Jieyi Zhu(Jz6282)

BIO382K Project1

### 1. IBM population simulation description

* There are two color phenotypes: Red and Orange.
* This model will consist of a collection of reproducing agents.
* Reproduction and mortality will be stochastic functions, with reproductive rate depending on the total number of agents

### 2.Parameter descriptions and definitions

The model is

where , and . These function give random births and deaths that occur over one update step.

* : The current population size
* : The population size in next time interval
* : The maximum possible growth rate
* : The population size where each mother will produce one offspring.
* : The total number of offspring
* : The total number of death.
* : The mortality rate in each time interval

Colors = c("orange","red") #population phenotypes  
t\_max = 100 # maximum time duration of simulation  
r0 = log(2) # maxium births is 2 per mother  
K = 100 # population size where births is 1 per monther  
delta = 0.1 # natural mortality  
N0 = 5\*length(Colors) # initial number of individuals

### 3. Initializing the model

* Initialize the model with a function which conatins the ID, Colors, and age
* Store the population size of each colore phenotype into a data frame.

# give the starting attributes (id, color, age)  
make.inds <- function(id=NaN, color=NaN, age=NaN){  
 inds <- data.frame(id = id, color = color, age = 0)   
 inds  
}  
  
# initial population size of each color phenotype  
inds <- make.inds(id=1:N0, color=as.factor(array(Colors, dim=N0)))   
  
# store the population sizes of each color phenotype  
N <- data.frame(NaN, dim=c(t\_max + 1, length(Colors)))  
colnames(N) <- Colors  
N[1,] <- summary(inds$color)

### 4.Simulation

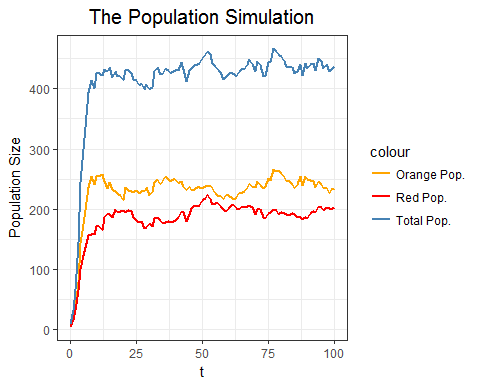
* looping the individual in the simulation

set.seed(12345)  
for (t in 1:t\_max + 1) # loop over time  
{  
 M = length(inds$age)# population size  
 lambda = exp(r0 \* (1 - M / K))   
 for (i in 1:M) # loop over individuals  
 {  
 j = rbinom(1, 1, delta) # will return a zero or one  
 if (j == 1) inds$age[i] = -1 # if j equals 1, mark as dead  
 else inds$age[i] = inds$age[i] + 1 # if j not equals to 1, the age will incrase by 1  
 n\_offsp = rpois(1, lambda) # the number of offspring per mother  
 if(n\_offsp>0){  
 offspring <- make.inds(id=seq(max(inds$id)+1, length.out=sum(n\_offsp)))  
 offspring$color <- inds[i,"color"]  
 inds = rbind(inds, offspring) # append the offrping to end of inds  
 }  
 }  
 inds= inds[inds$age != -1,] # exclude dead individuals  
 N[t,] = summary(inds$color) # put the population size of each color in N data frame  
}  
  
pop\_data = data.frame(t = 0:t\_max, N) # combine the population size and tiem interval  
pop\_data$total = pop\_data$orange + pop\_data$red # Get the total population size in each time interval

### 5. Plots

* Plot the population size in each color and plot the total population in each time interval
* Plot the Age distribution

library(ggplot2)  
# The population size plot  
cols <- c("Orange Pop."="orange","Red Pop."="red","Total Pop."= "steelblue")  
p\_pop <- ggplot(pop\_data, aes(x = t))  
p\_pop <- p\_pop+ geom\_line(aes(y = orange, col="Orange Pop."), lwd = 1 ) +  
 geom\_line(aes(y = red, color = "Red Pop."),lwd = 1) +  
 geom\_line(aes(y = total, color = "Total Pop."),lwd = 1)+ ylab("Population Size")  
  
p\_pop <- p\_pop +scale\_colour\_manual(values=cols)+ theme\_bw() + ggtitle("The Population Simulation") +  
 theme(plot.title = element\_text(hjust = 0.5, size = 15))   
p\_pop



# Age distribution  
p\_age <- ggplot(inds, aes(x = age)) + geom\_histogram(color='black', bins=50)  
p\_age <- p\_age + ggtitle("Age Distribution") + theme\_bw()+theme(plot.title = element\_text(hjust = 0.5, size=15))  
p\_age

