
Introduction to R

Download Files

Download this presentation as pdf

Download Files for today

To save either file…. Right-click –> ‘Save link as…’. Save file to location. Go to RStudio or Acrobat. File –> Open File… find your file.

Introductions

Instructors:

- Kyle Horton
- Georgia Titcomb
- Brian Gerber

Why learn to code?

- efficiency
- transparency
- flexibility in application
- shareable
- automated processes/report writing
- marketable skill
- needed for publications

Software

What is R?

R is a “suite of software facilities for data manipulation, calculation and graphical display.”

R uses **packages** that are collections of functions, data, and compiled code in a “well-defined format”.

Packages are downloaded from The Comprehensive R Archive Network (CRAN), R’s central software repository. Also, on GitHub, GitLab, BitBucket or other code sharing platforms.

Why use R?

- open-source and free
- small total user base / large in ecology and statistics
- find help online, e.g., stack**overflow**
- statistics
- plotting / graphics
- data management

What is RStudio?

RStudio is an “Integrated Development Environment (IDE)”.

RStudio brings tools/languages together.

We use R within RStudio.

Why use RStudio?

- Makes using R easier
- Projects (file mgmt)
- R Shiny: Interactive online apps
- R Markdown: Interactive documents
- Quarto: interactive articles, websites, blog, …
- Posit - Certified B corp

Online resources to learn R

- Intro to R for Biologists
- Introduction to R - tidyverse
- R for Data Science (2e)
- Advanced R
- Introduction to the R Language
- Introduction to R
- An Introduction to R for Research
- Introduction to Data Exploration ana Analysis with R
- Working with Data in R

Today

Goal

‘Get familiar with fundamentals of R useful for data’

‘To get beyond the initial shock or fear of programming and start using R’

Today

Learning Objectives

- Write and execute code in R via RStudio
- R language vocabulary
- Read/write data
- Find help
- Manipulate data efficiently
- Plot data/results

Today

Execution

- Presentation / code walk through
- Challenges (independent or in teams of 2-3)

Today

Schedule

- 900 - 930: Introductions and Setup
- 930 - 1015: RStudio and R (objects and functions)
- 1015 - 1130: Data Input and Output
- 1130- 1200: Finding Help
- 1200 - 1300: Lunch
- 1300 - 1400: Data Mgmt
- 1400 - 1500: Plotting
- 1500 - 1600: Final Challenge

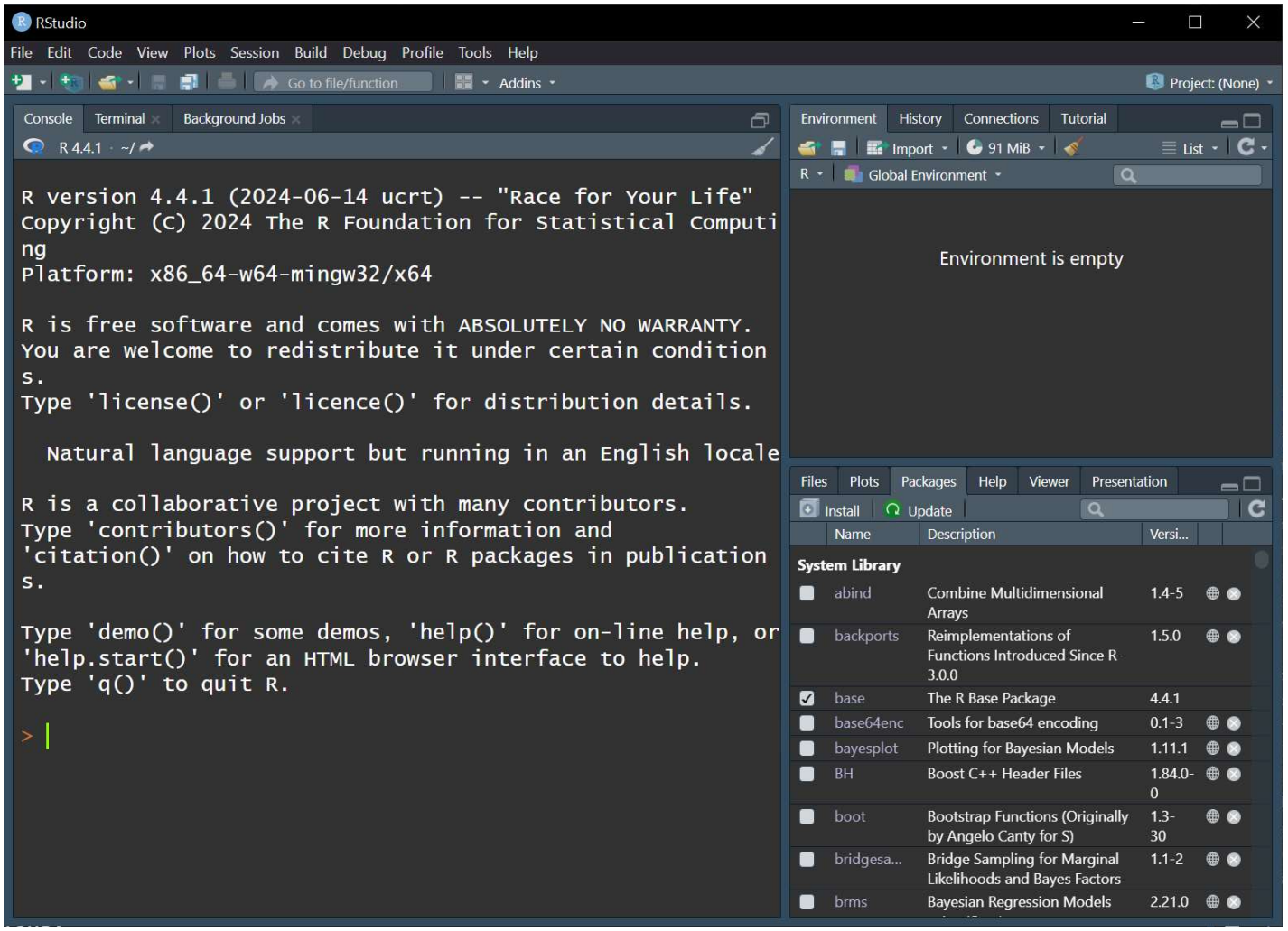
Showcases

Brian - R Shiny application

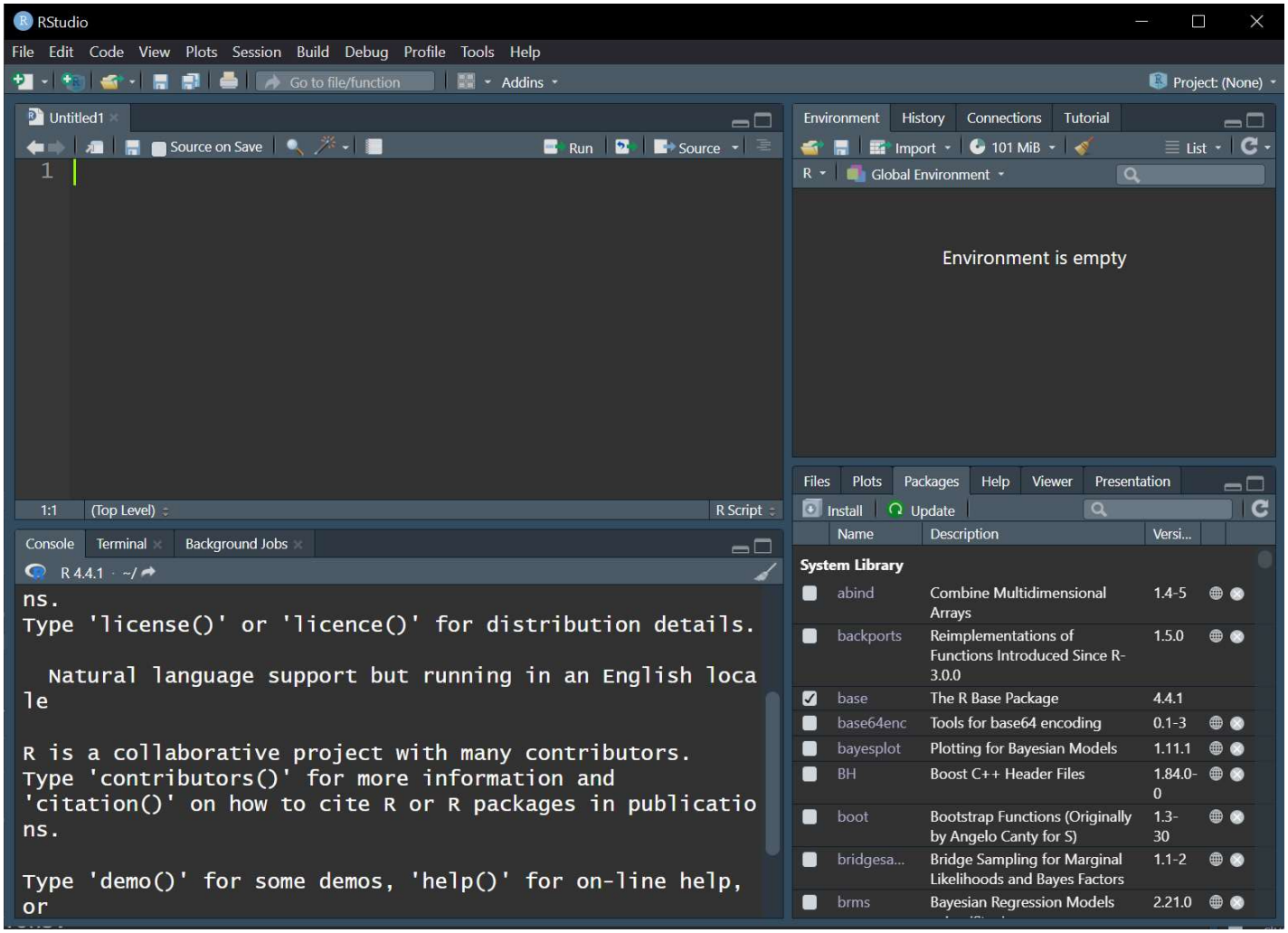
Kyle - Lights out alerts

Georgia - The Orion Nebula?

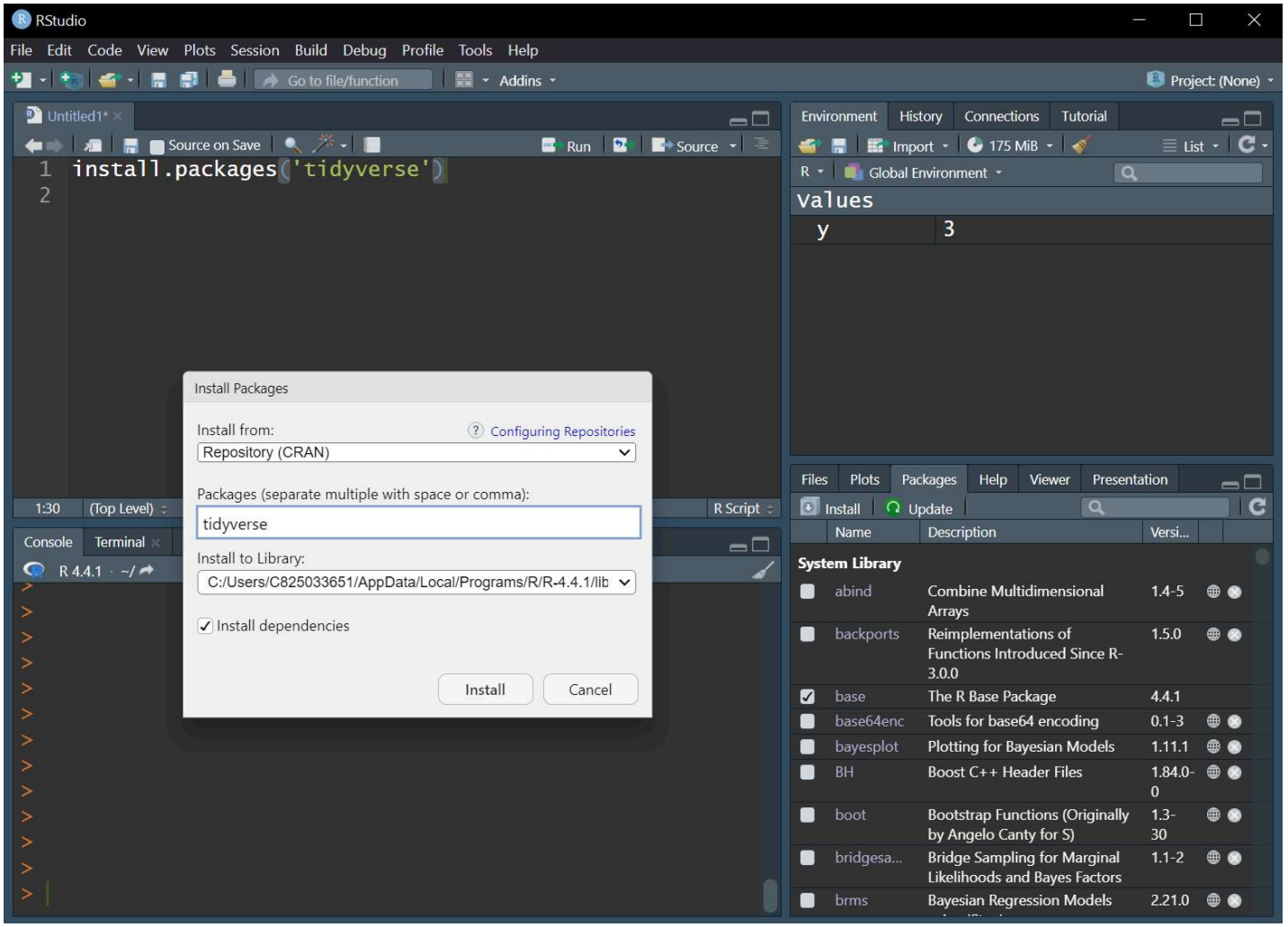
RStudio


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Console
Terminal
Background Jobs
R 4.4.1 · ~/
R version 4.4.1 (2024-06-14 ucrt) -- "Race for Your Life"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64
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.
>
Environment
History
Connections
Tutorial
Import
91 MiB
List
R
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
Presentation
Install
Update
Name
Description
Versi...
System Library
abind
Combine Multidimensional Arrays
1.4-5
backports
Reimplementations of Functions Introduced Since R-3.0.0
1.5.0
base
The R Base Package
4.4.1
base64enc
Tools for base64 encoding
0.1-3
bayesplot
Plotting for Bayesian Models
1.11.1
BH
Boost C++ Header Files
1.84.0-0
boot
Bootstrap Functions (Originally by Angelo Canty for S)
1.3-30
bridgesa...
Bridge Sampling for Marginal Likelihoods and Bayes Factors
1.1-2
brms
Bayesian Regression Models
2.21.0

RStudio


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Untitled1
Source on Save
Run
Source
1
1:1
(Top Level)
R Script
Console
Terminal
Background Jobs
R 4.4.1 · ~/
ns.
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
Environment
History
Connections
Tutorial
Import
101 MiB
List
R
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
Presentation
Install
Update
Name
Description
Versi...
System Library
abind
Combine Multidimensional Arrays
1.4-5
backports
Reimplementations of Functions Introduced Since R-3.0.0
1.5.0
base
The R Base Package
4.4.1
base64enc
Tools for base64 encoding
0.1-3
bayesplot
Plotting for Bayesian Models
1.11.1
BH
Boost C++ Header Files
1.84.0-0
boot
Bootstrap Functions (Originally by Angelo Canty for S)
1.3-30
bridgesa...
Bridge Sampling for Marginal Likelihoods and Bayes Factors
1.1-2
brms
Bayesian Regression Models
2.21.0

Installing Packages



Packages for Workshop

Please install from CRAN

- tidyverse
- readxl
- ggridges
- gridExtra

```
install.packages(c("tidyverse",
                   "readxl",
                   "ggridges",
                   "gridExtra")
                 )
```

The language of R

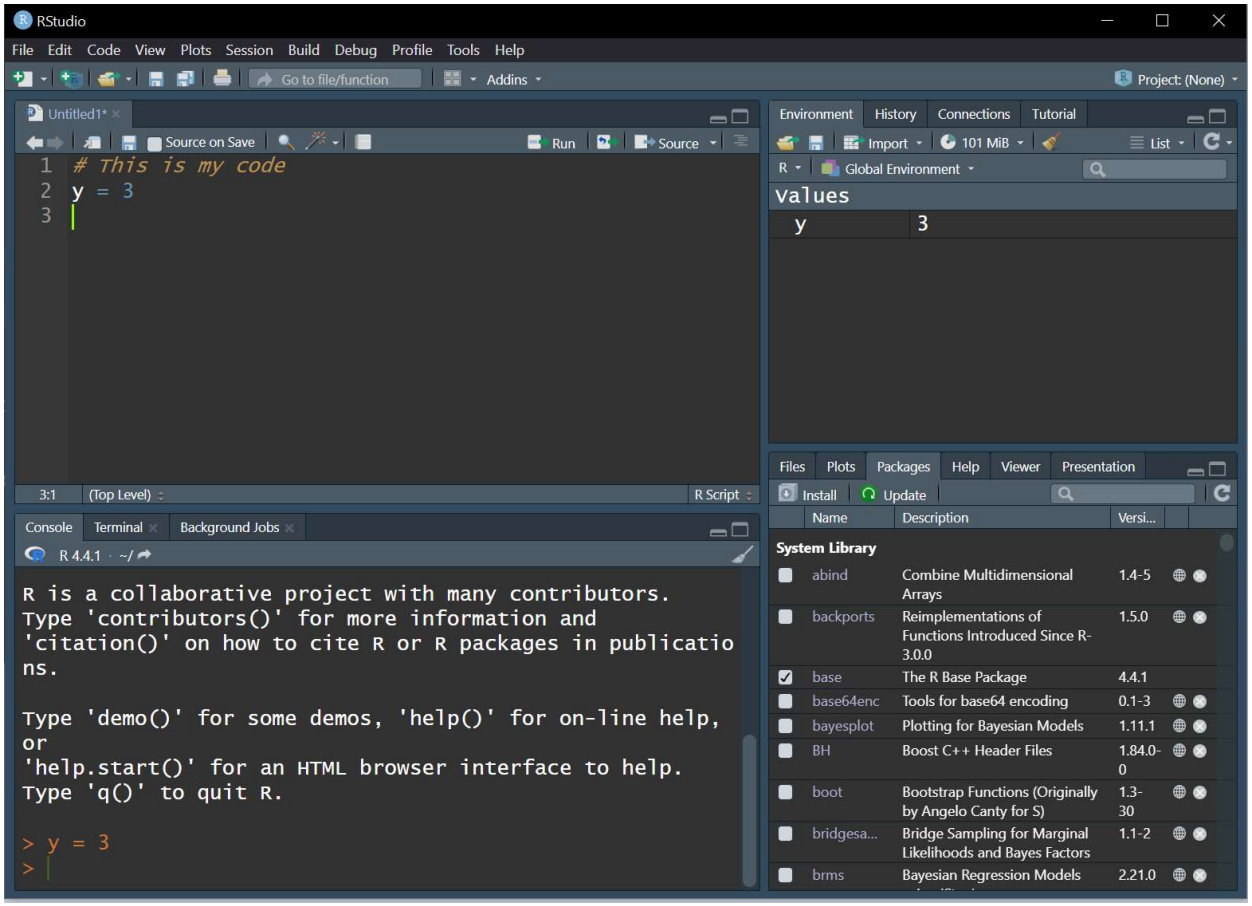
Objects

A storage place for information; stored in the “Environment”

‘Attributes’ describes the structure or information of the object

The language of R

Objects



The language of R

Objects

```
1 # y is an 'object' that is assigned the value 3
2 y = 3
3 y
```

```
[1] 3
```

```
1 # Same operation '=' '<-'
2 y <- 3
```

The language of R

Objects

```
1 # We can create new objects from objects
2 y2 = y-2
3 y2
```

```
[1] 1
```

```
1 # We can do math with our objects
2 # Mind your parentheses (order of operation)
3 y*2 / y*4
```

```
[1] 8
```

```
1 y*2 / (y*4)
```

```
[1] 0.5
```

The language of R

Functions

‘does stuff’; creates or manipulates objects

‘Arguments’ are the types of things a function is asking for; the inputs

The language of R

object = function(attribute1 = input1, attribute2 = input2)

object = function(input1, input2)

this = sign(x = -5)

```
1 sign(-5)
```

```
[1] -1
```

```
1 sign(5)
```

```
[1] 1
```

The language of R

Functions

```
1 # function - 'c' - concatenate
2 y = c(1,2,3,4,5,6)
```

```
1 is.numeric(y)
```

```
[1] TRUE
```

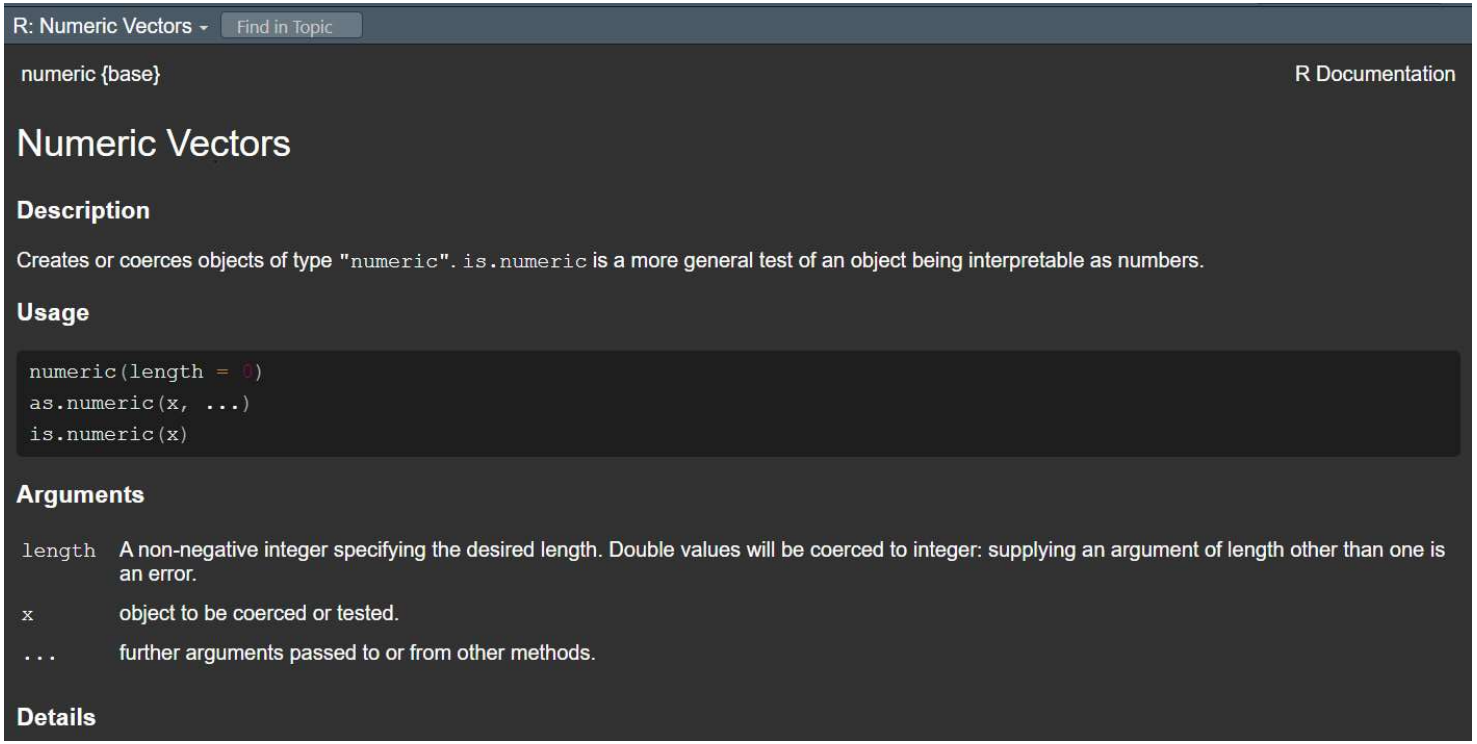
```
1 # The function 'class' has the argument 'x'
2 is.numeric(x = y)
```

```
[1] TRUE
```

The language of R

Functions

```
1 # How to find out the arguments of a function?
2 ?is.numeric
```



R: Numeric Vectors — Find in Topic

numeric {base} — R Documentation

Numeric Vectors

Description

Creates or coerces objects of type "numeric". is.numeric is a more general test of an object being interpretable as numbers.

Usage

```
numeric(length = 0)
as.numeric(x, ...)
is.numeric(x)
```

Arguments

length	A non-negative integer specifying the desired length. Double values will be coerced to integer: supplying an argument of length other than one is an error.
x	object to be coerced or tested.
...	further arguments passed to or from other methods.

Details

The language of R

Wrapping functions

```r
# Functions are commonly 1) wrapped, 2) have mul
x = matrix(
            data = c(1,2,3,4,5,6),
            nrow = 2,
            ncol = 3
            )
```

```r
x
```

```
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

The language of R

Values

- numeric
- integer
- character
- factor

Objects

- vector
- matrix
- array
- list
- dataframe
- S3, S4, S5, and beyond

Types of Values

Numeric

```
1 y = 3
2 class(y)
```

```
[1] "numeric"
```

Integer

```
1 y = integer(3)
2 class(y)
```

```
[1] "integer"
```

Character

```
1 y = "habitat"
2 class(y)
```

```
[1] "character"
```

Factor

```
y = factor("habitat")
class(y)
```

```
[1] "factor"
```

Types of Objects

Vector

```r
# An ordered collection indexed 1,2,...n
# Using the function 'c' to concetanate
z1 = c(4,5,6)
z1
```

```
[1] 4 5 6
```

The value 4 is in element/index/position 1 of the vector

The value 6 is in element/index/position 3 of the vector

```r
# the dimension of a vector
length(z1)
```

```
[1] 3
```

```r
# A vector of characters
z2 = c("dog","cat","horse")
z2
```

```
[1] "dog"   "cat"   "horse"
```

```
1  z3 = c("dog","1","horse")
2  z3
```

```
[1] "dog"   "1"     "horse"
```

:::

Types of Objects

Subsetting a vector

```
z3 = c("dog",
       "1",
       "horse",
       "chicken"
       )
z3[2]
```

```
[1] "1"
```

```
2:4
```

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

```
z3[2:4]
```

```
[1] "1"       "horse"   "chicken"
```

```
z3[c(2,4)]
```

```
[1] "1"       "chicken"
```

```
1 z3[-1]
```

```
[1] "1"       "horse"   "chicken"
```

Types of Objects

Vector of factors

```
z4 = factor(
               c("dog",
                 "dog",
                 "cat",
                 "horse"
                 )
             )
```

```
z4
```

```
[1] dog   dog   cat   horse
Levels: cat dog horse
```

```
levels(z4)
```

```
[1] "cat"   "dog"   "horse"
```

```r
summary(z4)
```

```
   cat    dog horse 
     1      2      1 
```

Types of Objects

Matrix

```r
x = matrix(
              c(1,2,3,4,5,6),
              nrow = 2,
              ncol = 3
             )
```

```r
x
```

```
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```r
#rows and columns
dim(x)
```

```
[1] 2 3
```

Types of Objects

Subsetting a matrix

```
1  # get element of row 1 and column 2
2  x[1,2]
```

```
[1] 3
```

```
1  # get all elements of row 2
2  x[2,]
```

```
[1] 2 4 6
```

```
1  # same as
2  x[2,1:3]
```

```
[1] 2 4 6
```

Types of Objects

Array

```r
# ARRAY - more than two dimensions
z5 = array(
            c("a","b","c","d","1","2","3","4"),
            dim = c(2,2,2)
           )
```

```r
z5
```

```
, , 1

     [,1] [,2]
[1,] "a"  "c"
[2,] "b"  "d"

, , 2

     [,1] [,2]
[1,] "1"  "3"
[2,] "2"  "4"
```

Types of Objects

List

```
1 # LIST - a bucket - will take anything
2 my.list = list(z1, z2, z3, z4, z5)
```

```
1 #Subset a list
2 my.list[[1]]
```

```
[1] 4 5 6
```

```
1 my.list[[4]]
```

```
[1] dog   dog   cat   horse
Levels: cat dog horse
```

Types of Objects

Data frame

E.g., a row for each observation and a column for each variable (can be different types).

```r
x = data.frame(outcome = c(1,0,1,1),
               exposure = c("yes", "yes", "no",
               age = c(24, 55, 39, 18)
               )
x
```

```
  outcome exposure age
1       1      yes  24
2       0      yes  55
3       1       no  39
4       1       no  18
```

Types of Objects

Subset data.frame

```
1 x$exposure
```

```
[1] "yes" "yes" "no"  "no"
```

```
1 x['exposure']
```

```
  exposure
1      yes
2      yes
3       no
4       no
```

```
1 x[,2]
```

```
[1] "yes" "yes" "no"  "no"
```

Next:
Data input and output (Kyle)