

# Package ‘backtest’

October 12, 2022

**Type** Package

**Title** Exploring Portfolio-Based Conjectures About Financial Instruments

**Version** 0.3-4

**Date** 2015-09-17

**Author** Jeff Enos <jeff@kanecap.com> and David Kane <dave@kanecap.com>, with contributions from Kyle Campbell <kyle.w.campbell@williams.edu>, Daniel Gerlanc <daniel@gerlanc.com>, Aaron Schwartz <Aaron.J.Schwartz@williams.edu>, Daniel Suo <danielsuo@gmail.com>, Alexei Colin <acolin@fas.harvard.edu>, and Luyi Zhao <luyizhao@gmail.com>

**Description** The backtest package provides facilities for exploring portfolio-based conjectures about financial instruments (stocks, bonds, swaps, options, et cetera).

**Maintainer** Daniel Gerlanc <dgerlanc@enplusadvisors.com>

**Depends** R (>= 2.10), methods, grid, lattice

**License** GPL (>= 2)

**LazyLoad** yes

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2015-09-17 22:50:01

## R topics documented:

backtest-package	2
backtest	2
backtest-class	5
starmine	8

<b>Index</b>	<b>10</b>
--------------	-----------

---

backtest-package	<i>Exploring portfolio-based conjectures about financial instruments</i>
------------------	--

---

### Description

The backtest package provides facilities for exploring portfolio-based conjectures about financial instruments (stocks, bonds, swaps, options, et cetera).

### Details

```

Package:    backtest
Type:      Package
Version:    0.3-1
Date:      2010-02-18
Depends:   R (>= 2.3.0), methods, grid, lattice
License:   GPL (>= 2)
LazyLoad:  yes

```

#### Index:

backtest	Creating an Object of Class Backtest
backtest-class	Class "backtest"
starmine	StarMine Rankings, 1995

Further information is available in the following vignettes:

backtest Using the backtest package (source, pdf)

### Author(s)

Jeff Enos <jeff@kanecap.com> and David Kane <dave@kanecap.com>, with contributions from Kyle Campbell <kyle.w.campbell@williams.edu>, Daniel Gerlanc <daniel@gerlanc.com>, Aaron Schwartz <Aaron.J.Schwartz@williams.edu>, and Daniel Suo <danielsuo@gmail.com>

Maintainer: Jeff Enos <jeff@kanecap.com>

---

backtest	<i>Creating an Object of Class Backtest</i>
----------	---

---

### Description

Conducts a backtest and returns the results as an object of class backtest.

**Usage**

```
backtest(x,
         in.var,
         ret.var,
         universe,
         by.var = NULL,
         date.var = NULL,
         id.var = NULL,
         buckets = 5,
         natural = FALSE,
         do.spread = TRUE,
         by.period = TRUE,
         overlaps = 1)
```

**Arguments**

<code>x</code>	A data frame containing the data to be analysed in the backtest. The details of what this data frame must contain are given below.
<code>in.var</code>	A character vector which indicates the name of the column or columns in <code>x</code> to be used as input variables.
<code>ret.var</code>	A character vector which indicates the name of the column or columns in <code>x</code> to be used as return variables.
<code>by.var</code>	An optional character value, specifying a second variable in <code>x</code> to be used for categorising the data. The details of how categories are created are given below.
<code>id.var</code>	An optional character value which indicates the name of the column in <code>x</code> containing a unique identifier for each observation. <code>id.var</code> must be specified if <code>natural</code> is <code>TRUE</code> .
<code>date.var</code>	An optional character vector which indicates the name of the column in <code>x</code> to be used as a date for each observation. <code>date.var</code> must be specified if <code>natural</code> is <code>TRUE</code> . In order to call <code>plot</code> , the contents of <code>date.var</code> must be of class <code>Date</code> or be coercible to an object of class <code>Date</code> via <code>as.Date</code> .
<code>buckets</code>	An optional numeric vector which specifies how many quantiles to create according to <code>in.var</code> and <code>by.var</code> .
<code>universe</code>	An optional expression for selecting a subset of <code>x</code> . The details of how this expression may be constructed are given below.
<code>natural</code>	An optional logical value. If <code>TRUE</code> , the <code>summary</code> method returns additional information and the backtest object may be plotted. The details of how a natural backtest differs from a pooled backtest are given below.
<code>do.spread</code>	Object of class "logical". If <code>TRUE</code> the <code>summary</code> method displays information about the spread between the extreme quantiles. If <code>FALSE</code> this information is suppressed. Defaults to <code>TRUE</code> .
<code>by.period</code>	Object of class "logical". If <code>TRUE</code> the quantiles are recalculated within each date period. If <code>FALSE</code> the quantiles are calculated all at once. Defaults to <code>TRUE</code> .
<code>overlaps</code>	An object of class "numeric" which specifies the number of prior periods to include in the current period's portfolio weights calculation. If <code>overlaps</code> is

the default of 1, `backtest` behaves as usual and only uses a period's own data to determine its portfolio. If `overlaps` is set to  $n > 1$ , a period's portfolio comprises the weighted mean of portfolio weights from the previous  $n$  periods, with period  $n$  having a weight of  $1/n$ .

## Details

Data frames for `backtest` must, at a minimum, contain a column of class `numeric` to be referenced by the `in.var` and `ret.var` arguments.

The `in.var` is the primary variable by which the `backtest` categorises observations. It must reference a numeric column in `x`. Using the values in `x`, `backtest` breaks the values into equal sized quantiles, or buckets.

The `by.var` is the secondary variable by which the `backtest` categorises observations. When specifying both `in.var` and `by.var`, `backtest` organises the observations into a  $n$  by  $j$  matrix where  $n$  is the number of quantiles or categories created for the `by.var` and  $j$  is the number of quantiles created for the `in.var`. By default, `backtest` creates 5 quantiles.

If `natural` is `TRUE`, the data and arguments must meet certain requirements. First, the frequency of the observations and `ret.var` must be the same. Second, an `id.var` and `date.var` are required. Third, a `by.var` is not allowed. Note that the code does not verify that the `backtest` is truly natural; `backtest` accepts the value passed by the user as valid.

## Value

Returns an object of class `backtest`.

The functions `show` and `summary` are used to obtain and print a short description and longer summary of the results of the `backtest`. The accessor functions `counts`, `totalCounts`, `marginals`, `means`, `naCounts`, and `turnover` extract different parts of the value returned by `backtest`.

## Author(s)

Kyle Campbell <kyle.w.campbell@williams.edu> and Jeff Enos <jeff@kanecap.com>

## See Also

[backtest-class](#)

## Examples

```
data(starmine)

## Backtest with 1 'in.var' and 1 'ret.var'

bt <- backtest(starmine, in.var = "smi", ret.var = "ret.0.1.m", by.period = FALSE)
summary(bt)

## Backtest with 2 'in.var' values, 1 'ret.var', and a 'by.var'

bt <- backtest(starmine, in.var = c("smi", "cap.usd"),
```

```

ret.var = "ret.0.1.m", by.var = "sector", by.period = FALSE)
summary(bt)

## Backtest with 1 'in.var', 1 'by.var', and 1 'ret.var'. Number of
## buckets changed from default of 5 to 4. Change in number of buckets
## only affects the 'in.var' because the 'by.var' column in 'starmine'
## contains character data. For each value in this column there is a
## unique category.

bt <- backtest(starmine, in.var = "smi", by.var = "sector",
ret.var = "ret.0.1.m", buckets = 4, by.period = FALSE)
summary(bt)

## Backtest with 1 'in.var', multiple 'ret.var', and a
## universe restriction

bt <- backtest(starmine, in.var = "smi",
ret.var = c("ret.0.1.m", "ret.0.6.m"),
universe = sector == "HiTec", by.period = FALSE)
summary(bt)

## Running a natural backtest with 2 'in.vars', 1 'ret.var'
## 10 buckets

bt <- backtest(starmine, in.var = c("smi","cap.usd"),
ret.var = "ret.0.1.m", date.var = "date",
id.var = "id", buckets = 10,
natural = TRUE, by.period = FALSE)
summary(bt)

## The same backtest, but calculating quantiles within periods.

bt <- backtest(starmine, in.var = c("smi","cap.usd"),
ret.var = "ret.0.1.m", date.var = "date",
id.var = "id", buckets = 10,
natural = TRUE, by.period = TRUE)
summary(bt)

plot(bt, type = "turnover")
plot(bt, type = "return")
plot(bt, type = "cumreturn")
plot(bt, type = "cumreturn.split")

```

---

backtest-class

*Class "backtest"*


---

### Description

Contains results from the backtest function.

## Details

The primary method for accessing the backtest results is through the `summary` method. `summary` provides different displays depending on the type of backtest object. These displays are shown in the examples section. Accessor methods such as `means`, `counts`, `marginals`, `naCounts`, `turnover`, and `ci` may be used to extract other types of information from the object.

A backtest object with a natural value of `TRUE` may be graphed by calling the `plot` method. The default `plot` method graphs `return`. The other plots, `turnover` and `cumulative return`, must be explicitly specified as `plot(object, type = "turnover")` or `plot(object, type = "cumreturn")`.

The backtest object does not store the data frame used to create the backtest. It only stores the results and the names of the vectors used in calculating these results.

The results of a backtest are stored in a 5-dimensional array, `results`. The 1st dimension contains one value for every element of `ret.var`. The 2nd dimension contains one value for every element of `in.var`. The 3rd dimension contains one value for every element in `1:buckets[1]`, a vector from 1 through the number of `by.var` buckets. The 4th dimension contains one value for every element in `1:buckets[2]`, a vector from 1 through the number of `in.var` buckets. The 5th dimension contains 4 elements: `means`, `counts`, `trim.means`, and `NAs`.

## Objects from the Class

Objects can be created by calls to the function `backtest(data, in.var, ret.var, ...)`.

## Slots

`in.var`: Object of class "character" specifying the `in.var` values for this backtest.

`ret.var`: Object of class "character" containing the `ret.var` values for this backtest.

`by.var`: Object of class "character" containing the `by.var`, if specified, for this backtest.

`date.var`: Object of class "character" containing the `date.var`, if specified, for this backtest.

`buckets`: Object of class "numeric" containing the number(s) of buckets used create quantiles from the `in.var` and `by.var` values.

`results`: A 5-dimensional "array" containing the results of the backtest.

`ret.stats`: Object of class "array" containing return statistics for the backtest.

`turnover`: Object of class "array" containing turnover statistics for the backtest.

`natural`: Object of class "logical" expressing whether or not the intervals between observations, as specified by `date.var`, and returns, as specified by `ret.var`, match. If the interval between dates is one month, the interval between returns should also be one month.

`do.spread`: Object of class "logical". If `TRUE` the `summary` method displays information about the spread between the extreme quantiles. If `FALSE` this information is suppressed. Defaults to `TRUE`.

`by.period`: Object of class "logical". If `TRUE` the quantiles are recalculated within each date period. If `FALSE` the quantiles are calculated all at once. Defaults to `TRUE`.

`overlaps`: An object of class "numeric" which specifies the number of prior periods to include in the current period's portfolio weights calculation. If `overlaps` is the default of 1, backtest behaves as usual and only uses a period's own data to determine its portfolio. If `overlaps` is set to `n > 1`, a period's portfolio comprises the weighted mean of portfolio weights from the previous `n` periods, with period `n` having a weight of `1/n`.

## Methods

- show** signature(object = "backtest"): Prints the variables used in this backtest.
- summary** signature(object = "backtest"): Prints the results of the backtest.
- summaryStats** signature(object = "backtest"): Returns a data frame with spreads for each date.var value and each in.var.
- means** signature(object = "backtest"): Returns a list of matrices, with one matrix for each in.var, where the value of each cell is the mean of the returns for that in.var and by.var combination.
- counts** signature(object = "backtest"): Returns a list of matrices, with one matrix for each in.var, where the value of each cell is the number of observations for that in.var and by.var combination.
- totalCounts** signature(object = "backtest"): Returns a data frame in the same format as the spreads data frame returned by summaryStats: contains the sum of counts for all buckets (or high and low buckets if argument low.high.only is set to TRUE) of non-NA in.var values that went into the spread calculations.
- marginals** signature(object = "backtest"): Returns a list of matrices, one matrix for each in.var, where the value of each cell is the number of observations for that in.var and by.var combination. Different from counts because the marginal sums have been appended to the matrices.
- naCounts** signature(object = "backtest"): Returns a list of matrices, with one matrix for each in.var, where the value of each cell is the number of NA observations for that in.var and by.var combination.
- ci** signature(object = "backtest"): Returns a matrix of confidence intervals for spreads.
- turnover** signature(object = "backtest"): Returns a data.frame of the turnovers if the backtest is natural.
- plot** signature(x = "backtest", y = "missing"): Plots returns, cumulative returns, or turnover, when passed a type argument of return, cumreturn, or turnover, respectively.

## Author(s)

Kyle Campbell <kyle.w.campbell@williams.edu>

## See Also

[backtest](#)

## Examples

```
data(starmine)
bt <- backtest(starmine, in.var = "smi", ret.var = "ret.0.1.m", by.period = FALSE)

## Summary for a pooled backtest

summary(bt)
```

```

## A natural backtest

bt <- backtest(starmine, in.var = "smi", ret.var = "ret.0.1.m",
              date.var = "date", id.var = "id", natural = TRUE, by.period = FALSE)

## Summary for a natural backtest

summary(bt)

## Other access methods

means(bt)
counts(bt)
marginals(bt)
naCounts(bt)

## Plotting methods

plot(bt, type = "turnover")
plot(bt, type = "return")
plot(bt, type = "cumreturn")

```

---

starmine

*StarMine Rankings, 1995*


---

### Description

StarMine rankings of some stocks in 1995, with corresponding returns and other data.

### Usage

```
data(starmine)
```

### Format

A data frame containing 53328 observations on the following 23 variables.

**date** Date on which the observation was recorded. The dates have a monthly frequency. Dates range from 1995-01-31 to 1995-11-30.

**id** Unique identifier for each stock.

**symbol** Company symbol.

**name** Full company name.

**country** Country of the exchange on which the company is listed. This factor has levels AUS, CHE, DEU, DNK, ESP, FIN, FRA, GBR, HKG, ITA, JPN, NLD, NOR, NZL, SWE and USA

**sector** Sector specification. This factor has levels Durbl, Enrgy, HiTec, Hlth, Manuf, Money, NoDur, Other, Shops, Telcm and Utils



`sec` An alternative sector specification. This factor has levels `CND`, `CNS`, `COM`, `ENE`, `FIN`, `HTH`, `IND`, `MAT`, `TEC` and `UTL`.

`ind` Industry specification. This factor has levels `AERDF`, `AIRLN`, `AUTOP`, `AUTOS`, `BANKS`, `BEVGS`, `BIOTC`, `BUILD`, `CHEMS`, `CNENG`, `CNFIN`, `CNMAT`, `COMEQ`, `COMPT`, `COMSS`, `CONGL`, `CPMKT`, `DICNS`, `DISTR`, `DVFIN`, `DVTEL`, `ELEQI`, `ELEQT`, `ELUTL`, `ENEQS`, `FDPRD`, `FDRET`, `GSUTL`, `HEPSV`, `HEQSP`, `HETEC`, `HOTEL`, `HSDUR`, `HSPRD`, `INSUR`, `INTSS`, `IPPET`, `ITCAT`, `ITCON`, `LEISR`, `LFSCI`, `LOGIS`, `MACHN`, `MEDIA`, `METAL`, `MGFIN`, `MLRET`, `MLUTL`, `OFFIC`, `OILGS`, `PACKG`, `PAPER`, `PHARM`, `PRPRD`, `REALE`, `REDEV`, `REITS`, `RRAIL`, `SEMIP`, `SEMIS`, `SHIPS`, `SMOKE`, `SOFTW`, `SPRET`, `TEXAP`, `TRADE`, `TRINF`, `WIREL` and `WTUTL`.

`size` `cap.usd` normalized to  $N(0,1)$ .

`smi` StarMine Indicator (`smi`) score

`liq` Liquidity of the company.

`ret.0.1.m` One-month forward return of the company.

`ret.0.6.m` Six-month forward return of the company.

`ret.1.0.m` One-month prior return of the company.

`ret.6.0.m` Six-month prior return of the company.

`ret.12.0.m` Twelve-month prior return of the company.

`mn.dollar.volume.20.d` Mean dollar volume of the past 20 days.

`md.dollar.volume.120.d` Median dollar volume of the past 120 days.

`cap.usd` Market capitalisation of the company in USD.

`cap` Market capitalisation of the company in local currency.

`sales` Annual gross sales of the company.

`net.income` Annual net income of the company.

`common.equity` Annual common equity of the company.

## Details

`starmine` contains selected attributes such as sector, market capitalisation, country, and various measures of return for a universe of approximately 6,000 stocks. The data is on a monthly frequency from January 31, 1995 to November 30, 1995.

## Note

We would like to thank StarMine Corporation for allowing us to include this data in the backtest package.

## Source

StarMine Corporation. For more information, see <http://www.starmine.com>.

## Examples

```
data(starmine)
head(starmine)
```

# Index

- \* **classes**
  - backtest-class, 5
- \* **datasets**
  - starmine, 8
- \* **file**
  - backtest, 2
- \* **package**
  - backtest-package, 2

backtest, 2, 7  
backtest-class, 5  
backtest-package, 2

ci (backtest-class), 5  
ci, backtest-method (backtest-class), 5  
counts (backtest-class), 5  
counts, backtest-method (backtest-class), 5

marginals (backtest-class), 5  
marginals, backtest-method (backtest-class), 5  
means (backtest-class), 5  
means, backtest-method (backtest-class), 5

naCounts (backtest-class), 5  
naCounts, backtest-method (backtest-class), 5

plot (backtest-class), 5  
plot, backtest, missing-method (backtest-class), 5

show, backtest-method (backtest-class), 5  
starmine, 8  
summary, backtest-method (backtest-class), 5  
summaryStats (backtest-class), 5  
summaryStats, backtest-method (backtest-class), 5

totalCounts (backtest-class), 5  
totalCounts, backtest-method (backtest-class), 5  
turnover (backtest-class), 5  
turnover, backtest-method (backtest-class), 5