Variables

CSC121
Mark Kazakevich

Last time

- We saw how the R Console works and wrote some expressions
- Learned about some binary and unary operators
- Saw some different numeric data types
    - ‘double’ and ‘integer’
- Introduced variables

Today

- Go a little deeper into variables
- See some ways we can work with variables in the console
- Introduce Functions
  - What they are
  - How to use them

Review: Why we need variables

- We need a way to easily store and access the data we work with

- We want to be able to label data
  - Make it human readable (a theme of programming languages)

- We want to keep track of how data changes

Variable Assignment Statement

- We need a way to create a variable, name it, and assign some data to it

- We do that with an **assignment statement:**

```
variableName <- expression
```

We say:

“variableName is assigned the value expression”

`variableName <- expression`

“`variableName` is assigned the value of `expression`”

- On the left side, you have the name of the variable you want to assign
- On the right side, you have the expression (the value) you want to assign to that variable

`<-` is the assignment operator

- It looks like an **arrow**, and you can think of it like that:
  - The expression is put **into** the variable in the direction of the arrow

Two steps to assign a variable

```
variableName <- expression
```

Step 1: Evaluate the expression on the right-hand side of the statement to produce a value

Step2: Assign the value of that expression to the variable name on the left-hand side of the statement

variableName <- expression

```
p <- 5
```

Step 1: Evaluate expression on right-hand side to get a value

5

Step 2: Assign that value to variable name on left-hand side

p is assigned the value 5

R Console:

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

`variableName <- expression`

```
q <- 6 * 3
```

Step 1: Evaluate expression on right-hand side to get a value

6 * 3 = 18

Step 2: Assign that value to variable name on left-hand side

q is assigned the value 18

R Console:

```
> q <- 6 * 3
> q
[1] 18
```

```
p <- 5
t <- p + 3
```

We saw how we assigned p before, so let's focus on t

Assignment steps for variable t

Step 1: Evaluate expression on right-hand side to get a value

p + 3 = 5 + 3 = 8

Step 2: Assign that value to variable name on left-hand side

t is assigned the value 8

R Console:

```
> p <- 5
> t <- p + 3
> t
[1] 8
```

```
b <- as.integer(7.76)
```

Assignment steps for variable b

Step 1: Evaluate expression on right-hand side to get a value

as.integer(7.76) = 7

Step 2: Assign that value to variable name on left-hand side

b is assigned the value 7

R Console:

```
> b <- as.integer(7.76)
> b
[1] 7
```

Variable data type

- The data type of the variable is the data type of its value

- Remember that we can use the `typeof()` command to find the type

```
> p <- 5
> typeof(p)
[1] “double”
```

```
> b <- as.integer(7.76)
> typeof(b)
[1] “integer”
```

Let’s look at how this all works in the R Console