

Package ‘strip’

May 9, 2026

Type Package

Title Lighten your R Model Outputs

Version 1.0.0

Date 2018-09-30

Description The strip function deletes components of R model outputs that are useless for specific purposes, such as predict[ing], print[ing], summary[izing], etc.

License MIT + file LICENSE

LazyData TRUE

Depends R (>= 3.1.3)

Imports rlist

Suggests caret, datasets, e1071, knitr, randomForest, stats, testthat, utils

URL <https://github.com/paulponcet/strip>

BugReports <https://github.com/paulponcet/strip/issues>

RoxygenNote 6.1.0

NeedsCompilation no

Author Paul Poncet [aut, cre]

Maintainer Paul Poncet <paulponcet@yahoo.fr>

Repository CRAN

Date/Publication 2018-10-29 15:30:03 UTC

Contents

strip	2
Index	5

`strip`*Lighten R model outputs*

Description

The `strip` function deletes components of R model outputs that are useless for specific purposes, such as `predict[ing]`, `print[ing]`, `summary[izing]`, etc.

The idea is to prevent the size of the model output to grow with the size of the training dataset. This is useful if one has to save the output for later use while limiting its size on disk.

The birth of this package originates with Nina Zumel's post [‘Trimming the Fat from glm\(\) Models in R’](#) on Win-Vector Blog.

Usage

```
strip(object, keep, ...)  
  
strip_(object, keep, ...)  
  
## Default S3 method:  
strip_(object, keep, ...)  
  
## S3 method for class 'gam'  
strip_(object, keep, ...)  
  
## S3 method for class 'glm'  
strip_(object, keep, ...)  
  
## S3 method for class 'kmeans'  
strip_(object, keep, ...)  
  
## S3 method for class 'lm'  
strip_(object, keep, ...)  
  
## S3 method for class 'loess'  
strip_(object, keep, ...)  
  
## S3 method for class 'randomForest'  
strip_(object, keep, ...)  
  
## S3 method for class 'train'  
strip_(object, keep, use_trim = FALSE, ...)
```

Arguments

`object` result of an R model, see 'Details'.

keep	character. A vector of values among "everything", "predict", "print", and "summary". Except for <code>strip.lm</code> , currently only the values "everything", "predict", and "print", are implemented.
...	Additional arguments to be passed to other methods.
use_trim	boolean. For the <code>strip.train</code> method, if <code>use_trim=TRUE</code> and if <code>keep="predict"</code> , then the function applied is (if it exists) the <code>trim</code> function embedded as <code>object\$modelInfo\$trim</code> .

Details

If `keep="predict"`, components inside the list object are kept if they are needed by the `predict` method, otherwise they are set to `NULL`. If `keep=c("predict", "print")`, components are kept as soon as they are needed by one of the `predict` or `print` methods. If `keep="everything"`, object is returned with no modifications.

Currently the models supported are limited to the following list:

- `lm` and `glm`, the linear and generalized linear regression function from package **stat**;
- `loess`, the local polynomial regression function from package **stat**;
- `randomForest`, from package **randomForest**.

There is also a `strip` function for 'train' objects built with the **caret** package.

Further developments of the package should include additional models, and should enable additional `keep` values (e.g. `keep="summary"`, `keep="anova"`, etc.)

Value

A list of the same class as object is returned.

Author(s)

The method for `glm` objects is adapted from [Nina Zumel's post](#) on Win-Vector Blog.

The method for `randomForest` objects is adapted from [ReKa's answer](#) on StackExchange.

See Also

See [Nina Zumel's post](#) on Win-Vector Blog for further insight, examples, and motivations; [ReKa's answer](#) on StackExchange for reducing the size of a `randomForest` object; [this discussion](#) for limiting the 'footprint' of regression and classification objects within the **caret** package.

Examples

```
data("mtcars")
set.seed(110)
i = sample(2, nrow(mtcars), replace = TRUE, prob=c(0.8, 0.2))
r1 = lm(mpg ~ ., data = mtcars[i==1,])
r2 = strip(r1, keep = "predict")

# Estimate the objects' size as the size of their serialization
length(serialize(r1, NULL))
```

```
length(serialize(r2, NULL))

# Check that predictions are the same
p1 = predict(r1, newdata = mtcars[i==2,])
p2 = predict(r2, newdata = mtcars[i==2,])
identical(p1, p2) # TRUE
```

Index

`strip`, 2
`strip_(strip)`, 2