Homework 2

Marta Font 1604517 and Nora Alquézar 1671727

May 3rd, 2024

1 Exercise 1

The dataset we have chosen is 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, 14230, 15026, 16157, 18773)
t <- 1:length(y)
```

1.1 Fitting the data using the exponential functions

We have the following exponential functions:

$$A_1(t) = e^{\beta_0+\beta_1 t}$$
$$A_2(t) = e^{\beta_0+\beta_1 t+\beta_2 t^2}$$

Firstly, we assume that the observations are Poisson distributed and then we fit the models using glm (Model A1) and nlm (Model A2) functions:

```
model_A1 <- glm(y ~ t, family = poisson)
```

Then, we calculate the log-likelihood:

```
log_likelihood_A2 <- function(parameters, y, t) {
  beta0 <- parameters[1]
  beta1 <- parameters[2]
  beta2 <- parameters[3]
  mu <- exp(beta0 + beta1 * t + beta2 * t^2)
  initial_guess <- c(0, 0, 0)  # Assuming initial values for the parameters
Model_A2 <- nlm(log_likelihood_A2, p = initial_guess, y = y, t = t)

  log_lik <- sum(dpois(y, lambda = mu, log = TRUE))
  return(-log_lik)
}
```

We know that the AIC penalizes the model for its complexity, meaning that it takes into account both the goodness of fit of the model to the data and the number of parameters used in the model. Therefore, a model with a lower AIC value achieves a good balance between goodness of fit and model complexity, suggesting that it provides a better trade-off between explaining the data and avoiding overfitting.

In order to do so, we compare both models using the AIC (Akaike Information Criterion) function:

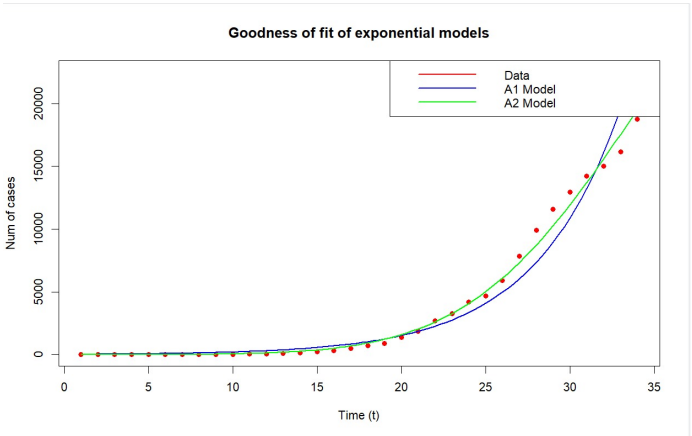
```
AIC_A1 <- AIC(model_A1)
AIC_A2 <- -2 * fit_A2$minimum + 2 * length(initial_guess)
```

```
> cat("AIC of the A1 model:", AIC_A1, "\n")
AIC of the A1 model: 8473.07
> cat("AIC of the A2 model:", AIC_A2, "\n")
AIC of the A2 model: -1573.807
```

We can clearly see than the Model A2 has a lower AIC, meaning that it fits better the data than the Model A1.

In addition, we create a plot to illustrate the fitted models:

```
predicted_A1 <- exp(predict(model_A1, newdata = data.frame(t = t_values)))
predicted_A2 <- exp(beta0_A2 + beta1_A2 * t_values + beta2_A2 * t_values^2)

plot(t, y, ylim = c(0, max(y)*1.2), pch=16, col="red", xlab = "Time (t)",
ylab = "Num of cases", main = "Goodness of fit of exponential models")
lines(t_values, predicted_A1, col = "blue", lwd = 2)
lines(t_values, predicted_A2, col = "green", lwd = 2)
legend("topright", legend = c("Data", "A1 Model", "A2 Model"), col =
c("red", "blue", "green"), lty = c(1, 1,1), lwd = 2)
```



In the plot above, we can observe how the good fit of both models diminishes as time progresses. Initially, both models fit the data nearly perfectly from time 0 to 20. However, beyond this point, we can observe a tendency in both models to underestimate the data until time 30, with model A1 exhibiting a more pronounced tendency. Nevertheless, after time 30, we can see that both models tend to overestimate the data, with model A1 being even more pronounced in this regard.

Lastly, we can anticipate the day on which the number of cases is expected to begin decreasing by using the following code with the model A2:

```
t_max <- -Model_A2$estimate[2] / (2 * Model_A2$estimate[3])
```

```
> # Print the day when the number of cases is expected to begin decreasing
> print(paste("Day when cases start decreasing (according to Model A2):", round
(t_max)))
[1] "Day when cases start decreasing (according to Model A2): 43"
```

In conclusion, we can see that the day on which the number of cases is expected to begin decreasing is on the 43th day.

1.2 Fitting the data using a two-parameters sigmoid function

We have the following two-parameters sigmoid function:

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

The function is not a standard Generalized Linear Model (GLM), due to the fact that in a GLM, the response variable is related to the predictor variables through a linear predictor via a link function, and the response variable follows a distribution from the exponential family.

However, the function provided is a nonlinear function, it's a sigmoid function commonly used to model growth or decay processes.

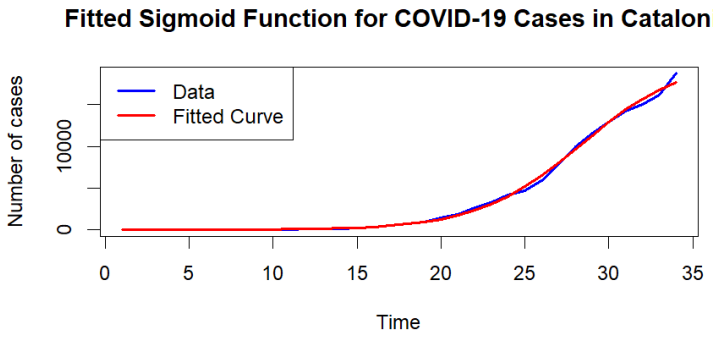
In this case, we fit the data using nlm and assuming again that the observations are Poisson distributed using the following code:

```
poisson_log_likelihood <- function(params, t, y) {
  A0 <- params[1]
  C <- params[2]
  beta <- params[3]

  lambda <- A0 * C / (A0 + (C - A0) * exp(-beta * t))

  # Calculate the negative log-likelihood
  -sum(dpois(y, lambda, log = TRUE))
}
```

```
# Initial parameter values
initial_params <- c(2,18773, 0.5)  # Adjusted C value

# Fit the model using nlm
fit <- nlm(poisson_log_likelihood, p = initial_params, t = t, y = y, hessian=TRUE)
```

Then, we print the plot in order to prove that the model fits the data correctly:

```
# Generate fitted curve using estimated parameters
fitted_curve <- A0 * C / (A0 + (C - A0) * exp(-beta * t))
# Plot the data and the fitted curve
plot(t, y, type = "l", col = "blue", lwd = 2,
xlab = "Time", ylab = "Number of cases",
main = "Fitted Sigmoid Function for COVID-19 Cases in Catalonia")
lines(t, fitted_curve, col = "red", lwd = 2)
legend("topleft", legend = c("Data", "Fitted Curve"),
col = c("blue", "red"), lty = 1, lwd = 2)
```



Subsequently, we display the estimated parameter values:

```
> cat("Estimated parameters:\n")
Estimated parameters:
> cat("A0:", A0, "\n")
A0: 2.122428
> cat("C:", C, "\n")
C: 20507.88
> cat("beta:", beta, "\n")
beta: 0.3234147
```

Then, we calculate the limit of A(t) when t tends to infinity:

```
fit <- nlm(poisson_log_likelihood, p = initial_params, t, y, hessian = TRUE)
# Calculate the limit of A(t) when t tends to infinity
A_infinity <- model_sigmoid$estimate[2]
```

```
# Print the limit of A(t) when t tends to infinity
cat("Limit of A(t) when t tends to infinity:", A_infinity, "\n")
```

```
> cat("Limit of A(t) when t tends to infinity:", limit_inf, "\n")
Limit of A(t) when t tends to infinity: 20507.88
```

Finally, we estimate C giving a 95% confidence interval:

```
# Estimate C and 95% confidence interval
C_estimate <- model_sigmoid$estimate[2]

# Estimate standard errors for parameters
par_se <- solve(model_sigmoid$hessian)
diag(par_se)^.5

# Calculate z-value for 95% confidence interval
z_value <- qnorm(0.975)

# Calculate confidence intervals for parameters
C_interval <- c(C_estimate - z_value * sqrt(par_se[2, 2]),
C_estimate + z_value * sqrt(par_se[2, 2]))


# Print the estimate and confidence interval for C
print(paste("Estimated value of C:", C_estimate))
print(paste("95% Confidence Interval for C:", C_interval[1], C_interval[2]))
```

```
> print(paste("95% Confidence Interval for C:", C_interval[1], C_interval[2]))
[1] "95% Confidence Interval for C: 20173.3237537138 20842.437351424"
```

Since the value 0 not being within the interval implies that the coefficient C is significant in the model. In other words, there is statistical evidence suggesting a significant relationship between the independent variable and the dependent variable, and the coefficient associated with the independent variable (C) is not zero.