

R Notebook

Emily Guthrie

5/18/2020

```
library(deSolve)
library(plsgenomics)
```

For any news related to the 'plsgenomics' package (update, corrected bugs), please check <http://thot>
C++ based sparse PLS routines will soon be available on the CRAN in the new 'fastPLS' package.

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.0      v purrr   0.3.4
## v tibble  3.0.1      v dplyr   0.8.5
## v tidyr   1.0.2      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.5.0
```

```
## Warning: package 'purrr' was built under R version 3.6.2
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

From literature: R0 is 2.2 duration for E is 3 duration for I is 7

Basic SEIR model with no handwashing

Model

```
OpenSEIR <- function(t, state, parameters) {
  with(as.list(c(state, parameters)), {
    N = S + E + I + R

    dS <- -beta * S * I/N + birth * N - death * S + omega *
      R
    dE <- beta * S * I/N - sigma * E - death * E
    dI <- sigma * E - death * I - gamma * I
    dR <- gamma * I - death * R - omega * R

    # return the rates of change as a list
    list(c(dS, dE, dI, dR))
  })
}
```

Parameters

```
R0 <- 2.2
latent <- 3
infectious <- 7

birth <- 0
death <- 0
omega <- 0 #waning immunity
gamma <- 1/infectious #recovery rate
sigma <- 1/latent
beta <- R0 * gamma #effective contact rate
```

Functions to analyze output

```
compute_cumulative_infection_time <- function(output) {
  df1 <- data.frame(output)
  return(sum(df1$I))
}

show_SEIR_model_results <- function(df1) {
  df1 %>% as_tibble() %>% mutate(S = as.numeric(S), E = as.numeric(E),
    I = as.numeric(I), R = as.numeric(R)) %>% pivot_longer(cols = c("S",
    "E", "I", "R"), names_to = "compartment", values_to = "count") %>%
    mutate(compartment = factor(compartment, levels = c("S",
    "E", "I", "R"))) %>% ggplot(aes(x = time, y = count,
    color = compartment)) + geom_line() + xlab("Time") +
    ylab("Count")
}
```

Run model

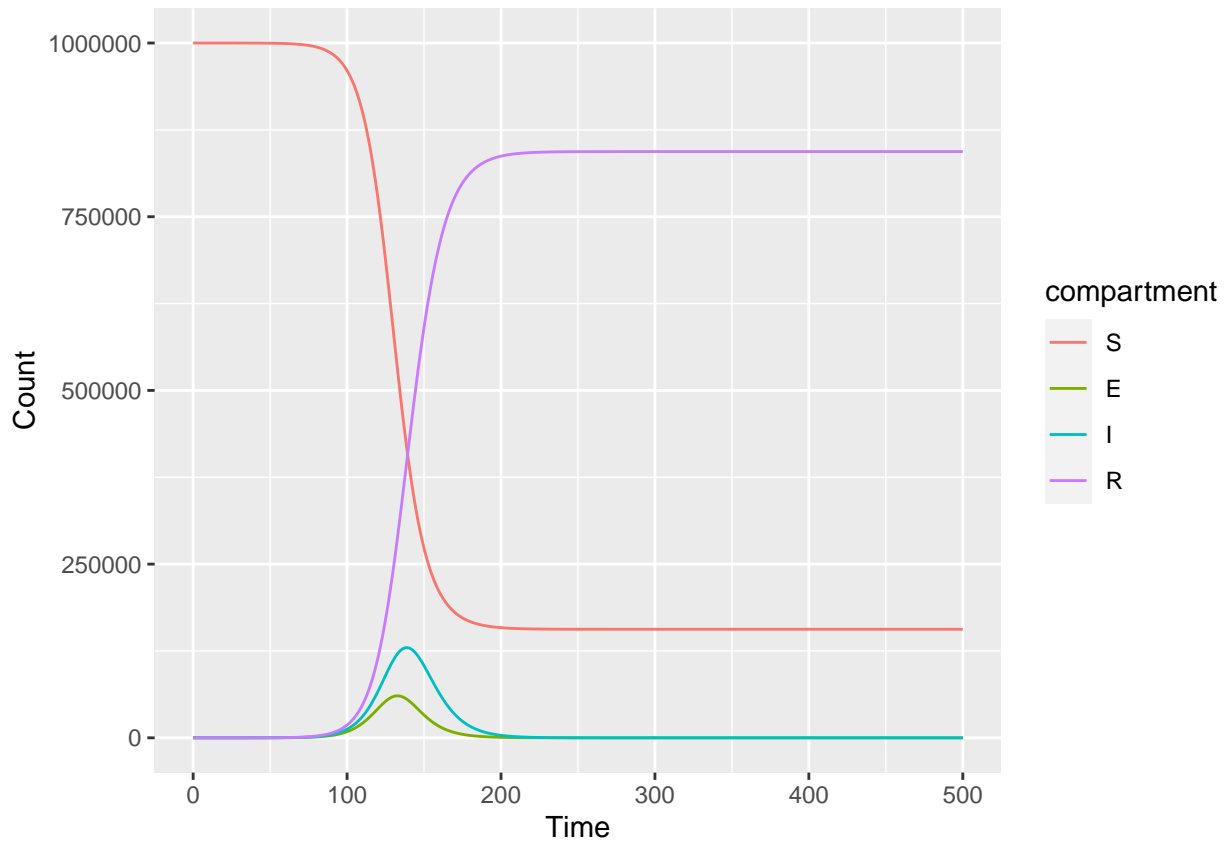
```
state <- c(S = 999999, #population of 1,000,000, 1 person starts of infected
  E = 0,
  I = 1,
  R = 0
)

T_end <- 500 #run model for 500 time steps
times <- seq(0, T_end, by = 1) #runs the model for 500 time steps, and computes output at each time step

output_basic_seir <- ode(y = state, times = times, func = OpenSEIR, parms = c(beta, gamma, omega, sigma))

output_basic_seir %>% show_SEIR_model_results()

## Don't know how to automatically pick scale for object of type deSolve/matrix. Defaulting to continuous
```



```
compute_cumulative_infection_time(output_basic_seir)
```

```
## [1] 5906174
```

Basic SEIR with handwashing - affects infectious only

Model assuming handwashing only impacts infectious people

```
handwash_I_SEIR <- function(t, state, parameters) {
  with(as.list(c(state, parameters)), {
    N <- S + E + I + R

    dW <- -theta * W + p * (N - W)

    dS <- -beta1 * S * (1 - W/N) * (I/N) - beta2 * S * (W/N) *
      (I/N) + birth * N - death * S + omega * R
    dE <- beta1 * S * (1 - W/N) * (I/N) + beta2 * S * (W/N) *
      (I/N) - sigma * E - death * E
    dI <- sigma * E - death * I - gamma * I
    dR <- gamma * I - death * R - omega * R

    # return the rates of change as a list
    list(c(dS, dE, dI, dR, dW))
  })
}
```

Parameters

```

R0 <- 2.2
latent <- 3
infectious <- 7
lambda <- 0.2 #this is the percent by which handwashing reduces effective contact rate
p <- 0.2 #prob that someone washes hands
theta <- 0.2 #rate that handwashing wears off

birth <- 0
death <- 0
omega <- 0 #waning immunity
gamma <- 1/infectious #recovery rate
sigma <- 1/latent
beta1 <- R0 * gamma #effective contact rate
beta2 <- (1 - lambda) * beta1

```

Run model

```

state <- c(S = 999999, #population of 1,000,000, 1 person starts of infected
          E = 0,
          I = 1,
          R = 0,
          W = 0
        )

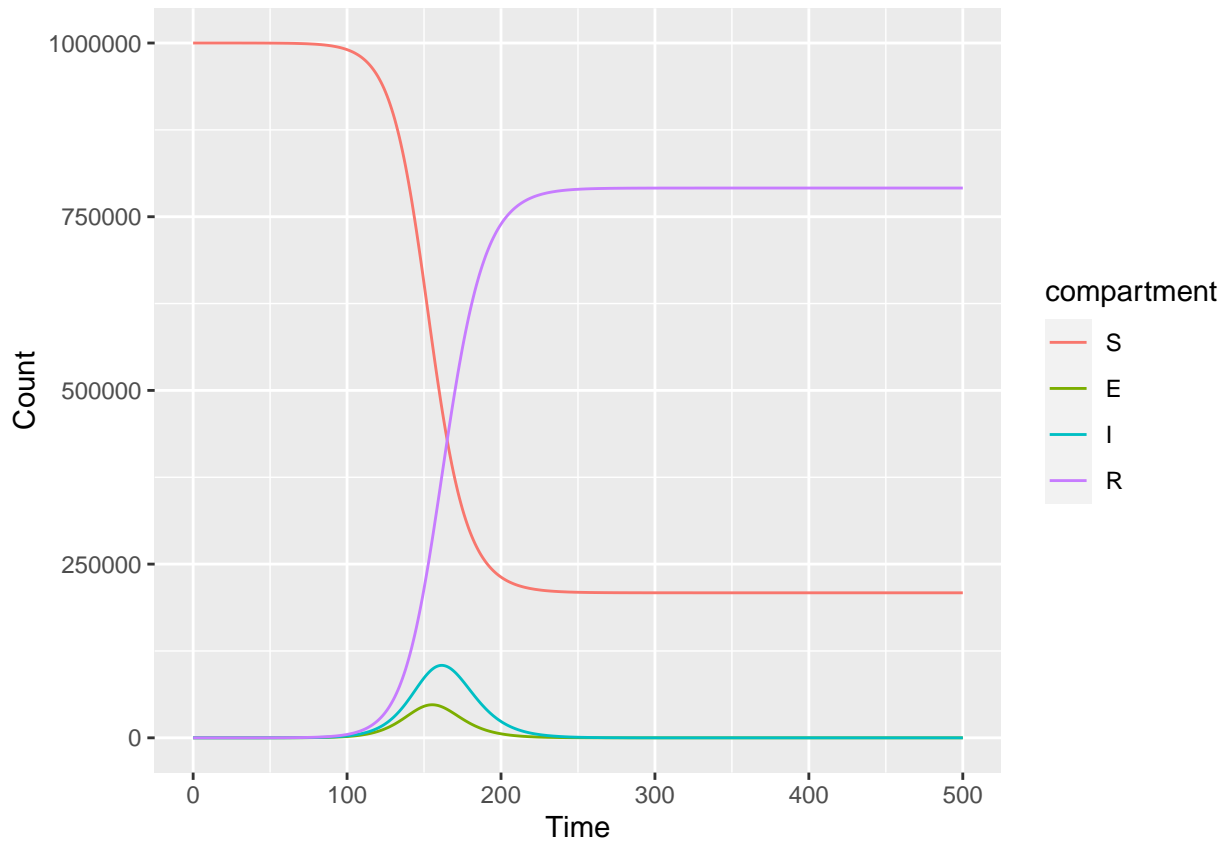
T_end <- 500 #run model for 500 time steps
times <- seq(0, T_end, by = 1) #runs the model for 500 time steps, and computes output at each time step

output_handwash_seir <- ode(y = state, times = times, func = handwash_I_SEIR, parms = c(beta1, beta2, gamma, omega, sigma, theta, lambda, p))

output_handwash_seir %>% show_SEIR_model_results()

```

Don't know how to automatically pick scale for object of type deSolve/matrix. Defaulting to continuous



```
compute_cumulative_infection_time(output_handwash_seir)
```

```
## [1] 5538896
```

Basic SEIR with handwashing - affects infectious and susceptible

We could model this with two separate E compartments – E1 would be people exposed to covid, but they could eliminate infection by handwashing, and E2 would be people who are exposed and are incubating the virus.

```
handwash_SI_SEIR <- function(t, state, parameters) {
  with(as.list(c(state, parameters)), {
    N <- S + E + I + R

    dW <- -theta * W + p * (N - W)

    dS <- -beta1 * S * (1 - W/N) * (I/N) - beta2 * S * (W/N) *
      (I/N) + birth * N - death * S + omega * R + alpha *
      (W/N) * E
    dE <- beta1 * S * (1 - W/N) * (I/N) + beta2 * S * (W/N) *
      (I/N) - sigma * E - death * E - alpha * (W/N) * E
    dI <- sigma * E - death * I - gamma * I
    dR <- gamma * I - death * R - omega * R

    # return the rates of change as a list
    list(c(dS, dE, dI, dR, dW))
  })
}
```

```
}
```

Parameters

```
R0 <- 2.2
latent <- 3
infectious <- 7
lambda <- 0.2 #this is the percent by which handwashing reduces effective contact rate
p <- 0.2 #prob that someone washes hands
theta <- 0.2 #rate that handwashing wears off
alpha <- 0.2 #proportion of people with washed hands return to susceptible (AKA, dont become infectious)

birth <- 0
death <- 0
omega <- 0 #waning immunity
gamma <- 1/infectious #recovery rate
sigma <- 1/latent
beta1 <- R0 * gamma #effective contact rate
beta2 <- (1 - lambda) * beta1
```

Run model

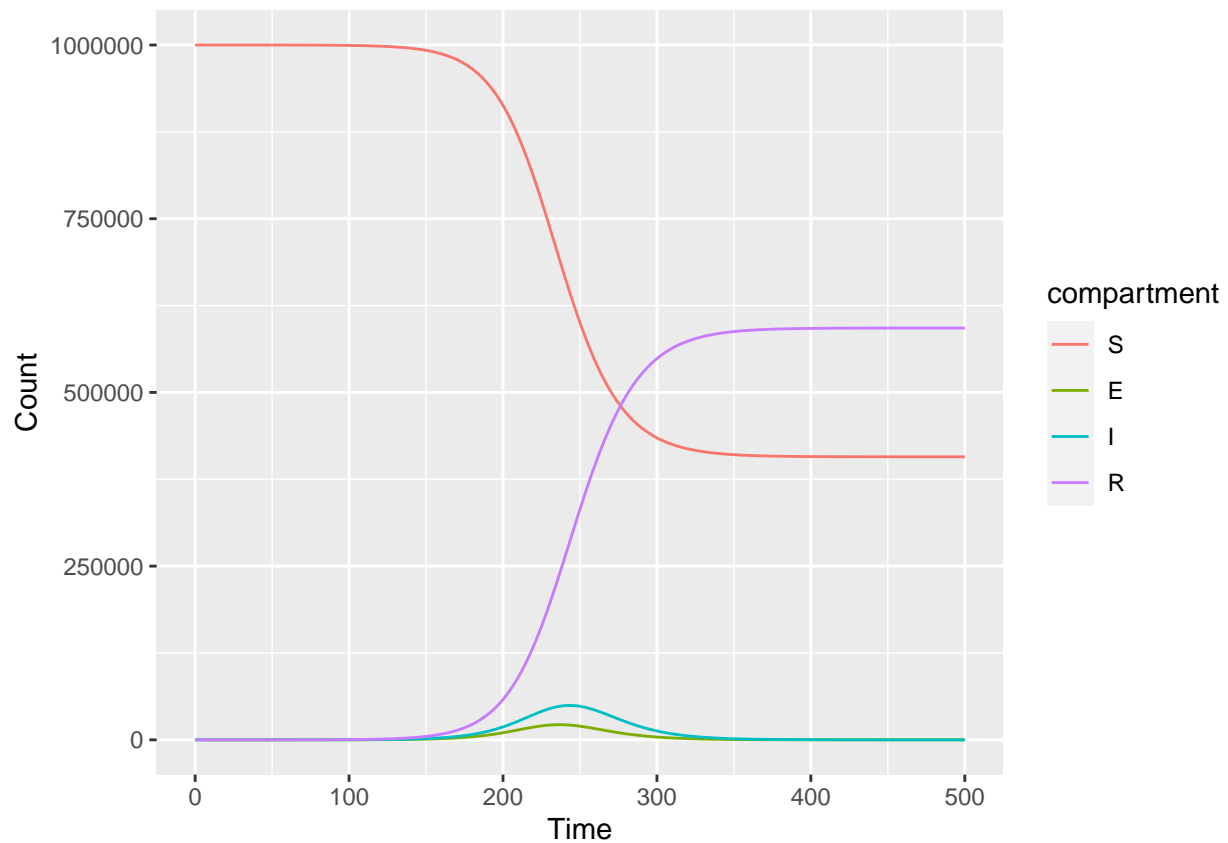
```
state <- c(S = 999999, #population of 1,000,000, 1 person starts of infected
          E = 0,
          I = 1,
          R = 0,
          W = 0
        )

T_end <- 500 #run model for 500 time steps
times <- seq(0, T_end, by = 1) #runs the model for 500 time steps, and computes output at each time step

output_handwash_si_seir <- ode(y = state, times = times, func = handwash_SI_SEIR, parms = c(beta1, beta2))

output_handwash_si_seir %>% show_SEIR_model_results()

## Don't know how to automatically pick scale for object of type deSolve/matrix. Defaulting to continuous
```



```
compute_cumulative_infection_time(output_handwash_si_seir)
```

```
## [1] 4148777
```

Potential additions:

Age stratification Isolation upon symptom onset Death/healthcare burden analysis Assumptions about number of visits out of the house per week Population density (impacts beta)