Week 1 - Solving DEs - Self

Sixtus Dakurah

3/24/2020

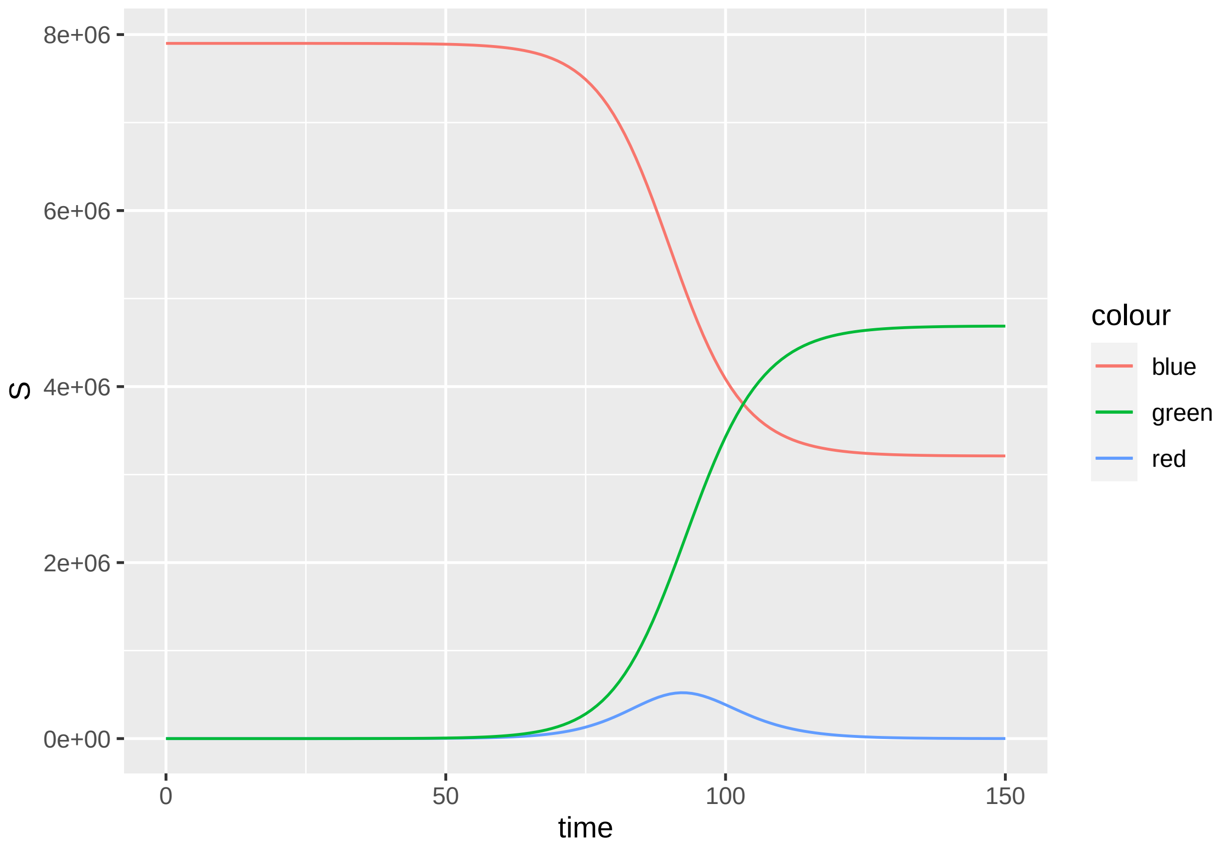
```r
# defining model parameters

# initialized params and state variables here ... these are fixed nnonetheless
alpha.0 = 1/3; beta.0 = 0.5; N.0=7900000; prop.val <- FALSE; delta <- 1
times <- seq(0, 150, by = delta)

S.0 = 7900000; I.0 = 10; R.0 = 0

# alpha: recovery rate, beta: contacts per given time sufficient for spread
params <- c(alpha = alpha.0, beta = beta.0, N=N.0)

# state variables
if(prop.val){
  state <- c(S = S.0/N.0, I=I.0/N.0, R = R.0/N.0)
}else{
  state <- c(S = S.0, I=I.0, R = R.0)
}
```

```r
# self implementation

RateOfChange.Self <- function(times, state, params)
{
  df <- data.frame(time = times)
  S.list <- c(NULL); I.list <- c(NULL); R.list <- c(NULL)
  # convert the rates into named list

  with(as.list(c(state, params)), {
    for (time in times){
      # this is to ensure that the first values are the initials
      # update the variables 1
      S.list <- c(S.list, S); I.list <- c(I.list, I); R.list <- c(R.list, R);

      # rate of change of subsceptible population
      S.change <- S*(1 - delta*beta*(I/N))
      #dS <- (S.change - S)/delta

      # rate of change of the recovered population
      #dR <- alpha*I
      R.change <- R + delta*alpha*I

      # rate of change of the infectious population
      I.change <- N - S.change - R.change
      #dI <- beta*(S/N)*I - alpha*I

      # update the variables 2
      S <- S.change; I <- I.change; R <- R.change

    }

    df$S <- S.list; df$I <- I.list; df$R <- R.list
    return(df)
  })

}
```

```r
data <- RateOfChange.Self(times, state, params)
head(data)
```

```
##   time       S         I        R
## 1    0 7900000 10.000000 0.000000
## 2    1 7899995  1.666667 3.333333
## 3    2 7899994  1.944444 3.888889
## 4    3 7899993  2.268517 4.537037
## 5    4 7899992  2.646602 5.293209
## 6    5 7899991  3.087701 6.175410
```

```r
# make a combine plot
ggplot(data = data, aes(time)) + geom_line(aes(y = S, colour = "blue")) +
  geom_line(aes(y = I, colour = "red")) +
  geom_line(aes(y = R, colour = "green"))
```


8e+06
6e+06
4e+06
2e+06
0e+00
S
0
50
100
150
time
colour
blue
green
red