

# Estadística Inferenziala

## R software librea erabiliz

R eta Rstudio:  
ezaugarri nagusiak

*\*Artxibo honetako irudi guztiak irakasle taldeak prestatutako irudi propioak dira.*

Eneko Arrospide, Gorka Bidegain, Xabier Erdocia, Aitziber Unzueta



# AURKIBIDEA

- 1.1. Sarrera
- 1.2. Leihoak
- 1.3. Oinarrizko ezagutza
- 1.4. Datu-egitura (bektoreak eta data markoak)
- 1.5. Funtzioak
- 1.6. Grafikoak
- 1.7. Paketeak
- 1.8. Lan direktorioa
- 1.9. Fitxategi batetik datuak irakurri
- 1.10. Datu-markoak fitxategi batean gorde



# 1.1. Sarrera





- Konputazio estatistikorako eta datuak grafiko bidez bistartzeko erabiltzen den programazio-lengoaia doakoa eta indartsua da. Windows, MAC OSX eta Linux-ekin bateragarria da.
- R GNU Lizenzia Publiko Orokorraren (GNU General Public License o GPL) baldintzapean eskuragarri dago. Beraz, R libreki eta doan banatu, kopiatu eta erabil daiteke.
- R-ren oinarriko instalazioak hainbat funtzio ditu datuak inportatzeko eta eraldatzeko, modelo estatistikoak ebaluatzeko, datuak irudikatzeko, etab. Ataza berriak egiteko funtzio edo kode berriak (paketeak) behar badira, R-k ematen duen erraztasuna handia da.



- R programarako Garapen Integratuko Ingurunea (IDE) bat da. Rstudiok diseinatutako plataforman aldagaiak aldatu, paketeak kargatu, R-ri buruzko laguntza eskatu edo irudiak bistaratu bezalako ekintzak oso erraz egiten dira.



# 1.2. Leihoak



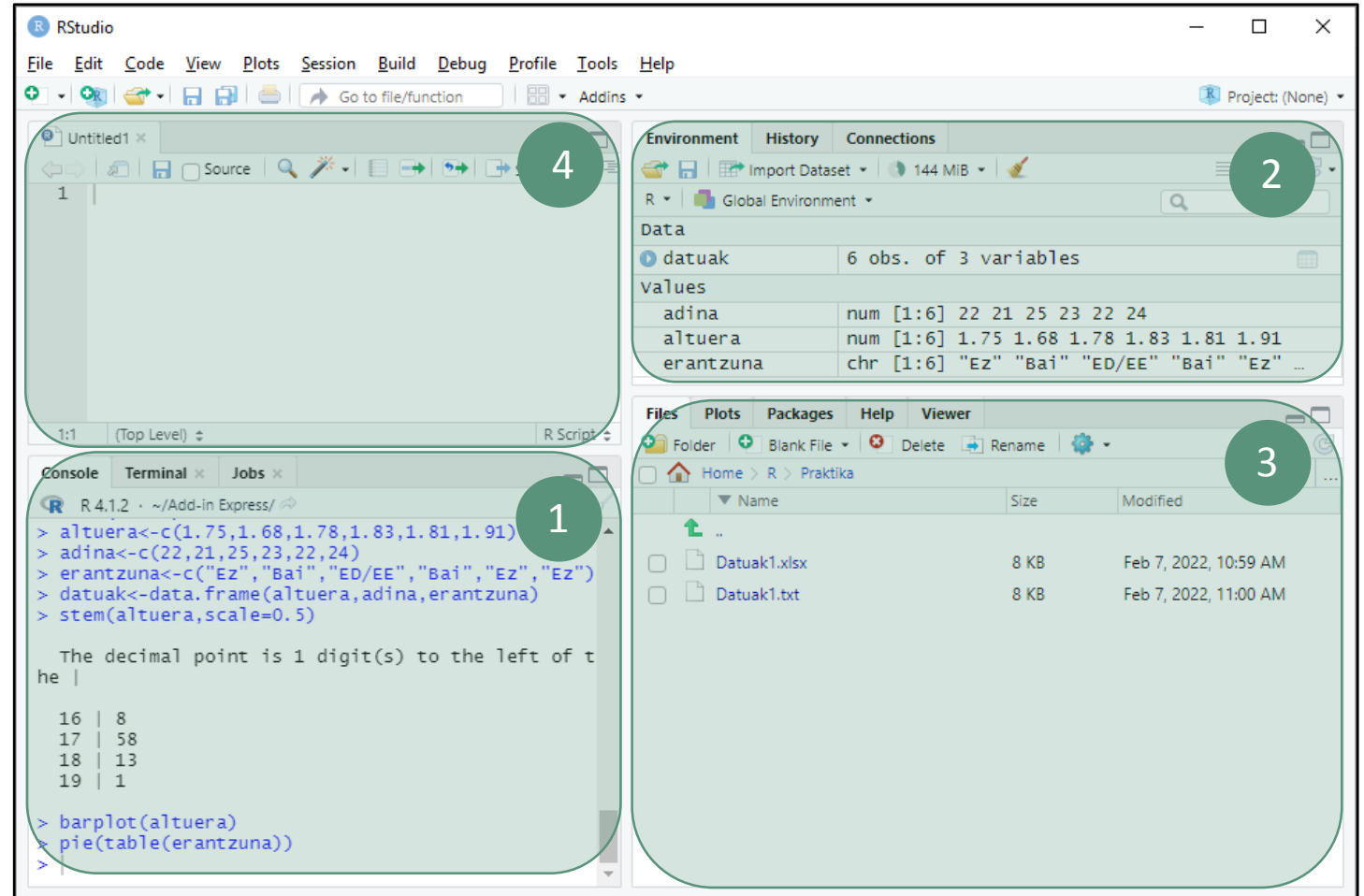
## Leihoak

1 Kontsola

2 Ingurunea + Historia

3 Fitxategiak + Irudiak +  
+ Paketeak + Laguntza

4 Script-en editorea +  
datuen nabigatzailea



The screenshot shows the RStudio interface with the following components highlighted by numbered callouts:

- 1 (Console):** Shows R code execution output:
 

```
> altuera<-c(1.75,1.68,1.78,1.83,1.81,1.91)
> adina<-c(22,21,25,23,22,24)
> erantzuna<-c("Ez","Bai","ED/EE","Bai","Ez","Ez")
> datuak<-data.frame(altuera,adina,erantzuna)
> stem(altuera,scale=0.5)

The decimal point is 1 digit(s) to the left of the |

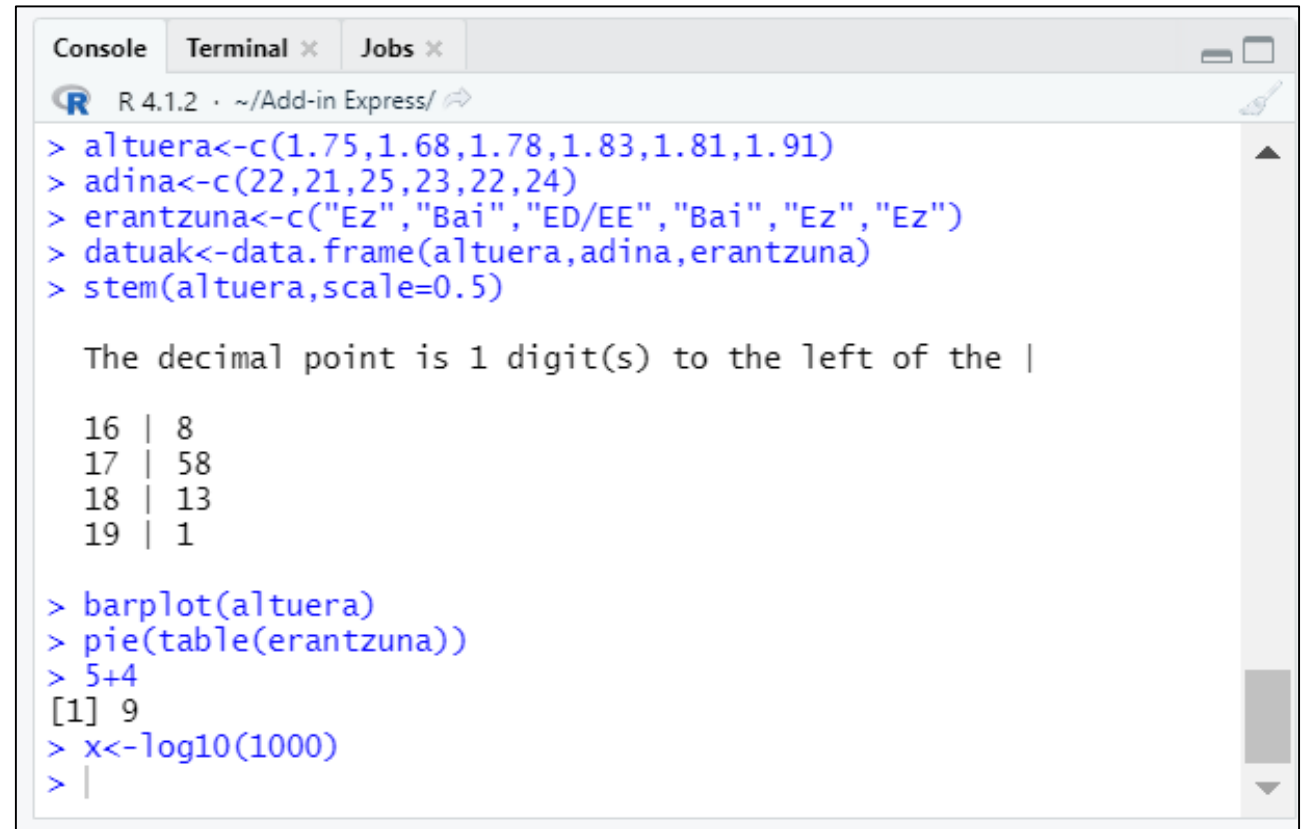
16 | 8
17 | 58
18 | 13
19 | 1

> barplot(altuera)
> pie(table(erantzuna))
>
```
- 2 (Environment/History):** Shows the Global Environment with a data frame named 'datuak' containing 6 observations of 3 variables: 'adina' (numeric), 'altuera' (numeric), and 'erantzuna' (character).
- 3 (Files/Plots/Packages):** Shows the file browser for the 'Praktika' project, listing files like 'Datuak1.xlsx' and 'Datuak1.txt'.
- 4 (Source Editor):** Shows the script editor with a single line of code: '1'.

## Leihoak

### 1 Kontsola

- R bakarrik instalatzen badugu, izango dugun leiho bakarra da.
- Kontsolan ordenagailuan idatzitako aginduak (komandoak) ebaluatzen edo balioztatzen dira.
- Gezia (>) agertzen bada, programa prest dago kodigoa irakurtzeko.
- Rstudio ez badugu instalatzen, guztia (laguntza eskatu, paketeak kargatu, etab.) kontsolan komando bidez egin beharko genuke.



```

R 4.1.2 · ~/Add-in Express/
> altuera<-c(1.75,1.68,1.78,1.83,1.81,1.91)
> adina<-c(22,21,25,23,22,24)
> erantzuna<-c("Ez","Bai","ED/EE","Bai","Ez","Ez")
> datuak<-data.frame(altuera,adina,erantzuna)
> stem(altuera,scale=0.5)

The decimal point is 1 digit(s) to the left of the |

16 | 8
17 | 58
18 | 13
19 | 1

> barplot(altuera)
> pie(table(erantzuna))
> 5+4
[1] 9
> x<-log10(1000)
> |

```

## Leihoak

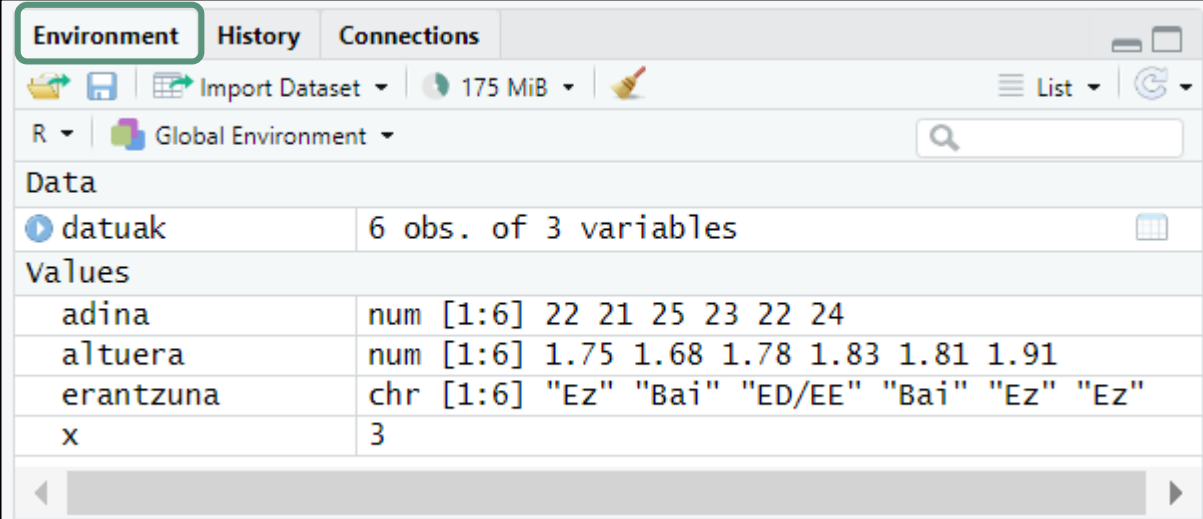
### 2 Ingurunea + Historia

#### Ingurunea:

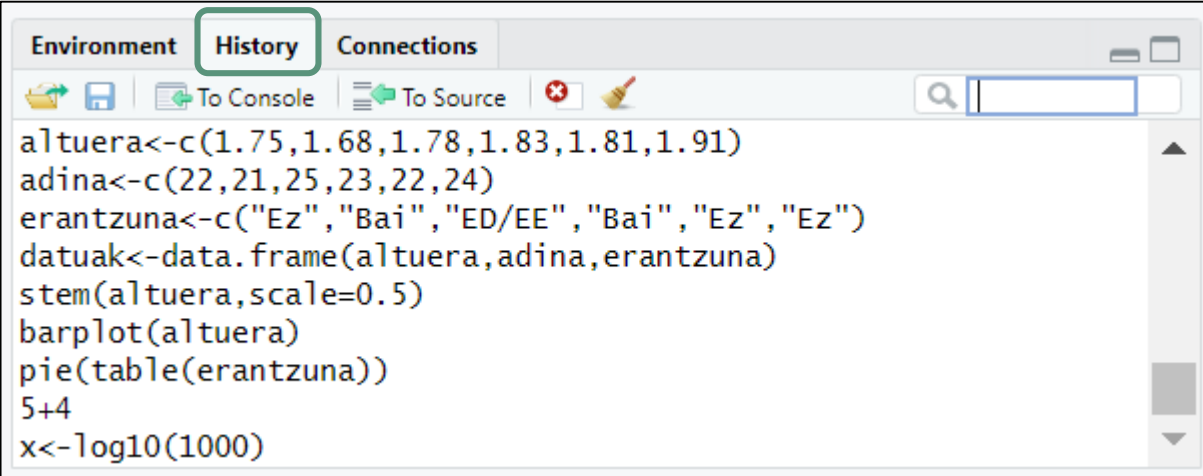
Gordetako aldagaiak, zein aldagai mota den eta beraien balioak agertzen dira.

#### Historia:

Sesioan idatzitako komandoak agertzen dira. Ez dira emaitzak agertzen.



Variable	Type	Values
adina	num [1:6]	22 21 25 23 22 24
altuera	num [1:6]	1.75 1.68 1.78 1.83 1.81 1.91
erantzuna	chr [1:6]	"Ez" "Bai" "ED/EE" "Bai" "Ez" "Ez"
x		3



```

altuera<-c(1.75,1.68,1.78,1.83,1.81,1.91)
adina<-c(22,21,25,23,22,24)
erantzuna<-c("Ez","Bai","ED/EE","Bai","Ez","Ez")
datuak<-data.frame(altuera,adina,erantzuna)
stem(altuera,scale=0.5)
barplot(altuera)
pie(table(erantzuna))
5+4
x<-log10(1000)

```



Leihoak

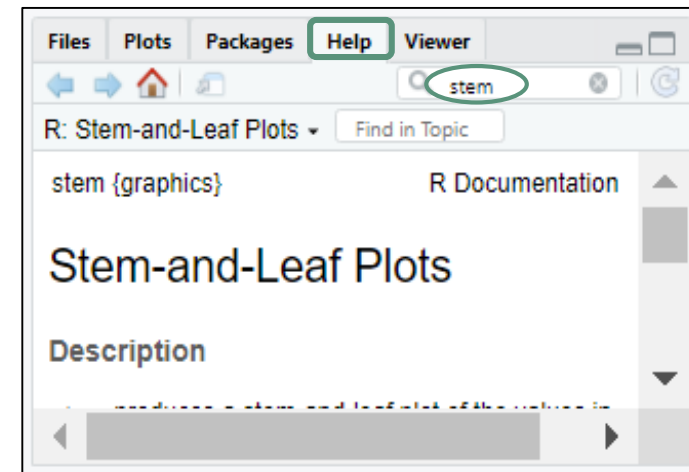
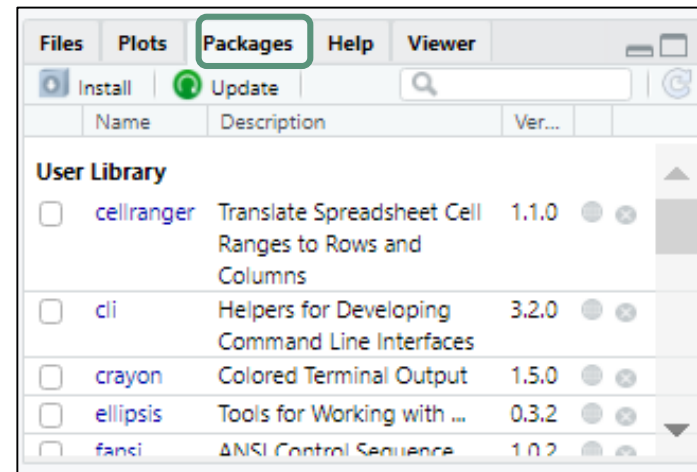
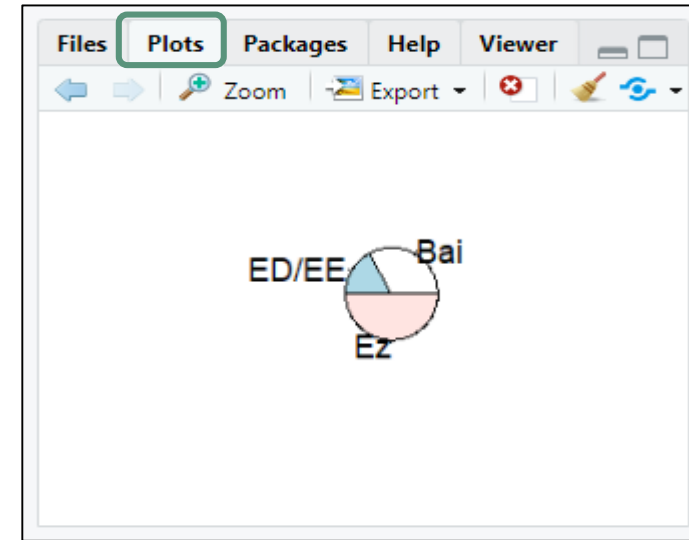
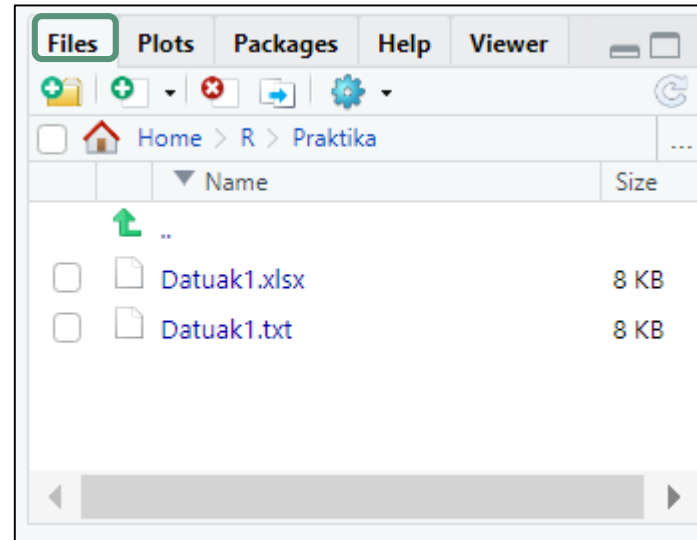
3 Fitxategiak + Irudiak +  
+ Paketeak + Laguntza

**Fitxategiak:**  
Lan direktorioan ditugun fitxategiak.

**Irudiak:**  
Grafikatu ditugun irudi guztiak agertzen dira. Ezker-eskuin geziekin irudi guztiak ditugu eskuragarri. Irudiak handitu (zoom) edo esportatu daitezke.

**Paketeak:**  
Instalatutako paketeak agertzen dira.

**Laguntza**



## Leihoak

### 4 Script-en editorea + datuen nabigatzailea

#### Script-en editorea:

Lehio honetan idatziko ditugu exekutatu nahi ditugun aginduen zerrenda. Agindu hauek .R fitxategi batean gorde daitezke.

#### Datuen nabigatzailea:

Ingurune fitxan definitutako datu-markoetan klikatuz, bere barnean dauden datuak agertzen dira.

```

1
2 altuera<-c(1.75,1.68,1.78,1.83,1.81,1.91)
3 adina<-c(22,21,25,23,22,24)
4 erantzuna<-c("Ez","Bai","ED/EE","Bai","Ez","Ez")
5 datuak<-data.frame(alteuera,adina,erantzuna)
6 stem(alteuera,scale=0.5)
7
8 barplot(alteuera)
9 pie(table(erantzuna))
10 5+4
11 x<-log10(1000)
12 x
13
11:1 (Top Level) R Script

```

altuera	adina	erantzuna
1.75	22	Ez
1.68	21	Bai
1.78	25	ED/EE
1.83	23	Bai
1.81	22	Ez
1.91	24	Ez

Variable	Type	Values
adina	num [1:6]	22 21 25 23 22 24
altuera	num [1:6]	1.75 1.68 1.78 1.83 1.81 1.91
erantzuna	chr [1:6]	"Ez" "Bai" "ED/EE" "Bai" "Ez" "Ez"
x	num	3

# 1.3. Oinarrizko ezagutza



## Eragiketa aritmetikoak: +, -, \*, /, ^, %/%, %%

- Eragiketen lehenengo parentesia barrukoak, gero berreketa eta erroketak, ondoren biderketak eta zatiketak. Amaitzeko, batuketa eta kenketak. Beti ezkerretik eskumara ebaluatzen da.

Eragiketa	Esanahia
+	Batuketa
-	Kenketa
*	Biderketa
/	Zatiketa
^	Berreketa
%/%	Zatiketaren zatidura
%%	Zatiketaren hondarra

```

> a<- 10 # a aldagaiari 10 balioa esleituko dio
> b<- 3 # b aldagaiari 3 balioa esleituko dio
> a+b
[1] 13
> a^b
[1] 1000
> a/b
[1] 3.333333
> a%/%b # zatiketaren zatidura kalkulatu du
[1] 3

```

## Aldagai motak

- Aldagaiaren izenak hizkiz, zenbakiz, puntuz edo azpimarrez osatuta egon daitezke. Hizki edo puntu batekin has daitezke, baina puntu batekin hasten badira ezin da zenbaki batekin jarraitu.

Aldagai mota	Esanahia
<b>Numeric</b>	Zenbaki errealak
<b>Integer</b>	Zenbaki osoak
<b>Complex</b>	Zenbaki konplexuak
<b>Character</b>	Karakterek
<b>Logical</b>	Aldagai logikoak: TRUE edo FALSE

```
> x1 <- c(1:20); class(x1)
[1] "integer"
> x2<- class(c(1.1, 2.3)) ; class(x2)
[1] "numeric"
> x3<- c("Arkaitz", "Ziortza", "Unai", "Amaia", "Eider")
> class(x3)
[1] "character"
> x4<- c(FALSE, TRUE, FALSE, TRUE) ; class(x4)
[1] "logical"
> x5 <- c(3, "Jon", 8, 3, 23, 0, 1) ; class(x5)
[1] "character"
> x5 # balio bat karakterea da, guztiak karaktere
bihurtzen dira.
[1] "3" "Jon" "8" "3" "23" "0" "1"
```

## Eragiketa logikoak: > , >= , < , <= , == , & , |

Eragiketa	Esanahia
>	handiago
>=	handiago edo berdin
<	txikiago
<=	txikiago edo berdin
==	berdin
!=	desberdin
&	eta
	edo

```
> a<- 10; b<- 5; c<- 2 # agindu anitz lerro bakarrean ;
erabiliz
```

```
> a==b
[1] FALSE
```

```
> a==5 | b<a
[1] TRUE
```

```
> c != 3
[1] TRUE
```

```
> a!=b & b!=c
[1] TRUE
```

## Funtzio aritmetikoak

Funtzioa	Esanahia
<code>log2(x)</code>	x-ren logaritmo bitarra
<code>log10(x)</code>	x-ren logaritmo hamartarra
<code>log(x,b)</code>	x-ren logaritmoa b oinarrian
<code>exp(x)</code>	x-ren esponentziala
<code>abs(x)</code>	x-ren balio absolutua
<code>sqrt(x)</code>	x-ren erro karratua
<code>cos(x)</code>	x-ren kosinua (x radianetan)
<code>sin(x)</code>	x-ren sinua (x radianetan)
<code>tan(x)</code>	x-ren tangentea (x radianetan)

```

> sin(pi)      # funtzio trigonometrikoak
[1] 1.224606e-16
> log(3)       # logaritmo nepertarra
[1] 1.098612
> log(100, base=10)
[1] 2
> sqrt(4)      # erro karratua
[1] 2
> pow(6, 3)
[1] 216
> abs(-20)     # balio absolutua
[1] 20
> cos(0)
[1] 1

```



## Oinarrizko aginduak

Agindua	Esanahia
<code>getwd()</code>	Lan direktorioa erakutsiko du
<code>setwd(“bidea/direktorioa”)</code>	Lan-direktorioa zehaztuko du
<code>ls()</code>	Lan-direktorioan ditugun objektuak zerrendatuko ditu
<code>dir()</code>	Lan-direktorioko karpeta eta fitxategiak erakutsiko ditu
<code>rm(objetua)</code>	lan-direktorioko <i>objektua</i> ezabatuko du
<code>history()</code>	Exekutatuak aginduak erakutsiko ditu
<code>help(objektua)</code>	<i>objektua</i> -ren laguntza emango du
<code>?objektua</code>	<i>objektua</i> -ren laguntza emango du
<code>install.packages(“objektua”)</code>	<i>objektua</i> paketea instalatuko du
<code>library(“objektua”)</code>	<i>objektua</i> paketea memorian kargatuko du



## Estatistika deskribatzaileko funtzioak

Funtzioa	Esanahia
<code>mean(x)</code>	x-ko elementuen batzbestekoa
<code>median(x)</code>	x-ko elementuen mediana
<code>quantile(x,prob)</code>	<i>prob</i> probabilitateari dagokion x-ko kuantila
<code>var(x)</code>	x-ko elementuen kuasibariantza
<code>sd(x)</code>	x-ko elementuen kuasidesbiderazio tipikoa
<code>cov(x, y)</code>	x eta y-ren arteko kobariantza
<code>summary(x)</code>	x-ko elementuen laburpena emango du (minimoa, 1. kuartila, mediana, batzbestekoa, 3. kuartila eta maximoa)
<code>boxplot.stats(x)\$out</code>	x-ko elementuen balio arraroak

## Estatistika deskribatzaileko funtzioak

Funtzioa	Esanahia
<b>max(x)</b>	x-ko elementuetatik maximoa
<b>min(x)</b>	x-ko elementuetatik minimoa
<b>range(x)</b>	x-ko elementu minimoa eta maximoa
<b>sum(x)</b>	x-ko elementu guztien batura
<b>prod(x)</b>	x-ko elementu guztien biderkadura
<b>diff(x)</b>	x-ko elementuen arteko diferentziak
<b>cumsum(x)</b>	x-ko elementuen batura metatua
<b>sort(x)</b>	x-ko elementuak ordena gorakorrean ordenatuko ditu
<b>table(x)</b>	x-ko elementuen balio desberdinak eta beraien maiztasun absolutua dituen taula itzuliko du
<b>round(x)</b>	x-ko elementuak borobilduko ditu

# 1.4. Datu-egitura (bektoreak eta datu-markoak)



## Bektoreak

- Dimentsio bakarra eta datu mota bera dituen datu-egitura. Bektoreak sortzeko aukera anitz daude:

## Bektoreak sortzea

- **c()**: bere osagaiak kateatu egiten ditu, bektore bat sortuz

```
> x1 <- c(1:20) ; x1
```

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

- **scan()**: teklatuz sartutako objektuak kateatuko ditu, bektore bat sortuz

```
> di ferentzi <- scan()
```

```
1: 1.2
```

```
2: 0.6
```

```
3: 2.4
```

```
4: 1.1
```

```
5: 0.9
```

```
6:
```

```
Read 5 items
```



- **seq()**: elementu-segida bat sortzen du. Funtzio hau bi eratara erabil daiteke: urratsaren tamaina ezarriz *by* parametroarekin edo bektorearen luzera zehaztuz *length* parametroarekin

```
> seq(1, 15, by=2) # edo seq(1, 15, 2). Segidaren lehen elementua 1 izango da, azkena 15 eta urratsa 2
```

```
[1] 1 3 5 7 9 11 13 15
```

```
> seq(1, 15, 7)
```

```
[1] 1 8 15
```

```
> seq(1, 15, length=5) # segidaren lehen elementua 1 izango da, azkena 15 eta luzera 5
```

```
[1] 1.0 4.5 8.0 11.5 15.0
```

- **rep()**: patroia bat errepikatuz sortuko du bektorea

```
> rep(c("bai", "ez", "agi an"), 2) # "bai" "ez" "agi an" patroia birritan errepikatu
```

```
[1] "bai" "ez" "agi an" "bai" "ez" "agi an"
```

```
> rep(c(1, 2, 3), c(1, 2, 3))
```

```
[1] 1 2 2 3 3 3
```

- **numeric()**: zeroz osatutako bektorea sortuko du

```
> numeric(10)
```

```
[1] 0 0 0 0 0 0 0 0 0 0
```

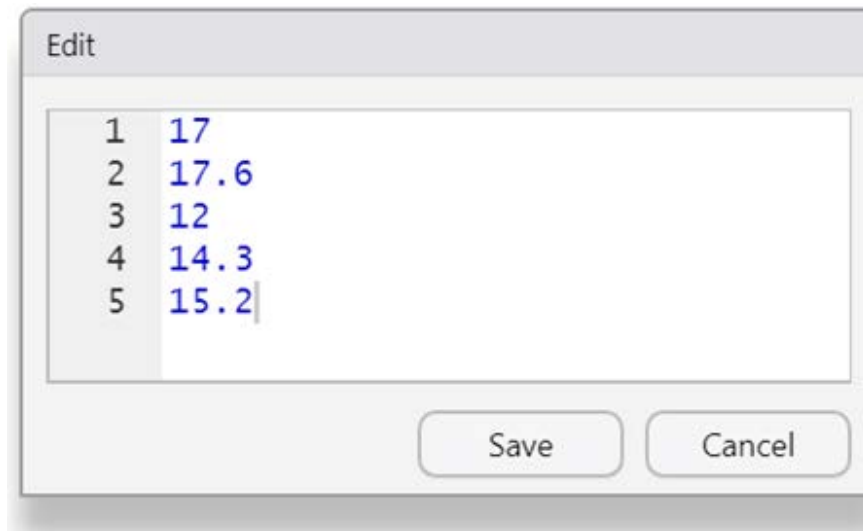


- **: operadorea:**

```
> 2:6 # eragile honek 2-tik 6-rako elementuekin bektore bat sortzen du eta z-ri esleitzen dio  
[1] 2 3 4 5 6
```

- **edit():** datu-bektore bat editatzen du

```
> datuak<- edit() # zabal tzen den lei hoan, ENTER erabiliz, bektorearen elementuak sar dai tezke
```



**Bektore baten elementuak erauzi:** bektoreak indexatuta daude. Bektore baten elementu bat lortzeko, elementuaren posizioa adierazten duen zenbakia kako zuzenen artean idatziko da.

- x bektorea sortu,

```
> x<-seq(11, 101, 10) # 11tik 101ra eta urratsa=10  
[1] 11 21 31 41 51 61 71 81 91 101
```

- x bektorearen 8. elementua erauzi

```
> x[8]  
[1] 81
```

- x bektorearen lehenengo bost elementuak erauzi

```
> x[1:5]  
[1] 11 21 31 41 51
```

- x bektorearen 4. 5. eta 8. elementuak erauzi

```
> x[c(4, 5, 8)]  
[1] 41 51 81
```

**Bektore baten elementuak ezabatu:** kako zuzenen artean zenbaki negatiboak sartzen baditugu posizio horietan dauden elementuak ezabatuko dira

- x bektorearen 5. elementua ezabatu

```
> x[-5]
[1] 11 21 31 41 61 71 81 91 101
```

- x bektorearen 4. 5. eta 8. elementuak ezabatu

```
> x[-c(4, 5, 8)] # edo x[c(-4, -5, -8)]
[1] 11 21 31 61 71 91 101
```

**Bektore baten elementuak gehitu edo aldatu**

- x bektorearen 1. elementua aldatu

```
> x[1] <- 11 # kasu honetan balio bera emango diogu
[1] 11 21 31 41 51 61 71 81 91 101
```

- x bektorean 11. elementua gehitu 111 balioarekin

```
> x[11] <- 111
[1] 11 21 31 41 51 61 71 81 91 101 111
```





## Bektore baten elementuen erauzketa baldintzatua

- x-ren elementuak 25 baino txikiago edo 50 baino handiago dira?

```
> x<25 | x>50
```

```
[1] TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

- x-ren elementuak 50 baino handiagoak dira?

```
> x>50
```

```
[1] FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

- 50 baino handiagoak diren x-ren elementuen posizioak

```
> which(x>50)
```

```
[1] 5 6 7 8 9 10 11
```

- 25 baino txikiago eta 50 baino handiago diren x-ren elementuen posizioak

```
> which(x<25 | x>50)
```

```
[1] 1 2 5 6 7 8 9 10 11
```

## Bektore baten elementuen erauzketa baldintzatua

- 71 balioa duten x-ren elementuen posizioak  

```
> which(x==71)
[1] 7
```
- x bektoretik 71 balioa duen elementua ezabatu  

```
> x[-which(x==71)]
[1] 11 21 31 41 51 61 81 91 101 111
```
- x bektoretik 71 balioa duen elementua ezabatu  

```
> x[x!=71]
[1] 11 21 31 41 51 61 81 91 101 111
```
- x bektoretik 71 balioa duen elementua ezabatu  

```
> subset(x, x!=71) # Definitutako baldintza batzuen arabera bektore baten azpimultzo
# bat itzuliko du
[1] 11 21 31 41 51 61 81 91 101 111
```

## Bektore baten elementuen erauzketa baldintzatua

- 50 baino handiagoak diren x-ren elementuen balioak  
`> x[x>50] # edo subset(x, x>50)`  
[1] 51 61 71 81 91 101
- 50 baino handiagoak diren x-ren elementu kopurua  
`> length(x[x>50])`  
[1] 7
- 30 baino txikiago edo 70 baino handiago diren x-ren elementuen balioak  
`> x[x<30 | x>70] # edo subset(x, x<30 | x>70)`  
[1] 11 21 71 81 91 101 111
- 22 baino handiago eta 70 baino txikiago diren x-ren elementuen balioak  
`> x[x>22 & x<70] # edo subset(x, x>22 & x<70)`  
[1] 31 41 51 61
- 22 baino handiago eta 70 baino txikiago diren x-ren elementuen balioak  
`> length(x[x>22 & x<70])`  
[1] 4



## Bektore baten zehaztu gabeko elementuen kudeaketa

- Balio galduak dituen bektorea sortu

```
> x<- c(11, 21, NA, 41, NA, 61, 71, 81, NA, 101)
> x
[1] 11 21 NA 41 NA 61 71 81 NA 101
```

- Bektore bateko balio galduak zehaztu

```
> is.na(x)
[1] FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE TRUE FALSE
```

- Bektore bateko balio galduen posizioak

```
> which(is.na(x))
[1] 3 5 9
```

- Balio galduak bektore batetik ezabatu

```
> x[!is.na(x)] # edo x[-which(is.na(x))]
[1] 11 21 41 61 71 81 101
```

## Bektore baten zehaztu gabeko elementuen kudeaketa

- Balio galduak dituen bektorearen elementuen batura  

```
> sum(x, na.rm = TRUE) # edo sum(x[!is.na(x)]) edo sum(na.omit(x))
[1] 387
```
- Balio galduak dituen bektorearen elementuen biderkadura  

```
> prod(x, na.rm = TRUE) # edo prod(x[!is.na(x)]) edo prod(na.omit(x))
[1] 335575629081
```
- Balio galduak dituen bektorearen elementuen batazbestekoa  

```
> mean(x, na.rm = TRUE) # edo mean(x[!is.na(x)]) edo mean(na.omit(x))
[1] 387
```
- Balio galduak dituen bektorearen elementuen batura  

```
> cumsum(na.omit(x)) # edo cumsum(x[!is.na(x)])
[1] 11 32 73 134 205 286 387
# cumsum() funtzioak ez du onartzen na.rm
```



## Bektore bateko balio arraroak

- **boxplot.stats()**: kutxa-diagramari buruzko informazioa bueltatzen du.

```
> boxplot.stats(c(17, 23, 21, 36, 21, 18, 19, 22, 35, 20, 25))
```

```
$stats
```

```
[1] 17.0 19.5 21.0 24.0 25.0
```

```
$n
```

```
[1] 11
```

```
$conf
```

```
[1] 18.85625 23.14375
```

```
$out
```

```
[1] 36 35
```

- **\$stats**: bost datu ematen ditu: beheko bibotearen muturra, lehen kuartila, mediana, hirugarren kuartila eta goiko bibotearen muturra.
- **\$n**: NA ez diren elementu kopurua.
- **\$conf**: medianaren konfidantza tartea (0.95 konfidantza maila).
- **\$out**: bektorearen balio arraroak.



## Bektore baten elementuen balio arraroak

- `boxplot.stats()`\$out: balio arraroak bueltatzen ditu.  

```
> x<- c(17, 23, 21, 36, 21, 18, 19, 22, 35, 20, 25)  
> boxplot.stats(x)$out  
[1] 36 35
```
- Balio arraroen posizioak bektorean  

```
> which(x %in% boxplot.stats(x)$out)  
[1] 4 9
```
- Balio arraroak ezabatu  

```
> x[-which(x %in% boxplot.stats(x)$out)]  
[1] 17 23 21 21 18 19 22 20 25
```

## Datu-markoak

- Datu-markoak (data frame-ak) datuak gordetzeko objekturik erabilienera da
- Datu-marko baten errenkada bakoitza laginaren norbanakotzat edo behaketatzat har daiteke eta, zutabe bakoitza norbanakoaren ezaugarri, atributu edo aldagai bat bezala
- Datu-marko baten zutabe bakoitza mota ezberdinekoa (zenbakia, karakterea, ...) izan daitezke. Baina zutabe bereko elementu guztiak mota berekoak izan behar dira.

## Datu-markoa sortzea

- **data.frame()**: luzera berdineko bektoreekin datu-markoa sortzen du
- Honako adibidean, datu marku bat sortuko da 9 ikasleen adinarekin eta aljebra eta kalkuluko ikasgaietan izandako kalifikazioekin:

```
> adi na <- c(18, 18, 19, 18, 20, 19, 18, 18, 18)
> aljebra <- c(4.2, 6.3, 5.8, 6.8, 7.3, 8.1, 3.5, 7.7, 6.5)
> kalkulua <- c(6.2, 3.3, 6.7, 5.2, 4.1, 7.8, 5.9, 7.1, 5.6)
> datuak <- data.frame(adi na, aljebra, kalkulua)
```





```
> datuak
```

	adi na	al j ebra	kal kul ua
1	18	4.2	6.2
2	18	6.3	3.3
3	19	5.8	6.7
4	18	6.8	5.2
5	20	7.3	4.1
6	19	8.1	7.8
7	18	3.5	5.9
8	18	7.7	7.1
9	18	6.5	5.6

- `rbind()`: norbanako (errenkada) berria gehitu du

```
> datuak<- rbind(datuak, c(18, 7.2, 7.4))
```

```
> datuak
```

	adi na	al j ebra	kal kul ua
1	18	4.2	6.2
...	...	...	...
9	18	6.5	5.6
10	18	7.2	7.4



- **cbind()**: aldagai (zutabe) berria gehitu du

```
> gradua <- c("el ekt roni ka", "el ektri ka", "mekani ka", "mekani ka", "el ekt roni ka",
"mekani ka", "mekani ka", "el ektri ka", "mekani ka", "mekani ka")
> datuak <- cbind(datuak, gradua)
> datuak
```

	adi na	al j ebra	kal kul ua	gradua
1	18	4. 2	6. 2	el ekt roni ka
2	18	6. 3	3. 3	el ektri ka
3	19	5. 8	6. 7	mekani ka
4	18	6. 8	5. 2	mekani ka
5	20	7. 3	4. 1	el ekt roni ka
6	19	8. 1	7. 8	mekani ka
7	18	3. 5	5. 9	mekani ka
8	18	7. 7	7. 1	el ektri ka
9	18	6. 5	5. 6	mekani ka
10	18	7. 2	7. 4	mekani ka

- **data()**: datu-markoa kargatuko du. Datu-marko batzuk erabiltzeko **data(datamarkoaren.izena)** kodearekin kargatu behar dira. Begiratu laguntzan IceSkating datu-markoaren erabilera:  
help(IceSkating)



- **str()**: datu-markoaren egitura bistaratzeko du

```
> str(datuak)
```

```
'data.frame':      10 obs. of  4 variables:
 $ adi na      : num  18 18 19 18 20 19 18 18 18 1
 $ aljebra     : num  4.2 6.3 5.8 6.8 7.3 8.1 3.5 7.7 6.5 7.2
 $ kalkulua    : num  6.2 3.3 6.7 5.2 4.1 7.8 5.9 7.1 5.6 7.4
 $ gradua      : chr   "el ektroni ka" "el ektri ka" "mekani ka" "mekani ka"
```

- **summary()**: datu-markoaren elementuen laburpena ematen du. Aldagai kuantitatiboak badira minimoa, 1. kuartila, mediana, batezbestekoa, 3. kuartila eta maximoa bueltatzen ditu.

```
> summary(datuak)
```

adi na	aljebra	kalkulua	gradua
Min. : 18.00	Min. : 3.500	Min. : 3.30	Length: 10
1st Qu.: 18.00	1st Qu.: 5.925	1st Qu.: 5.30	Class : character
Median : 18.00	Median : 6.650	Median : 6.05	Mode : character
Mean : 18.40	Mean : 6.340	Mean : 5.93	
3rd Qu.: 18.75	3rd Qu.: 7.275	3rd Qu.: 7.00	
Max. : 20.00	Max. : 8.100	Max. : 7.80	

- **head()**: datu-marcoaren lehenengo behaketak bistaritzen ditu

```
> head(datuak, n=6) # defektuz, n=6 (lehenengo 6 behaketak bistaritzen ditu)
```

	adi na	al j ebra	kal kul ua	gradua
1	18	4. 2	6. 2	el ekt roni ka
2	18	6. 3	3. 3	el ektri ka
3	19	5. 8	6. 7	mekani ka
4	18	6. 8	5. 2	mekani ka
5	20	7. 3	4. 1	el ekt roni ka
6	19	8. 1	7. 8	mekani ka

- **tail()**: datu-marcoaren azken behaketak bistaritzen ditu.

```
> tail(datuak, n=6) # defektuz, n=6 (azken 6 behaketak bistaritzen ditu)
```

	adi na	al j ebra	kal kul ua	gradua
5	20	7. 3	4. 1	el ekt roni ka
6	19	8. 1	7. 8	mekani ka
7	18	3. 5	5. 9	mekani ka
8	18	7. 7	7. 1	el ektri ka
9	18	6. 5	5. 6	mekani ka
10	18	7. 2	7. 4	mekani ka

- **names()**: datu-marcoaren aldagaien izenak bistaratu

```
> names(datuak)
```

```
[1] "adi na"      "al j ebra"   "kal kul ua" "gradua"
```

- **rownames()**: datu-marcoaren errenkaden identifikatzaileak bistaratu

```
> rownames(datuak)
```

```
[1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10"
```

- **dim()**: errenkada eta zutabe kopurua dituen bektorea sortzen du

```
> dim(datuak)
```

```
[1] 10  4
```

- **dimnames()**: errenkaden identifikatzaileak eta zutabeen izenak dituen zerrenda bat ematen du

```
> dimnames(datuak)
```

```
[[1]]
```

```
[1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10"
```

```
[[2]]
```

```
[1] "adi na"      "al j ebra"   "kal kul ua" "gradua"
```

## Datu-marko bateko osagaiak (zutabeak, errenkada, elementuak) erauzi:

- **\$**: **datumarkoaren\_izena\$zutabearen\_izena**, datu-markoaren zutabe bat lortuko da

```
> datuak$adina
[1] 18 19 18 20 19 18 18 18 18
> datuak$aljebra
[1] 4.2 6.3 5.8 6.8 7.3 8.1 3.5 7.7 6.5 7.2
> datuak$skalulua
[1] 6.2 3.3 6.7 5.2 4.1 7.8 5.9 7.1 5.6 7.4
```
- **attach()**: datu-markoaren aldagaiak (zutabeak) aldagai globalak bihurtzen dira, eta aldagaiaren izena erabiltzea nahikoa da bere elementuak lortzeko

```
> attach(datuak)
> adina
[1] 18 19 18 20 19 18 18 18 18
> aljebra
[1] 4.2 6.3 5.8 6.8 7.3 8.1 3.5 7.7 6.5 7.2
```
- **detach()**: datu-markoaren aldagaiek ingurune globaletik ezabatzen ditu

```
> detach(datuak)
```



Datu-marko bateko osagaiak (zutabeak, errenkada, elementuak) erauzi:

- Datu-markoaren 2. zutabea erauzi

```
> datuak$algebra
```

```
[1] 4.2 6.3 5.8 6.8 7.3 8.1 3.5 7.7 6.5 7.2
```

```
> datuak[2] # 2. zutabeko norbanako guztiak bueltatzen ditu
```

```
[1] 4.2 6.3 5.8 6.8 7.3 8.1 3.5 7.7 6.5 7.2
```

```
> datuak[, 2] # 2. zutabeko norbanako guztiak bueltatzen ditu
```

```
[1] 4.2 6.3 5.8 6.8 7.3 8.1 3.5 7.7 6.5 7.2
```

```
> datuak[, "algebra"] # "algebra" zutabeko norbanako guztiak bueltatzen ditu
```

```
[1] 4.2 6.3 5.8 6.8 7.3 8.1 3.5 7.7 6.5 7.2
```

- Datu-markoaren 2. eta 3. zutabeak erauzi

```
> datuak[, c(2, 3)]
```

```
algebra kalkulua
```

```
1      4.2      6.2
```

```
2      6.3      3.3
```

```
...
```

```
9      6.5      5.6
```

```
10     7.2      7.4
```





## Datu-marko bateko osagaiak (zutabeak, errenkada, elementuak) erauzi:

- Datu-markoaren 2. errenkada erauzi

```
> datuak[2, ]
```

```
adi na aljebra kalkulua        gradua
2      18      6.3      3.3 el ektri ka
```

- Datu-markoaren lehenengo 3 errenkadak erauzi

```
> datuak[1:3, ]
```

```
adi na aljebra kalkulua        gradua
1      18      4.2      6.2 el ektroni ka
2      18      6.3      3.3 el ektri ka
3      19      5.8      6.7 mekani ka
```

- Datu-markoaren lehenengo 8 errenkadak ezabatu

```
> datuak[-(1:8), ] # lehenengo zortzi errenkada (norbanako) ezabatu
```

```
adi na aljebra kalkulua        gradua
9      18      6.5      5.6 mekani ka
10     18      7.2      7.4 mekani ka
```





## Datu-marko bateko osagaiak (zutabeak, errenkada, elementuak) erauzi:

- Datu-markoaren 3. ikaslearen aljebra ko kalifikazioa erauzi

```
> datuak[3, 2] # edo datuak$algebra[3]
[1] 5.8
```

- Datu-markoaren 19 urte edo gehiago dituzten ikasleen zutabe guztiak erauzi

```
> datuak[datuak$adina >= 19, ]
```

	adina	algebra	kalkulua	gradua
3	19	5.8	6.7	mekanika
5	20	7.3	4.1	elektronika
6	19	8.1	7.8	mekanika

- Datu-markoaren 18 urte baino gehiago dituzten ikasleen kalkuluko kalifikazioa erauzi

```
> datuak$kalculua[datuak$adina > 18]
```

```
[1] 6.7 4.1 7.8
```

- Datu-markoaren elektriko graduko ikasleen adina

```
> datuak$adina[datuak$gradua == "elektronika"]
```

```
[1] 18 18
```



## Datu-marko bateko osagaiak (zutabeak, errenkada, elementuak) erauzi:

- Datu-markoaren 19 urte edo gehiago dituzten ikasleen posizioa  

```
> whi ch(datuak$adi na>=19)
[1] 3 5 6
```
- Datu-markoaren 19 urte edo gehiago dituzten ikasle kopurua  

```
> length(whi ch(datuak$adi na>=19))
[1] 3
```
- Datu-markoaren lehengo bi zutabeen batzbestekoa  

```
> apply(datuak[, 1:2], MARGIN=2, FUN=mean) # MARGIN: 1: errenkadak, 2: zutabeak
adi na alj ebra
18.40      6.34
```

## Datu-marko bateko osagaiak (zutabeak, errenkada, elementuak) erauzi:

- Datu-markoaren 18 urte baino gehiago dituzten ikasleen artean mekanikako graduko informazioa

```
> datuak[datuak$adi na>18 & datuak$gradua=="mekani ka", ]
```

	adi na	al j ebra	kal kul ua	gradua
3	19	5.8	6.7	mekani ka
6	19	8.1	7.8	mekani ka

- Datu-markoaren 18 urte baino gehiago edo elektronikako gradukoak diren ikasleen informazioa

```
> datuak[datuak$adi na>18 | datuak$gradua=="el ekt roni ka", ]
```

	adi na	al j ebra	kal kul ua	gradua
1	18	4.2	6.2	el ekt roni ka
3	19	5.8	6.7	mekani ka
5	20	7.3	4.1	el ekt roni ka
6	19	8.1	7.8	mekani ka

- Datu-markoaren 18 urte baino gehiago edo elektronikako gradukoak diren ikasle kopurua

```
> l e n g t h ( w h i c h ( d a t u a k $ a d i n a > 1 8 | d a t u a k $ g r a d u a == " e l e k t r o n i k a " ) )
```

```
[ 1 ] 4
```

# 1.5. Grafikoak



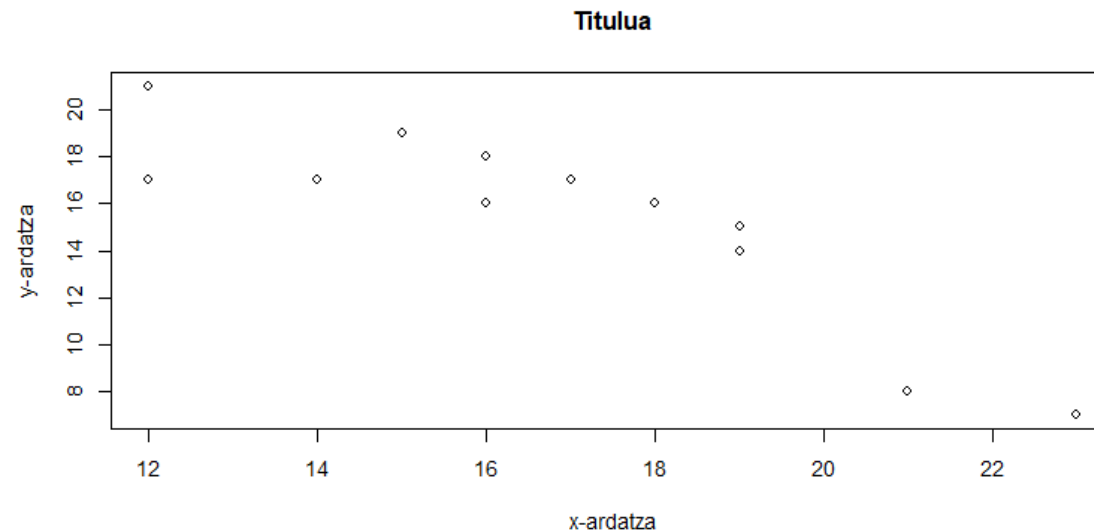
- plot()

```
plot(x, y=NULL, type="p", xlim=NULL, ylim=NULL, log="", main=NULL, sub=NULL,  
     xlab=NULL, ylab=NULL, axes=TRUE, ...)
```

```
> x <- c(15, 18, 17, 16, 12, 12, 14, 19, 21, 23, 19, 16)
```

```
> y <- c(19, 16, 17, 18, 21, 17, 17, 14, 8, 7, 15, 16)
```

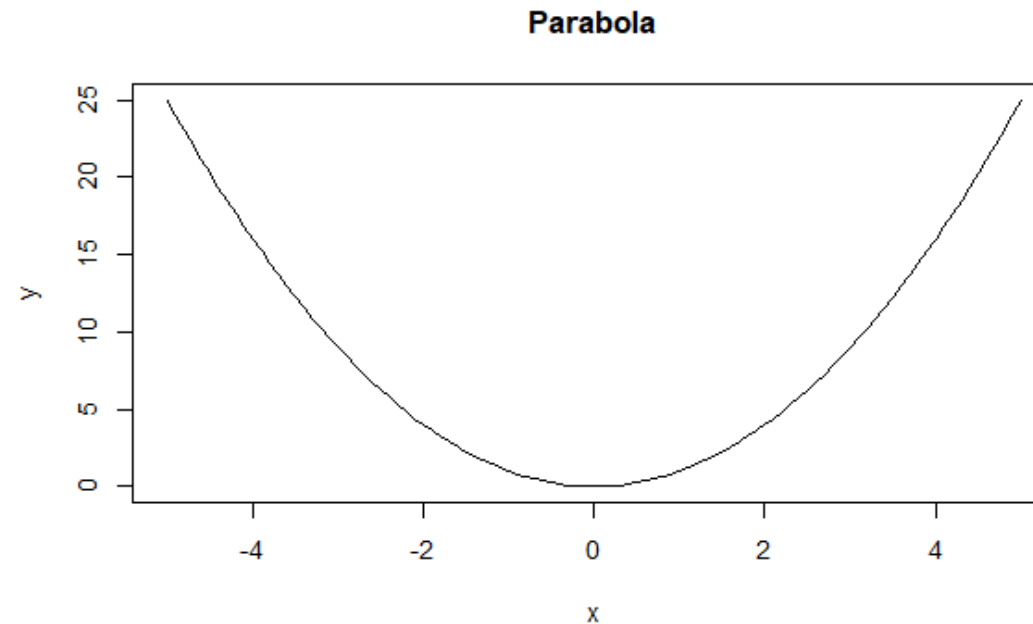
```
> plot(x, y, main="Titulua", xlab="x-ardatza", ylab="y-ardatza")
```



- curve()

```
curve(expr, from=NULL, to=NULL, n=101, add=FALSE, type="l", xname="x", xlab=xname,
      ylab=NULL, log=NULL, xlim=NULL, ...)
```

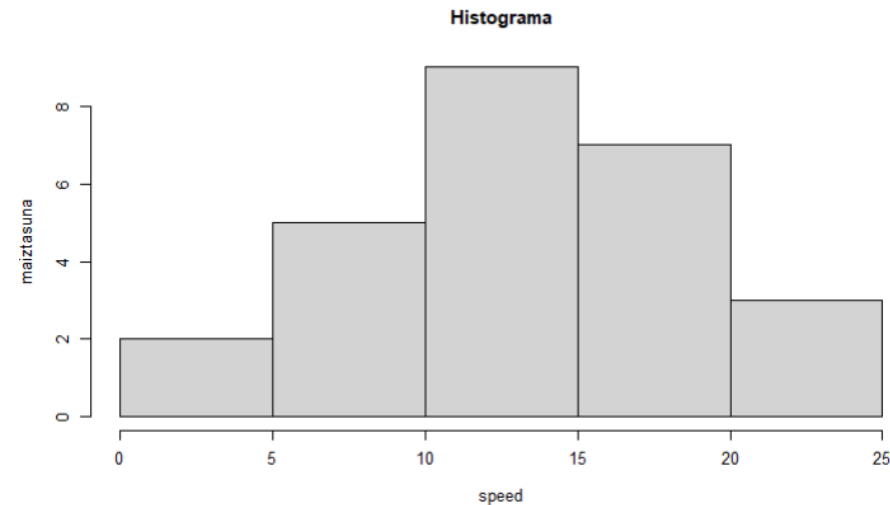
```
> curve(x^2, from=-5, to=5, main="Parabola", xlab="x", ylab="y" )
```



- hist()

```
hist(x, breaks = "Sturges", freq = NULL, probability = !freq, include.lowest = TRUE,
     right = TRUE, density = NULL, angle = 45, col = "lightgray", border = NULL,
     main = paste("Histogram of" , xname), xlim = range(breaks), ylim = NULL, xlab =
     xname, ylab, axes = TRUE, plot = TRUE, ...)
```

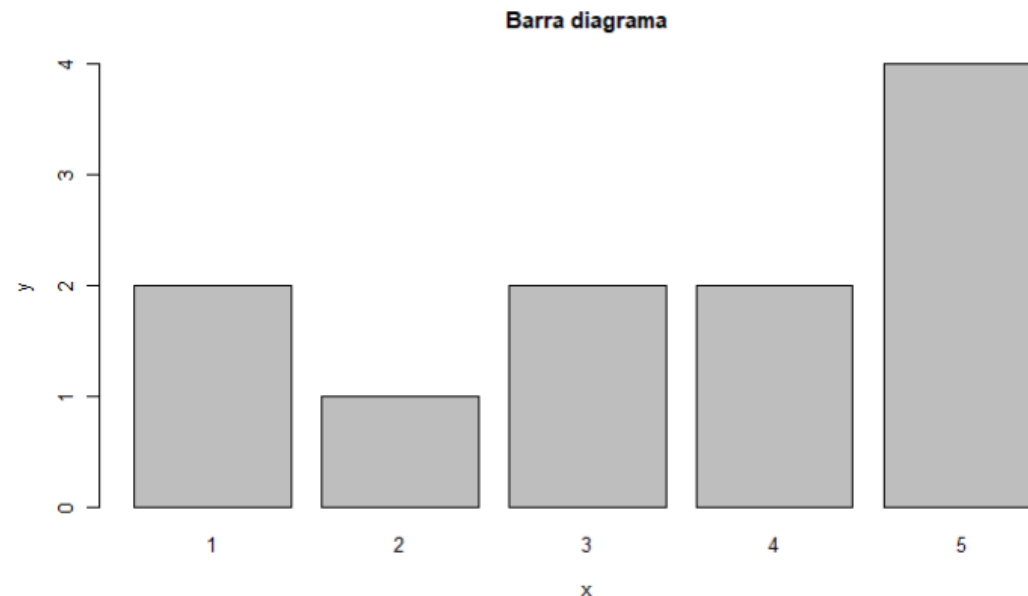
```
> speed<- c(4, 4, 7, 8, 9, 10, 10, 11, 12, 12, 13, 13, 14, 14, 15, 15, 16, 17, 18, 18, 19, 19, 20, 22, 23, 25)
> hist(speed, breaks=seq(0, 25, 5), main="Histograma", xlab="speed", ylab="maiztasuna" )
```



- barplot()

```
barplot(height, width = 1, space = NULL, names.arg = NULL, legend.text = NULL,  
        beside = FALSE, horiz = FALSE, density = NULL, angle = 45, col = NULL, border =  
        par("fg"), main = NULL, sub = NULL, xlab = NULL, ylab = NULL, xlim = NULL, ...)
```

```
> barplot(table(c(1, 1, 2, 3, 3, 4, 4, 5, 5, 5, 5)), main="Barra diagrama", xlab="x", ylab="y")
```

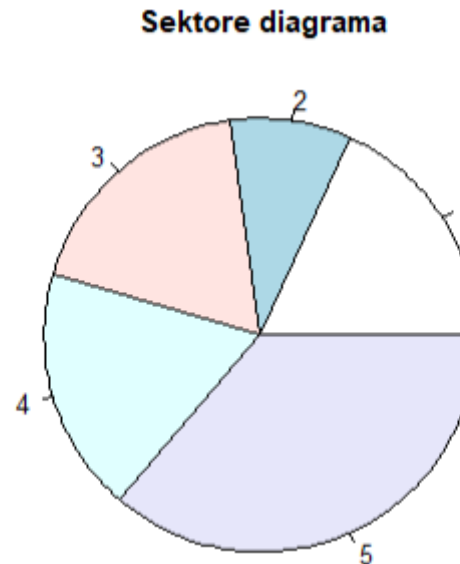




■ pie()

```
pie(x, labels = names(x), edges = 200, radius = 0.8, clockwise = FALSE, init.angle =  
  if(clockwise) 90 else 0, density = NULL, angle = 45, col = NULL, border = NULL,  
  lty = NULL, main = NULL, ...)
```

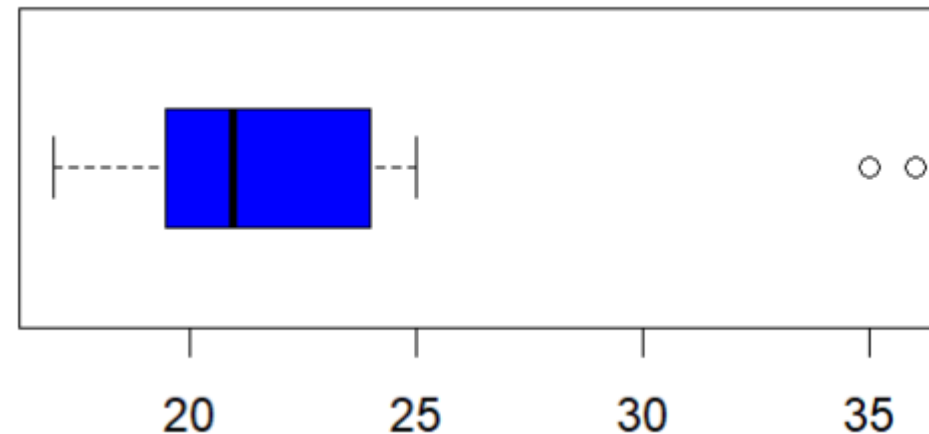
```
> pie(table(c(1, 1, 2, 3, 3, 4, 4, 5, 5, 5, 5)), main="Sektore diagrama")
```



- boxplot()

```
boxplot(x, ..., range = 1.5, width = NULL, varwidth = FALSE, notch = FALSE, outline = TRUE, names, plot = TRUE, border = par("fg"), col = "lightgray", log = "", pars = list(boxwex = 0.8, staplewex = 0.5, outwex = 0.5), ann = !add, horizontal = FALSE, add = FALSE, at = NULL)
```

```
> x<-c(17, 23, 21, 36, 21, 18, 19, 22, 35, 20, 25)  
> boxplot(x, horizontal=TRUE, col="blue")
```



# 1.6. Funtzioak



- function()

```
funtzioarenIzena <- function(arg_1, arg_2, ...) { Funtzioaren gorputza }
```

```
karratu <- function(x)
```

```
{  
  x^2  
}
```

```
> karratu(4) # funtzioa 4 baliotan ebaluatu
```

```
[1] 16
```

# 1.7. Lan direktorioa



## Lan direktorioa

- Fitxategi guztiak, bestela zehaztu ezean, lan-direktoriotik irakurri eta lan-direktorioan gordeko dira.
- Lan-direktorioa zehaztuta badago, datu-fitxategiak lan-direktoriotik R inguruan irakur daitezke fitxategiaren izena erabiliz soilik, fitxategiaren direktorioaren bidea aurrizki gisa gehitu gabe.
- Gainera, R-n sortutako diagrama, datu-markoa edo beste objektu bat gordetzen den bakoitzean, lan-direktorio gisa ezarrita dagoen karpetan gordeko dira.
- **Lanean ari garen lan direktorioa jakiteko**

Kontsolan idatzi:

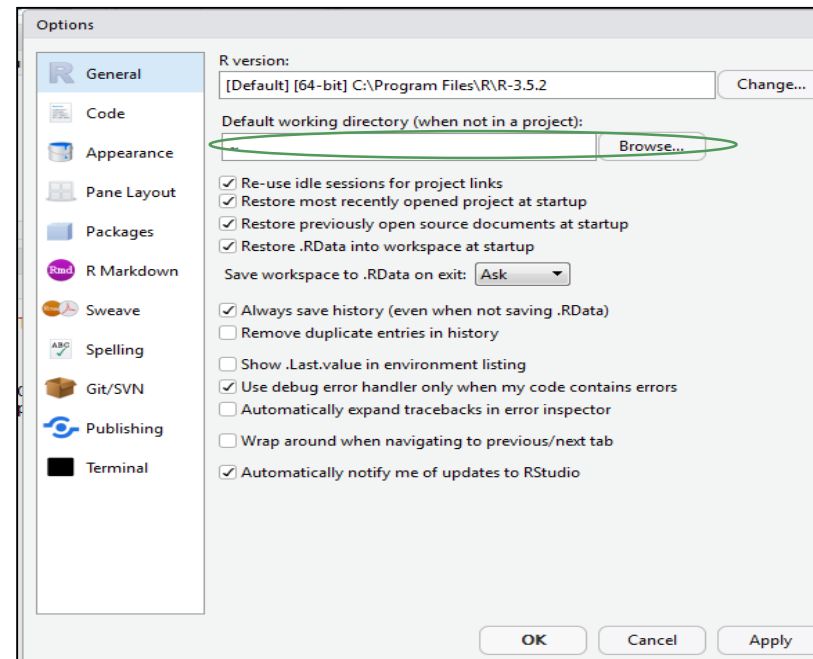
```
> getwd()
```

- Lan direktorioa aldatzeko aukera desberdinak daude:

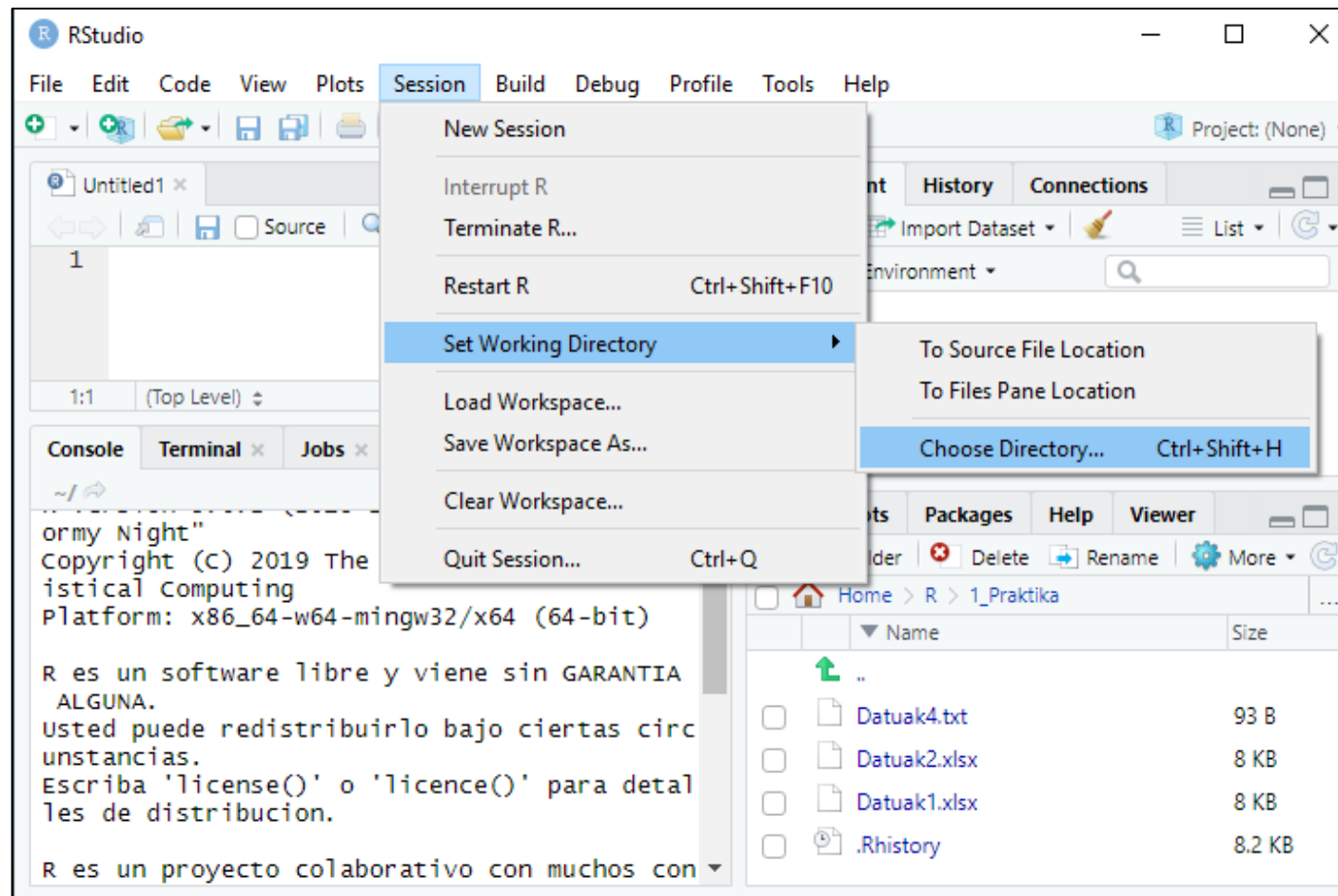
### I.) Kontsola erabiliz:

```
> setwd(“~/R/Praktika”)
```

### II.) Tools → Global Options → Browse:

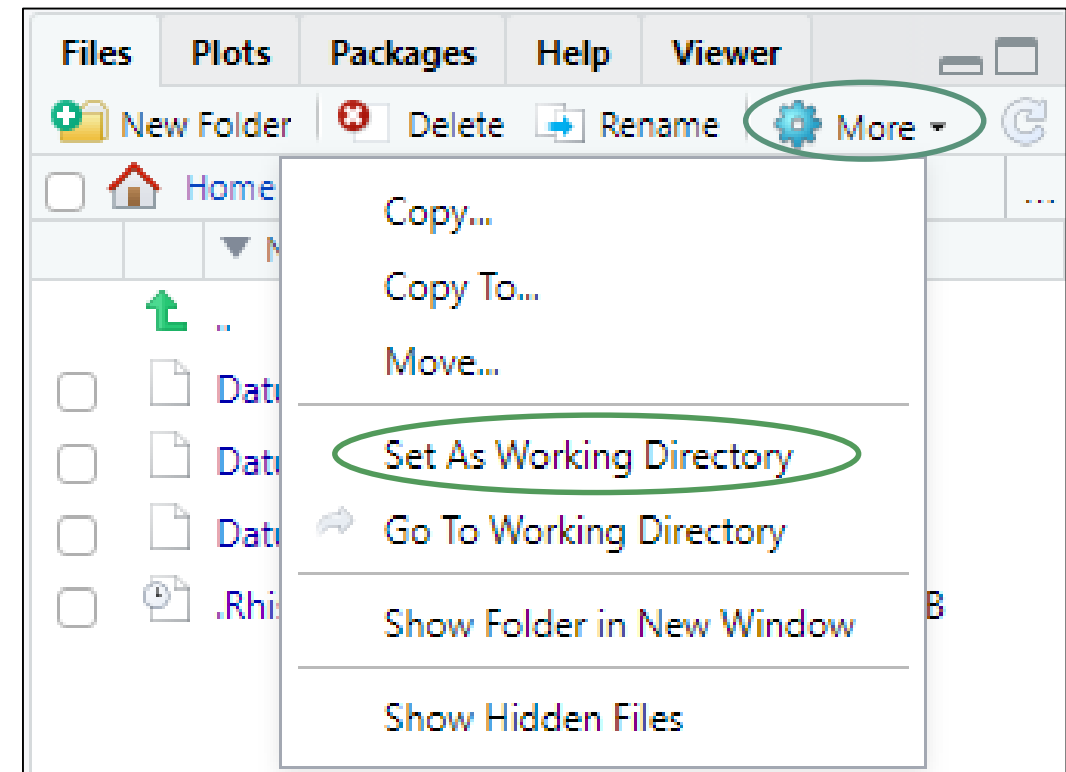
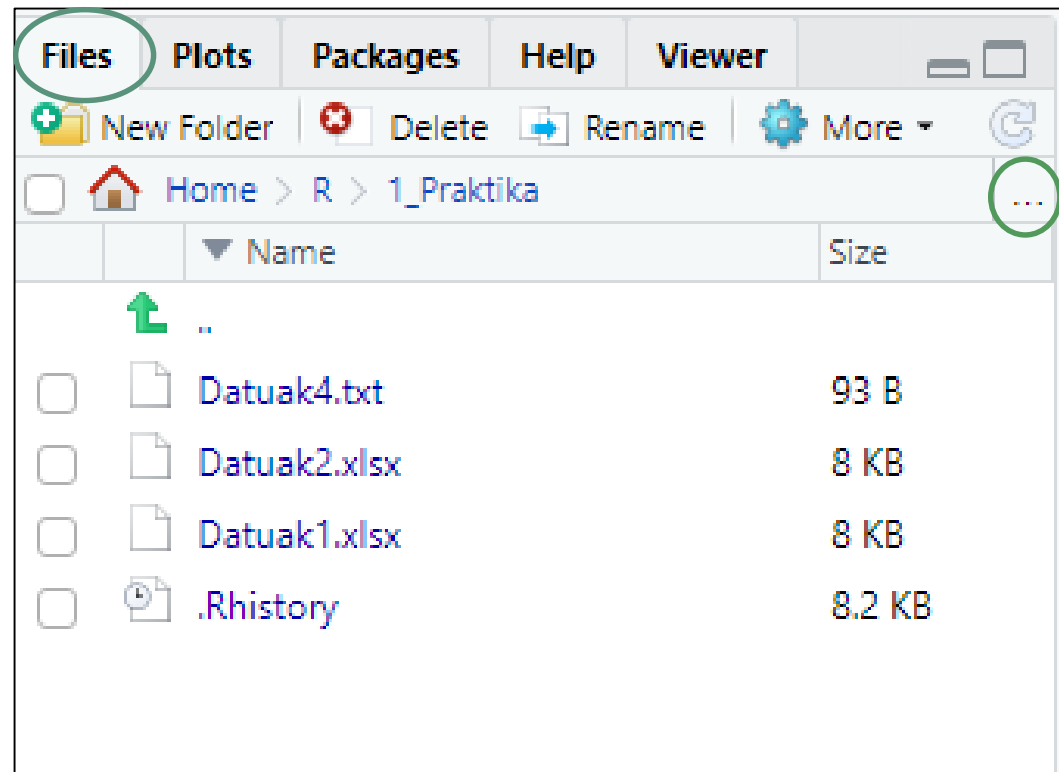


### III.) Session → Set Working Directory → Choose Directory (edo Ctrl+Shift+H teklen konbinazioa)





#### IV.) “File” fitxa: 1. direktorioa aukeratu, 2. Set As Working Directory → Choose Directory



# 1.8. Paketeak



## Paketeak

- R-ko paketeak R-ko funtzioen, lagin-datuaren eta konpilatutako kodeen bilduma dira.
- R-ko paketeak *Comprehensive R Archive Network* (CRAN) izeneko software biltegi zentralean gordeta daude
- R-ko paketeak instalatzen direnean, R ingurunearen “library” izeneko direktorioak instalatuko dira.

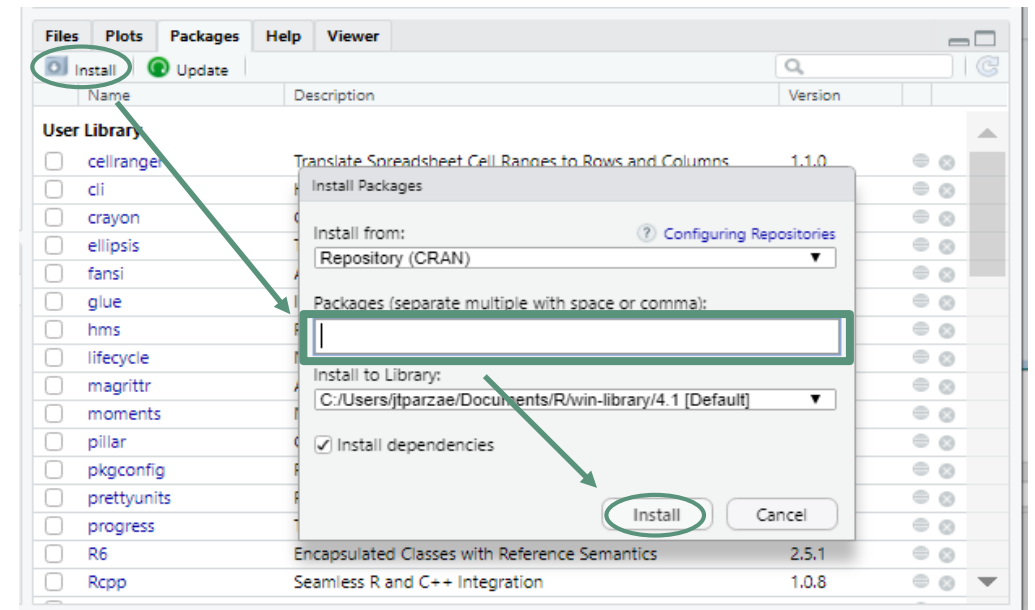
## Pakete bat instalatu

- Pakete baten instalazioa baten bakarrik egin behar da. Bi modu daude pakete bat instalatzeko:

I. **Kontsola erabiliz:** `> install.packages("paketearen_izena")`

II. **“Packages” fitxa erabiliz:**

- “Install” botoian sakatu
- Instalatu nahi den paketearen izena idatzi
- Behean agertzen den “Install” botoia sakatu



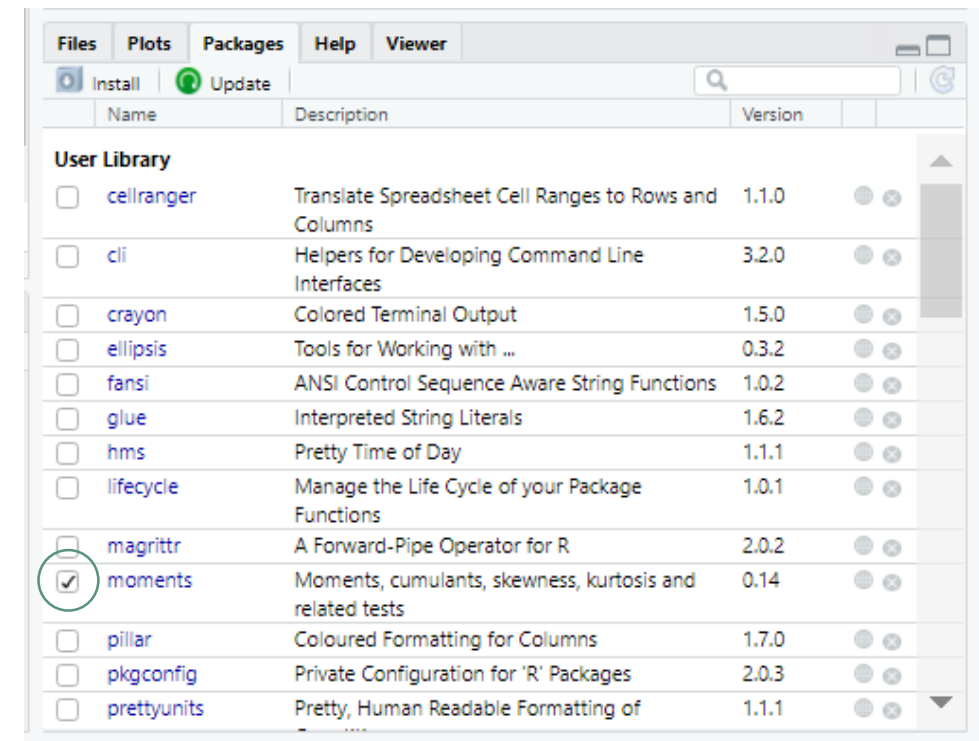
## Pakete bat kargatu

- Deskargatu den pakete bat erabiltzeko paketea kargatu egin behar da. Paketea kargatzea sesio bakoitzeko egin behar da. Pakete bat kargatzeko bi aukera daude:

I. **Kontsola erabiliz:** `> library(paketearen_izena)`

II. **“Packages” fitxa erabiliz:**

- Nahikoa da kargatu nahi den paketea aukeratzea



# 1.9. Fitxategi batetik datuak irakurri



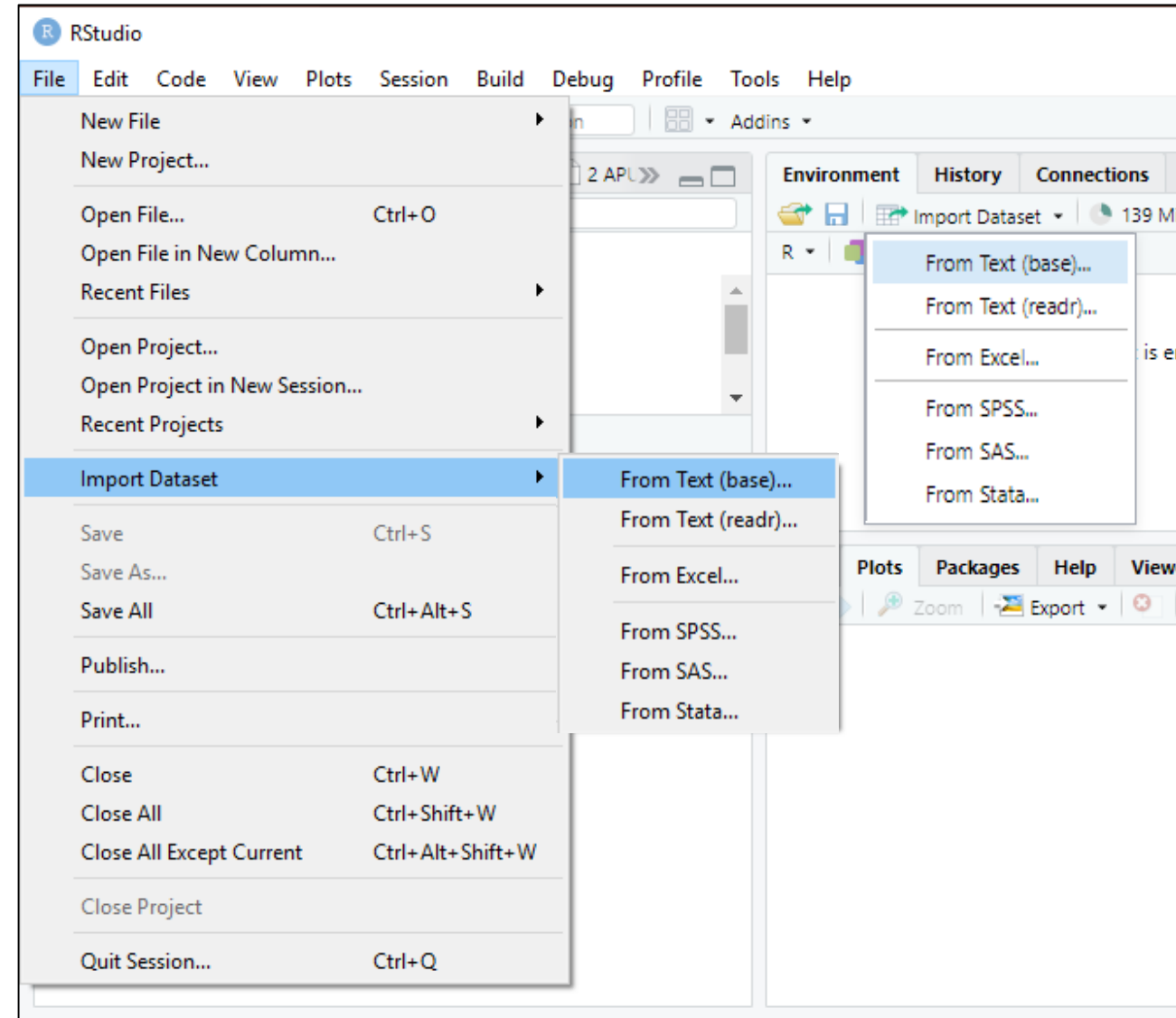
## TXT fitxategiak irakurri:

- File → Import Dataset → From Text (base)...

edo

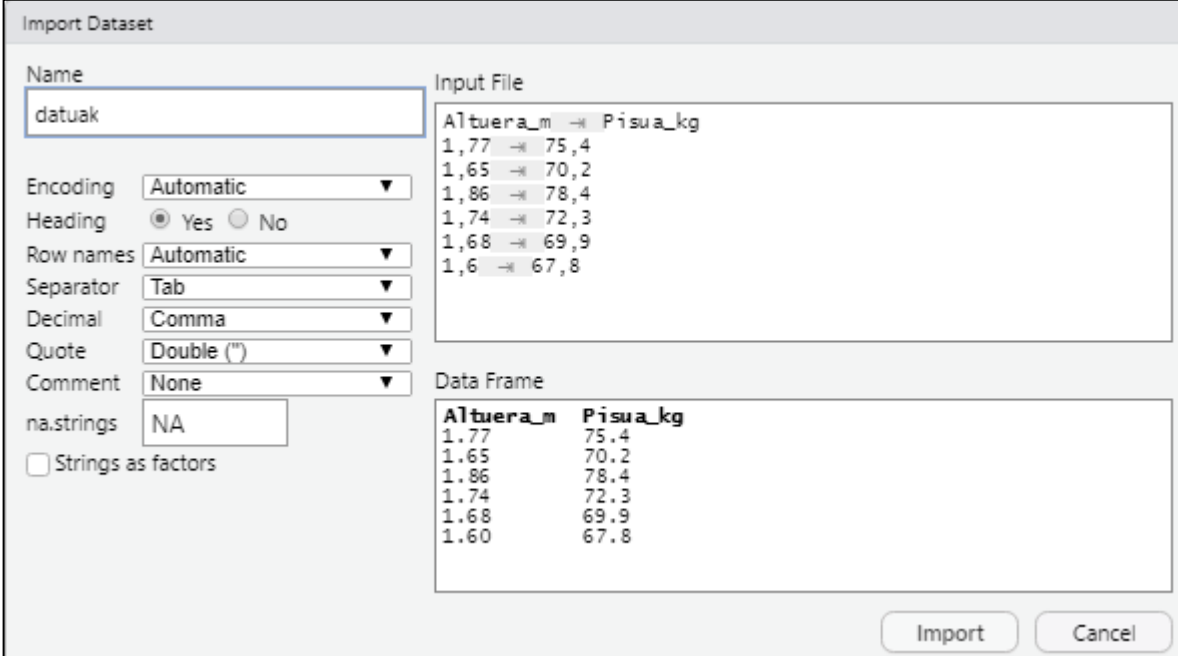
“Environment” fitxan → Import Dataset  
→ From Text (base)...

- Aukeratu irakurri nahi den .txt fitxategia.



## TXT fitxategiak irakurri:

- Aukeratu irakurri nahi den .txt fitxategia. Honako leihoa agertuko da:
- **Name:** data markoari esleituko zaion izena
- **Heading:** aukeratu YES datuek izenburua badaukate
- **Separator:** Tab aukeratu datuak tabuladore bidez banatuta badaude



Import Dataset

Name:

Input File:   
 1,77 → 75,4  
 1,65 → 70,2  
 1,86 → 78,4  
 1,74 → 72,3  
 1,68 → 69,9  
 1,6 → 67,8

Encoding:

Heading:  Yes  No

Row names:

Separator:

Decimal:

Quote:

Comment:

na.strings:

Strings as factors

Data Frame

Altuera_m	Pisua_kg
1.77	75.4
1.65	70.2
1.86	78.4
1.74	72.3
1.68	69.9
1.60	67.8

- Kongsolan honakoa agertuko da:
  - > `datuak <- read.delim2("~/R/Praktika/Datuak.txt")`
  - > `View(datuak)`



## TXT fitxategiak irakurri:

- **Kontsola erabiliz:**

```
read.table(file, header = FALSE, sep = "", quote = "\"'", dec = ".", numerals =
  c("allow.loss", "warn.loss", "no.loss"), row.names, col.names, as.is =
  !stringsAsFactors, na.strings = "NA", colClasses = NA, nrow = -1,...)
```

- **file:** irakurri nahi den fitxategiaren izena.
- **header=TRUE:** zutabeek izenburua dutela adierazteko.
- **sep:** karaktere banatzailea: hutsunea (" "), koma (", "), puntu eta koma ("; ") eta tabuladorea ("/t")
- Irakurri nahi den fitxategia ez badago lan-direktorioan, fitxategira heltzeko bide osoa jarri behar da:
  - > `dataframe_izena <- read.table("C:/Users/Documents/R/Praktika/Datuak.txt", header=T, sep="/t")`
- Irakurri nahi den fitxategia lan-direktorioan badago, fitxategiaren izenarekin nahikoa da:
  - > `dataframe_izena <- read.table("Datuak.txt", header=T, sep="/t")`

# 1.10. Datu-markoak fitxategi batean gorde



## TXT fitxategi batean gorde:

### Kontsola erabiliz:

```
write.table(x, file = "", append = FALSE, quote = TRUE, sep = " ", eol = "\n", na = "NA", dec = ".", row.names = TRUE, col.names = TRUE, qmethod = c("escape", "double"), fileEncoding = "")
```

- **file**: irakurri nahi den fitxategiaren izena.
- **header=TRUE**: zutabeek izenburua dutela adierazteko.
- **sep**: karaktere banatzailea: hutsunea (" "), koma (" , "), puntu eta koma (" ; ") eta tabuladorea (" /t ")
- Fitxategiaren izena bakarrik idazten bada, fitxategia lan-direktorioan gordeko da:
  - > `write.table(datuak, "datuak4.txt", header=T, sep="/t")`
- Fitxategiaren izenaren aurrean fitxategia gorde nahi den direktorioaren bide osoa jar daiteke:
  - > `write.table(datuak, "C:/Users/Documents/Praktika/datuak4.txt", header=T, sep="/t")`