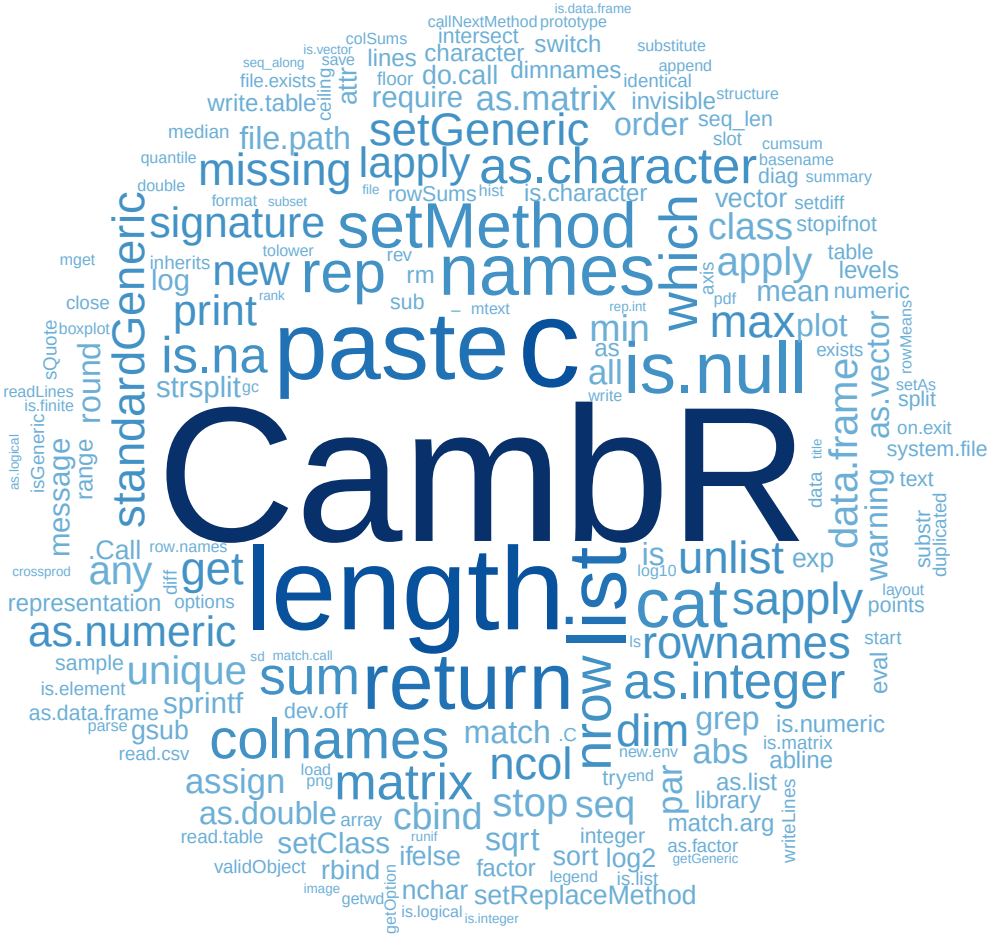
The CambR logo

Laurent Gatto and Robert Stojnic

October 29, 2012

The logo


CambR
length
paste
c
is.null
list
return
setMethod
setGeneric
standardGeneric
names
rep
new
signature
missing
lapply
as.character
which
apply
class
print
is.na
min
max
plot
unlist
cat
sapply
rownames
as.integer
colnames
matrix
dim
ncol
nrow
sum
unique
get
any
as.numeric
data.frame
as.vector
warning
stop
seq
cbind
rbind
sqrt
log2
sort
ifelse
setClass
as.double
assign
grep
abs
is.numeric
match
gsub
sprintf
as.data.frame
representation
sample
is.element
round
range
message
isGeneric
strsplit
inherits
log
mean
numeric
levels
table
vector
stopifnot
setdiff
diag
summary
basename
cumsum
slot
order
seq_len
median
quantile
file.path
write.table
require
as.matrix
invisible
identical
structure
append
substitute
intersect
switch
character
lines
do.call
dimnames
floor
file.exists
seq_along
save
is.vector
colSums
callNextMethod
prototype
is.data.frame
attr
ceiling
rowSums
hist
is.character
file
format
subset
double
close
boxplot
sQuote
readLines
is.finite
as.logical
mget
tolower
rev
rm
sub
rank
mtext
rep.int
axis
pdf
exists
all
as
write
gc
split
setAs
rowMeans
on.exit
system.file
text
substr
duplicated
data
title
.Call
row.names
crossprod
diff
options
log10
exp
points
layout
start
eval
ls
sd
match.call
dev.off
parse
read.csv
.C
new.env
try
end
load
png
array
runif
read.table
validObject
image
getwd
getOption
nchar
setReplaceMethod
is.logical
is.integer
legend
is.list
factor
integer
as.factor
getGeneric
as.list
library
match.arg
writeLines
par
abline
is.matrix

Motivation

Material for our *Advances R programming course*.
A colourful slide that says

> *This is what you are expected to know for this course.*

Material

- Get all the code from the Bioconductor project

```
svn co https://hedgehog.fhcrc.org/bioconductor/trunk/madman/Rp
```

- Extract only the `.R` and `.r` files

```
find -name "*.[rR]" | xargs cat > allR.R

$ ls -sh allR.R
36M allR.R
```

Methods - extracting relevant *words*

```
regexp <- "[a-zA-Z.][a-zA-Z0-9._]* *\\("
gregexpr(regexp, c("foo", "c (i,j,k)",
                   "setMethod()", "## comment"))
gregexpr(regexp, "foo = c (i,j,k); bar = c(l, m)")

t <- readLines("allR.R")
matches <- gregexpr(regexp, t)
length(matches) ## 1008501
k <- which(sapply(matches, function(x) x[1] != -1))
length(k)  ## 502941
```

Methods - counting words

Extract the matching pattern

```
word <- substr(t[i], start = m[j],
               stop = m[j] + attr(m, "match.length")[j]-2)
```

Trim each word by remove leading/ending \t, \n, \f, \r, \s

```
sub("^[\t\n\f\r ]*", "", word)
sub("^[\t\n\f\r ]*$", "", word)
```

Count/increment the word count if `is.function(word)`

```
if (is.function(word)) {
  if (!(word %in% names(words))) {
    words[[word]] <- 1
  } else {
    words[[word]] <- words[[word]] + 1
  }
}
```

The output

is a fun/freq `data.frame`

```
> head(out)
        fun  freq
2         c 38336
11   length 33491
100   paste 22251
17     list 15721
25   return 15236
26     stop 15041
...
```

that needs a bit of post-processing...

Post-processing

- Take the `sqrt(freq)`
- Get rid the embarrassing high-freq function `stop`.
- Add `CambR` with a `sqrt(freq)` of 300.

Plotting

```
library("wordcloud")
library("RColorBrewer")
pal <- brewer.pal(9,"Blues")[5:9]
wordcloud(out$fun, out$freq, c(6,.1), max.words = 200,
          random.order = FALSE, colors = pal)
```

https://github.com/lgatto/CambRlogo