Mathematics 4MB3/6MB3 Mathematical Biology

2018 ASSIGNMENT 2

Group Name: `πrates`

Group Members: Megan Hartwell, Ahmad Mahmood, Bradley Montgomery, Courtney Mulholland

This assignment is due in class on **Monday 5 February 2018 at 11:30am**.

1 Plot P&I mortality in Philadelphia in 1918

(a) Confirm that you have received this data file by e-mail:

`pim_us_phila_city_1918_dy.csv`

This plain text comma-separated-value file can be examined (if you wish) using any plain text editor, such as `Emacs`.

(b) Read the data into a data frame in R, using the `read.csv()` function. For example, the following chunk of R code should work:

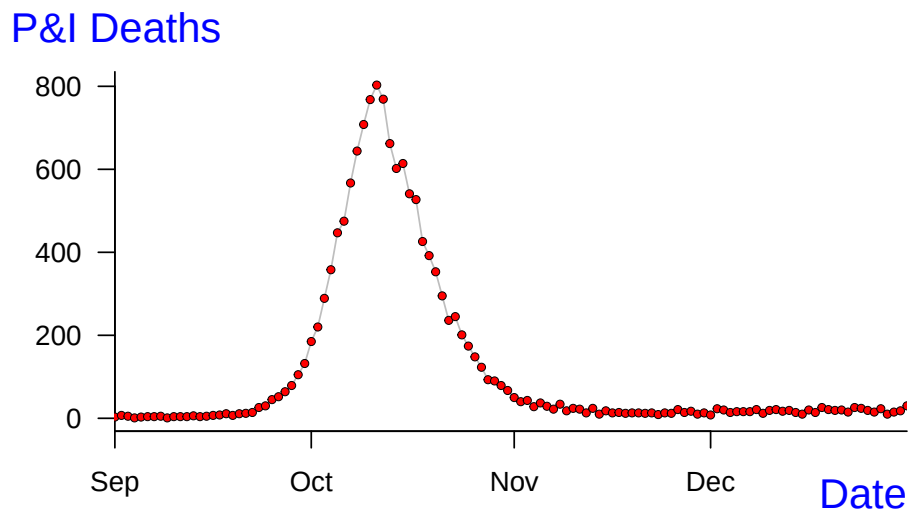
```
> datafile <- "pim_us_phila_city_1918_dy.csv"
> philadata <- read.csv(datafile)
> philadata$date <- as.Date(philadata$date)
```

The purpose of the last line of code above is to ensure that R encodes character strings such as `"1918-10-15"` as dates.

(c) Reproduce the Philadelphia 1918 P&I plot:

Solution. We read the data file using the code aboved and created the graph below.

```
> plot(philadata, pch=21, col="grey", type="l", bty="L",
+      ann=FALSE, las=1, xaxs="i") ## Plot underlying grey line
> points(philadata, pch=21, bg="red", cex=0.65, lwd=0.45) ## Add red pts
> ## Add blue labels
> mtext("Date", side=1, adj=1, line=1.5, font=1, cex=1.5, col='blue')
> mtext("P&I Deaths", side=2, padj=-7.0, line=-3.2, font=1,
+       las=1, cex=1.5, col='blue')
```



□

You'll need to use functions such as `plot()`, `points()` and `lines()`. For a comprehensive list of graphics parameters accepted by these functions, enter `?par` into the Console pane in `RStudio`. There are multiple ways to produce a graph exactly like the above, but the following steps work:

- Use `plot()` to draw the box and basic annotation and the grey line. Suppress labels when doing this (*e.g.,* `xlab=""`). The box type is controlled by the `bty` option and the orientation of annotation is controlled by the `las` option.
- Use `points()` to draw the heavy red dots with black borders. The most elegant way to do this is to set the point character type to 21 (`pch=21`) and the point background colour to red (`bg="red"`). Alternatively, you can use `points()` twice (first to draw the red dots and then to draw the black circles around them).
- Use `mtext()` to add the x and y axis labels in the margins of the plot.

2 Estimate $\mathcal{R}_0$ from the Philadelphia P&I time series

(a) The observed mortality time series $M(t)$ is certainly not equal to the prevalence $I(t)$ that appears in the SIR model. Suppose, however, that $I(t) = \eta M(t - \tau)$ for all time (where η and τ are constants), *i.e.,* that the mortality curve is exactly a scaled and translated version of the prevalence curve. Prove that if both I and M are growing exactly exponentially over some time period then their exponential rates are identical. Thus, if we compare them during the "exponential phase" on a logarithmic scale, then both curves will be perfectly straight with exactly the same slope.

Solution. Since we know both I and M are growing exactly exponentially we can write

each as an explicit exponential equation.

$$\begin{aligned} I(t) &= I_0 e^{K_0 t} \\ M(t) &= M_0 e^{K_1 t} \end{aligned} \tag{1}$$

Now we write M as a function of $t - \tau$:

$$\begin{aligned} M(t-\tau) &= M_0 e^{K_1(t-\tau)} \\ &= \frac{M_0}{e^{K_1 \tau}} e^{K_1 t} \end{aligned} \tag{2}$$

Next we use the equation given in the question, $I(t) = \eta M(t-\tau)$ and substiute the expressions we found above.

$$\begin{aligned} I(t) &= \eta M(t-\tau) \\ I_0 e^{K_0 t} &= \frac{\eta M_0}{e^{K_1 \tau}} e^{K_1 t} \end{aligned} \tag{3}$$

We know these expressions are equal for all time, and the constants have no time dependance, so we must set these equal. The only time dependance is in the exponentials, so these must also be equal, meaning $K_1 = K_0$.

$$\begin{aligned} I_0 &= \frac{\eta M_0}{e^{K_1 \tau}} \\ e^{K_0 t} = e^{K_1 t} &\Rightarrow K_1 = K_0 \end{aligned} \tag{4}$$

Therefore the rates of exponential growth for I and M must be equal. □

(b) Fit a straight line to the part of the Philadelphia 1918 mortality time series that looks straight on a logarithmic scale (and show your result in a plot). Once you get the hang of it, the easiest way to do this is to use the `lm()` function in R (lm stands for linear model). Note that the simplest way to draw a straight line with given slope and intercept is with the `abline()` function. If you find `lm()` counter-intutive to understand then experiment with `abline()` until your eyes tell you that you have discovered a line that provides a good fit.

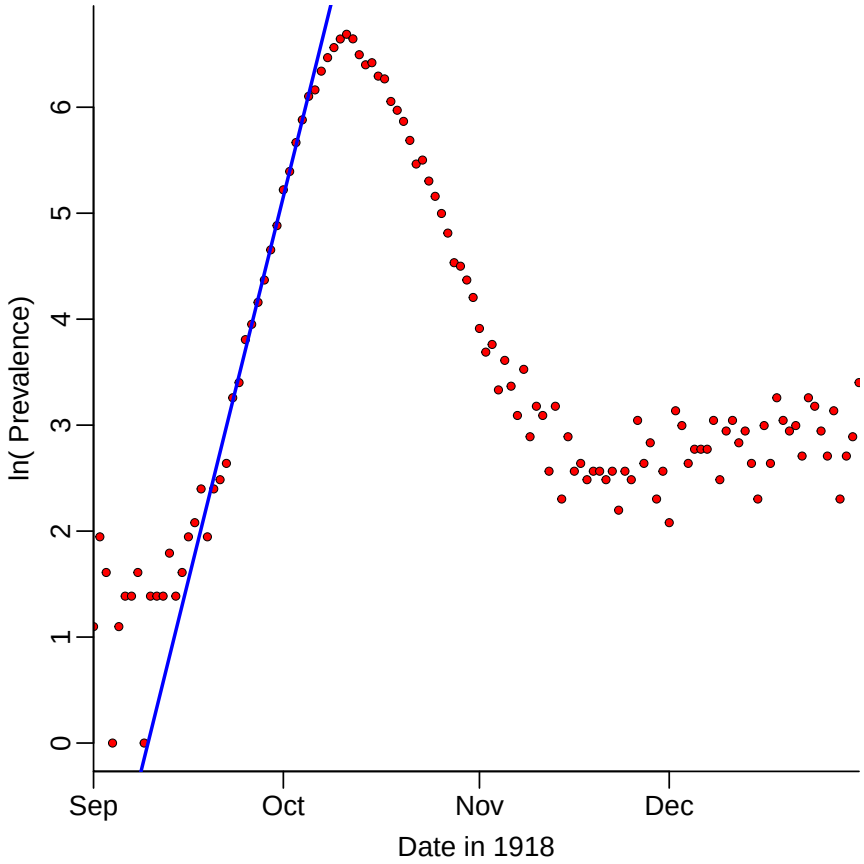
Solution. We took the logarithm of the prevalence of the same data used above, and fit a straight line to the linear part of the curve. The code and resulting graph are below.

```
> lnPI <- log(philadata$pim)
> # Adjust margins
> par(mar=c(4,4,3,3), mgp=c(1.75,0.5,0))
> # Create plot of logarithm of data
> plot(philadata$date, lnPI, pch=21, bg="red", type="p",
+      xlab="Date in 1918", ylab="ln( Prevalence)",
+      xaxs="i", cex=0.65, lwd=0.45, bty="L")
> #Fit linear model to subset of data
> m1 <- lm(lnPI[23:35]~philadata$date[23:35])
```

```
> # Add line to plot
> abline(m1, col="blue", lwd=2)
> text(x = 0,y=5,col="blue",
+      label=paste0("y=",round(m1$coefficients[1],2),"+",
+                   round(m1$coefficients[2],2),"t"))
> print(m1)

Call:
lm(formula = lnPI[23:35] ~ philadata$date[23:35])

Coefficients:
          (Intercept)  philadata$date[23:35]
            4517.082                  0.241
```



□

(c) How is the slope of your fitted line related to the parameters of the SIR model? (*Hint:* When I is small, $S \simeq 1$.) Why do you need an independent measure of the mean infectious period to estimate $\mathcal{R}_0$? If the mean infectious period is 4 days, what is your estimate of $\mathcal{R}_0$?

Solution. As shown in class, the slope of the logarithm of the prevalence curve is approximately $\beta - \gamma$ initially, as $S \approx 1$, as we derive below.

$$\begin{aligned} \frac{dI}{dt} &= \beta SI - \gamma I \\ &= (\beta S - \gamma) I \qquad \text{Assume } S \approx 1 \\ &\approx (\beta - \gamma) I \end{aligned} \tag{5}$$

Solving this equation we get $I(t) \approx I_0 e^{(\beta-\gamma)t}$ and we can see the slope of the exponential is $\beta - \gamma$. As measured in part (b) we found the slope of this line to be 0.241. We need an independent measure of the mean infectious period to estimate $\mathcal{R}_0$ as $\mathcal{R}_0$ is defined as the product of $\frac{1}{\gamma}$ and β, and we cannot solve for either of these using the linear combination of β and γ that we found from the slope. If we assume that the mean infectious period, $\frac{1}{\gamma}$, is 4, we can then solve for $\mathcal{R}_0$.

```
> sl=m1$coefficients[2][[1]]
> gam=1/4
> bet=sl+gam
> R0=bet/gam
> print(R0)

[1] 1.964087
```

□

3 Fit the basic SIR model to the Philadelphia P&I time series

(a) Install the `"deSolve"` package. This is done by typing the following command in the Console pane of `RStudio`:

```
install.packages("deSolve")
```

You will then be prompted to choose a mirror site from which to download the package. It doesn't matter which mirror you choose, but choosing a site in Ontario might save a fraction of a second. *Note:* This is a one-time operation. You do not want an `install.packages()` command inside your solutions code.

(b) Write an R function that plots the solution $I(t)$ of the SIR model for given parameter values ($\mathcal{R}_0$ and $1/\gamma$) and given initial conditions (S_0, I_0). Use the `ode()` function in the `deSolve` package. A few hints:

- Your code will first need to load the `deSolve` package:

  ```
  > library("deSolve")
  ```

- As an example of defining a function (without getting involved with a differential equation), here is a code chunk that defines a function to plot a sine curve, and then executes the function. Note that the default min and max x values are set in the parameter list of the function definition, but the max x value is changed when the function is executed:

```
> plot.sine <- function(xmin=0, xmax=2*pi ) {
+   x <- seq(xmin, xmax, length=100)
+   plot(x, sin(x), typ="l")
+   grid() # add a light grey grid
+ }
> plot.sine(xmax=4*pi)
```

- Here's another example. This time we first define the vector field for a differential equation. We then use this function inside another function that plots the solution of the associated differential equation. To understand the construction, you can, as usual, study the help page for the calling function (`?ode` in this case), but the most important issues are the following.

  One of the arguments of the `ode()` function is the function that evaluates the vector field at the current time. To avoid confusion, choose the arguments of your vector field function to be `t`, `vars` and `parms` (in that order):

`t`	The current time, which will be used within the vector field function if the system is non-autonomous.
`vars`	A named vector of the variables in the system (*e.g.,* S, I). The variables, as named vector passed to this function, are used in the code that defines the vector field within the function.
`parms`	A named vector of the parameters of the system (*e.g.,* β, γ). It is convenient—but not necessary—to specify default values for the parameters.

  It is strongly recommended that you follow exactly the style below when defining vector fields for differential equations that you wish to solve with the `ode()` function. In particular, the construction "`with(as.list(c(parms,vars)), ...)`" makes the variables and parameters visible within the section of code between the braces (`{...}`) without having to refer to the vectors or lists in which they are stored. For example, the code would be much harder to read if each instance of `x` were replaced by `vars$x` and each instance of `beta` were replaced by `parms$beta`; this issue becomes extremely important for complicated vector fields.

```
> ## Vector Field for SI model
> SI.vector.field <- function(t, vars, parms=c(beta=2, gamma=1)) {
+   with(as.list(c(parms, vars)), {
+     dx <- -beta*x*y # dS/dt
+     dy <- beta*x*y  # dI/dt
+     vec.fld <- c(dx=dx, dy=dy)
+     return(list(vec.fld)) # ode() requires a list
```

```
+   })
+ }
```

The following function plots a single solution of the ODE for a given initial condition (`ic`), integration time (`tmax`) and times at which the state is to be returned (`times`). The vector field function is passed as the `func` argument and the parameter vector is passed as the `parms` argument. If further arguments are given, they are passed to the `lines()` function that draws the solution.

```
> ## Draw solution
> draw.soln <- function(ic=c(x=1,y=0), tmax=1,
+                       times=seq(0,tmax,by=tmax/500),
+                       func, parms, ... ) {
+   soln <- ode(ic, times, func, parms)
+   lines(times, soln[,"y"], col="blue", lwd=3, ... )
+ }
```

Note here that the call to the `ode()` function gives the arguments in the default order so they are interpretted correctly. If we wished to write the arguments in a different order then we would have to be explicit about which argument is which. For example, if we wanted to list the initial conditions last for some deep reason then we would have to write:

```
>   soln <- ode(times=times, func=func, parms=parms, y=ic)
```

We can now use our `draw.soln()` function to plot a few solutions of the SI model.

```
> ## Plot solutions of the SI model
> tmax <- 10 # end time for numerical integration of the ODE
> ## draw box for plot:
> plot(0,0,xlim=c(0,tmax),ylim=c(0,1),
+      type="n",xlab="Time (t)",ylab="Prevalence (I)",las=1)
> ## initial conditions:
> I0 <- 0.001
> S0 <- 1 - I0
> ## draw solutions for several values of parameter beta:
> betavals <- c(1.5,2,2.5)
> for (i in 1:length(betavals)) {
+   draw.soln(ic=c(x=S0,y=I0), tmax=tmax,
+             func=SI.vector.field,
+             parms=c(beta=betavals[i],gamma=1),
+             lty=i # use a different line style for each solution
+             )
+ }
```

Solution. • We first wrote the following code to represent the ODE as a vector field that R can integrate numerically. We've chosen to use the parameters γ and β.

```
> SIR.vector.field <- function(t, vars, parms=c(beta=2,gamma=1)) {
+   # Writes vector field of these differential equations
+   with(as.list(c(parms, vars)), {
+     dS <- -1*beta*S*I # dS/dt
+     dI <- beta*S*I-gamma*I  # dI/dt
+     dR <- gamma*I  # dR/dt
+     vec.fld <- c(dS=dS, dI=dI, dR=dR)
+     return(list(vec.fld)) # ode() requires a list
+   })
+ }
```

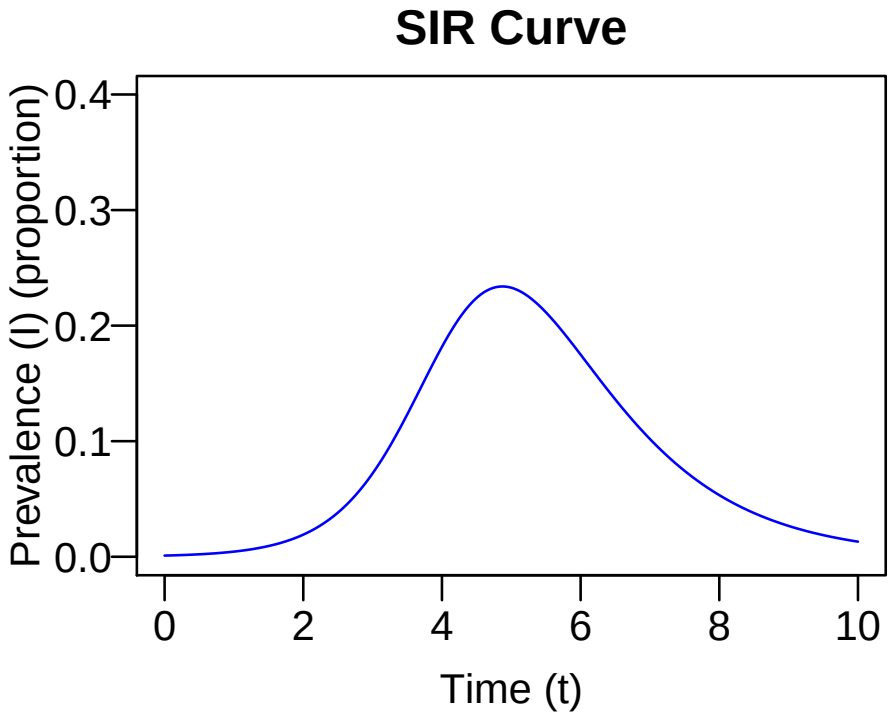
- To tell R that we want to solve this differential equation we need to tell it to integrate the equations above using the parameters we give it, from the initial conditions we specify. The chunk of code below takes the parameters, $\mathcal{R}_0$ and $\frac{1}{\gamma}$, and calculates the parameters β and γ and uses them to solve the ODE, starting from the initial conditions, `ic` that we give, using the time points `times`. We then add this solution to the current plot using the `lines` command.

```
> draw.soln <- function(ic=c(S=0.999,I=0.001,R=0), tmax=1,
+                       times=seq(0,tmax,length.out = 500),
+                       func, parms=c(R_0=2,gamma_inv=1),
+                       colour="blue", ... ) {
+   # Solve ode from vector field using deSolve package and
+   # initial condition and parameters given
+   with(as.list(parms),{
+     gamma=1/gamma_inv
+     beta=gamma*R_0
+     soln <- ode(y=ic, times=times, func,
+                 parms=c(beta=beta,gamma=gamma)) # Solve the ode
+     lines(times, soln[,"I"],
+           col=colour, ... ) # Add this line to our plot
+     return(data.frame(time=times,y=soln[,"I"]))
+   })
+ }
```

- Finally we use these two functions to solve the differential equations with our parameters and initial conditions.

```
> tmax <- 10 # end time for numerical integration of the ODE
> # Adjust margins:
> par(mar=c(3,3,2,2),mgp=c(1.75,0.5,0))
> ## draw box for plot:
> plot(0,0,xlim=c(0,tmax),ylim=c(0,0.4),
+      type="n",las=1,
+      xlab="Time (t)",ylab="Prevalence (I) (proportion)",
+      main="SIR Curve")
> ## initial conditions:
> I0 <- 0.001
```

```
> S0 <- 1 - I0
> R0 <-  0
> ## Set parameter values
> Rknot_vals <- 2.5
> gamma_inv <- 1
> # Solve ode for these parameters and initial conditions
> p <- draw.soln(ic=c(S=S0,I=I0,R=R0), tmax=tmax,
+           func=SIR.vector.field,
+           parms=c(R_0=Rknot_vals,gamma_inv=gamma_inv,lwd=5)
+ )
```



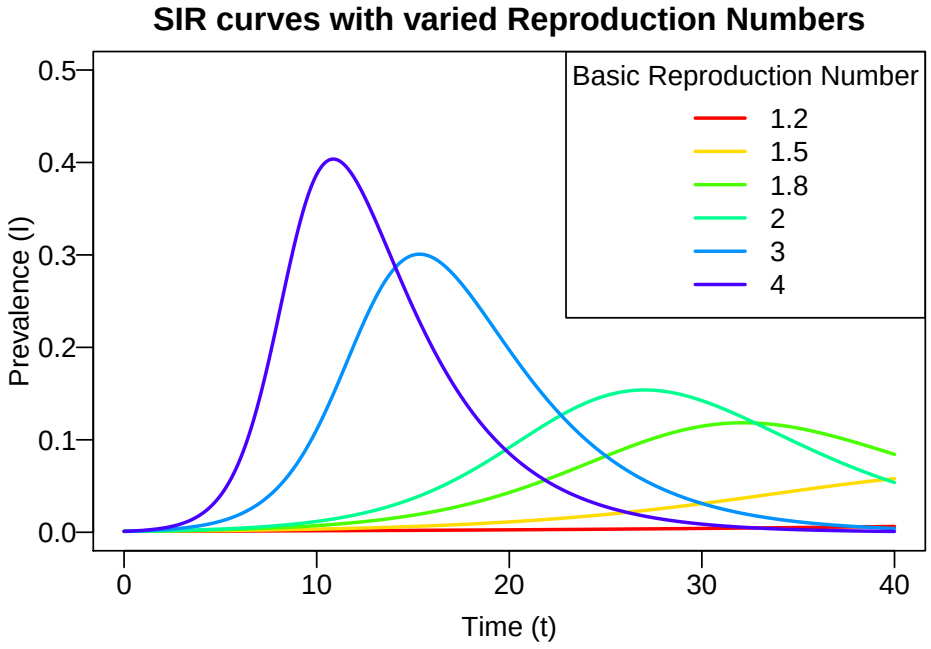
□

(c) For $I_0 = 10^{-3}$ and $S_0 = 1 - I_0$, plot the solutions of the SIR model assuming $1/\gamma = 4$ days and $\mathcal{R}_0 \in \{1.2, 1.5, 1.8, 2, 3, 4\}$. Use the `legend()` command to make a legend on the plot that shows which curves correspond to which values of $\mathcal{R}_0$.

Solution. Using the code written in part (b) we will make several calls to these functions to plot a series of curves with different $\mathcal{R}_0$ values.

```
> tmax <- 40 # end time for numerical integration of the ODE
> ## initial conditions:
> I0 <- 0.001
> R0 <-  0
> S0 <- 1 - I0- R0
> # Parameters:
```

```
> gamma_inv=4
> Rknot_vals <- c(1.2,1.5,1.8,2,3,4)
> # Adjust margins:
> par(mar=c(3,3,2,2),mgp=c(1.75,0.5,0))
> ## draw box for plot:
> plot(0,0,xlim=c(0,tmax),ylim=c(0,0.5),
+      type="n",las=1,
+      xlab="Time (t)",
+      ylab="Prevalence (I)",
+      main="SIR curves with varied Reproduction Numbers")
> ## draw solution for each value of Rknot:
> for (i in 1:length(Rknot_vals)) {
+   draw.soln(ic=c(S=S0,I=I0,R=R0), tmax=tmax,
+             times=seq(0,tmax,length.out = 500),
+             func=SIR.vector.field,
+             parms=c(R_0=Rknot_vals[i],gamma_inv=gamma_inv),
+             lwd=2, colour=rainbow(length(Rknot_vals)+1)[i]
+             # use a different line colour for each solution
+             )
+ }
> legend("topright",legend=Rknot_vals,title="Basic Reproduction Number",
+        col=rainbow(length(Rknot_vals)+1),lwd=2)
> # title(paste("SIR curves with varied","$R_0$","Values"))
```



□

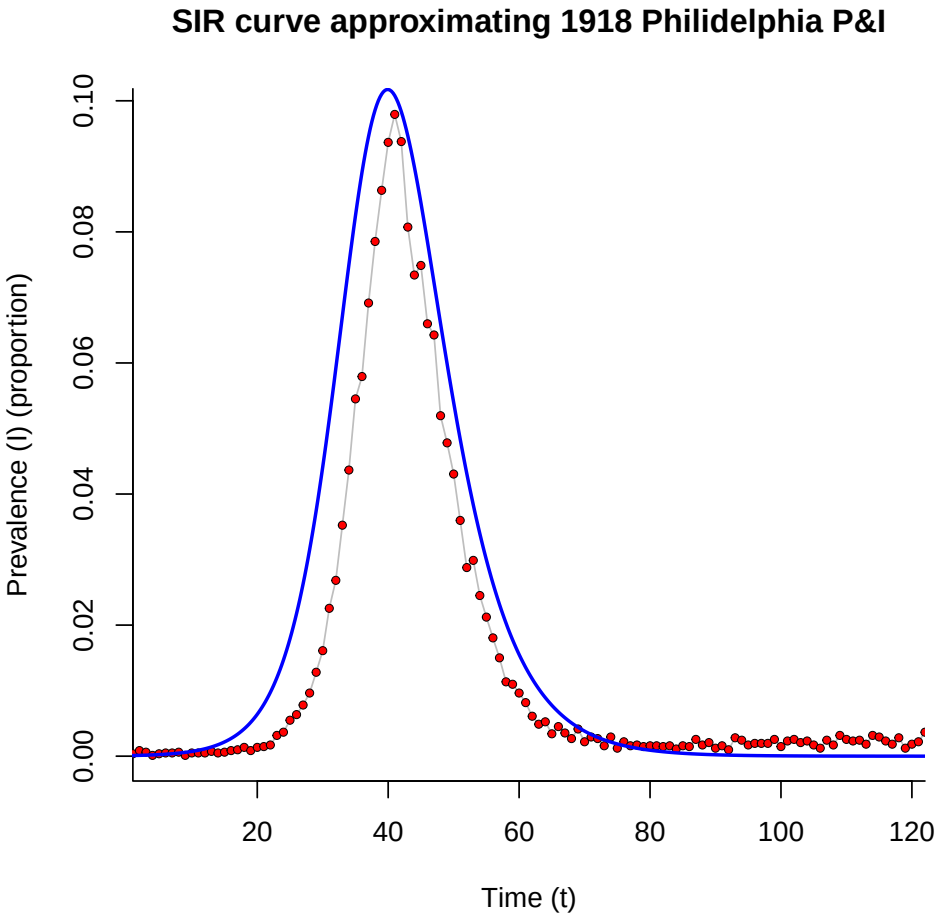
(d) By trial and error, find values of $\mathcal{R}_0$ and γ that yield a solution of the SIR model that fits the Philadelphia P&I times series reasonably well. You can assess the quality of fit using

the Euclidean distance between the model solution and the data. (*Note:* The trial and error approach is a valuable exercise, but not a suggestion of a method you would really use in practice. We'll discuss better methods for fitting ODE models to data later.)

Solution. When beginning this question we began by fitting the shape of the curve to the data points and then scaling the data points down to fit the curve. From previous questions and intuiton we estimated that the mean infectious period ($\frac{1}{\gamma}$) would be about 4 days, and the basic reproduction number ($\mathcal{R}_0$) to be about 2. From these points we varied the estimations of the population, I_0, R_0, and parameters $\frac{1}{\gamma}$ and $\mathcal{R}_0$.

The best estimate we found was with $I_0 = 8 \cdot 10^{-5}$, $R_0 = 0.1$, $N = 8200$, $\mathcal{R}_0 = 1.97$ and $\frac{1}{\gamma} = 3.5$. With these estimates the error between our curve and the data points (calculated using all integer values of time) is 2529.808.

```
> tmax <- 122 # end time for numerical integration of the ODE
> philadata$t <- seq(1,nrow(philadata))
> y1 <- data.frame(time=philadata$t,y=philadata$pim)
> euc.dist <- function(x1, x2) sqrt(sum((x1 - x2) ^ 2))
> I0 <- 0.00008
> R0 <-  0.1
> S0 <- 1 - I0 - R0
> pop <- 8200
> plot(x=philadata$t,y=philadata$pim/pop,
+      col="grey", "l", ## Plot underlying grey line
+      bty="L",xaxs="i",xlab="Time (t)",
+      ylab="Prevalence (I) (proportion)",
+      main="SIR curve approximating 1918 Philidelphia P&I")
> points(x=philadata$t,y=philadata$pim/pop, pch=21,
+        bg="red",cex=0.65,lwd=0.45) ## Add red pts
> Rknot_vals <- 1.97
> gamma_inv <- 3.5
> ## Draw the solution on our plot
> y2 <- draw.soln(ic=c(S=S0,I=I0,R=R0), tmax=tmax,
+                 times=seq(0,tmax,length.out = 488+1),
+                 func=SIR.vector.field,
+                 parms=c(R_0=Rknot_vals,gamma_inv=gamma_inv), lwd=2,
+                 colour="blue" # use a different line colour for each solution
+ )
> # Calculate the error in our approximation
> func <- y2[which(y2$time %in% y1$time),"y"]
> E <- euc.dist(y1[,2],func)
```



4 Executive summary for the Public Health Agency

The Public Health Agency of Canada (PHAC) is revising their pandemic plan and has asked your group to summarize what you learned from analyzing the 1918 Philadelphia P&I time series. Besides explaining what inferences you feel you can make from your analysis so far, PHAC wants to know what you would investigate if they were to fund you to continue your work full time for a month. They want a maximum of one page from your group.

Incidentally, you might be interested to know that rumour has it that all of the members of the pandemic planning committee took Math 2C03 at McMaster University between 1980 and 2003, but they all failed. Also, when the chair of the committee was recently asked "What is a differential equation?" he apparently bent over and vomited (it is hard to know quite what to make of this given that PHAC was investigating a norovirus outbreak at the time).

Note: When submitting your assignment solution, it is imperative that the one-page executive summary be printed on its own page. To start a new page in LaTeX, use the `\newpage`

command. Also, as usual, your summary should be in 12 point font. Don't try to cram in as much as possible. Make that page as clear and concise as you can, so that a public health planner can absorb its content quickly and easily.

Solution. After analyzing the 1918 Philidelphia P&I time series our group gained information about the epidemic from mathematical modeling and simulation of events. Two very important parameters for determining how the diseases were spread are the mean infectious period and the basic reproduction number, but other inferences can be made from mathematical analysis.

The mean infectious period from trial-and-error calculation with our mathematical model appeared to be 3.5 days. This means that on average, a person will be infected for 3.5 days, and during this time they are able to spread the diseases to other individuals. Having this information is critical for preparing for potential future epidemics that may arise, since necessary precautions, including isolation and other common tactics to prevent spread of infection, can be taken while it is known that someone is still infective.

A second important parameter that was analyzed and found using mathematical modeling from our group was the basic reproduction number. This parameter is the product of the previously mentioned mean infectious period, as well as the transmission rate. During our analysis, our group concluded that the basic reproduction number for the 1918 Philadelphia P&I time series was 1.97. The importance of knowing this number is quite high as it tells us the average number of secondary cases caused by each primary case; primary case meaning someone that is already infected. Since we found this number to be above 1, we know when someone is infected, they are guaranteed to spread it to other individuals and cause an epidemic. Once again, having this information is crucial for preparation for potential future outbreaks of the diseases.

Finally, inference can be made about the overall processes that our model used to accurately predict the data from the P&I time series. Without going into too much detail, in summary interactions between those that were infected and those that were susceptible were what caused the increase in prevalence of infections. After some time, a maximum proportion of the population was achieved, which is based on the basic reproduction number. After the maximum, those that were infected either recovered or passed away, and the proportion of the total population that was infected approached zero.

If you choose to fund our group full time for a month of research we will be investigating other epidemics of pneumonia and influenza to fit mathematical models that allow us to make predictions about future epidemics. Having information on epidemics from other areas of the world would provide great insight into whether location, culture or a number of other factors affects spread of the disease. □

— END OF ASSIGNMENT —

Compile time for this document: February 4, 2018 @ 12:53