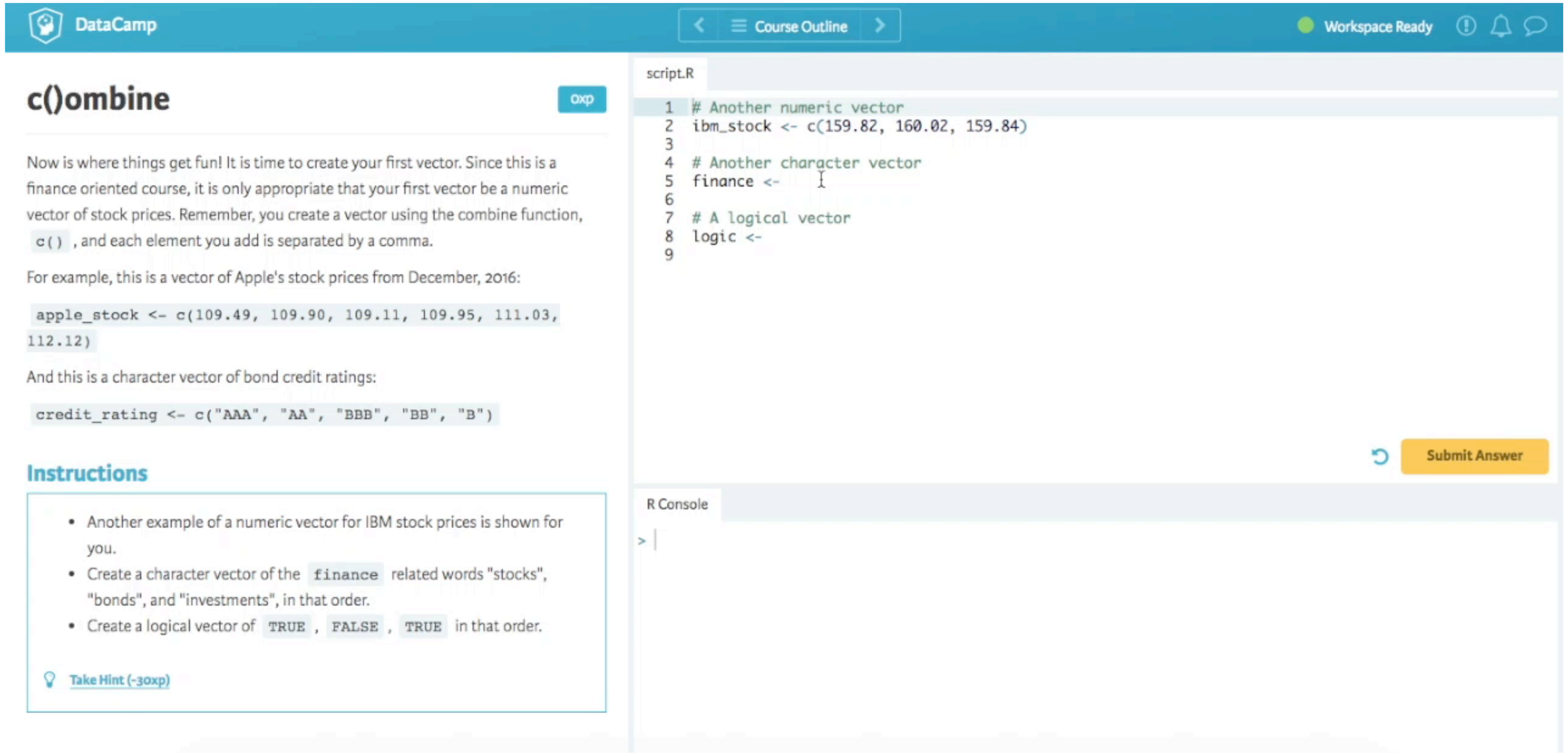
Welcome to Introduction to R for Finance!

Lore Dirick
Instructor, DataCamp

A Hands-On Course



c()ombine

Now is where things get fun! It is time to create your first vector. Since this is a finance oriented course, it is only appropriate that your first vector be a numeric vector of stock prices. Remember, you create a vector using the combine function, `c()`, and each element you add is separated by a comma.

For example, this is a vector of Apple's stock prices from December, 2016:

`apple_stock <- c(109.49, 109.90, 109.11, 109.95, 111.03, 112.12)`

And this is a character vector of bond credit ratings:

`credit_rating <- c("AAA", "AA", "BBB", "BB", "B")`

Instructions

- Another example of a numeric vector for IBM stock prices is shown for you.
- Create a character vector of the `finance` related words "stocks", "bonds", and "investments", in that order.
- Create a logical vector of `TRUE`, `FALSE`, `TRUE` in that order.

```r
# Another numeric vector
ibm_stock <- c(159.82, 160.02, 159.84)

# Another character vector
finance <-

# A logical vector
logic <-

```

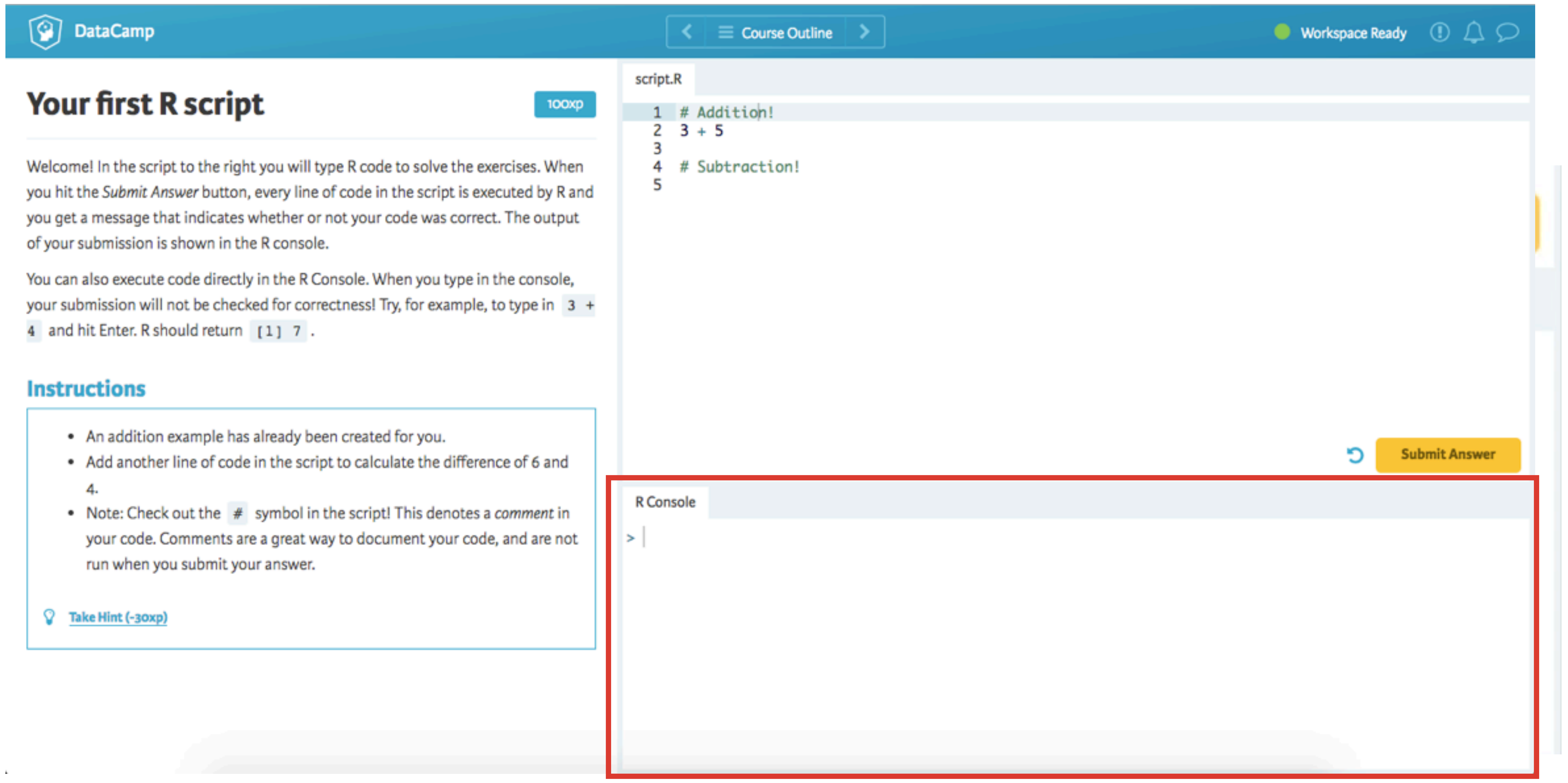
```
>
```

What will you learn?

- Basics of R
- Data structures
- Finance examples

Console

Execute R commands



Your first R script

Welcome! In the script to the right you will type R code to solve the exercises. When you hit the *Submit Answer* button, every line of code in the script is 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 R console.

You can also execute code directly in the R Console. When you type in the console, your submission will not be checked for correctness! Try, for example, to type in `3 + 4` and hit Enter. R should return `[1] 7`.

Instructions

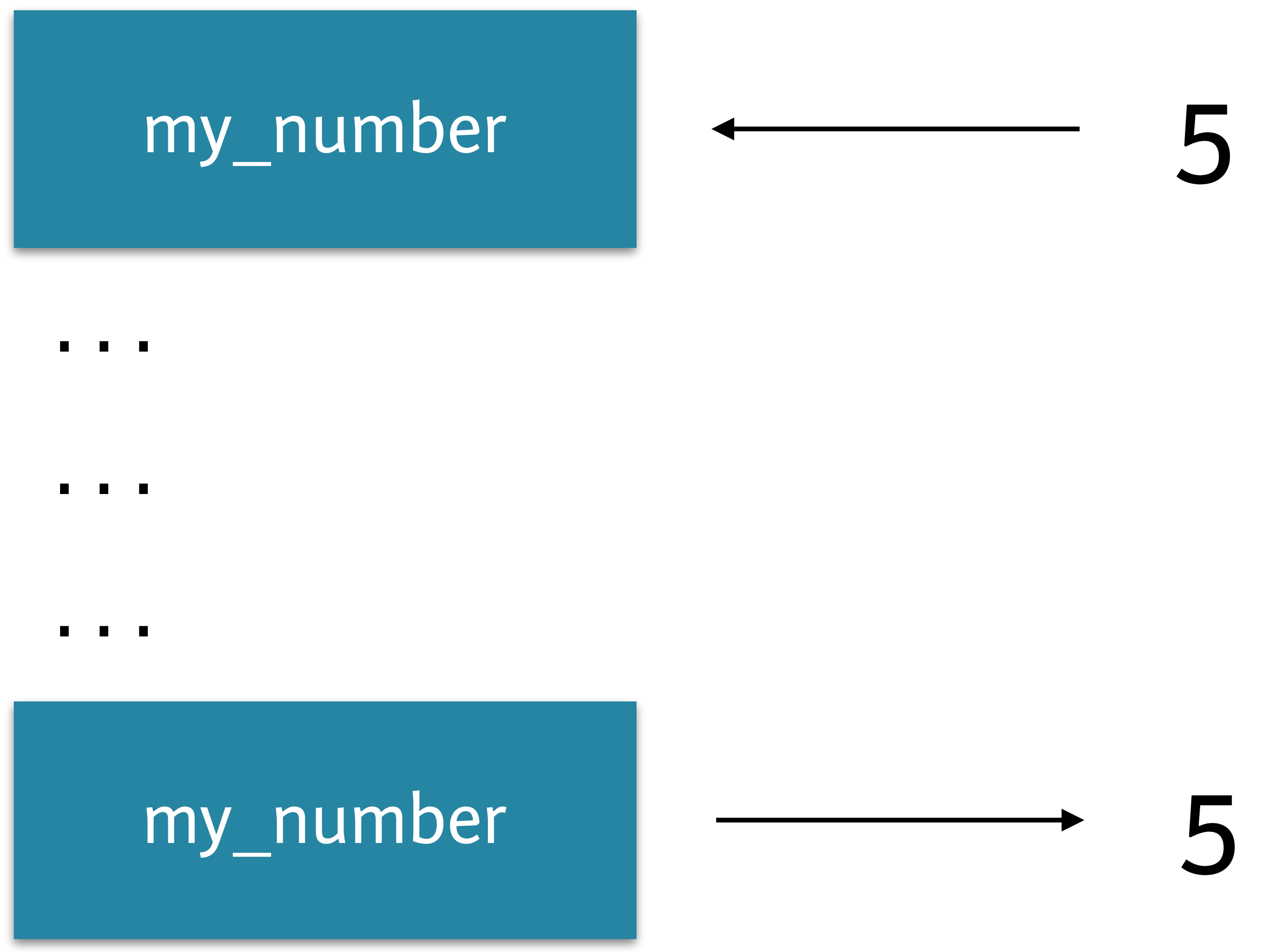
- An addition example has already been created for you.
- Add another line of code in the script to calculate the difference of 6 and 4.
- Note: Check out the `#` symbol in the script! This denotes a *comment* in your code. Comments are a great way to document your code, and are not run when you submit your answer.

```
# Addition!
3 + 5

# Subtraction!
```

```
>
```

Variables or objects `<-`



```
> my_number <- 5
> my_number
[1] 5
```

Arithmetic in R

```
> dan <- 100

> rob <- 50

> dan + rob
[1] 150

> total <- dan + rob

> total
[1] 150
```

R Scripts

script.R

```
dan <- 100
rob <- 50
total <- dan + rob
total
```

```
> dan <- 100

> rob <- 50

> total <- dan + rob

> total
[1] 150
```

Let’s practice!

Financial returns

Stock returns

- $50 worth of Apple stock
- 10% return in January

How much money do you have at the end of the month?

5 = 10% of 50

55 = 50 + 5

Stock returns

110% = 100% + 10%

$$\underbrace{1.10}_{\text{Return Multiplier}} = 1 + .10$$

55 = 50 * 1.10

```
> # Return Multiplier
> mult <- 1 + interest_rate / 100

> # New Amount
> new_cash <- starting_cash * mult
```

Stock returns - multiple periods

- $50 worth of Apple stock
- 10% return in January
- 5% return in February

57.75 = 55 * 1.05

57.75 = 50 * 1.10 * 1.05

```
> new_cash <- starting_cash * jan_mult * feb_mult
```

Let’s practice!

INTRODUCTION TO R FOR FINANCE

Basic data types

Numeric

```
> 42.5
[1] 42.5

> 5
[1] 5

> 5L
[1] 5
```

Character

```
> "Hello world"
[1] "Hello world"

> "forty"
[1] "forty"

> "5"
[1] "5"
```

Logical

```
> TRUE
[1] TRUE

> FALSE
[1] FALSE

> true
Error: object 'true' not found

> NA
[1] NA
```

Variables and data types

```
> my_answer <- TRUE
> my_answer
[1] TRUE

> food <- "carrots"
> food
[1] "carrots"
```

class()

```
> my_answer <- TRUE

> class(my_answer)
[1] "logical"

> class(5)
[1] "numeric"

> class(5L)
[1] "integer"
```

Let’s practice!