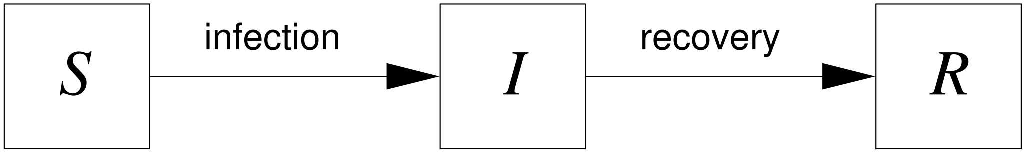
SIR Model for Spread of Infection

Compartmental model: Susceptibles, Infectives, Recovereds



$$dS/dt = -\beta SI/N$$

$$dI/dt = \beta SI/N - \gamma I$$

$$dR/dt = \gamma I$$

Ignore births and deaths (e.g. short-lived outbreak)

"Standard incidence" term $\beta SI/N$ β : "transmission parameter"

"well-mixed" population

Assume constant per-capita recovery rate of γ

$1/\gamma$ is average duration of infectiousness

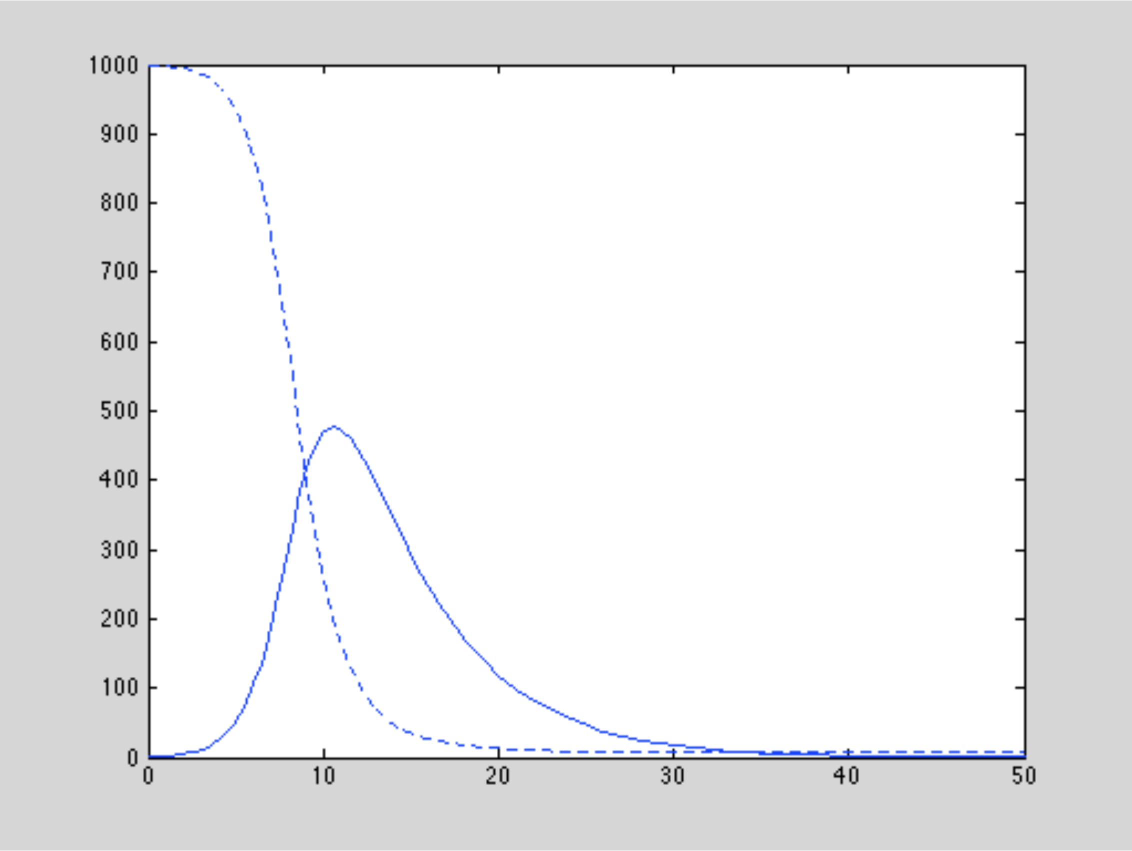
Note: $S + I + R = N$ (constant), so need only worry about S and I

Behavior of SIR Model

Behavior is governed by the value of the ratio $R_0 = \beta/\gamma$

Outbreak can occur if $R_0 > 1$, cannot occur if $R_0 < 1$

$R_0 > 1$ plot:



$\beta = 1$, $\gamma = 0.2$, $N = 1000$
$S(0) = 999$, $I(0) = 1$

$S(t)$: dashed line

$I(t)$: solid line

Behavior of SIR Model

Behavior is governed by the value of the ratio $R_0 = \beta/\gamma$

Outbreak can occur if $R_0 > 1$, cannot occur if $R_0 < 1$

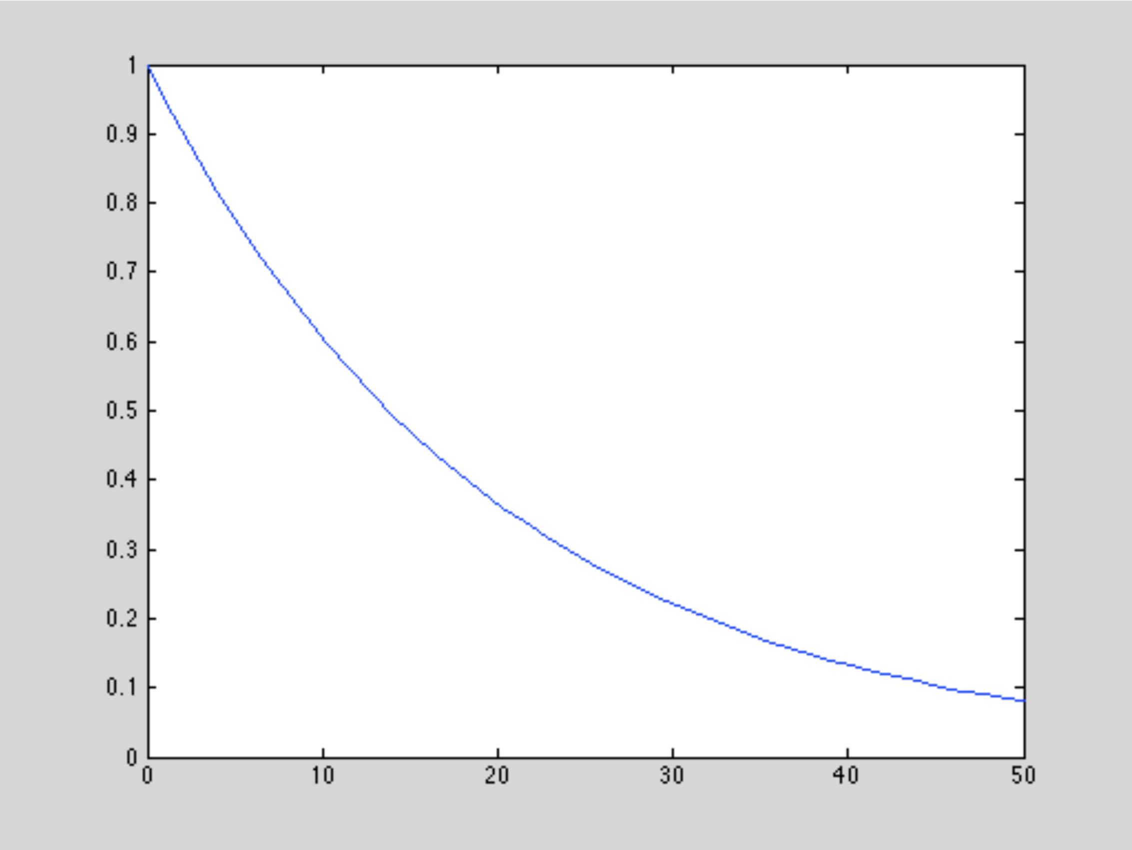
$R_0 < 1$ plot:

$\beta = 0.15$, $\gamma = 0.2$, $N = 1000$
$S(0) = 999$, $I(0) = 1$

$S(t)$: not shown (remains close to 999)

$I(t)$: solid line

note different scale on vertical axis



Simple Analysis of SIR Model in Terms of R_0

Consider dI/dt :

$$\begin{aligned} \frac{dI}{dt} &= \beta SI/N - \gamma I \\ &= \gamma \left(\frac{\beta}{\gamma} \frac{S}{N} - 1 \right) I \\ &= \gamma \left(R_0 \{ S/N \} - 1 \right) I \qquad (*) \end{aligned}$$

per-capita transmission maximized when $S \approx N$:

$$\frac{dI}{dt} = \gamma \left(R_0 - 1 \right) I$$

I increases if $R_0 > 1$, decreases if $R_0 < 1$

R_0 : basic reproductive number = $\beta \times 1/\gamma$ = β x (av. duration of infection)

average number of secondary infections caused by an infectious individual when the population is almost entirely susceptible

Epidemiological Importance of R_0

Can control infection if we can reduce R_0 ($= \beta/\gamma$) below one

(e.g. reduce β or increase γ)

Alternatively, from (*) on previous slide, if we can reduce S/N below $1/R_0$

e.g. vaccinate $p_c = 1 – 1/R_0$ or more of the population

Control is more difficult for a highly infectious agent (e.g. measles, with $R_0 \approx 15\text{-}18$) than for a less infectious agent (e.g. smallpox with $R_0 \approx 5\text{-}7$)

Critical for epidemiologists to estimate R_0 (i.e. β and γ), preferably also getting some idea of reliability of estimate(s)

Typical method used: fit model to some dataset

SIR Model : Forward Simulation

Nonlinearity of the transmission term means we cannot find an analytic solution of the model for *S* and *I* in terms of time

Numerically integrate (simulate) model in MATLAB, given a set of parameters and initial values for *S* and *I*

We shall use the `ode45` routine in MATLAB

MATLAB works with vectors, so we shall use the first element (e.g. `y(1)`) to denote *S* and the second (e.g. `y(2)`) to denote *I*

ode45

```
[t,y]=ode45(@odefun,tspan,y0,options,pars);
```

`odefun`	the name of the function that gives the right sides of our differential equations (replace "odefun" with something more descriptive, but keep "@")

`tspan`	vector that specifies the interval of times over which to integrate:
	`tspan = [t_initial, t_final]`
	or a vector of times at which we wish to obtain output :
	`tspan = [t_initial, t1, t2, … , t_final]`

`y0`	column vector of initial states (i.e. initial conditions) : `y0 = [ S0 ; I0 ]`
`options`	options for the ODE solver, e.g. solution tolerances use [ ] for no options; see `odeset` for information on options
`pars`	a vector of parameter values that gets passed to `odefun`
`t`	(returned) column vector of times at which output is given
`y`	(returned) matrix of numerically calculated values of state variables over time

each row refers to a different time point, each column to a different state variable
e.g. `y(1,:)` are initial states, `y(end,:)` final states,
`y(:,2)` is a column vector of I values at all times — this is what we want to make an $I(t)$ vs t plot

`odefun`

```
function f = odefun(t,y,pars)
```

Function `odefun` returns the entries of the right sides of the differential equations, *f* (*t*,**y**), as a column vector

`t`	(scalar) value of time at which to evaluate *f*
`y`	column vector containing values of state variables
`pars`	a vector of parameter values that gets passed to `odefun`

```
function f = sir_rhs(t,y,pars)

    f=zeros(2,1);

    beta=pars(1);
    gamma=pars(2);
    N=pars(3);

    S=y(1);
    I=y(2);

    f(1)=-beta*S*I/N;
    f(2)=beta*S*I/N-gamma*I;
end
```

need to return a column vector

could eliminate a number of these lines if we worked with y(1), pars(1) etc in the f(1) and f(2) lines

SIR Model Simulation

```
function sir_simulation

    beta=1.0;
    gamma=1.0/5.0;    % five day infectious period
    N=1000.0;

    pars=[beta,gamma,N];

    tspan=[0,50];      % simulate for 50 days

    y0=[999;1];       % one initial infective

    [t,y]=ode45(@sir_rhs,tspan,y0,[],pars);

    plot(t,y(:,2));   % plot prevalence of infection over time

end

function f = sir_rhs(t,y,pars)
    f=zeros(2,1);
    f(1)=-pars(1)*y(1)*y(2)/pars(3);
    f(2)=pars(1)*y(1)*y(2)/pars(3)-pars(2)*y(2);
end
```