

if-else conditions

Davinder Singh

2023-03-23

logical and conditional statements

These are pieces of code that return the 'TRUE' or 'FALSE' values, that is, a logical value

The common operators of logical statements are: -equality '==' -inequality '!=' -greater than '>' -less than '<' -greater or equal than '>=' -less or equal than '<='

The conditional statements allow to test several logical conditions at a time. The condition operators (or symbols) are -AND '&' (inside dplyr function we can represent AND using a ',') -OR '|'

We also have logical functions that test if something is 'TRUE' or 'FALSE', for example: -'is.na()' is a function that tests if a value is an 'NA' -This function is part of a whole family of functions, they also start with 'is.': -is.vector() -is.data.frame() -is.factor()

For next class: how to get all functions from a family (method).

-which() : takes logical vectors, it will give you the numerical index (position) of all values that are TRUE

```
which(letters == "r")
```

```
## [1] 18
```

```
letters[18]
```

```
## [1] "r"
```

exercise 6

```
w <- 10.2
x <- 1.3
y <- 2.8
z <- 17.5
colors <- c("red", "blue", "green")
masses <- c(45.2, 36.1, 27.8, 81.6, 42.4)
dna1 <- "attattaggaccaca"
dna2 <- "attattaggacacaca"
```

```
w > 10
```

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

```
colors == "green"
```

```
## [1] FALSE FALSE TRUE
```

```
x > y
```

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

```
masses > 40
```

```
## [1] TRUE FALSE FALSE TRUE TRUE
```

```
2^x + 0.2 == y
```

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

```
dna1 == dna2
```

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

```
dna1 != dna2
```

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

```
w > x | y > z
```

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

```
x * w < 13.5
```

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

```
x * w > 13.2
```

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

```
x * w < 13.5 & x * w > 13.2
```

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

```
masses > 30 & 50
```

```
## [1] TRUE TRUE FALSE TRUE TRUE
```

```
masses > 30
```

```
## [1] TRUE TRUE FALSE TRUE TRUE
```

```
masses < 50
```

```
## [1] TRUE TRUE TRUE FALSE TRUE
```

```
masses < 50 & masses > 30
```

```
## [1] TRUE TRUE FALSE FALSE TRUE
```

How to make simple choices with ‘if()’

```
if(condition is TRUE) {  
  Run all lines of code in this block of code  
}
```

If the condition is not TRUE, then nothing happens.

Exerciser 7: Handling one choice

```
age_class = "sapling"  
if (age_class == "sapling"){  
y <- 10  
}  
y
```

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

Case when we have two options: if-else structure

The general form of this structure:

```
if(condition){  
code that runs if condition is met  
} else{  
code that runs if condition is NOT met  
}
```

Exerciser 8 handling two choices

```

age_class = "seedling"
if (age_class == "sapling"){
y <- 10
} else{
  age_class == "seedling"
  y <- 5
}
y

```

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

handle more than 2 choices

In this case we are using the elseif structure:

```

if(condition1){
first block code that is executed if condition 1 is met
} else if(condition2){
second block code that executes if condition2 is met
} else if(condition3){
more code
} else{
this will cover all the conditions that are not specified before
}

```

you do not have to end up with an else block 'Else if' are more intentional with the conditions. A simple else will run in all other no matter what.

Exercise 9 handling 3 choices or more

```

age_class = "adult"
if (age_class == "sapling"){
y <- 10
} else if (age_class == "seedling"){
  y <- 5
} else{
  y <- 0
}
y

```

```
## [1] 0
```

Exercise 12 load or download file

```
list.files()
```

```
## [1] "acacia_by_treatment.jpg"      "acacia_by_treatment.pdf"
## [3] "acacia_vs_trees_homework.pdf"  "acacia_vs_trees_homework.Rmd"
## [5] "apply-prep.pdf"                "apply-prep.Rmd"
## [7] "choices-prep.pdf"              "choices-prep.Rmd"
## [9] "Davinder-life-cycle.pdf"       "Davinder-life-cycle.Rmd"
## [11] "functions-prep.pdf"            "functions-prep.Rmd"
## [13] "if-else-excerise.Rmd"          "joining-tables.pdf"
## [15] "joining-tables.Rmd"            "Joining-Vectors.pdf"
## [17] "Joining-Vectors.Rmd"           "shrub_volume.pdf"
## [19] "shrub_volume.Rmd"              "species.csv"
## [21] "surveys"                       "surveys (1).csv"
## [23] "surveys_test.csv"              "visualization-uhuru-day2.pdf"
## [25] "visualization-uhuru-day2.Rmd"  "visualization-uhuru.pdf"
## [27] "visualization-uhuru.Rmd"        "wrangling-pipes.pdf"
## [29] "wrangling-pipes.Rmd"           "wrangling-portal.pdf"
## [31] "wrangling-portal.Rmd"          "writing-functions.pdf"
## [33] "writing-functions.Rmd"
```

```
if (file.exists("surveys (1).csv")) {
  print("file exists")
} else{
  print("file does not exist")
}
```

```
## [1] "file exists"
```

?download.file

```
if (file.exists("surveys.csv")) {
  print("file exists")
} else{
  download.file("https://ndownloader.figshare.com/files/2292172", "surveys_test.csv")
  read.csv("surveys_test.csv") %>%
  head() %>%
  print()
}
```

```
##   record_id month day year plot_id species_id sex hindfoot_length weight
## 1         1     7  16 1977      2         NL   M             32      NA
## 2         2     7  16 1977      3         NL   M             33      NA
## 3         3     7  16 1977      2         DM   F             37      NA
## 4         4     7  16 1977      7         DM   M             36      NA
## 5         5     7  16 1977      3         DM   M             35      NA
## 6         6     7  16 1977      1         PF   M             14      NA
```

```
if (file.exists("species1.csv")) {
  print("file exists")
} else{
  download.file("https://ndownloader.figshare.com/files/3299483", "species.csv")
  read.csv("species.csv") %>%
  head() %>%
```

```
print()  
}
```

##	species_id	genus	species	taxa
## 1	AB	Amphispiza	bilineata	Bird
## 2	AH	Ammospermophilus	harrisi	Rodent
## 3	AS	Ammodramus	savannarum	Bird
## 4	BA	Baiomys	taylori	Rodent
## 5	CB	Campylorhynchus	brunneicapillus	Bird
## 6	CM	Calamospiza	melanocorys	Bird