

## PHE data

The following chunks of code retrieve data about daily COVID-19 cases and deaths up to the date of latest publication. To retrieve data used in code below use `load("pomp_phe_data.RData")`.

```
library("ukcovid19")

query_filters <- c(
  'areaType=nation'
)

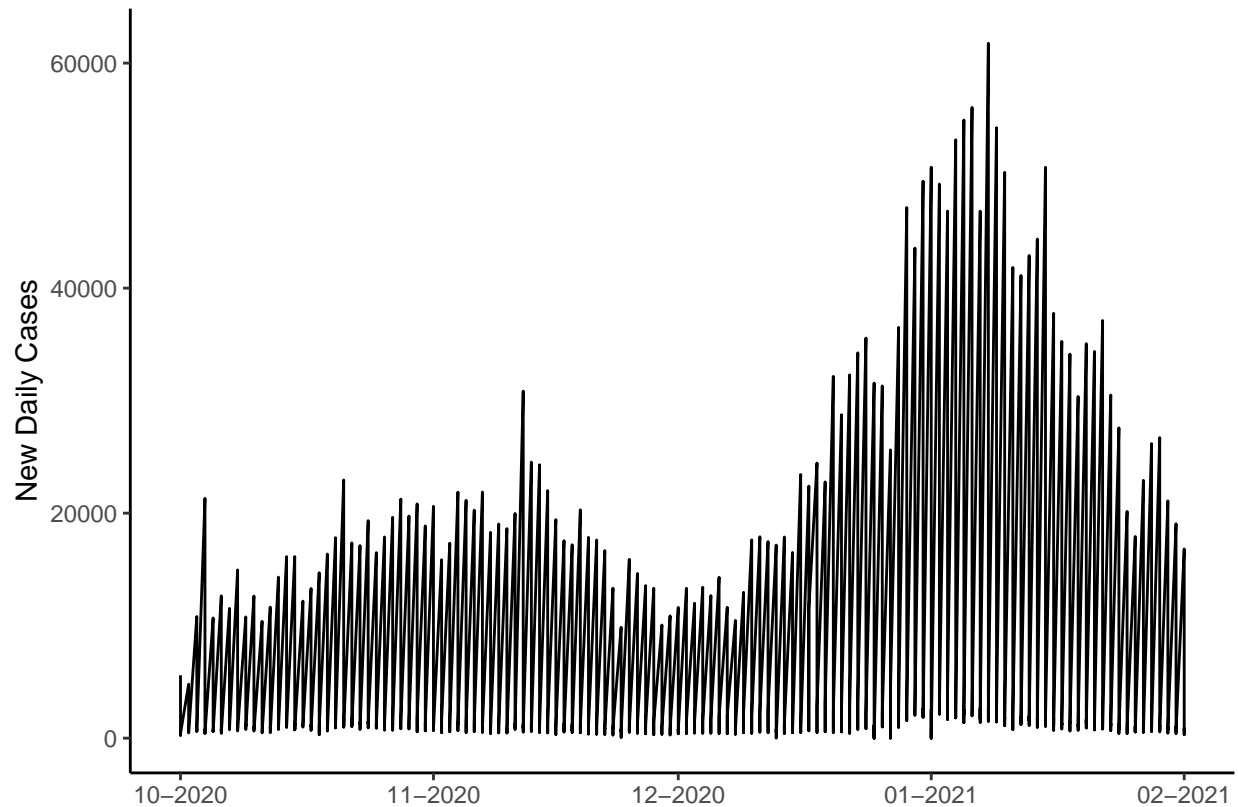
cases_and_deaths = list(
  date = "date",
  areaName = "areaName",
  areaCode = "areaCode",
  newCasesByPublishDate = "newCasesByPublishDate",
  cumCasesByPublishDate = "cumCasesByPublishDate",
  newDeaths28DaysByPublishDate = "newDeaths28DaysByPublishDate",
  cumDeaths28DaysByPublishDate = "cumDeaths28DaysByPublishDate"
)

data <- get_data(
  filters = query_filters,
  structure = cases_and_deaths
)

data$date = as.Date(data$date)
```

```
# Filtering cases between Oct 2020 and Feb 2021
# filter(date >= "2021-05-01")
covid = data %>% dplyr::filter(date <= "2021-02-01") %>%
  dplyr::filter(date >= "2020-10-01")
covid = dplyr::select(covid, newCasesByPublishDate, date)
```

```
ggplot(covid, aes(x = date, y = newCasesByPublishDate)) +
  geom_line() + theme_classic() +
  xlab("") + scale_x_date(date_labels = "%m-%Y") +
  ylab("New Daily Cases")
```



```
# df for pomp
covid2 = data.frame(cases = covid$newCasesByPublishDate,
                    time = seq_along(covid$newCasesByPublishDate))
```

## Tom's waning immunity code

Kindly borrowed as a guide for modifying the example given in “worked\_example.RMD”.

```
### Exponential times
S = S + omega * R - beta * S * I
E = E + beta * S * I - E * sigma
I = I + E * sigma - I * gamma
R = R + I * gamma - omega * R

### Gamma-distributed times
S[(i + 1)] = S[i] + omega * dt * R2[i] * o - beta * dt * S[i] * (I1[i] + I2[i])
E1[(i + 1)] = E1[i] + beta * dt * S[i] * (I1[i] + I2[i]) - E1[i] * sigma * dt * m
E2[(i + 1)] = E2[i] + sigma * dt * m * (E1[i] - E2[i])
E3[(i + 1)] = E3[i] + sigma * dt * m * (E2[i] - E3[i])
E4[(i + 1)] = E4[i] + sigma * dt * m * (E3[i] - E4[i])
I1[(i + 1)] = I1[i] + E4[i] * sigma * dt * m - I1[i] * gamma * dt * n
I2[(i + 1)] = I2[i] + gamma * dt * n * (I1[i] - I2[i])
R1[(i + 1)] = R1[i] + I2[i] * gamma * dt * n - omega * dt * R1[i] * o
R2[(i + 1)] = R2[i] + omega * dt * o * (R1[i] - R2[i])
```

## SEIR model

The simple model below runs, albeit giving odd results.

```
load("working_pomp_code.RData")
```

```
# change during each time step
SEIR_step <- function(S, E, I, R, N, Beta, sigma, gamma, Omega, delta.t, ...) {
  dN_RS = rbinom(1, R, 1 - exp(-Omega*(R/N)*delta.t)) #i'm not sure if this is correct...
  dN_SE = rbinom(1, S, 1 - exp(-Beta*((S*I)/N)*delta.t))
  dN_EI = rbinom(1, E, 1 - exp(-(E/N)*sigma*delta.t))
  dN_IR = rbinom(1, I, 1 - exp(-(I/N)*gamma*delta.t))
  S = S + dN_RS - dN_SE
  E = E + dN_SE - dN_EI
  I = I + dN_EI - dN_IR
  R = R + dN_IR - dN_RS
  return(c(S = S, E = E, I = I, R = R))
}

# iniital conditions
SEIR_step_rinit <- function(N, eta,...) {
  c(S = round(N*eta), E = 0, I = 1, R = round(N*(1 - eta)))
}

# parameters to be used in simulation
paramts = c(Beta = 0.91205, sigma = 0.2222, gamma = 0.3257, Omega = 0.01111,
             p = 0.5, k = 10, N = 66435550, eta = 0.999)
#Beta=30,mu_EI=0.8,mu_IR=1.3,p=0.5,k=10,eta=0.06,N=38000
#paramts = c(Beta = 0.91205, sigma = 0.2222, gamma = 0.3257, omega = 0.01111)

# dmeas process used for parameter estimation (likelohood)
SEIR_dmeas <- function(cases, p, k, log, ...) {
  dnbinom(x = cases, size = k, mu = p, log = log)
}

# rmeas process used for simulation
SEIR_rmeas <- function(p, k, ...) {
  c(cases = rnbinom(n = 1, size = k, mu = (p)))
} # mu = p*H I havw removed H as a state from the model which modifies the meaning of the rmeas proc

# coding the model as a pomp object
covidSEIR <- pomp(data = covid2,
                  times = "time", t0 = 0,
                  rprocess = euler(SEIR_step, delta.t = 0.1),
                  rinit = SEIR_step_rinit,
                  rmeasure = SEIR_rmeas,
                  dmeasure = SEIR_dmeas,
                  paramnames = c("Beta", "sigma", "gamma", "Omega",
                                  "p", "k", "N", "eta"),
                  statenames = c("S", "E", "I", "R")
)

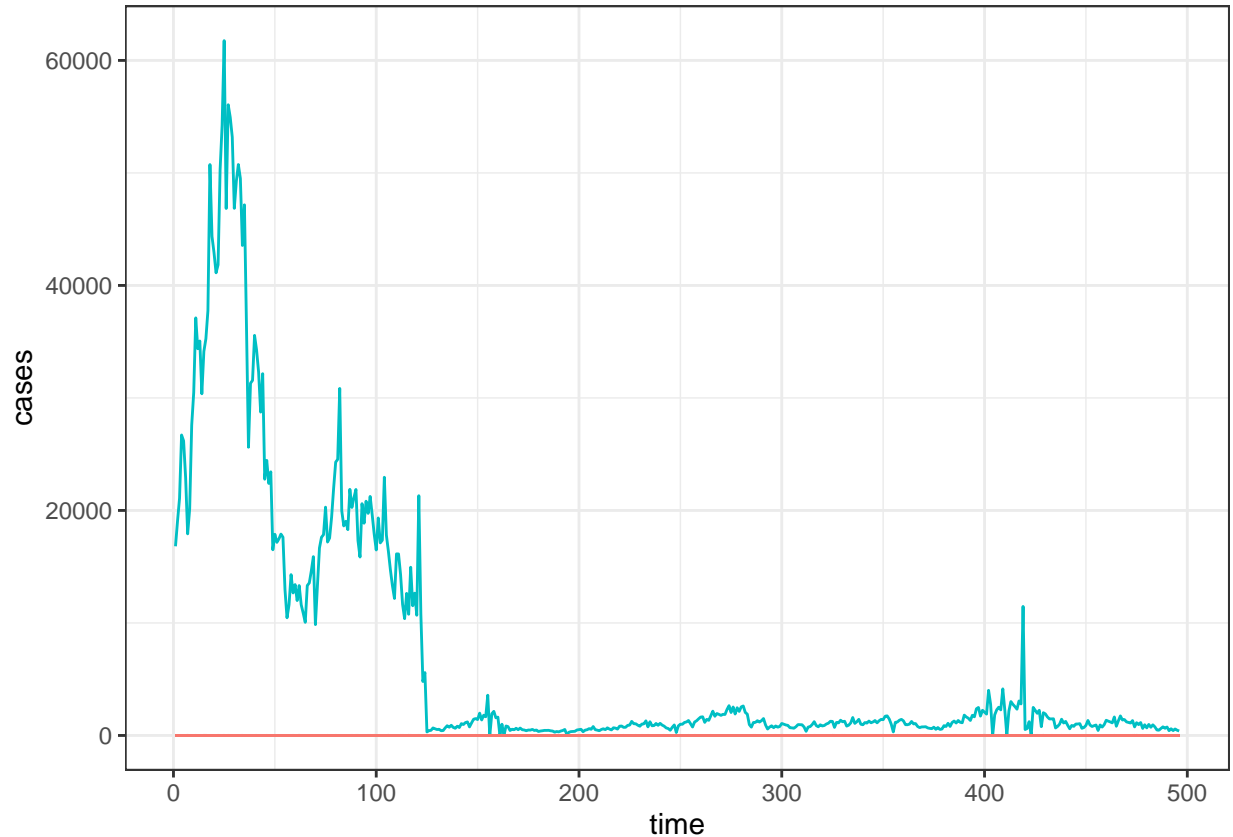
simSEIR = covidSEIR %>%
  simulate(params = c(Beta = 0.91205, sigma = 0.2222,
                     gamma = 0.3257, Omega = 0.01111,
```

```

      p = 0.75, k = 10, N = 66435550, eta = 0.999),
      nsim = 20, format = "data.frame", include.data = T)

ggplot(simSEIR, aes(x = time, y = cases, group = .id, color = .id == "data")) +
  geom_line() + guides(color = "none") + theme_bw()

```



*# the results do not make sense. could be attributed to poor parameter and initial condition choice. covid2*

```
str(simSEIR)
```

```

## 'data.frame':  10416 obs. of  7 variables:
## $ time : num  1 2 3 4 5 6 7 8 9 10 ...
## $ .id  : Ord.factor w/ 21 levels "data"<"1"<"2"<...: 1 1 1 1 1 1 1 1 1 1 ...
## $ cases: num  16815 19054 21089 26709 26182 ...
## $ S    : num  NA NA NA NA NA NA NA NA NA NA ...
## $ E    : num  NA NA NA NA NA NA NA NA NA NA ...
## $ I    : num  NA NA NA NA NA NA NA NA NA NA ...
## $ R    : num  NA NA NA NA NA NA NA NA NA NA ...

```

```
length(covid2$cases)
```

```
## [1] 496
```

```

## interestingly , all the data in covid2$cases is also present in simSEIR$cases
length(which(covid2$cases %in% simSEIR$cases)) == length(covid2$cases)

```

```
## [1] TRUE
```

```
identical(which(covid2$cases %in% simSEIR$cases), seq_along(covid2$cases))
```

```
## [1] TRUE
```

```
## but there are 10k observations in simSEIR$cases,  
## we have simulated SOMETHING albeit once  
length(simSEIR$cases)
```

```
## [1] 10416
```

## Next steps: attempting to adapt gamma-distributed compartment times

The following code does not work but would have been a good next step in adapting the model.

```
SEIR_step <- function(S, E, I, R, N, Beta, sigma, gamma, Omega, delta.t, ...) {  
  SE1 = rbinom(1, S, 1 - exp(-beta*dt*S*(I1 + I2)))  
  E1I = rbinom(1, E1, 1 - exp(-E1*sigma*dt*m))  
  E12 = rbinom(1, E1, 1 - exp(-sigma*dt*m*(E1 - E2)))  
  E23 = rbinom(1, E2, 1 - exp(-sigma*dt*m*(E2 - E3)))  
  E34 = rbinom(1, E3, 1 - exp(-sigma*dt*m*(E3 - E4)))  
  E4I1 = rbinom(1, E4, 1 - exp(-E4*sigma*dt*m))  
  I1R = rbinom(1, I1, 1 - exp(-I1*gamma*dt*n))  
  I12 = rbinom(1, I1, 1 - exp(-gamma*dt*n*(I1 - I2)))  
  I2R1 = rbinom(1, I2, 1 - exp(-I2*gamma*dt*n))  
  R1S = rbinom(1, R1, 1 - exp(-omega*dt*R1*o))  
  R12 = rbinom(1, R1, 1 - exp(-omega*dt*o*(R1 - R2)))  
  R2S = rbinom(1, R2, 1 - exp(-omega*dt*R2*o))  
  S = S + R2S - SE1  
  E1 = E1 + SE1 - E1I1  
  E2 = E2 + E12  
  E3 = E3 + E23  
  E4 = E4 + E34  
  I1 = I1 + E4I1 - I1R  
  I2 = I2 + I12  
  R1 = R1 + I2R1 - R1S  
  R2 = R2 + R12  
  c(S = S, E = E1+E2+E3+E4, I = I1+I2, R = R1+R2)  
}  
  
SEIR_step_rinit <- function(N, eta,...) {  
  c(S = round(N*eta), E = 0, I = 1, R = round(N*(1 - eta)))  
}  
  
#paramts = c(Beta = 0.91205, sigma = 0.2222, gamma = 0.3257, omega = 0.01111)  
paramts = c(Beta = 0.91205, sigma = 0.2222, gamma = 0.3257, Omega = 0.01111,  
  p = 0.5, k = 10, N = 66435550, eta = 0.999)  
#Beta=30,mu_EI=0.8,mu_IR=1.3,p=0.5,k=10,eta=0.06,N=38000),  
  
SEIR_dmeas <- function(cases, p, k, log, ...) {  
  dnbinom(x = cases, size = k, mu = p, log = log)  
}  
  
SEIR_rmeas <- function(p, k, ...) {  
  c(cases = rnbinom(n = 1, size = k, mu = (p)))  
}
```

```

covidSEIR <- pomp(data = covid2,
  times = "time", t0 = 0,
  rprocess = euler(SEIR_step, delta.t = 0.1),
  rinit = SEIR_step_rinit,
  rmeasure = SEIR_rmeas,
  dmeasure = SEIR_dmeas,
  paramnames = c("Beta", "sigma", "gamma", "Omega",
    "p", "k", "N", "eta"),
  statenames = c("S", "E", "I", "R")
)

```