

# sprof reference

August 29, 2013

**Encoding** UTF-8

**Type** Package

**Title** Profiling, timing and optimisation utilities

**Version** 0.1-0

**Date** 2013-08-28

**Author** Günther Sawitzki

**Maintainer** Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**Description** Support utilities for profiling and dynamic code analysis.

**License** GPL-2 | GPL-3

**Suggests** wordcloud, timeit, RColorBrewer

**URL** <http://sintro.r-forge.r-project.org>

**ByteCompile** FALSE

**KeepSource** TRUE

**BuildVignettes** FALSE

## R topics documented:

sprof-package . . . . .	2
adjacency . . . . .	3
asfactormodel . . . . .	4
barplot_s . . . . .	5
edgedf . . . . .	7
list.as.matrix . . . . .	8
nodepackage . . . . .	9
nodescloud . . . . .	10
nodesprofile . . . . .	11
nodesrunlength . . . . .	12
plot.sprof . . . . .	13
plot_nodes . . . . .	14
plot_profiles . . . . .	16
plot_stacks . . . . .	18

print.sprof . . . . .	19
print_profiles . . . . .	20
profiles_matrix . . . . .	20
readRprof . . . . .	21
rkindex . . . . .	23
roots_sprof . . . . .	24
rrle . . . . .	24
rrleb . . . . .	25
sampleRprof . . . . .	26
shownodes . . . . .	27
sprof01lm . . . . .	28
stacksasfactor . . . . .	29
stackstoadj . . . . .	30
str_prof . . . . .	30
summary.sprof . . . . .	31
summary_terminals . . . . .	32
trimstacks . . . . .	33
updateRprof . . . . .	33
writeRprof . . . . .	34
<b>Index</b>	<b>36</b>

---

sprof-package	<i>sprof: Analysis of R profiles</i>
---------------	--------------------------------------

---

**Description**

Profiling, timing and optimisation utilities

**Details**

Package: sprof  
Type: Package  
Version: 0.1-0  
License: GPL-2 | GPL-3

[readRprof\(\)](#) reads a profile file from [Rprof\(\)](#) or other profilers and returns a composite structure of class sprof. The basic components of sprof are (conceptually) four data frames

- info      general information and summaries
- nodes    node specific information
- stacks   node specific information. Stacks are random snapshots from the program execution, possibly including side information such as traces of the memory management.
- profiles   collected records of a profile, encoded as references to stacks

The data structure is subject to change. For more details, see the documentation of [readRprof\(\)](#).  
To create a profile on the fly, use [sampleRprof](#).  
To import profile information written by [Rprof](#) or other profilers, use [readRprof](#).

For sprof, the usual access functions are supported.

### Note

Version 0.0-6 is a clean-up version. Recommendations/requests for the interface definition are requested at this point.

### Author(s)

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

### Examples

```
## Not run:
data(sprof01lm)
str(sprof01lm)
plot(sprof01lm)

## End(Not run)
```

---

adjacency

*sprof to adjacency matrix*


---

### Description

convert node information from a sprof structure to adjacency matrix.

### Usage

```
adjacency(sprof, keep.names = TRUE, rmzero=TRUE, no.name="<nn>")
```

### Arguments

sprof	a sprof structure.
keep.names	boolean. Copy node names as row- and column names.
rmzero	boolean. Remove nodes with no edges.
no.name	If not null: replacement for empty strings as name.

### Value

a correspondence matrix

### Author(s)

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

### References

See the vignette of package sprof.

### Examples

```
data(sprof01lm)
adjacency(sprof01lm)
```

---

asfactormodel	<i>Convert to factor, like model</i>
---------------	--------------------------------------

---

## Description

Convert factor-like entries in a data structure to factor, with factor as in factormodel.

## Usage

```
asfactormodel(x, factormodel)
```

## Arguments

x	A data structure. Currently only integer vectors or lists of vectors are supported.
factormodel	A data structure to serve as model for the factor specification. Currently, a factor or a vector of type character.

## Value

a data structure of same type as x, with numeric vectors converted to factors.

## Note

This functionality could ge to R base function [factor](#).

Should be exxtended to cover a wide range of data structures, and identify substructures for conversion.

## Author(s)

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

## Examples

```
data(sprof01lm)

xf <- asfactormodel(sprof01lm$stacks$nodes, sprof01lm$nodes$name)

# To get a charcter listing, use
sapply(xf, function (x) {as.character(x)} )
```

## Description

Creates a sorted bar plot with vertical or horizontal bars.

## Usage

```
barplot_s(height,
horiz = FALSE,
sort_by,
decreasing = TRUE,
lowtrim, hightrim, trimlegend = TRUE,
col, coli, colfun,
main, ...)
```

## Arguments

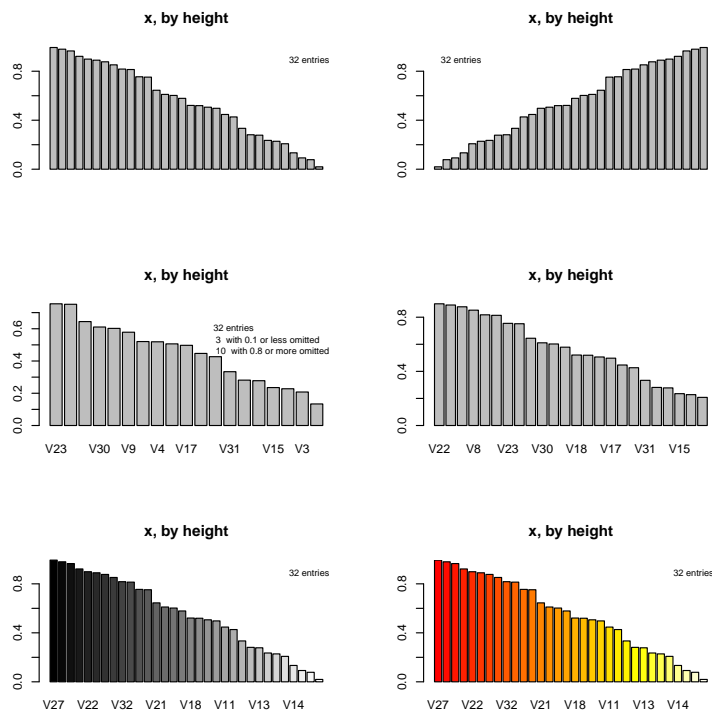
height	either a vector or matrix of values describing the bars which make up the plot. See <a href="#">barplot</a> .
horiz	boolean. Arrange bars horizontally.
sort_by	a variable to sort by. Defaults to height.
decreasing	boolean. Sorting direction.
lowtrim	A optional lower trim value. Observations with sort_by values up to lowtrim are discarded.
hightrim	A optional upper trim value. Observations with sort_by values from lowtrim are discarded.
trimlegend	Boolean. Show a legend about trimmed values.
col	a vector of colors for the bars or bar components. By default, grey is used if height is a vector, and a gamma-corrected grey palette if height is a matrix.
coli	An index into the col table, based on original sorting.
colfun	A function to generate a col palette. grey is rescaled to 1..n.
main	overall title for the plot
...	Passed to barplot

## Details

If no names are supplied, they are generated in the form x...

You man want to adjust the scale of the bar labels by using an additional argument such as `cex.names = 0.5`.

More to come. Plots are from this collection:



## Value

Invisible: A data frame with components

x	height
perm	the permutation applied
coli	the colour index applied
col	optional: the colours selected

## Note

Part of this could go to the R base function [barplot](#).

## Author(s)

Günther Sawitzki <[gsawitzki@users.r-forge.r-project.org](mailto:gsawitzki@users.r-forge.r-project.org)>

## See Also

[barplot](#)

## Examples

```
x <- runif(100)

barplot_s(x)

x <- rnorm(100)
```

```
barplot__s(x, colfun=heat.colors, lowtrim=-1)

rm(x)
```

---

**edgedf***Expand adjacency information to an edge table*

---

## Description

Expand adjacency information from an adjacency matrix or a sprof data structure to an edge table

## Usage

```
edgedf(data, counts = TRUE, na.rm = TRUE, no.name="<nn>")
```

## Arguments

data	an adjacency matrix or a sprof data structure.
counts	include a column of counts
na.rm	remove lines with a count NA.
no.name	If not null: replacement for empty strings as name.

## Details

The adjacency matrix is flattened. Lines with a count zero are eliminated.

## Value

A data frame.

from	Name of from node.
to	Name of to node.
count	optional. Frequencies of edges.

## Author(s)

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

## Examples

```
data(sprof01lm)
edgedf(sprof01lm)
```

---

`list.as.matrix`*Convert list to matrix*

---

**Description**

Convert list to matrix. List entries go to matrix columns, filled for equal length.

**Usage**

```
list.as.matrix(x, byrow=FALSE, filler = NA)
```

**Arguments**

<code>x</code>	a list of numeric vectors.
<code>byrow</code>	boolean. Arrange list entries as rows. Default is to use columns.
<code>filler</code>	a value to be used as a filler

**Value**

A matrix with the values from `x`, filled to matrix shape.

**To do**

Arguments should be as `as.matrix`.

A corresponding `list.as.data.frame` should be added for mixed types.

Name synchronisation/preservations should be added.

**Note**

This could go to `as.matrix()`.

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**See Also**

[as.matrix](#).

**Examples**

```
x <- list(x1=c(1,2,3),x2=3, x3=4:8)
list.as.matrix(x)
list.as.matrix(x,filler=0)
```



---

`nodepackage`*Find a package that may contain a node*

---

## Description

[getAnywhere\(\)](#) is used to look up `x`, and the package or namespace information is used to give a source package.

## Usage

```
nodepackage(x)
```

## Arguments

`x` a character string or name, or a vector.

## Details

There is no indication whether the information is from a namespace or from a package information.

No indication is given if multiple hits are encountered.

The information is based on the run time environment of this function. This may be different from the environment the object is taken from.

See the help information for [getAnywhere\(\)](#) for more warnings.

## Value

a character string or a vector of strings with the package names.

## Author(s)

Günther Sawitzki <[gsawitzki@users.r-forge.r-project.org](mailto:gsawitzki@users.r-forge.r-project.org)>

## See Also

[getAnywhere\(\)](#)

## Examples

```
nodepackage("getAnywhere")
```



**Note**

See the vignette of package sprof.

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**Examples**

```
## Not run:
data(sprof01lm)
nodescloud(sprof01lm)

## End(Not run)
```

---

nodesprofile

*Run length matrix*


---

**Description**

Extracts run length information from a sprof data structure with profiling information.

**Usage**

```
nodesprofile(sprof)
```

**Arguments**

sprof                      a sprof data structure with profiling information.

**Details**

Run length counts by node, level and run length.

**Value**

counts[ node, level, run length] .

**Note**

This expands a sparse matrix to full. Avoid to use it.

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**Examples**

```
data(sprof01lm)
str(nodesprofile(sprof01lm) )
```

---

nodesrunlength	<i>Marginal information for run length from profile</i>
----------------	---

---

## Description

Run length count, by node and run length, from profile.

## Usage

```
nodesrunlength(sprof, clean=TRUE)
```

## Arguments

sprof	a sprof data structure with profiling information.
clean	boolean.

## Details

If clean=TRUE, zero results are removed and the nodes are sorted by average time.

If clean=TRUE, zero results and sorting are preserved. Trailing nodes with zero count may have been lost in the process, and are added.

## Value

A matrix count[node, run length] with a column giving the number of runs by run length and three additional columns

nr_runs	sum of counts over all run lengths.
total_time	sum of count*run length
avg_time	total_time / count

## Author(s)

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

## Examples

```
data(sprof01lm)
nodesrunlength(sprof01lm)
```

plot.sprof

*plot for profiles*

## Description

plot a plot for the output of class `scode`.

## Usage

```
## S3 method for class 'sprof'
plot(x, ...)
```

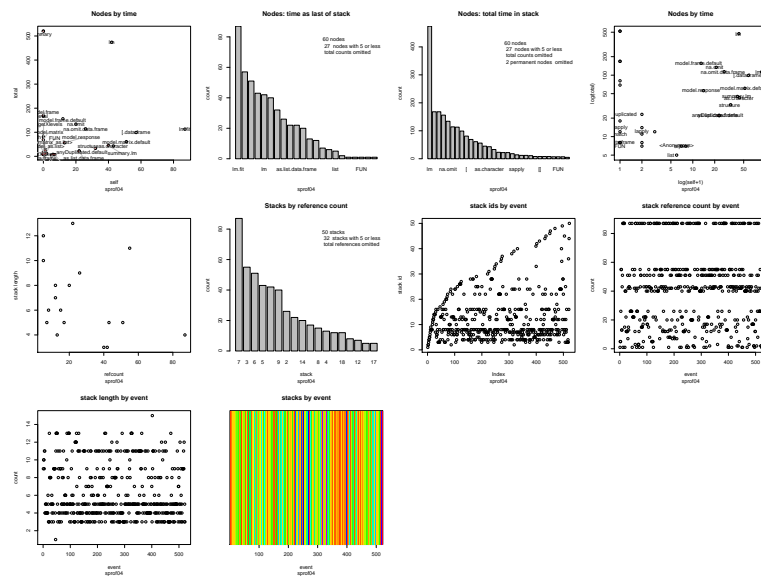
## Arguments

`x` A data structure as returned by [readRprof](#).

`...` further arguments passed to or from other methods.

## Details

These displays may be outdated. Please run the examples. Plots are from this collection:



## Value

subject to change

## Note

See the vignette for in-context explanations.

Displays of the graph structure are given in the vignette.

The `plot.sprof` method for `sprof` objects concatenates three plot functions. Using the plot functions one by one allows better control and will be preferred. [shownodes](#) may be a sufficient summary.

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**References**

<http://sintro.r-forge.r-project.org/>

**See Also**

[shownodes](#)

[summaryRprof](#)

[plot\\_nodes](#) [plot\\_stacks](#) [plot\\_profiles](#)

**Examples**

```
data(sprof01lm)
```

```
oldpar <- par(mfrow=c(3,4))
plot.sprof(sprof01lm)
par(oldpar)
```

---

plot\_nodes

*Plot profiling information on node level.*

---

**Description**

Various plots of a profile.

**Usage**

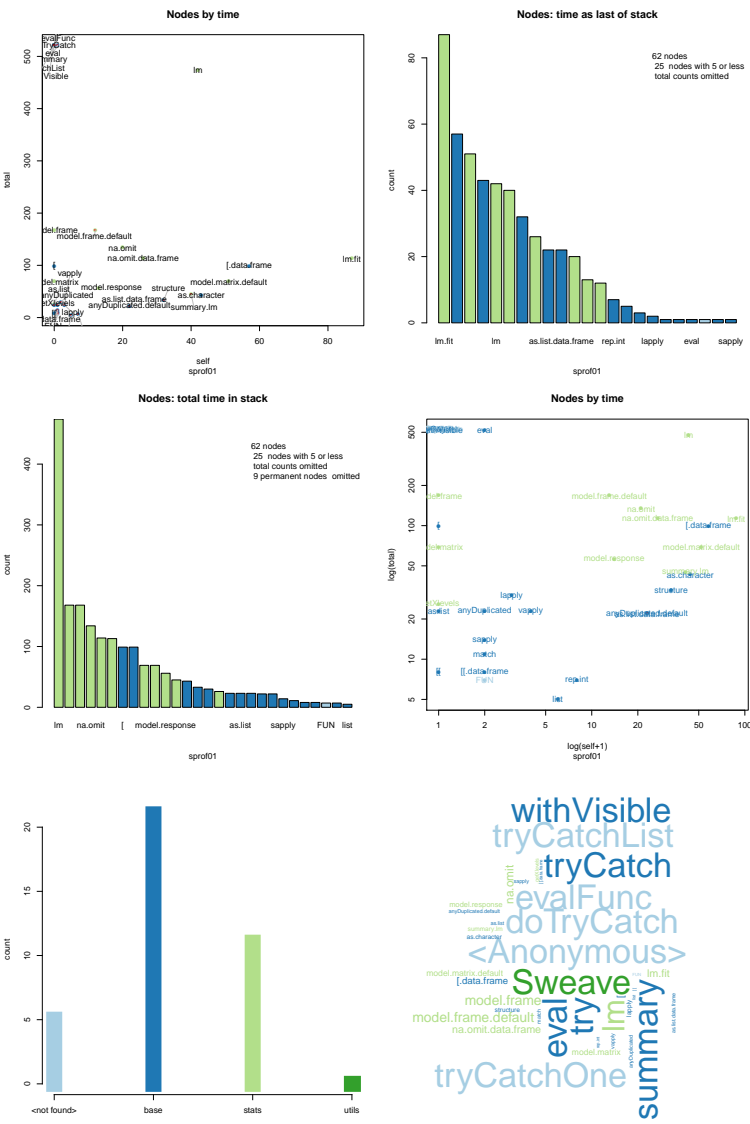
```
plot_nodes(x, which = c(1L, 2L, 3L, 4L), col = NULL,
  ask = prod(par("mfcol")) < length(which) && dev.interactive(),
  src = NULL, mincount = 5, horiz=FALSE, ...)
```

**Arguments**

x	preferably a sprof object. Other data structures may be extended
which	Selector of plots to show.
col	Colour table
ask	boolean. Ask for a new page?
src	String to be used as source identifier.
mincount	minimum total frequency count for node to be shown in barcharts.
horiz	draw horizontal bar plots.
...	passed.

Details

These displays may be outdated. Please run the examples. Plots are from this collection:



Value

To come.

Note

See the vignette of package sproft.

Author(s)

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

## References

See the vignette of package sprof.

## See Also

[plot.sprof](#)

## Examples

```
data(sprof01lm)
oldpar <- par(mfrow=c(2,2))
plot_nodes(sprof01lm)
par(oldpar)
```

---

plot_profiles	<i>Plot profiling information on profile level.</i>
---------------	---

---

## Description

Various plots of a profile.

## Usage

```
plot_profiles(x, which = c(1L, 2L, 3L, 4L), col,
ask = prod(par("mfcol")) < length(which) && dev.interactive(),
src = NULL, ...)
```

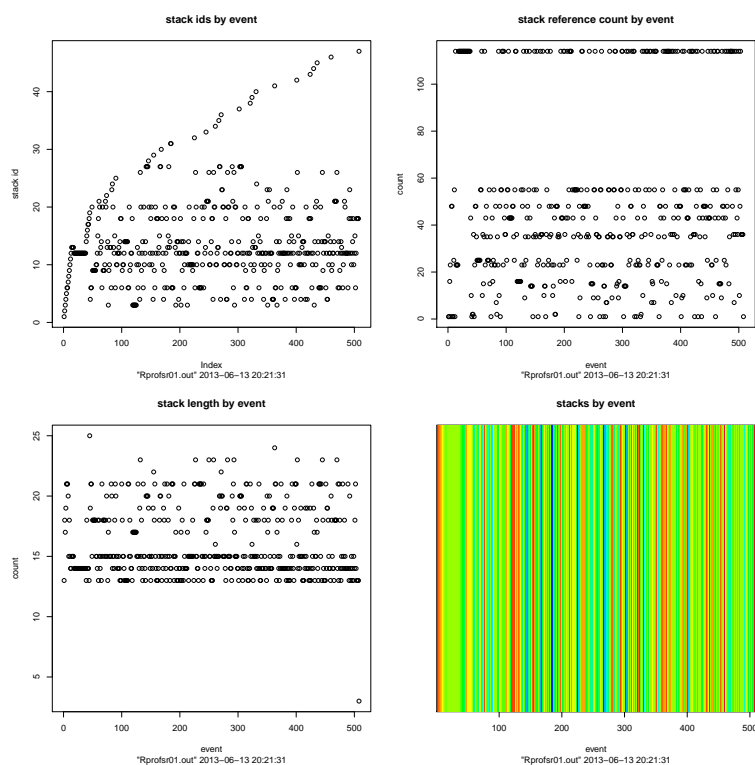
## Arguments

x	preferably a sprof object. Other data structures may be extended
which	Selector of plots to show.
col	Colour table
ask	boolean. Ask for a new page?
src	String to be used as source identifier.
...	passed.

## Details

Plots are from this collection:





## Note

See the vignette of package `sprof`.

## Author(s)

Günther Sawitzki <[gsawitzki@users.r-forge.r-project.org](mailto:gsawitzki@users.r-forge.r-project.org)>

## References

See the vignette of package `sprof`.

## See Also

[plot.sprof](#).

## Examples

```
data(sprof01lm)
oldpar <- par(mfrow=c(2,2))
plot_profiles(sprof01lm)
par(oldpar)
```

---

plot\_stacks

*Plot profiling information on stack level.*


---

## Description

Various plots of a profile.

## Usage

```
plot_stacks(x, which = c(1L, 2L),
ask = prod(par("mfcol")) < length(which) && dev.interactive(),
src = NULL, mincount = 5, horiz = FALSE, ...)
```

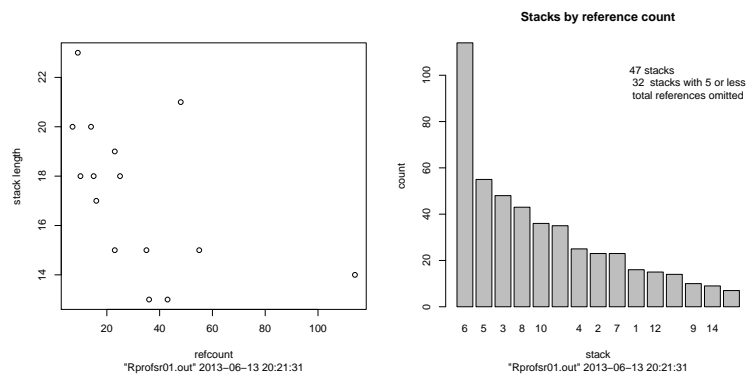
## Arguments

x	preferably a sprof object. Other data structures may be extended
which	Selector of plots to show.
ask	boolean. Ask for a new page?
src	String to be used as source identifier.
mincount	minimum total frequency count for stack to be shown in barcharts.
horiz	draw horizontal bar plots.
...	passed.

## Details

Note: these figures may be outdated. Please run the examples.

Plots are from this collection:



## Value

To come.

## Note

See the vignette of package sprof.

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**References**

See the vignette of package sprof.

**See Also**

[plot.sprof](#).

**Examples**

```
data(sprof01lm)
oldpar <- par(mfrow=c(2,2))
plot_stacks(sprof01lm)
par(oldpar)
```

---

print.sprof

*print for profiles*


---

**Description**

Print a print for the output of class scode.

**Usage**

```
## S3 method for class 'sprof'
print(x, ...)
```

**Arguments**

x	A data structure as returned by <a href="#">readRprof</a> .
...	further arguments passed to or from other methods.

**Value**

None.

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**References**

<http://sintro.r-forge.r-project.org/>

**See Also**

[summaryRprof](#) [plot.sprof](#)

**Examples**

```
data(sprof01lm)
print(sprof01lm)
```

---

print_profiles	<i>Print profile information</i>
----------------	----------------------------------

---

**Description**

Print profile information.

**Usage**

```
print_profiles(x)
```

**Arguments**

x                      a sprof data structure.

**Value**

none

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**Examples**

```
data(sprof01lm)
print_profiles(sprof01lm)
```

---

profiles_matrix	<i>Extract a node incidence matrix from profile information.</i>
-----------------	--

---

**Description**

Extract a node incidence matrix from profile information.

**Usage**

```
profiles_matrix(x)
```

**Arguments**

x                      an sprof data structure.

**Value**

an incidence matrix, NA filled.

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**Examples**

```
data(sprof01lm)
smat <- profiles_matrix(sprof01lm)
image(smat)
```

---

readRprof

Read [Rprof](#) Output files and Stack Logs

---

**Description**

Read a log of stack entries, such as the output of the [Rprof](#) function, and generate a more accessible representation.

**Usage**

```
readRprof(filename = "Rprof.out", chunksize = 5000,
interval = 0.02,
head = c("auto", "none", "Rprofmem"),
id = NULL)
```

**Arguments**

filename	Name of a file produced by <code>Rprof()</code> .
chunksize	Number of lines to read at a time.
interval	Real number: time interval between samples, in s. Defaults to 0.02s for consistency with <code>Rprof</code> , but shorter times should be used.
head	<code>c("auto", "none", "Rprofmem")</code> to interpret control information as provided by <code>Rprof</code> or <code>Rprofmem</code> . See details.
id	An optional identification string. Defaults to filename and date.

**Details**

This function reads a log file of stacks, one stack snapshot per line, stack entries separated by space.

As profiling output file could be very large, it is read in blocks of `chunksize` lines. Increasing `chunksize` will make the function run faster if sufficient memory is available.

(This data structure is subject to change.)

The input format is controlled by the `head` argument. Format "auto" tries to detect control lines as interspersed by `Rprof`. These lines are not included in the output.

"none" ignores all control information and includes these lines as strange stacks.

"Rprofmem" isolates headers as provided by `Rprofmem`. "Rprofmem" new page entries are encoded as malloc requests with length 0.

**Value**

This data structure is subject to change.

Temporarily: A list with components

info	Summary information.
nodes	A vector of node names. This may include stray entries from interspersed lines.
stacks	Stacks, represented as reference list to nodes, and stack frequencies.
profiles	Recorded data, as reference to stacks, and possibly additional data per reference.

nodes is (conceptually) a data frame with entries

name	node name
self.time	nr of events with node as terminal leaf
self.time	proportion of self.time with node as terminal leaf
total.time	nr of events with node in stack
self.time	proportion of total time with node as terminal leaf
nr.runs	number of runs, over all run lengths and levels
nr.runs	average of run length, over all run lengths and levels
icol	current colour index

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>, based on the code of [summaryRprof](#)

**References**

<http://sintro.r-forge.r-project.org/>

**See Also**

[summaryRprof](#)

[summaryRprof](#)

[flatProfile](#) in library(proftools).

[parse\\_rprof](#) in library(profr).

The chapter on “Tidying and profiling R code” in “Writing R Extensions” (see the ‘doc/manual’ subdirectory of the R source tree).

[Rprof](#) is a sampling profiler.

[tracemem](#) traces copying of an object via the C function duplicate.

[Rprofmem](#) is a non-sampling memory-use profiler.

<http://developer.r-project.org/memory-profiling.html>

**Examples**

```
## Not run:
## Rprof() is not available on all platforms
profintrval <- 0.001
simruns <- 100

n <- 10000
x <- runif(n)
```

```

y0 <- 2 + 3 * x

sinknull <- textConnection(NULL, "w"); sink(sinknull)
Rprof(tmp <- tempfile(), interval = profinterval)
for (i in 1:simruns) {y <- y0 + rnorm(n); xxx<- summary(lm(y~x))}
Rprof()

Rprof_out <- readRprof(tmp)

unlink(tmp)
sink(); close(sinknull)

str(Rprof_out)

## End(Not run)

```

---

rkindex	<i>Index, based on rank.</i>
---------	------------------------------

---

## Description

Convert  $x$  to an index in  $1 \dots \text{maxindex}$

## Usage

```
rkindex(x, maxindex = length(x), pwr = 1, ties.method = "random")
```

## Arguments

<code>x</code>	A vector of data to covert.
<code>maxindex</code>	Maximum for result
<code>pwr</code>	See details.
<code>ties.method</code>	passed to rank.

## Details

$x$  is transformed to a rank scale, using `ties.method`. It is then rescaled to  $[0,1]$ , and (optionally) a power transformation is applied. In visualisation terms, this is a gamma correction. The result is rescaled to  $1 \dots \text{maxindex}$ .

## Value

A vector of rescaled values.

## Author(s)

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

---

roots_sprof	<i>Root nodes list for sprof</i>
-------------	----------------------------------

---

**Description**

Return a list of root nodes of all stacks.

**Usage**

```
roots_sprof(sprof, stacks)
```

**Arguments**

sprof	a sprof data structure, if available
stacks	an optional list of stacks as references to nodes

**Value**

A vector of unique entries as root of stacks. If sprof is provided, nmaes are imported from the node name table of sprof

**To do**

stacks should support any reasonable representation of the stacks, and preserve format.

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**Examples**

```
data(sprof01lm)
roots_sprof(sprof01lm)
trimmed <- trimstacks(sprof01lm, trimnode= "summary")
roots_sprof(sprof01lm, stacks=trimmed)
```

---

rrle	<i>Recursive run length encoding.</i>
------	---------------------------------------

---

**Description**

Encode a matrix as run-length, top down. Encoding respects previous runs, e.g line 2 encodes runs in each run of line 1.

**Usage**

```
rrle(x, collapseNA =FALSE)
```



**Arguments**

`x` a matrix.

`collapseNA` boolean. Collapse runs of NA.

**Details**

By default, different NA data are not considered equal. `collapseNA` collapses runs of NA in the result. For recursion however they are treated as singular data, not as runs. This may need discussion.

**Value**

list of run length encoded lines

**Note**

This could go to `rle` in package `base`.

The data structure used for `rle` would be better represented as a `data.frame`.

Run length and other compressions might be implemented in `data.frames` by default.

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**See Also**

[rrleb](#)

**Examples**

```
x <- matrix(c(
  1,1,1,2,2,
  3,3,4,4,4,
  5,5,6,6,7,
  8,9,9,0,0
),nrow=4, ncol =5, byrow=TRUE)
xrrle <- rrle(x)
xrrle

t(sapply(xrrle, inverse.rle))
```

---

rrleb

---

Recursive run length encoding bottom up.

---

**Description**

Encode a matrix as run-length, bottom up. Encoding respects previous runs, e.g line `n-1` encodes runs in each run of line `n`.

May be removed.

**Usage**

```
rrleb(x)
```

**Arguments**

`x` a matrix.

**Value**

list of run length encoded lines

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**See Also**

[rle](#),

**Examples**

```
x <- matrix(c(
  3,3,4,4,4,
  5,5,6,6,7,
  8,9,9,0,0,
  1,1,1,2,2
),nrow=4, ncol =5, byrow=TRUE)
xrrleb <- rrleb(x)
xrrleb

t(sapply(xrrleb, inverse.rle))
```

---

sampleRprof

*Get a sample profile*

---

**Description**

Get a sample profile and return it as a sprof data structure.

**Usage**

```
sampleRprof(expr, runs = NULL, gcFirst = TRUE, interval = 0.001, ...)
```

**Arguments**

<code>expr</code>	an expression to be profiled.
<code>runs</code>	nr of runs to profile.
<code>gcFirst</code>	boolean. Bracket the total simulation with calls to <code>GC()</code> . If <code>TRUE</code> , the garbage collection information will be reported as components <code>gcin</code> , <code>codegcout</code> .
<code>interval</code>	Real: time interval between samples, in s.
<code>...</code>	additional parameters, passed to <code>Rprof</code>

**Value**

A list of type sprof

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**References**

<http://sintro.r-forge.r-project.org/>

**Examples**

```
## Not run:  
res_lm <- sampleRprof(for (i in 1:1000) yy<- lm(runif(1000)~rnorm(1000)), runs=100)  
## End(Not run)
```

---

shownodes	<i>Show node information from a profile</i>
-----------	---

---

**Description**

Plot node information from a profile in various plots.

**Usage**

```
shownodes(sprof, col)
```

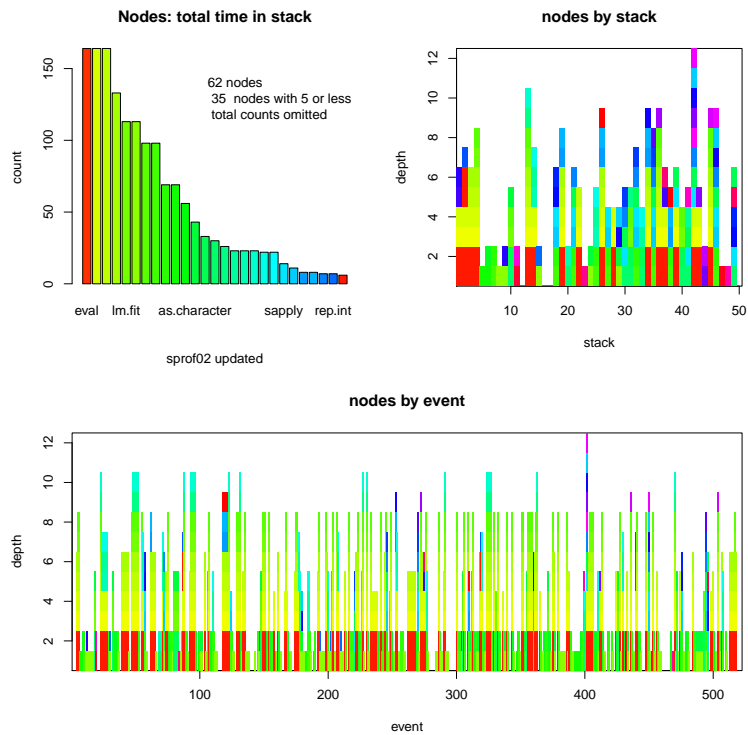
**Arguments**

sprof	A data structure as returned by <a href="#">readRprof</a> .
col	A colour palette for the plots.

**Details**

Note: these figures may be outdated. Please run the examples.

Plots are from this collection:



## Value

Used for the side effect of showing the plots.

## Author(s)

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

## Examples

```
## Not run:
data(sprof011m)
shownodes(sprof011m)

## End(Not run)
```

---

sprof011m

*sprof sample data*

---

## Description

An example data set for the functions in package sprof.

## Usage

```
data(sprof011m)
```

**Format**

The format is: A List of 4 \$ info : 'data.frame': 1 obs. of 8 variables: \$ nodes : 'data.frame': 62 obs. of 5 variables: \$ stacks : 'data.frame': 50 obs. of 7 variables: \$ profiles: List of 4

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**References**

See the vignette of package sprof.

**Examples**

```
data(sprof01lm)
str(sprof01lm)
plot(sprof01lm)
```

---

stacksasfactor	<i>Represent stacks as factor</i>
----------------	-----------------------------------

---

**Description**

Represent stacks as factor, using the node information of the profile record

**Usage**

```
stacksasfactor(sprof, sel, events)
```

**Arguments**

sprof	a sprof data structure.
sel	Indices of stacks to convert.
events	Events, stacks of which to convert

**Details**

sel and events are exclusive.

If none is selected, all stacks are given.

**Value**

A vector, or a list of vectors, representing the selected stacks as factors.

**Note**

event based selection should report event numbers.

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**Examples**

```
data(sprof01lm)
stacksasfactor(sprof01lm, 1:3)

stacksasfactor(sprof01lm, events=10)
```

---

stackstoadj	<i>Stacks to adjacency matrix</i>
-------------	-----------------------------------

---

**Description**

convert stack information to adjacency matrix

**Usage**

```
stackstoadj(xstacks, xfreq, maxnode)
```

**Arguments**

xstacks	list of stack ids
xfreq	vector of frequencies or weights
maxnode	maximum of nodes (maybe higher then in stacks)

**Value**

the adjacency matrix

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**Examples**

```
#
```

---

str_prof	<i>str for sprof objects</i>
----------	------------------------------

---

**Description**

str for sprof objects

**Usage**

```
str_prof(x)
```

**Arguments**

x	an sprof object
---	-----------------

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**Examples**

```
data(sprof01lm)
str__prof(sprof01lm)
```

---

summary.sprof	<i>Summary for profiles</i>
---------------	-----------------------------

---

**Description**

Print a summary for the output of class scode.

**Usage**

```
## S3 method for class 'sprof'
summary(object, ...)
```

**Arguments**

object	A data structure as returned by <a href="#">readRprof</a> .
...	further arguments passed to or from other methods.

**Value**

None.

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**References**

<http://sintro.r-forge.r-project.org/>

**See Also**

[summaryRprof](#)

**Examples**

```
## Not run:
## Rprof() is not available on all platforms
profinterval <- 0.001
simruns <- 100

n <- 10000
x <- runif(n)
y0 <- 2 + 3 * x
```

```

sinknull <- textConnection(NULL, "w"); sink(sinknull)
Rprof(tmp <- tempfile(), interval = profinterval)
for (i in 1:simruns) {y <- y0 + rnorm(n); xxx<- summary(lm(y~x))}
Rprof()

Rprof_out <- readProf(tmp)

unlink(tmp)
sink(); close(sinknull)

summary(Rprof_out)

### End(Not run)

```

---

summary_terminals	<i>Tabulate leaf nodes</i>
-------------------	----------------------------

---

## Description

Tabulate leaf nodes

## Usage

```
summary_terminals(x)
```

## Arguments

x                      an sprof data structure.

## Value

A table of frequencies, bystack.

## Author(s)

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

## Examples

```

data(sprof01lm)
summary_terminals(sprof01lm)

```



---

trimstacks	<i>Trim sprof stacks.</i>
------------	---------------------------

---

**Description**

Trim sprof stack information by top level or by node.

**Usage**

```
trimstacks(sprof, level, trimnode)
```

**Arguments**

sprof	a sprof data structure, if available, or a stacks\$nodes information.
level	Level to cut off.
trimnode	A node considered end of scaffold. Entries up to and including this node are trimmed. Nodes can be marked as index, or as node name.)

**Details**

The level information is evaluated first, and then the node information is evaluated.

**Value**

A data structure corresponding to stacks\$nodes, with the trimmed parts cut off. This may contain NULL entries.

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**Examples**

```
data(sprof01lm)
trimmed <- trimstacks(sprof01lm, trimnode="summary")
sprof01lm$stacks$nodes <- trimmed #now the refereces are update, and graphs will work
sprof02lm <- updateRprof(sprof01lm) #now the info is updated, and summeries will be consistent.
```

---

updateRprof	<i>Update statistics and tables in a sprof obejct</i>
-------------	---

---

**Description**

Synchronize information from profiles and stack tables, and update statistics.

**Usage**

```
updateRprof(sprof, id)
```

**Arguments**

sprof                    A data structure as returned by [readRprof](#).  
 id                        optional. A replacement for the info\$id string .

**Value**

An updated sprof data structure.

**Note**

See the vignette of package sprof.

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

---

 writeRprof

*Write profile data*


---

**Description**

Write a profile data file from a sprof data structure.

**Usage**

```
writeRprof(sprof, filename = "Rprof.Out")
```

**Arguments**

sprof                    a data structure from package sprof  
 filename                The file to be used for exporting the profiling results.

**Details**

writeRprof only writes the stack entries for the profile.

This can be used to export information after preprocessing with sprof to some package designed for Rprof output.

**Value**

An invisible list with the profile entries, headers removed.

**Note**

See the vignette for in-context explanations.

Displays of the graph structure are given in the vignette.

**Author(s)**

Günther Sawitzki <gsawitzki@users.r-forge.r-project.org>

**References**

<http://sintro.r-forge.r-project.org/>

**Examples**

```
## Not run:  
data(sprof01lm)  
writeRprof(sprof01lm)  
  
## End(Not run)
```

# Index

- \*Topic **datasets**
  - sprof01lm, 28
- \*Topic **hplot**
  - barplot\_s, 5
  - nodescloud, 10
  - plot.sprof, 13
  - plot\_nodes, 14
  - plot\_profiles, 16
  - plot\_stacks, 18
  - shownodes, 27
- \*Topic **list**
  - list.as.matrix, 8
- \*Topic **manip**
  - edgedf, 7
  - nodesprofile, 11
  - nodesrunlength, 12
  - rrle, 24
  - rrleb, 25
  - stackstoadj, 30
- \*Topic **matrix**
  - list.as.matrix, 8
  - rrle, 24
  - rrleb, 25
- \*Topic **methods**
  - summary.sprof, 31
- \*Topic **misc**
  - asfactormodel, 4
  - roots\_sprof, 24
  - trimstacks, 33
- \*Topic **print**
  - print.sprof, 19
  - print\_profiles, 20
- \*Topic **utilities**
  - adjacency, 3
  - list.as.matrix, 8
  - plot.sprof, 13
  - readRprof, 21
  - rrle, 24
  - rrleb, 25
  - sampleRprof, 26
  - sprof-package, 2
  - stacksasfactor, 29
  - stackstoadj, 30
  - str\_prof, 30
  - summary.sprof, 31
  - summary\_terminals, 32
  - updateRprof, 33
  - writeRprof, 34
- \*Topic **util**
  - nodepackage, 9
  - profiles\_matrix, 20
  - rkindex, 23
- adjacency, 3
- as.matrix, 8
- asfactormodel, 4
- barplot, 5, 6
- barplot\_s, 5
- edgedf, 7
- factor, 4
- flatProfile, 22
- getAnywhere, 9
- list.as.matrix, 8
- nodepackage, 9
- nodescloud, 10
- nodesprofile, 11
- nodesrunlength, 12
- parse\_rprof, 22
- plot.sprof, 13, 16, 17, 19
- plot\_nodes, 14, 14
- plot\_profiles, 14, 16
- plot\_stacks, 14, 18
- print.sprof, 19
- print\_nodes (print.sprof), 19
- print\_profiles, 20
- print\_stacks (print.sprof), 19
- profiles\_matrix, 20
- readRprof, 2, 10, 13, 19, 21, 27, 31, 34
- rkindex, 23
- rle, 26

roots\_sprof, [24](#)  
Rprof, [2](#), [21](#), [22](#)  
Rprofmem, [22](#)  
rrle, [24](#)  
rrleb, [25](#), [25](#)  
  
sampleRprof, [2](#), [26](#)  
shownodes, [13](#), [14](#), [27](#)  
sprof (sprof-package), [2](#)  
sprof-package, [2](#)  
sprof01lm, [28](#)  
stacksasfactor, [29](#)  
stackstoadj, [30](#)  
str\_prof, [30](#)  
summary.sprof, [31](#)  
summary\_nodes (summary.sprof), [31](#)  
summary\_profiles (summary.sprof), [31](#)  
summary\_stacks (summary.sprof), [31](#)  
summary\_terminals, [32](#)  
summaryRprof, [14](#), [19](#), [22](#), [31](#)  
  
tracemem, [22](#)  
trimstacks, [33](#)  
  
updateRprof, [33](#)  
  
writeRprof, [34](#)