

Heterogeneity - Part 2

Amy Greer, Department of Population Medicine, University of Guelph

2023-06-22

Note: Some of the examples and code provided in this laboratory have been extracted and/or modified from the textbook *Epidemics, Models and Data using R* by Ottar Bjornstad (<https://link.springer.com/book/10.1007/978-3-319-97487-3>).

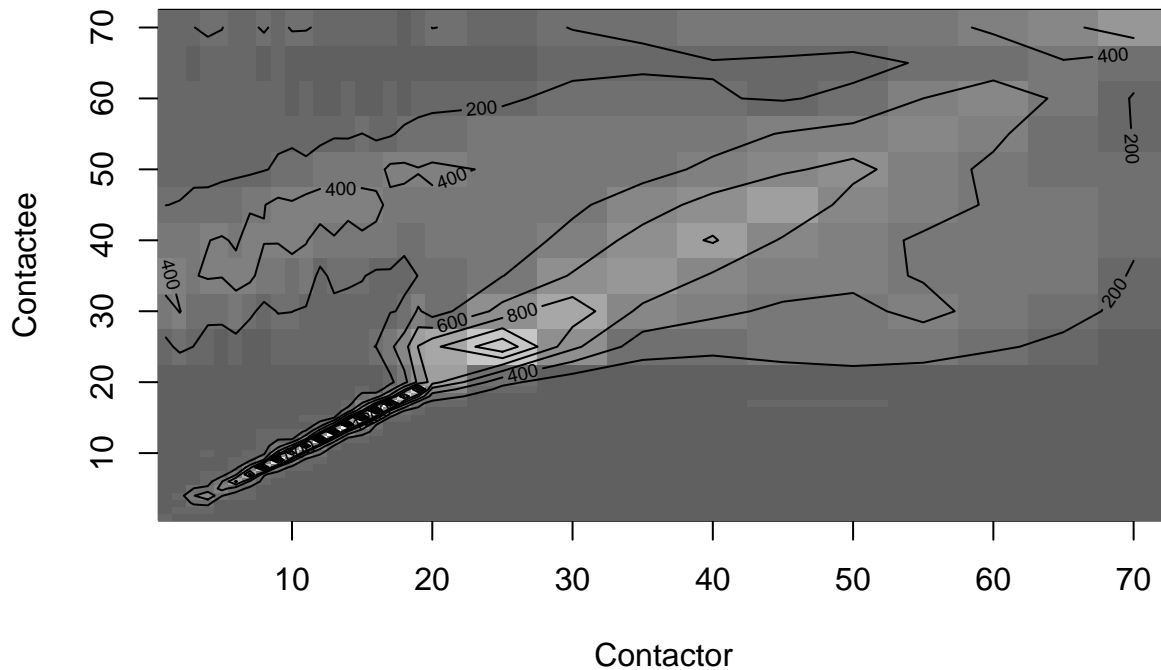
PART ONE - WAIFW

Age structured FOI arises from non-assortative mixing among different age groups. The Who-Acquires-Infection-From-Whom (WAIFW) matrix is used to describe non-homogenous mixing between different age groups. One of the most common data sources to parameterize WAIFW matrices is a study by Mossong et al. (2008), where they conducted contact diary surveys (and we have done this during the SARS-CoV-2 pandemic in Canada as well as we discussed in class). Mossong et al. provides age-stratified contact rates for various countries in Europe that were part of the POLY-MOD study. The contact rates by *contactor* and *contactee* are provided in the *mossong-data* set. We can visualize these diary data using an image plot with contours superimposed.

```
data(mossong)
head(mossong)

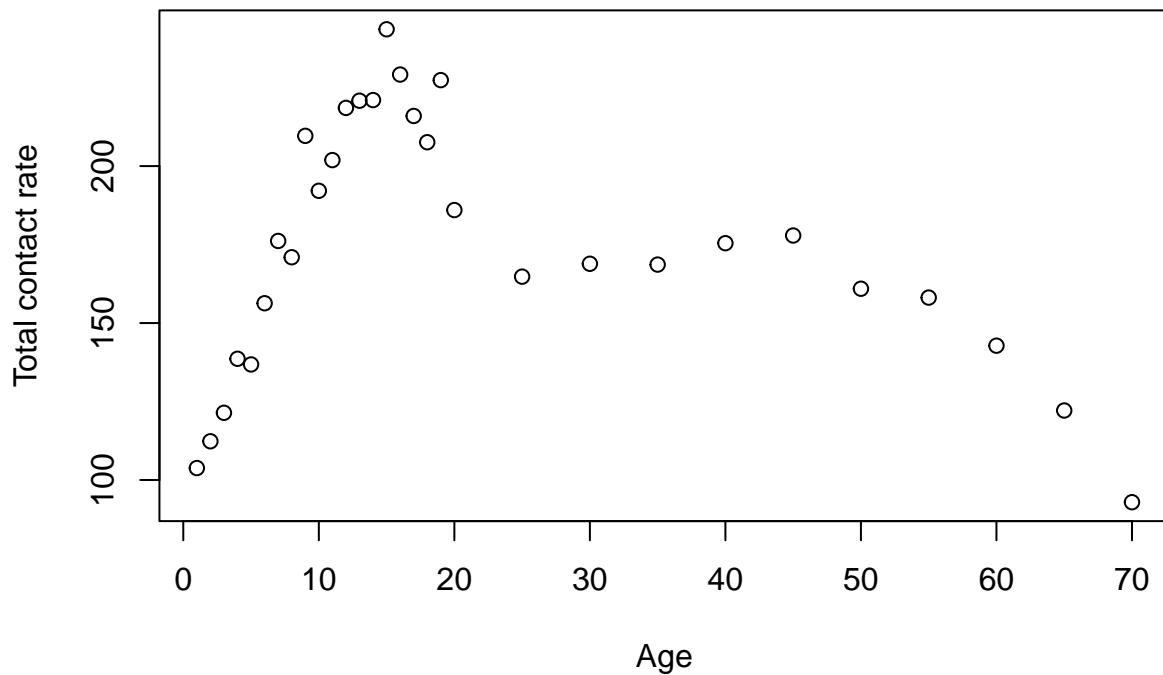
##   contactor contactee contact.rate
## 1         1         1    120.37234
## 2         2         1     33.45833
## 3         3         1     23.13380
## 4         4         1     24.33333
## 5         5         1     29.00662
## 6         6         1     14.50331

x=y=mossong$contactor[1:30]
z=matrix(mossong$contact.rate, ncol=30, nrow=30)
image(x=x, y=y, z=z, xlab="Contactor",
      ylab="Contactee", col=gray((12:32)/32))
contour(x=x, y=y, z=z, add=TRUE)
```



The reported contact rates are not symmetrical - which a WAIFW matrix will be - because of age-specific biases in diary entry rates and the age profiles of the *contactors* and the *contactees*. Before we “balance” the matrix we look at the reported marginal contact rate for each age group. Most contacts are among the same aged individuals and school-aged kids have the most contacts. We do also see some “off diagonal” ridges that represent parent-kid or kid-teacher interactions.

```
plot(apply(z,1,mean)~x, ylab="Total contact rate",
      xlab="Age")
```



This plot shows the age-specific contact rates reported by the POLY-MOD study by Mossong et al (2008). But we can’t use the raw data because we need the matrix to be symmetrical (why do you think that is?). The Mossong data are also found in the socialmixr package.

```

#install.packages('socialmixr')
#devtools::install_github('epiforecasts/socialmixr')
#library('socialmixr')
#?contact_matrix
data(polymod)
contact_matrix(polymod, countries = "United Kingdom", age.limits = c(0, 1, 5, 15), symmetric = TRUE)

## Using POLYMOD social contact data. To cite this in a publication, use the 'cite' function
## Removing participants that have contacts without age information. To change this behaviour, set the
## Warning in pop_age(survey.pop, part.age.group.present, ...): Not all age groups represented in popul
## Linearly estimating age group sizes from the 5-year bands.

## $matrix
##      contact.age.group
##      [0,1)  [1,5)  [5,15)  15+
## [1,] 0.40000000 0.6250000 0.764365 4.122919
## [2,] 0.15625000 1.9375000 1.406063 5.929829
## [3,] 0.07148821 0.5260153 7.946078 7.428739
## [4,] 0.05759306 0.3313352 1.109550 9.594101
##
## $demography
##      age.group population proportion year
## 1:      [0,1)      690734 0.01145748 2005
## 2:      [1,5)      2762936 0.04582990 2005
## 3:      [5,15)      7385454 0.12250542 2005
## 4:         15+      49447627 0.82020720 2005
##
## $participants
##      age.group participants proportion
## 1:      [0,1)           15 0.01483680
## 2:      [1,5)           80 0.07912957
## 3:      [5,15)          204 0.20178042
## 4:         15+          712 0.70425321

```

This generates a symmetrical contact matrix from the UK part of the POLYMOD study, with age groups 0-1, 1-5, 5-15 and 15+ years. It contains the mean number of contacts that each member of an age group (row) has reported with members of the same or another age group (column).

To get an idea of uncertainty of the contact matrices, a bootstrap can be used using the `sample.participants` argument of `contact_matrix()`. If this argument is set to `TRUE`, participants are sampled (with replacement, to get the same number of participants of the original study) every time the `contact_matrix()` function is called, and thus a different matrix returned every time. From these matrices, derived quantities can be obtained, for example the mean:

```

m <- replicate(n = 5,
  contact_matrix(
    polymod, countries = "United Kingdom", age.limits = c(0, 1, 5, 15),
    sample.participants = TRUE
  )
)

```

```

## Using POLYMOD social contact data. To cite this in a publication, use the 'cite' function
## Removing participants that have contacts without age information. To change this behaviour, set the
## Using POLYMOD social contact data. To cite this in a publication, use the 'cite' function

```

```
## Removing participants that have contacts without age information. To change this behaviour, set the
## Using POLYMOD social contact data. To cite this in a publication, use the 'cite' function
## Removing participants that have contacts without age information. To change this behaviour, set the
## Using POLYMOD social contact data. To cite this in a publication, use the 'cite' function
## Removing participants that have contacts without age information. To change this behaviour, set the
## Using POLYMOD social contact data. To cite this in a publication, use the 'cite' function
## Removing participants that have contacts without age information. To change this behaviour, set the
mr <- Reduce("+", lapply(m["matrix", ], function(x) x / ncol(m)))
mr
```

```
##           contact.age.group
## age.group   [0,1)   [1,5)   [5,15)   15+
## [0,1)  0.40383394 0.9298657 1.022857 5.536752
## [1,5)  0.10691033 1.8046816 1.433814 5.589765
## [5,15) 0.03310930 0.5015826 8.089758 6.232917
## 15+    0.03688354 0.3564376 1.221545 9.522006
```

We might also want to use contact rates per capita. Contact rate per capita is used in infectious disease modelling to calculate the force of infection, which is based on the likelihood that one susceptible individual of age i will be in contact with one infectious individual of age j .

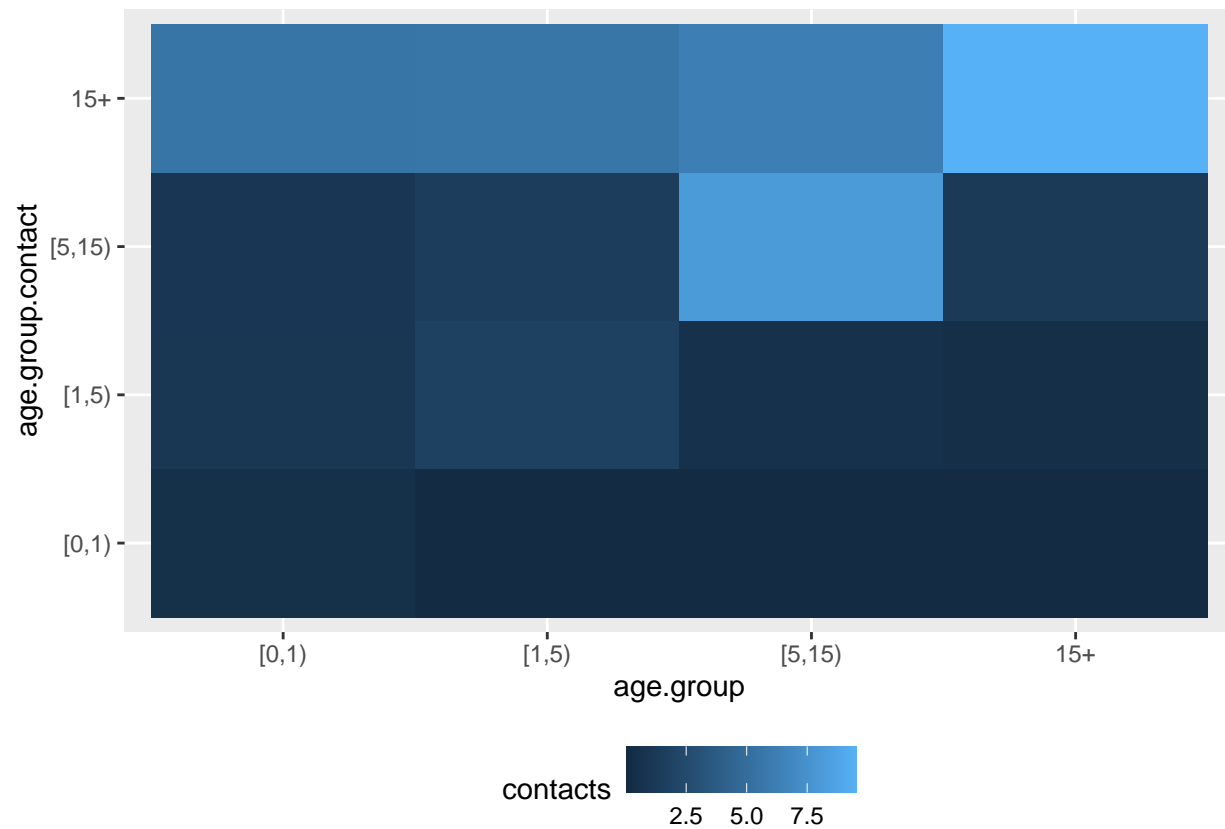
```
contact_matrix(survey = polymod, countries = "Germany", age.limits = c(0,60), symmetric = TRUE, per.capita = TRUE)

## Using POLYMOD social contact data. To cite this in a publication, use the 'cite' function
## Removing participants without age information. To change this behaviour, set the 'missing.participants' argument
## Removing participants that have contacts without age information. To change this behaviour, set the 'missing.participants' argument

## $matrix
##           contact.age.group
##           [0,60)   60+
## [1,] 7.743879 0.8967442
## [2,] 2.711694 2.1267606
##
## $matrix.per.capita
##           contact.age.group
##           [0,60)   60+
## [1,] 1.261735e-07 4.418248e-08
## [2,] 4.418248e-08 1.047852e-07
##
## $demography
## age.group population proportion year
## 1: [0,60) 61374868 0.7514869 2005
## 2: 60+ 20296375 0.2485131 2005
##
## $participants
## age.group participants proportion
## 1: [0,60) 1062 0.8329412
## 2: 60+ 213 0.1670588
```

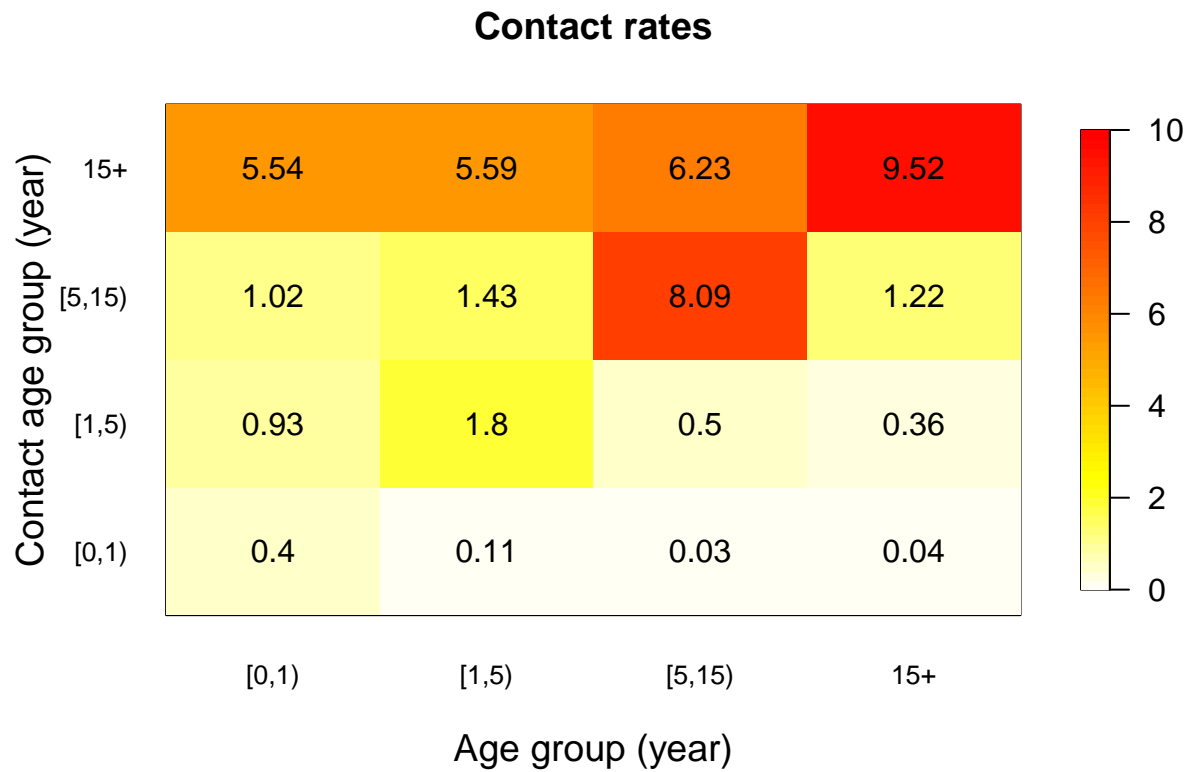
We can also plot the contact matrices (this is most often done using heat maps).

```
#library('reshape2')
#library('ggplot2')
df <- melt(mr, varnames = c("age.group", "age.group.contact"), value.name = "contacts")
ggplot(df, aes(x = age.group, y = age.group.contact, fill = contacts)) +
  theme(legend.position = "bottom") +
  geom_tile()
```

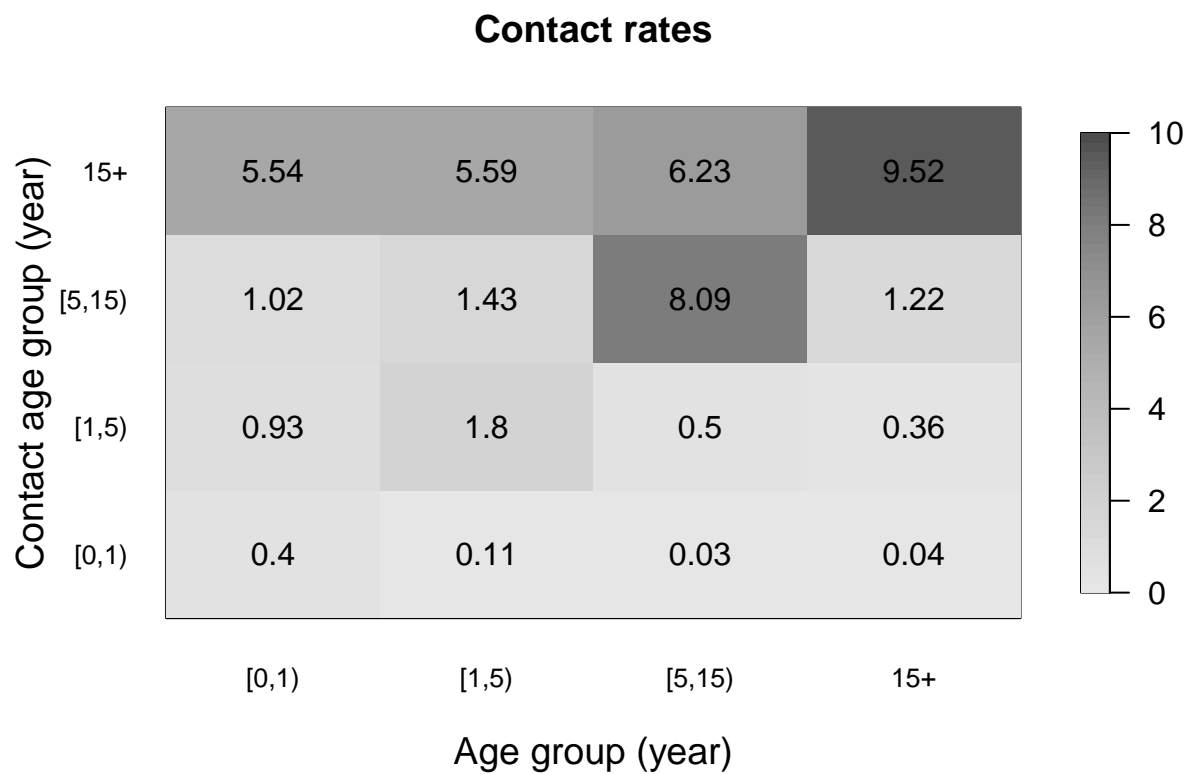


We can also plot this in other ways.

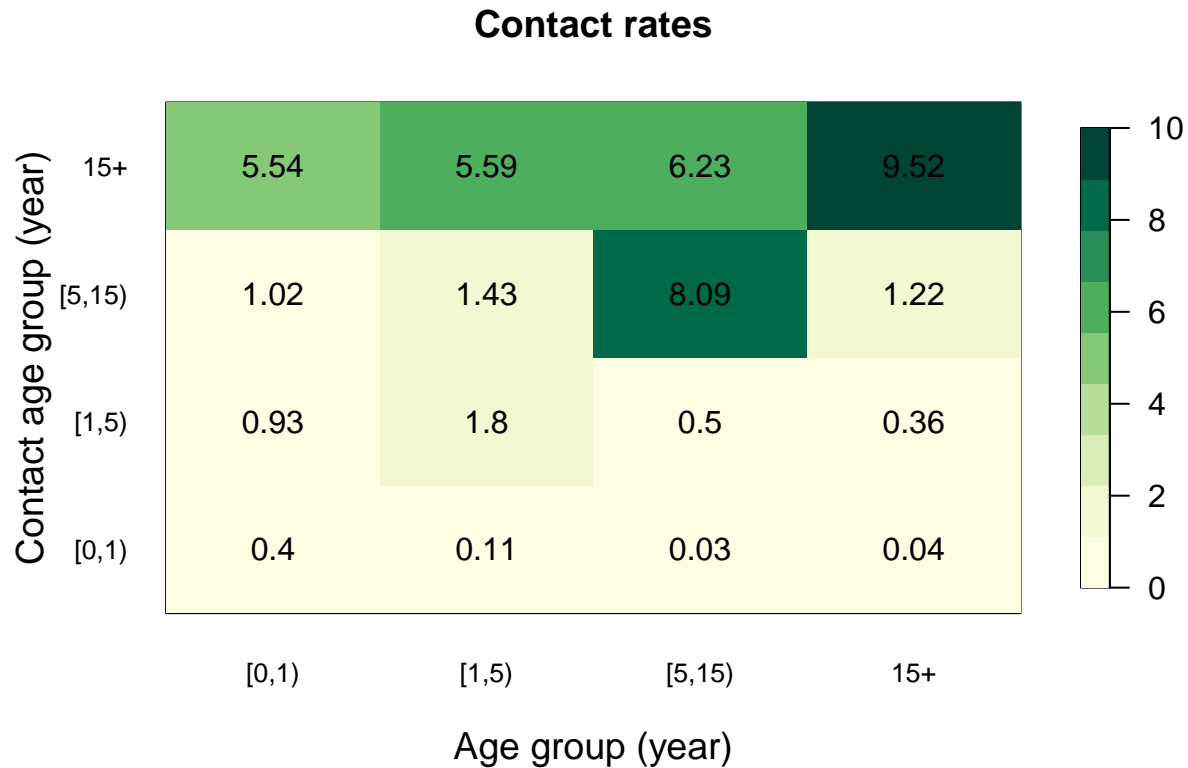
```
matrix_plot(mr)
```



```
matrix_plot(mr, color.palette = gray.colors)
```



```
usr_col <- function(x){ return(grDevices::hcl.colors(x, 'YlGn')) }
matrix_plot(mr, num.colors = 9, color.palette = usr_col)
```

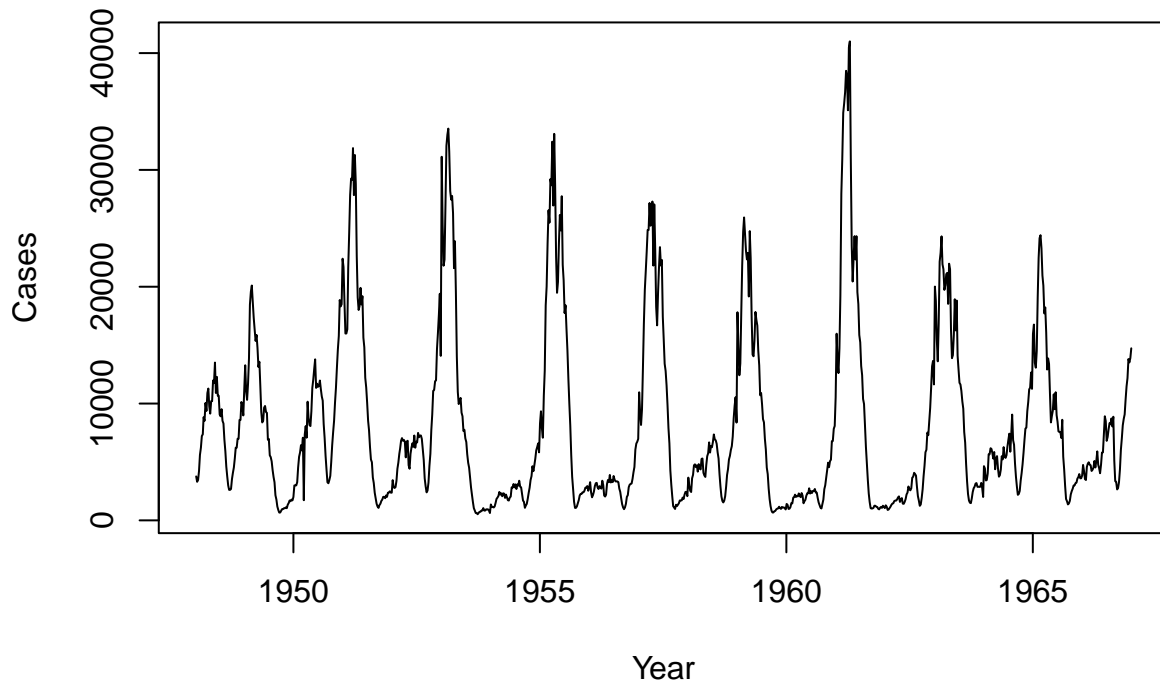


In contrast to the model with homogeneous mixing which predicts that age-intensity curves decay exponentially with age, the realistic age structure model can lead to a variety of age-incidence curves including the hump-shaped curve with a mode at around 10 years.

****PART TWO - INCORPORATING WAIFW into a compartment model**

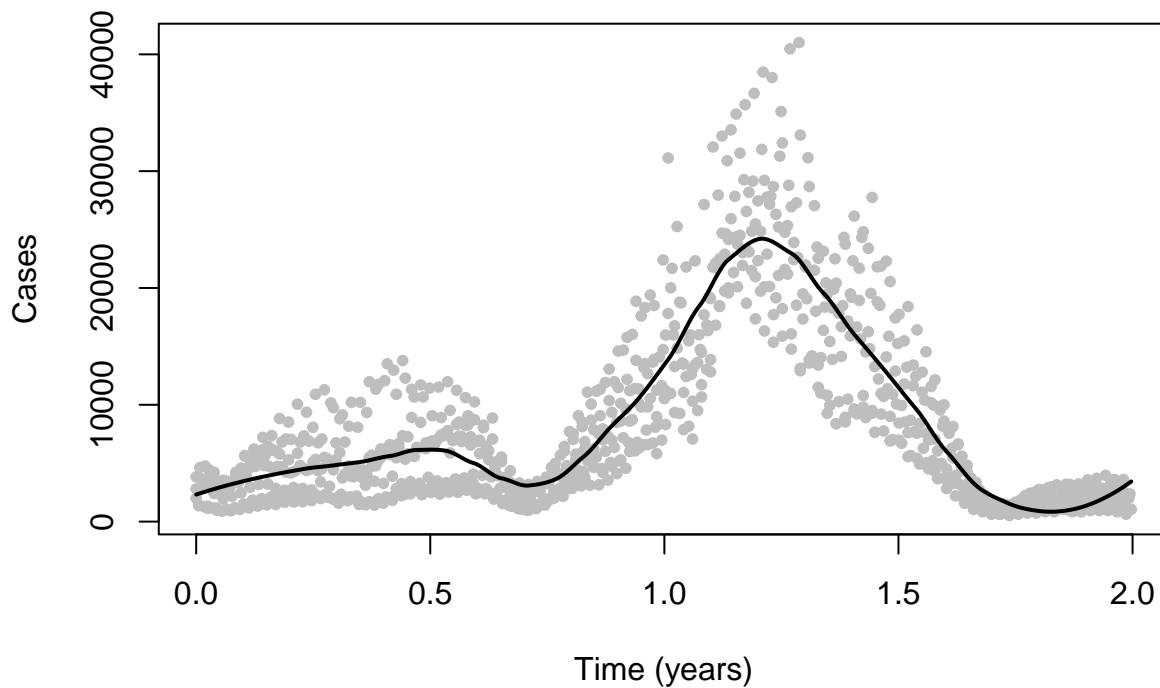
For this example, we will study the endemic dynamics of measles in England and Wales prior to vaccination, which exhibit periodic cycles. Our goal is to understand how age-structure and mixing among age-classes has given rise to these patterns. First, we look at the raw time series

```
load("/Users/amygreer/OneDrive - University of Guelph/Guelph/2022-2023 Academic Year/Teaching/Bonne Bay/
##load('data.RData')    #load the data (this is the same R data file from week 1)
plot(measles$Time,measles$Cases,type='l', xlab='Year',ylab='Cases') #plot cases over time
```



Now, we overplot each two year period to look at the predictability of this cycle. Notice that we use a local linear regression smoother loess to draw a curve through the scatterplot.

```
#plot measles cases in subsequent two year intervals using the constructed variable "TwoYEAR"
plot(measles$TwoYear,measles$Cases,type='p',pch=20,col='grey',xlab='Time (years)',ylab='Cases')
#fit a smooth line using loess -- notice data must be ordered for loess to fit properly
smooth.cases<-loess(measles$Cases[order(measles$TwoYear)]~measles$TwoYear[order(measles$TwoYear)],
                    span=0.3)
lines(smooth.cases$x,smooth.cases$fitted,lwd=2) #add smooth fit
```



One approach to modeling age structure is to allow individuals to age continuously. Such a model would consist of coupled partial differential equations. Another approach is to have individuals age all at once, once per

year (perhaps at the start of the school year, since we are concerned with the mixing of different aged children). This approach would consist of simultaneously solving a large number (several hundreds) of ordinary differential equations. Both of these approaches are cumbersome programming and prone to errors of different kinds. Therefore, we will adopt a model that is a bit more of an approximation, but nevertheless might adequately represent the key dynamical processes.

In what follows, we divide the population into four age classes, reflecting our best understanding of the main categories of transmission potential: pre-school (<5 years), primary school (6-9 years), secondary school (10-19 years), and adults (20 +). As described in the lecture portion for the model with two age classes, we allow the transmission among all age classes with pairwise rates (requiring 16 different transmission parameters—though we will assume that some of these 16 are the same. We will then have a system of 12 ODEs representing the rate of change of each class (S, E, and I) in each age group.

The equations will be solved with lsoda. Births and deaths will be represented in the usual way: births enter the youngest age class at a rate proportional to size of the population aged 20 and over, mortality occurs at rate 4.98×10^{-5} per day for the oldest age class and zero for other age classes, giving an average life span of 75 years.

Aging is accomplished once per year by moving 1 of the first age class to the second age class, 1 of the 64 second age class to the third age class, and 1 of the third age class to the fourth age class.

```
age.model<-function(t,x,parms){ #a function to return derivatives of age structured model
  S<-x[1:4]      #S are the first four elements of x
  E<-x[5:8]      #E are the next four elements of x
  I<-x[9:12]     #I are the last four elements of x
  dx<-vector(length=12) #a vector to store the derivatives
  for(a in 1:4){ #loop over age classes
    tmp <- (parms$beta[a,]*I)*S[a] #temporary variable with infection rate
    dx[a] <- parms$nu[a]*55/75 - tmp - parms$mu[a]*S[a] #dS
    dx[a+4] <- tmp - parms$sigma*E[a] - parms$mu[a]*E[a] #dE
    dx[a+8] <- parms$sigma*E[a] - parms$gamma*I[a] - parms$mu[a]*I[a] #dI
  }
  return(list(dx)) #return the result
}
```

Now we set some initial conditions and parameters. By inspection of age.model we see that the first four elements are S for the four age classes, elements 5 through 8 are E for the four age classes, and so forth. We use the R function matrix to indicate that β is a matrix object and therefore an appropriate object for multiplication with the vector I in the first line of the for loop in the function age.model

```
y0<-c(0.05, 0.01, 0.01, 0.008, 0.0001, 0.0001, 0.0001, 0.0001, 0.0001, 0.0001, 0.0001, 0.0001)
#initialize state variables

#a list of model parameters
parms<-list(beta=matrix(c(2.089, 2.089, 2.086, 2.037, 2.089, 9.336, 2.086, 2.037, 2.086, 2.086,
                          2.086, 2.037, 2.037, 2.037, 2.037, 2.037),nrow=4,byrow=TRUE),
            sigma=1/8, gamma=1/5, nu=c(1/(55*365),0,0,0), mu=c(0,0,0,1/(55*365)))
parms

## $beta
##      [,1] [,2] [,3] [,4]
## [1,] 2.089 2.089 2.086 2.037
## [2,] 2.089 9.336 2.086 2.037
## [3,] 2.086 2.086 2.086 2.037
## [4,] 2.037 2.037 2.037 2.037
##
## $sigma
```

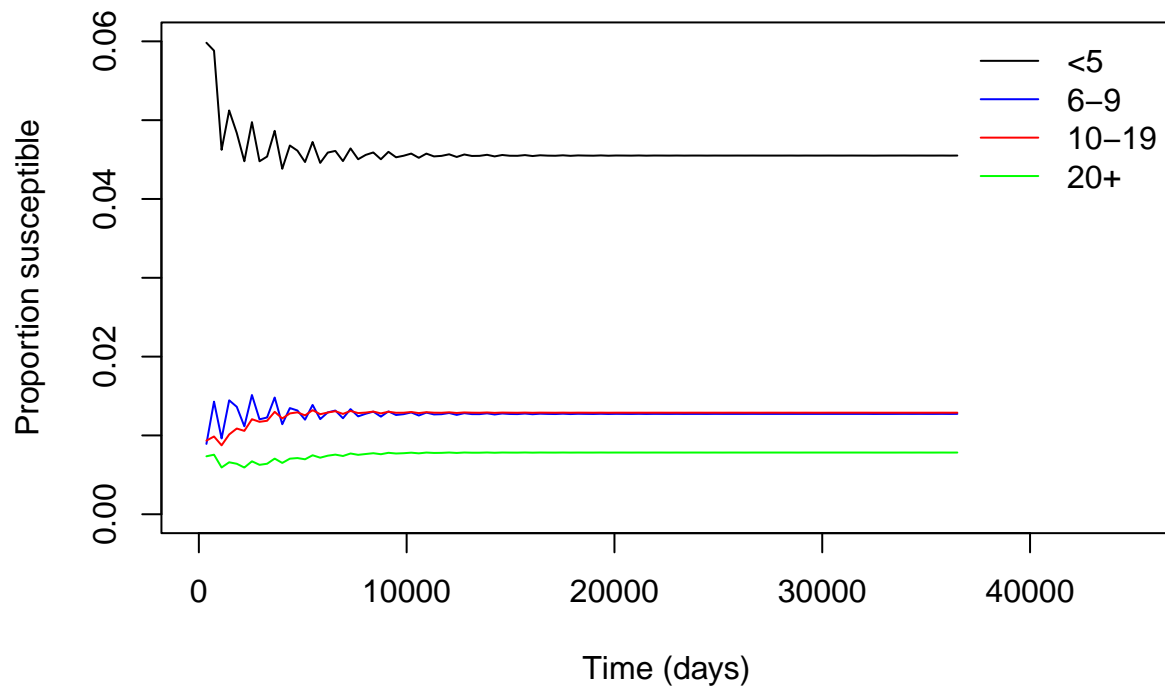
```
## [1] 0.125
##
## $gamma
## [1] 0.2
##
## $nu
## [1] 4.98132e-05 0.00000e+00 0.00000e+00 0.00000e+00
##
## $mu
## [1] 0.00000e+00 0.00000e+00 0.00000e+00 4.98132e-05
```

Finally, we solve the equations one year at a time, moving age classes up between years, and plot.

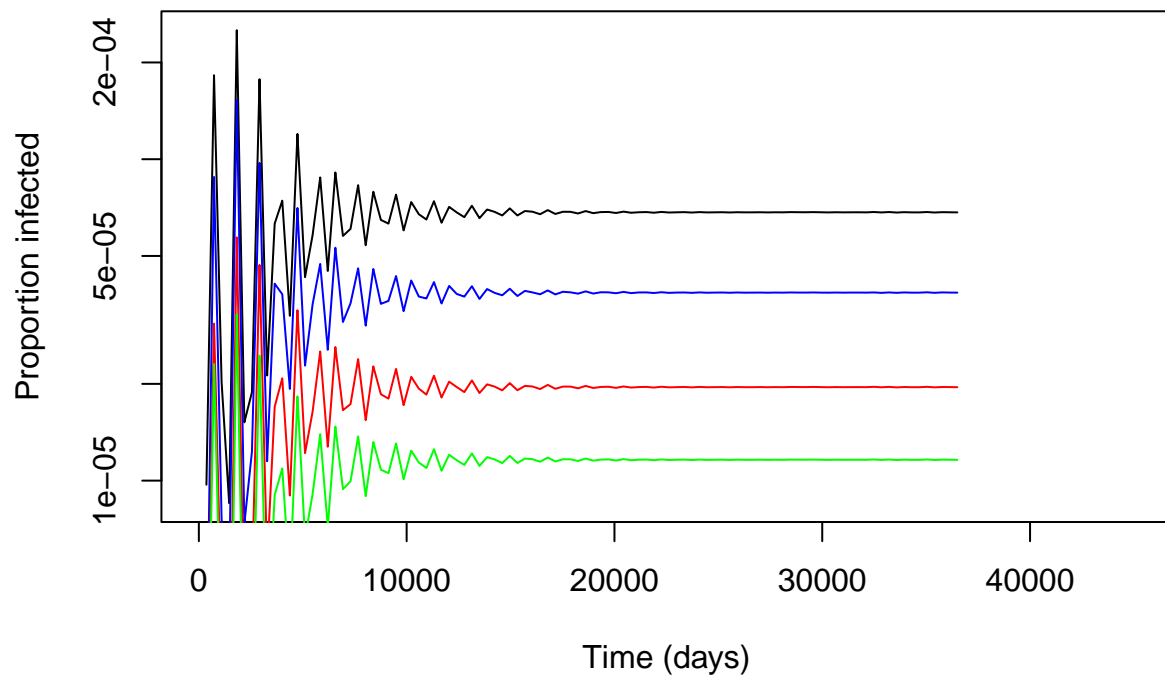
```
n=c(6,4,10,55)/75 #number of years in each age class
maxTime <- 100*365 #number of days in 100 years
T0=0 #initial time
S=c() #initialize S
E=c() #initialize E
I=c() #initialize E
T=c() #initialize T, a vector to hold times
while(T0<maxTime){ #loop over times
  y=lsoda(y0,c(T0, T0+365),age.model,parms) #solve diff'l equation for each time
  T=rbind(T, y[2,1]) #store results
  S=rbind(S, y[2,2:5])
  E=rbind(E, y[2,6:9])
  I=rbind(I, y[2,10:13])
  #Now do the yearly movements
  #Note use of "tail" to pull off the last value in a vector
  y0[1]=tail(y,1)[2]-tail(y,1)[2]/6
  y0[2]=tail(y,1)[3]+tail(y,1)[2]/6 - tail(y,1)[3]/4
  y0[3]=tail(y,1)[4]+tail(y,1)[3]/4 - tail(y,1)[4]/10
  y0[4]=tail(y,1)[5]+tail(y,1)[4]/10
  y0[5]=tail(y,1)[6]-tail(y,1)[6]/6
  y0[6]=tail(y,1)[7]+tail(y,1)[6]/6 - tail(y,1)[7]/4
  y0[7]=tail(y,1)[8]+tail(y,1)[7]/4 - tail(y,1)[8]/10
  y0[8]=tail(y,1)[9]+tail(y,1)[8]/10
  y0[9]=tail(y,1)[10]-tail(y,1)[10]/6
  y0[10]=tail(y,1)[11]+tail(y,1)[10]/6 - tail(y,1)[11]/4
  y0[11]=tail(y,1)[12]+tail(y,1)[11]/4 - tail(y,1)[12]/10
  y0[12]=tail(y,1)[13]+tail(y,1)[12]/10
  T0=tail(T,1)
}
```

Plot the results.

```
#plot
##par(mfrow=c(2,1)) #set up plotting region
plot(T,S[,1],type='l',xlim=c(0,45000),ylim=c(0,0.06),xlab='Time (days)',
      ylab='Proportion susceptible') #plot susceptibles in youngest age class
lines(T,S[,2],col='blue') #susceptibles in second age class
lines(T,S[,3],col='red') #susceptibles in third age class
lines(T,S[,4],col='green') #susceptibles in oldest age class
legend(x='topright',legend=c('<5','6-9','10-19','20+'), #add legend
       col=c('black','blue','red','green'),lty=1,bty='n')
```



```
plot(T,I[,1],type='l',log='y',xlim=c(0,45000),xlab='Time (days)', #plot infected
      ylab='Proportion infected')
lines(T,I[,2],col='blue')
lines(T,I[,3],col='red')
lines(T,I[,4],col='green')
```



PART THREE - HANDS-ON EXERCISES - Age-structured model for measles

1. Draw out a compartment model for the structured model of measles described in the text (age.model). (Make sure you make it clear that your model has multiple age groups and that individuals age in the model)
2. Review the text that describes the aging process for the model (e.g. moving 1/6 of the first age class to the second age class). Can you explain what this actually means? Where do the proportions come from?
3. Walk through the set-up of the structured model R Code. Specifically focus on the WAIFW matrix. Where does this get added in and how?
4. Take a look at the code that sets up the “aging” process in the model. What is this code doing and how?
5. Now modify the age-structured model to have greater assortativity. What does that mean in terms of the beta transmission matrix (WAIFW matrix)? In practical terms how do you do this? What are some of the important considerations?
6. When you re-run the model using the updated (more assortative) mixing matrix, what do you find and how does that compare to the previous matrix you examined?