Vacamole

This example is taken from the documentation of the R package [epidemics](https://raw.githubusercontent.com/epiverse-trace/epidemics/main/vignettes/vacamole.Rmd).

library(epidemics)  
library(data.table)  
library(ggplot2)  
library(colorspace)  
library(scales)

## Prepare population and initial conditions

Prepare population and contact data.

# load contact and population data from socialmixr::polymod  
polymod <- socialmixr::polymod  
contact\_data <- socialmixr::contact\_matrix(  
 polymod,  
 countries = "United Kingdom",  
 age.limits = c(0, 20, 65),  
 symmetric = TRUE  
)

Using POLYMOD social contact data. To cite this in a publication, use the 'cite' function

Removing participants that have contacts without age information. To change this behaviour, set the 'missing.contact.age' option

# prepare contact matrix  
contact\_matrix <- t(contact\_data$matrix)  
  
# prepare the demography vector  
demography\_vector <- contact\_data$demography$population  
names(demography\_vector) <- rownames(contact\_matrix)

Prepare initial conditions for each age group. The Vacamole model has 11 compartments and therefore requires a matrix with 11 columns.

# initial conditions  
initial\_i <- 1e-6  
  
# // 0| 1| 2|3| 4|5| 6|7| 8|9|10  
# // S|V1|V2|E|EV|I|IV|H|HV|D|R  
  
# make initial conditions - order is important  
initial\_conditions <- c(  
 S = 1 - initial\_i,  
 V1 = 0, V2 = 0,  
 E = 0, EV = 0,  
 I = initial\_i, IV = 0,  
 H = 0, HV = 0, D = 0, R = 0  
)  
initial\_conditions <- rbind(  
 initial\_conditions,  
 initial\_conditions,  
 initial\_conditions  
)  
  
# assign rownames for clarity  
rownames(initial\_conditions) <- rownames(contact\_matrix)

Prepare the time in days over which to model the epidemic, with the outbreak beginning at day zero.

epidemic\_days <- 300

Prepare a population as a population class object.

uk\_population <- population(  
 name = "UK",  
 contact\_matrix = contact\_matrix,  
 demography\_vector = demography\_vector,  
 initial\_conditions = initial\_conditions  
)

## Prepare pathogen parameters

# make infection class for Vacamole model  
# note extra arguments passed as ...  
infect <- infection(  
 name = "covid", r0 = 3, infectious\_period = 10,  
 preinfectious\_period = 5,  
 eta = 1 / 1000, omega = 1 / 1000,  
 susc\_reduction\_vax = 0.5,  
 hosp\_reduction\_vax = 0.5,  
 mort\_reduction\_vax = 0.5  
)

## Prepare a two dose vaccination campaign

We prepare a two-dose vaccination campaign by concatenating two single dose vaccination regimes, i.e., applying the function c() to two vaccination objects.

# prepare a two dose vaccination regime for a single age group  
# prepare the first dose  
dose\_1 <- vaccination(  
 name = "two-dose vaccination", # name given to first dose  
 nu = matrix(1e-2, nrow = 3),  
 time\_begin = matrix(30, nrow = 3),  
 time\_end = matrix(epidemic\_days, nrow = 3)  
)  
  
# prepare the second dose with a 30 day interval in start date  
dose\_2 <- vaccination(  
 name = "two-dose vaccination", # name given to first dose  
 nu = matrix(1e-2, nrow = 3),  
 time\_begin = matrix(60, nrow = 3),  
 time\_end = matrix(epidemic\_days, nrow = 3)  
)  
  
# use `c()` to combine the two doses  
double\_vaccination <- c(dose\_1, dose\_2)  
  
# print to visualise  
double\_vaccination

<vaccination>  
Vaccination name: "two-dose vaccination"  
  
Time begin:  
 dose\_1 dose\_2  
[1,] 30 60  
[2,] 30 60  
[3,] 30 60  
  
Time end:  
 dose\_1 dose\_2  
[1,] 300 300  
[2,] 300 300  
[3,] 300 300  
  
Vaccination rate:  
 dose\_1 dose\_2  
[1,] 0.01 0.01  
[2,] 0.01 0.01  
[3,] 0.01 0.01

We prepare a dummy vaccination regime of no vaccination against which to compare the two-dose vaccination campaign. This is done by using the no\_vaccination() function and specifying the number of doses (here, 2 doses).

# prepare a null vaccination regime with two doses  
# use the convenience function `no\_vaccination()`  
no\_vaccination <- no\_vaccination(population = uk\_population, doses = 2)

## Model epidemic using Vacamole

First, we run the model with no vaccination.

data <- epidemic(  
 model\_name = "vacamole",  
 population = uk\_population,  
 infection = infect,  
 vaccination = no\_vaccination, # note custom no\_vaccination object  
 time\_end = epidemic\_days  
)

Next we run the model with a two dose vaccination regime.

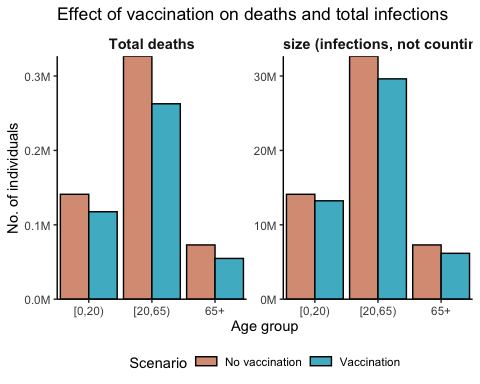
data\_vaccination <- epidemic(  
 model\_name = "vacamole",  
 population = uk\_population,  
 infection = infect,  
 vaccination = double\_vaccination, # note custom object for two doses  
 time\_end = epidemic\_days  
)

## Visualise model outcomes

First, we calculate the total number of infections resulting in recoveries and deaths over the course of the simulation; this is the epidemic’s final size.

# collect data from the two scenarios  
data\_scenarios <- list(data, data\_vaccination)  
  
# get deaths and recoveries from infection  
data\_scenarios <- Map(  
 data\_scenarios, c("no\_vax", "vax"),  
 f = function(df, sc) {  
 df\_ <- unique(df[, "demography\_group"], by = "demography\_group")  
 # get total deaths per group  
 df\_$total\_deaths <- df[time == max(time) & compartment == "dead", ]$value  
  
 # get total recoveries per group using helper function `epidemic\_size()`  
 df\_$total\_recovered <- epidemic\_size(df, deaths = FALSE) # do not count dead  
  
 # add scenario information  
 df\_$scenario <- sc  
  
 # return data  
 df\_  
 }  
)  
  
# collect data  
data\_scenarios <- rbindlist(data\_scenarios)  
  
# transform to long format  
data\_scenarios <- melt(  
 data\_scenarios,  
 id.vars = c("demography\_group", "scenario")  
)

ggplot(data\_scenarios) +  
 geom\_col(  
 aes(demography\_group, value, fill = scenario),  
 position = "dodge",  
 colour = "black"  
 ) +  
 facet\_wrap(  
 ~variable,  
 scales = "free\_y",  
 labeller = labeller(  
 variable = c(  
 "total\_deaths" = "Total deaths",  
 "total\_recovered" = "Epidemic size (infections, not counting deaths)"  
 )  
 )  
 ) +  
 scale\_fill\_discrete\_qualitative(  
 palette = "Dynamic",  
 labels = c("No vaccination", "Vaccination"),  
 name = "Scenario",  
 na.value = "lightgrey"  
 ) +  
 scale\_y\_continuous(  
 labels = label\_comma(  
 scale = 1e-6, suffix = "M"  
 )  
 ) +  
 theme\_classic() +  
 theme(  
 legend.position = "bottom",  
 legend.key.height = unit(2, "mm"),  
 strip.background = element\_blank(),  
 strip.text = element\_text(  
 face = "bold",  
 size = 11  
 )  
 ) +  
 expand\_limits(  
 x = c(0.5, length(unique(data\_scenarios$demography\_group)) + 0.5)  
 ) +  
 coord\_cartesian(  
 expand = FALSE  
 ) +  
 labs(  
 x = "Age group",  
 y = "No. of individuals",  
 title = "Effect of vaccination on deaths and total infections"  
 )



Finally, we can compare the peak of hospital bed occupancy in each scenario — this can be a rough indication of how much hospital capacity would be required if a pandemic of these characteristics were to occur, as well as another way to examine the effect of vaccination in reducing this requirement.

# collect data from the two scenarios  
data\_scenarios <- list(data, data\_vaccination)  
  
peak\_hospital\_occupancy <- vapply(data\_scenarios, function(df) {  
 # get highest hospital occupancy  
 # first get total hospitalisations among vaccinated and un- or part-vacc.  
 df <- dcast(  
 df[compartment %like% "hospitalised", ],  
 time + demography\_group ~ compartment,  
 value.var = "value"  
 )  
 df[, total\_hosp := hospitalised + hospitalised\_vaccinated]  
  
 # sum all age groups  
 df <- df[, list(total\_hosp = sum(total\_hosp)), by = "time"]  
  
 df[total\_hosp == max(total\_hosp), ]$total\_hosp  
}, FUN.VALUE = numeric(1))  
  
# set names for comprehensibility  
names(peak\_hospital\_occupancy) <- c("No vaccination", "Vaccination")  
  
# show peak hospital occupancy in a readable format  
format(peak\_hospital\_occupancy, big.mark = ",", digits = 1)

No vaccination Vaccination   
 "95,063" "69,827"

This example demonstrates that implementing vaccination can substantially reduce peak hospital occupancy (by about 27%) compared to a scenario in which no vaccines are deployed.