

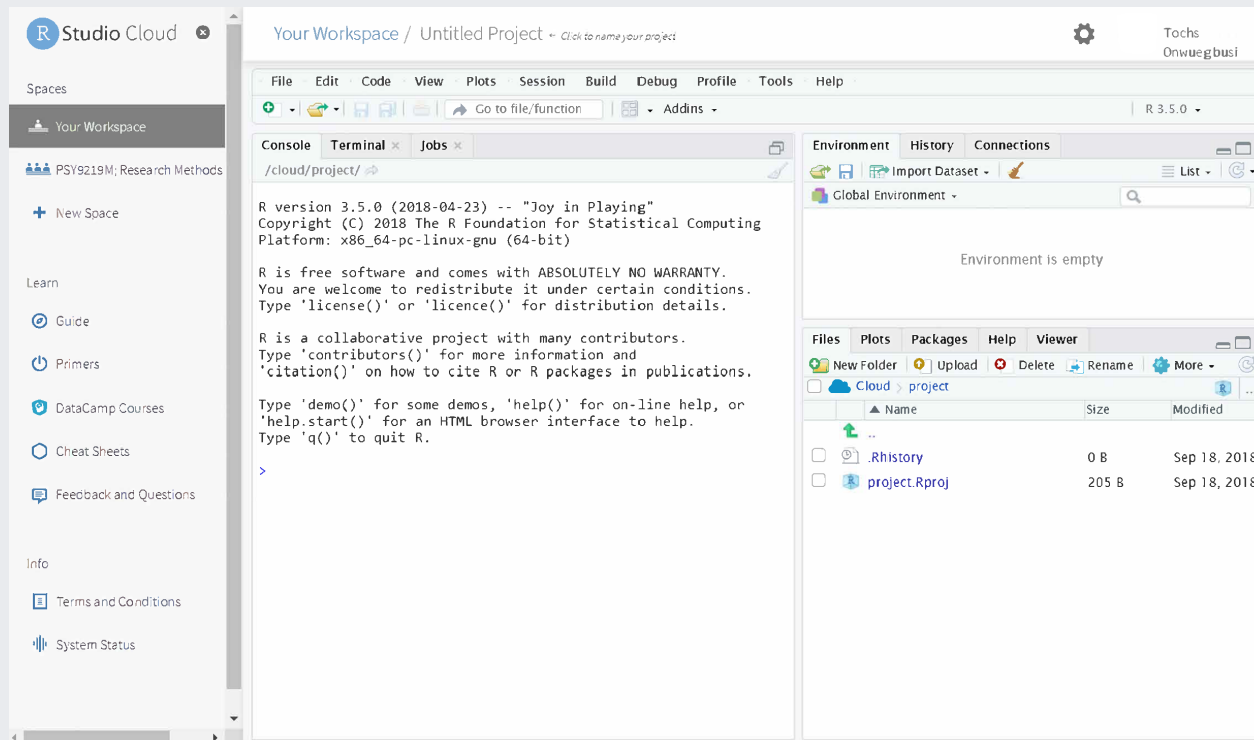
# Introduction to R, part 2

## Research Methods and Skills

10/10/2023

# Interacting with R

- The R Console
  - REPL: Read Evaluate Print Loop
  - Type stuff in, it tries to do it



# Basic use of R

## Use of R like a calculator

The R console allows you to use it like a calculator, as below:

```
5 + 5
```

```
## [1] 10
```

```
10 - 6 * 13
```

```
## [1] -68
```

# Basic use of R

## Creating objects to store information

You assign values to objects using `<-`

```
test_object <- 5
```

`<-` can be read as "is now", making the code above roughly mean

```
The object "test_object" is now 5 # Do not run!
```

Objects "stand-in" for their values:

```
test_object
```

```
## [1] 5
```

# Basic use of R

## Creation of vectors

Vectors are simply a 1-dimensional collection of values of the same type.

E.g. We can create a numeric vector using the `c()` function.

```
c(5, 10, 3, -1, -5)
```

```
## [1]  5 10  3 -1 -5
```

This is a one-dimensional vector of length *five*, since it has 5 values.

# Basic use of R

## Using functions on objects

**Functions** do things to objects.

Brackets after a word in these slides indicate that something is a function, e.g. `c()`, `mean()`

```
mean(c(5, 8, 2, 4, 5))
```

```
## [1] 4.8
```

```
test_object <- c(5, 8, 2, 4, 5)  
mean(test_object)
```

```
## [1] 4.8
```

# R Scripts

# R Scripts

*Scripts* are a way of writing out a sequence of commands that you want R to execute.

A typical script looks something like this:

```
# Load in required packages using library()  
library(tidyverse)  
  
# Define any custom functions here (we haven't covered this!)  
  
# Now load any data you want to work on. (again, we'll cover this later!)  
test_data <-  
  read_csv("data/a-random-RT-file.csv") %>% # I'll explain what %>% means later  
  rename(RT = `reaction times`)  
  
# The rest of the script then runs whatever analyses or plotting you want to do  
ggplot(test_data,  
       aes(x = RT,  
           fill = viewpoint)) +  
  geom_density()
```



# Why is this useful?

Somebody asks you how you performed a particular analysis. In particular, they want detailed instructions of how you created a plot, filtered out outliers or missing data, and performed a linear regression.

Q1: *How would you do that if you used SPSS?*

Q2: *How would you do that if you used R?*

**Let's create a script!**

```
R version 4.0.5 (2021-03-31) -- "Shake and Throw"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
```






































```
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.
```

```
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 is empty

Name	Description	Version	
<b>User Library</b>			
<input type="checkbox"/> abind	Combine Multidimensional Arrays	1.4-5	 
<input type="checkbox"/> ade4	Analysis of Ecological Data: Exploratory and Euclidean Methods in Environmental Sciences	1.7-22	 
<input type="checkbox"/> afex	Analysis of Factorial Experiments	1.1-1	 
<input type="checkbox"/> aod	Analysis of Overdispersed Data	1.3.1	 
<input type="checkbox"/> apaTables	Create American Psychological Association (APA) Style Tables	2.0.8	 
<input type="checkbox"/> arm	Data Analysis Using Regression and Multilevel/Hierarchical Models	1.11-2	 
<input type="checkbox"/> AsioHeaders	'Asio' C++ Header Files	1.22.1-1	 
<input type="checkbox"/> askpass	Safe Password Entry for R, Git, and SSH	1.1	 
<input type="checkbox"/> assertthat	Easy Pre and Post Assertions	0.2.1	 
<input type="checkbox"/> backports	Reimplementations of Functions Introduced Since R-3.0.0	1.4.1	 
<input type="checkbox"/> base64enc	Tools for base64 encoding	0.1-3	 
<input type="checkbox"/> BayesFactor	Computation of Bayes Factors for Common Designs	0.9.12-4.2	 
<input type="checkbox"/> bayesplot	Plotting for Bayesian Models	1.8.0	 
<input type="checkbox"/> bayestestR	Understand and Describe Bayesian Models and Posterior Distributions	0.13.0	 
<input type="checkbox"/> BBmisc	Miscellaneous Helper Functions for B. Bischl	1.12	 
<input type="checkbox"/> beeswarm	The Bee Swarm Plot, an Alternative to Stripchart	0.4.0	 
<input type="checkbox"/> bfw	Bayesian Framework for Computational Modeling	0.4.2	 
<input type="checkbox"/> BH	Boost C++ Header Files	1.81.0-1	 
<input type="checkbox"/> bit	Classes and Methods for Fast Memory-Efficient Boolean Selections	4.0.4	 
<input type="checkbox"/> bit64	A S3 Class for Vectors of 64bit Integers	4.0.5	 

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

New File  
New Project...

Open File... Ctrl+O  
Open File in New Column...  
Recent Files

Open Project...  
Open Project in New Session...  
Recent Projects

Import Dataset

Save Ctrl+S  
Save As...  
Save All Alt+Ctrl+S

Print...

Close Ctrl+W  
Close All Ctrl+Shift+W  
Close All Except Current Alt+Ctrl+Shift+W

Close Project

Quit Session... Ctrl+Q

R Script Ctrl+Shift+N

Quarto Document...  
Quarto Presentation...

R Notebook  
R Markdown...

Shiny Web App...  
Plumber API...

C File  
C++ File  
Header File

Markdown File  
HTML File  
CSS File  
JavaScript File  
D3 Script

Python Script  
Shell Script  
SQL Script  
Stan File  
Text File

R Sweave  
R HTML  
R Documentation...

Environment History Connections Tutorial

Import Dataset 41 MiB

Global Environment

Environment is empty

Files Plots Packages Help Viewer Presentation

R: Sling messages and warnings with flair Find in Topic

say {cowsay} R Documentation

### Sling messages and warnings with flair

#### Description

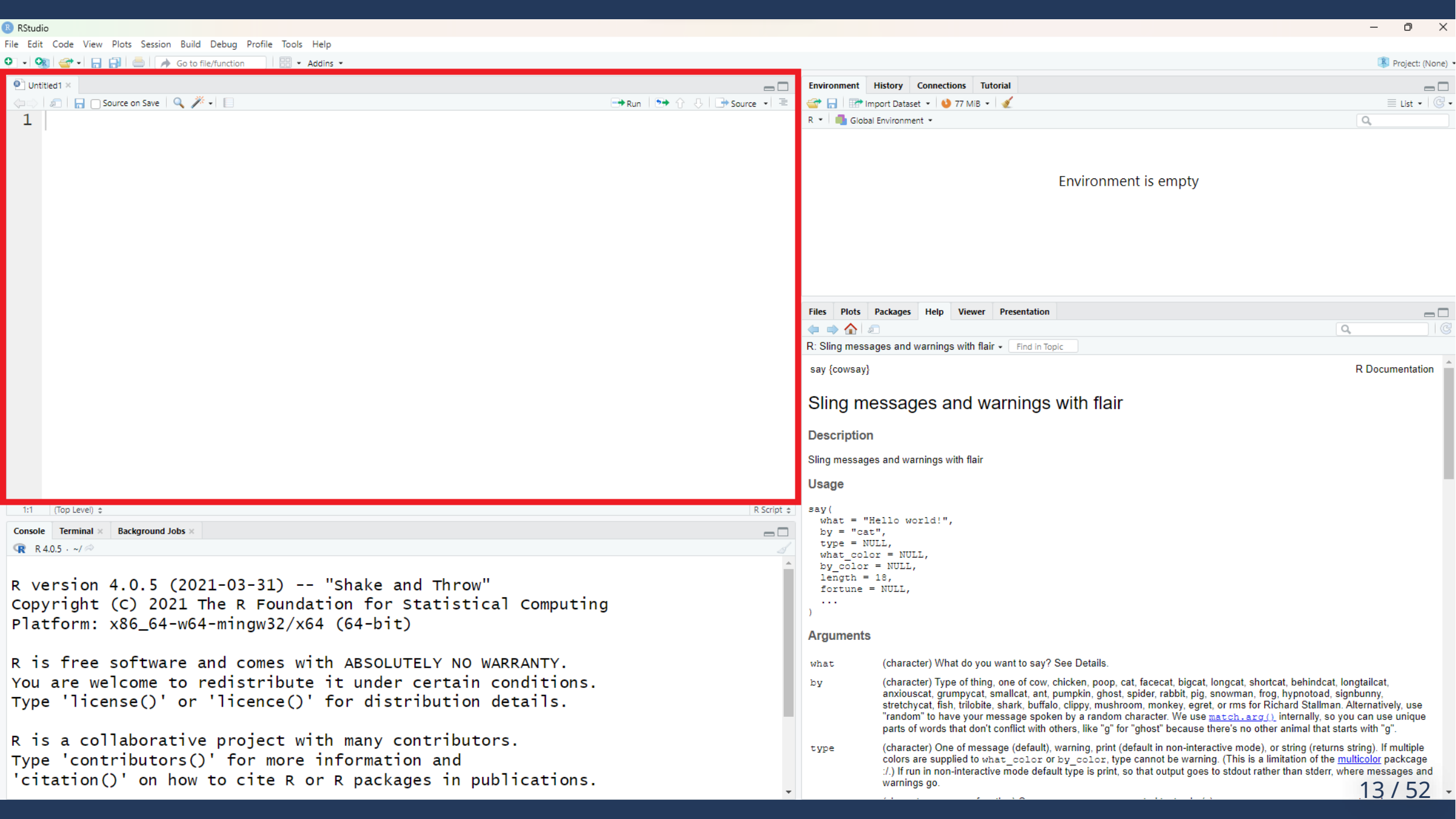
Sling messages and warnings with flair

#### Usage

```
say(  
  what = "Hello world!",  
  by = "cat",  
  type = NULL,  
  what_color = NULL,  
  by_color = NULL,  
  length = 18,  
  fortune = NULL,  
  ...  
)
```

#### Arguments

what	(character) What do you want to say? See Details.
by	(character) Type of thing, one of cow, chicken, poop, cat, facecat, bigcat, longcat, shortcat, behindcat, longtailcat, anxiouscat, grumpycat, smallcat, ant, pumpkin, ghost, spider, rabbit, pig, snowman, frog, hypnotoad, signbunny, stretchycat, fish, trilobite, shark, buffalo, clippy, mushroom, monkey, egret, or rms for Richard Stallman. Alternatively, use "random" to have your message spoken by a random character. We use <a href="#">match.arg()</a> internally, so you can use unique parts of words that don't conflict with others, like "g" for "ghost" because there's no other animal that starts with "g".
type	(character) One of message (default), warning, print (default in non-interactive



Environment is empty

say {cowsay}

## Sling messages and warnings with flair

### Description

Sling messages and warnings with flair

### Usage

```
say(  
  what = "Hello world!",  
  by = "cat",  
  type = NULL,  
  what_color = NULL,  
  by_color = NULL,  
  length = 18,  
  fortune = NULL,  
  ...  
)
```

### Arguments

**what** (character) What do you want to say? See Details.

**by** (character) Type of thing, one of cow, chicken, poop, cat, facecat, bigcat, longcat, shortcat, behindcat, longtailcat, anxiouscat, grumpycat, smallcat, ant, pumpkin, ghost, spider, rabbit, pig, snowman, frog, hypnotoad, signbunny, stretchycat, fish, trilobite, shark, buffalo, clippy, mushroom, monkey, egret, or rms for Richard Stallman. Alternatively, use "random" to have your message spoken by a random character. We use `match.arg()` internally, so you can use unique parts of words that don't conflict with others, like "g" for "ghost" because there's no other animal that starts with "g".

**type** (character) One of message (default), warning, print (default in non-interactive mode), or string (returns string). If multiple colors are supplied to `what_color` or `by_color`, type cannot be warning. (This is a limitation of the `multicolor` package :/) If run in non-interactive mode default type is print, so that output goes to stdout rather than stderr, where messages and warnings go.

R version 4.0.5 (2021-03-31) -- "Shake and Throw"  
Copyright (C) 2021 The R Foundation for Statistical Computing  
Platform: x86\_64-w64-mingw32/x64 (64-bit)

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.

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.

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function

Addins

Untitled1\*

Run

Source

```
1 # Load the necessary packages
2 library(cowsay)
3
4 # Define some custom objects
5 say_what <- "This is what I want you to see"
6 by_animal <- "cow"
7
8 # Write the function out
9 say (what = say_what, by = by_animal)
```

8:25 (Top Level) R Script

Console

Terminal

Background Jobs

R 4.0.5 ~/ >

Environment

History

Connections

Tutorial

Import Dataset 95 MiB

R Global Environment

Environment is empty

Files Plots Packages Help Viewer Presentation

R: Sling messages and warnings with flair Find in Topic

say {cowsay}

R Documentation

Sling messages and warnings with flair

Description

Sling messages and warnings with flair

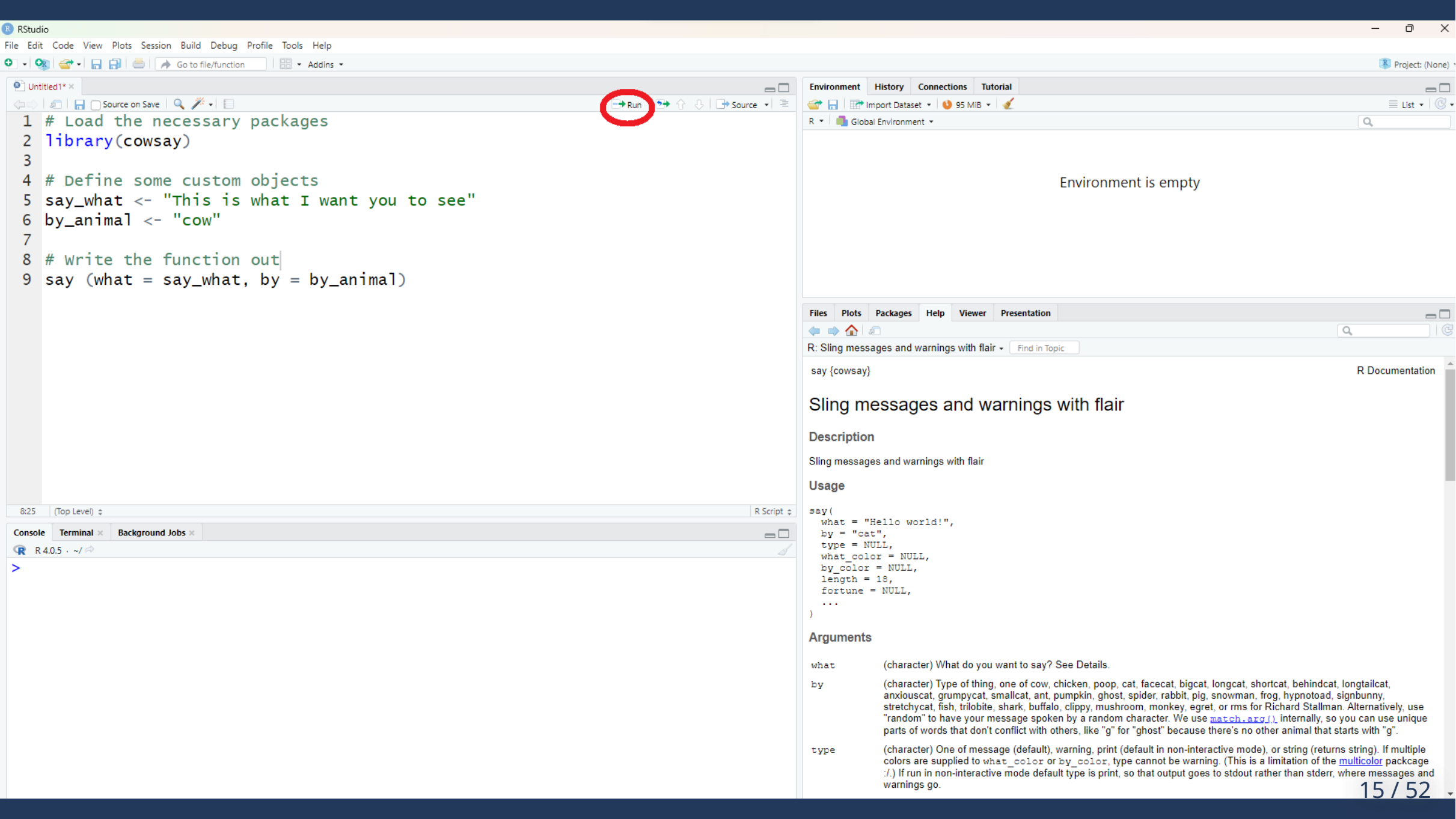
Usage

```
say(
  what = "Hello world!",
  by = "cat",
  type = NULL,
  what_color = NULL,
  by_color = NULL,
  length = 18,
  fortune = NULL,
  ...
)
```

Arguments

what	(character) What do you want to say? See Details.
by	(character) Type of thing, one of cow, chicken, poop, cat, facecat, bigcat, longcat, shortcat, behindcat, longtailcat, anxiouscat, grumpycat, smallcat, ant, pumpkin, ghost, spider, rabbit, pig, snowman, frog, hypnotoad, signbunny, stretchycat, fish, trilobite, shark, buffalo, clippy, mushroom, monkey, egret, or rms for Richard Stallman. Alternatively, use "random" to have your message spoken by a random character. We use <code>match.arg()</code> internally, so you can use unique parts of words that don't conflict with others, like "g" for "ghost" because there's no other animal that starts with "g".
type	(character) One of message (default), warning, print (default in non-interactive mode), or string (returns string). If multiple colors are supplied to <code>what_color</code> or <code>by_color</code> , type cannot be warning. (This is a limitation of the <code>multicolor</code> package .) If run in non-interactive mode default type is print, so that output goes to stdout rather than stderr, where messages and warnings go.

14 / 52



```
1 # Load the necessary packages
2 library(cowsay)
3
4 # Define some custom objects
5 say_what <- "This is what I want you to see"
6 by_animal <- "cow"
7
8 # Write the function out
9 say (what = say_what, by = by_animal)
```

Environment is empty

say {cowsay}

R Documentation

## Sling messages and warnings with flair

### Description

Sling messages and warnings with flair

### Usage

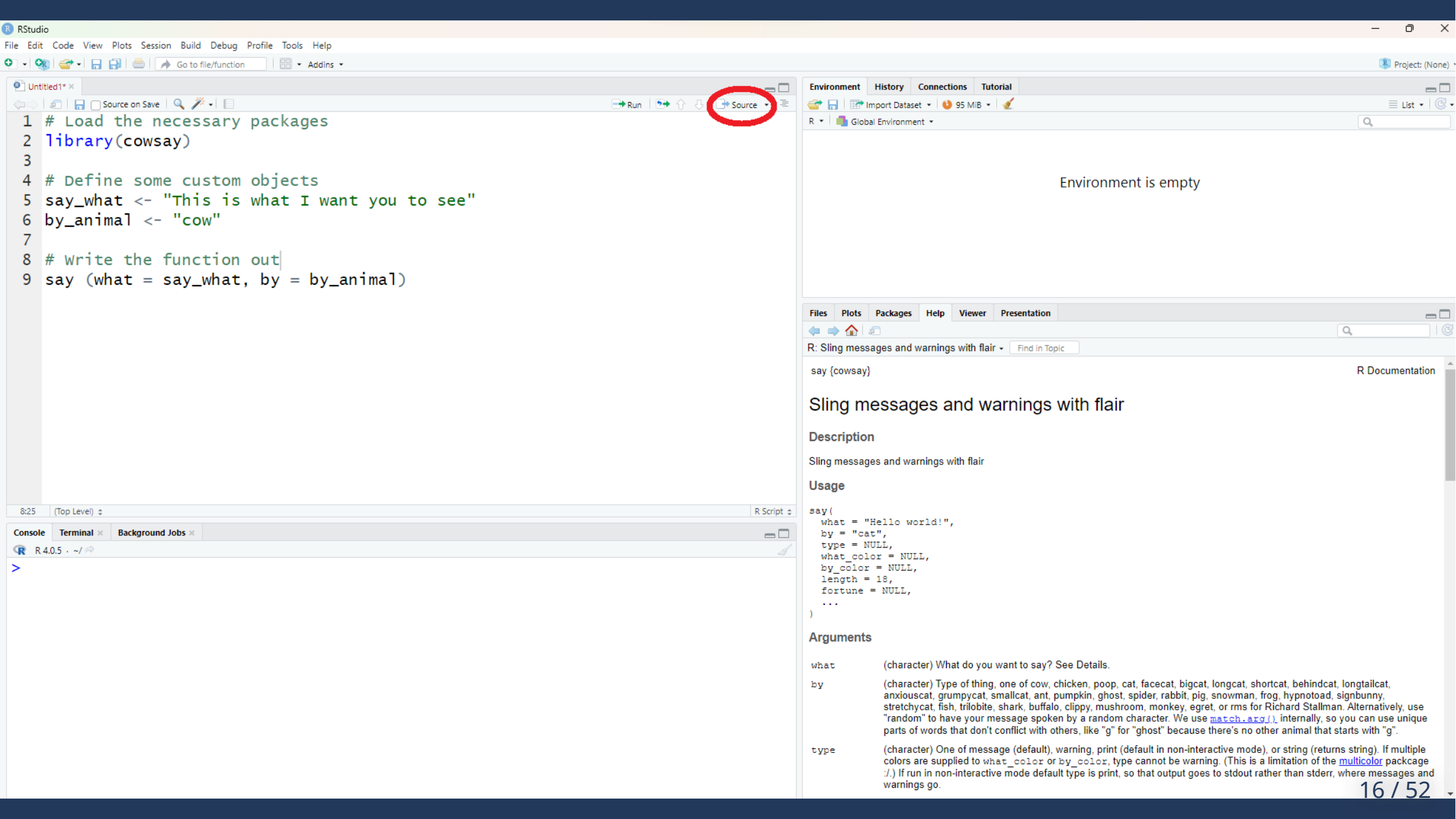
```
say(
  what = "Hello world!",
  by = "cat",
  type = NULL,
  what_color = NULL,
  by_color = NULL,
  length = 18,
  fortune = NULL,
  ...
)
```

### Arguments

**what** (character) What do you want to say? See Details.

**by** (character) Type of thing, one of cow, chicken, poop, cat, facecat, bigcat, longcat, shortcat, behindcat, longtailcat, anxiouscat, grumpycat, smallcat, ant, pumpkin, ghost, spider, rabbit, pig, snowman, frog, hypnotoad, signbunny, stretchycat, fish, trilobite, shark, buffalo, clippy, mushroom, monkey, egret, or rms for Richard Stallman. Alternatively, use "random" to have your message spoken by a random character. We use `match.arg()` internally, so you can use unique parts of words that don't conflict with others, like "g" for "ghost" because there's no other animal that starts with "g".

**type** (character) One of message (default), warning, print (default in non-interactive mode), or string (returns string). If multiple colors are supplied to `what_color` or `by_color`, type cannot be warning. (This is a limitation of the `multicolor` package .) If run in non-interactive mode default type is print, so that output goes to stdout rather than stderr, where messages and warnings go.





FileEditCodeViewPlotsSessionBuildDebugProfileToolsHelp

Go to file/functionAddins

Untitled1\*

RunSource

```
1 # Load the necessary packages
2 library(cowsay)
3
4 # Define some custom objects
5 say_what <- "This is what I want you to see"
6 by_animal <- "cow"
7
8 # Write the function out
9 say (what = say_what, by = by_animal)
```

8:25 (Top Level)R Script

ConsoleTerminalBackground Jobs

R 4.0.5 ~/  
> source("~/active-rstudio-document")  
  
-----  
This is what I want you to see  
-----  
 \ ^\_\_^  
 \ (oo)\\_\_\_\_\_  
 (\_\_)\\ )\/ \  
 ||----w |  
 || ||

> |

EnvironmentHistoryConnectionsTutorial

Import Dataset78 MiB

Global Environment

Values

by_animal	"cow"
say_what	"This is what I want you to see"

FilesPlotsPackagesHelpViewerPresentation

R: Sling messages and warnings with flairFind in Topic

R Documentation

say {cowsay}

Sling messages and warnings with flair

Description

Sling messages and warnings with flair

Usage

say(  
 what = "Hello world!",  
 by = "cat",  
 type = NULL,  
 what\_color = NULL,  
 by\_color = NULL,  
 length = 18,  
 fortune = NULL,  
 ...  
)

Arguments

what (character) What do you want to say? See Details.

by (character) Type of thing, one of cow, chicken, poop, cat, facecat, bigcat, longcat, shortcat, behindcat, longtailcat, anxiouscat, grumpycat, smallcat, ant, pumpkin, ghost, spider, rabbit, pig, snowman, frog, hypnotoad, signbunny, stretchycat, fish, trilobite, shark, buffalo, clippy, mushroom, monkey, egret, or rms for Richard Stallman. Alternatively, use "random" to have your message spoken by a random character. We use match.arg() internally, so you can use unique parts of words that don't conflict with others, like "g" for "ghost" because there's no other animal that starts with "g".

type (character) One of message (default), warning, print (default in non-interactive mode), or string (returns string). If multiple colors are supplied to what\_color or by\_color, type cannot be warning. (This is a limitation of the multicolor package :.) If run in non-interactive mode default type is print, so that output goes to stdout rather than stderr, where messages and warnings go.

17 / 52

# R Markdown

# R Markdown

**Literate programming** is a mixture of plain text and code.

Whereas in scripts you need to use the **#** symbol to indicate comments, as here

```
# This is a comment
```

...with R Markdown you can mix plain text and code using **chunks** to delineate sections of code.

This allows you to create elaborate documents following the structure *you* want!

New File

Open File...  
Ctrl+O

Open File in New Column...  
Reopen with Encoding...

Recent Files

Import Dataset

Save  
Ctrl+S

Save As...

Save with Encoding...

Save All  
Ctrl+Alt+S

Compile Report...

Print...

Close  
Ctrl+W

Close All  
Ctrl+Shift+W

Close All Except Current  
Ctrl+Alt+Shift+W

R Script  
Ctrl+Alt+Shift+N

Quarto Document...

Quarto Presentation...

R Notebook

R Markdown...

Shiny Web App...

Plumber API...

C File

C++ File

Header File

Markdown File

HTML File

CSS File

JavaScript File

D3 Script

Python Script

Shell Script

SQL Script

Stan File

Text File

R Sweave

R HTML

R Documentation...

Run

Source

Create a new R Markdown document

1:1 (Top Level)

Console

Terminal

Background Jobs

R 4.3.1 · /cloud/project/

>

R 4.3.1

Environment

History

Connections

Tutorial

Import Dataset

183 MiB

Global Environment

Environment is empty

Files

Plots

Packages

Help

Viewer

Presentation

R: Arithmetic Mean

Find in Topic

mean {base}

Arithmetic Mean

Description

Generic function for the (trimmed) arithmetic mean.

Usage

mean(x, ...)






## Default S3 method:

mean(x, trim = 0, na.rm = FALSE, ...)

Arguments

An R object. Currently there are methods for numeric/logical vectors and data frames.

File Edit Code View Plots Session Build Debug Profile Tools Help


    Go to file/function  Addins

R 4.3.1

Untitled1

     Run   Source

Environment History Connections Tutorial

 Import Dataset 249 MiB

R Global Environment

List

Environment is empty

## Install Required Packages



Creating R Markdown documents requires updated versions of the following packages: base64enc, digest, evaluate, glue, highr, htmltools, jsonlite, knitr, magrittr, markdown, mime, rmarkdown, stringi, stringr, xfun, yaml.

Do you want to install these packages now?

Yes

No

packages Help Viewer Presentation

Find in Topic

R Documentation

1:1 (Top Level)

Console Terminal Background Jobs

R 4.3.1 · /cloud/project/

&gt;

Generic function for the (trimmed) arithmetic mean.

## Usage

`mean(x, ...)`

## Default S3 method:

`mean(x, trim = 0, na.rm = FALSE, ...)`

## Arguments

An **R** object. Currently there are methods for numeric/logical vectors and [date](#).

File Edit Code View Plots Session Build Debug Profile Tools Help

R 4.3.1

Untitled1 x

Go to file/function

Addins

Run

Environment History Connections Tutorial

Import Dataset 249 MiB

List

1

R Global Environment

Q

Environment is empty

Viewer Presentation

R Documentation

(trimmed) arithmetic mean.

```
## Default S3 method:  
mean(x, trim = 0, na.rm = FALSE, ...)
```

## Arguments

An **R** object. Currently there are methods for numeric/logical vectors and [Date](#), [Date](#), [Date](#)

24/52

### New R Markdown

- ☒ Document
- ☐ Presentation
- ☐ Shiny
- ☐ From Template

Title: GOOD MORNING LOL

Author: Tochs Onwuegbusi

Date: 2023-10-08

☐ Use current date when rendering document

#### Default Output Format:

☒ HTML

Recommended format for authoring (you can switch to PDF or Word output anytime).

☐ PDF

PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

☐ Word







Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).

Create Empty Document

OK

Cancel

File Edit Code View Plots Session Build Debug Profile Tools Help

     Go to file/function  Addins

R 4.3.1

Untitled1 x Untitled2 x  
Source Visual Outline

```
1 ---
2 title: "GOOD MORNING LOL"
3 author: "Tochs Onwuegbusi"
4 date: "2023-10-08"
5 output: html_document
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ```
11
12 ## R Markdown
13
14 This is an R Markdown document. Markdown is a simple formatting
15 syntax for authoring HTML, PDF, and MS Word documents. For more
16 details on using R Markdown see <http://rmarkdown.rstudio.com>.
17
18 When you click the **Knit** button a document will be generated that
19 includes both content as well as the output of any embedded R code
20 chunks within the document. You can embed an R code chunk like this:
21
22 ```{r cars}
23 summary(cars)
24 ```
25
26 ## Including Plots
```

2:1 GOOD MORNING LOL




R Markdown

Console Terminal x Background Jobs x

R 4.3.1 · /cloud/project/

&gt;

Environment History Connections Tutorial

  Import Dataset 250 MiB 

R Global Environment

List

Environment is empty

Files Plots Packages Help Viewer Presentation

    Find in Topic

R: Arithmetic Mean

R Documentation

mean {base}

## Arithmetic Mean

### Description

Generic function for the (trimmed) arithmetic mean.

### Usage

mean(x, ...)

## Default S3 method:

mean(x, trim = 0, na.rm = FALSE, ...)

### Arguments

An **R** object. Currently there are methods for numeric/logical vectors and [data](#).

23/52

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

R 4.3.1

Untitled1 x Untitled2 x

Source Visual Knit ⚙️

Run ↻

Outline

```
1 ---
2 title: "GOOD MORNING LOL"
3 author: "Tochs Onwuegbusi"
4 date: "2023-10-08"
5 output: html_document
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ```
```

```
11 ## R Markdown
```

```
12 This is an R Markdown document. Markdown is a simple formatting
13 syntax for authoring HTML, PDF, and MS Word documents. For more
14 details on using R Markdown see <http://rmarkdown.rstudio.com>.
```

```
15
16 When you click the **Knit** button a document will be generated that
17 includes both content as well as the output of any embedded R code
18 chunks within the document. You can embed an R code chunk like this:
```

```
18 ```{r cars}
19 summary(cars)
20 ```
```

```
21
22 ## Including Plots
```

Console Terminal x Background Jobs x

R 4.3.1 · /cloud/project/

&gt;

# CODE CHUNK

Environment is empty

Environment History Connections Tutorial

Import Dataset 250 MiB

R Global Environment

Files Plots Packages Help Viewer Presentation

R: Arithmetic Mean Find in Topic

mean {base}

R Documentation

## Arithmetic Mean

### Description

Generic function for the (trimmed) arithmetic mean.

### Usage

```
mean(x, ...)
```

```
## Default S3 method:
```

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

### Arguments

An **R** object. Currently there are methods for numeric/logical vectors and [data frames](#).



The screenshot displays the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The toolbar contains icons for saving, running, and other functions. The main editor window shows a code chunk with R Markdown syntax:

```
1 ---  
2 title: "GOOD MORNING LOL"  
3 author: "Tochs Onwuegbusi"  
4 date: "2023-10-08"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ```  
11  
12 ## R Markdown  
13  
14 This is an R Markdown document. Markdown is a simple formatting  
15 syntax for authoring HTML, PDF, and MS Word documents. For more  
16 details on using R Markdown see <http://rmarkdown.rstudio.com>.  
17  
18 When you click the **Knit** button a document will be generated that  
19 includes both content as well as the output of any embedded R code  
20 chunks within the document. You can embed an R code chunk like this:  
21  
22 ```{r cars}  
23 summary(cars)  
24 ```  
25  
26 ## Including Plots
```

A context menu is open over the code, listing various run options:

- Run Selected Line(s) Ctrl+Enter
- Run Current Chunk Ctrl+Shift+Enter
- Run Next Chunk Ctrl+Alt+N
- Run Setup Chunk
- Run Setup Chunk Automatically
- Run All Chunks Above Ctrl+Alt+Shift+P
- Run All Chunks Below
- Restart R and Run All Chunks
- Restart R and Clear Output
- Run All Ctrl+Alt+R

The Environment pane on the right shows "Environment is empty". The bottom pane displays the R Documentation for the `mean` function:

### Arithmetic Mean

#### Description

Generic function for the (trimmed) arithmetic mean.

#### Usage

```
mean(x, ...)
```

## Default S3 method:

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

#### Arguments

An **R** object. Currently there are methods for numeric/logical vectors and `data.frame`.

File Edit Code View Plots Session Build Debug Profile Tools Help

R 4.3.1

Untitled1 x Untitled2 x  
Source Visual Outline

```
1 ---
2 title: "GOOD MORNING LOL"
3 author: "Tochs Onwuegbusi"
4 date: "2023-10-08"
5 output: html_document
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ```
11
12 ## R Markdown
13
14 This is an R Markdown document. Markdown is a simple formatting
15 syntax for authoring HTML, PDF, and MS Word documents. For more
16 details on using R Markdown see <http://rmarkdown.rstudio.com>.
17
18 When you click the **Knit** button a document will be generated that
19 includes both content as well as the output of any embedded R code
20 chunks within the document. You can embed an R code chunk like this:
21
22 ```{r cars}
23 summary(cars)
24 ```
25
26 ## Including Plots
```



Environment History Connections Tutorial

Import Dataset 250 MiB

R Global Environment

Environment is empty

Files Plots Packages Help Viewer Presentation

R: Arithmetic Mean Find in Topic

mean {base}

R Documentation

## Arithmetic Mean

### Description

Generic function for the (trimmed) arithmetic mean.

### Usage

```
mean(x, ...)
```

```
## Default S3 method:
```

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

### Arguments

An **R** object. Currently there are methods for numeric/logical vectors and [data](#).

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Untitled1 Untitled2\*

Knit

Knit to HTML

Knit to PDF

Knit to Word

Knit with Parameters...

Knit Directory

Clear Knitr Cache...

# CLICK KNIT

```
GOOD MORNING LOL"
Onwuegbusi"
08"
document
```

```
8- ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
```

```
10-
11-
12- ## R Markdown
```

```
13-
14- This is an R Markdown document. Markdown is a simple formatting
15- syntax for authoring HTML, PDF, and MS Word documents. For more
16- details on using R Markdown see <http://rmarkdown.rstudio.com>.
```

```
17- When you click the **Knit** button a document will be generated that
18- includes both content as well as the output of any embedded R code
19- chunks within the document. You can embed an R code chunk like this:
```

```
20- ```{r cars}
21- summary(cars)
```

```
22- ## Including Plots
```

2:1 GOOD MORNING LOL

R Markdown

Environment History Connections Tutorial

Import Dataset 181 MiB

R Global Environment

Environment is empty

Files Plots Packages Help Viewer Presentation

R: Arithmetic Mean Find in Topic

mean {base}

R Documentation

## Arithmetic Mean

### Description

Generic function for the (trimmed) arithmetic mean.

### Usage







```
mean(x, ...)
```

```
## Default S3 method:
```

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

### Arguments

File Edit Code View Plots Session Build Debug Profile Tools Help

     Go to file/function  Addins

R 4.3.1

Untitled1 x test\_rmd.Rmd x

     Knit ⚙️   Run 

Source Visual

Outline




```
1 ---
2 title: "GOOD MORNING LOL"
3 author: "Tochs Onwuegbusi"
4 date: "2023-10-08"
5 output: html_document
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ```
11
12 ## R Markdown
13
14 This is an R Markdown document. Markdown is a simple formatting
15 syntax for authoring HTML, PDF, and MS Word documents. For more
16 details on using R Markdown see <http://rmarkdown.rstudio.com>.
17
18 When you click the Knit button a document will be generated that
19 includes both content as well as the output of any embedded R code
20 chunks within the document. You can embed an R code chunk like this:
21
22 ```{r cars}
23 summary(cars)
24 ```
25
26 ## Including Plots
```

2:1 GOOD MORNING LOL

R Markdown

Console Terminal x Render x Background Jobs x

Environment History Connections Tutorial








  Import Dataset 198 MiB 


R Global Environment

List

Environment is empty

Files Plots Packages Help Viewer Presentation

 New Folder  New Blank File  Upload  Delete  Rename  More Cloud > project

	Name	Size	Modified
	..		
<input type="checkbox"/>	.Rhistory	0 B	Oct 8, 2023, 2:29 PM
<input type="checkbox"/>	project.Rproj	205 B	Oct 8, 2023, 3:14 PM
<input type="checkbox"/>	test_rmd.Rmd	850 B	Oct 8, 2023, 3:14 PM
<input type="checkbox"/>	test_rmd.html	631.3 KB	Oct 8, 2023, 3:15 PM

# Some very important advice

R Markdown documents are like *recipes*.

Every step needs to be written down.

When you press the knit button, R forgets everything and follows the instructions line-by-line.

So be thorough, and write down everything in the order you want it to happen!

(One exception: NEVER use `install.packages()` in a script)

**Let's write some RMarkdown!**

# Basic data types

# Basic data types

There are five basic data types in R:

Type	Description	Examples
integer	Whole numbers	1, 2, 3
numeric	Any real number, fractions	3.4, 2, -2.3
character	Text	"Hi there", "8.5", "ABC123"
logical	Assertion of truth/falsity	TRUE, FALSE
complex	Real and imaginary numbers	0.34+5.3i

There are some additional types to be aware of, particularly *factors*, but we'll come back to them in a later session.



# Checking data types

We can use the **class()** function to check what type a given object is.

```
class(10)
```

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

```
class(10L) # using L after the number turns it into an *integer*
```

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

```
class(TRUE)
```

```
## [1] "logical"
```

```
class("Wednesday")
```

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

# Basic containers



# Vectors

A vector is a collection of values which all have the same basic **type**.

A numeric vector is thus a collection of numeric values:

```
some_numbers <- c(5, 3, 6, 8)
some_numbers
```

```
## [1] 5 3 6 8
```

... and a character vector is a collection of character values

```
char_example <- c("Monday", "Tuesday", "Wednesday", "Thursday")
char_example
```

```
## [1] "Monday"      "Tuesday"     "Wednesday"   "Thursday"
```

# More about vectors

The colon (:) operator can be used to produce a sequence of numbers:

```
one_to_ten <- 1:10  
one_to_ten
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

Vectors can also be given names:

```
one_to_four <- 1:4  
names(one_to_four) <- char_example  
one_to_four
```

```
## Monday Tuesday Wednesday Thursday  
## 1 2 3 4
```

# Extracting values

Sometimes you only want a specific subset of a vector. For example, suppose that you only want the third value. For this, we need the `[]` (square brackets) operator.

We put an *index* inbetween the `[]` operator.

```
char_example[3]
```

```
## [1] "Wednesday"
```

Note that you can also supply *multiple* values:

```
char_example[2:3]
```

```
## [1] "Tuesday" "Wednesday"
```

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

```
## [1] "Tuesday" "Thursday"
```

# Extracting values

If your vector is *named*, you can also use the names as *indices*.

```
one_to_four
```

```
##      Monday   Tuesday Wednesday  Thursday  
##           1         2         3         4
```

```
one_to_four["Wednesday"]
```

```
## Wednesday  
##          3
```

```
one_to_four[c("Monday", "Wednesday")]
```

```
##      Monday Wednesday  
##           1         3
```

# Matrices





# Matrices

Matrices are 2-dimensional collections of values.

All values must be of the same type.

```
matrix(1:9, nrow = 3, ncol = 3)
```

```
##      [,1] [,2] [,3]  
## [1,]    1    4    7  
## [2,]    2    5    8  
## [3,]    3    6    9
```

This is quite a common format. For example, each row could represent an individual participant, while each column could represent a different numerical measure.

# Accessing matrices

Since matrices are two-dimensional, you need to give two indices to make sure you get the value you want. Again, you can use the `[]` operator.

```
[row, col]
```

Here I created a 3 x 3 matrix and then extracted the number from the 2nd row down, 3rd column across.

```
test_matrix <- matrix(1:9, nrow = 3, ncol = 3)
test_matrix
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

```
test_matrix[2, 3]
```

```
## [1] 8
```

# Lists



# Lists

Lists are a collection of objects of varying length and type.

```
album_list <-  
  list(The_Beatles = c(  
    "Sgt. Pepper",  
    "The White Album",  
    "Revolver",  
    "Abbey Road"),  
    Nirvana = c(  
      "Bleach",  
      "Nevermind",  
      "In Utero")  
  )
```

Each element is labelled, just like a mason jar on a shelf.

Each element has different contents, just like our mason jars.

# Lists

```
names(album_list)
```

```
## [1] "The_Beatles" "Nirvana"
```

```
length(album_list)
```

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

```
album_list["The_Beatles"]
```

```
## $The_Beatles
```

```
## [1] "Sgt. Pepper"      "The White Album" "Revolver"        "Abbey Road"
```






# Tabular data

*Tabular* data is also a collection of different types of data, arranged in a rectangular, tabular format. Most of the data you encounter in psychology is in this kind of format.

In tabular data, each column contains only values of one *type*, and each row thus contains different types of information about one thing.

Show  entries

Search:

	mpg 	cyl 	disp 	hp 	drat 
Mazda RX4	21	6	160	110	3.9
Mazda RX4 Wag	21	6	160	110	3.9
Datsun 710	22.8	4	108	93	3.85
Hornet 4 Drive	21.4	6	258	110	3.08
Hornet Sportabout	18.7	8	360	175	3.15

Showing 1 to 5 of 32 entries

Previous

1

2

3

4

5

6

7

Next

Spaces

Your Workspace

PSY9219M; Research Method

+ New Space

Learn

Guide

Primers

DataCamp Courses

Cheat Sheets

Feedback and Questions

Info

Terms and Conditions

System Status

File Edit Code View Plots Session Build Debug Profile Tools Help



Go to file/function

Addins

R 3.5.0

FearofCrime x

Filter

	ResponseID	ResponseSet	Name	ExternalDataReference	Status
1	R_ai4tgG1GHNdVdqt	Default Response Set	Anonymous	NA	0
2	R_d5OiATV0IiBbMx	Default Response Set	Anonymous	NA	0
3	R_aaBVZUe9mIGiDpH	Default Response Set	Anonymous	NA	0
4	R_6nxlnLKQv2bucQZ	Default Response Set	Anonymous	NA	0
5	R_6SCYbhOP9BG5CgR	Default Response Set	Anonymous	NA	0
6	R_5pCxWA6qOQdnVyd	Default Response Set	Anonymous	NA	0
7	R_d1ni6V75Cnn0v	Default Response Set	Anonymous	NA	0

Showing 1 to 8 of 301 entries

Console

Terminal x

Jobs x

/cloud/project/

```
> library(readr)
> FearofCrime <- read_csv("http://www.research.lancs.ac.uk/portal/files/104824495/FearofCrime.csv")
Parsed with column specification:
cols(
  .default = col_integer(),
  ResponseID = col_character(),
  ResponseSet = col_character(),
  Name = col_character(),
  ExternalDataReference = col_character(),
  StartDate = col_character(),
  EndDate = col_character(),
  hexaco_First_Click = col_double(),
  hexaco_Last_Click = col_double(),
  hexaco_Page_Submit = col_double(),
```

Environment

History

Connections



Import Dataset

List

Global Environment

Data

FearofCrime 301 obs. of 169 variables

Files

Plots

Packages

Help

Viewer



New Folder

Upload

Delete

Rename

More

Cloud &gt; project

	Name	Size	Modified
<input type="checkbox"/>	..		
<input type="checkbox"/>	.Rhistory	0 B	Oct 21, 2018, 10:47 F
<input type="checkbox"/>	data		
<input type="checkbox"/>	project.Rproj	205 B	Oct 22, 2018, 10:01 A
<input type="checkbox"/>	scripts		
<input type="checkbox"/>	solved		

# Creating tabular data

In R, this type of structure is called a *data frame*.

```
days_of_the_week <-  
  data.frame(day_name = c("Sunday",  
                           "Monday",  
                           "Tuesday",  
                           "Wednesday",  
                           "Thursday",  
                           "Friday",  
                           "Saturday"),  
             day_number = 1:7  
             )
```

days\_of\_the\_week

##	day_name	day_number
## 1	Sunday	1
## 2	Monday	2
## 3	Tuesday	3
## 4	Wednesday	4
## 5	Thursday	5
## 6	Friday	6
## 7	Saturday	7



# Extracting information from data frames

You can use the `[]` operator to extract single elements, rows, or columns:

```
days_of_the_week[1, 2]
```

```
## [1] 1
```

```
days_of_the_week[5, ]
```

```
##   day_name day_number  
## 5 Thursday         5
```

```
days_of_the_week[, 1]
```

```
## [1] "Sunday"    "Monday"    "Tuesday"   "Wednesday" "Thursday"  "Friday"  
## [7] "Saturday"
```

# Extracting information from data frames

A special operator you can use for data frame columns is the dollar sign, \$

Combine the data frame's name with the column name as below:

```
days_of_the_week$day_name
```

```
## [1] "Sunday"      "Monday"      "Tuesday"     "Wednesday"  "Thursday"   "Friday"
## [7] "Saturday"
```

Question: what **class()** is this?

Wrapping up

# This week's concepts

- R Markdown - Chapter 27 of R4DS - see also <https://rmarkdown.rstudio.com>
- **vectors** and **lists** in Chapter 20 of R4DS

# Prep for next week

- Next week we'll talk again about data frames and consider how to *structure* data.
- Look at Section 2 (Wrangle) of R4DS for information on **tibbles** (which are essentially data frames...).