2657 Functions

Ananda Mahto

May 4, 2012

Contents

concat.split

What it does

The `concat.split` function takes a column with multiple values, splits the values into separate columns, and returns a new `data.frame`.

Arguments

- `data`: is the source `data.frame`
- `split.col`: the variable that needs to be split
- `mode`: can be either `binary` or `value` (where `binary` is default and it recodes values to `1` or `NA`)
- `sep`: the character separating each value (defaults to `","`)
- `drop.col`: logical (whether to remove the original variable from the output or not; defaults to `TRUE`).

The Function

```
concat.split = function(data, split.col, mode = NULL, sep = ",",
    drop.col = FALSE) {

    if (is.numeric(split.col))
        split.col = split.col else split.col = which(colnames(data) %in% split.col)

    a = as.character(data[, split.col])
    b = strsplit(a, sep)
```

```
    if (suppressWarnings(is.na(try(max(as.numeric(unlist(b)))))) {
        what = "string"
        ncol = max(unlist(lapply(b, function(i) length(i))))
    } else if (!is.na(try(max(as.numeric(unlist(b)))))) {
        what = "numeric"
        ncol = max(as.numeric(unlist(b)))
    }

    m = matrix(nrow = nrow(data), ncol = ncol)
    v = vector("list", nrow(data))

    if (identical(what, "string")) {
        temp = as.data.frame(t(sapply(b, "[", 1:ncol)))
        names(temp) = paste(names(data[split.col]), "_", 1:ncol, sep = "")
        temp1 = cbind(data, temp)
    } else if (identical(what, "numeric")) {
        for (i in 1:nrow(data)) {
            v[[i]] = as.numeric(strsplit(a, sep)[[i]])
        }

        temp = v

        for (i in 1:nrow(data)) {
            m[i, temp[[i]]] = temp[[i]]
        }

        m = data.frame(m)
        names(m) = paste(names(data[split.col]), "_", 1:ncol, sep = "")

        if (is.null(mode) || identical(mode, "binary")) {
            temp1 = cbind(data, replace(m, m != "NA", 1))
        } else if (identical(mode, "value")) {
            temp1 = cbind(data, m)
        }
    }

    if (isTRUE(drop.col))
        temp1[-split.col] else temp1

}
```

Examples

First load some data from a CSV stored at github. The URL is an HTTPS, so we need to use `getURL` from `RCurl`.

```
require(RCurl)
```

```
## Loading required package: RCurl
```

```
## Loading required package: bitops
```

```
baseURL = c("https://raw.github.com/mrdwab/2657-R-Functions/master/")
temp = getURL(paste0(baseURL, "data/concatenated-cells.csv"))
concat.test = read.csv(textConnection(temp))
```

```
rm(temp)

# How big is the dataset?
dim(concat.test)

## [1] 48  3

# Just show me the first few rows
head(concat.test)

##     Name     Likes                   Siblings
## 1   Boyd 1,2,4,5,6 Reynolds , Albert , Ortega
## 2  Rufus 1,2,4,5,6  Cohen , Bert , Montgomery
## 3   Dana 1,2,4,5,6                     Pierce
## 4 Carole 1,2,4,5,6 Colon , Michelle , Ballard
## 5 Ramona   1,2,5,6           Snyder , Joann ,
## 6 Kelley   1,2,5,6          James , Roxanne ,
```

Notice that the data have been entered in a very silly manner. Let's split it up!

```
# Split up the second column, selecting by column number
head(concat.split(concat.test, 2))

##     Name     Likes                   Siblings Likes_1 Likes_2 Likes_3
## 1   Boyd 1,2,4,5,6 Reynolds , Albert , Ortega       1       1      NA
## 2  Rufus 1,2,4,5,6  Cohen , Bert , Montgomery       1       1      NA
## 3   Dana 1,2,4,5,6                     Pierce       1       1      NA
## 4 Carole 1,2,4,5,6 Colon , Michelle , Ballard       1       1      NA
## 5 Ramona   1,2,5,6           Snyder , Joann ,       1       1      NA
## 6 Kelley   1,2,5,6          James , Roxanne ,       1       1      NA
##   Likes_4 Likes_5 Likes_6
## 1       1       1       1
## 2       1       1       1
## 3       1       1       1
## 4       1       1       1
## 5      NA       1       1
## 6      NA       1       1

# ... or by name, and drop the offensive first column
head(concat.split(concat.test, "Likes", drop.col = TRUE))

##     Name                   Siblings Likes_1 Likes_2 Likes_3 Likes_4
## 1   Boyd Reynolds , Albert , Ortega       1       1      NA       1
## 2  Rufus  Cohen , Bert , Montgomery       1       1      NA       1
## 3   Dana                     Pierce       1       1      NA       1
## 4 Carole Colon , Michelle , Ballard       1       1      NA       1
## 5 Ramona           Snyder , Joann ,       1       1      NA      NA
## 6 Kelley          James , Roxanne ,       1       1      NA      NA
##   Likes_5 Likes_6
## 1       1       1
## 2       1       1
## 3       1       1
## 4       1       1
## 5       1       1
## 6       1       1
```

```
# Retain the original values
head(concat.split(concat.test, 2, mode = "value", drop.col = TRUE))
```

```
##     Name                   Siblings Likes_1 Likes_2 Likes_3 Likes_4
## 1   Boyd Reynolds , Albert , Ortega       1       2      NA       4
## 2  Rufus  Cohen , Bert , Montgomery       1       2      NA       4
## 3   Dana                     Pierce       1       2      NA       4
## 4 Carole Colon , Michelle , Ballard       1       2      NA       4
## 5 Ramona           Snyder , Joann ,       1       2      NA      NA
## 6 Kelley          James , Roxanne ,       1       2      NA      NA
##   Likes_5 Likes_6
## 1       5       6
## 2       5       6
## 3       5       6
## 4       5       6
## 5       5       6
## 6       5       6
```

```
# Let's try splitting some strings... Same syntax
head(concat.split(concat.test, 3, drop.col = TRUE))
```

```
##     Name     Likes Siblings_1 Siblings_2  Siblings_3
## 1   Boyd 1,2,4,5,6  Reynolds     Albert       Ortega
## 2  Rufus 1,2,4,5,6     Cohen       Bert   Montgomery
## 3   Dana 1,2,4,5,6     Pierce       <NA>        <NA>
## 4 Carole 1,2,4,5,6     Colon   Michelle      Ballard
## 5 Ramona   1,2,5,6    Snyder      Joann         <NA>
## 6 Kelley   1,2,5,6     James    Roxanne         <NA>
```

To Do

- Modify the function so that you can split multiple columns in one go?
- Strip whitespace from string output.

Reference

See: http://stackoverflow.com/q/10100887/1270695