Practical Exam

Luigi Echaveria

2024-03-07

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

```
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
```

```
numeric_columns <- sapply(warpbreaks, is.numeric)
integer_columns <- sapply(warpbreaks, is.integer)
```

#2. How many observations does it have?

```
nrow(warpbreaks)
```

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

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

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

#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.

```
x <- c("1", "2", "3", "4")
x <- as.numeric(x)
y <- x + 1
```

#Error message: #Error in x + 1 : non-numeric argument to binary operator

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

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

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

```
comments <- grepl("^#", ReadExample)
comments_vec <- ReadExample[comments]
data_vec <- ReadExample[!comments]

cat("Comments:\n")
```

```
## Comments:
```

```
print(comments_vec)
```

```
## character(0)
```

```
cat("\nData:\n")
```

```
##
## Data:
```

```
print(data_vec)
```

```
## [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"
```

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

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

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

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

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