Gil Sales M. Neto & João Victor de Fonseca

Projeto 3
MAE001 - Modelagem Mat. em Finanças I
Cálculo de Volatilidade Implicita
& Plot do Smile (Volatilidade x Strike)

Universidade Federal do Rio de Janeiro

Instituto de Matemática

Bacharelado em Matemática Aplicada

Prof.: Marco Cabral

Brasil

Junho, 2019

Sumário

1 Os Algoritmos

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

1.1 Black & Scholes

Utilizamos a linguagem Python3 para a implementação do algoritmo de Black & Scholes e cálculo da volatilidade implicita. Esse algoritmo tem depedência dos pacotes: Scipy para estatística, Math para cálculos matemáticos, Pandas para lidar com os dados e MatPlotLib.pyplot para os gráficos.

```
In [78]: from scipy import stats
         import scipy
         import matplotlib.pyplot as plt
         import pandas as pd
         import math
```

```
In [64]: """ Precificação utilizando Black and Scholes.
             cp: +1 -> call; -1 put
             s: valor da ação
             k: strike
             t: tempo em dias até expirar a opção
             v: volatilidade
             rf: taxa de juros neutra risco
         """
         def black_scholes (cp, s, k, t, rf, v, div = 0):
                 d1 =
                 ↪  (math.log(s/k)+(rf+0.5*math.pow(v,2))*t)/(v*math.sqrt(t))
                 d2 = d1 - v*math.sqrt(t)

                 optprice =
                 ↪  (cp*s*math.exp(-div*t)*stats.norm.cdf(cp*d1)) -
                 ↪  (cp*k*math.exp(-rf*t)*stats.norm.cdf(cp*d2))
```

```
        return optprice


""" Calculo da volatilidade implicita
    cp: +1 -> call; -1 put
    s: valor da ação
    k: strike
    t: tempo em dias até expirar a opção
    rf: taxa de juros neutra risco
    price: cotação da opção pelo mercado
    Função H(vol): Seja B(vol) o valor calculado por B&S dada
↪   volatilidade, e P a cotação da opção, H(vol) = B(vol) - P
    É a função a ser usada na bisseção
"""
def volat_impl(cp, s, k, t, rf, price):
    def h(vol):
        return black_scholes(cp, s, k, t, rf, vol) - price
    return scipy.optimize.bisect(h,1e-6,5,xtol=1e-16)
```

1.1.1 Importando os dados do CSV

```
In [65]: data = pd.read_csv('BRFOODS.csv')
```

```
In [163]: '''
          Setting CONSTANTS
          '''

          sigla_acao = 'BRFS3'
          empresa = 'BRFoods S.A.'
          preco_acao = 27.00
          dias = np.arange(6,29)


          puts = data['Tipo'] == 'PUT'
          calls = data['Tipo'] == 'CALL'
          dia1706 = data['TF'] == '17-06-2019'
          dia1507 = data['TF'] == '15-07-2019'
```

1.1.2 Plots dos gráficos

1.1.2.1 Plot Put 6 dias

In [172]:
```
## PUT Com 6 dias a vencer

    ## Buscando as informações no DataFrame
    df_k = data[puts & dia1706].iloc[0:,2:3]
    df_s = data[puts & dia1706].iloc[0:,3:4]
    ks_put_6 = df_k.values.flatten()
    Ss_put_6 = df_s.values.flatten()

    ## Setando o array com as volatilidades a serem plotadas
    vs_put_6 = []
    for (k,s) in zip(ks_put_6,Ss_put_6):

        ↪ vs_put_6.append(volat_impl(-1,preco_acao,k,dias[0]/365,0.065,s))

    ## Plot do gráfico
    plt.figure(figsize=(10,6))
    plt.plot(ks_put_6,vs_put_6, marker='o', linestyle='--',
    ↪ color='g', markerfacecolor='r')
    plt.xlabel('Strikes - em R$')
    plt.ylabel('Volatilidade')
    plt.title('Gráfico Smile - Volatilidade x Strike - PUT Com 6
    ↪ dias a vencer')
    plt.grid()
    plt.show()
```

1.1.2.2 Plot Call 6 dias

In [98]:
```
## CALL Com 6 dias a vencer

    ## Buscando as informações no DataFrame
    df_k = data[calls & dia1706].iloc[0:,2:3]
    df_s = data[calls & dia1706].iloc[0:,3:4]
    ks_call_6 = df_k.values.flatten()
    Ss_call_6 = df_s.values.flatten()
```

```
## Setando o array com as volatilidades a serem plotadas
vs_call_6 = []
for (k,s) in zip(ks_call_6,Ss_call_6):

    ↪ vs_call_6.append(volat_impl(1,preco_acao,k,dias[0]/365,0.065,s))

## Plot do gráfico
plt.figure(figsize=(10,6))
plt.plot(ks_call_6,vs_call_6, marker='o', linestyle='--',
↪ color='g', markerfacecolor='r')
plt.xlabel('Strikes - em R$')
plt.ylabel('Volatilidade')
plt.title('Gráfico Smile - Volatilidade x Strike - CALL Com 6
↪ dias a vencer')
plt.grid()
plt.show()
```

1.1.2.3 Plot Put 28 dias

```
In [186]: ## PUT Com 28 dias a vencer

## Buscando as informações no DataFrame
df_k = data[puts & dia1507].iloc[0:,2:3]
df_s = data[puts & dia1507].iloc[0:,3:4]
ks_put_28 = df_k.values.flatten()
Ss_put_28 = df_s.values.flatten()

## Setando o array com as volatilidades a serem plotadas
vs_put_28 = []
for (k,s) in zip(ks_put_28,Ss_put_28):

    ↪ vs_put_28.append(volat_impl(-1,preco_acao,k,dias[1]/365,0.065,s))

## Plot do gráfico
plt.figure(figsize=(10,6))
```

```
plt.plot(ks_put_28,vs_put_28, marker='o', linestyle='--',
↪  color='g', markerfacecolor='r')
plt.xlabel('Strikes - em R$')
plt.ylabel('Volatilidade')
plt.title('Gráfico Smile - Volatilidade x Strike - PUT com 28
↪  dias a vencer')
plt.grid()
plt.show()
```

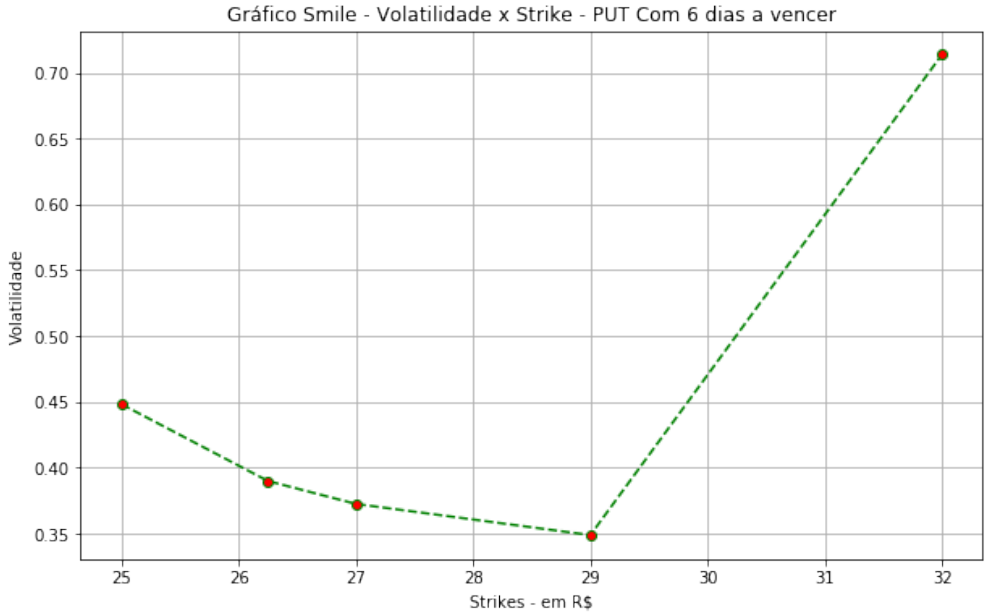
1.1.2.4 Plot Call 28 dias

In [173]:
```
## CALL Com 28 dias a vencer

## Buscando as informações no DataFrame
df_k = data[calls & dia1507].iloc[0:,2:3]
df_s = data[calls & dia1507].iloc[0:,3:4]
ks_call_28 = df_k.values.flatten()
Ss_call_28 = df_s.values.flatten()

## Setando o array com as volatilidades a serem plotadas
vs_call_28 = []
for (k,s) in zip(ks_call_28,Ss_call_28):

 ↪  vs_call_28.append(volat_impl(1,preco_acao,k,dias[1]/365,0.065,s))

## Plot do gráfico
plt.figure(figsize=(10,6))
plt.plot(ks_call_28,vs_call_28, marker='o', linestyle='--',
↪  color='g', markerfacecolor='r')
plt.xlabel('Strikes - em R$')
plt.ylabel('Volatilidade')
plt.title('Gráfico Smile - Volatilidade x Strike - CALL com 28
↪  dias a vencer')
plt.grid()
plt.show()
```

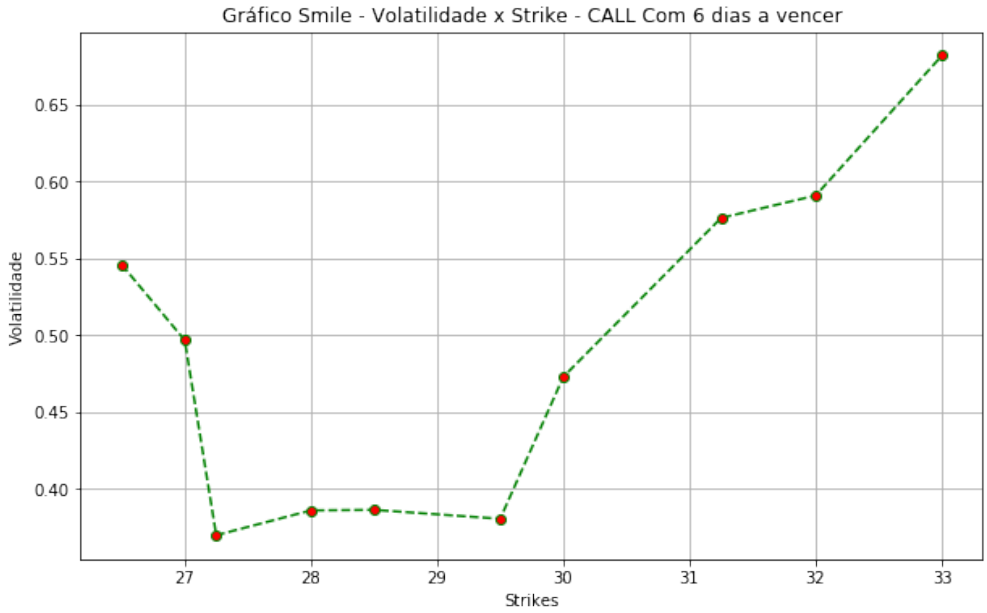
2 Atividade

Usamos como fonte de dados para preços de opções o site do InfoMoney: <https://www.infomoney.com.br/mercados/ferramentas/cotacoes-opcoes-acoes>, os dados obtidos foram salvos em um arquivo CSV
Utilizamos opções para ações da empresa BRFoods S.A. - Ações: BRFS3
As opções começam no dia 11/06/2019 e tem vencimento dia 17/06/2019 (6 dias) e 15/07/2019 (28 dias)

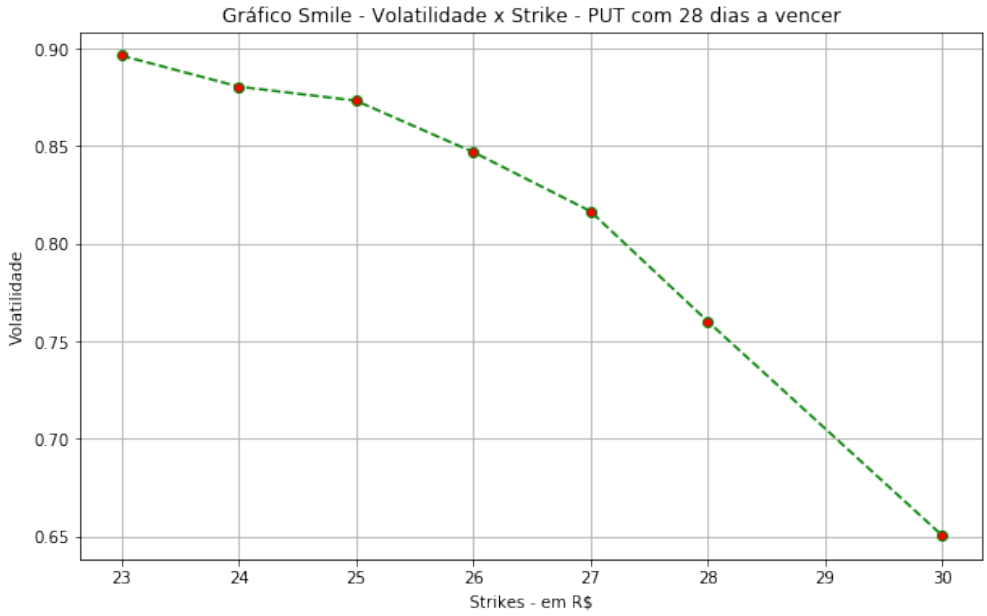
2.0.1 Gráfico de volatilidade implicita para as opções PUT com vencimento em 6 dias



2.0.2 Gráfico de volatilidade implicita para as opções CALL com vencimento em 6 dias



2.0.3 Gráfico de volatilidade implicita para as opções PUT com vencimento em 28 dias



2.0.4 Gráfico de volatilidade implicita para as opções CALL com vencimento em 28 dias

3 Conclusão

Foi possível observar um SMILE no **PUT** de 6 dias, no **CALL** de 6 dias e no **CALL** de 28 dias.
No **PUT** de 28 dias a curva ficou invertida