Practical Exam

2024-02-08

A. Load the built-in warpbreaks dataset.

```
#1.Find out, in a single command, which columns of warpbreaks are either numeric or integer. What are t

data(warpbreaks)
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
```

```
sapply(warpbreaks, class)
```

```
##    breaks      wool   tension
## "numeric"  "factor"  "factor"
```

```
#2.How many observations does it have?
nrow(warpbreaks)
```

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

```
# The number of observations the data set have is 54.
```

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

numeric <-as.integer(warpbreaks$breaks)
class(warpbreaks$breaks)
```

```
## [1] "numeric"
```

```
#4Explain what is the error all about. Do not just copy the error message that was displayed.

# I did not counter an error withg my previous codes.
```

B.Load the exampleFile.txt

```
#1.Read the complete file using readLines.
ReadExample <- readLines("exampleFile.txt")
ReadExample
```

```
## [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. Hin

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

```r
comment_lines <- grepl("^#", ReadExample)

comments <- ReadExample[comment_lines]
data <- ReadExample[!comment_lines]

data
```

```
## [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"
```

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

ReadExample <- readLines("exampleFile.txt")
firstLine <- ReadExample[1]
dateString <- gsub("Created : (\\d{2} [A-Za-z]+ \\d{4})", "\\1", firstLine)
cat("It was created", dateString, ".\n")
```

```
## It was created // Survey data. 21 May 2013 .
```

```r
#4 Read the data into a matrix as follows.

#a. Split the character vectors in the vector containing data lines by semicolon (;) using strsplit.
splitmtrx <- lapply(data, function(line) unlist(strsplit(line, split = ";")))
splitmtrx
```

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

max <- max(sapply(splitmtrx, length))
max
```

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

```
withNA <- lapply(splitmtrx, function(line) {
  line_with_na <- c(line, rep(NA, max - length(line)))
  return(line_with_na)
})

withNA
```

```
## [[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. Use unlist and matrix to transform the data to row-column format.

rowcol <- matrix(unlist(withNA), nrow = max, byrow = TRUE)

rowcol
```

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

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

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

#comment_lines <- grepl("^#", ReadExample)
#comments <- ReadExample[comment_lines]
#data <- ReadExample[!comment_lines]

#data <- matrix(unlist(strsplit(data, split = ";")), nrow = max(sapply(strsplit(data, split = ";"), #le
#colnames(data) <- gsub("^#", "", comments[2:4])

#ERROR : Error in dimnames(x) <- dn :  length of 'dimnames' [2] not equal to array extent
```