P8477 EPI MODELING FOR INFECTIOUS DISEASES

Lab 2: SIR simulations

House Keeping Issues

- Office hours
- Background survey
  - Some time next week, for group project

Lab Learning Objectives

- How to simulate an epidemic using the SIR
- Test the epidemic threshold phenomenon
- Test the exponential growth period
- Test “epidemic burnout”
- Final epidemic size v. R0 (example in excel on courseworks)

Ordinary Differential Equations (ODE)

- For simple ODEs, we can solve them analytically, by integration, e.g.:

$$\frac{dy}{dx} = x^2$$

$$dy = x^2 dx$$

$$\int dy = \int x^2 dx$$

$$y = \frac{x^3}{3} + C$$

To find C, we need the inital condition:

$$x = 0, y = 0 => C = 0$$

- For more complicated ODEs, no analytic solution is available, so we do numerical integration

$$\frac{dS}{dt} = -\frac{\beta IS}{N}$$

$$\frac{dI}{dt} = \frac{\beta IS}{N} - \gamma I$$

ODE solvers

- There are many algorithms for numerical integration
  - Euler’s method
  - Runge-Kutta (ODE in R or ODE45 in Matlab)
- In R, use the package ‘deSolve’ (note: case sensitive)
- Function:
  ode(y, times, func, parms, method = c("lsoda", "lsode", "lsodes", "lsodar", "vode", "daspk”, "euler", "rk4", "ode23", "ode45", "radau", "bdf", "bdf_d", "adams", "impAdams", "impAdams_d", "iteration"), ...)
  - y: the initial condition (e.g. S0, I0, R0 in the SIR model)
  - times: the variable with which we are differentiating
  - func: the function, i.e. the ode equations (e.g., the SIR model)
  - parms: parameters in the ode equations (e.g., β)

Toy example

```r
library(deSolve)

## PRE-LAB: TRY OUT THE ODE SOLVER
## SIMPLE ODE: dy/dx=x^2
myfunc=function(x,y,parms){
    dy=x^2
    list(dy)
}
xs=seq(0,10,by=.1)
state=c(y=0);
out=ode(y=state,times=xs,func=myfunc,parms=NULL)

y.true=1/3*xs^3

## check if the ode is working
plot(out[,1],out[,2],xlab='x',ylab='y',pch=1)
lines(xs,y.true,lwd=2,col='red')

```

$$\frac{dy}{dx} = x^2$$

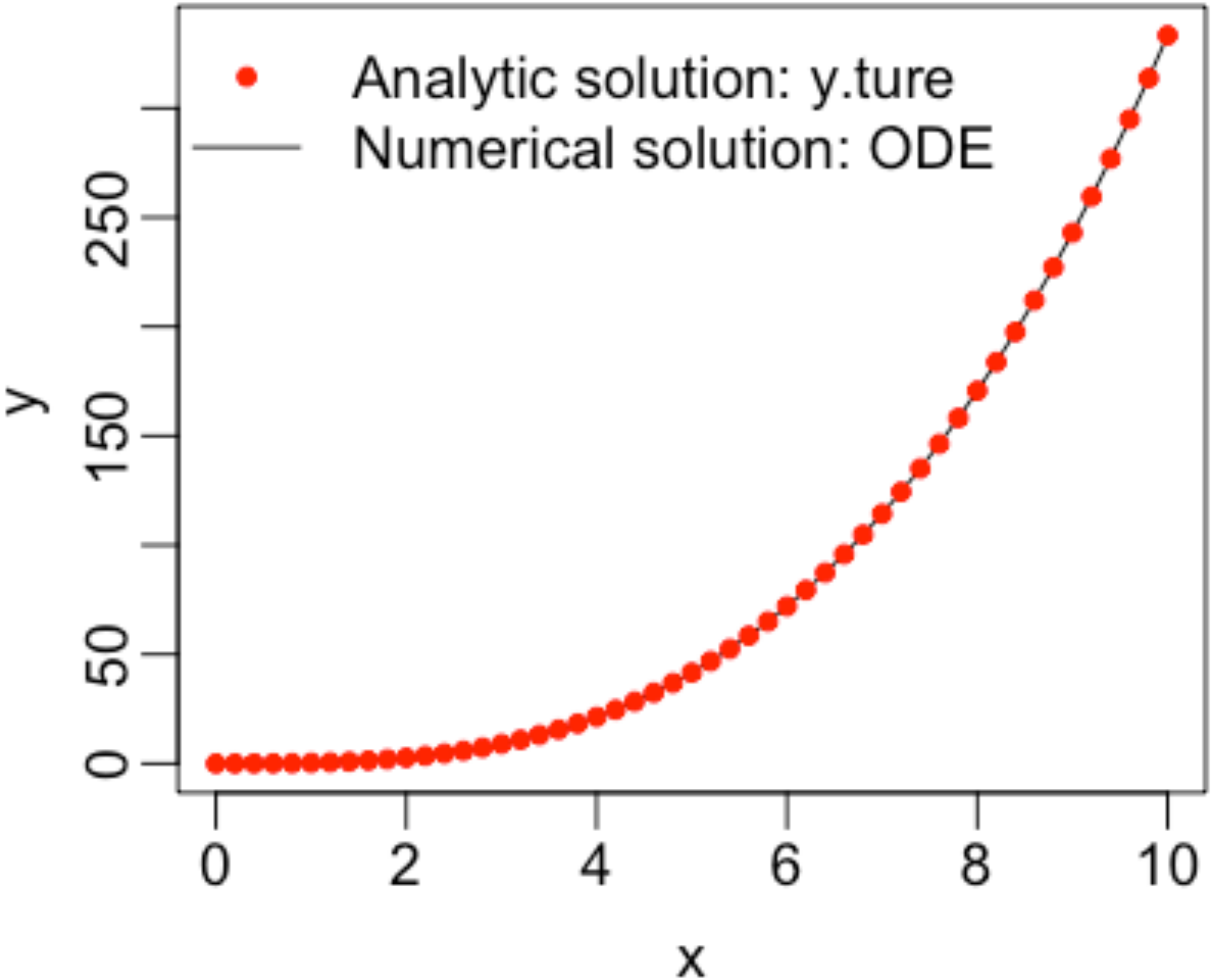
Note: we only enter ‘dy~~/dx~~’ on the LHS (i.e., omit ‘/dx’)

Initial conditions:

$$x = 0, y = 0$$

Note: you have to run the ‘ode’ function to get the solution

Toy Example



Analytic solution: y.ture
Numerical solution: ODE
0
50
150
250
y
0
2
4
6
8
10
x

SIR Model

```r
## try the SIR
SIR=function(t,state,parameters){
  with(as.list(c(state,parameters)),{
    # rate of change
    dS=-beta*S*I/N;
    dI=beta*S*I/N-gamma*I;
    # return the rate of change
    list(c(dS,dI))
  }) # end with(as.list...)
}

# specify initial conditions/parameters
N=1e5; # population size
I0=10; # initial No. of Infectious pe
S0=N-I0;  # initial No. of Susceptibl
state=c(S=S0,I=I0);  # store the initial conditions I & S together
parameters=c(beta=.5,gamma=.3);  # store the model parameters

times=seq(0,100,by=1);
sim=ode(y=state,times=times,func=SIR,parms=parameters);
```

$$\frac{dS}{dt} = -\frac{\beta IS}{N}$$

$$\frac{dI}{dt} = \frac{\beta IS}{N} - \gamma I$$

Note: the variables in the (return) list should match with the order of the equations

~~list(c(dI, dS))~~

initial conditions/parameters: The order of S, I, etc. in ‘state’ should match with the ODE

SIR: Model Output

```
sim=ode(y=state,times=times,func=SIR,parms=parameters);

View(sim)  # to see what's in the model output 'sim'
```

Lab2_SIR_for_students.R | sim

Filter

	time	S	I
1	0	99990.00	10.00000
2	1	99984.47	12.21326
3	2	99977.71	14.91591
4	3	99969.45	18.21594
5	4	99959.38	22.24507
6	5	99947.07	27.16385
7	6	99932.05	33.16803
8	7	99913.70	40.49597
9	8	99891.32	49.43788
10	9	99863.99	60.34680
11	10	99830.66	73.65169
12	11	99789.99	89.87340
13	12	99740.39	109.64328
14	13	99679.92	133.72533

Use the View () function to see model output

SIR Model

- Demo: changes in S, I, R over time

Pre-lab:

i. Try code up and run the toy example dy/dx=x^2 (x=0,y=0)

ii. Code up the SIR function and run it with the following conditions and parameters:

N=1e5; I0=10; S0=N-I0;

beta=0.5; gamma=0.3

times=seq(0,100,by=1)

iii. Plot the outputs I and S vs. time from question ii

Lab Report Questions

1. The SIR model includes 3 equations, but in the SIR code, we only use 2 equations. How do we find R (# recovered) in Eqn3?

```
SIR=function(t,state,parameters){
  with(as.list(c(state,parameters)),{
    # rate of change
    dS=-beta*S*I/N;
    dI=beta*S*I/N-gamma*I;
    # return the rate of change
    list(c(dS,dI))
  }) # end with(as.list...)
}
```

$$\frac{dS}{dt} = -\frac{\beta IS}{N} \quad (1)$$

$$\frac{dI}{dt} = \frac{\beta IS}{N} - \gamma I \quad (2)$$

$$\frac{dR}{dt} = \gamma I \quad (3)$$

[Q1] Compute and plot R over time for the following epidemic setting. What is the final epidemic size (i.e. R at the end of the epidemic)?

N=1e5; I0=10; S0=N-I0;   beta=0.5; gamma=0.3;

times=seq(0,365,by=1); # run for a year

Hint: Method 1: S+I+R=? or

Method 2: Can we add Eqn 3 in the SIR code to compute R?

Lab Report Questions

2. To test the epidemic threshold, try the following exercise: Use the same initial state conditions as in #1, and try different parameters:

(1) beta=0.5; gamma=0.3

(2) beta=0.5; gamma=0.4

(3) beta=0.5; gamma=0.5

(4) beta=0.5; gamma=0.6

[Q2] What are the differences among the simulations? Explain what you see.

Hint: think about R0 (how it is related to beta and gamma, and R0 vs the epidemic threshold)

Lab Report Questions

- 3. Test the exponential period

$$\frac{dI}{dt} = \left(\frac{\beta S}{N} - \gamma\right) I \approx (\beta - \gamma) I$$

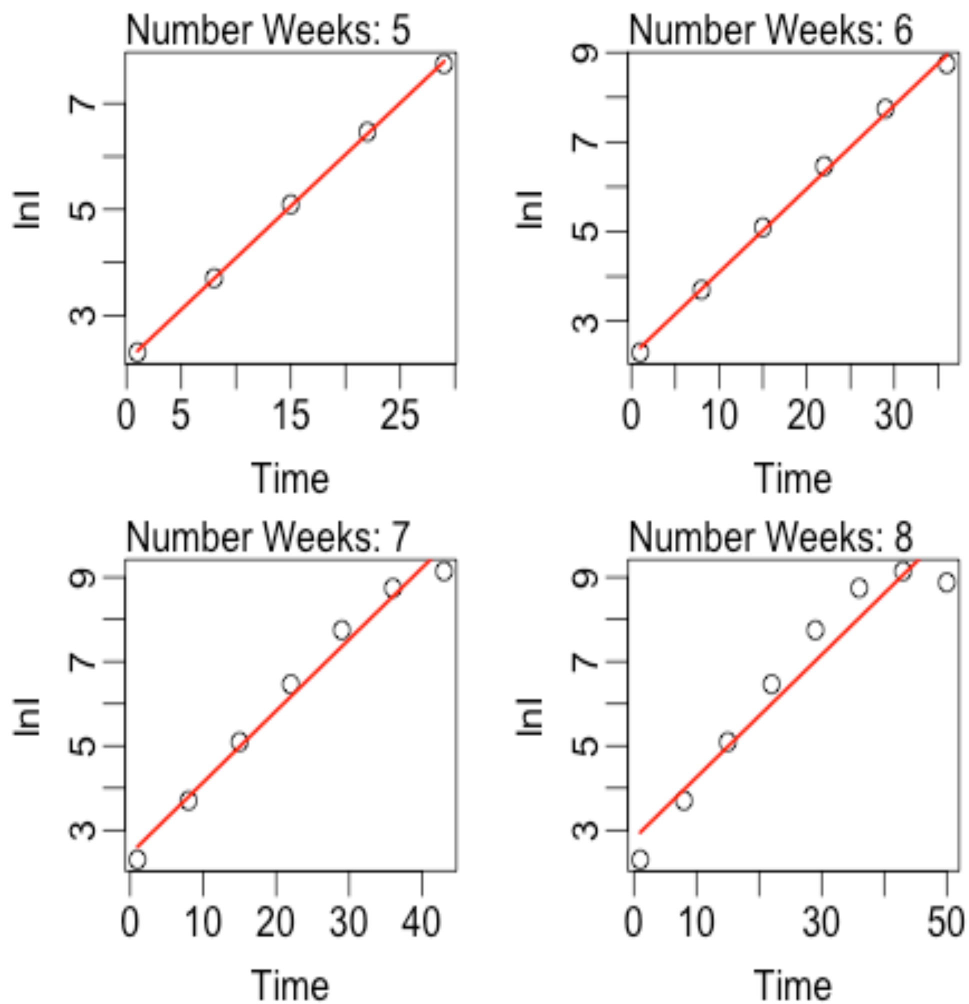
$$dI = (\beta - \gamma) I dt$$

$$\int_{I(0)}^{I(t)} \frac{dI}{I} = \int_{0}^{t} (\beta - \gamma) dt$$

$$ln \frac{I(t)}{I(0)} = (\beta - \gamma) t$$

$$\underline{\ln(I(t))} = \ln(I(0)) + \underline{(\beta\text{-}\gamma) t}$$

$lnI \propto t \rightarrow Exponential\ growth$



Lab Report Questions

3. Test the exponential period

Run the SIR model using the conditions and parameters below:

N=1e5; I0=10; S0=N-I0; beta=0.5; gamma=0.3

times=seq(0,100,by=1) # in days

[Q3] Plot log(I) vs. t (try t=3 to 8 weeks) and see when the exponential period ends.

[Q4] Fit log(I) to t *over the exponential period*, and find the slope of the linear regression. What does the slope represent?

Hint: function for linear regression in R: lm()

e.g.: x=1:10; y=3*x;

fit=lm(y~x)

To extract the slope: slope=fit$coeff[2]

Lab Report Question #4

```
##################################################################################
N=1e5; I0=10; S0=N-I0;
state=c(S=S0,I=I0);
parameters=c(beta=.5,gamma=.3);

times=seq(0,100,by=1);
sim=ode(y=state,times=times,func=SIR,parms=parameters);
s=sim[,'S']/N
i=sim[,'I']/N

WkExp=4; # CHANGE THE NUMBER OF WEEK HERE
Iexp=sim[seq(1,length=WkExp,by=7),'I'];
# NOTE: THE TIME STEP IN THE SIMULATION IS DAY, BUT WE ARE LOOKING AT WEEK HERE
# SO WE USE 'seq(1,length=WkExp,by=7)' TO EXTRACT THE CORRESPONDING DATE FOR EACH
lnI=log(Iexp); # TAKE THE LOG
tt=seq(1,length=WkExp,by=7)
fit=lm(lnI~tt) # LINEAR REGRESSION
slope=fit$coeff[2] ## extract the slope for the linear regression
```

seq(1,length=4,by=7)?

Lab Report Questions

4. Epidemic burnout

Set the final time to 600 days and run the model with the following conditions and parameters:

N=1e5; I0=10; S0=N-I0;

beta=0.5; gamma=0.3

times=seq(0,600,by=1)

[Q5] Show your results and check the finial S and I (is S=0? is I=0?). Why does the epidemic die out eventually?

hint: tail(X,1): print the last row of X