

## 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

$$N_{t+1} - N_t = Birth_t - Death_t$$

where  $Birth_t \sim \text{Poisson}(\lambda)$ ,  $\lambda = N_t e^{r_0(1-\frac{N_t}{K})}$  and  $Death_t \sim \text{Binom}(N_t, \delta)$ . These function give random births and deaths that occur over one update step.

- $N_t$  : The current population size
- $N_{t+1}$  : The population size in next time interval
- $r_0$  : The maximum possible growth rate
- $K$  : The population size where each mother will produce one offspring.
- $Birth_t$  : The total number of offspring
- $Death_t$  : The total number of death.
- $\delta$  : 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
    increase 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 offrrping 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
```

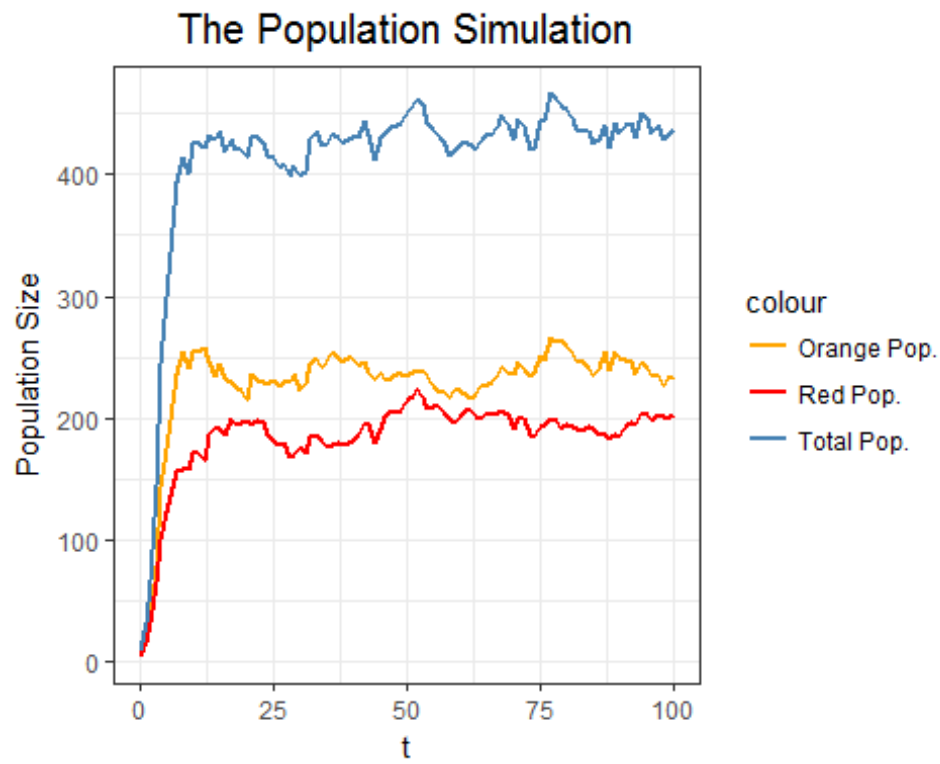
## 5. Plots

- 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
```

Age Distribution

