



# ST 117

## 2. Basic R



WARWICK

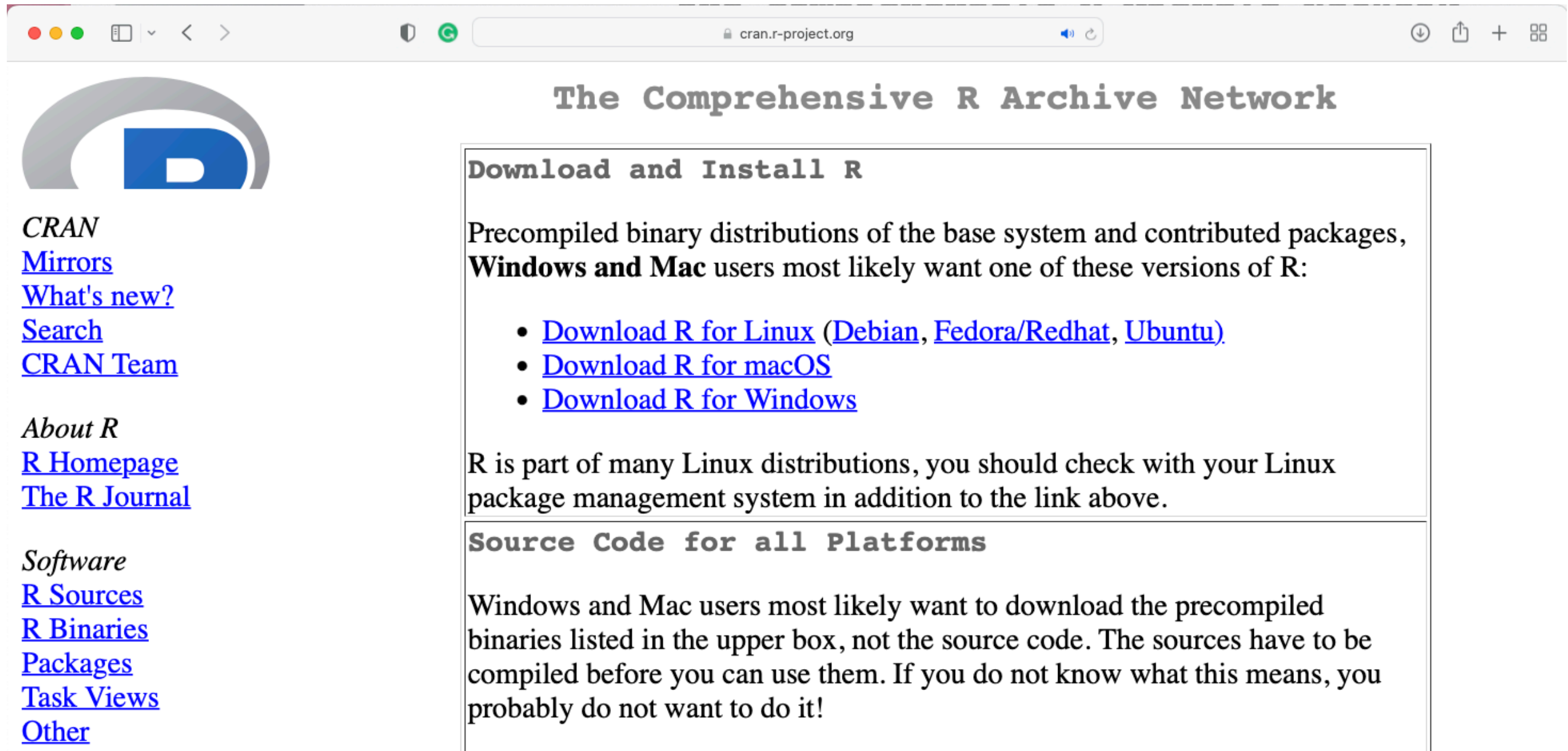
### Lecture 3 (Week 1)

Getting started  
Data types and structures  
Predefined functions

# Download: R

Please download this as soon as possible (essential)

<https://cran.r-project.org>



The screenshot shows the CRAN website in a web browser. The browser's address bar displays 'cran.r-project.org'. The website's header features the CRAN logo on the left and the title 'The Comprehensive R Archive Network' in the center. On the left side, there is a vertical menu with links: 'CRAN', 'Mirrors', 'What's new?', 'Search', 'CRAN Team', 'About R', 'R Homepage', 'The R Journal', 'Software', 'R Sources', 'R Binaries', 'Packages', 'Task Views', and 'Other'. The main content area is divided into two sections. The first section, titled 'Download and Install R', contains text about precompiled binary distributions and lists three download links for Linux (Debian, Fedora/Redhat, Ubuntu), macOS, and Windows. The second section, titled 'Source Code for all Platforms', contains text advising users to download precompiled binaries instead of source code.

**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](#)

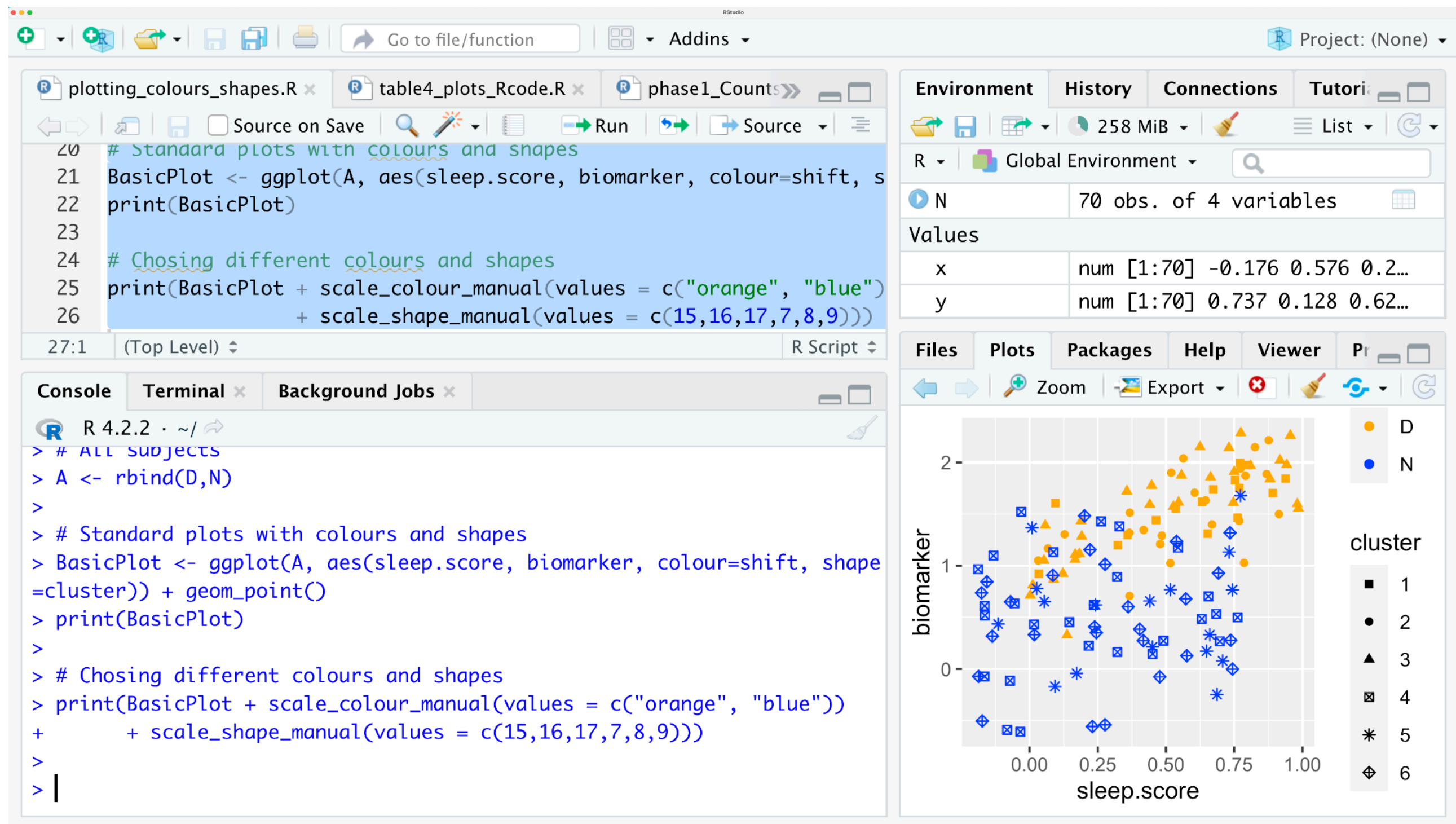
R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

**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!

# Integrated development environment: RStudio

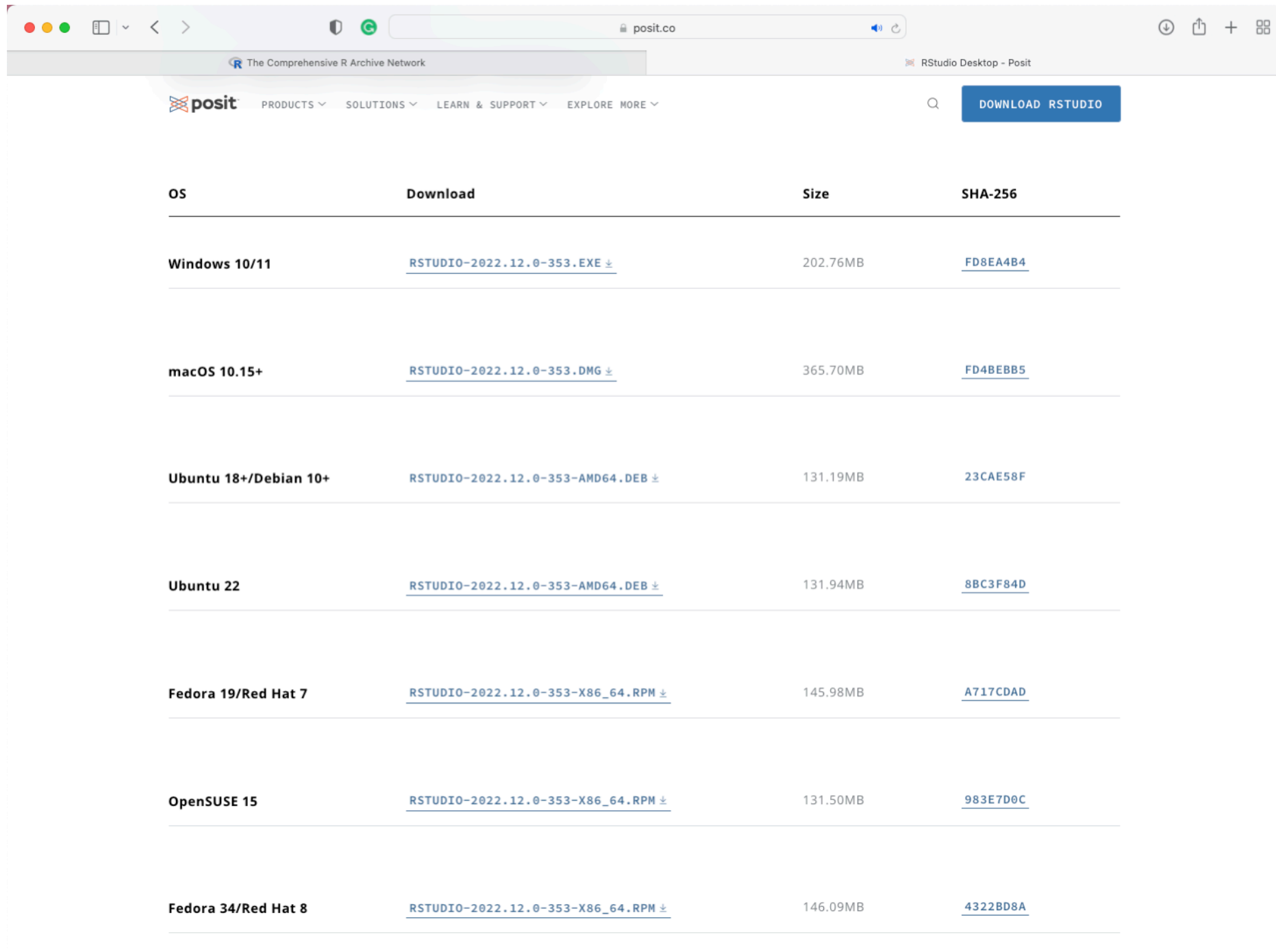
R console integrated with editor, var list, output, help etc



# Download: RStudio

Please download this, too (convenient)

<https://posit.co/download/rstudio-desktop/>



OS	Download	Size	SHA-256
Windows 10/11	<a href="#">RSTUDIO-2022.12.0-353.EXE</a>	202.76MB	<a href="#">FD8EA4B4</a>
macOS 10.15+	<a href="#">RSTUDIO-2022.12.0-353.DMG</a>	365.70MB	<a href="#">FD4BE8B5</a>
Ubuntu 18+/Debian 10+	<a href="#">RSTUDIO-2022.12.0-353-AMD64.DEB</a>	131.19MB	<a href="#">23CAE58F</a>
Ubuntu 22	<a href="#">RSTUDIO-2022.12.0-353-AMD64.DEB</a>	131.94MB	<a href="#">8BC3F84D</a>
Fedora 19/Red Hat 7	<a href="#">RSTUDIO-2022.12.0-353-X86_64.RPM</a>	145.98MB	<a href="#">A717CDAD</a>
OpenSUSE 15	<a href="#">RSTUDIO-2022.12.0-353-X86_64.RPM</a>	131.50MB	<a href="#">983E7D0C</a>
Fedora 34/Red Hat 8	<a href="#">RSTUDIO-2022.12.0-353-X86_64.RPM</a>	146.09MB	<a href="#">4322BD8A</a>

Windows

Mac

Linux (various)



# Warwick online resource

Moodle course "Basic R with pointers"

<https://moodle.warwick.ac.uk/course/view.php?id=41822>

## Basic R with pointers

[Dashboard](#) / [My Moodle](#) / [Stats-Basic R](#)

*We developed **Basic R with pointers** for the students of the [Department of Statistics](#) at University of Warwick to be used as a gentle introduction or refresher to R and its core functionality, or as a complement to the various modules that use R in the [undergraduate](#) and [postgraduate](#) courses offered by the Department of Statistics. While designing the resource, it became clear to us that it could be handy to students and staff across the University, who engage or want to engage with R. As a result, we are offering direct enrolment to anyone interested.*

*The Basic R with pointers team, September 2020*

This resource is formally part of the [WOLC material](#) the [Department of Statistics](#) provides to its students.

# R language basics

Rough summary of topics we will cover in the next four lectures

<https://github.com/rstudio/cheatsheets/blob/main/base-r.pdf>

## Base R Cheat Sheet

### Getting Help

#### Accessing the help files

`?mean`  
Get help of a particular function.

`help.search('weighted mean')`  
Search the help files for a word or phrase.

`help(package = 'dplyr')`  
Find help for a package.

#### More about an object

`str(iris)`  
Get a summary of an object's structure.

`class(iris)`  
Find the class an object belongs to.

### Using Packages

`install.packages('dplyr')`  
Download and install a package from CRAN.

`library(dplyr)`  
Load the package into the session, making all its functions available to use.

`dplyr::select`  
Use a particular function from a package.

`data(iris)`

### Working Directory

`getwd()`  
Find the current working directory (where inputs are found and outputs are sent).

`setwd('C://file/path')`  
Change the current working directory.

Use projects in RStudio to set the working directory to the folder you are working in.

## Vectors

### Creating Vectors

<code>c(2, 4, 6)</code>	<code>2 4 6</code>	Join elements into a vector
<code>2:6</code>	<code>2 3 4 5 6</code>	An integer sequence
<code>seq(2, 3, by=0.5)</code>	<code>2.0 2.5 3.0</code>	A complex sequence
<code>rep(1:2, times=3)</code>	<code>1 2 1 2 1 2</code>	Repeat a vector
<code>rep(1:2, each=3)</code>	<code>1 1 1 2 2 2</code>	Repeat elements of a vector

### Vectors Functions

<code>sort(x)</code>	<code>rev(x)</code>
Return x sorted.	Return x reversed.
<code>table(x)</code>	<code>unique(x)</code>
See counts of values.	See unique values.

### Selecting Vector Elements

#### By Position

<code>x[4]</code>	The fourth element.
<code>x[-4]</code>	All but the fourth.
<code>x[2:4]</code>	Elements two to four.
<code>x[-(2:4)]</code>	All elements except two to four.
<code>x[c(1, 5)]</code>	Elements one and five.

#### By Value

<code>x[x == 10]</code>	Element which are equal to 10.
<code>x[which(x==10)]</code>	Element which are equal to 10.
<code>x[x &lt; 0]</code>	All elements less than zero.
<code>x[x %in% c(1, 2, 5)]</code>	Elements in the set {1, 2, 5}.

#### Named Vectors

<code>x['apple']</code>	Element with name 'apple'.
-------------------------	----------------------------

## Programming

### For Loop

```
for (variable in sequence) {  
  Do something  
}
```

#### Example

```
for (i in 1:4) {  
  j <- i + 10  
  print(j)  
}
```

### While Loop

```
while (condition) {  
  Do something  
}
```

#### Example

```
while (i < 5) {  
  print(i)  
  i <- i + 1  
}
```

### If Statement

```
if (condition) {  
  Do something  
} else {  
  Do something  
}
```

#### Example

```
if (i > 3) {  
  print('Yes')  
} else {  
  print('No')  
}
```

### Functions

```
func_name <- function(var) {  
  Do something  
  return(new_variable)  
}
```

#### Example

```
square <- function(x) {  
  squared <- x*x  
  return(squared)  
}
```

### Reading and Writing Data

Input	Output	Description
<code>df &lt;- read.table('file.txt')</code>	<code>write.table(df, 'file.txt')</code>	Read and write a delimited text file.
<code>df &lt;- read.csv('file.csv')</code>	<code>write.csv(df, 'file.csv')</code>	Read and write a comma separated value file. This is a special case of read.table/write.table.
<code>load('file.RData')</code>	<code>save(df, file = 'file.RData')</code>	Read and write a n R data file, a file type special for R.

#### Also see the readr package.

Conditions	a == b	Are equal	a > b	Greater than	a >= b	Greater than or equal to	is.na(a)	Is missing
	a != b	Not equal	a < b	Less than	a <= b	Less than or equal to	is.null(a)	Is null
	c    e	c or e	c && y	c and y				

## Strings

#### Also see the stringr package.

<code>cat(x, y, sep = '')</code>	Join and print multiple vectors together.
<code>cat(x, collapse = '')</code>	Join and print elements of a vector together.
<code>grep(pattern, x)</code>	Find regular expression matches in x.
<code>gsub(pattern, replace, x)</code>	Replace matches in x with a string.
<code>toupper(x)</code>	Convert to uppercase.
<code>tolower(x)</code>	Convert to lowercase.
<code>nchar(x)</code>	Number of characters in a string.

## Factors

<code>factor(x)</code>	Turn a vector into a factor. Can set the levels of the factor and the order.
<code>cut(x, breaks = 4)</code>	Turn a numeric vector into a factor by 'cutting' into sections.

## Statistics

<code>lm(y ~ x, data=df)</code> Linear model.	<code>t.test(x, y)</code> Perform a t-test for difference between means.	<code>prop.test</code> Test for a difference between proportions.
<code>glm(y ~ x, data=df)</code> Generalized linear model.	<code>pairwise.t.test</code> Perform a t-test for paired data.	<code>aov</code> Analysis of variance.
<code>summary</code> or <code>fivenum</code> Get more detailed information out a model.		

## Distributions

	Random Variates	Density Function	Cumulative Distribution	Quantile
Normal	<code>rnorm</code>	<code>dnorm</code>	<code>pnorm</code>	<code>qnorm</code>
Poisson	<code>rpois</code>	<code>dpois</code>	<code>ppois</code>	<code>qpois</code>
Binomial	<code>rbinom</code>	<code>dbinom</code>	<code>pbinom</code>	<code>qbinom</code>
Uniform	<code>runif</code>	<code>dunif</code>	<code>punif</code>	<code>qunif</code>

## Plotting

#### Also see the ggplot2 package.

<code>plot(x)</code> Values of x in order.	<code>plot(x, y)</code> Values of x against y.	<code>hist(x)</code> Histogram of x.
-----------------------------------------------	---------------------------------------------------	-----------------------------------------



---

# What's good about R?

---

- ❖ It's free.
- ❖ Open source.
- ❖ Basic statistical applications are very straightforward.
- ❖ Full-function programming language, relatively easy to write and read
- ❖ Very easy to make simple default plots, but not too hard to exert fine control over the graphics.
- ❖ Flexible syntax.
- ❖ Huge community writing packages and keeping an eye on the basics.
- ❖ High-quality random number generators.



---

# What's not so good about R?

---

- ❖ On-board documentation is generally mediocre.
- ❖ Slow, particularly for any application involving loops. Such programs need to be written in C++ and embedded in R.
- ❖ Not the best for numerical methods.
- ❖ Doesn't do algebraic manipulation or high-precision integer calculations.
- ❖ Packages are of mixed quality. Outside of core R, *caveat emptor*.



---

# Finding help

---

- ❖ If you know what command you need, typing ?command brings up a help page, e.g. ?sqrt
- ❖ If you know approximately what command, or a keyword, ??word searches for help pages.
- ❖ Lots of free online help is available.
  - ❖ Short intro: <http://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf>
  - ❖ Short reference card: <http://cran.r-project.org/doc/contrib/Short-refcard.pdf>
  - ❖ Long reference: [http://web.udl.es/Biomath/Bioestadistica/R/Manuals/r\\_in\\_a\\_nutshell.pdf](http://web.udl.es/Biomath/Bioestadistica/R/Manuals/r_in_a_nutshell.pdf)
  - ❖ Online course (if you like watching video lectures): <https://www.coursera.org/course/rprog>
- ❖ Lots of non-free books are now available
  - ❖ General books, e.g., Venables and Ripley, [Modern Applied Statistics with S-Plus](#); Crawley, [The R Book](#) and [Statistics: An introduction using R](#).
  - ❖ Specific books, with titles like [Statistical Method X with R](#), often published by CRC Press.



---

# Interacting with R

---

- ❖ Console: Type here for immediate action. Text output comes here.
- ❖ R script: Write programs, longer work that you want to save, modify, etc.
- ❖ Graphics windows are generated by R commands.
- ❖ Important shortcuts:
  - ❖ Up-arrow repeats previous line. Repeating goes back through your history.
  - ❖ Ctrl-Enter (Cmd-Enter on Mac) after selecting text in an R-script window runs the selected commands in the console.
- ❖ Integrated development environments (IDEs) such as RStudio (available free) help to organise the windows, scripts, variables, etc.



# Basic arithmetic in R

```
> 2+3
[1] 5
> 2*3
[1] 6
> 2^3
[1] 8
> 7%%3
[1] 1
> 7%/%3
[1] 2
```

```
> x=27
> sqrt(x)
[1] 5.196152
> y<-sqrt(x)
> y^2
[1] 27
> 27 -> x
> x
[1] 27
> 27=x
```

Error in 27 = x : invalid  
(do\_set) left-hand side to  
assignment

Boolean operators

```
> y<5
[1] FALSE
> y^2==27
[1] TRUE
> (x/y)==y
[1] TRUE
> (x/y)<=y
[1] TRUE
```





## "Translate" into R:

---

$$(-17)^2$$

$$x = 23, y = (x - 89)^{-2}$$

$$0^0 + 1^1 + 2^2 + 3^3 + 4^4 + 5^5 + 6^6 + 7^7$$

The formula for the solution of quadratic equations

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \text{ (either case) and calculate the value for a few cases}$$

of values  $a, b, c$ . Try to use some values such that the discriminant is negative to see what happens then.

An expression that returns TRUE or FALSE depending on whether the discriminant in the expression above.



---

# Data types in R

---

**Numeric:** Decimal values, this is the default computational data type

**Integer:** Values that can be written without a fractional component

**Character:** A piece of text is represented as a sequence of characters (letters, number, symbols)

**Logic:** TRUE or FALSE values

**Factor:** A categorical value, also called "level" of a categorical variable

```
> x <- 243.6
class(x)
[1] "numeric"
> is.numeric(x)
[1] TRUE
> is.integer(x)
[1] FALSE

> as.integer(x)
[1] 243
> floor(243)
[1] 243
> ceiling(x)
[1] 244
```



# Logical operators

Operator	Description
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
==	exactly equal to
!=	not equal to
!x	not x
x y	x or y
x&y	x and y

```
> x=1
> x!=2
[1] TRUE
> x>2
[1] FALSE
> x<=1
[1] TRUE
> (x|y)==1
[1] TRUE
> x&y==1
[1] FALSE
```



---

# Find out which class these expression are:

---

`"Hello world"`

`'Hello world'`

``Hello world``

*Note: One of these create a problem that is worth to have encountered, so you will remember how to get unstuck. You will get unstuck by pressing control C.*

`7`

`"7"`

`as.integer(7)`

`as.integer("7")`

`as.integer(7)^2`

`floor(8.9)`

`7==8`

`7==7`



---

# Generating sequences

---

Look at help entries:

```
>help(rep)
```

```
> rep(5,3)
```

```
[1] 5 5 5
```

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

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

```
> rep(3:1,each=3)
```

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

```
> rep(4:1,1:4)
```

```
[1] 4 3 3 2 2 2 1 1 1 1
```

```
> rep(-1:3,length.out=6)
```

```
[1] -1 0 1 2 -1 0
```

```
>help("sequence")
```

```
>help(seq)
```

```
> seq(from=0,to=1,by=.1)
```

```
[1] 0.0 0.1 0.2 0.3 0.4 0.5
```

```
0.6 0.7 0.8 0.9 1.0
```

```
> seq(0,1)
```

```
[1] 0 1
```

```
> seq(0,1,length.out=6)
```

```
[1] 0.0 0.2 0.4 0.6 0.8 1.0
```

```
> seq(1,0,along.with=c(17,9,-4,0,NA))
```

```
[1] 1.00 0.75 0.50 0.25 0.00
```





## Generating sequences:

---

Generate a sequence of numbers divisible by 3 between 1 and 31.

Find an alternative expression that returns the same result.

Create a sequence of even numbers between 1 and 6 that show each number half of the times as its value. Do the same but showing them in reverse order.

Show all numbers between 99 and 100 in steps of 0.01.

Create a sequence that shows ABC three times. Do this in another way, too.

# Timelines

	Due	Lectures	Lab	Posted	<u>Indiv. tasks</u>
<b>Term 2</b>					
	A/E/WR Tue 1pm Q Wed	W1 Thu 11-12pm & 1-2pm & Fri 11-12pm W2+ Mon 1-2pm & Tue 1-3pm	W2+ Tue-Fri	A/E/W Tue Q Wed	
<b>1. Introduction &amp; 2. R basics</b>					
<b>1</b>		Teaching team, module assessment & org., tasters, syllabus  R basics (covering material on cheat sheet): getting started, data types & structures, predefined functions,		A0	A0 Install R, R Studio Small exercises from lecture
<b>2</b>	A0 <u>indiv.</u>	data input & output, tables, graphics, constructing functions, controlling flow, wrapping up and looking ahead, finding resources	Practice R with your A0 datasets		Small exercises from lecture  Practice for Q1

A=Activity, Q=Quiz, E=Exercise set



# First Year R Course

## Overview of topics:

- Basics of R (R studio, operations in R and logical operators)
- Data Types & Data Structures
- Control Structures
- Custom Functions
- Brief insight into Applications of R

**Access to slides (with theory, code demos and questions), demonstrations in R, practice questions and more...**



### When & Where?

Every **Wednesday 3-4pm**  
(After Stats Café)

**MB0.07** (W2-4,6-7,9-10)  
MB0.08 (W5,8)

Mathematical Sciences (Stats)  
Building



# Warwick Statistics Society



## **Academic Help**

- R Course
- Stats Café – Every Wednesday 1-3pm – CV checks with our careers officer, academic help from students in years 3&4, board games & activities and pizza!

## **Social Events**

- Check @warwickstats on Instagram for up-to-date info
- Past events include: Ball (with Maths & Physics), BBQ, Board Games, Poker night etc.

## **Sports**

- Badminton & Football Teams in Society Leagues