

A short introduction to
individual-based simulations
on R

rstudio-user-guide - RStudio

Go to file/function

Addins

rstudio-user-guide

Source

```

1 library(ggplot2)
2 mpg_plot <- ggplot(mpg, aes(x = displ, y = hwy)) +
3   geom_point(aes(colour = class))
4
5 mpg_plot
6

```

5:9 (Top Level) R Script

Console

```

> library(ggplot2)
> mpg_plot <- ggplot(mpg, aes(x = displ, y = hwy)) +
+   geom_point(aes(colour = class))
>
> mpg_plot
>

```

Console

Environments

Environment History Connections Build Git

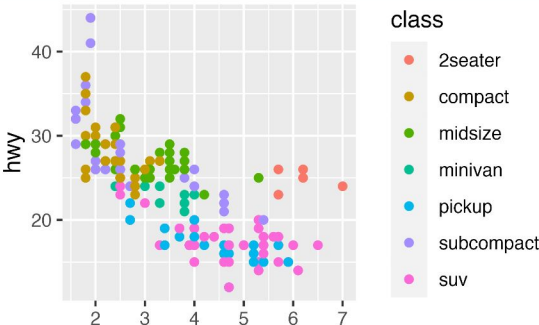
Import 220 MiB Grid

R Global Environment

Name	Type	Len...	Size	Value
mpg_plot	gg	9	29.1...	List of 9

Files Plots Packages Help Tutorial View

Zoom Export



class

- 2seater
- compact
- midsize
- minivan
- pickup
- subcompact
- suv

Output

Reminder

$$\mathbf{n}_{t+1} = L\mathbf{n}_t$$

$$\mathbf{n}_1 = L\mathbf{n}_0$$

$$\mathbf{n}_2 = L\mathbf{n}_1 = L^2\mathbf{n}_0$$

$$\mathbf{n}_3 = L\mathbf{n}_2 = L^3\mathbf{n}_0$$

⋮

$$\mathbf{n}_t = L^t\mathbf{n}_0$$

Age a (years)	p_a	m_a	f_a
0	0.25		
1	0.46	1.28	0.32
2	0.77	2.28	0.57
3	0.65	2.28	0.57
4	0.67	2.28	0.57
5	0.64	2.28	0.57
6	0.88	2.28	0.57
7		2.28	0.57

$$L = \begin{pmatrix} 0.32 & 0.57 & 0.57 & 0.57 & 0.57 & 0.57 & 0.57 \\ 0.46 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.77 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.65 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.67 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.64 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.88 & 0 \end{pmatrix}$$



Individual-based simulation

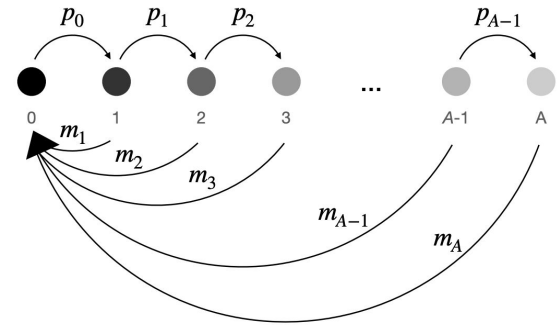
- An individual-based simulation (IBS) is a **computational model/tool** that simulates the behavior and interactions of individuals within a population.
- Each individual possesses unique **features** and follows **predefined rules** governing its behavior (e.g. reproduction, movement, and interaction with other individuals and the environment).
- It helps simulate **complex evolutionary scenarios** and understand how various factors contribute to the emergence of specific traits or adaptations within a population.

Today's goal:

We model the dynamics of a population, including **reproduction**, **aging** and **death**.

For simplicity, we assume individuals cannot live for more than six years, such that the population is structured into a newborn age-class (year 0) and six reproducing age-classes (from year 1 to year 6).

The fecundity and survival probability are equal across all ages.



INDIVIDUAL

What is an individual?

Which **features** characterize an individual?

AGE

in the
range

1, 2, 3, 4, 5, 6

1

6

3

4

1

2

1

4

1

2

1

Population

```
> possible_ages = c(1, 2, 3, 4, 5, 6)
> possible_ages
[1] 1 2 3 4 5 6
> A = sample(possible_ages, 10, replace = T)
> A
[1] 6 6 3 2 4 6 2 1 6 4
> A[1]
[1] 6
> A[5]
[1] 4
> length(A)
[1] 10
```

INDIVIDUAL

What is an individual?

Which **features** characterize an individual?

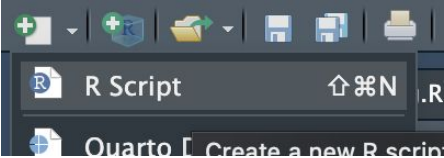
AGE

in the
range

1, 2, 3, 4, 5, 6

1 6 3 4 1 2 1 4 1 2 1

Population



```
Untitled1* x  Untitled2 x  Untitled3 x  R
← →  ↵  Source on Save  🔍  ✨  📄
1  # This is a comment...
2  ### Initialization
3  Population = rep(1, 10) # CTRL+Enter
4
```

```
> Population
[1] 1 1 1 1 1 1 1 1 1 1
```

INDIVIDUAL

What is an individual?

Which **features** characterize an individual?

What does the individual do?

What is the **life-cycle** of such individuals?
How does time progress?

Reproduce

$F_{ec} =$

1

$P_{suv} =$

0.5

```
# Reproduction
```

```
fec = 1
```

```
no_offspring = rpois(n = 1, lambda = fec)
```

```
> no_offspring
```

```
[1] 3
```


INDIVIDUAL

What is an individual?

Which **features** characterize an individual?

What does the individual do?

What is the **life-cycle** of such individuals?
How does time progress?

Reproduce

$F_{ec} =$

1

$P_{suv} =$

0.5

```
# Reproduction  
fec = 1  
psuv = 0.5  
no_offspring = rpois(n = 1, lambda = fec)  
no_new_n1 = rbinom(1, size = no_offspring, prob = psuv)
```

```
> no_new_n1  
[1] 2
```

INDIVIDUAL

What is an individual?

Which **features** characterize an individual?

What does the individual do?

What is the **life-cycle** of such individuals?
How does time progress?

Reproduce

$F_{ec} =$

1

$P_{suv} =$

0.5

```
# Reproduction  
fec = 1  
psuv = 0.5  
no_offspring = rpois(n = 1, lambda = fec)  
no_new_n1 = rbinom(1, size = no_offspring, prob = psuv)  
new_n1 = rep(1, no_new_n1)
```

```
> new_n1  
[1] 1 1
```

INDIVIDUAL

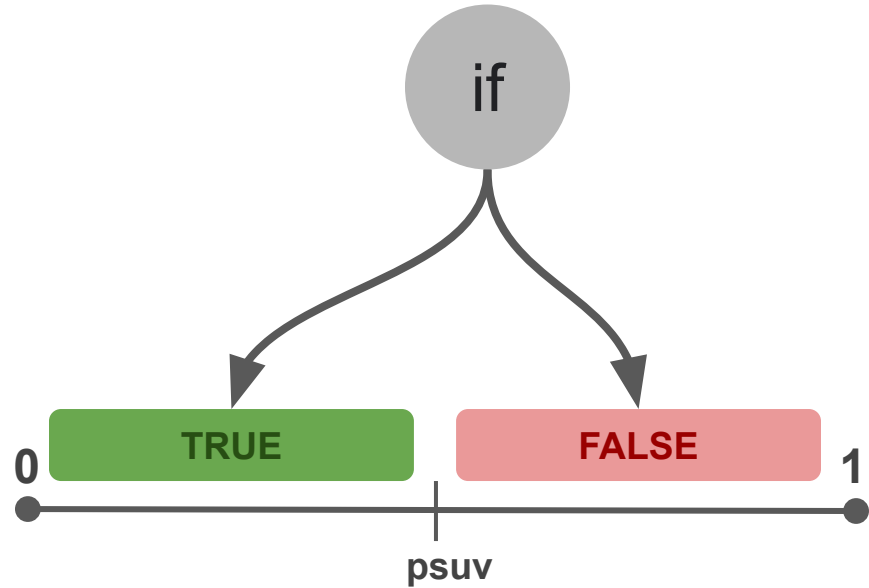
What does the individual do?

What is the **life-cycle** of such individuals?
How does time progress?

Try to survive
to next year

$P_{\text{suv}} = 0.5$

```
if (runif(1) < psuv){  
  # Survives and ages  
} else {  
  # Dies  
}
```



INDIVIDUAL

What does the individual do?

What is the **life-cycle** of such individuals?
How does time progress?

**Try to survive
to next year**

$P_{\text{suv}} =$ **0.5**

```
> runif(1) < 0.5
[1] TRUE
> runif(1) < 0.5
[1] TRUE
> runif(1) < 0.5
[1] TRUE
> runif(1) < 0.5
[1] FALSE
> runif(1) < 0.5
[1] FALSE
> runif(1) < 0.5
[1] FALSE
> runif(1) < 0.5
[1] TRUE
> runif(1) < 0.5
[1] FALSE
```

INDIVIDUAL

What does the individual do?

What is the **life-cycle** of such individuals?
How does time progress?

**Try to survive
to next year**

$P_{\text{suv}} =$ **0.5**

```
8 psuv = 0.5
9 age = 2
10 if (runif(1) < psuv){
11     # Survives and ages
12     age = age + 1
13 } else {
14     # Dies
15     age = NA
16 }
```

Reproduce

$F_{ec} =$

1.0

Try to survive
to next year

$P_{suv} =$

0.5

```
IND_ACT = function(age, fec, psuv){  
  # Reproduction  
  no_offspring = rpois(n = 1, lambda = fec)  
  no_new_n1 = rbinom(1, size = no_offspring, prob = psuv)  
  new_n1 = rep(1, no_new_n1)  
  
  # Try to survive  
  if (runif(1) < psuv){  
    # Survives and ages  
    age = age + 1  
  } else {  
    # Dies  
    age = NA  
  }  
  
  # Returns new adults and individual  
  return(c(new_n1, age))  
} # CTRL+Enter
```

Reproduce

$F_{ec} = 1.0$

Try to survive
to next year

$P_{suv} = 0.5$

```
> IND_ACT(age = 1, fec = 1, psuv = 0.5)
[1] 1 1 1 NA
> IND_ACT(age = 1, fec = 1, psuv = 0.5)
[1] 1 2
> IND_ACT(age = 1, fec = 1, psuv = 0.5)
[1] 1 NA
> IND_ACT(age = 1, fec = 1, psuv = 0.5)
[1] NA
> IND_ACT(age = 1, fec = 1, psuv = 0.5)
[1] 2
> IND_ACT(age = 1, fec = 1, psuv = 0.5)
[1] 1 2
```

```
> IND_ACT(age = 6, fec = 1, psuv = 0.5)
[1] 7
```

Reproduce

$F_{ec} =$

1.0

Try to survive
to next year

$P_{suv} =$

0.5

```
IND_ACT = function(age, fec, psuv){
  # Reproduction
  no_offspring = rpois(n = 1, lambda = fec)
  no_new_n1 = rbinom(1, size = no_offspring, prob = psuv)
  new_n1 = rep(1, no_new_n1)

  # Try to survive
  if (runif(1) < psuv){
    # Survives and ages
    age = age + 1
    if (age > 6){ age = NA } # Caps max age at 6
  } else {
    # Dies
    age = NA
  }

  # Returns new adults and individual
  return(c(new_n1, age))
} # CTRL+Enter
```



```


1 # This is a comment...
2
3 IND_ACT = function(age, fec, psuv){
4   # Reproduction
5   no_offspring = rpois(n = 1, lambda = fec)
6   no_new_n1 = rbinom(1, size = no_offspring, prob = psuv)
7   new_n1 = rep(1, no_new_n1)
8
9   # Try to survive
10  if (runif(1) < psuv){
11    # Survives and ages
12    age = age + 1
13    if (age > 6){ age = NA } # Caps max age at 6
14  } else {
15    # Dies
16    age = NA
17  }
18
19  # Returns new adults and individual
20  return(c(new_n1, age))
21 } # CTRL+Enter
22
23 ### Initialization
24 Population = rep(1, 10) # CTRL+Enter
25
26
27

```

```

1 # This is a comment...
2
3 IND_ACT = function(age, fec, psuv){ } # CTRL+Enter
22
23 ### Initialization
24 Population = rep(1, 10) # CTRL+Enter
25

```

```
1 # This is a comment...
2
3 ▶ IND_ACT = function(age, fec, psuv){} # CTRL+Enter
22
23 ### Initialization
24 Population = rep(1, 100) # CTRL+Enter ← Let's increase initial population size
25
26
27 NextPopulation = c()
28 ▼ for(i in 1:length(Population)){
29   # Append to NextPopulation all "families"
30   NextPopulation = c(NextPopulation, IND_ACT(age = Population[i], fec = 1.0, psuv = 0.5))
31 ▲ }
32
```

```
> table(NextPopulation)
```

```
NextPopulation
```

```
 1  2
50 51
```

```
> NextPopulation
```

```
[1] 2 NA 1 2 1 2 2 2 1 NA 1 1 NA 1 2 2 2 2 1 NA 2 NA 1 NA 2 2 2 1 NA 1 NA NA NA NA 1 2 NA NA  
[39] 1 NA 1 1 2 2 NA 2 NA 1 NA 2 NA 2 1 1 1 2 2 1 1 1 NA NA 1 2 1 NA 1 NA 2 NA NA 1 NA 2 2 2  
[77] NA 1 1 2 1 1 NA NA NA 1 NA 1 NA 1 1 2 NA 2 NA 1 NA 1 2 1 2 2 2 1 NA NA 2 1 2 1 1 NA 1 2  
[115] 2 1 NA NA 1 NA NA 2 2 NA 2 NA 2 2 2 2 2 2 NA 2 1 NA 2 NA 1 NA 1 1 2 2 1 NA 1 2 1 NA
```

```
> is.na(NextPopulation)
```






```
[1] FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE  
[20] TRUE FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE  
[39] FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE  
[58] FALSE FALSE FALSE TRUE TRUE FALSE FALSE FALSE TRUE FALSE TRUE FALSE TRUE TRUE FALSE TRUE FALSE FALSE  
[77] TRUE FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE FALSE TRUE FALSE TRUE FALSE FALSE FALSE TRUE FALSE TRUE  
[96] FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE  
[115] FALSE FALSE TRUE TRUE FALSE TRUE TRUE FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE  
[134] FALSE FALSE TRUE FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE
```


```
> !is.na(NextPopulation)
```

```
[1] TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE  
[20] FALSE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE FALSE FALSE FALSE FALSE TRUE TRUE FALSE FALSE  
[39] TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE  
[58] TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE FALSE TRUE FALSE TRUE FALSE FALSE TRUE FALSE TRUE TRUE TRUE  
[77] FALSE TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE TRUE FALSE TRUE FALSE TRUE TRUE TRUE FALSE TRUE FALSE  
[96] TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE  
[115] TRUE TRUE FALSE FALSE TRUE FALSE FALSE TRUE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE  
[134] TRUE TRUE FALSE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE
```

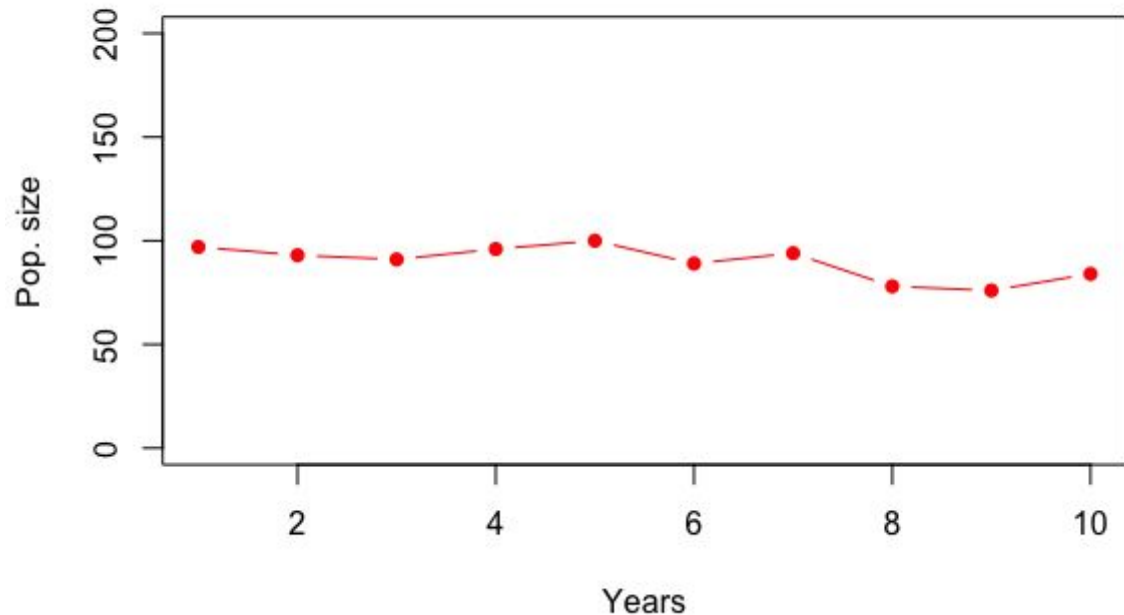
```
> NextPopulation[!is.na(NextPopulation)]
```

```
[1] 2 1 2 1 2 2 2 1 1 1 1 2 2 2 2 1 2 1 2 2 2 1 1 1 2 1 1 1 2 2 2 1 2 2 1 1 1 1 2 1 1 2 1 2 2 2 1 1 2 1 1 1 1  
[59] 1 1 2 2 1 1 2 1 2 2 2 1 2 1 2 1 1 1 2 2 1 1 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 2 2 1 1 2 1
```


```
1 # This is a comment...
2
3 ▶ IND_ACT = function(age, fec, psuv){} # CTRL+Enter
22
23 ### Initialization
24 Population = rep(1, 100) # CTRL+Enter
25 FinalYear = 10 
26
27 ▼ for (year in 1:FinalYear){ 
28     NextPopulation = c()
29
30 ▼ for(i in 1:length(Population)){
31     # Append to NextPopulation all "families"
32     NextPopulation = c(NextPopulation, IND_ACT(age = Population[i], fec = 1.0, psuv = 0.5))
33 ▲ }
34
35     Population = NextPopulation[!is.na(NextPopulation)] 
36 ▲ } 
```


```
1 # This is a comment...
2
3 ▶ IND_ACT = function(age, fec, psuv){ # CTRL+Enter
22
23 ### Initialization
24 Population = rep(1, 100) # CTRL+Enter
25 FinalYear = 10
26 PopulationSize = rep(0, FinalYear) ←
27
28 ▼ for (year in 1:FinalYear){
29     NextPopulation = c()
30
31 ▼ for(i in 1:length(Population)){
32     # Append to NextPopulation all "families"
33     NextPopulation = c(NextPopulation, IND_ACT(age = Population[i], fec = 1.0, psuv = 0.5))
34 ▲ }
35
36     Population = NextPopulation[!is.na(NextPopulation)]
37     PopulationSize[year] = length(Population) ←
38 ▲ }
39
```

```
plot(x = 1:FinalYear, y = PopulationSize,  
     type = "b",  
     col = "red", pch = 16,  
     ylim = c(0, 2*max(PopulationSize)),  
     xlab = "Years", ylab = "Pop. size")
```



```
1 # This is a comment...
2
3 IND_ACT = function(age, fec, psuv){ } # CTRL+Enter
22
23 DYN = function(n0, FinalYear, fec, psuv){ ←
24   ### Initialization
25   Population = rep(1, n0) ←
26   PopulationSize = rep(0, FinalYear)
27
28   for (year in 1:FinalYear){
29     NextPopulation = c()
30
31     for(i in 1:length(Population)){
32       # Append to NextPopulation all "families"
33       NextPopulation = c(NextPopulation, IND_ACT(age = Population[i], fec = fec, psuv = psuv))
34     }
35
36     Population = NextPopulation[!is.na(NextPopulation)]
37     PopulationSize[year] = length(Population)
38   }
39
40   return(PopulationSize) ←
41 } ←
42
43 # Runs one simulation with a set of parameters
44 sim = DYN(n0 = 100, FinalYear = 100, fec = 1.0, psuv = 0.5) ←
45
46 # Plots the output of the simulation
47 plot(x = 1:100, y = sim, ←
48       type = "b",
49       col = "red", pch = 16,
50       ylim = c(0, 2*max(sim)), ←
51       xlab = "Years", ylab = "Pop. size")
52
```

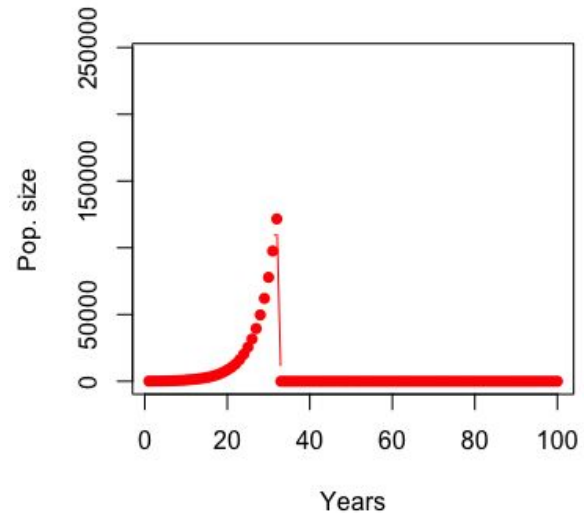
```
1 # This is a comment...
2
3 ▸ IND_ACT = function(age, fec, psuv){ # CTRL+Enter
22
23 ▾ DYN = function(n0, FinalYear, fec, psuv){
24   ### Initialization
25   Population = rep(1, n0)
26   PopulationSize = rep(0, FinalYear)
27
28 ▾   for (year in 1:FinalYear){
29     NextPopulation = c()
30
31 ▾     for(i in 1:length(Population)){
32       # Append to NextPopulation all "families"
33       NextPopulation = c(NextPopulation, IND_ACT(age = Population[i], fec = fec, psuv = psuv))
34 ▸     }
35
36     Population = NextPopulation[!is.na(NextPopulation)]
37     PopulationSize[year] = length(Population)
38
39     if( length(Population) == 0 ){ break } # Population went extinct
40     if( length(Population) > 100000 ){ break } # Population skyrocketed
41 ▸   }
42
43   return(PopulationSize)
44 ▸ }
```




```

1 # This is a comment...
2
3 ▶ IND_ACT = function(age, fec, psuv){ } # CTRL+Enter
22
23 ▶ DYN = function(n0, FinalYear, fec, psuv){ }
45
46 # Runs one simulation with a set of parameters
47 sim = DYN(n0 = 100, FinalYear = 100, fec = 1.5, psuv = 0.5)
48
49 # Plots the output of the simulation
50 plot(x = 1:100, y = sim,
51      type = "b",
52      col = "red", pch = 16,
53      ylim = c(0, 2*max(sim)),
54      xlab = "Years", ylab = "Pop. size")
55
56

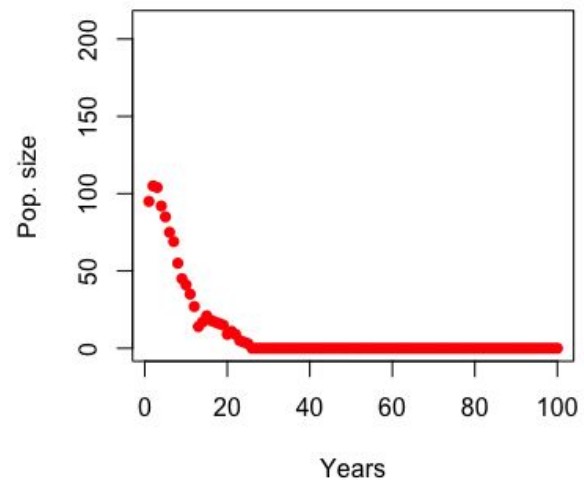
```



```

1 # This is a comment...
2
3 ▶ IND_ACT = function(age, fec, psuv){ } # CTRL+Enter
22
23 ▶ DYN = function(n0, FinalYear, fec, psuv){ }
45
46 # Runs one simulation with a set of parameters
47 sim = DYN(n0 = 100, FinalYear = 100, fec = 0.8, psuv = 0.5)
48
49 # Plots the output of the simulation
50 plot(x = 1:100, y = sim,
51      type = "b",
52      col = "red", pch = 16,
53      ylim = c(0, 2*max(sim)),
54      xlab = "Years", ylab = "Pop. size")
55
56

```



Now, it's up to you:

We model the dynamics of a population, including **reproduction**, **aging** and **death**.

For simplicity, we assume individuals cannot live for more than six years, such that the population is structured into a newborn age-class (year 0) and six reproducing age-classes (from year 1 to year 6).

The **fecundity** and **survival probability** differ across ages.

Age a (in years)	Fecundity m_a	Survival probability p_a
0	-	0.8
1	0.57	0.52
2	2.10	0.60
3	4.25	0.71
4	4.25	0.71
5	4.25	0.71
6	4.25	-

Table 1: *Sus scrofa* life-table.