

PracticalExam

Ulgasan

2024-03-06

Load the built-in warpbreaks dataset.

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

Find out, in a single command, which columns of warpbreaks are either numeric or integer. What are the data types of each column? —Answer— the “breaks” column of “warpbreak” are numerical the “wool” column of “warpbreak are factors w/ 2 levels “A”, “B”: 1 1 1 1 1 1 1 1 1 1 ... the “tension” column of “warpbreak” are also factors w/ 3 levels “L”, “M”, “H”: 1 1 1 1 1 1 1 1 1 2 ...

```
str(warpbreaks)
```

```
## 'data.frame': 54 obs. of 3 variables:
## $ breaks : num 26 30 54 25 70 52 51 26 67 18 ...
## $ wool : Factor w/ 2 levels "A","B": 1 1 1 1 1 1 1 1 1 1 ...
## $ tension: Factor w/ 3 levels "L","M","H": 1 1 1 1 1 1 1 1 1 2 ...
```

2. How many observations does it have? —Answer— it does have “54” observations

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

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

3. Is numeric a natural data type for the columns which are stored as such? Convert to integer when necessary. —Answer— Yes, numeric is a natural data type

```
data("warpbreaks")
warpbreaks$tension <- as.integer(warpbreaks$tension)
warpbreaks$wool <- as.integer(warpbreaks$wool)
warpbreaks$breaks <- as.integer(warpbreaks$breaks)
str(warpbreaks)
```

```
## 'data.frame': 54 obs. of 3 variables:
## $ breaks : int 26 30 54 25 70 52 51 26 67 18 ...
## $ wool : int 1 1 1 1 1 1 1 1 1 1 ...
## $ tension: int 1 1 1 1 1 1 1 1 1 2 ...
```

```
#View(warpbreaks)
```

4. Error messages in R sometimes report the underlying type of an object rather than the user-level class. Derive from the following code and error message what the underlying type. Explain what is the error all about. Do not just copy the error message that was displayed. —Answer— This mistake arises from the fact that the x object has character strings (such as “NA”) that represent integers. R meets the “NA” string, which cannot be immediately converted to an integer, when attempting to force these character strings to numbers. Before converting the character vector to integers, you must handle the “NA” values appropriately, either by deleting them or substituting a suitable value.

```
x <- c("1", "2", "3", "4", "5", "NA")
as.integer(x)
```

```
## Warning: NAs introduced by coercion
```

```
## [1] 1 2 3 4 5 NA
```

Load the exampleFile.txt

1. Read the complete file using readLines.

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

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

```
lines <- c("// 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")
```

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

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

```
print("Comments:")
```

```
## [1] "Comments:"
```

```
print(comments)
```

```
## [1] "// Survey data. Created : 21 May 2013"
```

```
## [2] "// Field 1: Gender"
```

```
## [3] "// Field 2: Age (in years)"
```

```
## [4] "// Field 3: Weight (in kg)"
```

```
print("Data:")
```

```
## [1] "Data:"
```

```
print(data)
```

```
## [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.”

```
lines <- c("// 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")
```

```
first_comment <- lines[grepl("^//", lines)][1]
date <- gsub("^.*Created : ", "", first_comment)
```

```
cat("It was created on:", date)
```

```
## It was created on: 21 May 2013
```

4. Read the data into a matrix as follows. a. Split the character vectors in the vector containing data lines by semicolon (;) using strsplit.

```
lines <- c("M;28;81.3",
          "male;45;",
          "Female;17;57,2",
          "fem.;64;62.8")
```

```
split_data <- lapply(strsplit(lines, ";"), function(x) trimws(x))
```

```
matrix_data <- do.call(rbind, split_data)
```

```
## Warning in (function (... , deparse.level = 1) : number of columns of result is
## not a multiple of vector length (arg 2)
```

```
print(matrix_data)
```

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

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

```
lines <- c("M;28;81.3",
          "male;45;",
          "Female;17;57,2",
          "fem.;64;62.8")
```

```
split_data <- lapply(strsplit(lines, ";"), function(x) trimws(x))
```

```
max_fields <- max(lengths(split_data))
```

```
for (i in seq_along(split_data)) {
  if (length(split_data[[i]]) < max_fields) {
    split_data[[i]] <- c(split_data[[i]], rep(NA, max_fields - length(split_data[[i]])))
  }
}
```

```
matrix_data <- do.call(rbind, split_data)
```

```
print(matrix_data)
```

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

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

```
split_data<-unlist(split_data)
```

```
split_data
```

```
## [1] "M"      "28"      "81.3"   "male"   "45"      NA        "Female" "17"
## [9] "57,2"   "fem."    "64"     "62.8"
```

```
dataMatrix <- matrix(split_data, nrow = length(data),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"
```

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

```
fieldNames <- c("Gender", "Age", "Weight")
```

```
colnames(dataMatrix) <- fieldNames
```

```
print(dataMatrix)
```

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