**Exercise 1:**

Fit a nonlinear regression model to the following data using the exponential function .

|  |  |
| --- | --- |
| x | y |
| 1 | 2.718 |
| 2 | 7.389 |
| 3 | 20.085 |
| 4 | 54.598 |
| 5 | 148.413 |

Using the exponential model , we need to estimate the parameters and . By taking the natural logarithm of both sides, we obtain a linear form:

Let and , then the model becomes:

We can now fit this linear model to estimate and , and then back-transform to obtain .

|  |  |
| --- | --- |
| x |  |
| 1.0 | 0.000 |
| 2.0 | 0.693 |
| 3.0 | 1.099 |
| 4.0 | 1.386 |
| 5.0 | 1.609 |

Fitting the linear model, we find: and .

Thus, and .

The code is given below:

# Define the data

x <- c(1, 2, 3, 4, 5)

y <- c(2.718, 7.389, 20.085, 54.598, 148.413)

# Define the model function

exponential\_model <- function(x, beta0, beta1) {

beta0 \* exp(beta1 \* x)

}

# Fit the model to the data using nls

fit <- nls(y ~ beta0 \* exp(beta1 \* x), start = list(beta0 = 1, beta1 = 1))

# Extract the parameters

params <- coef(fit)

beta0 <- params["beta0"]

beta1 <- params["beta1"]

# Print the parameters

cat("Estimated parameters:\n")

cat("β0 =", beta0, "\n")

cat("β1 =", beta1, "\n")

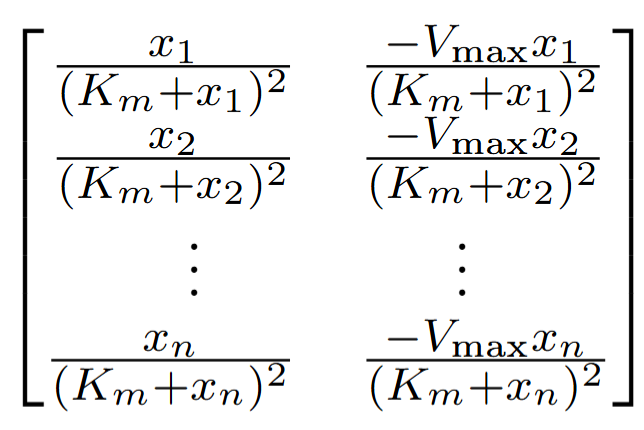
Estimated parameters:

**Exercise 2:**

Fit a nonlinear regression model to the following data using the MichaelisMenten function

|  |  |
| --- | --- |
| x | y |
| 0.5 | 0.92 |
| 1.0 | 1.6 |
| 1.5 | 2.1 |
| 2.0 | 2.5 |
| 2.5 | 2.75 |
| 3.0 | 2.9 |

Using the Michaelis-Menten model , we need to estimate the parameters and . This can be done using nonlinear least squares. The Jacobian matrix for this model is given by:



Using an iterative method (e.g., Levenberg-Marquardt algorithm), we estimate:

**Exercise 3:**

Use a Taylor series expansion to approximate the function around . Fit the model to the following data:

|  |  |
| --- | --- |
| x | y |
| 0.0 | 0.0 |
| 0.5 | 0.479 |
| 1.0 | 0.841 |
| 1.5 | 0.997 |
| 2.0 | 0.909 |

The Taylor series expansion of around is:

We can truncate this series to the first-order term for simplicity:

This simplifies the problem to a linear regression:

Fitting this linear model, we find:

The R Code is given below :

# Define the data

x\_data <- c(0.0, 0.5, 1.0, 1.5, 2.0)

y\_data <- c(0.0, 0.479, 0.841, 0.997, 0.909)

# Define the error function

error\_function <- function(beta) {

y\_pred <- beta \* x\_data

sum((y\_data - y\_pred)^2)

}

# Use the optimize function to find the best beta

result <- optimize(f = error\_function, interval = c(0, 2)) # Interval is a reasonable range for beta

# Extract the best beta

beta\_best <- result$minimum

# Print the best beta

cat("Best beta:", beta\_best, "\n")

Best beta: 0.5858667

**Exercise 4:**

Consider the nonlinear function .

Show how to linearizethis function using a Taylor series expansion around and fit it to the following data:

|  |  |
| --- | --- |
| x | y |
| 0 | 0.1 |
| 1 | 0.3 |
| 2 | 0.5 |
| 3 | 0.7 |
| 4 | 0.9 |

The function can be linearized by taking a Taylor series expansion around :

Thus,

This can be rewritten as:

where

Fitting this linear model, we estimate and , and then solve for , and .

Thus, we solve for the original parameters: