Project

```r
# Initializing the libraries
library(deSolve)
library(ggplot2)
library(reshape2)
```

```r
# Setting the population size (21 kids in the school)
N=21
```

```r
# Building the SIR function
sir_model <- function(time, var, param){
  with(as.list(c(var, param)),{
    S_factor <- -(beta*I*S)
    I_factor <- (beta*I*S)-(gamma*I)
    R_factor <- gamma*I
    return(list(c(S_factor, I_factor, R_factor)))
  })
}
```

```r
# Setting up the parameters
# S=No of Susceptible population
# I=No of Infected population
# R=No of Recovered population
init_vals <- c(S=20,I=1,R=0)

# Seting up the time period to 50 days
time_vals <- seq(0, 50)

# Setting up rates
# beta=infection rate (per person/no of days)
# gamma=recovery rate (/no of days)
param_vals <- c(beta=0.02, gamma=1/3)
```

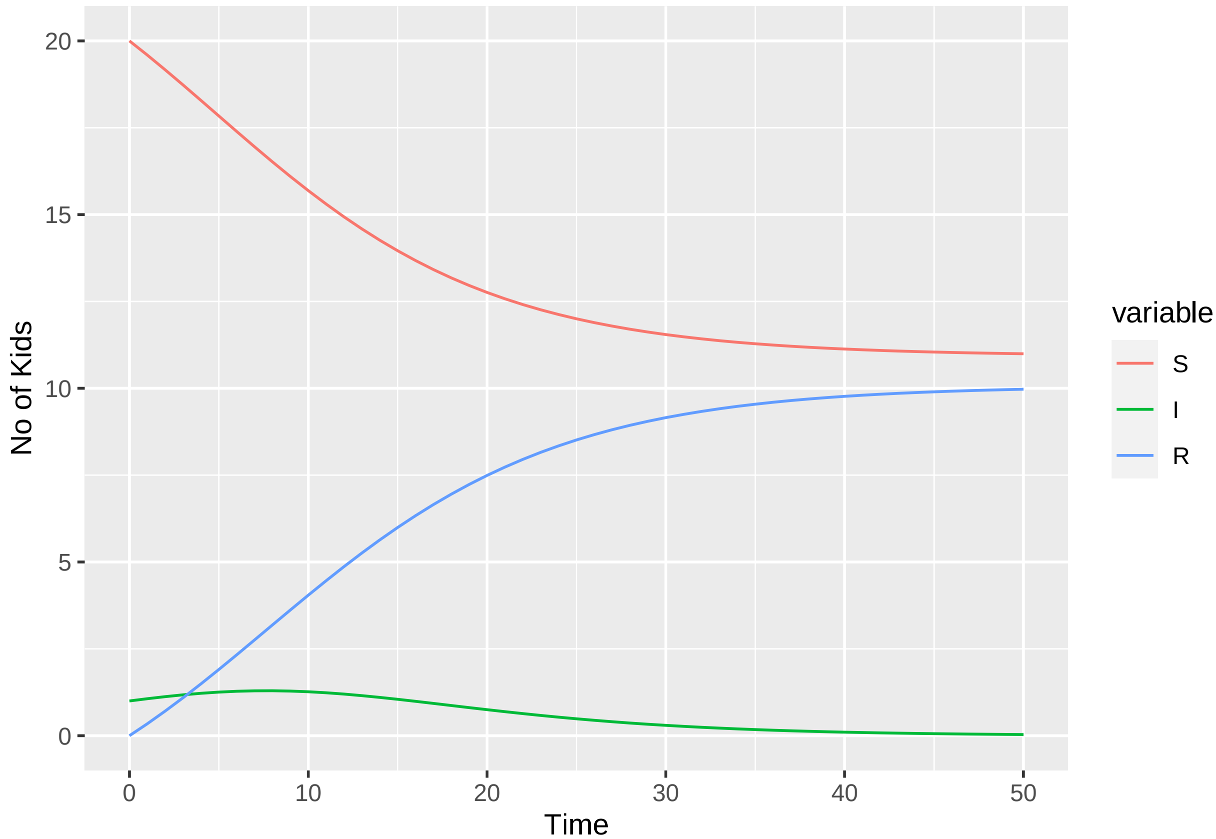
```r
# Calling the SIR function with the parameters
sir_results <- ode(init_vals, time_vals, sir_model, param_vals)
head(sir_results)
```

```
##      time        S        I         R
## [1,]    0 20.00000 1.000000 0.0000000
## [2,]    1 19.59117 1.064608 0.3442226
## [3,]    2 19.16686 1.123981 0.7091552
## [4,]    3 18.73073 1.176491 1.0927803
## [5,]    4 18.28679 1.220655 1.4925558
## [6,]    5 17.83929 1.255229 1.9054849
```

```r
sir_results <- as.data.frame(sir_results)

# Melting the results for visualization
sir_plot<-melt(sir_results,id="time")

# Plotting the SIR model with the parameters setup
ggplot(data = sir_plot, aes(x = time, y = value, colour = variable, group = variable)) + geom_line() + :
```




```r
# Building the SEIR function
seir_model <- function(time, var, param){
  with(as.list(c(var, param)),{
    S_factor <- -(beta*I*S)
    I_factor <- (beta*I*S)-(gamma*I)
    R_factor <- gamma*I
    E_factor <- (beta*I*S)-(delta*E)
    return(list(c(S_factor, E_factor, I_factor, R_factor)))
  })
}
```

```r
# Setting up the parameters
# S=No of Susceptible population
# E=No of Exposed population
# I=No of Infected population
# R=No of Recovered population
init_vals2 <- c(S=20,E=0, I=1,R=0)
```

```r
# Seting up the time period to 50 days
time_vals2 <- seq(0, 50)

# Setting up rates
# beta=infection rate (per person/no of days)
# gamma=recovery rate (/no of days)
# delta=exposure rate (/no of days)
param_vals2 <- c(beta=0.02, gamma=1/3, delta=1/2)
```

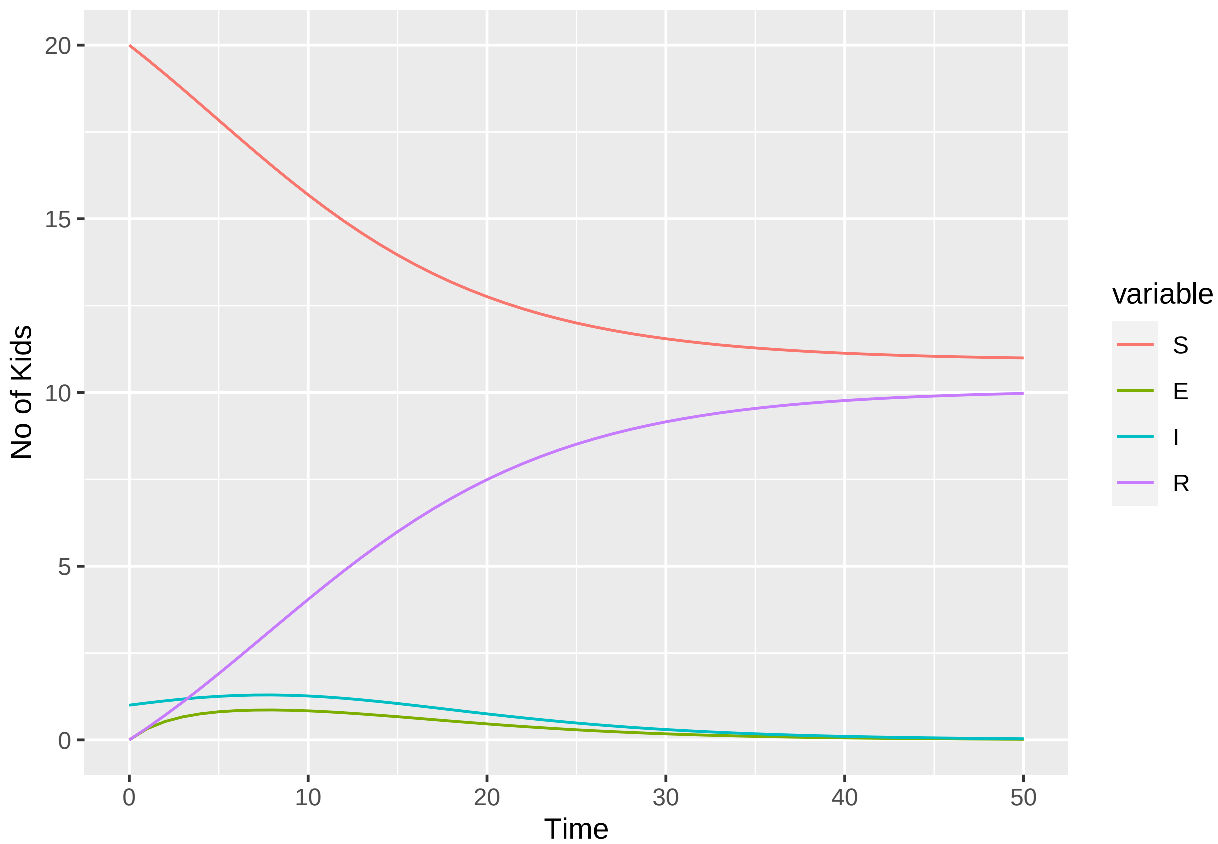
```r
# Calling the SEIR function with the parameters
seir_results <- ode(init_vals2, time_vals2, seir_model, param_vals2)
head(seir_results)
```

```
##      time        S         E        I         R
## [1,]    0 20.00000 0.0000000 1.000000 0.0000000
## [2,]    1 19.59117 0.3222844 1.064608 0.3442236
## [3,]    2 19.16686 0.5298273 1.123981 0.7091570
## [4,]    3 18.73073 0.6648865 1.176490 1.0927811
## [5,]    4 18.28679 0.7528116 1.220654 1.4925558
## [6,]    5 17.83929 0.8088040 1.255228 1.9054839
```

```r
seir_results <- as.data.frame(seir_results)

# Melting the results for visualization
seir_plot<-melt(seir_results,id="time")

# Plotting the SEIR model with the parameters setup
ggplot(data = seir_plot, aes(x = time, y = value, colour = variable, group = variable)) + geom_line() +
```




```
# Building the SIRS function
sirs_model <- function(time, var, param){
  with(as.list(c(var, param)),{
    S_factor <- -(beta*I*S)+(sigma*R)
    I_factor <- (beta*I*S)-(gamma*I)
    R_factor <- (gamma*I)-(sigma*R)
    return(list(c(S_factor, I_factor, R_factor)))
  })
}
```

```
# Setting up the parameters
# S=No of Susceptible population
# I=No of Infected population
# R=No of Recovered population
init_vals3 <- c(S=20,I=1,R=0)

# Seting up the time period to 50 days
time_vals3 <- seq(0, 50)

# Setting up rates
# beta=infection rate (per person/no of days)
# gamma=recovery rate (/no of days)
# sigma=re-infection rate (/no of days)
param_vals3 <- c(beta=0.02, gamma=1/3, sigma=1/20)
```

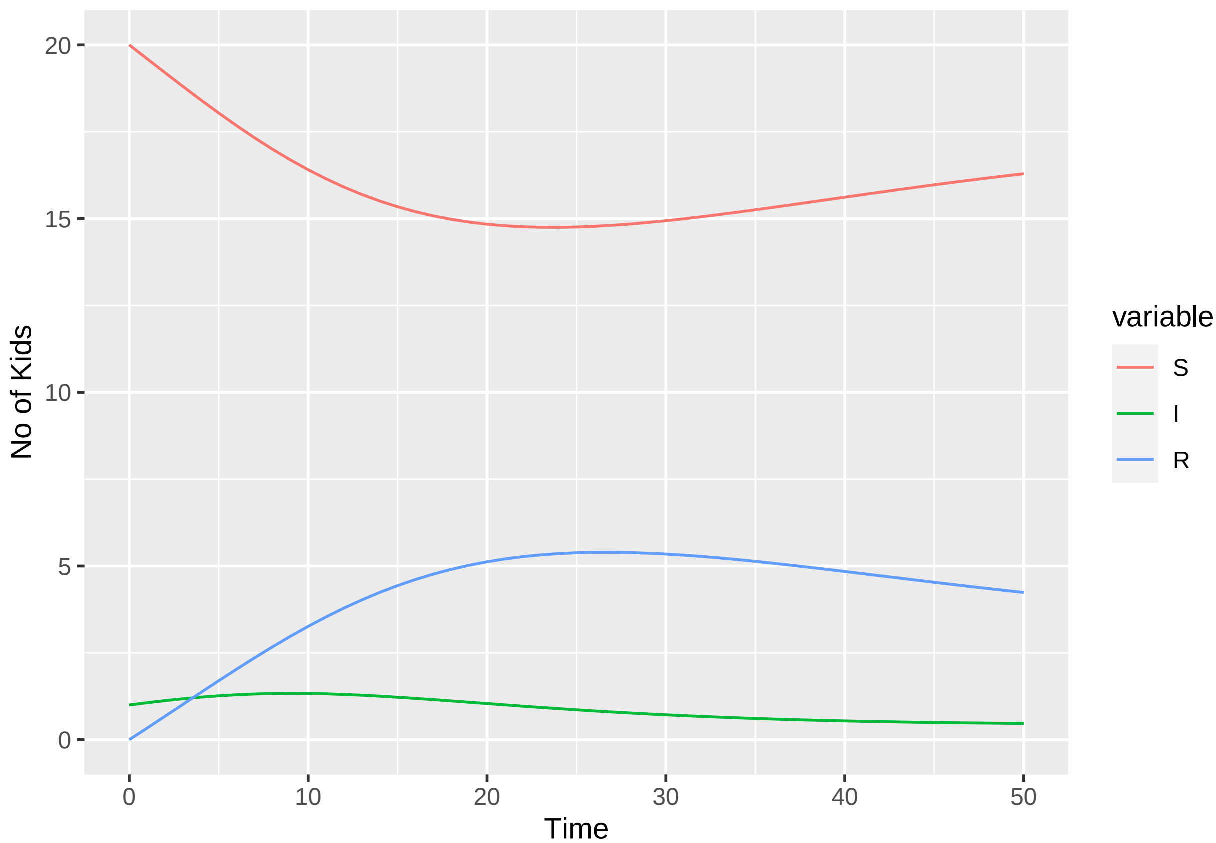
```r
# Calling the SIRS function with the parameters
sirs_results <- ode(init_vals3, time_vals3, sirs_model, param_vals3)
head(sirs_results)
```

```
##      time        S        I         R
## [1,]    0 20.00000 1.000000 0.0000000
## [2,]    1 19.59948 1.064668 0.3358516
## [3,]    2 19.19993 1.124479 0.6755872
## [4,]    3 18.80452 1.178239 1.0172405
## [5,]    4 18.41649 1.224930 1.3585768
## [6,]    5 18.03907 1.263753 1.6971747
```

```r
sirs_results <- as.data.frame(sirs_results)

# Melting the results for visualization
sirs_plot<-melt(sirs_results,id="time")

# Plotting the SIRS model with the parameters setup
ggplot(data = sirs_plot, aes(x = time, y = value, colour = variable, group = variable)) + geom_line() +
```




```r
# Building the SEIRS function
seirs_model <- function(time, var, param){
  with(as.list(c(var, param)),{
    S_factor <- -(beta*I*S)+(sigma*R)
```

```r
    I_factor <- (beta*I*S)-(gamma*I)
    R_factor <- gamma*I-(sigma*R)
    E_factor <- (beta*I*S)-(delta*E)
    return(list(c(S_factor, E_factor, I_factor, R_factor)))
  })
}
```

```r
# Setting up the parameters
# S=No of Susceptible population
# E=No of Exposed population
# I=No of Infected population
# R=No of Recovered population
init_vals4 <- c(S=20,E=0, I=1,R=0)

# Seting up the time period to 50 days
time_vals4 <- seq(0, 50)

# Setting up rates
# beta=infection rate (per person/no of days)
# gamma=recovery rate (/no of days)
# delta=exposure rate (/no of days)
# sigma=re-infection rate (/no of days)
param_vals4 <- c(beta=0.02, gamma=1/3, delta=1/2, sigma=1/20)
```

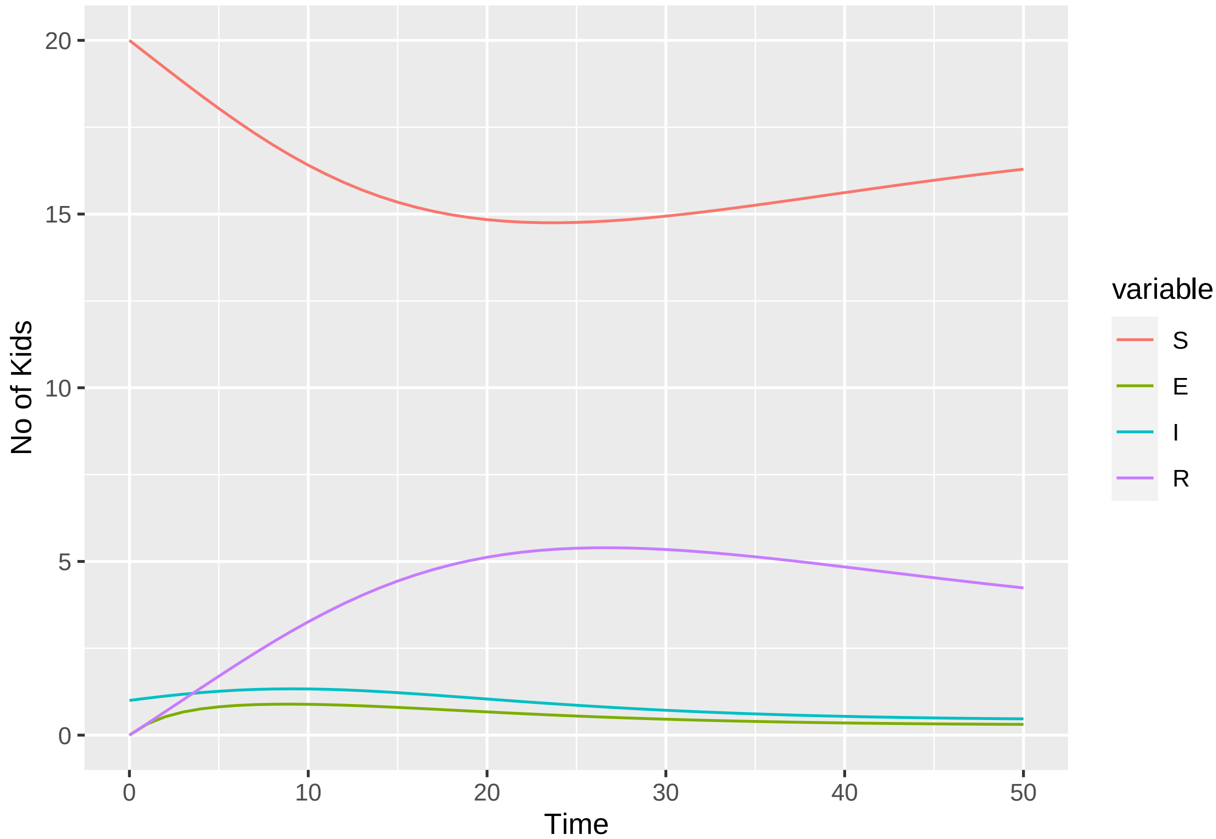
```r
# Calling the SEIRS function with the parameters
seirs_results <- ode(init_vals4, time_vals4, seirs_model, param_vals4)
head(seirs_results)
```

```
##      time        S         E        I         R
## [1,]    0 20.00000 0.0000000 1.000000 0.0000000
## [2,]    1 19.59948 0.3223412 1.064667 0.3358520
## [3,]    2 19.19993 0.5302907 1.124478 0.6755881
## [4,]    3 18.80452 0.6664715 1.178239 1.0172403
## [5,]    4 18.41650 0.7565943 1.224928 1.3585752
## [6,]    5 18.03908 0.8161884 1.263751 1.6971722
```

```r
seirs_results <- as.data.frame(seirs_results)

# Melting the results for visualization
seirs_plot<-melt(seirs_results,id="time")

# Plotting the SEIRS model with the parameters setup
ggplot(data = seirs_plot, aes(x = time, y = value, colour = variable, group = variable)) + geom_line() +
```




```
# Building the SIRS function with vaccinations
vac_model <- function(time, var, param){
  with(as.list(c(var, param)),{
    S_factor <- -(beta*I*S)
    I_factor <- (beta*I*S)-(gamma*I)
    R_factor <- gamma*I
    return(list(c(S_factor, I_factor, R_factor)))
  })
}
```

```
# Setting up the parameters
# S=No of Susceptible population (60% unvaccinated)
# I=No of Infected population
# R=No of Recovered population (40% vaccinated)
init_vals5 <- c(S=0.6*20,I=1,R=0.4*20)

# Seting up the time period to 50 days
time_vals5 <- seq(0, 50)

# Setting up rates
# beta=infection rate (per person/no of days)
# gamma=recovery rate (/no of days)
param_vals5 <- c(beta=0.02, gamma=1/3)
```

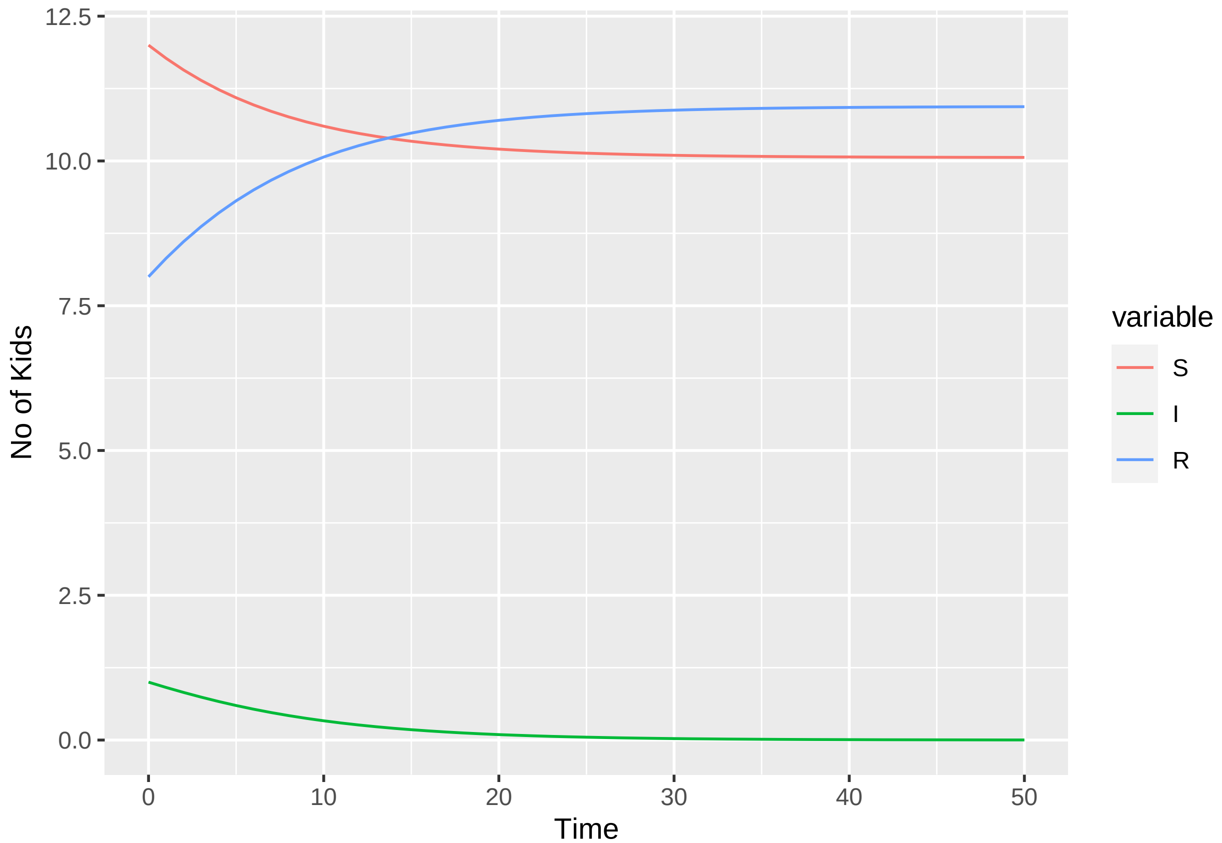
```r
# Calling the SIRS function accounting for vaccinations with the parameters
vac_results <- ode(init_vals5, time_vals5, vac_model, param_vals5)
head(vac_results)
```

```
##      time        S         I        R
## [1,]    0 12.00000 1.0000000 8.000000
## [2,]    1 11.77320 0.9087868 8.318010
## [3,]    2 11.57124 0.8223622 8.606393
## [4,]    3 11.39182 0.7413278 8.866852
## [5,]    4 11.23273 0.6660232 9.101247
## [6,]    5 11.09190 0.5965776 9.311518
```

```r
vac_results <- as.data.frame(vac_results)

# Melting the results for visualization
vac_plot<-melt(vac_results,id="time")

# Plotting the SIRS model accounting for vaccinations with the parameters setup
ggplot(data = vac_plot, aes(x = time, y = value, colour = variable, group = variable)) + geom_line() + 
```




```r
# Simulating SIR Model 1000 times using Binomial(20,0.02)
set.seed(100)
x=rbinom(1000,size=20,p=0.02)/20
x[6]
```

```
## [1] 0
```

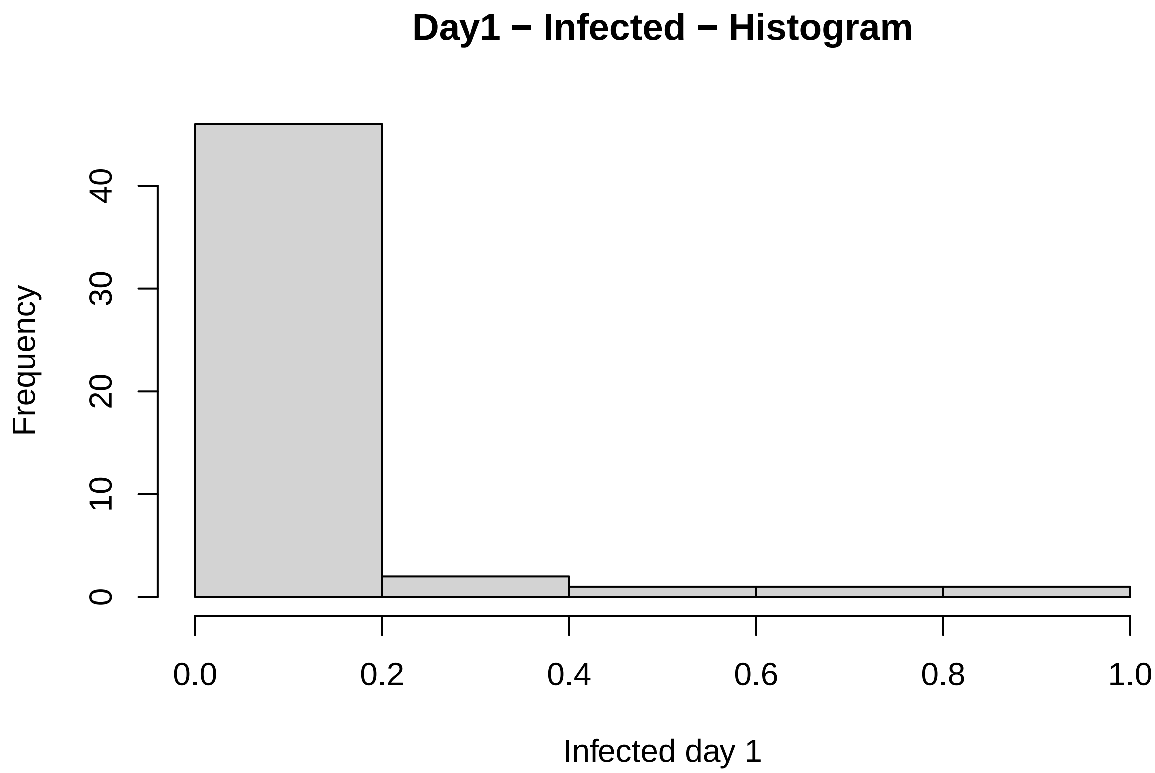
```r
s_result = data.frame(matrix(ncol=1000,nrow=51))
i_result = data.frame(matrix(ncol=1000,nrow=51))
r_result = data.frame(matrix(ncol=1000,nrow=51))
for (val in c(1:1000))
{
  init_vals <- c(S=20,I=1,R=0)
  time_vals <- seq(0, 50)
  param_vals <- c(beta=x[val], gamma=1/3)
  sir_results <- ode(init_vals, time_vals, sir_model, param_vals)

  i_result[,val] = sir_results[,3]
  s_result[,val] = sir_results[,2]
  r_result[,val] = sir_results[,4]
}
```

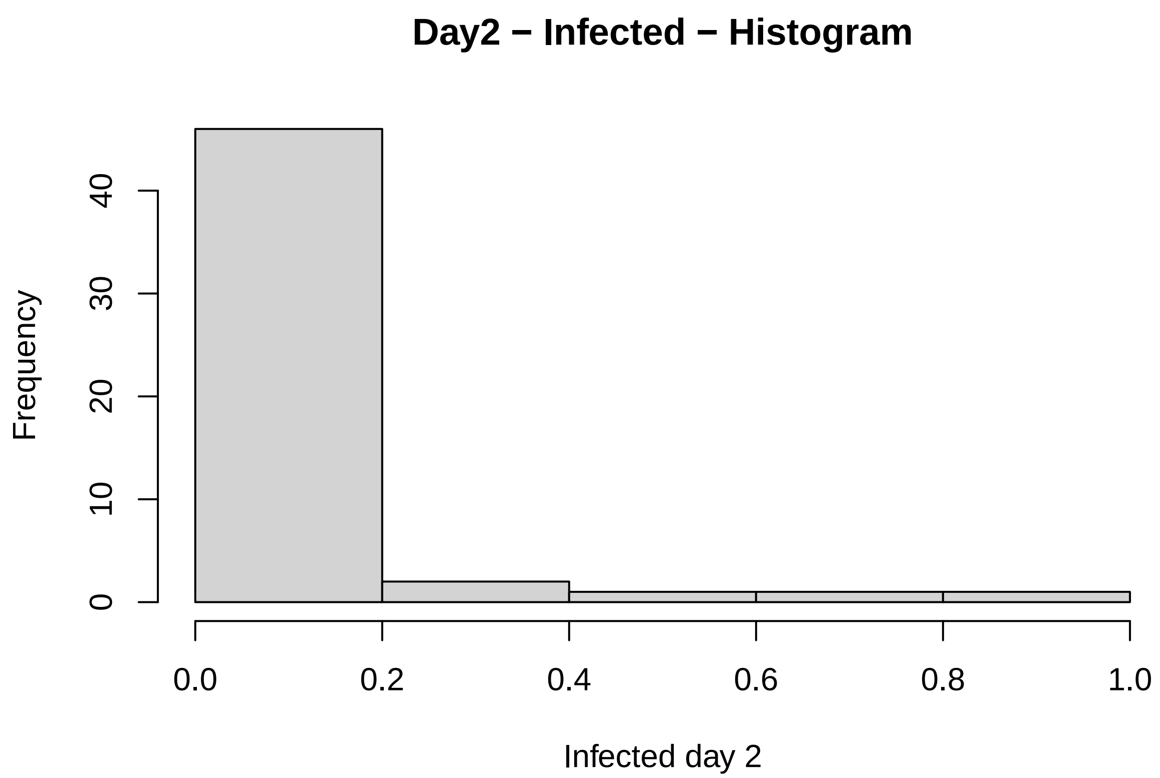
```r
I=rowMeans(i_result)
S=rowMeans(s_result)
R=rowMeans(r_result)
gg = cbind(S,I,R)
head(gg)
```

```
##             S        I         R
## [1,] 20.00000 1.000000 0.0000000
## [2,] 19.26641 1.355881 0.3777079
## [3,] 18.07107 1.991583 0.9373454
## [4,] 16.90575 2.415140 1.6791102
## [5,] 15.86351 2.613557 2.5229333
## [6,] 15.00759 2.595153 3.3972558
```

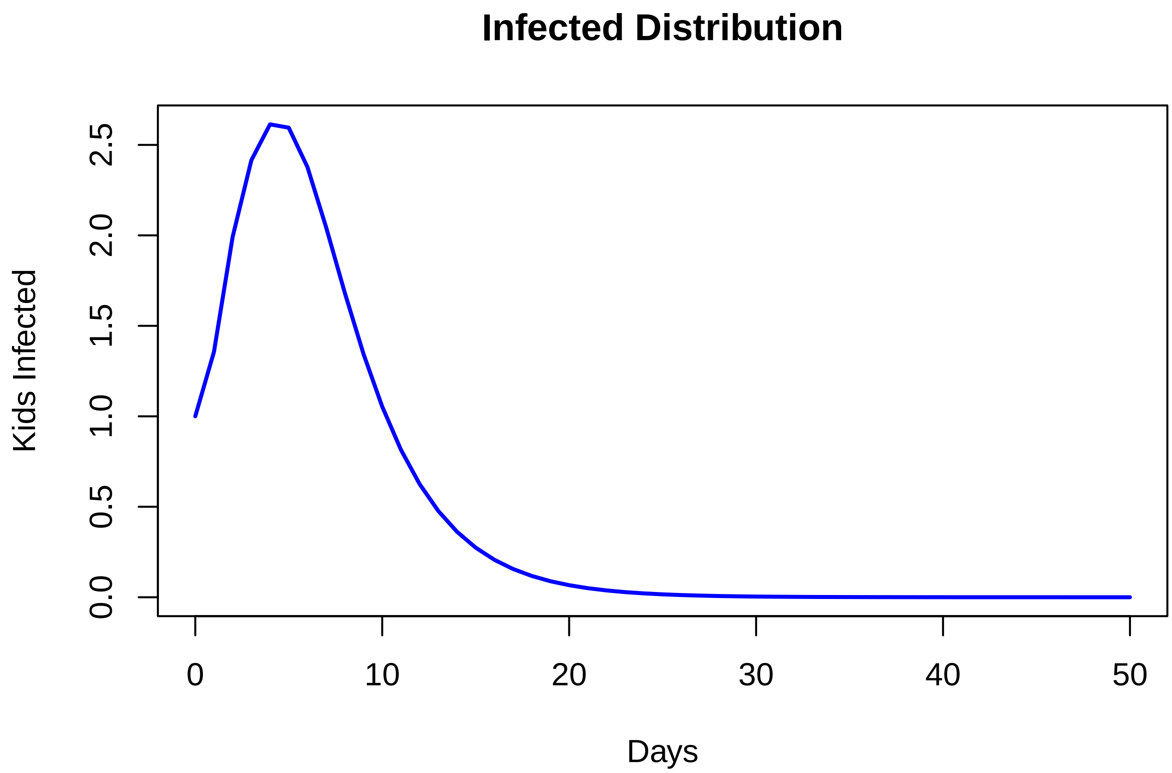
```r
##Q(a)
hist(as.numeric(i_result[,2]),main="Day1 - Infected - Histogram", xlab="Infected day 1")
```




```r
##Q(c)
hist(as.numeric(i_result[,3]),main="Day2 - Infected - Histogram", xlab="Infected day 2")
```

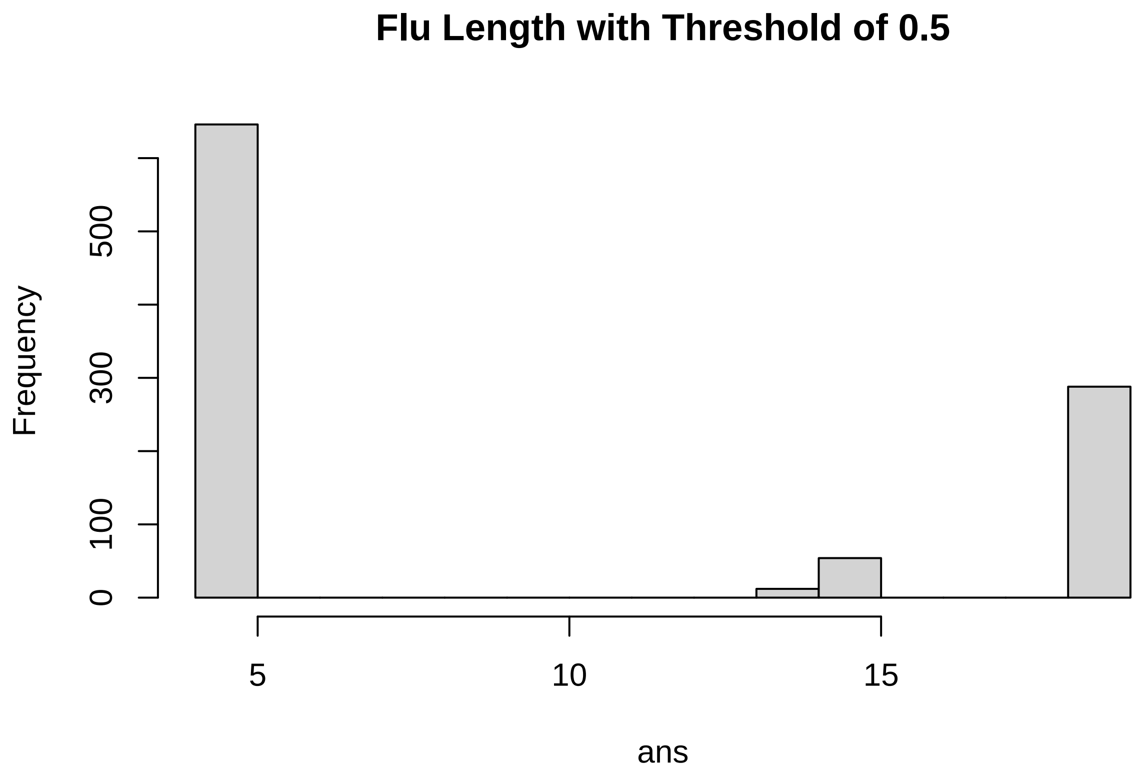



```
##Q(d1)
plot(c(0:50),I, type="l", col="blue", lwd=2, xlab="Days", ylab="Kids Infected", main="Infected Distribu
```




```r
##Q(d2)
# Threshold > 0.5

ans=c()
for (val in c(1:1000))
{
  ans[val] = min(which(i_result[,val] < 0.5))
}
h=hist(ans, main="Flu Length with Threshold of 0.5")
```




```r
unique(ans)
```

```
## [1]  4 19 15 14
```

```r
h$breaks
```

```
##  [1]  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
```

```r
h$counts
```

```
##  [1] 646   0   0   0   0   0   0   0   0  12  54   0   0   0 288
```

```r
h$counts/1000
```

```
##  [1] 0.646 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.012 0.054 0.000
## [13] 0.000 0.000 0.288
```

```r
# Calling the SIR function with the parameters
sir_results <- ode(init_vals, time_vals, sir_model, param_vals)
head(sir_results)
```

```
##      time         S         I         R
```

```
## [1,]    0 20.000000 1.000000 0.000000
## [2,]    1 18.642765 1.888740 0.468495
## [3,]    2 16.421372 3.264304 1.314324
## [4,]    3 13.380124 4.940133 2.679743
## [5,]    4 10.063191 6.357823 4.578986
## [6,]    5  7.180642 6.990403 6.828954
```

```
sir_results[,3]
```

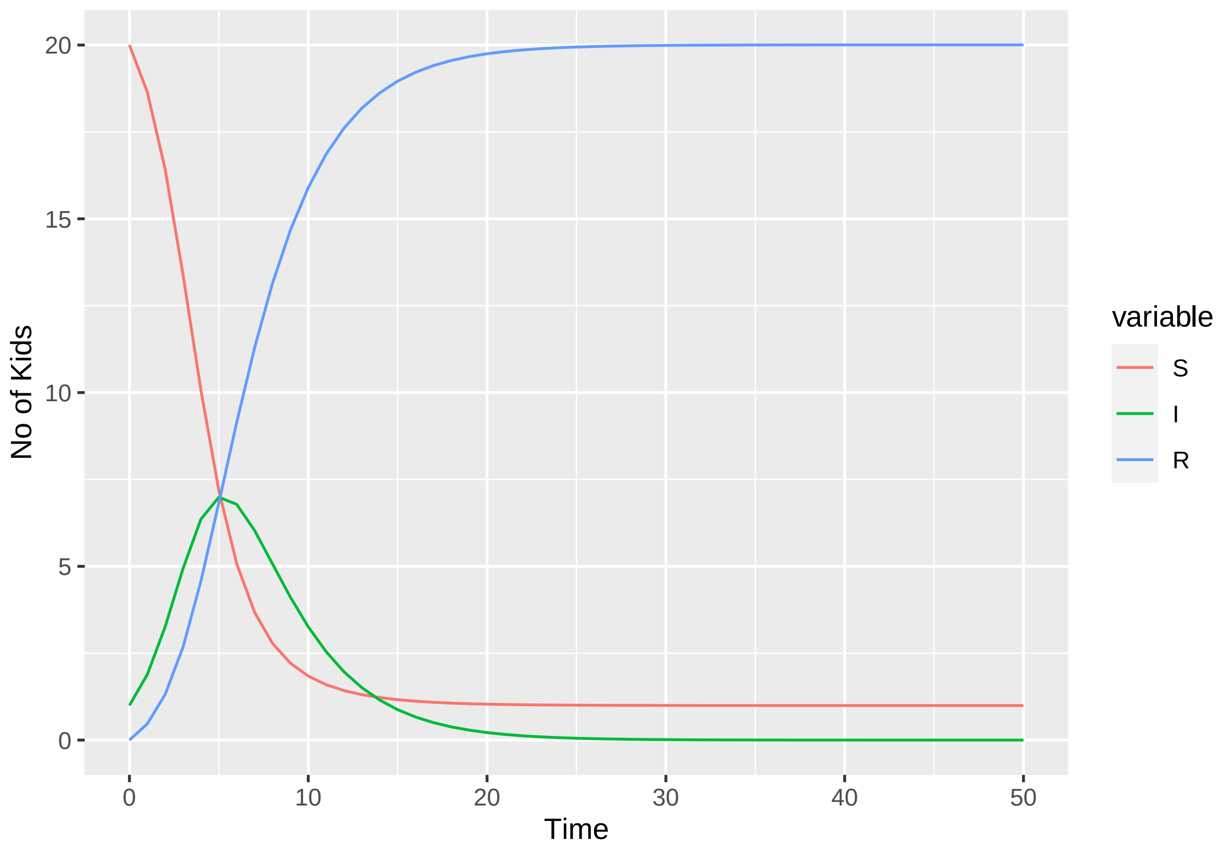
```
##  [1] 1.000000e+00 1.888740e+00 3.264304e+00 4.940133e+00 6.357823e+00
##  [6] 6.990403e+00 6.781902e+00 6.032457e+00 5.071857e+00 4.113667e+00
## [11] 3.260098e+00 2.544530e+00 1.965702e+00 1.507814e+00 1.150836e+00
## [16] 8.752551e-01 6.639564e-01 5.027246e-01 3.801211e-01 2.871250e-01
## [21] 2.167160e-01 1.634805e-01 1.232702e-01 9.292074e-02 7.002686e-02
## [26] 5.276421e-02 3.975176e-02 2.994537e-02 2.255642e-02 1.698971e-02
## [31] 1.279627e-02 9.637558e-03 7.258386e-03 5.466446e-03 4.116834e-03
## [36] 3.100408e-03 2.334939e-03 1.758439e-03 1.324266e-03 9.972919e-04
## [41] 7.510484e-04 5.655847e-04 4.259111e-04 3.207393e-04 2.415385e-04
## [46] 1.818941e-04 1.369785e-04 1.031539e-04 7.768169e-05 5.849867e-05
## [51] 4.405801e-05
```

```
report_result = data.frame(matrix(ncol=1000,nrow=51))
```

```
sir_results <- as.data.frame(sir_results)

# Melting the results for visualization
sir_plot<-melt(sir_results,id="time")

# Plotting the SIR model with the parameters setup
ggplot(data = sir_plot, aes(x = time, y = value, colour = variable, group = variable)) + geom_line() + :
```



20
15
10
5
0
No of Kids
0
10
20
30
40
50
Time
variable
S
I
R