
Introduction to R

Introductions

Instructors:

- Kyle Horton
- Georgia Titcomb
- Brian Gerber

Why learn to code?

- efficiency
- transparency
- flexibility in application
- shareable
- 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** and unmarked group
- data management
- statistics
- plotting / graphics

What is RStudio?

RStudio is an “Integrated Development Environment (IDE)”.
It bring tools/languages together. We use R within RStudio.

Why use RStudio?

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

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
- Find help
- Read/write data
- Manipulate data efficiently
- Plot data or results

Execution

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

Schedule

- 900 - 930: Introductions and setup
- 930 - 1000: RStudio and R (objects and functions)
- 1000 - 1130: Data input and output
- 1130- 1200: Finding help
- 1200 - 1300: Lunch
- 1300 - 1400: Data mgmt
- 1400 - 1500: Plotting
- 1500 - 1600: Challenge

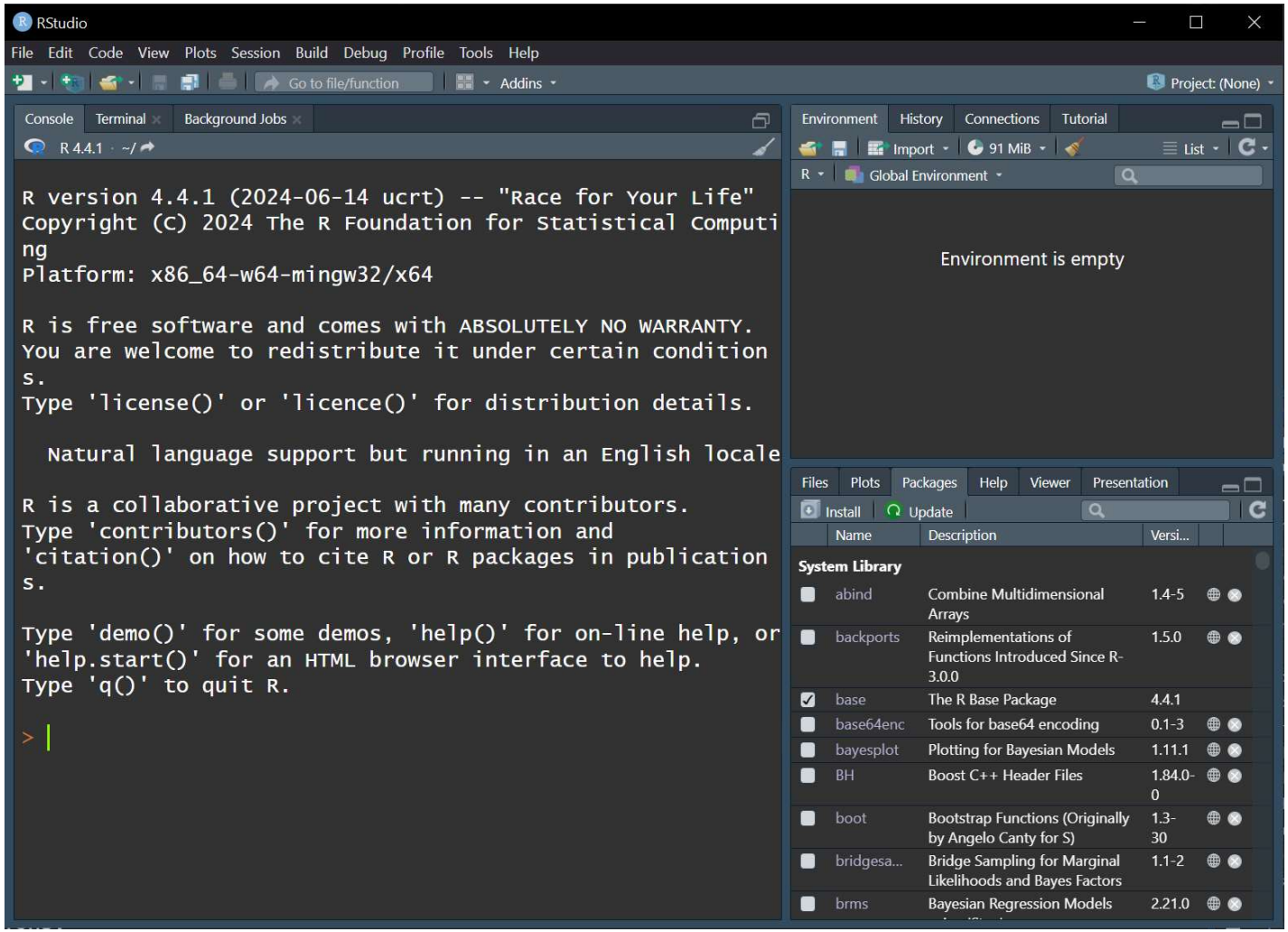
Instructor Showcases

Brian - R Shiny application that allows users to subset data and visualize 14,586 results

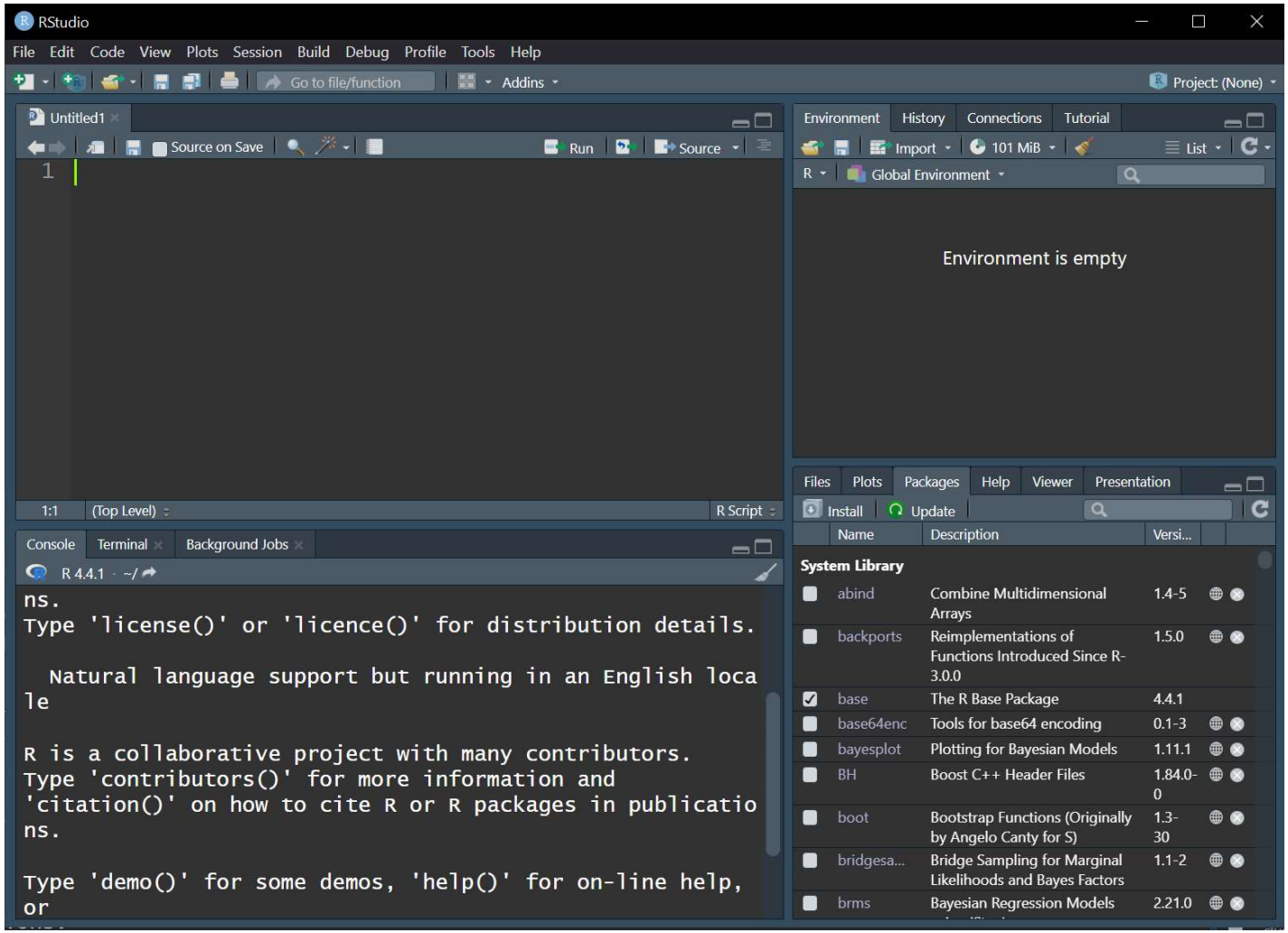
Kyle - example here

Georgia - example here

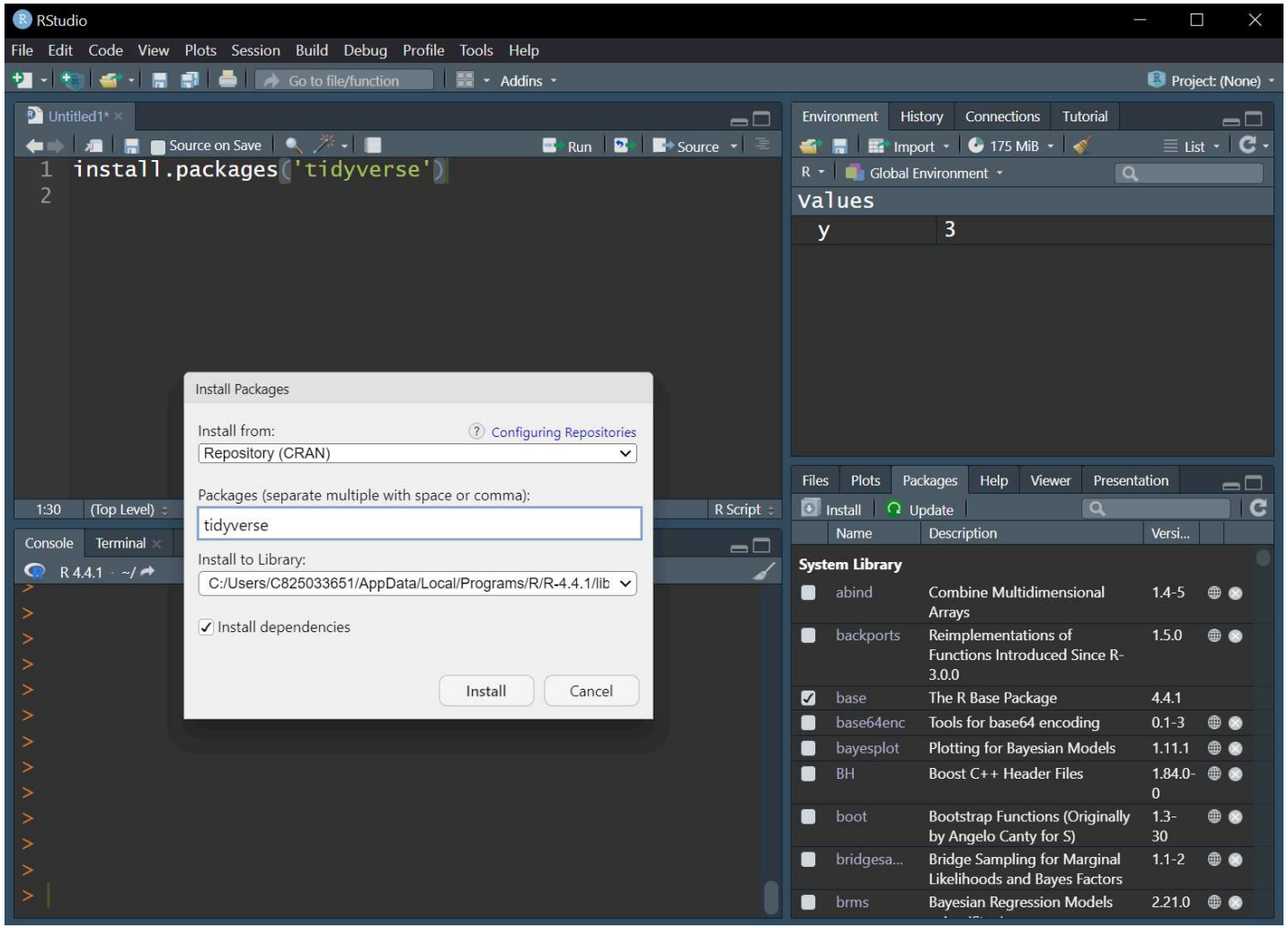
RStudio


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



The language of R

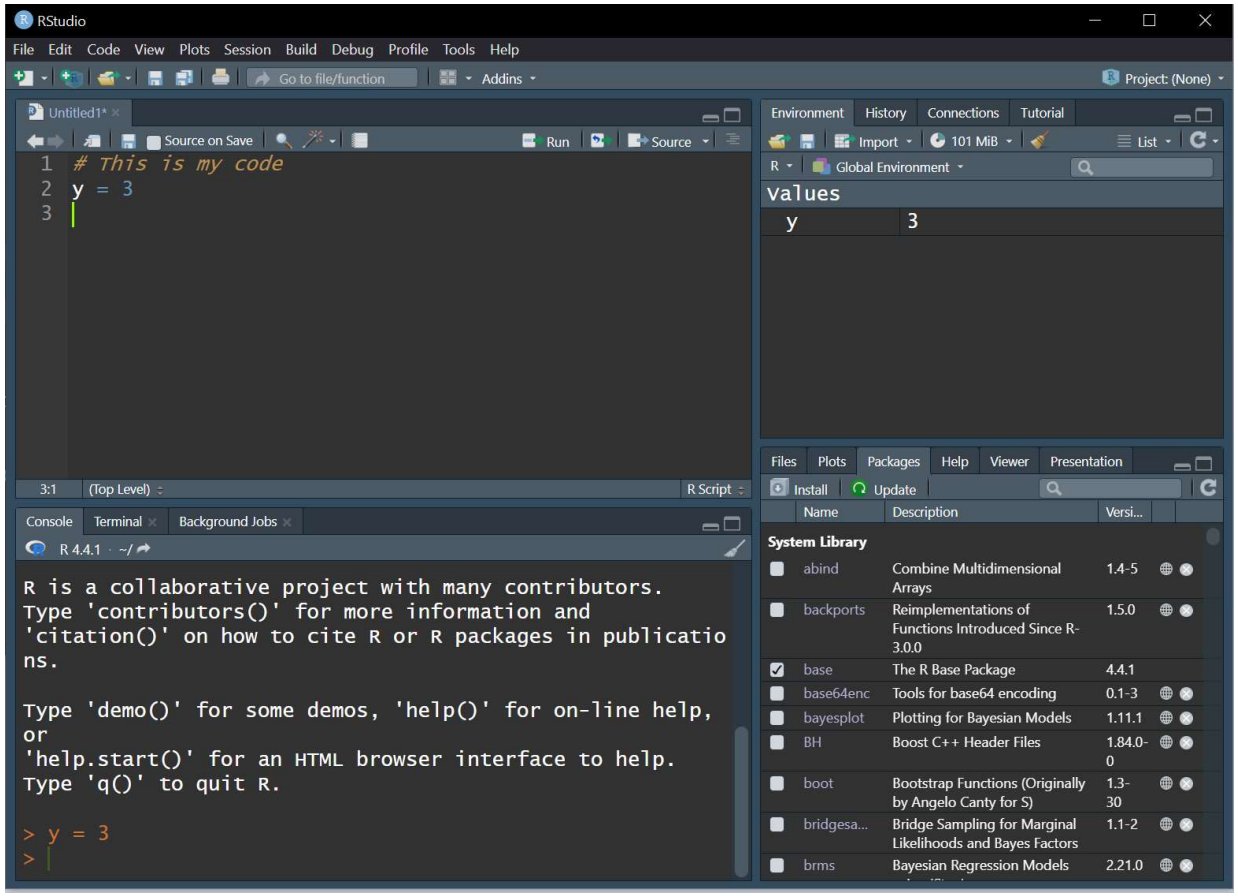
Objects

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

‘Attributes’ describes the structure or information of the object

The language of R

Objects



Code for Presentation

R Code Script for the remaining code is here

Left-Click ‘here’ –> Ctrl A –> Ctrl C. Go to RStudio, left-click on an empty script. Ctrl V

OR

Right-click ‘here’ –> ‘Save link as…’. Save file to location. Go to RStudio. File –> Open File… find your file.

The language of R

Objects

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

```
[1] 3
```

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

The language of R

Objects

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

```
[1] 1
```

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

```
[1] 8
```

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

```
[1] 0.5
```

Challenge 1

Compute the diameter (d) of the Earth (in km) at the equator using this formula for the circumfrance (c)...

$$d = \frac{c}{\pi}$$

- c = 24,901.55 miles
- 1 km = 0.621 miles
- **Hint** type in 'pi' to see what you get

1. Convert the circumference from miles to km.
2. Write the formula in R and plug in the value for circumference

▶ Click for Answer

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(54)
```

```
[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
```

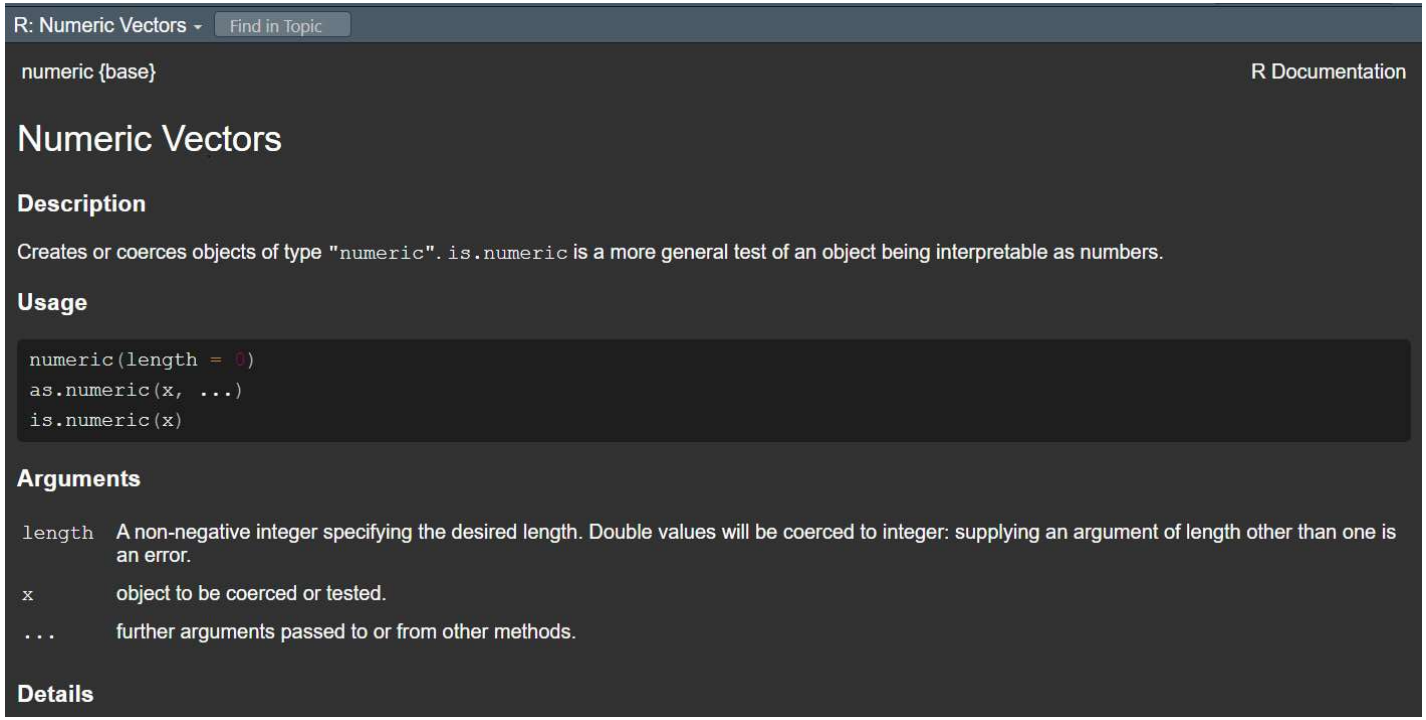
```
1 # The function 'class' has the argument '
2 is.numeric(x = y)
```

```
[1] TRUE
```

The language of R

Functions

```
1 # How to find out the arguments of a func
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 can be wrapped around each ot
# Functions commonly have multiple argume

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

Types of Values

Numeric

```
1 y = 3.3
2 class(y)
```

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

Integer

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

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

Character

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

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

Factor

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

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

Types of Objects

Vector

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

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

```
1 # 4 is in element/index/position 1 of the
2 # 6 is in element/index/position 3 of the
```

```
1 # the dimension of a vector
2 length(z1)
```

```
[1] 3
```

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

```
3 z2
```

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

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

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

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

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

Types of Objects

Subsetting a vector

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

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

```
1  z3[2]
```

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

```
1  z3[2:3]
```

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

```
1  z3[c(2,3)]
```

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

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

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

Types of Objects

Vector of factors, a special kind of character string

```
1  z4 = factor(
2                c("dog", "dog", "cat","horse"
3              )
```

```
1  z4
```

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

```
1  levels(z4)
```

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

```
1  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
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 element of row 2 and column 6
2 x[2,3]
```

```
[1] 6
```

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

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

::: fragment

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

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

Types of Objects

Array

```
1  # ARRAY - more than just two dimensions
2  z5 = array(
3                c("a","b","c","d","e","f"),
4                dim=c(2,2,2)
5              )
```

```
1  dim(z5)
```

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

```
1  z5
```

```
, , 1

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

```
, , 2

     [,1] [,2]
[1,] "e"  "a"
[2,] "f"  "b"
```

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

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",
               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"
```

Challenge 2

1. Create a vector of numbers that has length 6; call this object ‘vec1’.
2. Use the function ‘mean’ to find the mean of the values of vec1.
3. Subset vec1 to only elements 4 through 6. Call this new object ‘vec1a’.
4. Create a new vector (length 3) of characters called “hab1”, “hab2”, and “hab3”. Call this object ‘vec2’.
5. Put vec1a and vec2 together into a data frame and call this object ‘dat’