

# Package ‘benchmark’

February 19, 2015

**Type** Package

**Title** Benchmark Experiments Toolbox

**Version** 0.3-6

**Date** 2014-05-02

**Depends** R (>= 2.10), utils, proto, ggplot2, relations, psychotools

**Imports** reshape, scales, plyr

**Suggests** coin, multcomp, lme4, e1071, entropy, archetypes, Rgraphviz, psychotree

**Description** The benchmark package provides a toolbox for setup, execution and analysis of benchmark experiments. Main focus is the analysis of data accumulating during the execution -- one primary objective is the statistical correct computation of the candidate algorithms' order.

**License** GPL (>= 2)

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

**Collate** 'algerperf-beplot0.R' 'testprocedure.R' 'proto.R'  
'algerperf-paircomp.R' 'algerperf-preference.R'  
'algerperf-visualizations.R' 'warehouse.R' 'algerperf.R'  
'as.psychobench.R' 'as.warehouse.R' 'benchmark.R' 'bsgraph.R'  
'bsplot.R' 'datachar-visualizations.R'  
'dataset-characteristics.R' 'dataset.R'  
'dataset-characterization.R' 'testres-analysis.R'  
'testres-visualizations.R'

**Author** Manuel J. A. Eugster [aut, cre]

**Maintainer** Manuel J. A. Eugster <manuel@mjae.net>

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-05-02 21:48:47

## R topics documented:

as.dataset	2
as.psychobench	3
as.relation.PaircompDecision	4
as.warehouse.mlr.bench.result	4
benchmark	5
beplot0	6
boxplot.AlgorithmPerformance	7
bs.sampling	8
bsgraph0	9
bsplot0	10
characterize	11
fittime	12
ghraw	12
monks3raw	13
paircomp	13
plot.DatasetCharacterization	14
plot.TestResult	15
StatlogCharacteristics	15
subset.AlgorithmPerformance	16
TestProcedure	17
uci621raw	17
warehouse	18
<b>Index</b>	<b>20</b>

---

as.dataset	<i>Dataset abstraction</i>
------------	----------------------------

---

### Description

Dataset abstraction to simplify characterization.

### Usage

```
as.dataset(formula, data, ordered.as.factor = TRUE,
           integer.as.numeric = TRUE)
```

### Arguments

formula	A symbolic description of the dataset
data	The data frame
ordered.as.factor	Interpret ordered factors as factors
integer.as.numeric	Interpret integer variables as numerics

**Value**

A proto object with an additional S3 class dataset

**See Also**

Other dataset.characterization: [StatlogCharacteristics](#); [characterize](#); [plot.DatasetCharacterization](#)

**Examples**

```
data("iris")
ds <- as.dataset(Species ~ ., iris)
ds

str(ds$response())
str(ds$dataparts(c("input", "numeric")))
```

---

as.psychobench

*Coerce benchmark experiment warehouse to preference table*

---

**Description**

Coerce benchmark experiment warehouse to preference table

**Usage**

```
as.psychobench(x, comparisons = TRUE)
```

**Arguments**

x                   A [warehouse](#) object

comparisons       Return preference or performance table

**Value**

Data.frame with preference or performance table

---

as.relation.PaircompDecision

*Relations based on pairwise comparisons*

---

### Description

Infer a [relation](#) based on pairwise decisions.

### Usage

```
## S3 method for class 'PaircompDecision'
as.relation(x, verbose = FALSE, ...)

relation_is_strict_weak_order(x)
```

### Arguments

x	A <a href="#">PaircompDecision</a> object
verbose	Show information during execution
...	Ignored

### Value

A [relation](#) object

---

as.warehouse.mlr.bench.result

*Coerce objects to benchmark experiment warehouse*

---

### Description

Coerce objects to benchmark experiment warehouse

### Usage

```
as.warehouse.mlr.bench.result(x, ...)

as.warehouse.array4dim(x, ...)
```

### Arguments

x	An object to coerce
...	Ignored

**Details**

`as.warehouse.mlr.bench.result`: Coerces a `bench.result` object from package `mlr` to a [warehouse](#) object.

`as.warehouse.array4dim`: Coerces a four dimensional array (1st: sampling, 2nd: algorithms, 3rd: performance measures, 4th: datasets) to a [warehouse](#) object.

**Value**

A [warehouse](#) object

---

benchmark

*Benchmark experiment setup and execution*

---

**Description**

Function to execute benchmark experiments and to collect all data the package can analyze. For more sophisticated benchmark experiments we suggest the usage of the `mlr` package.

**Usage**

```
benchmark(datasets, sampling, algorithms = NULL, performances = NULL,
           characteristics = NULL, test = NULL, test.burnin = 3, verbose = TRUE)
```

**Arguments**

<code>datasets</code>	List of <code>data.frames</code>
<code>sampling</code>	Sampling function, see <a href="#">benchmark-sampling</a> .
<code>algorithms</code>	List of algorithms; i.e., functions which take a model formula and a <code>data.frame</code> to fit a model. Note that a <a href="#">predict</a> function must be defined as well.
<code>performances</code>	List of performance measure functions; i.e., functions with arguments <code>yhat</code> and <code>y</code> . See, e.g., <a href="#">benchmark-comptime</a> .
<code>characteristics</code>	DatasetCharacteristics object; e.g., <a href="#">StatlogCharacteristics</a>
<code>test</code>	<a href="#">TestProcedure</a> object
<code>test.burnin</code>	Number of burn-in replications
<code>verbose</code>	Show information during execution

**Value**

A [warehouse](#) object

**See Also**

[warehouse](#), [as.warehouse](#), [benchmark-sampling](#), [benchmark-comptime](#)

---

 beplot0

*Benchmark experiment plot*


---

## Description

The benchmark experiment plot visualizes each benchmark experiment run. The x-axis is a podium with as many places as algorithms. For each benchmark run, the algorithms are sorted according to their performance values and a dot is drawn on the corresponding place. To visualize the count of an algorithm on a specific position, a bar plot is shown for each of podium places.

## Usage

```
beplot0(x, ...)
```

```
## S3 method for class 'AlgorithmPerformance'
beplot0(x, xlab = NULL, ylab = NULL,
  lines.show = FALSE, lines.alpha = 0.2, lines.lwd = 1, lines.col = col,
  dots.pch = 19, dots.cex = 1, places.lty = 2, places.col = 1,
  legendfn = function(algs, cols) { legend("topleft", algs, lwd = 1, col =
  cols, bg = "white") }, ...)
```

```
## S3 method for class 'matrix'
beplot0(x, col = 1:ncol(x), xlab = NULL, ylab = NULL,
  lines.show = FALSE, lines.alpha = 0.2, lines.lwd = 1, lines.col = col,
  dots.pch = 19, dots.cex = 1, places.lty = 2, places.col = 1,
  legendfn = function(algs, cols) { legend("topleft", algs, lwd = 1, col =
  cols, bg = "white") }, ...)
```

## Arguments

x	A matrix or <a href="#">AlgorithmPerformance</a> object
xlab	A title for the x axis
ylab	A title for the y axis
lines.show	Connect dots of same benchmark runs
lines.col	Line color
lines.alpha	Alpha value of the line color
lines.lwd	Line width
dots.pch	Dot symbol
dots.cex	Dot symbol expansion
places.lty	Type of separator line between podium places
places.col	Color of separator line between podium places
legendfn	Function which draws a legend
...	Ignored
col	Colors

**Value**

Return value of underlying beplot0.matrix; currently undefined

**References**

See *Eugster and Leisch (2008)* and *Eugster et al. (2008)* in citation("benchmark").

**See Also**

Other algperf.visualization: [boxplot.AlgorithmPerformance](#), [densityplot](#), [densityplot.AlgorithmPerformance](#), [stripchart.AlgorithmPerformance](#); [bsgraph0](#), [bsgraph0.dist](#), [bsgraph0.graphNEL](#); [bsplot0](#), [bsplot0.matrix](#), [bsplot0.relation\\_ensemble](#)

---

boxplot.AlgorithmPerformance

*Common visualizations of algorithm performances*

---

**Description**

Common visualizations of algorithm performances

**Usage**

```
## S3 method for class 'AlgorithmPerformance'
boxplot(x, order.by = median,
        order.performance = 1, dependence.show = c("outliers", "all", "none"),
        dependence.col = alpha("black", 0.1), ...)
```

```
densityplot(x, ...)
```

```
## S3 method for class 'AlgorithmPerformance'
densityplot(x, ...)
```

```
## S3 method for class 'AlgorithmPerformance'
stripchart(x, order.by = median,
           order.performance = 1, dependence.show = c("none", "all"),
           dependence.col = alpha("black", 0.1), ...)
```

**Arguments**

x	An <a href="#">AlgorithmPerformance</a> object
order.by	Function like <a href="#">mean</a> , <a href="#">median</a> , or <a href="#">max</a> to calculate a display order of the algorithms; or NULL for no specific order.
order.performance	Name or index of the reference performance measure to calculate the order.

dependence.show Show dependence of observations for all, none or outlier observations.  
 dependence.col Color of the dependence line.  
 ... Ignored.

**Value**

A [ggplot](#) object.

**See Also**

Other `algerf.visualization`: [beplot0](#), [beplot0.AlgorithmPerformance](#), [beplot0.matrix](#); [bsgraph0](#), [bsgraph0.dist](#), [bsgraph0.graphNEL](#); [bsplot0](#), [bsplot0.matrix](#), [bsplot0.relation\\_ensemble](#)

---

<code>bs.sampling</code>	<i>Sampling functions</i>
--------------------------	---------------------------

---

**Description**

Functions to create a set of learning and test samples using a specific resampling method.

**Usage**

```
bs.sampling(B)
sub.sampling(B, psize)
cv.sampling(k)
```

**Arguments**

B	Number of learning samples
psize	Size of subsample
k	Number of cross-validation samples

**Value**

List with corresponding learning and test samples

**See Also**

[benchmark](#)



---

 bsgraph0

*Benchmark experiment graph*


---

## Description

The benchmark summary plot takes the individual benchmark experiment results into account. The y-axis represents the data sets, the x-axis a podium with as many places as candidate algorithms.

## Usage

```
bsgraph0(x, ...)

## S3 method for class 'dist'
bsgraph0(x, ndists.show = length(sort(unique(x))),
         edge.col = gray(0.7), edge.lwd = 1, node.fill = NULL, ...)

## S3 method for class 'graphNEL'
bsgraph0(x, layoutType = "neato", ...)
```

## Arguments

x	A <a href="#">dist</a> or <a href="#">graphNEL-class</a> object
ndists.show	The number of distance levels to show
edge.col	The color of edges (one or one for each distance level)
edge.lwd	The line width of edges (one or one for each distance level)
node.fill	The colors of nodes
...	Arguments passed to underlying function
layoutType	Defines the layout engine

## Value

The return value of [bsgraph0.graphNEL](#)

## See Also

Other `algerperf.visualization`: [beplot0](#), [beplot0.AlgorithmPerformance](#), [beplot0.matrix](#); [boxplot.AlgorithmPerformance](#), [densityplot](#), [densityplot.AlgorithmPerformance](#), [stripchart.AlgorithmPerformance](#); [bsplot0](#), [bsplot0.matrix](#), [bsplot0.relation\\_ensemble](#)

---

 bsplot0

*Benchmark experiment summary plot.*


---

### Description

The benchmark summary plot takes the individual benchmark experiment results into account. The y-axis represents the data sets, the x-axis a podium with as many places as candidate algorithms.

### Usage

```
bsplot0(x, ...)

## S3 method for class 'relation_ensemble'
bsplot0(x, stat = NULL, ds.order = NULL,
        alg.order = NULL, ...)

## S3 method for class 'matrix'
bsplot0(x, stat = NULL, col = structure(seq_len(nrow(x)) +
    1, names = rownames(x)), ylab = "Datasets", xlab = "Podium",
    sig.lwd = 4, stat.col = NULL, ylab.las = NULL, ...)
```

### Arguments

x	A <a href="#">relation_ensemble</a> or matrix object
stat	A matrix with statistics to display (rows are the algorithms, columns the data sets)
ds.order	Data set order
alg.order	Algorithm order
...	Arguments passed to underlying function
col	Colors of the algorithms
xlab	A title for the x axis
ylab	A title for the y axis
sig.lwd	Line width of the significance sperator line
stat.col	Colors of the statistics
ylab.las	las of the labels of the y axis

### See Also

Other alperf.visualization: [beplot0](#), [beplot0.AlgorithmPerformance](#), [beplot0.matrix](#); [boxplot.AlgorithmPerformance](#), [densityplot](#), [densityplot.AlgorithmPerformance](#), [stripchart.AlgorithmPerformance](#); [bsgraph0](#), [bsgraph0.dist](#), [bsgraph0.graphNEL](#)

---

characterize	<i>Dataset characterization framework</i>
--------------	---

---

## Description

Implementation of a map/reduce approach to characterize a dataset with given dataset characteristics.

## Usage

```
characterize(x, y, verbose = FALSE, index = NULL, ...)
```

## Arguments

x	A <a href="#">dataset</a> object
y	A <a href="#">DatasetCharacteristics</a> object; e.g., <a href="#">StatlogCharacteristics</a>
verbose	Show information during execution
index	Characterize only a subset
...	Ignored

## Value

The characterization matrix (1 row and as many columns as characteristics)

## References

See *Eugster et al. (2010)* in `citation("benchmark")`.

## See Also

Other `dataset.characterization`: [StatlogCharacteristics](#); [as.dataset](#); [plot.DatasetCharacterization](#)

## Examples

```
data("iris")
ds <- as.dataset(Species ~ ., iris)
characterize(ds, StatlogCharacteristics)
```

fittime

*Performance measures*

---

**Description**

Dummy functions to enable fitting and prediction time as performance measures.

**Usage**

```
fittime(yhat, y)
```

```
predicttime(yhat, y)
```

**Arguments**

yhat            Ignored

y               Ignored

**Value**

Time (User and System) used for the model fitting or prediction

**See Also**

[benchmark](#)

---

ghraw*Grasshopper domain benchmark experiment results*

---

**Description**

Misclassification error and various other performance measures on the data set a domain of 33 data sets with information if a specific grasshopper species is available in a territory or not. The candidate algorithms are lda, rf, knn, rpart, svm and naiveBayes.

ghrel are precomputed relations.

Used in demo("lsbenchplot-gh").

**Usage**

```
ghraw
```

**Format**

A data frame with 99000 observations and 5 variables (alg, samp, perf, value, ds).

---

 monks3raw

*monks3 benchmark experiment results*


---

**Description**

Misclassification error and computation time on the data set monks3 of the candidate algorithms lda, rf, knn, rpart, svm and nnet.

Used in demo("benchplot").

**Usage**

monks3raw

**Format**

An array of dimension 250 x 6 x 2 x 1 (sampling x algorithms x performances x data sets).

**References**

Manuel J. A. Eugster, Torsten Hothorn and Friedrich Leisch. "Exploratory and Inferential Analysis of Benchmark Experiments", Technical Report 30, Department of Statistics, Ludwig-Maximilians-Universitaet Muenchen, Germany. 2008. <http://epub.ub.uni-muenchen.de/4134/>

---

 paircomp

*Pairwise comparison of algorithm performances*


---

**Description**

Pairwise comparison of algorithm performances

**Usage**

paircomp(x, family, type = c("<", "="), ...)

**Arguments**

x	An <a href="#">AlgorithmPerformance</a> object
family	A <a href="#">Paircomp</a> object; see details section
type	Draw strict or indifference decision
...	Ignored

**Details**

Available TestPaircomp implementations:

FriedmanTestPaircomp	Pairwise comparison based on the non parametric friedman test
LmerTestPaircomp	Pairwise comparison based on a mixed effects model (function lmer in package lme4)
PercintTestPaircomp	Pairwise comparison based on the bootstrap percentile intervals

Available PointPaircomp implementations:

GenericPointPaircomp	Pairwise comparison based on point estimates.
----------------------	---

## Value

A PaircompDecision object; a list with the elements:

decision	The incidence matrix representing the pairwise comparisons
type	The decision type
base	A list with information on the decision base

## References

See *Eugster and Leisch (2008)* and *Eugster et al. (2008)* in citation("benchmark").

---

plot.DatasetCharacterization  
*Visualization of dataset characteristics*

---

## Description

Visualization of dataset characteristics

## Usage

```
## S3 method for class 'DatasetCharacterization'
plot(x, y = NULL, lines = TRUE,
     points = TRUE, null.line = TRUE, null.line.col = gray(0.7),
     basis = TRUE, basis.col = NULL, ...)
```

## Arguments

x	A <a href="#">DatasetCharacterization</a> object
y	Ignored
lines	Draw observation dependency lines
points	Draw observation points
null.line	Draw null line
null.line.col	Null line color
basis	Draw basis characterization of the dataset
basis.col	Color of basis characterization
...	Ignored

**Value**

A `ggplot` object.

**See Also**

Other dataset.characterization: [StatlogCharacteristics](#); [as.dataset](#); [characterize](#)

---

plot.TestResult      *Visualization methods for (sequential) test results.*

---

**Description**

Visualization methods for (sequential) test results.

**Usage**

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

**Arguments**

x                    An `TestResult` object  
...                   Ignored.

**Value**

A `ggplot` object.

---

StatlogCharacteristics  
                          *StatLog dataset characteristics*

---

**Description**

Implementation of the StatLog project dataset characteristics.

**Usage**

```
StatlogCharacteristics
```

**Format**

```
proto object
$ requirements:function (., ...)
$ name           : chr "Statlog"
$ reduce         :function (.)
$ map            :function (.)
parent: proto object
```

**References**

See *Eugster et al. (2010)* in `citation("benchmark")`.

R. D. King, C. Feng and A. Sutherland. STATLOG: Comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence*, 9, 1995.

**See Also**

Other `dataset.characterization`: [as.dataset](#); [characterize](#); [plot.DatasetCharacterization](#)

---

subset.AlgorithmPerformance

*Return subsets of AlgorithmPerformance objects*

---

**Description**

Return subsets of `AlgorithmPerformance` objects

**Usage**

```
## S3 method for class 'AlgorithmPerformance'
subset(x, subset, ...)
```

**Arguments**

x	An <code>AlgorithmPerformance</code> object
subset	Logical expression indicating rows to keep
...	Passed to the underlying <code>subset.data.frame</code> call

**Value**

An `AlgorithmPerformance` object with just the selected observations



---

TestProcedure	<i>Test procedures infrastructure</i>
---------------	---------------------------------------

---

**Description**

Test procedures infrastructure

**Usage**

```
FriedmanTest
LmerTest
PercintTest
```

**Format**

```
proto object
$ requirements:function (., ...)
$ pairwiseTest:function (., ...)
$ globalTest :function (., ...)
$ new :function (., ...)
```

**Details**

Implemented TestProcedure and corresponding TestResult are:

```
FriedmanTest Test procedure based on the non parametric friedman test
LmerTest Test procedure based on a mixed effects model (function lmer in package lme4)
PercintTest Test procedure based on the bootstrap percentile intervals
```

**References**

See *Eugster and Leisch (2008)* and *Eugster et al. (2008)* in citation("benchmark").

---

uci621raw	<i>uci621 benchmark experiment results</i>
-----------	--

---

**Description**

Misclassification error and computation time on the data set 21 UCI Machine Learning Repository data sets of the candidate algorithms lda, rf, knn, rpart, svm and nnet.

uci621rel are precomputed relations.

Used in demo("lsbenchplot-cs621").

**Usage**

```
uci621raw
```

**Format**

An array of dimension 250 x 6 x 2 x 1 (sampling x algorithms x performances x data sets).

**References**

Manuel J. A. Eugster, Torsten Hothorn and Friedrich Leisch. "Exploratory and Inferential Analysis of Benchmark Experiments", Technical Report 30, Department of Statistics, Ludwig-Maximilians-Universitaet Muenchen, Germany. 2008. <http://epub.ub.uni-muenchen.de/4134/>

---

warehouse	<i>Benchmark experiment warehouse</i>
-----------	---------------------------------------

---

**Description**

warehouse is the constructor function for a benchmark experiment warehouse.

**Usage**

```
warehouse(datasets, B, algorithms = NULL, performances = NULL,
          characteristics = NULL, tests = NULL)
```

**Arguments**

datasets	Names of the datasets
B	Number of benchmark runs
algorithms	Names of the candidate algorithms
performances	Names of the performance measures
characteristics	Names of the dataset characteristics
tests	Names of the monitored test measures

**Details**

A benchmark experiment warehouse collects all data during the execution of a benchmark experiment; see [benchmark](#). Different views (based on the collected data) provide cleaned parts of the data for further analyses.

Implemented views:

1. `viewAlgorithmPerformance()`: returns a data frame (S3 class `AlgorithmPerformance`) with columns `samples`, `datasets`, `algorithms`, `performances` (factors with the corresponding levels) and the column `value` with the corresponding performance value.
2. `viewDatasetCharacterization()`: returns a data frame (S3 class `DatasetCharacterization`) with columns `samples`, `datasets`, `characteristics`, `value`.
3. `viewDatasetBasisCharacterization()`: returns a data frame (S3 class `DatasetBasisCharacterization`) with columns `datasets`, `characteristics`, `value`.
4. `viewTestResult()`: returns a data frame (S3 class `TestResult`) with columns `samples`, `datasets`, `tests`, `value`.

**Value**

Proto object with different views (see Details).

**See Also**

[benchmark, as .warehouse](#)

# Index

## \*Topic **datasets**

- ghraw, 12
- monks3raw, 13
- StatlogCharacteristics, 15
- TestProcedure, 17
- uci621raw, 17

AlgorithmPerformance, 6, 7, 13, 16

AlgorithmPerformance (warehouse), 18

as.dataset, 2, 11, 15, 16

as.psychobench, 3

as.relation.PaircompDecision, 4

as.warehouse, 5, 19

as.warehouse

- (as.warehouse.mlr.bench.result), 4

as.warehouse.mlr.bench.result, 4

benchmark, 5, 8, 12, 18, 19

benchmark-comptime (fittime), 12

benchmark-sampling (bs.sampling), 8

beplot0, 6, 8–10

beplot0.AlgorithmPerformance, 8–10

beplot0.matrix, 8–10

boxplot.AlgorithmPerformance, 7, 7, 9, 10

bs.sampling, 8

bsgraph0, 7, 8, 9, 10

bsgraph0.dist, 7, 8, 10

bsgraph0.graphNEL, 7–10

bsplot0, 7–9, 10

bsplot0.matrix, 7–9

bsplot0.relation\_ensemble, 7–9

characterize, 3, 11, 15, 16

cv.sampling (bs.sampling), 8

dataset, 11

DatasetBasisCharacterization

- (warehouse), 18

DatasetCharacterization, 14

DatasetCharacterization (warehouse), 18

densityplot, 7, 9, 10

densityplot

- (boxplot.AlgorithmPerformance), 7

densityplot.AlgorithmPerformance, 7, 9, 10

dist, 9

fittime, 12

FriedmanTest (TestProcedure), 17

FriedmanTestPaircomp (paircomp), 13

GenericPointPaircomp (paircomp), 13

ggplot, 8, 15

ghraw, 12

ghrel (ghraw), 12

LmerTest (TestProcedure), 17

LmerTestPaircomp (paircomp), 13

max, 7

mean, 7

median, 7

monks3raw, 13

paircomp, 13

PaircompDecision, 4

PaircompDecision (paircomp), 13

PercintTest (TestProcedure), 17

PercintTestPaircomp (paircomp), 13

plot.DatasetCharacterization, 3, 11, 14, 16

plot.TestResult, 15

PointPaircomp (paircomp), 13

predict, 5

predicttime (fittime), 12

relation, 4

relation\_ensemble, 10

relation\_is\_strict\_weak\_order  
    (as.relation.PaircompDecision),  
    4

StatlogCharacteristics, [3](#), [5](#), [11](#), [15](#), [15](#)

stripchart.AlgorithmPerformance, [7](#), [9](#),  
    [10](#)

stripchart.AlgorithmPerformance  
    (boxplot.AlgorithmPerformance),  
    7

sub.sampling (bs.sampling), [8](#)

subset.AlgorithmPerformance, [16](#)

subset.data.frame, [16](#)

TestPaircomp (paircomp), [13](#)

TestProcedure, [5](#), [17](#)

TestResult, [15](#)

TestResult (warehouse), [18](#)

uci621raw, [17](#)

uci621rel (uci621raw), [17](#)

warehouse, [3](#), [5](#), [18](#)