**R0 > 1**

pacotes <- c("plotly","tidyverse","ggrepel","sjPlot","reshape2","knitr",

"kableExtra","FactoMineR")

dias <- 1:9

casos <- c(1,20,40,50,70,100,200,300,500)

exponential.model <-lm(log(casos)~dias)

summary (exponential.model)

somatoria <- sum(casos)

a <- exp (exponential.model$coefficients[[1]])

b <- exponential.model$coefficients[[2]]

gama = 1/9 ##gama is 1/(number of days) (equação , 1/Nd , onde Nd é a quantidade de dias para recuperação)

beta <- b + gama

##Verificando valores

a

b

beta

ro = beta/gama ##Coefienciente de infeccao

ro

## Load deSolve package (carregar pacote) -- disponivel em : https://cran.r-project.org/web/packages/deSolve/

library(deSolve)

## Create an SIR function (criacao da funcao SIR)

sir <- function(time, state, parameters) {

with(as.list(c(state, parameters)), {

dS <- -alpha \* S \* I

dI <- alpha \* S \* I - theta \* I

dR <- theta \* I

return(list(c(dS, dI, dR)))

})

}

### Set parameters (setar parametros)

init <- c(S = 1280/1281, I = 1/1281, R = 0)

## alpha: infection parameter (parametro de infeccao); theta: recovery parameter (parametro de recuperacao)

parameters <- c(alpha = ro, theta = gama)

## Time frame (frame de tempo)

times <- seq(0, 70, by = 1)

## Solve using ode (General Solver for Ordinary Differential Equations) (utilizando EDO na solucao)

out <- ode(y = init, times = times, func = sir, parms = parameters, maxsteps = 1e5)

## change to data frame (mudanca do dataframe)

out <- somatoria\*as.data.frame(out)

## Change number of decimal digits to 2 (quantidade de digitos decimais = 2)

options(digits=2)

print(out)

## Delete time variable (deletando a variavel tempo)

out$time <- NULL

## Show data (apresentando os dados)

head(out, 10)

## Plot (plotando grafico)

matplot(x = times, y = out, type = "l",

xlab = "Time", ylab = "Susceptible and Recovered", main = "SIR Model",

lwd = 1, lty = 1, bty = "l", col = 2:4)

## Add legend (adicionar a legenda)

legend(50, 600, c("Sucetivel", "Infectado", "Recuperado"), pch = 1, col = 2:4, bty = "n")

**R0 < 1**

pacotes <- c("plotly","tidyverse","ggrepel","sjPlot","reshape2","knitr",

"kableExtra","FactoMineR")

dias <- 1:9

casos <- c(1,20,40,50,70,100,200,300,500)

exponential.model <-lm(log(casos)~dias)

summary (exponential.model)

somatoria <- sum(casos)

a <- exp (exponential.model$coefficients[[1]])

b <- exponential.model$coefficients[[2]]

gama = 1/9 ##gama is 1/(number of days) (equação , 1/Nd , onde Nd é a quantidade de dias para recuperação)

##beta <- b + gama

##Verificando valores

a

b

beta

##ro = beta/gama ##Coefienciente de infeccao

ro = 0.5

## Load deSolve package (carregar pacote) -- disponivel em : https://cran.r-project.org/web/packages/deSolve/

library(deSolve)

## Create an SIR function (criacao da funcao SIR)

sir <- function(time, state, parameters) {

with(as.list(c(state, parameters)), {

dS <- -alpha \* S \* I

dI <- alpha \* S \* I - theta \* I

dR <- theta \* I

return(list(c(dS, dI, dR)))

})

}

### Set parameters (setar parametros)

init <- c(S = 1280/1281, I = 1/1281, R = 0)

## alpha: infection parameter (parametro de infeccao); theta: recovery parameter (parametro de recuperacao)

parameters <- c(alpha = ro, theta = gama)

## Time frame (frame de tempo)

times <- seq(0, 70, by = 1)

## Solve using ode (General Solver for Ordinary Differential Equations) (utilizando EDO na solucao)

out <- ode(y = init, times = times, func = sir, parms = parameters, maxsteps = 1e5)

## change to data frame (mudanca do dataframe)

out <- somatoria\*as.data.frame(out)

## Change number of decimal digits to 2 (quantidade de digitos decimais = 2)

options(digits=2)

print(out)

## Delete time variable (deletando a variavel tempo)

out$time <- NULL

## Show data (apresentando os dados)

head(out, 10)

## Plot (plotando grafico)

matplot(x = times, y = out, type = "l",

xlab = "Time", ylab = "Susceptible and Recovered", main = "SIR Model",

lwd = 1, lty = 1, bty = "l", col = 2:4)

## Add legend (adicionar a legenda)

legend(50, 600, c("Sucetivel", "Infectado", "Recuperado"), pch = 1, col = 2:4, bty = "n")

**R0 = 1**

pacotes <- c("plotly","tidyverse","ggrepel","sjPlot","reshape2","knitr",

"kableExtra","FactoMineR")

dias <- 1:9

casos <- c(1,20,40,50,70,100,200,300,500)

exponential.model <-lm(log(casos)~dias)

summary (exponential.model)

somatoria <- sum(casos)

a <- exp (exponential.model$coefficients[[1]])

b <- exponential.model$coefficients[[2]]

gama = 1/9 ##gama is 1/(number of days) (equação , 1/Nd , onde Nd é a quantidade de dias para recuperação)

##beta <- b + gama

##Verificando valores

a

b

beta

##ro = beta/gama ##Coefienciente de infeccao

ro = 1

## Load deSolve package (carregar pacote) -- disponivel em : https://cran.r-project.org/web/packages/deSolve/

library(deSolve)

## Create an SIR function (criacao da funcao SIR)

sir <- function(time, state, parameters) {

with(as.list(c(state, parameters)), {

dS <- -alpha \* S \* I

dI <- alpha \* S \* I - theta \* I

dR <- theta \* I

return(list(c(dS, dI, dR)))

})

}

### Set parameters (setar parametros)

init <- c(S = 1280/1281, I = 1/1281, R = 0)

## alpha: infection parameter (parametro de infeccao); theta: recovery parameter (parametro de recuperacao)

parameters <- c(alpha = ro, theta = gama)

## Time frame (frame de tempo)

times <- seq(0, 70, by = 1)

## Solve using ode (General Solver for Ordinary Differential Equations) (utilizando EDO na solucao)

out <- ode(y = init, times = times, func = sir, parms = parameters, maxsteps = 1e5)

## change to data frame (mudanca do dataframe)

out <- somatoria\*as.data.frame(out)

## Change number of decimal digits to 2 (quantidade de digitos decimais = 2)

options(digits=2)

print(out)

## Delete time variable (deletando a variavel tempo)

out$time <- NULL

## Show data (apresentando os dados)

head(out, 10)

## Plot (plotando grafico)

matplot(x = times, y = out, type = "l",

xlab = "Time", ylab = "Susceptible and Recovered", main = "SIR Model",

lwd = 1, lty = 1, bty = "l", col = 2:4)

## Add legend (adicionar a legenda)

legend(50, 600, c("Sucetivel", "Infectado", "Recuperado"), pch = 1, col = 2:4, bty = "n")