Tutorial 1

Introduction to R

A brief expectations management

- The course site is:
  https://webcourse.cs.technion.ac.il/236523/Spring2021/
- The course is 2.5 academic points
- Two hours of lecture and one hour of tutorial a week
- The grade consists of 60% HWs + 40% project
- Four homeworks, that will be submitted in pairs. The submission is at the course site.
- The project is submitted in pairs.
- My email for your questions regarding course administration, material taught at the tutorials and the content of HWs: alonar@technion.ac.il
- All the questions regarding the HW grades should be referred to Mahmoud Yazbek mahmudi@campus.technion.ac.il

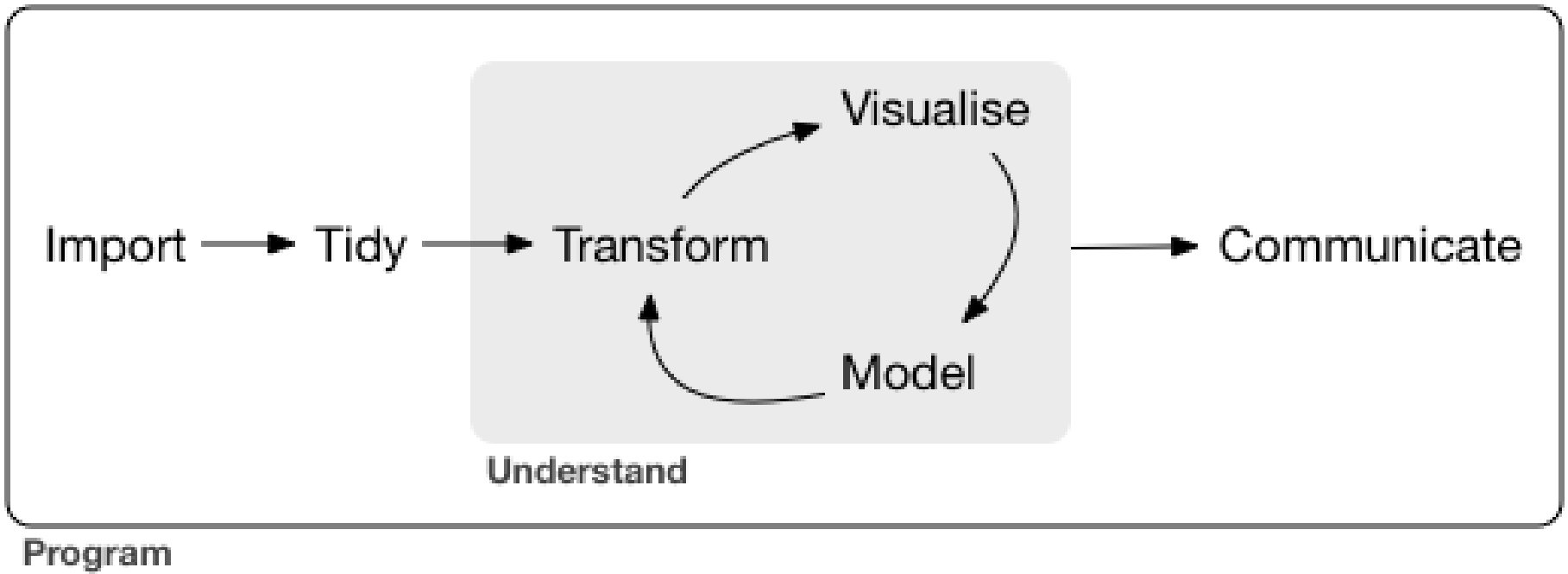
What is R ?

R is a programming language, like C, C++.

R language can be interpreted by computer, however almost everyone uses R with an Rstudio application.

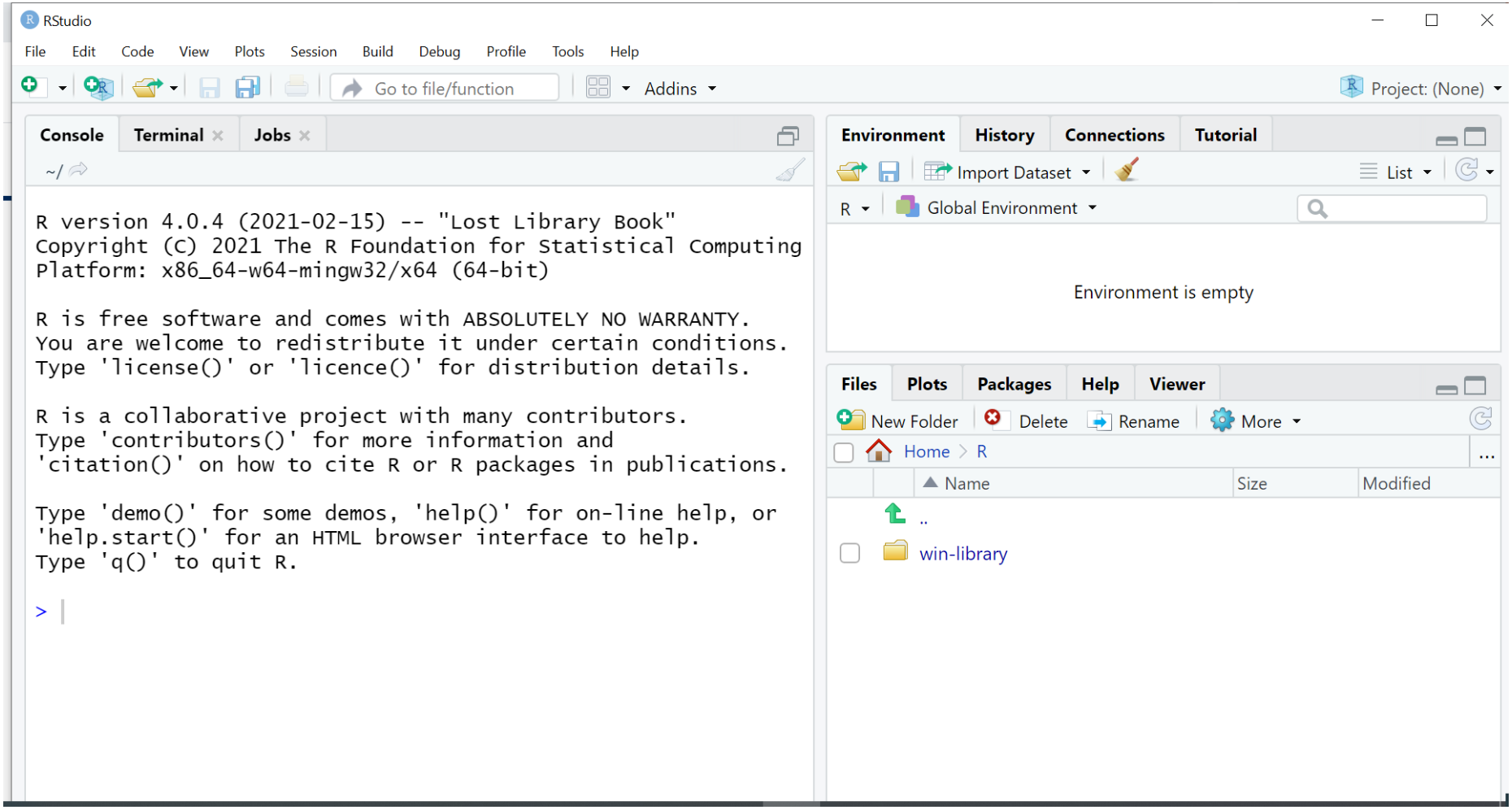
Current version of R: 4.0.4

What are the main data analysis steps ?



https://r4ds.had.co.nz/index.html

How do we do it ?



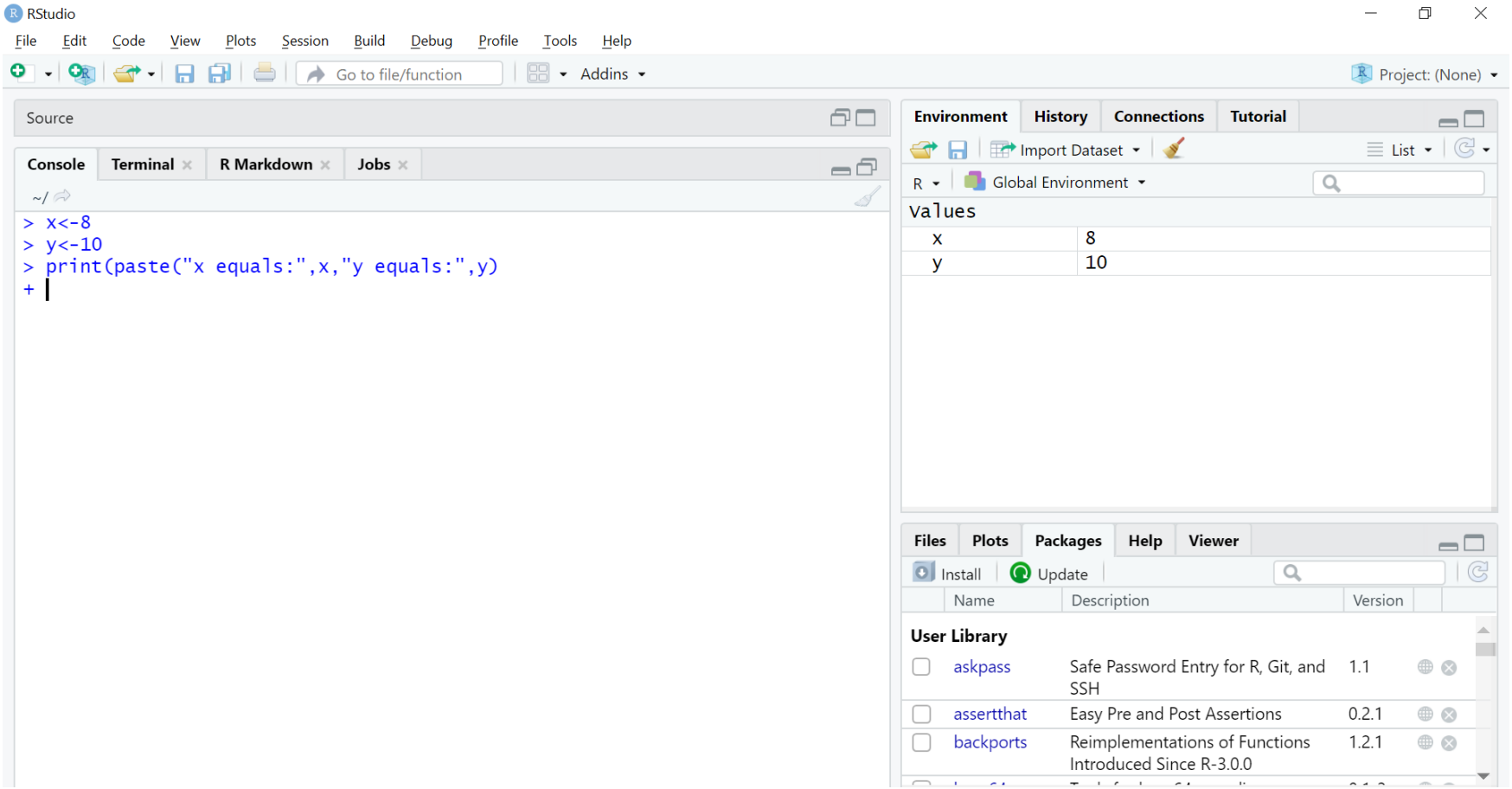
Naming convention

- Must start with letter (A-Z, a-z)
- Can contain letters, digits, underscore, period “.”
- Case sensitive (Cat.Food differs from cat.food)

Assignment

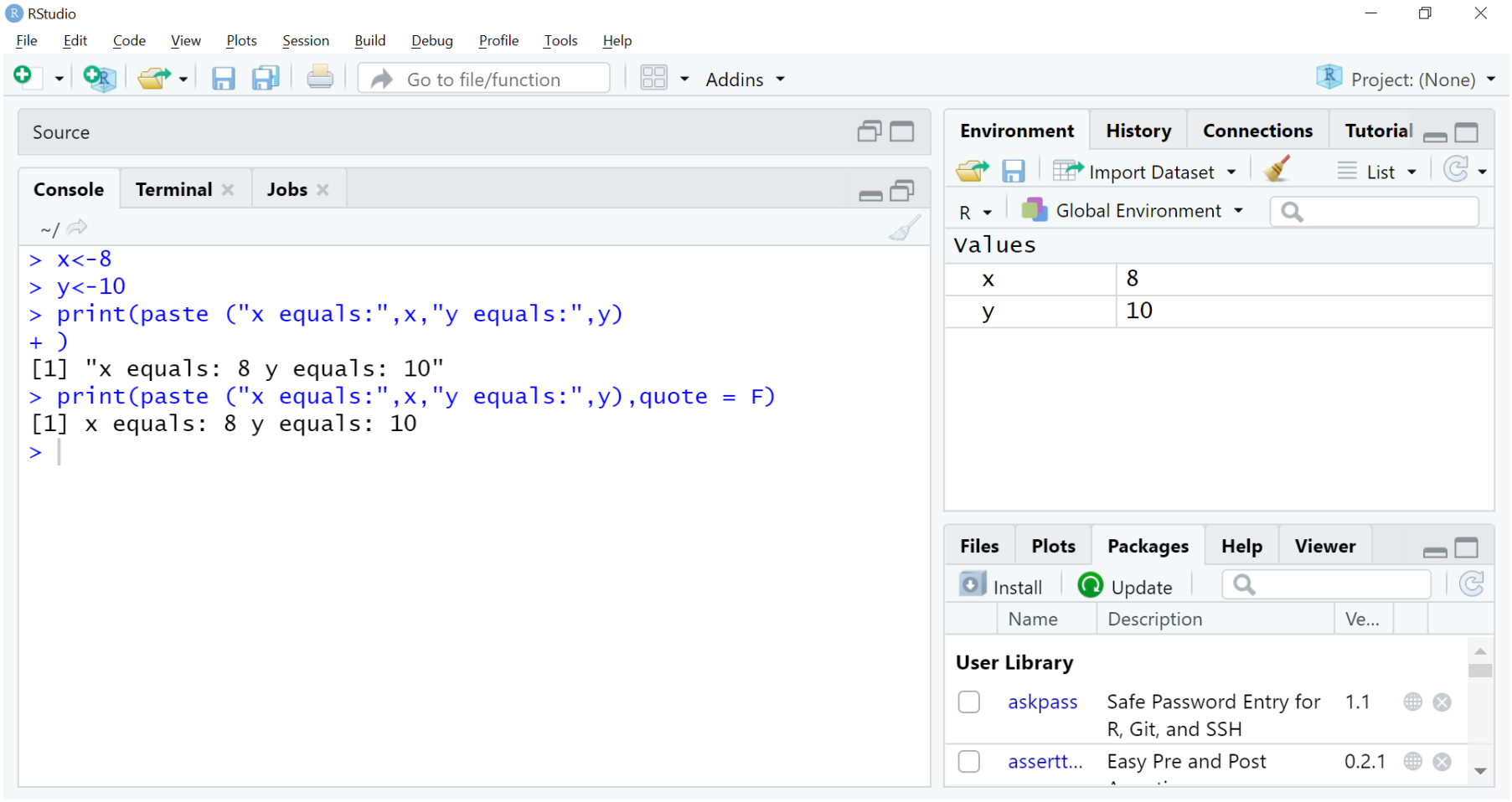
- Assignment is performed by “<-”

object_name <- value



RStudio’s keyboard shortcut: Alt + - (the minus sign) gives <-

Assignment



Ctrl+L clears the console window

Data structures

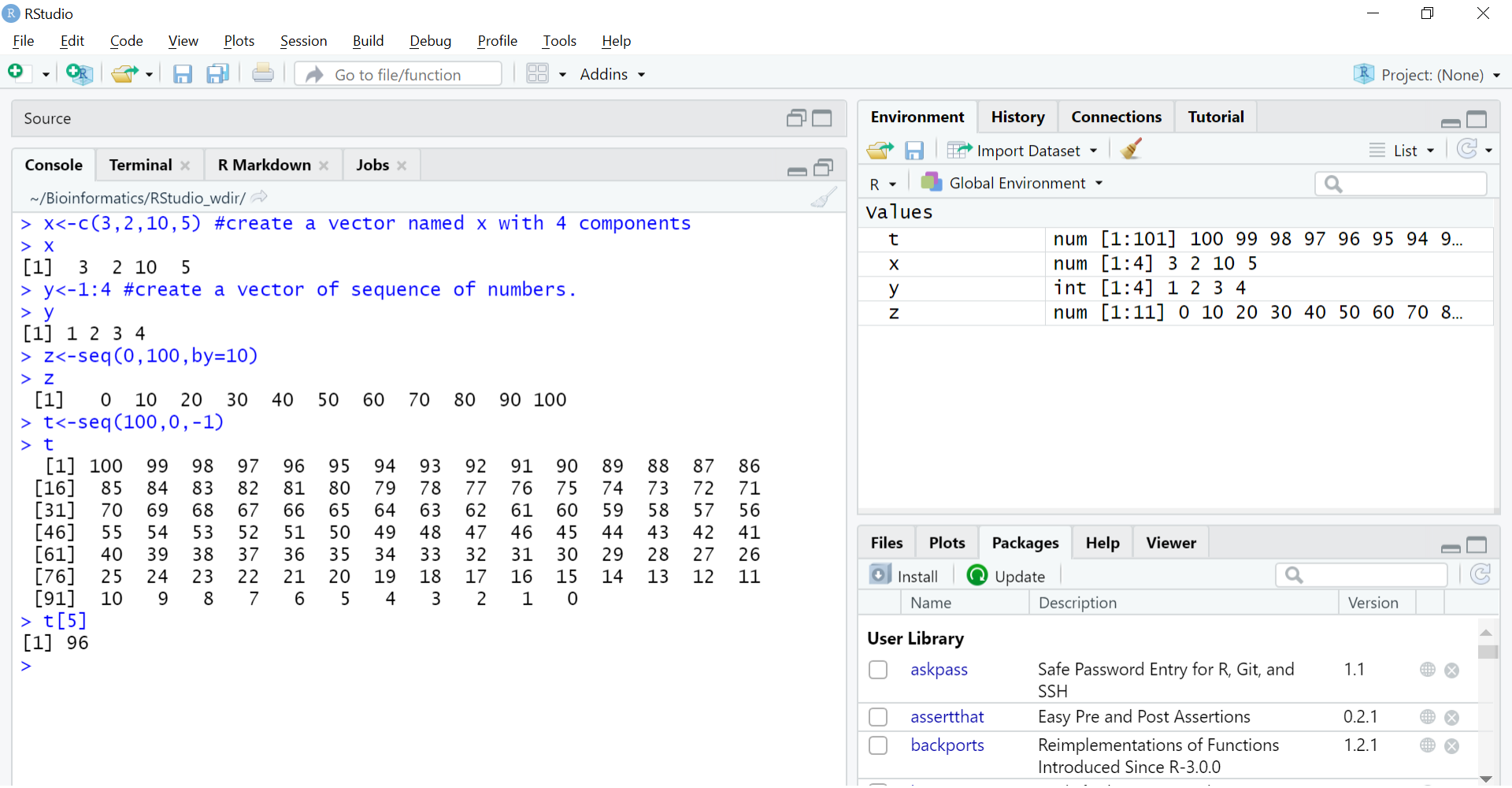
- Vectors
- Matrices
- Data frames
- Lists
- Factors

Data structures

- R has multiple data structures:
- **Vectors**
- Matrices
- Data frames
- Lists
- Factors

Vector is sequence of data elements of the same type (numeric, character or logical). Every column of a table will be represented as a vector.

Vectors

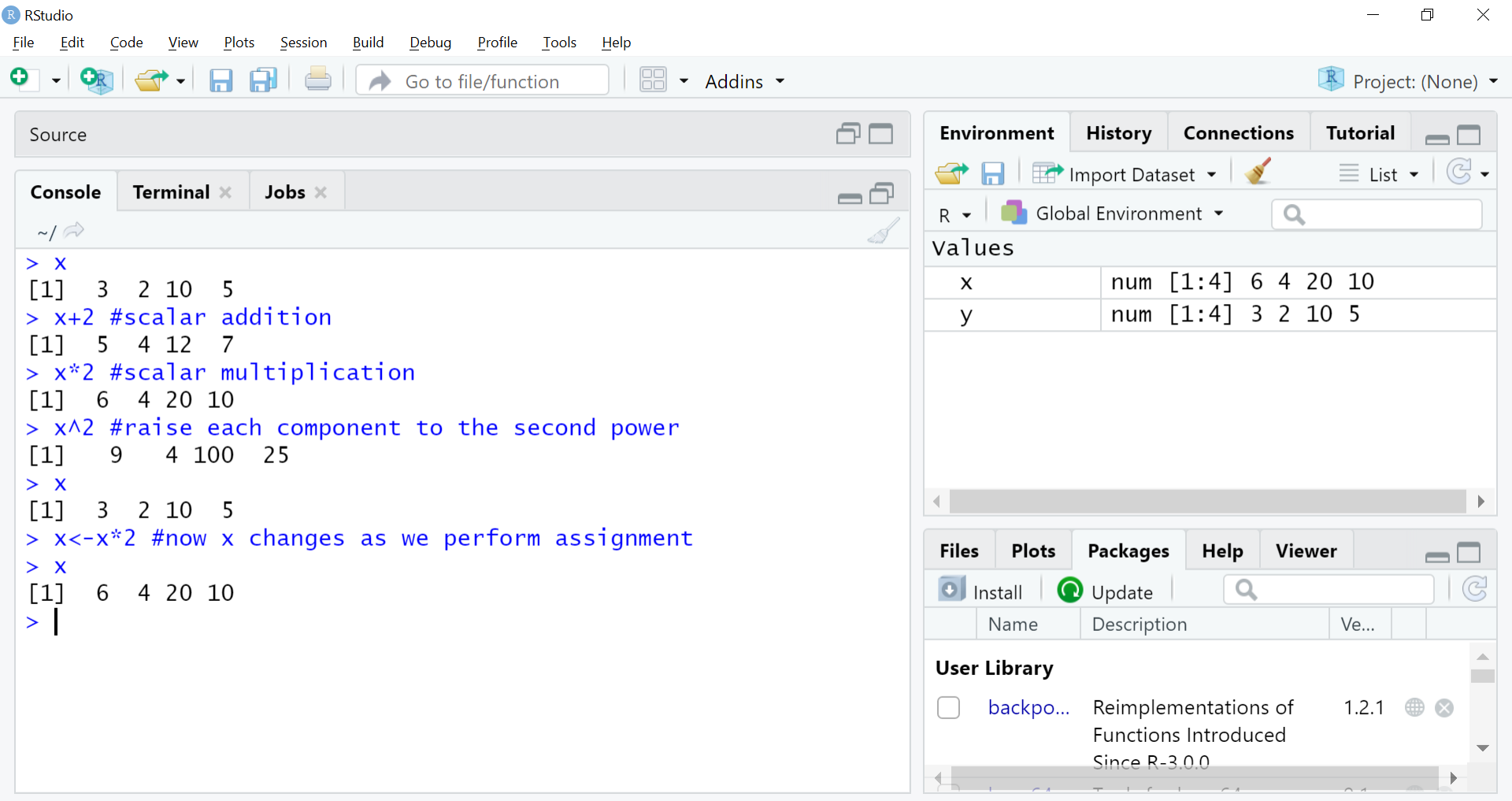


Use # to comment

[1] is indicating that the first element of the vector can be accessed with [1]. Subsequent lines show the appropriate index to access the first number in the line.

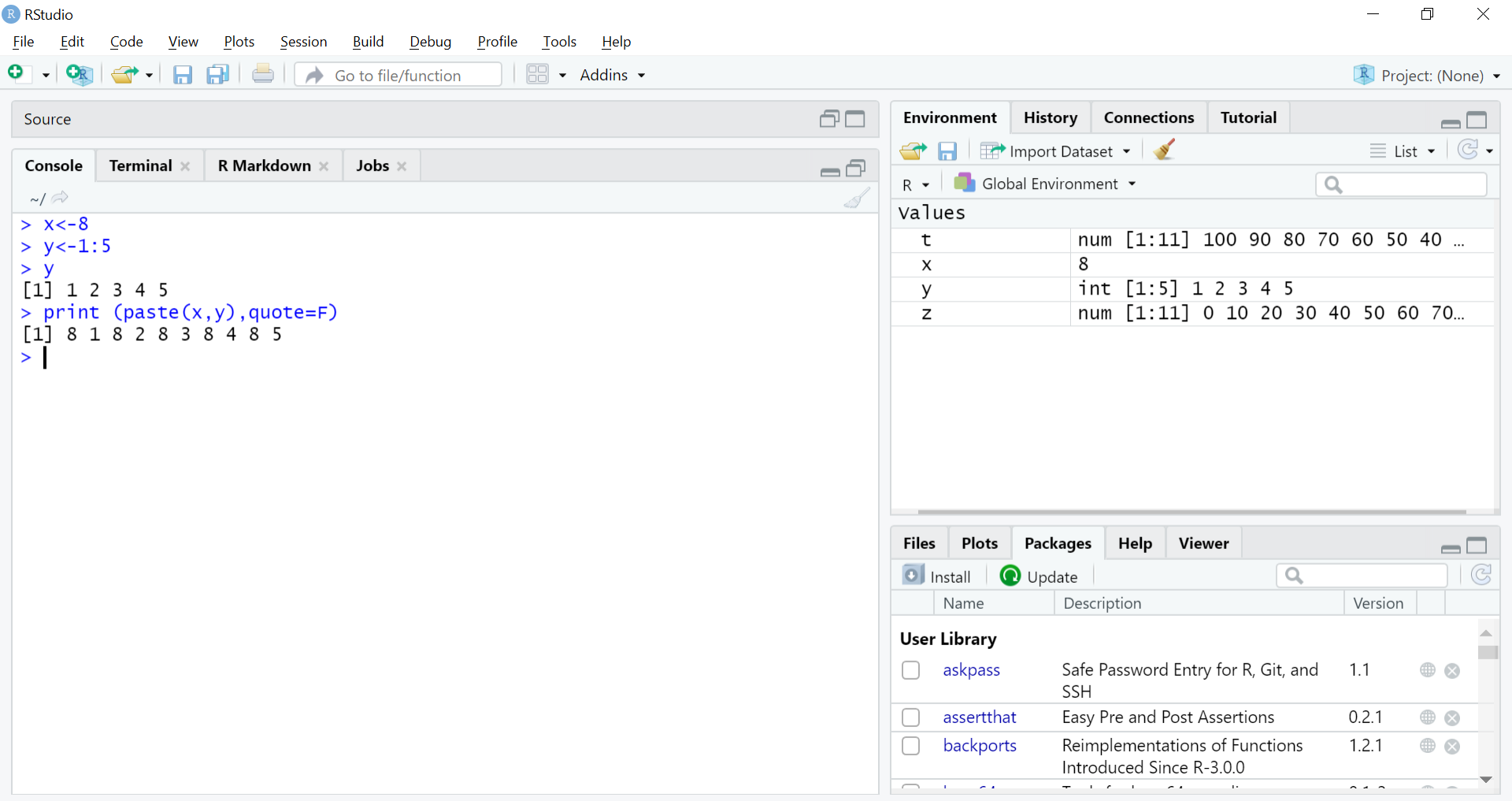
https://monashbioinformaticsplatform.github.io/r-intro/start.html

Some math …

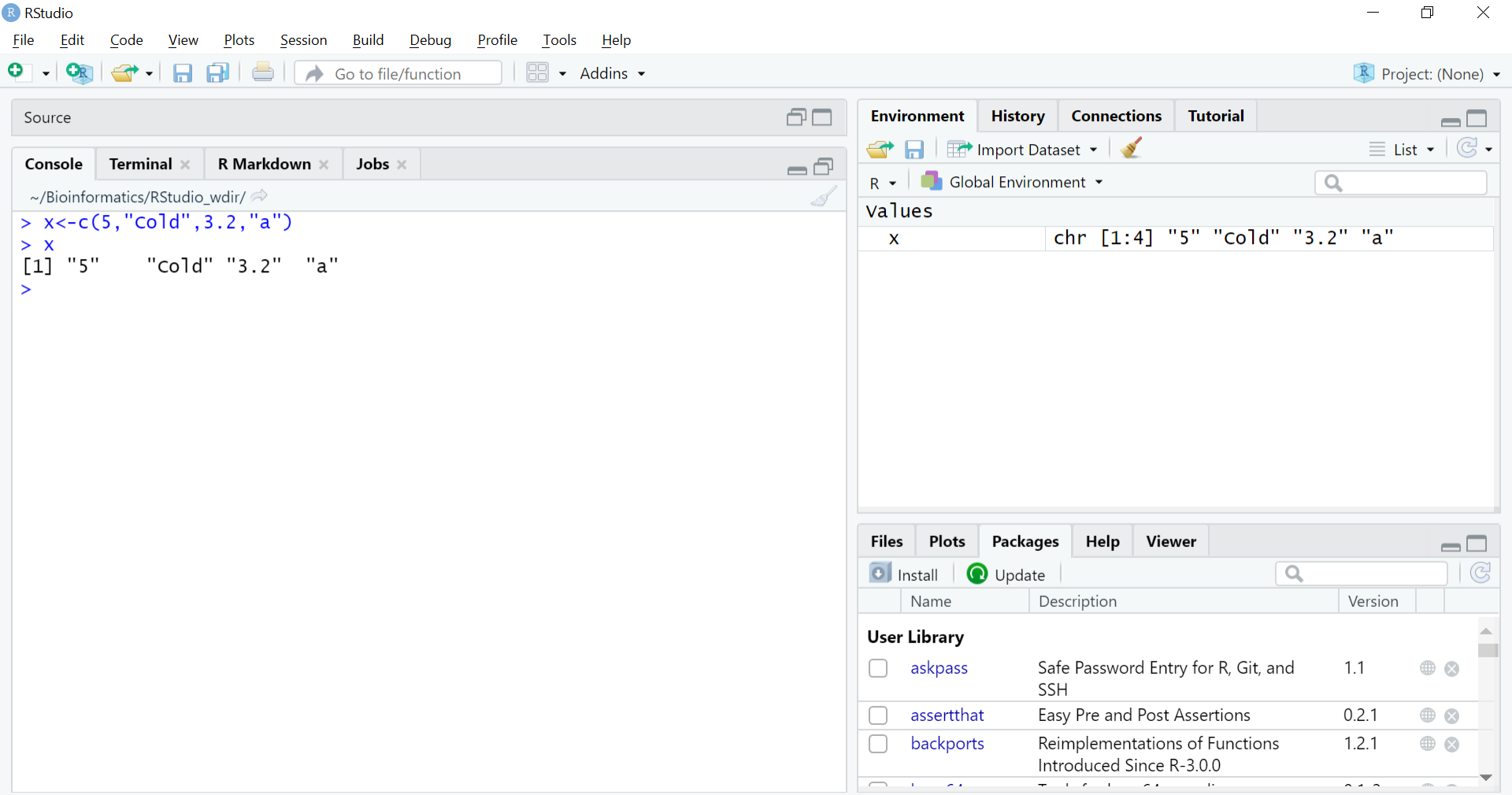


Up arrow gives you the last typed command.

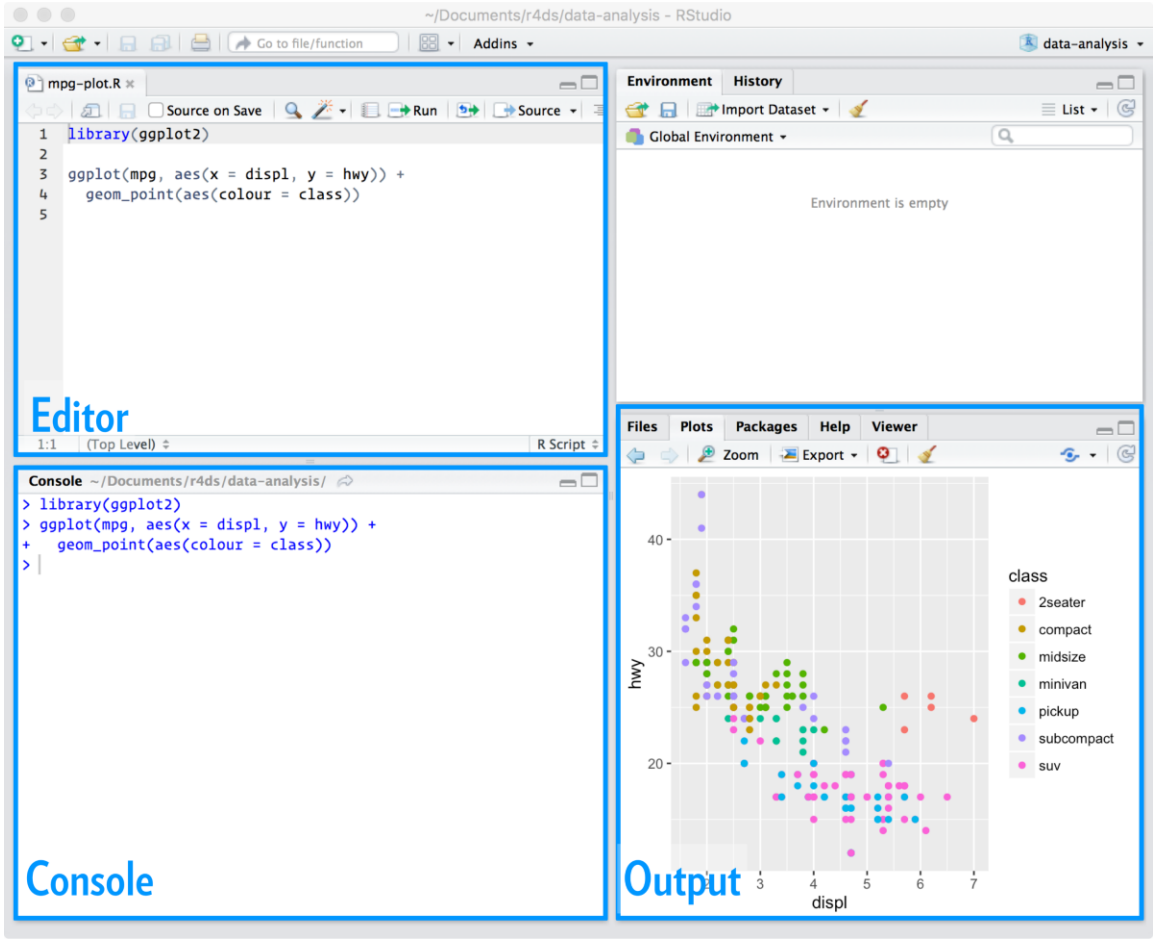
Some math …


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Source
Console Terminal R Markdown Jobs
~/
> x<-8
> y<-1:5
> y
[1] 1 2 3 4 5
> print (paste(x,y),quote=F)
[1] 8 1 8 2 8 3 8 4 8 5
>
Environment History Connections Tutorial
Import Dataset
List
R
Global Environment
Values
t num [1:11] 100 90 80 70 60 50 40 …
x 8
y int [1:5] 1 2 3 4 5
z num [1:11] 0 10 20 30 40 50 60 70…
Files Plots Packages Help Viewer
Install
Update
Name Description Version
User Library
askpass Safe Password Entry for R, Git, and SSH 1.1
assertthat Easy Pre and Post Assertions 0.2.1
backports Reimplementations of Functions Introduced Since R-3.0.0 1.2.1

Assignment of mixed data types



Script editor vs console



- Ctrl + Enter shortcut allows you to execute every line/code chunk at the script editor.

https://r4ds.had.co.nz/workflow-scripts.html

Script editor vs console

- Instead of executing expression-by-expression, you can also execute the complete script in one step using Ctrl+Shift+Enter.

- Start your script with the packages that you need. If you'll share your code with others, they can easily see what packages they need to install/load.

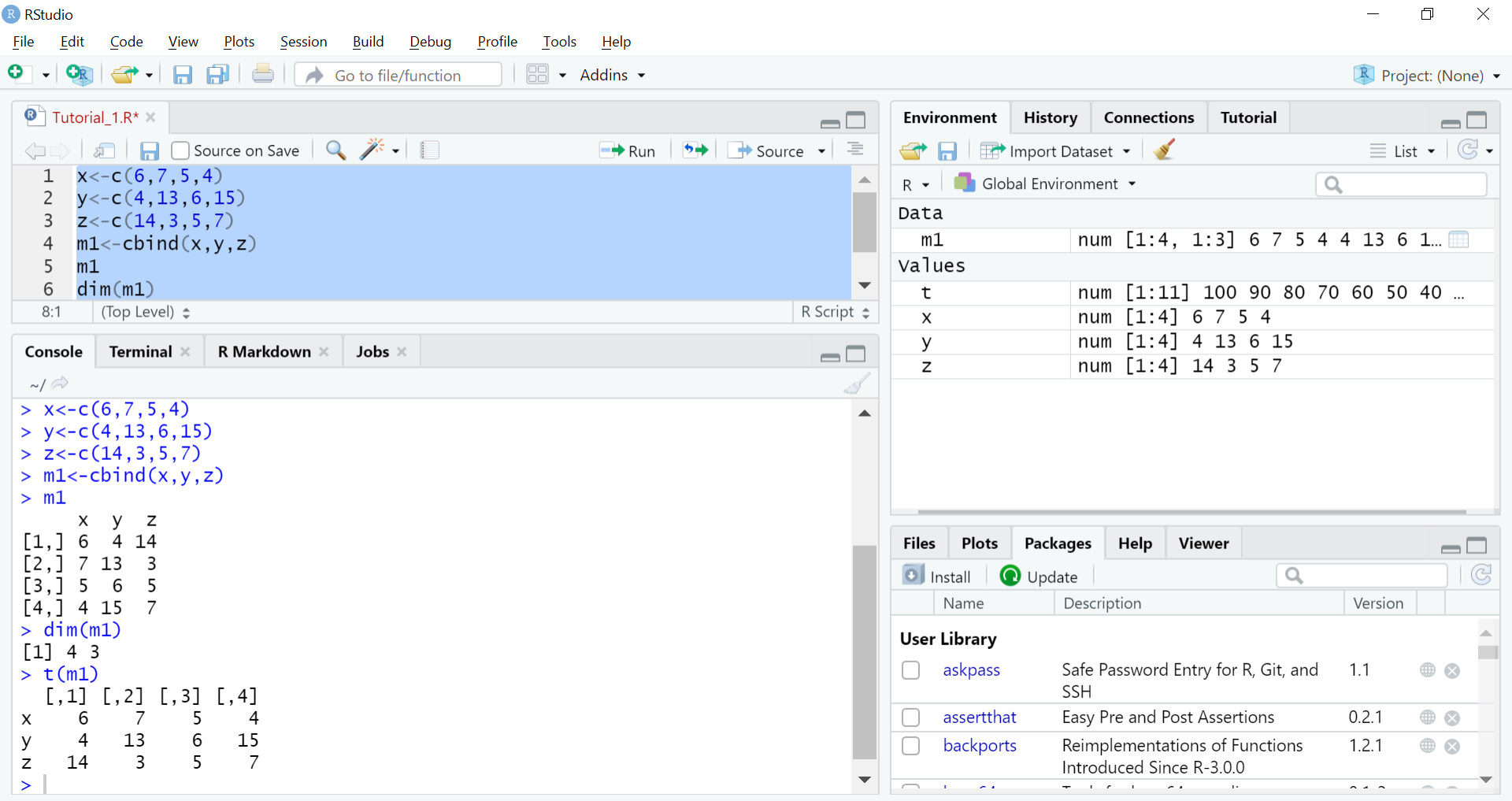
- You can use a console to execute parts of code, or you can use it observe the results of the script/parts of script execution

Data structures

- R has multiple data structures:
- Vectors
- **Matrices**
- Data frames
- Lists
- Factors

- A matrix is an array of rows and columns of the same data type. One of the easiest ways to create a matrix is to combine vectors of equal length using cbind() function.

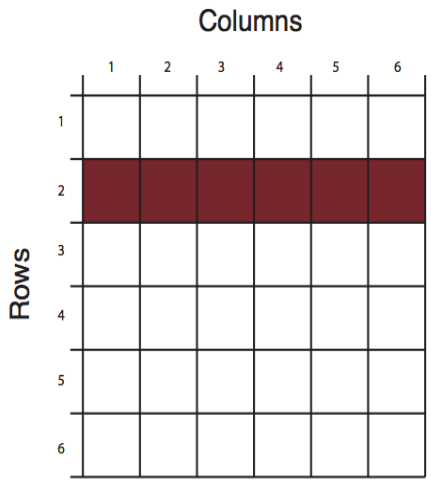
Matrix


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Tutorial_1.R*
Source on Save
Run
Source
1 x<-c(6,7,5,4)
2 y<-c(4,13,6,15)
3 z<-c(14,3,5,7)
4 m1<-cbind(x,y,z)
5 m1
6 dim(m1)
8:1 (Top Level)
R Script
Console
Terminal
R Markdown
Jobs
~/
> x<-c(6,7,5,4)
> y<-c(4,13,6,15)
> z<-c(14,3,5,7)
> m1<-cbind(x,y,z)
> m1
x y z
[1,] 6 4 14
[2,] 7 13 3
[3,] 5 6 5
[4,] 4 15 7
> dim(m1)
[1] 4 3
> t(m1)
[,1] [,2] [,3] [,4]
x 6 7 5 4
y 4 13 6 15
z 14 3 5 7
>
Environment
History
Connections
Tutorial
Import Dataset
List
R
Global Environment
Data
m1 num [1:4, 1:3] 6 7 5 4 4 13 6 1...
Values
t num [1:11] 100 90 80 70 60 50 40 ...
x num [1:4] 6 7 5 4
y num [1:4] 4 13 6 15
z num [1:4] 14 3 5 7
Files
Plots
Packages
Help
Viewer
Install
Update
Name
Description
Version
User Library
askpass Safe Password Entry for R, Git, and SSH 1.1
assertthat Easy Pre and Post Assertions 0.2.1
backports Reimplementations of Functions Introduced Since R-3.0.0 1.2.1

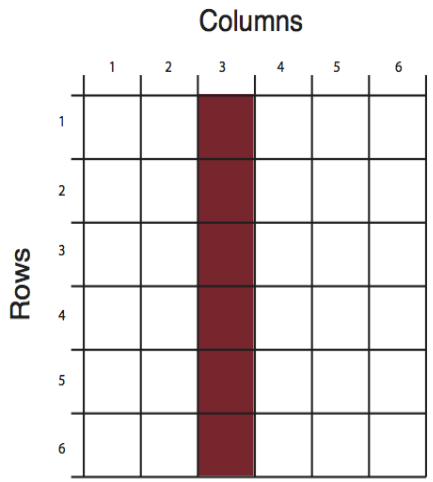
Matrix

Matrices and data frames, are tabular data structures. These are two dimensional datasets in which data in the any specific column is of the same type. You can subset those data structures using [] and providing desired rows and columns to subset.

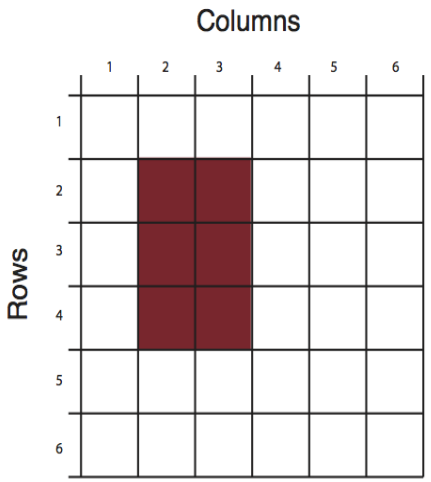
`mat[2,]`



`mat[,3]`

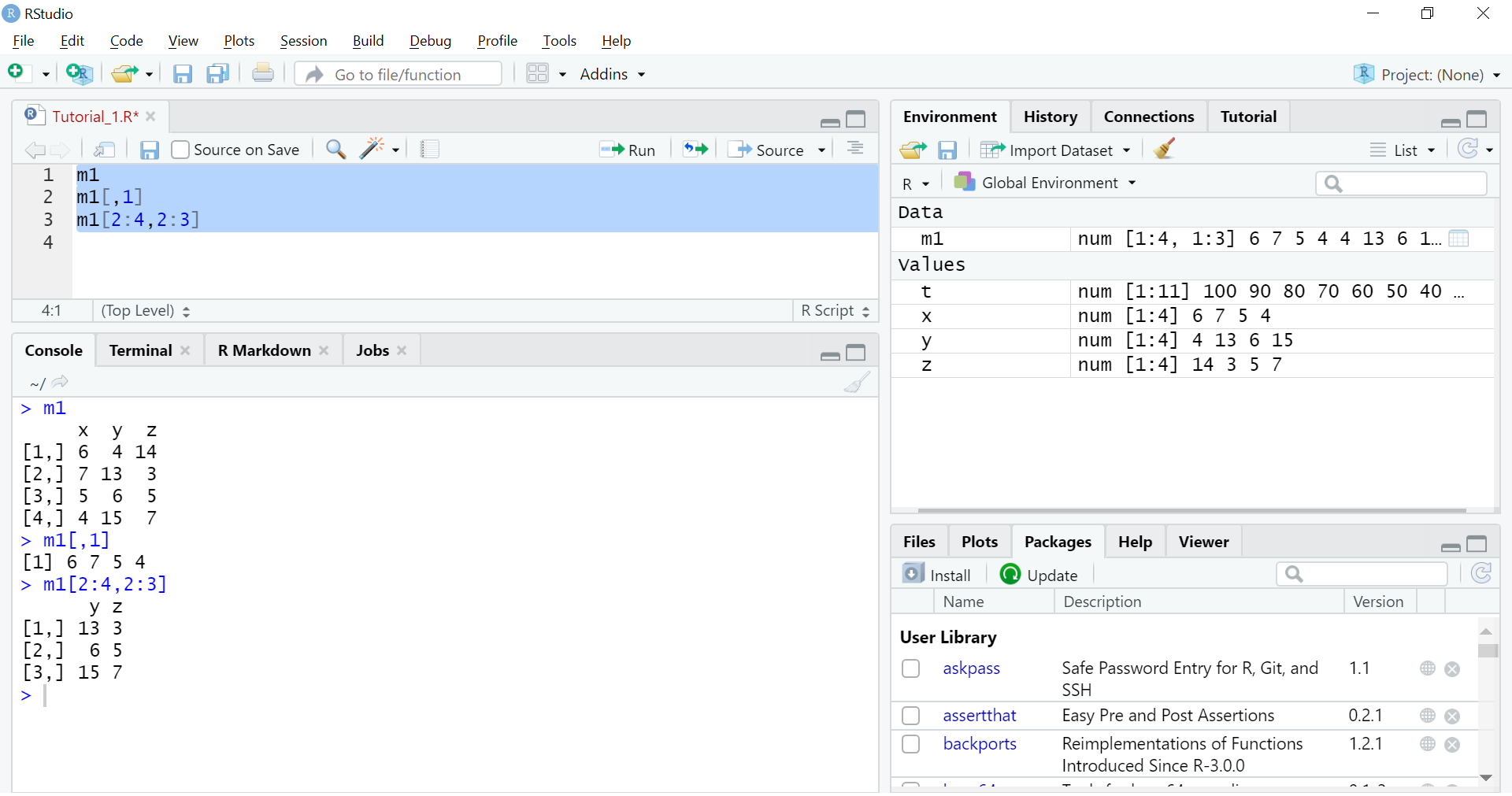


`mat[2:4,2:3]`

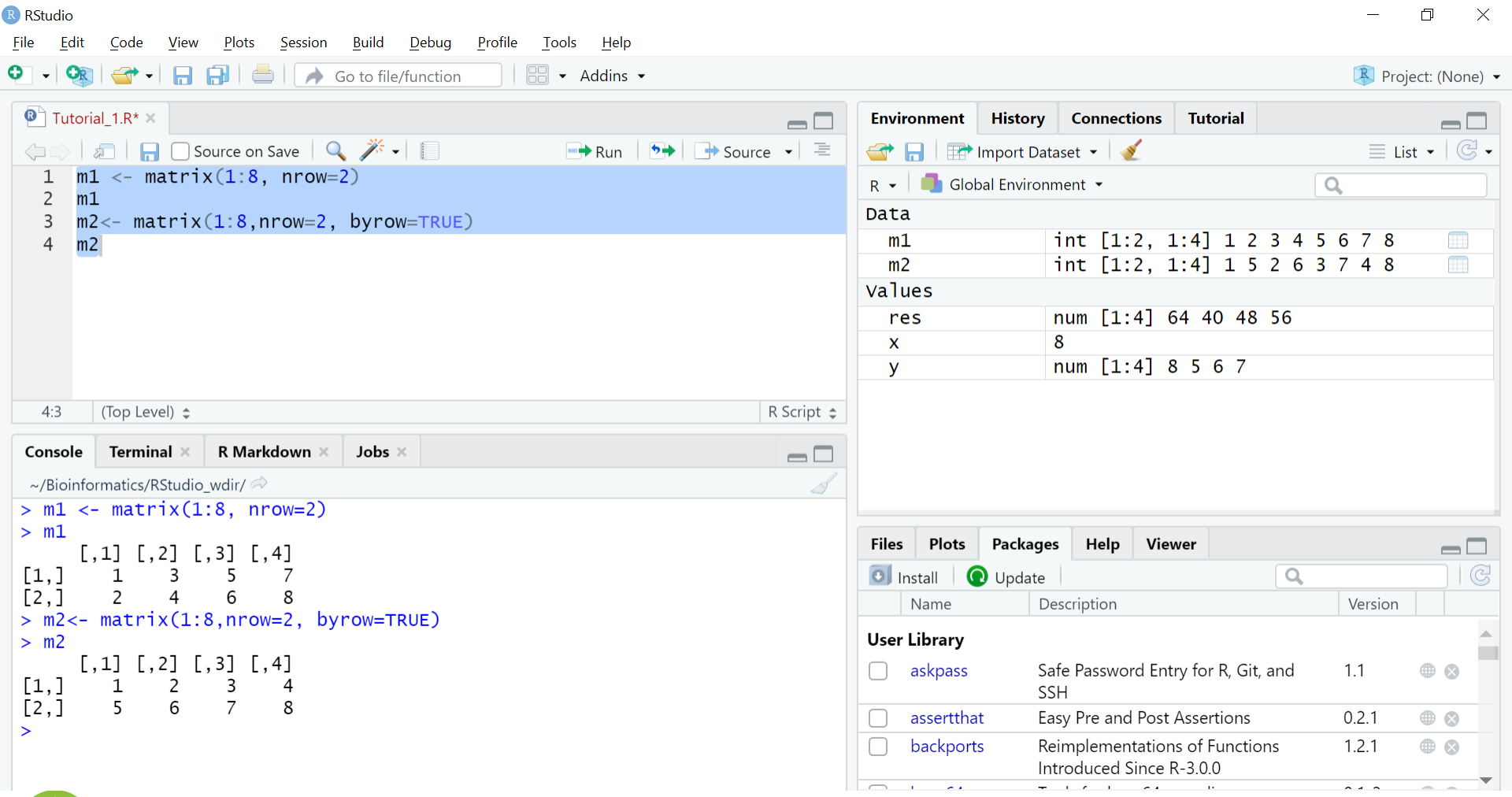


https://compgenomr.github.io/book/data-structures.html#matrices

Matrix


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Tutorial_1.R*
Source on Save
Run
Source
1 m1
2 m1[,1]
3 m1[2:4,2:3]
4
4:1
(Top Level)
R Script
Console
Terminal
R Markdown
Jobs
~/
> m1
x y z
[1,] 6 4 14
[2,] 7 13 3
[3,] 5 6 5
[4,] 4 15 7
> m1[,1]
[1] 6 7 5 4
> m1[2:4,2:3]
y z
[1,] 13 3
[2,] 6 5
[3,] 15 7
>
Environment
History
Connections
Tutorial
Import Dataset
List
R
Global Environment
Data
m1 num [1:4, 1:3] 6 7 5 4 4 13 6 1...
Values
t num [1:11] 100 90 80 70 60 50 40 ...
x num [1:4] 6 7 5 4
y num [1:4] 4 13 6 15
z num [1:4] 14 3 5 7
Files
Plots
Packages
Help
Viewer
Install
Update
Name
Description
Version
User Library
askpass
Safe Password Entry for R, Git, and SSH
1.1
assertthat
Easy Pre and Post Assertions
0.2.1
backports
Reimplementations of Functions Introduced Since R-3.0.0
1.2.1

Matrix



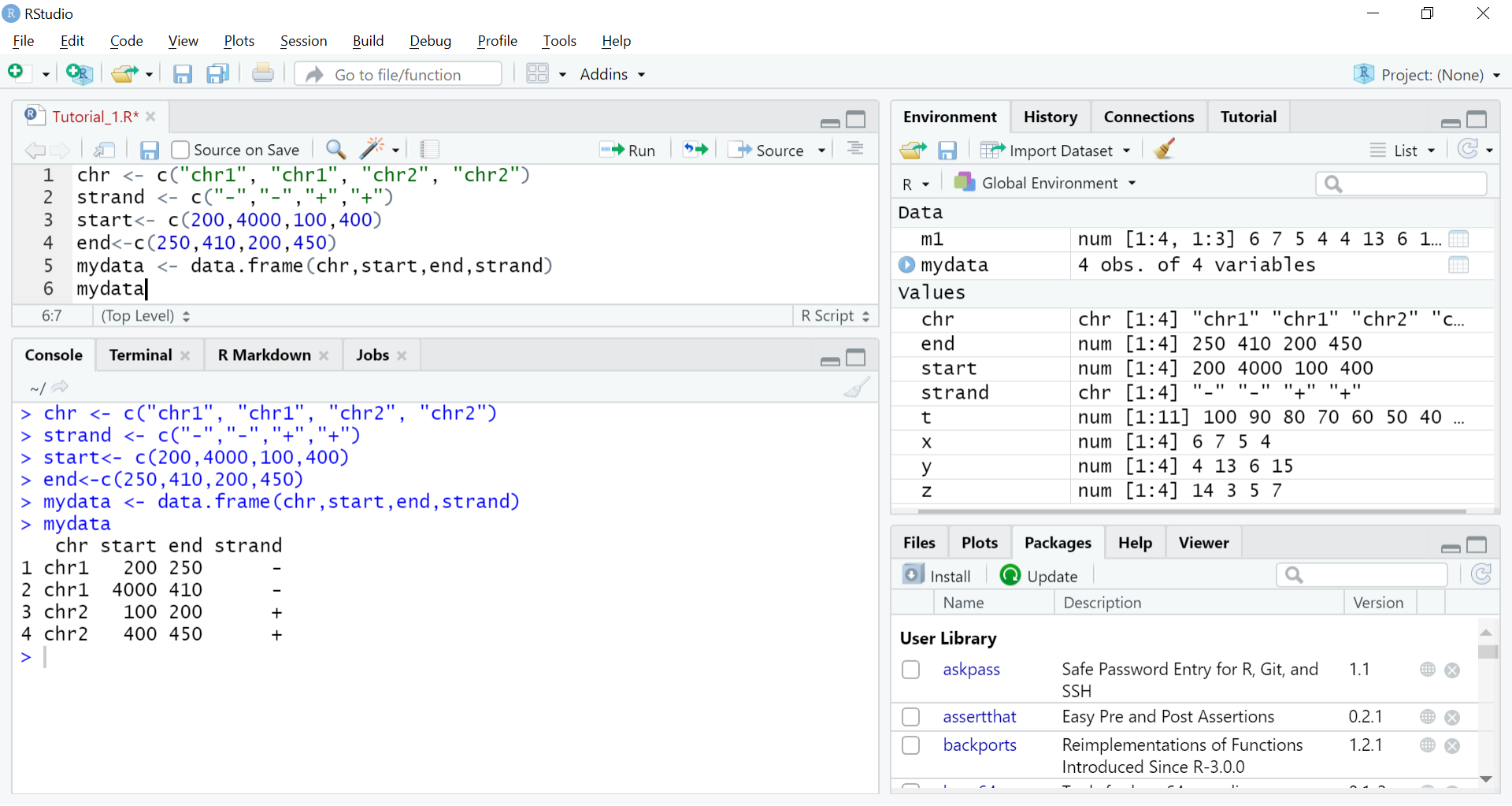
By default, the matrix is filled column-by-column. If you set byrow=TRUE the matrix will be filled row by row

Data structures

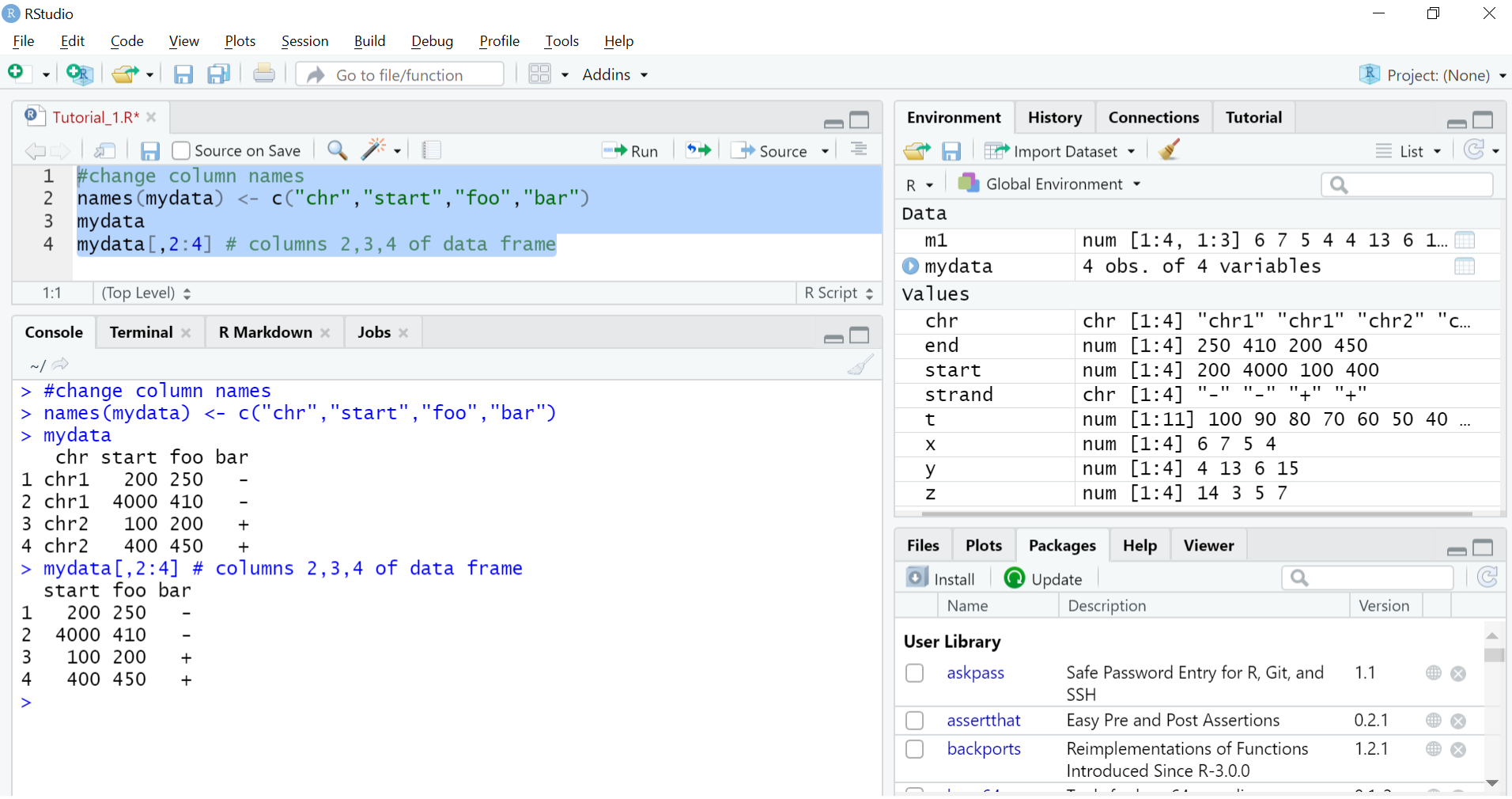
- R has multiple data structures:
- Vectors
- Matrices
- **Data frames**
- Lists
- Factors

- A data frame is a nxm collection of variables (in the columns) and observations (in the rows). Data frame is more general data structure then matrix, as its columns can have different data types (numeric, character, factor, etc.)

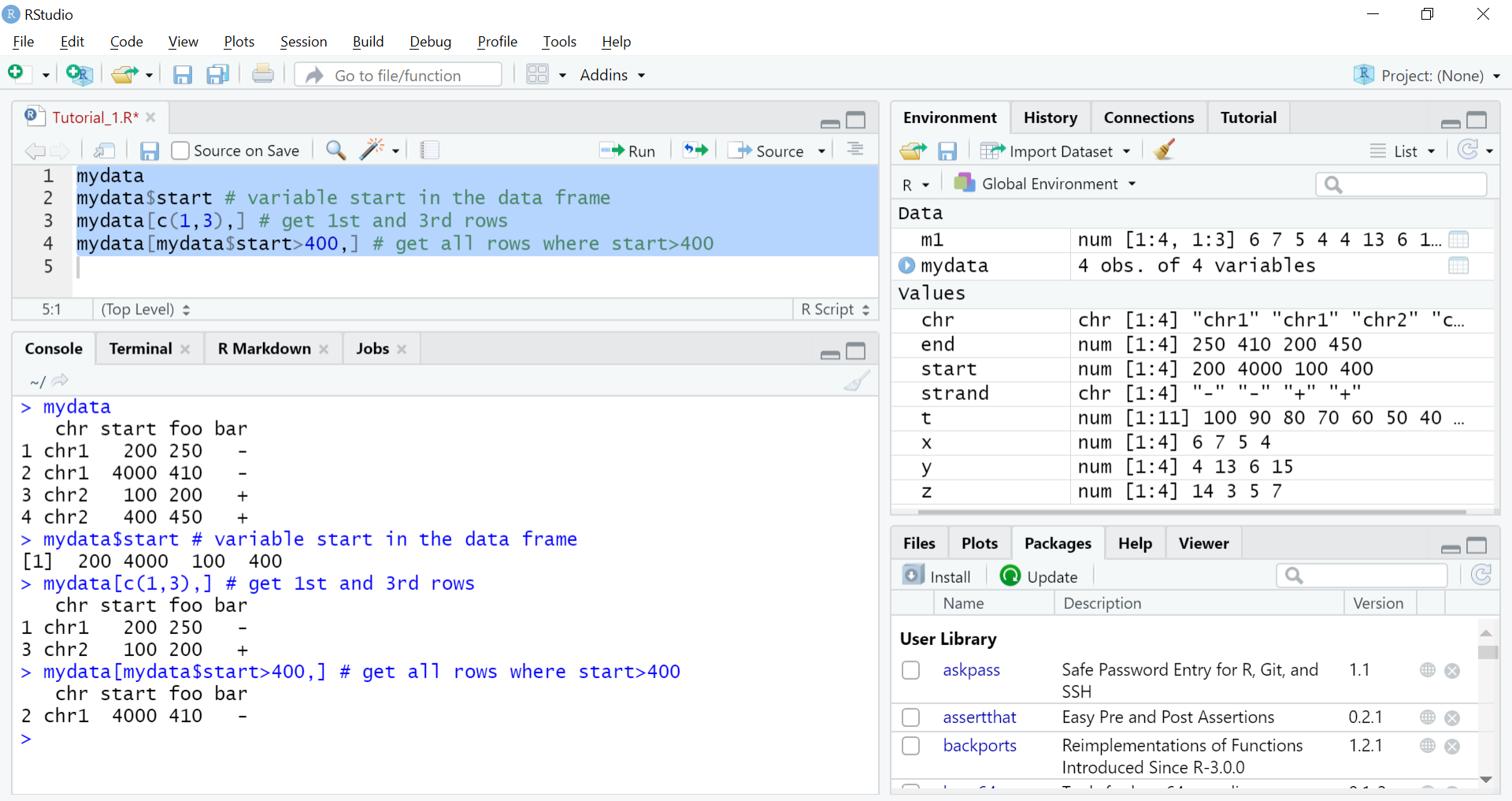
Data frame


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Tutorial_1.R*
Source on Save
Run
Source
1 chr <- c("chr1", "chr1", "chr2", "chr2")
2 strand <- c("-","-","+","+")
3 start<- c(200,4000,100,400)
4 end<-c(250,410,200,450)
5 mydata <- data.frame(chr,start,end,strand)
6 mydata
6:7 (Top Level) R Script
Console Terminal R Markdown Jobs
~/
> chr <- c("chr1", "chr1", "chr2", "chr2")
> strand <- c("-","-","+","+")
> start<- c(200,4000,100,400)
> end<-c(250,410,200,450)
> mydata <- data.frame(chr,start,end,strand)
> mydata
chr start end strand
1 chr1 200 250 -
2 chr1 4000 410 -
3 chr2 100 200 +
4 chr2 400 450 +
Environment History Connections Tutorial
Import Dataset
List
R
Global Environment
Data
m1 num [1:4, 1:3] 6 7 5 4 4 13 6 1...
mydata 4 obs. of 4 variables
Values
chr chr [1:4] "chr1" "chr1" "chr2" "c...
end num [1:4] 250 410 200 450
start num [1:4] 200 4000 100 400
strand chr [1:4] "-" "-" "+" "+"
t num [1:11] 100 90 80 70 60 50 40 ...
x num [1:4] 6 7 5 4
y num [1:4] 4 13 6 15
z num [1:4] 14 3 5 7
Files Plots Packages Help Viewer
Install Update
Name Description Version
User Library
askpass Safe Password Entry for R, Git, and SSH 1.1
assertthat Easy Pre and Post Assertions 0.2.1
backports Reimplementations of Functions Introduced Since R-3.0.0 1.2.1

Data frame


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Tutorial_1.R*
Source on Save
Run
Source
1 #change column names
2 names(mydata) <- c("chr","start","foo","bar")
3 mydata
4 mydata[,2:4] # columns 2,3,4 of data frame
1:1
(Top Level)
R Script
Console
Terminal
R Markdown
Jobs
~/
> #change column names
> names(mydata) <- c("chr","start","foo","bar")
> mydata
chr start foo bar
1 chr1 200 250 -
2 chr1 4000 410 -
3 chr2 100 200 +
4 chr2 400 450 +
> mydata[,2:4] # columns 2,3,4 of data frame
start foo bar
1 200 250 -
2 4000 410 -
3 100 200 +
4 400 450 +
>
Environment
History
Connections
Tutorial
Import Dataset
List
R
Global Environment
Data
m1 num [1:4, 1:3] 6 7 5 4 4 13 6 1...
mydata 4 obs. of 4 variables
Values
chr chr [1:4] "chr1" "chr1" "chr2" "c...
end num [1:4] 250 410 200 450
start num [1:4] 200 4000 100 400
strand chr [1:4] "-" "-" "+" "+"
t num [1:11] 100 90 80 70 60 50 40 ...
x num [1:4] 6 7 5 4
y num [1:4] 4 13 6 15
z num [1:4] 14 3 5 7
Files
Plots
Packages
Help
Viewer
Install
Update
Name
Description
Version
User Library
askpass Safe Password Entry for R, Git, and SSH 1.1
assertthat Easy Pre and Post Assertions 0.2.1
backports Reimplementations of Functions Introduced Since R-3.0.0 1.2.1

Data frame

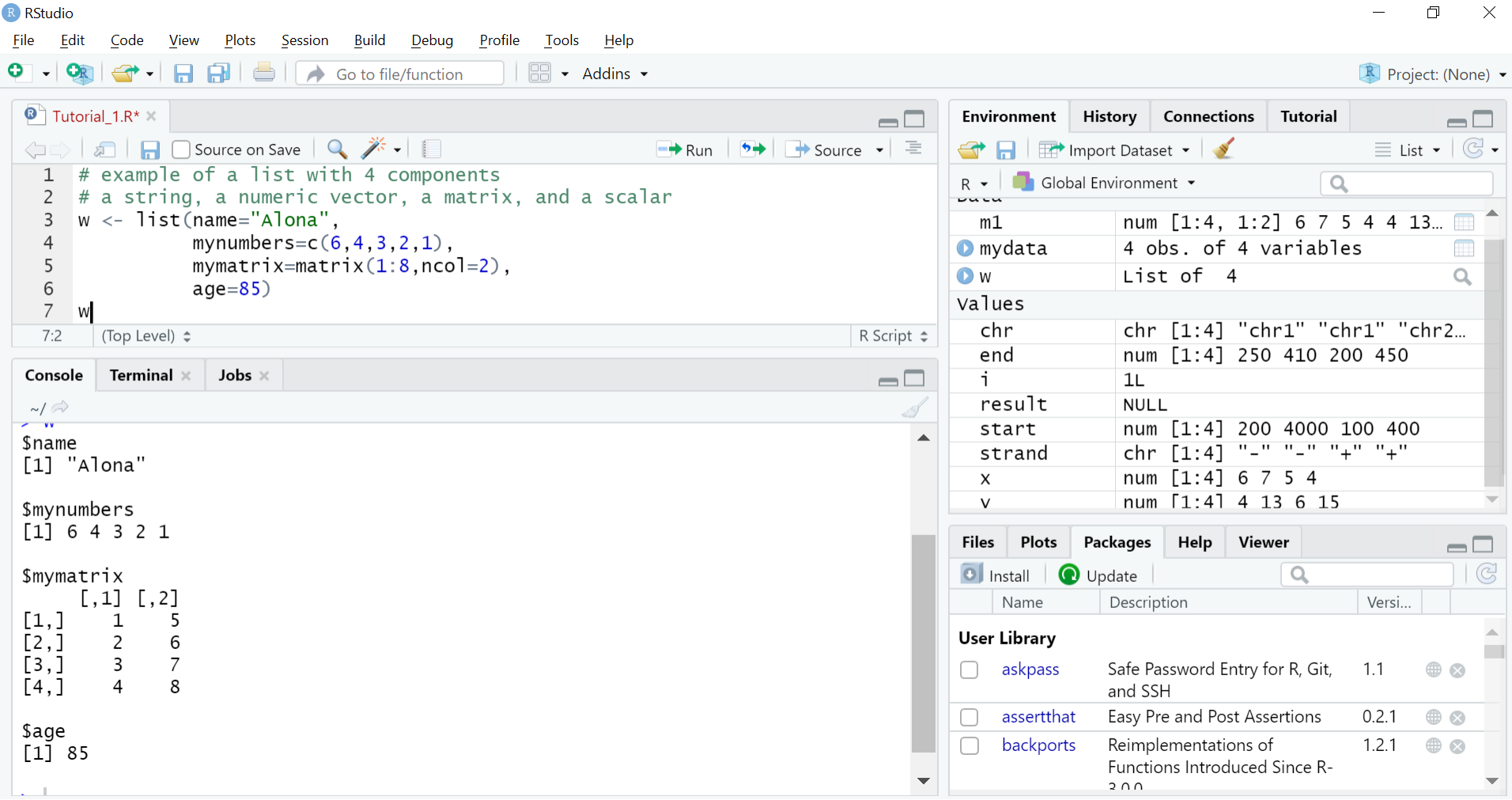

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Tutorial_1.R*
Source on Save
Run
Source
1 mydata
2 mydata$start # variable start in the data frame
3 mydata[c(1,3),] # get 1st and 3rd rows
4 mydata[mydata$start>400,] # get all rows where start>400
5
5:1
(Top Level)
R Script
Console
Terminal
R Markdown
Jobs
~/
> mydata
chr start foo bar
1 chr1 200 250 -
2 chr1 4000 410 -
3 chr2 100 200 +
4 chr2 400 450 +
> mydata$start # variable start in the data frame
[1] 200 4000 100 400
> mydata[c(1,3),] # get 1st and 3rd rows
chr start foo bar
1 chr1 200 250 -
3 chr2 100 200 +
> mydata[mydata$start>400,] # get all rows where start>400
chr start foo bar
2 chr1 4000 410 -
>
Environment
History
Connections
Tutorial
Import Dataset
List
R
Global Environment
Data
m1 num [1:4, 1:3] 6 7 5 4 4 13 6 1...
mydata 4 obs. of 4 variables
Values
chr chr [1:4] "chr1" "chr1" "chr2" "c...
end num [1:4] 250 410 200 450
start num [1:4] 200 4000 100 400
strand chr [1:4] "-" "-" "+" "+"
t num [1:11] 100 90 80 70 60 50 40 ...
x num [1:4] 6 7 5 4
y num [1:4] 4 13 6 15
z num [1:4] 14 3 5 7
Files
Plots
Packages
Help
Viewer
Install
Update
Name
Description
Version
User Library
askpass Safe Password Entry for R, Git, and SSH 1.1
assertthat Easy Pre and Post Assertions 0.2.1
backports Reimplementations of Functions Introduced Since R-3.0.0 1.2.1

Data structures

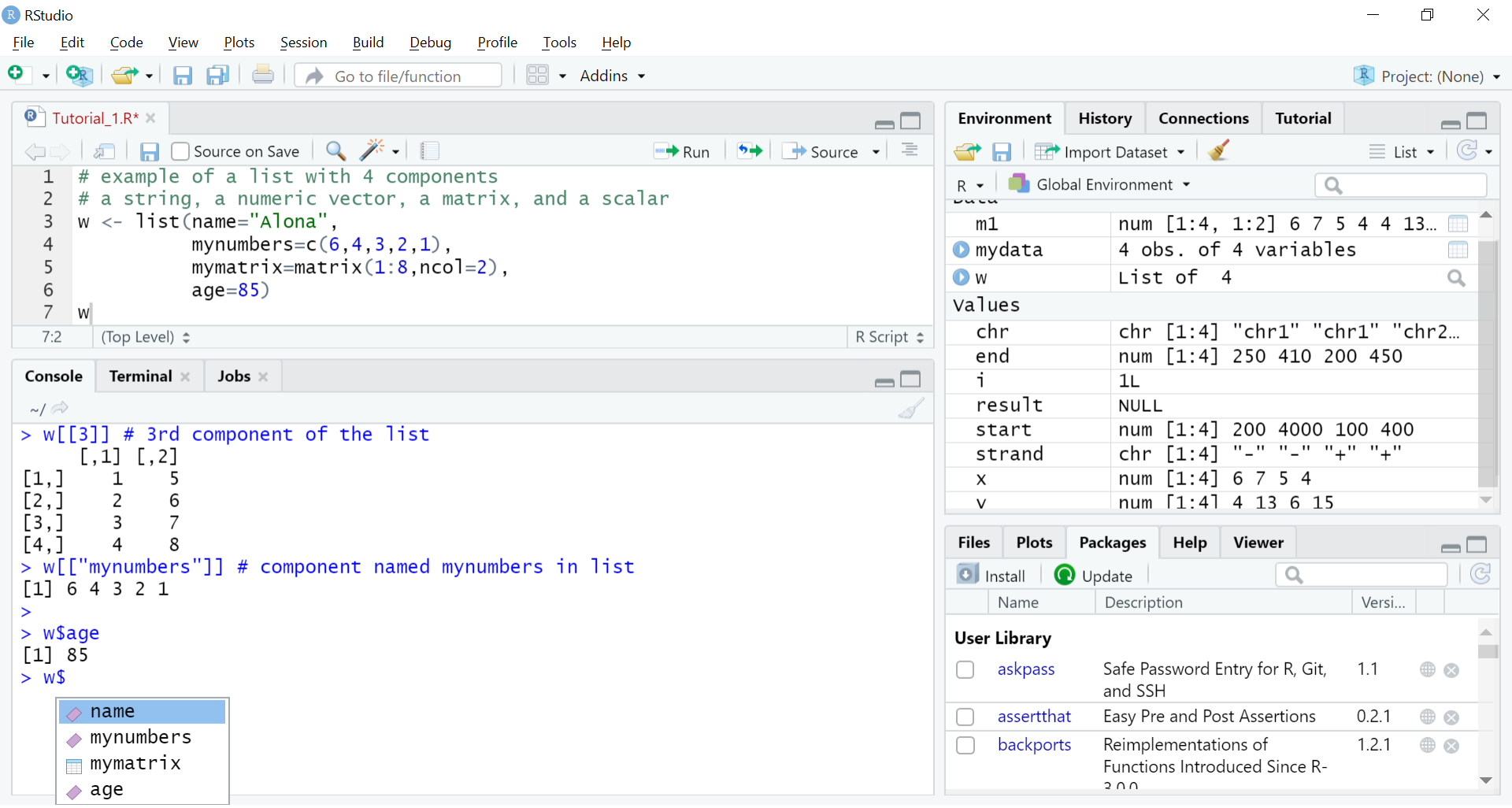
- R has multiple data structures:
- Vectors
- Matrices
- Data frames
- **Lists**
- Factors

- A list is an ordered collection of components. A list allows you to gather a variety of often unrelated objects of different length and type under one name. Each object or element in the list has a numbered position and can have names. The list can be created with the list() function.

Lists


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Tutorial_1.R*
Source on Save
Run
Source
1 # example of a list with 4 components
2 # a string, a numeric vector, a matrix, and a scalar
3 w <- list(name="Alona",
4 mynumbers=c(6,4,3,2,1),
5 mymatrix=matrix(1:8,ncol=2),
6 age=85)
7 w
7:2 (Top Level) R Script
Console Terminal Jobs
~/
$name
[1] "Alona"
$mynumbers
[1] 6 4 3 2 1
$mymatrix
[,1] [,2]
[1,] 1 5
[2,] 2 6
[3,] 3 7
[4,] 4 8
$age
[1] 85
Environment History Connections Tutorial
Import Dataset
List
R
Global Environment
m1 num [1:4, 1:2] 6 7 5 4 4 13...
mydata 4 obs. of 4 variables
w List of 4
Values
chr chr [1:4] "chr1" "chr1" "chr2...
end num [1:4] 250 410 200 450
i 1L
result NULL
start num [1:4] 200 4000 100 400
strand chr [1:4] "-" "-" "+" "+"
x num [1:4] 6 7 5 4
v num [1:4] 4 13 6 15
Files Plots Packages Help Viewer
Install
Update
Name
Description
Versi...
User Library
askpass
Safe Password Entry for R, Git, and SSH
1.1
assertthat
Easy Pre and Post Assertions
0.2.1
backports
Reimplementations of Functions Introduced Since R-
1.2.1

Lists


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Tutorial_1.R*
Source on Save
Run
Source
1 # example of a list with 4 components
2 # a string, a numeric vector, a matrix, and a scalar
3 w <- list(name="Alona",
4 mynumbers=c(6,4,3,2,1),
5 mymatrix=matrix(1:8,ncol=2),
6 age=85)
7 w
7:2 (Top Level) R Script
Console Terminal Jobs
~/
> w[[3]] # 3rd component of the list
[,1] [,2]
[1,] 1 5
[2,] 2 6
[3,] 3 7
[4,] 4 8
> w[["mynumbers"]] # component named mynumbers in list
[1] 6 4 3 2 1
>
> w$age
[1] 85
> w$
name
mynumbers
mymatrix
age
Environment History Connections Tutorial
Import Dataset
List
R
Global Environment
m1 num [1:4, 1:2] 6 7 5 4 4 13...
mydata 4 obs. of 4 variables
w List of 4
Values
chr chr [1:4] "chr1" "chr1" "chr2...
end num [1:4] 250 410 200 450
i 1L
result NULL
start num [1:4] 200 4000 100 400
strand chr [1:4] "-" "-" "+" "+"
x num [1:4] 6 7 5 4
v num [1:4] 4 13 6 15
Files Plots Packages Help Viewer
Install
Update
Name Description Versi...
User Library
askpass Safe Password Entry for R, Git, and SSH 1.1
assertthat Easy Pre and Post Assertions 0.2.1
backports Reimplementations of Functions Introduced Since R-3.0.0 1.2.1

Data structures

- R has multiple data structures:
- Vectors
- Matrices
- Data frames
- Lists
- **Factors**

- Factors are data structures that are used to store categorical data. We will elaborate on this type of data structure at our next tutorials.

If/else/ifelse

```
if (condition) {
  # executed when condition is TRUE
} else {
  # executed when condition is FALSE
}
```

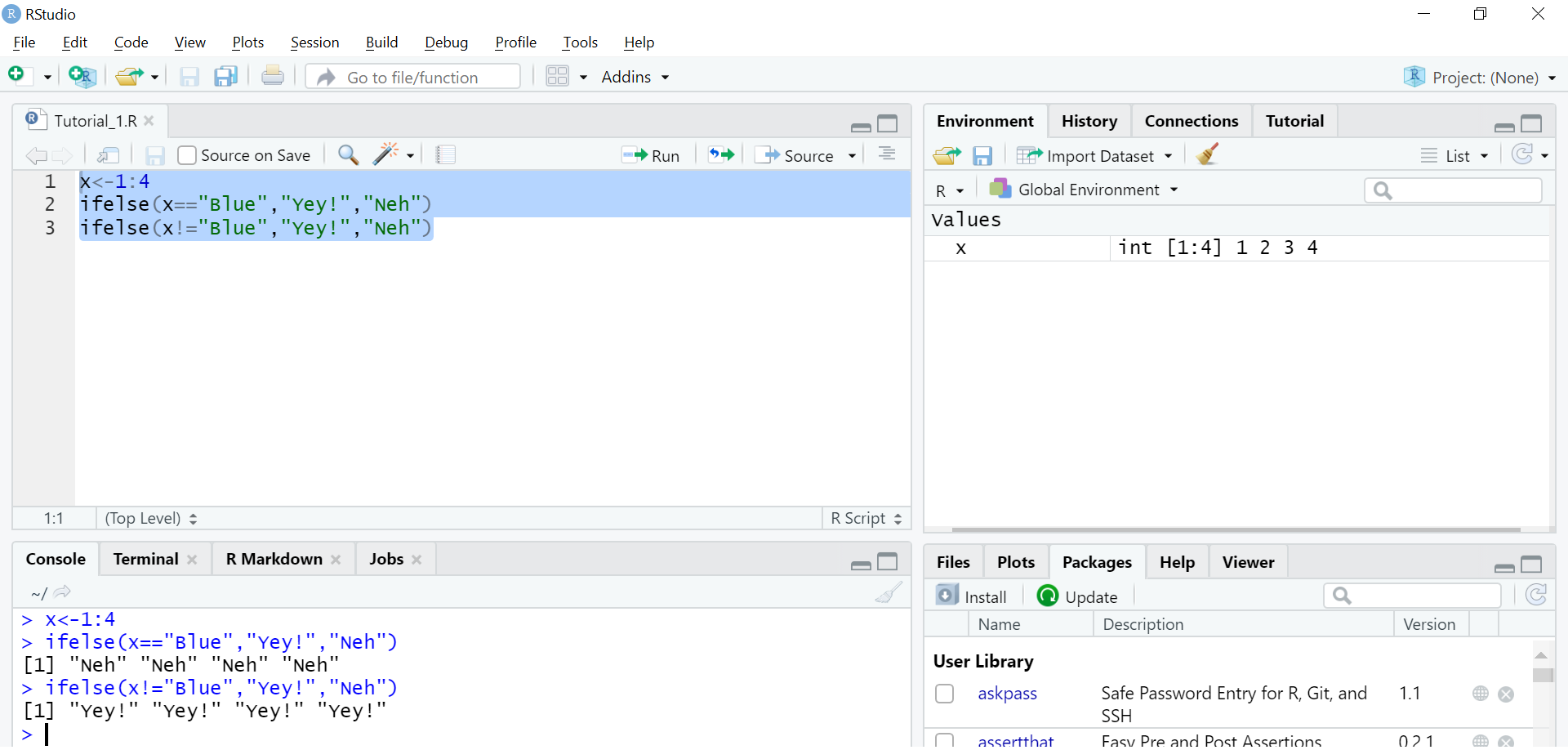
An opening curly brace should never go on its own line and should always be followed by a new line. A closing curly brace should always go on its own line, unless it's followed by else. Always indent the code inside curly braces.

https://r4ds.had.co.nz/functions.html

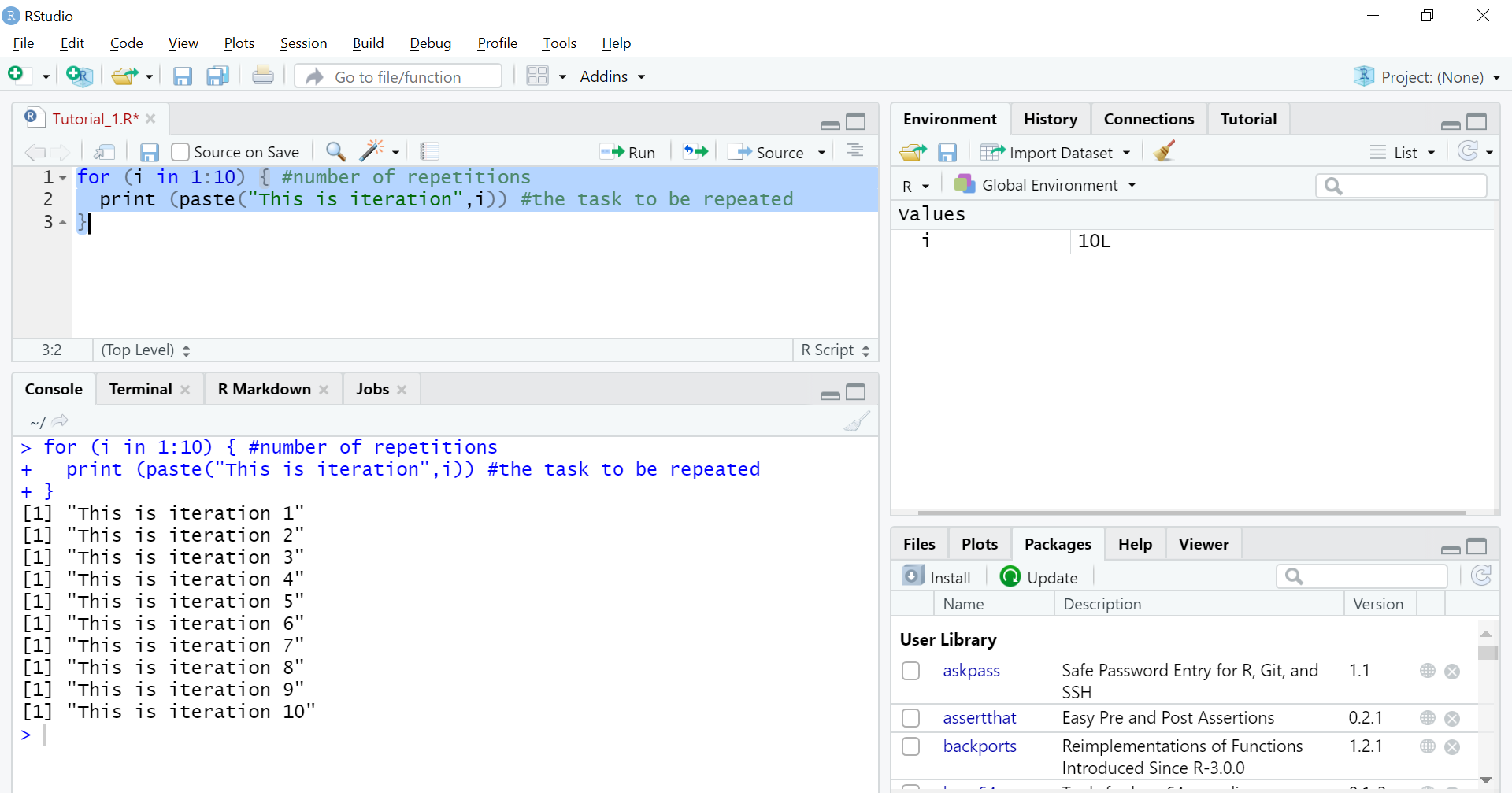
If/else/ifelse

Ifelse(test,yes,no)

ifelse returns a value with the same shape as test which is filled with elements selected from either yes or no depending on whether the element of test is TRUE or FALSE.

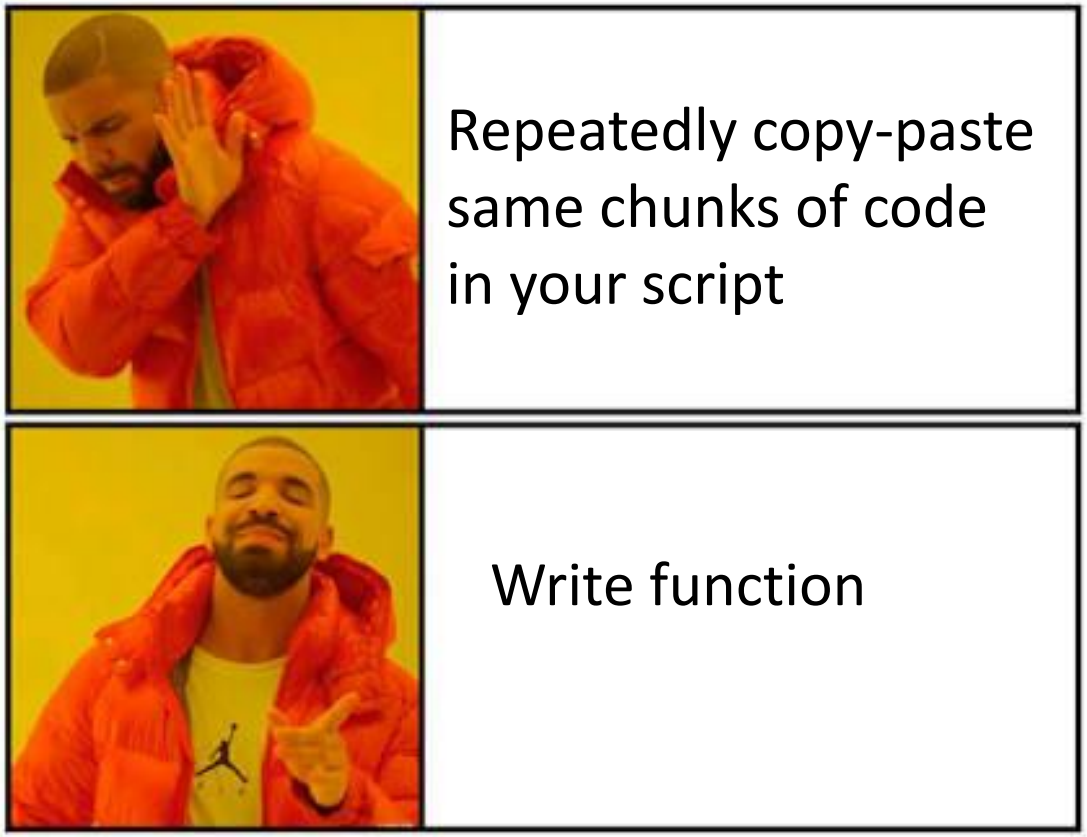


For

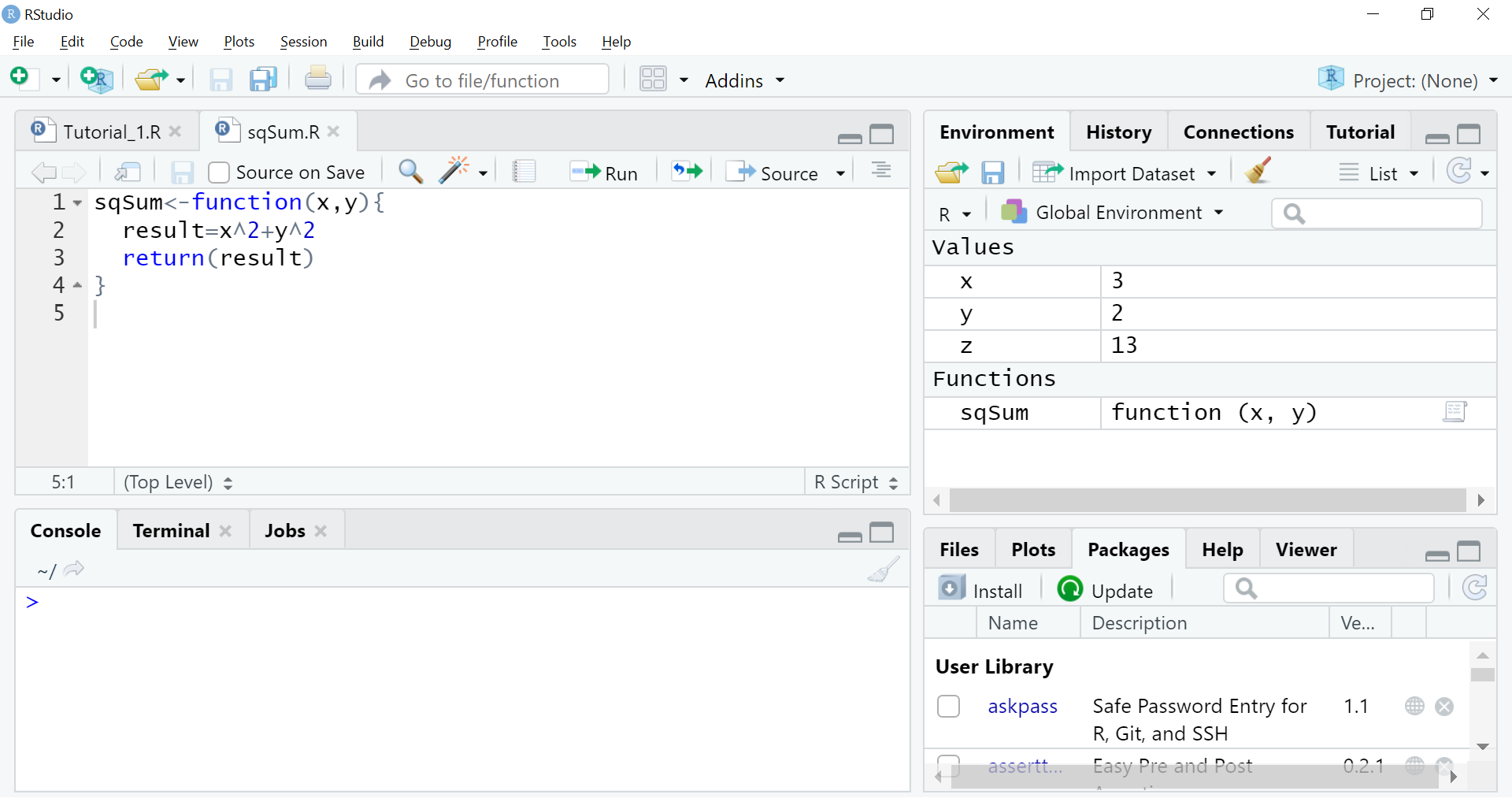

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Tutorial_1.R*
Source on Save
Run
Source
1 for (i in 1:10) { #number of repetitions
2 print (paste("This is iteration",i)) #the task to be repeated
3 }
3:2
(Top Level)
R Script
Environment History Connections Tutorial
Import Dataset
List
R
Global Environment
Values
i 10L
Console Terminal R Markdown Jobs
~/
> for (i in 1:10) { #number of repetitions
+ print (paste("This is iteration",i)) #the task to be repeated
+ }
[1] "This is iteration 1"
[1] "This is iteration 2"
[1] "This is iteration 3"
[1] "This is iteration 4"
[1] "This is iteration 5"
[1] "This is iteration 6"
[1] "This is iteration 7"
[1] "This is iteration 8"
[1] "This is iteration 9"
[1] "This is iteration 10"
>
Files Plots Packages Help Viewer
Install
Update
Name
Description
Version
User Library
askpass
Safe Password Entry for R, Git, and SSH
1.1
assertthat
Easy Pre and Post Assertions
0.2.1
backports
Reimplementations of Functions Introduced Since R-3.0.0
1.2.1

Functions

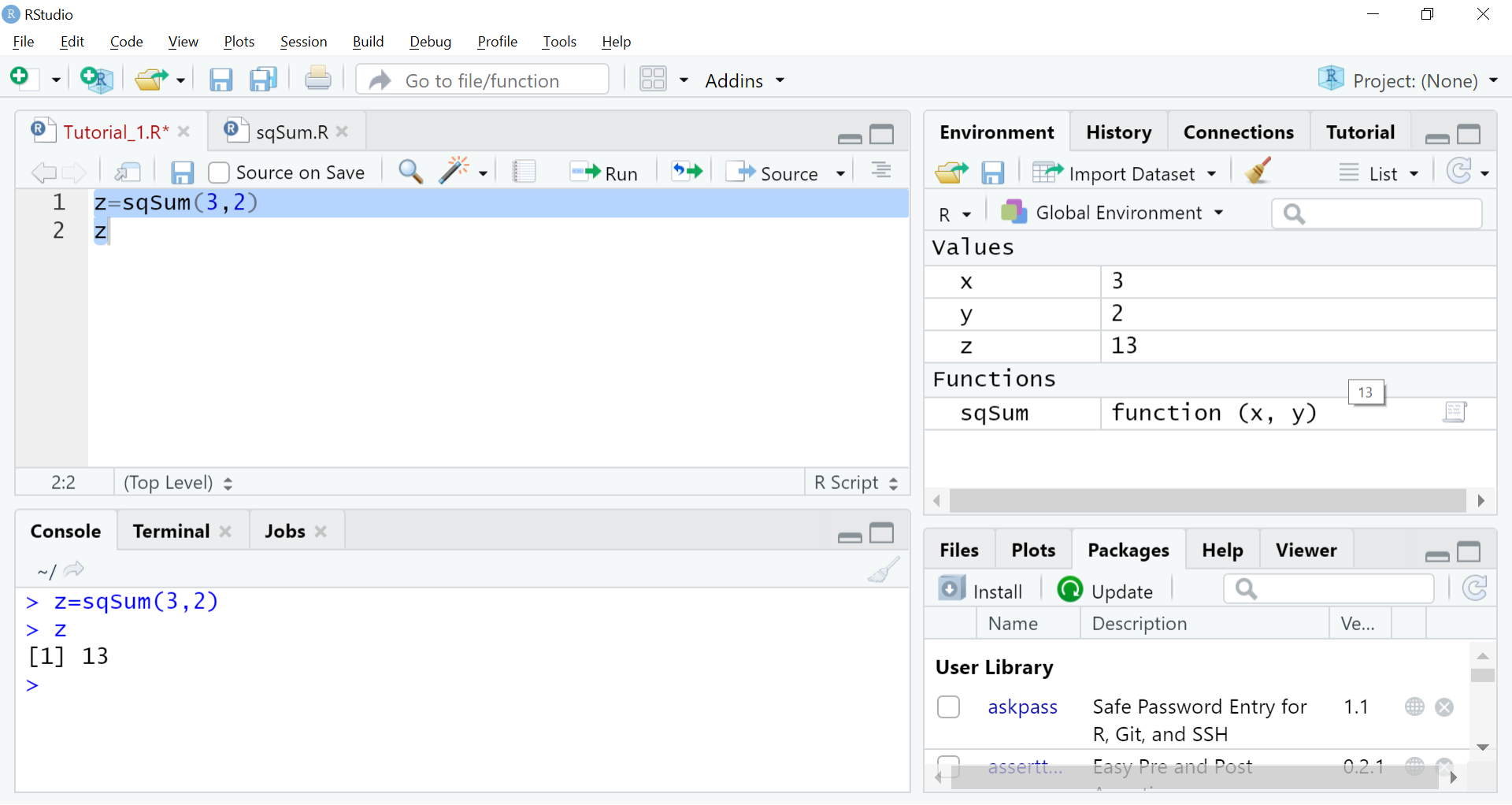
- You should consider writing a function if you used the same block of code more than twice. Function name must be clear. Don’t worry about name length as Rstudio has a perfect autocomplete.



Functions

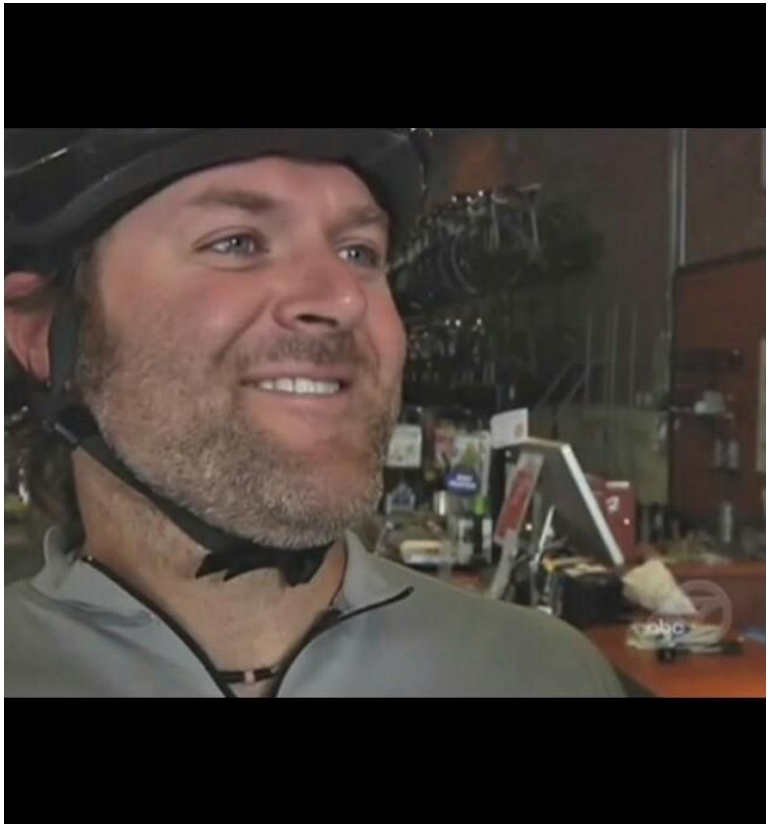

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Tutorial_1.R
sqSum.R
Source on Save
Run
Source
1 sqSum<-function(x,y){
2 result=x^2+y^2
3 return(result)
4 }
5
5:1
(Top Level)
R Script
Console
Terminal
Jobs
~/
>
Environment
History
Connections
Tutorial
Import Dataset
List
R
Global Environment
Values
x 3
y 2
z 13
Functions
sqSum function (x, y)
Files
Plots
Packages
Help
Viewer
Install
Update
Name
Description
Ve...
User Library
askpass
Safe Password Entry for R, Git, and SSH
1.1
assertt...
Easy Pre and Post
0.2.1

Functions


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Tutorial_1.R*
sqSum.R
Source on Save
Run
Source
1 z=sqSum(3,2)
2 z
2:2
(Top Level)
R Script
Console
Terminal
Jobs
~/
> z=sqSum(3,2)
> z
[1] 13
>
Environment
History
Connections
Tutorial
Import Dataset
List
R
Global Environment
Values
x 3
y 2
z 13
Functions
sqSum function (x, y)
13
Files
Plots
Packages
Help
Viewer
Install
Update
Name
Description
Ve...
User Library
askpass
Safe Password Entry for R, Git, and SSH
1.1
assertt...
Easy Pre and Post
0.2.1

Apply functions family

- Try to do your best to replace every “for” by the helper functions from the “apply” family.

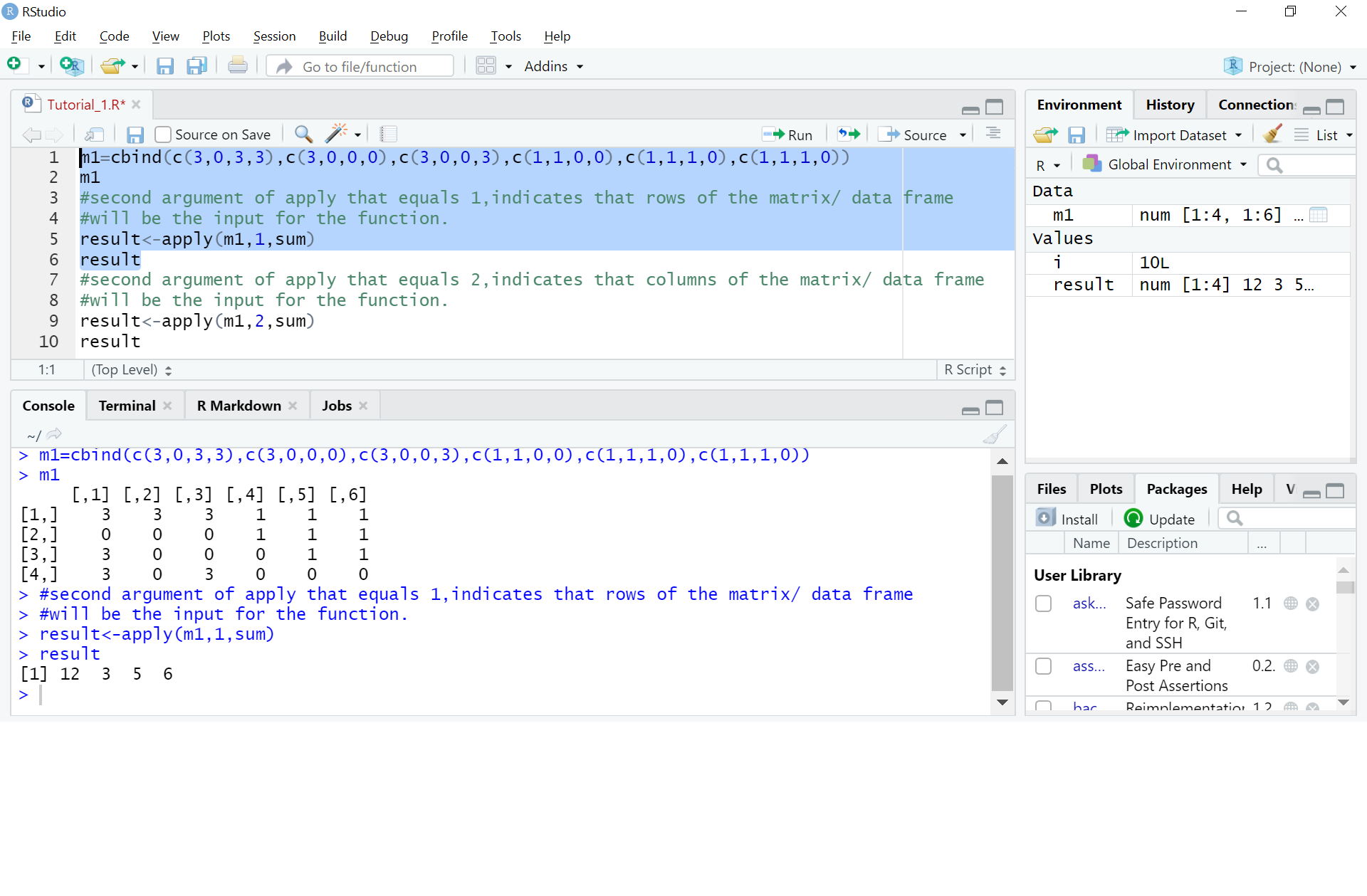


Apply functions family

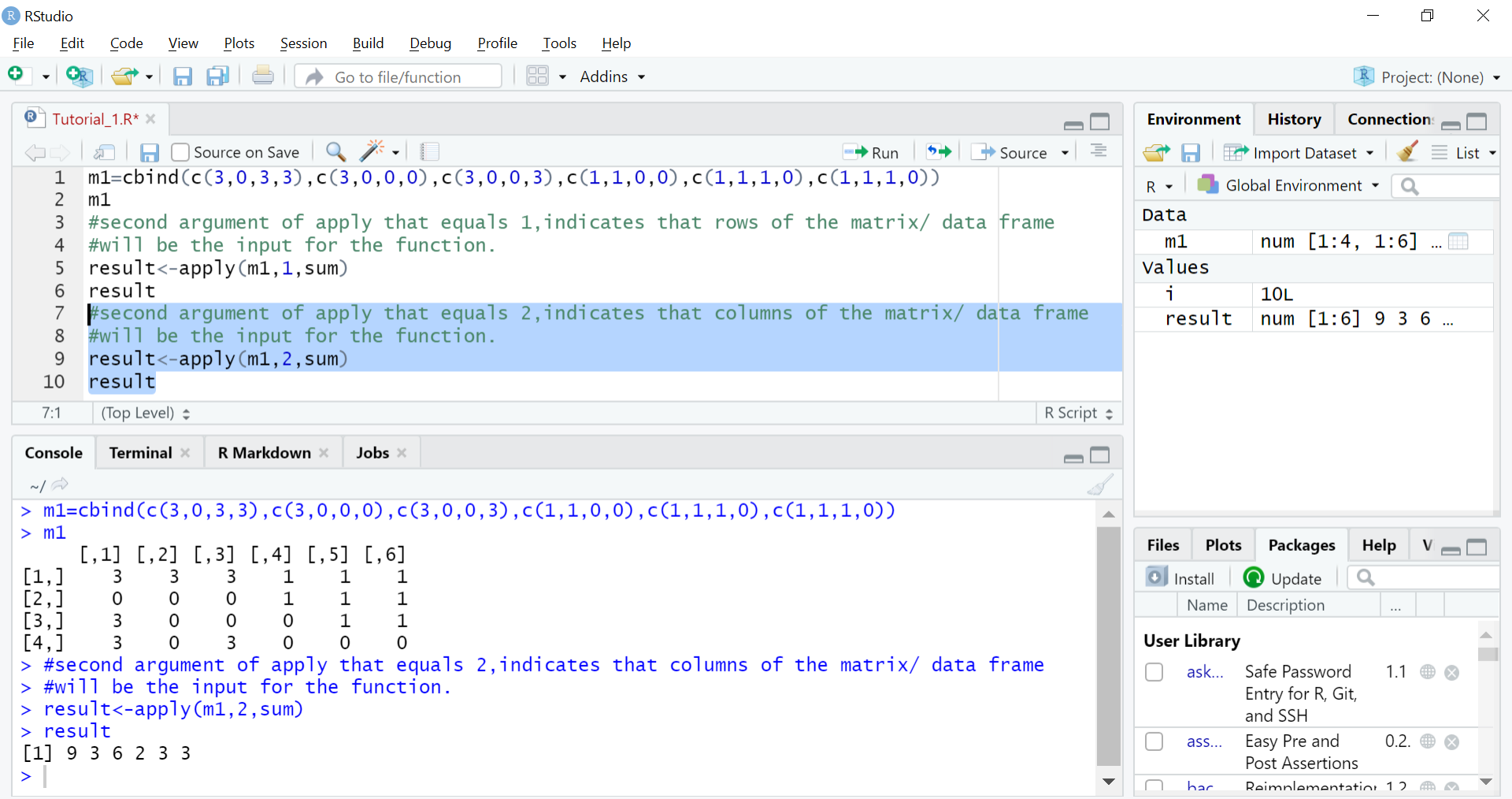
- The apply family of functions in base R (apply(), lapply(), tapply(), etc) consists of functions that eliminate the need for many common “for” loops.
- The difference between these is that they take different types of inputs. For example, apply works on data frames or matrices and applies the function on each row or column of the data structure.

Another option to get rid of “for” loops is to get familiar with the functions in the purrr package
purrr cheatsheet is here:
https://github.com/rstudio/cheatsheets/blob/master/purrr.pdf

Apply functions family

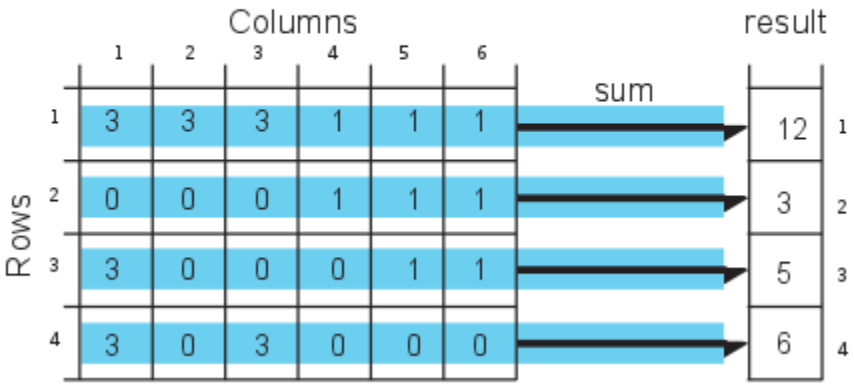


Apply functions family

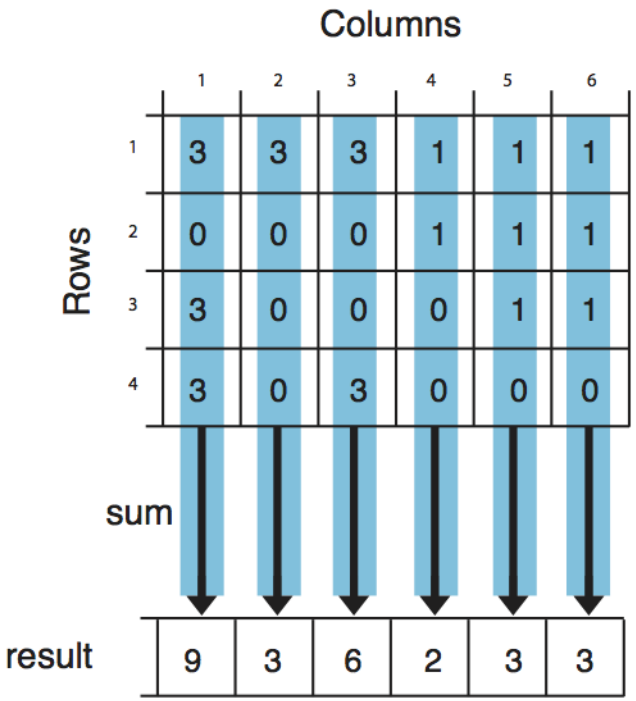


Apply functions family

```
result<-apply(mat,1,function(x) sum(x) )
result<-apply(mat,1,sum)
```



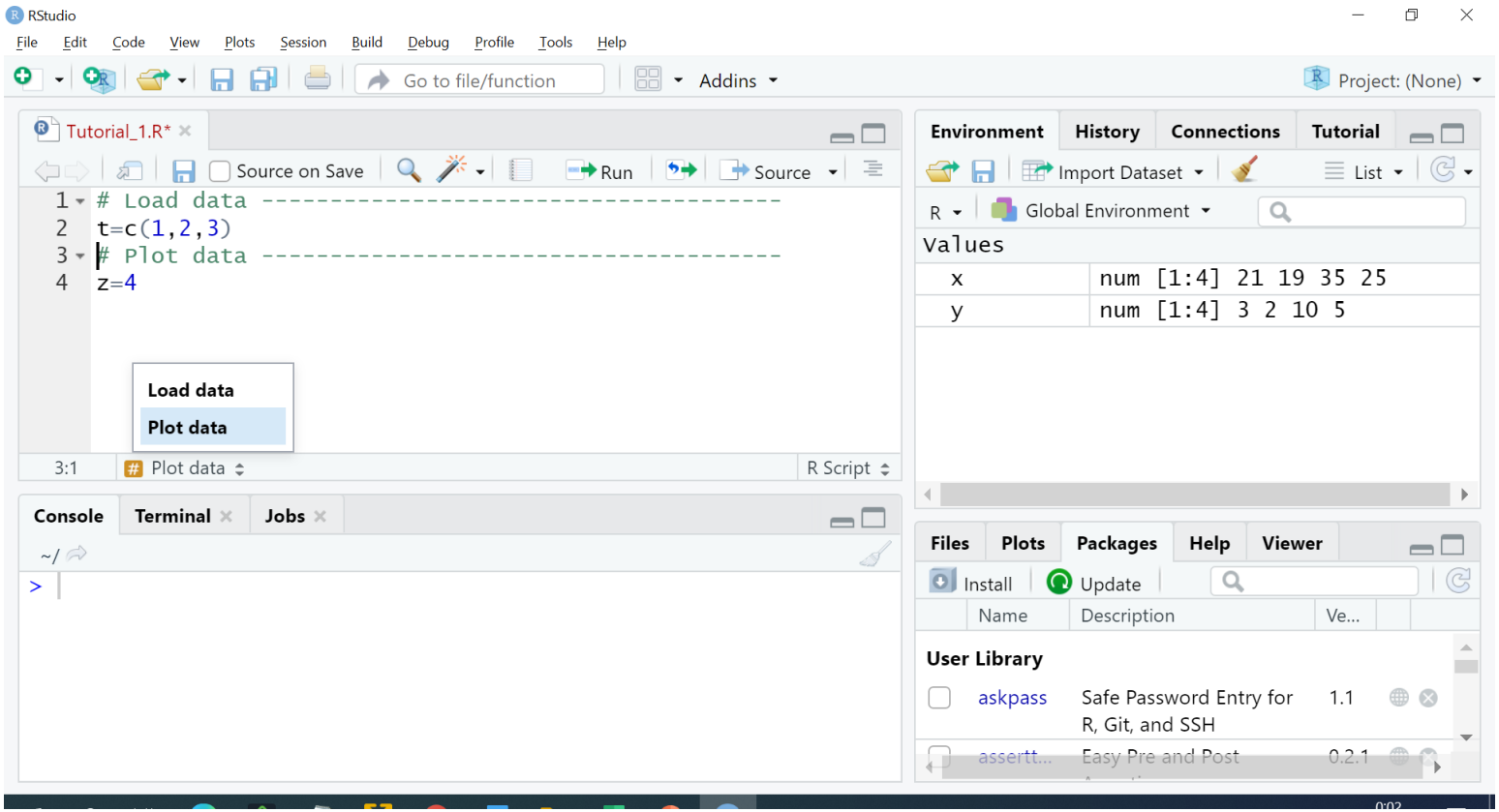

```
result<-apply(mat,2,function(x) sum(x))
result<-apply(mat,2,sum)
```



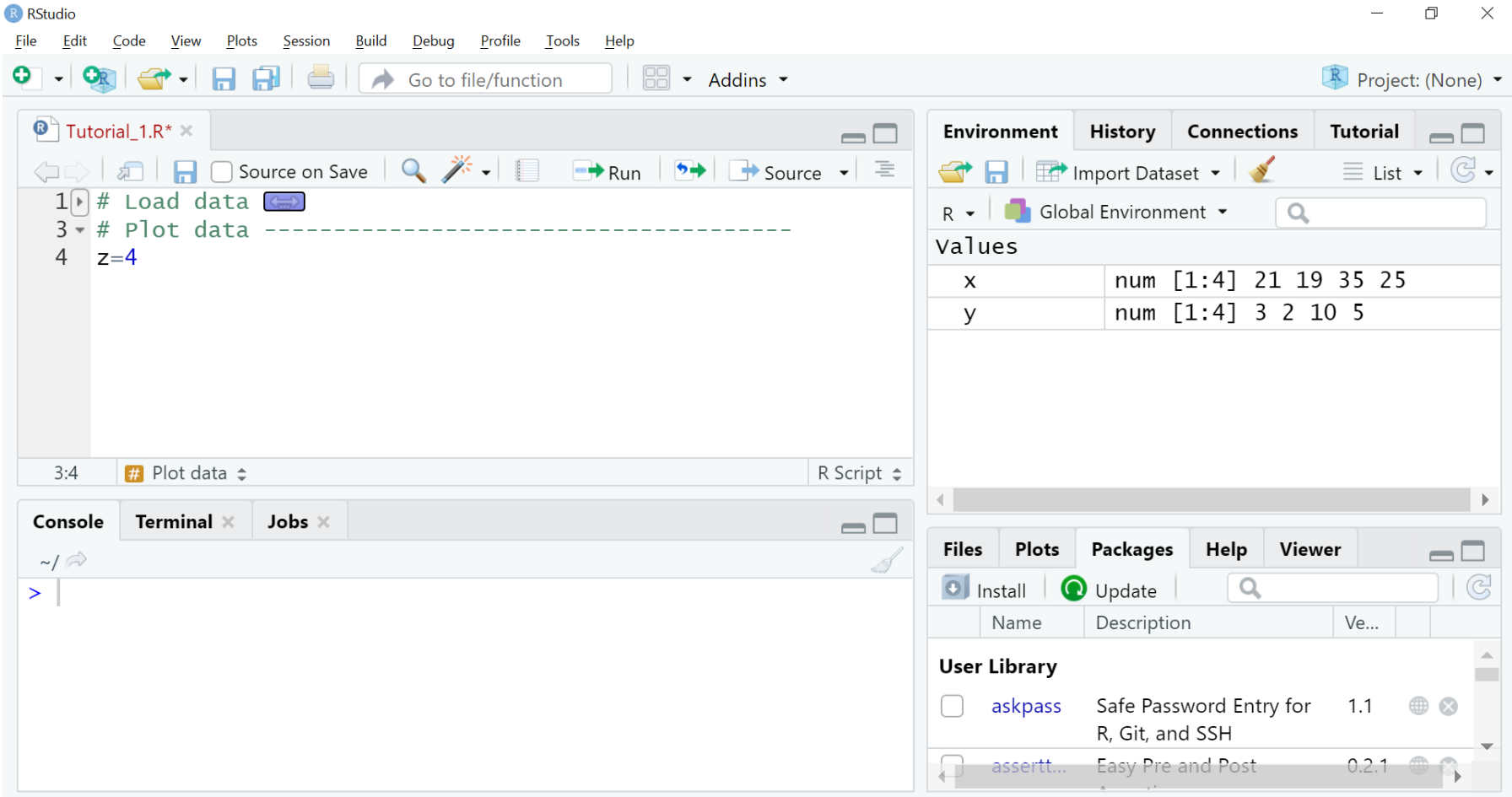
https://compgenomr.github.io/book/functions-and-control-structures-for-ifelse-etc-.html#loops-and-looping-structures-in-r

More tips…

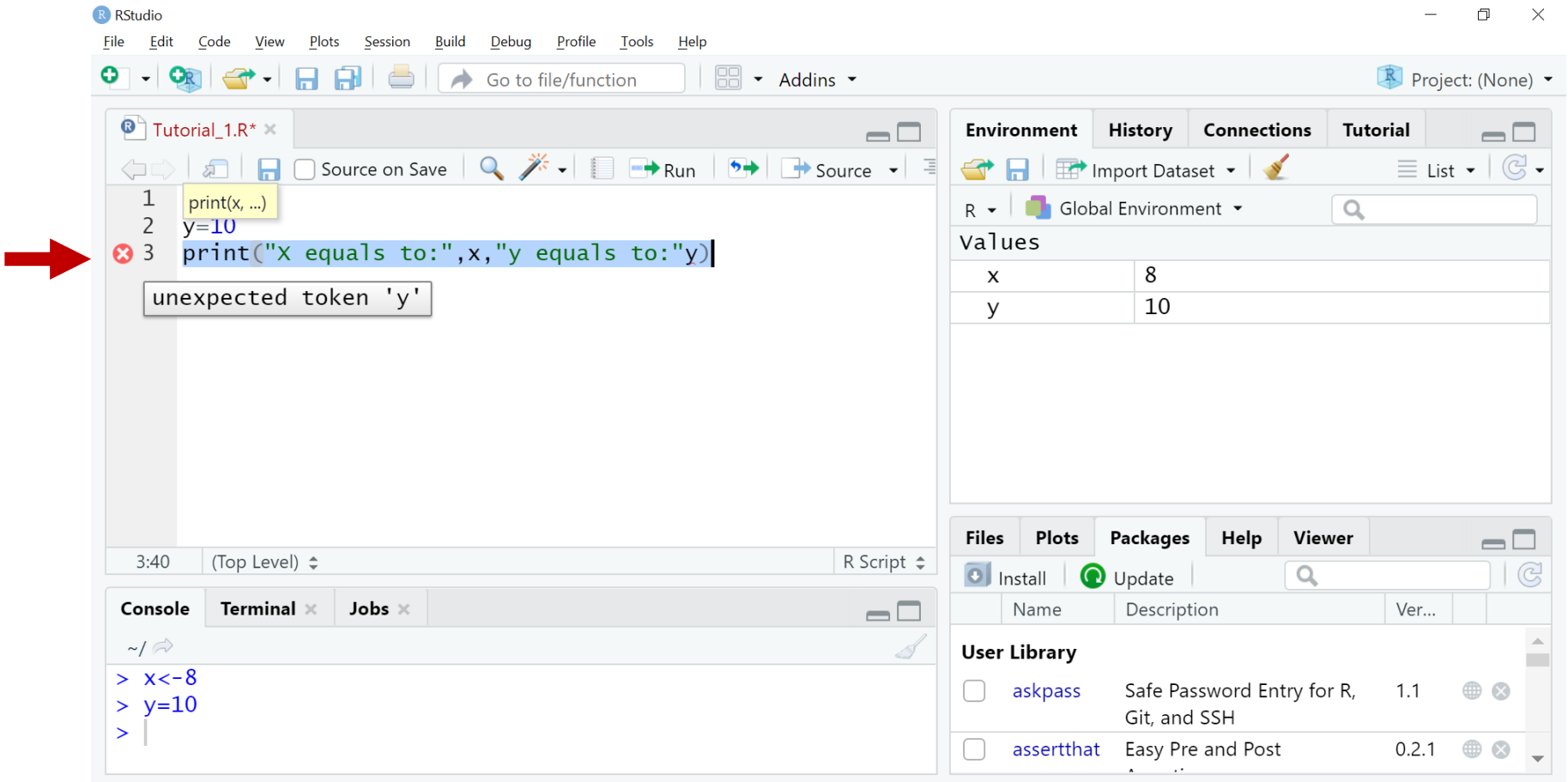
- Break your code into a well explained blocks



More tips…


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Tutorial_1.R*
Source on Save
Run
Source
1 # Load data
3 # Plot data ----------
4 z=4
3:4
Plot data
R Script
Console
Terminal
Jobs
~/
>
Environment
History
Connections
Tutorial
Import Dataset
List
R
Global Environment
Values
x num [1:4] 21 19 35 25
y num [1:4] 3 2 10 5
Files
Plots
Packages
Help
Viewer
Install
Update
Name
Description
Ve...
User Library
askpass
Safe Password Entry for R, Git, and SSH
1.1
assertt...
Easy Pre and Post
0.2.1

Error messages in the editor window



What’s next ?

- At our next tutorial we’ll discover the wonders of the plotting and we’ll make statistics our dedicated partner for the rest of the semester.