

---

**turingquant**

*Versão 0.1.6*

**09 mar., 2021**



---

## Contents:

---

<b>1 Módulos</b>	<b>3</b>
<b>2 Módulo metrics</b>	<b>5</b>
<b>3 Módulo benchmark</b>	<b>9</b>
<b>4 Módulo support</b>	<b>11</b>
<b>Índice de Módulos do Python</b>	<b>13</b>
<b>Índice</b>	<b>15</b>



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.

---





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.

**Parâmetros**

- **start\_price** (*float*) – preço inicial.
- **end\_price** (*float*) – preço final.
- **dps** (*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.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` (`return_series`, `plot=True`)

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

#### Parâmetros

- **return\_series** (`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` (`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

**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=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

**Retorno** série das estimativas de volatilidade

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

`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

**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 portfólio 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=True*)

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=True*)

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=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 (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)`

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.

**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_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`.

### 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



### t

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



**A**

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

**B**

`benchmark()` (no módulo `turingquant.benchmark`), 9

`benchmark_ibov()` (no módulo `turingquant.benchmark`), 9

`benchmark_sp500()` (no módulo `turingquant.benchmark`), 9

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

**C**

`cumulative_returns()` (no módulo `turingquant.metrics`), 5

**D**

`daily()` (no módulo `turingquant.support`), 11

`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`), 11

`get_fundamentus()` (no módulo `turingquant.support`), 11

`get_ibov()` (no módulo `turingquant.support`), 11

`get_sp500_tickers()` (no módulo `turingquant.support`), 11

`get_tickers()` (no módulo `turingquant.support`), 11

**I**

`intraday()` (no módulo `turingquant.support`), 11

**P**

`parkinson_volatility()` (no módulo `turingquant.metrics`), 6

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

**R**

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

`rolling_beta()` (no módulo `turingquant.metrics`), 7

`rolling_sharpe()` (no módulo `turingquant.metrics`), 7

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

**S**

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

**T**

`turingquant` (módulo), 1

`turingquant.benchmark` (módulo), 9

`turingquant.metrics` (módulo), 5

`turingquant.support` (módulo), 11