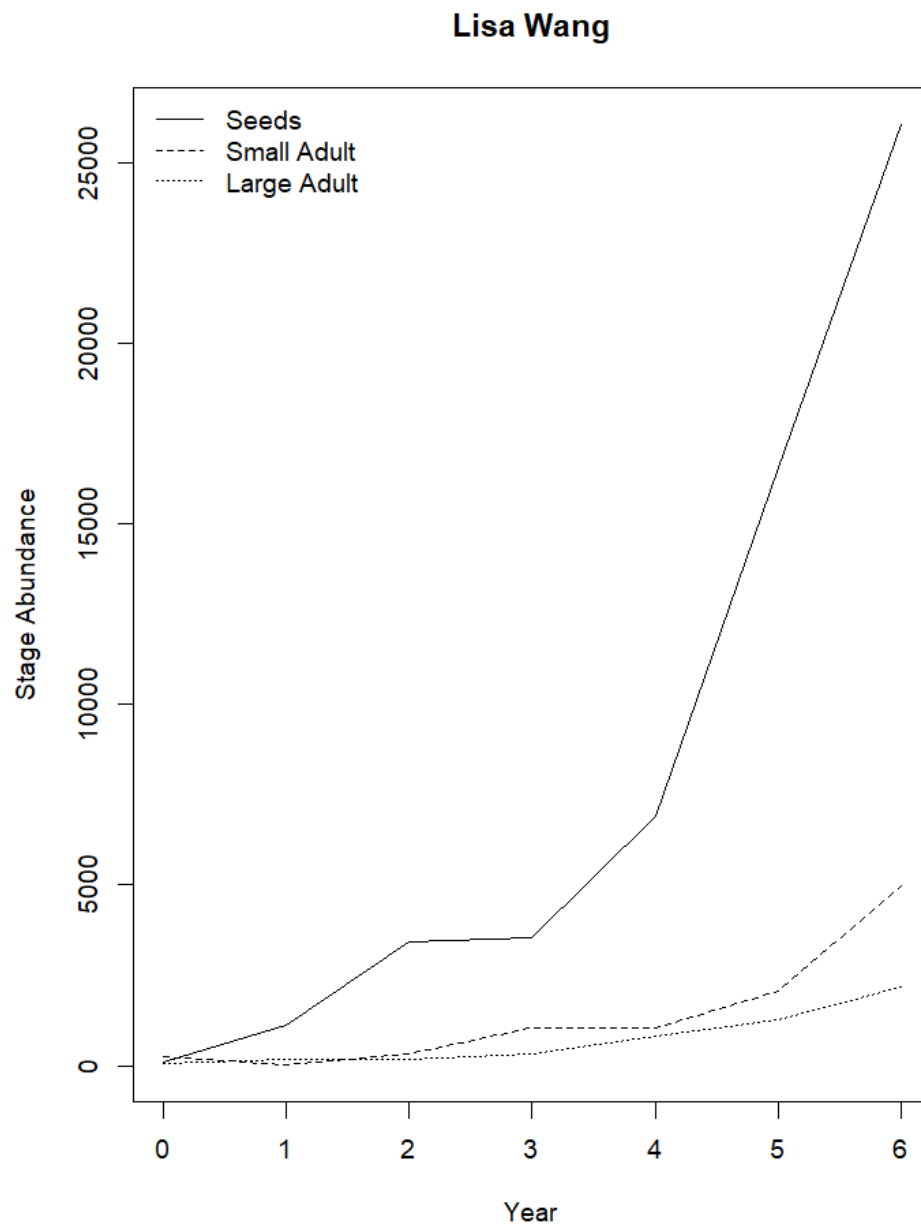
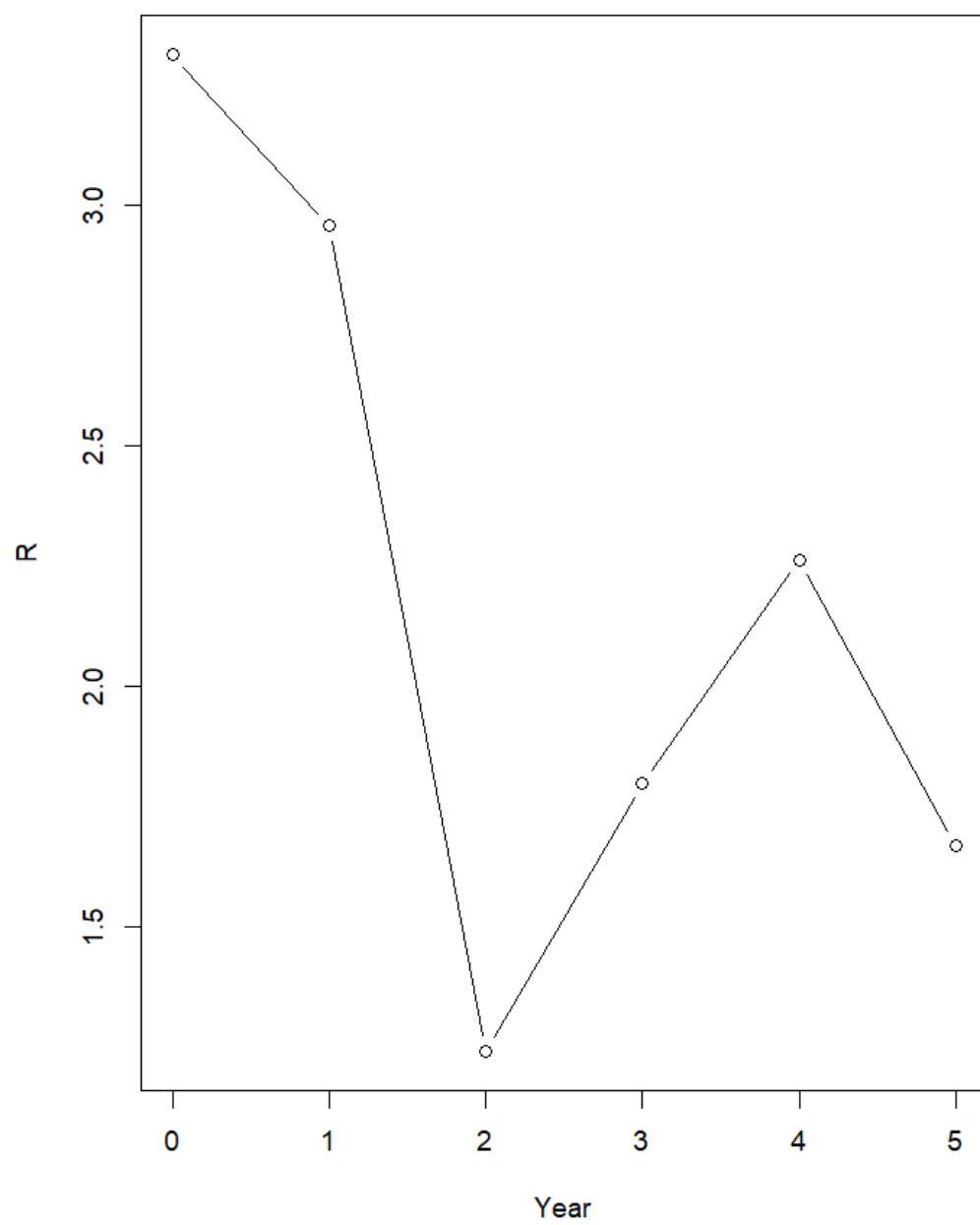


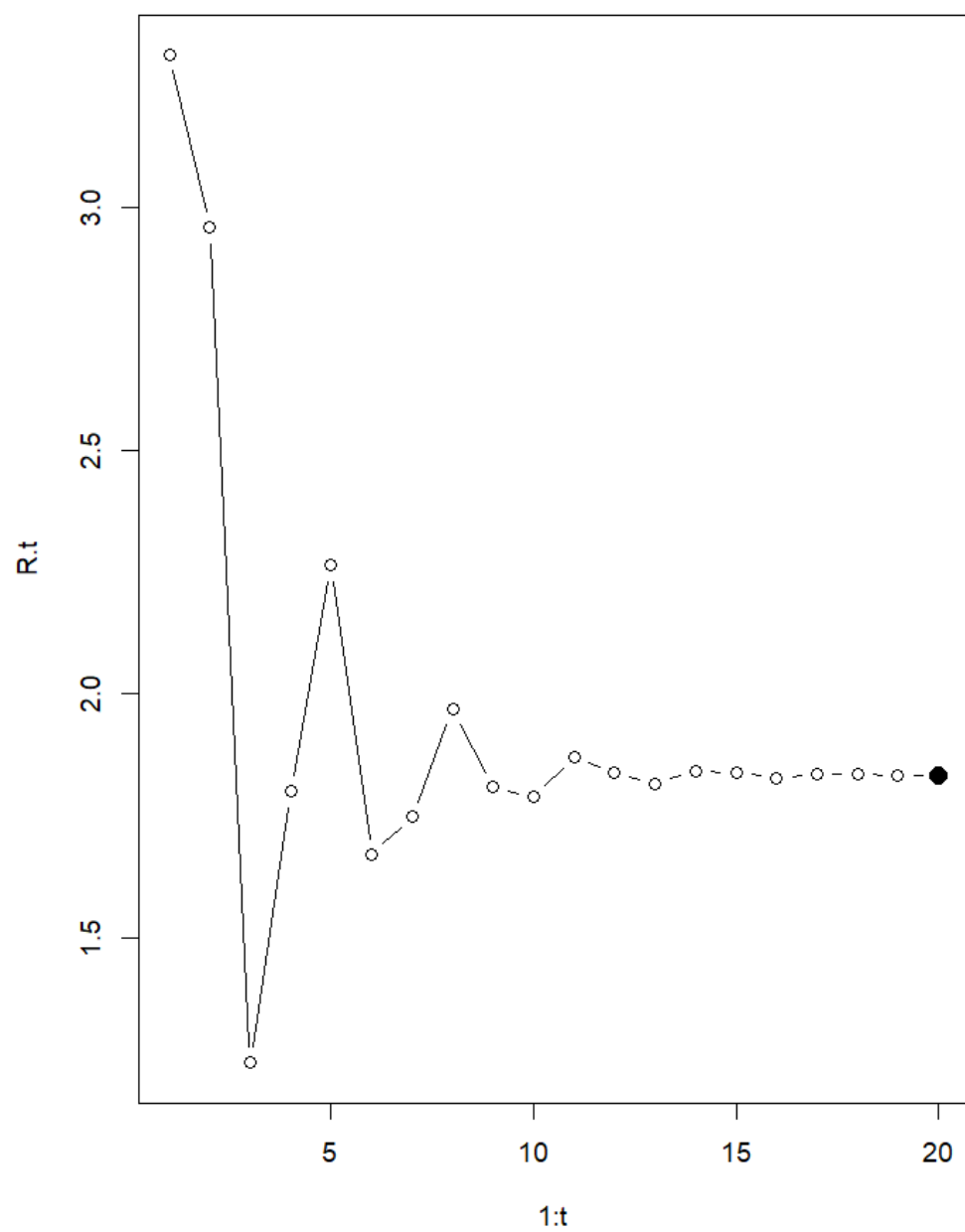
1a.



# Lisa Wang



Lisa Wang



```

1 #PS 4
2 #1a-----
3 A <- matrix(c(0, 0.5, 20, 0.3, 0, 0, 0, 0.5, 0.9), nr = 3, byrow = TRUE)
4 A
5 N0 <- matrix(c(100, 250, 50), ncol = 1)
6 N1 <- A %>% N0
7 N1
8
9 years <- 6
10 N.projections <- matrix(0, nrow = nrow(A), ncol = years + 1)
11 N.projections[,1] <- N0
12 #now we perform the iteration with the for loop
13 for (i in 1:years){
14   N.projections[, i + 1] <- A %>% N.projections[,i]
15 }
16
17 matplot(0:years, t(N.projections), type = "l", lty = 1:3, col = 1, ylab = "Stage Abundance", xlab = "Year", main = "Lisa Wang")
18 legend("topleft", legend = c("Seeds", "Small Adult", "Large Adult"), lty = 1:3, col = 1, bty = "n")
19
20
21 #Annual growth rate
22 N.totals <- apply(N.projections, 2, sum)
23 Rs <- N.totals[-1]/N.totals[-(years +1)]
24 plot(0:(years - 1), Rs, type = "b", xlab = "Year", ylab = "R", main = "Lisa Wang")
25
26 eigs.A <- eigen(A)
27 eigs.A
28
29 here1 <- eigs.A$values
30 here1 <- eigs.A$vectors
31
32 #Finding lambda
33 dom.pos <- which.max(eigs.A[["values"]])
34 L1 <- Re(eigs.A[["values"]][dom.pos])
35 L1
36
37
38 #Power iteration method of eigenanalysis
39 t <- 20
40 Nt <- N0/sum(N0)
41
42 R.t <- numeric(t)
43 for (i in 1:t) R.t[i] <- {
44   Nt1 <- A %>% Nt
45   R <- sum(Nt1)/sum(Nt)
46   Nt <- Nt1/sum(Nt1)
47   R
48 }
49 par(mar = c(5, 4, 3, 2))
50 plot(1:t, R.t, type = "b", main = "Lisa Wang")
51 points(t, L1, pch = 19, cex = 1.5)

```

1b.

```
54 #1b-----
55 w <- Re(eigs.A[["vectors"]][,dom.pos])
56 ssd <- w/sum(w)
57 round(ssd, 3)
58
59 #calculating reproductive value
60 M <- eigen(t(A))
61 v <- Re(M$vectors[, which.max(Re(M$values))])
62 RV <- v/v[1]
63 RV
64
65 #Calculating sensitivity
66 vw.s <- v %>% t(w)
67 (S <- vw.s/as.numeric(v %>% w))
68
69 #Response:
70
71 #we are interested in the demographic parameters that have the biggest influence on lambda, the long term
72 #growth rate of the population. Sensitivity describes the absolute response of lambda to absolute changes
73 #L_ij, if the rest of matrix are held constant. S_ij = d(lambda)/d(L_ij) . S31 is the highest sensitivity
74 #because the lambda increases the most if the plant can undergo survivorship
75 #directly to reproductive stage, because the greatest proportion of population is in seedling stage
76 #and adult age exhibits the largest reproductive rate. However, this is not possible because of
77 #no skipping stage 2.
```

#Response:

#we are interested in the demographic parameters that have the biggest influence on lambda, the long term

#growth rate of the population. Sensitivity describes the absolute response of lambda to absolute changes

#L<sub>ij</sub>, if the rest of matrix are held constant.  $S_{ij} = d(\lambda)/d(L_{ij})$  . S<sub>31</sub> is the highest sensitivity

#because the lambda increases the most if the plant can undergo survivorship

#directly to reproductive stage, because the greatest proportion of population is in seedling stage

#and adult age exhibits the largest reproductive rate. However, this is not possible because of

#no skipping stage 2.

#The plant does not exhibit this transition, and it means the magnitude of

#yield if all seedlings are fully surviving the second stage, and they become

#able to reproduce in adult stage.

#This is still biologically useful information, but we would prefer to use elasticity instead.

1c.

```
85 #1c-----
86 elas <- (A/L1) * S
87 round(elas, 3)
88
89 #Response:
90
91 #The question is asking whether reducing survivorship of stage 3 by 20%
92 #or reducing fecundity of stage 3 by 20% is going to have a bigger effect
93 #on the long term population growth rate lambda. |
94
95 #Based on the elasticity matrix, we know
96 #0.246 = elasticity of fecundity of stage 3 plants
97 #0.238 = elasticity of survivorship of stage 3 plants on self-survivorship, remaining in the system
98 #from these results, we can deduce that reducing the fecundity will decrease lambda more since 0.246 > 0.238
99 #we would consider the proportion change in a transition element.
100
```

#Response:

#The question is asking whether reducing survivorship of stage 3 by 20%  
#or reducing fecundity of stage 3 by 20% is going to have a bigger effect  
#on the long term population growth rate lambda.

#Based on the elasticity matrix, we know  
#0.246 = elasticity of fecundity of stage 3 plants  
#0.238 = elasticity of survivorship of stage 3 plants on self-survivorship, remaining in the system  
#from these results, we can deduce that reducing the fecundity will decrease lambda more since  
0.246 > 0.238  
#we would consider the proportion change in a transition element.

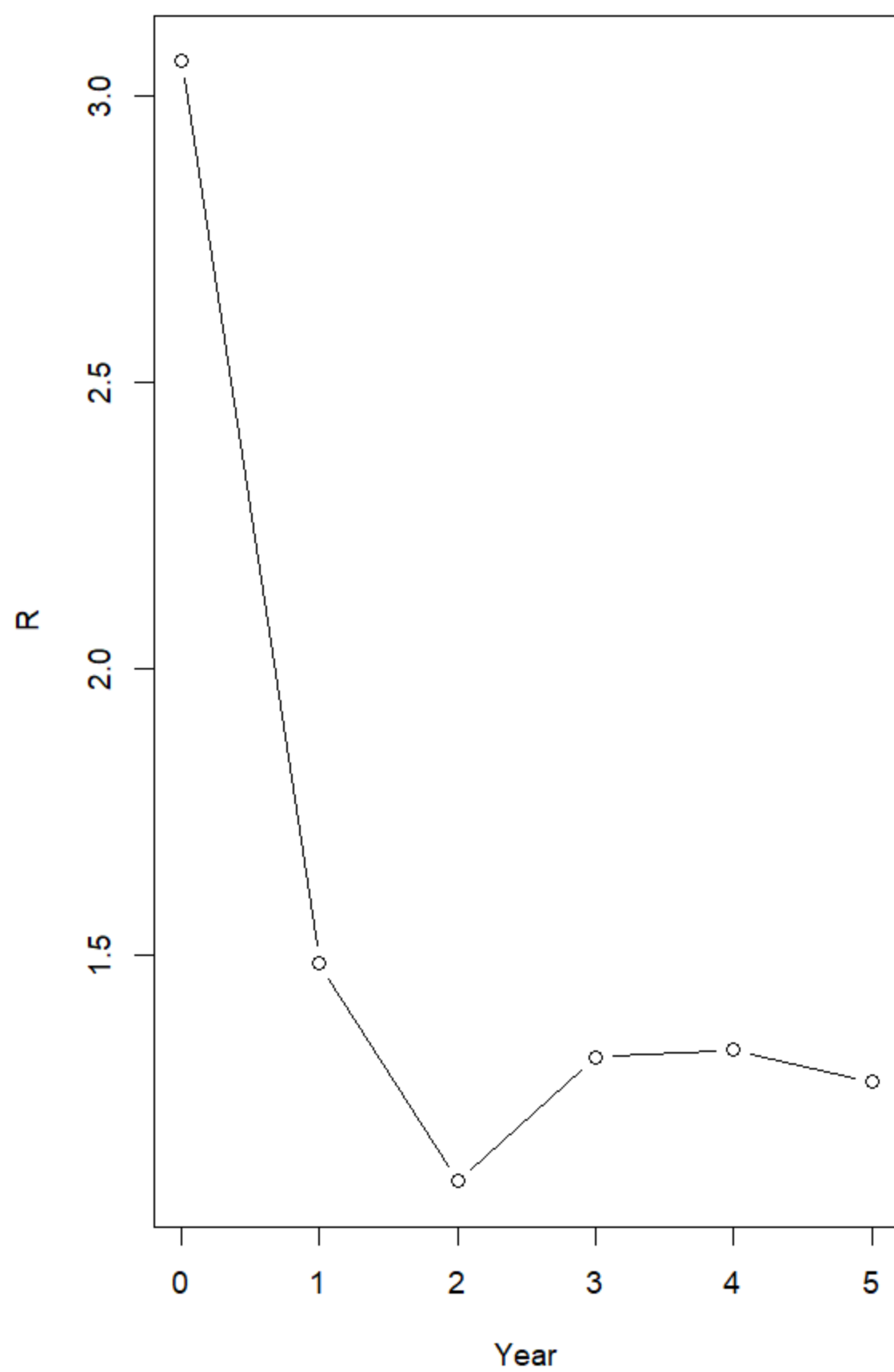
1d.

```
100
101 #1d-----
102 B <- matrix(c(0, 0.5, 20, 0.3, 0, 0, 0, 0.1, 0.9), nr = 3, byrow = TRUE)
103 B
104 N0 <- matrix(c(100, 250, 50), ncol = 1)
105 N1 <- B %*% N0
106 N1
107 |
108 #the transition parameter is reduced to 0.1, after 80% reduction
109
```

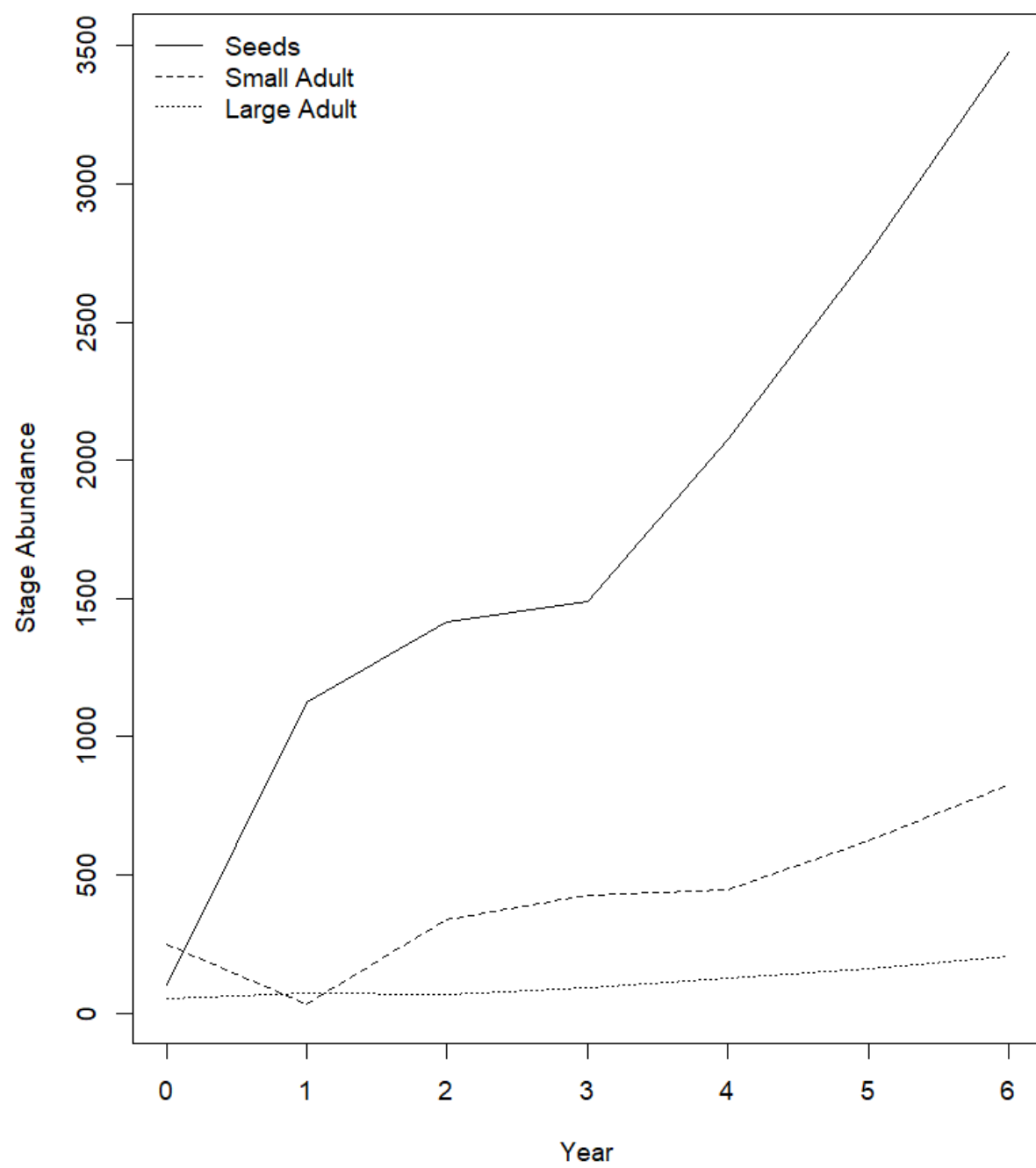
#the transition parameter is reduced to 0.1, after 80% reduction

1e.

**New plot of Annual Growth rate**



### New plot





```

110 #le-----
111 #making plots of the projected population sizes and annual growth rate
112
113 years <- 6
114 N.projections <- matrix(0, nrow = nrow(B), ncol = years + 1)
115 N.projections[,1] <- N0
116 #now we perform the iteration with the for loop
117 for (i in 1:years){
118   N.projections[, i + 1] <- B %*% N.projections[,i]
119 }
120
121 matplot(0:years, t(N.projections), type = "l", lty = 1:3, col = 1, ylab = "Stage Abundance", xlab = "Year", main = "New plot")
122 legend("topleft", legend = c("Seeds", "Small Adult", "Large Adult"), lty = 1:3, col = 1, bty = "n")
123
124
125 N.totals <- apply(N.projections, 2, sum)
126 Rs <- N.totals[-1]/N.totals[-(years +1)]
127 plot(0:(years - 1), Rs, type = "b", xlab = "Year", ylab = "R", main = "New plot of Annual Growth rate")
128
129

```

1f.

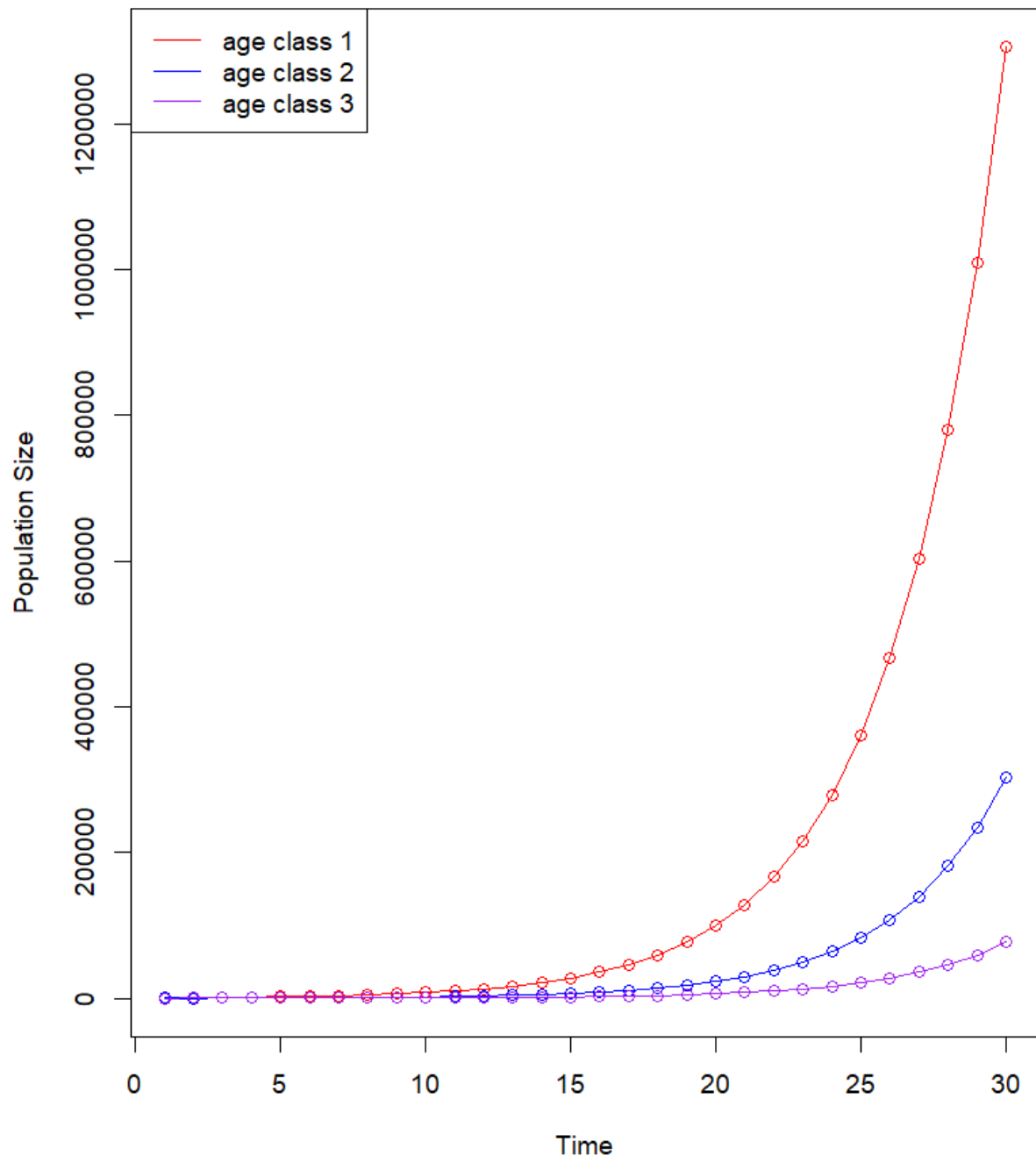
```

131 #1f-----
132 #calculating the long term population growth rate, lambda
133
134 eigs.B <- eigen(B)
135 eigs.B
136
137 here <- eigs.B$values
138 here1 <- eigs.B$vectors
139 dom.pos <- which.max(eigs.B[["values"]])
140 L1 <- Re(eigs.B[["values"]][dom.pos])
141 L1
142
143 #lambda = 1.294
144 stable_stage_distribution <- eigs.B$vectors[,which.max(abs(eigs.B$values))]
145 stable_stage_distribution
146 #the corresponding vector is 0.97255374+0i 0.22551915+0i 0.05727425+0i
147

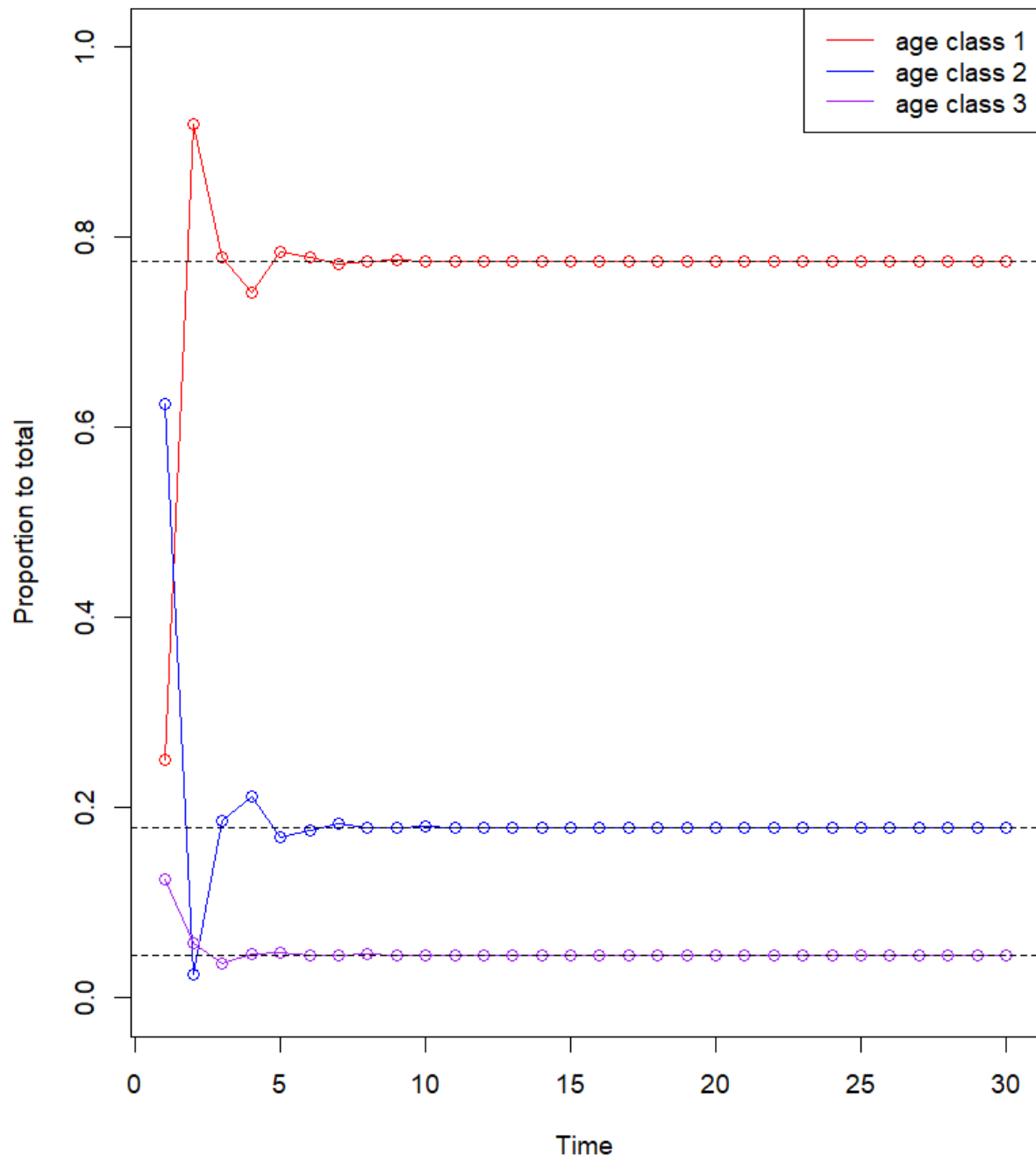
```

1g.

Population vs. Time



Prop to total vs. Time



```

148 #lg-----
149 #proportion of individuals for new projection matrix
150
151 B <- matrix(c(0, 0.5, 20, 0.3, 0, 0, 0, 0.1, 0.9), nr = 3, byrow = TRUE)
152 N0 <- matrix(c(100, 250, 50), ncol = 1) #initial population vector
153
154 pop <- matrix(NA, ncol = 30, nrow = 3)
155 pop[,1] <- N0
156
157 #loop over time steps of interest
158 for(ii in 2:30){
159   pop[,ii] <- B %>% pop[, ii -1]
160 }
161
162 plot(seq(1,30), pop[,1,], type = "o", xlab = 'Time', ylab = 'Population Size', col = 'red', main = 'Population vs. Time')
163 lines(seq(1,30), pop[,2,], type = "o", xlab = 'Time', ylab = 'Population Size', col = 'blue', main = 'Population vs. Time')
164 lines(seq(1,30), pop[,3,], type = "o", xlab = 'Time', ylab = 'Population Size', col = 'purple', main = 'Population vs. Time')
165 legend(x = "topleft", legend = c("age class 1", "age class 2", "age class 3"), col = c('red', 'blue', 'purple'), lty = 1)
166
167 totalPop <- colSums(pop) #calculates total population sizes
168 totalPop
169 juvProp <- pop[,1,]/totalPop #calculates proportion of juveniles
170 twoProp <- pop[,2,]/totalPop
171 threeProp <- pop[,3,]/totalPop
172 plot(seq(1,30), juvProp, ylim = c(0,1), type = "o", xlab = 'Time', ylab = 'Proportion to total', col = 'red', main = 'Prop to total vs. Time')
173 lines(seq(1,30), twoProp, type = "o", xlab = 'Time', ylab = 'Proportion to total', col = 'blue', main = 'Prop to total vs. Time')
174 lines(seq(1,30), threeProp, type = "o", xlab = 'Time', ylab = 'Proportion to total', col = 'purple', main = 'Prop to total vs. Time')
175 legend(x = "topright", legend = c("age class 1", "age class 2", "age class 3"), col = c('red', 'blue', 'purple'), lty = 1)
176
177
178 eigs.B$eigenvectors[,1]/sum(eigs.B$eigenvectors[,1])
179 #prints 0.775, 0.179, 0.045
180
181 abline(h = 0.77472892, lty = "dashed")
182 abline(h = 0.17964685, lty = "dashed")
183 abline(h = 0.04562423, lty = "dashed")
184
185 #We have shown, over 30 iterations, that the proportion of age groups approaches stability
186

```

1h.

```

187 #1h-----
188 #calculating sensitivity and elasticity |
189 w <- Re(eigs.B[["vectors"]][,dom.pos])
190 ssd <- w/sum(w)
191 round(ssd, 3)
192
193 #calculating reproductive value
194 M <- eigen(t(B))
195 v <- Re(M$eigenvectors[, which.max(Re(M$values))])
196 RV <- v/v[1]
197 RV
198
199 #Calculating sensitivity matrix
200 vw.s <- v %>% t(w)
201 (S <- vw.s/as.numeric(v %>% w))
202
203 #Calculating elasticity matrix
204
205 elas <- (A/L1) * S
206 round(elas, 3)
207
208 #The proportional change I would make to the mugwort
209 #population would be introducing some external agent that
210 #reduces the survivorship from year 1 to year 2. Specifically, because
211 #the elasticity of s12 = 0.912, highest of all elements
212 #and the sensitivity of s12 = 2.35, highest of all elements. I would
213 #reduce by 100%.
214

```

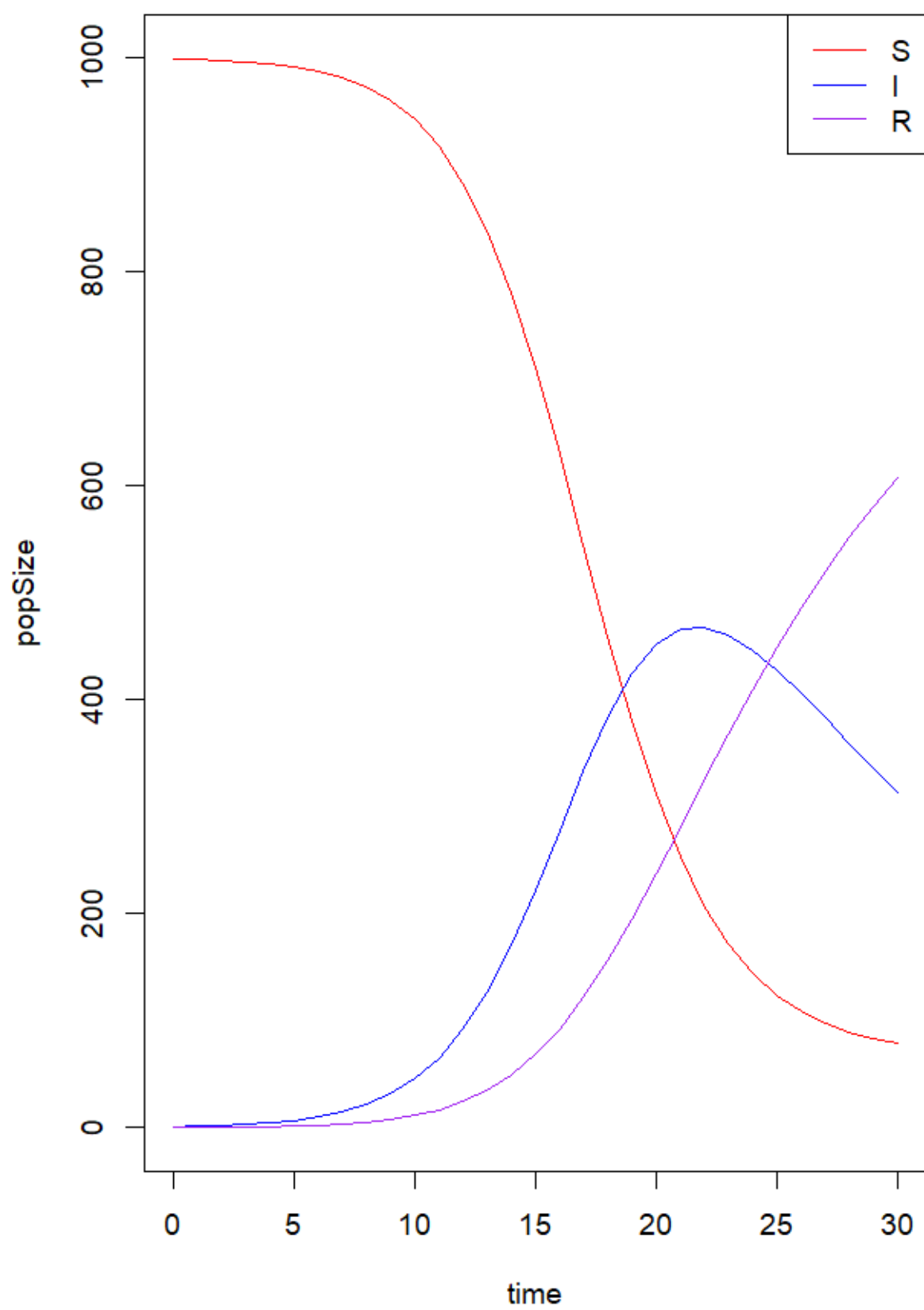
1i.

```
215 #1i-----
216 #recalculating lambda, except projection matrix changed from
217 C <- matrix(c(0, 0.5, 20, 0.3, 0, 0, 0, 0.1, 0.45), nr = 3, byrow = TRUE)
218 C
219
220 eigs.C <- eigen(C)
221 eigs.C
222
223 eigenvalues <- eigs.C$values
224 eigenvectors <- eigs.C$vectors
225 dom.pos <- which.max(eigs.C[["values"]])
226 L1 <- Re(eigs.C[["values"]][dom.pos])
227 L1
228
229 #lambda = 1.0626
230 stable_stage_distribution1 <- eigs.B$vectors[,which.max(abs(eigs.B$values))]
231 stable_stage_distribution1
232 #This is still a growing population in long term speculation, because
233 #it is larger than 1
234
235 D <- matrix(c(0, 0.5, 20, 0.3, 0, 0, 0, 0.1, 0.293), nr = 3, byrow = TRUE)
236 D
237
238 eigs.D <- eigen(D)
239 eigs.D
240
241 eigenvalues <- eigs.D$values
242 eigenvectors <- eigs.D$vectors
243 dom.pos <- which.max(eigs.D[["values"]])
244 L1 <- Re(eigs.D[["values"]][dom.pos])
245 L1
246
247 #lambda = 0.999
248 stable_stage_distribution1 <- eigs.D$vectors[,which.max(abs(eigs.D$values))]
249 stable_stage_distribution1
250 #This lambda value is now lower than 1, so it looks like a over 63% reduction
251 # of adult survivorship would result in decreasing population of mugworts.
252
253 "
```

Question 2

2a.

PopSize vs. Time



```

253 #2a-----
254 #I am choosing the SIR model, with Initial Conditions, parameters, and can return
255 #a vector of values as a function of time
256 install.packages("deSolve")
257 library("deSolve")
258
259 IC <- c(S = 999, I = 1, R = 0)
260 N <- 1000
261 times <- seq(0, 30, 1) #Create a sequence of times
262 pars <- c(beta = 0.5, gamma = 0.1, mu = 0.01)
263
264 SIRODE <- function(t, vars, pars) {
265   with(as.list(c(vars, pars)), {
266     #This is a function that allows you to refer to the named
267     #parameters inside the vector pars directly by their name, without having to index pars.
268     dSdt <- mu*N - ((beta*S*I)/N) - (mu*S)
269     dIdt <- (beta*S*I)/N - mu*I - gamma*I
270     dRdt <- gamma*I - mu*R
271     return (list(c(dSdt, dIdt, dRdt)))
272   })
273 }
274 output <- lsoda(IC, times, SIRODE, pars, rtol = 1e-6, atol = 1e-6)
275 head(output)
276 #Then make the plot
277
278 plot(output[,1], output[,2], type = "l", ylim = c(0,1000), xlab = "time", col = "red", ylab = "popSize", main = "PopSize vs. Time")
279 lines(output[,1], output[,3], type = "l", xlab = "time", col = "blue", ylab = "popSize", main = "PopSize vs. Time")
280 lines(output[,1], output[,4], type = "l", xlab = "time", col = "purple", ylab = "popSize", main = "PopSize vs. Time")
281
282 legend(x = "topright", legend = c("S", "I", "R"), col = c("red", "blue", "purple"), lty = 1)
283

```

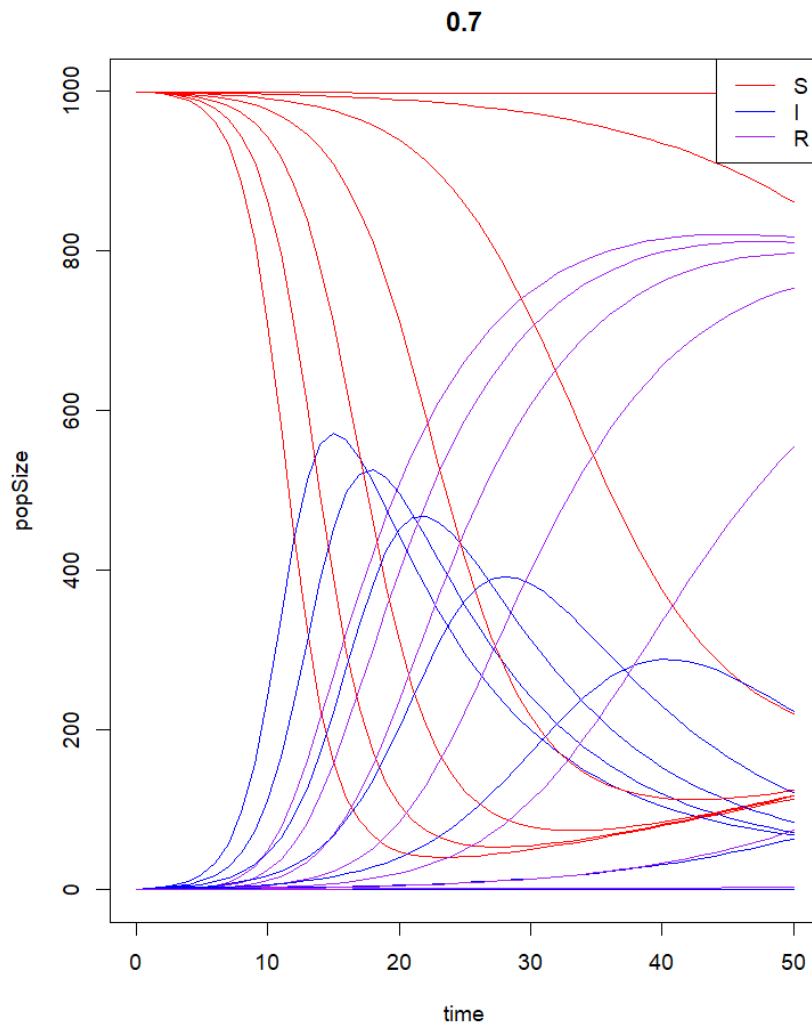
2b.

```

284 #2b-----
285 #variation of key parameter: beta
286
287 IC <- c(S = 999, I = 1, R = 0)
288 N <- 1000
289 times <- seq(0, 50, 1) #Create a sequence of times
290 pars <- c(beta = 0.5, gamma = 0.1, mu = 0.01)
291
292 SIRODE <- function(t, vars, pars) {
293   with(as.list(c(vars, pars)), {
294     #This is a function that allows you to refer to the named
295     #parameters inside the vector pars directly by their name, without having to index pars.
296     dSdt <- mu*N - ((beta*S*I)/N) - (mu*S)
297     dIdt <- (beta*S*I)/N - mu*I - gamma*I
298     dRdt <- gamma*I - mu*R
299     return (list(c(dSdt, dIdt, dRdt)))
300   })
301 }
302 output <- lsoda(IC, times, SIRODE, pars, rtol = 1e-6, atol = 1e-6)
303 head(output)
304 #Then make the plot
305
306 plot(output[,1], output[,2], type = "l", ylim = c(0,1000), xlab = "time", col = "red", ylab = "popSize", main = "PopSize vs. Time")
307 lines(output[,1], output[,3], type = "l", xlab = "time", col = "blue", ylab = "popSize", main = "PopSize vs. Time")
308 lines(output[,1], output[,4], type = "l", xlab = "time", col = "purple", ylab = "popSize", main = "PopSize vs. Time")
309
310 legend(x = "topright", legend = c("S", "I", "R"), col = c("red", "blue", "purple"), lty = 1)
311
312 install.packages("plotly")
313 library(plotly)
314
315 batch <- function(trials){
316   pars <- c(beta = trials*0.1, gamma = 0.1, mu = 0.01)
317   output <- lsoda(IC, times, SIRODE, pars, rtol = 1e-6, atol = 1e-6)
318   plot(output[,1], output[,2], type = "l", ylim = c(0,1000), xlab = "time", col = "red", ylab = "popSize", main = trials*0.1)
319   while(trials > 0){
320     pars <- c(beta = trials*0.1, gamma = 0.1, mu = 0.01)
321     output <- lsoda(IC, times, SIRODE, pars, rtol = 1e-6, atol = 1e-6)
322
323     lines(output[,1], output[,2], type = "l", ylim = c(0,1000), xlab = "time", col = "red", ylab = "popSize", main = trials*0.1)
324     lines(output[,1], output[,3], type = "l", xlab = "time", col = "blue", ylab = "popSize")
325     lines(output[,1], output[,4], type = "l", xlab = "time", col = "purple", ylab = "popSize")
326     legend(x = "topright", legend = c("S", "I", "R"), col = c("red", "blue", "purple"), lty = 1)
327     trials = trials - 1
328   }
329 }
330
331 batch(7) #running the batch function
332
333 #Commentary on biological significance: It looks like that the effects of the SIR model
334 #simply propagates out. Beta is the disease transmission rate, and the higher the beta value
335 #the faster the drop of S curve and sharper the peak for Infected. Therefore, we
336 #notice the height of the I peak go higher and higher.
337

```

#Commentary on biological significance: It looks like that the effects of the SIR model  
#simply propagates out. Beta is the disease transmission rate, and the higher the beta value  
#the faster the drop of S curve and sharper the peak for Infected. Therefore, we  
#notice the height of the I peak go higher and higher.





2c.

```
337
338 # 2c-----
339 #changing the initial conditions
340
341 initial <- function(trials){
342   IC <- c(S = 999 - trials*30, I = 1 + trials*30, R = 0)
343   N <- 1000
344   times <- seq(0, 30, 1) #Create a sequence of times
345   pars <- c(beta = 0.5, gamma = 0.1, mu = 0.01)
346
347   output <- lsoda(IC, times, SIRODE, pars, rtol = 1e-6, atol = 1e-6)
348
349   plot(output[,1], output[,2], type = "l", ylim = c(0,1000), xlab = "time", col = "red", ylab = "popSize", main = "PopSize vs. Time")
350
351   while(trials > 0){
352     IC <- c(S = 999 - trials*30, I = 1 + trials*30, R = 0)
353     output <- lsoda(IC, times, SIRODE, pars, rtol = 1e-6, atol = 1e-6)
354     lines(output[,1], output[,2], type = "l", ylim = c(0,1000), xlab = "time", col = "red", ylab = "popSize", main = "PopSize vs. Time")
355     lines(output[,1], output[,3], type = "l", xlab = "time", col = "blue", ylab = "popSize", main = "PopSize vs. Time")
356     lines(output[,1], output[,4], type = "l", xlab = "time", col = "purple", ylab = "popSize", main = "PopSize vs. Time")
357     legend(x = "topright", legend = c("S", "I", "R"), col = c("red", "blue", "purple"), lty = 1)
358     trials = trials -1
359   }
360 }
361
362 initial(4)
363
364 #Commentary on biological significance: whereas the shape of the curves were actually different
365 #for changing beta, changing the initial conditions seems to have shifted the curves
366 #incrementally to the right more. As we decrease the number of susceptibles and increasing the
367 # count for infected, there seem to be similar shifting behavior on recovered individuals.
368
369
370
```

#Commentary on biological significance: whereas the shape of the curves were actually different  
#for changing beta, changing the initial conditions seems to have shifted the curves  
#incrementally to the right more. As we decrease the number of susceptibles and increasing the  
# count for infected, there seem to be similar shifting behavior on recovered individuals.

PopSize vs. Time

