
turingquant

Release 0.1.6

Oct 21, 2020

Contents:

1 Módulos	3
2 Módulo metrics	5
3 Módulo benchmark	9
4 Módulo support	11
Python Module Index	13
Index	15

turingquant é uma biblioteca para coleta, análise e backtesting de ativos e estratégias financeiras. O projeto está em desenvolvimento ativo pelos membros de Finanças Quantitativas do [Grupo Turing](#).

A API de onde obtemos os dados fundamentalistas é a Alpha Vantage e você pode [obter a chave de uso gratuitamente](#). Essa chave será necessária sempre que você utilizar as funções `daily` e `intraday`.

`pip install turingquant` instala a última versão estável.

CHAPTER 1

Módulos

<i>turingquant.metrics</i>	Módulo para metrificação de ativos e retornos.
<i>turingquant.benchmark</i>	Módulo para comparação e benchmarking de ativos e retornos.
<i>turingquant.support</i>	Módulo para coletar informações de ações do mercado financeiro.

Módulo para metrificação de ativos e retornos.

`turingquant.metrics.alpha` (*start_price*, *end_price*, *dps*)

Essa função, com o fornecimento do preço final, dos dividendos por ação e do preço inicial, a calcula o alfa de um ativo.

Parameters

- **start_price** (*float*) – preço inicial.
- **end_price** (*float*) – preço final.
- **dps** (*float*) – dividendos por ação.

Returns alpha do ativo

Return type float

`turingquant.metrics.beta` (*returns*, *benchmark*)

Essa função, a partir do fornecimento dos retornos do ativo e do benchmark, calcula o beta do ativo.

Parameters

- **returns** (*pd.series*) – série com o retorno do ativo.
- **benchmark** (*pd.series*) – série com o retorno do benchmark.

Returns Beta do ativo

Return type float

`turingquant.metrics.drawdown` (*return_series*, *plot=True*)

Calcula e plota o drawdown percentual para uma série de retornos.

Parameters

- **return_series** (*pd.Series*) – série de retornos para o qual será calculado o Drawdown.
- **plot** (*bool*) – se *True*, plota um gráfico de linha com o Drawdown ao longo do tempo.

Returns uma série com os valores percentuais do Drawdown.

Return type pd.Series

`turingquant.metrics.ewma_volatility(close_prices, return_type, window, plot=True)`

Essa função possibilita a visualização da volatilidade a partir do cálculo da EWMA e da plotagem do gráfico dessa métrica ao longo de um período.

Parâmetros: `close_prices` (pd.DataFrame): série de preços de fechamento que será utilizado de base para o cálculo da EWMA; `return_type` (string): tipo de retorno (simple - 'simp' ou logarítmico - 'log') que será utilizado de base para cálculo; `window` (int): janela móvel para cálculo da EWMA; `plot` (bool): se True, plota o gráfico de linha da EWMA ao longo do tempo

Retorna: `ewma_volatility` (pd.DataFrame): um dataframe indexado à data com os valores de EWMA dos últimos `window` dias

`turingquant.metrics.garman_klass_volatility(high_prices, low_prices, close_prices, open_prices, window, time_scale=1, plot=True)`

Estima a volatilidade a partir dos seguintes preços: alta, baixa, abertura e fechamento

Parâmetros: `high_prices` (pd.DataFrame): série de preços de alta de uma ação `low_prices` (pd.DataFrame): série de preços de baixa de uma ação `close_prices` (pd.DataFrame): série de preços de fechamento de uma ação `open_prices` (pd.DataFrame): série de preços de abertura de uma ação `window` (int): janela das estimativa de volatilidade `time_scale` (int): fator de escala da volatilidade, por padrão é 1 (diária) `plot` (bool): se 'True', plota um gráfico da volatilidade móvel

Retorna: `garman_klass_vol` (pd.Series): série das estimativas de volatilidade

`turingquant.metrics.parkinson_volatility(high_prices, low_prices, window, time_scale=1, plot=True)`

Estimando a volatilidade a partir dos preços de Alta e de Baixa

Parâmetros: `high` (pd.DataFrame): série de preços de alta de uma ação `low` (pd.DataFrame): série de preços de baixa de uma ação `window` (int): janela das estimativa de volatilidade `time_scale` (int): fator de escala da volatilidade, por padrão é 1 (diária) `plot` (bool): se 'True', plota um gráfico da volatilidade móvel

Retorna: `garman_klass_vol` (pd.Series): série das estimativas de volatilidade

`turingquant.metrics.rolling_beta(returns, benchmark, window=60, plot=True)`

Plota o beta móvel para um ativo e um benchmark de referência, na forma de séries de retornos.

Parameters

- **returns** (*array*) – série de retornos para o qual o beta será calculado.
- **benchmark** (*array*) – série de retornos para usar de referência no cálculo do beta.
- **window** (*int*) – janela móvel para calcular o beta ao longo do tempo.
- **plot** (*bool*) – se *True*, plota um gráfico de linha com o beta ao longo do tempo.

Returns uma série com os valores do Beta para os últimos `window` dias. A série não possui os `window` primeiros dias.

Return type pd.Series

`turingquant.metrics.rolling_sharpe(returns, window, risk_free=0, plot=True)`

Plota o sharpe móvel para um ativo e um benchmark de referência, na forma de séries de retornos.

Parameters

- **returns** (*array*) – série de retornos para o qual o Sharpe Ratio será calculado.
- **window** (*int*) – janela móvel para calcular o Sharpe ao longo do tempo.

- **risk_free** (*float*) – valor da taxa livre de risco para cálculo do Sharpe.
- **plot** (*bool*) – se *True*, plota um gráfico de linha com o Sharpe ao longo do tempo.

Returns uma série com os valores do Sharpe para os últimos *window* dias. A série não possui os *window* primeiros dias.

Return type `pd.Series`

`turingquant.metrics.rolling_std(close_prices, return_type, window, plot=True)`

Essa função possibilita a visualização da volatilidade a partir do cálculo da desvio padrão móvel e da plotagem do gráfico dessa métrica ao longo de um período.

Parâmetros: `close_prices` (`pd.DataFrame`): série de preços de fechamento que será utilizado de base para o cálculo do desvio padrão móvel; `return_type` (`string`): tipo de retorno (`simple` - 'simp' ou `logaritmico` - 'log') que será utilizado de base para cálculo; `window` (`int`): janela móvel para cálculo do desvio padrão móvel; `plot` (`bool`): se `True`, plota o gráfico de linha do desvio padrão móvel ao longo do tempo

Retorna: `rolling_std` (`pd.DataFrame`): um dataframe indexado à data com os valores de desvio padrão móvel dos últimos `window` dias

`turingquant.metrics.sharpe_ratio(returns, risk_free=0)`

Essa função, a partir da definição do parâmetro de retorno, fornece o sharpe ratio do ativo, com base na média histórica e desvio padrão dos retornos. O `risk free` considerado é nulo.

Parameters

- **returns** (`pd.series`) – série com o retorno do ativo.
- **risk_free** (`float`) – risk free utilizado para cálculo do sharpe ratio.

Returns índice de sharpe do ativo.

Return type `float`

Módulo benchmark

Módulo para comparação e benchmarking de ativos e retornos.

```
turingquant.benchmark.benchmark(ticker, start: datetime.datetime, end: datetime.datetime,  
                                source='yahoo', plot=True)
```

Essa função fornece um plot de retorno acumulado de um ativo ao longo de um dado intervalo de tempo, definido pelos parâmetros `start` e `end`. Os dados são coletados da API do yahoo, caso haja dados faltantes, os retornos são contabilizados como nulos.

Parameters

- **ticker** (*str*) – recebe o ticker do papel que será obtido.
- **start** (*datetime*) – início do intervalo.
- **end** (*datetime*) – final do intervalo.
- **plot** (*bool*) – opcional; exhibe o gráfico caso *True*.

Returns uma série de ativos indexados com o tempo com o retorno cumulativo para o período.

Return type `pd.series`

```
turingquant.benchmark.benchmark_ibov(start: datetime.datetime, end: datetime.datetime,  
                                       source='yahoo', plot=True)
```

Essa função produz um plot da evolução do Índice Bovespa ao longo de um dado intervalo, definido pelos parâmetros `start` e `end`.

Parameters

- **start** (*datetime*) – início do intervalo.
- **end** (*datetime*) – final do intervalo.
- **plot** (*bool*) – opcional; exhibe o gráfico caso *True*.

Returns uma série temporal com o retorno acumulado do Ibovespa para o período.

Return type `pd.series`

```
turingquant.benchmark.benchmark_sp500(start: datetime.datetime, end: datetime.datetime,  
                                         source='yahoo', plot=True)
```

Essa função produz um plot da evolução do Índice S&P500 ao longo de um dado intervalo, definido pelos parâmetros *start* e *end*.

Parameters

- **start** (*datetime*) – início do intervalo.
- **end** (*datetime*) – final do intervalo.
- **plot** (*bool*) – opcional; exibe o gráfico caso *True*.

Returns uma série temporal com o retorno acumulado do S&P500 para o período.

Return type `pd.series`

Módulo para coletar informações de ações do mercado financeiro.

`turingquant.support.daily` (*key*, *ticker*, *br=True*)

Essa função entrega a cotação dia a dia de um produto negociado em bolsa com melhor formatação de dados que a biblioteca `alpha_vantage`.

Parameters

- **key** (*str*) – recebe a chave de uso do AlphaVantage
- **ticker** (*str*) – recebe o ticker do papel que será obtido
- **br** (*str*) – se *True*, adiciona “.SA” ao final do ticker, necessário para papéis brasileiros

Returns um dataframe contendo a cotação dia a dia do ativo.

Return type `pd.DataFrame`

`turingquant.support.get_annual_hpr` (*ticker*, *period=252*)

Essa função calcula o holding period return anual de junho

`turingquant.support.get_fundamentus` (*ticker*)

Essa função obtém os dados patrimoniais de empresas por meio do site `fundamentus.com.br`.

`turingquant.support.get_ibov` (*atual=True*)

Essa função obtém a composição atual do Índice Bovespa

`turingquant.support.get_sp500_tickers` ()

Essa função obtém todas o ticker de todas as ações do S&P 500.

`turingquant.support.get_tickers` (*setor='Todos'*)

Essa função obtém os tickers listados no site `fundamentus.com.br`, seja um setor específico, uma lista de setores ou todos os tickers de todos os setores.

`turingquant.support.intraday` (*key*, *ticker*, *br=True*, *interval='1min'*)

Essa função entrega a cotação intraday dos últimos 5 dias de um produto negociado em bolsa com melhor formatação de dados que a biblioteca `alpha_vantage`.

Parameters

- **key** (*str*) – recebe a chave de uso do AlphaVantage
- **ticker** (*str*) – recebe o ticker do papel que será obtido
- **br** (*bool*) – se *True*, adiciona “.SA” ao final do ticker, necessário para papéis brasileiros.
- **interval** (*str*) – recebe o período entre cada informação (1min, 5min, 15min, 30min, 60min)

Returns DataFrame contendo a cotação intraday dos últimos 5 dias.

Return type pd.DataFrame

t

turingquant, 1
turingquant.benchmark, 9
turingquant.metrics, 5
turingquant.support, 11

A

`alpha()` (in module `turingquant.metrics`), 5

B

`benchmark()` (in module `turingquant.benchmark`), 9

`benchmark_ibov()` (in module `turingquant.benchmark`), 9

`benchmark_sp500()` (in module `turingquant.benchmark`), 9

`beta()` (in module `turingquant.metrics`), 5

D

`daily()` (in module `turingquant.support`), 11

`drawdown()` (in module `turingquant.metrics`), 5

E

`ewma_volatility()` (in module `turingquant.metrics`), 6

G

`garman_klass_volatility()` (in module `turingquant.metrics`), 6

`get_annual_hpr()` (in module `turingquant.support`), 11

`get_fundamentus()` (in module `turingquant.support`), 11

`get_ibov()` (in module `turingquant.support`), 11

`get_sp500_tickers()` (in module `turingquant.support`), 11

`get_tickers()` (in module `turingquant.support`), 11

I

`intraday()` (in module `turingquant.support`), 11

P

`parkinson_volatility()` (in module `turingquant.metrics`), 6

R

`rolling_beta()` (in module `turingquant.metrics`), 6

`rolling_sharpe()` (in module `turingquant.metrics`), 6

`rolling_std()` (in module `turingquant.metrics`), 7

S

`sharpe_ratio()` (in module `turingquant.metrics`), 7

T

`turingquant` (module), 1

`turingquant.benchmark` (module), 9

`turingquant.metrics` (module), 5

`turingquant.support` (module), 11