R Stats Bootcamp

R language

Megan Lewis

2024-11-14

How the R stats bootcamp works

- Practical, open instructional materials for learning R
- Traditional statistics in R
- Reproducible research and collaboration tools
- Self-guided learning and self-assessment
- Friendly community

R stats bootcamp - Module 1

Schedule:

- ~~Session 1: An introduction and script workflow~~
- Session 2: R language
- Session 3: R functions
- Session 4: Data objects
- Session 5: Data frames
- Session 6: Data subsetting

Session 2 objectives:

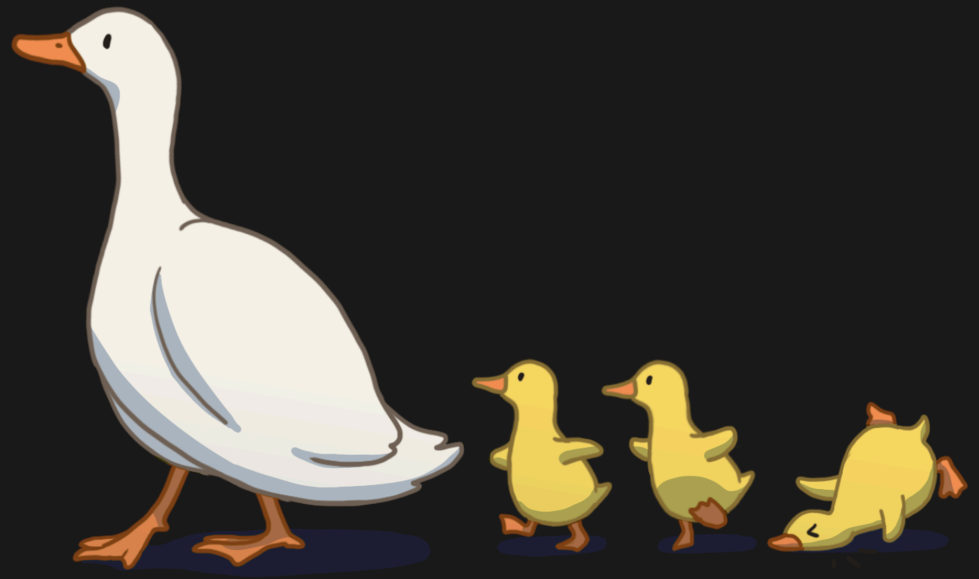
- Example script, comments, help and pseudocode
- Math operators
- Logical boolean operators
- Regarding “base R” and the Tidyverse
- Practice exercises

R Stats Bootcamp Session 2

- R as a passive aggressive butler

Example Script, comments, help and pseudocode

- Demo: The concept and purpose of a script
- Tips:
  - Work through bootcamp by coding
  - Type your own code
  - Document code with comments



Example script

Download and open the example script in R Studio:

Example Script

```
## HEADER ####
## Who: <Your name>
## What: Bootcamp 1_2 R language
## Last edited: <yyyy-mm-dd format>
####
```

Contents > Organisation

```
1 ## CONTENTS ####
2 ## 2 Example script, help, pseudocode
3 ## 3 Math operators
4 ## 4 Logical Boolean operators
5 ## 5 Regarding base R and the Tidyverse
6 ## 6 Practice exercises
```

Comment syntax

- All comments start with at least one hash tag

```
1  # This is a comment
2
3  ## Using two hash tags at the beginning to visually separate section title
4
5  ## Section heading ####
```

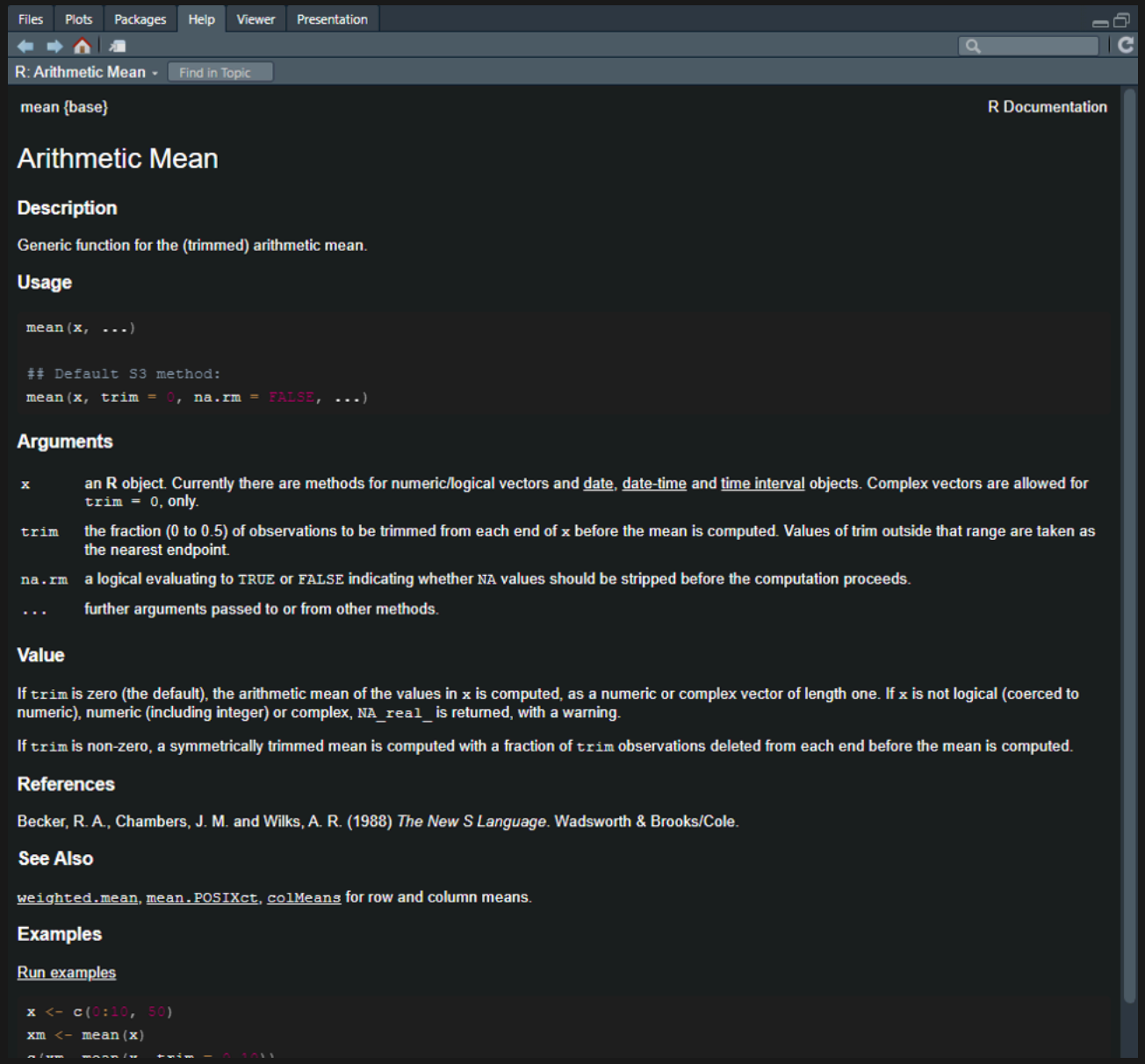
- The ## and #### make it a chunk!
- Organised and good for navigation

Help

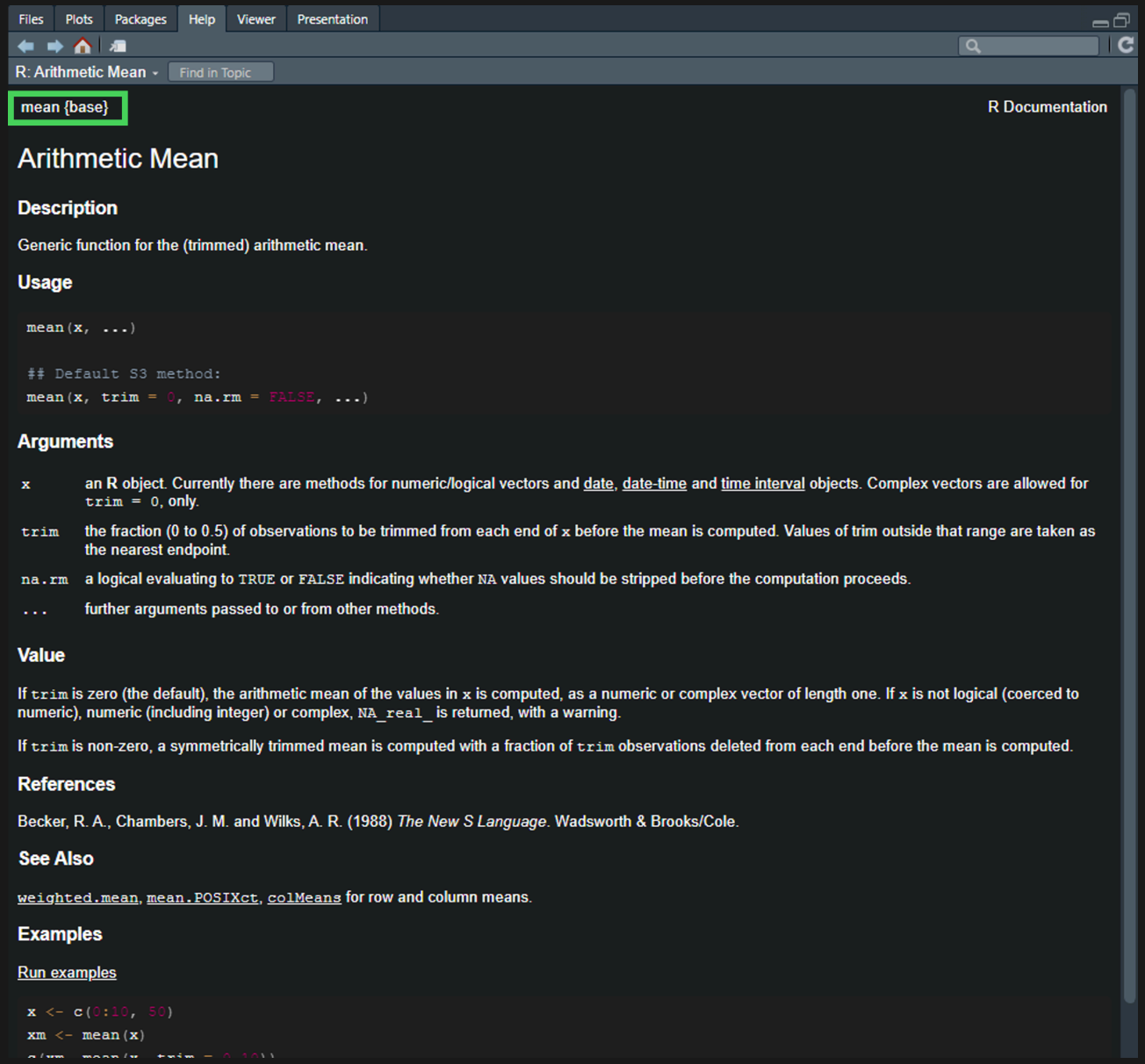
- In R Studio help

```
1 # Display help page for the function 'mean'
2 help(mean)
3
4 # Alternative way of getting help
5 ?mean
```

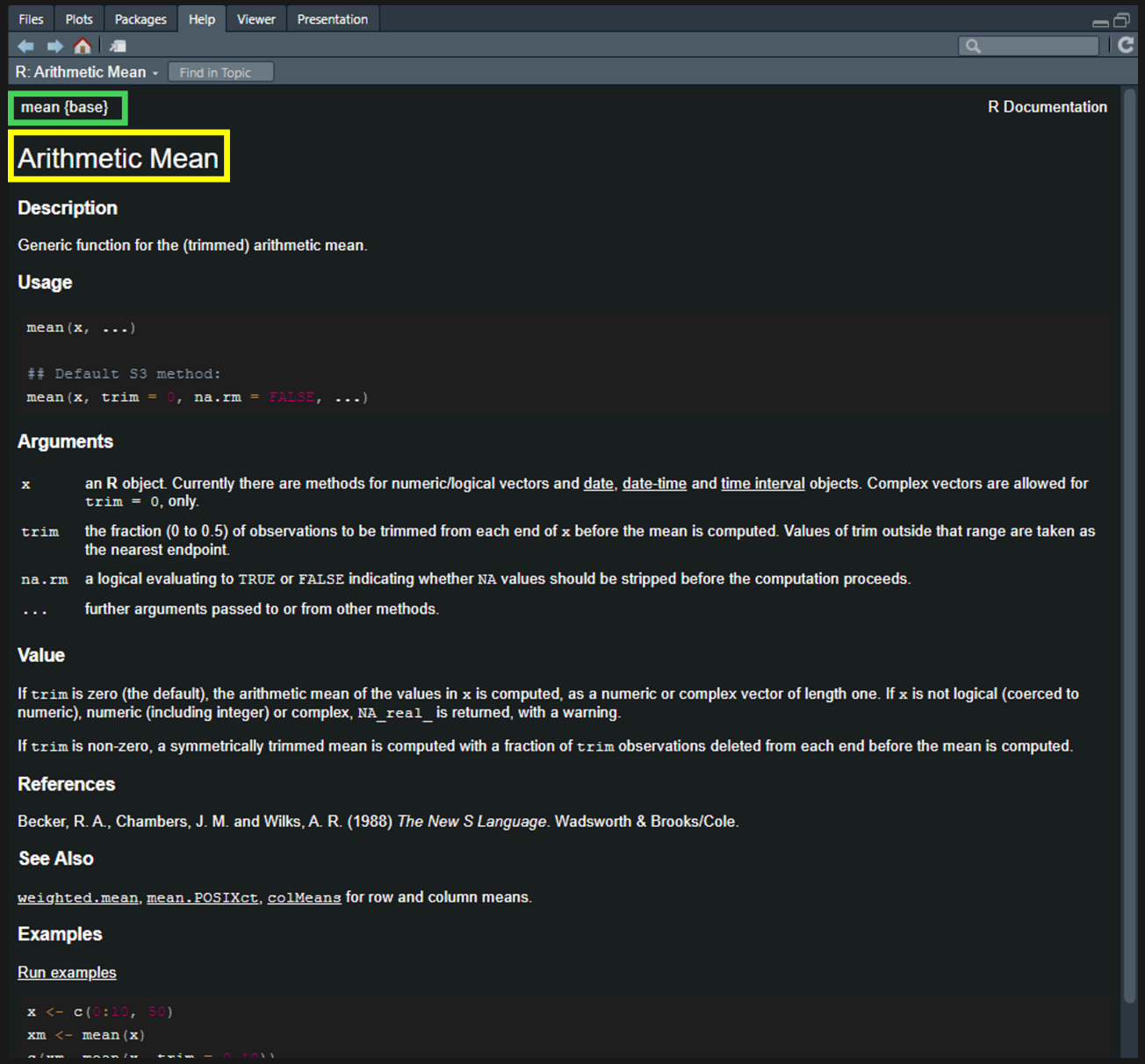
Anatomy of help pages



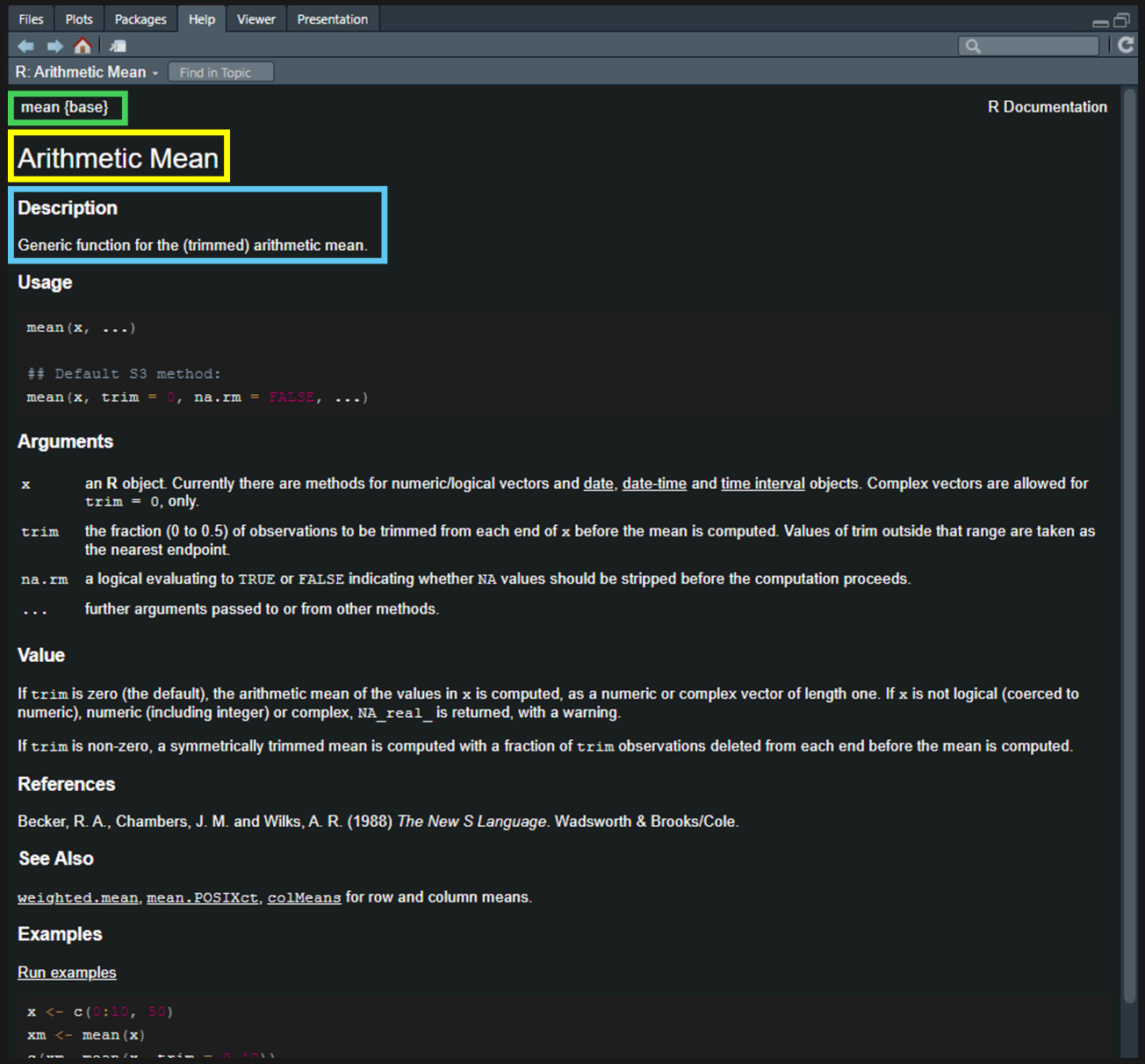
Anatomy of help pages



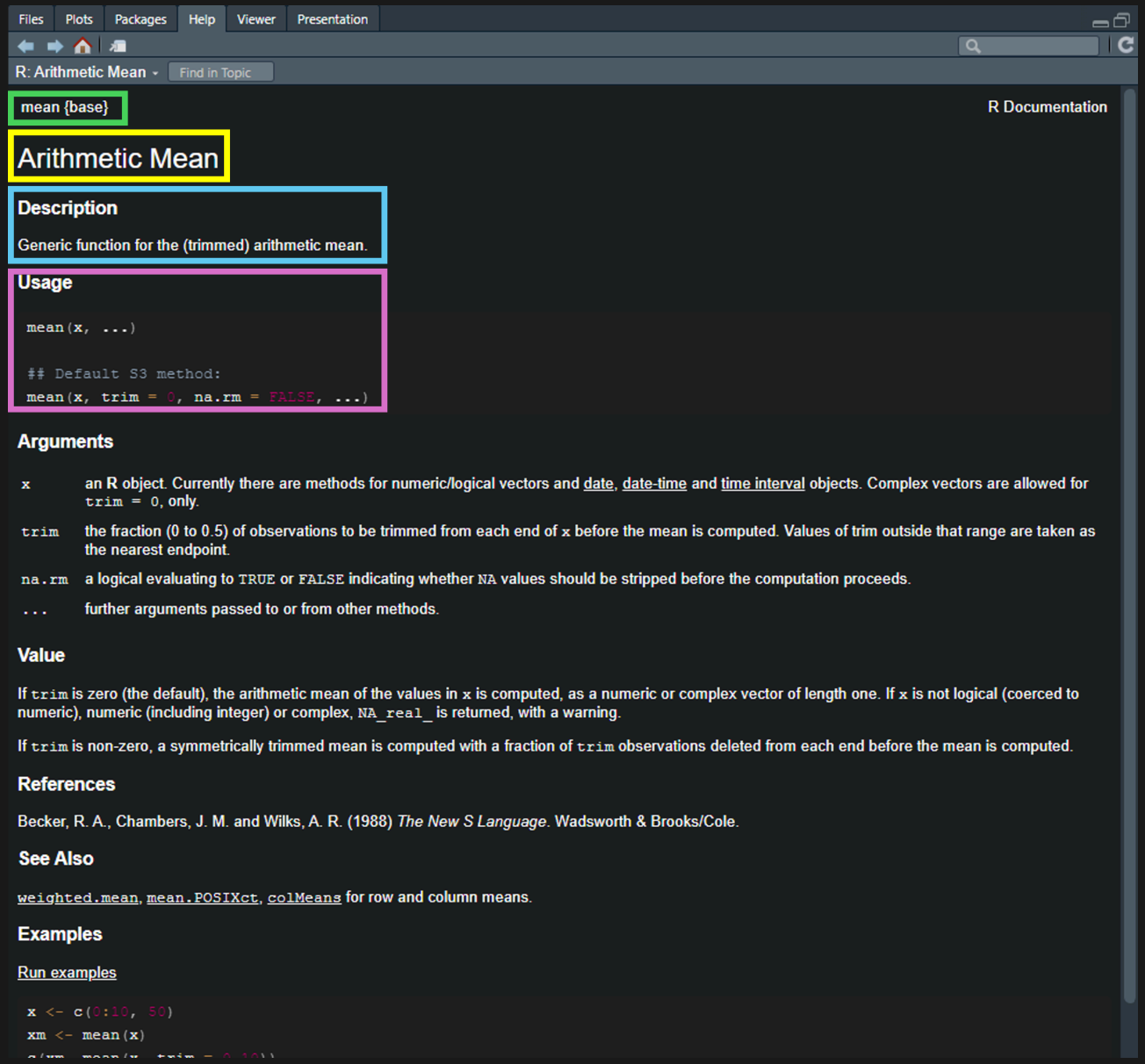
Anatomy of help pages



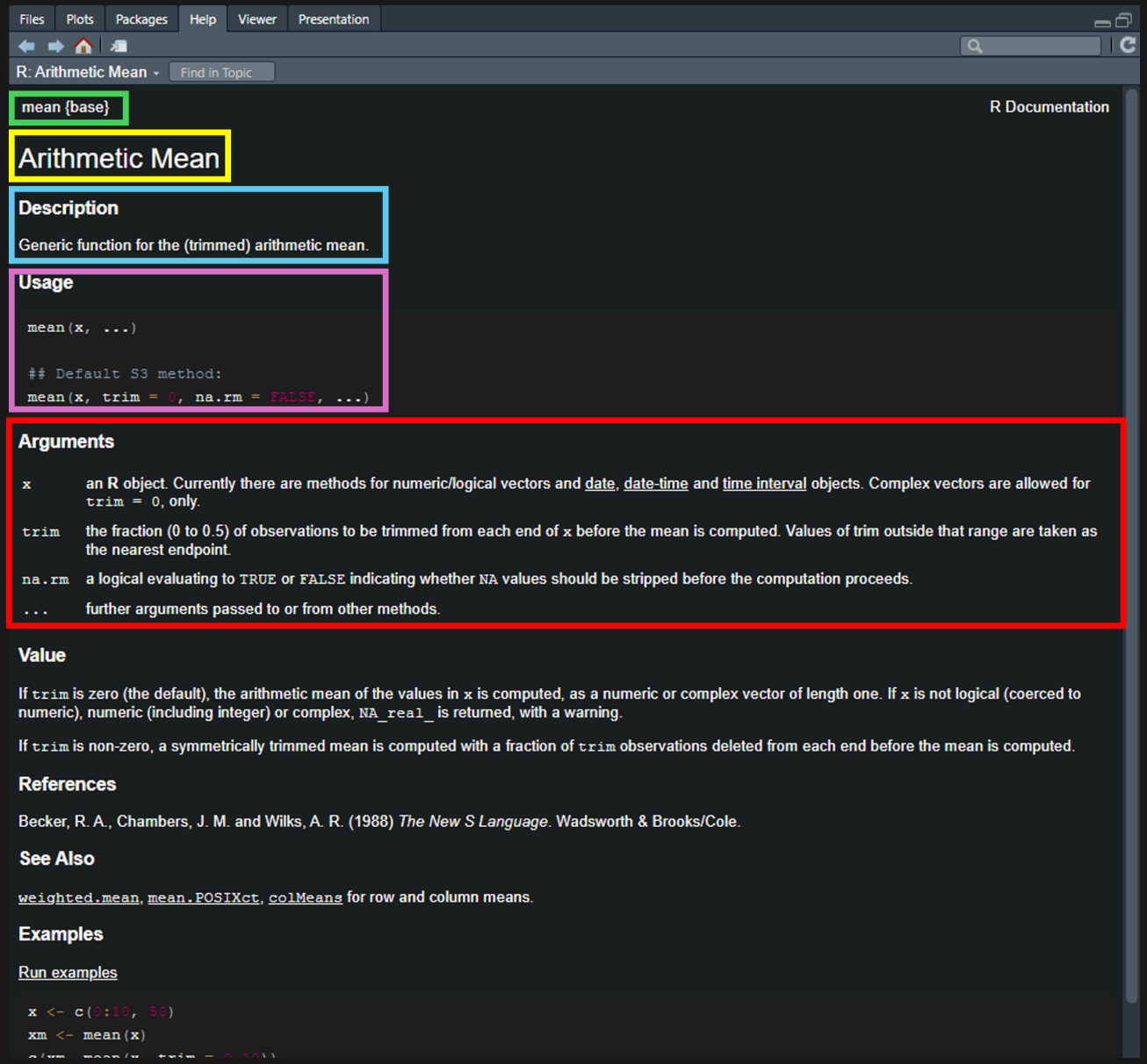
Anatomy of help pages



Anatomy of help pages



Anatomy of help pages



Pseudocode - Tell it to the duck!

- Break problem down into smaller chunks
- Helps with understanding code and problems
- Give example

Math Operators

- R as a calculator
- Addition +
- Subtraction -
- Multiplication *
- Division /

Math operators

```
1  # Add with "+"
2  2 + 5
```

```
[1] 7
```

Math operators

```
1 # Add with "+"
2 2 + 5
```

```
[1] 7
```

```
1 # Subtract with "-"
2 10 - 15
```

```
[1] -5
```

Order of operation

```
1  # Try this
2  4 + 2 * 3
```

```
[1] 10
```

Order of operation

```
1 # Try this
2 4 + 2 * 3
```

```
[1] 10
```

```
1 # Order control - same
2 4 + (2 * 3)
```

```
[1] 10
```

Order of operation

```
1 # Try this
2 4 + 2 * 3
```

```
[1] 10
```

```
1 # Order control - same
2 4 + (2 * 3)
```

```
[1] 10
```

```
1 # Order control - different
2 (4 + 2) * 3
```

```
[1] 18
```

Use of spaces

- Spaces don’t matter, but style…

```
1  # Try this
2  6+10          # No spaces
```

```
[1] 16
```

Use of spaces

- Spaces don’t matter, but style…

```
# Try this
6+10          # No spaces
```

```
[1] 16
```

```
7     - 5     # Uneven spaces
```

```
[1] 2
```

Use of spaces

- Spaces don’t matter, but style…

```
# Try this
6+10          # No spaces
```

```
[1] 16
```

```
7     - 5     # Uneven spaces
```

```
[1] 2
```

```
1.6   /   2.3 # Large spaces
```

```
[1] 0.6956522
```

Use of spaces

- Spaces don’t matter, but style…

```
# Try this
6+10          # No spaces
```

```
[1] 16
```

```
7     - 5     # Uneven spaces
```

```
[1] 2
```

```
1.6   /   2.3 # Large spaces
```

```
[1] 0.6956522
```

```
16 * 3        # Exactly one space
```

```
[1] 48
```

Code formatting

{{< fa lightbulb >}} Formatting shortcut

Shortcut to help reformat selected/highlighted code
Windows/Linux: `Ctrl + Shift + a`
Mac: `Cmd + Shift + a`

Logical Boolean operators

TRUE and FALSE

```
1 # Try this
2 3 > 5 # True, yes?
```

```
[1] FALSE
```

Logical Boolean operators

TRUE and FALSE

```
1 # Try this
2 3 > 5 # True, yes?
```

```
[1] FALSE
```

```
1 # 3 is compared to each element
2 3 < c(1, 2, 3, 4, 5, 6)
```

```
[1] FALSE FALSE FALSE  TRUE  TRUE  TRUE
```

Logical Boolean operators

Useful booleans

```
1  # Try this
2
3  x <- c(21, 3, 5, 6, 22)
4  x
```

```
[1] 21  3  5  6 22
```

Logical Boolean operators

Useful booleans

```
1  # Try this
2
3  x <- c(21, 3, 5, 6, 22)
4  x
```

```
[1] 21  3  5  6 22
```

Selecting with Booleans

```
1  x[x > 20]
```

```
[1] 21 22
```

Logical Boolean operators

The “not” operator, ! (**Sorry !sorry**)

```
1 # Try this
2 TRUE # plain true
```

```
[1] TRUE
```

Logical Boolean operators

The “not” operator, ! (**Sorry !sorry**)

```
1 # Try this
2 TRUE # plain true
```

```
[1] TRUE
```

```
1 !FALSE # not false is true!
```

```
[1] TRUE
```

Logical Boolean operators

The “not” operator, ! (**Sorry !sorry**)

```
# Try this
TRUE # plain true
```

```
[1] TRUE
```

```
!FALSE # not false is true!
```

```
[1] TRUE
```

```
6 < 5 #definitely false
```

```
[1] FALSE
```

Logical Boolean operators

The “not” operator, ! (**Sorry !sorry**)

```
1 # Try this
2 TRUE # plain true
```

```
[1] TRUE
```

```
1 !FALSE # not false is true!
```

```
[1] TRUE
```

```
1 6 < 5 #definitely false
```

```
[1] FALSE
```

```
1 !(6 < 5) #not false...
```

```
[1] TRUE
```

Logical Boolean operators

The “not” operator, ! (**Sorry !sorry**)

```
# Try this
TRUE # plain true
```

```
[1] TRUE
```

```
!FALSE # not false is true!
```

```
[1] TRUE
```

```
6 < 5 #definitely false
```

```
[1] FALSE
```

```
!(6 < 5) #not false...
```

```
[1] TRUE
```

```
!(c(23, 44, 16, 51, 12) > 50)
```

```
[1]  TRUE  TRUE  TRUE FALSE  TRUE
```

Base R vs Tidyverse

Tidyverse

- System of packages for data manipulation, exploration and visualization which share a common design philosophy
- Originally developed by Hadley Wickham Book
- Some find it difficult to understand
- Divergence of R languages

Personal preference - R Bootcamp use Base R

Practice exercises

- Understanding the first two sections of an R script
- Subset in the boxplot function
- R as a calculator
- Pseudocode
- Understanding outputs
- Boolean operators