

Matematika Diskretua

Laboratorio saioa: RStudio-rekin hasten

Laboratorio saio honen helburua R lengoaia erabiltzen hastea da.

Ariketak

1. Laguntza aztertu. RStudio irekitzean, pantaila lau zatitan banatuta ikusiko duzu. Rrekin hasten laguntzeko, erabil ezazu Rren laguntza. Horretarako, behean ezkerretan `help.start()` idatzi beharko duzu. Behean eskuinean laguntza agertuko da. "Manuals" ataleko "An Introduction to R" aukeran laguntzarako informazioa agertuko zaizu. Har ezazu tarte bat zer dagoen ikusteko. Begira itzazu bereziki honakoak: (2) Simple manipulations numbers and vectors, (9) Grouping, loops and conditional execution, (10) Writing your own functions, (13) Packages, (Appendix D) Function and variable index.

R softwarearen oinarrizko funtzioak "base" izeneko paketeen daude. RStudio bertan `library(help = "base")` aginduaren bidez funtzio guztien zerrenda aterako zaizu. On-line manuala hemen aurkituko duzu: "The R Base Package":

<http://stat.ethz.ch/R-manual/R-devel/library/base/html/00Index.html>

2. Dokumentazioa aztertuz funtzio eta datu egitura oso erraz batzuk erabiliko ditugu. Beheko ezkerreko lehioan idazten duzuna interpretatua izango da. Interpretatua izan den guztia RStudio-ren memorian gordeta geratzen da, eta goiko eskuineko lehio zatian ageri da. Goiko ezkerreko lehio zatian idazten duzuna .R formatua izango duen fitxategi batean gordetzeko aukera izango duzu. Hor idazten dira, normalean, R funtzioak. Laboratorio saioaren amaieran bertan idatzitakoarekin sortuko duzun .R fitxategia eGela plataformara igoko duzu.

- Assignment Operators (`<`, `-`, `>`, `=`),

```
x<-2
3-> y
z=4
```

- Arithmetic Operators (`+`, `-`, `*`, `/`, `^`, `%%`, `%/%`),

```
x+y
(x+y)-z
((x+y)-z)*z/x
x^y
```

Laguntzan sakatu `%%` ze eragiketa den ikusteko

Value atalean zera dio:

`%%` indicates `x mod y` and `%/%` indicates integer division.

```
(y+z) %% x
```

```
(y+z) %/% x
```

- Relational Operators (<, >, <=, >=, ==, !=)

```
x<y      x>y      x<=y
x>=y     x==y     x!=y
```

- Logical Operators (logical negation !, logical AND &, logical OR |),

```
!(x<y)
x<y
y<x
(x<y)&(y<x)
(x<y)|(y<x)
```

- Sequence Generation (c, colon ":", seq), length

```
a<-c(1,2,3)
b<-c(x,y,z)
length(a)
Bektoreko osagai bat hartzeko:
a[1]
1:10
4:20
x:z
2tik 20ra binaka    seq(2,20,2)
```

- Print, Concatenate and print

```
print(5+5)
cat("Hiru gehi hiru (", y, "+", y, ") sei da", y+y)
```

- Control flow (if, if else, while, for)

```
if (x<y) y-x
if (y<x) x-y else 0
while (x<z) {print(x); x<-x+1}
for (i in 1:10) {print(i)}
for (i in 1:10) {print("a")}
```

- Balio bat itzultzen duen funtzio baten definizioa (function, return)

```
luzera_itzuli<-function(v)
{bek_luzera=length(v);
  return(bek_luzera);
}
v<-c(1,2)
luzera_itzuli(v)
```

- Bestelakoak (is.numeric, as.numeric)

```
is.numeric(5)      is.numeric("5")
v=c("kaixo", 1,2,3)
is.numeric(v[1])  is.numeric(v[2])
v[2]+v[3]
as.numeric(v[2])+as.numeric(v[3])
```

3. Behean duzun R kodean "batuketa" funtzio bat definitzen da, 2 osagaiko bektore bat parametro moduan jaso eta bien batura itzuliko duena.

```
batuketa<-function(v)
{v1<-v[1];
  v2<-v[2];
  cat("Bektorea=(",v1,",",v2,")", "Bi osagaien batura=", v1+v2, "\n");
}
```

Funtzioari dei egingo diogu, ea espero dugun emaitza lortzen dugun. Probarako, hona hemen 2 osagaiko bektore baten definizioa, eta deia "batuketa" funtzioari.

```
v<-c(1,2)
length(v)
## [1] 2
batuketa(v)
## Bektorea=( 1 , 2 ), Bi osagaien batura= 3
```

Adibide horretan oinarrituz, asma itzazu zenbait funtzio bektore horretako osagaien arteko eragiketa aritmetikoak egin eta emaitza itzultzeko. Adibidez,

- 3.1 Edozein luzerako bektore bateko osagai guztien batura.

```
batuketa2<-function(v)
{
}
```

Dei egin iezaiozu zure funtzioari parametro moduan honako bektoreak erabiliz eta egiaztatu espero duzun emaitza lortzen duzula.

```
## Deiak funtzioari probak egiteko:
w1<-c(3,8,10,2,4,9)
w2<-c(3,8,10)
w3<-c(3,8,10,0,-1,9)
batuketa2(w1)
## [1] 36
batuketa2(w2)
## [1] 21
batuketa2(w3)
## [1] 29
```

- 3.2 Bektore bat emanik, lehenengo posizioan adieraziko da gainerako batura ala biderkadura egin behar den.

```
batuketa3<-function(v)
{
}
```

Dei egin iezaiozu zure funtzioari parametro moduan honako bektoreak erabiliz eta egiaztatu espero duzun emaitza lortzen duzula.

```
v1=c("batu", 1,2,3,4,5,6)
batuketa3(v1)
# [1] 21
```

```
v2=c("biderkatu", 1,2,3,4,5,6)
batuketa3(v2)
# [1] 720
```

Hobetu dezakezu pixka bat funtzioa? Errore mezuak eman ditzala funtzioak honako itxura duten bektoreak parametro moduan jasoz gero:

```
v3=c(1,2,3,4,5,6)
batuketa3(v3)
# Ezin da batu ala biderkatu.
```

```
v4=c("batu")
batuketa3(v4)
# Ezin da batu ala biderkatu.
```

3.3 Edozein luzerako bektore bateko bi balio handienak zein diren eta zein posizioetan dauden itzuliko duena.

```
bi_handienak<-function(v)
{
}
```

Dei egin iezaiozu zure funtzioari parametro moduan honako bektoreak erabiliz eta egiaztatu espero duzun emaitza lortzen duzula.

```
v5<-c(3,8,10,2,4,9)
bi_handienak(v5)
# Handiena: 10 Posizioa: 3 Bigarren handiena: 9 Posizioa: 6
v6<-c(3,8)
bi_handienak(v6)
# Handiena: 8 Posizioa: 2 Bigarren handiena: 3 Posizioa: 1
v7<-c(8,3)
bi_handienak(v7)
# Handiena: 8 Posizioa: 1 Bigarren handiena: 3 Posizioa: 2
v8<-c(8,8)
bi_handienak(v8)
# Handiena: 8 Posizioa: 1 Bigarren handiena: 8 Posizioa: 2
v9<-c(8)
bi_handienak(v9)
# Bektoreak gutxienez 2 osagai izan behar ditu
v10<-c(8,8,0,2,12,8,12,12)
bi_handienak(v10)
# Handiena: 12 Posizioa: 5 Bigarren handiena: 12 Posizioa: 7
```