loops

Davinder Singh

2023-04-06

For loops are another way we can tell a computer to repeat task for us. They are versatile and very explicit, so that means that we are are controlling everything that is run on each iteration of the loops (mostly). As opposed to apply functions, where the iterations happen kind of under the hood, and the apply functions can only be used to loop over (iterate) on one function loops can let us iterate over multiple functions and whole blocks of code ## The general structure of a for loop

```
for (variable_used_inside_the_loops in object_with_values){
  do something with(variable_used_inside_the_loops)
}
```

```
lengths <- c(13.3, 15, 100)

for (length in lengths){
  mass <- 0.73 * length^2
  print(mass)
  #we can't use return() in for loops
}
```

```
## [1] 129.1297
## [1] 164.25
## [1] 7300
```

excerise 1 basic for loops

```
numbers <- c(1, 2, 3, 4, 5)
for (number in numbers){
  number <- numbers * 3
  print(number)
}
```

```
## [1]  3  6  9 12 15
## [1]  3  6  9 12 15
## [1]  3  6  9 12 15
## [1]  3  6  9 12 15
## [1]  3  6  9 12 15
```

```
mass_lbs <- c(2.2, 3.5, 9.6, 1.2)
for (mass in mass_lbs){
  mass_kg <- 2.2 * mass
  print(mass_kg)
}
```

```
## [1] 4.84
## [1] 7.7
## [1] 21.12
## [1] 2.64
```

looping over an index

what is an index in R? It is the numeric position of values inside any data structure in R. For example in the following vector

```
flowers <- c("lilacs", "daises", "jasmins")
str(flowers)
```

```
##  chr [1:3] "lilacs" "daises" "jasmins"
```

```
#to access the second element in the vector, I need to use the index number 2 inside the square bracket
flowers[2]
```

```
## [1] "daises"
```

We can use numbers as indicies to loop over values inside a vector

```
for (i in 1:3){
  print(i)
  print(flowers[i])
}
```

```
## [1] 1
## [1] "lilacs"
## [1] 2
## [1] "daises"
## [1] 3
## [1] "jasmins"
```

```
birds = c('robin', 'woodpecker', 'blue jay', 'sparrow')
for (i in 1:length(birds)){
  print(birds[i])
}
```

```
## [1] "robin"
## [1] "woodpecker"
## [1] "blue jay"
## [1] "sparrow"
```

Storing results from a for loop indices

so far we have just printed some values and results from some equations.

Usually what we need is to save results of running a for loop, so that we can use them later.

When we are using a function what we do we use to result of the function? we assign the result to a variable name

```
my_results <- 0.73 * lengths^2

# but in for loops we do not have that option:

#we can't do:
```

```
my_result <- for(varaible in vector){

}
```

The only way to save results from each iteration of the loop is by saving them into an empty object.

```
# To create an empty vector, we use the function vector()

my_results <- vector(mode = "character", length = length(flowers))

for (i in 1:length(flowers)){
  upper <- toupper(flowers[i])
  print(upper)
}
```

```
## [1] "LILACS"
## [1] "DAISES"
## [1] "JASMINS"
```

```
my_results <- vector(mode = "character", length = length(flowers))
my_results
```

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

Now we can use this empty vector and indices inside a loop to store results

```
for (i in 1:length(flowers)){
  upper <- toupper(flowers[i])
  my_results[i] <- upper
}
my_results
```

```
## [1] "LILACS"  "DAISES"  "JASMINS"
```

```
radius <- c(1.3, 2.1, 3.5)
areas <- vector(mode = "numeric", length = length(radius))
for (i in 1:length(radius)){
  areas[i] <- pi * radius[i] ^ 2
}
areas
```

```
## [1]  5.309292 13.854424 38.484510
```

looping over multiple objects with indicies

We have 3 vectors

```
dino_names <- c("T-rex", "Ankylosaur", "Triceratops")
# we have diffent a values for each of these dino species
a_values <- c(0.73, 5.4, 100)
b_values <- c(2, 0.5, 1.2)
dino_lengths <- c(15, 3, 20)
dino_masses <- vector(mode = "numeric", length = length(dino_names))
```

We can iterate through these values within a loop

```
for (i in 1:length(dino_names)){
  print(dino_names[i])
  mass <- a_values[i] * dino_lengths[i]^b_values[i]
  dino_masses[i] <- mass
}
```

```
## [1] "T-rex"
## [1] "Ankylosaur"
## [1] "Triceratops"
```

```
dino_masses
```

```
## [1]  164.250000    9.353074 3641.128406
```

```
lengths = c(1.1, 2.2, 1.6)
widths = c(3.5, 2.4, 2.8)
areas <- vector(mode = "numeric", length = length(widths))
for (i in 1:length(lengths)) {
  areas[i] <- lengths[i] * widths[i]
}
areas
```

```
## [1] 3.85 5.28 4.48
```

Excerise 6

```
read.csv(file = "../data-raw/dinosaur_lengths (1).csv") -> dinosaur_length
```

```
mass_from_length <- function(length, species){
  if (species == "Stegosauria"){
    a = 10.95
    b = 2.64
  } else if (species == "Theropoda"){
```

```
    a = 0.73
    b = 3.63

  } else if ( species == "Sauropoda"){
    a = 214.44
    b = 1.46

  } else {
              a = 25.37
              b = 2.49

  }
mass <- a * length^b
return(mass)
}
```

```
my_results <-vector(mode = "numeric", length = nrow(dinosaur_length))
for (i in 1:nrow(dinosaur_length)){
   mass_i <- mass_from_length(length = dinosaur_length$lengths[i], species = dinosaur_length$species[i])
  #print(mass_i)
  my_results[i] <- mass_i

}
head(my_results)
```

```
## [1] 24341.68 27017.90 67453.38 22114.19 53884.76 52026.34
```

```
dinosaur_length$masses <- my_results
head(dinosaur_length)
```

```
##        species  lengths   masses
## 1  Stegosauria 18.52588 24341.68
## 2 Ankylosauria 16.43598 27017.90
## 3 Ankylosauria 23.73421 67453.38
## 4    Sauropoda 23.93411 22114.19
## 5 Ankylosauria 21.68718 53884.76
## 6 Ankylosauria 21.38363 52026.34
```

```
dinosaur_length %>%
  group_by(species) %>%
  summarise(mean_mass = mean(masses))
```

```
## # A tibble: 4 x 2
##   species      mean_mass
##   <chr>            <dbl>
## 1 Ankylosauria    46819.
## 2 Sauropoda       16104.
## 3 Stegosauria     31924.
## 4 Theropoda       45572.
```

Excerise 7

```
download.file(url = "http://www.datacarpentry.org/semester-biology/data/individual_collar_data.zip", des
```

```
unzip("../data-raw/individual_collar_data.zip",exdir = "../data-raw/")
```

```
list.files(path = "../data-raw", pattern = "collar-data-.*.txt")
```

```
##  [1] "collar-data-A1-2016-02-26.txt"  "collar-data-B2-2016-02-26.txt"
##  [3] "collar-data-C3-2016-02-26.txt"  "collar-data-D4-2016-02-26.txt"
##  [5] "collar-data-E5-2016-02-26.txt"  "collar-data-F6-2016-02-26.txt"
##  [7] "collar-data-G7-2016-02-26.txt"  "collar-data-H8-2016-02-26.txt"
##  [9] "collar-data-I9-2016-02-26.txt"  "collar-data-J10-2016-02-26.txt"
```