

Simular la evolución

Un ejemplo en R:
Animales del desierto

Referencias

- A.K. Dewdney (1989) Simulated evolution: wherein bugs learn to hunt bacteria, Scientific American, mayo: 138-141
- C. Barski (2011) Land of Lisp 202-218

Simular la evolución



Escenario



Escenario

```
## desierto toroidal
```

```
## con un oasis en el centro
```

```
ancho <- 100
```

```
alto <- 30
```

```
oasis <- c(45, 10, 10, 10)
```

Plantas

```
energia.planta <- 80  # días-vida que proporciona
plantas        <- matrix (FALSE, ancho, alto)
planta.aleatoria <-
function (izquierda, arriba, ancho, alto)
{
  pos.x <- izquierda + sample(ancho,1)
  pos.y <- arriba     + sample(alto,1)
  plantas[pos.x,pos.y] <<- TRUE
}
```

Plantas

```
poner.plantas <- function ()  
  ## añade dos plantas al mundo  
  ## (una en general, otra en el oasis)  
  {  
    planta.aleatoria (oasis[1], oasis[2],  
                      oasis[3], oasis[4])  
    planta.aleatoria (0,      0,  
                      ancho,  alto  )  
  }
```

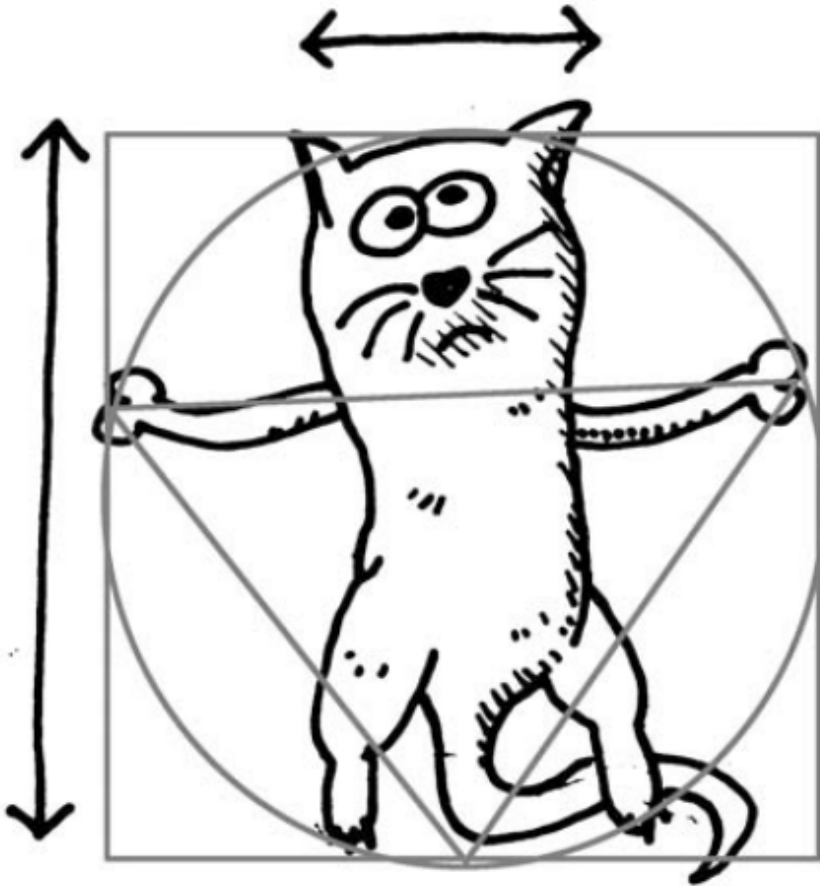
Animales



Animales

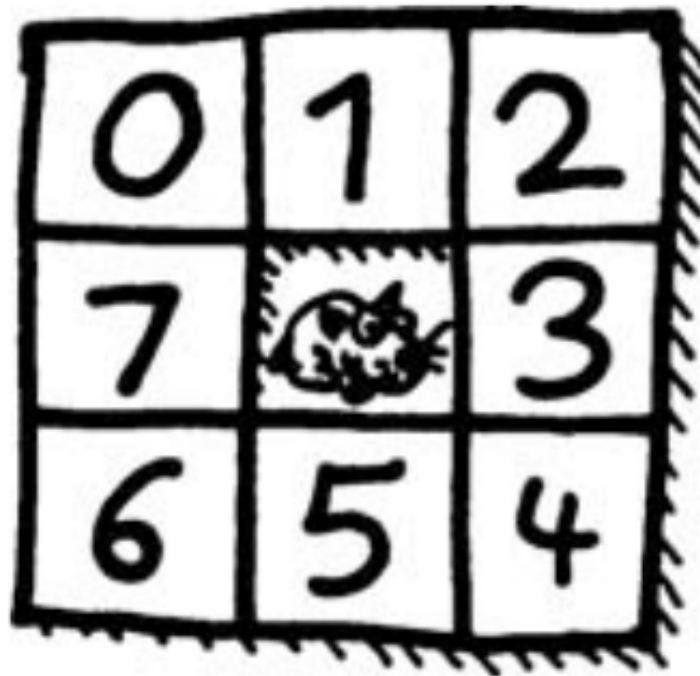
```
hacer.animal <- function (x, y,  
                           energia, dir, genes)  
  list (x          = x,          # coordenadas  
        y          = y,  
        energia    = energia,    # días de vida  
        dir        = dir,        # 0=N0, 1=N... 7=0  
        genes      = genes)      # pesos de cada  
                                   # ángulo: 0=0°,  
                                   # 1=45°... 7 = 315°
```

Animales: x, y

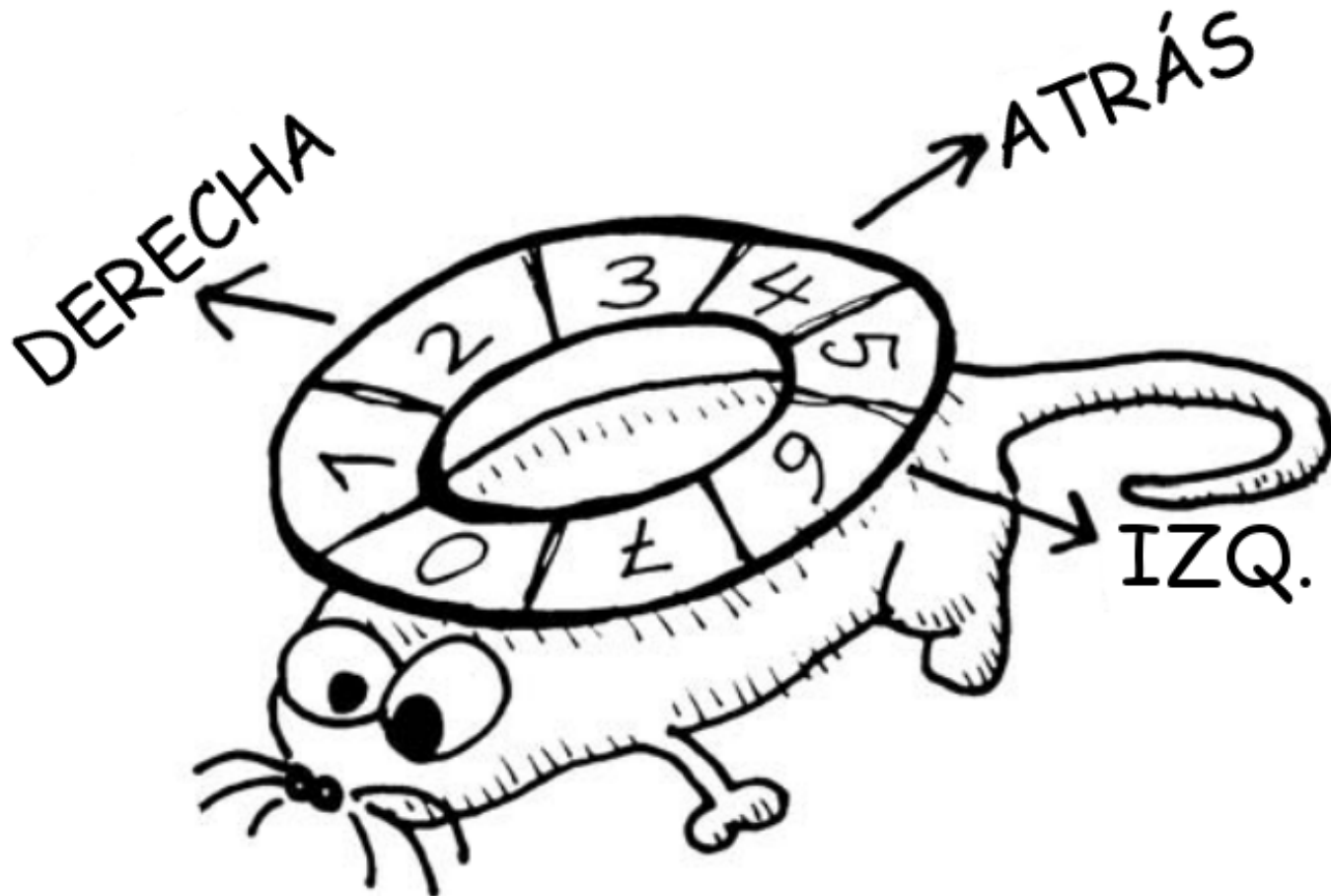


- ¡no!, tamaño no
- posición en el escenario

Animales: dir



Animales: genes



Animales: genes

- Desplazamiento según ruleta
- Ejemplo de la derecha: en círculos mayormente

locus	alelo
0	1
1	1
2	10
3	1
4	1
5	1
6	1
7	1

Animales

```
## población primigenia: 1 animal  
animales <-  
  list (hacer.animal (x      = ancho/2,  
                      y      = alto/2,  
                      energia = 1000,  
                      dir     = 0, # N.O.  
          genes = sample (10, 8,  
                          replace=TRUE) ) )
```

[illegible]

Animales: mover



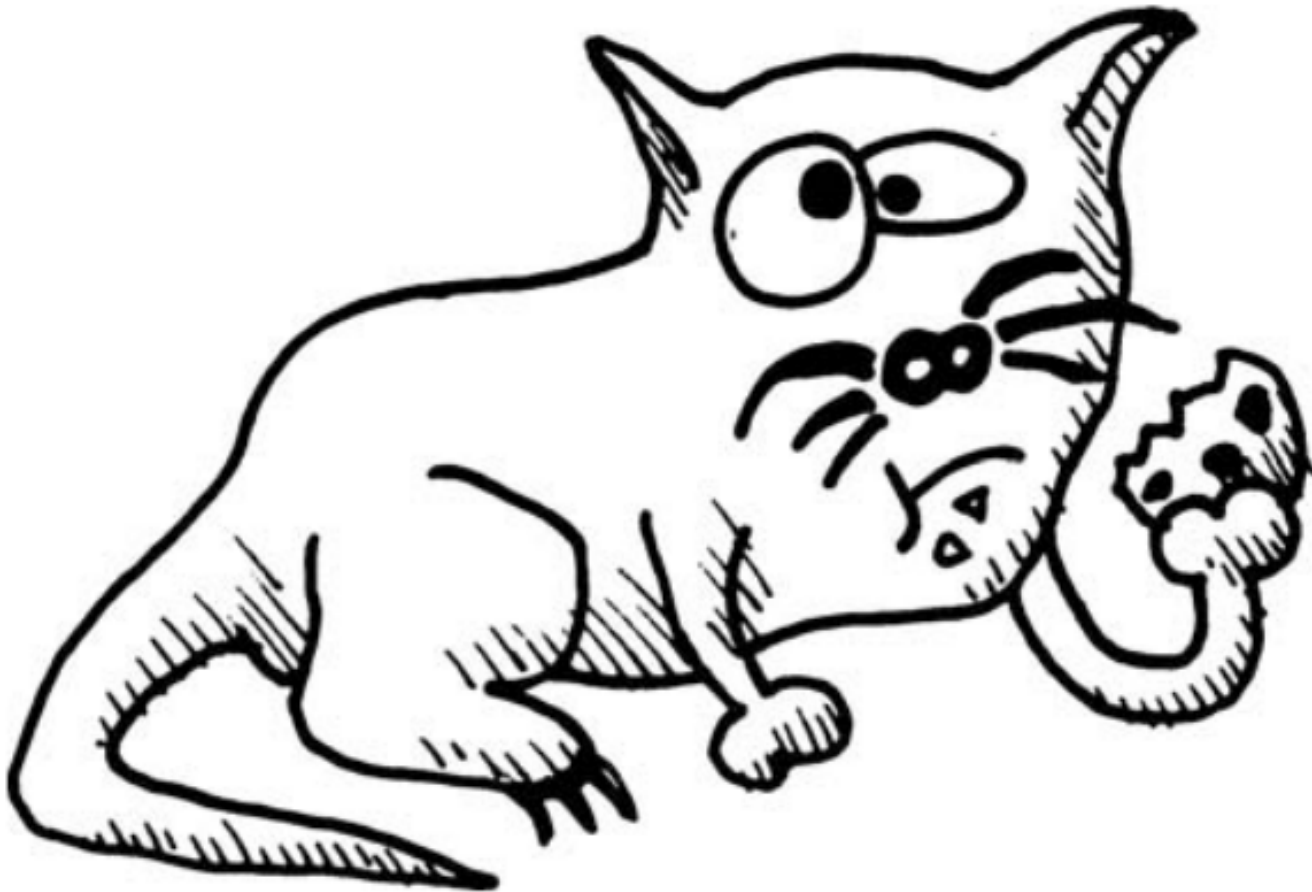

```
mover <- function (animal)
{
  dir      <- animal$dir
  x        <- animal$x
  y        <- animal$y
  animal$x <- (x - 1 +
    ifelse (2<=dir&dir<=4,
            1,
            ifelse (dir==1|dir==5,
                    0,
                    -1))) %% ancho + 1
```

```
animal$y <- (y - 1 +  
  ifelse (0<=dir&dir<=2,  
    -1,  
    ifelse (4<=dir&dir<=6,  
      1,  
      0))) %% alto + 1  
animal$energia <- animal$energia - 1  
animal  
}
```

Animal: giro

```
girar <- function (animal)
{
  angulo <- sample (0:7, 1,
                    prob=animal$genes)
  animal$dir <- (animal$dir +
                angulo) %% 8
  animal
}
```

Nutrición



```
comer <- function (animal)
{
  posx <- animal$x
  posy <- animal$y
  if (plantas[posx,posy])
  {
    animal$energia <-
      animal$energia + energia.planta
    plantas[posx,posy] <<- FALSE
  }
  animal
}
```

Reproducción



```
energia.reproduccion <- 200
# energía mínima para ser progenitor
reproducir <- function (animal)
{
  e <- animal$energia
  if (e >= energia.reproduccion)
  {
    animal$energia <- round (animal$energia / 2)
    animal.nuevo    <- animal # hacemos una copia
    genes           <- animal.nuevo$genes
```

```
mutacion <- sample (8, 1)
genes[mutacion] <- max (1,
  genes[mutacion] + sample (-1:1,1))
# "-" unario
animal.nuevo$genes <- genes
animales <<- c (animales,
                list (animal.nuevo))
}
animal
}
```


Iterar: cómo transcurre un día

```
iterar <- function ()  
{ ## eliminamos animales muertos  
  animales <- animales [sapply (animales,  
                                function (animal) animal$energia >= 0)]  
  ## los restantes viven; nacen nuevas plantas  
  for (i in 1:length(animales)) {  
    animales[[i]] <- girar      (animales[[i]])  
    animales[[i]] <- mover      (animales[[i]])  
    animales[[i]] <- comer      (animales[[i]])  
    animales[[i]] <- reproducir (animales[[i]])  
  }  
  poner.plantas () }
```

```
evolucion <- function () {  
  dibujar ()  
  cat ("Intro->seguir; fin->salir; nº iter:")  
  cadena <- readline () #espera a pulsar Intro  
  if (cadena == "") {iterar (); evolucion ()}  
  else if (cadena == "fin") return ()  
  else { x <- as.integer (cadena)  
    if (!is.na(x))  
      for (i in 1:x) {  
        iterar ()  
        if (i %% 1000 == 0) cat (".") }  
    evolucion () } }
```

Pruebas

- `R> evolution() # intro intro intro ...`
- `: 100`
- `: 100000`
- `: fin`
- `R> animales[[1]]`
- `R> sapply (animales,
 function (a) a$genes)`

Tareas

- ¿Cuántas «especies»?
 - Componentes principales
 - Conglomerados
 - K-medias
- ¿Cómo caracterizarlas fenotípicamente?
 - ¿distancia? ¿energía? ¿terreno cubierto?
- ¿Función adaptación?

R

- `evolucion ()`
- `g <- sapply (animales, function (x) x$genes)`
- `p <- princomp (t (g))`
- `plot (p) ; biplot (p)`
- `plot (hclust (dist (t (g))))`
- `kmeans (t (g),2) ; kmeans (t (g),2)`

