

Sumário

Gil Sales M. Neto & João Victor de Fonseca

Projeto I
Modelagem Matemática em Finanças I
O Modelo Binomial

Universidade Federal do Rio de Janeiro

Instituto de Matemática

Bacharelado em Matemática Aplicada

Brasil

Maio, 2019

Sumário

1	OS ALGORITMOS	3
1.1	Simulação do modelo binomial	3
1.2	Plotando o gráfico da questão 1	4
1.2.1	Plot usual	4
1.2.2	Plot com escala Log em Y	5
1.3	Boxplots da questão 2	5
1.4	Boxplots da questão 3	5
2	QUESTÃO 1	7
2.0.1	Plot com escala comum	7
2.0.2	Plot com escala log	8
3	QUESTÃO 2	9
4	QUESTÃO 3	10
5	QUESTÃO 4	11
5.0.1	Valor esperado para o primeiro caso	11
5.0.2	Valor esperado para o segundo caso	11
5.0.3	Média dos valores para o primeiro caso	11
5.0.4	Média dos valores para o segundo caso	11
5.0.5	Conclusão	12
6	QUESTÃO 5	13

1 Os Algoritmos

Esta seção tem como objetivo apresentar todos os códigos utilizados nas simulações.

1.1 Simulação do modelo binomial

Utilizamos a linguagem [Python3](#) para a implementação do algoritmo que simula os caminhos da ação

```
In [1]: ## Função para simular os valores da ação dados os parametros
def binomial(S0, T, dt, u, d, p):
    Si = S0
    S = []
    t = np.arange(0,T,dt)
    for ti in t:
        rnd = np.random.rand()
        if rnd < p:
            Si *= u
        else:
            Si *= d
        S.append(Si)
    return S
```

Esse algoritmo tem dependência do pacote [Numpy](#) do Python, e os gráficos foram plotados com o pacote [Matplotlib.pyplot](#)

1.2 Plotando o gráfico da questão 1

Foram plotados dois gráficos, os quais seguem os códigos

1.2.1 Plot usual

```
In [2]: ## Definição dos parametros
        T = 20
        dt = 0.5
        S0 = 8
        u = 1.1
        d = 0.8
        p = 0.65

        ## Construção da lista de valores da ação pelo tempo
        x = [binomial(S0, T, dt, u, d, p) for i in range(0,20)]

        ## Inserir o valor inicial em cada lista
        for i in range(0,20):
            x[i].insert(0,S0)

        ## Discretização do tempo
        ts = np.linspace(0,20,41)

        ## Plot do gráfico
        plt.figure(figsize=(20,10))
        for i in range(0,20):
            plt.plot(ts,x[i],label="{0:.5f}".format(x[i][-1]))
        plt.title('Simulação com 20 caminhos de valor da ação')
        plt.legend(title='Valor final', loc='upper center',
            ↪ fancybox=True, shadow=True, ncol=7,
            ↪ bbox_to_anchor=(0.5,1.05))
        plt.xlabel('Tempo')
        plt.ylabel('Valor da Ação.')
        plt.show()
```

1.2.2 Plot com escala Log em Y

```
In [3]: plt.figure(figsize=(20,10))
        for i in range(0,20):
            plt.semilogy(ts,x[i],label="{0:.5f}".format(x[i][-1]))
        plt.legend(title='Valor final', loc='upper center',
        ↪ fancybox=True, shadow=True, ncol=7,
        ↪ bbox_to_anchor=(0.5,1.05))
        plt.title('Caminhos de valor da ação')
        plt.show()
```

1.3 Boxplots da questão 2

```
In [30]: vs = []
        med_1 = []
        vf = []
        for j in [250,500,1000,2000]:
            vs.append([binomial(S0, T, dt, u, d, p)[-1] for i in range(0,j)])
        plt.boxplot(vs,showfliers=False)
        plt.title('Boxplot com simulações')
        plt.xlabel('Número de caminhos executados na simulação')
        plt.xticks([1,2,3,4],[250,500,1000,2000])
        plt.show()
```

1.4 Boxplots da questão 3

```
In [83]: u_n = np.sqrt(u)
        d_n = np.sqrt(d)
        dt_n = dt/2

        vs = []
        for j in [250,500,1000,2000]:
            vs.append([binomial(S0, T, dt_n, u_n, d_n, p)[-1] for i in range(0,j)])

        plt.boxplot(vs,showfliers=False)
```

```
plt.title('Boxplot com simulações')
plt.ylabel('Valores das ações ao tempo final')
plt.xlabel('Número de caminhos executados na simulação')
plt.xticks([1,2,3,4],[250,500,1000,2000])
plt.show()
```

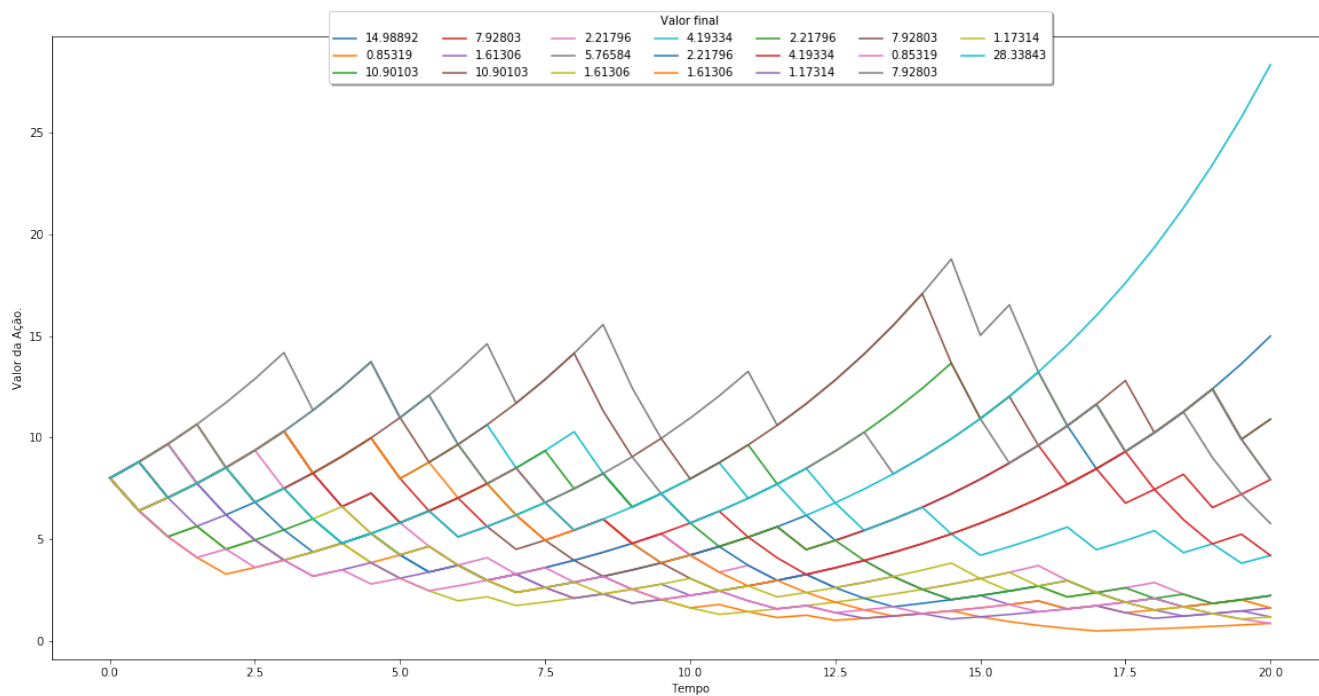
2 Questão 1

Foi pedido para fixar os parâmetros $T = 20$ e $\Delta t = 0.5$ e escolher os outros, nossos parâmetros então ficaram:

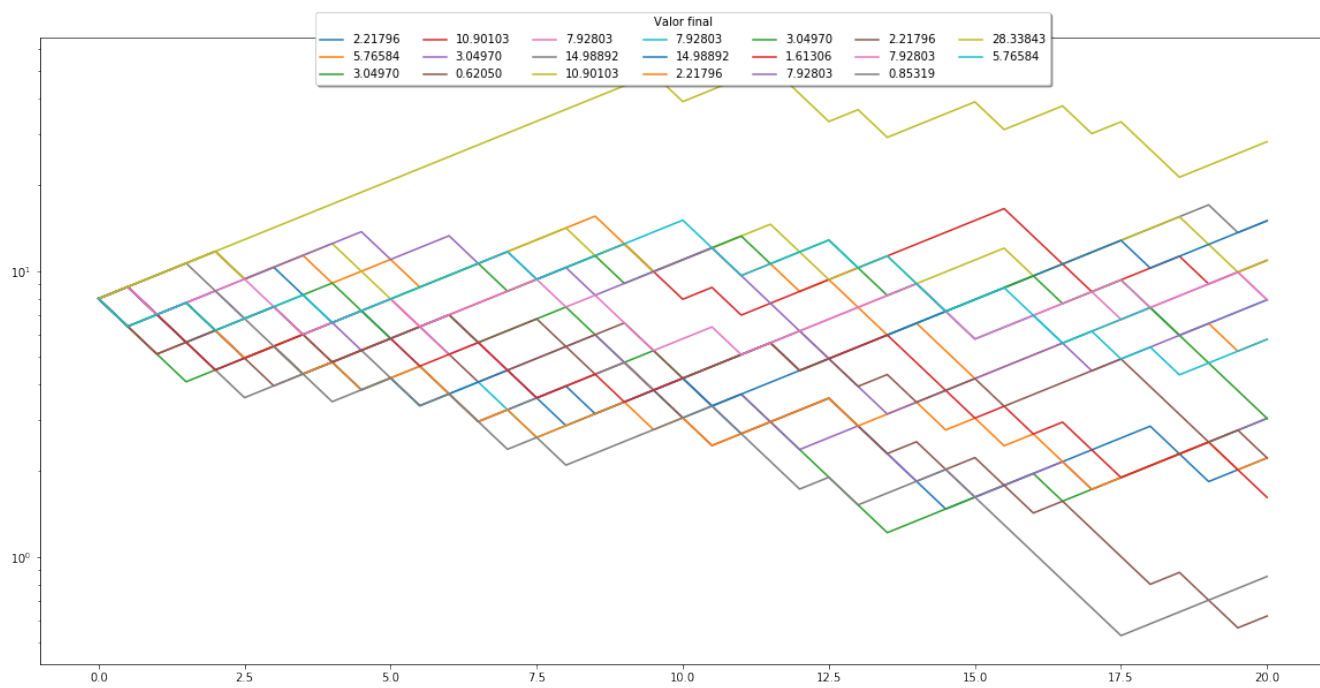
- $T = 20$
- $\Delta t = 0.5$
- $u = 1.1$
- $d = 0.8$
- $p = 0.65$

Como estamos tratando de um modelo binomial que lida com exponenciais, faz-se necessário também visualizar os valores com escala log.

2.0.1 Plot com escala comum

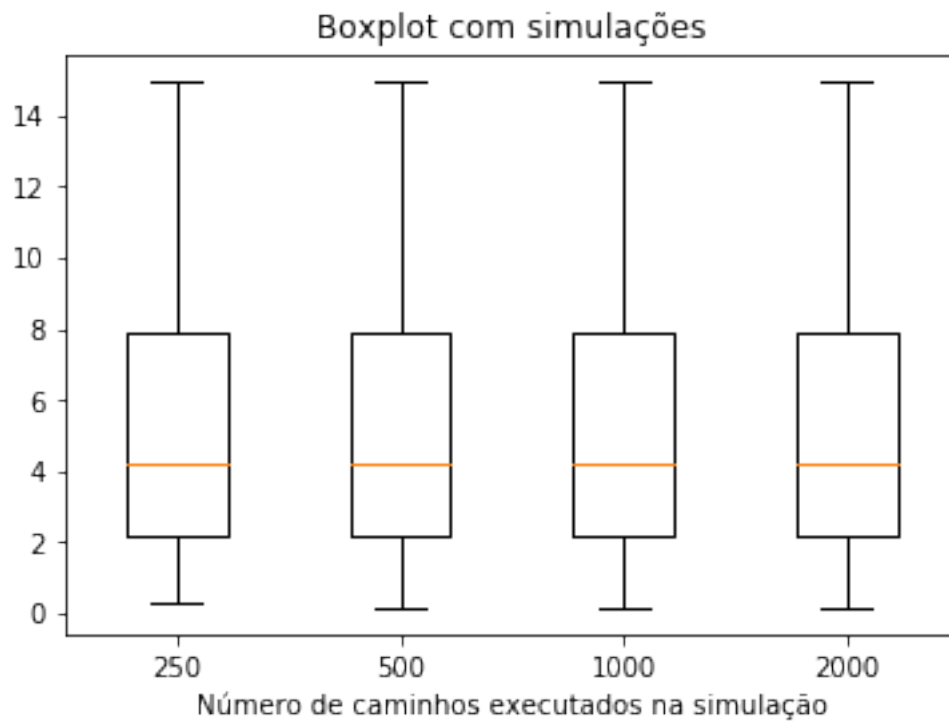


2.0.2 Plot com escala log



3 Questão 2

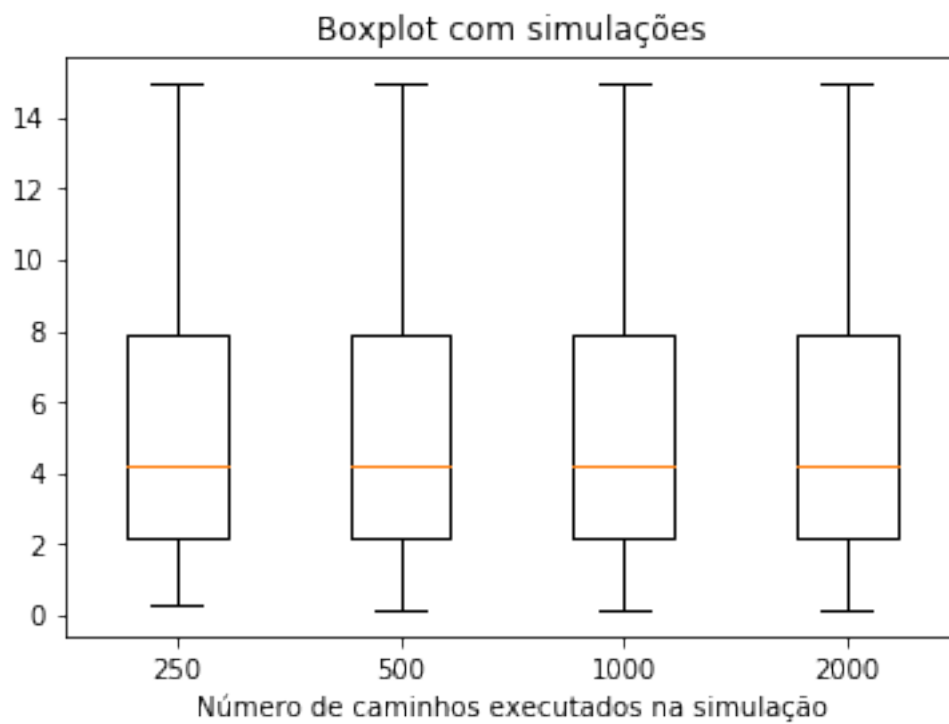
Foi pedido um boxplot para 250,500,1000,2000 caminhos de valor de ação



4 Questão 3

Foi pedido um mesmo boxplot, mas com novos parametros, que ficaram:

- $T = 20$
- $\Delta t = 0.25$
- $u = \sqrt{(1.1)}$
- $d = \sqrt{(0.8)}$
- $p = 0.65$



5 Questão 4

A média esperada da binomial pode ser dada pela seguinte fórmula:

$$E[X] = (p \cdot u + q \cdot d)^n \cdot S_0$$

onde $q = (1 - p)$, $n = T$, e os outros são os parametros do algoritmo

A média do conjunto de dados é obtida pela média aritmetica, utilizando o método `mean` do numpy.

5.0.1 Valor esperado para o primeiro caso

$$\begin{aligned} E[X] &= (0.65 \cdot 1.1 + 0.35 \cdot 0.8)^{20} \cdot 8 \\ &\approx (0.995)^{20} \cdot 8 \\ &\approx (0.904) \cdot 8 \\ &\approx 7.237 \end{aligned}$$

5.0.2 Valor esperado para o segundo caso

$$\begin{aligned} E[X] &= (0.65 \cdot 1.049 + 0.35 \cdot 0.894)^{20} \cdot 8 \\ &\approx (0.994)^{20} \cdot 8 \\ &\approx (0.900) \cdot 8 \\ &\approx 7.2 \end{aligned}$$

5.0.3 Média dos valores para o primeiro caso

250 ações: 7.229332844074394 500 ações: 6.576994633671657 1000 ações: 6.659109025949532
2000 ações: 6.508780520361858

5.0.4 Média dos valores para o segundo caso

250 ações: 5.320103652152599 500 ações: 5.152399440097309 1000 ações: 5.410913075158868
2000 ações: 5.238002583867098

5.0.5 Conclusão

falta escrever

6 Questão 5

Plotar gráfico do erro e concluir sobre