

What we have covered so far

- Vectors
- Functions

What we will cover today

- Scripts
- Logicals
- Vector extraction using logicals
- Data Frames (if we have the time)

Useful Functions

- `seq(start, end, skip)`: generate regular sequences
- `sample(n)` : generate a permutation of integers $[1, \dots, n]$
- `sample(n, m)`: generate m integers in the range $[1, \dots, n]$
- `1:10` is a shortcut for `seq(1, 10)`
- `rnorm(n)` : generate a vector of “ n ” numbers that satisfy a normal distribution of mean 0 and standard deviation 1
- `mean(x)` : compute the mean value of x
- `sd(x)` : compute the standard deviation of x
- `var(x)` : compute the variance of x

Scripts

- As our knowledge of R increases,
- We type longer and longer sequences
- We do not wish to type the same instructions over and over again
-
- Solution: script files

What is a Script File

- A “script file” is simply a text file
- This file is created with any editor
 - Notepad, edit, vi, Microsoft Word, etc.
- Use a special function in R to read this file
 - The function name is “source(filename)”

Example of a script file

- Step 1: create a file with the following commands

```
# this is an example script file
heights = c(5.6, 6.2, 5.)
higher = heights[2]
cat("higher = ", higher, "\n")    # "\n" is a carriage return
```

- Step 2: name this file: "heights.r"
- Step 3: in RStudio, use "setwd()" to change to the directory that contains the script file "heights.r"
- Step 4: execute the command:
 > source("heights.r")
- Result: the commands in the file are executed

Script files

- You will use script files when doing your assignments to simplify your task
- Easier to keep a record of what you are doing
- Can cut and past from one script file to another when you need R code that only changes slightly from code you already have written

Logical Variables

- Logical variables take two possible values:
 - TRUE and FALSE
 - T and F
- Arithmetic with logical variables
 - $T + T$ returns 2
 - $T + F$ returns 1
 - $3 * T$ returns 3
- Examples of use

Logical Variables

- `3 > 5`
 - Is this true or false? What does R return?
- `5 > 2 && 7 > 2`
 - `&&` means “and”
 - `5 > 2` is TRUE
 - `7 > 2` is TRUE
 - `TRUE && TRUE` is TRUE

Data Frames

- A vector is a collection of elements of the *same type*
- A data frame is “essentially” a collection of columns
 - Each column is a vector
 - Elements in each column are of the same type
 - Different columns can have different types
- Create a data frame from several vectors
-

Data Frames

- `heights = c(5.7, 6.2, 5.1, 5.5)`
`names = c("John", "James", "Sophie", "Christine")`
`table = data.frame(names, heights)`
`table = data.frame(heights, names)`
`table = data.frame(nms=names, h=heights)`
- (try the above out and discuss)
- One of the columns has numeric elements
- The other column string elements

More complex data.frame

- ```
heights = c(5.7, 6.2, 5.1, 5.5)
names = c("John", "James", "Sophie", "Christine")
table = data.frame(names, heights, heights > 5)
```
- The third column is a vector of logical elements

# Logical Operators

- `&&` is the “and” operator
  - `&&` compares two logical expressions
  - The result is only TRUE if both expressions are TRUE
- `||` is the “or” operator
  - `||` compares two logical expressions
  - The results is only FALSE if both expressions are FALSE

# Examples

is.higher = 5 > 2

is.heavier = 170 > 150

one.or.the.other = is.higher || is.heavier

one.and.the.other = is.higher && is.heavier

no.is.higher = !is.higher

# “!” returns opposite (negation) of logical  
expression

# Data Frames

heights = (5.6, 6.2, 5.8, 5.9)  
and the logical expression

`if.heights = heights > 5.8`

- `heights[c(T,T,F,F)]` returns `c(5.6, 6.2)`
- `heights[c(if.heights)]` returns `c(6.2, 5.9)`
- The logical vector is used to select which vector elements to extract.

# Logical Vectors

- We know that a number is really a vector with a single element
- Similarly, a logical values (T or F) is a vector with a single element
- Therefore, one can manipulate vectors such as

`c(T,T,F,F,T,F,T)`

whose elements all have the same type



# Logical Vectors

- `5 > 8` is a logical statement (the value is TRUE or FALSE)
- The vector `c(T,F,T)` is identical to  
`c(8 > 5, -1 > 0, 55 > 50)`  
(each element is either TRUE or FALSE)
- Consider the vector:  
`heights = (5.6, 6.2, 5.8, 5.9)`  
and the logical expression
- `Heights > 5.8`  
What does R return?

# Logical Vectors

- `5 > 8` is a logical statement (the value is TRUE or FALSE)
- The vector `c(T,F,T)` is identical to  
`c(8 > 5, -1 > 0, 55 > 50)`  
(each element is either TRUE or FALSE)
- Consider the vector:  
`heights = (5.6, 6.2, 5.8, 5.9)`  
and the logical expression
- `Heights > 5.8`  
What does R return?

# Logical Vectors

heights = (5.6, 6.2, 5.8, 5.9)  
and the logical expression

Heights > 5.8

R returns an error!!! Why?

# Logical Vectors

heights = (5.6, 6.2, 5.8, 5.9)  
and the logical expression

`heights > 5.8`

R returns

`F, T, F, T`

There are 4 elements in heights, and 4 elements in the logical vector.

We could have also written:

- `if.heights = heights > 5.8`

in which case, `if.heights` contains the vector `c(F,T,F,T)`

# Logical Vectors

heights = (5.6, 6.2, 5.8, 5.9)  
and the logical expression

`heights > 5.8`

R returns

`F, T, F, T`

There are 4 elements in heights, and 4 elements in the logical vector.

We could have also written:

- `if.heights = heights > 5.8`

in which case, `if.heights` contains the vector `c(F,T,F,T)`

# Extraction with logical vectors

heights = (5.6, 6.2, 5.8, 5.9)  
and the logical expression

`if.heights = heights > 5.8`

- `heights(c(T,T,F,F))` returns `c(5.6, 6.2)`
- `heights(c(if.heights))` returns `c(6.2, 5.9)`
- The logical vector is used to select which vector elements to extract.