
turingquant

Versão 0.2.1

12 mai., 2021

Contents:

1	Módulos	3
2	Módulo metrics	5
3	Módulo benchmark	11
4	Módulo support	13
5	Módulo optimizers	17
	Índice de Módulos do Python	19
	Índice	21

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.

CAPÍTULO 1

Módulos

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

Módulo para metrificação de ativos e retornos.

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

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

Parâmetros

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

Retorno alpha do ativo

Tipo de retorno float

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

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

Parâmetros

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

Retorno Beta do ativo

Tipo de retorno float

`turingquant.metrics.cagr` (*returns*, *time_scale=252*)

Calcula o CAGR que é a taxa composta de crescimento anual. :param returns: série de retornos para a qual será calculado o drawdown. :type returns: pd.Series :param time_scale: fator de escala do cagr, que é o número de amostras em um ano. Caso fosse uma série temporal diária: 252; série temporal mensal: 12 :type time_scale: int

Retorno cagr do ativo.

Tipo de retorno float

`turingquant.metrics.capm` (*returns, market_returns, risk_free*)

Essa função, com o fornecimento dos retornos de um portfólio ou ativo, dos retornos do mercado e da retorno sem risco, calcula o retorno esperado pela abordagem CAPM. Essa abordagem considera o mercado (benchmark) e as relações com os ativos como parâmetro para estimar o retorno esperado.

Parâmetros

- **returns** (*pd.Series* ou *np.array*) – vetor de retornos
- **market_returns** (*pd.Series* ou *np.array*) – vetor de retornos do mercado ou benchmark
- **risk_free** (*float*) – retorno livre de risco

Retorno retorno esperado pela abordagem CAPM

Tipo de retorno float

`turingquant.metrics.cumulative_returns` (*returns, return_type*)

Essa função permite o cálculo do retorno cumulativo ao longo do tempo.

Parâmetros

- **returns** (*pd.Series*) – série de retornos da ação ao longo do tempo;
- **return_type** (*string*) – tipo de retorno (simples - “simp” ou logarítmico - “log”) presente na série.

Retorno série com os valores de retorno cumulativo ao longo do tempo

Tipo de retorno cumulative_returns (*pd.Series*)

`turingquant.metrics.drawdown` (*returns, plot=False*)

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

Parâmetros

- **returns** (*pd.Series*) – série de retornos para a qual será calculado o drawdown.
- **plot** (*bool*) – se *True*, plota o gráfico de underwater (drawdown consoante o tempo).

Retorno uma série com os valores percentuais do Drawdown.

Tipo de retorno *pd.Series*

`turingquant.metrics.ewma_volatility` (*returns, window, plot=False*)

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

- **returns** (*pd.DataFrame*) – série de retornos para o qual o EWMA será calculado.
- **window** (*int*) – janela móvel para cálculo da EWMA;
- **plot** (*bool*) – se *True*, plota o gráfico de linha da EWMA ao longo do tempo

Retorno um dataframe indexado à data com os valores de EWMA dos últimos window dias

Tipo de retorno ewma_volatility (*pd.DataFrame*)

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

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

Retorno série das estimativas de volatilidade

Tipo de retorno *garman_klass_vol* (*pd.Series*)

`turingquant.metrics.mar_ratio` (*returns, time_window, time_scale=252*)

Calcula e plota o drawdown percentual para uma série de retornos. :param returns: série de retornos para a qual será calculado o mar ratio. :type returns: *pd.Series* :param time_window: janela de tempo que o mar ratio será calculado em relação a escala de tempo. `time_window = 3` e `time_scale = 252` denota uma janela de 3 anos (Calmar Ratio). :type time_window: *float* :param time_scale: fator de escala do mar ratio, que é o número de amostras em um ano. Caso fosse uma série temporal diária: 252; série temporal mensal: 12 :type time_scale: *int*

Retorno valor do mar ratio do ativo

Tipo de retorno *float*

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

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

Retorno série das estimativas de volatilidade

Tipo de retorno *garman_klass_vol* (*pd.Series*)

`turingquant.metrics.plot_allocation` (*dictionary*)

Essa função permite a visualização da distribuição de pesos em um portfolio através da plotagem de um gráfico de pizza.

Parâmetros dictionary (*dict*) – dicionário com o nome da ação e sua respectiva porcentagem na carteira, no formato ação:porcentagem.

`turingquant.metrics.returns` (*close_prices, return_type='log', cumulative=False*)

Essa função permite o cálculo rápido do retorno de uma ação ao longo do tempo.

Parâmetros

- **close_prices** (*pd.DataFrame*) – série de preços de fechamento que será utilizada de base para o cálculo do retorno;
- **return_type** (*string*) – tipo de retorno (simples - “simp” ou logarítmico - “log”) a ser calculado;

- **cumulative** (*bool*) – se True, calculará o retorno cumulativo

Retorno série com os valores do retorno ao longo do tempo

Tipo de retorno returns (pd.Series)

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

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

Parâmetros

- **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.

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

Tipo de retorno pd.Series

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

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

Parâmetros

- **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.

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

Tipo de retorno pd.Series

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

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 (simples - “simp” ou logarítmico - “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

Retorno um dataframe indexado à data com os valores de desvio padrão móvel dos últimos *window* dias

Tipo de retorno rolling_std (pd.DataFrame)

turingquant.metrics.sharpe_ratio (returns, risk_free=0, time_scale=252)

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.

Parâmetros

- **returns** (*pd.series*) – série com o retorno do ativo.
- **risk_free** (*float*) – risk free utilizado para cálculo do sharpe ratio.
- **time_scale** (*int*) – fator de escala do sharpe ratio, que é o número de amostras em um ano. Caso fosse uma série temporal diária: 252; série temporal mensal: 12

Retorno índice de sharpe do ativo.

Tipo de retorno 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.

Parâmetros

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

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

Tipo de retorno `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*.

Parâmetros

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

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

Tipo de retorno `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`.

Parâmetros

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

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

Tipo de retorno `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`.

Parâmetros

- **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

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

Tipo de retorno `pd.DataFrame`

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

Essa função calcula o holding period return anual de junho

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

Obtém o Balance Sheet ou o Balanço Patrimonial para a companhia do ticker desejado por meio do Yahoo! Finance.

Parâmetros

- **ticker** (*str*) – recebe o ticker do papel que será obtido
- **br** (*str*) – se *True*, adiciona «.SA» ao final do ticker, necessário para papéis brasileiros

Retorno dataframe com os dados do relatório nos últimos anos.

Tipo de retorno `pd.DataFrame`

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

Obtém o Cash Flow ou o Fluxo de Caixa para a companhia do ticker desejado por meio do Yahoo! Finance.

Parâmetros

- **ticker** (*str*) – recebe o ticker do papel que será obtido

- **br** (*str*) – se *True*, adiciona «.SA» ao final do ticker, necessário para papéis brasileiros

Retorno dataframe com os dados do relatório nos últimos anos.

Tipo de retorno `pd.DataFrame`

`turingquant.support.get_financials(url)`

Função de suporte, base para as funções `get_income_statement()`, `get_balance_sheet()` e `get_cashflow()`.

`turingquant.support.get_fundamentus(tickers)`

Essa função obtém os dados patrimoniais de empresas por meio do site `fundamentus.com.br`, voltado para companhias com papeis na B3.

Parâmetros **tickers** (*str* / *list*) – string com tickers separados por espaço ou lista de tickers

Retorno dataframe contendo os dados patrimoniais (linhas) para os tickers dados (colunas)

Tipo de retorno `pd.DataFrame`

`turingquant.support.get_ibov()`

Essa função obtém informações sobre a composição atual do Índice Bovespa por meio do site da B3.

Retorno dataframe contendo ticker, nome, tipo, quantidade e participação das companhias constituintes do índice

Tipo de retorno `pd.DataFrame`

`turingquant.support.get_income_statement(ticker, br=True)`

Obtém o Income Statement ou a Demonstração do Resultado do Exercício (DRE) para a companhia do ticker desejado por meio do Yahoo! Finance.

Parâmetros

- **ticker** (*str*) – recebe o ticker do papel que será obtido
- **br** (*str*) – se *True*, adiciona «.SA» ao final do ticker, necessário para papéis brasileiros

Retorno dataframe com os dados do relatório nos últimos anos.

Tipo de retorno `pd.DataFrame`

`turingquant.support.get_sp500_tickers()`

Essa função obtém os tickers de todas as atuais constituintes do S&P500.

Retorno lista com todos os tickers atuais do índice.

Tipo de retorno `list`

`turingquant.support.get_tickers(setores='Todos')`

Essa função obtém os tickers listados no site `fundamentus.com.br` consoante seus setores. Observação: o “setor” no site `fundamentus.com.br` corresponde ao “subsetor” na B3, e o “subsetor” nesse site corresponde ao “segmento” na B3.

Parâmetros **setores** (*str* / *list*) – “Todos” para considerar todos os setores ou lista com os setores desejados

Retorno lista com todos os tickers listados para os setores pedidos

Tipo de retorno `list`

`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`.

Parâmetros

- **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)

Retorno DataFrame contendo a cotação intraday dos últimos 5 dias.

Tipo de retorno pd.DataFrame

Módulo optimizers

Módulo para otimização de portfólios.

class `turingquant.optimizers.Markowitz` (*df_close*, *num_portfolios=10000*, *risk_free=0*)

Otimizador baseado na Teoria Moderna do Portfólio, de Harry Markowitz. A partir dos dados de fechamento, gera portfólios com pesos aleatórios e calcula os melhores pesos utilizando o risco e retorno da carteira.

Parâmetros: *df_close* (pd.DataFrame): DataFrame com os preços de fechamento dos ativos *num_portfolios* (int): números de portfólios gerados *risk_free* (float): taxa de risco livre utilizada para cálculo do sharpe ratio.

Atributos:

wallets (dict): dicionário contendo os valores “weights”, “returns”, “vol” e “sharpe_ratio” de todos os portfólios gerados

best_portfolio (*method='sharpe_ratio'*)

Retorna os pesos do melhor portfólio de acordo com o método escolhido.

Parâmetros method (*string*) - Método utilizado para indicar o melhor portfólio “sharpe_ratio” - Portfólio com melhor Sharpe ratio “volatility” - Portfólio com menor volatilidade “return” - Portfólio com maior retorno

Retorno Numpy array contendo os pesos do melhor portfólio.

Tipo de retorno weights (np.array)

plot_efficient_frontier (*method='sharpe_ratio'*)

Plota gráfico com a fronteira eficiente dos portfólios gerados.

Parâmetros method (*string*) - Método utilizado para indicar o melhor portfólio “sharpe_ratio” - Portfólio com melhor Sharpe ratio “volatility” - Portfólio com menor volatilidade “return” - Portfólio com maior retorno

t

- `turingquant`, 1
- `turingquant.benchmark`, 11
- `turingquant.metrics`, 5
- `turingquant.optimizers`, 17
- `turingquant.support`, 13

A

alpha() (no módulo *turingquant.metrics*), 5

B

benchmark() (no módulo *turingquant.benchmark*), 11

benchmark_ibov() (no módulo *turingquant.benchmark*), 11

benchmark_sp500() (no módulo *turingquant.benchmark*), 11

best_portfolio() (método *turingquant.optimizers.Markowitz*), 17

beta() (no módulo *turingquant.metrics*), 5

C

cagr() (no módulo *turingquant.metrics*), 5

capm() (no módulo *turingquant.metrics*), 5

cumulative_returns() (no módulo *turingquant.metrics*), 6

D

daily() (no módulo *turingquant.support*), 13

drawdown() (no módulo *turingquant.metrics*), 6

E

ewma_volatility() (no módulo *turingquant.metrics*), 6

G

garman_klass_volatility() (no módulo *turingquant.metrics*), 6

get_annual_hpr() (no módulo *turingquant.support*), 13

get_balance_sheet() (no módulo *turingquant.support*), 13

get_cash_flow() (no módulo *turingquant.support*), 13

get_financials() (no módulo *turingquant.support*), 14

get_fundamentus() (no módulo *turingquant.support*), 14

get_ibov() (no módulo *turingquant.support*), 14

get_income_statement() (no módulo *turingquant.support*), 14

get_sp500_tickers() (no módulo *turingquant.support*), 14

get_tickers() (no módulo *turingquant.support*), 14

I

intraday() (no módulo *turingquant.support*), 14

M

mar_ratio() (no módulo *turingquant.metrics*), 7

Markowitz (classe em *turingquant.optimizers*), 17

P

parkinson_volatility() (no módulo *turingquant.metrics*), 7

plot_allocation() (no módulo *turingquant.metrics*), 7

plot_efficient_frontier() (método *turingquant.optimizers.Markowitz*), 17

R

returns() (no módulo *turingquant.metrics*), 7

rolling_beta() (no módulo *turingquant.metrics*), 8

rolling_sharpe() (no módulo *turingquant.metrics*), 8

rolling_std() (no módulo *turingquant.metrics*), 8

S

sharpe_ratio() (no módulo *turingquant.metrics*), 8

T

turingquant (módulo), 1

turingquant.benchmark (módulo), 11

turingquant.metrics (módulo), 5

turingquant.optimizers (módulo), 17

turingquant.support (módulo), 13