ICL Coursera

I'm very much not an expert, and these resources only scratch the surface of infectious disease modelling. Nevertheless, I think the resources are a handy primer to learn the basic structure of a popular kind of model, and understand the premise of major extensions and complications that are made by modellers in the real world. My notes may be of some use, though I highly recommend reading the original series of posts. Similarly, my Anki cards might be helpful, though I have found much more value when I make my own.

I think a reader could cover these basics quickly. The main barrier to understanding would be if you were intimidated by differential equations. But you shouldn't be intimidated - to cover the basics you only need to be able to read off what differential equations are saying, with some grasp of the meaning. If you know the notation, all the pieces of each equation make intuitive sense and follow from the model. 3Blue1Brown is a helpful conceptual guide if you're rusty.

I know that if I had come across some of the later equations in a random paper, my eyes would have probably glazed over and I'd mentally skip over it. Developing my understanding from the ground up has helped me to not be intimated, and instead feel comfortable working out why each piece of the equation is present, and what it does.

Being able to code in R would be a boon. I imagine it would greatly speed up the practical exercises. However, I think a reader could develop a decent understanding of the basics without the practical coding exercises.

ICL-1 Developing the SIR Model

Week 1

Infected cohorts and simple flows

With two compartments of infected and recovered, and everyone starting off infected:
Assuming a constant hazard rate γ, you get
$di/dt = - \gamma i$
and
$dR/Dt = \gamma i$

The solutions to these ODEs are:
$i = i_0 e^{-\gamma t}$
$R = = i_0(1-e^{-\gamma t})$

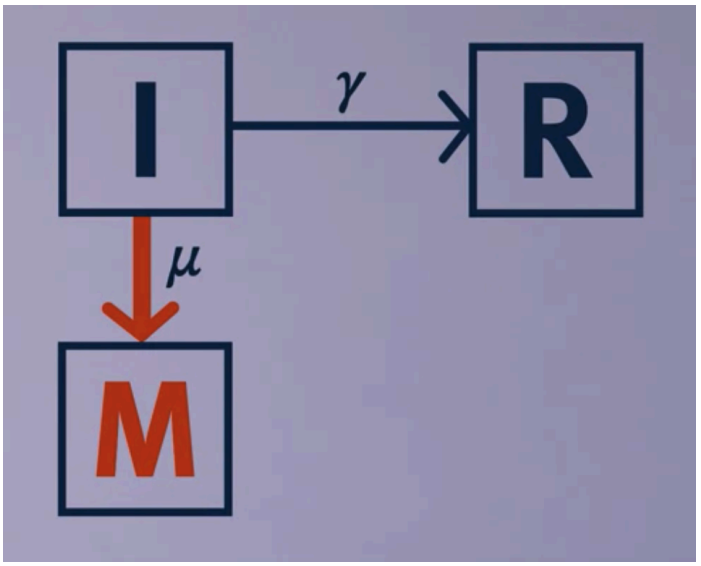
Note that:

- rates like γ are in inverse time, e.g. per-day or per-year
- the inverse of a rate gives the average duration

Competing hazards

Competing hazards = hazards acting on the same compartment

M = no. people died, with rate of transition μ



Case fatality rate = μ/μ+γ
Survival rate = γ/μ+γ

Generically, the amount who end up in compartment A out of [A,B,C] = a/(a+b+c)

di/dt = -(μ+γ)i
dM/dt = μi
dR/dt = γi

initial conditions for our examples:
i = 10^6
M=R=0
at time t = 0

The Force of Infection

SIR models assumptions

Compartmental models - all individuals in one compartment are identical - assumed as the ‘average representative member’. This is the assumption of **homogeneity**.

Force of infection is lambda - the rate of transition from susceptible to infected.

But force of infection will not be constant - with more infected people around, the chance of getting infected becomes higher.
To model a dynamic force of infection, we define:
c = average number of contacts a susceptible makes per day
I/N = the proportion of contacts that are made with an infectious person
p = probability of infection per infectious contact
and lambda = p*c*(I/N)

Key assumptions of the SIR model: homogeneity, and well-mixed

Week 2

Transmission drivers of an epidemic

Basic discussion of R-nought.

Can we control transmission?

Important to know about and model latent and symptomatic phases, as they can skew the real world away from a simple SIR model fit.

Reading: Fraser et al. 2004 - 'Factors that make an infectious disease outbreak controllable'

Uses a quantitative model to assess two interventions: isolation of sx individuals, tracing and quarantining of contacts.
The key pathogen properties for these interventions are the proportion of transmission that is asymptomatic or pre-symptomatic, in addition to transmissibility itself.
Assesses the examples of SARS-1, HIV, smallpox, pandemic influenza.

They suggest theta as a useful statistic - the proportion of transmission occurring before symptom onset or asymptomatically. If theta is less than 1/R-0, then isolation alone could control an outbreak. If theta is above 1/R-0, contact tracing is needed.
But theta can be increased by delays between in implementation isolation. The efficacy of tracing and isolation can determine the overall outcomes of control efforts.
R-0 and theta may be critical early parameters for assessing an emerging pathogen, as they can be easily collected from the initial few hundred cases.
This paper was in 2004 - has theta caught on as a useful measure?

They note that SARS-1 is the easiest infection to control among the cases they consider because of low R-0 and theta. (**How much worse was SARS-CoV-2 in these metrics?**)

Week 3 - Combining modelling and insights

SIR Dynamics

Okay so every person infects 1 person every 2 days and is infectious for 4 days, so what are the values of β and γ?
Beta is 0.5 (infection rate, right?)
Gamma is just 1/D where D is the number of days spent infectious, so gamma is 0.25

For R-0 >1, and beta = 0.1, Tau (the infectious period in days) must be >10, therefore the recovery rate must be less than 0.1

R-0 and R-eff

Yep - R-0 is beta/gamma

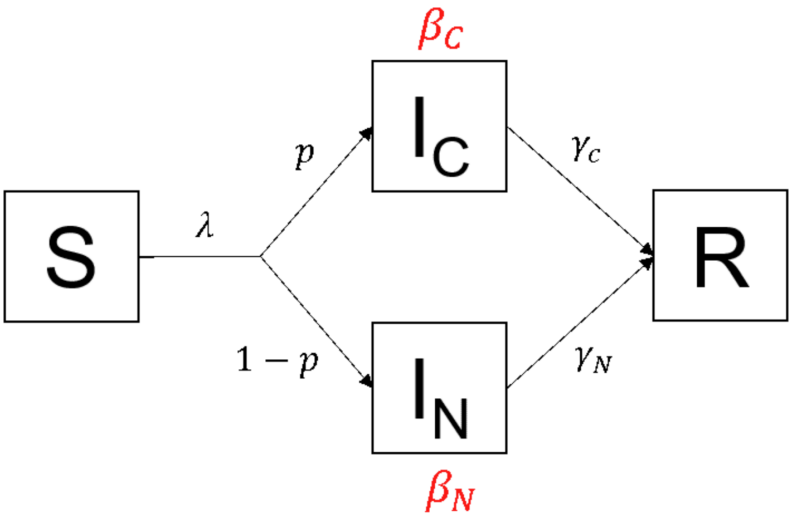
I compartment = prevalence
But lots of public health data comes as incidence

Think of bathtub with water flowing in through a tap, and out through a plug. The prevalence is the level of water in the tub, and incidence is the inflow through the tap.

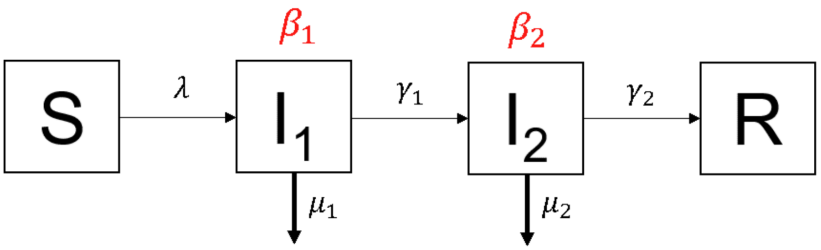
Prevalence = the number of infected cases at a given point in time
Incidence = number of new cases in a govern interval of time (e.g. per week).

You calculate the incidence with the integral of the infection hazard rate multiplied by the number in the susceptible compartment over the time period you care about.

R-0 complexities



So, r-0 is beta over gamma. We basically just wanted a weighed average here, so I would guess it's = p(B-c/Yc) + (1-p)(B-n/Y-n)



So the difference in this example is we're using mortality rate mu, and the infectious compartments are sequential, not parallel.
My guess is = B-1/(Y1+u1) + Y1*B-2/(Y2+u2)

Actual notes and calculation here.

Yep, in the parallel case it's just a average of the betas and gammas weighted by proportion:

$$R_0 = p\frac{\beta_C}{\gamma_C} + (1-p)\frac{\beta_N}{\gamma_N}$$

For the sequential case, I was nearly there, but needed to think more about the proportion proceeding to the second infectious state. It's not weighted by Y-1, but by Y-1/(Y-1 + u-1), following the formula for competing hazards:

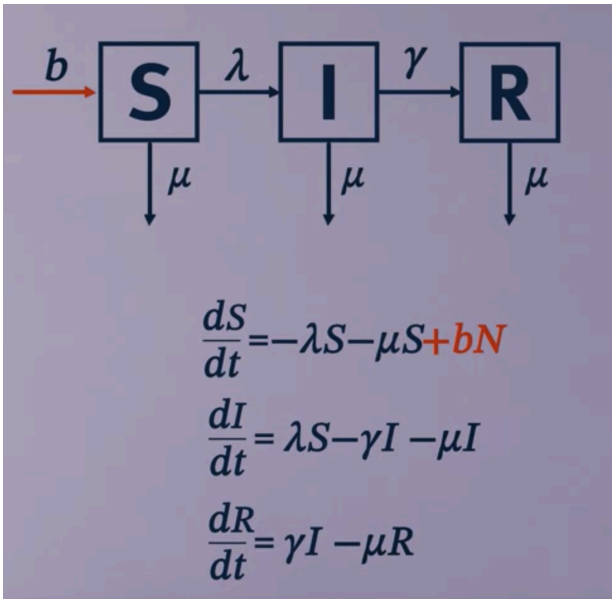
$$R_0 = \frac{\beta_1}{\gamma_1 + \mu_1} + \frac{\gamma_1}{\gamma_1 + \mu_1} \times \frac{\beta_2}{\gamma_2 + \mu_2}$$

Week 4

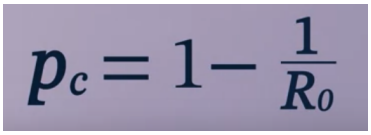
Population turnover and vaccination

To model population turnover, we first add a constant hazard of mortality from each of the SIR compartments. For some special cases like Ebola, we include the dead in a 'Mortality' compartment, as they can still play a role in transmission. But we'll ignore such cases. Also, the mortality hazard here is not disease-induced mortality, just mortality from all other causes.

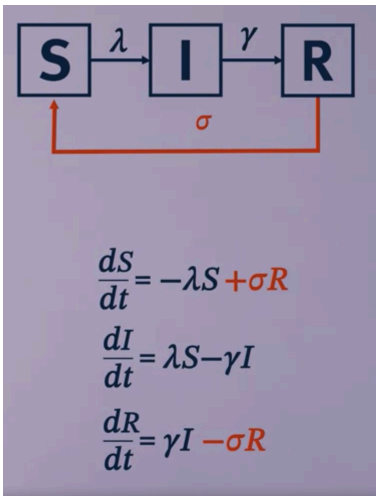
We can model births just by adding a term for influx to the S compartment. If you wanted to include maternal transmission of immunity, or neonatal vaccination, you could add them to the R compartment too.



Critical vaccination threshold:



We model waning immunity with an additional flow with constant hazard sigma from R to S:



ICL-2 Interventions and Calibration

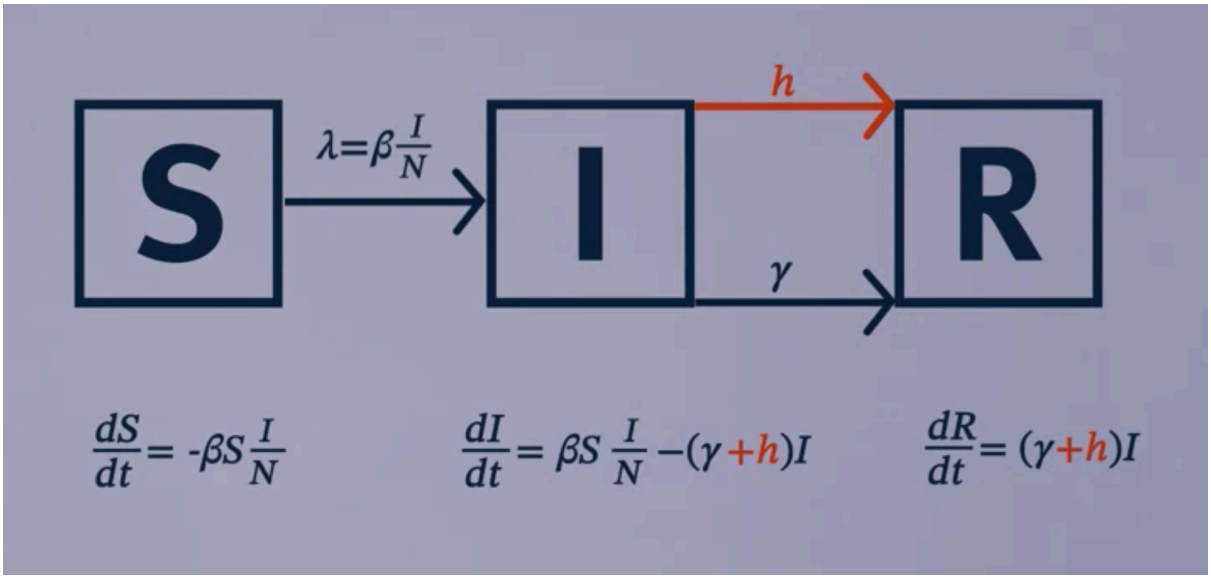
Reminder: glossary here

Week 1

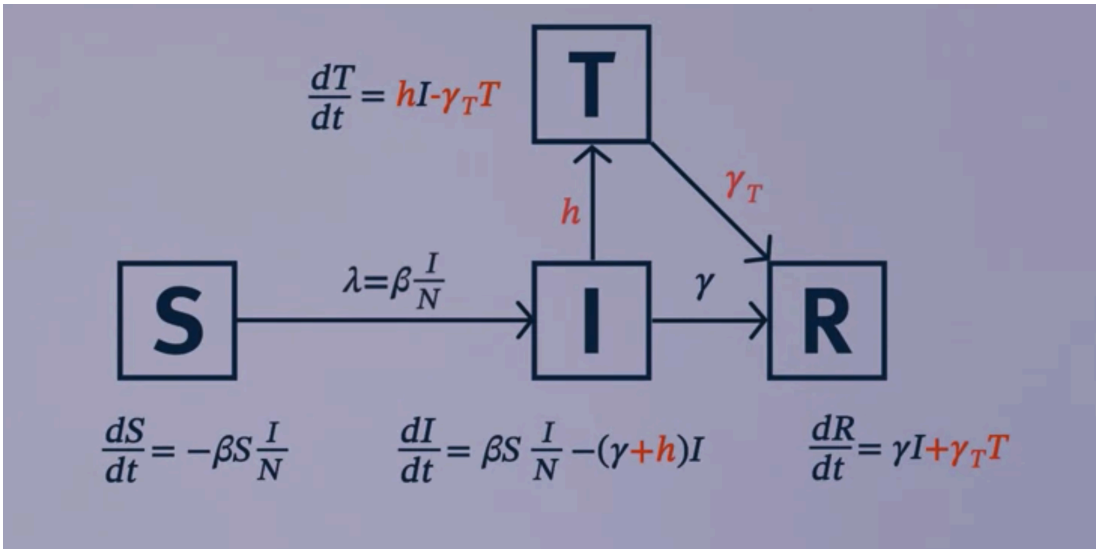
Further reading: “Modeling infectious disease dynamics in the complex landscape of global health” https://pubmed.ncbi.nlm.nih.gov/25766240/

Modelling curative treatment

How is treatment modelled?
For a perfectly curative treatment, you can model it a s increasing the rate that people go from I to R. You can keep the natural rate of recovery gamma, but also add a treatment parameter h, as below:



For the more realistic case of imperfect treatments, you can add a compartment T. Influx to the compartment would still be I*h, while outflux would be the rate that treated individuals go to the recovered compartment - call it γ_T, as below:



Modelling vaccination: first steps

A previous simplistic way of modelling vaccination was just to move a fixed proportion of people from S to R directly, in advance of the epidemic (modelling a situation where a % of the population are immunised prior to an epidemic starting).

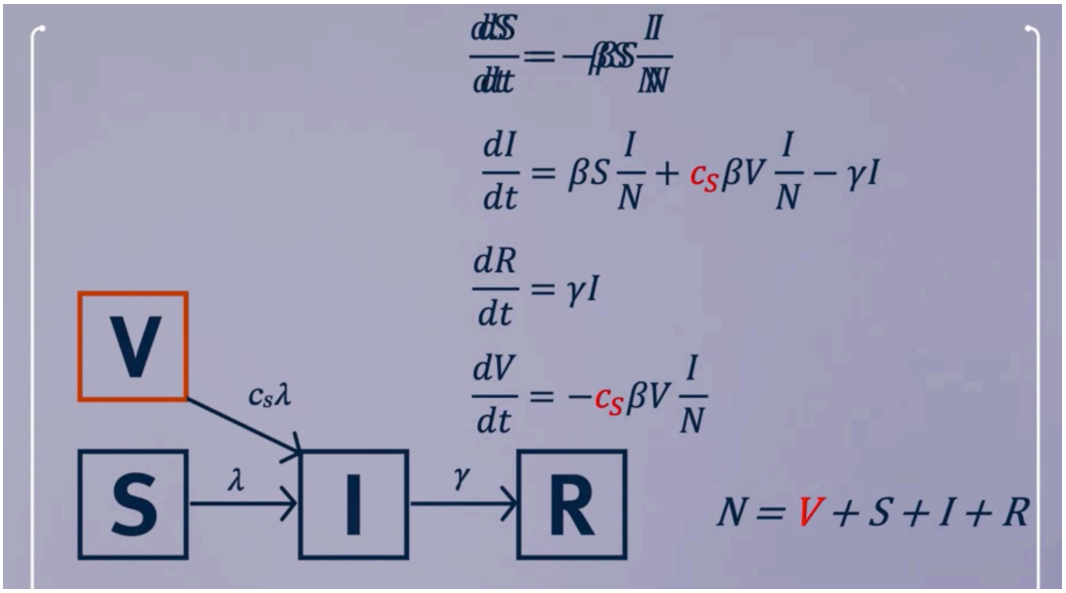
We need a more nuanced model for more realistic situations, such as imperfect vaccines, vaccines that reduce symptom severity, etc. We build this with an additional compartment, called V.

Leaky vaccines

All-or-nothing vaccines vs. leaky vaccines
E.g. All or nothing: 60% effective = 60% of people have perfect protection, 40% have no protection.
Leaky = everyone has 60% protection

All or nothing vaccines are easy to model, as you can just adjust the size of the V compartment to adjust for the effectiveness (e.g. a 50% effective vaccine at 80% coverage is equivalent to a perfect vaccine with 40% coverage).

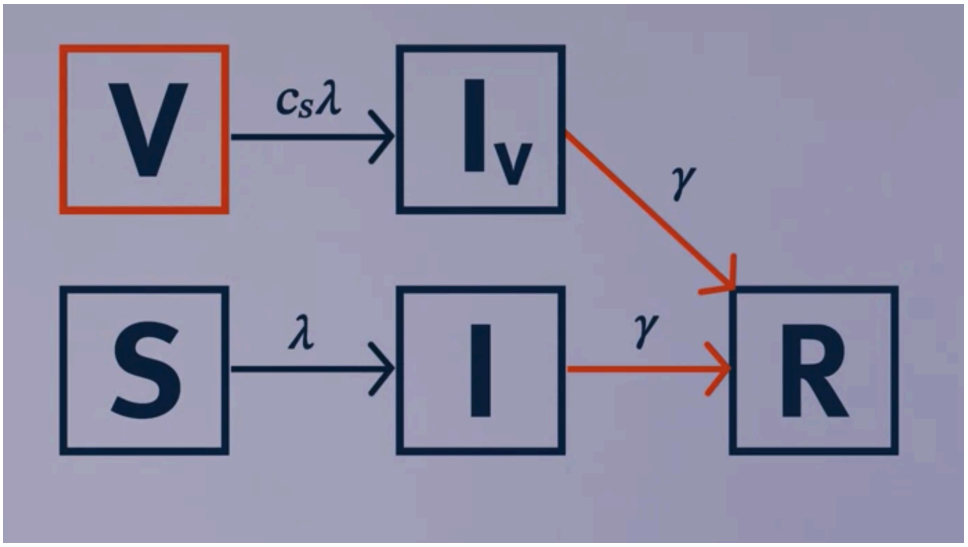
You can model leaky vaccines as a compartment V with a different (lower) force of infection, $c_S\lambda$, as below:



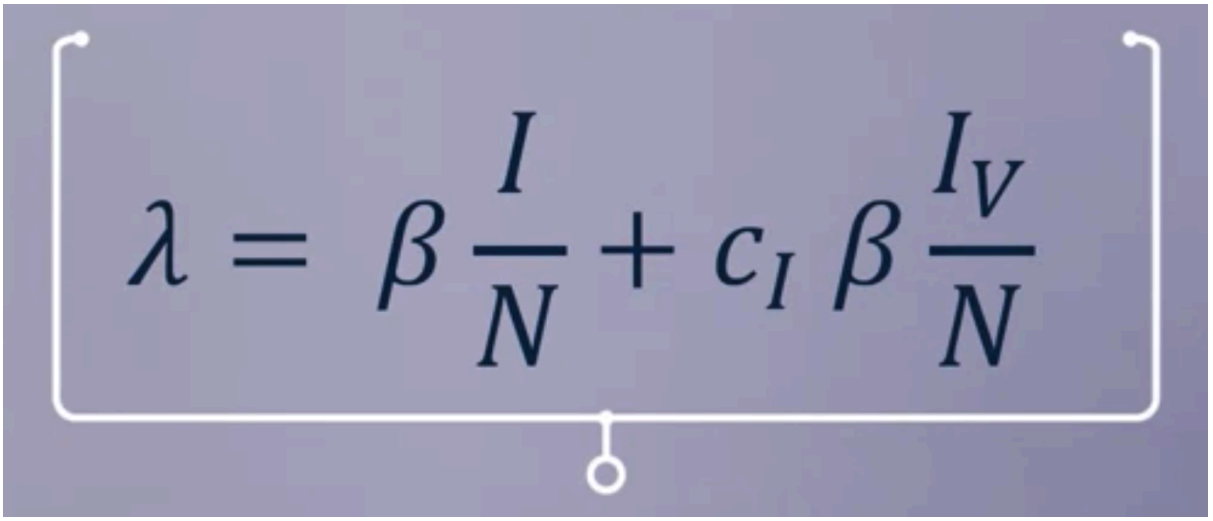
Additional vaccine effects

Vaccines may reduce severity and/or infectiousness, in addition to likelihood of infection.

To model this, we add a new compartment to distinguish those infected who have been vaccinated. Call it I_V
I.e. if we assume the vaccine doesn't affect recovery time, the schematic would be:



If the vaccine reduces infectivity, we can modify the force of infection from the total infected group ($I + I_V$) with a term c_I, which captures this aspect of vaccine effectiveness. If c_I is 0, the vaccinated people are completely non-infectious. If c_I is 1, the vaccine has no effect on infectivity.



Week 2

Models and data

Manually adjusting parameters to fit the data, and thereby estimating β and ɣ

With a simple enough model, we can make analytic progress.
Given in an early stage S/N ~= 1, then dI/dt ~= βI - ɣI
The solution to this is $I = I_0 \exp(\beta - \gamma)t$

Because the early exponential growth is driven by the difference between β and ɣ, it is difficult to estimate either one confidently without much data - there are (infinitely?) many solutions that match the data.

With sufficient data, you can estimate these parameters. But what if you have more unknowns, as in a more realistic scenario? E.g. if there's some previous immunity, such that S_0 is also an unknown (in addition to β and ɣ). So we need more data.

Week 3

Computer based calibration

The basic scheme is comparing the data and model simulation using a **distance function** (how closely the model matches the data), and then find the minimum of that function using an **optimisation function** to arrive at the best fitting values of β and ɣ.

Least squares calibration

Basic form of a distance function.

For two parameters, you can visualise this minimum as the lowest point on a 2-dimensional plane with varying height. Dealing with n unknown parameters just shifts this into finding the minimum in an n-dimensional space.

Least-squares is a good method for this toy, simple model. Modellers in the real world use something more complicated, called likelihoods.

Week 4

Constructing likelihoods

The basic premise of using likelihoods is:
'What is the likelihood that an epidemic with these parameters will give me the data that I observe?'
So we're trying to find the parameters that maximise the likelihood of actually seeing the observed data.

Likelihood for a coin estimate -> binomial probability model (do ID-models also use binomial distributions?)

So - we're trying to find β and ɣ in an SIR model, with some prevalence data. The observed prevalence is D, and the model-based predicted prevalence is N, of whom a proportion p are reported0. What's the likelihood of D given N?
Different approaches use binomial, Poisson, or normal distributions. We'll focus on Poisson. That is:

$$\text{Poisson Likelihood of data } D \text{ given } p, N = \frac{(pN)^D \exp(-pN)}{D!}$$

This is just for one data point. To do it for the full epidemic curve, we multiply them all up, as below:

$$\text{Overall Likelihood of data for full epidemic curve} = \frac{(pN_1)^{D_1} \exp(-pN_1)}{D_1!} \times \frac{(pN_2)^{D_2} \exp(-pN_2)}{D_2!} \times \frac{(pN_3)^{D_3} \exp(-pN_3)}{D_3!} \times \cdots$$

$$= \prod_i \frac{(pN_i)^{D_i} \exp(-pN_i)}{D_i!} \quad \text{(in more compact mathematical notation)}$$

And we want to find the parameters that maximise the likelihood of the data given those parameters.

The likelihood can range massively over the possible parameters. This can be a problem, e.g. round-off errors. To address this we use log-likelihood. I.e. For a given parameter L, we instead use $\log_e L = x$, where $e^x = L$

So now let's look at the logarithm of the Poisson distribution for a given data point:

$$D_1 \log_e (pN_1) - pN_1 - \log (D_1!)$$

Taking the log changes the formula for finding the overall distribution to a sum of the log-likelihoods:

$$\text{Log (Overall Poisson Likelihood)} = \begin{aligned} & D_1 \log(pN_1) - pN_1 - \log(D_1!) \\ & + D_2 \log(pN_2) - pN_2 - \log(D_2!) \\ & + D_3 \log(pN_3) - pN_3 - \log(D_3!) \\ & + \cdots \end{aligned}$$

(Usually approximated using Stirling's formula)

Log-likelihoods are almost always more useful.

Likelihood terms let us model uncertainty, which is an extremely useful feature.

A rule of thumb: the most likely parameter range is that which is within 1.92 log-likelihood of the maximum. This gives the 95% uncertainty range.

Building on the SIR model

Week 1 - Stochasticity

Prior to the introduction of a vaccine in the1960s, measles epidemics occurred every 2-3 years.

A deterministic model is often appropriate for large population numbers or frequent infection.

But the early stage of an epidemic requires a stochastic approach - i.e. in small populations, and/or when infections are rare.

What is a basic approach to stochastically modelling infectious disease?
The Gillespie algorithm - where you attach probabilities to transmission, recovery, or death, and generate a random number to decide the outcome for each case.

Week 2 - Heterogeneity

Model-informed UK influenza policy: Vaccinating the elderly to reduce deaths, and vaccinating children to reduce transmission.

https://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.0050074

How are age groups notated in infectious disease modelling?
One example: children = $_1$ and adults = $_2$
So number of infected children = I_1 , infected adults = I_2 etc.
Can denote contact rates similarly, e.g. per adult number of daily contacts with children = c_{21}
Can also have c_{12} , c_{22} , c_{11}

Then, by getting and summing the various potential contact rates, we can calculate the overall force of infection for a group:

$$\lambda_2 = bc_{21}\frac{I_1}{N_1} + bc_{22}\frac{I_2}{N_2}$$

We can use shorthand notation:

$$\lambda_i = \Sigma_j bc_{ij}\frac{I_j}{N_j}$$

Paper with modelling supporting the UK policy shift on influenza vaccination (adding childhood vaccination to quell transmission, in addition to vaccinating the elderly to reduce deaths)

Week 3 - Vector-borne diseases

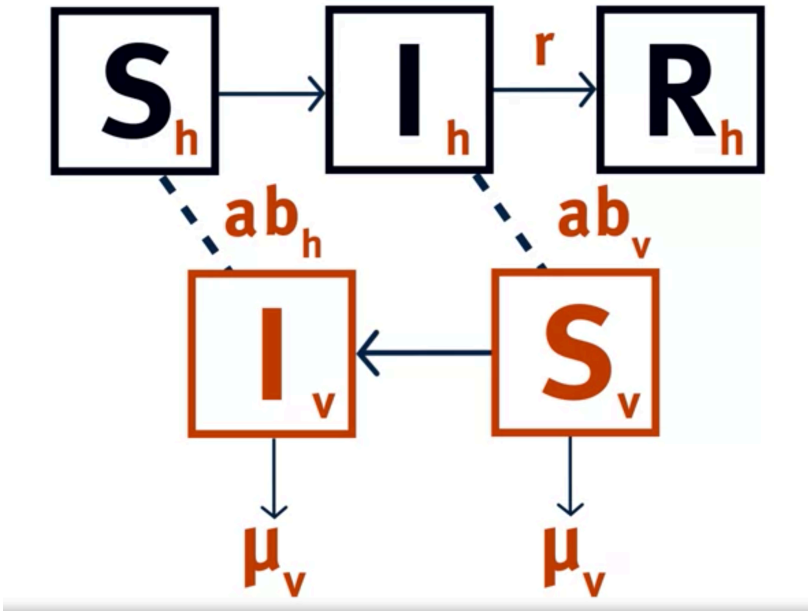
One popular form of compartmental model for vector-borne diseases is the Ross-MacDonald model.

We add two vector compartments - I and S. To differentiate members, we use the subscript $_h$ for host and $_v$ for vector

Because vectors of Dengue, once infected, remain infectious for life, there's no exiting the I_v compartment (what about vector death? Wouldn't absolute vector numbers be important to include? - Ah yes - with μ_V applying to the I_V and S_V compartments - they ignore host mortality in this model as it is so small compared to vector mortality)

Disease transmission interactions between vectors and hosts are denoted with a doted line, and are moderated by a rate *a*, which in this case might mean the biting rate - the number of blood meals a mosquito takes over a set of days.
b_h denotes the probability of an infection being transmitted from a vector to a host, and b_V denotes an infection transmitting from a host to a vector.
h denotes the proportion of blood meals a vector takes from a host, as some vectors feed from multiple hosts (i.e. other non-human species?). For dengue, h=1 (and so can be ignored).

All up, this results in:



We get a differential equation by denoting the number of vectors, N_V , and the birth rate of vectors μ_V (note - equal to the mortality rate).

$$\frac{dS_v}{dt} = \mu_v N_v - a b_v S_v \frac{I_h}{N_h} - \mu_v S_v$$

and

$$\frac{dI_v}{dt} = a\, b_v S_v \frac{I_h}{N_h} - \mu_v I_v$$

Then for the susceptible hosts:

$$\frac{dS_h}{dt} = - a\, b_h \frac{S_h}{N_h} I_v$$

And for infected hosts:

$$\frac{dI_h}{dt} = a\, b_h \frac{S_h}{N_h} I_v - r\, I_h$$

Note that r represents the recovery rate of infected hosts, such that finally:

$$\frac{dR_h}{dt} = r\, I_h$$

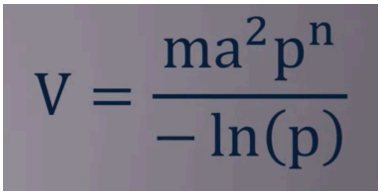
For vector-borne diseases, R_0 is the product of the R_0 from the vector to the host times the R_0 from the host to the vector.

$$R_o = R_{o\ v\text{-}h} * R_{o\ h\text{-}v}$$

Using the equations above, we get:

$$R_o = \left(\frac{N_v}{N_h}\right) a^2 b_h b_v / r \mu_v$$

Vectorial capacity = the number of secondary cases arising per day from a single infective case in a totally susceptible population.



where m = the ratio of mosquitoes to humans, or N_V / N_H
a = biting rate
p = probability of a mosquito surviving one day
n = extrinsic incubation period of the virus, usually in days
Spraying insecticide (fogging) would reduce N_V and so reduce m in the vectorial capacity equation. It could also affect p

PPE or repellant reduces a, the biting rate.

The propensity of different species to bite leads to variation in a

Changes in the environment, like humidity or temperature, could reduce p. However, increasing the temperature could also speed up transmission, thus lowering the extrinsic incubation period n

So interventions can affect multiple parameters, and so have complex consequences. Not to mention heterogeneity.

What about larvicidal campaigns? There'd be less mosquitoes - reducing m - though this would be hard to estimate confidently.

Appendix - Epi 101 (Data4Sci blogposts)

Some complications to add to a SIR model:
Asymptomatic and mildly infectious (i.e. variance in infectiousness) cases
Dynamic lags
Lockdown procedures
Structured populations (e.g. two neighbouring cities with total population P is very different from a single pot of size P).
Incubation period
Temporal immunity

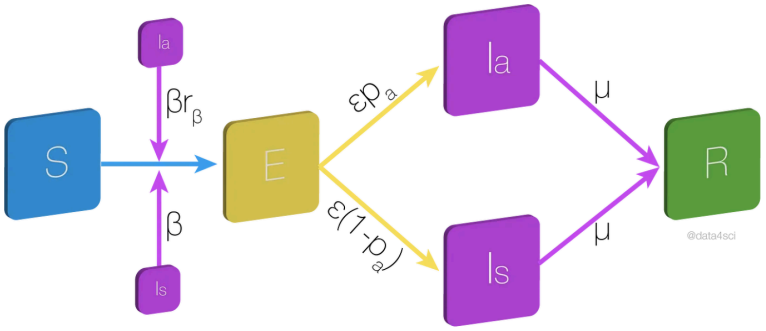
Mortality rate
Stochastic variability
Seasonal variation

Maybe useful generic code (in Python though):
https://github.com/DataForScience/Epidemiology101/blob/master/EpiModel.py

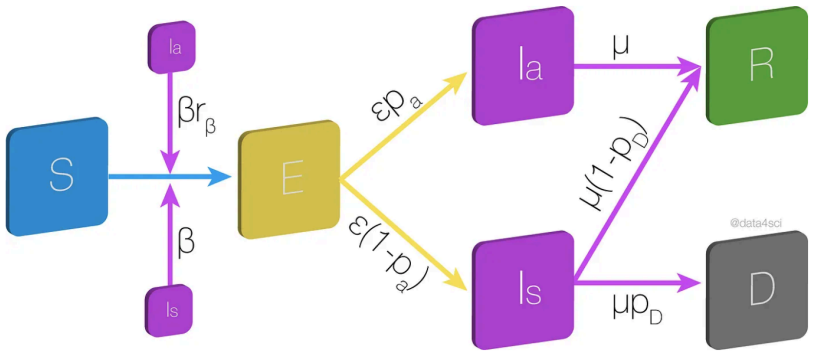
Notes that interactions have terms involving two compartments (e.g. from susceptible to infected depending on force of infection which depends on compartments), while spontaneous transitions have terms involving just one compartment (e.g. the linear transition from infected to recovery).

A simple way to model re-infection is a spontaneous transition from the recovered to the susceptible compartment with rate ρ (rho), making a SEIRS model.
A SEIRS model allows an 'endemic' disease where the fuel of susceptible people never burns out.
The rate ρ at which immunity is lost has a determinant effect in the progress of the epidemic and the rise of endemicity. E.g. a sufficiently low ρ can allow peaks and declines before endemicity.

Also, asymptomatic individuals are often less infectious than symptomatic ones, by some fraction r_β
We can model that with a subdivided infectious compartment, e.g.:



Also included mortality results in 6 compartments with 7 transitions and 6 parameters:



And early in a pandemic most of these parameters are partially or completely unknown.

Next step is making the model probabilistic or stochastic, with a Monte Carlo method.

Appendix - my_model_code.R

```
# LOAD THE PACKAGES:
library(deSolve)
library(reshape2)
library(ggplot2)

# MODEL INPUTS:

# Vector storing the initial number of people in each compartment (at timestep 0)
initial_state_values <- c(S = 1000000-1,  # the whole population we are modelling is
susceptible to infection
                          I = 1,          # the epidemic starts with a single infected person
                          R = 0)          # there is no prior immunity in the population

# Vector storing the parameters describing the transition rates in units of days^-1
parameters <- c(beta = 0.5,     # the infection rate, which acts on susceptibles
                gamma = 0.25)   # the rate of recovery, which acts on those infected

# TIMESTEPS:

# Vector storing the sequence of timesteps to solve the model at
times <- seq(from = 0, to = 100, by = 1)   # from 0 to 100 days in daily intervals

# SIR MODEL FUNCTION:

# The model function takes as input arguments (in the following order): time, state and
parameters
sir_model <- function(time, state, parameters) {

   with(as.list(c(state, parameters)), {   # tell R to unpack variable names from the state and
parameters inputs

   # Calculating the total population size N (the sum of the number of people in each
compartment)
     N <- S+I+R

   # Defining lambda as a function of beta and I:
     lambda <- beta * I/N

   # The differential equations
     dS <- -lambda * S              # people move out of (-) the S compartment at a rate lambda
(force of infection)
```

```
    dI <- lambda * S - gamma * I    # people move into (+) the I compartment from S at a rate
lambda,
                                # and move out of (-) the I compartment at a rate gamma (recovery)
    dR <- gamma * I                 # people move into (+) the R compartment from I at a rate
gamma

  # Return the number of people in the S, I and R compartments at each timestep
  # (in the same order as the input state variables)
  return(list(c(dS, dI, dR)))
  })

}

# MODEL OUTPUT (solving the differential equations):

# Solving the differential equations using the ode integration algorithm
output <- as.data.frame(ode(y = initial_state_values,
                            times = times,
                            func = sir_model,
                            parms = parameters))

# PLOTTING THE OUTPUT
output_long <- melt(as.data.frame(output), id = "time")                 # turn output dataset into
long format

# Adding a column for the proportion of the population in each compartment at each
timestep
# One way of calculating this is dividing the number in each compartment by the total initial
population size
# We can do this in this case because our population is closed, so the population size stays
the same
# at every timestep
output_long$proportion <- output_long$value/sum(initial_state_values)

# Plot this new column
ggplot(data = output_long,                                          # specify object containing data to
plot
      aes(x = time, y = proportion, colour = variable, group = variable)) +  # assign columns to
axes and groups
  geom_line() +                                                   # represent data as lines
  xlab("Time (days)")+                                              # add label for x axis
  ylab("Proportion of the population") +                                  # add label for y axis
  labs(colour = "Compartment")                                          # add legend title
```