Practical Exam 1

Barbie Joy Obas

2024-03-06

#A. Load the built-in warpbreaks dataset. data(warpbreaks)

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

```
numeric_col <- sapply(warpbreaks, is.numeric)
numeric_col
```

```
##  breaks    wool tension
##    TRUE   FALSE   FALSE
```

```
#The data type of each column are breaks, wool and tension.
```

#A.2. How many observations does it have?

```
observations <- nrow(warpbreaks)
observations
```

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

```
#There are 54 observations
```

```
#The error is an object 'numeric_or_integer_columns' not found suggests that R couldn't find the object
```

#B. Load the exampleFile.txt

```
data <- read.csv("exampleFile.txt")
```

#B.1. Read the complete file using readLines.

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

```
## Warning in readLines("exampleFile.txt"): incomplete final line found on
## 'exampleFile.txt'
```

```
cat(lines, sep = "\n")
```

```
## // Survey data. Created : 21 May 2013
## // Field 1: Gender
## // Field 2: Age (in years)
## // Field 3: Weight (in kg)
## M;28;81.3
## male;45;
## Female;17;57,2
## fem.;64;62.8
```

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

```
comments <- lines[grepl("^//", lines)]
comments
```

```
## [1] "// Survey data. Created : 21 May 2013"
## [2] "// Field 1: Gender"
## [3] "// Field 2: Age (in years)"
## [4] "// Field 3: Weight (in kg)"
```

```
data_lines <- lines[!grepl("^//", lines)]
data_lines
```

```
## [1] "M;28;81.3"      "male;45;"       "Female;17;57,2" "fem.;64;62.8"
```

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

```
date <- gsub("^// Survey data. Created : ", "", comments[1])
date
```

```
## [1] "21 May 2013"
```

#B.4. Read the data into a matrix as follows.

#4.a. Split the character vectors in the vector containing data lines by semicolon (;) using strsplit.

```
split_data <- strsplit(data_lines, ";")
split_data
```

```
## [[1]]
## [1] "M"    "28"   "81.3"
##
## [[2]]
## [1] "male" "45"
##
## [[3]]
## [1] "Female" "17"     "57,2"
##
## [[4]]
## [1] "fem." "64"   "62.8"
```

#4.b. Find the maximum number of fields retrieved by split. Append rows that are shorter with NA's.

```
max_fields <- max(sapply(split_data, length))
max_fields
```

```
## [1] 3
```

```
split_data <- lapply(split_data, function(x) c(x, rep(NA, max_fields - length(x))))
split_data
```

```
## [[1]]
## [1] "M"    "28"   "81.3"
##
## [[2]]
## [1] "male" "45"   NA
##
## [[3]]
## [1] "Female" "17"     "57,2"
##
## [[4]]
## [1] "fem." "64"   "62.8"
```

#4.c. Use unlist and matrix to transform the data to row-column format.

```
data_matrix <- matrix(unlist(split_data), ncol = max_fields, byrow = TRUE)
data_matrix
```

```
##      [,1]     [,2] [,3]
## [1,] "M"      "28" "81.3"
## [2,] "male"   "45" NA
## [3,] "Female" "17" "57,2"
## [4,] "fem."   "64" "62.8"
```

#4.d. From comment lines 2-4, extract the names of the fields. Set these as colnames for the matrix you just created.

```
fieldNames <- gsub("^// Field [0-9]+: ", "", comments[2:4])
fieldNames
```

```
## [1] "Gender"         "Age (in years)" "Weight (in kg)"
```

```
colnames(data_matrix) <- fieldNames
colnames(data_matrix)
```

```
## [1] "Gender"         "Age (in years)" "Weight (in kg)"
```