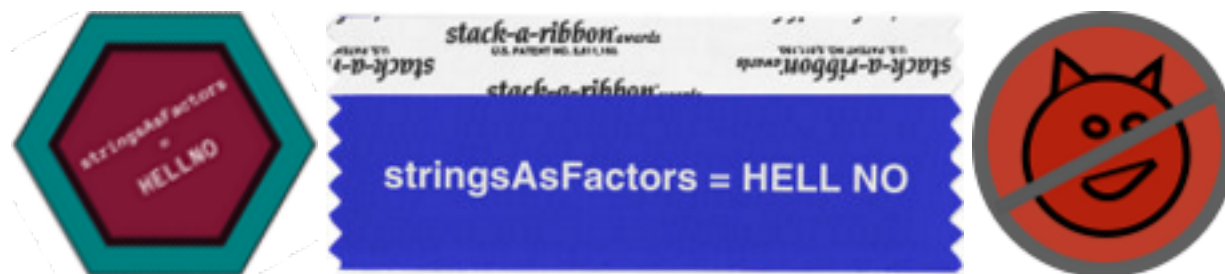


# One Solution to the ‘stringsAsFactors’-Problem Or: Hell-Yeah there is HELLNO

*Peter Meißner*

*2015-12-14*



## Introduction

Base R's `stringsAsFactors` default setting is supposedly the most often complained about piece of code in the whole R infrastructure. A search through the source code of all CRAN packages in December 2015 ([Link](#)) resulted in 3,492 mentions of `stringsAsFactors`. Most of the time these explicit mentions were found within calls to `data.frame()` or `as.data.frame()` and simply set the value to `FALSE`.

The `hellno` package provides an explicit solution to the problem without changing R itself or having to mess around with options. One could use e.g.: `options("stringsAsFactors" = FALSE)` to re-set the global default behavior. Instead `hellno` tackles the problem from another direction, namely by providing alternative implementations of `data.frame()` and `as.data.frame()`. Those *re-implementations* are in fact simple wrappers around base R's very own `data.frame()` and `as.data.frame()` with `stringsAsFactors` option set to `HELLNO` - which in turn equals to `FALSE` and gives the package its name.

Some info material and crediting for ‘hellno’ as catch phrase - thanks Clint?:

- [Some girls and Clint](#)
- [THE \(official\) LOGO](#)
- [twitter-search](#)
- [sillylogic](#) (The one and only package with hand-crafted logical constants)

## Using hellno interactively

Using the package is simple - load it, note the message indicating masking two base functions and code on - from now on no type conversion will take place within `data.frame()` and `as.data.frame()`:

```
# options(repos = c(CRAN = "https://cran.rstudio.com"))
# install.packages("hellno")
library(hellno)
```

```
##
## Attaching package: 'hellno'
```

```
##
## Die folgenden Objekte sind maskiert von 'package:base':
##
##      as.data.frame, data.frame
```

```
df2 <- data.frame(a=letters[1:3])
class(df2$a)
```

```
## [1] "character"
```

## Using hellno for package development

While using hellno in interactive R is nice, in fact its real strength is that it can be imported when writing packages. Once imported `stringsAsFactors=FALSE` will be the default for all uses of `data.frame()` and `as.data.frame()` within all package functions BUT NOT OUTSIDE OF IT.

Thus it provides a way to ease programming while also ensuring that package users can still choose which flavor of `stringsAsFactors` they like best.

Let us see how this works following a little example. Again, let us start with loading hellno package:

```
library(hellno)
data.frame(a=letters[1:2])$a
```

```
## [1] "a" "b"
```

As shown before, character vectors are not transformed to factor when hellno is loaded.

We unload hellno again to start clean.

```
unloadNamespace("hellno")
```

Now we install the hellnotest package from GitHub and load it. The hellnotest package imports hellno and therefore its function `hellno_df()` will not convert character vectors to factor while functions outside the packages scope will not be affected:

```
if( !("hellnotests" %in% installed.packages()) ){
  devtools::install_github("petermeissner/hellnotests")
}

library(hellnotests)
```

While all functions within the package use hellno's alternative implementations:

```
hellno_df
```

```
## function ()
## {
##      data.frame(a = letters[1:3])$a
## }
## <environment: namespace:hellnotests>
```

... and hence for them character vector conversion does not happen anymore:

```
hellno_df()
```

```
## [1] "a" "b" "c"
```

... functions outside the package (like `data.frame()` from the base package) are not affected at all:

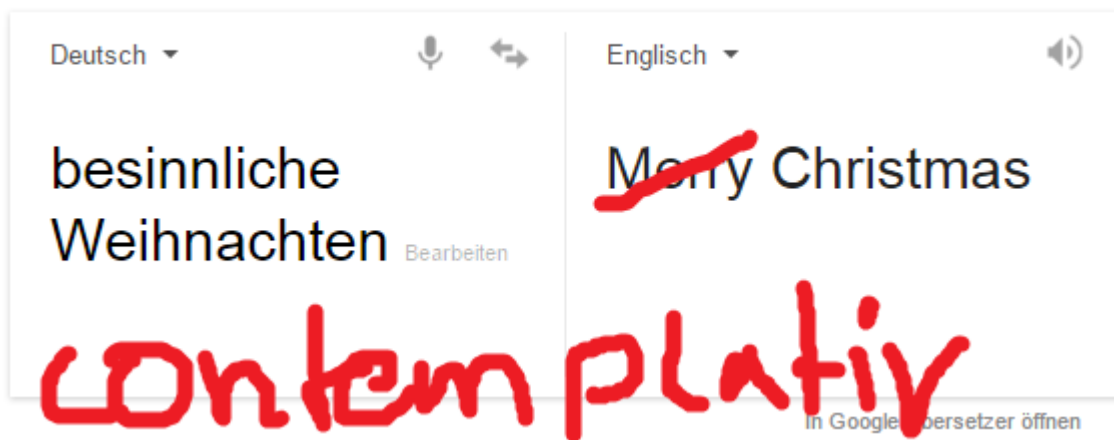
```
data.frame(a=letters[1:2])$a
```

```
## [1] a b  
## Levels: a b
```

## Summing it up

- Using `hellno` interactively makes the change of the default setting very explicit.
- Writing packages with `hellno` does not change outside behavior.
- R is Rsome.

Have fun and ...



## Discussion

If you have thoughts/ideas on the “stringsAsFactors”-problem, e.g. you do not like this solution because ... I [herewith open the issues section of the package’s GitHub repository](#) for general discussion of the theme and related stuff. I am very much interested on what you think.