PracExam

Meigelyn Joy Matias

2024-03-07

```
#A
data ("warpbreaks")


#1. Find out, in a single command, which columns of warpbreaks are either numeric or integer.
numeric_col <- sapply(warpbreaks, is.numeric)
numeric_col
```

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

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

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

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

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

```
numeric_or_integer_columns <- warpbreaks[, numeric_col | integer_col]
numeric_or_integer_columns
```

```
##  [1] 26 30 54 25 70 52 51 26 67 18 21 29 17 12 18 35 30 36 36 21 24 18 10 43 28
## [26] 15 26 27 14 29 19 29 31 41 20 44 42 26 19 16 39 28 21 39 29 20 21 24 17 13
## [51] 15 15 16 28
```

```
#4. Error messages in R sometimes report the underlying type of an object rather than the user-level cl

#for (i in 1:ncol(numeric_or_integer_columns)) {
#if (!is.integer(numeric_or_integer_columns[, i])) {
    #numeric_or_integer_columns[, i] <- as.integer(numeric_or_integer_columns[, i])
  #}
#}

#The given code tries to loop through each column in `numeric_or_integer_columns` and convert any non-i
```

```
#B

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

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

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

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

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

```
#3.Extract the date from the first comment line and display on the screen "It was created data."
date <- gsub("^// Survey data. Created : ", "It was created data ", comments[1])
date
```

```
## [1] "It was created data 21 May 2013"
```

```
#4a. Split the character vectors in the vector containing data lines by semicolon (;) using strsplit.
splitdata <- strsplit(datalines, ";")
splitdata
```

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

```
#4b. Find the maximum number of fields retrieved by split. Append rows that are shorter with NA's.
maxfields <- max(sapply(splitdata, length))
maxfields
```

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

```
splitdata <- lapply(splitdata, function(x) c(x, rep(NA, maxfields - length(x))))
splitdata
```

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

```
#4c. Use unlist and matrix to transform the data to row-column format.
datamatrix <- matrix(unlist(splitdata), ncol = maxfields, byrow = TRUE)
datamatrix
```

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

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

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

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

```
colnames(datamatrix) <- fieldNames
colnames(datamatrix)
```

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