

PractExam

Pineda, Andrea Patrice A.

2024-03-07

#A

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

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

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

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

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

```
Obs <- nrow(warpbreaks)
Obs
```

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

#There are 54 observations. The variables are breaks, wool, and tension.

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

```
IntCol <- sapply(warpbreaks, is.integer)
IntCol
```

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

```
NumOrIntCol <- warpbreaks[, NumericCol | IntCol]
NumOrIntCol
```

```
## [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
```

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

#Without a valid dataset, the code can't determine the number of columns and generates an error.

#B

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

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

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

```
## 'exampleFile.txt'
#B.2 - Separate the vector of lines into a vector containing comments and a vector containing the data.

Comments <- Lines[grep("//", 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[!grep("//", Lines)]
DataLines

## [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.

Date <- gsub("// Survey data. Created : ", "", Comments[1])
Date

## [1] "21 May 2013"
cat("It was created data", Date)

## It was created data 21 May 2013
#B.4a. - Split the character vectors in the vector containing data lines by semicolon (;) using strsplit

DataSplit <- strsplit(DataLines, ";")
DataSplit

## [[1]]
## [1] "M"      "28"      "81.3"
##
## [[2]]
## [1] "male" "45"
##
## [[3]]
## [1] "Female" "17"      "57,2"
##
## [[4]]
## [1] "fem." "64"      "62.8"
#B.4b - Find the maximum number of fields retrieved by split. Append rows that are shorter with NA's.

max_fields <- max(sapply(DataSplit, length))
max_fields

## [1] 3
DataSplit <- lapply(DataSplit, function(x) c(x, rep(NA, max_fields - length(x))))
DataSplit

## [[1]]
## [1] "M"      "28"      "81.3"
##
## [[2]]
## [1] "male" "45"      NA
```

```
##
## [[3]]
## [1] "Female" "17"      "57,2"
##
## [[4]]
## [1] "fem." "64"     "62.8"
```

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

```
Matrix_Data <- matrix(unlist(DataSplit), ncol = max_fields, byrow = TRUE)
Matrix_Data
```

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

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

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

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

```
colnames(Matrix_Data) <- Field_Names
colnames(Matrix_Data)
```

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