INTRODUCTION TO R

R: The true basics

What is R?

- Language for Statistical Computing
- Ihaka & Gentleman
- Auckland, New Zealand
- Open-source implementation of S
- Statistical Techniques
- Visualization Capabilities
- Highly extensible

Advantages

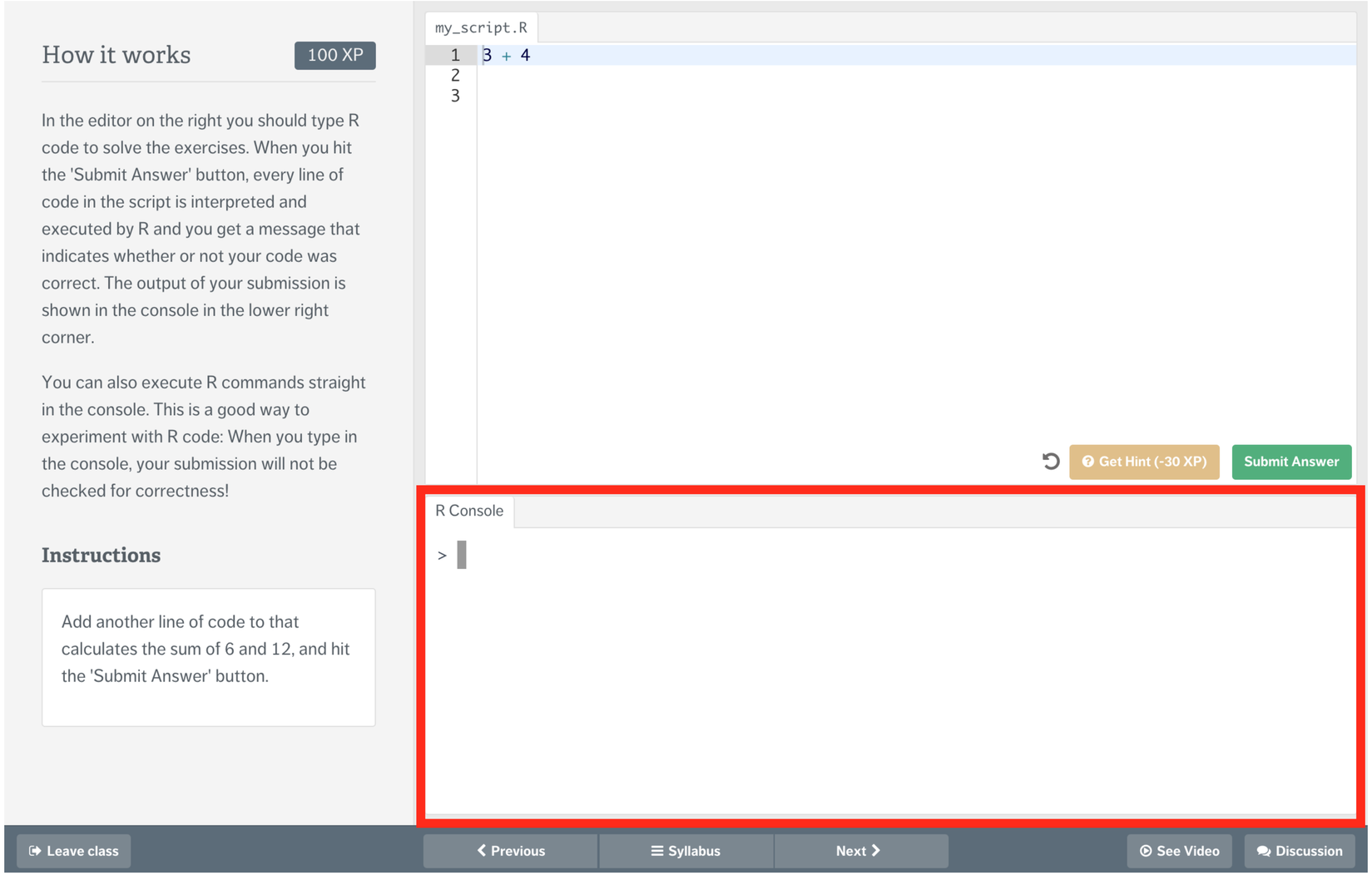
- Open source! free!
- Master at graphics
- Command-line interface
- Reproducibility through R scripts
- R packages: extensions of R
- Vibrant community

Disadvantages

- Easy to learn, hard to master
- Command-line interface daunting at first
- Poorly written code hard to read/maintain
- Poorly written code is slow

RDocumentation.org

Console


How it works
100 XP
In the editor on the right you should type R code to solve the exercises. When you hit the 'Submit Answer' button, every line of code in the script is interpreted and executed by R and you get a message that indicates whether or not your code was correct. The output of your submission is shown in the console in the lower right corner.
You can also execute R commands straight in the console. This is a good way to experiment with R code: When you type in the console, your submission will not be checked for correctness!
Instructions
Add another line of code to that calculates the sum of 6 and 12, and hit the 'Submit Answer' button.
my_script.R
1 3 + 4
2
3
Get Hint (-30 XP)
Submit Answer
R Console
>
Leave class
Previous
Syllabus
Next
See Video
Discussion

Console

```
> 1 + 2
[1] 3

> "Hi there, console!"
[1] "Hi there, console!"

> 2
[1] 2
```

Variables

- Store a variable to reuse later
- <-

```
> height <- 2

> height
[1] 2

> width <- 4

> width
[1] 4
```

Workspace

```
> ls()
[1] "height" "width"

> depth
Error: object 'depth' not found

> height * width
[1] 8

> area <- height * width
> area
[1] 8

> ls()
[1] "area"   "height" "width"
```

R script

- Text file with R commands
- Automate your work

rectangle.R

```
height <- 2
width <- 4
area <- height * width
area
```

Submit Answer

R script

rectangle.R

```
height <- 2
width <- 4
area <- height * width
area
```

```
> height <- 2

> width <- 4

> area <- height * width

> area
[1] 8
```

R script

rectangle.R

```
height <- 3
width <- 6
area <- height * width
area
```

```
> height <- 3

> width <- 6

> area <- height * width

> area
[1] 18
```

Comments

rectangle.R

```
# Create variables height and width
height <- 3
width <- 6

# Calculate the area
area <- height * width

# Print the area
area

# x <- 3    not executed!
```

Workspace (2)

```
> ls()
[1] "area"   "height" "width"

> rm(area)

> ls()
[1] "height" "width"

> area
Error: object 'area' not found
```

INTRODUCTION TO R

Basic Data Types

logical

`class()` **to reveal type**

```
> TRUE
[1] TRUE
> class(TRUE)
[1] "logical"

> FALSE
[1] FALSE

> class(NA)
[1] "logical"

> T
[1] TRUE
> F
[1] FALSE
```

numeric

```
> 2
[1] 2

> 2.5
[1] 2.5

> 2L
[1] 2

> class(2)
[1] "numeric"

> class(2L)
[1] "integer"
```

numeric

```
> is.numeric(2)
[1] TRUE

> is.numeric(2L)
[1] TRUE

> is.integer(2)
[1] FALSE

> is.integer(2L)
[1] TRUE
```

`integer` **is** `numeric`
`numeric` **not always** `integer`

character

```
> "I love data science!"
[1] "I love data science!"

> class("I love data science!")
[1] "character"
```

Other atomic types

- `double`: higher precision
- `complex`: complex numbers
- `raw`: store raw bytes

Coercion

```
> as.numeric(TRUE)
[1] 1
> as.numeric(FALSE)
[1] 0
> as.character(4)
[1] "4"
> as.numeric("4.5")
[1] 4.5

> as.integer("4.5")
[1] 4
> as.numeric("Hello")
[1] NA
Warning message:
NAs introduced by coercion
```