Homework 3 - Covid cases over time

Pau Blasco Roca. NIU: 1600959

15/5/2022

Covid cases evolution analysis

The number of registered cases of COVID-19 in Catalonia, day by day, from 27/02/2020 to 31/03/2020:

y = c(2, 3, 5, 6, 15, 15, 15, 24, 24, 24, 49, 75, 124, 156, 260, 316, 509, 715, 903, 1394, 1866, 2702, 3270, 4203, 4704, 5925, 7864, 9937, 11592, 12940, 1423, 15026, 16157, 18773)

We want to analyze the evolution of the number of affected individuals as a function of time A(t).

Exercise 1: Exponential fit

We will fit the data using these two exponential functions.

$A_1(t) = \exp(\beta_0 + \beta_1 t), \quad A_2(t) = \exp(\beta_0 + \beta_1 t + \beta_2 t^2)$

We are going to assume our data is Poisson distributed. Let's first store our data in a dataset.

```
cases = c(2, 3, 5, 6, 15, 15, 15, 24, 24, 24, 49, 75, 124, 156, 260, 316, 509, 715, 903, 1394, 1866, 27
days = seq(1,length(cases),1)
myData <- data.frame("cases" = cases, "days" = days)
```

We will fit the model with both functions and compare their results. We will start using the **glm** method.

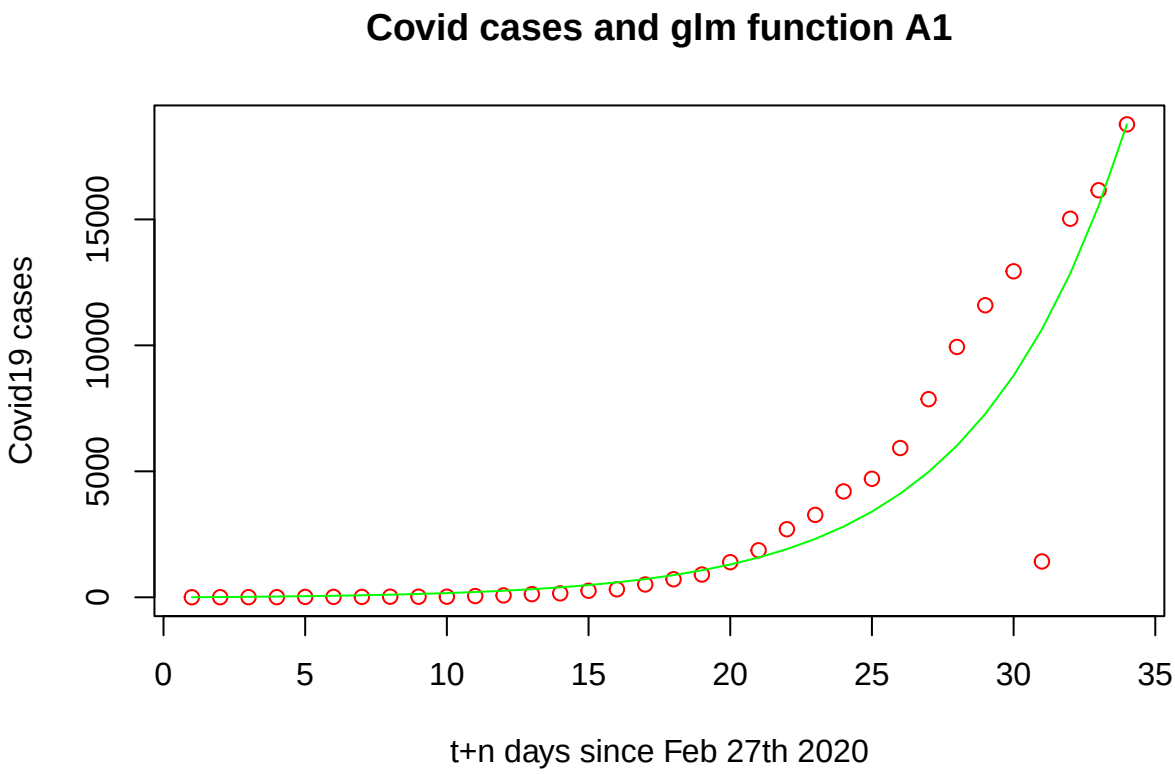
```
t <- seq(1, length(myData$days), 1)
t2 <- t*t
f2 <- glm(cases ~ t, family="poisson", data=myData)
summary(f2)$aic
```

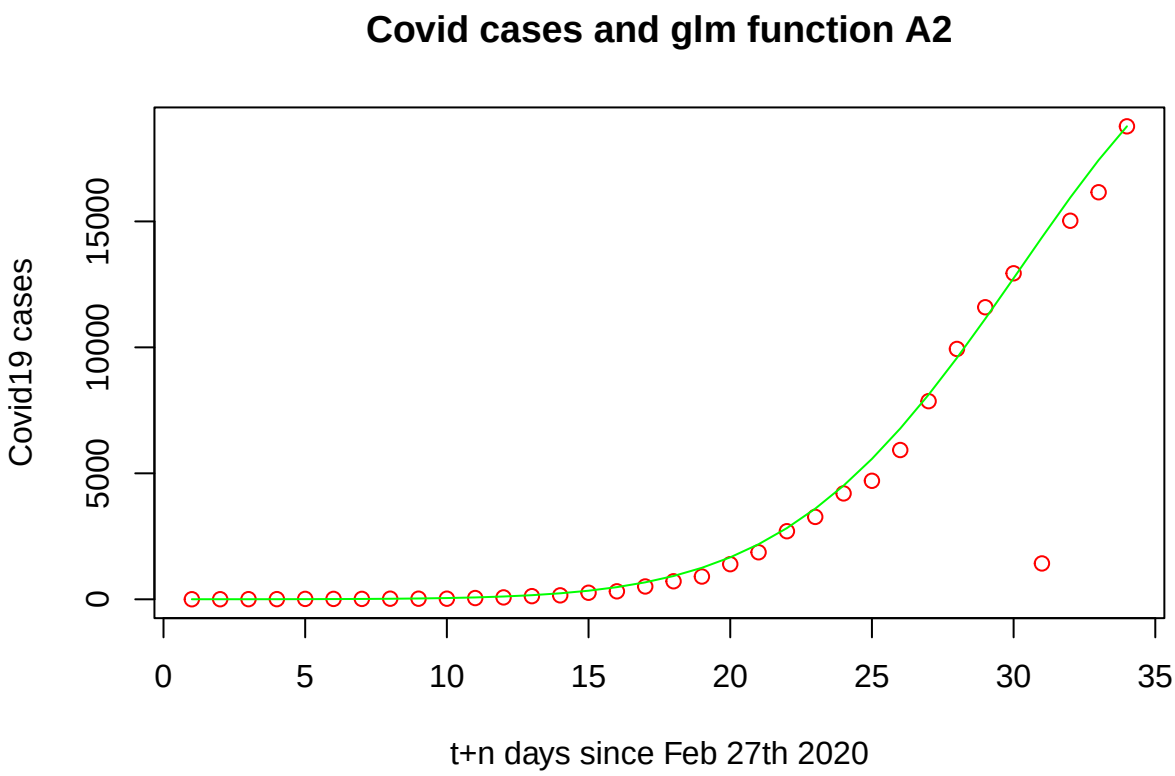
```
## [1] 24278.81
```

```
f4 <- glm(cases ~ t + t2, family="poisson",data=myData)
summary(f4)$aic
```

```
## [1] 17978.65
```

Our results are AIC=24278.81 for the first function and AIC=17978.65 for the second function. We see that, as the second one has a lower value, is the best option for us. Let's plot both functions and study how well they adjust to the data.

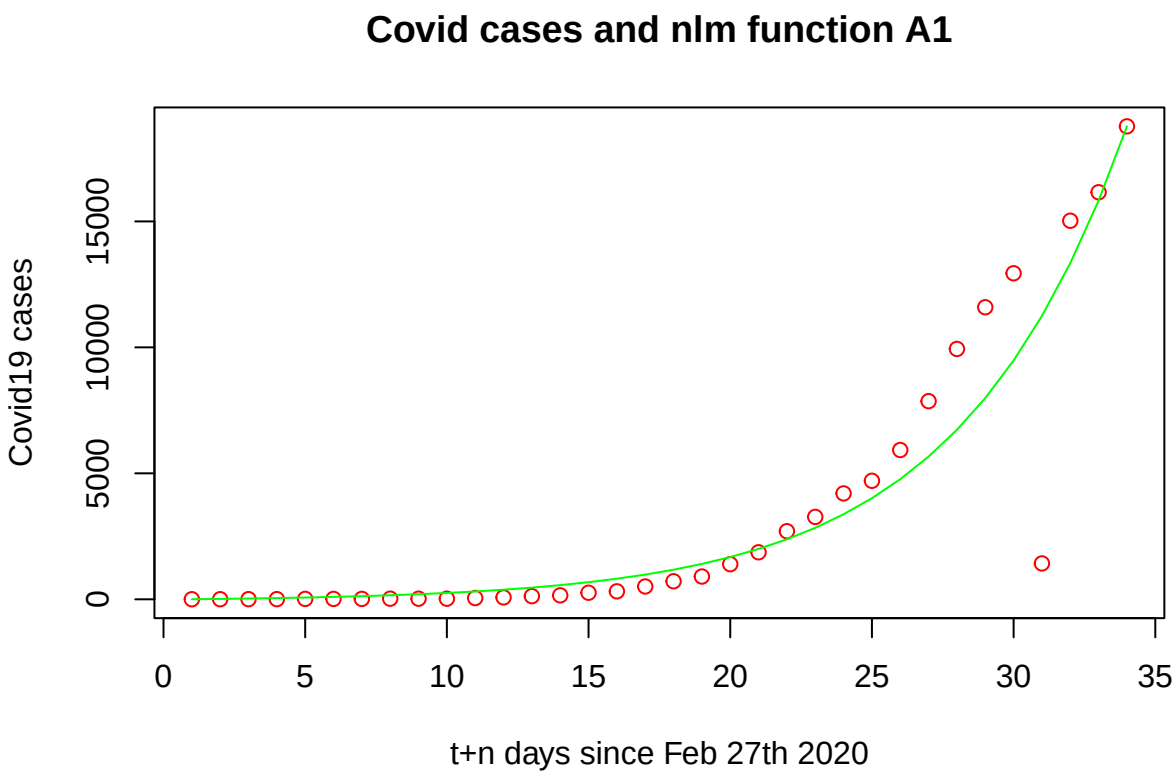

Covid cases and glm function A1
Covid19 cases
0
5000
10000
15000
0
5
10
15
20
25
30
35
t+n days since Feb 27th 2020



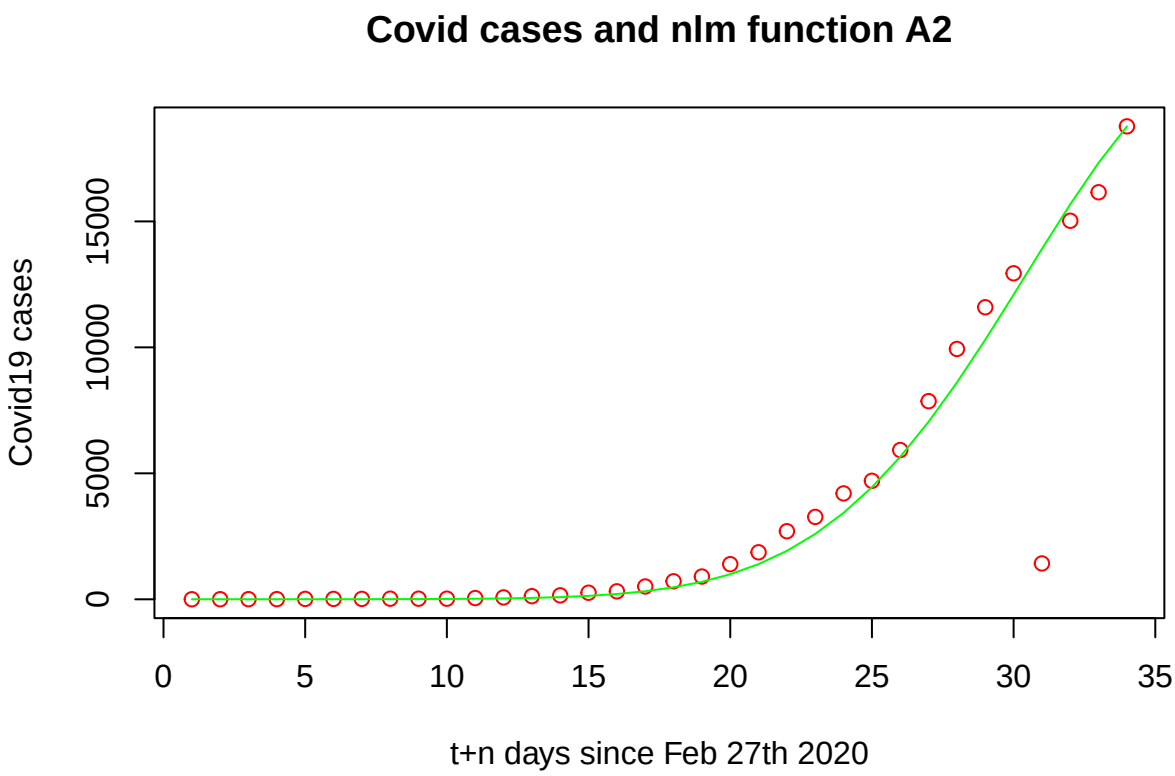
After plotting both options, we can see how the second one adjusts slightly better to our data.

We will now study the model using the **nlm** method. For that, we will first need to compute the maximum likelihood estimator for each function. Luckily, R makes that process really simple.

```
loglik=function(par){
  f = (exp(b0 + b1*t))
  return(-sum(myData$cases*log(f)-f))
}
#These initial parameters seem to give the best results
param=nlm(loglik, p=c(1,0.17),hessian = TRUE)
b0 = param$estimate[1]
b1 = param$estimate[2]
f2c <- function(t) { return (exp(b0 + b1*t)) }
plot(myData$days,  myData$cases, type="p", col="red", ylab="Covid19 cases", xlab="t+n days since Feb 27
par(new=TRUE)
plot(myData$days, f2c(myData$days), type="l", col="green", axes=FALSE, main="Covid cases and nlm functi
```




```
loglik=function(par){
  f = (exp(b0 + b1*t + b2*t**2))
  return(-sum(myData$cases*log(f)-f))
}
#These initial parameters seem to give the best results
param=nlm(loglik, p=c(2,0.75,-0.01),hessian = TRUE)
b0 = param$estimate[1]
b1 = param$estimate[2]
b2 = param$estimate[3]
f4c <- function(t) { return (exp(b0 + b1*t + b2*t**2)) }
plot(myData$days,  myData$cases, type="p", col="red", ylab="Covid19 cases", xlab="t+n days since Feb 27
par(new=TRUE)
plot(myData$days, f4c(myData$days), type="l", col="green", axes=FALSE, main="Covid cases and nlm functi
```

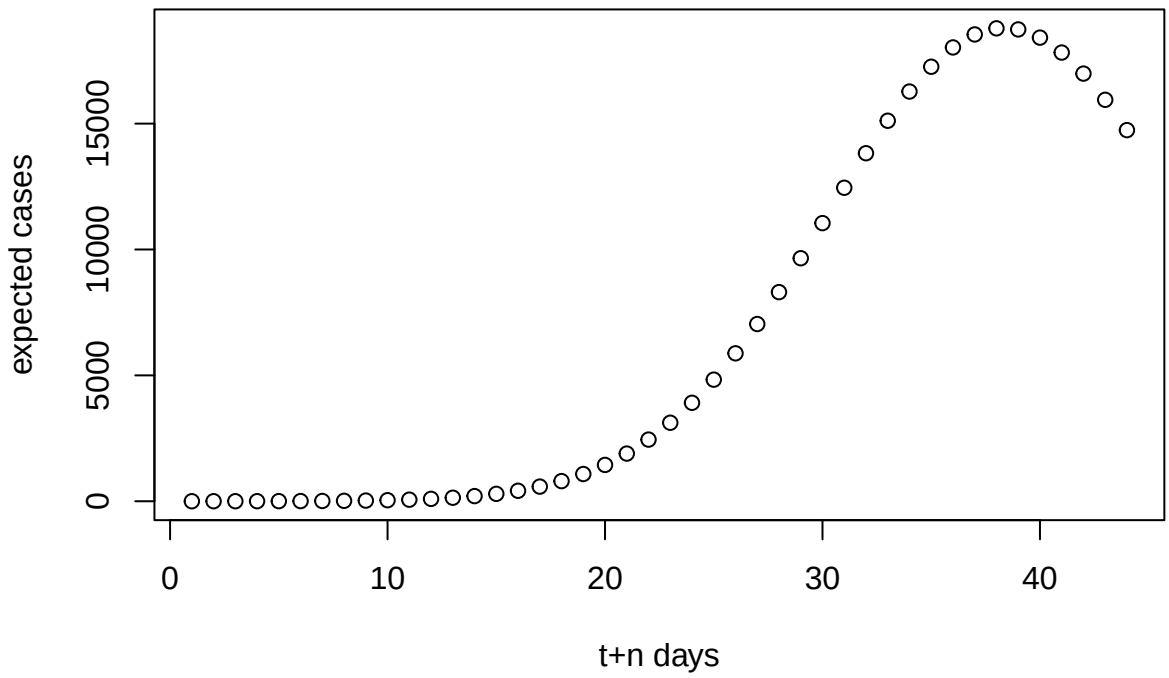


Now we must find the expected time at where the cases will stop growing. For that, we will just optimize A_2.

```
f4 <- glm(cases ~ t + t2, family="poisson",data=myData)
b0 <- summary(f4)$coefficients[1]
b1 <- summary(f4)$coefficients[2]
b2 <- summary(f4)$coefficients[3]
f4c <- function(t) { return (exp(b0 + b1*t + b2*t**2)) }
optimize(f4c, c(1, length(cases)+10), maximum=TRUE)
```

```
## $maximum
## [1] 38.35119
##
## $objective
## [1] 18801.67
```

```
plot(seq(1,length(cases)+10,1), f4c(seq(1,length(cases)+10,1)), xlab="t+n days", ylab="expected cases")
```



As we can clearly see, the maximum takes place in day 38.35 (between days 38 and 39). I have plotted the function for 10 more days to make it clear that the maximum takes place in that day.

Exercise 2: Two-parameters Sigmoid fit

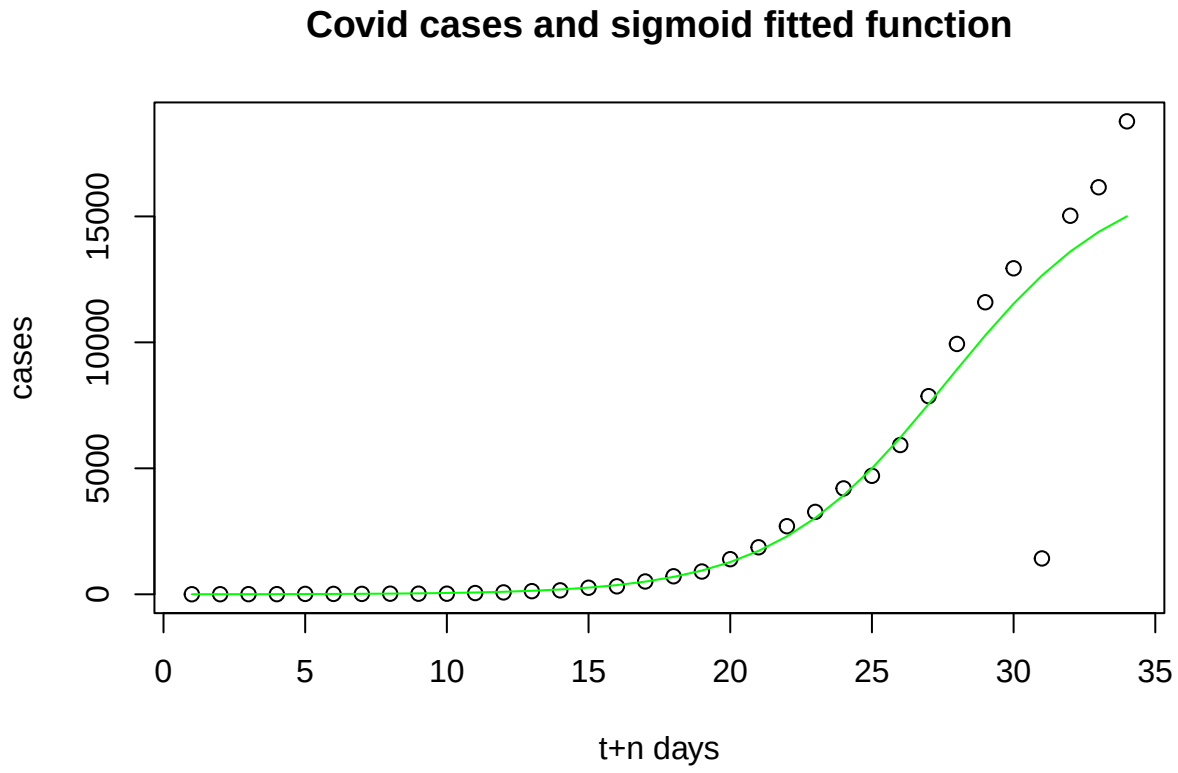
We will fit the data using this sigmoid-like function.

$A(t) = \frac{A_0 C}{A_0 + (C - A_0)\exp(-\beta t)}$, where $A(0) = A_0$ and $\beta > 0$.

This model is not a **glm** model, as the function is not linear and therefore, glm won't be able to handle it. For this, we need to use **nlm**.

```
A0 = myData$cases[1]
loglik=function(par){
  f = ((A0*par[1])/(A0+(par[1]-A0)*exp(-par[2]*myData$days)))
  return(-sum(myData$cases*log(f)-f))
}
#We use the initial parameters as the first value of the sample.
param=nlm(loglik, p=c(2,1),hessian = TRUE)
C = param$estimate[1]
b1 = param$estimate[2]
f6 <- function(a0,x){ return((a0*C)/(a0+(C-a0)*exp(-b1*x)))}

plot(myData$days, myData$cases, pch=1, xlab="t+n days", ylab="cases", main="Covid cases and sigmoid fit
lines(myData$days, f6(A0,myData$days), col="green")
```



We can easily compute the limit of this function by studying it mathematically. Our function, fitted to our data, is the following:

$$A(t) = \frac{2\text{ů}16888}{2 + 16888\exp(-0.32698t)}, \quad \text{where } A(0) = A_0 \text{ and } \beta > 0.$$

Which clearly tends to C, in our case 16888, as time tends to infinity.

To estimate C with a 95% confidence interval, we need to use the following formula:

$$C = \hat{C} \pm 1.96\frac{1}{\sqrt{J(\hat{C})}}$$

Where $J(\hat{C}) = -l''(\hat{C})$). Differentiating two times the log-likelihood, we get the hessian matrix:

```
fi= param$hessian
sol <- solve(fi)
sigm <- sol[1,1]
sqrt(sigm)
```

```
## [1] 101.9012
```

Then, we get a window of ± 1.96ů101.9 which gives us the following CI: $(16688.276, 17087.724)$.