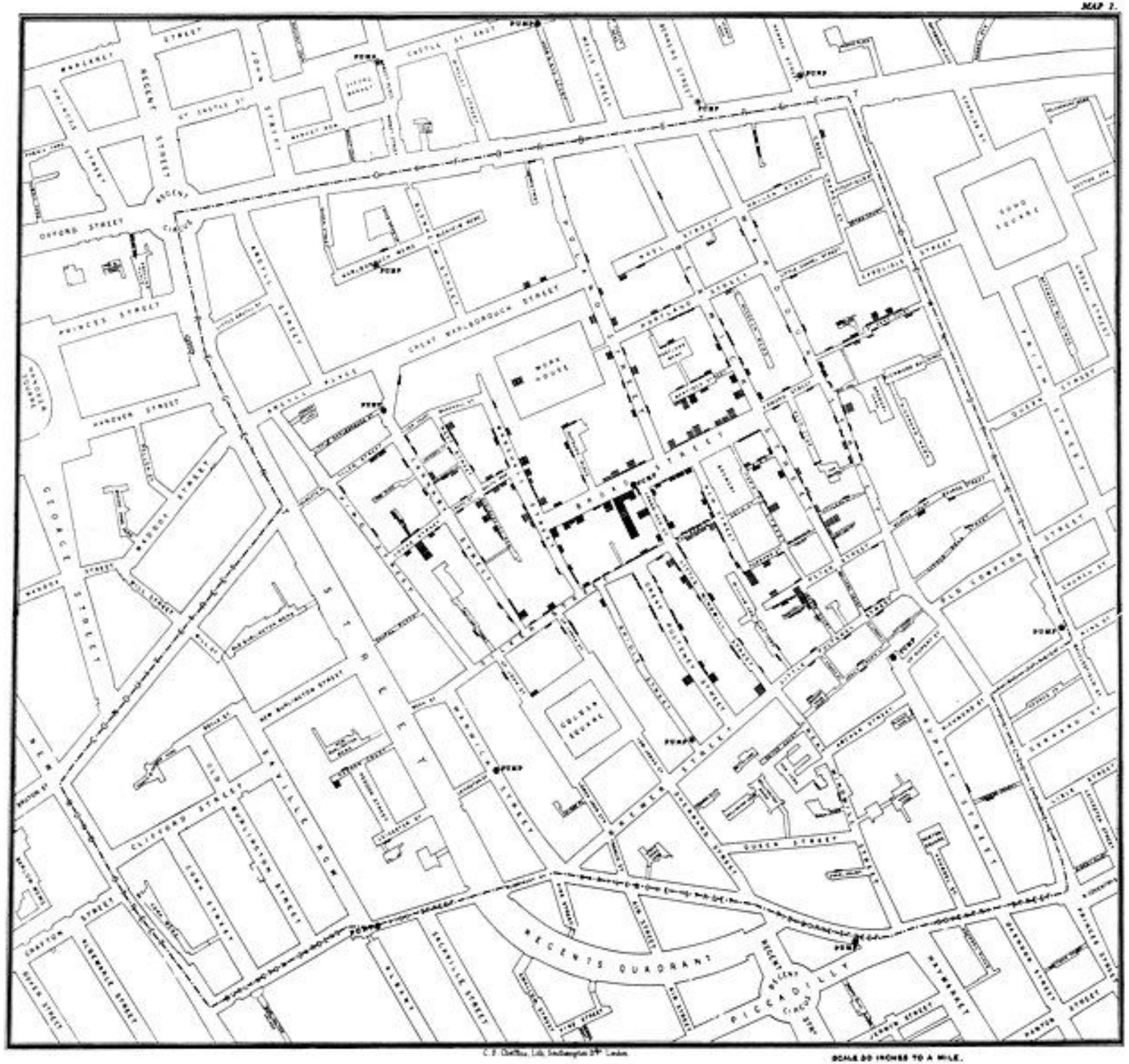
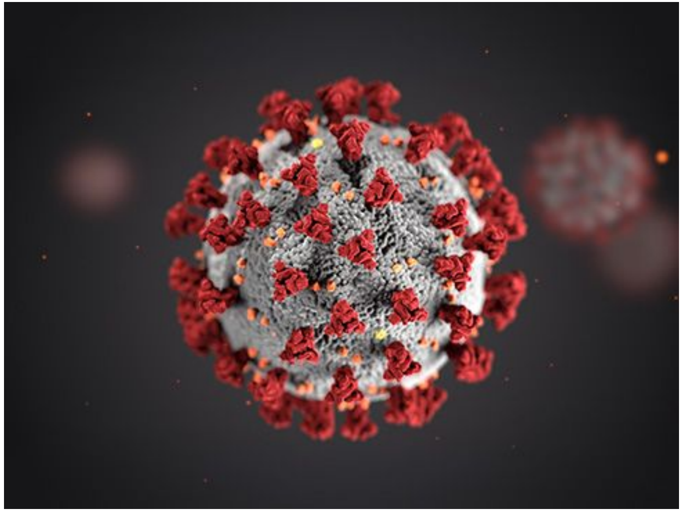


Modeling Contagion

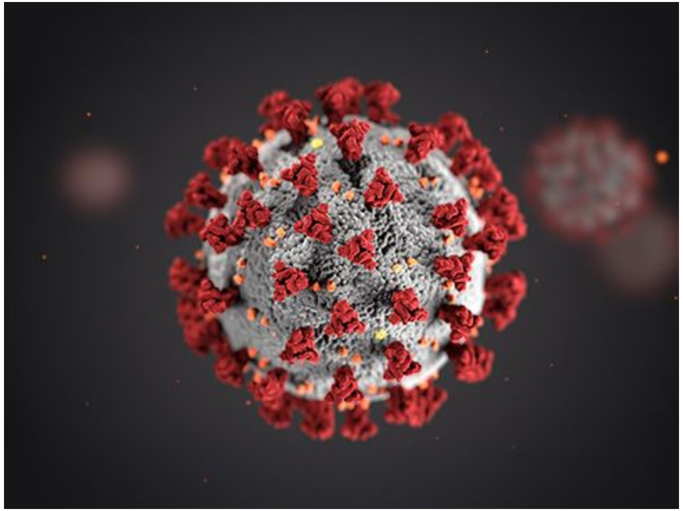
Nicolò Gozzi, ISI Foundation

SOCIAL COMQUANT

Epidemics



Epidemics



Epidemics

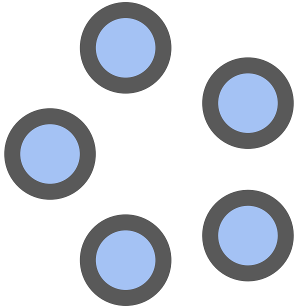
Epidemics

stonks
.560
0.168
1.4563
1.0287
False
Wrong Data
ANTI SCIENCE
NEWS
False
True
Misleading
Disinformation
False
False

Epidemics have three fundamental ingredients:

Epidemics have three fundamental ingredients:

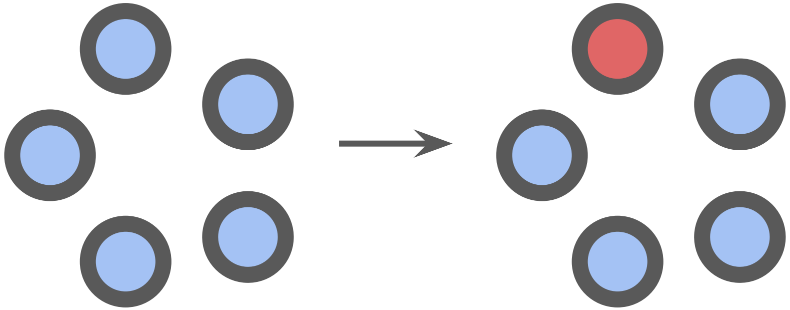
1) Individuals (susceptibles)



Epidemics have three fundamental ingredients:

1) Individuals (susceptibles)

2) Something that *spreads*

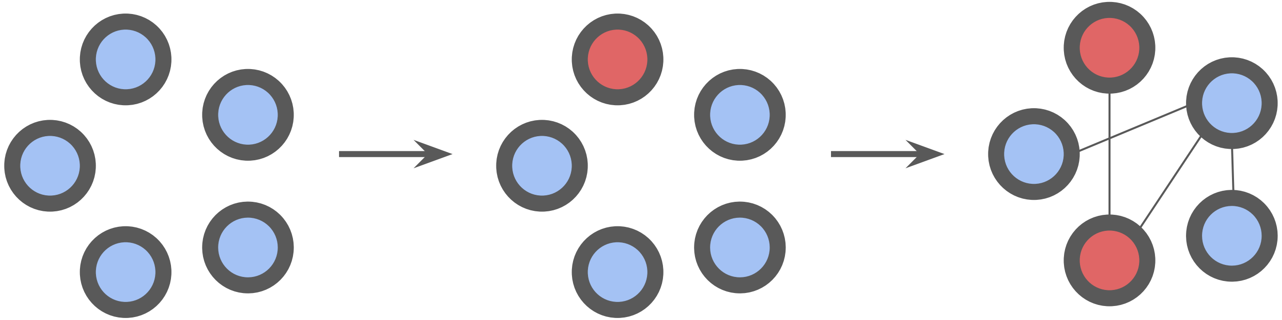


Epidemics have three fundamental ingredients:

1) Individuals (susceptibles)

2) Something that *spreads*

3) Interactions



Why are complexity scientists interested in Epidemics?

Epidemics are a perfect example of complex systems:

- Phase transitions, tipping points
- Emerging phenomena
- Feedback between spread and *behavior*
- Simple processes, complex outcomes

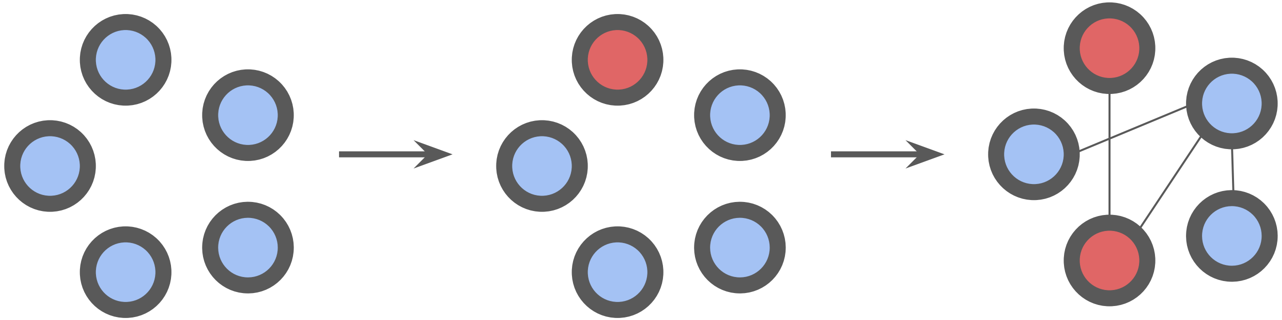
Why (Computational) Social Scientists should care about Epidemics?

Why (Computational) Social Scientists should care about Epidemics?

1) Individuals (susceptibles)

2) Something that *spreads*

3) Interactions

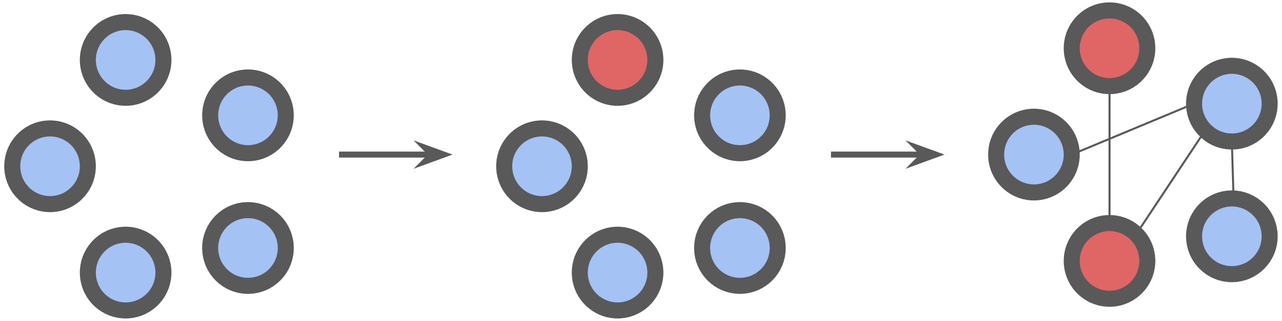


Why (Computational) Social Scientists should care about Epidemics?

1) Individuals (susceptibles)

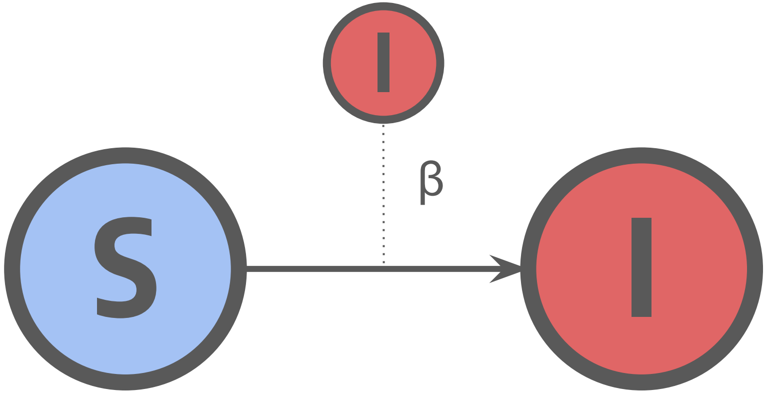
2) Something that *spreads*

3) Interactions



Compartmental Models

A more detailed example (SI model)

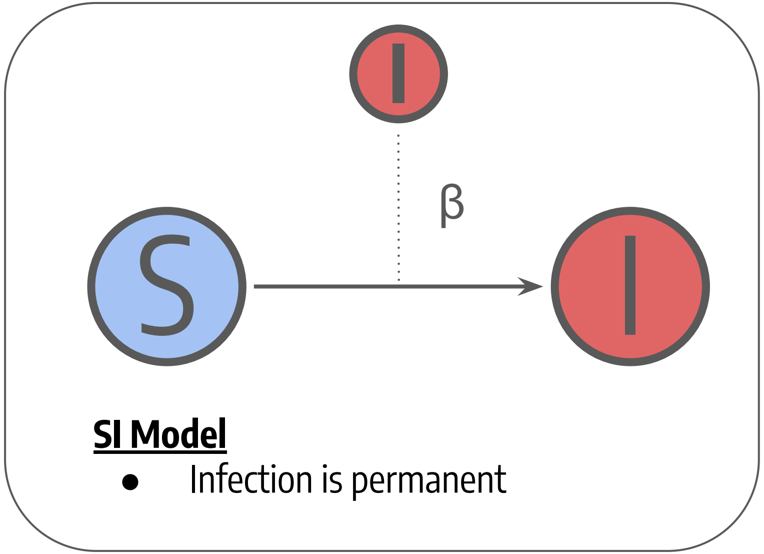


- Susceptible individuals interact with infectious
- With a certain probability β, the interaction leads to new infection

Compartmental models

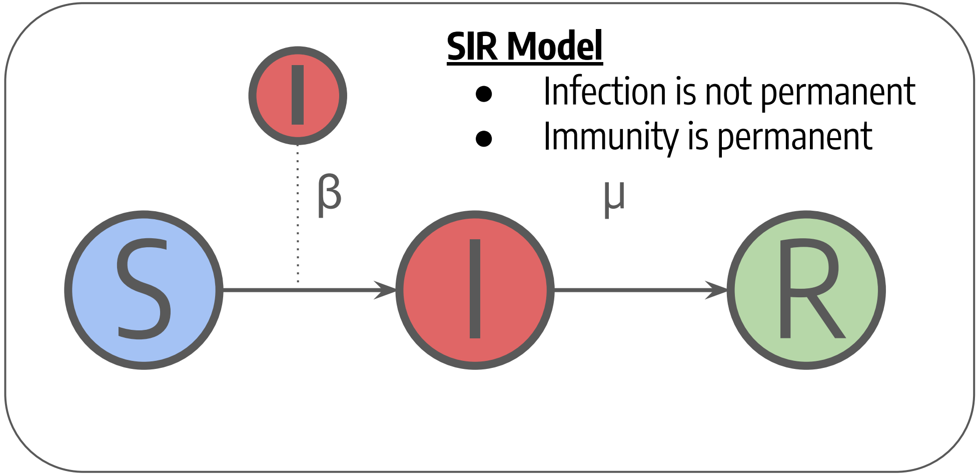
- The framework we have just seen is a compartmental model
- In compartmental models, individuals are divided into compartments (groups) according to their (health) status
- Individuals transition among compartments at given rates
- Transitions can be divided into two main categories:
  - infection processes (they are mediated by infectious agents)
  - spontaneous processes (e.g., recovery)

SI, SIR, SIS



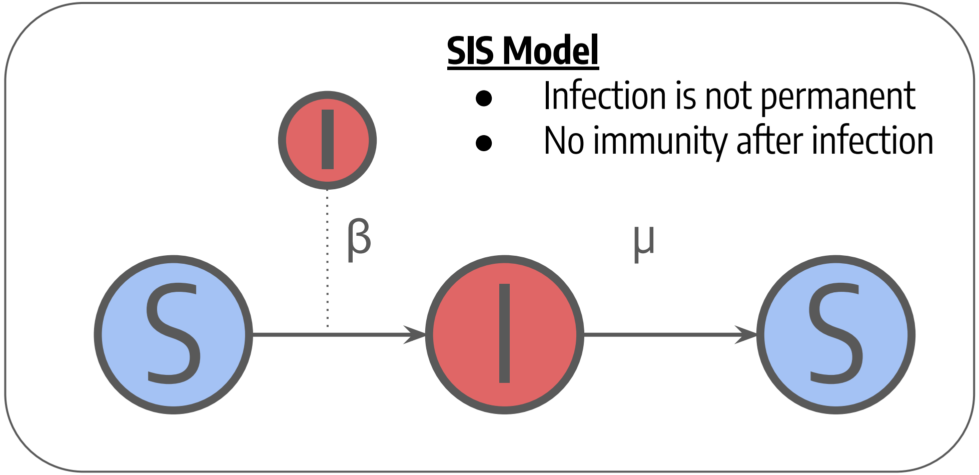
SI Model

- Infection is permanent



SIR Model

- Infection is not permanent
- Immunity is permanent



SIS Model

- Infection is not permanent
- No immunity after infection

Compartmental Models as ODEs

- Compartmental models are convenient because they can be easily turned into systems of differential equations (ODE)
- The ODE system describe the evolution in time of number of people in different compartments
- Consider the SIR model:

$$\frac{dS_t}{dt} = -\beta C \frac{I_t}{N} S_t$$

$$\frac{dI_t}{dt} = \beta C \frac{I_t}{N} S_t - \mu I_t$$

$$\frac{dR_t}{dt} = \mu I_t$$

Note: we are assuming that S, I, R are *continuous* quantities (i.e., we can have 0.5 individuals infected)

What are the factors influencing the force of infection?

Understanding the force of infection

$$\frac{I_t}{N}$$

probability of meeting and infectious individual

Understanding the force of infection

$$C\frac{I_t}{N}$$

The number of contacts an individual has

Understanding the force of infection

The infection probability of a single contact

$$\beta C \frac{I_t}{N}$$

Understanding the force of infection

$$\beta C \frac{I_t}{N}$$

- Main Limitation: Homogeneous Mixing Assumption
- Individuals bump into each other like gas particles (or marbles in a box)
- The force of infection is proportional to the density of infected individuals (mass-action law)

Understanding the infection term

$$\beta C \frac{I_t}{N}$$

Disclaimer: in the following we will omit C and when we write “β” we implicitly means “β · C”

Solving the System under early-stage assumption

When the number of infected is small, the system of ODE for the SIR can be solved *easily*:

$$\frac{dI_t}{dt} = \beta \frac{I_t}{N} S_t - \mu I_t$$

Solving the System under early-stage assumption

When the number of infected is small, the system of ODE for the SIR can be solved *easily*:

Early-stage assumption

$$\frac{dI_t}{dt} = \beta \frac{I_t}{N} S_t - \mu I_t$$

$$\frac{dI_t}{dt} \simeq \beta \frac{I_t}{N} N - \mu I_t$$

At start:

I∼0

R∼0

S∼N

Solving the System under early-stage assumption

When the number of infected is small, the system of ODE for the SIR can be solved *easily*:

$$\frac{dI_t}{dt} = \beta \frac{I_t}{N} S_t - \mu I_t$$

$$\frac{dI_t}{dt} \simeq \beta \frac{I_t}{\cancel{N}} \cancel{N} - \mu I_t$$

$$\frac{dI_t}{dt} \simeq \beta I_t - \mu I_t$$

At start:
I∼0
R∼0
S∼N

Solving the System under early-stage assumption

When the number of infected is small, the system of ODE for the SIR can be solved *easily*:

$$\frac{dI_t}{dt} = \beta \frac{I_t}{N} S_t - \mu I_t$$

$$\frac{dI_t}{dt} \simeq \beta \frac{I_t}{N} N - \mu I_t$$

$$\frac{dI_t}{dt} \simeq \beta I_t - \mu I_t$$

At start:
I∼0
R∼0
S∼N

$$\frac{dI_t}{dt} \simeq (\beta - \mu) I_t$$

Solving the System under early-stage assumption

When the number of infected is small, the system of ODE for the SIR can be solved *easily*:

$$\frac{dI_t}{dt} = \beta\frac{I_t}{N}S_t - \mu I_t$$

$$\frac{dI_t}{dt} \simeq \beta\frac{I_t}{N}N - \mu I_t$$

At start:
I∼0
R∼0
S∼N

$$\frac{dI_t}{dt} \simeq \beta I_t - \mu I_t$$

$$\frac{dI_t}{dt} \simeq (\beta - \mu)I_t$$

$$\frac{dI_t}{dt} \simeq \mu(\boxed{\frac{\beta}{\mu}} - 1)I_t$$

$$\frac{dI_t}{dt} \simeq \mu(\boxed{R_0} - 1)I_t$$

Define:

$$R_0 = \frac{\beta}{\mu}$$

Solving the System under early-stage assumption

When the number of infected is small, the system of ODE for the SIR can be solved *easily*:

$$\frac{dI_t}{dt} = \beta \frac{I_t}{N} S_t - \mu I_t$$

At start:
I∼0
R∼0
S∼N

$$\frac{dI_t}{dt} \simeq \beta \frac{I_t}{N} N - \mu I_t$$

$$\frac{dI_t}{dt} \simeq \beta I_t - \mu I_t$$

$$\frac{dI_t}{dt} \simeq (\beta - \mu) I_t$$

$$\frac{dI_t}{dt} \simeq \mu(\frac{\beta}{\mu} - 1) I_t$$

Define:

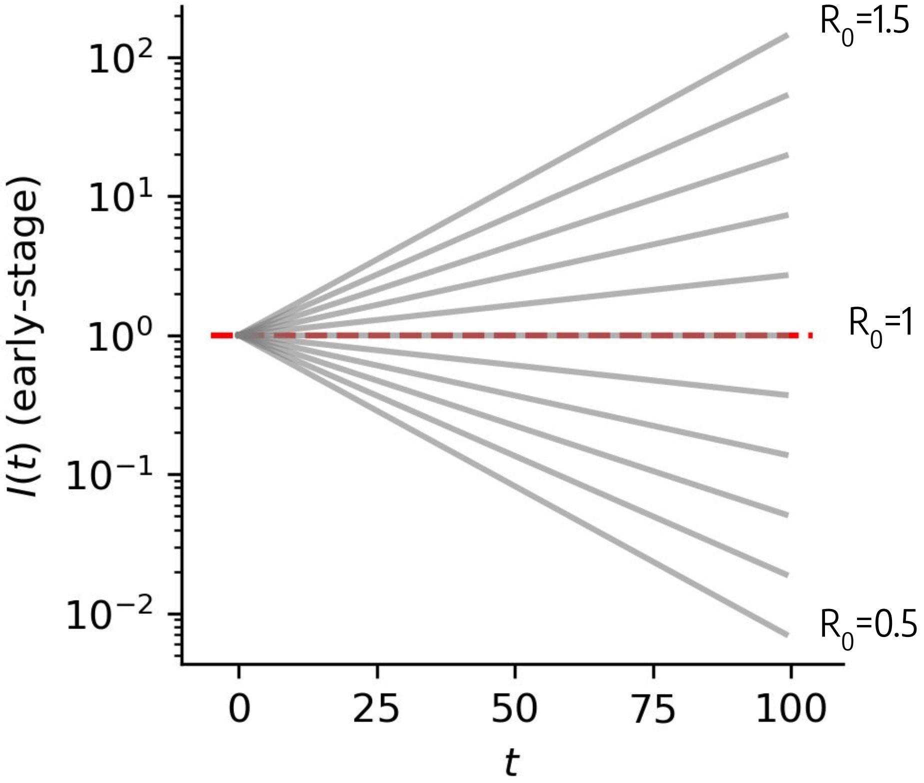
$$R_0 = \frac{\beta}{\mu}$$

$$\frac{dI_t}{dt} \simeq \mu(R_0 - 1) I_t$$

$$\boxed{I(t) = I_0 e^{\mu(R_0 - 1)}}$$

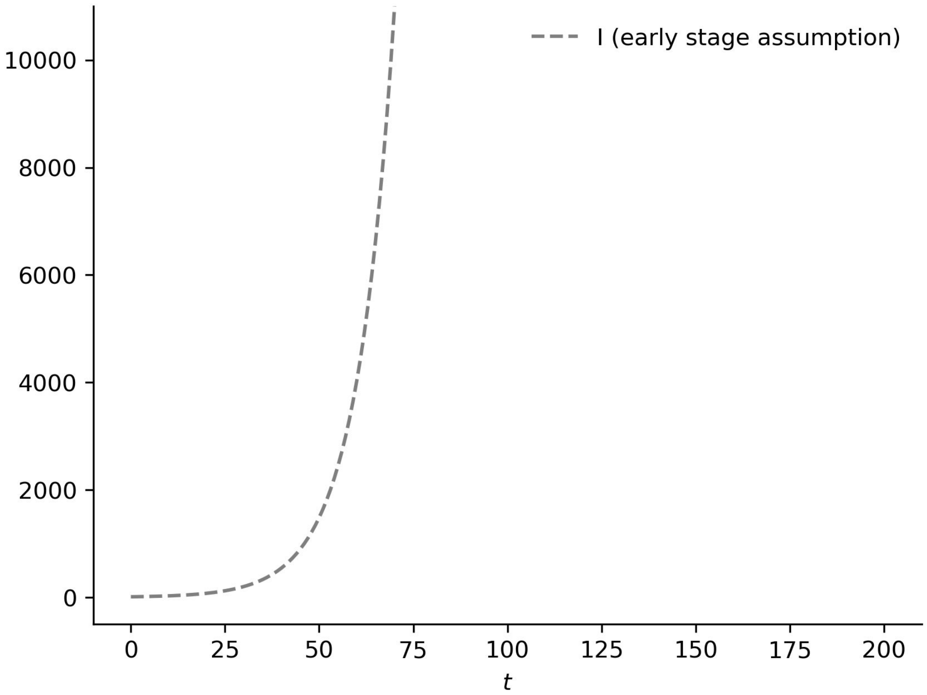
Interpreting R_0

- R_0 is the number of secondary infections generated by a infected case in a fully susceptible population of infinite size
- In practice, if $R_0 > 1$ the epidemics will grow, if $R_0 < 1$ it will decrease
- Think of it as the number of offsprings needed to maintain the size of a population



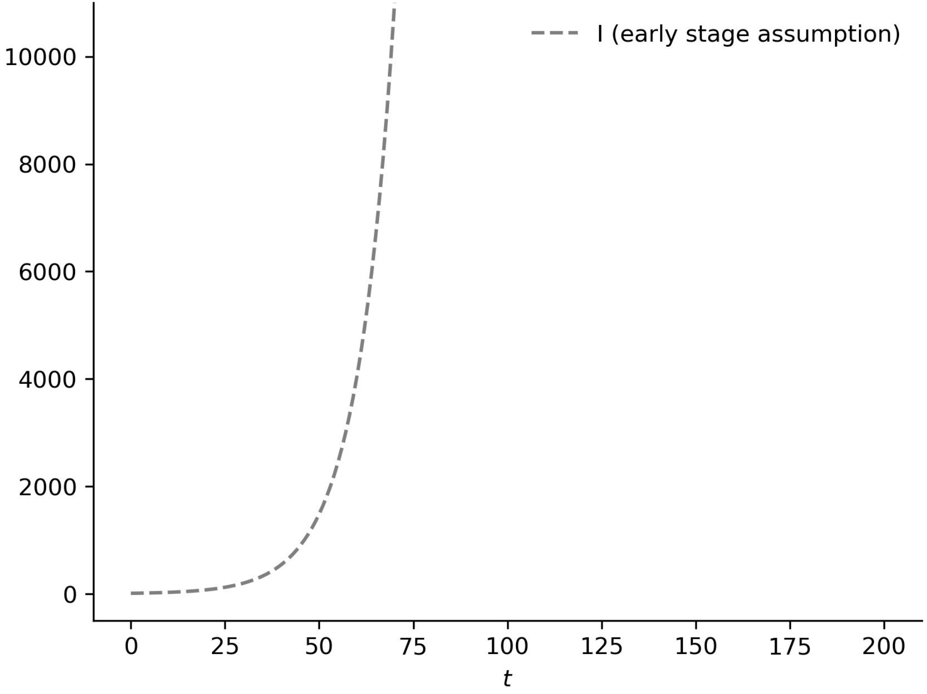
$$I(t) = I_0 e^{\mu(R_0-1)}$$

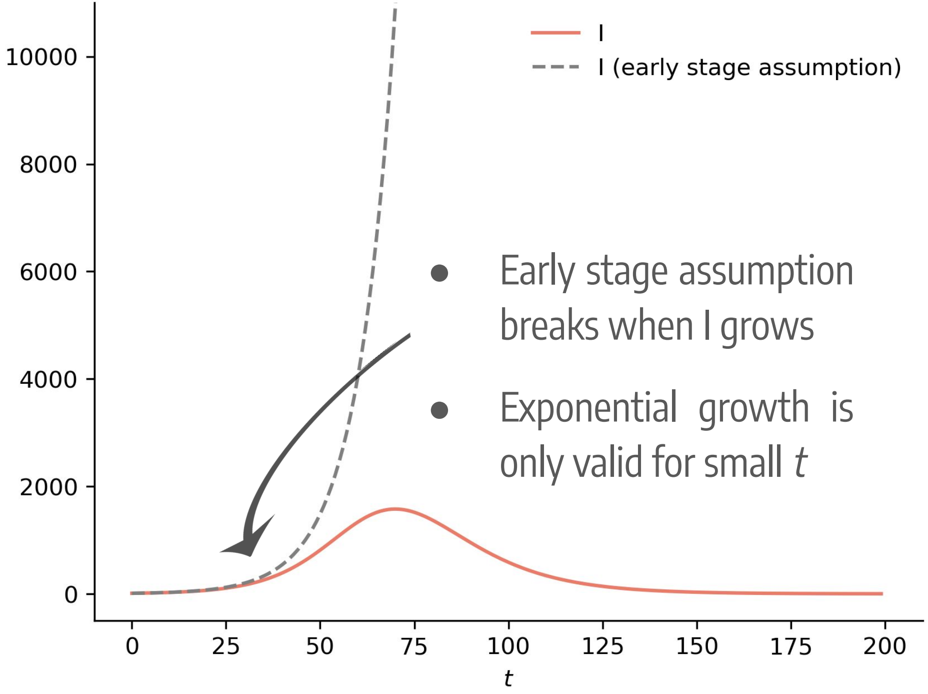
Testing the early stage assumption



If we plot the solution for I(t) this is what we got, the number of infected grows indefinitely...

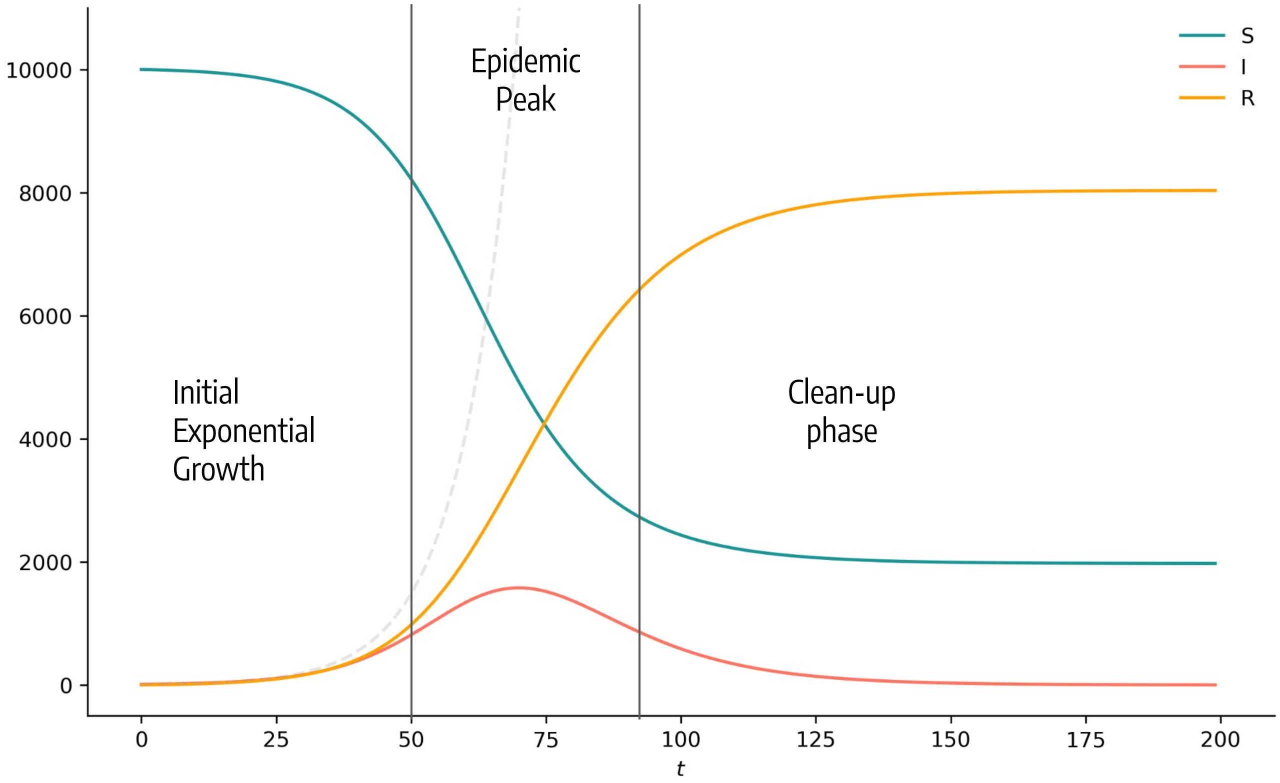
Testing the early stage assumption





- Early stage assumption breaks when I grows
- Exponential growth is only valid for small t

Epidemic Phases



Initial Exponential Growth

Epidemic Peak

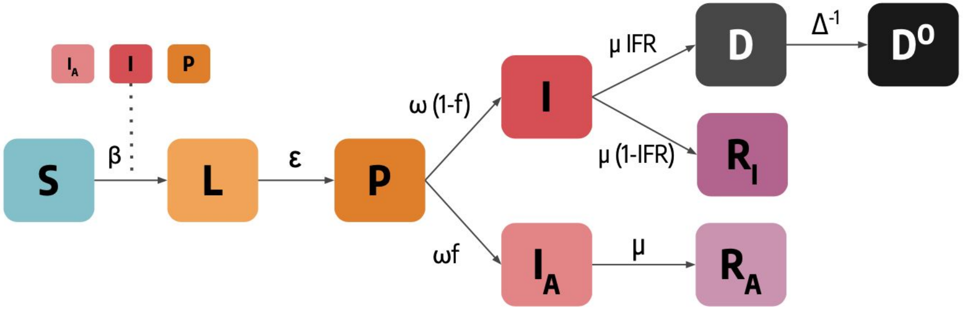
Clean-up phase

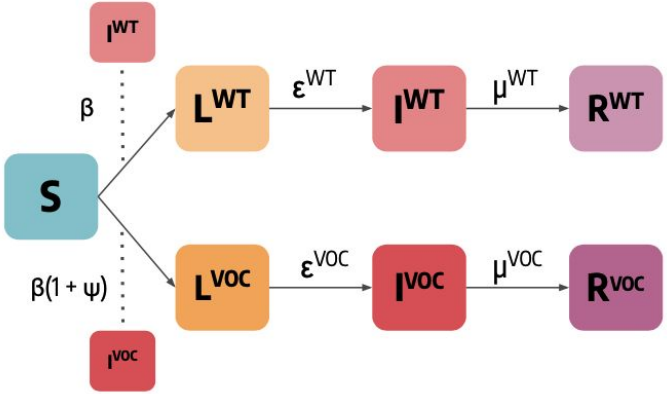
Extending the Compartments

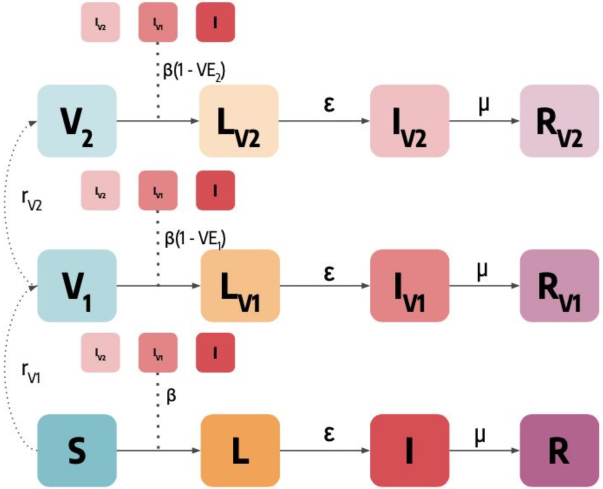
The compartmental structure can be changed and adapted according to the phenomenon we are studying:

- Compartments for Vaccinated, Asymptomatic, Pre-Symptomatic, Hospitalized, ...

Extending the Compartments - More Complex Disease





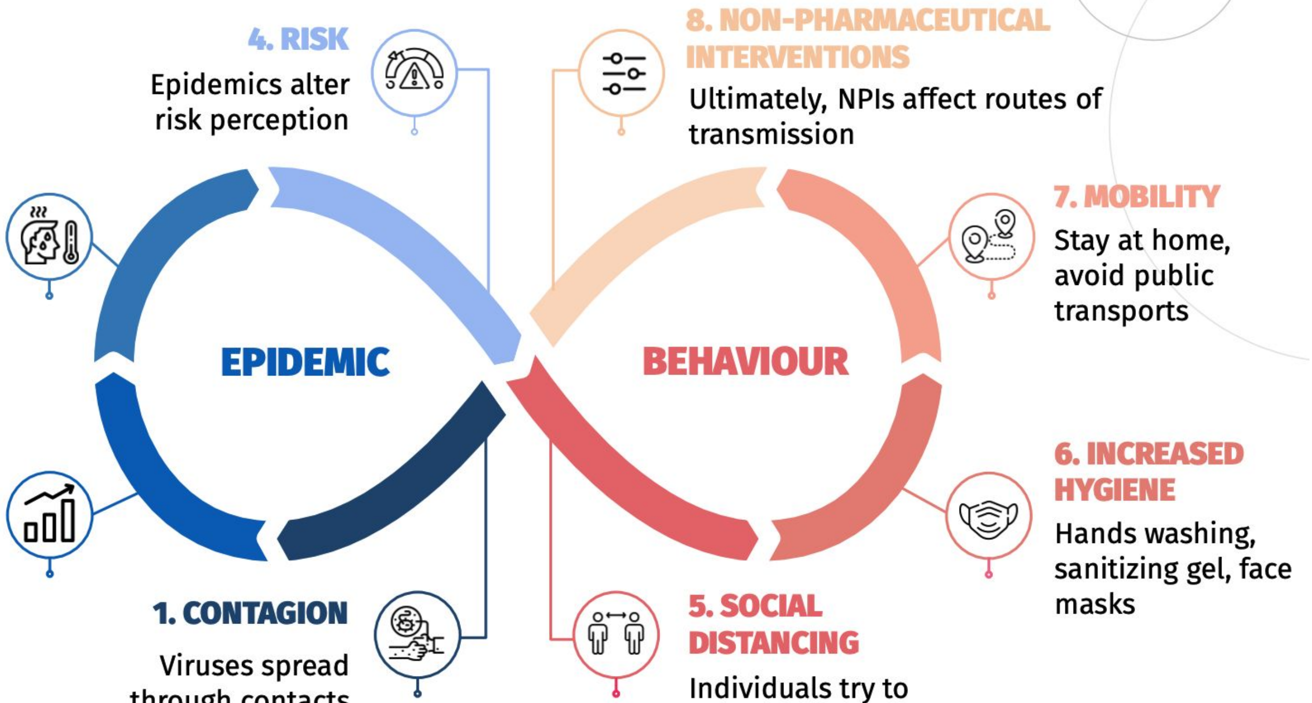


Extending the Compartments

The compartmental structure can be changed and adapted according to the phenomenon we are studying:

- Compartments for Vaccinated, Asymptomatic, Pre-Symptomatic, Hospitalized, ...
- Compartments may represent also “behavioral” classes:
  - Individuals may be more or less suceptible to the “virus” (behavioral preventive measures, level of information, gullibility, etc)

Epidemic-Behavior Feedback Loop



1. CONTAGION

Viruses spread through contacts

2. GROWTH

Epidemics growth is influenced by rate of interaction

3. SEVERITY AND SUSCEPTIBILITY

Diseases affect people differently, leading to heterogeneous risk perception

4. RISK

Epidemics alter risk perception

5. SOCIAL DISTANCING

Individuals try to avoid contacts

6. INCREASED HYGIENE

Hands washing, sanitizing gel, face masks

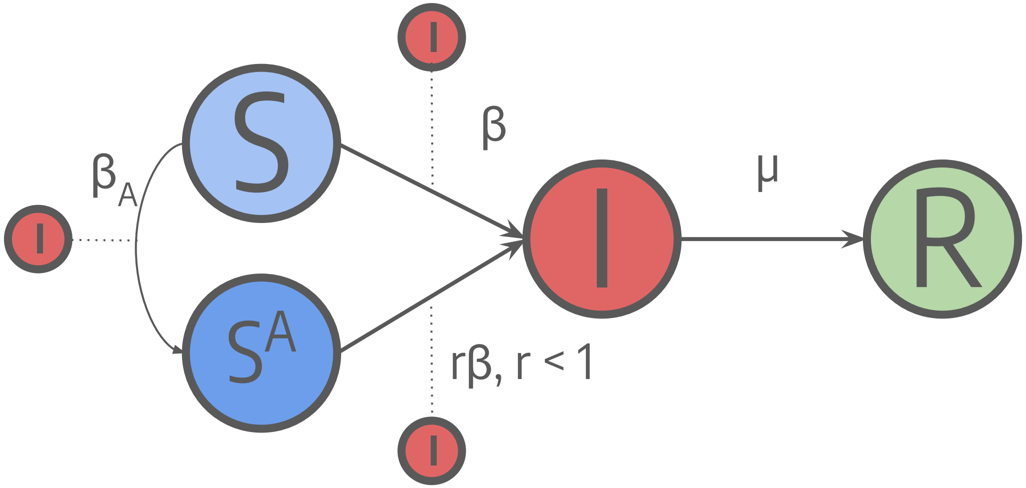
7. MOBILITY

Stay at home, avoid public transports

8. NON-PHARMACEUTICAL INTERVENTIONS

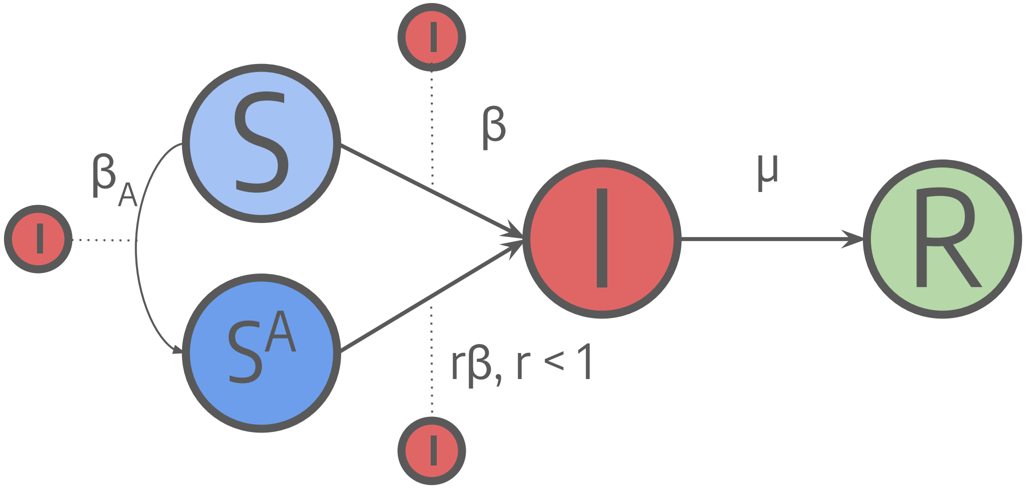
Ultimately, NPIs affect routes of transmission

Extending the Compartments - Behavioral Information



- Compartments can also represent “behavioral” classes
- For example, some individuals may be less susceptible due to behavior change / awareness

Extending the Compartments - Behavioral Information

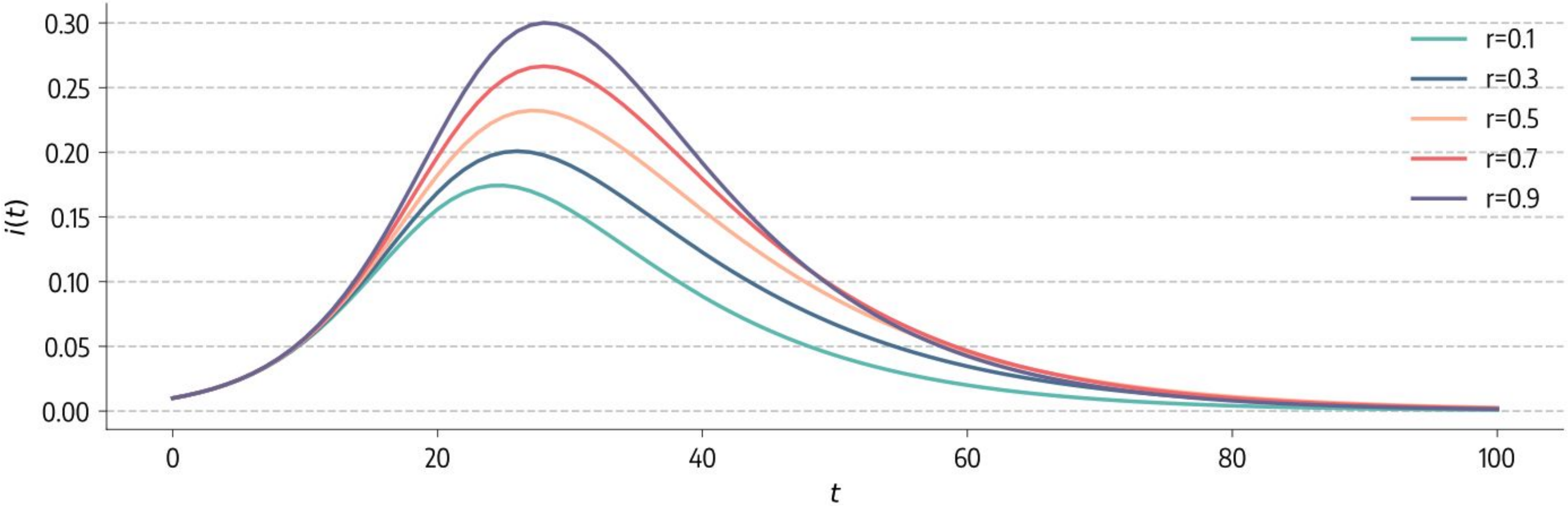


- Compartments can also represent "behavioral" classes
- For example, some individuals may be less susceptible due to behavior change / awareness

ODE System

$$\begin{aligned}
\frac{dS_t}{dt} &= -\beta\frac{I_t}{N}S_t - \beta_A\frac{I_t}{N}S_t \\
\frac{dS_t^A}{dt} &= -r\beta\frac{I_t}{N}S_t^A + \beta_A\frac{I_t}{N}S_t \\
\frac{dI_t}{dt} &= \beta\frac{I_t}{N}S_t + r\beta\frac{I_t}{N}S_t^A - \mu I_t \\
\frac{dR_t}{dt} &= \mu I_t
\end{aligned}$$

Extending the Compartments - Behavioral Information



A note on Machine Learning Models

- What we considered so far are Mechanistic Models
- In a Mechanistic model, we explicitly encode the *physics* of the problem
- In a Statistical model, we learn the mechanisms behind the system from the data
- Main pro of mechanistic model: easy to run counterfactuals (*“what if tomorrow R_0 is 2x?”*)

Technical Aspects

Simulation

- We can derive analytically some interesting characteristics of the system (such as R_0)
- But when the system becomes more complicated simulation become essential
- Simulation: writing a computer program that *simulates* the system

Deterministic and Stochastic Simulations

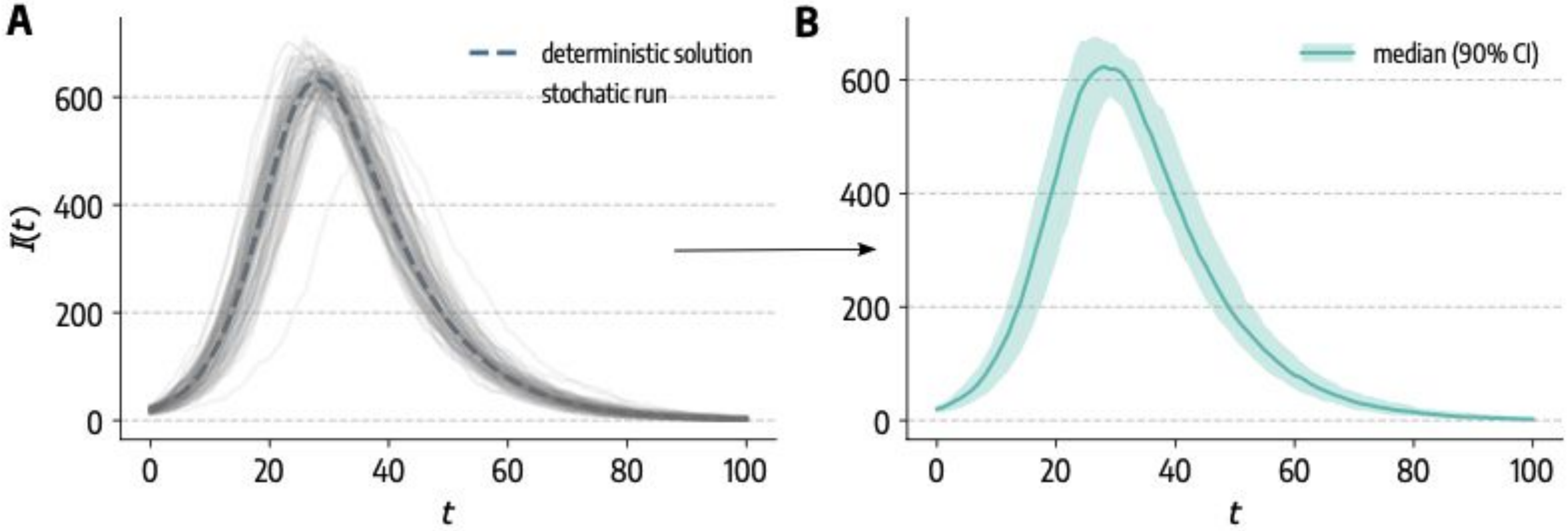
Deterministic Simulations

- Numerical integration of the system of ODEs
- Pros:
  - Easy to implement, can leverage all methods for ODE system integration
  - Faster
  - For a given set of parameters and initial conditions 1 run is enough
- Cons:
  - No uncertainty
  - Number of individuals is approximated as a continuous quantity

Stochastic Simulations

- Simulate transitions among compartments via probabilistic sampling
- Pros:
  - Allow to include uncertainty (important especially for scenarios, forecast, early stage)
- Cons:
  - Multiple runs needed
  - Slower

Deterministic and Stochastic Simulations



Deterministic Simulation - Python Implementation (i)

In order to be represented in a programming language, equations must be discretized in time, according to the well-known Euler method:

$$\frac{dy}{dt} \simeq \frac{y(t+\Delta t) - y(t)}{\Delta t}$$

For the SIR Model:

$$\begin{aligned}
\frac{S(t+\Delta t) - S(t)}{\Delta t} &= -\beta \frac{I(t)}{N} S(t) \\
\frac{I(t+\Delta t) - I(t)}{\Delta t} &= \beta \frac{I(t)}{N} S(t) - \mu I(t) \\
\frac{R(t+\Delta t) - R(t)}{\Delta t} &= \mu I(t)
\end{aligned}$$

Deterministic Simulation - Python Implementation (ii)

Last expressions provide us with a convenient recursive formula to get the number of individuals in different compartments in the next step:

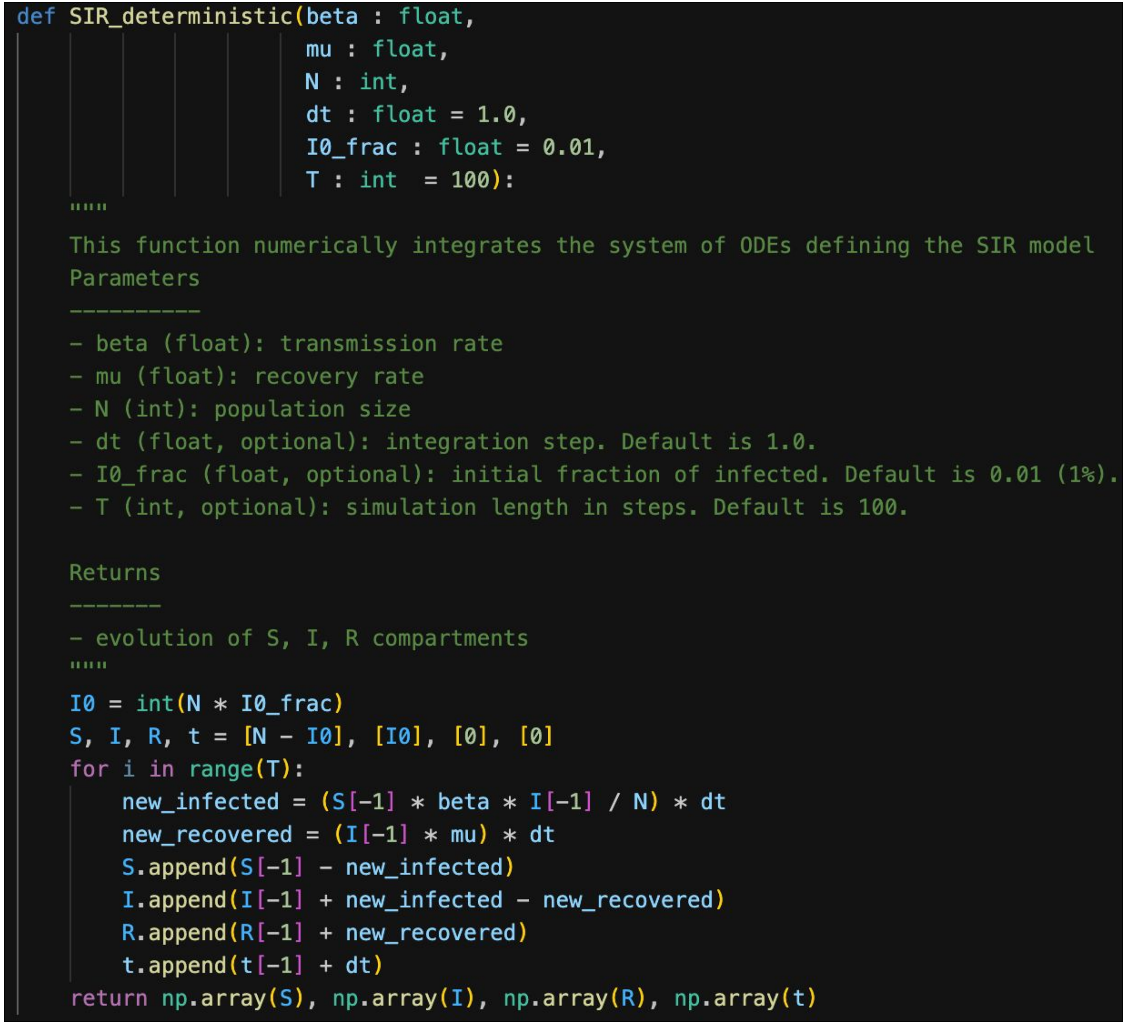
$$S(t+\Delta t) = S(t) - \beta \frac{I(t)}{N} S(t)\Delta t$$

$$I(t+\Delta t) = I(t) + (\beta \frac{I(t)}{N} S(t) - \mu I(t))\Delta t$$

$$R(t+\Delta t) = R(t) + \mu I(t)\Delta t$$

Setting initial conditions S(0),I(0),R(0), one can iteratively apply these equations to find S(1), I(1), R(1), . . . S(T), I(T), R(T)

Deterministic Simulation - Python Implementation (iii)



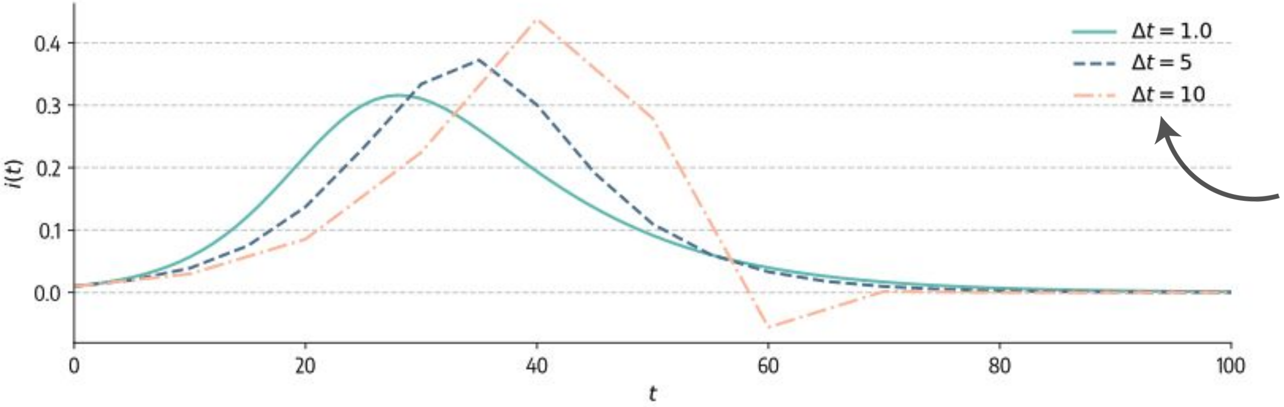
```python
def SIR_deterministic(beta : float,
                      mu : float,
                      N : int,
                      dt : float = 1.0,
                      I0_frac : float = 0.01,
                      T : int  = 100):
    """
    This function numerically integrates the system of ODEs defining the SIR model
    Parameters
    ----------
    - beta (float): transmission rate
    - mu (float): recovery rate
    - N (int): population size
    - dt (float, optional): integration step. Default is 1.0.
    - I0_frac (float, optional): initial fraction of infected. Default is 0.01 (1%).
    - T (int, optional): simulation length in steps. Default is 100.

    Returns
    -------
    - evolution of S, I, R compartments
    """
    I0 = int(N * I0_frac)
    S, I, R, t = [N - I0], [I0], [0], [0]
    for i in range(T):
        new_infected = (S[-1] * beta * I[-1] / N) * dt
        new_recovered = (I[-1] * mu) * dt
        S.append(S[-1] - new_infected)
        I.append(I[-1] + new_infected - new_recovered)
        R.append(R[-1] + new_recovered)
        t.append(t[-1] + dt)
    return np.array(S), np.array(I), np.array(R), np.array(t)
```

https://github.com/ngozzi/modeling-contagion/

Deterministic Simulation - Python Implementation (iv)

The choose of Δt can influence the results:



higher Δt lead to instability of solutions

- As a rule of thumb, smaller Δt lead to more accurate results but slower simulations
- More sophisticated techniques can be used to overcome the issue of the stability of the solutions, such as the Runge-Kutta or Jacobi methods

Stochastic Simulation - Python Implementation (i)

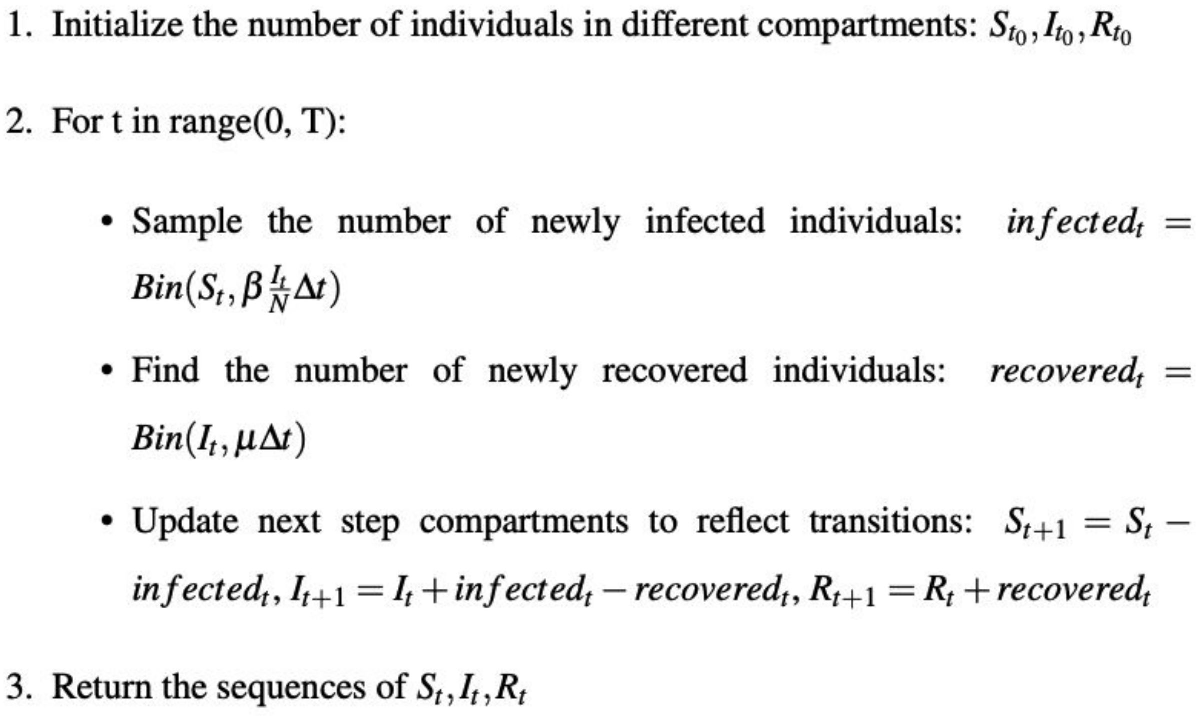
Stochastic simulations provide a framework for a more realistic representation of the dynamics of the system under study. Indeed, transitions among compartments are intrinsically probabilistic events:

- At each time step, the number of individuals transitioning among compartments (during Δt) is sampled from a probability distribution
- After the sampling step, the number of individuals in different compartments is updated to reflect the transitions.
- This process is repeated iteratively to get the time evolution of different compartments.

A common approach to represent the discrete and stochastic nature of transitions among compartments is through chain binomial processes.

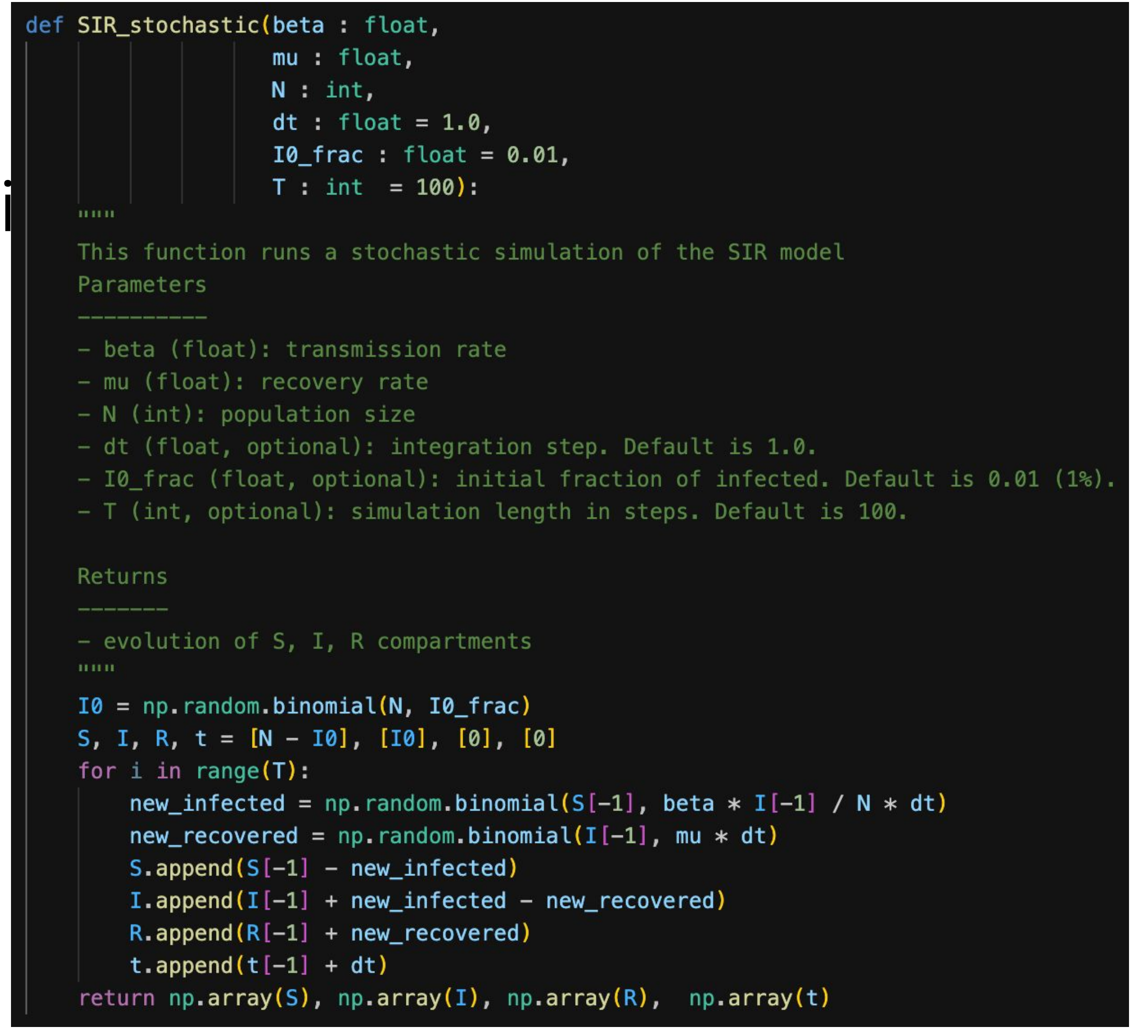
Stochastic Simulation - Python Implementation (ii)

SIR model algorithm:



1. Initialize the number of individuals in different compartments: $S_{t_0}, I_{t_0}, R_{t_0}$
2. For t in range(0, T):
   - Sample the number of newly infected individuals: $infected_t = Bin(S_t, \beta \frac{I_t}{N} \Delta t)$
   - Find the number of newly recovered individuals: $recovered_t = Bin(I_t, \mu \Delta t)$
   - Update next step compartments to reflect transitions: $S_{t+1} = S_t - infected_t$, $I_{t+1} = I_t + infected_t - recovered_t$, $R_{t+1} = R_t + recovered_t$
3. Return the sequences of S_t, I_t, R_t

Stochastic Simulation - Python Implementation (iii



```python
def SIR_stochastic(beta : float,
                   mu : float,
                   N : int,
                   dt : float = 1.0,
                   I0_frac : float = 0.01,
                   T : int  = 100):
    """
    This function runs a stochastic simulation of the SIR model
    Parameters
    ----------
    - beta (float): transmission rate
    - mu (float): recovery rate
    - N (int): population size
    - dt (float, optional): integration step. Default is 1.0.
    - I0_frac (float, optional): initial fraction of infected. Default is 0.01 (1%).
    - T (int, optional): simulation length in steps. Default is 100.

    Returns
    -------
    - evolution of S, I, R compartments
    """
    I0 = np.random.binomial(N, I0_frac)
    S, I, R, t = [N - I0], [I0], [0], [0]
    for i in range(T):
        new_infected = np.random.binomial(S[-1], beta * I[-1] / N * dt)
        new_recovered = np.random.binomial(I[-1], mu * dt)
        S.append(S[-1] - new_infected)
        I.append(I[-1] + new_infected - new_recovered)
        R.append(R[-1] + new_recovered)
        t.append(t[-1] + dt)
    return np.array(S), np.array(I), np.array(R),  np.array(t)
```

https://github.com/ngozzi/modeling-contagion/

Model Calibration

We have a model, we have data, so what? →**Model Calibration**

- Estimating the parameters of the mathematical model from the data
- Epidemic models depend on a range of parameters (β, μ, ...):
    - The values of some of them can be informed considering the literature
    - The remaining free parameters need to be determined through a statistical procedure based on the data the model is trying to reproduce

Approximate Bayesian Computation for Epidemic Models

- The main concept of Bayesian inference is the idea of updating beliefs with new evidence
- Within the Bayesian framework, the parameters of a model are considered as random variables characterized by probability distributions, rather than exact values
- These ideas are condensed in the famous Bayes theorem:

$$P(\theta|D) \propto P(\theta)P(D|\theta)$$

Approximate Bayesian Computation for Epidemic Models

- The main concept of Bayesian inference is the idea of updating beliefs with new evidence
- Within the Bayesian framework, the parameters of a model are considered as random variables characterized by probability distributions, rather than exact values
- These ideas are condensed in the famous Bayes theorem:

$$P(\theta|D) \propto \boxed{P(\theta)}P(D|\theta)$$

Prior distribution of model's parameters
This is what we decide (prior beliefs)

Approximate Bayesian Computation for Epidemic Models

- The main concept of Bayesian inference is the idea of updating beliefs with new evidence
- Within the Bayesian framework, the parameters of a model are considered as random variables characterized by probability distributions, rather than exact values
- These ideas are condensed in the famous Bayes theorem:

$$P(\theta|D) \propto P(\theta)\boxed{P(D|\theta)}$$

likelihood function

This is what we have to compute

Approximate Bayesian Computation for Epidemic Models

- The main concept of Bayesian inference is the idea of updating beliefs with new evidence
- Within the Bayesian framework, the parameters of a model are considered as random variables characterized by probability distributions, rather than exact values
- These ideas are condensed in the famous Bayes theorem:

$$\boxed{P(\theta|D)} \propto P(\theta)P(D|\theta)$$

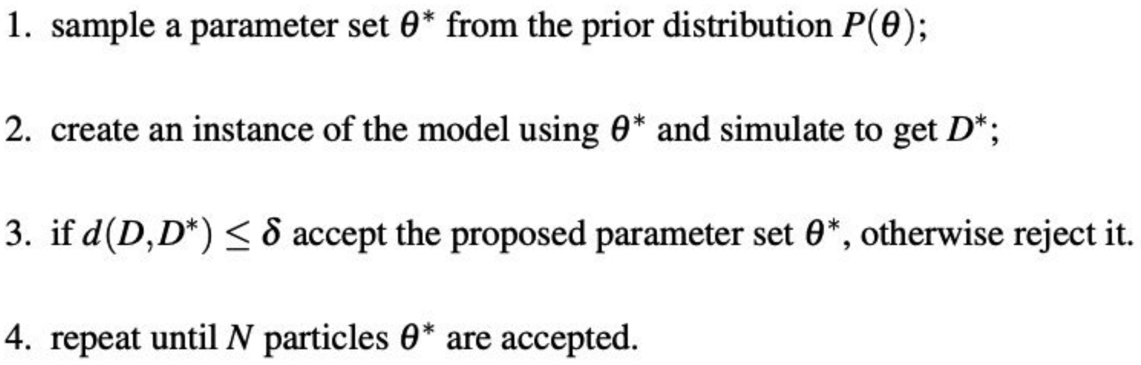
Posterior distribution of model's parameters
This is what we want to get

Approximate Bayesian Computation for Epidemic Models

- In practice, the Bayes theorem is rarely applied directly to get the expression of P(θ|D)
- Indeed, apart from trivial cases, it is hard to get an analytical expression for P(D|θ)
- This is where Approximate Bayesian Computation (ABC) techniques come into play
- The goal of ABC is to estimate the posterior distribution of the parameters without computing the likelihood function

A simple ABC technique: the Rejection Algorithm

This algorithm accepts proposed parameter values (also named *particles*) if a distance metric d between the real (D) and the model simulated (D*) data is less than or equal to a predefined threshold δ. In steps:



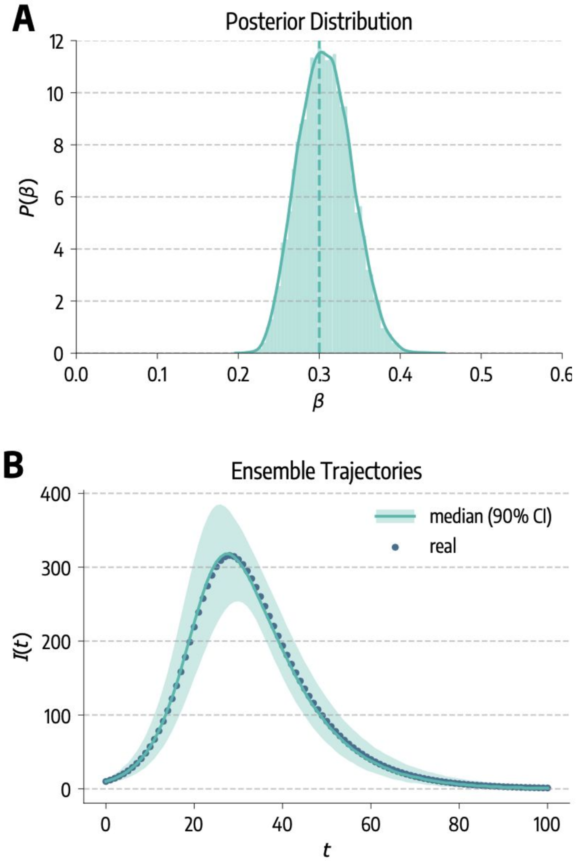
1. sample a parameter set θ^* from the prior distribution $P(\theta)$;
2. create an instance of the model using θ^* and simulate to get D^*;
3. if $d(D,D^*) \leq \delta$ accept the proposed parameter set θ^*, otherwise reject it.
4. repeat until N particles θ^* are accepted.

The N accepted samples are an approximation of the actual posterior distribution of model parameters

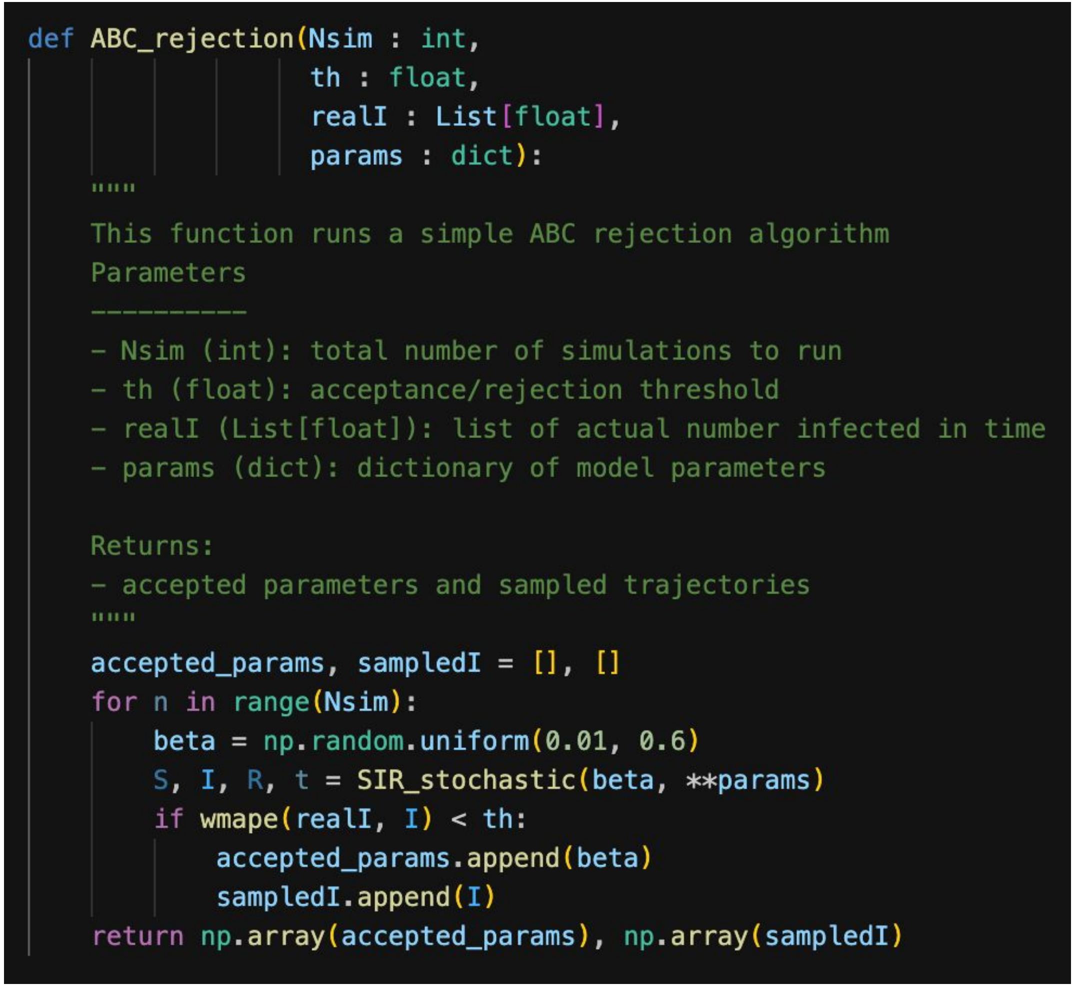
ABC Rejection Algorithm: Practical Example

We want to find the posterior distribution of β using a stochastic SIR model

- We set a uniform prior on the only free parameter β ∼ U(0.01,0.6)
- As distance metric, we use the weighted mean absolute percentage error on the number of infected at each time step t
- We set a threshold δ = 0.25 and we run 10,000 iterations of the ABC-rejection algorithm
- Other parameters are set to μ = 0.1, total number of individuals N = 1000, initial fraction of infected 1%



ABC Rejection Algorithm: Python Code



```python
def ABC_rejection(Nsim : int,
                  th : float,
                  realI : List[float],
                  params : dict):
    """
    This function runs a simple ABC rejection algorithm
    Parameters
    ----------
    - Nsim (int): total number of simulations to run
    - th (float): acceptance/rejection threshold
    - realI (List[float]): list of actual number infected in time
    - params (dict): dictionary of model parameters

    Returns:
    - accepted parameters and sampled trajectories
    """
    accepted_params, sampledI = [], []
    for n in range(Nsim):
        beta = np.random.uniform(0.01, 0.6)
        S, I, R, t = SIR_stochastic(beta, **params)
        if wmape(realI, I) < th:
            accepted_params.append(beta)
            sampledI.append(I)
    return np.array(accepted_params), np.array(sampledI)
```

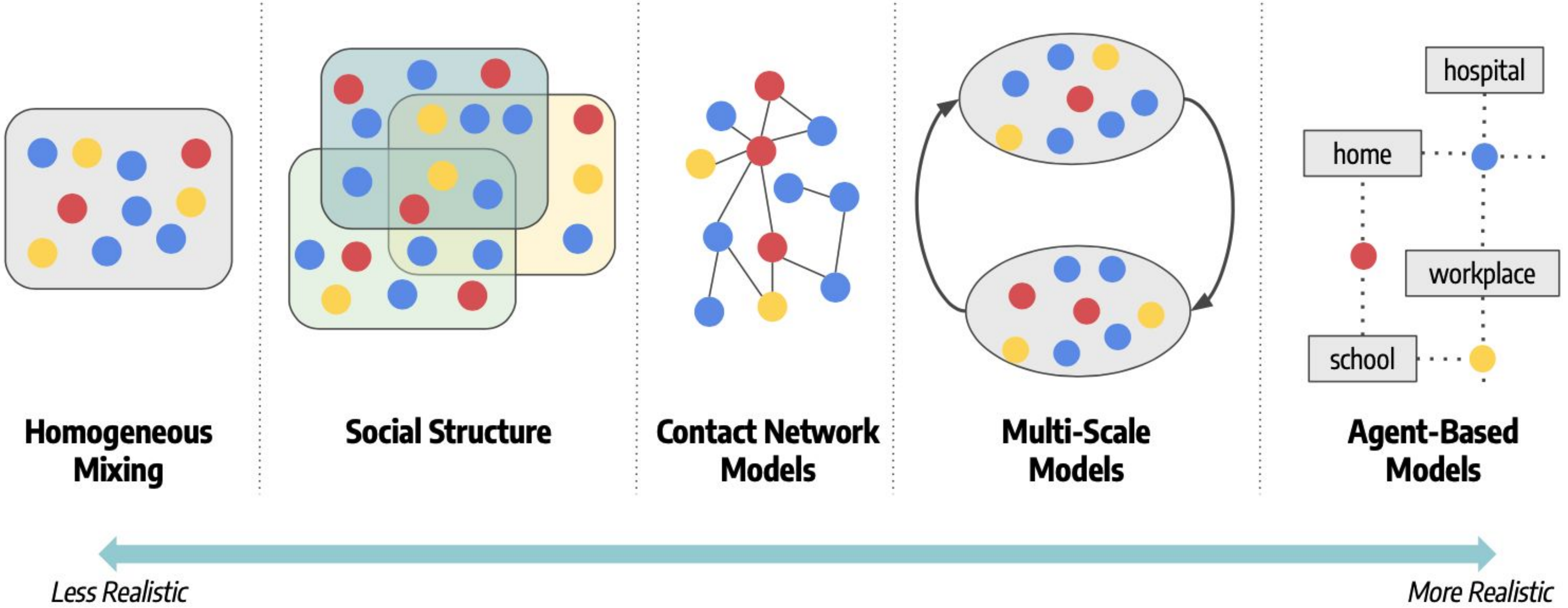
https://github.com/ngozzi/modeling-contagion/

Advanced ABC: The ABC-SMC Algorithm

- The simple rejection algorithm is easy to implement but slow to converge
- Indeed, the information from previous iterations is not integrated into the next ones
- The ABC-SMC algorithm solves this issue (SMC stands for Sequential Monte Carlo) by implementing iteratively the rejection algorithm
- In the initial step a high tolerance is used, this will lead to the exclusion of very unlikely parameters
- At next steps, the tolerance is lowered and the prior distribution will we constituted by the parameters accepted in the previous step (perturbed via a kernel function)

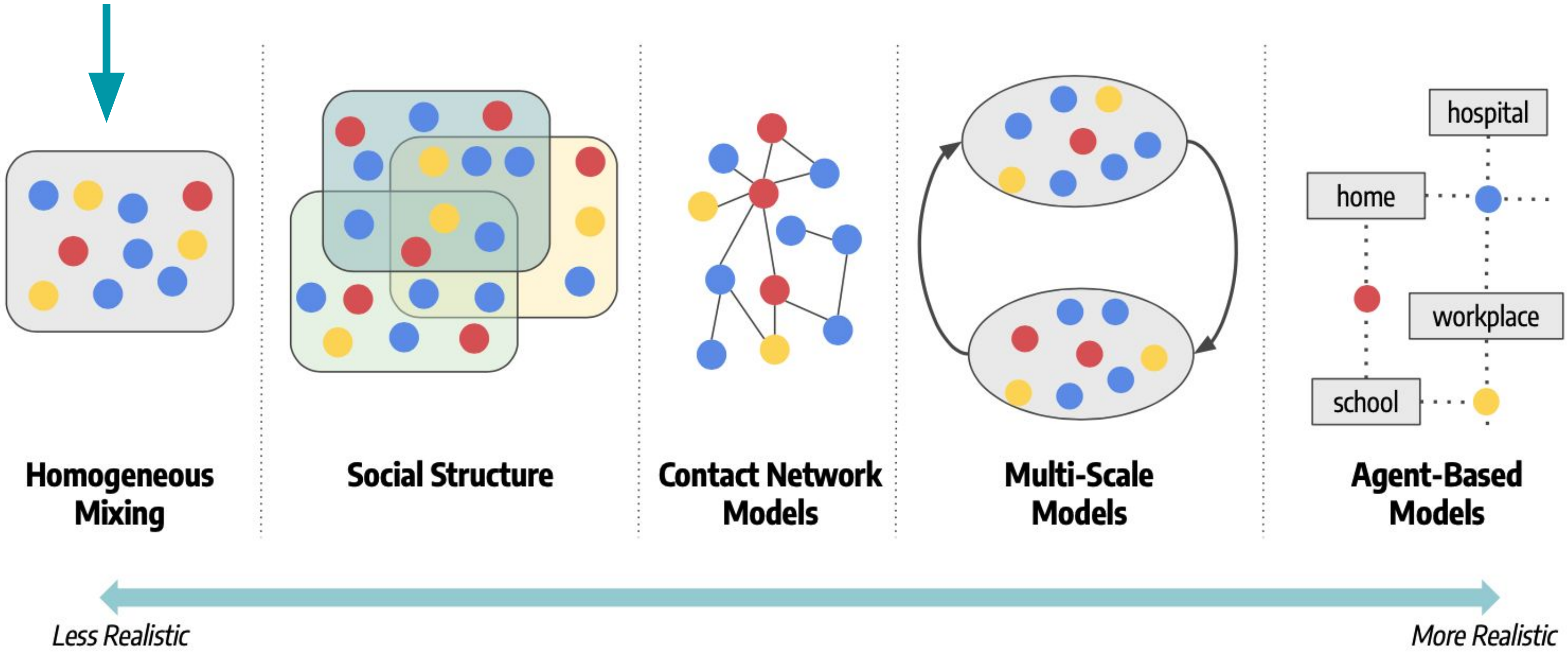
Beyond Homogeneous Mixing

Beyond Homogeneous Mixing



Plot adapted from *Vittoria Colizza, Marc Barthelemy, Alain Barrat, and Alessandro Vespignani. Epidemic modeling in complex realities*

Beyond Homogeneous Mixing



Homogeneous Mixing

Social Structure

Contact Network Models

Multi-Scale Models

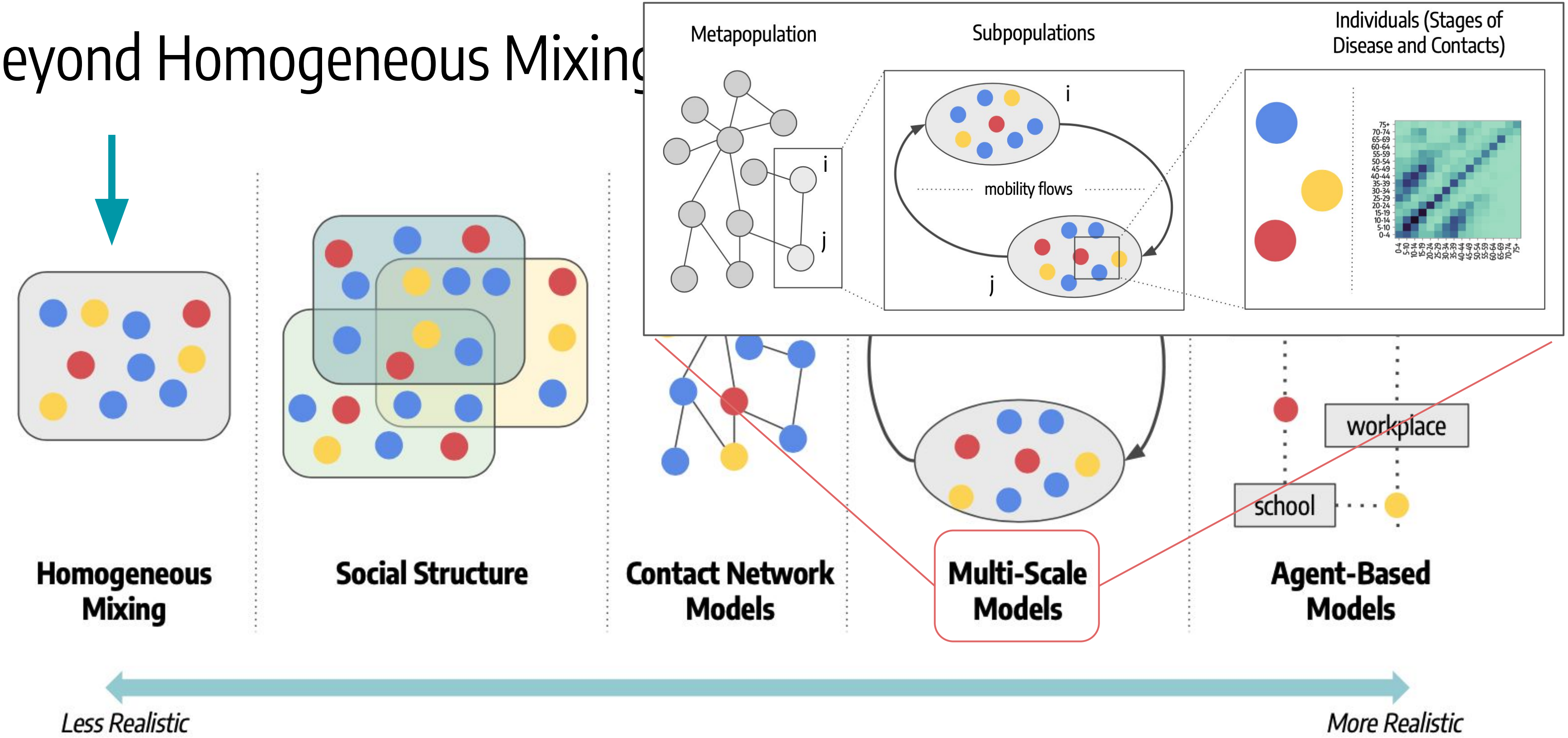
Agent-Based Models

Less Realistic

More Realistic

Plot adapted from *Vittoria Colizza, Marc Barthelemy, Alain Barrat, and Alessandro Vespignani. Epidemic modeling in complex realities*

Beyond Homogeneous Mixing



Homogeneous Mixing

Social Structure

Contact Network Models

Multi-Scale Models

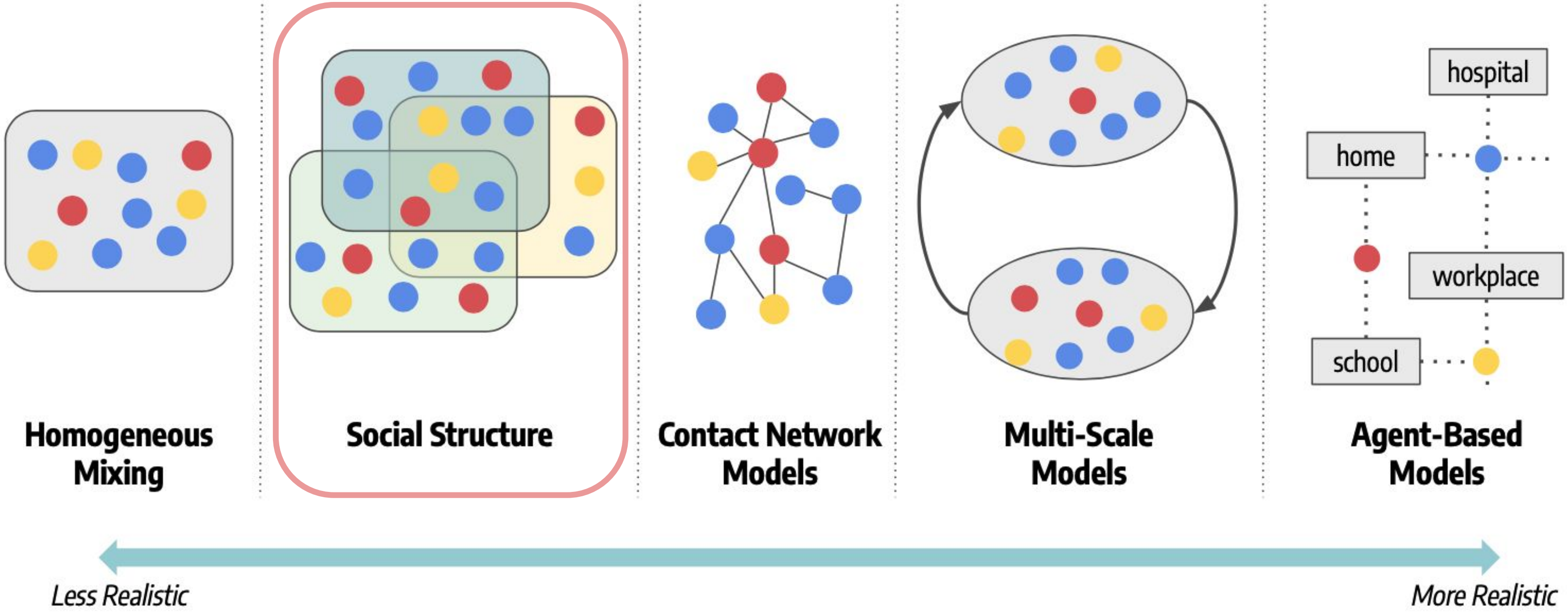
Agent-Based Models

Less Realistic

More Realistic

Plot adapted from *Vittoria Colizza, Marc Barthelemy, Alain Barrat, and Alessandro Vespignani. Epidemic modeling in complex realities*

Beyond Homogeneous Mixing



Homogeneous Mixing

Social Structure

Contact Network Models

Multi-Scale Models

Agent-Based Models

Less Realistic

More Realistic

Plot adapted from *Vittoria Colizza, Marc Barthelemy, Alain Barrat, and Alessandro Vespignani. Epidemic modeling in complex realities*

Adding a Social Structure to Compartmental Models

- Consider a society formed by two groups of individuals (A and B)
- The two groups mix at different rates (homophily)
- Contacts can be represented with a matrix:

$$C = \begin{pmatrix} C_{AA} & C_{AB} \\ C_{BA} & C_{BB} \end{pmatrix}$$

Adding a Social Structure to Compartmental Models

- Consider a society formed by two groups of individuals (A and B)
- The two groups mix at different rates (homophily)
- Contacts can be represented with a matrix:

$$C = \begin{pmatrix} C_{AA} & \boxed{C_{AB}} \\ C_{BA} & C_{BB} \end{pmatrix}$$

Number of contacts that an individual of group A has, on average, with an individual of group B (in a day)

Adding a Social Structure to Compartmental Models

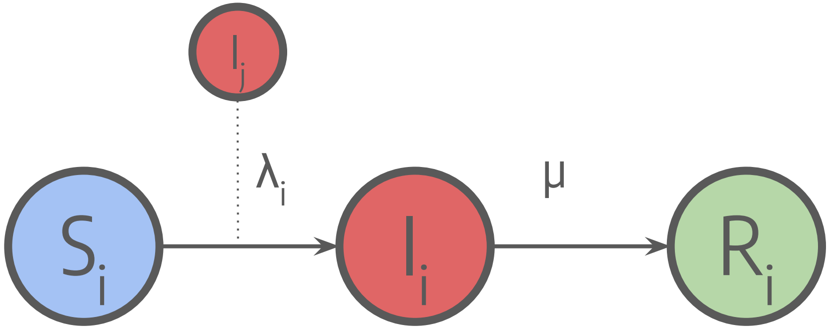
- Consider a society formed by two groups of individuals (A and B)
- The two groups mix at different rates (homophily)
- Contacts can be represented with a matrix:

Number of contacts that an individual of group A has, on average, with an individual of group B (in a day)

$$C = \begin{pmatrix} C_{AA} & \boxed{C_{AB}} \\ C_{BA} & C_{BB} \end{pmatrix}$$

Is C symmetric? (i.e., is $C_{AB} = C_{BA}$?)

Extending the SIR model to a complex social structure



What is the expression of the Force of Infection (λ_i) now?

- We were starting from:

$$\beta C \frac{I}{N}$$

- But now C is a matrix and we have K groups:

$$\lambda_i = \beta \sum_{j=1}^{K} C_{ij} \frac{I_j}{N_j}$$

Extending the SIR model to a complex social structure

The equation system will now have *3 x K* equations (3 compartments, K groups):

$$\frac{dS_i}{dt} = -\beta \sum_{j=1}^{K} C_{ij} \frac{I_j}{N_j} S_i$$

$$\frac{dI_i}{dt} = \beta \sum_{j=1}^{K} C_{ij} \frac{I_j}{N_j} S_i - \mu I_i$$

$$\frac{dR_i}{dt} = \mu I_i$$

Extending the SIR model to a complex social structure

The equation system will now have *3 x K* equations (3 compartments, K groups):

$$\frac{dS_i}{dt} = -\beta \sum_{j=1}^{K} C_{ij} \frac{I_j}{N_j} S_i$$

$$\frac{dI_i}{dt} = \beta \sum_{j=1}^{K} C_{ij} \frac{I_j}{N_j} S_i - \mu I_i$$

$$\frac{dR_i}{dt} = \mu I_i$$

The R_0 is now:

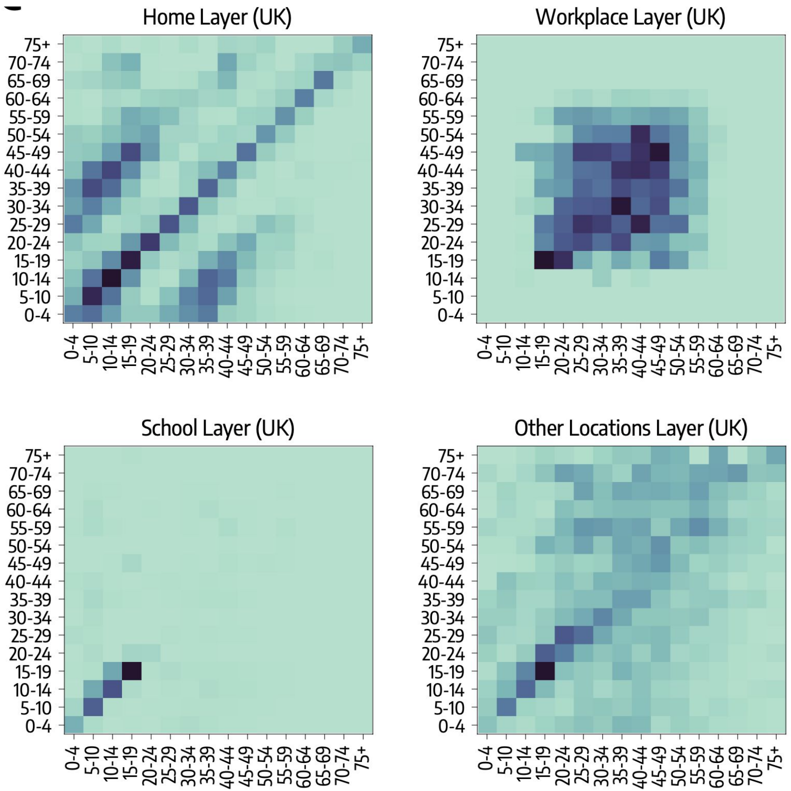
$$R_0 = \rho(\tilde{C}) \frac{\beta}{\mu}$$

Where ρ(·) indicates the spectral radius (i.e., the largest eigenvalue) and “C tilde” is the contact matrix weighted by the number of individuals in different groups

Example: Age Structure

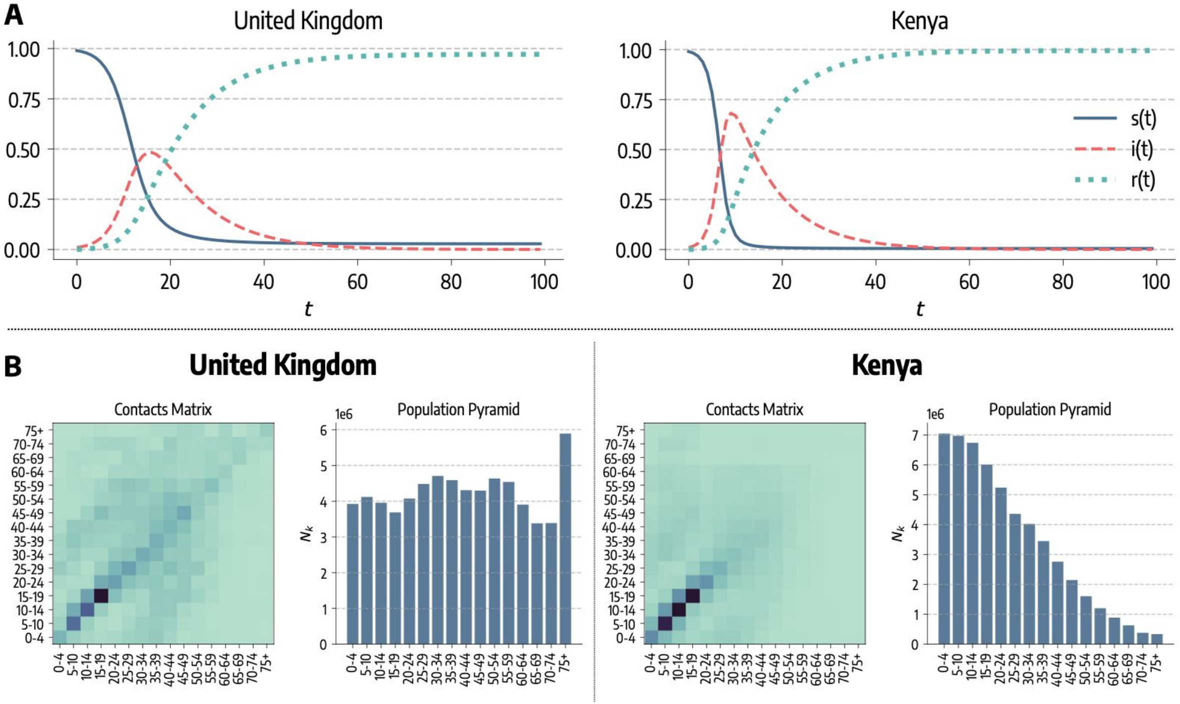
A common approach is to introduce a contact matrix describing contacts among different age groups:

- Contacts among age groups correlates with the setting of the contact
- Age is also generally an important risk factor (disease)



Example: Age Structure

- Age distribution and contacts significantly vary among context
- We run SIR model with same parameters but different demographics (UK and Kenya)



Conclusions

- Epidemics are social phenomena
- Three ingredients: individuals, something that spread, interactions
- Build on simple models
- Implementation is as important as theory (simulation, calibration)