

Package ‘profr’

February 20, 2015

Version 0.3.1

Title An alternative display for profiling information

Description profr provides an alternative data structure and visual rendering for the profiling information generated by Rprof.

Author Hadley Wickham <h.wickham@gmail.com>

Maintainer Hadley Wickham <h.wickham@gmail.com>

URL <https://github.com/hadley/profr>

BugReports <https://github.com/hadley/profr/issues>

Imports stringr, plyr

Suggests ggplot2

License MIT + file LICENSE

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2014-04-22 16:35:46

R topics documented:

ggplot.profr	2
parse_rprof	2
plot.profr	3
profr	4
sample-data	5

Index	6
--------------	----------

<code>ggplot.profr</code>	<i>Visualise profiling data with ggplot2. Visualise profiling data stored in a profr data.frame.</i>
---------------------------	--

Description

This will plot the call tree of the specified stop watch object. If you only want a small part, you will need to subset the object

Usage

```
ggplot.profr(data, ..., minlabel = 0.1, angle = 0)
```

Arguments

<code>data</code>	profile output to plot
<code>...</code>	other arguments passed on to ggplot
<code>minlabel</code>	minimum percent of time for function to get a label
<code>angle</code>	function label angle

See Also

[plot.profr](#)

Examples

```
if (require("ggplot2")) {  
  ggplot(nesting_prof)  
  ggplot(reshape_prof)  
}
```

<code>parse_rprof</code>	<i>Parse Rprof output.</i>
--------------------------	----------------------------

Description

Parses the output of [Rprof](#) into an alternative format described in [profr](#). This produces a flat data frame, which is somewhat easier to summarise and visualise.

Usage

```
parse_rprof(path, interval = 0.02)
```

Arguments

path path to [Rprof](#) output
interval real-time interval between samples (in seconds)

Value

[data.frame](#) of class `profr`

See Also

[profr](#) for profiling and parsing

Examples

```
nesting_ex <- system.file("samples", "nesting.rprof", package="profr")
nesting <- parse_rprof(nesting_ex)

reshape_ex <- system.file("samples", "reshape.rprof", package="profr")
diamonds <- parse_rprof(reshape_ex)
```

plot.profr	<i>Visualise profiling data with base graphics. Visualise profiling data stored in a profr data.frame.</i>
------------	--

Description

If you only want a small part of the total call tree, you will need to subset the object as demonstrated by the example.

Usage

```
## S3 method for class 'profr'
plot(x, ..., minlabel = 0.1, angle = 0)
```

Arguments

x profile output to plot
... other arguments passed on to [plot.default](#)
minlabel minimum percent of time for function to get a label
angle function label angle

See Also

[ggplot.profr](#)

Examples

```
plot(nesting_prof)
plot(reshape_prof)
```

`profr`*Profile the performance of a function call.*

Description

This is a wrapper around [Rprof](#) that provides results in an alternative data structure, a `data.frame`. The columns of the `data.frame` are:

Usage

```
profr(expr, interval = 0.02, quiet = TRUE)
```

Arguments

<code>expr</code>	expression to profile
<code>interval</code>	interval between samples (in seconds)
<code>quiet</code>	should output be discarded?

Details

f name of function

level level in call stack

time total time (seconds) spent in function

start time at which control entered function

end time at which control exited function

leaf TRUE if the function is a terminal node in the call tree, i.e. didn't call any other functions

source guess at the package that the function came from

Value

`data.frame` of class `profr`

See Also

[parse_rprof](#) to parse standalone [Rprof](#) file, [plot.profr](#) and [ggplot.profr](#) to visualise the profiling data

Examples

```
## Not run:
glm_ex <- profr({Sys.sleep(1); example(glm)}, 0.01)
head(glm_ex)
summary(glm_ex)
plot(glm_ex)

## End(Not run)
```

`sample-data`*Sample profiling datasets*

Description

These two datasets illustrate the results of running `parse_rprof` on the sample `Rprof` output stored in the `samples` directory. The output was generated by the code in `samples/generate.r`.

Usage`nesting_prof``reshape_prof`**Format**

a data frame

Index

- *Topic **datasets**
 - sample-data, 5
- *Topic **debugging**
 - parse_rprof, 2
 - profr, 4
- *Topic **hplot**
 - ggplot.profr, 2
 - plot.profr, 3

data.frame, 3, 4

ggplot, 2

ggplot.profr, 2, 3, 4

nesting_prof (sample-data), 5

parse_rprof, 2, 4, 5

plot.default, 3

plot.profr, 2, 3, 4

profr, 2, 3, 4

reshape_prof (sample-data), 5

Rprof, 2–5

sample-data, 5