

---

# Guia d'RStudio

---

PID\_00265510

Jordi Mas Elias

---

Temps mínim de dedicació recomanat: 2 hores

---



**Jordi Mas Elias**

L'encàrrec i la creació d'aquest recurs d'aprenentatge UOC han estat coordinats pel professor: Jordi Mas Elias (2019)

Primera edició: setembre 2019  
© Jordi Mas Elias  
Tots els drets reservats  
© d'aquesta edició, FUOC, 2019  
Av. Tibidabo, 39-43, 08035 Barcelona  
Realització editorial: FUOC

*Cap part d'aquesta publicació, incloent-hi el disseny general i la coberta, no pot ser copiada, reproduïda, emmagatzemada o transmesa de cap manera ni per cap mitjà, tant si és elèctric com químic, mecànic, òptic, de gravació, de fotocòpia o per altres mètodes, sense l'autorització prèvia per escrit dels titulars dels drets.*

# Índex

<b>Introducció.....</b>	<b>5</b>
<b>1. Primer pas: baixar R i RStudio.....</b>	<b>9</b>
<b>2. Segon pas: situar-nos.....</b>	<b>11</b>
2.1. Console .....	12
2.2. Script .....	14
2.3. Environment .....	16
2.4. Files .....	17
<b>3. Tercer pas: els paquets.....</b>	<b>20</b>
<b>4. Quart pas: script.....</b>	<b>22</b>
<b>5. Cinquè pas: possibles errors.....</b>	<b>24</b>
<b>6. Glossari.....</b>	<b>26</b>
<b>Resum.....</b>	<b>28</b>



## Introducció

L'ús dels programes de gestió de dades és una tasca cada vegada més estesa en els estudis internacionals. Si bé el coneixement d'aquest tipus de programes no és un requisit indispensable en el món laboral en l'àmbit de les ciències socials, també és veritat que tenir-ne un domini bàsic està esdevenint una plataforma molt útil per a accedir a llocs de treball en aquest sector. Una primera aproximació de l'estudiant a la gestió de dades és, doncs, important en dos sentits. En primer lloc, fer els primers passos amb aquests programes pot ajudar a crear, en alguns estudiants, l'interès pel món de les dades i les ganes de continuar aprenent el llenguatge d'R o altres programes com ara Python, Stata o SPSS. En segon lloc, l'aprenentatge inicial d'un llenguatge com el d'R dota l'estudiant d'unes habilitats que li poden facilitar oportunitats laborals que requereixin un coneixement mínim de la gestió i l'anàlisi de dades.

Aquest mòdul té com a principal objectiu ajudar a l'estudiant a familiaritzar-se amb l'entorn del programa d'R. És per això que les següents pàgines pretenen ser una guia pràctica per iniciar-se als programaris R i RStudio i complementar altres mòduls d'UOC que utilitzen R per als seus propòsits docents. Els altres objectius d'aquests materials són:

- Ajudar a descarregar i instal·lar els programes R i RStudio.
- Facilitar el coneixement de la interfície d'RStudio de manera que ajudi l'estudiant a familiaritzar-se amb les principals finestres del programa.
- Explicar com es comença un projecte amb RStudio, quins són els primers passos i com es guarda un projecte.
- Fer una petita incursió en el llenguatge d'R, mostrant alguns exercicis pràctics que l'estudiant pot fer abans de començar a treballar els mòduls de l'assignatura en qüestió.

També cal, però, fer dues advertències inicials. En primer lloc, ens hem de prendre aquesta guia com un manual d'ajuda que ens permetrà fer les primeres passes amb un software desconegut fins ara per nosaltres com és R. Tot el que s'ensenya en aquestes pàgines va dirigit a poder seguir amb més facilitat les indicacions i activitats que trobarem en altres mòduls d'aquesta assignatura. Aprendre, doncs, a fer una primera immersió en el llenguatge d'R i, molt important, serem capaços de resoldre errors que ens puguin sobrevenir durant la realització de les activitats de l'assignatura. La guia no és, per tant, un manual que ensenyi a programar o analitzar dades amb R. Qui estigui interessat a endinsar-se en el món de l'anàlisi de dades pot consultar guies per a aprendre a fer funcionar R i RStudio que trobarà en el seminari en línia i a la secció

<sup>(1)</sup>Per a guies més avançades, proveu Advanced R (<http://adv-r.had.co.nz>) o R for Data Science (<https://r4ds.had.co.nz/>).

d'aprenentatge en línia del mateix web d'RStudio, o bé pot buscar diferents guies, fòrums o tutorials, fàcils de trobar a internet (vegeu per exemple Stack Overflow). Aquestes guies permeten començar a utilitzar el llenguatge R des d'un nivell inicial i aprendre'l de manera progressiva fins a obtenir un nivell avançat<sup>1</sup>.

La segona advertència es refereix a la dificultat d'iniciar-se en els programes de gestió de dades. Aquest tipus de programari acostuma a tenir una corba d'aprenentatge molt llarga. Per a una persona que es posi per primera vegada al davant d'aquest tipus de programes, les primeres hores poden ser difícils i feixugues. Se n'aprèn molt a poc a poc, a base d'equivocar-se i entrebancar-se moltes vegades, i només una formació continuada en el temps pot comportar grans retorns en l'ús d'aquest tipus de programes. Per tant, l'estudiant haurà de tenir paciència i seguir molt atentament aquesta guia per a poder superar les sempre difícils hores inicials. Cal dir que les hores invertides no són en va, sinó que són un guany en experiència acumulada que ajudarà en el futur a incrementar la productivitat en l'ús del programa.

## R i RStudio

L'assignatura que treballareu requereix l'ús dels programes R i RStudio. Són dos programes diferents i els dos s'han d'instal·lar a l'ordinador. Una vegada instal·lats, totes les operacions indicades en aquest manual les farem utilitzant RStudio, però també és necessari tenir instal·lat el programa R, encara que no l'executem en cap moment, perquè és el motor que utilitza RStudio per a funcionar. En el següent apartat, s'indiquen els passos que s'han de seguir per a instal·lar els dos programes.

Al llarg d'aquesta guia utilitzarem indistintament els termes R i RStudio. Això no vol dir que en els casos en què ens referim a R hàgim d'utilitzar el programa R. Sempre haurem d'utilitzar RStudio en totes les nostres operacions, però utilitzarem els dos termes en aquesta guia, entenent normalment R com el motor que fa funcionar RStudio i processa la informació, i RStudio com la interfície, entès com l'entorn de programació que ens permetrà utilitzar les eines del programa R d'una manera més visual i eficient. Podem utilitzar R i RStudio des d'un sistema operatiu Windows, Mac OS X o Linux.

## Què farem?

Aquesta guia inicial està estructurada de la manera següent. En primer lloc, veurem com baixar i executar els programes R i RStudio. En segon lloc, executarem el programa RStudio i n'explicarem la interfície, en especial les quatre finestres amb les quals treballarem. Aquesta guia conté exercicis pràctics i està pensada per a ser utilitzada en paral·lel amb RStudio. D'aquesta manera, l'estudiant podrà aplicar directament al programa les indicacions proposades.

També aprendrem a utilitzar els paquets o llibreries d'R i a preparar els primers codis amb l'script. Finalment, aquesta guia conté un apartat d'ajuda per a resoldre errors.



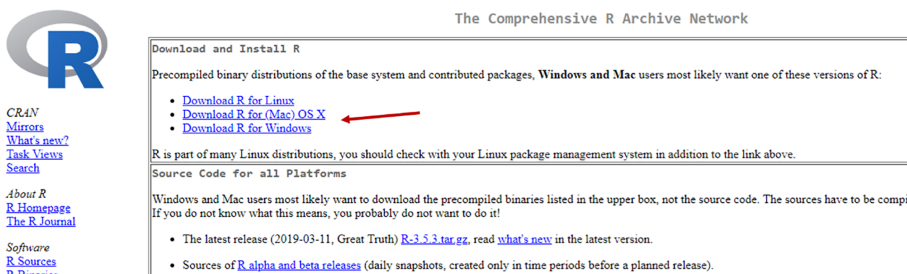


## 1. Primer pas: baixar R i RStudio

R i RStudio són dos programes diferents que s'han de baixar des de pàgines web diferents. Necessitarem, però, tenir els dos programes instal·lats per a treballar-hi. R és el motor i té els engranatges suficients per a funcionar de manera autònoma, però la seva visualització és austera i poc pràctica per a un usuari que s'inicia en llenguatges de programació. RStudio, en canvi, no és autònom, necessita R per a funcionar, però incorpora moltes característiques que en faciliten el seu maneig, com ara finestres, elements visuals i diferents opcions que ens ajudaran a treballar amb R. Per tant, nosaltres treballarem amb RStudio però necessitarem tenir els dos programes instal·lats, ja que RStudio no pot funcionar si no tenim prèviament instal·lat R.

Trobarem els dos programes a les seves respectives webs. Recordeu triar l'opció correcta en funció de si el vostre sistema operatiu és Linux, Mac o Windows. Per a instal·lar R, el primer pas és accedir a la seva web i seleccionar el sistema operatiu. Tingueu en compte que la versió del programa pot ser diferent a la que veieu a les següents figures. En aquest manual veiem la versió 3.5.3., però és molt possible que quan us descarregueu el programa ja tingueu a la vostra disposició una versió més nova.

Figura 1. Baixar R: primer pas, seleccionar el sistema operatiu.



The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages. **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#) ←
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

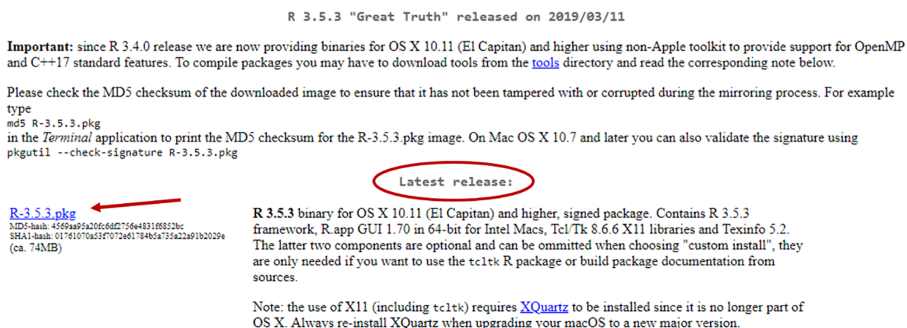
Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled if you do not know what this means, you probably do not want to do it!

- The latest release (2019-03-11, Great Truth) [R-3.5.3.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).

En cas d'utilitzar el sistema Mac, a la següent finestra trobareu l'enllaç per a baixar l'arxiu `.pkg`. En cas de Windows, cal seleccionar 'Install R for the first time' i a la següent finestra en podeu baixar l'última versió.

Figura 2. Baixar R: baixar l'arxiu `.pkg` en cas de treballar amb sistema operatiu Mac.



R 3.5.3 "Great Truth" released on 2019/03/11

**Important:** since R 3.4.0 release we are now providing binaries for OS X 10.11 (El Capitan) and higher using non-Apple toolkit to provide support for OpenMP and C++17 standard features. To compile packages you may have to download tools from the [tools](#) directory and read the corresponding note below.

Please check the MD5 checksum of the downloaded image to ensure that it has not been tampered with or corrupted during the mirroring process. For example type

```
md5 R-3.5.3.pkg
```

in the *Terminal* application to print the MD5 checksum for the R-3.5.3.pkg image. On Mac OS X 10.7 and later you can also validate the signature using

```
pkgutil --check-signature R-3.5.3.pkg
```

**Latest release:**

[R-3.5.3.pkg](#) ←

R 3.5.3 binary for OS X 10.11 (El Capitan) and higher, signed package. Contains R 3.5.3 framework, R.app GUI 1.70 in 64-bit for Intel Macs, Tcl/Tk 8.6.6 X11 libraries and Texinfo 5.2. The latter two components are optional and can be omitted when choosing "custom install", they are only needed if you want to use the `tc1tk` R package or build package documentation from sources.

Note: the use of X11 (including `tc1tk`) requires [XQuartz](#) to be installed since it is no longer part of OS X. Always re-install XQuartz when upgrading your macOS to a new major version.

Figura 3. Baixar R: seleccionar 'Install R for the first time' en cas de treballar amb sistema operatiu Windows.

**R for Windows**

Subdirectories:

- [base](#) Binaries for base distribution. This is what you want to [install R for the first time](#).
- [contrib](#) Binaries of contributed CRAN packages (for R  $\geq$  2.13.x; managed by Uwe Ligges). There is also information on [third party software](#) available for CRAN Windows services and corresponding environment and make variables.
- [old contrib](#) Binaries of contributed CRAN packages for outdated versions of R (for R  $<$  2.13.x; managed by Uwe Ligges).
- [Rtools](#) Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

Figura 4. Baixar R: baixar l'última versió en cas de treballar amb sistema operatiu Windows.

**R-3.5.3 for Windows (32/64 bit)**

[Download R 3.5.3 for Windows](#) (79 megabytes, 32/64 bit)

[Installation and other instructions](#)

[New features in this version](#)

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of the .exe to the [fingerprint](#) on the master server. You will need a version of md5sum for windows: both [graphical](#) and [command line versions](#) are available.

Un cop descarregat R, passarem a baixar RStudio. Hem d'accedir a la web de descàrrega del programa i seleccionar el sistema operatiu corresponent. Fixeu-vos bé que a la part superior ens deixa clar que RStudio no pot treballar sense tenir R instal·lat.

Figura 5. Baixar RStudio: escollir el sistema operatiu.

#### RStudio Desktop 1.1.463 — Release Notes

RStudio requires R 3.0.1+. [If you don't already have R, download it here.](#)

Linux users may need to import RStudio's public code-signing key prior to installation, depending on the operating system's security policy.

#### Installers for Supported Platforms

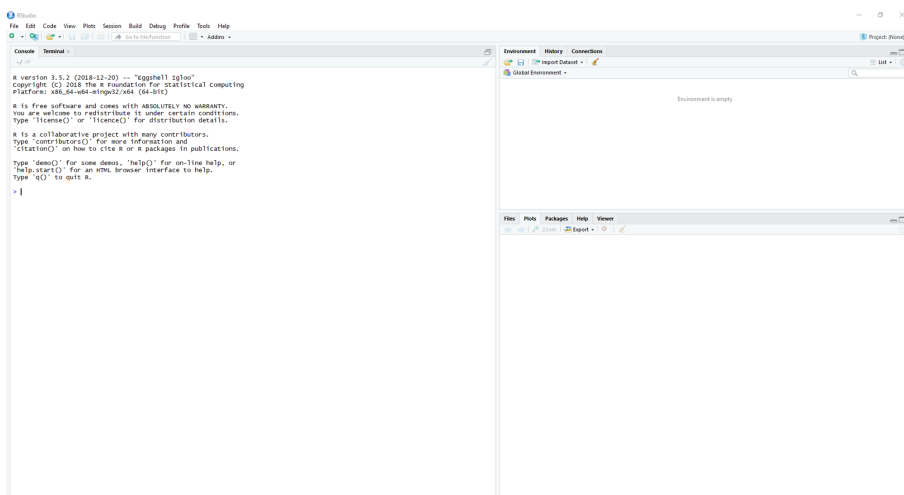
Installers	Size	Date	MD5
RStudio 1.1.463 - Windows Vista/7/8/10	85.8 MB	2018-10-29	58b3d796d8cf96fb8580c62f46ab64d4
RStudio 1.1.463 - Mac OS X 10.6+ (64-bit)	74.5 MB	2018-10-29	a79032ba4d7daaa86a8da01948278d94
RStudio 1.1.463 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	89.3 MB	2018-10-29	8a6755fa9fae2bafce289df3358aaf63
RStudio 1.1.463 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	97.4 MB	2018-10-29	bc50d6bd34926c1cc3ae4a209d67d649
RStudio 1.1.463 - Ubuntu 16.04+/Debian 9+ (64-bit)	65 MB	2018-10-29	cf6d59db18619cc78d1592fefaa7c753
RStudio 1.1.463 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	88.1 MB	2018-10-29	742f0bad60dfeaa3281576e14ad6699e
RStudio 1.1.463 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	90.6 MB	2018-10-29	c7303067a0ca99deea7e427b856952d1

Recordeu que, un cop baixats, heu d'instal·lar els dos programes. El programa R caldrà que estigui instal·lat al nostre ordinador però no serà necessari executar-lo. Sí que instal·larem i executarem RStudio, que serà el programa amb el qual farem totes les operacions.

## 2. Segon pas: situar-nos

Quan entrem a RStudio per primera vegada, veurem un escriptori format per tres finestres principals. La interfície del programa és molt simple, però si no estem acostumats a aquest tipus d'entorns de programació ens costarà un temps habitar-nos-hi. A l'esquerra veiem una finestra més gran que les altres, amb les pestanyes 'Console' i 'Terminal', mentre que a la dreta en veiem dues de més petites: la de sobre té les pestanyes 'Environment', 'History' i 'Connections' mentre que la de sota té les pestanyes 'Files', 'Plots', 'Packages', 'Help' i 'Viewer'.

Figura 6. Pantalla inicial d'RStudio.



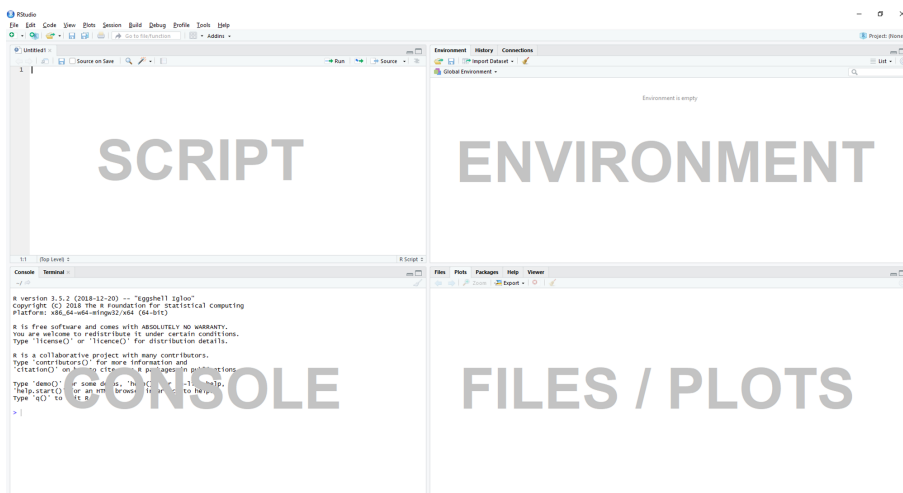
El primer que farem és afegir una quarta finestra a l'escriptori d'RStudio:

- A la barra superior de la pantalla anirem a: File -> New File -> **R Script** (alternativament podeu prémer Ctrl+Shift+N). Normalment la tecla equivalent al Ctrl en PC serà la tecla Command en Mac.
- R acaba de crear una quarta finestra amb el nom '**Untitled1**' i l'ha ubicada a la part superior esquerra de l'escriptori. Acabem de crear un script, una de les eines fonamentals per a utilitzar R. El primer que farem és guardar aquest document en blanc (podem guardar clicant a la icona del disquet, prement Ctrl+S o bé anant a File -> Save As).
- Quan intentem guardar el document, el més probable és que per defecte ens suggereixi guardar-lo a la carpeta "Documents" (o "Documentos") del nostre ordinador. El més recomanable és crear una carpeta a dins de la carpeta de l'assignatura titulada 'R Scripts'. N'hauré de crear uns quants i baixar-los durant el curs, per tant tenir una carpeta on guardar-los ens serà útil.

- Aquest primer script el guardarem amb el nom, per exemple 'R\_Script\_prova'.

En la següent imatge veiem les quatre finestres d'RStudio. Per comoditat a l'hora de treballar, farem les finestres de l'esquerra més amples que les de la dreta de manera que ens quedi més o menys com a la imatge.

Figura 7. Les quatre finestres d'RStudio.



En els propers apartats repassem les quatre finestres (Console, Script, Environment i Files), una a una:

## 2.1. Console

La *console* o consola és l'instrument que utilitzem per a comunicar-nos amb RStudio. El que nosaltres li indiquem aquí, R ho llegirà i farà les operacions necessàries. En aquesta finestra mantindrem la visualització de la pestanya 'Console' i tindrem amagada la de 'Terminal'. A la consola, fixem-nos en el símbol > que apareix al final de tot del text inicial. Se'n diu operador. A l'operador és on podem introduir les nostres primeres indicacions per tal de comunicar-nos amb R. Teclegeu els codis que apareixen a la taula 1 i a continuació premeu la tecla Enter.

Taula 1. Operacions inicials amb la consola

Codi	Return
1+1	[1] 2
1:50	[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 [21] 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 [41] 41 42 43 44 45 46 47 48 49 50
"Hola"	[1] "Hola"

A cada codi, veiem que R ens retorna primer un número entre claudàtors [1]. Aquest número, que trobarem al principi de cada línia, ens indica en quina posició es troba el primer valor de cada fila de tota la cadena de valors que ens està mostrant. Com veiem, el primer valor de la primera fila és el valor 1. Quan arriba a la segona fila, ja ens ha mostrat 20 valors i per tant el primer de la segona fila és el 21 en aquest exemple (en el vostre ordinador hi pot aparèixer un valor diferent). El primer de la tercera és el 41. En aquest cas, el número entre claudàtors ens coincideix amb el primer número de cada fila, però no ha de ser així necessàriament. Si demanem a R que ens mostri els mateixos números d'una manera desordenada veurem com ja no coincideixen. Per exemple, teclegeu `sample(50)` i premeu Enter.

A continuació, tenint la consola seleccionada, premeu la fletxa amunt del vostre teclat. Veureu que apareixen els codis que hem entrat prèviament. La consola memoritza tots els comandaments que hem demanat en la sessió actual que tenim iniciada a RStudio. Amb les fletxes amunt i avall del teclat podem recuperar codis anteriors i modificar-los. Seguim, a la taula 2, amb una darrera prova.

Taula 2. Errors a R

Codi	Retorn
R	Error: object 'R' not found
\$	Error: unexpected '\$' in "\$"

Si entrem els dos codis de la taula anterior veurem que ens dona error. La consola, en una anotació en vermell, ens indica en el primer cas que no ha trobat l'objecte 'R'. En el segon cas ens indica que ha trobat un símbol que no ha entès. En el nostre procés d'aprenentatge amb RStudio toparem amb desenes d'errors a la consola. Hem de considerar-ho un fet habitual, que amb paciència solucionarem, i saber que el primer pas per a arribar a la solució requereix llegir molt atentament el que comenta R en aquests missatges en vermell.

Per exemple, quan hem introduït R, el programa no ens ha trobat cap objecte amb el nom R. Efectivament, no recordem haver-ne creat cap. Doneu ara la següent indicació a l'operador: `R <- c(1:2, 5, 7:10)` i premeu Enter. A continuació torneu a teclejar R i premeu Enter. Com veieu, el que hem fet primer és crear un objecte amb el nom R per mitjà del comandament `<-` (símbol 'menor que' seguit del símbol 'menys')<sup>2</sup>, que serveix per a crear objectes. Després hem demanat a R que ens imprimeixi aquest objecte cridant el seu nom. Com que l'hem creat prèviament, ens l'ha mostrat. La funció `c()` que hem utilitzat significa *concatenat* i l'haurem d'utilitzar de manera molt habitual quan treballem amb objectes que tenen més d'un valor. Per a crear aquestes cadenes de valors introduïrem la lletra c, seguida de parèntesis. A dins dels parèntesis, separats per comes, hi posarem els valors que vulguem.

<sup>(2)</sup>Una drecera per a crear aquest símbol és amb el teclat introduint Alt+- (signe menys).

#### Cridar i imprimir

Els termes cridar i imprimir seran habituals en aquest mòdul. Quan cridem un objecte d'R vol dir que el demanem a la consola. Per a fer-ho, hem de teclejar el nom de l'objecte a la consola. Quan R ens mostra aquest objecte a la consola moltes vegades direm que ens l'imprimeix.

La consola ens mostra els resultats de totes les operacions que fem amb R. En alguns casos, però, veurem que no ens retorna cap resultat. Això passa principalment quan creem un objecte. Quan anteriorment hem creat l'objecte `R`, no hem rebut cap indicació a la consola conforme hem creat satisfactòriament l'objecte. Això, encara que pugui semblar el contrari, és bon senyal. Quan creem objectes, RStudio ens emmagatzema l'objecte en memòria però no ens el mostra en pantalla. Per a veure'l imprès a la consola, haurem de cridar-lo un cop creat, o bé podem posar tot el codi entre parèntesis en el moment de crear-lo perquè ens el guardi i ens l'imprimeixi en el mateix moment. Feu la prova escrivint, per exemple, `(S <- c(1:100))`.

#### El símbol : a R

Quan vulguem demanar una cadena llarga de valors consecutius no cal que introduïm cada un d'aquests valors. El símbol `:` és una drecera que ens permet introduir només l'inici i el final. Si volem imprimir els valors de l'1 al 100, només cal que teclegem `1:100`.

També cal remarcar que R no guardarà res del que fem si no li ho indiquem explícitament amb el comandament `<-`. Per exemple, quan hem efectuat l'operació `sample(50)`, R ens ha mostrat el resultat a la consola, però no ens ha guardat el resultat en forma d'objecte. Per a guardar-lo, hem d'introduir primer el nom que tindrà l'objecte, després el comandament `<-` i finalment teclejar el contingut que vulguem que tingui l'objecte. Feu el següent exercici: creeu i imprimeu al mateix temps un objecte anomenat `mostra` que contingui `sample(50)`.

Cada ordre completa que escrivim a la consola es diu línia de comandament. Ens hi referirem indistintament tant per aquest nom com pel de codi. Perquè R ens llegeixi qualsevol codi o línia de comandament a la consola, sempre haurem de prémer la tecla Enter al final. R ho llegirà, ho processarà i ens retornarà els resultats.

## 2.2. Script

El guió o *script* és un document amb el qual podem anotar i transmetre a R les nostres indicacions. A partir d'ara, en aquest mòdul deixarem de donar les indicacions per mitjà de la consola i ho farem amb l'script, ja que és com ho farem habitualment quan treballem amb R. Tot i que les funcions de l'script i la consola són semblants, a l'script podem registrar totes les nostres operacions i guardar línies i línies d' anotacions. Hem de considerar l'script com un document de text que podem fer servir d'esborrany per a apuntar i modificar tantes vegades com faci falta els nostres codis d'R. Apuntar-ho a l'script ens permetrà preparar, modificar, guardar i compartir codis més fàcilment.

Veureu com totes les línies de l'script estan numerades a la part esquerra. A continuació teniu en el següent quadre de color gris diverses línies de comandament. Copieu-les totes i enganxeu-les a la primera línia numerada de codi del vostre script. Un cop enganxades, situeu el cursor en aquesta primera línia de codi. A diferència de la consola, per a executar un codi o línia de comandament a l'script hem de prémer `Ctrl+Enter` als PC i `Command+Enter` als Mac<sup>3</sup>.

<sup>(3)</sup>Si volem ordenar a R que llegeixi tot l'script, de dalt a baix, haurem de prémer `Ctrl+Shift+Enter` als PC i `Ctrl+Shift+Command` als Mac.

Cada vegada que executeu una línia de comandament a l'script, el cursor saltarà al principi de la línia següent. Aneu executant totes les línies de l'script fins a arribar a l'última.

```
#Base de dades Mtcars
mtcars
hist(mtcars$mpg) #Histograma de la variable mpg
plot(mtcars$disp, mtcars$wt)
plot(mtcars$  disp,
      mtcars$qsec)
```

En aquest simple procés hem fet diverses coses que hem de tenir en compte per a saber com funcionen les ordres que donem a l'script. El primer que hem de saber és que el símbol # (coixinet) desactiva tota la resta d'informació que hi hagi en la mateixa línia de comandament. Això succeeix tant a la primera línia amb #Base de dades Mtcars com a la tercera línia, on a partir de l'aparició del coixinet, R entendreà que ha de deixar de llegir el codi en aquesta línia i passar a la línia següent. El coixinet, doncs, ens serà molt útil per a fer-nos anotacions. Podem utilitzar-lo, per exemple, com a títol per a indicar les operacions que vindran a continuació, i també podem utilitzar-lo al final d'una operació per a donar algun detall sobre la línia en qüestió.

La segona cosa que hem pogut observar en l'exemple anterior és com funcionen els espais. Fixeu-vos que en la quarta línia del codi anterior no hi ha espai entre \$ i disp, mentre que a la cinquena línia aquests dos termes sí que estan separats per diversos espais. Hem de saber que R no llegeix espais, de manera que tant en una línia com en l'altra R ens ha llegit el mateix objecte mtcars\$disp. Els espais ens són útils per a poder llegir i interpretar els codis amb més facilitat, però R no els té en compte quan llegeix un codi. Per exemple, per a R serà exactament el mateix el codi `v<-c(3,2)` que el codi `v < - c ( 3 , 2 )`.

Finalment, cal assenyalar que quan demanem a R que llegeixi una línia, el programa ens saltarà automàticament a la següent si interpreta que el codi no està finalitzat. Els codis d'R no acaben mai amb una coma. Com que la darrera línia del codi anterior acaba amb una coma, R ha passat a llegir automàticament la línia següent. Hi ha altres símbols, com un parèntesi obert ( ( ), una cometa oberta ( " ) o un símbol de suma ( + ) que requereixen codi a la seva dreta. R interpretarà que ha de continuar llegint i/o saltar de línia si és necessari. Els salts de línia ens són útils per a organitzar-nos el codi i facilitar la nostra lectura.

També és important saber que si el cursor està situat en qualsevol posició en una línia de comandament de l'script, R llegirà tota la frase sencera (i això pot comportar que R també ens llegeixi línies superiors o inferiors). R anirà a buscar on és el principi i el final del codi complet i ens el llegirà sencer. Si veu que està incomplet ens retornarà error. Si en lloc de situar el cursor en una part del codi, seleccionem una part de la línia de comandament il·luminant-la

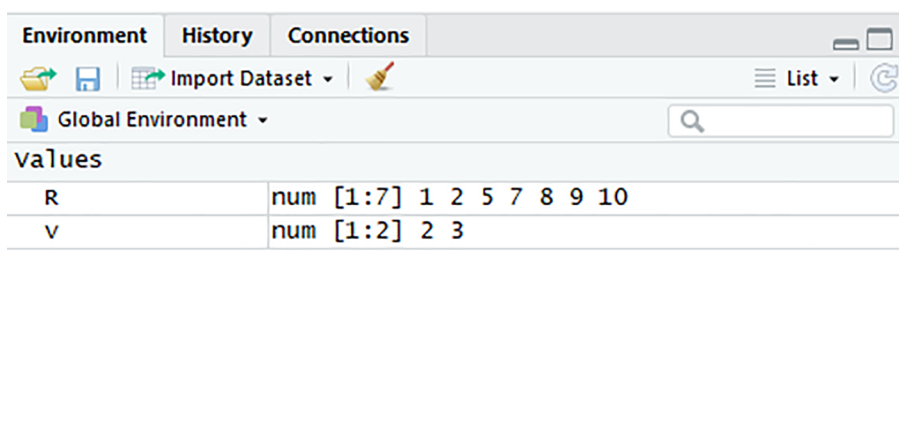
amb el cursor, R només ens llegirà aquesta part. Proveu, del codi anterior, de seleccionar només `mtcars$disp`. Després feu el mateix amb `mtcars$qsec`. R ens llegirà únicament la selecció (sempre que tingui sentit gramatical per a R).

Utilitzar l'script per a apuntar tots els passos que anem seguint quan creem i desenvolupem codis és molt útil tant per a nosaltres com per a terceres persones. Nosaltres ens assegurem de conservar en detall tot el procediment que hem seguit per a generar unes dades determinades. Així, podem repetir el procediment en qualsevol moment, canviar-lo o bé examinar-lo per a detectar possibles errors. Anotar els codis a l'script també ajuda el nostre rol d'analista de dades de cara a terceres persones. Si volem compartir la nostra feina, no caldrà que guardem i enviem diverses taules i gràfics. A vegades tampoc serà necessari que guardem una base de dades. En un simple script, que ocupa poquíssim al nostre ordinador, tenim detallat tot el procediment, de manera que l'altra persona només haurà de reproduir al seu ordinador els codis que hem preparat.

### 2.3. Environment

A la finestra Environment o entorn podem consultar tot el que R té emmagatzemat en memòria. A la finestra principal hi trobem els objectes que ha creat. Si hem seguit l'exercici anterior de crear els objectes `R` i `v`, R ens mostrarà aquests dos objectes a la finestra juntament amb una breu descripció. Com veieu a la figura 8, `R` és un objecte numèric de set caràcters mentre que `v` és un objecte també numèric però de dos caràcters. Des de la consola o l'script podem obtenir informació dels objectes creats amb les funcions `ls()` o `objects()`.

Figura 8. Global Environment.



The screenshot shows the RStudio Global Environment window. At the top, there are tabs for 'Environment', 'History', and 'Connections'. Below the tabs, there are icons for file operations and a search bar. The main area displays a table of variables in the Global Environment. The table has two columns: the variable name and its value. The variable 'R' is of type 'num' and has values [1:7] 1 2 5 7 8 9 10. The variable 'v' is also of type 'num' and has values [1:2] 2 3.

values	
R	num [1:7] 1 2 5 7 8 9 10
v	num [1:2] 2 3

Els objectes també es poden eliminar. Per a eliminar-ne un de concret, introduïu el nom de l'objecte a dins de la funció `rm()` o `remove()`. Es poden eliminar tots els objectes amb la funció `rm(list=ls())`.



A part d'objectes, R també pot emmagatzemar altres tipus de dades. Si cliqueu en l'apartat de 'Global Environment', us apareixerà un desplegable on podreu comprovar els paquets instal·lats a l'entorn d'R. Un paquet o llibreria es refereix al mateix concepte, un grup de funcions i objectes, i ens hi referirem en aquest mòdul de les dues maneres (vegeu l'apartat Pas 3: Paquets). A dins el desplegable de 'Global Environment' veureu les set llibreries amb què R ve per defecte: *stats*, *graphics*, *grDevices*, *utils*, *datasets*, *methods* i *base*. Més endavant afegirem altres llibreries en aquest desplegable. També podem consultar les llibreries carregades amb la funció `search()`.

A sobre de l'apartat 'Global Environment' hi trobem 'Import Dataset'. Amb aquest apartat podrem importar diferents bases de dades a R. El més senzill és baixar fitxers amb extensió *.csv* o *.txt*. Si voleu importar fitxers en format Excel, Stata o SPSS, R us demanarà que baixeu la llibreria necessària per a poder-los importar en cada cas. L'ideal és importar dades fent servir codi, ja que ens permet replicar un comandament en altres ordinadors, però també és útil saber que podem importar manualment bases de dades des de la finestra Environment.

Les altres pestanyes d'aquesta finestra no les farem servir. 'History' fa una recopilació dels procediments que ha fet RStudio en la sessió actual. Podeu fer-hi un cop d'ull per a veure l'historial d'operacions que heu fet fins ara.

## 2.4. Files

Com veieu, aquesta finestra té diverses pestanyes: 'Files', 'Plots', 'Packages', 'Help' i 'Viewer'. A la pestanya 'Files' ens hauria d'aparèixer la carpeta del nostre ordinador on R anirà per defecte a buscar els arxius, que acostuma a ser la carpeta 'Documents' o 'Documentos'. R necessita una carpeta del nostre ordinador per a operar. Serà on ens guardarà les bases de dades que exportem i el primer lloc on anirà a buscar les bases de dades que vulguem importar. Aquesta carpeta l'anomenarem directori de treball. Podem consultar el contingut del directori de treball per mitjà de `dir()` i la seva ubicació dins el nostre ordinador amb `getwd()`.

Si tenim una carpeta especial on vulguem tenir tots els arxius que anem generant amb R, serà convenient indicar-la com a directori de treball per a facilitar-nos la feina. Podem, per exemple, anar a la carpeta de l'assignatura que tinguem creada al nostre ordinador i crear-hi a dins una nova carpeta anomenada 'R Docs'. Un cop siguem en aquesta nova carpeta, hem d'especificar a R que volem fixar-la com a directori de treball. Per a canviar manualment la ubicació del directori de treball ho podem fer mitjançant els tres punts suspensius (...) que trobem a l'extrem dret de la finestra 'Files'. Un cop haguem seleccionat el nou directori de treball, podem anar a l'engranatge blau de la finestra 'Files' i marcar 'Set As Working Directory'. Aquest procés també es pot fer per mitjà de l'script amb la funció `setwd()`, en què a dins dels parèntesis especificarem la

### Utilitzar arxius d'Excel, Stata i SPSS

Les llibreries més comunes per a baixar bases de dades d'Excel són *readxl* i *gdata*, mentre que per a Stata i SPSS utilitzarem els paquets *haven* o *foreign*.

direcció de la carpeta. Per exemple, posem posar `setwd("C:/RRII/Indicadors/R")` en PC i `setwd("H:\\myfolder\\data")` amb Mac. De manera semiautomàtica tenim l'opció `setwd(choose.dir())`.

Ja hem vist la utilitat de la segona pestanya, 'Plots'. Quan R crea un gràfic, el visualitzarem automàticament en aquesta finestra.

La pestanya 'Packages' és extremament important en els nostres primers passos amb R per a familiaritzar-nos amb el sistema de paquets (encara que a mesura que guanyem habilitat amb el programa l'anirem oblidant progressivament). En aquest apartat podem visualitzar tots els paquets instal·lats a RStudio de manera semblant a com hem fet quan visitàvem 'Global Environment'. La llista que veiem són els paquets que R té instal·lats, i hi trobem el nom del paquet, una breu descripció i la versió. També podem visualitzar-los a la consola amb `installed.packages()`. Que un paquet estigui en aquesta llista no vol dir que R l'estigui fent servir. R pot tenir un paquet instal·lat, però no carregat. Per tant, hem de diferenciar dos processos:

- *Paquets instal·lats*: els paquets que es troben a la llista que apareix a la finestra són els que RStudio té baixats. Això no vol dir que els estigui utilitzant.
  - De manera manual, podem instal·lar un paquet nou clicant a l'apartat 'Install'. Ens apareixerà una finestra on podem introduir el nom del paquet que volem que instal·li. Per a instal·lar els paquets bàsics que ens demanen a la guia de l'assignatura, buscarem *dplyr*, *ggplot2* i *tidyr* a la llista i, si no els trobem, anirem a l'apartat 'Install' per a instal·lar-los.
  - A l'script, podem utilitzar la funció `install.packages()` per a instal·lar paquets. El nom del paquet ha d'anar sempre entre cometes. Si volem instal·lar a la vegada més d'un paquet, ho haurem de fer amb el concatenat. Per a instal·lar els paquets que ens demanen a la guia de l'assignatura, utilitzarem: `install.packages(c("dplyr", "ggplot2", "tidyr"))`.
- *Paquets carregats*: perquè R pugui treballar amb el paquet, ha d'estar carregat. Si ens trobem amb problemes quan fem una determinada operació amb R, és molt possible que el paquet no estigui carregat. El podem carregar de dues maneres:
  - Manualment, activant el paquet que ens interessi de la llista amb el quadradet que apareix al seu costat. R ens el carregarà automàticament. Si no apareix a la llista l'haurem d'instal·lar (vegeu l'apartat anterior).
  - Una segona opció per a instal·lar i carregar els paquets és des de la consola o l'script. Per a fer-ho, utilitzarem la funció `library()`. No podem carregar diversos paquets a la vegada. Els haurem de carregar individualment. Per exemple: `library(dplyr)`, `library(ggplot2)`, `library(tidyr)`.

El millor amic d'un usuari d'R és la finestra 'Help'. En aquest apartat podrem demanar informació sobre un paquet, una funció o un objecte determinats. Només cal que posem el símbol `?` abans del nom del qual volem demanar informació. Per exemple, `?sum`, `?dplyr` o `?datasets`. En aquesta fase encara no ens serà de gran utilitat, però a mesura que ens familiaritzem amb R descobrirem el gran valor que té.

Finalment, tenim la finestra 'Viewer', que per ara no utilitzarem.

### 3. Tercer pas: els paquets

Les llibreries o paquets són els manuals que consulta R per a poder treballar. Per a fer un símil amb els prestatges de llibres que tenim a casa nostra, direm que després d'instal·lar R al nostre ordinador, el programa té al seu estudi una prestatgeria gegant amb la majoria de prestatges buits. Només hi té uns quants llibrets bàsics que ja venen incorporats al programa i que R ja se sap de memòria. Aquests llibrets són els que podem veure inicialment a la pestanya de Global Environment o bé si teclegem `search()`. Amb el que ha après d'aquests llibres, R sap fer operacions bàsiques com ara sumar i restar, crear gràfics senzills i algunes coses més.

Quan donem l'ordre `install.packages()`, el que fem és dir a R que vagi a buscar nous llibres i els col·loqui als prestatges de l'estudi. A mesura que instal·lem més llibres o paquets, la llibreria se'ns va omplint de material, encara que R només els té col·locats decorativament. Ocupen espai als prestatges (al nostre disc dur), però compte, tenir molts llibres no vol dir que R els hagi llegit! R no els llegirà fins que li ho indiquem.

El pas següent és dir a R que ens llegeixi de dalt a baix alguns dels llibres amb la funció `library()`. Els llibres són com manuals d'instruccions. Quan diem a R `library(jardineria)`, el programa llegeix el manual de jardineria amb dècimes de segon. De cop i volta, R ja sap absolutament tot el que s'ha de fer per a ser un bon jardiner. Si li diem `library(carnetB)`, R es traurà el carnet B de conduir en poques dècimes de segon i estarà preparat per a pilotar el cotxe que li indiquem.

La qüestió és, doncs, omplir la llibreria de llibres que necessitem per a tenir-los fàcilment a l'abast i indicar a R que llegeixi els manuals que utilitzarem per a fer les operacions que necessitarem. De base, R ja té alguns coneixements. Sap sumar, restar, fer diverses operacions, crear objectes, llegir bases de dades, dibuixar alguns gràfics..., però nosaltres li demanarem molt més. Principalment, li direm que aprengui a fer tres tasques. La més important és la de manipular bases de dades. R aprendrà a crear i eliminar columnes i files, ordenar files o fer operacions estadístiques més sofisticades. Això ens ho permetrà un llibre en el lloc del qual diu *dplyr*. Una segona tasca que també ens serà molt útil és la de dibuixar gràfics complexos. Això ens ho permetrà fer el manual *ggplot2*. Finalment, també utilitzarem *tidyr*, un paquet que manipula les files i les columnes de les bases de dades. Hi ha més paquets que utilitzarem, com ara *readxl*, *stringr* o *lubridate*, però ja en parlarem més endavant.

El CRAN és la nostra biblioteca central. Quan li diem `install.packages()`, R va a la biblioteca, agafa els llibres que li diem i els deixa al prestatge de l'estudi. Quan li diem `library()`, R va al prestatge i llegeix el manual que

li diem. En l'apartat anterior ja hem après a diferenciar entre aquests dos processos i també hem conegut les dues maneres que hi ha per a baixar paquets: per mitjà de la finestra 'Packages' o per codi mitjançant l'script o la consola. És possible que primer ens sentim més còmodes fent servir la finestra, però més endavant de ben segur que ens sentirem més còmodes amb el codi.

### **Diferència clau entre instal·lar i carregar**

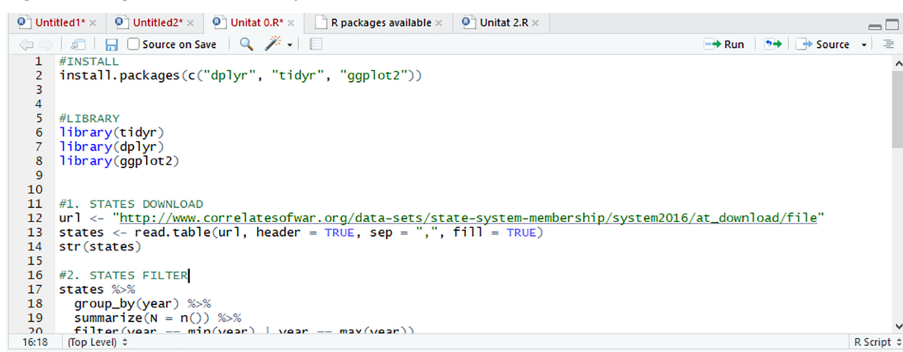
És molt important saber que quan instal·lem un paquet, R se'l guardarà en memòria i no necessitarem tornar-lo a instal·lar cada vegada que iniciem sessió a R (tret que n'aparegui una nova versió que vulguem utilitzar, cas en què sí que l'haurem de instal·lar de nou). En canvi, quan sortim d'R els paquets deixaran d'estar carregats. La propera vegada que tornem a executar el programa haurem de tornar a carregar els paquets. Per tant, tingueu en compte, de manera sistemàtica, d'entrar aquests tres codis sempre que entreu a R.

```
library(dplyr)
library(tidyr)
library(ggplot2)
```

## 4. Quart pas: script

Ara que ja coneixem les parts essencials de la interfície d'R, ja estem preparats per a començar a treballar amb els materials de l'assignatura. En aquest apartat utilitzarem, com a exemple, els scripts que figuren al mòdul inicial de l'assignatura *Fonts d'Informació i Indicadors en Estudis Internacionals*, encara que podrien valdre els de qualsevol assignatura. El primer que podem fer és organitzar-nos l'script al nostre gust i començar a entrar els codis que tenim als materials docents. A la figura 9 trobareu una proposta de com començar.

Figura 9. Organització de l'script.



```

1 #INSTALL
2 install.packages(c("dplyr", "tidyr", "ggplot2"))
3
4
5 #LIBRARY
6 library(tidyr)
7 library(dplyr)
8 library(ggplot2)
9
10
11 #1. STATES DOWNLOAD
12 url <- "http://www.correlatesofwar.org/data-sets/state-system-membership/system2016/at_download/file"
13 states <- read.table(url, header = TRUE, sep = ",", fill = TRUE)
14 str(states)
15
16 #2. STATES FILTER|
17 states %>%
18   group_by(year) %>%
19   summarize(N = n()) %>%
20   filter(year == min(year) | year == max(year))

```

En primer lloc, hem posat un títol d'encapçalament precedit pel coixinet (#). Com sabeu, quan RStudio veu que hi ha un coixinet al davant d'una línia entén que no ha de llegir res a partir del símbol del que hi hagi a la mateixa línia. A nosaltres, en canvi, ens anirà bé per a fer-nos anotacions a l'script. A #INSTALL instal·lem els tres paquets bàsics que utilitzarem. Recordeu que els paquets s'instal·len amb cometes. Molt de compte si copiem i enganxem codis amb cometes des de programes com ara Word. Normalment aquests programes arrodoneixen o inclinen les cometes<sup>4</sup>. Si enganxem aquestes cometes modificades a RStudio, el programa no entendreà quin símbol són i ens donarà error.

<sup>(4)</sup>Fixeu-vos en la diferència entre escriure "dplyr" o "dplyr". R reconeixerà el primer terme però ens donarà error en el segon cas.

Per a instal·lar els paquets hem d'estar connectats a internet, ja que R va a buscar-los a la biblioteca CRAN. Veurem que la instal·lació d'alguns paquets pot trigar una estona, en especial la de *ggplot2*. Paciència. Sabreu si R encara està processant si hi ha un senyal vermell de stop al marge superior esquerre de la finestra de la consola.

Un cop instal·lats els paquets, els haurem de carregar o activar. Podem fer la mateixa operació que abans i marcar amb un coixinet # l'inici d'un altre procés. En aquest cas, hem posat #PACKAGES. Els paquets s'han de carregar per separat: `library(tidyr)`, `library(dplyr)` i `library(ggplot2)`. Normal-

ment només necessitarem instal·lar una sola vegada els paquets. Després ja es queden guardats al nostre ordinador. El que sí que haurem de fer és carregar-los cada cop que iniciem una nova sessió.

La millor manera de comprovar que R ens ha instal·lat correctament el paquet serà utilitzant-los. A la taula 3 s'indica com comprovar que els paquets *ggplot2* i *dplyr* funcionen amb dues operacions fetes mitjançant funcions d'aquests paquets: crearem un histograma amb *ggplot2* i filtrarem unes dades amb *dplyr*.

Taula 3. Prova dels paquets *ggplot2* i *dplyr*.

<b>Prova ggplot2</b> <code>qplot(mtcars\$mpg)</code>
<b>Prova dplyr</b> <code>filter(mtcars, mpg &gt; 25)</code>

Si en el primer codi ens apareix un gràfic significarà que tenim *ggplot2* funcionant correctament. Si en el segon codi ens apareix a la consola una taula de sis files també significarà que el paquet *dplyr* funciona correctament.

Amb les indicacions donades fins aquí ja podeu repetir les instruccions del manual de l'assignatura al vostre ordinador. Personalitzeu l'script a la vostra manera i aneu-hi introduint les línies de comandament.

## 5. Cinquè pas: possibles errors

No hi ha programador o analista de dades, per més que en sàpiga, que no topi amb desenes d'errors quan utilitza R. Les hores invertides en el programa i la nostra familiarització amb l'entorn, manuals d'ajuda o fòrums determinaran la nostra capacitat de poder resoldre aquests errors. Per tant, al principi haurem de tenir molta paciència, ja que encara tindrem pocs recursos per a poder resoldre les situacions que ens sobrevinguin.

A la taula 4 s'ofereix una guia dels problemes més habituals amb què l'usuari es pot trobar en les primeres hores d'R. Òbviament, no és una llista exhaustiva, però ens pot ajudar a identificar i resoldre més d'un error.

Taula 4. Errors principals que us podeu trobar amb R.

1	No haver instal·lat RStudio ni R. Treballarem amb RStudio, però també haurem de tenir instal·lat R perquè RStudio funcioni adequadament.
2	No reconeix paràmetres com <code>%&gt;%</code> , <code>group_by()</code> o <code>summarize()</code> . No està instal·lat el paquet <code>dplyr</code> .
3	Alguns errors es donen perquè no tenim les llibreries corresponents instal·lades o baixades. Comprova que tens totes les llibreries necessàries instal·lades i també carregades per a fer l'exercici. Pots comprovar a la finestra inferior dreta 'Packages' quines són les llibreries instal·lades (les que figuren a la llista de la finestra) i quines són les que estan carregades (les que tenen un signe ✓ al costat).
4	Si copiem les dades primer en un Word i després a RStudio ens podem trobar un problema de format. Per exemple, Word modifica les cometes " _ " dels codis per a fer-les més arrodonides o inclinades, i en aquest format RStudio no n'entendrà el significat.
5	És un error molt comú copiar les dades de manera inexacta. Fixa't bé en l'orientació de les barres, ja que no és el mateix \ que /. Fes un recompte del nombre de parèntesis que obris en una línia de codi, ja que per cada parèntesi ( que obrim n'hi haurà d'haver un altre ) que tanqui.
6	Quan a la consola ens surt + és que R espera instruccions, que no hem acabat bé el codi i que possiblement falta tancar un parèntesi o unes cometes.
7	No posis mai un # davant d'una línia de codi. Les línies de codi que R ha d'interpretar han d'estar lliures al davant.
8	R és sensible a les majúscules. "SPAIN" no és el mateix que "Spain". Tecleja "SPAIN" == "Spain" i observa el resultat.

A mesura que anem guanyant més experiència amb R, cada cop serem més capaços d'entendre quin és l'error si llegim el text que R ens retorna a la consola. Per exemple, amb "unexpected symbol" o "unexpected end of input" és molt possible que hi hagi algun problema de tecleig en el codi i que hàgim de revisar que tots els parèntesis estiguin tancats, els caràcters entre



cometes i els noms dels objectes i les funcions entrats correctament. Si a la consola apareix un “+” és perquè R espera que afegim algun comandament més per a completar el codi.

## 6. Glossari

### Glossari de dreceres amb el teclat (PC)

Alt+-	Drecera per <-
Ctrl+Enter	Executa una línia de codi a l'script
Ctrl+S	Guarda l'script
Ctrl+Shift+Enter	Executa tot el codi de l'script
Ctrl+Shift+N	Crea un script
Enter	Executa una línia de codi a la consola

### Glossari de funcions i operadors del mòdul

<-	Crea un objecte
?	Obre el quadre d'ajuda d'una funció o objecte
??	Fes una cerca per paraula clau
\$	Estableix la separació entre el marc de dades i la variable
c()	Crea un vector
dir()	Mostra els arxius del directori de treball
getwd()	Mostra la ubicació del directori de treball
help()	Obre el quadre d'ajuda d'una funció o objecte
installed.packages()	Mostra les llibreries instal·lades
install.packages()	Instal·la una o més llibreries
library()	Mostra els paquets instal·lats
ls()	Mostra els objectes creats
objects()	Mostra els objectes creats
remove()	Elimina un objecte creat
rm()	Elimina un objecte creat
rm(list=ls())	Elimina tots els objectes que hem creat
sample()	Retorna una mostra aleatòria
search()	Mostra els paquets carregats a R.
setwd()	Estableix la ubicació del directori de treball



## Resum

En aquest mòdul hem vist com introduir-nos a RStudio a través de cinc passos principals. Els continguts que hem estudiat en les pàgines anteriors no pretenen ser una guia exhaustiva sobre l'ús del programa, sinó que tenen com a objectiu principal que l'estudiant es pugui moure per RStudio amb suficient habilitat per a poder treballar els materials i les activitats d'una assignatura que utilitzi l'entorn d'R com a eina docent.

És important completar aquest aprenentatge amb altres materials, principalment amb vídeos que trobareu disponibles a l'aula i també amb els recursos que cregueu necessaris. Tingueu en compte que R és un programari lliure amb una comunitat d'usuaris molt gran i que hi ha moltíssims llocs a la xarxa on podeu trobar tutorials o guies d'ajuda.