
Radium

Release 0.1.0

Ivan Erlic and Callum Rai

Mar 04, 2021

CONTENTS:

1	Equity	1
2	Pair	3
3	Strategy	7
	Python Module Index	9
	Index	11

EQUITY

class `radium.Equity` (*symbol, start_date, end_date, key*)

Bases: `object`

Class for a single equity between two dates.

data

Contains all daily signals with date as index

Type `pd.DataFrame`

high

Contains daily high prices with date as index

Type `pd.Series`

low

Contains daily low prices with date as index

Type `pd.Series`

open

Contains daily open prices with date as index

Type `pd.Series`

closed

Contains daily adjusted closed prices with date as index

Type `pd.Series`

symbol

Symbol for equity as found on exchange

Type `str`

start_date

First date of interest

Type `datetime.date`

end_date

Last date of interest

Type `datetime.date`

key

Alpha-vantage API key

Type `str`

Initialises equity class

Parameters

- **symbol** (*str*) – Symbol for equity as found on exchange
- **start_date** (*str or datetime or datetime.date*) – First date of interest in YYYY-MM-DD form
- **end_date** (*str of datetime or datetime.date*) – Last date of interest in YYYY-MM-DD form
- **key** (*str*) – Alpha-vantage API key

Raises

- **ValueError** – API Key is invalid Equity symbol does not exist End date is same as or before start date
- **RuntimeError** – API Call limit reached

plot (*start_date=None, end_date=None*)

Plots closed prices of equity between two dates as a line graph

Parameters

- **start_date** (*(optional) str or datetime or datetime.date*) – First date to plot in YYYY-MM-DD form, defaults to equity start date
- **end_date** (*(optional) str of datetime or datetime.date*) – Last date to plot in YYYY-MM-DD form, defaults to equity end date

Raises ValueError – End date is same as or before start date

class radium.**Pair**(*equity1, equity2*)

Bases: object

Class for a pair of equities.

price_spread

Price spread of equities for self.hedge_ratios

Type float np.ndarray[]

equity1

Type *radium.Equity*

equity2

Type *radium.Equity*

start_date

Type datetime.date

end_date

Type datetime.date

hedge_ratios

Day-wise pairs of hedge ratios

Type float np.ndarray[][2]

Initialise Pair class

Parameters

- **equity1** (*radium.Equity*) –
- **equity2** (*radium.Equity*) –

Raises

- **TypeError** – If equity1 or equity2 is not of type radium.Equity.
- **ValueError** – If equity1 and equity2 do not share any date ranges

budget (*hedge_ratio, dec*)

Calculates budget needed to buy integer number of equities.

Parameters

- **hedge_ratio** (*int np.ndarray[2]*) – Hedge ratios of pair
- **dec** (*int*) – Number of decimals to truncate to

Returns `budget` – Budget needed rounded to 2 d.p.

Return type `float`

Raises

- **TypeError** – If `hedge_ratio` isn't a list of floats, or `dec` isn't an integer.
- **ValueError** – If `hedge_ratio` isn't length 2 or `dec` < 0.

hedge (*method*, *lookback*)

Calculates the `hedge_ratios` given a method and lookback and stores it in `self.hedge_ratios`

Parameters

- **method** (*str*) – Method for calculating hedge ratios ('ols')
- **lookback** (*int*) – Number of signals to lookback on when calculating hedge ratios

Raises

- **TypeError** – If method isn't a string. If lookback isn't an integer.
- **ValueError** – If lookback <= 0. If method isn't available.

Notes

Available methods: 'OLS'

plot_closed (*start_date=None*, *end_date=None*)

Plots closed prices of both equities between two dates as a line graph

Parameters

- **start_date** ((*optional*) *str* or *datetime* or *datetime.date*) – First date to plot in YYYY-MM-DD form, defaults to equity start date
- **end_date** ((*optional*) *str* of *datetime* or *datetime.date*) – Last date to plot in YYYY-MM-DD form, defaults to equity end date

Raises **ValueError** – End date is same as or before start date

plot_price_spread ()

Plots price spread of the pair given `hedge_ratios`

Raises **Exception** – If `self.hedge_ratios` is not defined.

property price_spread

Price spread of equities for `self.hedge_ratios`

Raises **TypeError** – If `self.hedge_ratios` isn't defined.

Notes

Spread calculated using $y = h1*y1 + h2*y2$.

Type `float np.ndarray[]`

`radium.pair.cadf_test` (*pair*)

Conducts a Cointegrated Augmented Dickey Fuller Test on a pair of equities.

Prints t-statistic, p-value and critical values

Parameters **pair** (`radium.Pair`) –

Raises `TypeError` – If pair is not of type `radium.Pair`.

`radium.pair.johansen_test` (*pair*)

Conducts a Johansen Test on a pair of equities.

Print trace/eigenvalue statistics and critical values.

Parameters `pair` (`radium.Pair`) –

Raises `TypeError` – If pair is not of type `radium.Pair`.

STRATEGY

class `radium.strategy.BollingerPair` (*pair*, *entry_z*, *exit_z*, *lookback*)

Bases: `radium.strategy.pair_strategy.PairStrategy`

Bollinger band strategy on a pair of equities

pair

Type *radium.Pair*

entry_z

Z-score to enter position at

Type float

exit_z

Z-score to exit position at

Type float

lookback

Days looked back at on when calculating optimal positions

Type int

th_positions

Theoretical optimum positions calculated by strategy

Type `np.float[][2]`

See also:

radium.PairStrategy Parent class

Initialises strategy

Parameters

- **pair** (*radium.Pair*) –
- **entry_z** (*float*) – Z-score to enter position at
- **exit_z** (*float*) – Z-score to exit position at
- **lookback** (*int*) – Days to lookback on when calculating optimal positions

calculate_positions ()

Calculates optimum positions calculated by strategy

Returns **th_positions**

Return type `np.float[][2]`

```
class radium.strategy.PairStrategy(pair)
    Bases: object

    Base class for equity pair trading strategies.

    daily_returns
        Daily returns from trading by optimal positions
        Type float np.ndarray[]

    cum_returns
        Daily cumulative returns from trading by optimal positions.
        Type float np.ndarray[]

    CAGR
        Compound annual growth rate based upon daily cumulative returns.
        Type float

    sharpe
        Sharpe ratio of the strategy.
        Type float

    pair
        Type radium.pair

    Initialises PairStrategy class

    Parameters pair (radium.Pair) –

    Raises TypeError – If pair isn't radium.Pair.

property CAGR
    Compound annual growth rate based upon daily cumulative returns.
        Raises Exception – If self.th_positions is not defined.
        Type float

property cum_returns
    Daily cumulative returns from trading by optimal positions.
        Raises Exception – If self.th_positions is not defined
        Type float np.ndarray[]

property daily_returns
    Daily returns from trading by optimal positions
        Raises Exception – If self.th_positions isn't defined
        Type float np.ndarray[]

property sharpe
    Sharpe ratio of the strategy.

    Measures the performance of an investment compared to a risk-free asset, after adjusting for its risk.
    Exception If self.th_positions is not defined

    Type float
```

PYTHON MODULE INDEX

r

radium.pair, 4

B

BollingerPair (class in *radium.strategy*), 7
 budget () (*radium.Pair* method), 3

C

cadf_test () (in module *radium.pair*), 4
 CAGR (*radium.strategy.PairStrategy* attribute), 8
 CAGR () (*radium.strategy.PairStrategy* property), 8
 calculate_positions () (*radium.strategy.BollingerPair* method), 7
 closed (*radium.Equity* attribute), 1
 cum_returns (*radium.strategy.PairStrategy* attribute), 8
 cum_returns () (*radium.strategy.PairStrategy* property), 8

D

daily_returns (*radium.strategy.PairStrategy* attribute), 8
 daily_returns () (*radium.strategy.PairStrategy* property), 8
 data (*radium.Equity* attribute), 1

E

end_date (*radium.Equity* attribute), 1
 end_date (*radium.Pair* attribute), 3
 entry_z (*radium.strategy.BollingerPair* attribute), 7
 Equity (class in *radium*), 1
 equity1 (*radium.Pair* attribute), 3
 equity2 (*radium.Pair* attribute), 3
 exit_z (*radium.strategy.BollingerPair* attribute), 7

H

hedge () (*radium.Pair* method), 4
 hedge_ratios (*radium.Pair* attribute), 3
 high (*radium.Equity* attribute), 1

J

johansen_test () (in module *radium.pair*), 5

K

key (*radium.Equity* attribute), 1

L

lookback (*radium.strategy.BollingerPair* attribute), 7
 low (*radium.Equity* attribute), 1

M

module
 radium.pair, 4
 radium.strategy, 7

O

open (*radium.Equity* attribute), 1

P

Pair (class in *radium*), 3
 pair (*radium.strategy.BollingerPair* attribute), 7
 pair (*radium.strategy.PairStrategy* attribute), 8
 PairStrategy (class in *radium.strategy*), 7
 plot () (*radium.Equity* method), 2
 plot_closed () (*radium.Pair* method), 4
 plot_price_spread () (*radium.Pair* method), 4
 price_spread (*radium.Pair* attribute), 3
 price_spread () (*radium.Pair* property), 4

R

radium.pair
 module, 4
radium.strategy
 module, 7

S

sharpe (*radium.strategy.PairStrategy* attribute), 8
 sharpe () (*radium.strategy.PairStrategy* property), 8
 start_date (*radium.Equity* attribute), 1
 start_date (*radium.Pair* attribute), 3
 symbol (*radium.Equity* attribute), 1

T

th_positions (*radium.strategy.BollingerPair* attribute), 7