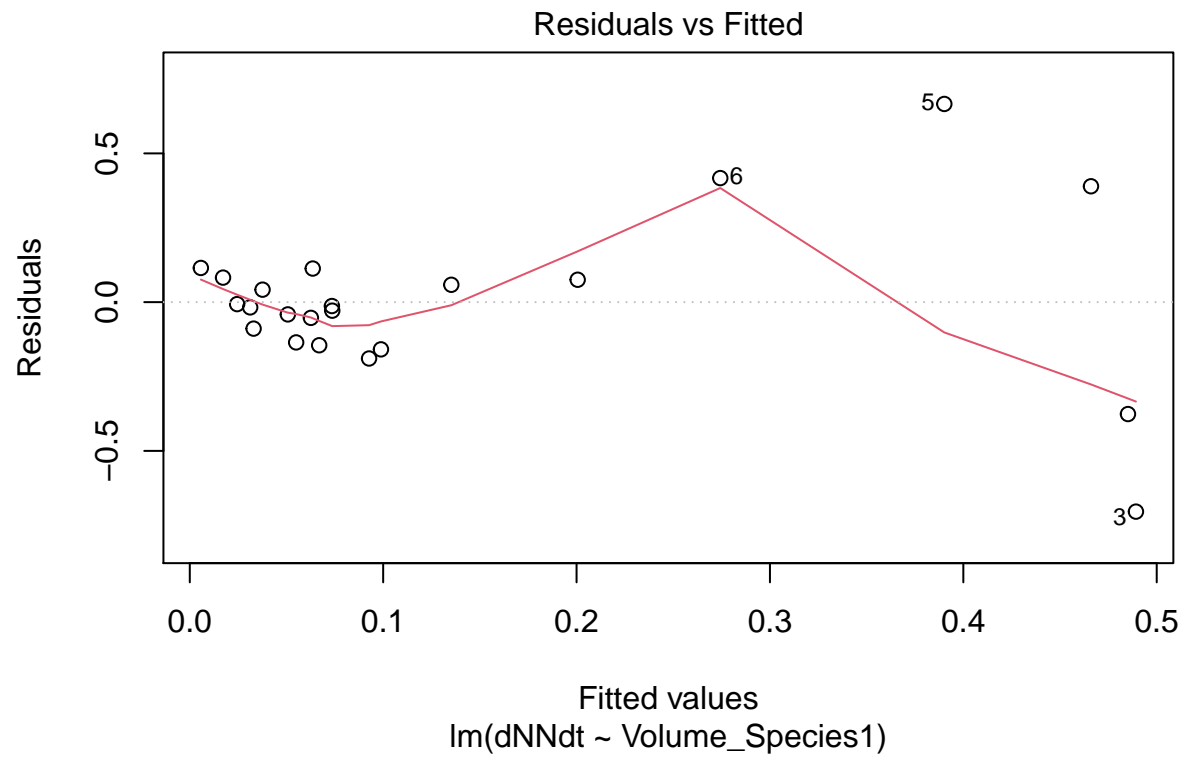


*#Encontramos una regresión lineal para los datos*

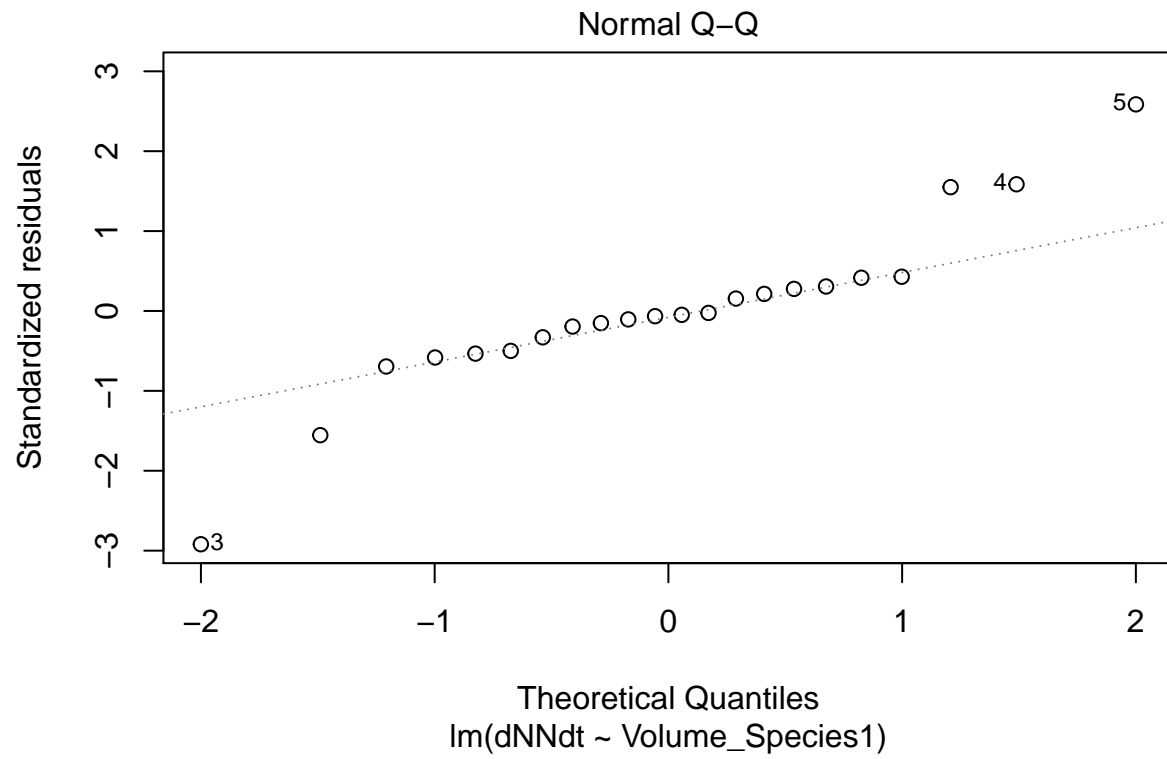
```
modelo_PC = lm(dNNdt ~ Volume_Species1, data = Paramecium-caudatum)
```

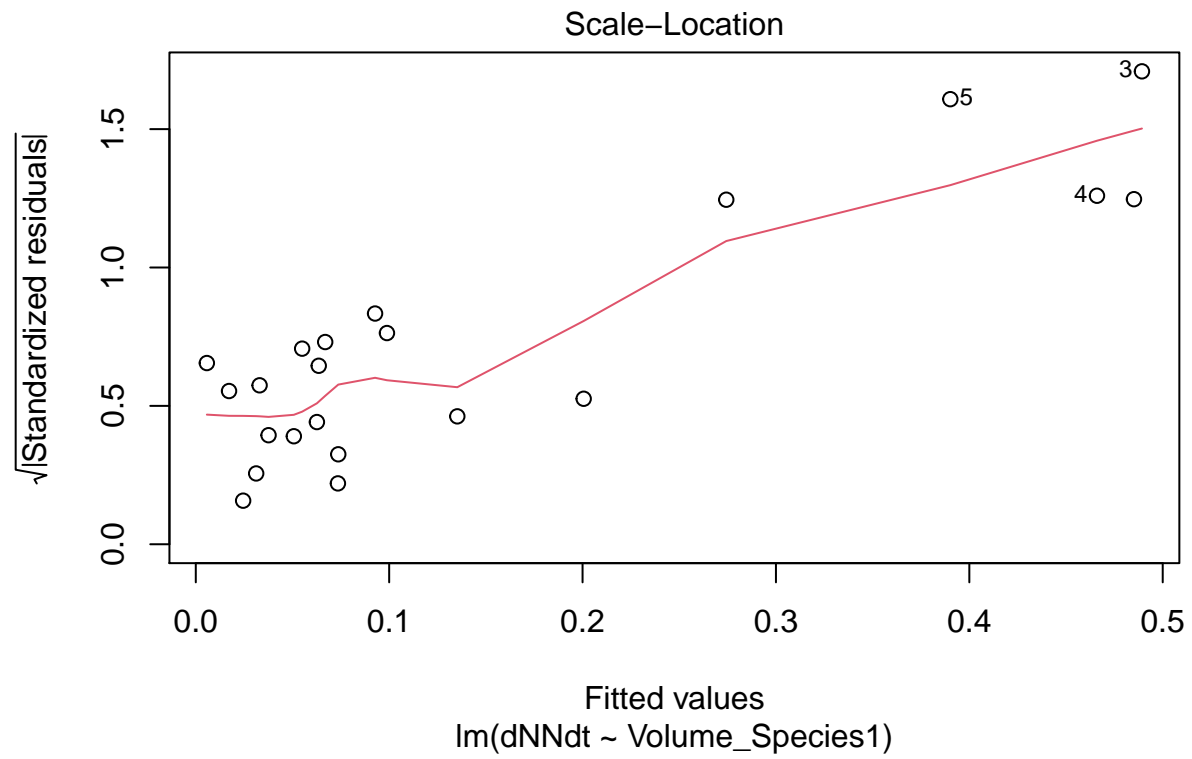
*#Graficamos el modelo como una línea roja*

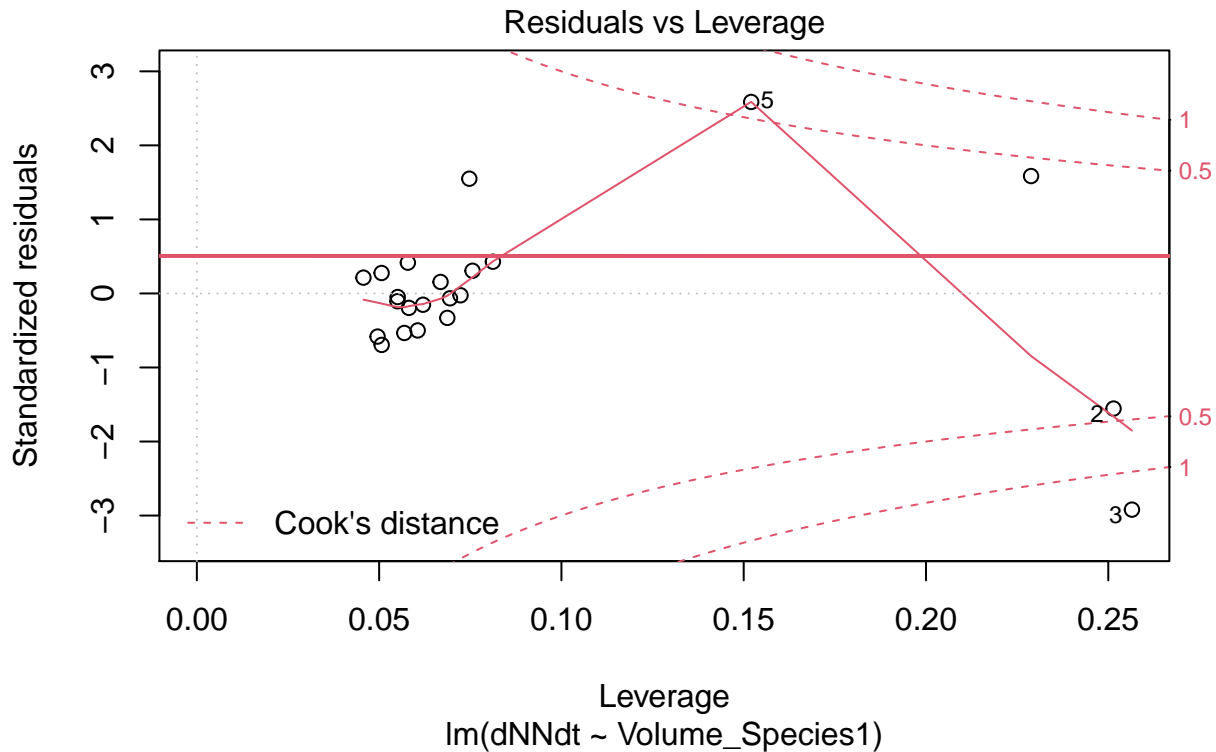
```
plot(modelo_PC) + abline(modelo_PC, lwd = 2, col = 2)
```











```
## integer(0)
```

```
parametros_rsn <- coef(modelo_PC)
```

**Extrayendo los parámetros  $r$  y  $s$  del modelo de la regresión lineal :** Ahora utilizaremos los parámetros extraídos empezando con los valores del ajuste no lineal de mínimos cuadrados (*nls*). Esta parte es generalmente no requerida ya que se pueden utilizar los parámetros obtenidos para graficar la función de crecimiento logístico, sin embargo, no contamos con valores para el volumen inicial. Para solucionar este problema, se podrían utilizar diferentes suposiciones para el valor del volumen inicial y elegir la que mejor se adapte al modelo. No usaremos ese método, optimizaremos el ajuste del modelo para obtener los valores del volumen inicial.

Los valores del volumen inicial no son requeridos para obtener una respuesta. Sin embargo se necesita ya que el comando *nls* utiliza el método de gradiente descendiente. Éste método, al ser un algoritmo de minimización, puede encontrar los mínimos locales pero no los mínimos globales deseados.

Identificamos la capacidad de carga con el comando `summary`. *¿Qué podemos decir acerca del comportamiento de las especies de largo plazo?*

```
summary(Paramecium_caudatum)
```

```
##          Paper    Figure    Day          Species1
##  gause_1934_science:23  2-3:23  Min.   : 2.0  Paramecium caudatum:23
##                               1st Qu.: 7.5
##                               Median :13.0
##                               Mean   :13.0
##                               3rd Qu.:18.5
```

```
##                               Max.      :24.0
##
## Volume_Species1              Species2 Volume_Species2 Treatment
## Min.      : 7.633   Paramecium aurelia:23   Min.      :0      Mixture: 0
## 1st Qu.:119.452              1st Qu.:0      Pa      : 0
## Median :192.254              Median :0      Pc      :23
## Mean      :153.168              Mean      :0
## 3rd Qu.:205.275              3rd Qu.:0
## Max.      :222.350              Max.      :0
##
##      dNNdt
## Min.      : -0.21491
## 1st Qu.: -0.03964
## Median : 0.05266
## Mean      : 0.14671
## 3rd Qu.: 0.16252
## Max.      : 1.05628
## NA's      :1
# Transformamos los parámetros a una escala logística
par_logist <- c(r = unname(parametros_rsn["(Intercept)"]),
               K=unname(-parametros_rsn["(Intercept)"]/parametros_rsn["Volume_Species1"]))
```

Ajustando con el comando nls:

```
modelo_nls <- nls(Volume_Species1 ~ get_logistic(time = Day, NO, r, K),
                  data = Paramecium_caudatum,
                  start = c(NO = unname(Paramecium_caudatum$Volume_Species1[which.min(Paramecium_caudatum$Volume_Species1)]),
                             r = 0.1, K = 200))

summary(modelo_nls)
```

```
##
## Formula: Volume_Species1 ~ get_logistic(time = Day, NO, r, K)
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## NO    0.2182      0.1820   1.199   0.244
## r     0.9593      0.1169   8.209 7.81e-08 ***
## K    202.3046      3.0331  66.698 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.17 on 20 degrees of freedom
##
## Number of iterations to convergence: 11
## Achieved convergence tolerance: 1.391e-06
```

### Graficando resultados

```
#Graficamos los resultados

plot(Volume_Species1 ~ Day, data = Paramecium_caudatum, type = "b", ylab = "P.caudatum Volume")

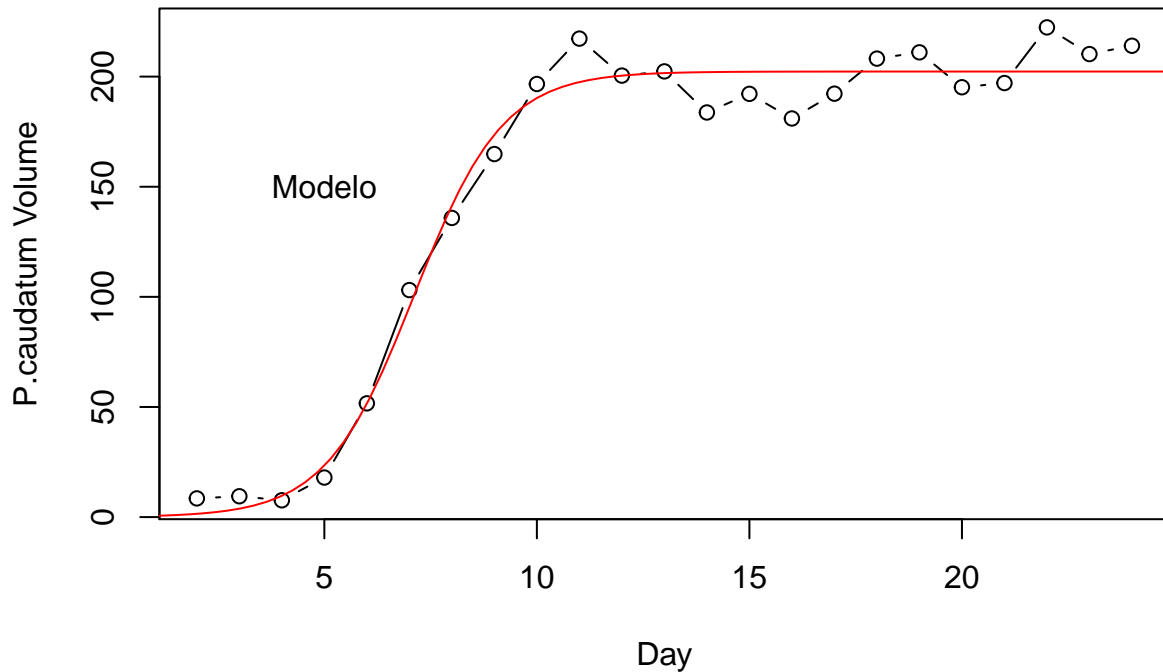
time_seq = seq(0,30,length = 100)
Ntest = get_logistic(time = time_seq, NO=coef(modelo_nls)["NO"],r=coef(modelo_nls)["r"],
```

```

K=coef(modelo_nls)["K"]

#Graficamos el modelo como una línea roja
plot(Volume_Species1 ~ Day, data = Paramecium-caudatum, type = "b", ylab = "P.caudatum Volume")+
lines(time_seq, Ntest, col="red")+ text(5, 150, "Modelo")

```



```
## integer(0)
```

Ahora probaremos que tan bueno es nuestro modelo con el valor `r` utilizando la función `test_goodnes_of_fit` del paquete `gauserR`. En donde valores en o debajo del 0 significan un mal ajuste.

*#Creamos unsmuestra de nuestro modelo. Una pequeña predicción.*

```

model_test <- get_logistic(time = Paramecium-caudatum$Day, NO=coef(modelo_nls)["NO"], r = coef(modelo_nls)["r"])
goodnesModelFit <- test_goodness_of_fit(observed = Paramecium-caudatum$Volume_Species1, predicted = model_test$predicted)
goodnesModelFit #Encontramos que su valor es de 0.9810307

```

```
## [1] 0.9810307
```

El valor R de mínimos cuadrados es mayor a 0. Muy cercano a 1 indicando un ajuste bastante bueno.

Análisis de *Paramecium aurelia* crecidas en monocultivo.