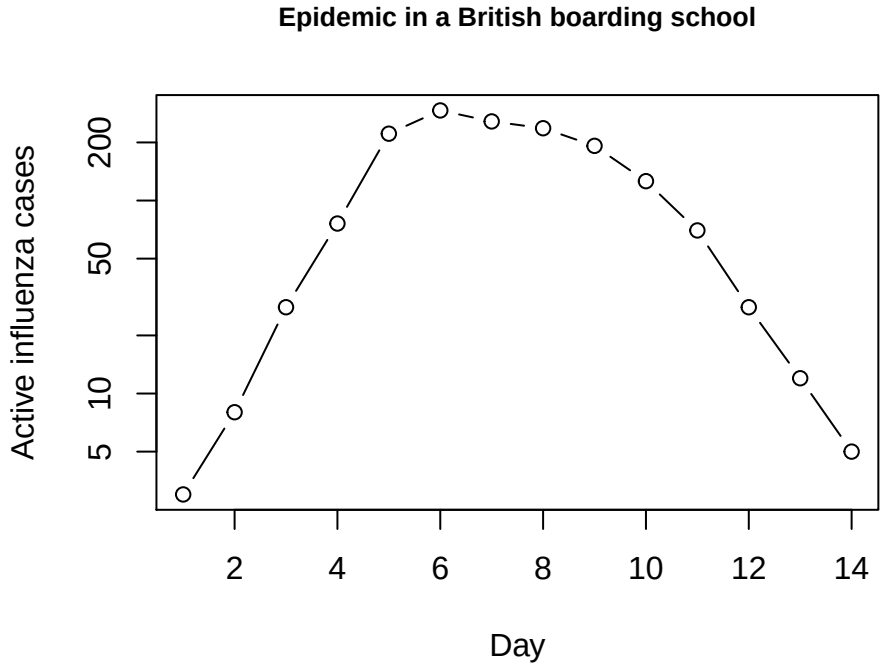
Estimation*

John M. Drake & Pejman Rohani

1 Estimating R_0

So far in this class we have focused on the *theory* of infectious disease. Often, however, we will want to apply this theory to particular situations. One of the key applied problems in epidemic modeling is the estimation of R_0 from outbreak data. In this session, we study two methods for estimating R_0 from an epidemic curve. As a running example, we will use the data on influenza in a British boarding school.

```
> load('data.RData')     #load the data and plot flu cases
> plot(flu,type='b',log='y',main='Epidemic in a British boarding school', cex.main=0.85,
+  xlab='Day', ylab='Active influenza cases')
```



2 Estimating R_0 from the final outbreak size

Our first approach is to estimate R_0 from the final outbreak size. Although unhelpful at the early stages of an epidemic (before the final epidemic size is observed), this method is nonetheless a useful tool for *post hoc* analysis. The method is general and can be motivated by the following argument (Keeling and Rohani 2007): First, we assume that the epidemic is started by a single infectious individual in a completely susceptible population. On average, this individual infects R_0 others. The probability a particular individual escaped infection is therefore $e^{-R_0/N}$. If Z individuals have been infected, the probablility of an individual escaping infection from all potential sources is $e^{-ZR_0/N}$. It follows that at the end of the epidemic a proportion $R(\infty) = Z/N$ have been infected and the fraction remaining susceptible is $S(\infty) = e^{-R(\infty)R_0}$, which is equal to $1 - R(\infty)$, giving

$$1 - R(\infty) - e^{-R(\infty)R_0} = 0 \tag{1}$$

Rearranging, we have the estimator

$$\hat{R_0} = \frac{log(1 - Z/N)}{-Z/N}, \tag{2}$$

which, in this case, evaluates to $\frac{log(1-512/764)}{-512/764} = 1.655$.

Exercise 1. This equation shows the important one-to-one relationship between R_0 and the final epidemic size. Plot the relationship between the total epidemic size and R_0 for the complete range of values between 0 and 1.

3 Linear approximation

The next method we introduce takes advantage of the fact that during the early stages of an outbreak, the number of infected individuals is given approximately as $Y(t) \approx Y_0 e^{((R_0-1)(\gamma+\mu)t)}$. Taking logarithms of both sides, we have $log(Y(t)) \approx log(Y_0) + (R_0 - 1)(\gamma + \mu)t$, showing that the log of the number of infected individuals is approximately linear in time with a slope that reflects both R_0 and the recovery rate.

This suggests that a simple linear regression fit to the first several data points on a log-scale, corrected to account for γ and μ, provides a rough and ready estimate of R_0. For flu, we can assume $\mu = 0$ because the epidemic occurred over a time period during which natural mortality is negligible. Further, assuming an infectious period of about 2.5 days, we use $\gamma = (2.5)^{-1} = 0.4$ for the correction. Fitting to the first four data points, we obtain the slope as follows.

```
> model<-lm(log(flu[1:4])~day[1:4],data=flu);  #fit a linear model
> summary(model)         #summary statistics for fit model


Call:
lm(formula = log(flu[1:4]) ~ day[1:4], data = flu)

Residuals:
       1        2        3        4
 0.03073 -0.08335  0.07450 -0.02188
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.02703    0.10218  -0.265  0.81611
day[1:4]     1.09491    0.03731  29.346  0.00116 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.08343 on 2 degrees of freedom
Multiple R-squared:  0.9977,        Adjusted R-squared:  0.9965
F-statistic: 861.2 on 1 and 2 DF,  p-value: 0.001159

> slope<-coef(model)[2]  #extract slope parameter
> slope                  #print to screen

day[1:4]
1.094913
```

Rearranging the linear equation above and denoting the slope coefficient by $\hat{\beta}_1$ we have the estimator $\hat{R}_0 = \hat{\beta}_1/\gamma + 1$ giving $\hat{R}_0 = 1.094913/0.4 + 1 \approx 3.7$.

Exercise 2. Our estimate assumes that boys remained infectious during the natural course of infection. The original report on this epidemic indicates that boys found to have syptoms were immediately confined to bed in the infirmary. The report also indicates that only 1 out of 130 adults at the school exhibited any symptoms. It is reasonable, then, to suppose that transmission in each case ceased once he had been admitted to the infirmary. Supposing admission happened within 24 hours of the onset of symptoms. How does this affect our estimate of R_0? Twelve hours?

Exercise 3. Biweekly data for outbreaks of measles in three communities in Niamey, Niger are provided in the dataframe `niamey`. Use this method to obtain estimates of R_0 for measles from the first community assuming that the infectious period is approximately two weeks or $14/365 \approx 0.0384$ years.

Exercise 4. A defect with this method is that it uses only a small fraction of the information that might be available, *i.e.*, the first few data points. Indeed, there is nothing in the method that tells one how many data points to use–this is a matter of judgment. Further, there is a tradeoff in that as more and more data points are used the precision of the estimate increases, but this comes at a cost of additional bias. Plot the estimate of R_0 obtained from $n = 3, 4, 5, ...$ data points against the standard error of the slope from the regression analysis to show this tradeoff.

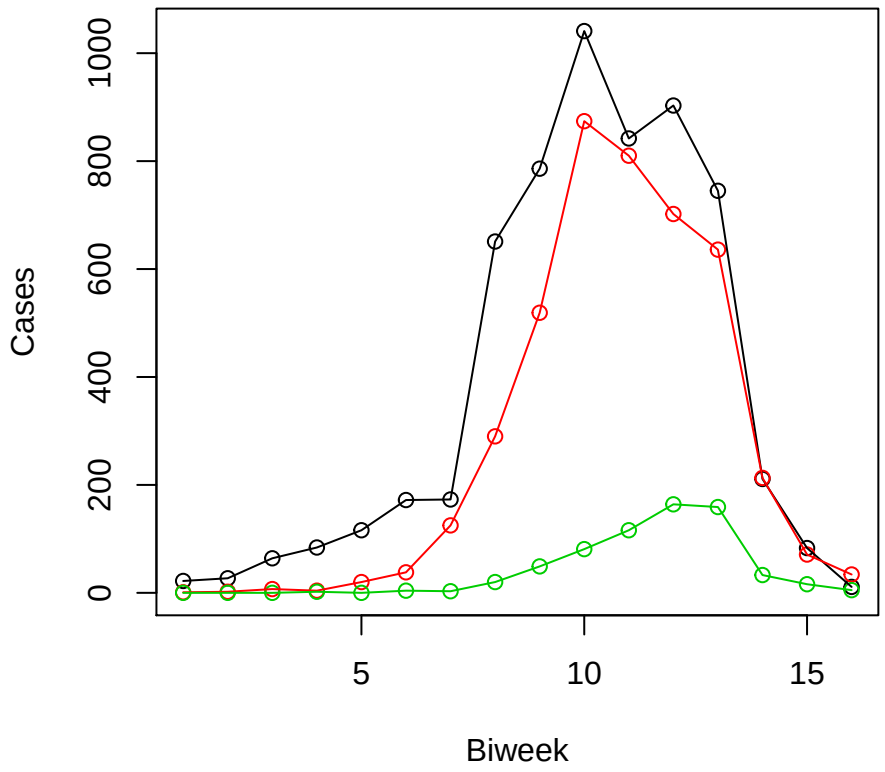
4 Estimating dynamical parameters with least squares

The objective of the previous exercise was to estimate R_0. Knowing R_0 is critical to understanding the dynamics of any epidemic system. It is, however, a composite quantity and is not sufficient to completely describe the epidemic trajectory. For this, we require estimates for all parameters of the model. In this exercise, we introduce a simple approach to model estimation called *least squares fitting*, sometimes called *trajectory matching*. The basic idea is that we find the values of the model parameters that minimize the squared differences between model predictions and the observed data. To demonstrate least squares fitting, we consider an outbreak of measles in Niamey, Niger, reported on by Grais et al. 2006 (Grais, R.F., *et al.* 2006. Estimating transmission intensity for a measles outbreak in Niamey, Niger: lessons for intervention. *Transactions of the Royal Society of Tropical Medicine and Hygiene* 100:867-873.).

```
> load('data.RData')
> niamey[5,3]<-0  #replace a "NA"
```

```
> niamey<-data.frame(biweek=rep(seq(1,16),3),site=c(rep(1,16),rep(2,16),rep(3,16)),
+                    cases=c(niamey[,1],niamey[,2],niamey[,3])) #define "biweeks"

> plot(niamey$biweek,niamey$cases,type='p',col=niamey$site,xlab='Biweek',ylab='Cases')
> lines(niamey$biweek[niamey$site==1],niamey$cases[niamey$site==1])
> lines(niamey$biweek[niamey$site==2],niamey$cases[niamey$site==2],col=2)
> lines(niamey$biweek[niamey$site==3],niamey$cases[niamey$site==3],col=3)
```



5 Dynamical model

First, we write a specialized function for simulating the SIR model in a case where the removal rate is "hard-wired" and with no demography.

```
> closed.sir.model <- function (t, x, params) {  #SIR model equations
+   X <- x[1]
+   Y <- x[2]
+   beta <- params
+   dX <- -beta*X*Y
```

```
+   dY <- beta*X*Y-(365/13)*Y
+   list(c(dX,dY))
+ }
```

6 Objective function

Now we set up a function that will calculate the sum of the squared differences between the observations and the model at any parameterization (more commonly known as "sum of squared errors"). In general, this is called the *objective function* because it is the quantity that optimization seeks to minimize.

```
> sse.sir <- function(params0,data,site){  #function to calculate squared errors
+   data<-data[data$site==site,]    #working dataset, based on site
+   t <- data[,1]*14/365            #time in biweeks
+   cases <- data[,3]               #number of cases
+   beta <- exp(params0[1])            #parameter beta
+   X0 <- exp(params0[2])           #initial susceptibles
+   Y0 <- exp(params0[3])           #initial infected
+   out <- as.data.frame(ode(c(X=X0,Y=Y0),times=t,closed.sir.model,beta,hmax=1/120))
+   sse<-sum((out$Y-cases)^2)       #sum of squared errors
+ }
>
```

Notice that the code for `sse.sir` makes use of the following modeling trick. We know that β, X_0, and Y_0 must be positive, but our search to optimize these parameters will be over the entire number line. We could constrain the search using a more sophisticated algorithm, but this might introduce other problems (i.e., stability at the boundaries). Instead, we parameterize our objective function (`sse.sir`) in terms of some alternatve variables $log(\beta)$, $log(X_0)$, and $log(Y_0)$. While these numbers range from $-\infty$ to ∞ (the range of our search) they map to our model parameters on a range from 0 to ∞ (the range that is biologically meaningful).

7 Optimization

Our final step is to use the function `optim` to find the values of β, X_0, and Y_0 that minimize the sum of squared errors as calculated using our function.

```
> library(deSolve)   #differential equation library
> params0<-c(-3.2,7.3,-2.6)  #initial guess
> fit1 <- optim(params0,sse.sir,data=niamey,site=1) #fit
> exp(fit1$par)  #back-transform parameters
```

```
[1] 5.463181e-03 9.110385e+03 2.331841e+00
```

```
> fit2 <- optim(params0,sse.sir,data=niamey,site=2) #fit
> exp(fit2$par)  #back-transform parameters
```

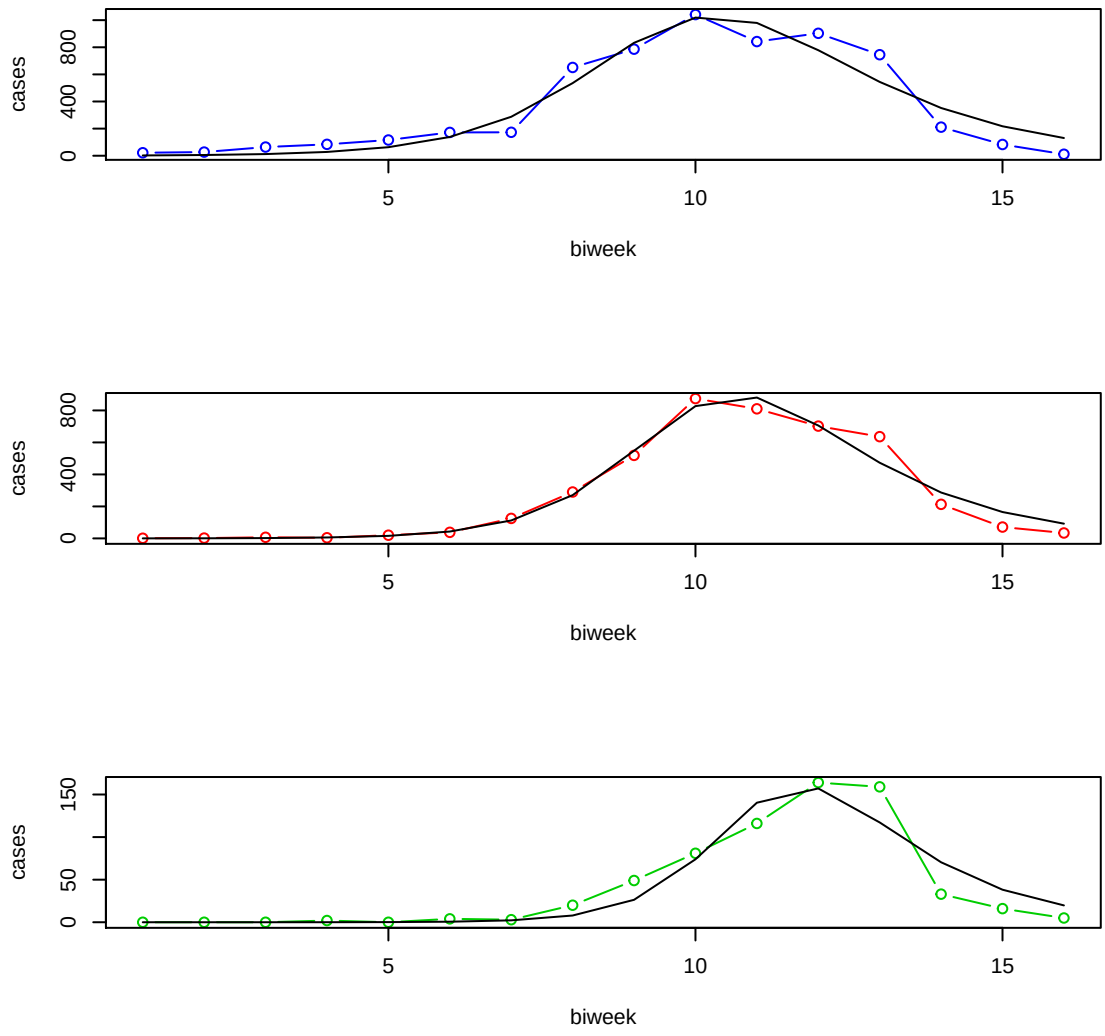
```
[1] 8.666138e-03 6.276503e+03 2.843753e-01
```

```
> fit3 <- optim(params0,sse.sir,data=niamey,site=3) #fit
> exp(fit3$par)  #back-transform parameters

[1] 7.130417e-02 8.625791e+02 1.031319e-03
```

Finally, we plot these fits against the data.

```
> par(mfrow=c(3,1))   #set up plotting area for multiple panels
> plot(cases~biweek,data=subset(niamey,site==1),type='b',col='blue', pch=21) #plot site 1
> t <- subset(niamey,site==1)[,1]*14/365
> mod.pred<-as.data.frame(ode(c(X=exp(fit1$par[2]),Y=exp(fit1$par[3])),times=t,
+                             closed.sir.model,exp(fit1$par[1]),hmax=1/120))
>                             #obtain model predictions
> lines(mod.pred$Y~subset(niamey,site==1)[,1]) #and plot as a line
> plot(cases~biweek,data=subset(niamey,site==2),type='b',col=site) #site 2
> t <- subset(niamey,site==2)[,1]*14/365
> mod.pred<-as.data.frame(ode(c(X=exp(fit2$par[2]),Y=exp(fit2$par[3])),times=t,
+                             closed.sir.model,exp(fit2$par[1]),hmax=1/120))
> lines(mod.pred$Y~subset(niamey,site==2)[,1])
> plot(cases~biweek,data=subset(niamey,site==3),type='b',col=site) #site 3
> t <- subset(niamey,site==3)[,1]*14/365
> mod.pred<-as.data.frame(ode(c(X=exp(fit3$par[2]),Y=exp(fit3$par[3])),times=t,
+                             closed.sir.model,exp(fit3$par[1]),hmax=1/120))
> lines(mod.pred$Y~subset(niamey,site==3)[,1])
>
```



Exercise 5. To make things easier, we have assumed the infectious period is known to be 14 days. In terms of years, $\gamma = (365/14)^{-1} \approx 0.0384$. Now, modify the code above to estimate γ and β simultaneously.

Exercise 6. What happens if one or both of the other unknowns (X_0 and Y_0) is fixed instead of γ?

8 From least squares to likelihood

Many researchers use the theory of *likelihood* as the basis for optimization. Likelihood is the probability that a the data were generated by a candidate model. By comparing models, one can identify that with the *maximum likelihood*. For a more developed introduction to the theory of fitting via maximum likelihood, see the optional exercise "Estimating model parameters by maximum likelihood".

A special case in likelihood theory is where the data are drawn from a normal probability density with mean given by the model and constant variance. That is, there is observation noise, but not process

noise; the observation errors are symmetrical and Gaussian, and the noise does not scale with the state of the system.

If Y_t is the number of observed infectives at time t and $\hat{Y}_t$ is the model's prediction, then the log of the probability of the data given the model is:

$$\begin{aligned}\log P(Y_t|\hat{Y}_t) &= \log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\,\exp\left(-\frac{(Y_t-\hat{Y}_t)^2}{2\sigma^2}\right)\right)\\ &= -\tfrac{1}{2}\,\log 2\pi\sigma^2 - \tfrac{1}{2}\,\frac{(Y_t-\hat{Y}_t)^2}{\sigma^2}\end{aligned}$$

From this, it follows that

$$\log\mathcal{L} = -\tfrac{1}{2}\,\left(\frac{1}{\sigma^2}\,\sum_t (Y_t-\hat{Y}_t)^2 + \log\left(\sigma^2\right) + \log\left(2\pi\right)\right),$$

where this last equation is the *negative log likeliood* of the data given the model. But, what is σ^2? It is a new parameter – the theoretical variance of the normally distributed errors, which is unknown. We can approximate it, however, with the variance of the deviations $Y_t - \hat{Y}_t$.

Exercise 7. Modify your optimizer so that it returns the negative log-likelihood. What is it for the three districts in Niamey?