

# Package ‘profr’

February 15, 2012

**Type** Package

**Title** An alternative display for profiling information

**Version** 0.2

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

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

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

**Imports** digest

**Suggests** ggplot2

**License** MIT

**LazyData** true

**Collate** ‘output.r’ ‘parse.r’ ‘profile.r’

**Repository** CRAN

**Date/Publication** 2011-01-28 20:02:09

## R topics documented:

ggplot.profr . . . . .	2
parse_rprof . . . . .	2
plot.profr . . . . .	3
profr . . . . .	4
Sample profiling data . . . . .	5

<b>Index</b>	<b>6</b>
--------------	----------

---

`ggplot.profr`*Visualise profiling data with ggplot2.*

---

### Description

Visualise profiling data with ggplot2. Visualise profiling data stored in a `profr` `data.frame`.

### Usage

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

### Arguments

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

### Details

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

### See Also

[plot.profr](#)

### Examples

```
if (require("ggplot2", quiet = TRUE)) {  
  ggplot(nesting_prof)  
  ggplot(reshape_prof)  
}
```

---

`parse_rprof`*Parse Rprof output.*

---

### Description

Parse Rprof output. Parses the output of [Rprof](#) into an alternative format described in [profr](#).

### Usage

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

**Arguments**

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

**Details**

This produces a flat data frame, which is somewhat easier to summarise and visualise.

**Value**

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

**See Also**

[profr](#) for profiling and parsing

**Examples**

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

---

plot.profr

*Visualise profiling data with base graphics.*


---

**Description**

Visualise profiling data with base graphics. Visualise profiling data stored in a `profr` `data.frame`.

**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

**Details**

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

**See Also**

[ggplot.profr](#)

**Examples**

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

---

profr

*Profile the performance of a function call.*

---

**Description**

Profile the performance of a function call.

**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**

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

**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 profiling 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
```

# Index

\*Topic **datasets**

Sample profiling 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](#), [4](#)

nesting\_prof (Sample profiling data), [5](#)

parse\_rprof, [2](#), [4](#), [5](#)

plot.default, [3](#)

plot.profr, [2](#), [3](#), [4](#)

profr, [2](#), [3](#), [4](#)

reshape\_prof (Sample profiling data), [5](#)

Rprof, [2-5](#)

Sample profiling data, [5](#)