

BIOS-IN5410

Introduction to R programming

Learning goals

Introduce you to R and Rstudio

Basic R functionality

Find and install packages

Be able to read package manuals and find help

Read and write files

Plotting data

(Very rough) time plan

Friday Nov 15

13:15-14:00

- Introduction to R and RStudio
- Set up and get going
- Do Exercise 1

14:15 - 16:00

- Go through Exercise 1
- R packages and the Tidyverse
- Rectangular and tidy data
- Working with files
- Exercise 2
- Go through Exercise 2

Thursday Nov 21

09:15 - 10:30

- Manipulating data with dplyr
- Exercise 3

10:45 - 12:30

- Go through Exercise 3
- Basic plotting
- Exercise 4
- Go through exercise 4 together

13:00 - 17:00

- Programming basics
 - For loops + Ex 5 (13:00 - 14:15)
 - Ex 5 + If statements + Ex 6 (14:30 - 15:30)
 - Go through exercise 6 (15:45 - 16:15)
- Wrap-up

Friday Nov 22

09:15 – 12:00

- R scripts
 - Running R on the command line
 - Command line arguments
- Plotting with ggplot2 (not curriculum – brief demo + exercise)

R resources

Introduction to Data Science - free online book (most of the material in this course is taken from here): <https://rafalab.github.io/dsbook/>

R for Data Science - free online book: [R for Data Science \(2e\)](#)

Software Carpentry - <https://swcarpentry.github.io/r-novice-gapminder/>

Course material and exercises: [jonbra/BIOS-IN5410: Repository for the R lectures in BIOS-IN5410](#)

The R project

Environment for statistical computing and graphics

It's free

Can be run on Windows, Mac, Unix...

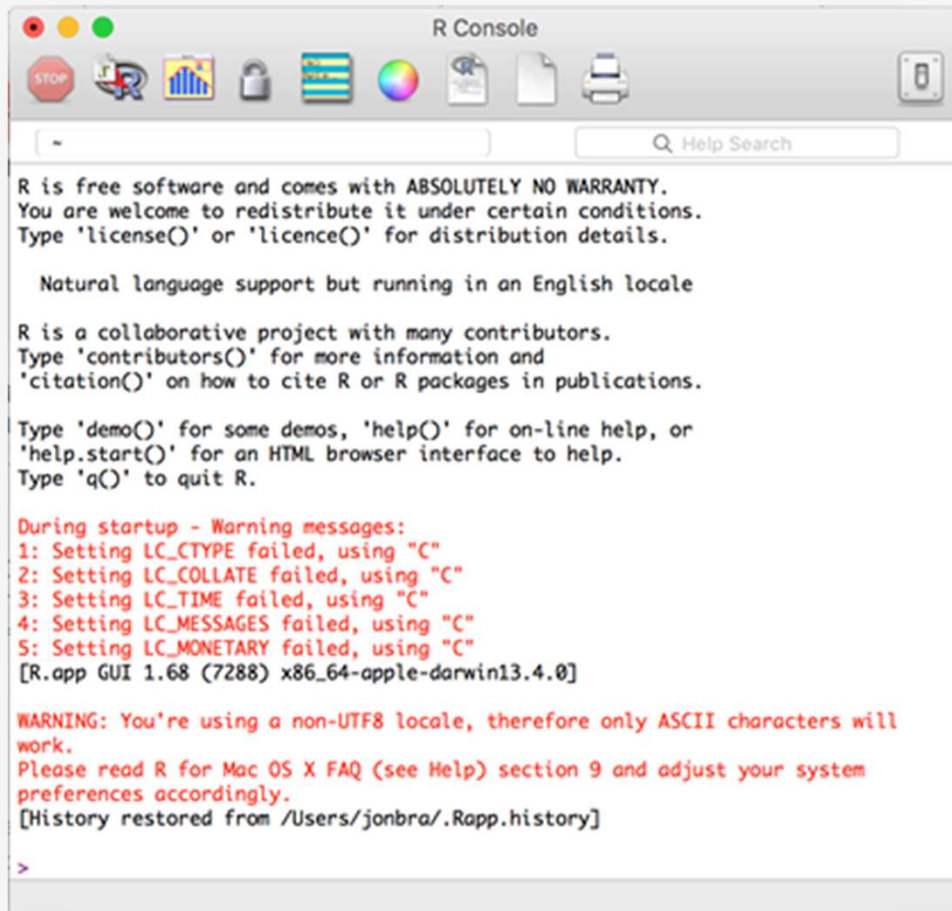
Extremely rich selection of packages

Analyze data, statistics, ...

Very good for graphics and plotting



The R console



```
R Console

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

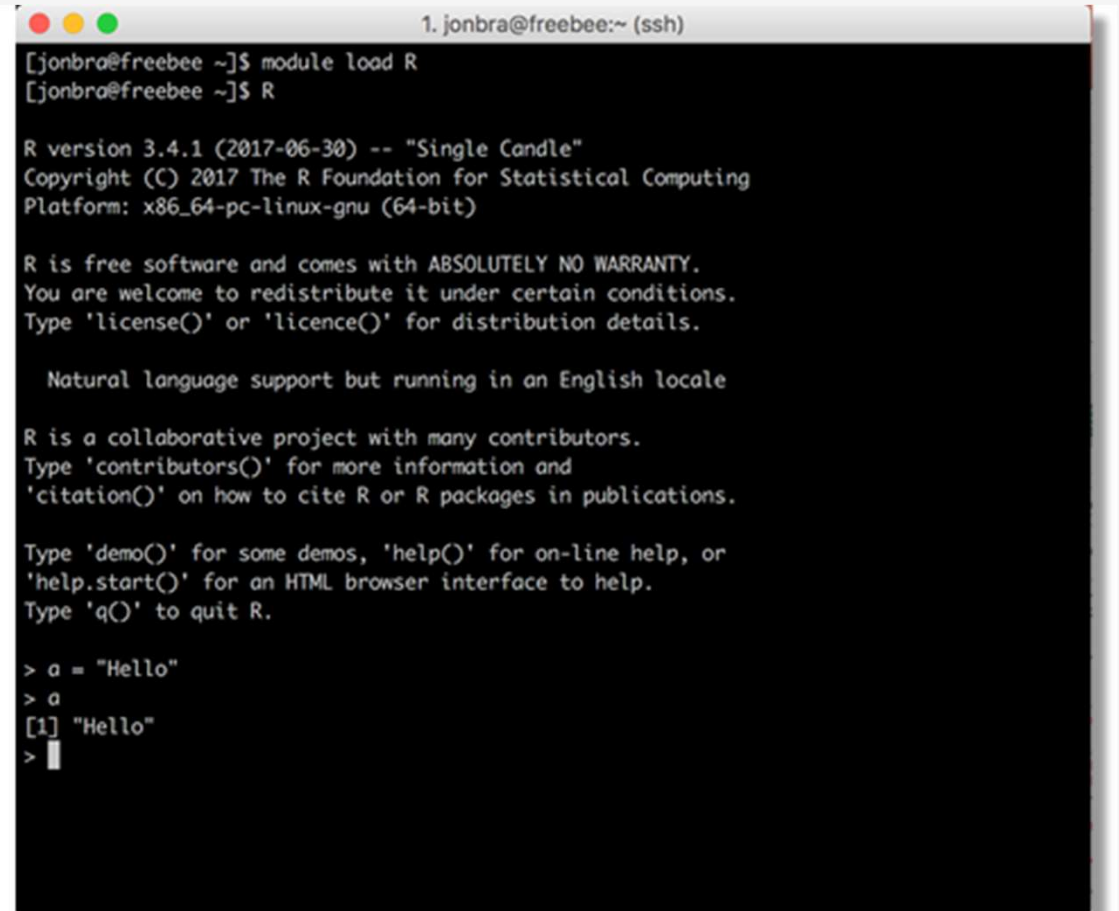
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

During startup - Warning messages:
1: Setting LC_CTYPE failed, using "C"
2: Setting LC_COLLATE failed, using "C"
3: Setting LC_TIME failed, using "C"
4: Setting LC_MESSAGES failed, using "C"
5: Setting LC_MONETARY failed, using "C"
[R.app GUI 1.68 (7288) x86_64-apple-darwin13.4.0]

WARNING: You're using a non-UTF8 locale, therefore only ASCII characters will
work.
Please read R for Mac OS X FAQ (see Help) section 9 and adjust your system
preferences accordingly.
[History restored from /Users/jonbra/.Rapp.history]

>
```



```
1. jonbra@freebee:~ (ssh)

[jonbra@freebee ~]$ module load R
[jonbra@freebee ~]$ R

R version 3.4.1 (2017-06-30) -- "Single Candle"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

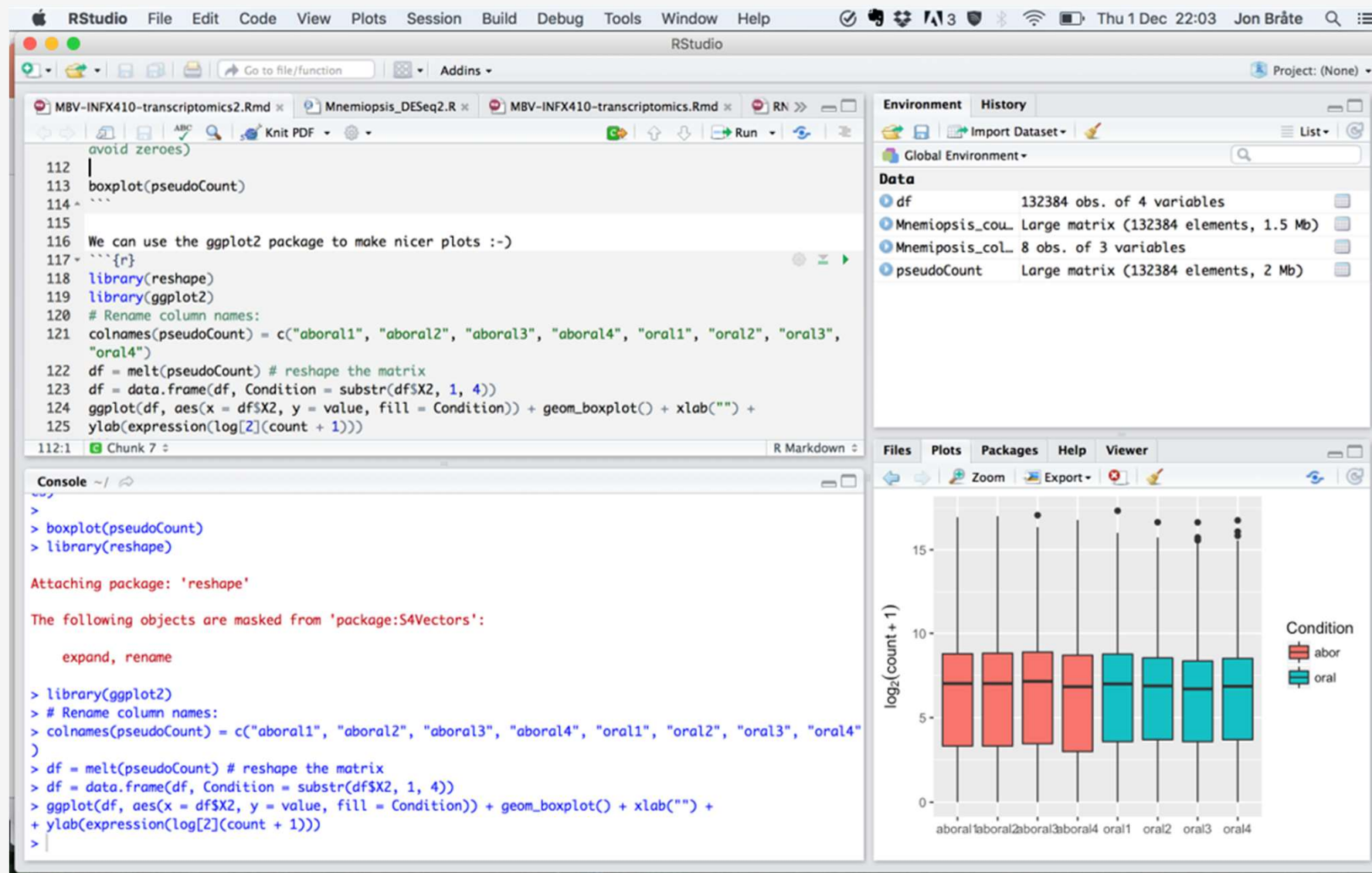
Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> a = "Hello"
> a
[1] "Hello"
> █
```

RStudio - an R IDE



RStudio - an R IDE

Text editor

```
RStudio File Edit Code View Plots Session Build Debug Tools Window Help
RStudio
Go to file/function Addins
MBV-INF410-transcriptomics2.Rmd x Mnemiposis_DESeq2.R x MBV-INF410-transcriptomics.Rmd x RN >>
avoid zeroes)
112 |
113 boxplot(pseudoCount)
114 ~~~
115
116 We can use the ggplot2 package to make nicer plots :-)
117 ~~~{r}
118 library(reshape)
119 library(ggplot2)
120 # Rename column names:
121 colnames(pseudoCount) = c("aboral1", "aboral2", "aboral3", "aboral4", "oral1", "oral2", "oral3",
122 "oral4")
122 df = melt(pseudoCount) # reshape the matrix
123 df = data.frame(df, Condition = substr(df$X2, 1, 4))
124 ggplot(df, aes(x = df$X2, y = value, fill = Condition)) + geom_boxplot() + xlab("") +
125 ylab(expression(log[2](count + 1)))
112:1 Chunk 7 R Markdown
```

Environment window

Environment History

Global Environment

Data

- df 132384 obs. of 4 variables
- Mnemiposis_cou... Large matrix (132384 elements, 1.5 Mb)
- Mnemiposis_col... 8 obs. of 3 variables
- pseudoCount Large matrix (132384 elements, 2 Mb)

Console

```
Console ~/
>
> boxplot(pseudoCount)
> library(reshape)

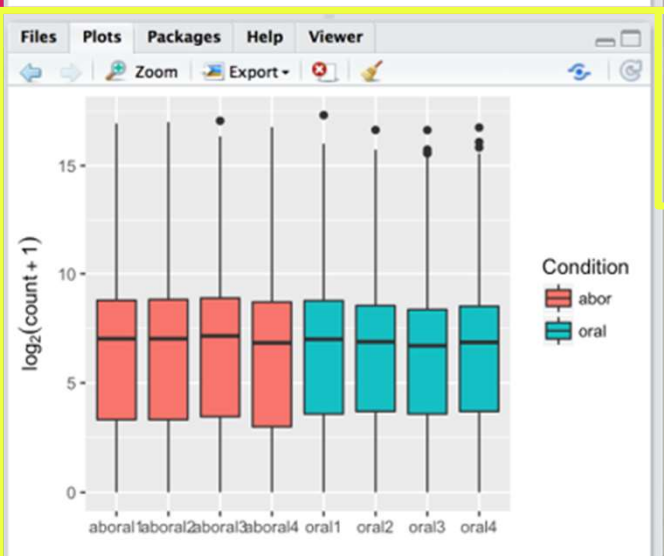
Attaching package: 'reshape'

The following objects are masked from 'package:S4Vectors':

    expand, rename

> library(ggplot2)
> # Rename column names:
> colnames(pseudoCount) = c("aboral1", "aboral2", "aboral3", "aboral4", "oral1", "oral2", "oral3", "oral4")
> df = melt(pseudoCount) # reshape the matrix
> df = data.frame(df, Condition = substr(df$X2, 1, 4))
> ggplot(df, aes(x = df$X2, y = value, fill = Condition)) + geom_boxplot() + xlab("") +
+ ylab(expression(log[2](count + 1)))
>
```

View plots, packages, files, help and more



RStudio - cheat sheet

Check out the [RStudio cheat sheet](#) in the GitHub repo - especially the shortcuts.

Keyboard Shortcuts

RUN CODE

	Windows/Linux	Mac
Search command history	Ctrl+↑	Cmd+↑
Interrupt current command	Esc	Esc
Clear console	Ctrl+L	Ctrl+L

NAVIGATE CODE

	Windows/Linux	Mac
Go to File/Function	Ctrl+.	Ctrl+.

WRITE CODE

Attempt completion	Tab or Ctrl+Space	Tab or Ctrl+Space
Insert <- (assignment operator)	Alt+-	Option+-
Insert %>% (pipe operator)	Ctrl+Shift+M	Cmd+Shift+M
(Un)Comment selection	Ctrl+Shift+C	Cmd+Shift+C

MAKE PACKAGES

	Windows/Linux	Mac
Load All (devtools)	Ctrl+Shift+L	Cmd+Shift+L
Test Package (Desktop)	Ctrl+Shift+T	Cmd+Shift+T
Document Package	Ctrl+Shift+D	Cmd+Shift+D

DOCUMENTS AND APPS

Knit Document (knitr)	Ctrl+Shift+K	Cmd+Shift+K
Insert chunk (Sweave & Knitr)	Ctrl+Alt+I	Cmd+Option+I
Run from start to current line	Ctrl+Alt+B	Cmd+Option+B

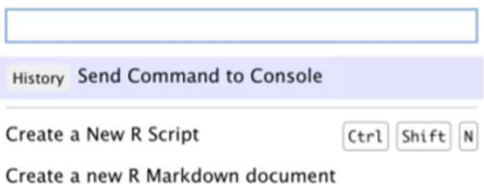
MORE KEYBOARD SHORTCUTS

Keyboard Shortcuts Help	Alt+Shift+K	Option+Shift+K
Show Command Palette	Ctrl+Shift+P	Cmd+Shift+P

View the Keyboard Shortcut Quick Reference with **Tools > Keyboard Shortcuts** or **Alt/Option + Shift + K**



Search for keyboard shortcuts with **Tools > Show Command Palette** or **Ctrl/Cmd + Shift + P**.



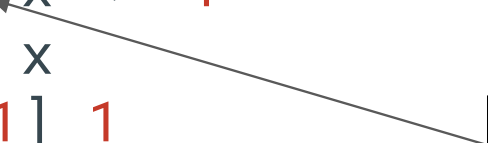
A (super) short introduction to R functionality

(you don't need to remember all the details. Use the slides as
a reference)

Variable assignment

We assign values to variables with the assignment operator "<-" (can also use "="). Just typing the variable by itself at the prompt will print out the value.

```
> x <- 1
> x
[1] 1
> x = 1
> x
[1] 1
> y <- 2
> x + y
[1] 3
```



The prompt (like the \$ in the Unix terminal)

R is very good for mathematics

```
> 1+1 # Simple arithmetic
[1] 2
> 2 + 3 * 4 # Operator precedence
[1] 14
> 3 ^ 2 # Exponentiation
[1] 9
> exp(1) # Basic mathematical functions are available
[1] 2.718282
> sqrt(10)
[1] 3.162278
> pi # The constant pi is predefined
[1] 3.141593
> 2*pi*6378 # Circumference of earth at equator (in km)
[1] 40074.16
```

Functions

R functions are invoked by its name, then followed by the parenthesis, and zero or more arguments. The following apply the function `c()` to combine three numeric values into a vector.

```
> c(1, 2, 3)
[1] 1 2 3
```

Function name

Arguments (separated by comma)

Comments

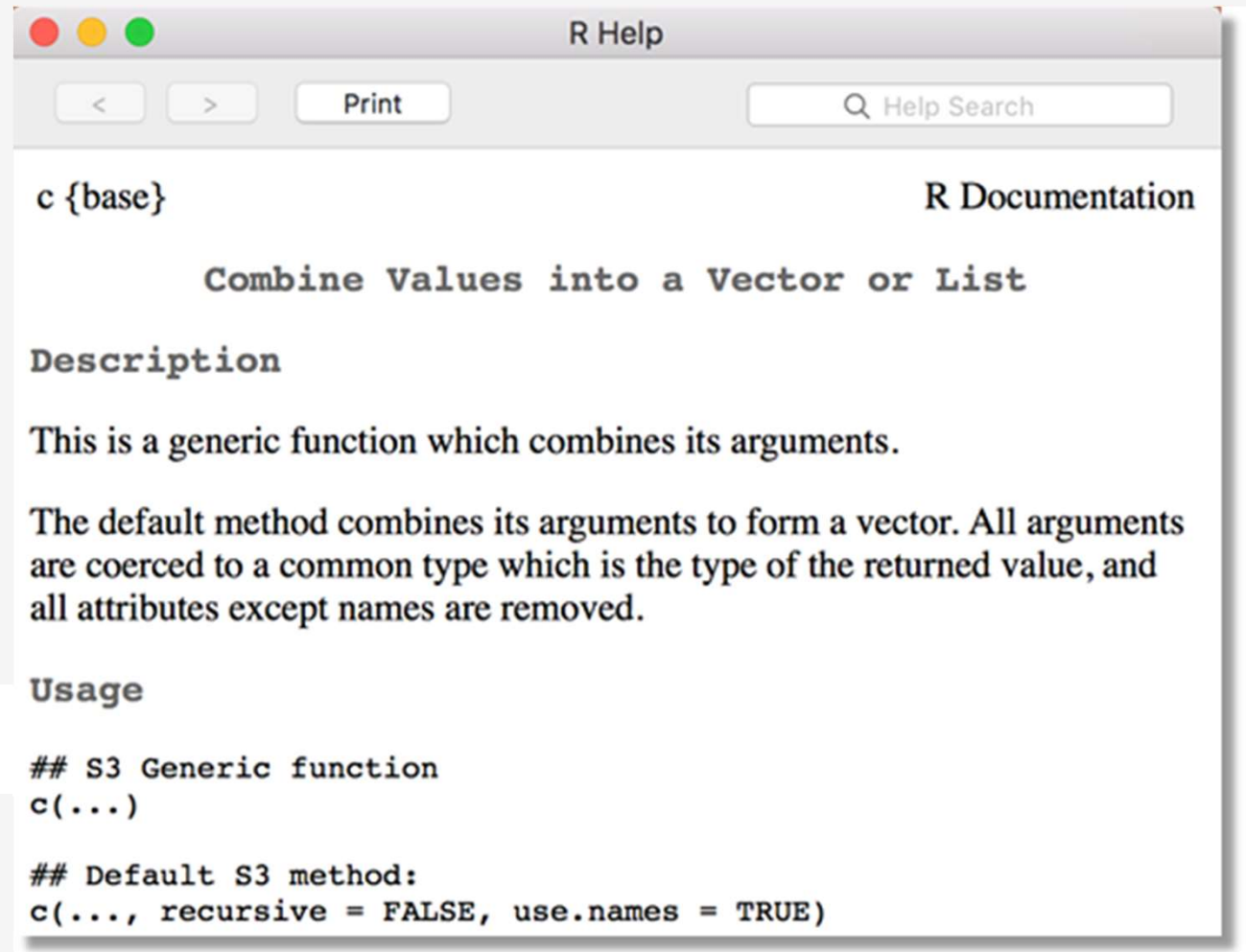
Just like in unix/bash, all text after the hash tag "#" within the same line is considered a comment.

```
> 1 + 1 # This is a comment  
[1] 2
```

Getting help

R provides extensive documentation. For example, entering `?c` or `help(c)` at the prompt gives documentation of the function `c` in R.

```
> help(c)
```



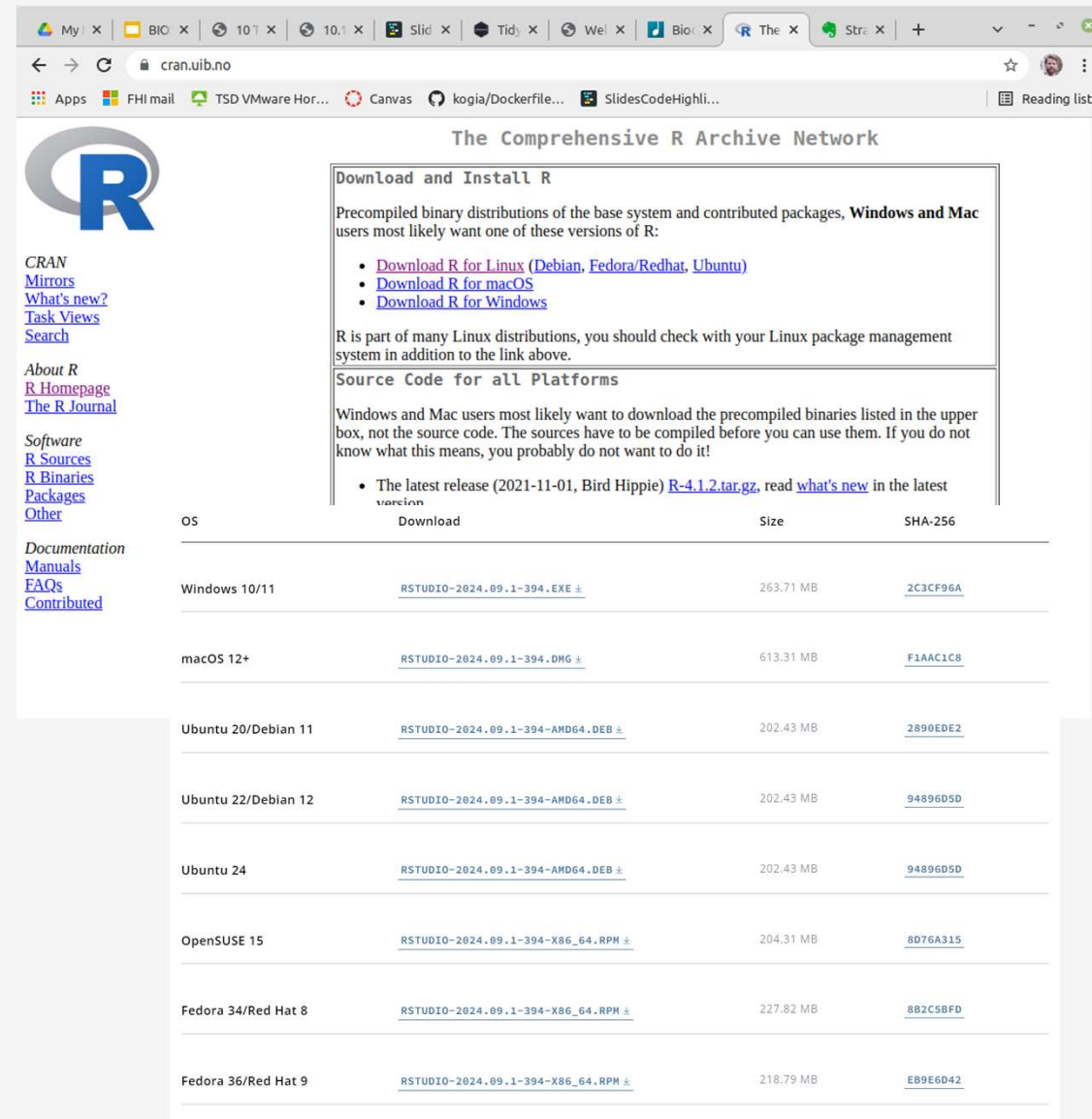
Get started with R

Install R (r-project.org)

cran.uib.no

Choose the right OS

Download and install RStudio ([RStudio Desktop - Posit](https://www.rstudio.com/products/rstudio/download/#download)). Choose the right OS



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 \(Debian, Fedora/Redhat, Ubuntu\)](#)
- [Download R for macOS](#)
- [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.

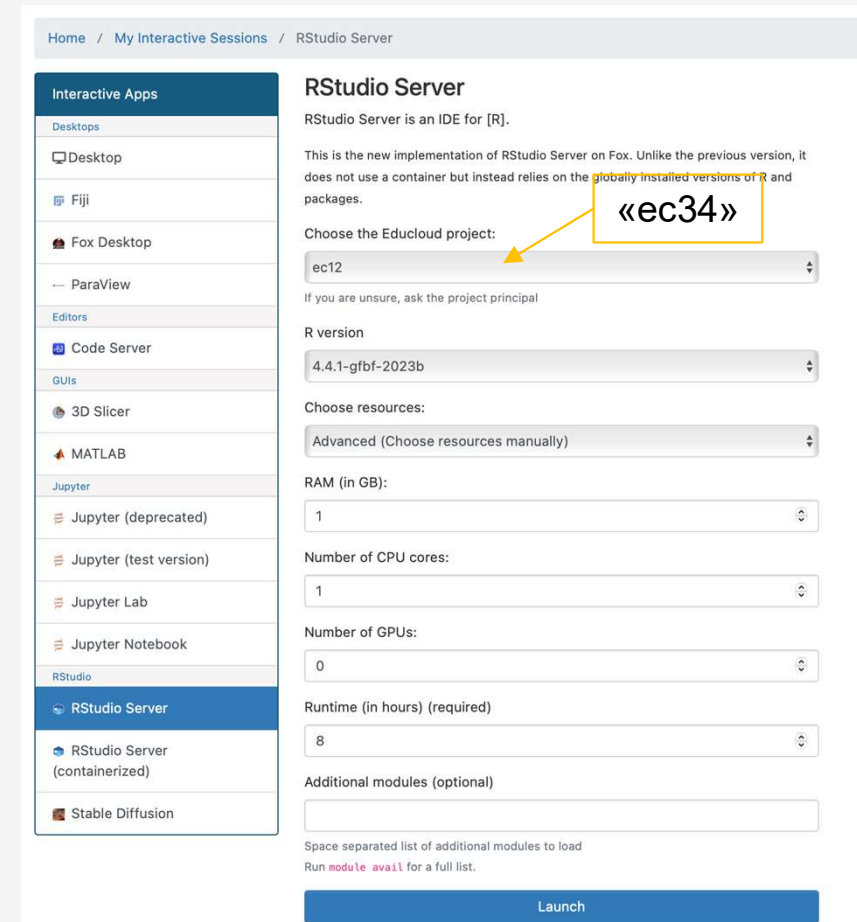
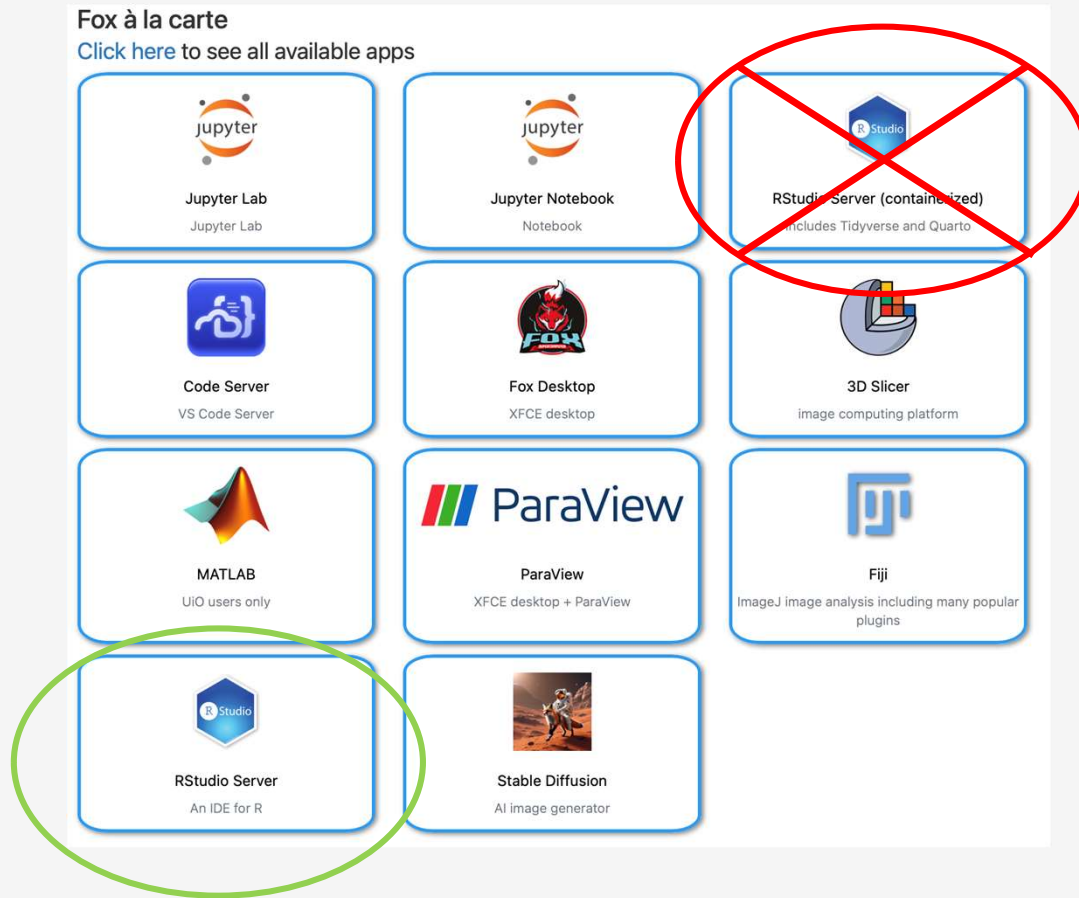
Source Code for all Platforms

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 before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2021-11-01, Bird Hippie) [R-4.1.2.tar.gz](#), read [what's new](#) in the latest version

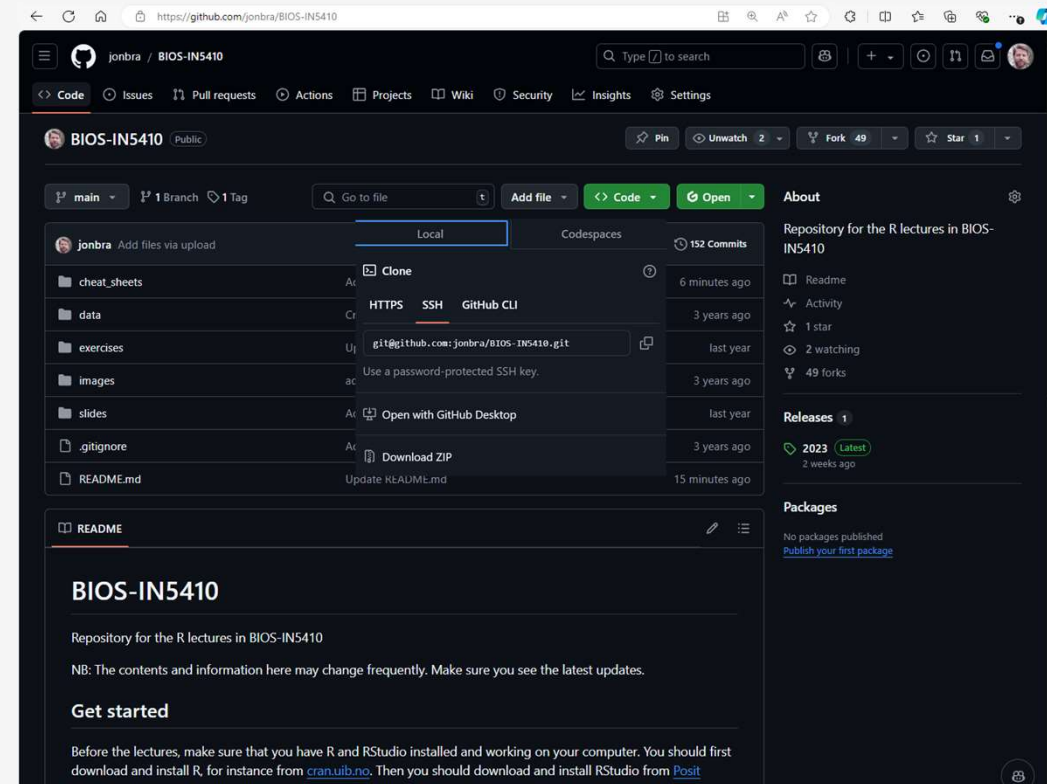
OS	Download	Size	SHA-256
Windows 10/11	RSTUDIO~2024.09.1~394.EXE ±	263.71 MB	2C3CF96A
macOS 12+	RSTUDIO~2024.09.1~394.DMG ±	613.31 MB	F1AAC1C8
Ubuntu 20/Debian 11	RSTUDIO~2024.09.1~394-AMD64.DEB ±	202.43 MB	2890EDE2
Ubuntu 22/Debian 12	RSTUDIO~2024.09.1~394-AMD64.DEB ±	202.43 MB	94896D5D
Ubuntu 24	RSTUDIO~2024.09.1~394-AMD64.DEB ±	202.43 MB	94896D5D
OpenSUSE 15	RSTUDIO~2024.09.1~394-X86_64.RPM ±	204.31 MB	8076A315
Fedora 34/Red Hat 8	RSTUDIO~2024.09.1~394-X86_64.RPM ±	227.82 MB	8B2C5BFD
Fedora 36/Red Hat 9	RSTUDIO~2024.09.1~394-X86_64.RPM ±	218.79 MB	EB9E6D42

Experimental: use R and Rstudio on Educcloud



Time to try R for yourself

- Make sure R and RStudio is installed and working.
- Test writing and executing commands, both in the editor and the console.
- Try to assign some variables, change them, etc.
- Download a copy of the GitHub repo (either by *git clone* or downloading a zip file)
- Do [Exercise 1](#) in your repo (we will always go through the exercises together).
- And just play around in R and RStudio (e.g. check out the cheat sheet).
- ***And help each other! I haven't given you all the details you need so you need to check the help menus and search the web.***



First break

Go through exercise 1

R-packages

In addition to “base R”, there are thousands of so-called “packages” (libraries) that gives additional functionality to R.

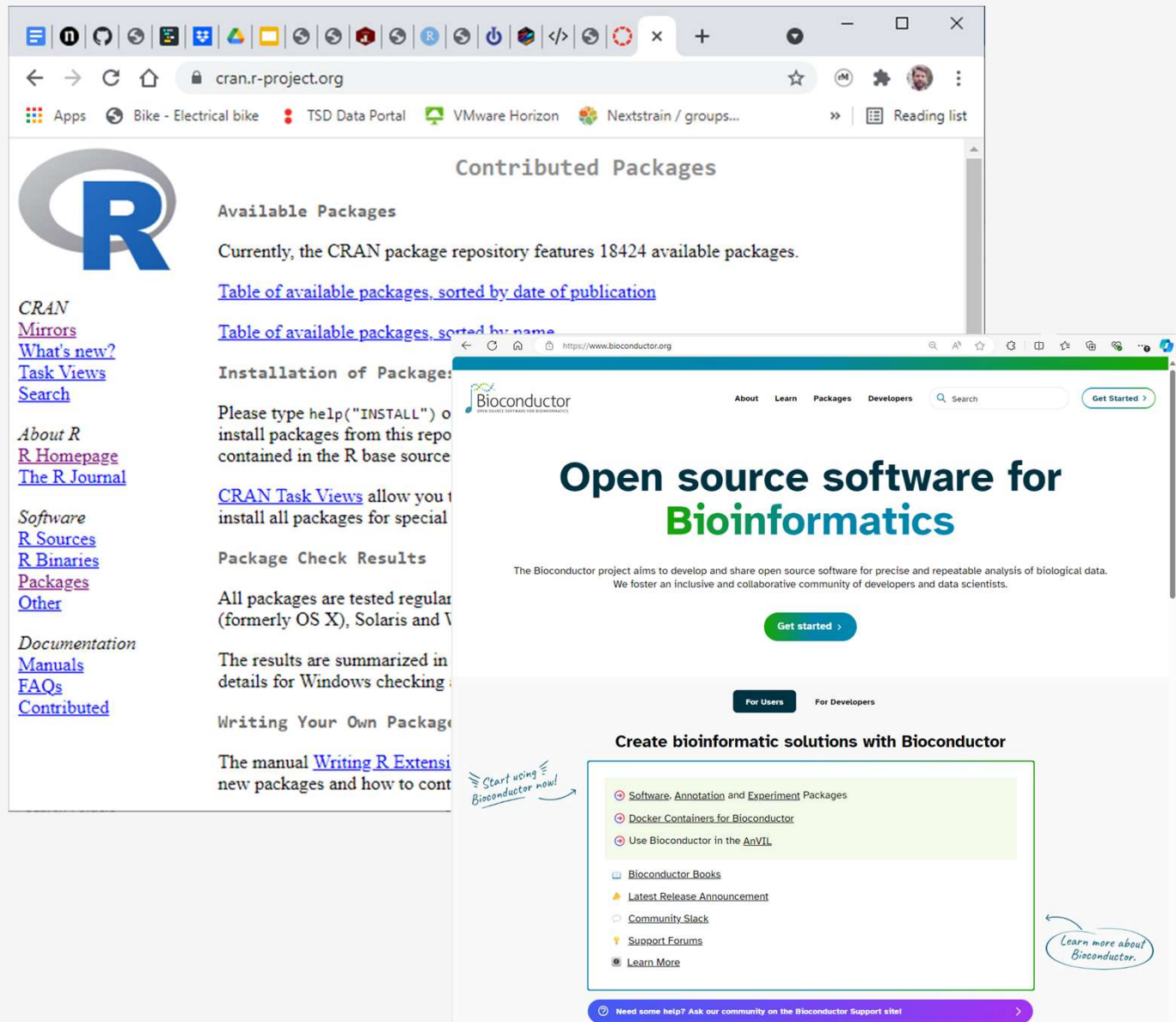
CRAN and Bioconductor are the main repositories for packages.

Packages needs to be installed, e.g. by typing

```
install.packages("package")
```

And activated before use by typing

```
library("package")
```



The image shows two web browser windows. The top window is the CRAN (Comprehensive R Archive Network) website at cran.r-project.org. It features the R logo, a sidebar with links like 'CRAN Mirrors', 'What's new?', 'Task Views', and 'Search', and a main section titled 'Contributed Packages' with 'Available Packages' and 'Installation of Packages' subsections. The bottom window is the Bioconductor website at <https://www.bioconductor.org>. It has a green header with navigation links, a large 'Open source software for Bioinformatics' title, and a 'Get started' button. Below this, it lists resources for users and developers, including 'Software, Annotation and Experiment Packages', 'Bioconductor Books', and 'Support Forums'. Handwritten blue annotations are present: 'Start using Bioconductor now!' with an arrow pointing to the 'Get started' button, and 'Learn more about Bioconductor.' with an arrow pointing to the 'Learn More' link in the Bioconductor resources list.

Tidyverse

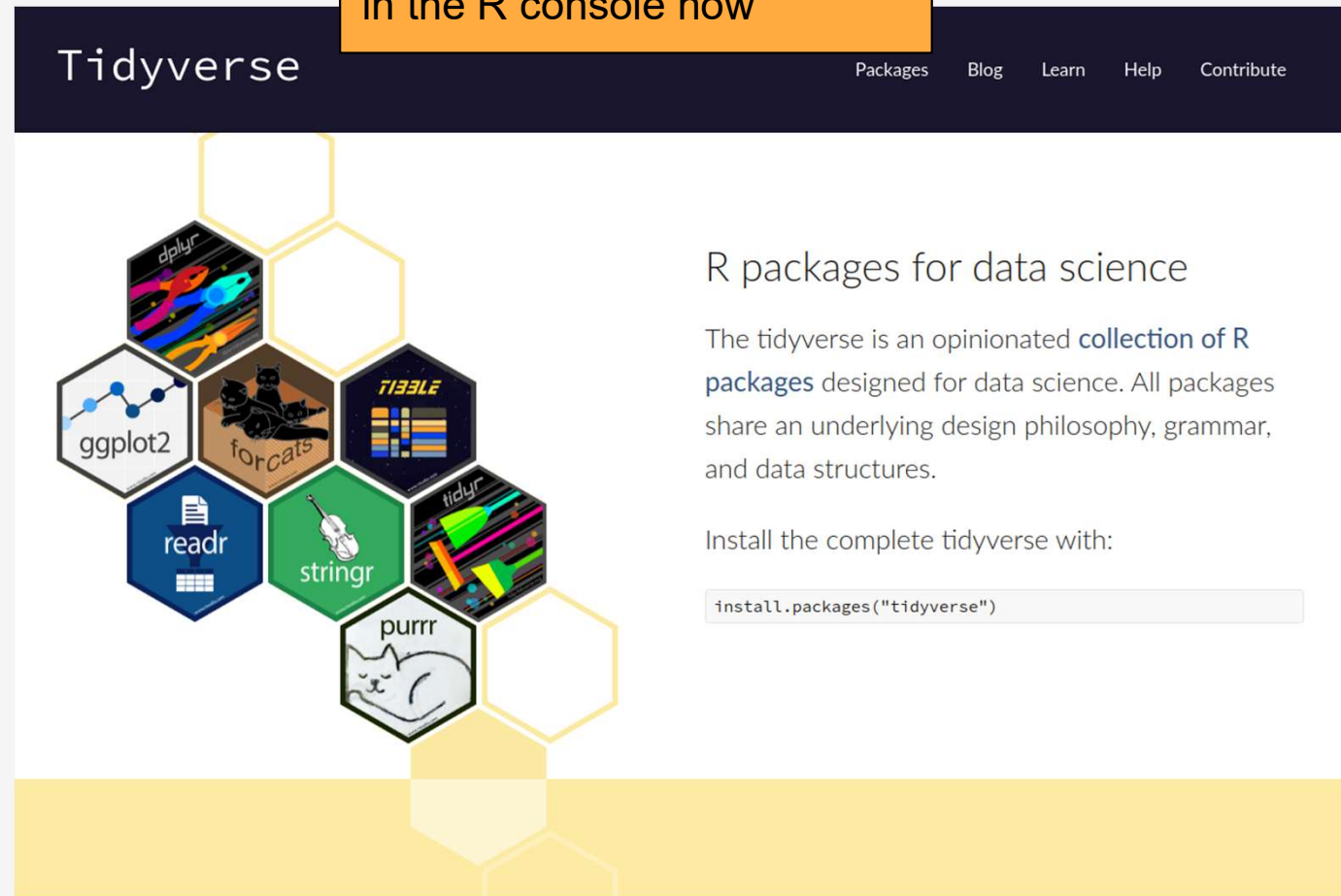
*“A system of packages for **data manipulation, exploration and visualization** that share a common design philosophy.”*

Centered around “Rectangular data structures” (e.g. data frames, matrices..)

tidyverse.org

```
install.packages("tidyverse")
```

Everyone should try to run
`install.packages("tidyverse")`
in the R console now



Free online book for learning R and the tidyverse: [R for Data Science \(2e\)](#)

The rectangular data type

A lot of the work you will do in R is centered around “rectangular data”, or data frames. Data frames are like tables with each row is a record and the columns are the different variables.

different variables.

Columns

Header

←

state

abb

region

population

total

Rows

1

Alabama

AL

South

4779736

135

2

Alaska

AK

West

710231

19

3

Arizona

AZ

West

6392017

232

4

Arkansas

AR

South

2915918

93

5

California

CA

West

37253956

1257

6

Colorado

CO

West

5029196

65

Tidy data

1. Each variable is a column;
each column is a variable.

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	127291272
China	2000	213766	128042583

variables

2. Each observation is a row;
each row is an observation.

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	127291272
China	2000	213766	128042583

observations

3. Each value is a cell; each cell
is a single value.

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	127291272
China	2000	213766	128042583

values

Contain all *values* that measure the same underlying attribute (e.g., country, year...).

An *observation* contains all *values* measured on the same unit (e.g., country) across attributes (notice multiple observations on the same row).

Strings (text) or numbers. Belong to a *variable* and an *observation*.

"tidy datasets are all alike but every messy dataset is messy in its own way."

<https://www.jstatsoft.org/article/view/v059i10>

R for Data Science, Hadley Wickham

Tidy data

We say that a data table is in ***tidy format*** if each row represents *one observation* and columns represent the different *variables* available for each of these observations.

Each variable forms a column

One observation per row

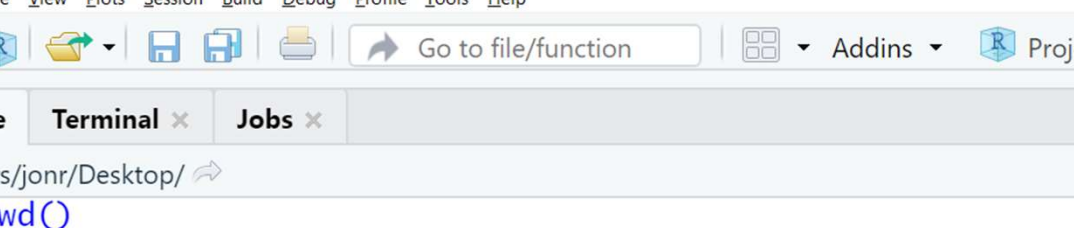
	country	year	fertility
1	Germany	1960	2.41
2	South Korea	1960	6.16
3	Germany	1961	2.44
4	South Korea	1961	5.99
5	Germany	1962	2.47
6	South Korea	1962	5.79

Multiple observations per row

	country	1960	1961	1962
1	Germany	2.41	2.44	2.47
2	South Korea	6.16	5.99	5.79

Working directory

The `getwd()` function let's you see where on your file system R is currently working. Change the working directory with `setwd()`.



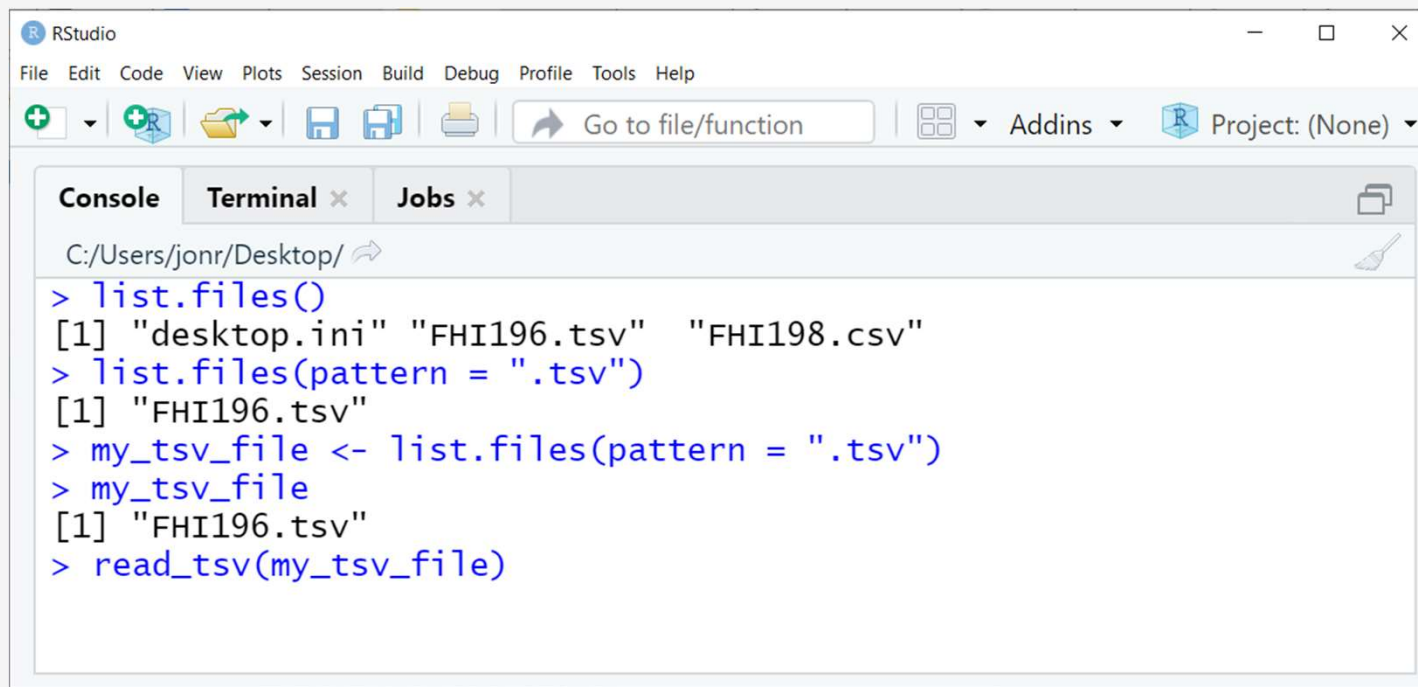
The screenshot shows the RStudio application window. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with icons for creating a new file, opening a file, saving, and other functions. The main workspace area is currently empty. The bottom pane is divided into three tabs: Console, Terminal, and Jobs. The Terminal tab is active, showing a command prompt with the following text:

```
C:/Users/jonr/Desktop/
```

```
> getwd()
[1] "C:/Users/jonr/OneDrive - Folkehelsetestitutet/Documents"
> setwd("C:/Users/jonr/Desktop/")
> getwd()
[1] "C:/Users/jonr/Desktop"
>
```

File system - access files

list.files() and *list.dirs()* will show the files and the directories in the working directory. Use the *pattern* argument to filter what kind of files or directories to be listed.

A screenshot of the RStudio application window. The title bar says 'RStudio'. The menu bar includes 'File', 'Edit', 'Code', 'View', 'Plots', 'Session', 'Build', 'Debug', 'Profile', 'Tools', and 'Help'. The toolbar contains icons for creating a new file, opening a file, saving, and other standard file operations, along with a search bar labeled 'Go to file/function'. Below the toolbar, there are tabs for 'Console', 'Terminal', and 'Jobs'. The 'Console' tab is active, showing the current working directory as 'C:/Users/jonr/Desktop/'. The console contains the following R code and its output:

```
> list.files()
[1] "desktop.ini" "FHI196.tsv"  "FHI198.csv"
> list.files(pattern = ".tsv")
[1] "FHI196.tsv"
> my_tsv_file <- list.files(pattern = ".tsv")
> my_tsv_file
[1] "FHI196.tsv"
> read_tsv(my_tsv_file)
```

Getting data into R - the readr package

There are many ways of getting data from files into R. The [readr](#) package offers several functions for reading different data types.

`read_csv()`: comma separated (CSV) files

`read_tsv()`: tab separated files

`read_delim()`: general delimited files

`read_fwf()`: fixed width files

`read_table()`: tabular files where columns are separated by white-space.

`read_log()`: web log files

Getting data into R - the readr package

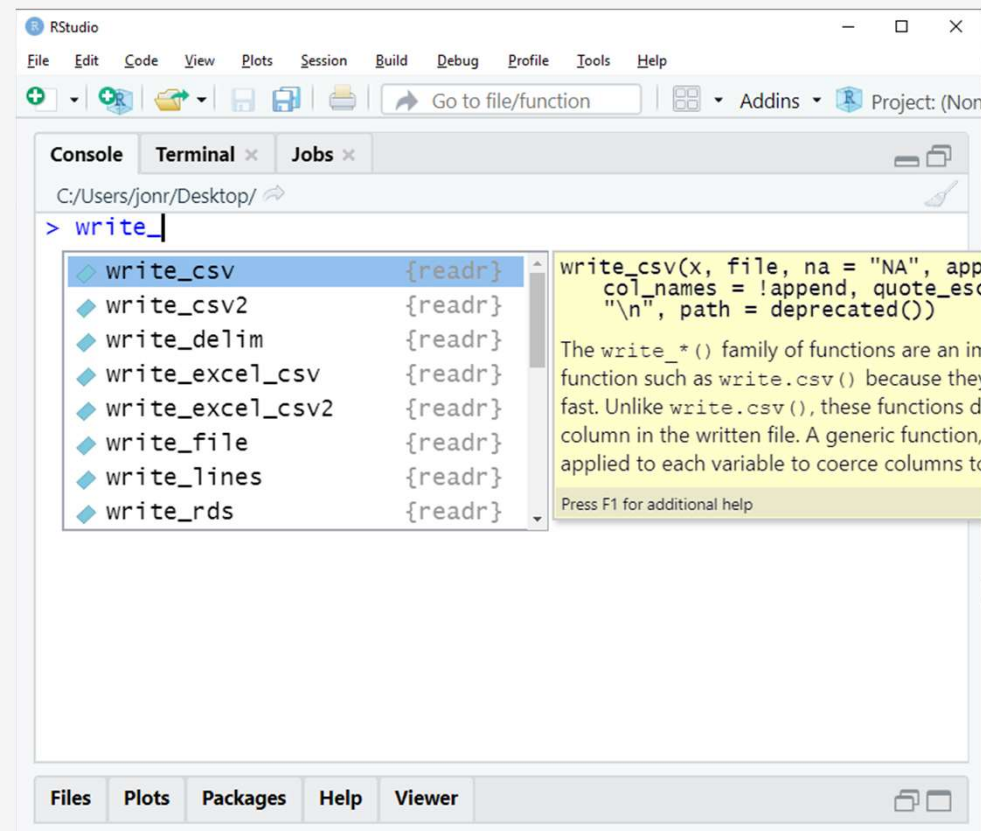
The functions have different arguments that can be used to further specify the structure of the file to be read. E.g., does the file have a header line? What type of symbol separates the columns? Are there any lines that should be skipped? Etc.



Notice the pop-up help menu. The different arguments are shown, with default values.

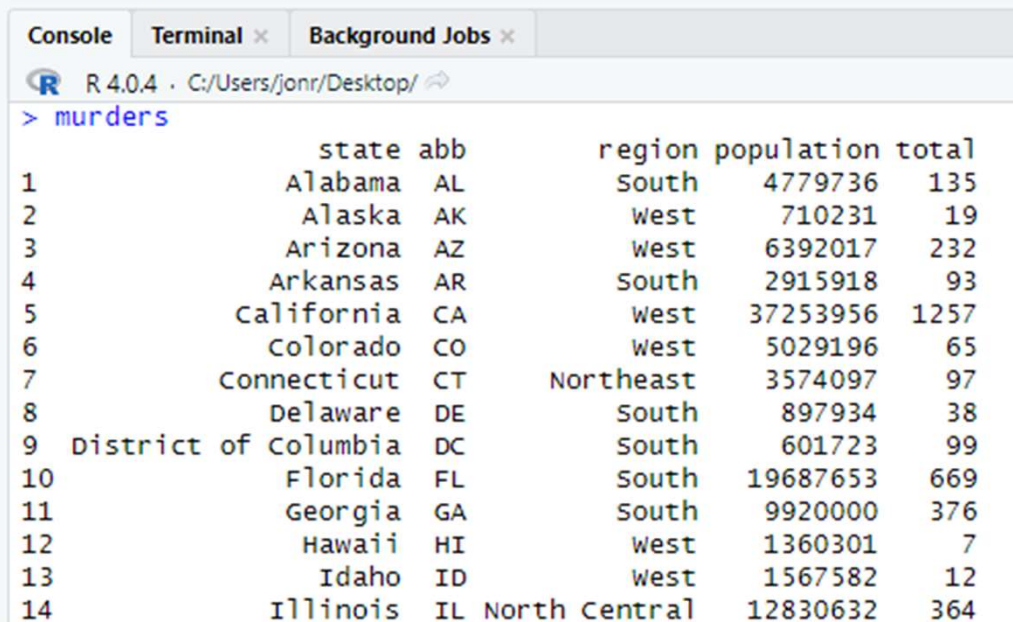
Getting data out of R

The readr package also comes with complementary write functions that can write files in different formats.



Tibbles

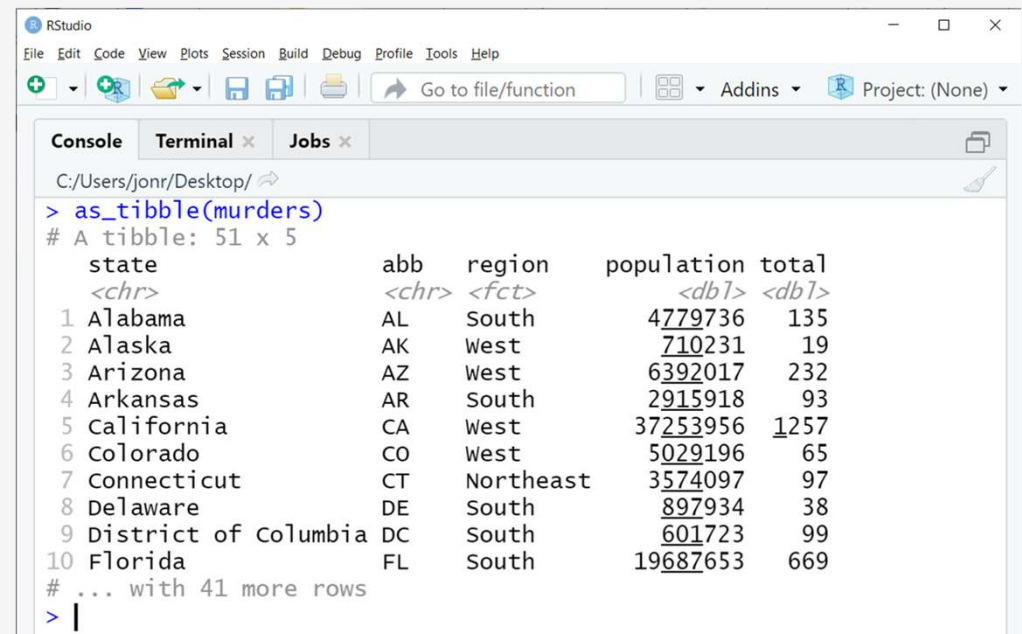
A tibble is a special kind of data frame. Tibbles are the preferred format in the tidyverse and most tidyverse operations result in a tibble. Tibbles also display better when printed in R.



R 4.0.4 · C:/Users/jonr/Desktop/

```
> murders
```

	state	abb	region	population	total
1	Alabama	AL	South	4779736	135
2	Alaska	AK	West	710231	19
3	Arizona	AZ	West	6392017	232
4	Arkansas	AR	South	2915918	93
5	California	CA	West	37253956	1257
6	Colorado	CO	West	5029196	65
7	Connecticut	CT	Northeast	3574097	97
8	Delaware	DE	South	897934	38
9	District of Columbia	DC	South	601723	99
10	Florida	FL	South	19687653	669
11	Georgia	GA	South	9920000	376
12	Hawaii	HI	West	1360301	7
13	Idaho	ID	West	1567582	12
14	Illinois	IL	North Central	12830632	364



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

C:/Users/jonr/Desktop/

```
> as_tibble(murders)
# A tibble: 51 x 5
  state      abb region      population total
  <chr>    <chr> <fct>      <dbl>    <dbl>
1 Alabama  AL    South      4779736    135
2 Alaska   AK    West        710231     19
3 Arizona  AZ    West      6392017    232
4 Arkansas AR    South     2915918     93
5 California CA    West    37253956   1257
6 Colorado CO    West     5029196     65
7 Connecticut CT    Northeast 3574097     97
8 Delaware DE    South      897934     38
9 District of Columbia DC    South      601723     99
10 Florida  FL    South    19687653   669
# ... with 41 more rows
> |
```