

# Model 4: SIR Model with seasonality

Katie Montovan, Aaron King, Eli Goldwyn & Julie Blackwood

2025-1-22

This is a modification of the SIR model (model 2) to show how you can include seasonal variation in your model.

We assume that you have already read the chapter and worked through the “Model 1 SI” and “Model 2 SIR” r codes and understand them reasonably well.

## Import data

```
library(readr) #this library is only used to load the csv file.  
simuldata <- read_csv("https://raw.githubusercontent.com/kmontovan/pomp/refs/heads/main/SimulatedSIRdata")
```

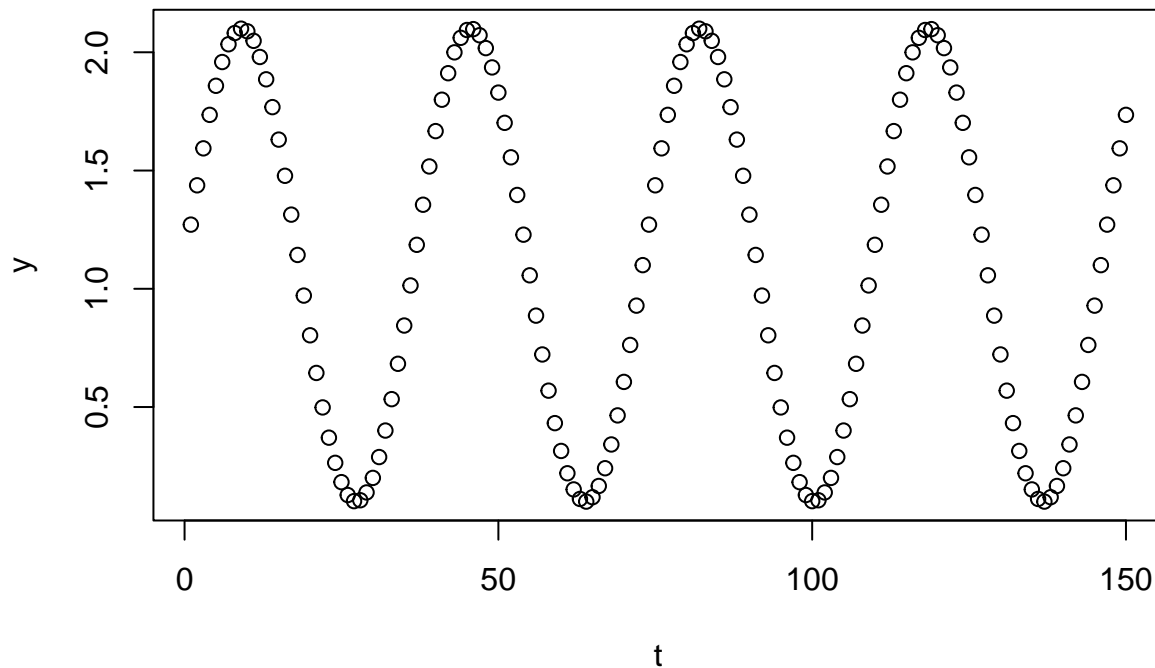
## Model setup

These are the things you may need to adjust for each new model, or, when making a major change to your model. It can be helpful to have this information all in one place as it makes changing the model a bit easier.

We will assume that the rate of spread of the disease varies throughout the year.

Lets start by figuring out a function that varies like we would like. We will multiply this by **Beta**, the rate of spread of the disease. We are going to make an unrealistic function so that we can more clearly see the affects in the simulations

```
t=1:150  
y=(cos(20*pi/365*t-pi/2)+1.1)  
plot(t,y)
```



Now, where we had **Beta** in the model we can put in the function in terms of time. We make a new variable called **BetaV** to hold the time varying Beta value.

```

sir_step <- Csnippet("
  double BetaV = (cos(20*3.14/365*t-3.14/2)+1.1)*Beta;
  double infections = rbinom(S,1-exp(-BetaV*I*dt));
  double recoveries = rbinom(I,1-exp(-gamma*dt));
  double loseimmunity = rbinom(R,1-exp(-r*dt));
  S += -infections + loseimmunity;
  I += infections - recoveries;
  R += recoveries - loseimmunity;
  NewI += infections;
")

sir_rinit <- Csnippet("
  S = nearbyint((1-fI)*N);
  I = nearbyint(fI*N);
  R = 0;
  NewI = 0;
")

si_rmeas <- Csnippet("
  NewCases = rnbinom_mu(k,rho*NewI);
")

```

```

model <-
  pomp(
    data=simuldata,
    times="day",
    t0=0,
    rprocess=euler(
      sir_step,
      delta.t=1/4
    )
  )

```

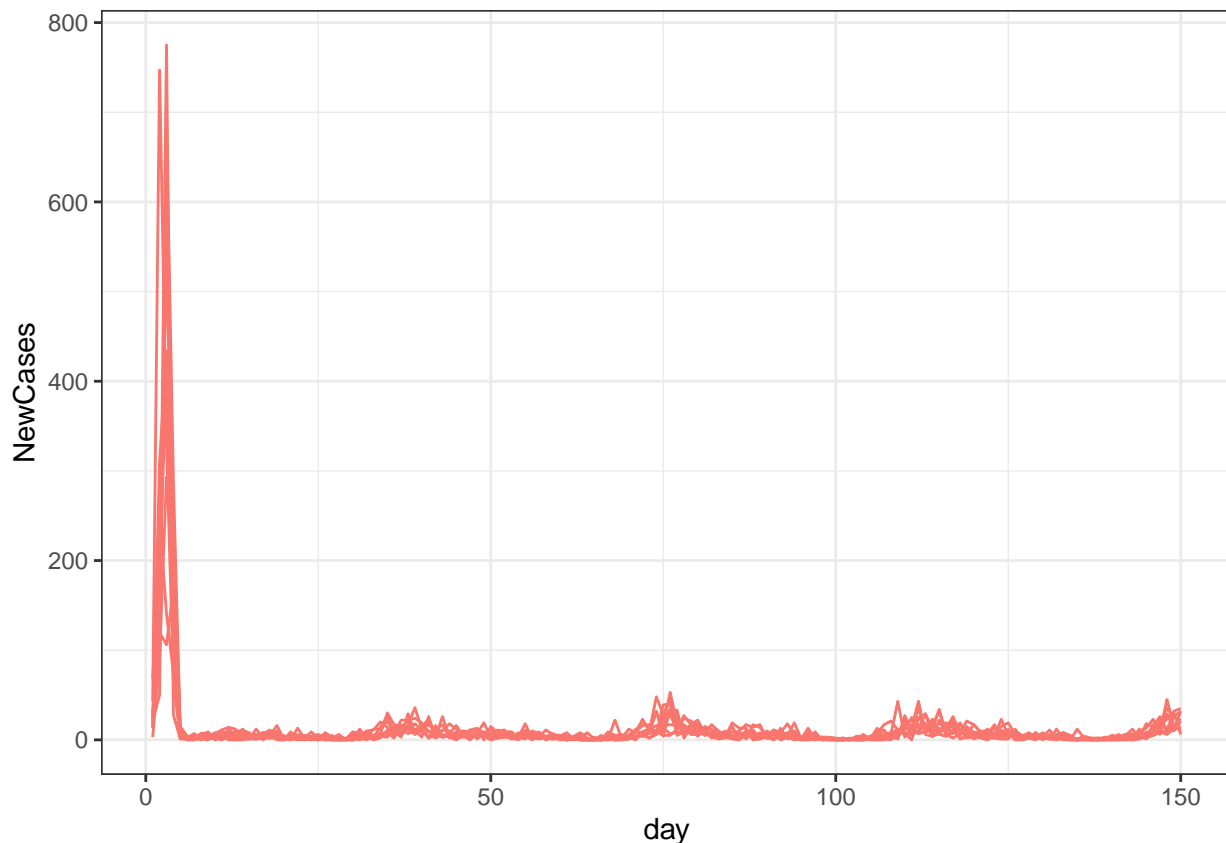
```

    ),
    rinit=sir_rinit,
    rmeasure=si_rmeas,
    accumvars="NewI",
    statenames=c("S", "I", "R", "NewI"),
    params=c(N=1000, #population size
             fI = 0.004, #number of infections at time 0
             Beta=0.0024, # rate of spread of illness - per year
             gamma=0.1, #recovery rate
             r = 0.01, #rate of individuals losing immunity
             rho=0.8, #fraction of new infections that are recorded
             k=5), #changes the variability of reporting accuracy
    paramnames=c("N", "fI", "Beta", "gamma", "r", "rho", "k")
  )
}

model|>
simulate(
  nsim=10,
  format="data.frame",
  include.data=FALSE
)-> sims

sims |>
  ggplot(aes(x=day, y=NewCases, group=.id, color=.id=="data"))+
  geom_line()+
  guides(color="none")+
  theme_bw()

```



You can see the increase in infections that happens at about days 40, 75, 115, and 150.

You also might want to be able to use an if function.

If you want to type in “if  $t > 100$ , then  $\text{BetaV} = 0$ , else  $\text{BetaV} = \text{Beta}$ ” in the Csnippet one way to do this is to type `BetaV= (t >100) ? 0.0 : Beta`. Basically This is saying check whether the thing in the parentheses is true, if it is, return the value after the question mark, if it isn't, return the value after the colon.

In the code, this looks like:

```
sir_step <- Csnippet("
  double BetaV= (t >100) ? 0.0 : Beta;
  double infections = rbinom(S,1-exp(-BetaV*I*dt));
  double recoveries = rbinom(I,1-exp(-gamma*dt));
  double loseimmunity = rbinom(R,1-exp(-r*dt));
  S += -infections + loseimmunity;
  I += infections - recoveries;
  R += recoveries - loseimmunity;
  NewI += infections;
")

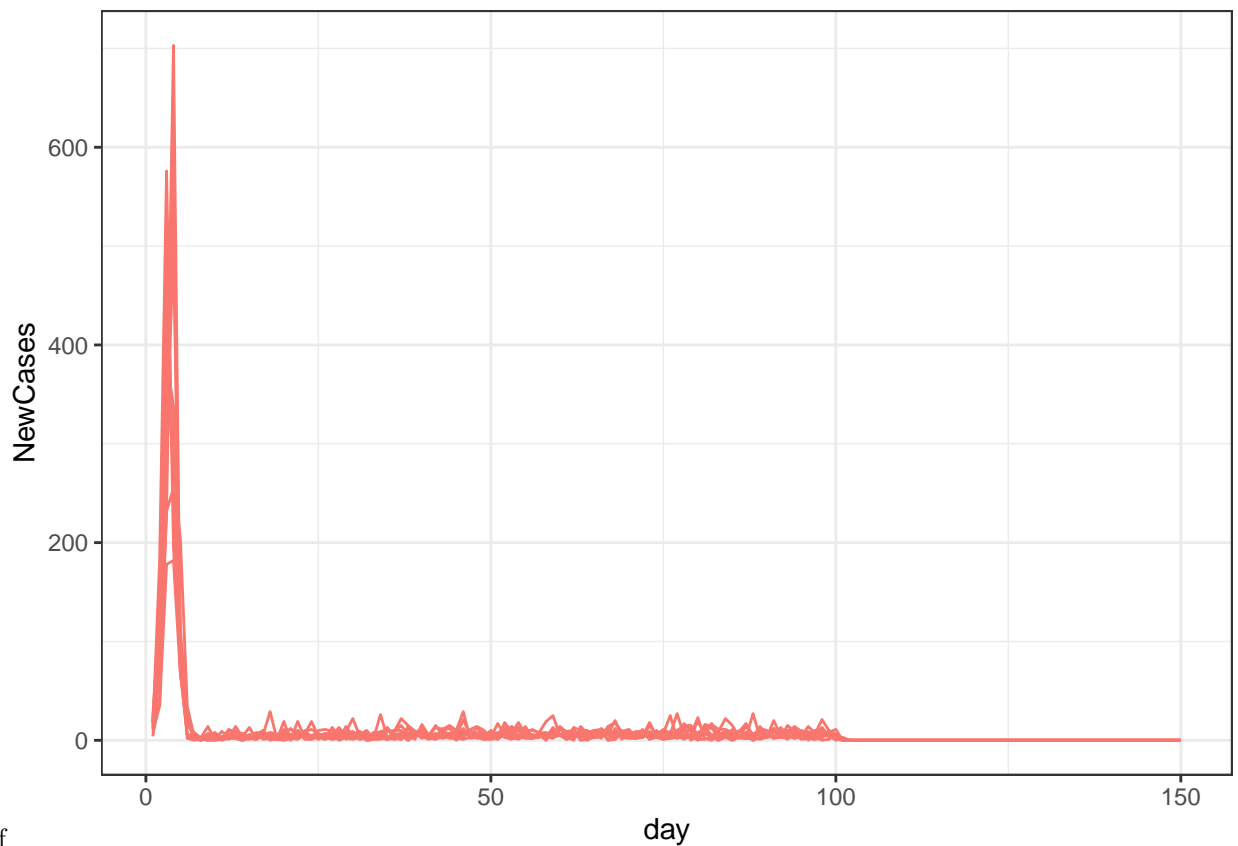
model2 <- pomp(
  data=simuldata, # the timeseries data
  times="day", # timesteps in the simuldata dataset
  t0=0, #the time to start the simulations.
  rprocess=euler(
    sir_step,
    delta.t=1/4
  ),
)
```

```

rinit=sir_rinit,
rmeasure=si_rmeas,
accumvars="NewI",
statenames=c("S","I","R","NewI"),
params=c(N=1000, #population size
         fI = 0.004, #number of infections at time 0
         Beta=0.0024, # rate of spread of illness - per year
         gamma=0.1, #recovery rate
         r = 0.01, #rate of individuals loosing immunity
         rho=0.8, #fraction of new infections that are recorded
         k=5),
paramnames=c("N","fI","Beta","gamma","r","rho","k")
)
model2|>
simulate(
  nsim=10,
  format="data.frame",
  include.data=FALSE
)-> sims

sims |>
ggplot(aes(x=day,y=NewCases,group=.id,color=.id=="data"))+
  geom_line()+
  guides(color="none")+
  theme_bw()

```



model 2-1.pdf

We can see that at day 100, Beta is changed to zero and there are no new cases from then on.

There are almost endless ways you might want to change this, but trying to find the right code to include can be frustrating and sometimes it might be better to add the seasonality through a covariate if what you want is at all complicated.