Kable functions

Define Kable variants to extend possibilities

Bruno Fischer Colonimos

27 juin 2017

Contents

1 Auxiliary code **1**

2 Example data **2**
2.1 Data Frame 2
2.2 Example Matrices 2

3 Experiments with kable + matrices **3**
3.1 Analysis of Kable results+ modifications of those 5

---

1 Auxiliary code

```
# def getlib ==> Works in scripts, but not in Rmd
getlib <- function(libname) {
        if (!require(libname, character.only = TRUE)) {
                install.packages(libname)
                library(libname, character.only = TRUE)

        }
}
#  get libraries
getlib("knitr")
```

```
## Loading required package: knitr
```

```
getlib("psych")
```

```
## Loading required package: psych
```

```
getlib("reshape2")
```

```
## Loading required package: reshape2
```

```
getlib("kernlab")
```

```
## Loading required package: kernlab
```

```
##
## Attaching package: 'kernlab'
```

```
## The following object is masked from 'package:psych':
##
##     alpha
```

2 Example data

2.1 Data Frame

```
set.seed(358)

size <- 50
# one factor
fv <- factor(sample(x = c("a", "b", "c"), size = size, replace = TRUE))

# conditional factor values

cfv<- factor( sapply(X = fv, FUN = function(x) {
        if (x == "a") {y <- sample(c("x", "y"), size = 1, prob = c(0.8, 0.2))
        } else if (x == "b") {y <- sample(c("x", "y"), size = 1, prob = c(0.5, 0.5))
        } else {y <- sample(c("x", "y"), size = 1, prob = c(0.1, 0.9))
        }
        y
} ) )

# conditional numeric values
#
cvals <- sapply(X = fv, FUN = function(x) {
        if (x == "a") {y <- rnorm(n = 1, mean = 10, sd = 5)
        } else if (x == "b") {y <- rnorm(n = 1, mean = 15, sd = 5)
        } else {y <- rnorm(n = 1, mean = 20, sd = 5)
        }
        }
        )

# the dataframe
df <- data.frame(
        fv <- fv,
        cfv <- cfv,
        cvals <- cvals,
        vals = rnorm(n = size, mean = 10, sd = 5)
)
```

2.2 Example Matrices

```
# matrix without names
m0 <- matrix(1:12, nrow = 3)
m0
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    4    7   10
## [2,]    2    5    8   11
## [3,]    3    6    9   12
```

```
# matrix with column names
# mcol
```

```
mcol <- m0
colnames(mcol) <- rep("", ncol(mcol))
colnames(mcol) <- paste0("Col ", 1:ncol(mcol))
mcol
```

```
##      Col 1 Col 2 Col 3 Col 4
## [1,]     1     4     7    10
## [2,]     2     5     8    11
## [3,]     3     6     9    12
```

```
# matrix with row names
# mrow
mrow <- m0
rownames(mrow) <- rep("", nrow(mrow))
rownames(mrow) <- paste0("Row ", 1:nrow(mrow))
mrow
```

```
##       [,1] [,2] [,3] [,4]
## Row 1    1    4    7   10
## Row 2    2    5    8   11
## Row 3    3    6    9   12
```

```
# matrix with column and rox names
# mcr
mcr <- m0
colnames(mcr) <- rep("", ncol(mcr))
colnames(mcr) <- paste0("Col ", 1:ncol(mcr))
rownames(mcr) <- rep("", nrow(mcol))
rownames(mcr) <- paste0("Row ", 1:nrow(mcol))
mcr
```

```
##       Col 1 Col 2 Col 3 Col 4
## Row 1     1     4     7    10
## Row 2     2     5     8    11
## Row 3     3     6     9    12
```

3 Experiments with kable + matrices

```
kable(m0) #fails : no header
```

1	4	7	10
2	5	8	11
3	6	9	12

```
kable(mrow) #fails  : no header
```

Row 1	1	4	7	10
Row 2	2	5	8	11
Row 3	3	6	9	12

```
kable(mcol) # works ok
```

Col 1	Col 2	Col 3	Col 4
1	4	7	10
2	5	8	11
3	6	9	12

```
kable(mcr) # works ok
```

	Col 1	Col 2	Col 3	Col 4
Row 1	1	4	7	10
Row 2	2	5	8	11
Row 3	3	6	9	12

```
try(expr = kable(m0), silent = TRUE)
```

1	4	7	10
2	5	8	11
3	6	9	12

```
try
```

```
## function (expr, silent = FALSE, outFile = getOption("try.outFile",
##     default = stderr()))
## {
##     tryCatch(expr, error = function(e) {
##         call <- conditionCall(e)
##         if (!is.null(call)) {
##             if (identical(call[[1L]], quote(doTryCatch)))
##                 call <- sys.call(-4L)
##             dcall <- deparse(call)[1L]
##             prefix <- paste("Error in", dcall, ": ")
##             LONG <- 75L
##             msg <- conditionMessage(e)
##             sm <- strsplit(msg, "\n")[[1L]]
##             w <- 14L + nchar(dcall, type = "w") + nchar(sm[1L],
##                 type = "w")
##             if (is.na(w))
##                 w <- 14L + nchar(dcall, type = "b") + nchar(sm[1L],
##                   type = "b")
##             if (w > LONG)
##                 prefix <- paste0(prefix, "\n  ")
##         }
##         else prefix <- "Error : "
##         msg <- paste0(prefix, conditionMessage(e), "\n")
##         .Internal(seterrmessage(msg[1L]))
##         if (!silent && identical(getOption("show.error.messages"),
##             TRUE)) {
##             cat(msg, file = outFile)
##             .Internal(printDeferredWarnings())
##         }
##         invisible(structure(msg, class = "try-error", condition = e))
```

```
##     })
## }
## <bytecode: 0x0716a89c>
## <environment: namespace:base>
```

3.1 Analysis of Kable results+ modifications of those

```
k1 <- kable(mcol) # works ok
k1
```

Col 1	Col 2	Col 3	Col 4
1	4	7	10
2	5	8	11
3	6	9	12

```
# str(k1)
# Class 'knitr_kable'
# atomic [1:5]  | Col 1| Col 2| Col 3| Col 4|
#               |-----:|-----:|-----:|-----:
#               |     1|     4|     7|    10|
#               |     2|     5|     8|    11| ...
#
# ..- attr(*, "format")= chr "markdown"

# attr(k1, "format")

k1
```

Col 1	Col 2	Col 3	Col 4
1	4	7	10
2	5	8	11
3	6	9	12

```
k1[1]
## [1] " Col 1   Col 2   Col 3   Col 4"
k1[2]
## [1] "------  ------  ------  ------"
k1[3]
## [1] "     1       4       7      10"
k1[4]
## [1] "     2       5       8      11"
k1[5]
## [1] "     3       6       9      12"
# ajouter une ligne ? OK
```

```
k1[6] <- k1[3]
k1[7] <- "|...|...|...|...|"

k1
```

Col 1	Col 2	Col 3	Col 4
1	4	7	10
2	5	8	11
3	6	9	12
1	4	7	10
|...|...	|...|...	|	

```
# Key attributes of a kable result
#
Kclass <- class(k1)
Kformat <- attr(k1, "format")


# make a kable result
# make lines
k2 <- c( "|  a|  b|", "|---|---|", "|2500|7000|")
k2

## [1] "|  a|  b|"   "|---|---|"   "|2500|7000|"

class(k2) <- Kclass
attr(k2, "format") <-  Kformat

k2
```

a	b
2500	7000