sb_rmd_ex_07

Sander Bossier

2024-01-17

1.

Consider the character vector x:

```
x <- c("Swan swam over the pond, Swim swan swim!",
"Swam swam back again - Well swum swan!")
```

Split the strings in `x` into words, removing any *commas* or *hyphens*. Call the result `y`.

```
y <- strsplit(
  x,
  split = ",? -? ?"
)
y
```

```
## [[1]]
## [1] "Swan"  "swam"  "over"  "the"   "pond"  "Swim"  "swan"  "swim!"
##
## [[2]]
## [1] "Swam"  "swam"  "back"  "again" "Well"  "swum"  "swan!"
```

2.

The .rda file `hafu` in the RStudio Project `Text` contains a data set `hafu` on manga characters. In the `Father` and `Mother` columns, some values have question marks after the country name, indicating that the author was uncertain about the nationality of the parent:

```
load("hafu.rda")
head(hafu[, -ncol(hafu)], n = 10)
```

```
##    Year                 Series                Character Gender        Father
## 1  1963           Yuki no Taiyou                    Sanae      F      Japanese
## 2  1964               Cyborg 009            Joe Shimamura      M      American
## 3  1967                Lupin III                Lupin III      M       French?
## 4  1967             Nekome Kozou             Cat-Eyed Boy      M      Japanese
## 5  1972                Gatchaman              Jun the Swan      F          <NA>
## 6  1974           Great Mazinger                 Jun Hono      F      American
## 7  1975        Ichigo Monogatari                   Ichigo      F Scandinavian?
## 8  1976 Kochira Katsushika-ku... Reiko Katherine Akimoto      F     Japanese?
## 9  1977      Haikara-san ga Tooru          Shinobu Iijyuin      M      Japanese
## 10 1979       Mobile Suit Gundam                Amuro Ray      M      Japanese
##       Mother  Eyes  Hair
## 1   American  <NA>  <NA>
## 2   Japanese brown brown
## 3  Japanese? black black
## 4    Fantasy brown  <NA>
## 5       <NA> green green
## 6   Japanese black black
## 7  Japanese? brown brown
## 8    French?  blue blond
## 9     German  blue blond
## 10  Canadian  blue brown
```

Create two vectors `FatherUncertain` and `MotherUncertain` denoting whether or not there was a question mark in the `Father` or `Mother` column, respectively.

```r
FatherUncertainSB <- endsWith(
  x = as.character(hafu$Father),
  suffix = "?"
)
MotherUncertainSB <- endsWith(
  x = as.character(hafu$Mother),
  suffix = "?"
)

FatherUncertain <- with(hafu, grepl("\\?", Father))
MotherUncertain <- with(hafu, grepl("\\?", Mother))

identical(FatherUncertainSB, FatherUncertain)
```

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

```r
identical(MotherUncertainSB, MotherUncertain)
```

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

3.

Create a new data frame `hafu2` by removing the question marks from the `Father` and `Mother` columns in `hafu` (i.e. replace them by an empty string).

```r
hafu2 <- within(hafu, {
  Father <- gsub(pattern = "\\?", replacement = "", x = Father)
  Mother <- gsub(pattern = "\\?", replacement = "", x = Mother)
  }
)
```

Challenges

4.

The .rda file `Dataline` contains a vector `Dataline` that contains a single value. This value is the result of a webscraper. It contains 1500 fields, separated by a semicolon:

```r
load("Dataline.rda")
```

Create a vector `Fields` which contains the 1500 fields as separate values.

```r
Fields <- strsplit(Dataline, split = ";")[[1]]
```

5.

On the basis of the vector `Fields` create: * A character vector `Label` containing the labels of every field: possible values are "Low", "Middle" and "High". * A numeric vector `ID` containing the ID numbers of every field (i.e. the value following the underscore). * A character vector `newLabel` containing labels constructed as `ID_xxx_Label`, with `xxx` the value of `ID` for that field and `Label` the value of `Label` for that field.

```r
head(Fields, n = 10)
```

```
##  [1] "MiddleID_54"  "HighID_115"   "LowID_109"    "HighID_346"   "MiddleID_298"
##  [6] "HighID_139"   "MiddleID_221" "MiddleID_447" "HighID_27"    "LowID_331"
```

```r
Label <- gsub(
  pattern = "ID_\\d+", #KP solution is more general by no requiring the underscore followed by any numb
  replacement = "",
  x = Fields
)

ID <- as.numeric(
  gsub(
    pattern = "\\w+ID_", # again KP solution is more general as it matches a sequence of one or more no
    replacement = "",
    x = Fields
  )
)

newLabel <- paste(
  "ID",
  ID,
  Label,
  sep = "_"
)
```

```r
# Solution 5:
LabelKP <- gsub("ID.*", "", Fields)
IDKP <- as.numeric(gsub("[^_]+_", "", Fields))
newLabelKP <- paste("ID", ID, Label, sep = "_")
```

```r
identical(Label, LabelKP)
```

```
## [1] TRUE
```

```r
identical(ID, IDKP)
```

```
## [1] TRUE
```

```r
identical(newLabel, newLabelKP)
```

```
## [1] TRUE
```

6.

The file genes.rda contains a named vector **genes**. This vector contains the function descriptions of a set of genes. The names of this vector give the COG codes of these genes:

```r
load("genes.rda")
head(genes, n = 10)
```

```
##                                                                    RCOG3478
## "Predicted nucleic-acid-binding protein containing a Zn-ribbon domain "
##                                                                CCOG1031
##                                   "Uncharacterized Fe-S oxidoreductase "
##                                                                SCOG3771
##                                            "Predicted membrane protein "
##                                                                KCOG1191
##                  "DNA-directed RNA polymerase specialized sigma subunit "
##                                                                SCOG5230
##                                     "Uncharacterized conserved protein "
##                                                                JCOG0532
##                         "Translation initiation factor 2 (IF-2; GTPase) "
##                                                                CCOG1905
##                           "NADH:ubiquinone oxidoreductase 24 kD subunit "
##                                                                FCOG0461
##                                     "Orotate phosphoribosyltransferase "
##                                                                SCOG2904
##                          "Uncharacterized protein conserved in bacteria "
##                                                                ECOG0509
##                    "Glycine cleavage system H protein (lipoate-binding) "
```

Use this vector to create a character vector `geneclass` with the values “Predicted”, “Uncharacterized” and “Described”. These are defined as follows: * If the gene description contains “Uncharacterized”, it is always called “Uncharacterized” * If the gene description contains “Predicted” but not “Uncharacterized”, it is called “Predicted” * All other genes are called “Described”

```
geneclass <- rep("Described", length(genes))
IDunc <- grep(pattern = "uncharacterized", x = genes, ignore.case = TRUE)
IDpred <- grep(pattern = "predicted", x = genes, ignore.case = TRUE)
geneclass[IDpred] <- "Predicted"
geneclass[IDunc] <- "Uncharacterized"
```

7.

Use the `genes` vector to find all COG codes for functions containing an enzyme name. An enzyme can be recognised by their name ending on “ase” or “ases”. For this, you need to: * Extract the COG code from the names: this consists of the numbers following the letters COG in the name * Find at which positions in the vector genes an enzyme is described * Use the information from the previous two steps to find only the COG codes of the enzymes

```
COGcodes <- gsub(
  pattern = ".*COG",
  replacement = "",
  x = names(genes)
)
```