$\frac{68}{110}$ = 62%

Workbook 2 – Epidemiology

Jamie Hall (jah79)

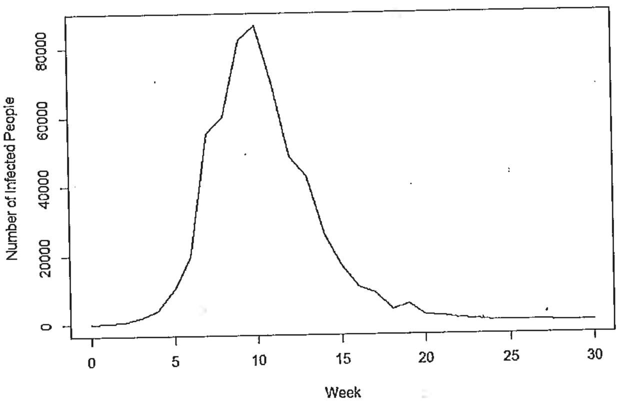
Q2 $\frac{24}{34}$

Question 2:

2.a:

```
> attach(epidemiology.data)
> plot(Week, No.infected, main="Number of Infected People Over a 30 
Week Period", xlab="Week", ylab="Number of Infected People")
```

Number of Infected People Over a 30 Week Period



✓

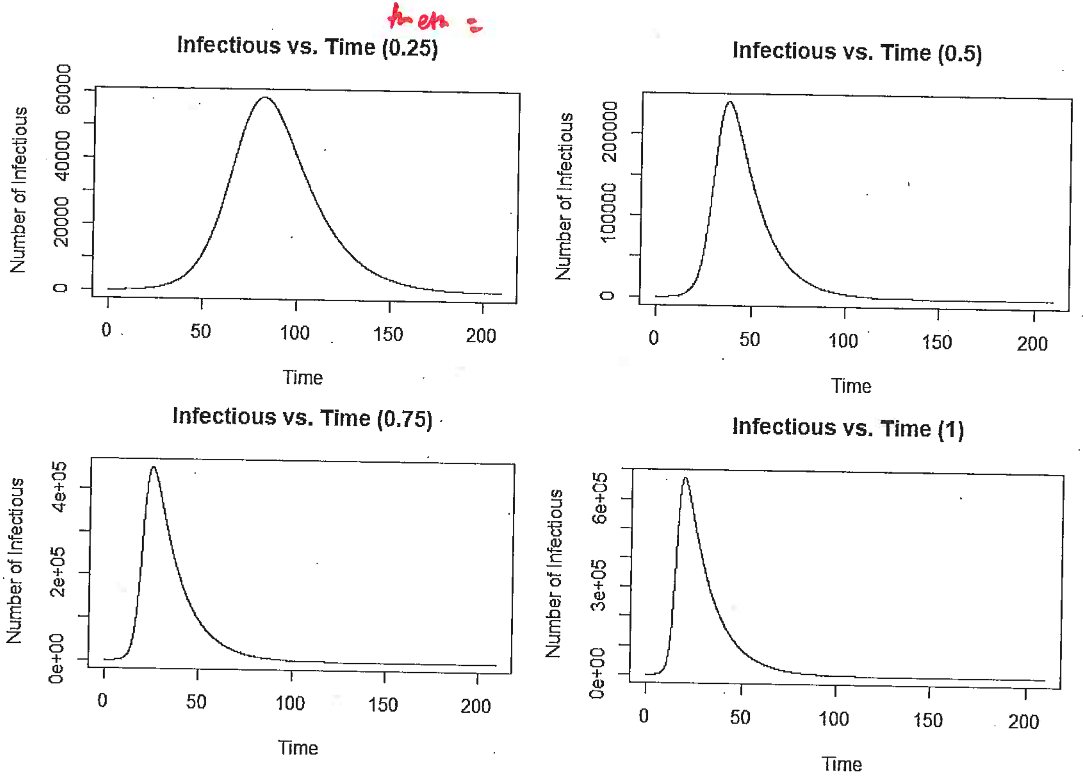
5.

Low numbers of people infected for first 3 weeks followed by gradually larger numbers each week until reaching week 10, at which ✓ the highest number of infected people is reached. This shows that the disease takes roughly around 7 to 14 weeks for most people to become infected with week 10 being the most likely. Followed by decline to very low numbers.

2.b:

```
> inf.time<-function(epidemiology.data,N.time.steps=2100, I0=61, R0=10, N=
1000000, theta, delta.t=0.1, gamma=1/14){
+       S0<-N*theta
+       beta<-R0*gamma/N
+       S<-numeric(N.time.steps+1)
+       I<-numeric(N.time.steps+1)
+       S[1]<-S0
+       I[1]<-I0
+       for(i in 1:N.time.steps){
+           S[i+1]<-S[i]-beta*S[i]*I[i]*delta.t
+           I[i+1]<-I[i]+beta*S[i]*I[i]*delta.t-gamma*I[i]*delta.t
+       }
+       time.vector<-seq(0,N.time.steps*delta.t,by=delta.t)
+       inf.time.list<-list(time.vector,I)
+       return(inf.time.list)
+ }

> t1<-inf.time(epidemiology.data,theta=0.25)
> t2<-inf.time(epidemiology.data,theta=0.5)
> t3<-inf.time(epidemiology.data,theta=0.75)
> t4<-inf.time(epidemiology.data,theta=1)
> plot(unlist(t1[1]),unlist(t1[2]),type="l")
> plot(unlist(t2[1]),unlist(t2[2]),type="l")
> plot(unlist(t3[1]),unlist(t3[2]),type="l")
> plot(unlist(t4[1]),unlist(t4[2]),type="l")
```

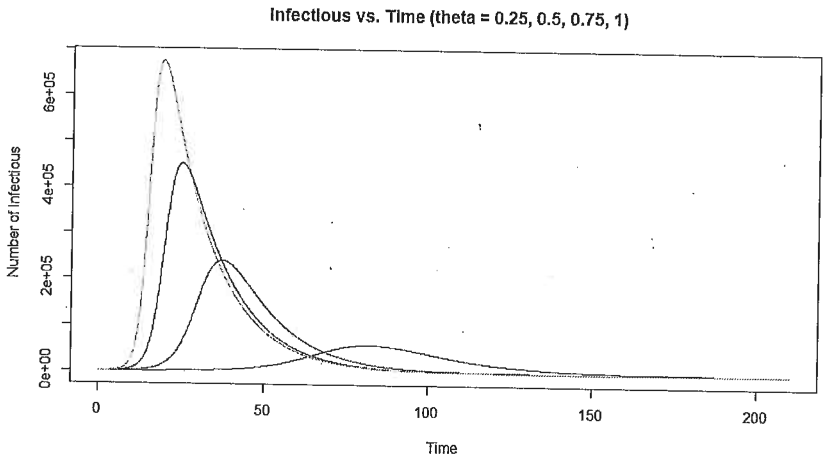
theta =



4.

These four graphs show that the spikes range between significantly larger values with each increase of the Theta value. They all follow a similar pattern, but the pattern seems thinner and shaper with higher theta values. With value ranges on each graph being different, it's best to plant all plots on one graph for a better comparison.

```
> plot(unlist(t1[1]),unlist(t1[2]),type="l",xlab="Time",ylab="Number of In
fectious",main="Infectious vs. Time (theta = 0.25, 0.5, 0.75, 1)",ylim=c(0
,max(unlist(t4[2]))))
> lines(unlist(t2[1]),unlist(t2[2]),col="red")
> lines(unlist(t3[1]),unlist(t3[2]),col="blue")
> lines(unlist(t4[1]),unlist(t4[2]),col="green")
```



Black: theta = 0.25

Blue: theta = 0.5

Red: theta = 0.75

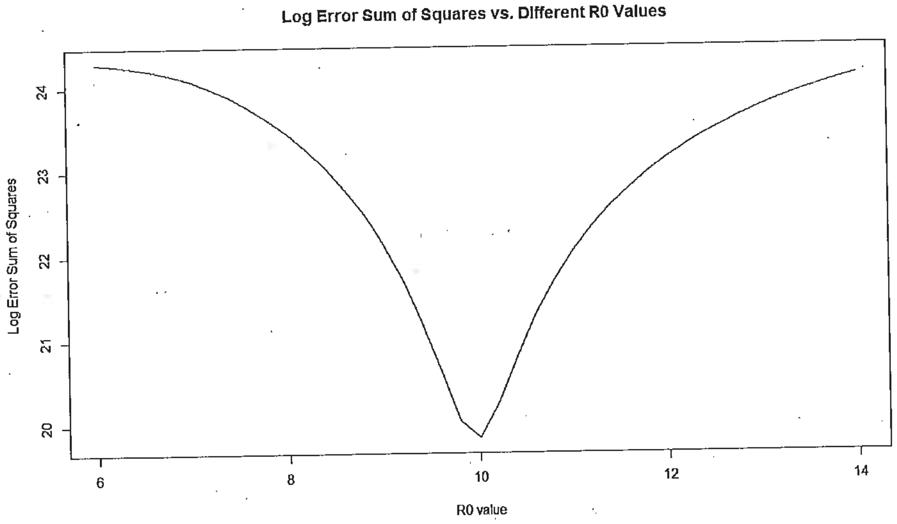
Green: theta = 1

✓

2.c:

```
> errorSS<-function(epidemiology.data, N=1000000, R0, D, theta, N.time.ste
ps=2100, delta.t=0.1){
+       gamma<-1/D
+       beta<-R0*gamma/N
+       S0<-theta*N
+       I0<-epidemiology.data$No.infected[1]
+       sample.index<-epidemiology.data$Time.index
+       S<-numeric(N.time.steps+1)
+       I<-numeric(N.time.steps+1)
+       S[1]<-S0
+       I[1]<-I0
+       for(i in 1:N.time.steps){
+           S[i+1]<-S[i]-beta*S[i]*I[i]*delta.t
+           I[i+1]<-I[i]+beta*S[i]*I[i]*delta.t-gamma*I[i]*delta.t
+       }
+       err.ss<-sum((epidemiology.data$No.infected-I[sample.index])^2)
+       return(err.ss)
+ }

> V<-seq(6,14,0.2)
> le.vals<-numeric(length(V))

> for(i in 1:length(V)){
+       le.vals[i]<-log(errorSS(epidemiology.data,R0=V[i],D=14,theta=0.3))
+ }

> plot(V,le.vals, main="Log Error Sum of Squares vs. Different R0 Values", xlab="R0 va
uares", type="l")
```

S.

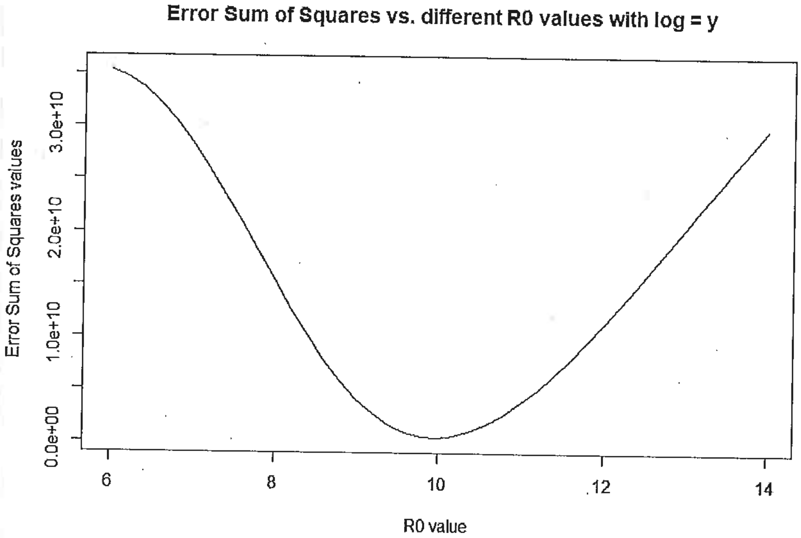


✓

The ✓graph above shows gradual drop slowly increase into a sharper drop before stopping at R0 value 10 and doing the opposite, rising back up in the same way. To see how logarithms are helping this dataset I've created another graph without the log function included to see how the dataset would fair without it. This graph is shown below.

log is minimised in same place as original data.

```
> for(i in 1:length(V)){
+       le.vals[i]<- errorSS(epidemiology.data,R0=V[i],D=14,theta=0.3)
+ }
```



Y axis starts at 0 and the values on the Y-axis are extremely small and can be difficult to read. However, with the logarithm included, it solves this and makes the Y axis easier to read. The overall shape of the line graph is different, as this graph shows a smoother and wider curve whereas the logarithm graph showed a gradually increasing drop until R0 10.

2.d:

```
> R<-seq(6,14,0.2)
> D<-seq(10,18,0.2)

> log.rd<-numeric(length(R))
> for(i in 1:length(R)){
+      log.rd[i]<-log(errorSS(epidemiology.data, R0=R[i],D=D[i], theta=0.3)
)
+ }
```

Only looking along the diagonal

Minimum loge:

```
> min(log.rd)
[1] 19.7924
```

✗ 19.76.

Corresponding R0 value:

```
> R[which.min(log.rd)]
[1] 9.8
```

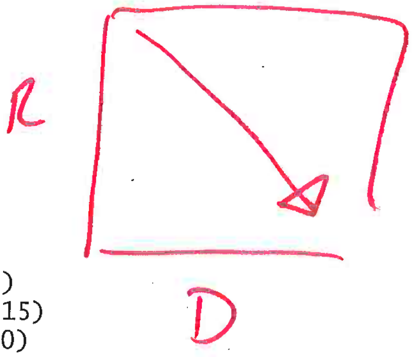
✓

2.

Corresponding D value:

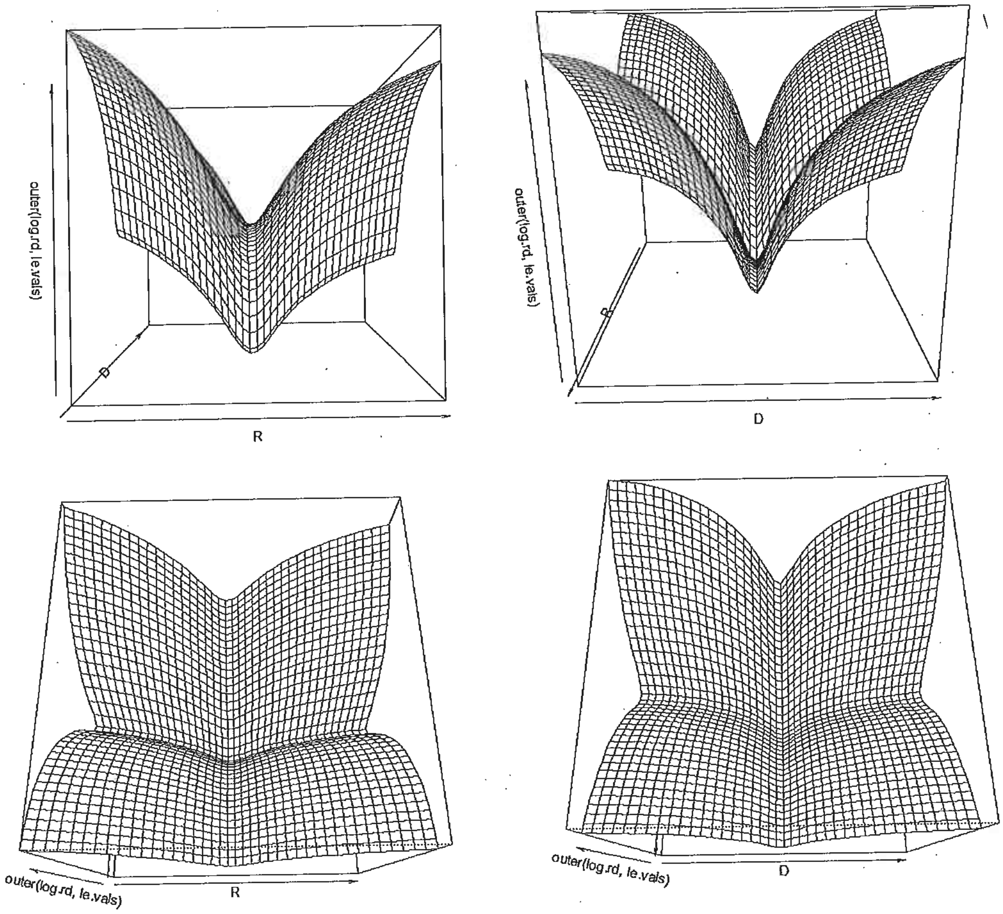
```
> D[which.min(log.rd)]
[1] 13.8
```

✗. 13.6



2.e:

```
> persp(R, D, outer(log.rd,le.vals), theta=0, phi=0)
> persp(R, D, outer(log.rd,le.vals), theta=90, phi=15)
> persp(R, D, outer(log.rd,le.vals), theta=0, phi=70)
> persp(R, D, outer(log.rd,le.vals), theta=90, phi=70)
```



plots not correct.

5.

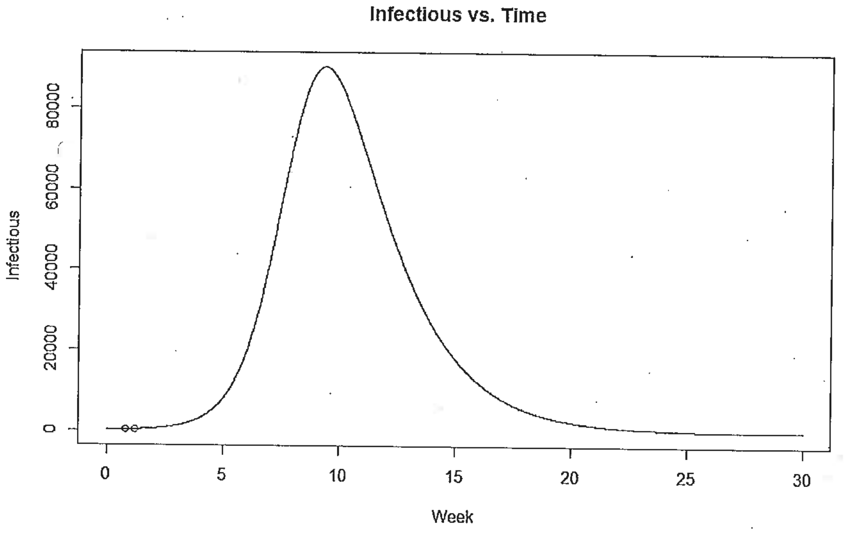
The matrix used in the persp() function contains the loge (le.vals) values from part 2.c and the log.rd values from 2.d. The first graph with theta 0 and phi 0 shows a close resemblance to the plot from 2.d. ✓ R value appears to have greater impact on the loge values with a higher starting point, lowering to a low midpoint and raising up again thereafter. The D value side shows a similar pattern but with lower values. ✓.

2.f:

```
> t03<-inf.time(epidemiology.data,theta=0.3)
> plot((unlist(t03[1])/7),unlist(t03[2]),type="l",main="Infectious vs. Time", xlab=
"Week", ylab="Infectious")

> points((unlist(t03[1])[R[which.min(log.rd)]]),(unlist(t03[2])[R[which.min(log.rd)
]]))

> points((unlist(t03[1])[D[which.min(log.rd)]]),(unlist(t03[2])[D[which.min(log.rd)
]]),col="red")
```

+ data points ?

3.



```
> unlist(t03[1])[R[which.min(log.rd)]]
[1] 0.8

> unlist(t03[2])[R[which.min(log.rd)]]
[1] 68.32996


> unlist(t03[1])[D[which.min(log.rd)]]
[1] 1.2

> unlist(t03[2])[D[which.min(log.rd)]]
[1] 72.31873
```

If I've plot correctly here, the minimum R0 and D values are just before the infectious number begins to climb for the peak. With the theta = 0.3 model the peak is at it's highest at week 10 at a value of around 90,000.

3.a:

Using equation:

Q3 $\frac{9}{22}$

$$R_0 = \frac{\beta N}{\gamma}$$

Rearranging to:

$$\beta = \frac{R_0 * \gamma}{N}$$

```
> B=(20*0.1)/N
> B
[1] 4e-07
```

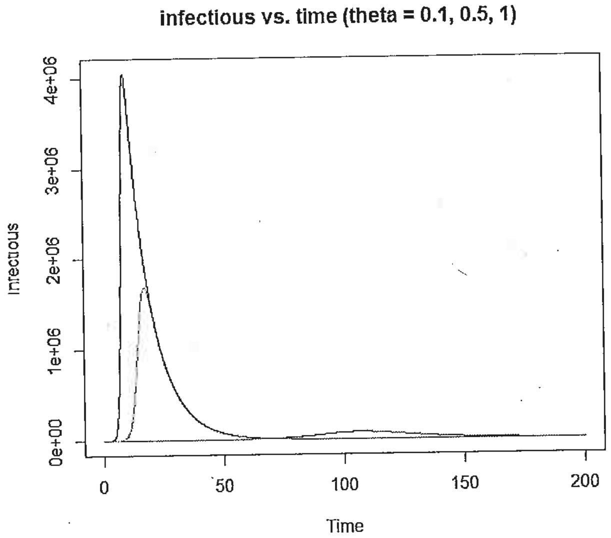
✓

2.

Infection rate = 0.0000004

3.b:

```
> q3.inf.time<- function(epidemiology.data, N.time.steps=2000,I0,R0=20, N=
5000000, theta, delta.t=0.1, gamma=0.1){
+ beta<-R0*gamma/N
+ S<-numeric(N.time.steps+1)
+ I<-numeric(N.time.steps+1)
+ S0<-N*theta
+ S[1]<-S0
+ I[1]<-I0
+ for(i in 1:N.time.steps){
+     S[i+1]<-S[i]-beta*S[i]*I[i]*delta.t
+     I[i+1]<-I[i]+beta*S[i]*I[i]*delta.t-gamma*I[i]*delta.t
+     }
+ t.vector<-seq(0,N.time.steps*delta.t,by=delta.t)
+ i.t.list<-list(t.vector,I)
+ return(i.t.list)
+ }
>

> theta1<-q3.inf.time(epidemiology.data,I0=10,theta=0.1)
> theta2<-q3.inf.time(epidemiology.data,I0=10,theta=0.5)
> theta3<-q3.inf.time(epidemiology.data,I0=10,theta=1)


> plot(unlist(theta1[1]), unlist(theta1[2]), type="l", col="red", xlim=c(0
,max(unlist(theta1[1]))), ylim=c(0,max(unlist(theta3[2]))), main="infectio
us vs. time (theta = 0.1, 0.5, 1)", xlab="Time", ylab="Infectious")
> lines(unlist(theta2[1]), unlist(theta2[2]), type="l", col="green")
> lines(unlist(theta3[1]), unlist(theta3[2]), type="l", col="blue")
```



Red: theta = 0.1

Green: theta = 0.5

Blue: theta = 1

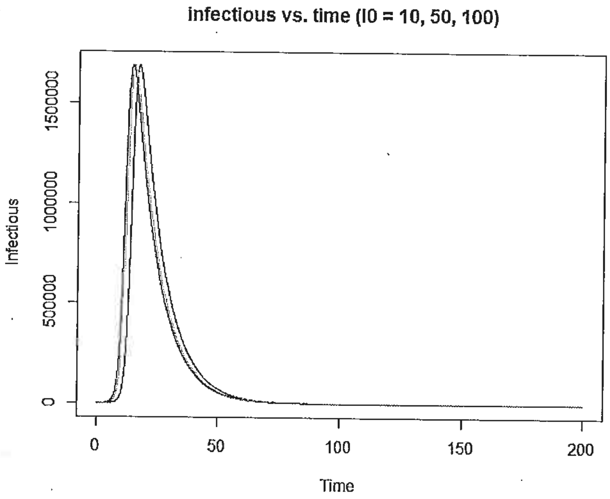
✓

```
> plot(unlist(I1[1]), unlist(I1[2]), type="l", col="red", xlim=c(0,max(unl
ist(I1[1]))), ylim=c(0,max(unlist(I3[2]))), main="infectious vs. time (I0
= 10, 50, 100)", xlab="Time", ylab="Infectious")
> lines(unlist(I2[1]), unlist(I2[2]), type="l", col="green")
> lines(unlist(I3[1]), unlist(I3[2]), type="l", col="blue")

> I1<-q3.inf.time(epidemiology.data,I0=10,theta=0.5)
> I2<-q3.inf.time(epidemiology.data,I0=50,theta=0.5)
> I3<-q3.inf.time(epidemiology.data,I0=100,theta=0.5)
```

Comments on effects of varying parameters.

Is the effect of varying θ the same for different I_0 values? ✓

7



Red: theta = 10
Green: theta = 50
Blue: theta = 100

Decided to go with theta values 0.1, 0.5, 1 as the 0.1 will show the effect of a small theta value, 1 will show the effect of a high theta value and the 0.5 is a steady mid-point. Following this 10%, 50%, 100% pattern for choice of values, the I0 values chosen were 10, 50, 100.

Q4 $\frac{31}{38}$

4.a:

B = birth rate, rate of new individuals entering susceptible pool

βSI = influx from susceptible pool to infectious pool

µS = death rate for susceptible pool individuals ✓

3.

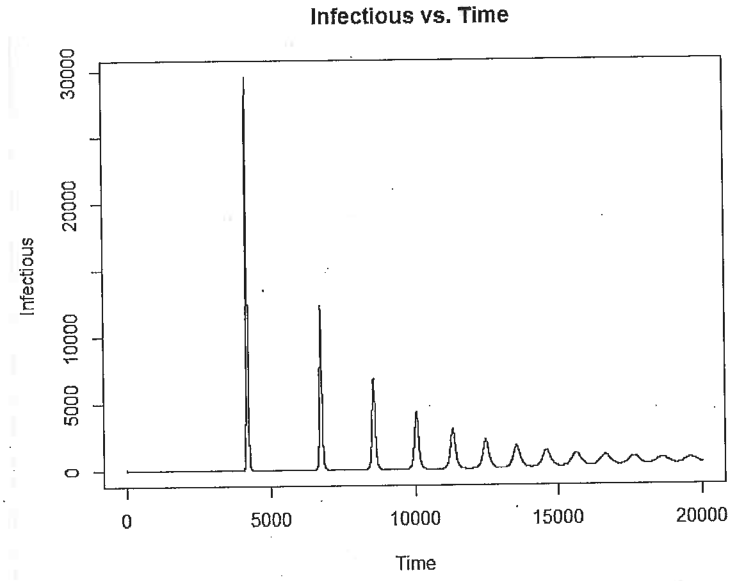
4.b:

```
> generate.S.I.by.time.vital.dynamics<-
+     function(N.time.steps,delta.t=0.1,S0,I0,R0,gamma,mu,N){
+         S<-numeric(N.time.steps+1)
+         I<-numeric(N.time.steps+1)
+         S[1]<-S0
+         I[1]<-I0
+         beta<-R0*gamma/N
+         for (i in 1:N.time.steps){
+             S[i+1]<-S[i]+mu*N*delta.t-beta*S[i]*I[i]*delta.t-mu*S[i]*delta.t
+             I[i+1]<-I[i]+beta*S[i]*I[i]*delta.t-gamma*I[i]*delta.t-mu*I[i]*delta.t
+         }
+         time.vector<-seq(0,N.time.steps*delta.t,by=delta.t)
+         out<-list(S=S,I=I,time.vector=time.vector)
+         return(out)
+     }
```

```
> q4.data<-generate.S.I.by.time.vital.dynamics(N.time.steps=200000, S0=0.0
2*800000, I0=50, R0=6, gamma=0.1, mu=65.75/800000, N=800000)
```

q4.data[1] = susceptible
q4.data[2] = infected
q4.data[3] = time

```
> plot(unlist(q4.data[3]), unlist(q4.data[2]), main="Infectious vs. Time",
xlab="Time", ylab="Infectious", type="l")
```



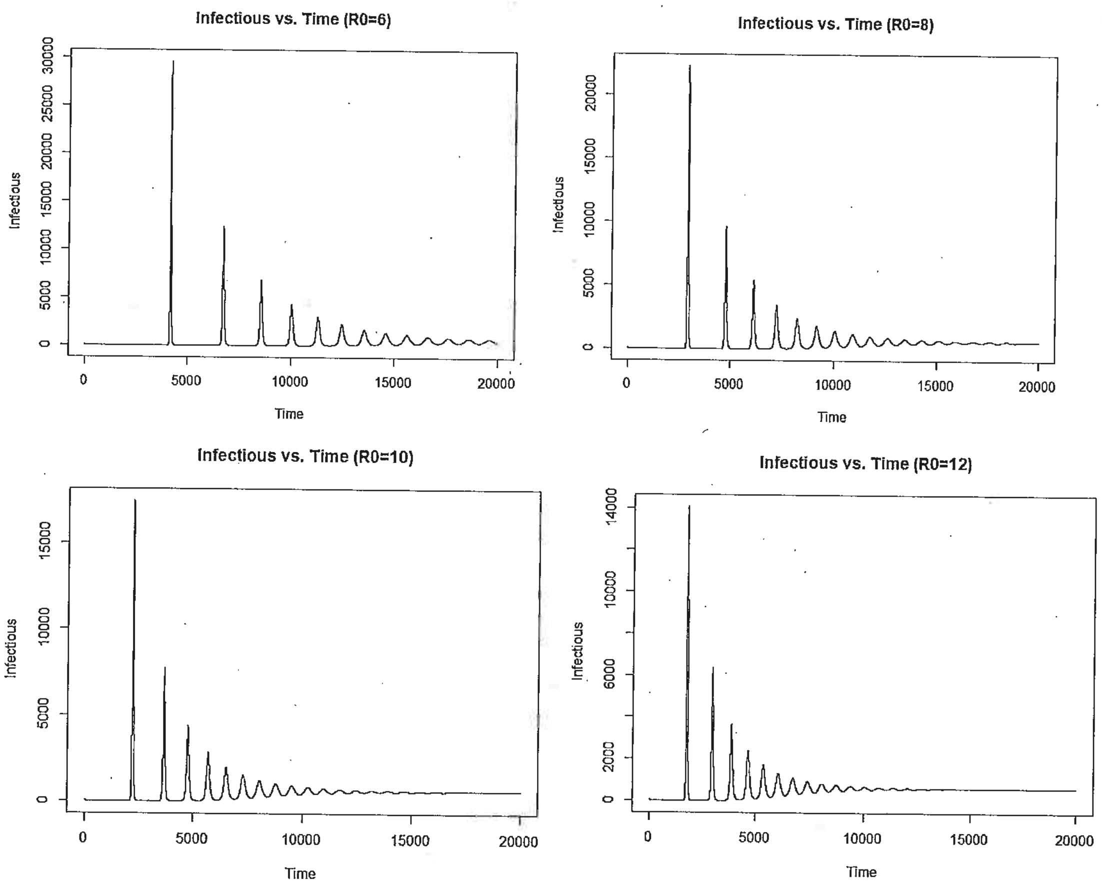
5.

Large initial spike at around 4300 time with an interval of around 1500 length in time. The spikes get gradually shorter and wider and the intervals between each spike also appear to get smaller. This plot shows a gradual decrease in number of infections with each outbreak over time, with the number of infectious pool individuals more than halving from the first outbreak to the second outbreak. From there on, the amount each outbreaks' infectious pool count decreases gets smaller over time. For example, as mentioned previously, the second outbreak has less than half that of the first outbreak infections whereas the difference between the final two recorded outbreaks is almost unnoticeable.

4.c:

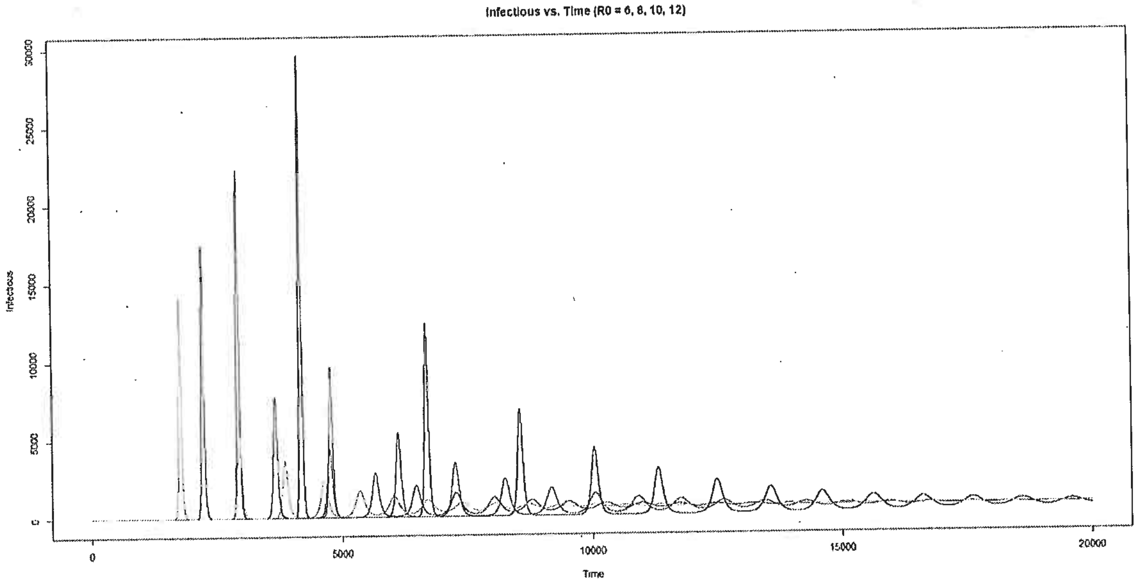
```
> q4.data.r6<-generate.S.I.by.time.vital.dynamics(N.time.steps=200000, S0=c(0.02*800000), I0=
50, R0=6, gamma=0.1, mu=c(65.75/800000), N=800000)

> q4.data.r8<-generate.S.I.by.time.vital.dynamics(N.time.steps=200000, S0=c(0.02*800000), I0=
50, R0=8, gamma=0.1, mu=c(65.75/800000), N=800000)

> q4.data.r10<-generate.S.I.by.time.vital.dynamics(N.time.steps=200000, S0=c(0.02*800000), I0
=50, R0=10, gamma=0.1, mu=c(65.75/800000), N=800000)

> q4.data.r12<-generate.S.I.by.time.vital.dynamics(N.time.steps=200000, S0=c(0.02*800000), I0
=50, R0=12, gamma=0.1, mu=c(65.75/800000), N=800000)


> plot(unlist(q4.data.r6[3]), unlist(q4.data.r6[2]), main="Infectious vs. Time (R0=6)", xlab=
"Time", ylab="Infectious", type="l", xlim=c(0,max(unlist(q4.data.r6[3]))))

> plot(unlist(q4.data.r8[3]), unlist(q4.data.r8[2]), main="Infectious vs. Time (R0=8)", xlab=
"Time", ylab="Infectious", type="l", xlim=c(0,max(unlist(q4.data.r8[3]))))

> plot(unlist(q4.data.r10[3]), unlist(q4.data.r10[2]), main="Infectious vs. Time (R0=10)", xl
ab="Time", ylab="Infectious", type="l", xlim=c(0,max(unlist(q4.data.r10[3]))))
```

```
> plot(unlist(q4.data.r12[3]), unlist(q4.data.r12[2]), main="Infectious vs. Time (R0=12)", xl
ab="Time", ylab="Infectious", type="l", xlim=c(0,max(unlist(q4.data.r12[3]))))
```



✓

```
> plot(unlist(q4.data.r6[3]), unlist(q4.data.r6[2]), main="Infectious vs. Time (R0 = 6, 8, 10
, 12)", xlab="Time", ylab="Infectious", type="l", xlim=c(0,max(unlist(q4.data.r6[3]))))
> lines(unlist(q4.data.r8[3]), unlist(q4.data.r8[2]),col="red")
> lines(unlist(q4.data.r10[3]), unlist(q4.data.r10[2]),col="blue")
> lines(unlist(q4.data.r12[3]), unlist(q4.data.r12[2]),col="green")
```

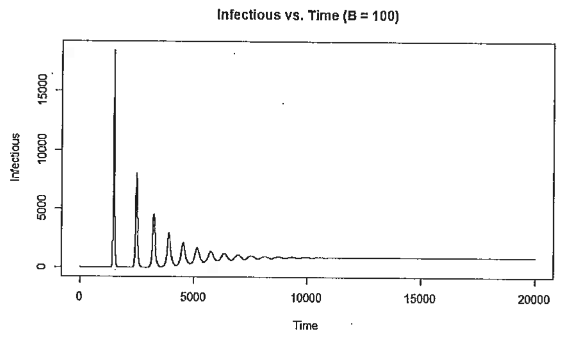


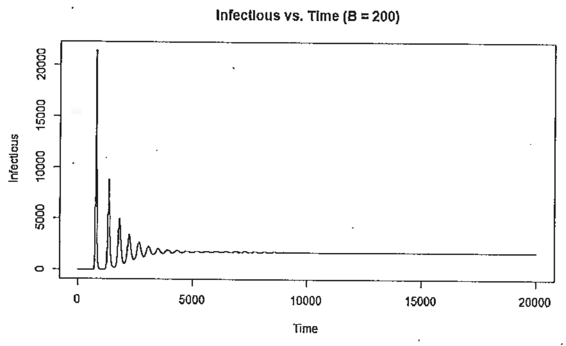
6

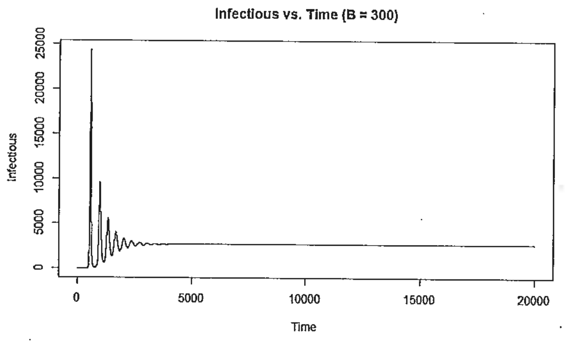
From the results shown on the graph above, it appears that the higher the R0 value, the earlier the initial infection begins but at a lower total of individuals in the infectious pool. The larger R0 values also seem to show a quicker/proportionally larger decrease in number of individuals in the infectious pool for each following outbreak. None of the R0 values show the number of individuals return to zero after so many outbreaks and the number of individuals in the infectious pool appears to stabilize at around 500 to 1000 individuals. Larger R0 values appear to have a higher frequency of outbreaks but the number of individuals in the infectious pool also seems to stabilize faster in terms of time. ✓

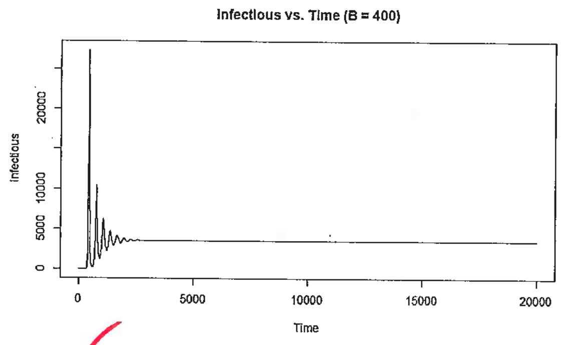
4.d:

```
> q4.data.b100<-generate.S.I.by.time.vital.dynamics(N.time.steps=200000, S0=c(0.02*800000), I
0=50, R0=10, gamma=0.1, mu=c(100/800000), N=800000)

> q4.data.b200<-generate.S.I.by.time.vital.dynamics(N.time.steps=200000, S0=c(0.02*800000), I
0=50, R0=10, gamma=0.1, mu=c(200/800000), N=800000)

> q4.data.b300<-generate.S.I.by.time.vital.dynamics(N.time.steps=200000, S0=c(0.02*800000), I
0=50, R0=10, gamma=0.1, mu=c(300/800000), N=800000)

> q4.data.b400<-generate.S.I.by.time.vital.dynamics(N.time.steps=200000, S0=c(0.02*800000), I
0=50, R0=10, gamma=0.1, mu=c(400/800000), N=800000)

> plot(unlist(q4.data.b100[3]), unlist(q4.data.b100[2]), main="Infectious vs. Time (B = 100)"
, xlab="Time", ylab="Infectious", type="l", xlim=c(0,max(unlist(q4.data.b100[3]))))

> plot(unlist(q4.data.b200[3]), unlist(q4.data.b200[2]), main="Infectious vs. Time (B = 200)"
, xlab="Time", ylab="Infectious", type="l", xlim=c(0,max(unlist(q4.data.b200[3]))))

> plot(unlist(q4.data.b300[3]), unlist(q4.data.b300[2]), main="Infectious vs. Time (B = 300)"
, xlab="Time", ylab="Infectious", type="l", xlim=c(0,max(unlist(q4.data.b300[3]))))

> plot(unlist(q4.data.b400[3]), unlist(q4.data.b400[2]), main="Infectious vs. Time (B = 400)"
, xlab="Time", ylab="Infectious", type="l", xlim=c(0,max(unlist(q4.data.b400[3]))))
```









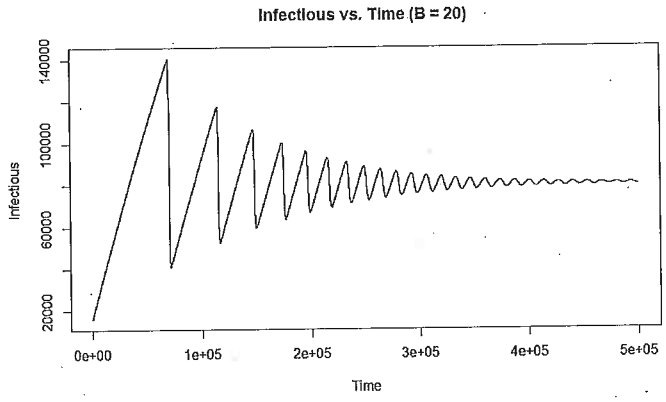
4

Larger B values appear to have a higher frequency of outbreaks, but the outbreaks also die out quicker with the graph of B = 400 having no further outbreaks past time 2600. It also appears there's a trend where the higher the B value, the more individuals are in the infectious group for the initial outbreak. The higher B values also show that the number of infectious pool individuals stabilizes at a higher number. For example, graph (B=100) shows a stabilization of individuals in the pool at around 1000 whereas the number at which it stabilizes for graph (B=400) is around 4000 individuals, although there are more individuals overall to take into consideration.

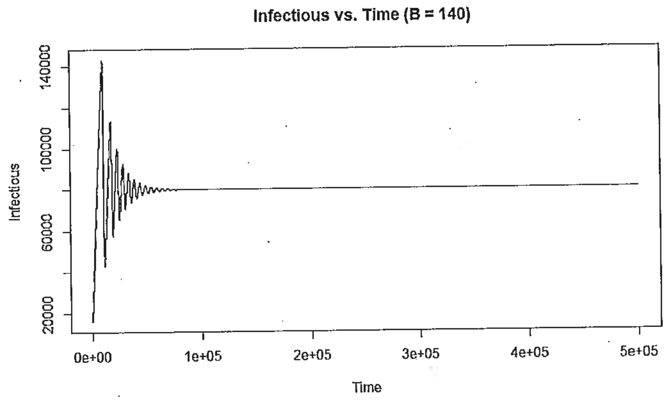
4.e:

```
> value.seq<-seq(20,140,10)
> returned.values<-numeric(length(value.seq))
> for (i in value.seq){
+     index<-((i-10)/10)
+     returned.values[index]<-generate.S.I.by.time.vital.dynamics(N.time.steps=500000, S0=c(0
.02*800000), I0=50, R0=10, gamma=0.1, mu=c(i/800000), N=800000)
+     }
```

```
> plot(unlist(returned.values[1]),type="l", ylab="Infectious", xlab="Time", main="Infectious
vs. Time (B = 20)")

> plot(unlist(returned.values[13]),type="l", ylab="Infectious", xlab="Time", main="Infectious
vs. Time (B = 140)")
```



X strange-looking plot



```
> infectious.info = matrix(nrow=14,ncol=10)
> infectious.info[nrow=1,ncol=1]<-"B_Value"
> infectious.info[nrow=1,ncol=2]<-"Equil_Avg"
> infectious.info[nrow=1,ncol=3]<-"1st_peak_value"
> infectious.info[nrow=1,ncol=4]<-"2nd_peak_value"
> infectious.info[nrow=1,ncol=5]<-"3rd_peak_value"
> infectious.info[nrow=1,ncol=6]<-"4rd_peak_value"
> infectious.info[nrow=1,ncol=7]<-"1st_peak_index"
> infectious.info[nrow=1,ncol=8]<-"2nd_peak_index"
> infectious.info[nrow=1,ncol=9]<-"3rd_peak_index"
> infectious.info[nrow=1,ncol=10]<-"4th_peak_index"
```

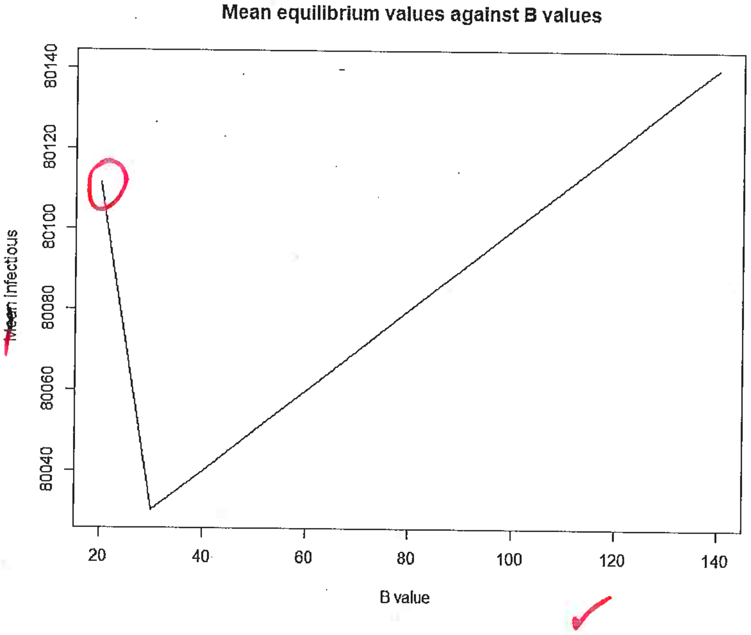
```
> for(set in 1:13){
+     setseq<-seq(20,140,10)
+     infectious.info[(set+1), 1]<-setseq[set]
+
+     firstfour<-0
+     infectious.info[(set+1),2]<-mean(unlist(returned.values[set])[490000:500001])
+     peak.vals<-numeric(4)
+     peak.ind<-numeric(4)
+
+     for(index in 1:length(unlist(returned.values[set]))){
+         if(index>1 & diff(unlist(returned.values[set]))[index]<0){
+             if(diff(unlist(returned.values[set]))[(index-1)]>0){
+                 firstfour<-(firstfour+1)
+                 infectious.info[(set+1), (2+firstfour)]<-unlist(returned.values[set])[(index-1)]
+                 infectious.info[(set+1), (6+firstfour)]<-index
+                 if(firstfour==4){
+                     break
+                 }
+             }
+         }
+     }
+ }
```

```
> infectious.info
       [,1]      [,2]              [,3]                [,4]                [,5]
 [1,] "B_Value" "Equil_Avg"       "1st_peak_value"    "2nd_peak_value"    "3rd_peak_value"
 [2,] "20"      "80111.3880150357" "140541.810289602" "117327.505494995" "106445.151131593"
 [3,] "30"      "80030.3179967512" "140654.457516738" "116830.283439621" "105675.16818715"
 [4,] "40"      "80039.9822755424" "140803.688091002" "116385.439431065" "104974.913609216"
 [5,] "50"      "80049.9960270642" "140979.696430372" "115978.483401318" "104325.945550394"
 [6,] "60"      "80060.0003215758" "141176.700265992" "115600.957805854" "103717.948713264"
 [7,] "70"      "80070.0000017541" "141390.738250386" "115247.346430984" "103144.542294575"
 [8,] "80"      "80080.0000007251" "141619.020053438" "114913.802833705" "102601.525620977"
 [9,] "90"      "80090.0000000173" "141859.480908372" "114597.44458341"  "102085.865614458"
[10,] "100"     "80099.9999999999" "142110.384053703" "114296.142823771" "101595.218718729"
[11,] "110"     "80110.0000000005" "142370.481051497" "114008.204454849" "101127.694818087"
[12,] "120"     "80119.9999999987" "142638.588213328" "113732.273371852" "100681.62200775"
[13,] "130"     "80130.0000000005" "142913.880244857" "113467.23936559"  "100255.546604696"
[14,] "140"     "80139.9999999991" "143195.52681327"  "113212.268313692" "99848.1971401297"
```

```
      [,6]               [,7]              [,8]              [,9]              [,10]
 [1,] "4rd_peak_value"   "1st_peak_index" "2nd_peak_index" "3rd_peak_index" "4th_peak_index"
 [2,] "100042.467196972" "69336"          "114015"         "146976"         "173334"
 [3,] "99101.8678887997" "46314"          "76422"          "98722"          "116697"
 [4,] "98250.0182256522" "34814"          "57640"          "74623"          "88440"
 [5,] "97466.5746663093" "27920"          "46380"          "60183"          "71536"
 [6,] "96740.8354687006" "23329"          "38879"          "50571"          "60306"
 [7,] "96066.0245339438" "20052"          "33525"          "43717"          "52315"
 [8,] "95437.0659242307" "17596"          "29513"          "38588"          "46344"
 [9,] "94849.740675945"  "15688"          "26394"          "34608"          "41715"
[10,] "94300.3905266672" "14162"          "23901"          "31431"          "38023"
[11,] "93785.7452906531" "12915"          "21863"          "28838"          "35010"
[12,] "93302.8561582746" "11876"          "20166"          "26681"          "32504"
[13,] "92849.0855760901" "10998"          "18731"          "24859"          "30387"
[14,] "92422.0519190671" "10246"          "17503"          "23301"          "28575"
```

```
> B<-infectious.info[2:14,1]
> equil<-infectious.info[2:14,2]
> peak.val<-infectious.info[2:14,3:6]
> peak.time<-infectious.info[2:14,7:10]
```

```
> plot(B, equil, type="l", main="Mean equilibrium values against B values"
, xlab="B value", ylab="Mean Infectious")
```



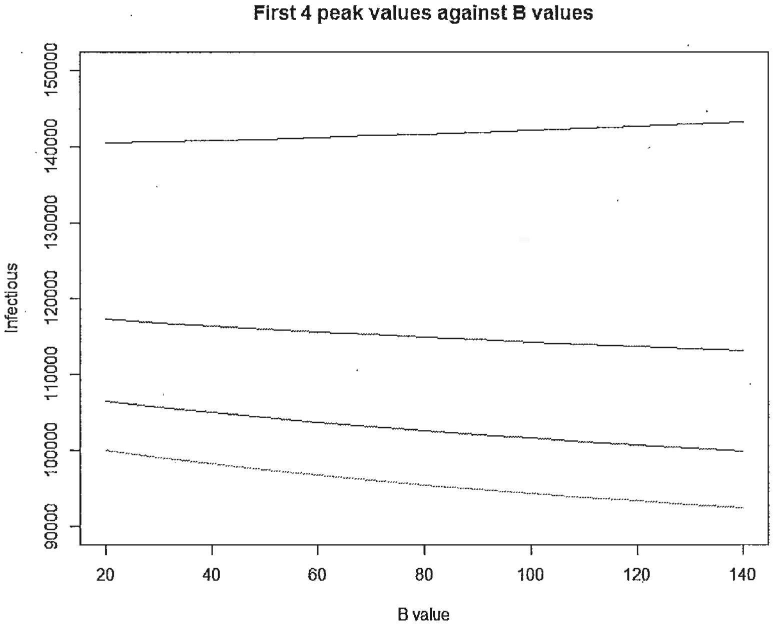
Explain

Should be linear, but y-values too high.

✓

This graph shows the mean equilibrium value of infectious of the final 1000 values for the set of data for each variation of B. The mean value drops significantly from B = 20 to B = 30, the value drops by 80. From B = 30 onwards, there is a gradual increase and a positive correlation between increasing mean and increasing B value. The broad pattern shows that the higher the B value, the higher the equilibrium infectious value.

```
> plot(B, peak.val[1:13,1], type="l", main="First 4 peak values against B
values", xlab="B value", ylab="Infectious", ylim=c(90000,150000))
> lines(B,peak.val[1:13,2],col="red")
> lines(B,peak.val[1:13,3],col="blue")
> lines(B,peak.val[1:13,4],col="green")
```



Black: First peak

Red: Second peak

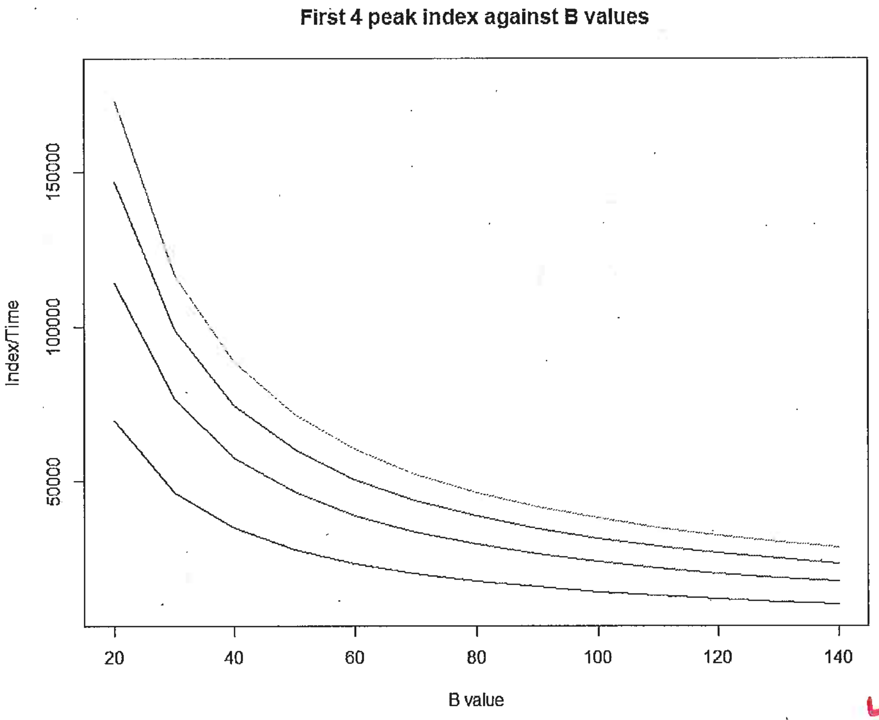
Blue: Third peak ✓

Green: Fourth peak

This line graph shows values of each of the four peaks with each B value tested. This shows that the initial infectious peak is higher in value with larger B values. However, with the three following infectious peaks, they appear to be lower in value with the increasing B values. The difference in values of peak 2 to 3 and 3 to 4 may also decrease by a larger amount with larger B values although it is difficult to tell if this is such a pattern on this graph as this difference seems small and almost not visible. But it looks as if this is occurring as the gap between the red line and the blue line looks as if it gets larger towards the higher B values, though only slightly.

```
> plot(B, peak.time[1:13,1], type="l", main="First 4 peak index against B
values", xlab="B value", ylab="Index/Time", ylim=c(10000,180000))
```

```
> lines(B,peak.time[1:13,2],col="red")
> lines(B,peak.time[1:13,3],col="blue")
> lines(B,peak.time[1:13,4],col="green")
```



✓

Black: First peak

Red: Second peak

Blue: Third peak

Green: Fourth peak

✓

This line graph shows the time at which each of the four first infectious peaks occur by index. The time for the initial peak for B=20 is 170000 and as B increases this value decreases rapidly before decreasing at a slower pace towards the higher end of the B value vector. This pattern shows the same for each of the following three peaks just at lower values. The frequency of the peaks increases with higher B values, this can be seen by looking at the size of the difference between each peak line for each B value. The differences between the peaks at B=140 are much smaller in comparison to large differences at B=20.

13

Q5 4/16

5

```
>   generate.S.I.by.time.vital.dynamics.seasonal.forcing<-
+           function(N.time.steps,delta.t=0.1,S0,I0,R0,gamma,mu,N,alpha){
+                   S<-numeric(N.time.steps+1)
+                   I<-numeric(N.time.steps+1)
+                   S[1]<-S0
+                   I[1]<-I0
+                   beta<-(R0*gamma/N)*(1+alpha*cos(2*pi*seq(0,20,1)))
+                   for (i in 1:N.time.steps){
+                           S[i+1]<-S[i]+mu*N*delta.t-beta*S[i]*I[i]*delta.t-mu*S[i]*delta.t
+                           I[i+1]<-I[i]+beta*S[i]*I[i]*delta.t-gamma*I[i]*delta.t-mu*I[i]*delta.t
+                       }
+                   time.vector<-seq(0,N.time.steps*delta.t,by=delta.t)
+                   out<-list(S=S,I=I,time.vector=time.vector)
+                   return(out)
+           }
```

X

this is not time in years.

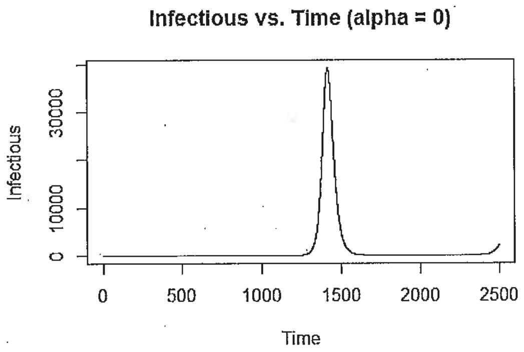
The main change to the function is the following line of code:
`beta<-(R0*gamma/N)*(1+alpha*cos(2*pi*seq(0,20,1)))`

Using sequence of 0 to 20 in intervals of 1 to replicate the time scale of 20 years as t is in years.

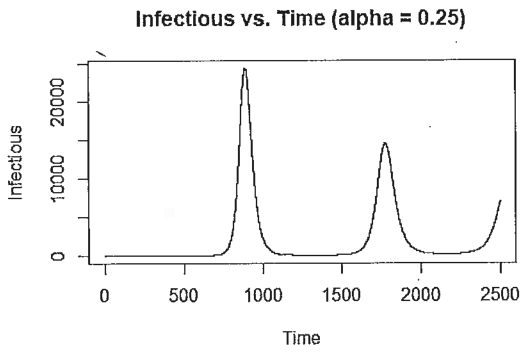
```
> q5.0<-generate.S.I.by.time.vital.dynamics.seasonal.forcing(N.time.steps=25000, S0=c(0.04*5
000000), I0=(500/30), R0=14, gamma=0.1, mu=c(100000/(365*5000000)), N=5000000, alpha=0)

> q5.25<-generate.S.I.by.time.vital.dynamics.seasonal.forcing(N.time.steps=25000, S0=c(0.04*
5000000), I0=(500/30), R0=14, gamma=0.1, mu=c(100000/(365*5000000)), N=5000000, alpha=0.25)

> q5.5<-generate.S.I.by.time.vital.dynamics.seasonal.forcing(N.time.steps=25000, S0=c(0.04*5
000000), I0=(500/30), R0=14, gamma=0.1, mu=c(100000/(365*5000000)), N=5000000, alpha=0.5)

> q5.75<-generate.S.I.by.time.vital.dynamics.seasonal.forcing(N.time.steps=25000, S0=c(0.04*
5000000), I0=(500/30), R0=14, gamma=0.1, mu=c(100000/(365*5000000)), N=5000000, alpha=0.75)
```

4

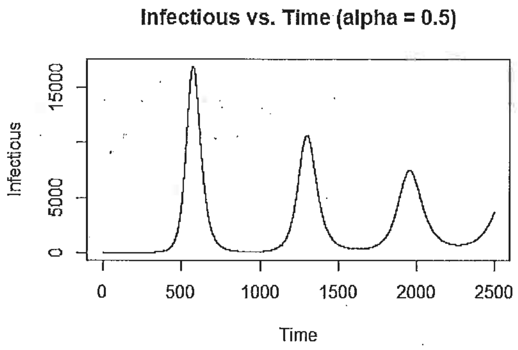
```
> plot(unlist(q5.0[3]), unlist(q5.0[2]), main="Infectious vs. Time (alpha = 0)", xlab="Time",
ylab="Infectious", type="l")

> plot(unlist(q5.25[3]), unlist(q5.25[2]), main="Infectious vs. Time (alpha = 0.25)", xlab="T
ime", ylab="Infectious", type="l")

> plot(unlist(q5.5[3]), unlist(q5.5[2]), main="Infectious vs. Time (alpha = 0.5)", xlab="Time
", ylab="Infectious", type="l")

> plot(unlist(q5.75[3]), unlist(q5.75[2]), main="Infectious vs. Time (alpha = 0.75)", xlab="T
ime", ylab="Infectious", type="l")
```

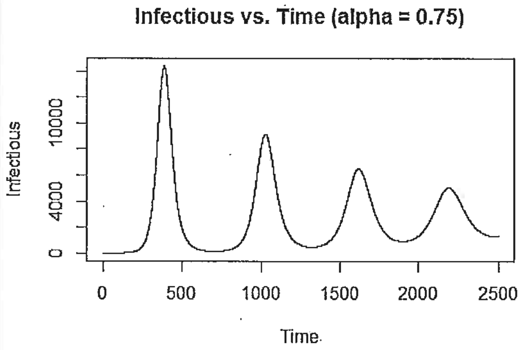


Infectious vs. Time (alpha = 0)



Infectious vs. Time (alpha = 0.25)





Graphs too small.