Practical Exam

Dianah Marie Canonicato

2024-03-07

A. Load the built-in warpbreaks dataset.

```
#1

# 1.
num_col <- sapply(warpbreaks, function(x) is.numeric(x) || is.integer(x))
print(num_col)
```

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

```
#2. How many observations does it have?
num_observations <- nrow(warpbreaks)
print(paste("Number of observations:", num_observations))
```

```
## [1] "Number of observations: 54"
```

```
#3. Is numeric a natural data type for the columns which are stored as such? Convert to integer when ne
#YES
warpbreaks[, num_col] <- sapply(warpbreaks[, num_col], as.integer)
print(warpbreaks)
```

```
##    breaks wool tension
## 1      26    A       L
## 2      30    A       L
## 3      54    A       L
## 4      25    A       L
## 5      70    A       L
## 6      52    A       L
## 7      51    A       L
## 8      26    A       L
## 9      67    A       L
## 10     18    A       M
## 11     21    A       M
## 12     29    A       M
## 13     17    A       M
## 14     12    A       M
## 15     18    A       M
## 16     35    A       M
## 17     30    A       M
## 18     36    A       M
## 19     36    A       H
## 20     21    A       H
## 21     24    A       H
## 22     18    A       H
## 23     10    A       H
## 24     43    A       H
## 25     28    A       H
## 26     15    A       H
## 27     26    A       H
## 28     27    B       L
## 29     14    B       L
## 30     29    B       L
## 31     19    B       L
## 32     29    B       L
## 33     31    B       L
## 34     41    B       L
## 35     20    B       L
## 36     44    B       L
## 37     42    B       M
## 38     26    B       M
## 39     19    B       M
## 40     16    B       M
## 41     39    B       M
## 42     28    B       M
## 43     21    B       M
## 44     39    B       M
## 45     29    B       M
## 46     20    B       H
## 47     21    B       H
## 48     24    B       H
## 49     17    B       H
## 50     13    B       H
## 51     15    B       H
## 52     15    B       H
## 53     16    B       H
## 54     28    B       H
```

```
# 4. Error messages in R sometimes report the underlying type of an object rather than the user-level c
# Example code:
# my_list <- list(1, "a", TRUE)
# my_matrix <- matrix(my_list)

# The error message would be:
# Error in matrix(my_list, nrow = 3) :
#   data must be of a vector type, was 'list'

# From this error message, we can derive that the underlying type of the object 'my_list' is 'list'.
# The error message indicates that the 'matrix' function expects data to be of a vector type, but 'my_l
# This error occurs because the 'matrix' function can only accept vectors as input, not lists.
```

B. Load the exampleFile.txt

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

file_path <- "/cloud/project/PractExam/exampleFile (2).txt"
lines <- readLines(file_path, warn = FALSE)
print(lines)
```

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

```
#2. Separate the vector of lines into a vector containing comments and a vector containing the data. Hi

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

```
## character(0)
```

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

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

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

# Find the comment line containing the date information
date_line_index <- grep("^# Date: ", comments)

if (length(date_line_index) > 0) {
  # Extract the date from the first comment line containing the date information
  date_line <- comments[date_line_index]
  print(date_line)
  date <- gsub("# Date: ", "", date_line)
  print(date)
} else {
  print("Date information not found in the comments.")
}
```

```
## [1] "Date information not found in the comments."
```

4. Read the data into a matrix as follows.

```
# a.
split_data_lines <- strsplit(data_lines, ";")
print(split_data_lines)
```

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

```
## [[4]]
## [1] "// Field 3: Weight (in kg)"
##
## [[5]]
## [1] "M"    "28"   "81.3"
##
## [[6]]
## [1] "male" "45"
##
## [[7]]
## [1] "Female" "17"     "57,2"
##
## [[8]]
## [1] "fem." "64"   "62.8"
```

```
# b.
max_fields <- max(sapply(split_data_lines, length))
print(max_fields)
```

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

```
split_data_lines <- lapply(split_data_lines, function(x) {
  if (length(x) < max_fields) {
    c(x, rep(NA, max_fields - length(x)))
  } else {
    x
  }
})
print(split_data_lines)
```

```
## [[1]]
## [1] "// Survey data. Created : 21 May 2013"
## [2] NA
## [3] NA
##
## [[2]]
## [1] "// Field 1: Gender" NA                   NA
##
## [[3]]
## [1] "// Field 2: Age (in years)" NA
## [3] NA
##
## [[4]]
## [1] "// Field 3: Weight (in kg)" NA
## [3] NA
##
## [[5]]
## [1] "M"    "28"   "81.3"
##
## [[6]]
## [1] "male" "45"   NA
##
## [[7]]
## [1] "Female" "17"     "57,2"
##
```

```
## [[8]]
## [1] "fem." "64"   "62.8"
```

```
# c.
data_matrix <- matrix(unlist(split_data_lines), nrow = length(split_data_lines), byrow = TRUE)
print(data_matrix)
```

```
##      [,1]                                    [,2] [,3]
## [1,] "// Survey data. Created : 21 May 2013" NA   NA
## [2,] "// Field 1: Gender"                    NA   NA
## [3,] "// Field 2: Age (in years)"            NA   NA
## [4,] "// Field 3: Weight (in kg)"            NA   NA
## [5,] "M"                                     "28" "81.3"
## [6,] "male"                                  "45" NA
## [7,] "Female"                                "17" "57,2"
## [8,] "fem."                                  "64" "62.8"
```

```
# d.
field_names_vector <- gsub("# ", "", comments[2:4])
print(field_names_vector)
```

```
## [1] NA NA NA
```

```
dim(data_matrix)
```

```
## [1] 8 3
```

```
field_names_vector<- strsplit(field_names_vector, ": ")[[1]]
print(field_names_vector)
```

```
## [1] NA
```

```
length_field_names <- length(field_names_vector)
print(length_field_names)
```

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

```
if (ncol(data_matrix) == length_field_names) {
  colnames(data_matrix) <- field_names_vector
} else {
  # Handle the mismatch (adjust your code accordingly)
  print("The length of the column names does not correspond with the number of columns.")
}
```

```
## [1] "The length of the column names does not correspond with the number of columns."
```

#C. Pushing into GitHub #1. The .rmd should be knitted into pdf form. #2. Create a folder and named it as PractExam. The PractExam will contain the .rmd and the pdf files. #3. Push the folder – PractExam into your GitHub repo. There is no need to change the repo. Just use the repo you have created before.