

Package ‘IndexConstruction’

March 29, 2019

Type Package

Title Index Construction for Time Series Data

Version 0.1-1

Date 2019-03-25

Author Simon Trimborn <simon.trimborn@nus.edu.sg>

Maintainer Simon Trimborn <simon.trimborn@nus.edu.sg>

LazyLoad yes

LazyData true

Depends R (>= 2.10)

Imports KernSmooth, fGarch, lubridate, xts, RcppBDT, zoo

Description Derivation of indexes for benchmarking purposes. The methodology of the CRyptocurrency IndeX (CRIX) family with flexible number of constituents is implemented. Also functions for market capitalization and volume weighted indexes with fixed number of constituents are available. The methodology behind the functions provided gets introduced in Trimborn and Haerdle (2018) <doi:10.1016/j.jempfin.2018.08.004>.

License GPL (>= 3)

NeedsCompilation no

Repository CRAN

Date/Publication 2019-03-29 16:50:03 UTC

R topics documented:

CRIX	2
ECRIX	3
EFCRIX	4
IndexComp	5
IndexMemberSelection	7
IndexMembersUpdate	9
IndexUpdate	10
LCRIX	11
LECRIX	12

LEFCRIX	13
market	14
price	14
RelativeWeights	15
SwitchDates	15
vol	16
Index	18

 CRIX

Derivation of the CRIX index

Description

CRIX is a wrapper function for IndexComp which derives the index according the methodology for CRIX as described in Trimborn and Haerdle (2018) and visualized on <http://thecrix.de/>.

Usage

```
CRIX(market, price, vol = NULL, days.line)
```

Arguments

market	An xts object with the market capitalization data. The default is NULL, an entry is necessary if weighting is set to "market".
price	An xts object with the price data. An entry is always required.
vol	An xts object with the trading volume (liquidity) data. The default is NULL, an entry is necessary iw weighting is set to "volume".
days.line	The days of the month to perform the recalculation on. Can be calculated from SwitchDates.

Details

For more details, please see the methodology section of the paper Trimborn and Haerdle (2018).

Value

A list, entry 1 is the optimal index, entry 2 the index of all constituents, entry 3 the index of all constituents rebased at the index (entry 1) each time after altering the number of index constituents which is useful for comparisons with the market, entry 4 the number of assets available for analysis in each period, entry 5 the absolute weight given to each assets price in the respective periods

References

Trimborn, S. and Haerdle, W.K. (2018). CRIX an Index for cryptocurrencies, *Journal of Empirical Finance* 49, pp. 107-122. <https://doi.org/10.1016/j.jempfin.2018.08.004>

Examples

```

data(CryptoData)

price = price["2014-03-31::2015-01-31"]
vol = vol["2014-03-31::2015-01-31"]
days.line = SwitchDates(price, specificDate = "1")

CRIX(market = market, price = price, vol = vol, days.line = days.line)

```

ECRIX

*Derivation of the ECRIX index***Description**

ECRIX is a wrapper function for IndexComp which derives the index according the methodology for ECRIX as described in Trimborn and Haerdle (2018) and visualized on <http://thecrix.de/>.

Usage

```
ECRIX(market, price, vol = NULL, days.line)
```

Arguments

market	An xts object with the market capitalization data. The default is NULL, an entry is necessary if weighting is set to "market".
price	An xts object with the price data. An entry is always required.
vol	An xts object with the trading volume (liquidity) data. The default is NULL, an entry is necessary iw weighting is set to "volume".
days.line	The days of the month to perform the recalculation on. Can be calculated from SwitchDates.

Details

For more details, please see the methodology section of the paper Trimborn and Haerdle (2018).

Value

A list, entry 1 is the optimal index, entry 2 the index of all constituents, entry 3 the index of all constituents rebased at the index (entry 1) each time after altering the number of index constituents which is useful for comparisons with the market, entry 4 the number of assets available for analysis in each period, entry 5 the absolute weight given to each assets price in the respective periods

References

Trimborn, S. and Haerdle, W.K. (2018). CRIX an Index for cryptocurrencies, *Journal of Empirical Finance* 49, pp. 107-122. <https://doi.org/10.1016/j.jempfin.2018.08.004>

Examples

```

data(CryptoData)

price = price["2014-03-31::2015-01-31"]
vol = vol["2014-03-31::2015-01-31"]
days.line = SwitchDates(price, specificDate = "1")

EFCRIX(market = market, price = price, vol = vol, days.line = days.line)

```

EFCRIX

*Derivation of the EFCRIX index***Description**

EFCRIX is a wrapper function for IndexComp which derives the index according the methodology for EFCRIX as described in Trimborn and Haerdle (2018) and visualized on <http://thecrix.de/>.

Usage

```
EFCRIX(market, price, vol = NULL, days.line)
```

Arguments

market	An xts object with the market capitalization data. The default is NULL, an entry is necessary if weighting is set to "market".
price	An xts object with the price data. An entry is always required.
vol	An xts object with the trading volume (liquidity) data. The default is NULL, an entry is necessary iw weighting is set to "volume".
days.line	The days of the month to perform the recalculation on. Can be calculated from SwitchDates.

Details

For more details, please see the methodology section of the paper Trimborn and Haerdle (2018).

Value

A list, entry 1 is the optimal index, entry 2 the index of all constituents, entry 3 the index of all constituents rebased at the index (entry 1) each time after altering the number of index constituents which is useful for comparisons with the market, entry 4 the number of assets available for analysis in each period, entry 5 the absolute weight given to each assets price in the respective periods

References

Trimborn, S. and Haerdle, W.K. (2018). CRIX an Index for cryptocurrencies, *Journal of Empirical Finance* 49, pp. 107-122. <https://doi.org/10.1016/j.jempfin.2018.08.004>

Examples

```

data(CryptoData)

price = price["2014-03-31::2014-10-31"]
vol = vol["2014-03-31::2014-10-31"]
days.line = SwitchDates(price, specificDate = "1")

EFCRIX(market = market, price = price, vol = vol, days.line = days.line)

```

IndexComp

*Index derivation for price and liquidity indices***Description**

IndexComp derives an Index from the given price and market capitalization or liquidity data. The number of constituents can be fixed or being chosen based on the methodology from Trimborn and Haerdle (2018).

Usage

```

IndexComp(market, price, vol = NULL, weighting = "market", weighting.all = "market",
IC = "AIC", EvalSeq = c("Sequential", "AllTogether"),
optimum = c("local", "global"), start.const = 1, steps = 1, fixed.value = NULL,
base.value = 1000, derivation.period = 1, derivation.period.ic = 3, days.line)

```

Arguments

market	An xts object with the market capitalization data. The default is NULL, an entry is necessary if weighting is set to "market".
price	An xts object with the price data. An entry is always required.
vol	An xts object with the trading volume (liquidity) data. The default is NULL, an entry is necessary iw weighting is set to "volume".
weighting	The weighting scheme to be applied. "market" refers to weighting by market capitalization, "volume" refers to weighting by trading volume.
weighting.all	The weighting scheme to be applied to the full market index. "market" refers to weighting by market capitalization, "volume" refers to weighting by trading volume.
IC	Information Criterion to be used for the evaluation of the appropriate index to be used. Possible entries are "AIC", "GCV", "GFCV", "SH", "Cp" and "FPE".
EvalSeq	Indicates how the evaluation of the candidate indices by the IC shall be performed. "AllTogether" evaluates all indices against each other, "Sequential" evaluates always two consecutive indices against each other.
optimum	Define how to choose the optimal index. Either a "local" optimum is chosen, thus the derivation stops the first time the results become worse under the chosen IC, or a "global" optimum is chosen, thus all indices are derived and the best fitting one under the IC is chosen.

<code>start.const</code>	The number of constituents to start constructing the indices with. The default is 1.
<code>steps</code>	The step width for the number of constituents to construct the next index from. The default is 1.
<code>fixed.value</code>	In case no IC for the number of constituents for the index shall be applied, give the number of constituents the index shall contain. In that case, "IC", "EvalSeq", "optimum", "start.const" and "steps" are inactive parameters. The default is NULL.
<code>base.value</code>	The starting value for the index. The default is 1000.
<code>derivation.period</code>	The number of month after which the weights of the index are reallocated. The default is 1.
<code>derivation.period.ic</code>	The number of month after which the composition of the index is derived again, thus the number of constituents is reevaluated. The default is 3.
<code>days.line</code>	The days of the month to perform the recalculation on. Can be calculated from SwitchDates.

Details

For more details, please see the methodology section of the paper Trimborn and Haerdle (2018).

Value

A list, entry 1 is the optimal index, entry 2 the index of all constituents, entry 3 the index of all constituents rebased at the index (entry 1) each time after altering the number of index constituents which is useful for comparisons with the market, entry 4 the number of assets available for analysis in each period, entry 5 the absolute weight given to each assets price in the respective periods

References

Trimborn, S. and Haerdle, W.K. (2018). CRIX an Index for cryptocurrencies, *Journal of Empirical Finance* 49, pp. 107-122. <https://doi.org/10.1016/j.jempfin.2018.08.004>

Examples

```
data(CryptoData)

price = price["2014-03-31::2015-01-31"]
market = market["2014-03-31::2015-01-31"]
vol = vol["2014-03-31::2015-01-31"]
days.line = SwitchDates(price, specificDate = "1")

IndexComp(market = market, price = price, vol = vol, weighting = "market",
weighting.all = "market", IC = "AIC", EvalSeq = "Sequential", optimum = "local",
start.const = 5, steps = 5, days.line = days.line)
```

IndexMemberSelection *Number of Index Members Derivation*

Description

IndexMemberSelection derives the number of index members for the coming period based on an Information Criterion, e.g. AIC. The methodology is according to Trimborn and Haerdle (2018). The method derives the new weights according to the specifications of the weight reevaluation. The function expects the data period provided to be twice the number of months specified in derivation.period.ic. In case of a mismatch, a warning is given.

Usage

```
IndexMemberSelection(market, price, vol, weighting = "market",
weighting.all = "market", IC = "AIC", EvalSeq = c("Sequential", "AllTogether"),
optimum = c("local", "global"), start.const = 1, steps = 1, fixed.value = NULL,
derivation.period = 1, derivation.period.ic = 3, base.value = 1000, days.line)
```

Arguments

market	An xts object with the market capitalization data. The default is NULL, an entry is necessary if weighting is set to "market".
price	An xts object with the price data. An entry is always required.
vol	An xts object with the trading volume (liquidity) data. The default is NULL, an entry is necessary if weighting is set to "volume".
weighting	The weighting scheme to be applied. "market" refers to weighting by market capitalization, "volume" refers to weighting by trading volume.
weighting.all	The weighting scheme to be applied to the full market index. "market" refers to weighting by market capitalization, "volume" refers to weighting by trading volume.
IC	Information Criterion to be used for the evaluation of the appropriate index to be used. Possible entries are "AIC", "GCV", "GFCV", "SH", "Cp" and "FPE".
EvalSeq	Indicates how the evaluation of the candidate indices by the IC shall be performed. "AllTogether" evaluates all indices against each other, "Sequential" evaluates always two consecutive indices against each other.
optimum	Define how to choose the optimal index. Either a "local" optimum is chosen, thus the derivation stops the first time the results become worse under the chosen IC, or a "global" optimum is chosen, thus all indices are derived and the best fitting one under the IC is chosen.
start.const	The number of constituents to start constructing the indices with. The default is 1.
steps	The step width for the number of constituents to construct the next index from. The default is 1.

<code>fixed.value</code>	In case no IC for the number of constituents for the index shall be applied, give the number of constituents the index shall contain. In that case, "IC", "EvalSeq", "optimum", "start.const" and "steps" are inactive parameters. The default is NULL.
<code>base.value</code>	The starting value for the index. The default is 1000.
<code>derivation.period</code>	The number of month after which the weights of the index are reallocated. The default is 1.
<code>derivation.period.ic</code>	The number of month after which the composition of the index is derived again, thus the number of constituents is reevaluated. The default is 3.
<code>days.line</code>	The days of the month to perform the recalculation on. Can be calculated from SwitchDates.

Details

IndexMemberSelection derives the number of index members for the coming period based on an Information Criterion, e.g. AIC. The methodology is according to Trimborn and Haerdle (2018). The method derives the new weights according to the specifications of the weight reevaluation. The function expects the data period provided to be twice the number of months specified in `derivation.period.ic`. In case of a mismatch, a warning is given. The data from the first period are used to derived the likelihood, the second period is used for out-of-sample derivation of the number of constituents. Hence for a 3 month reevaluation period, 6 month of data are required by this function. For more details, please see the methodology section of the paper Trimborn and Haerdle (2018).

Value

Returns the number of index members for application in the next period.

References

Trimborn, S. and Haerdle, W.K. (2018). CRIX an Index for cryptocurrencies, *Journal of Empirical Finance* 49, pp. 107-122. <https://doi.org/10.1016/j.jempfin.2018.08.004>

Examples

```
data(CryptoData)

price = price["2016-07-31::2017-01-31"]
market = market["2016-07-31::2017-01-31"]
vol = vol["2016-07-31::2017-01-31"]
days.line = SwitchDates(price, specificDate = "1")

IndexMemberSelection(market = market, price = price, vol = vol,
weighting = "market", weighting.all = "market", IC = "AIC", EvalSeq = "Sequential",
optimum = "local", start.const = 5, steps = 5, days.line = days.line)
```

IndexMembersUpdate *Reevaluation of Index constituents weights*

Description

IndexMembersUpdate derives the new weights for the coming period. The methodology is according to Trimborn and Haerdle (2018). The method derives the new weights over the data period provided. The data input defines the length of the period, hence it can be different from full month.

Usage

```
IndexMembersUpdate(market, price, vol, weighting, index.const, last.value)
```

Arguments

market	An xts object with the market capitalization data. The default is NULL, an entry is necessary if weighting is set to "market".
price	An xts object with the price data. An entry is always required.
vol	An xts object with the trading volume (liquidity) data. The default is NULL, an entry is necessary if weighting is set to "volume".
weighting	The weighting scheme to be applied. "market" refers to weighting by market capitalization, "volume" refers to weighting by trading volume.
index.const	Number of Index constituents. The number can be derived from IndexComp, IndexMemberSelection or be chosen by alternative means.
last.value	The last index value before rederivation.

Details

IndexMembersUpdate derives the new weights for the coming period. The methodology is according to Trimborn and Haerdle (2018). The method derives the new weights over the data period provided. The data input defines the length of the period, hence it can be different from full month. For more details, please see the methodology section of the paper Trimborn and Haerdle (2018).

Value

A list, entry 1 is the ordered names of index members, entry 2 the respective consideration of the index constituents, entry 3 the weights of the index members which gives multiplied with entry 2 the actual weight and entry 4 the new divisor of the index.

References

Trimborn, S. and Haerdle, W.K. (2018). CRIX an Index for cryptocurrencies, *Journal of Empirical Finance* 49, pp. 107-122. <https://doi.org/10.1016/j.jempfin.2018.08.004>

Examples

```

data(CryptoData)

price = price["2017-01-01::2017-01-31"]
market = market["2017-01-01::2017-01-31"]
vol = vol["2017-01-01::2017-01-31"]
IndexMembersUpdate(market = market, price = price, vol = vol,
weighting = "market", index.const = 5, last.value = 1000)

```

IndexUpdate

Updating an existing index with new index values

Description

IndexUpdate derives the next values of an Index from the given price, weights and its divisor.

Usage

```
IndexUpdate(price, index.weights, divisor)
```

Arguments

price	An xts object with the price data. An entry is always required.
index.weights	A vector with the absolute weights expressed as number of shares of each asset. The weights are provided by IndexComp. They can be also easily derived from the market capitalization by dividing with the respective price.
divisor	The divisor required for the index derivation. The divisor is provided by IndexComp. For details on its derivation, see Trimborn and Haerdle (2018).

Details

For more details, please see the methodology section of the paper Trimborn and Haerdle (2018).

Value

The next value(s) of the Index.

References

Trimborn, S. and Haerdle, W.K. (2018). CRIX an Index for cryptocurrencies, *Journal of Empirical Finance* 49, pp. 107-122. <https://doi.org/10.1016/j.jempfin.2018.08.004>

Examples

```
data(CryptoData)

const.names = c("btc", "eth", "xrp", "ltc", "xmr")
index.weights = c(16136712, 88440036, 36856524148, 49589181, 13859864)
divisor = 17185084

IndexUpdate(price["2017-02-01", const.names], index.weights = index.weights, divisor = divisor)
```

LCRIX

*Derivation of the LCRIX index***Description**

LCRIX is a wrapper function for IndexComp which derives the index according the methodology for LCRIX as described in Trimborn and Haerdle (2018) and visualized on <http://thecrix.de/>.

Usage

```
LCRIX(market, price, vol = NULL, days.line)
```

Arguments

market	An xts object with the market capitalization data. The default is NULL, an entry is necessary if weighting is set to "market".
price	An xts object with the price data. An entry is always required.
vol	An xts object with the trading volume (liquidity) data. The default is NULL, an entry is necessary iw weighting is set to "volume".
days.line	The days of the month to perform the recalculation on. Can be calculated from SwitchDates.

Details

For more details, please see the methodology section of the paper Trimborn and Haerdle (2018).

Value

A list, entry 1 is the optimal index, entry 2 the index of all constituents, entry 3 the index of all constituents rebased at the index (entry 1) each time after altering the number of index constituents which is useful for comparisons with the market, entry 4 the number of assets available for analysis in each period, entry 5 the absolute weight given to each assets price in the respective periods

References

Trimborn, S. and Haerdle, W.K. (2018). CRIX an Index for cryptocurrencies, *Journal of Empirical Finance* 49, pp. 107-122. <https://doi.org/10.1016/j.jempfin.2018.08.004>

Examples

```

data(CryptoData)

price = price["2014-03-31::2014-10-31"]
market = market["2014-03-31::2014-10-31"]
vol = vol["2014-03-31::2014-10-31"]
days.line = SwitchDates(price, specificDate = "1")

LECRIX(market = market, price = price, vol = vol, days.line = days.line)

```

LECRIX

*Derivation of the LECRIX index***Description**

LECRIX is a wrapper function for IndexComp which derives the index according the methodology for LECRIX as described in Trimborn and Haerdle (2018) and visualized on <http://thecrix.de/>.

Usage

```
LECRIX(market, price, vol = NULL, days.line)
```

Arguments

market	An xts object with the market capitalization data. The default is NULL, an entry is necessary if weighting is set to "market".
price	An xts object with the price data. An entry is always required.
vol	An xts object with the trading volume (liquidity) data. The default is NULL, an entry is necessary iw weighting is set to "volume".
days.line	The days of the month to perform the recalculation on. Can be calculated from SwitchDates.

Details

For more details, please see the methodology section of the paper Trimborn and Haerdle (2018).

Value

A list, entry 1 is the optimal index, entry 2 the index of all constituents, entry 3 the index of all constituents rebased at the index (entry 1) each time after altering the number of index constituents which is useful for comparisons with the market, entry 4 the number of assets available for analysis in each period, entry 5 the absolute weight given to each assets price in the respective periods

References

Trimborn, S. and Haerdle, W.K. (2018). CRIX an Index for cryptocurrencies, *Journal of Empirical Finance* 49, pp. 107-122. <https://doi.org/10.1016/j.jempfin.2018.08.004>

Examples

```

data(CryptoData)

price = price["2014-03-31::2014-10-31"]
market = market["2014-03-31::2014-10-31"]
vol = vol["2014-03-31::2014-10-31"]
days.line = SwitchDates(price, specificDate = "1")

LEFCRIX(market = market, price = price, vol = vol, days.line = days.line)

```

LEFCRIX

*Derivation of the LEFCRIX index***Description**

LEFCRIX is a wrapper function for IndexComp which derives the index according the methodology for LEFCRIX as described in Trimborn and Haerdle (2018) and visualized on <http://thecrix.de/>.

Usage

```
LEFCRIX(market, price, vol = NULL, days.line)
```

Arguments

market	An xts object with the market capitalization data. The default is NULL, an entry is necessary if weighting is set to "market".
price	An xts object with the price data. An entry is always required.
vol	An xts object with the trading volume (liquidity) data. The default is NULL, an entry is necessary iw weighting is set to "volume".
days.line	The days of the month to perform the recalculation on. Can be calculated from SwitchDates.

Details

For more details, please see the methodology section of the paper Trimborn and Haerdle (2018).

Value

A list, entry 1 is the optimal index, entry 2 the index of all constituents, entry 3 the index of all constituents rebased at the index (entry 1) each time after altering the number of index constituents which is useful for comparisons with the market, entry 4 the number of assets available for analysis in each period, entry 5 the absolute weight given to each assets price in the respective periods

References

Trimborn, S. and Haerdle, W.K. (2018). CRIX an Index for cryptocurrencies, *Journal of Empirical Finance* 49, pp. 107-122. <https://doi.org/10.1016/j.jempfin.2018.08.004>

Examples

```
data(CryptoData)

price = price["2014-03-31::2014-10-31"]
market = market["2014-03-31::2014-10-31"]
vol = vol["2014-03-31::2014-10-31"]
days.line = SwitchDates(price, specificDate = "1")

LEFCRIX(market = market, price = price, vol = vol, days.line = days.line)
```

market	<i>Market capitalization data for Cryptocurrencies.</i>
--------	---

Description

The dataset contains market capitalization information for cryptocurrencies.

Usage

```
data(CryptoData)
```

Format

A dataset with a xts matrix. Load the R library xts for proper visualization of the dataset.

Source

The dataset was provided by CoinGecko. Up-to-date data are accessible via <https://www.coingecko.com/api>.

price	<i>Pricing data for Cryptocurrencies.</i>
-------	---

Description

The dataset contains pricing information for cryptocurrencies.

Usage

```
data(CryptoData)
```

Format

A dataset with a xts matrix. Load the R library xts for proper visualization of the dataset.

Source

The dataset was provided by CoinGecko. Up-to-date data are accessible via <https://www.coingecko.com/api>.

RelativeWeights	<i>Retrieving the relative weights of the assets in the index</i>
-----------------	---

Description

RelativeWeights retrieves the relative weights of the assets in the index from the absolute weights expressed in shares of the assets. The latter is a direct output of IndexComp.

Usage

```
RelativeWeights(price, index.weights)
```

Arguments

price	An xts object with the price data. An entry is always required.
index.weights	A vector with the absolute weights expressed as number of shares of each asset. The weights are provided by IndexComp. They can be also easily derived from the market capitalization by dividing with the respective price.

Value

The relative weights of the assets in the Index.

References

Trimborn, S. and Haerdle, W.K. (2018). CRIX an Index for cryptocurrencies, *Journal of Empirical Finance* 49, pp. 107-122. <https://doi.org/10.1016/j.jempfin.2018.08.004>

Examples

```
data(CryptoData)

const.names = c("btc", "eth", "xrp", "ltc", "xmr")
index.weights = c(16136712, 88440036, 36856524148, 49589181, 13859864)

RelativeWeights(price = price["2017-02-01", const.names], index.weights = index.weights)
```

SwitchDates	<i>Deriving the dates on which the index constituents are going to be reevaluated</i>
-------------	---

Description

SwitchDates derives the dates on which the index constituents are going to be reevaluated.

Usage

```
SwitchDates(price, specificDate = NULL, WeekDay = NULL, Appearance = 1)
```

Arguments

price	An xts object with the price data. An entry is always required.
specificDate	A specific date of each month on which the index members get reevaluated. A common date would be the 1st of each month or the 15th of each month. specificDate is dominating WeekDay.
WeekDay	Only active when specificDate is NULL. A specific weekday of each month on which the index members get reevaluated. The input has to be a character describing the weekday in English. By default the first weekday with this appearance is returned. The argument Appearance defines if it is the 1st, 2nd or another appearance of this weekday. E.g. the 3rd Friday of each month can be returned.
Appearance	Defines if the 1st, 2nd or another appearance of a weekday gets returned. E.g. the 3rd Friday of each month can be returned. Only active when specificDate is NULL. The argument works in combination with WeekDay.

Value

A vector of class date with the respective dates on which the index members become reevaluated. This is a necessary input to IndexComp.

References

Trimborn, S. and Haerdle, W.K. (2018). CRIX an Index for cryptocurrencies, *Journal of Empirical Finance* 49, pp. 107-122. <https://doi.org/10.1016/j.jempfin.2018.08.004>

Examples

```
data(CryptoData)

SwitchDates(price, specificDate = "1")
```

 vol

Volume data for Cryptocurrencies.

Description

The dataset contains trading volume information for cryptocurrencies.

Usage

```
data(CryptoData)
```


Format

A dataset with a xts matrix. Load the R library xts for proper visualization of the dataset.

Source

The dataset was provided by CoinGecko. Up-to-date data are accessible via <https://www.coingecko.com/api>.

Index

CRIX, [2](#)

ECRIX, [3](#)

EFCRIX, [4](#)

IndexComp, [5](#)

IndexMemberSelection, [7](#)

IndexMembersUpdate, [9](#)

IndexUpdate, [10](#)

LCRIX, [11](#)

LECRIX, [12](#)

LEFCRIX, [13](#)

market, [14](#)

price, [14](#)

RelativeWeights, [15](#)

SwitchDates, [15](#)

vol, [16](#)