Practical 2: Social Structure in SEIR Models

Consider the SEIR model from section 6.2 of the notes. In section 6.2 we considered the impact of individual variability in the transmission rate parameter β on the dynamics of an epidemic. The simulations suggest that higher variability leads to smaller epidemics. Another refinement of the basic SEIR model would be to represent various aspects of social structure in the models, rather than assuming that people bump into (and infect) each other completely at random like molecules of gas.

A simple model of social structure would represent the fact that people live in households, where household members are more likely to infect each other than people outside the household. It would also represent the fact that people tend to have relatively fixed groups of people with whom they interact regularly, in addition to more general mixing with other people of the sort represented in the original model.

To be mathematically concrete, there are n people each in state S, E, I or R. If in state I they have a daily probability of δ of moving to state R. If in state E they have a daily probability of γ of moving to state I. The interesting part of the model is the transition from state S to state E as a result of infection by someone in state I. This depends on two characteristics of each person: who they share a household with, and who is in their network of regular contacts. These are modelled as follows:

1. The household characteristic is easy, we simply divide the n people into households of sizes between 1 and $h_{\max}$. Assume a uniform distribution of household sizes.

2. The contact network model is more complicated. Suppose each person has a 'sociability' parameter with β_i being the parameter for the ith person. Then the network is created randomly by assigning a link between persons i and j with probability
$$\frac{n_c \beta_i \beta_j}{\bar{\beta}^2 (n-1)}$$
where n_c is the average number of contacts per person and $\bar{\beta}$ the mean sociability parameter. People in the same household are excluded from such contacts.

Given this structure, there are several ways for a person, i, in state I to infect a person j in state S.

1. If j is a household member there is a daily probability α_h of i infecting j.

2. If j is in i's regular network of contacts then there is a daily probability α_c of i infecting j.

3. Irrespective of household or regular network relations there is a daily probability
$$\frac{\alpha_r n_c \beta_i \beta_j}{\bar{\beta}^2 (n-1)}$$
of i infecting j (random mixing).

The use of the same n_c in the random mixing and regular network expressions is a simplification.

Your task is to implement the model and provide illustration of its use to investigate the role of household and network structure on epidemic dynamics. Your code should work with any population sizes, n, up to at least 10000, but you might want to test and develop with $n = 1000$.

1. Write code to produce an n vector, `h`, of integers indicating which household each person belongs to. Households sizes should be uniformly distributed between 1 and $h_{\max}$. For example if h_1, h_{56} and h_{907} are all 13, and these are the only 13s in h, then people 1, 56 and 907 all live in the same 3 person household. `h` can be created with one line of code by careful use of `rep` and `sample`. Use $h_{\max} = 5$ by default.

2. Write a function `get.net(beta,nc=15)` where `beta` is the n vector of β_i value for each person, and `nc` is n_c. The function should return a list, the ith element of which is a vector of the indices of the regular (non-household) contacts of person i. You need to be careful to implement the model properly so that you do not *create* any links twice (links need to be *recorded* twice of course).

3. Write a function

```
nseir(beta,h,alink,alpha=c(.1,.01,.01),delta=.2,gamma=.4,nc=15, nt = 100,pinf = .005)
```

where `beta` and `h` are as above, `alink` is a list defining the regular contacts of each person as returned by `get.net`, `nt` is the number of days to simulate and `pinf` is the proportion of the initial population to randomly start in the I state. The function should implement the model given above and return a list with elements `S`, `E`, `I`, `R` and `t` giving the total population in each class each day and the day, respectively. Note that while looping through individuals in the I state is probably inevitable, the code can still be made to run in a few seconds for $n = 10000$, especially with careful use of expressions like `x[ind1][ind2] <- y` in places (assignment to a subvector of a subvector).

4. Write a function to nicely plot the dynamics of the simulated population states as returned by `nseir` (a better version of the sort of plots in 6.2 of the notes).

5. Setting `beta` to a vector of $U(0, 1)$ random variables, use the model to compare 4 scenarios and plot them next to each other (suitably labelled). First the full model with default parameters. Next look at what happens when you remove the household and regular network structure, while keeping the average initial number of infectious contacts per day the same for each person, by setting $\alpha_h = \alpha_c = 0$ and $\alpha_r = 0.04$. Then consider the full model, but with the `beta` vector set to simply contain the average of the previous `beta` vector for every element. Finally combine the previous two scenarios (constant beta, random mixing). Comment on the apparent effect of the household and network structure, relative to random mixing.

As usual, comment your code so that it can be fully understood without this sheet and without knowing what it is for in advance. Lay the code out clearly and ensure that the comments concisely help to convey how it works and what it is intended to do. Please put all function definitions first, with the code required to use the functions to run the model and answer the final part at the end.

One piece of work - the text file containing your commented R code - is to be submitted for each group of 3 on Learn by 12:00 (noon) 17th October 2025. Include a link to your github repo in a comment, and make your github repo public by the end of the submission day, but not before submission. Start your submitted file with a short statement of roughly the proportion contributed by each team member. No extensions are available on this course, because of the frequency with which work has to be submitted. So late work will automatically attract a penalty (of 100% after work has been marked and returned). Technology failures will not be accepted as a reason for lateness (unless it is provably the case that Learn was unavailable for an extended period), so aim to submit ahead of time.

Full marks will be obtained for code that:

1. does what it is supposed to do, and has been coded in R approximately as indicated using base R (that is marks will be lost for simply finding a package or online code that simplifies the task for you).

2. is carefully commented, so that someone reviewing the code can easily tell exactly what it is for, what it is doing and how it is doing it without having read this sheet, or knowing anything else about the code. Note that *easily tell* implies that the comments must also be as clear and *concise* as possible. You should assume that the reader knows basic R, but not that they know exactly what every function in R does.

3. is well structured and laid out, so that the code itself, and its underlying logic, are easy to follow.

4. is reasonably efficient. As a rough guide the whole code should take at most a few 10s of seconds to run - much longer than that and something is probably wrong.

5. was prepared collaboratively using git and github in a group of 3.

6. contains no evidence of having been copied, in whole or in part, from anyone else (AI counts as anyone else for this purpose), other students on this course, students at other universities (there are now tools to help detect this, including internationally), online sources, ChatGPT etc.

7. includes a comment stating team member contributions.

Individual marks may be adjusted within groups if contributions are widely different.