Introduction to R

Introductions

Instructors:

- Kyle Horton
- Georgia Titcomb
- Brian Gerber

Download presentation as pdf

FISH, WILDLIFE, AND CONSERVATION BIOLOGY

COLORADO STATE UNIVERSITY

USGS

science for a changing world

Colorado Cooperative
Fish and Wildlife Research Unit

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**
- data management
- statistics
- plotting / graphics

What is RStudio?

RStudio is an “Integrated Development Environment (IDE)”. It 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
- Find help
- Read/write data
- Manipulate data efficiently
- Plot data or results

Today

Execution

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

Today

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: Final Challenge

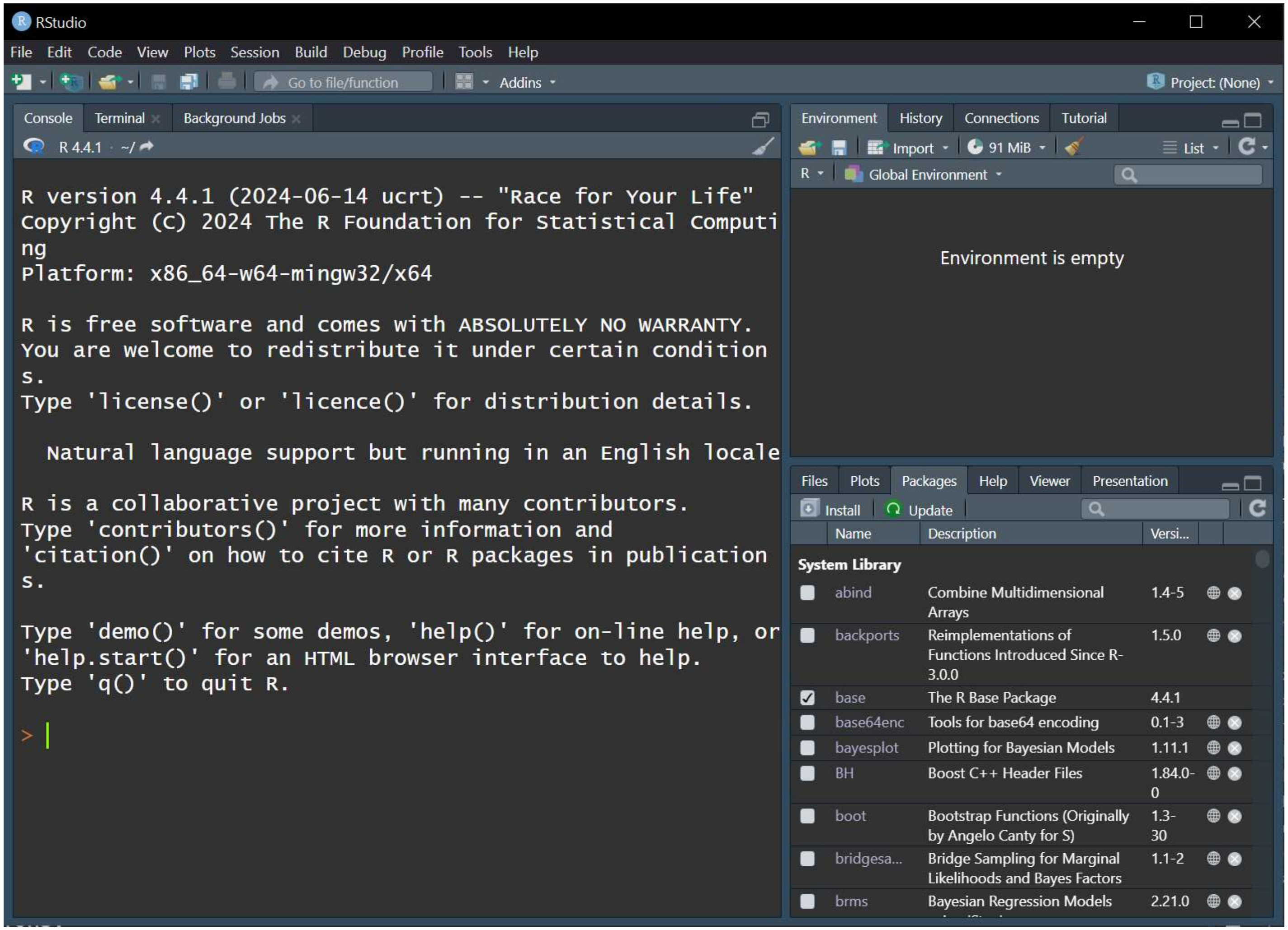
Showcases

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

Kyle - example here

Georgia - example here

RStudio

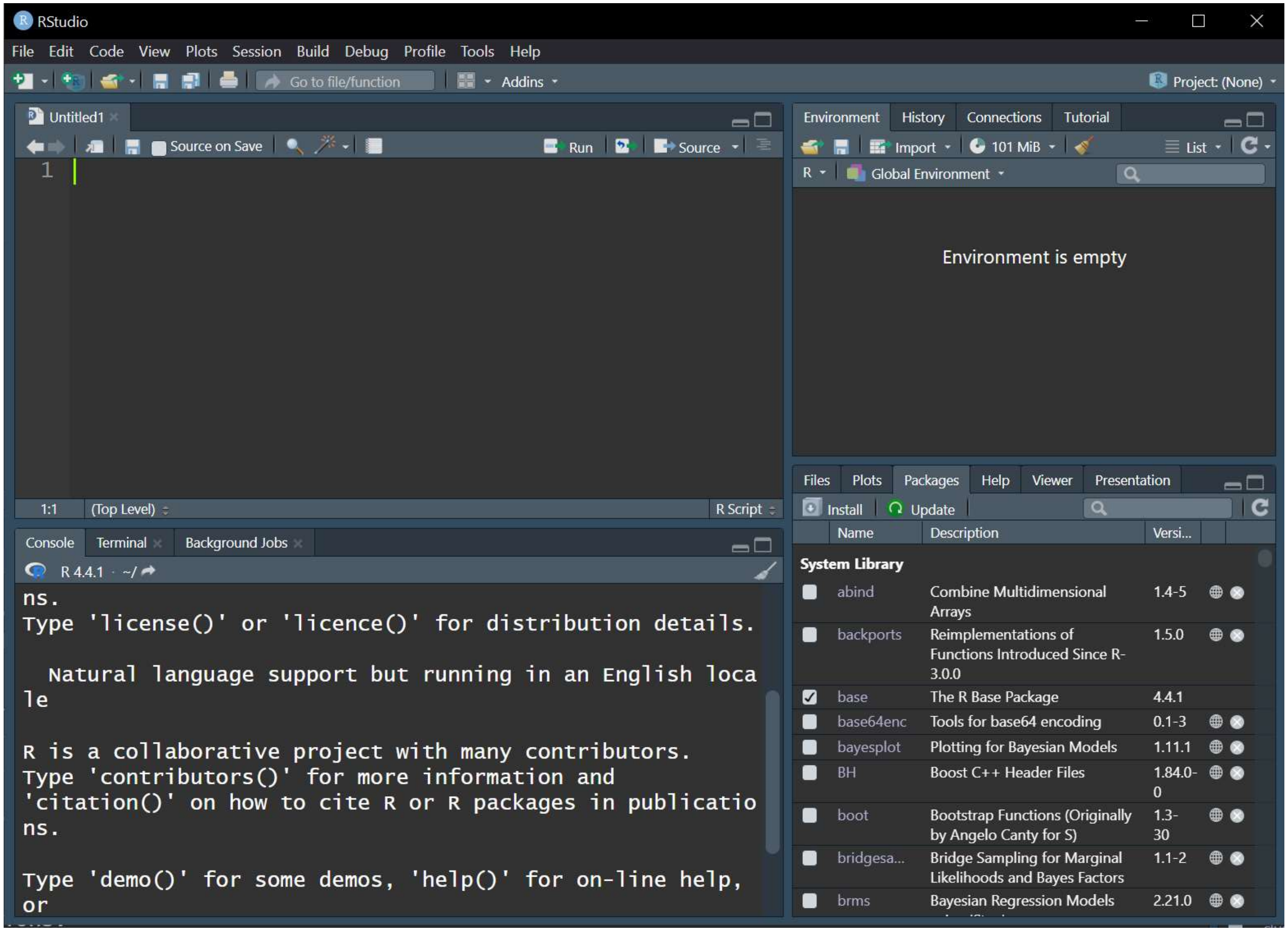


```
R version 4.4.1 (2024-06-14 ucrt) -- "Race for Your Life"
Copyright (C) 2024 The R Foundation for Statistical Computi
ng
Platform: x86_64-w64-mingw32/x64

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain condition
s.
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 publication
s.

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 is empty

	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

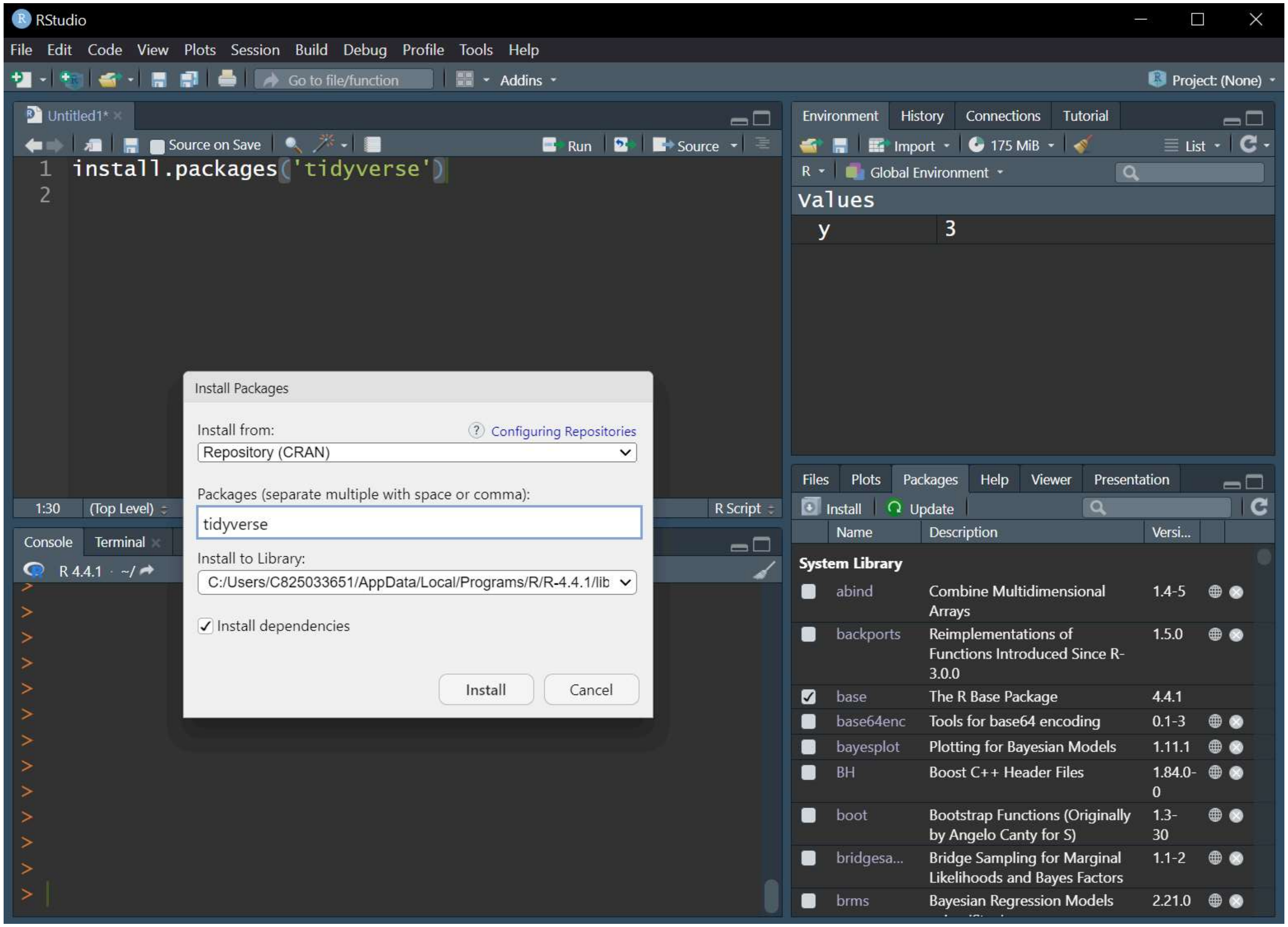

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Untitled1
Source on Save
Run
Source
1:1
(Top Level)
R Script
Environment History Connections Tutorial
Import
101 MiB
List
R
Global Environment
Files Plots Packages Help Viewer Presentation
Install
Update
Console Terminal Background Jobs
R 4.4.1 · ~/

Environment is empty

```
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
```

	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



```
install.packages('tidyverse')
```

Values

y	3

Install Packages

Install from: Configuring Repositories

Repository (CRAN)

Packages (separate multiple with space or comma):

tidyverse

Install to Library:

C:/Users/C825033651/AppData/Local/Programs/R/R-4.4.1/lib

☑ Install dependencies

	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

The language of R

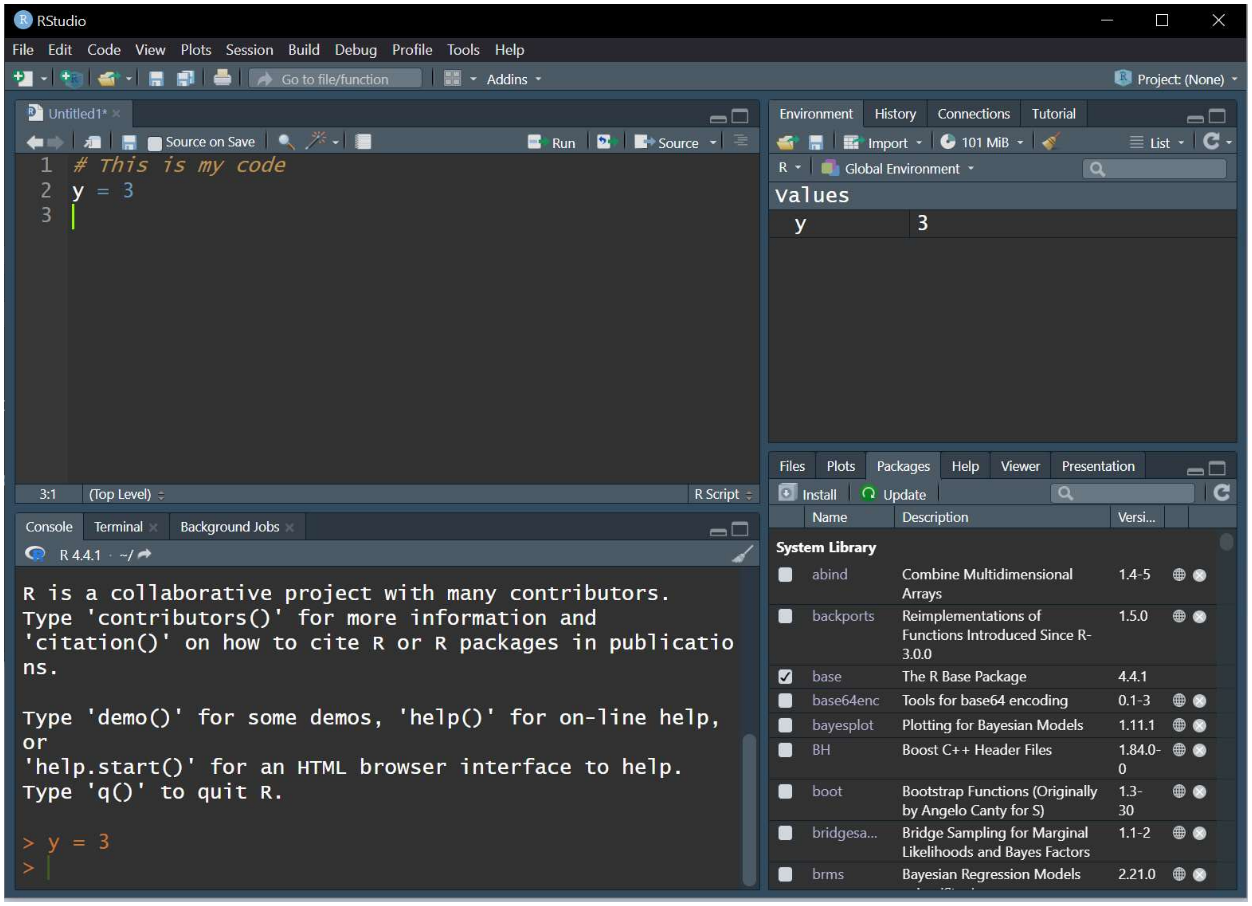
Objects

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

‘Attributes’ describes the structure or information of the object

The language of R

Objects



```r
# This is my code
y = 3

```

```
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
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> y = 3
> 
```

```
Values
  y            3
```

System Library

	Name	Description	Versi...
□	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

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

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

```
[1] 3
```

```r
# Same operation '=' '<-'
y <- 3
```

The language of R

Objects

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

```
[1] 1
```

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

```
[1] 8
```

```r
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 by defining objects and the values given to computer d in km.

▶ 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)

```r
sign(-5)
```

```
[1] -1
```

```r
sign(54)
```

```
[1] 1
```

The language of R

Functions

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

```r
is.numeric(y)
```

```
[1] TRUE
```

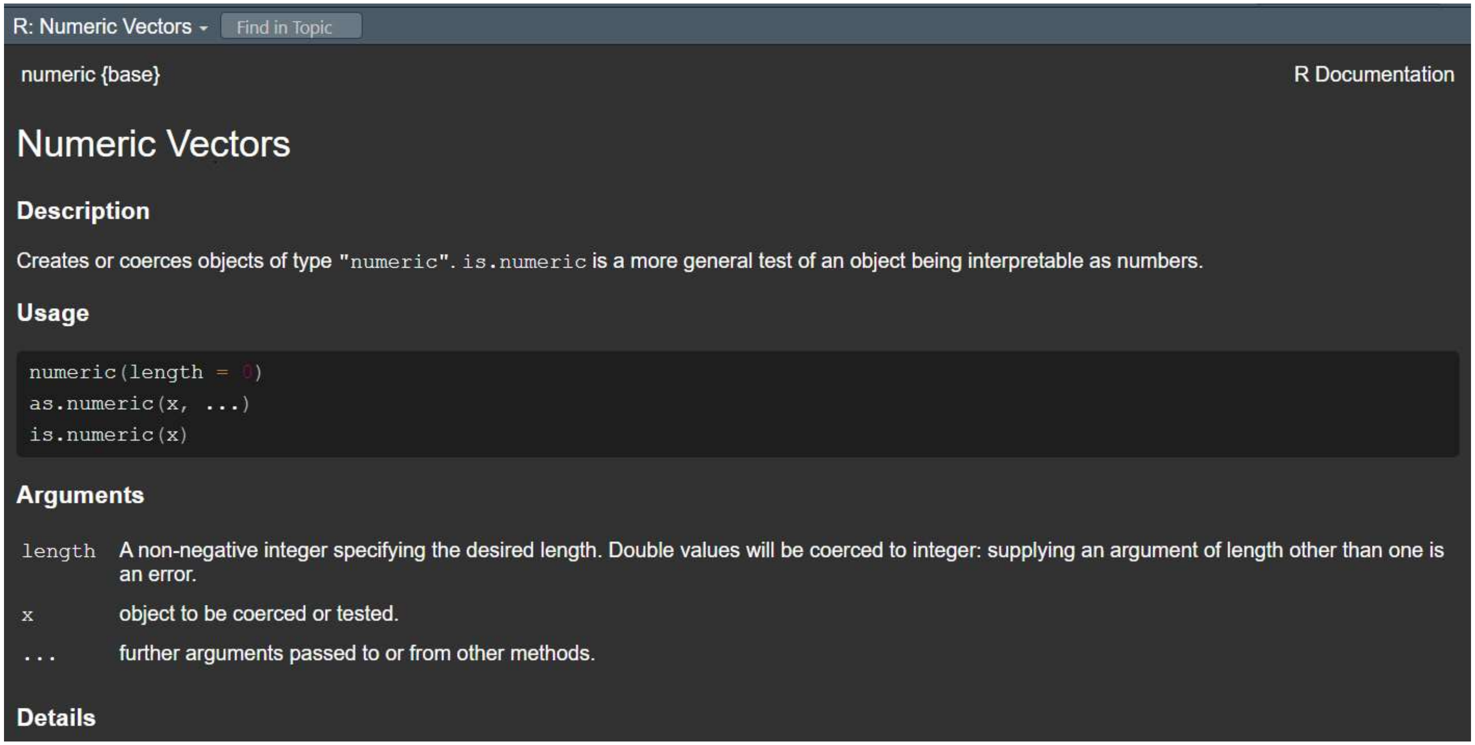
```r
# The function 'class' has the argument 'x'
is.numeric(x = y)
```

```
[1] TRUE
```

The language of R

Functions

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



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

```
# Functions are commonly 1) wrapped, 2) have mul
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
- S3, S4, S5, and beyond

Types of Values

Numeric

```r
y = 3
class(y)
```

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

Integer

```r
y = integer(3)
class(y)
```

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

Character

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

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

Factor

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

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

Types of Objects

Vector

```
# 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

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

```
[1] 3
```

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

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

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

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

```r
z3 = c("dog",1,"horse")
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(2,3)]
```

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

```
z3[-1]
```

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

Types of Objects

Vector of factors, a special kind of character string

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

```r
z4
```

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

```r
levels(z4)
```

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

```r
summary(z4)
```

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

Types of Objects

Matrix

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

```
x
```

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

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

```
[1] 2 3
```

Types of Objects

Subsetting a matrix

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

```
[1] 3
```

```r
# get element of row 2 and column 3
x[2,3]
```

```
[1] 6
```

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

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

```r
# same as
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","e","f"),
            dim = c(2,2,2)
           )
```

```r
dim(z5)
```

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

```r
z5
```

```
, , 1

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

, , 2

     [,1] [,2]
```

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

Types of Objects

List

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

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

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

```r
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).

```
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

```r
x$exposure
```

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

```r
x['exposure']
```

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

```r
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 ‘vec1’, thereby overwriting the original vec1.
4. Create a new vector (length 3) of characters called “hab1”, “hab2”, and “hab3”. Call this object ‘vec2’.
5. Put vec1 and vec2 together into a data frame and call this object ‘dat’

▶ Click for Answer