if-else conditions

Anita Dhillon

2023-03-23

Logical and conditional statements

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

The common operators of logical statements are - equality ‘==’ - inequality ‘!=’ - greater than ‘>’ - less than ‘<’ - greater than or equal to ‘>=’ - less than or equal to ‘<=’

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 a part of a whole family of functions that all 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 numerical index (position) of all values that are TRUE

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

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

```
letters[18]
```

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

Exercise 6:Choice Operators

1. Create the following variables

```
 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 <- "attattaggaacaca"
```

2. Use them to print whether or not the following statements are TRUE or FALSE.

a) w is greater than 10

```r
w > 10
```

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

b) “green” is in colors

```r
("green" == colors)
```

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

c)x is greater than y

```r
x > y
```

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

d)Each value in masses is greater than 40.

```r
is.na(masses > 40)
```

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

e)2 * x + 0.2 is equal to y

```r
2 * x + 0.2 ==  y
```

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

f)dna1 is the same as dna2

```r
(dna1 == dna2)
```

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

g)dna1 is not the same as dna2

```r
dna1 != dna2
```

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

h)w is greater than x, or y is greater than z

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

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

i) x times w is between 13.2 and 13.5

```
x*w < 13.5
```

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

```
x*w > 13.2
```

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

```
# 13.2 < x*w < 13.5 : This is how we would write in on paper
# in R we have to compare things in pairs
x * w < 13.5 & x *w > 13.2
```

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

j) Each mass in masses is between 30 and 50.

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

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

How to make simple choices w ’ if()’

The general structure of an if statement :

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

If condition is not TRUE, then nothing happens

Exercise 7 Handing one choice

1. Copy the following code and complete the if statement so that if age_class is equal to “sapling” it sets y <- 10.

```
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 strucuteL

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

Exercise 8 : Handling 2 choices

Copy the following code and complete the if statement so that if age_class is equal to “sapling” it sets y <- 10 and if age_class is equal to “seedling” it sets y <- 5.

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

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

Exercise 9 : Handling two or more choices

In this case we are using an elseif structure:

```
if(condition1) {
first block code that executes if condition 1 is met
} else if (condition2) {
else if (condition 3) {
more code
} else {
this will cover all the conditions that are not specified before
}
```

1. Copy the following code and complete the if statement so that if age_class is equal to “sapling” it sets y <- 10 and if age_class is equal to “seedling” it sets y <- 5 and if age_class is something else then it sets the value of y <- 0.

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

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

Exercise 10 : value of y by age class

Convert your conditional statement from the last exercise into a function called get_y, that takes age_class as an argument and returns y.

Call the function you just created 5 times, once with each of the following values for age_class, and print the values of y:

Exercise 12

```r
list.files()
```

```
##  [1] "acacia_by_treatment.jpg"     "acacia_by_treatment.pdf"
##  [3] "choices-prep.pdf"            "choices-prep.Rmd"
##  [5] "DataLifeCycle.Rmd"           "functions-prep.pdf"
##  [7] "functions-prep.Rmd"          "Functions.html"
##  [9] "Functions.Rmd"               "FunctionsHW-3-21-.pdf"
## [11] "FunctionsHW(3:21).Rmd"       "HWACACIA3-1-23.html"
## [13] "HWACACIA3:1:23 copy.Rmd"     "HWACACIA3:1:23.Rmd"
## [15] "if-else-exercises.pdf"       "if-else-exercises.Rmd"
## [17] "joining-tables.pdf"          "joining-tables.Rmd"
## [19] "species.csv"                 "surveys_test.csv"
## [21] "visualization-uhuru_files"   "visualization-uhuru.html"
## [23] "visualization-uhuru.log"     "visualization-uhuru.Rmd"
## [25] "visualization-uhuru.tex"     "Visualization-uhuruday2.log"
## [27] "Visualization-uhuruday2.Rmd" "Visualization-uhuruday2.tex"
## [29] "wrangling-pipes.pdf"         "wrangling-pipes.Rmd"
## [31] "wrangling-portal.log"        "wrangling-portal.Rmd"
```

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

```
## [1] "file does not exist"
```

?download.file

```r
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
```