

CS102_BSIT2C_JUNTANILLA_PRAC

JOSE ROLAND JUNTANILLA

2024-03-06

A. Load the built-in warpbreaks dataset.

```
structure(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
```

1. Find out, in a single command, which columns of warpbreaks are either numeric or integer. What are the data types of each column?

```
df_datatype <- sapply(warpbreaks, function(x) typeof(x))
df_datatype
```

```
##      breaks      wool      tension
## "double" "integer" "integer"
```

#The data types of each column is double, integer, integer.

2. How many observations does it have?

```
(obs_df <- nrow(warpbreaks))
```

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

The dataframe warpbreaks has a 54 Observations.

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

```
is.integer(warpbreaks$breaks)
```

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

```
warpbreaks$wool <- as.integer(warpbreaks$wool)
warpbreaks$tension <- as.integer(warpbreaks$tension)
warpbreaks
```

```
##      breaks wool tension
## 1      26      1        1
## 2      30      1        1
## 3      54      1        1
## 4      25      1        1
## 5      70      1        1
## 6      52      1        1
## 7      51      1        1
## 8      26      1        1
## 9      67      1        1
## 10     18      1        2
## 11     21      1        2
```

```
## 12      29      1      2
## 13      17      1      2
## 14      12      1      2
## 15      18      1      2
## 16      35      1      2
## 17      30      1      2
## 18      36      1      2
## 19      36      1      3
## 20      21      1      3
## 21      24      1      3
## 22      18      1      3
## 23      10      1      3
## 24      43      1      3
## 25      28      1      3
## 26      15      1      3
## 27      26      1      3
## 28      27      2      1
## 29      14      2      1
## 30      29      2      1
## 31      19      2      1
## 32      29      2      1
## 33      31      2      1
## 34      41      2      1
## 35      20      2      1
## 36      44      2      1
## 37      42      2      2
## 38      26      2      2
## 39      19      2      2
## 40      16      2      2
## 41      39      2      2
## 42      28      2      2
## 43      21      2      2
## 44      39      2      2
## 45      29      2      2
## 46      20      2      3
## 47      21      2      3
## 48      24      2      3
## 49      17      2      3
## 50      13      2      3
## 51      15      2      3
## 52      15      2      3
## 53      16      2      3
## 54      28      2      3
```

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.

B. Load the exampleFile.txt 1. Read the complete file using readLines.

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

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

```
print(file_cont)
```

```
## [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. Hint: use `grepl`.

```
(com <- file_cont[grepl("//", file_cont)])
```

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

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

```
## [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(".*Created : ", "", com[1]))
```

```
## [1] "21 May 2013"
```

```
paste0("It was created ", date)
```

```
## [1] "It was created 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`.

```
(splitLines <- strsplit(data, ";"))
```

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

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

```
(veclength <- sapply(splitLines, length))
```

```
## [1] 3 2 3 3
```

```
(Fieldfmax <- max(veclength))
```

```
## [1] 3
(splitLines <- lapply(splitLines, function(row) c(row, rep(NA, Fieldfmax - length(row)))))

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