Importing data from flat files with utils

Mohamad Osman

2022-06-21

01-read.csv

- Use `read.csv()` to import `"swimming_pools.csv"` as a data frame with the name `pools`.
- Print the structure of `pools` using `str()`.

```
swimming_pools_path <- file.path("..","00_Datasets","swimming_pools.csv")
swimming_pools_path
```

```
## [1] "../00_Datasets/swimming_pools.csv"
```

```
# Import swimming_pools.csv: pools
pools <- read.csv(swimming_pools_path)

# Print the structure of pools
str(pools)
```

```
## ’data.frame’:    20 obs. of  4 variables:
##  $ Name     : chr  "Acacia Ridge Leisure Centre" "Bellbowrie Pool" "Carole Park" "Centenary Pool (in
##  $ Address  : chr  "1391 Beaudesert Road, Acacia Ridge" "Sugarwood Street, Bellbowrie" "Cnr Boundary
##  $ Latitude : num  -27.6 -27.6 -27.6 -27.5 -27.4 ...
##  $ Longitude: num  153 153 153 153 153 ...
```

02-stringsAsFactors

- Use `read.csv()` to import the data in `"swimming_pools.csv"` as a data frame called `pools`; make sure that strings are imported as characters, not as factors.
- Using `str()`, display the structure of the dataset and check that you indeed get character vectors instead of factors.

```
# Import swimming_pools.csv correctly: pools
pools <- read.csv(swimming_pools_path, stringsAsFactors = FALSE)

# Check the structure of pools
str(pools)
```

```
## ’data.frame’:    20 obs. of  4 variables:
##  $ Name     : chr  "Acacia Ridge Leisure Centre" "Bellbowrie Pool" "Carole Park" "Centenary Pool (in
##  $ Address  : chr  "1391 Beaudesert Road, Acacia Ridge" "Sugarwood Street, Bellbowrie" "Cnr Boundary
##  $ Latitude : num  -27.6 -27.6 -27.6 -27.5 -27.4 ...
##  $ Longitude: num  153 153 153 153 153 ...
```

03-read.delim

- Import the data in "hotdogs.txt" with read.delim(). Call the resulting data frame hotdogs. The variable names are **not** on the first line, so make sure to set the header argument appropriately.
- Call summary() on hotdogs. This will print out some summary statistics about all variables in the data frame.

```
hotdogs_path <- file.path("..","00_Datasets","hotdogs.txt")

# Import hotdogs.txt: hotdogs
hotdogs <- read.delim(hotdogs_path, sep = "\t", header = FALSE,
                      stringsAsFactors = FALSE)

# Summarize hotdogs
summary(hotdogs)
```

```
##       V1                  V2              V3
##  Length:54          Min.   : 86.0   Min.   :144.0
##  Class :character   1st Qu.:132.0   1st Qu.:362.5
##  Mode  :character   Median :145.0   Median :405.0
##                     Mean   :145.4   Mean   :424.8
##                     3rd Qu.:172.8   3rd Qu.:503.5
##                     Max.   :195.0   Max.   :645.0
```

04-read.table

- Finish the read.table() call that's been prepared for you. Use the path variable, and make sure to set sep correctly.
- Call head() on hotdogs; this will print the first 6 observations in the data frame.

```
# Path to the hotdogs.txt file: path
hotdogs_path <- file.path("..", "00_Datasets", "hotdogs.txt")

# Import the hotdogs.txt file: hotdogs
hotdogs <- read.table(hotdogs_path,
                      sep = '\t',
                      col.names = c("type", "calories", "sodium"))

# Call head() on hotdogs
head(hotdogs)
```

```
##   type calories sodium
## 1 Beef      186    495
## 2 Beef      181    477
## 3 Beef      176    425
## 4 Beef      149    322
## 5 Beef      184    482
## 6 Beef      190    587
```

05-Arguments

- Finish the `read.delim()` call to import the data in `"hotdogs.txt"`. It's a tab-delimited file without names in the first row.
- The code that selects the observation with the lowest calorie count and stores it in the variable `lily` is already available. It uses the function `which.min()`, that returns the index the smallest value in a vector.
- Do a similar thing for Tom: select the observation with the *most sodium* and store it in `tom`. Use `which.max()` this time.
- Finally, print both the observations `lily` and `tom`.

```r
# Finish the read.delim() call
hotdogs <- read.delim(hotdogs_path, header = FALSE, col.names = c("type", "calories", "sodium"))

# Select the hot dog with the least calories: lily
lily <- hotdogs[which.min(hotdogs$calories), ]

# Select the observation with the most sodium: tom
tom <- hotdogs[which.max(hotdogs$sodium),]

# Print lily and tom
print(lily)
```

```
##       type calories sodium
## 50 Poultry       86    358
```

```r
print(tom)
```

```
##    type calories sodium
## 15 Beef      190    645
```

06-Column classes

- The `read.delim()` call from before is already included and creates the `hotdogs` data frame. Go ahead and display the structure of `hotdogs`.
- **Edit** the second `read.delim()` call. Assign the correct vector to the `colClasses` argument. `NA` should be replaced with a character vector: `c("factor", "NULL", "numeric")`.
- Display the structure of `hotdogs2` and look for the difference.

```r
# Previous call to import hotdogs.txt
hotdogs <- read.delim(hotdogs_path, header = FALSE, col.names = c("type", "calories", "sodium"))

# Display structure of hotdogs
str(hotdogs)
```

```
## ’data.frame’:    54 obs. of  3 variables:
##  $ type    : chr  "Beef" "Beef" "Beef" "Beef" ...
##  $ calories: int  186 181 176 149 184 190 158 139 175 148 ...
##  $ sodium  : int  495 477 425 322 482 587 370 322 479 375 ...
```

```r
# Edit the colClasses argument to import the data correctly: hotdogs2
hotdogs2 <- read.delim(hotdogs_path, header = FALSE,
                       col.names = c("type", "calories", "sodium"),
                       colClasses = c("factor", "NULL", "numeric"))


# Display structure of hotdogs2
str(hotdogs2)
```

```
## 'data.frame':    54 obs. of  2 variables:
##  $ type  : Factor w/ 3 levels "Beef","Meat",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ sodium: num  495 477 425 322 482 587 370 322 479 375 ...
```

```r
head(hotdogs,3)
```

```
##   type calories sodium
## 1 Beef      186    495
## 2 Beef      181    477
## 3 Beef      176    425
```

```r
head(hotdogs2,3)
```

```
##   type sodium
## 1 Beef    495
## 2 Beef    477
## 3 Beef    425
```

The END