MATH 142: Mathematical Modeling, Homework 2

Darren Tsang 405433124

Produced on Wednesday, Apr. 15 2020 @ 03:01:15 AM

```r
library(ggplot2)
```

Question 1

Part A

From $N_{t+1} = (1 + R_0)N_t = N_t + R_0N_t$, you notice that the population increases by R_0N_t. Thus, you can easily derive $\frac{dN}{dt} = R_0N$.

Part B

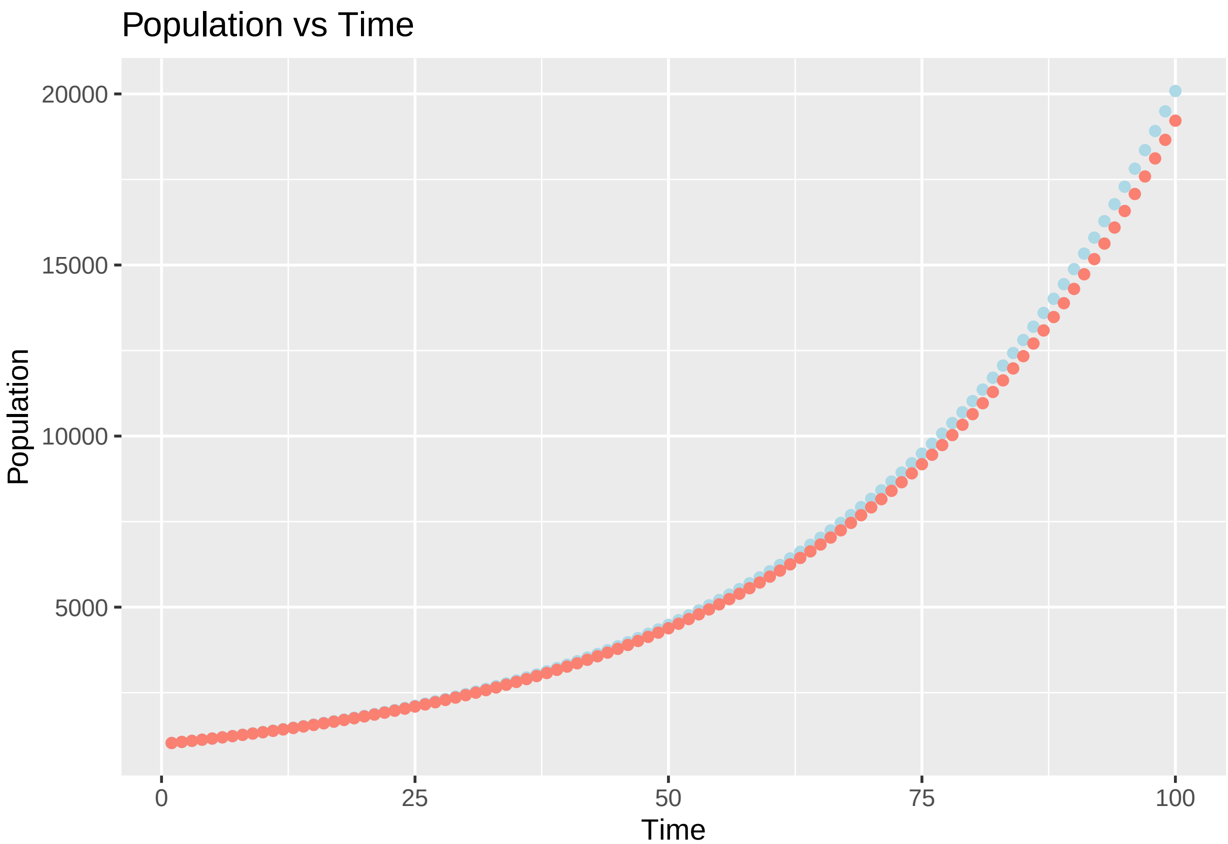
$$\frac{dN}{dt} = R_0N$$
$$\rightarrow \int \frac{dN}{N} = \int R_0 dt$$
$$\rightarrow ln(N) = R_0t + c$$
$$\rightarrow N = ce^{R_0t}$$
$$\rightarrow N(t_0) = c_1e^{R_0t_0} = N_0$$
$$\rightarrow c_1 = \frac{N_0}{e^{R_0t_0}} = \frac{1000}{e^{.03*0}} = 1000$$

Thus, $N(t) = 1000e^{.03t}$

Part C

```r
x <- c(1:100)
ode <- 1000*exp(.03*x)
reccurence <- (1.03)^x*1000

ggplot(aes(x = x, y = ode), data = NULL) +
  geom_point(color = "lightblue") +
  geom_point(aes(x = x, y = reccurence), data = NULL, color = "salmon") +
  labs(title="Population vs Time", x ="Time", y = "Population")
```



From the graph, I notice that the two solutions are very similar, but eventually, it becomes clear that the solution to the reccurence equation grows slower as time increases.

Question 2

Part A

To find the eigenvalues, $det(L - \lambda I) = det(\begin{bmatrix} 1-\lambda & 1 \\ \frac{1}{2} & \frac{3}{2}-\lambda \end{bmatrix}) = (1-\lambda)(\frac{3}{2}-\lambda) - \frac{1}{2} = \lambda^2 - \frac{5}{2}\lambda + 1 = \frac{1}{2}(2x-1)(x-2)$. Thus, $\lambda_1 = \frac{1}{2}, \lambda_2 = 2$.

The largest eigenvalue in magnitude is $\lambda_2 = 2 > 1$, which means the population will grow without bounds.

Part B

To find the eigenvector for $\lambda_1 = \frac{1}{2}$, $L - \lambda_1 I = \begin{bmatrix} 1-\frac{1}{2} & 1 \\ \frac{1}{2} & \frac{3}{2}-\frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 1 \\ \frac{1}{2} & 1 \end{bmatrix}$. Thus, the eigenvector is $\begin{bmatrix} -2 \\ 1 \end{bmatrix}$.

To find the eigenvector for $\lambda_2 = 2$, $L - \lambda_1 I = \begin{bmatrix} 1-2 & 1 \\ \frac{1}{2} & \frac{3}{2}-2 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix}$. Thus, the eigenvector is $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

$$\begin{bmatrix} 1 \\ 4 \end{bmatrix} = a \begin{bmatrix} -2 \\ 1 \end{bmatrix} + b \begin{bmatrix} 1 \\ 1 \end{bmatrix} \rightarrow \begin{cases} 1 & = & -2a+b \\ 4 & = & a+b \end{cases} \rightarrow \begin{cases} a & = & 1 \\ b & = & 3 \end{cases}$$

$$N_k = \begin{bmatrix} 1 & 1 \\ \frac{1}{2} & \frac{3}{2} \end{bmatrix}^k (1 \begin{bmatrix} -2 \\ 1 \end{bmatrix} + 3 \begin{bmatrix} 1 \\ 1 \end{bmatrix}) = (\frac{1}{2})^k \begin{bmatrix} -2 \\ 1 \end{bmatrix} + 3(2)^k \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$N_4 = (\frac{1}{2})^4 \begin{bmatrix} -2 \\ 1 \end{bmatrix} + 3(2^4) \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -.125 \\ .0625 \end{bmatrix} + \begin{bmatrix} 48 \\ 48 \end{bmatrix} = \begin{bmatrix} 47.875 \\ 48.0625 \end{bmatrix}$$

```r
L <- matrix(data = c(1, 1, 1/2, 3/2), nrow = 2, byrow = T)
(L %*% L %*% L %*% L) %*% (1*matrix(c(-2, 1)) + 3*matrix(c(1, 1)))
```

```
##         [,1]
## [1,] 47.8750
## [2,] 48.0625
```

Part C

The eigenvector associated with the largest eigenvalue in magnitude is $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$. This means that the stable age distribution is $\begin{bmatrix} \frac{1}{1+1} \\ \frac{1}{1+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$.

Question 3

Part A

$L = \begin{bmatrix} 0 & b \\ s & 1-m \end{bmatrix}$

Part B

$det(L - \lambda I) = det(\begin{bmatrix} 0-\lambda & .9 \\ .4 & .9-\lambda \end{bmatrix}) = (-\lambda)(.9-\lambda) - .36 = \lambda^2 - .9\lambda - .36 = (\lambda + .3)(\lambda - 1.2) \doteq 0 \rightarrow \lambda_1 = -.3, \lambda_2 = 1.2$ Thus, the long-term growth factor is 1.2.

$L - 1.2I = \begin{bmatrix} -1.2 & .9 \\ .4 & -.3 \end{bmatrix} \rightarrow u_2 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$. Thus, the stable age distribution is $\begin{bmatrix} \frac{3}{3+4} \\ \frac{4}{3+4} \end{bmatrix} = \begin{bmatrix} \frac{3}{7} \\ \frac{4}{7} \end{bmatrix}$

Question 4

Part A

$L_{1,1}$: m of all cells die in each day and t of juvinile become tethered, $(1-m)(1-t)$

$L_{1,2}$: it is impossible to go from juvinile to freely swimming, 0

$L_{1,3}$: m of all cells die in each day and u of the tethered become freely swimming, $(1-m)u$

$L_{2,1}$: freely swimming does not give birth to juvinile, 0

$L_{2,2}$: m of all cells die in each day and c of the juvinile become tethered, $(1-m)(1-c)$

$L_{2,3}$: m of all cells die in each day and b of the tethered will divide and create a juvinile, $(1-m)b$

$L_{3,1}$: m of all cells die in each day, $(1-m)$

$L_{3,2}$: m of all cells die in each day and c of juvinile become tethered, $(1-m)c$

$L_{3,3}$: m of all cells die in each day and u tethered become freely swimming , $(1-m)(1-u)$

Thus, $L = \begin{bmatrix} (1-m)(1-t) & 0 & (1-m)u \\ 0 & (1-m)(1-c) & (1-m)b \\ (1-m)t & (1-m)c & (1-m)(1-u) \end{bmatrix}$

Part B

$L = \begin{bmatrix} .6 & 0 & .7 \\ 0 & .4 & .5 \\ .3 & .5 & .2 \end{bmatrix}$

(i)

```r
L <- matrix(c(.6, 0, .7, 0, .4, .5, .3, .5, .2), nrow = 3, byrow = T)
L
```

```
##      [,1] [,2] [,3]
## [1,]  0.6  0.0  0.7
## [2,]  0.0  0.4  0.5
## [3,]  0.3  0.5  0.2
```

```r
eigen(L)
```

```
## eigen() decomposition
## $values
## [1]  1.0504880  0.5021306 -0.3526186
##
## $vectors
##            [,1]       [,2]       [,3]
## [1,] -0.7764170  0.8197641 -0.5220384
## [2,] -0.3840705 -0.5611153 -0.4719745
## [3,] -0.4996665 -0.1146141  0.7104336
```

From above, $\lambda_1 = 1.05$, $\vec{u_1} = \begin{bmatrix} .78 \\ .38 \\ .5 \end{bmatrix}$,

$\lambda_2 = .5$, $\vec{u_2} = \begin{bmatrix} .82 \\ -.56 \\ -.11 \end{bmatrix}$,

$\lambda_3 = -.35$, $\vec{u_3} = \begin{bmatrix} -.52 \\ -.47 \\ .71 \end{bmatrix}$

The largest eigenvalue in magnitude is $\lambda_1 = 1.05 > 1$, which means the population will grow without bounds.

The stable fraction is $\begin{bmatrix} \frac{.78}{.78+.38+.5} \\ \frac{.38}{.78+.38+.5} \\ \frac{.5}{.78+.38+.5} \end{bmatrix} = \begin{bmatrix} .47 \\ .23 \\ .3 \end{bmatrix}$

(ii)

```r
results <- t(as.matrix(c(0, 100, 0)))
results <- rbind(results, t(L %*% as.matrix(c(0, 100, 0))))

L_new <- L
for(i in 1:9){
  L_new <- L_new %*% L
  results <- rbind(results, t(L_new %*% as.matrix(c(0, 100, 0))))
}

colnames(results) <- c("F", "J", "T")
results <- cbind("k" = 0:10, results)
round(results, 4)
```

```
##        k       F        J       T
##  [1,]  0  0.0000 100.0000  0.0000
##  [2,]  1  0.0000  40.0000 50.0000
##  [3,]  2 35.0000  41.0000 30.0000
##  [4,]  3 42.0000  31.4000 37.0000
##  [5,]  4 51.1000  31.0600 35.7000
##  [6,]  5 55.6500  30.2740 38.0000
##  [7,]  6 59.9900  31.1096 39.4320
##  [8,]  7 63.5964  32.1598 41.4382
##  [9,]  8 67.1646  33.5830 43.4465
## [10,]  9 70.7113  35.1565 45.6302
## [11,] 10 74.3679  36.8777 47.9177
```

```r
F.Growth <- c()
J.Growth <- c()
T.Growth <- c()

F.Fraction <- c()
J.Fraction <- c()
T.Fraction <- c()

for(i in 9:11){
```

```r
  F.Growth <- cbind(F.Growth, results[i, "F"] / results[i - 1, "F"])
  J.Growth <- cbind(J.Growth, results[i, "J"] / results[i - 1, "J"])
  T.Growth <- cbind(T.Growth, results[i, "T"] / results[i - 1, "T"])

  F.Fraction <- cbind(F.Fraction, results[i, "F"] / sum(results[i, c("F", "J", "T")]))
  J.Fraction <- cbind(J.Fraction, results[i, "J"] / sum(results[i, c("F", "J", "T")]))
  T.Fraction <- cbind(T.Fraction, results[i, "T"] / sum(results[i, c("F", "J", "T")]))
}


growth <- cbind(8:10,
                t(F.Growth),
                t(J.Growth),
                t(T.Growth),
                t(F.Fraction),
                t(J.Fraction),
                t(T.Fraction))

colnames(growth) <- c("k",
                      "F_Growth",
                      "J_Growth",
                      "T_Growth",
                      "F_Fraction",
                      "J_Fraction",
                      "T_Fraction")
round(growth, 4)
```

```
##       k F_Growth J_Growth T_Growth F_Fraction J_Fraction T_Fraction
## [1,]  8   1.0561   1.0443   1.0485     0.4658     0.2329     0.3013
## [2,]  9   1.0528   1.0469   1.0503     0.4667     0.2321     0.3012
## [3,] 10   1.0517   1.0490   1.0501     0.4672     0.2317     0.3011
```

Ignoring rounding error, these numbers agree with the results I obtained in part (i).