Practical Exam

Rey Angelo Calopez

2024-03-06

A. Load the built-in warpbreaks dataset.

```
data("warpbreaks")
```

#1. Find out, in a single command, which columns of warpbreaks are either numeric or integer. What are the data types of each column?

```
str(warpbreaks)
```

```
## 'data.frame':    54 obs. of  3 variables:
##  $ breaks : num  26 30 54 25 70 52 51 26 67 18 ...
##  $ wool   : Factor w/ 2 levels "A","B": 1 1 1 1 1 1 1 1 1 1 ...
##  $ tension: Factor w/ 3 levels "L","M","H": 1 1 1 1 1 1 1 1 1 2 ...
```

#2. There are 54 observations.

```
nrow(warpbreaks)
```

```
## [1] 54
```

#3.Is numeric a natural data type for the columns which are stored as such? Convert to integer when necessary.

```
#warpbreaks <- as.integer(warpbreaks)
```

#4. The error message indicates that the wool column in the warpbreaks dataset is categorized as 'factor'. The function as.integer() is unable to convert this type into an 'integer' vector. This is due to the wool column being a categorical variable, and converting it to an integer would lead to an arbitrary encoding of the categories. Therefore, it is recommended to retain it as a factor variable or convert it to a character variable instead.

B. Load the exampleFile.txt

#1. Read the complete file using readLines.

```
#exampleFile <- readLines("exampleFile.txt")
#exampleFile
```

#2 Separate the vector of lines into a vector containing comments and a vector containing the data. Hint: use grepl.

```
#comments <- file_content[grepl("^//", file_content)]
#data_lines <- file_content[!grepl("^//", file_content)]
```

#3 Extract the date from the first comment line and display on the screen "It was created data."

```
#comments <- file_content[grepl("^//", file_content)]
#data_lines <- file_content[!grepl("^//", file_content)]
#date_line <- comments[1]
```

```
#date <- gsub("^// Created on ", "", date_line)
#cat("It was created on", date, "\n")
```

Output: It was created on // Survey data. Created : 21 May 2013

#4 Read the data into a matrix as follows. #a. Split the character vectors in the vector containing data lines by semicolon (;) using strsplit. #b. Find the maximum number of fields retrieved by split. Append rows that are shorter with NA's. #c. Use unlist and matrix to transform the data to row-column format. #d. From comment lines 2-4, extract the names of the fields. Set these as colnames for the matrix you just created.

```
#split_Exampledata <- strsplit(data_lines, ";")

#max_fields <- max(sapply(split_Exampledata, length))
#filled_split_data <- lapply(split_Exampledata, function(x) c(x, rep(NA, max_fields - length(x))))

#data_matrix <- matrix(unlist(filled_split_data), ncol = max_fields, byrow = TRUE)

#field_names <- strsplit(comments[2:4], ";")
#colnames(data_matrix) <- unlist(field_names)

# Display the resulting matrix
#print(data_matrix)
```

Output:

// Field 1: Gender // Field 2: Age (in years) // Field 3: Weight (in kg) [1,] "M" "28" "81.3"
[2,] "male" "45" NA
[3,] "Female" "17" "57,2"
[4,] "fem." "64" "62.8"