

# Package ‘prcbench’

March 5, 2019

**Type** Package

**Title** Testing Workbench for Precision-Recall Curves

**Version** 0.8

**Date** 2019-03-04

**Description** A testing workbench for evaluating precision-recall curves under various conditions.

**URL** <http://takayasaito.github.io/prcbench/>,  
<https://github.com/takayasaito/prcbench>

**BugReports** <https://github.com/takayasaito/prcbench/issues>

**Depends** R (>= 3.2.3)

**License** GPL-3

**LazyData** TRUE

**Imports** ROCR (>= 1.0-7), PRROC (>= 1.1), precrec (>= 0.1), rJava (>= 0.9-7), R6 (>= 2.1.1), assertthat (>= 0.1), grid, gridExtra (>= 2.0.0), graphics, ggplot2 (>= 2.1.0), methods, memoise (>= 1.0.0)

**RoxygenNote** 6.1.1

**Encoding** UTF-8

**Suggests** microbenchmark (>= 1.4-2.1), PerfMeas (>= 1.2.1), testthat (>= 0.11.0), knitr (>= 1.11), rmarkdown (>= 0.8.1)

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Takaya Saito [aut, cre],  
Marc Rehmsmeier [aut]

**Maintainer** Takaya Saito <takaya.saito@outlook.com>

**Repository** CRAN

**Date/Publication** 2019-03-05 12:50:03 UTC

## R topics documented:

autoplot.evalcurve . . . . .	2
C1DATA . . . . .	3
C2DATA . . . . .	4
C3DATA . . . . .	4
C4DATA . . . . .	4
create_example_func . . . . .	5
create_testset . . . . .	5
create_toolset . . . . .	7
create_usrdata . . . . .	8
create_usrtool . . . . .	9
prcbench . . . . .	10
run_benchmark . . . . .	11
run_evalcurve . . . . .	12
TestDataB . . . . .	12
TestDataC . . . . .	13
ToolAUCCalculator . . . . .	15
ToolIFBase . . . . .	16
ToolPerfMeas . . . . .	17
Toolprecrec . . . . .	18
ToolPRROC . . . . .	19
ToolROCR . . . . .	20
<b>Index</b>	<b>22</b>

---

autoplot.evalcurve      *Plot the result of Precision-Recall curve evaluation*

---

### Description

The plot\_eval\_results function validates Precision-Recall curves and creates a plot.

### Usage

```
## S3 method for class 'evalcurve'
autoplot(object, base_plot = TRUE,
         ret_grob = FALSE, ncol = NULL, nrow = NULL, use_category = FALSE,
         ...)
```

### Arguments

object	An S3 object that contains evaluation results of Precision-Recall curves.
base_plot	A Boolean value to specify whether the base points are plotted.
ret_grob	A Boolean value to specify whether the function returns a grob object.
ncol	An integer used for the column size of multiple panes.
nrow	An integer used for the row size of multiple panes.

<code>use_category</code>	A Boolean value to specify whether the categorical summary instead of the total summary.
<code>...</code>	Not used by this function.

**Value**

A data frame with validation results.

**Examples**

```
library(ggplot2)

## Plot evaluation results on test datasets r1, r2, and r3
testset <- create_testset("curve", c("c1", "c2", "c3"))
toolset <- create_toolset(set_names = "crv5")
eres1 <- run_evalcurve(testset, toolset)
autoplot(eres1)
```

---

C1DATA

*C1: Pre-calculated Precision-Recall curve*


---

**Description**

A list contains scores, labels, and pre-calculated recall and precision values as x and y.

**Usage**

```
data(C1DATA)
```

**Format**

A list with 5 items.

**scores** input scores

**labels** input labels

**bp\_x** pre-calculated recall values for curve evaluation

**bp\_y** pre-calculated precision values for curve evaluation

**tp\_x** x position for displaying the test result in a plot

**tp\_y** y position for displaying the test result in a plot

C2DATA

*C2: Pre-calculated Precision-Recall curve*

---

**Description**

A list contains scores, labels, and pre-calculated recall and precision values as x and y.

**Usage**

```
data(C2DATA)
```

**Format**

See [C1DATA](#).

---

C3DATA

*C3: Pre-calculated Precision-Recall curve*

---

**Description**

A list contains scores, labels, and pre-calculated recall and precision values as x and y.

**Usage**

```
data(C3DATA)
```

**Format**

See [C1DATA](#).

---

C4DATA

*C4: Pre-calculated Precision-Recall curve*

---

**Description**

A list contains scores, labels, and pre-calculated recall and precision values as x and y.

**Usage**

```
data(C4DATA)
```

**Format**

See [C1DATA](#).

---

create\_example\_func     *Create an example for the func argument of the create\_usrtool function*

---

**Description**

The create\_example\_func function creates an example for the [create\\_usrtool](#) function.

**Usage**

```
create_example_func()
```

**Value**

A function as an example for [create\\_usrtool](#)

**See Also**

[create\\_usrtool](#) requires the same format. [create\\_testset](#) for testset.

**Examples**

```
## Create a function
func <- create_example_func()
func
```

---

create\_testset     *Create a list of test datasets*

---

**Description**

The create\_testset function creates test datasets either for benchmarking or curve evaluation.

**Usage**

```
create_testset(test_type, set_names = NULL)
```

**Arguments**

test_type	A single string to specify the type of dataset generated by this function. <b>"bench"</b> Create test datasets for benchmarking <b>"curve"</b> Create test datasets for curve evaluation
set_names	A character vector to specify the names of test datasets.

1. For benchmarking (`test_type = "bench"`)  
 This function uses a naming convention for randomly generated data for benchmarking. The format is a prefix ('i' or 'b') followed by the number of dataset. The prefix 'i' indicates a balanced dataset, whereas 'b' indicates an imbalanced dataset. The number can be used with a suffix 'k' or 'm', indicating respectively 1000 or 1 million.  
 Below are some examples.  
**"b100"** A balanced data set with 50 positives and 50 negatives.  
**"b10k"** A balanced data set with 5000 positives and 5000 negatives.  
**"b1m"** A balanced data set with 500,000 positives and 500,000 negatives.  
**"i100"** An imbalanced data set with 25 positives and 75 negatives.  
 The function returns a list of [TestDataB](#) objects.
2. For curve evaluation (`test_type = "curve"`)  
 The following three predefined datasets can be specified for curve evaluation.

set name	S3 object	data source
c1 or C1	<a href="#">TestDataC</a>	<a href="#">C1DATA</a>
c2 or C2	<a href="#">TestDataC</a>	<a href="#">C2DATA</a>
c3 or C3	<a href="#">TestDataC</a>	<a href="#">C3DATA</a>
c4 or C4	<a href="#">TestDataC</a>	<a href="#">C4DATA</a>

The function returns a list of [TestDataC](#) objects.

## Value

A list of R6 test dataset objects.

## See Also

[run\\_benchmark](#) and [run\\_evalcurve](#) require the list of the datasets generated by this function. [TestDataB](#) for benchmarking test data. [TestDataC](#), [C1DATA](#), [C2DATA](#), [C3DATA](#), and [C4DATA](#) for curve evaluation test data. [create\\_usrdata](#) for creating a user-defined test set.

## Examples

```
## Create a balanced data set with 50 positives and 50 negatives
tset1 <- create_testset("bench", "b100")
tset1

## Create an imbalanced data set with 25 positives and 75 negatives
tset2 <- create_testset("bench", "i100")
tset2

## Create P1 dataset
tset3 <- create_testset("curve", "c1")
tset3

## Create P1 dataset
```

```
tset4 <- create_testset("curve", c("c1", "c2"))
tset4
```

---

create\_toolset      *Create a set of tools*

---

### Description

The `create_toolset` function takes names of predefined tools and generates a list of wrapper functions for Precision-Recall curve calculations.

### Usage

```
create_toolset(tool_names = NULL, set_names = NULL, calc_auc = TRUE,
               store_res = TRUE)
```

### Arguments

tool_names	A character vector to specify the names of performance evaluation tools. The names for the following five tools can be currently used. <ul style="list-style-type: none"> <li>• ROCR</li> <li>• AUCCalculator</li> <li>• PerfMeas</li> <li>• PRROC</li> <li>• precrec</li> </ul>
set_names	A character vector to specify a predefined set name. Following six sets are currently available. <p>"def5" A set of 5 tools with <code>calc_auc = TRUE</code> and <code>store_res = TRUE</code></p> <p>"auc5" A set of 5 tools with <code>calc_auc = TRUE</code> and <code>store_res = FALSE</code></p> <p>"crv5" A set of 5 tools with <code>calc_auc = FALSE</code> and <code>store_res = TRUE</code></p> <p>"def4" A set of 4 tools with <code>calc_auc = TRUE</code> and <code>store_res = TRUE</code></p> <p>"auc4" A set of 4 tools with <code>calc_auc = TRUE</code> and <code>store_res = FALSE</code></p> <p>"crv4" A set of 4 tools with <code>calc_auc = FALSE</code> and <code>store_res = TRUE</code></p>
calc_auc	A Boolean value to specify whether the AUC score should be calculated.
store_res	A Boolean value to specify whether the calculated curve is retrieved and stored

### Value

A list of R6 tool objects.

### See Also

[run\\_benchmark](#) and [run\\_evalcurve](#) require the list of the tools generated by this function [ToolROCR](#), [ToolAUCCalculator](#), [ToolPerfMeas](#), [ToolPRROC](#), and [Toolprecrec](#) as R6 tool classes.

## Examples

```
## Create ROCR and precrec
toolset1 <- create_toolset(c("ROCR", "precrec"))
toolset1

## Create auc5 tools
toolset2 <- create_toolset(set_names = "auc5")
toolset2
```

---

create\_usrdata      *Create a user-defined test dataset*

---

## Description

The create\_usrdata function creates various types of test datasets.

## Usage

```
create_usrdata(test_type, scores = NULL, labels = NULL,
               tsname = NULL, base_x = NULL, base_y = NULL, text_x = NULL,
               text_y = NULL, text_x2 = text_x, text_y2 = text_y)
```

## Arguments

test_type	A single string to specify the type of dataset generated by this function. <b>"bench"</b> Create a test dataset for benchmarking <b>"curve"</b> Create a test dataset for curve evaluation
scores	A numeric vector to set scores.
labels	A numeric vector to set labels.
tsname	A single string to specify the name of the dataset.
base_x	A numeric vector to set pre-calculated recall values for curve evaluation.
base_y	A numeric vector to set pre-calculated precision values for curve evaluation.
text_x	A single numeric value to set the x position for displaying the test result in a plot
text_y	A single numeric value to set the y position for displaying the test result in a plot
text_x2	A single numeric value to set the x position for displaying the test result (group into categories) in a plot
text_y2	A single numeric value to set the y position for displaying the test result (group into categories) in a plot

## Value

A list of R6 test dataset objects.



**See Also**

[create\\_testset](#) for creating a predefined test set. [TestDataB](#) for benchmarking test data. [TestDataC](#) for curve evaluation test data.

**Examples**

```
## Create a test dataset for benchmarking
testset2 <- create_usrdata("bench", scores = c(0.1, 0.2), labels = c(1, 0),
                          tsname = "m1")

testset2

## Create a test dataset for curve evaluation
testset <- create_usrdata("curve", scores = c(0.1, 0.2), labels = c(1, 0),
                         base_x = c(0, 1.0), base_y = c(0, 0.5))

testset
```

---

create_usrtool	<i>Create a set of tools</i>
----------------	------------------------------

---

**Description**

The `create_toolset` function takes names of predefined tools and generates a list of wrapper functions for Precision-Recall curve calculations.

**Usage**

```
create_usrtool(tool_name, func, calc_auc = TRUE, store_res = TRUE,
              x = NA, y = NA)
```

**Arguments**

<code>tool_name</code>	A single string to specify the name of a user-defined tool.
<code>func</code>	A function to calculate a Precision-Recall curve and the AUC. It should take an element of the test dataset generated by <a href="#">create_testset</a> as an argument. It also should return a list with three elements - 'x', 'y', and 'auc' that represent calculated recall and precision values plus the AUC score. See <a href="#">create_example_func</a> for an example.
<code>calc_auc</code>	A Boolean value to specify whether the AUC score should be calculated.
<code>store_res</code>	A Boolean value to specify whether the calculated curve is retrieved and stored.
<code>x</code>	Set pre-calculated recall values.
<code>y</code>	Set pre-calculated precision values.

**Value**

A list of R6 tool objects.

## See Also

[create\\_toolset](#) to create a predefined tool set. [create\\_testset](#) for testset. [create\\_example\\_func](#) to create an example function.

## Examples

```
## Create a new tool interface called "xyz"
efunc <- create_example_func()
toolset1 <- create_usrtool("xyz", efunc)
toolset1

## Example function with a correct argument
testset <- create_usrdata("bench", scores = c(0.1, 0.2), labels = c(1, 0))
retf <- efunc(testset[[1]])
retf
```

---

prcbench

*prcbench: A package to provide a testing workbench for precision-recall curves*

---

## Description

The prcbench package provides four categories of important functions: tool interface, test data interface, benchmarking, and curve evaluation.

### Tool interface

The [create\\_toolset](#) function creates a common interface for five different tools that calculate Precision-Recall curves. These tools are [ROCR](#), [AUCCalculator](#), [PerfMeas](#), [PRROC](#), and [precrec](#).

The [create\\_usrtool](#) function helps users to make the same interface of the predefined ones for their own tools.

### Test data interface

The [create\\_testset](#) function creates two different types of test data sets. The first type is for benchmarking, and the second type is for curve evaluation.

The [create\\_usrdata](#) function helps users to make their own test data sets.

### Benchmarking

The [run\\_benchmark](#) function takes a tool set and a test data set and run [microbenchmark](#) for them.

### Curve evaluation

The [run\\_evalcurve](#) function takes a tool set and a test data set and evaluates the accuracy of Precision-Recall curves for them.

---

run_benchmark	<i>Run microbenchmark with specified tools and test sets</i>
---------------	--

---

## Description

The run\_benchmark function runs [microbenchmark](#) for specified tools and test datasets

## Usage

```
run_benchmark(testset, toolset, times = 5, unit = "ms",
              use_sys_time = FALSE)
```

## Arguments

testset	A character vector to specify a test set generated by <a href="#">create_testset</a> .
toolset	A character vector to specify a tool set generated by <a href="#">create_toolset</a> .
times	The number of iteration used in <a href="#">microbenchmark</a> .
unit	A single string to specify the unit used in <a href="#">summary.microbenchmark</a> .
use_sys_time	A Boolean value to specify <a href="#">system.time</a> is used instead of <a href="#">summary.microbenchmark</a> .

## Value

A data frame of microbenchmark results with additional columns.

## See Also

[create\\_testset](#) to generate a test dataset. [create\\_toolset](#) to generate a tool set. [microbenchmark](#) for benchmarking details.

## Examples

```
## Not run:
## Benchmarking for b10 and i10 test sets and crv5, auc5, and def5 tool sets
testset <- create_testset("bench", c("b10", "i10"))
toolset <- create_toolset(set_names = "def5")
res1 <- run_benchmark(testset, toolset)
res1

## End(Not run)
```

---

run_evalcurve	<i>Evaluate Precision-Recall curves with specified tools and test sets</i>
---------------	--

---

**Description**

The run\_evalcurve function runs several tests to evaluate the accuracy of Precision-Recall curves.

**Usage**

```
run_evalcurve(testset, toolset, auto_combo = TRUE)
```

**Arguments**

testset	A character vector to specify a test set generated by <a href="#">create_testset</a> .
toolset	A character vector to specify a tool set generated by <a href="#">create_toolset</a> .
auto_combo	A Boolean value to specify whether a combination of test and tool sets is automatically created.

**Value**

A data frame with validation results.

**See Also**

[create\\_testset](#) to generate a test dataset. [create\\_toolset](#) to generate a tool set.

**Examples**

```
## Evaluate curves for c1, c2, c3 test sets and crv5 tool set
testset <- create_testset("curve", c("c1", "c2", "c3"))
toolset <- create_toolset(set_names = "crv5")
res1 <- run_evalcurve(testset, toolset)
res1
```

---

TestDataB	<i>R6 class of test dataset for performance evaluation tools</i>
-----------	--

---

**Description**

TestDataB is a class that contains scores and label for performance evaluation tools. It provides necessary methods for benchmarking.

**Usage**

```
TestDataB
```

**Format**

An R6 class object.

**Methods**

- `get_tname()`: Get the dataset name.
- `get_scores()`: Get a vector of scores.
- `get_labels()`: Get a vector of labels.
- `get_fg()`: Get a vector of positive scores.
- `get_bg()`: Get a vector of negative scores.
- `get_fname()`: Get a file name that contains scores and labels.
- `del_file()`: Delete the file with scores and labels.

**See Also**

[create\\_testset](#) for creating a list of test datasets. [TestDataC](#) is derived from this class for curve evaluation.

**Examples**

```
## Initialize with scores, labels, and a dataset name
testset <- TestDataC$new(c(0.1, 0.2, 0.3), c(0, 1, 1), "m1")
testset
```

---

TestDataC

*R6 class of test dataset for Precision-Recall curve evaluation*

---

**Description**

TestDataC is a class that contains scores and label for performance evaluation tools. It provides necessary methods for curve evaluation.

**Usage**

```
TestDataC
```

**Format**

An R6 class object.

## Methods

- `set_basepoints_x(x)`: Set pre-calculated recall values for curve evaluation
- `set_basepoints_y(y)`: Set pre-calculated precision values for curve evaluation
- `get_basepoints_x()`: Get pre-calculated recall values for curve evaluation
- `get_basepoints_y()`: Get pre-calculated precision values for curve evaluation
- `set_textpos_x(x)`: Set the x position for displaying the test result in a plot
- `set_textpos_y(y)`: Set the y position for displaying the test result in a plot
- `get_textpos_x()`: Get the x position for displaying the test result in a plot
- `get_textpos_y()`: Get the y position for displaying the test result in a plot

Following seven methods are inherited from [TestDataB](#). See [TestDataB](#) for the method descriptions.

- `get_datname()`
- `get_scores()`
- `get_labels()`
- `get_fg()`
- `get_bg()`
- `get_fname()`
- `del_file()`

## See Also

[create\\_testset](#) for creating a list of test datasets. It is derived from [TestDataB](#).

## Examples

```
## Initialize with scores, labels, and a dataset name
testset <- TestDataC$new(c(0.1, 0.2), c(1, 0), "c4")
testset

## Set base points
testset$set_basepoints_x(c(0.13, 0.2))
testset$set_basepoints_y(c(0.5, 0.6))
testset
```

---

ToolAUCCalculator      *R6 class of the AUCCalculator tool*

---

## Description

ToolAUCCalculator is a wrapper class for the [AUCCalculator](#) tool, which is a Java library that provides calculations of ROC and Precision-Recall curves.

## Usage

```
ToolAUCCalculator
```

## Format

An R6 class object.

## Inheritance

[ToolIFBase](#)

## Methods

`set_jarpath(jarpath)` It sets an AUCCalculator jar file.

`jarpath` File path of the AUCCalculator jar file, e.g. `"/path1/path2/auc2.jar"`.

Following nine methods are inherited from [ToolIFBase](#). See [ToolIFBase](#) for the method descriptions.

- `call((testset, calc_auc, store_res)`
- `get_toolname()`
- `set_toolname(toolname)`
- `get_setname()`
- `set_setname(setname)`
- `get_result()`
- `get_x()`
- `get_y()`
- `get_auc()`

## See Also

This class is derived from [ToolIFBase](#). `create_toolset` for creating a list of tools.

**Examples**

```
## Initialization
toolaucalc <- ToolAUCCalculator$new()

## Show object info
toolaucalc

## create_toolset should be used for benchmarking and curve evaluation
toolaucalc2 <- create_toolset("AUCCalculator")
```

---

ToolIFBase

*Base class of performance evaluation tools*


---

**Description**

ToolIFBase is an abstract class to provide a uniform interface for performance evaluation tools.

**Usage**

```
ToolIFBase
```

**Format**

An R6 class object

**Methods**

- `call(testset, calc_auc, store_res)`: It calls an actual tool to calculate Precision-Recall curves.  
`testset` R6 object generated by the `create_testset` function.  
`calc_auc` A Boolean value to specify whether the AUC score should be calculated.  
`store_res` A Boolean value to specify whether the calculated curve is retrieved and stored.
- `get_toolname()`: Get the name of the tool.
- `set_toolname(toolname)`: Set the name of the tool.
- `get_setname()`: Get the name of the tool set.
- `set_setname(setname)`: Set the name of the tool set.
- `get_result()`: Get a list with curve values and the AUC score.
- `get_x()`: Get calculated recall values.
- `get_y()`: Get calculated precision values.
- `get_auc()`: Get the AUC score.

**See Also**

[ToolROCR](#), [ToolAUCCalculator](#), [ToolPerfMeas](#), [ToolPRROC](#), and [Toolprecrec](#) are derived from this class. [create\\_toolset](#) for creating a list of tools.



---

ToolPerfMeas	<i>R6 class of the PerfMeas tool</i>
--------------	--------------------------------------

---

## Description

ToolPerfMeas is a wrapper class for the [PerfMeas](#) tool, which is an R library that provides several performance measures.

## Usage

```
ToolPerfMeas
```

## Format

An R6 class object.

## Inheritance

[ToolIFBase](#)

## Methods

Following nine methods are inherited from [ToolIFBase](#). See [ToolIFBase](#) for the method descriptions.

- `call(testset, calc_auc, store_res)`
- `get_toolname()`
- `set_toolname(toolname)`
- `get_setname()`
- `set_setname(setname)`
- `get_result()`
- `get_x()`
- `get_y()`
- `get_auc()`

## See Also

This class is derived from [ToolIFBase](#). [create\\_toolset](#) for creating a list of tools.

## Examples

```
## Initialization
toolperf <- ToolPerfMeas$new()

## Show object info
toolperf

## create_toolset should be used for benchmarking and curve evaluation
toolperf2 <- create_toolset("PerfMeas")
```

---

Toolprecrec

*R6 class of the precrec tool*

---

## Description

Toolprecrec is a wrapper class for the [precrec](#) tool, which is an R library that provides calculations of ROC and Precision-Recall curves.

## Usage

```
Toolprecrec
```

## Format

An R6 class object.

## Inheritance

[ToolIFBase](#)

## Methods

Following nine methods are inherited from [ToolIFBase](#). See [ToolIFBase](#) for the method descriptions.

- `call(testset, calc_auc, store_res)`
- `get_toolname()`
- `set_toolname(toolname)`
- `get_setname()`
- `set_setname(setname)`
- `get_result()`
- `get_x()`
- `get_y()`
- `get_auc()`

**See Also**

This class is derived from [ToolIFBase](#). [create\\_toolset](#) for creating a list of tools.

**Examples**

```
## Initialization
toolprecrec <- Toolprecrec$new()

## Show object info
toolprecrec

## create_toolset should be used for benchmarking and curve evaluation
toolprecrec2 <- create_toolset("precrec")
```

---

ToolPRROC

*R6 class of the PRROC tool*

---

**Description**

ToolPRROC is a wrapper class for the **PRROC** tool, which is an R library that provides calculations of ROC and Precision-Recall curves.

**Usage**

```
ToolPRROC
```

**Format**

An R6 class object.

**Inheritance**

[ToolIFBase](#)

**Methods**

`set_curve(val)` A Boolean value to specify whether a Precision-Recall curve is calculated.

`set_minStepSize(val)` A numeric value to specify the minimum step size between two intermediate points.

Following nine methods are inherited from [ToolIFBase](#). See [ToolIFBase](#) for the method descriptions.

- `call(testset, calc_auc, store_res)`
- `get_toolname()`
- `set_toolname(toolname)`
- `get_setname()`

- `set_setname(setname)`
- `get_result()`
- `get_x()`
- `get_y()`
- `get_auc()`

### See Also

This class is derived from [ToolIFBase](#). `create_toolset` for creating a list of tools.

### Examples

```
## Initialization
toolprroc <- ToolPRROC$new()

## Show object info
toolprroc

## create_toolset should be used for benchmarking and curve evaluation
toolprroc2 <- create_toolset("PRROC")
```

---

ToolROCR

*R6 class of the ROCR tool*

---

### Description

ToolROCR is a wrapper class for the **ROCR** tool, which is an R library that provides calculations of various performance evaluation measures.

### Usage

```
ToolROCR
```

### Format

An R6 class object.

### Inheritance

[ToolIFBase](#)

## Methods

Following nine methods are inherited from [ToolIFBase](#). See [ToolIFBase](#) for the method descriptions.

- `call(testset, calc_auc, store_res)`
- `get_toolname()`
- `set_toolname(toolname)`
- `get_setname()`
- `set_setname(setname)`
- `get_result()`
- `get_x()`
- `get_y()`
- `get_auc()`

## See Also

This class is derived from [ToolIFBase](#). [create\\_toolset](#) for creating a list of tools.

## Examples

```
## Initialization
toolrocr <- ToolROCR$new()

## Show object info
toolrocr

## create_toolset should be used for benchmarking and curve evaluation
toolrocr2 <- create_toolset("ROCR")
```

# Index

## \*Topic **datasets**

C1DATA, 3

C2DATA, 4

C3DATA, 4

C4DATA, 4

TestDataB, 12

TestDataC, 13

ToolAUCCalculator, 15

ToolIFBase, 16

ToolPerfMeas, 17

Toolprecrec, 18

ToolPRROC, 19

ToolROCR, 20

ToolPerfMeas, 7, 16, 17

Toolprecrec, 7, 16, 18

ToolPRROC, 7, 16, 19

ToolROCR, 7, 16, 20

autoplot.evalcurve, 2

C1DATA, 3, 4, 6

C2DATA, 4, 6

C3DATA, 4, 6

C4DATA, 4, 6

create\_example\_func, 5, 9, 10

create\_testset, 5, 5, 9–14

create\_toolset, 7, 10–12, 15–17, 19–21

create\_usrdata, 6, 8, 10

create\_usrtool, 5, 9, 10

microbenchmark, 10, 11

prcbench, 10

prcbench-package (prcbench), 10

run\_benchmark, 6, 7, 10, 11

run\_evalcurve, 6, 7, 10, 12

summary.microbenchmark, 11

system.time, 11

TestDataB, 6, 9, 12, 14

TestDataC, 6, 9, 13, 13

ToolAUCCalculator, 7, 15, 16

ToolIFBase, 15, 16, 17–21