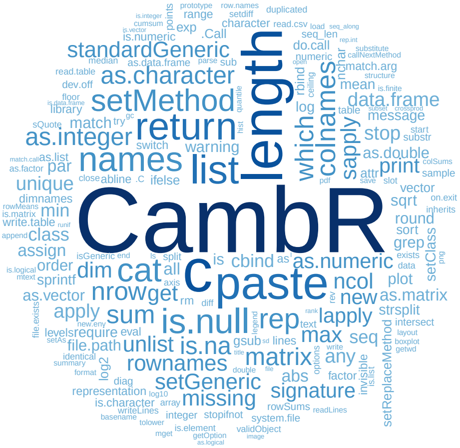
The CambR logo

Laurent Gatto and Robert Stojnic

October 29, 2012

The logo


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

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
$ wc allR.R
1008315  3351122 37166763 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

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
  }
}
```

(with some error catching not shown here.)

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)
```

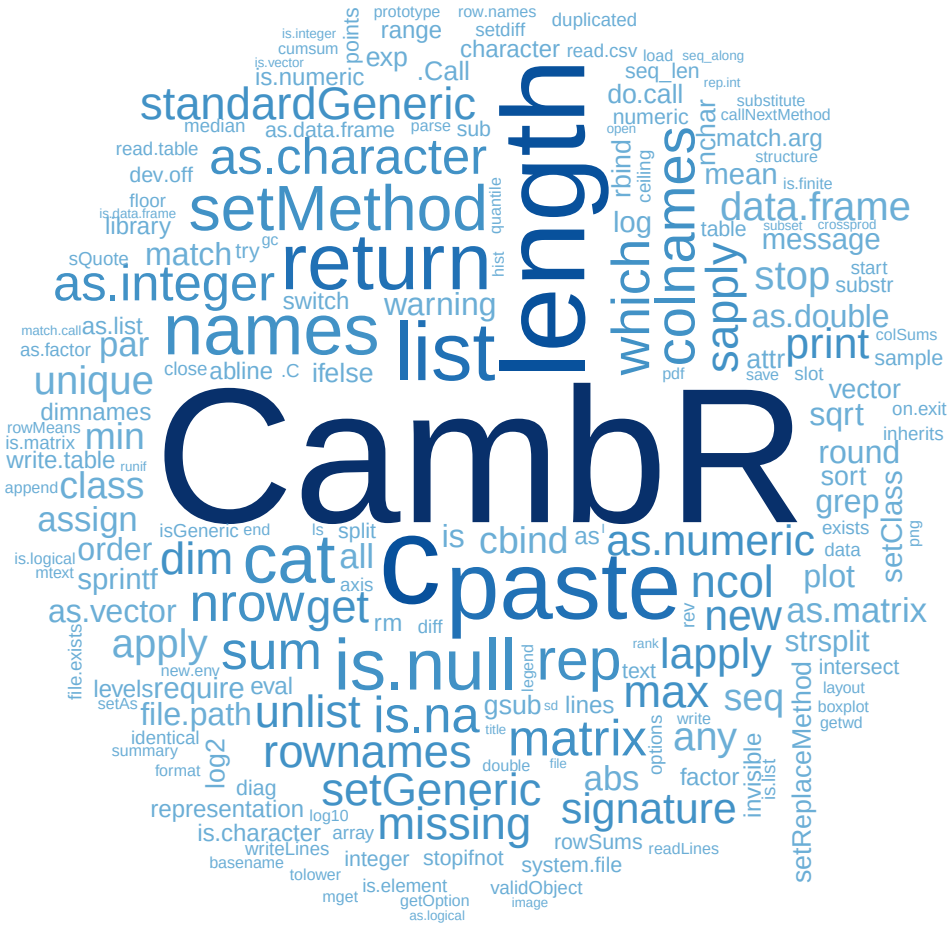


Figure: https://github.com/lgatto/CambRlogo