Cálculo de retorno dos Swaps

O retorno diário do swap é dado por:

$$r_t^{Swap} = r_t^{ETF} - r_t^{short} \tag{1}$$

Dessa forma, usando como base o preço do *total return index (gross dividends)* (p_t) dos ETFs usados, calculamos o log-retorno diário dos ETFs como:

$$r_t^{ETF} = log(p_t) - log(p_{t-1}) \tag{2}$$

```
import pandas as pd
import numpy as np
def calculate_log_return(df : pd.DataFrame, price_col : str):
        return np.log(df[price_col]) - np.log(df[price_col].shift(1))
```

A base da **LIBOR** (índice usado na perna short do swap) foi dado em taxa anual, incialmente precisamos transformar a taxa anual em taxa diária e em seguida transformar em log-retorno:

$$r_t^{\text{daily-log-return}} = log((1 + r_t^{annual})^{1/365}) \tag{3}$$

```
import pandas as pd
import numpy as np
def transform_annual_rate_in_daily_log_raturn(df : pd.DataFrame, annual_rate_col: str):
        df2 = df.reset_index()
        return np.log((1 + df2[annual_rate_col])**(1/365))
```

Com isso, os log-retornos diários dos swaps das ETFs (r_t^{Swap}) foram calcualdos.

Cálculo da Volatilidade

O cálculo da vol de cada asset e a vol. da estratégia é primordial para o controle geral das estratégias e para a execução individual de cada estratégia, tanto na vol-targeting como na implementação usada pela trend-following.

O cálculo de cada componente está descrito abaixo. Para todas, o número de pontos n usados para o cálculo da vol. foi definido como 90 (embora pudesse ser definido um número diferente para cada estratégia e book).

1. Swaps

1.1. Swaps - Cálculo da volatilidade

Dado a série histórica dos log-retornos dos swaps usados em uma determinada estratégia, obtemos os últimos n pontos antes da data t e calculamos o desvio padrão de cada swap:

$$V_t^{Swaps} = Std[R_{t-n-1 \,\leq\, d \,<\, t}^{Swaps}] \tag{1}$$

```
def get_assets_volatility(self, target_date, last_n_points):
    df = self.log_returns[self.log_returns[ias.DATE] < target_date].tail(last_n_points)
    std = df[self.asset_names].std()
    return std
```

1.2. Swaps - Cálculo da covariância

Dado a série histórica dos log-retornos dos swaps usados em uma determinada estratégia, obtemos os últimos n pontos antes da data t e a matriz de covariância:

$$COV_t^{Swaps} = Cov[R_{t-n-1 \,\leq\, d \,<\, t}^{Swaps}] \tag{2}$$

```
def get_covariance_matrix(self, target_date, last_n_points):
    df = self.log_returns[self.log_returns[ias.DATE] < target_date].tail(last_n_points)
    covariance_matrix = df[self.asset_names].cov()
    return covariance_matrix
```

2. Estratégias

2.1. Estratégias - Cálculo da volatilidade

Em determinada data t, um swap (ou asset, para generalizar) possui uma posição $w_{a,t}$ em determinada estratégia, sendo que cada posição pode variar entre $-\lambda_{max} \leq w_{a,t} \leq \lambda_{max}$, onde λ_{max} é a máxima alavangacem (*leverage*) permitida para essa estratégia.

Uma posição $w_{a,t} > 0$ significa que a estratégia está long no swap e uma posição $w_{a,t} < 0$ significa que a estratégia está short no swap.

Sendo uma estratégia com k swaps, o vetor de posições para cada swap usado nessa estratégia é definido por:

$$W_t = \begin{bmatrix} w_{1,t} \\ w_{2,t} \\ \vdots \\ w_{k,t} \end{bmatrix}^T \tag{3}$$

Assim, dado o vetor de posições da estratégia e a matriz de covariância dos log-retornos dos swaps COV_t^{Swaps}, a volatilidade da estratégia é dada por:

$$V_t^{Estratégia} = \sqrt{W_t \times COV_t^{Swaps} \times W_t^T} \tag{4}$$

```
def get_portfolio_volatility(self, target_date, last_n_points):
    covariance = self.get_covariance_matrix(target_date, last_n_points)

    # calculando vol da estratégia
    current_asset_weights_series = pd.Series(self.current_asset_weights)
    portfolio_var =  current_asset_weights_series @ covariance @ current_asset_weights_
    portfolio_vol = np.sqrt(portfolio_var)

    return portfolio_vol
```

3. Vol. do Trading Book

O Trading Book é formado por um conjunto de estratégias e o peso em cada estratégia pode mudar ao longo do tempo. Nesse sentido, uma estratégia posssui um peso $\rho_{s,t}$ no boook, sendo que:

$$\begin{cases} \sum_{s=1}^{m} \rho_{s,t} = 1 & \forall t \\ 0 \leq \rho_{s,t} \leq 1 & \forall t, \end{cases} \tag{5}$$

onde m é o número de estratégias. Com isso, o vetor de pesos para cada estratégia no book é definido por:

$$P_t = \begin{bmatrix} \rho_{1,t} \\ \rho_{2,t} \\ \vdots \\ \rho_{m,t} \end{bmatrix}^T \tag{6}$$

Para calcular a matriz de correlação das estratégias, usamos como *proxy* do log-retorno o P&L diário da estratégia, desconsiderando a média de P&L diário dos últimos n dias.

Nesse sentido, para uma data t, a covariânica das estratégias $COV_t^{Estratégias}$ seria dada por:

$$COV_t^{Estratégias} = Cov[(PnL_{t-n-1 \,\leq\, d \,<\, t}^{Estratégias}) - \mathbb{E}(PnL_{t-n-1 \,\leq\, d \,<\, t}^{Estratégias})] \tag{7}$$

Assim, sendo o vetor de pesos das estratégias P_t e a matriz de covariância dos pnls diários das estratégias $COV_t^{Estratégias}$, a volatilidade do book é dada por:

$$V_t^{Book} = \sqrt{P_t \times COV_t^{Estratégias} \times P_t^T} \tag{8}$$

```
def calc_portfolio_volatility_with_weights(self, target_date, last_n_points, current_as
    covariance = self.get_covariance_matrix(target_date, last_n_points)

    # calculando vol do book
    portfolio_var =  current_asset_weights @ covariance @ current_asset_weights.T
    portfolio_vol = np.sqrt(portfolio_var)

    return portfolio_vol
```