

# Practical Approaches to Bioanalysis in R

## Day 1 – Intro to R and the Tidyverse Ecosystem

\*Many of these slides have been contributed by or modified from  
slides contributed by Dr. Claus Wilke

# R: The premier data analysis and visualization platform

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



## 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.

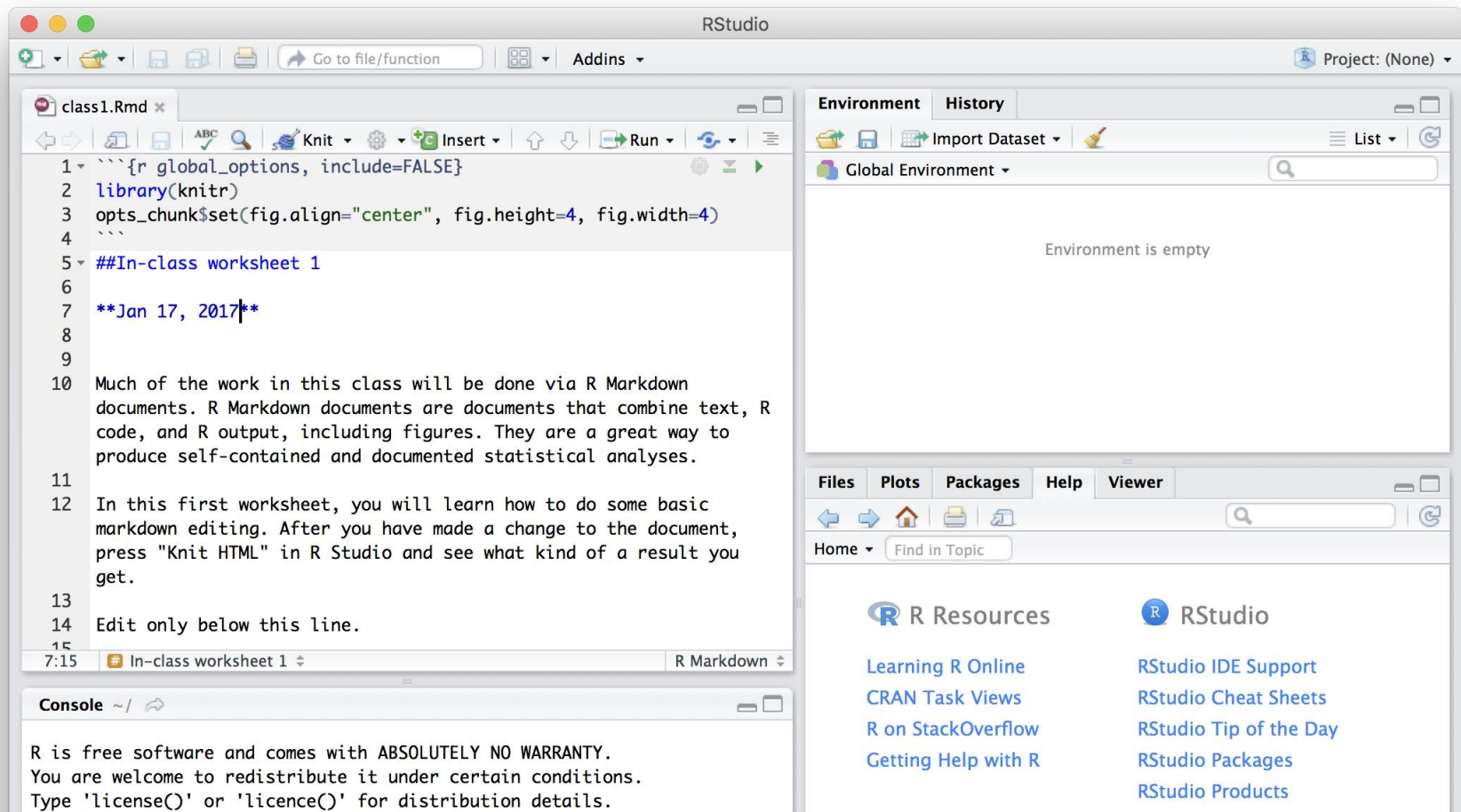
### Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper

# R Studio:

## A nice user interface for R

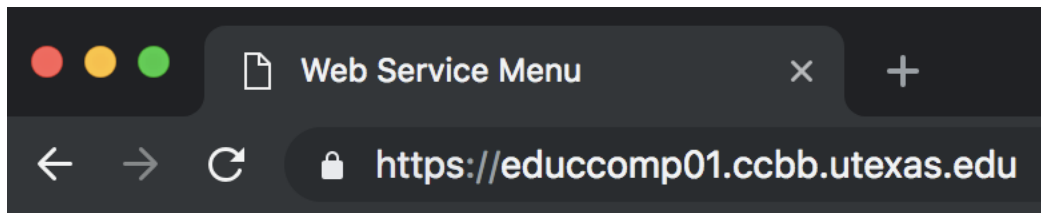
<https://www.rstudio.com/products/rstudio/download/>



# Access R Studio through your web browser

1. <https://gsafcomp01.ccb.b.utexas.edu/>
2. <https://gsafcomp02.ccb.b.utexas.edu/>
3. <https://gsafcbig01.ccb.b.utexas.edu/>

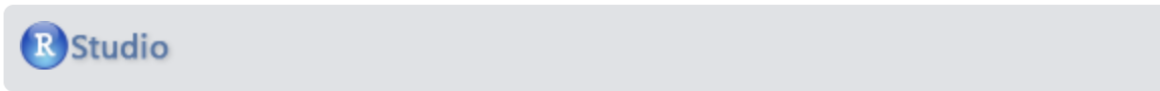
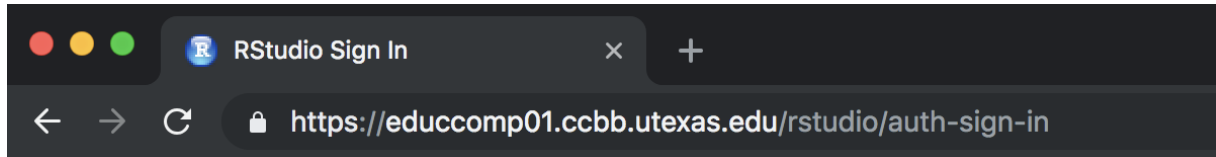
# Select RStudio



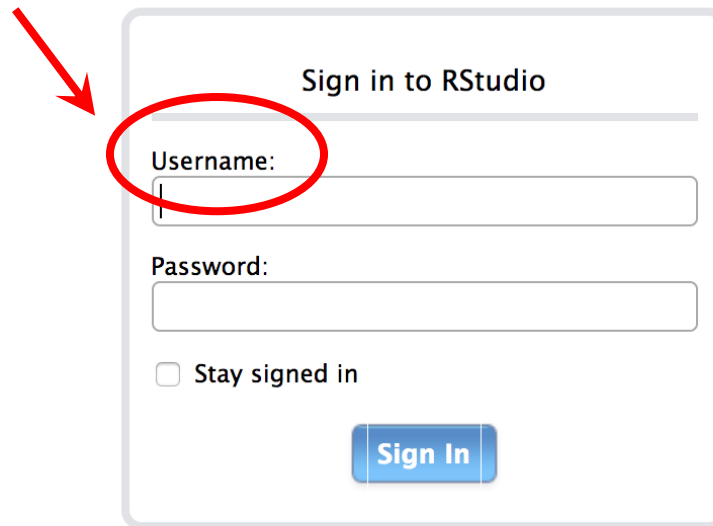
Please choose one of the following applications:

- [RStudio](#) ←
- [Jupyterhub](#)

# Sign in with your student# and password

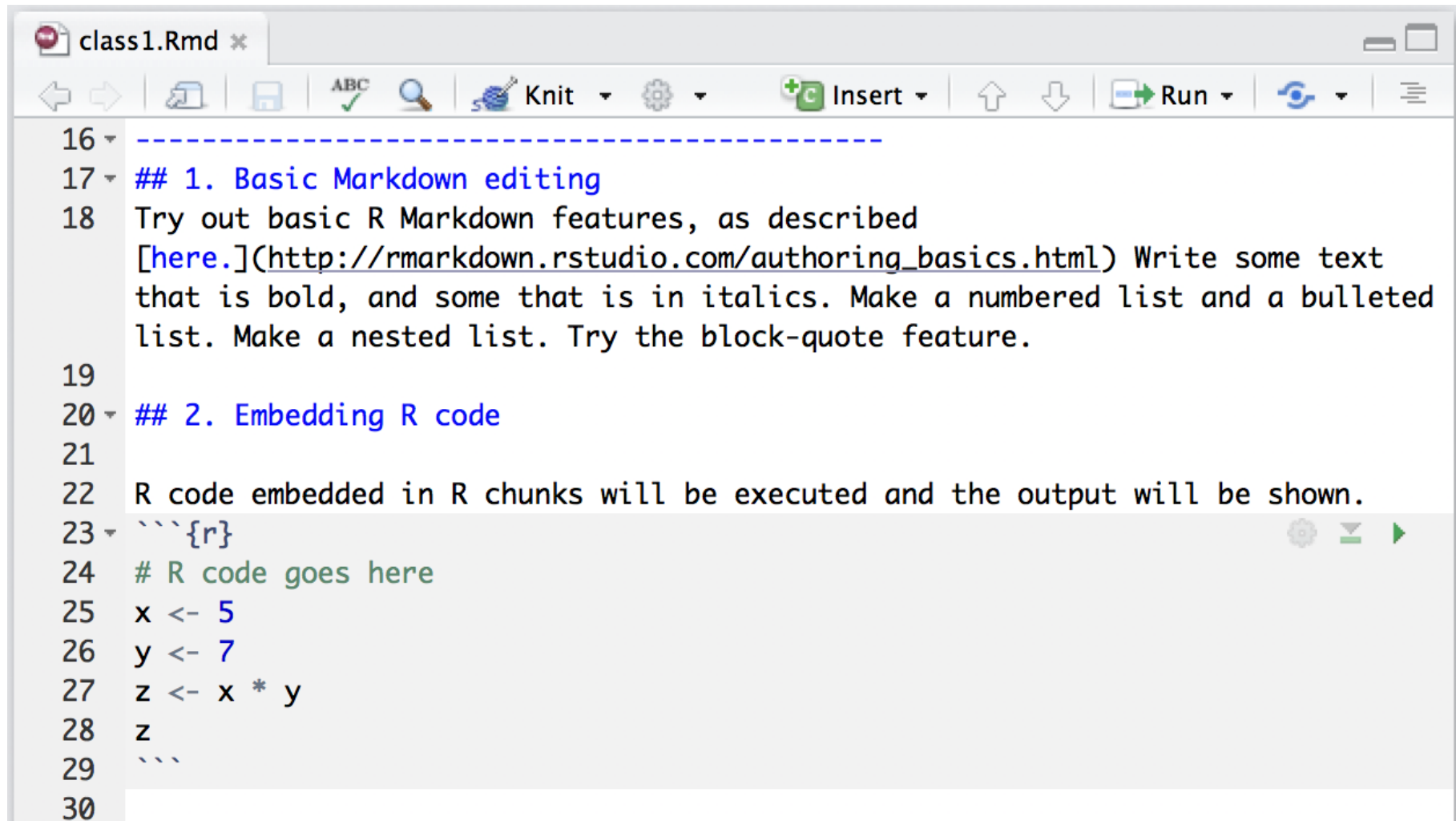


Refer to class email for your individual username

A screenshot of the 'Sign in to RStudio' form. The form is titled 'Sign in to RStudio' and contains two input fields: 'Username:' and 'Password:'. A red arrow points to the 'Username:' field, which is also circled in red. Below the password field is a checkbox labeled 'Stay signed in'. At the bottom of the form is a blue button labeled 'Sign In'.

# R Markdown

# R Markdown: Writing documents with embedded R code



```
class1.Rmd x
-----
## 1. Basic Markdown editing
Try out basic R Markdown features, as described
[here.](http://rmarkdown.rstudio.com/authoring_basics.html) Write some text
that is bold, and some that is in italics. Make a numbered list and a bulleted
list. Make a nested list. Try the block-quote feature.

## 2. Embedding R code
R code embedded in R chunks will be executed and the output will be shown.
```${r}
# R code goes here
x <- 5
y <- 7
z <- x * y
z
```
```



# R Markdown:

## Writing documents with embedded R code

### 1. Basic Markdown editing

Try out basic R Markdown features, as described [here](#). Write some text that is bold, and some that is in italics. Make a numbered list and a bulleted list. Make a nested list. Try the block-quote feature.

### 2. Embedding R code

R code embedded in R chunks will be executed and the output will be shown.

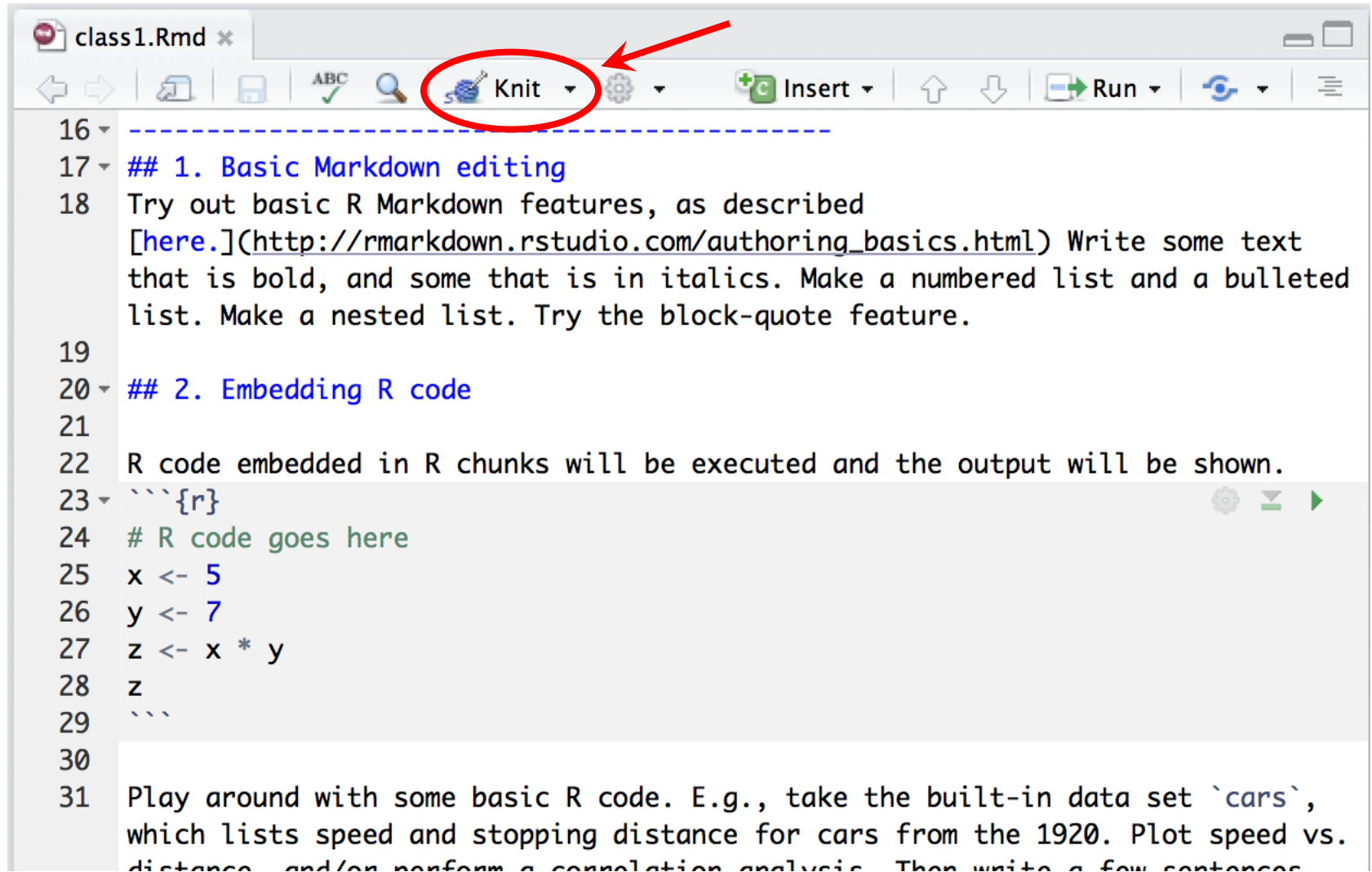
```
# R code goes here
```

```
x <- 5  
y <- 7  
z <- x * y  
z
```

```
## [1] 35
```

Play around with some basic R code. E.g., take the built-in data set `cars`, which lists speed and stopping distance for cars from the 1920. Plot speed vs. distance, and/or perform a correlation analysis. Then write a few sentences describing what you see.

# We convert R Markdown to HTML by “knitting” the Markdown file



# Markdown basics

[http://rmarkdown.rstudio.com/authoring\\_basics.html](http://rmarkdown.rstudio.com/authoring_basics.html)

normal text

*\*italics\**

**\*\*bold\*\***

# Header 1

## Header 2

List:

1. Item 1
2. Item 2
3. Item 3



normal text

*italics*

**bold**

# Header 1

## Header 2

List:

1. Item 1
2. Item 2
3. Item 3

# Markdown basics

Embedded R code will be evaluated and printed

```
```{r}  
head(cars)  
plot(cars$speed, cars$dist)  
```
```

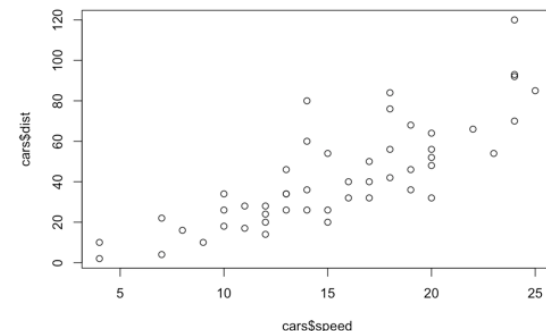


Embedded R code will be evaluated and printed

```
head(cars)
```

```
##   speed dist  
## 1     4    2  
## 2     4   10  
## 3     7    4  
## 4     7   22  
## 5     8   16  
## 6     9   10
```

```
plot(cars$speed, cars$dist)
```



# R Programming Basics

# Assignments, numbers, vectors

```
> x <- 5
```

Assign number 5 to variable x

```
> x
```

```
[1] 5
```

```
> 5*x^2+7
```

Calculate  $5x^2+7$

```
[1] 132
```

```
> y <- c(1, 2, 3, 4, 5)
```

Create vector, assign  
to variable y

```
> y
```

```
[1] 1 2 3 4 5
```

```
> x*y
```

Multiply each element  
in y with the number in x

```
[1] 5 10 15 20 25
```

# Strings

A string contains text:

```
> name <- "Rachael Cox"
> name
[1] "Rachael Cox"
```

A vector of strings:

```
> animals <- c("cat", "mouse", "mouse",
"cat", "rabbit")
> animals
[1] "cat"      "mouse"    "mouse"    "cat"
"rabbit"
```

# Factors

Factors keep track of distinct categories (levels) in a vector:

```
> animals  
[1] "cat"      "mouse"    "mouse"    "cat"  
"rabbit"
```

```
> factor(animals)  
[1] cat      mouse    mouse    cat      rabbit  
Levels: cat mouse rabbit
```



# Data frames

We use data frames to store data sets with multiple variables:

```
> pets <- data.frame(  
  family = c(1, 2, 3, 4, 5),  
  pet = animals  
)
```

```
> pets
```

|   | family | pet   |
|---|--------|-------|
| 1 | 1      | cat   |
| 2 | 2      | mouse |
| 3 | 3      | mouse |
| 4 | 4      | cat   |

# Data frames

We access individual columns in a data frame with \$ + the column name:

```
> pets$family  
[1] 1 2 3 4 5
```

```
> pets$pet  
[1] cat      mouse    mouse    cat      rabbit  
Levels: cat mouse rabbit
```

# Data frames

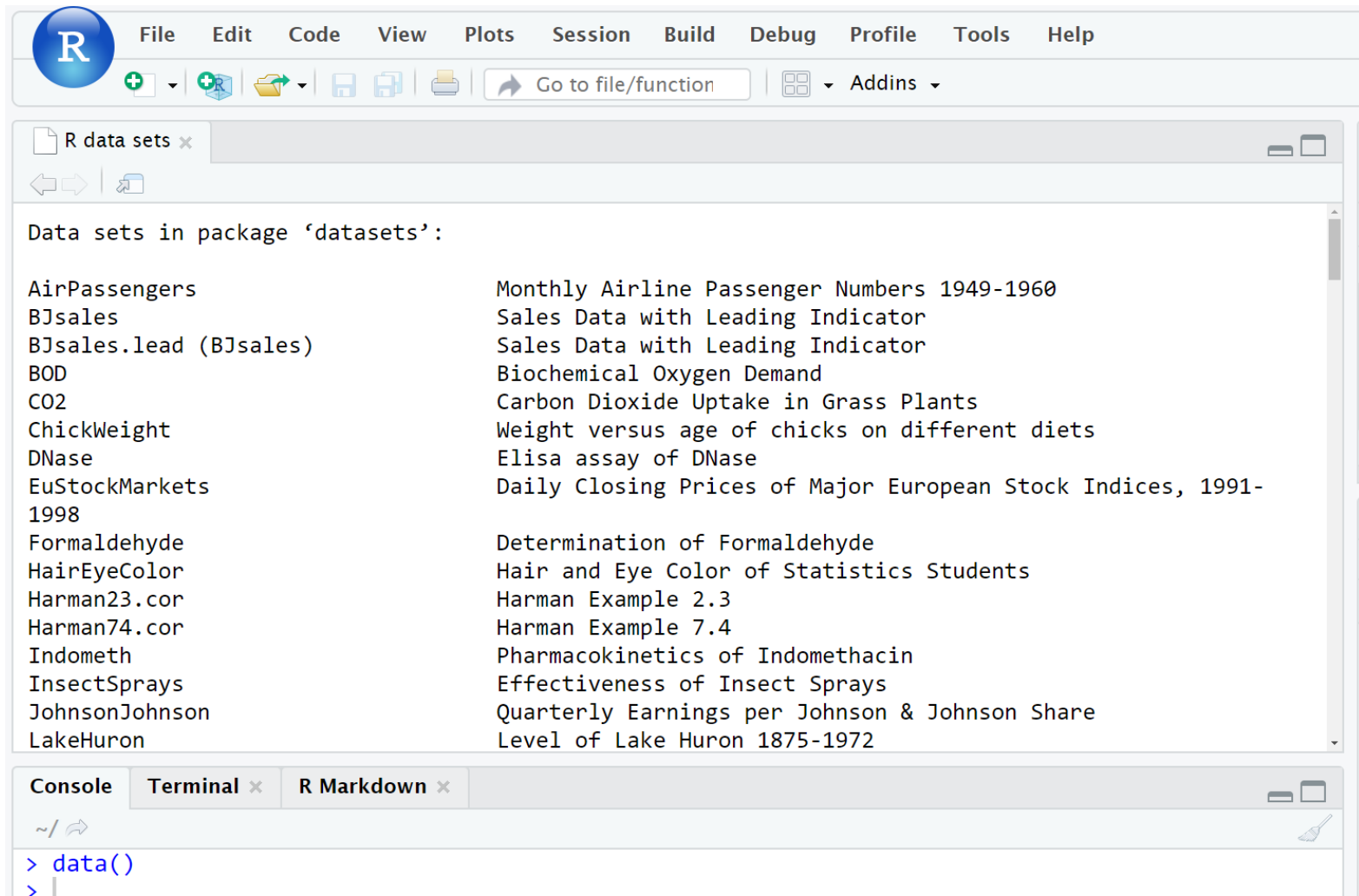
R has many built-in data frames:

```
> cars
```

|    | speed | dist |
|----|-------|------|
| 1  | 4     | 2    |
| 2  | 4     | 10   |
| 3  | 7     | 4    |
| 4  | 7     | 22   |
| 5  | 8     | 16   |
| 6  | 9     | 10   |
| 7  | 10    | 18   |
| 8  | 10    | 26   |
| 9  | 10    | 34   |
| 10 | 11    | 17   |

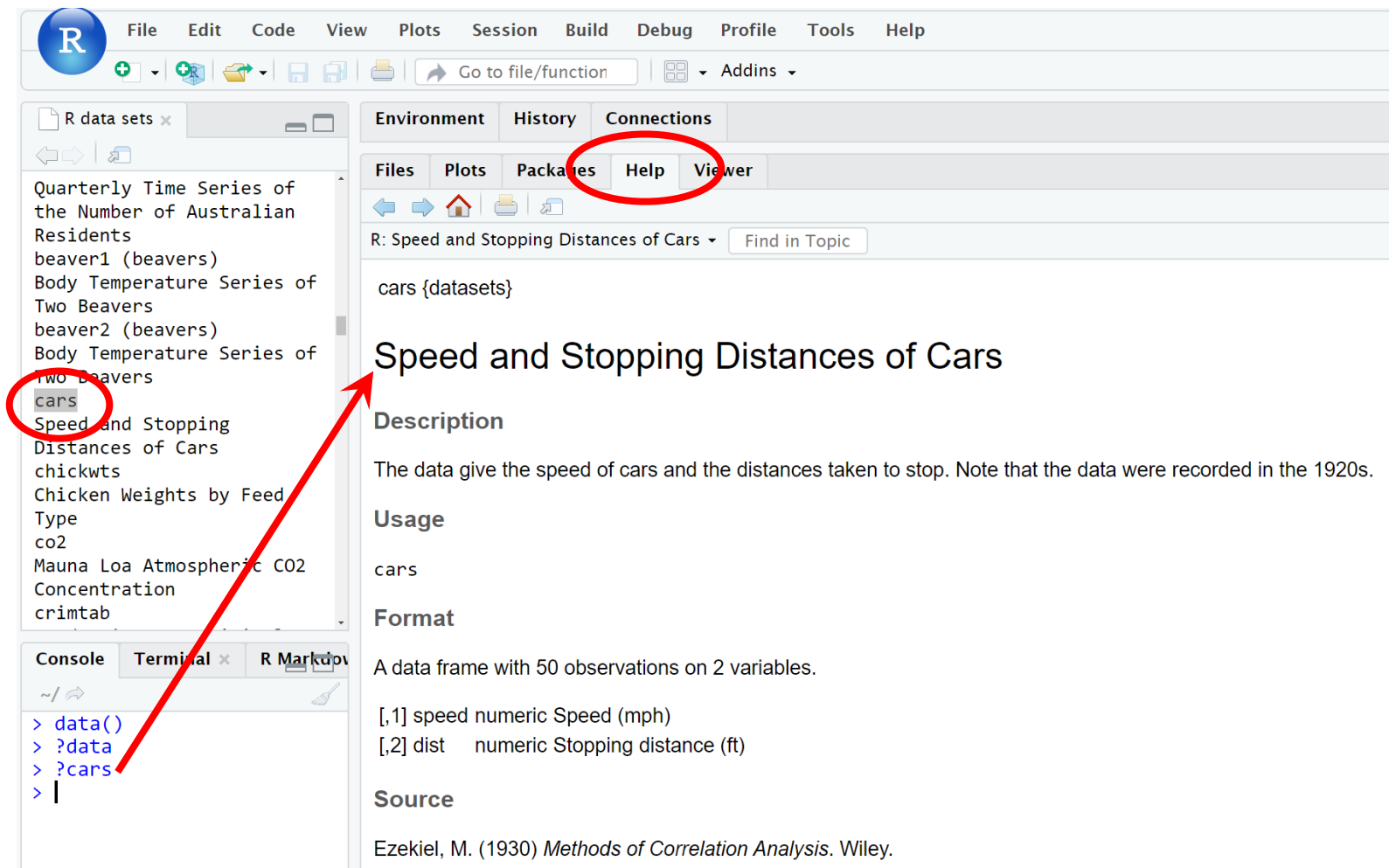
# Data frames

Available built-in datasets can be accessed with `data()`



# Data frames

Dataset information can be accessed with `?<dataset>`



The screenshot shows the RStudio interface. On the left, the 'R data sets' pane lists various datasets, with 'cars' highlighted by a red circle. A red arrow points from this circle to the 'Speed and Stopping Distances of Cars' help page on the right. The help page is titled 'Speed and Stopping Distances of Cars' and includes sections for Description, Usage, Format, and Source. The console at the bottom shows the command `?cars` being entered.

**Environment** **History** **Connections**

**Files** **Plots** **Packages** **Help** **Viewer**

R: Speed and Stopping Distances of Cars

`cars {datasets}`

## Speed and Stopping Distances of Cars

### Description

The data give the speed of cars and the distances taken to stop. Note that the data were recorded in the 1920s.

### Usage

```
cars
```

### Format

A data frame with 50 observations on 2 variables.

```
[,1] speed numeric Speed (mph)
[,2] dist  numeric Stopping distance (ft)
```

### Source

Ezekiel, M. (1930) *Methods of Correlation Analysis*. Wiley.

**Console** **Terminal** **R Markdown**

```
> data()
> ?data
> ?cars
> |
```

# Data frames

The `head()` function shows the first few lines of a data frame:

```
> head(cars)
  speed  dist
1     4     2
2     4    10
3     7     4
4     7    22
5     8    16
6     9    10
>
```

# Functions

Functions are called in the format `function (argument)`

```
> head(cars)
```



Function name



First argument

# Functions

Functions can have any number of required arguments or optional arguments

> head(cars, 8)

Function name

First argument (required)

Second argument (optional; default = 6)



# Functions

`head(cars, 8)` will show the first 8 lines of the data frame instead of the default 6:

```
> head(cars, 8)
```

```
  speed dist
```

```
1      4     2
```

```
2      4    10
```

```
3      7     4
```

```
4      7    22
```

```
5      8    16
```

```
6      9    10
```

```
7     10    18
```

```
8     10    26
```

# Functions

More information about what a function does and/or requires can be accessed with `?<function>`

The screenshot shows the RStudio interface. On the left, a data frame with columns 'speed' and 'dist' is displayed. The 'Console' tab is active, showing the command `> ?head` entered, with a red circle around it. A red arrow points from this command to the help page for `head()` on the right. The help page includes the title 'Return the First or Last Part of an Object', a description, and usage examples.

|   | speed | dist |
|---|-------|------|
| 1 | 4     | 2    |
| 2 | 4     | 10   |
| 3 | 7     | 4    |
| 4 | 7     | 22   |
| 5 | 8     | 16   |
| 6 | 9     | 10   |
| 7 | 10    | 18   |
| 8 | 10    | 20   |

Showing 1 to 8 of 50 entries

**Environment** **History** **Connections**

**Files** **Plots** **Packages** **Help** **Viewer**

R: Return the First or Last Part of an Object

head {utils}

## Return the First or Last Part of an Object

### Description

Returns the first or last parts of a vector, matrix, table, data frame or function. Since `head()` and `tail()` are generic functions.

### Usage

```
head(x, ...)  
## Default S3 method:  
head(x, n = 6L, ...)  
## S3 method for class 'data.frame'  
head(x, n = 6L, ...)  
## S3 method for class 'matrix'  
head(x, n = 6L, ...)  
## S3 method for class 'ftable'  
head(x, n = 6L, ...)  
## S3 method for class 'table'  
head(x, n = 6L, ...)  
## S3 method for class 'function'  
head(x, n = 6L, ...)  
  
tail(x, ...)
```

# Functions

?<function> has argument information

The screenshot shows the RStudio interface. On the left, a data frame with columns 'speed' and 'dist' is visible. The console at the bottom left shows the command `> ?head` entered. A red circle highlights this command, and a red arrow points from it to the 'Arguments' section of the help viewer on the right. The help viewer displays the documentation for the `tail` function, which includes the title 'R: Return the First or Last Part of an Object', S3 method definitions for 'ftable', 'table', and 'function' classes, and a detailed list of arguments.

|   | speed | dist |
|---|-------|------|
| 1 | 4     |      |
| 2 | 4     |      |
| 3 | 7     |      |
| 4 | 7     |      |
| 5 | 8     |      |
| 6 | 9     |      |
| 7 | 10    |      |

Showing 1 to 8 of 50 entries

Console

```
> ?head
> |
```

Environment History Connections

Files Plots Packages Help Viewer

R: Return the First or Last Part of an Object Find in Topic

```
## S3 method for class 'ftable'
tail(x, n = 6L, addrownums = FALSE, ...)
## S3 method for class 'table'
tail(x, n = 6L, addrownums = TRUE, ...)
## S3 method for class 'function'
tail(x, n = 6L, ...)
```

**Arguments**

|            |  |
|------------|--|
| x          | an object  |
| n          | a single integer. If positive, size for the resulting object: number of elements for a vector (including lists), rows for a matrix or data frame or lines for a function. If negative, all but the n last/first number of elements of x. |
| addrownums | if there are no row names, create them from the row numbers.   |
| ...        | arguments to be passed to or from other methods.   |

# Functions

We can implicitly or explicitly pass arguments

```
> head(cars, 8)
```

```
  speed dist
```

```
1      4     2
```

```
2      4    10
```

```
3      7     4
```

```
4      7    22
```

```
5      8    16
```

```
6      9    10
```

```
7     10    18
```

```
8     10    26
```

```
>
```

```
> head(x=cars, n=8)
```

```
  speed dist
```

```
1      4     2
```

```
2      4    10
```

```
3      7     4
```

```
4      7    22
```

```
5      8    16
```

```
6      9    10
```

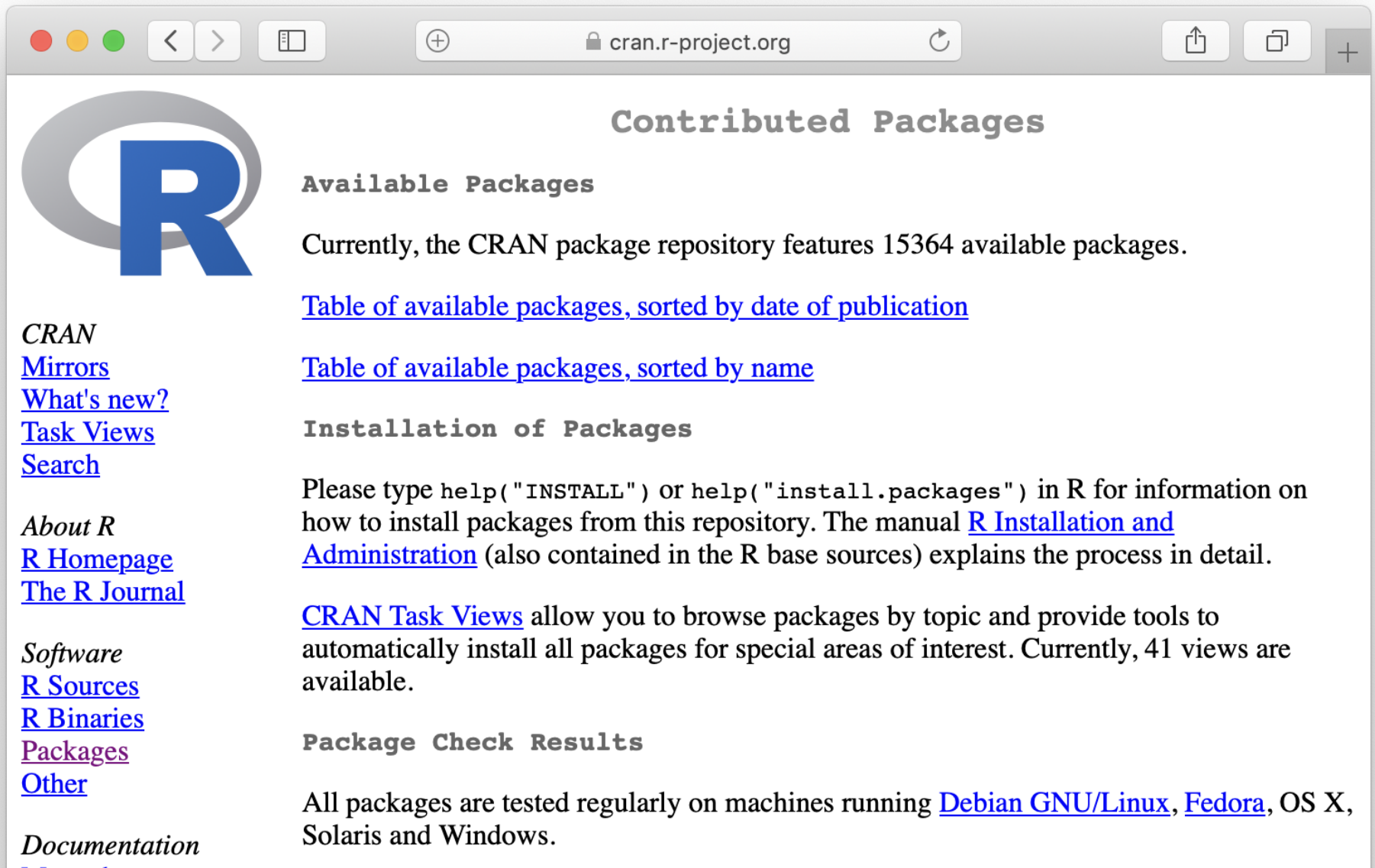
```
7     10    18
```

```
8     10    26
```

```
>
```

Extending R through packages:  
There's a package for everything

# R packages are available on CRAN (Comprehensive R Archive Network)

A screenshot of a web browser displaying the CRAN (Comprehensive R Archive Network) website. The browser's address bar shows 'cran.r-project.org'. The website features the R logo on the left and a main content area with various links and information. The browser window has standard macOS-style window controls (red, yellow, green buttons) and navigation icons (back, forward, home, search, and a plus sign for additional tabs).

**Contributed Packages**

**Available Packages**

Currently, the CRAN package repository features 15364 available packages.

[Table of available packages, sorted by date of publication](#)

[Table of available packages, sorted by name](#)

**Installation of Packages**

Please type `help("INSTALL")` or `help("install.packages")` in R for information on how to install packages from this repository. The manual [R Installation and Administration](#) (also contained in the R base sources) explains the process in detail.

[CRAN Task Views](#) allow you to browse packages by topic and provide tools to automatically install all packages for special areas of interest. Currently, 41 views are available.

**Package Check Results**

All packages are tested regularly on machines running [Debian GNU/Linux](#), [Fedora](#), OS X, Solaris and Windows.

**CRAN**

[Mirrors](#)

[What's new?](#)

[Task Views](#)

[Search](#)

*About R*

[R Homepage](#)

[The R Journal](#)

*Software*

[R Sources](#)

[R Binaries](#)

[Packages](#)

[Other](#)

*Documentation*

# Bio-specific R packages are available on Bioconductor



Search:

[Home](#)

[Install](#)

[Help](#)

[Developers](#)

[About](#)

## About Bioconductor

*Bioconductor* provides tools for the analysis and comprehension of high-throughput genomic data.

*Bioconductor* uses the R statistical programming language, and is open source and open development. It has two releases each year, and an active user community. *Bioconductor* is also available as an [AMI](#) (Amazon Machine Image) and [Docker](#) images.

## News

- See our [google calendar](#) for events, conferences, meetings, forums, etc. Add your event with email to events at [bioconductor.org](#).
- *Bioconductor* 3.11 is available.
- Nominate an outstanding community member for a *Bioconductor* Award! See the [support site](#) for more information.
- Registration open for [BioC2020](#).
- Core team **job opportunities** available, contact Martin.Morgan at [RoswellPark.org](#)
- *Bioconductor* [F1000 Research Channel](#) is

## BioC 2020

Get the latest updates on the [BioC 2020 Conference!](#)

- BioC 2020 is going virtual July 27 - July 31. Please see the [Registration Page](#) for more information.
- Nominate an outstanding *Bioconductor* community member for a *Bioconductor* Award! See [posting](#) for more information.
- Call for birds-of-feather, hack-a-thon, and how-to sections. Please see [posting](#) for more information.
- Registration is now open. [Register today.](#)

## Install »

- Discover [1903 software packages](#) available in *Bioconductor* release 3.11.

Get started with *Bioconductor*

- [Install Bioconductor](#)
- [Get support](#)
- [Latest newsletter](#)
- [Follow us on twitter](#)
- [Install R](#)

## Learn »

Master *Bioconductor* tools

- [Courses](#)
- [Support site](#)
- [Package vignettes](#)
- [Literature citations](#)
- [Common work flows](#)
- [FAQ](#)
- [Community resources](#)
- [Videos](#)