Review

Luna L Sanchez Reyes

2022-11-29

Exercise 1: Load or download a file?

1. Write a conditional statement that checks if surveys.csv exists in the working directory, if it doesn't then:

a. downloads it from https://ndownloader.figshare.com/files/2292172 using download.file(), and finally
b. loads the file into a data frame and
c. displays the first few rows using the head() function. The url needs to be in quotes since it is character data.

```
survey.list <- list.files("../data-raw")

any(survey.list == "surveys.csv")
```

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

```
any(c(FALSE, FALSE, FALSE))
```

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

```
any(c(FALSE, TRUE))
```

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

```
is.element("surveys.csv", survey.list)
```

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

```
if (is.element("surveys.csv", survey.list)) {
  print("The file surveys.csv is already downladed")
} else if (!is.element("surveys.csv", survey.list)) {
  print("The file is not downloaded!")
  # a.  downloads it from https://ndownloader.figshare.com/files/2292172 using download.file(),
  # b. loads the file into a data frame and
  # c. displays the first few rows using the head() function.
}
```

```
## [1] "The file is not downloaded!"
```

```
help("download.file")
?download.file
download.file(url = "https://ndownloader.figshare.com/files/2292172",
              destfile = "../data-raw/surveys-download.csv")

?data.frame
data.frame(names = c("Azul", "Jai", "Marcos", "Luna"), color = c("green", "pink", "gray", "lavender"))
```

```
##    names    color
## 1   Azul    green
## 2    Jai     pink
## 3 Marcos     gray
## 4   Luna lavender
```

```
data.frame("../data-raw/surveys-download.csv")
```

```
##   X....data.raw.surveys.download.csv.
## 1    ../data-raw/surveys-download.csv
```

```
surveys_table <- read.csv("../data-raw/surveys-download.csv")

head(surveys_table)
```

```
##   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 (!is.element("surveys.csv", list.files("../data-raw/"))) {
  download.file(url = "https://ndownloader.figshare.com/files/2292172",
             destfile = "../data-raw/surveys-download.csv")

  surveys <- read.csv(file = "../data-raw/surveys-download.csv")

  head(surveys_table)
}
```

```
##   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
```

2. Make a version of this conditional statement that is a function, where the name of the file is the first argument and the link for downloading the file is the second argument.

The function should return the resulting data frame. Add some documentation to the top of the function describing what it does. Call this function using "species.csv" as the file name and https://ndownloader.figshare.com/files/3299483 as the link. Print the first few rows of the resulting data frame using head().

```
# This function tests if a file is in the data raw directory and if not it download it and read it as d

reading_csv <- function(file_name, file_link) {

  # 1. test if file_name is in the data-raw folder
  # file_name <- "species.csv"
  test <- !is.element(file_name, list.files(path = "../data-raw"))

  # 2. if test is FALSE, download the file

  if (test) {
    # Option 1: save it with a random name:
```

```
    # download.file(url = file_link, destfile = "../data-raw/temporary.csv")

    # result <- read.csv(file = "../data-raw/temporary.csv")

    # Option 2: save it with the name given in file name:
    destination_file <- stringr::str_c("../data-raw/", file_name)
    download.file(url = file_link, destfile = destination_file)

    result <- read.csv(file = destination_file)

  }

  # return(result)
}
```

```
reading_csv(file_name = "species.csv",
            file_link = "https://ndownloader.figshare.com/files/3299483")
```

Multi-file Analysis

You have a satellite collars on a number of different individuals and want to be able to quickly look at all of their recent movements at once. The data is posted daily to a URL, as a zip file, and contains one csv file for each individual: http://www.datacarpentry.org/semester-biology/data/individual_collar_data.zip.

Start your solution by:

1. If individual_collar_data.zip is not already in your working directory download the zip file using download.file()

```
?list.files
list_of_files <- list.files(path = "../data-raw/.")

file_name <- "individual_collar_data.zip"

file_present <- is.element(file_name, list_of_files)

if (!file_present) {
  download.file(url = "http://www.datacarpentry.org/semester-biology/data/individual_collar_data.zip",
}

library(stringr)
str_c("Azul", "Marcos", "Jai", "Emily Jane")
```

```
## [1] "AzulMarcosJaiEmily Jane"
```

```
str_c("../data-raw/", file_name)
```

```
## [1] "../data-raw/individual_collar_data.zip"
```

2. Unzip it using unzip()

```
?unzip
# dir.create(path = "../data-raw/individual_collar_data")
unzip("../data-raw/individual_collar_data.zip", exdir = "../data-raw/individual_collar_data")
```

3. Obtain a list of all of the files with file names matching the pattern "collar-data-.*.txt" (using list.files())

```
collar_data_files <- list.files(pattern = "collar-", path = "../data-raw/individual_collar_data")
collar_data_files
```

```
##  [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"
```

4. Use a loop to load each of these files into R and make a line plot (using geom_path()) for each file with long on the x axis and lat on the y axis. Graphs, like other types of output, won't display inside a loop unless you explicitly display them, so you need to put your ggplot() command inside a print() statement. Include the name of the file in the graph as the graph title using labs().

```
# load file:
read.csv(file = "../data-raw/individual_collar_data/collar-data-J10-2016-02-26.txt")
str_c("apple", "1", "seven")
str_c("../data-raw/", collar_data_files)

library(ggplot2)
for (i in collar_data_files) {
  print(i)
  collar_data_table <- read.csv(file = i)
  #print(head(collar_data_table))
  collar_data_graph <- ggplot(collar_data_table, aes(x = long, y = lat)) +
    geom_path() +
    labs(title = i)
  print(collar_data_graph)
}
```

5. Add code to the loop to calculate the minimum and maximum latitude in the file, and store these values, along with the name of the file, in a data frame. Show the data frame as output.

Solution 1, using empty vectors

```
# ?vector
# empty vectors
all_min <- vector()
all_max <- vector()
length(all_min)
```

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

```
for (i in collar_data_files) {
  print(i)
  file_name <- str_c("../data-raw/individual_collar_data/", i)
  collar_data_table <- read.csv(file = file_name)
  min_lat <- min(collar_data_table[, "lat"])
  # min_lat <- min(collar_data_table$lat)
  all_min <- c(all_min, min_lat)
  print(all_min)
  #max_lat <- max(collar_data_table$lat)
}
```

```
## [1] "collar-data-A1-2016-02-26.txt"
## [1] 25.2108
## [1] "collar-data-B2-2016-02-26.txt"
## [1] 25.21080 26.70509
## [1] "collar-data-C3-2016-02-26.txt"
## [1] 25.21080 26.70509 28.86998
## [1] "collar-data-D4-2016-02-26.txt"
## [1] 25.21080 26.70509 28.86998 21.34315
## [1] "collar-data-E5-2016-02-26.txt"
## [1] 25.21080 26.70509 28.86998 21.34315 21.85565
## [1] "collar-data-F6-2016-02-26.txt"
## [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788
## [1] "collar-data-G7-2016-02-26.txt"
## [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788 27.67120
## [1] "collar-data-H8-2016-02-26.txt"
## [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788 27.67120 19.70875
## [1] "collar-data-I9-2016-02-26.txt"
## [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788 27.67120 19.70875
## [9] 25.70252
## [1] "collar-data-J10-2016-02-26.txt"
##  [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788 27.67120 19.70875
##  [9] 25.70252 24.71200
```

Solution 2, using vectors of a predetermined length

```r
# create vectors of a predetermined length

all_min_lat <- vector(mode = "numeric", length = length(collar_data_files))
all_max_lat <- all_min_lat
# create a vector for the path file names

for (i in 1:length(collar_data_files)) {
  file_name_and_path <- str_c("../data-raw/individual_collar_data/", collar_data_files[i])
  print(file_name_and_path)
  collar_data_table <- read.csv(file = file_name_and_path)
  min_lat <- min(collar_data_table$lat)
  all_min_lat[i] <- min_lat
  print(all_min_lat)
}
```

```
## [1] "../data-raw/individual_collar_data/collar-data-A1-2016-02-26.txt"
##  [1] 25.2108  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
## [10]  0.0000
## [1] "../data-raw/individual_collar_data/collar-data-B2-2016-02-26.txt"
##  [1] 25.21080 26.70509  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000
##  [9]  0.00000  0.00000
## [1] "../data-raw/individual_collar_data/collar-data-C3-2016-02-26.txt"
##  [1] 25.21080 26.70509 28.86998  0.00000  0.00000  0.00000  0.00000  0.00000
##  [9]  0.00000  0.00000
## [1] "../data-raw/individual_collar_data/collar-data-D4-2016-02-26.txt"
##  [1] 25.21080 26.70509 28.86998 21.34315  0.00000  0.00000  0.00000  0.00000
##  [9]  0.00000  0.00000
## [1] "../data-raw/individual_collar_data/collar-data-E5-2016-02-26.txt"
##  [1] 25.21080 26.70509 28.86998 21.34315 21.85565  0.00000  0.00000  0.00000
##  [9]  0.00000  0.00000
## [1] "../data-raw/individual_collar_data/collar-data-F6-2016-02-26.txt"
##  [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788  0.00000  0.00000
##  [9]  0.00000  0.00000
## [1] "../data-raw/individual_collar_data/collar-data-G7-2016-02-26.txt"
##  [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788 27.67120  0.00000
##  [9]  0.00000  0.00000
## [1] "../data-raw/individual_collar_data/collar-data-H8-2016-02-26.txt"
##  [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788 27.67120 19.70875
##  [9]  0.00000  0.00000
## [1] "../data-raw/individual_collar_data/collar-data-I9-2016-02-26.txt"
##  [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788 27.67120 19.70875
##  [9] 25.70252  0.00000
## [1] "../data-raw/individual_collar_data/collar-data-J10-2016-02-26.txt"
##  [1] 25.21080 26.70509 28.86998 21.34315 21.85565 17.90788 27.67120 19.70875
##  [9] 25.70252 24.71200
```

```
all_min_lat[1] <- 0.10987563839
```