

MAE001: Modelagem Matemática em Finanças I

Ramon Duarte de Melo
ramonduarte@poli.ufrj.br
Alex Teixeira
ramonduarte@poli.ufrj.br

Universidade Federal do Rio de Janeiro (UFRJ) — 12 de junho de 2019

Introdução

O objetivo do Projeto III é implementar, avaliar e comparar o algoritmo recursivo proposto pelo livro no capítulo 1.3 e o Método de Monte-Carlo, aplicados à determinação do valor de contratos americanos e europeus de opção de compra e venda, realizando comparações de cunho matemático-estatístico e produzindo gráficos com tais observações.

Para tal, foi utilizada a linguagem *Python 3.6.7*, com os módulos *numpy* (métodos numéricos) e *matplotlib.pyplot* (visualização de dados). O programa requer a instalação destes módulos, mas possui uma ferramenta de instalação automatizada das dependências (*pipenv*).

O código utilizado neste trabalho, bem como o deste relatório e as imagens geradas, foi aberto e disponibilizado publicamente no repositório <https://github.com/ramonduarte/mmftab2>.

Atividade a

Nesta atividade, foi implementado o algoritmo sugerido no livro em seu capítulo 1.3. O procedimento é composto de três etapas:

1. obtenção dos valores finais, com especial atenção para evitar a explosão combinatória típica do modelo binomial.
2. cálculo recursivo dos valores intermediários utilizando os valores finais.
3. dedução do valor inicial V_0 do contrato, que também representa seu custo pela teoria de precificação da arbitragem.

Os contratos escolhidos possuem os mesmos parâmetros, tanto para a opção de compra, quanto para a opção de venda:

- valor inicial do ativo: 4
- taxa de valorização: 100
- taxa de desvalorização: 50
- taxa de renda fixa: 25
- preço de *strike*: 5

Estes parâmetros foram utilizados junto às probabilidades neutras a risco $\tilde{p} = \tilde{q} = 0.5$. Foram calculados os valores para 10 simulações, tais que $N = 2^k; k = 1, 2, \dots, 10$.

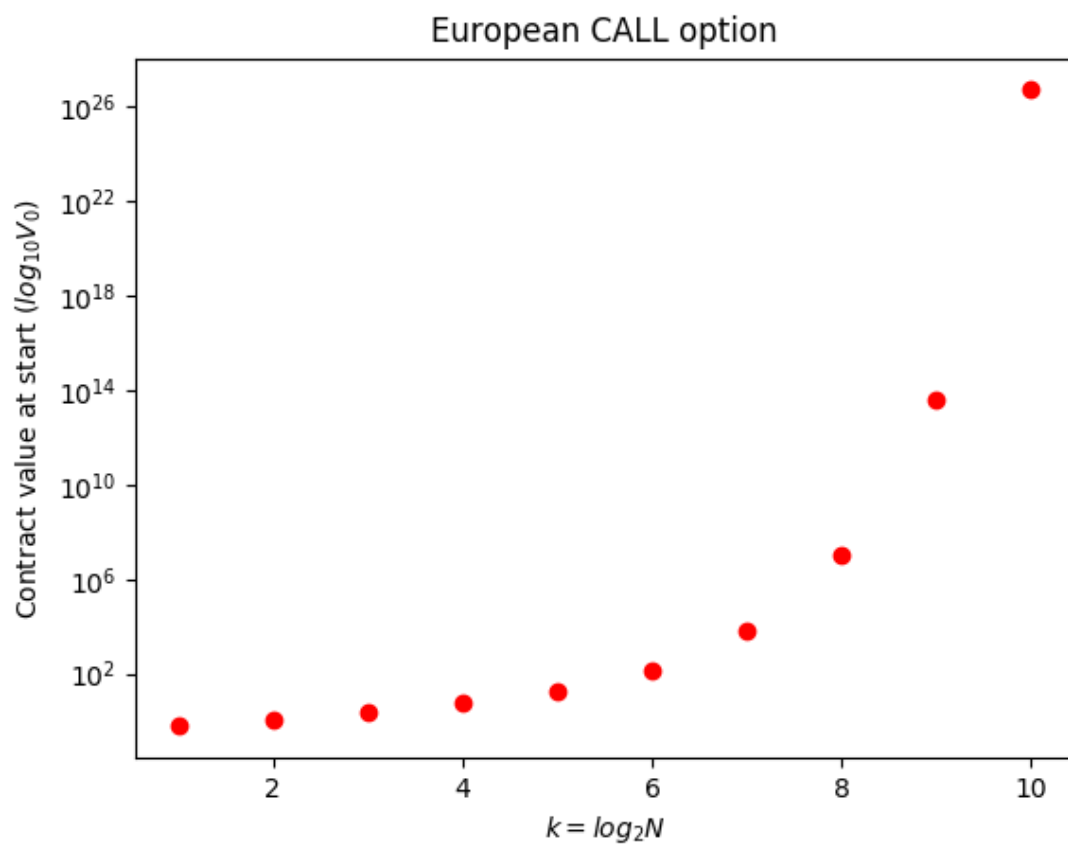


Figura 1: Passeios aleatórios com $S_0 = 100$, $u = 1.024$, $d = 1/u = 0.9765625$ e $p = q = 0.5$.

Atividade b

Os boxplots foram gerados usando o módulo `matplotlib.pyplot.boxplot`. Para facilitar a comparação dos boxplots, eles foram agrupados em subplots no mesmo gráfico. *Outliers* foram marcados com quadrados vermelhos.

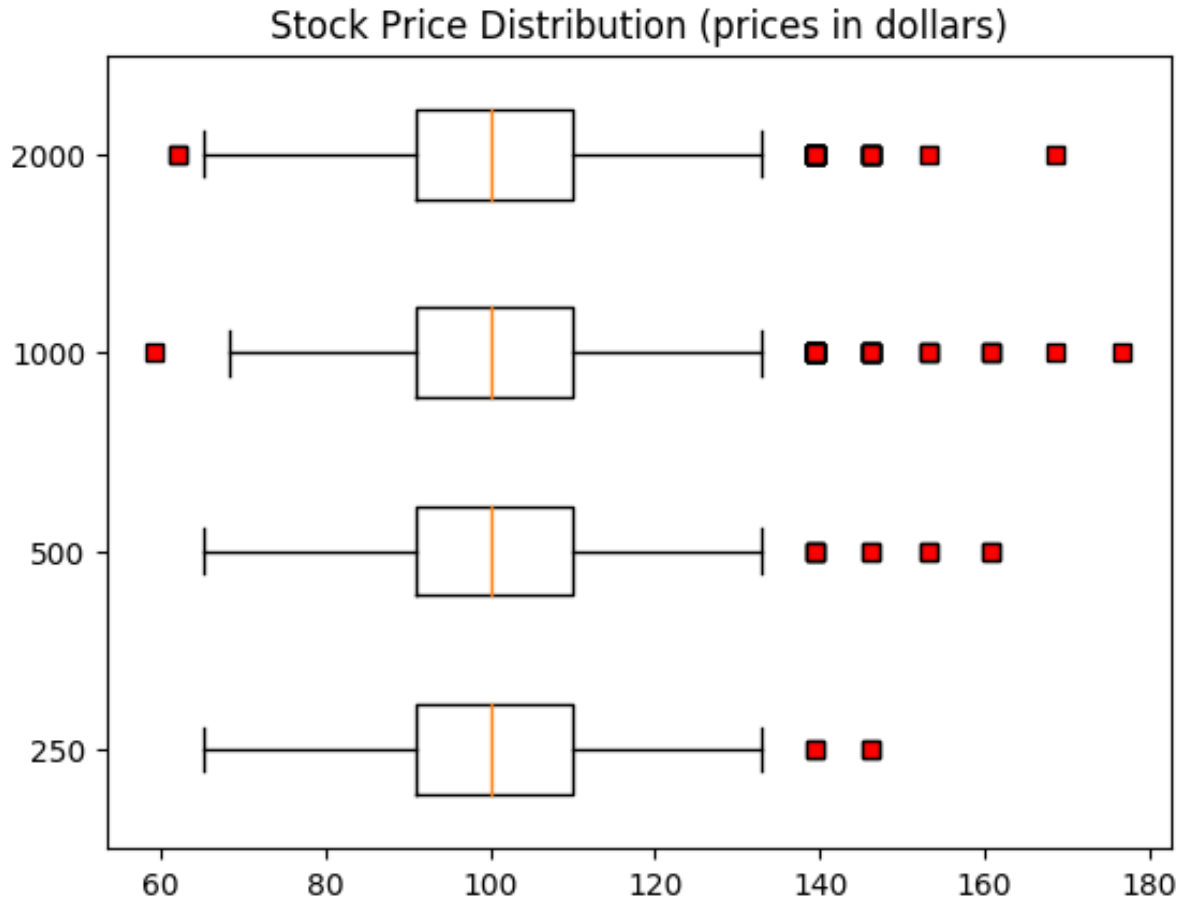


Figura 2: Boxplots com 250, 500, 1000 e 2000 iterações, respectivamente, de baixo para cima.

Atividade c

Para esta questão, os boxplots foram refeitos utilizando os parâmetros $\Delta t = 0.25$ e $u = 1/d = \sqrt{1.024}$. Esta configuração gerou uma variância consideravelmente menor, bem como uma quantidade significativamente menor de *outliers*. Embora todos os ativos pudessem atingir as mesmas taxas de crescimento e decrescimento que no cenário anterior - apesar das taxas serem menores, o maior número de iterações compensa proporcionalmente a redução -, foram raros os ativos que obtiveram mais de 30% de variação, um cenário bastante comum na segunda questão.

Este fenômeno foi observado porque, como $E[\bar{S}] = E[S_0]$ e $E[S(n)] = E[S(n+1)]$, um número maior de repetições tende a aproximar os resultados da média. A redução da variância em distribuições probabilísticas por conta da ampliação do espaço amostral é bastante conhecida e documentada.

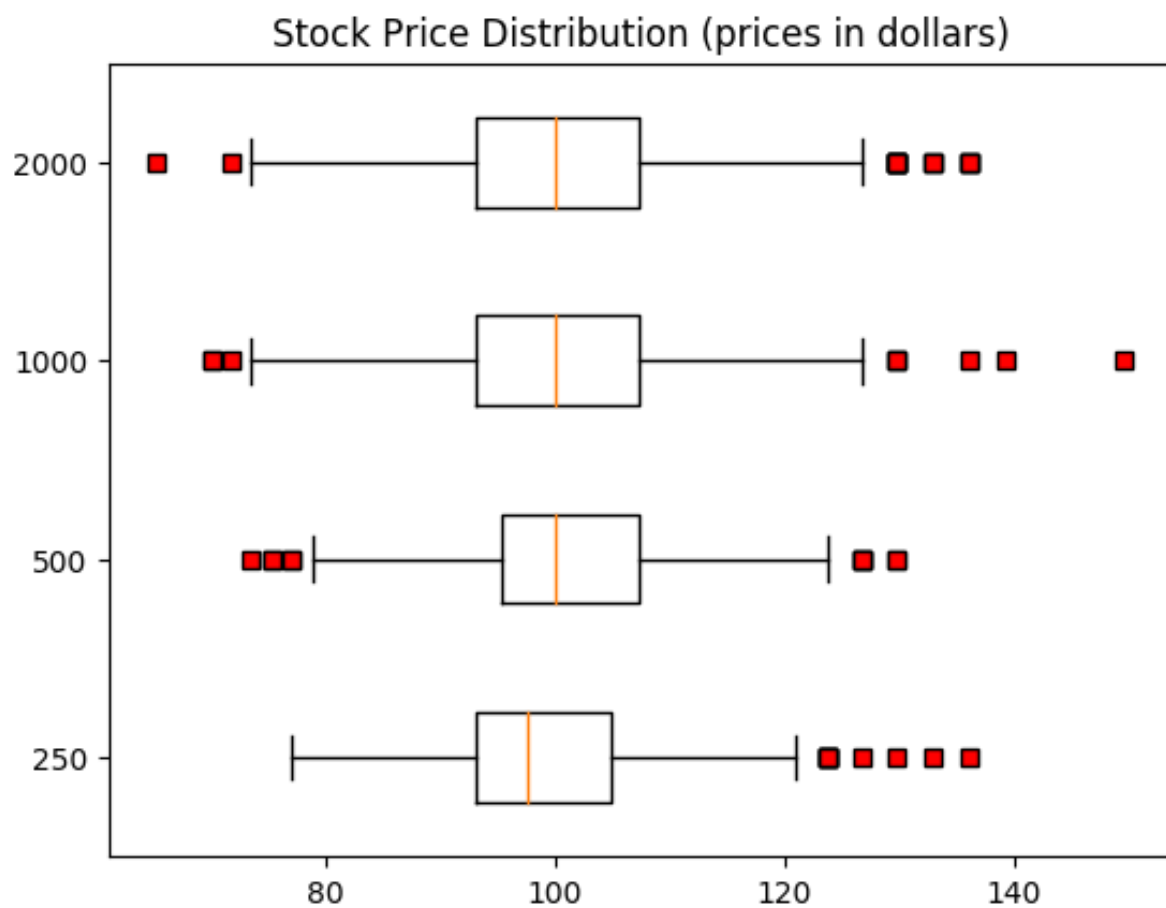


Figura 3: Boxplots com 250, 500, 1000 e 2000 iterações, respectivamente, de baixo para cima.