

Script2

Yosune Miquelajauregui

14/12/2017

Script de la segunda clase de estadística y modelación de sistemas socioecológicos en R.

Generación de datos, bucles y funciones

Uso de las funciones seq() y rep()

```
Años <- seq(1:1000)
str(Años)

## int [1:1000] 1 2 3 4 5 6 7 8 9 10 ...

tmp <- seq(1, 20, by = 3)
tmp

## [1] 1 4 7 10 13 16 19

tmp2 <- seq(0, 1, length.out=10)
tmp2

## [1] 0.0000000 0.1111111 0.2222222 0.3333333 0.4444444 0.5555556
## [2] 0.6666667
## [8] 0.7777778 0.8888889 1.0000000
```

Generación de factores. Uso de c() y rep()

```
Municipio <- c("Tlahuac", "Tlalpan", "Miguel Hidalgo", "Cuauhtemoc",
"Venustiano Carranza", "Coyoacan", "Benito Juarez",
"Cuajimalpa", "Magdalena Contreras", "Xochimilco", "Azcapotzalco", "Gustavo
A. Madero", "Iztacalco", "Iztapalapa", "Milpa Alta", "Alvaro Obregon")
Mu <- rep(Municipio, each=8)
str(Mu)

## chr [1:128] "Tlahuac" "Tlahuac" "Tlahuac" "Tlahuac" ...
```

Uso de rep()

```
rep(1:20)

## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

```
rep(1:20, each = 2)

## [1] 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10 11 11
12
## [24] 12 13 13 14 14 15 15 16 16 17 17 18 18 19 19 20 20
```

Generar datos aleatorios

```
Datosaleatoriosnormal <- rnorm(1000,0.5,0.5)
Datosaleatoriosbinom <- rbinom (50, 50,0.5)
Datosaleatoriospoi <- rpois(1000,0.2)
str(Datosaleatoriosnormal)

## num [1:1000] -4.173 2.594 -5.407 -0.298 0.911 ...

str(Datosaleatoriosbinom)

## int [1:50] 26 21 29 25 23 26 24 29 28 17 ...

str(Datosaleatoriospoi)

## int [1:1000] 0 0 0 0 0 0 1 0 1 0 ...
```

Bucles

Generar bucles “for”

```
tmp<-seq(30,1000,by=2)
rc <- numeric(length(tmp))
for (i in 1:length(tmp)){
rc[i] <- sqrt(tmp[i])
rc
}
head(rc)

## [1] 5.477226 5.656854 5.830952 6.000000 6.164414 6.324555

m <- 20
n <- 0
for (k in 1:length(m)){
n[k] <- m*2/4
n
}
n

## [1] 10

PoblacionBallenas <- seq (1200,2000,by=100)
NuevaPoblacion <-
  numeric(length(PoblacionBallenas))
Nacimientos <- 2
```

```

for (b in 1:length(PoblacionBallenas)){
  NuevaPoblacion[b] <- PoblacionBallenas [b] + Nacimientos
  NuevaPoblacion
}
head(PoblacionBallenas)

## [1] 1200 1300 1400 1500 1600 1700

head(NuevaPoblacion)

## [1] 1202 1302 1402 1502 1602 1702

PoblacionBallenas <- seq (1200,2000,by=100)
CambioPob <- numeric(length(PoblacionBallenas)-1)

for (b in 1:length(PoblacionBallenas)-1){
  CambioPob[b] <- PoblacionBallenas [b+1] - PoblacionBallenas [b]
  CambioPob
}
CambioPob

## [1] 100 100 100 100 100 100 100 100

```

Generar bucles “while”

```

i <- 1
while (i < 6) {
  print(i)
  i = i+1
}

## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5

numero <- 1
while (numero <5){
  print("este número es menor que cinco")
  numero <- numero+1
}

## [1] "este número es menor que cinco"
## [1] "este número es menor que cinco"
## [1] "este número es menor que cinco"
## [1] "este número es menor que cinco"

```

Generar bucles “repeat”

```

x <- 1
repeat {
  print(x)
}

```

```
x = x+1
if (x == 6){
  break
}
}

## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
```

Funciones

```
primerafuncion <- function (x){
  x <- x*5
  x
}

segundafuncion <- function (x, y){
  tmp <- x*y
  tmp
}

tercerafuncion <- function (x){
  tmp1 <- mean(x[,1])
  tmp2 <- mean(x[,2])
  tmp3 <- mean(x[,3])
  return (c(tmp1,tmp2,tmp3))
}
```

Evaluar las funciones

```
primerafuncion(5)
```

```
## [1] 25
```

```
primerafuncion(8)
```

```
## [1] 40
```

```
primerafuncion(1000)
```

```
## [1] 5000
```

```
segundafuncion(3,3)
```

```
## [1] 9
```

```
segundafuncion(1000,3)
```

```
## [1] 3000
```

```
segundafuncion(8789,8768)
```

```
## [1] 77061952
```

```
Var1 <- seq(0, 1, length.out=100)
```

```
Var2 <- seq(20, 700, length.out=100)
```

```
Var3 <-seq(103, 305, length.out=100)
```

```
Datos3<- cbind(Var1,Var2, Var3)
```

```
head(Datos3)
```

```
##           Var1      Var2      Var3
```

```
## [1,] 0.00000000 20.00000 103.0000
```

```
## [2,] 0.01010101 26.86869 105.0404
```

```
## [3,] 0.02020202 33.73737 107.0808
```

```
## [4,] 0.03030303 40.60606 109.1212
```

```
## [5,] 0.04040404 47.47475 111.1616
```

```
## [6,] 0.05050505 54.34343 113.2020
```

```
str(Datos3)
```

```
## num [1:100, 1:3] 0 0.0101 0.0202 0.0303 0.0404 ...
```

```
## - attr(*, "dimnames")=List of 2
```

```
## ..$ : NULL
```

```
## ..$ : chr [1:3] "Var1" "Var2" "Var3"
```

```
tercerafuncion(Datos3)
```

```
## [1] 0.5 360.0 204.0
```

Podemos guardar las funciones como objetos en R para tener acceso a ellas en cualquier momento.

```
save(segundafuncion,file="segundafuncion.Rdata")
```

```
load(file="segundafuncion.Rdata")
```