Hackhaton Itau Asset 2021

Construir um robô-consultor

Objetivo

Seu objetivo é entregar um pacote python **documentado** com códigos comentados, dependências explicitadas em um arquivo `requirements.txt` que possa ser instalado e usado por um colega de empresa em outra área e que possua as funcionalidades úteis para o problema relatado.

Preparação

Para consultar os dados do terminal Bloomberg através do Python, é necessário instalar a API Bloomberg para python e é útil também uma API intermediária:

1. Interface `xbbg`: https://pypi.org/project/xbbg/
2. Bloomberg API: https://bcms.bloomberg.com/pip/simple/blpapi/

Introdução do problema

Como se deve recomendar uma carteira de investimento? Há várias abordagens possíveis discutidas em economia financeira, que incluem métodos econométricos sofisticados até economia comportamental. O trabalho pioneiro foi do economista Harry Markowitz e abordagem dele será usada como introdução básica conceitual. Markowitz ponderou que um investidor escolhe sua alocação de investimentos de modo a maximizar o que ele espera de retorno, que é o percentual do ganho ou perda do investimento relativo ao investido, para uma tolerância de risco. Risco em finanças é sinônimo de variável aleatória: a incerteza sobre os resultados futuros. Como medida de risco, Markowitz propos usar o desvio-padrão dos retornos da carteira do investidor.

Seja S o conjunto de N instrumentos financeiros (ações, fundos, etc.) disponíveis para o investidor, r_i o retorno do i-ésimo instrumento. Considere que r_i é uma variável aleatória. Seja $E[r_i]$ o retorno que o investidor espera obter do i-ésimo investimento após investir neste por um tempo fixo T. Se o investidor investir uma fração w_i em i do total de sua carteira de investimentos, o retorno total desta carteira após o período T será

$$E[r_p] = \sum_i w_i E[r_i]$$

Se for possível estimar a matriz de covariância Ω da distribuição de probabilidades das variáveis aleatórias r_i, a variância σ^2 da carteira será dada por

$$\sigma^2 = \vec{w} \cdot \Omega \vec{w}$$

Desta forma, um investidor que busca uma carteira com desvio-padrão σ_p deve escolher uma alocação $\vec{w}$ no ponto em que $E[r_p]$ atinge um máximo com o vínculo $\sigma(w) = \sigma_p$. Esse problema é usualmente considerado na forma inversa: para um dado r_p, o investidor deve alocar $\vec{w}$ tal que $\sigma(\vec{w})$ é um mínimo.

Parte 1

(a)

Seja Ω uma matriz dada sobre corpo dos números reais $\mathbb{R}$ simétrica definida-positiva de dimensão $N \times N$, e r_p um número real dado. **Desenvolva** um algoritmo que encontra a solução numérica do problema de otimização de mínima variância com vínculos:

$$\begin{aligned} & \min_w \vec{w} \cdot \Omega \vec{w} \\ \text{restrito a} & \\ & \sum_i w_i = 1 \\ & \sum_i w_i r_i = r_p \\ & w_i \in [0,1] \; \forall i \end{aligned}$$

Este é um caso particular do problema de programação quadrática.

(b)

Escreva um módulo python que implemente o seu algoritmo. Função ou classe são aceitáveis. Seu módulo pode depender do pacote `numpy`, mas não pode utilizar algoritmos prontos, pacotes de otimização não-linear ou pacotes de programação quadrática.

(c)

Escreva um módulo python que constrói a *fronteira eficiente* de média-variância, que é a curva $\gamma \in \mathbb{R}^2$ dos valores r_p como função de σ_p que são soluções do problema (a) para um Ω e r_i dados e para todos os valores permissíveis de r_p. O seu módulo deve ser capaz de produzir um conjunto de pontos que existem, dado o vetor de r_i, do menor r_p possível ao maior r_p possível, com o número de pontos da curva n_p dado como parâmetro e os pontos da curva espaçados igualmente em r_p (ou seja, a diferença do r_p entre dois pontos consecutivos é fixa). Inclua uma visualização gráfica da curva $r_p(\sigma_p)$ e também de w_i para todos os i ao longo de σ_p. Seu módulo pode depender de outros módulos de visualização gráfica, mas deve usar seus resultados do item (b).

Parte 2

Como parte do seu pacote python com as funcionalidades da Parte 1, inclua um exemplo pronto de uso com a seguinte aplicação. Seu exemplo pode ser um script python ou um caderno jupyter.

Considere que um investidor possui um perfil de risco e objetivo consistentes em investir no seguinte conjunto de cotas de fundos de investimento disponíveis para negociação na B3, chamados ETFs:

Ticker	Nome	Ticker do índice
BOVV11 BZ Equity	It Now Ibovespa	IBOV Index
SPXI11 BZ Equity	It Now SP500 Total Return	SPXT Index
IMAB11 BZ Equity	It Now ID IMA-B	BZRFIMAB Index
IRFM11 BZ Equity	It Now IRFM P2	BZRFIRFM Index

(a)

Obtenha uma estimação da matriz de covariância dos retornos diários dos ETFs. Você pode usar os pacotes `numpy` e `pandas`. Para obter os preços ao longo do tempo, você pode usar a função `bdh` do pacote `xbbg` que serve de interface a API Bloomberg. Por exemplo, para obter os preços históricos do BOVV11:

```
from xbbg import blp


blp.bdh('BOVV11 BZ Equity', 'PX_LAST', '2010-12-30', '2021-04-30')
```

A função `bdh` aceita no primeiro argumento o ticker Bloomberg do instrumento financeiro. Os retornos diários $r_i(t)$ no dia t são as variações percentuais dos preços $P_i(t)$ no fechamento de mercado do dia t calculados como:

$$r_i(t) = \frac{P_i(t) - P_i(t-1)}{P_i(t-1)} = P_i(t)/P_i(t-1) - 1$$

Para alguns ETFs que tenham um histórico muito curto (menor que 3 anos), você pode completar o histórico faltante de *retornos* com os retornos dos índices correspondentes que os ETFs replicam.

(b)

Usando como a média histórica dos retornos diários para cada ETF como o valor esperado $E[r_i]$ dos retornos futuros e a matriz Ω estimada em (a), obtenha um conjunto de 100 portfólios igualmente espaçados em retorno da fronteira eficiente de média-variância. Construa o gráfico da fronteira $r_p(\sigma_p)$ e também uma visualização dos pesos $w_i(\sigma_p)$.

Parte 3

Utilizando o conjunto de investimentos da Parte 2, considere um investidor que informa que após $T \geq 1$ dias aceita perder uma fração de até ES_T do seu investimento com probabilidade de 5%. Assumindo que os retornos de um ponto (r_p, σ_p) da fronteira eficiente calculada na Parte 2 são dados por uma distribuição normal de média r_p e desvio-padrão σ_p, adicione ao seu pacote uma função ou método de classe que recebe ES_T e T como argumentos e retorna o vetor $\vec{w}$ dos pesos em cada instrumento que o investidor pode investir. Você pode adicionar métodos gráficos de visualização da carteira de investimentos resultante, usando pacotes python.

Note que os retornos calculados na Parte 2, como r_p e r_i, referem-se ao período de investimento de apenas um dia, e que o investidor deseja especificar seu horizonte T além da fração ES_T.