Session 1 - Exercise 1

Daniel Viana

January 30, 2018

Table of Contents

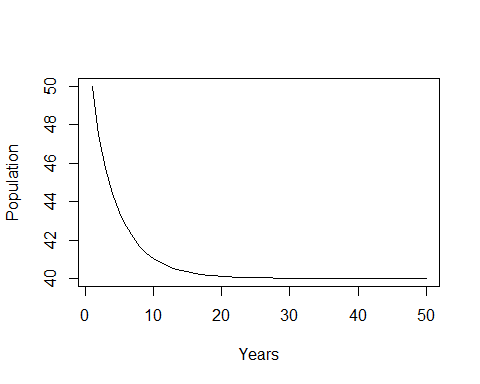
Exercise 1

# a

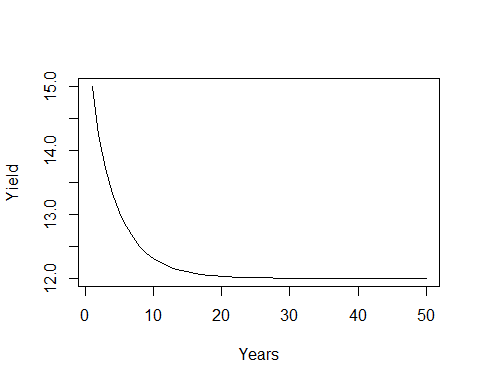
#Define variables   
g = 0.5  
f = 0.3  
K=100  
t=50  
  
#Create vectors to store population size and yield  
pop = rep(0, t)  
yield = rep(0, t)  
  
#Set initial population size  
pop[1] = 50  
yield[1]=50\*f  
  
#Run the loop  
for(n in 2:t){  
 pop[n] = pop[n-1] + pop[n-1]\*g\*(1-pop[n-1]/K) - pop[n-1]\*f   
 yield[n] = pop[n]\*f  
 }

## Plots

plot(pop, type="l", ylab="Population", xlab="Years")



plot(yield, type="l", ylab="Yield", xlab="Years")



## Questions

sum(yield)

## [1] 613.2043

#613.2043

pop[50]

## [1] 40.00013

yield[50]

## [1] 12.00004

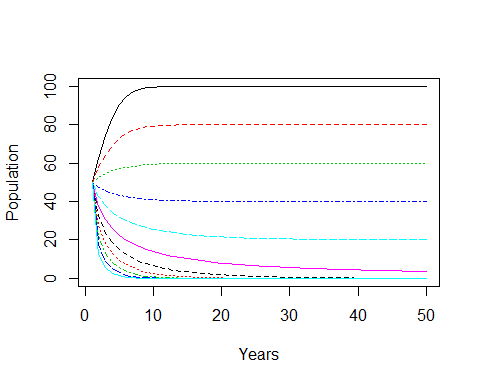
#Population size in year 50 is 40, and yield in year 50 is 12

# b

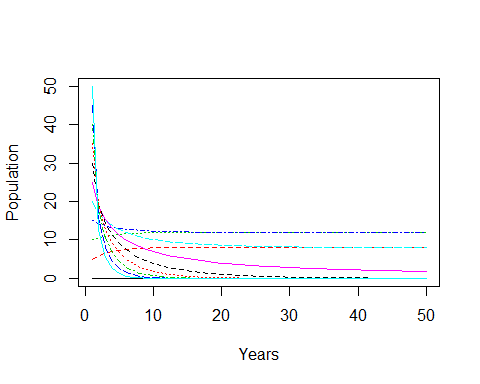
#Define variables   
g = 0.5  
f = seq(0, 1, by=0.1)  
K=100  
t=50  
  
#Create matrices to store population size and yield  
pop = matrix(0, t, length(f))  
yield = matrix(0, t, length(f))  
  
#Set initial population size and yield  
pop[1,] = 50  
yield[1,] = 50\*f  
  
#Run the loop  
for(i in 1:ncol(pop)){  
 for(n in 2:t){  
 pop[n, i] = pop[n-1, i] + pop[n-1, i]\*g\*(1-pop[n-1,i]/K) - pop[n-1, i]\*f[i]   
 yield[n, i] = pop[n, i]\*f[i]  
 }  
}

## Plots

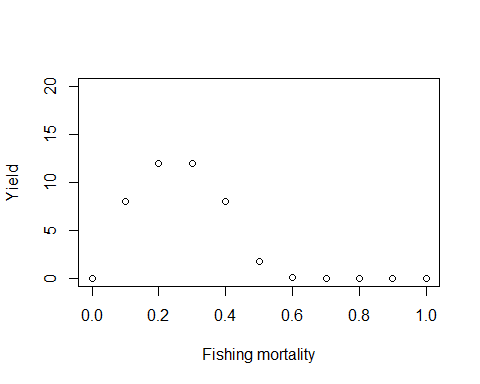
matplot(pop, type="l", ylab="Population", xlab="Years")



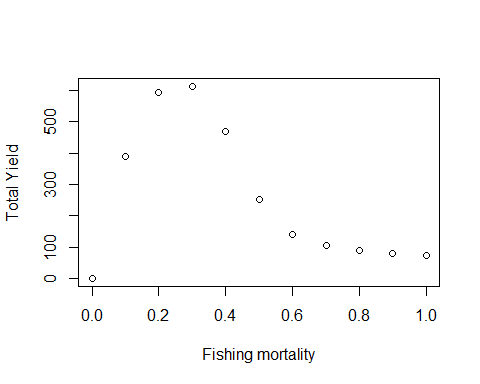
matplot(yield, type="l", ylab="Population", xlab="Years")



plot(yield[50,]~f, ylim=c(0, 20), xlim=c(0, 1), ylab="Yield", xlab="Fishing mortality")



#Create a vector to store the sum of yields for all fishing mortalities  
sum.yield = rep(0, ncol(yield))  
  
#Run a loop to calculate the yield sum for all values of f   
for(n in 1:length(sum.yield)){  
 sum.yield[n] = sum(yield[,n])  
}  
#Plot  
plot(sum.yield~f, ylab="Total Yield", xlab="Fishing mortality")



## Questions

x = cbind(sum.yield, f)  
#You can either look at the matrix to figure out the highest f, or you can use this function called "match" to find the f that gives you the highest total yield. This function tells you the row in which the highest yield is located and you can use that to find the f you are looking for. This can be useful when you have a large dataset and can't just look for the highest value. For this function you need to first put the value (or values) you are looking for, then the data to be matched against. For example:  
h=match(max(sum.yield), x)  
x[h,"f"]

## f   
## 0.3

#f=0.3 gives you the highest total yield

#Run the model again for t=3  
  
#Define variables   
g = 0.5  
f = seq(0, 1, by=0.1)  
K=100  
t=3  
  
#Create matrices to store population size and yield  
pop = matrix(0, t, length(f))  
yield = matrix(0, t, length(f))  
  
#Set initial population size  
pop[1,] = 50  
yield[1,] = 50\*f  
  
#Run the loop  
for(i in 1:ncol(pop)){  
 for(n in 2:t){  
 pop[n, i] = pop[n-1, i] + pop[n-1, i]\*g\*(1-pop[n-1,i]/K) - pop[n-1, i]\*f[i]   
 yield[n, i] = pop[n, i]\*f[i]  
 }  
}  
  
#Create a vector to store the sum of yields for all fishing mortalities  
sum.yield = rep(0, ncol(yield))  
  
#Run a loop to calculate the yield sum for all values of f   
for(n in 1:length(sum.yield)){  
 sum.yield[n] = sum(yield[,n])  
}  
#Form a matrix with the sum of yields and f  
x = cbind(sum.yield, f)  
#match the maximum value  
h=match(max(sum.yield), x)  
x[h,"f"]

## f   
## 0.9

#Now f=0.9 gives you the highest yield

#Run the model again for t=50  
  
#Define variables   
g = 0.5  
f = seq(0, 1, by=0.1)  
K=100  
t=50  
  
#Create matrices to store population size and yield  
pop = matrix(0, t, length(f))  
yield = matrix(0, t, length(f))  
  
#Set initial population size  
pop[1,] = 50  
yield[1,] = 50\*f  
  
#Run the loop  
for(i in 1:ncol(pop)){  
 for(n in 2:t){  
 pop[n, i] = pop[n-1, i] + pop[n-1, i]\*g\*(1-pop[n-1,i]/K) - pop[n-1, i]\*f[i]   
 yield[n, i] = pop[n, i]\*f[i]  
 }  
}  
  
#Form a matrix with the yield[50,] and f  
x = cbind(yield[t,], f)  
  
#match the maximum value  
h=match(max(x[,1]), x)  
x[h,"f"]

## f   
## 0.3

##f=0.3 gives you the highest equilibrium yield

#Run the model again for g=0.2  
  
#Define variables   
g = 0.2  
f = seq(0, 1, by=0.1)  
K=100  
t=50  
  
#Create matrices to store population size and yield  
pop = matrix(0, t, length(f))  
yield = matrix(0, t, length(f))  
  
#Set initial population size  
pop[1,] = 50  
yield[1,] = 50\*f  
  
#Run the loop  
for(i in 1:ncol(pop)){  
 for(n in 2:t){  
 pop[n, i] = pop[n-1, i] + pop[n-1, i]\*g\*(1-pop[n-1,i]/K) - pop[n-1, i]\*f[i]   
 yield[n, i] = pop[n, i]\*f[i]  
 }  
}  
  
#Form a matrix with the yield[50,] and f  
x = cbind(yield[t,], f)  
  
#match the maximum value  
h=match(max(x[,1]), x)  
x[h,"f"]

## f   
## 0.1

#Now f=0.1 gives you the highest equilibrium yield