

Gil Sales M. Neto & João Victor de Fonseca

Projeto 4
MAE001 - Modelagem Mat. em Finanças I
CAPM & Estrutura a Termo

Universidade Federal do Rio de Janeiro

Instituto de Matemática

Bacharelado em Matemática Aplicada

Prof.: Marco Cabral

Brasil

Junho, 2019

Sumário

1	OS ALGORITMOS	3
1.1	Funções principais	3
1.1.1	Carregando dados e Calculando Estrutura a Termo	4
1.1.2	Plots dos gráficos	5
1.1.2.1	Função para plotagem	5
1.1.2.2	Plot Emitidos em 2017	5
1.1.2.3	Plot Emitidos em 2018	5
1.1.2.4	Plot Emitidos em 2019	5
1.1.3	Atividade 3	5
1.1.3.1	Importando Dados	5
1.1.3.2	Preparando Dados	6
1.1.3.3	Algoritmos	7
1.1.3.4	Plot Retorno	7
1.1.3.5	Plot Sharpe	7
2	ATIVIDADE 1	8
2.0.1	Estrutura a termo de juros	8
2.0.2	Estrutura a Termo para Título emitido em 2017	9
2.0.3	Estrutura a Termo para Título emitido em 2018	10
2.0.4	Estrutura a Termo para Título emitido em 2019	11
3	ATIVIDADE 3	12
3.0.1	Retorno esperado	12
3.0.2	Índice Sharpe	13

1 Os Algoritmos

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

1.1 Funções principais

Utilizamos a linguagem **Python3** para a implementação do algoritmo que determina a estrutura a termo de juros. Esse algoritmo tem dependência dos pacotes: **Scipy** para Spline, **Numpy** e **Matplotlib.pyplot** para os gráficos.

```
In [165]: import numpy as np
          import matplotlib.pyplot as plt
          from scipy.interpolate import spline
```

```
In [259]: # Carregando dados
          ltns_2017 = np.array([[897.78,2018],[812.14,2019],[754.04,2020],[651.41,2021]])
          ltns_2018 = np.array([[936.03,2019],[855.85,2020],[770.38,2021],[622.14,2023]])
          ltns_2019 = np.array([[937.58,2020],[866.29,2021],[804.61,2022],[719.12,2023]])
```

1.1.1 Carregando dados e Calculando Estrutura a Termo

```
In [260]: # Algoritmo para estrutura a termo
def term_structure(valores, ano):
    ys = np.array([(1000/i[0])**((1/(i[1]-ano)) - 1) for i in
    ↪ valores])
    ys *= 100
    return ys

# Estrutura a termo para cada ano
ts_2017 = [(i[1]-2017) for i in ltns_2017]
ys_2017 = term_structure(ltns_2017, 2017)

ts_2018 = [(i[1]-2018) for i in ltns_2018]
ys_2018 = term_structure(ltns_2018, 2018)

ts_2019 = [(i[1]-2019) for i in ltns_2019]
ys_2019 = term_structure(ltns_2019, 2019)
```

1.1.2 Plots dos gráficos

1.1.2.1 Função para plotagem

In [269]: *# Funções para plotar gráficos*

```
def plot_spline(ts, ys, ano, lim):  
    xs = np.linspace(1,lim,100)  
    spl = spline(ts, ys, xs)  
    plt.figure(figsize=(12,8))  
    plt.title('Yield Curve - Titulos emitidos em %d' % ano)  
    plt.xlabel('Tempo até maturidade (em anos)')  
    plt.ylabel('Juros (%)')  
    plt.grid(True)  
    plt.plot(xs,spl, 'r')  
    plt.plot(ts,ys, 'ro')  
    plt.show()
```

1.1.2.2 Plot Emitidos em 2017

In [273]: plot_spline(ts_2017, ys_2017, 2017, 4)

1.1.2.3 Plot Emitidos em 2018

In [272]: plot_spline(ts_2018, ys_2018, 2018, 5)

1.1.2.4 Plot Emitidos em 2019

In [270]: plot_spline(ts_2019, ys_2019, 2019,4)

1.1.3 Atividade 3

1.1.3.1 Importando Dados

```
In [90]: kroton = pd.read_csv('KROT3.csv')[::-1]  
        mglu = pd.read_csv('MGLU3.csv')[::-1]  
        itau = pd.read_csv('ITUB4.csv')[::-1]  
        raias = pd.read_csv('RADL3.csv')[::-1]
```

1.1.3.2 Preparando Dados

```
In [91]: itau_fecho = itau.iloc[0:,1:2].values.flatten()
        kroton_fecho = kroton.iloc[0:,1:2].values.flatten()
        raias_fecho = raias.iloc[0:,1:2].values.flatten()
        mglu_fecho = mglu.iloc[0:,1:2].values.flatten()

        beta_itau = 1.2663
        rm_itau = 0.2039

        beta_mglu = 1.2737
        rm_mglu = 0.2598

        beta_raia = 0.3499
        rm_raia = 0.1397

        beta_kroton = 1.1226
        rm_kroton = 0.0738

        re_itau = retorno_esp(selic, beta = beta_itau, rm = rm_itau)
        re_mglu = retorno_esp(selic, beta = beta_mglu, rm = rm_mglu)
        re_raia = retorno_esp(selic, beta = beta_raia, rm = rm_raia)
        re_kroton = retorno_esp(selic, beta = beta_kroton, rm = rm_kroton)

        vol_itau = 0.513346
        vol_mglu = 0.726408
        vol_raia = 0.164342
        vol_kroton = 0.226435

        shp_itau = sharpe(vol_itau, re_itau)
        shp_mglu = sharpe(vol_mglu, re_mglu)
        shp_raia = sharpe(vol_raia, re_raia)
        shp_kroton = sharpe(vol_kroton, re_kroton)
```

1.1.3.3 Algoritmos

```
In [92]: selic = 0.065

def retorno_esp(rf = selic, beta = 1, rm = 1):
    return rf + beta*(rm - rf)

def sharpe(vol, re, rf = selic):
    return (re - rf)/vol
```

1.1.3.4 Plot Retorno

```
In [93]: res = np.array([re_itaui, re_mglu, re_raia, re_kroton])
plt.figure(figsize=(12,8))
plt.bar(['Itau', 'Magazine Luiza', 'Drogas Raia', 'Kroton'], res*100)
plt.ylabel('Retorno (em %)')
plt.grid(alpha = 0.4)
plt.show()
```

1.1.3.5 Plot Sharpe

```
In [94]: sharpes = [shp_itaui, shp_mglu, shp_raia, shp_kroton]
plt.figure(figsize=(12,8))
plt.bar(['Itau', 'Magazine Luiza', 'Drogas Raia', 'Kroton'], sharpes)
plt.grid(alpha = 0.4)
plt.ylabel('Índice Sharpe')
plt.show()
```

2 Atividade 1

Usamos como fonte de dados para valor das LTNs o histórico fornecido pelo site do Tesouro Nacional: <<https://sisweb.tesouro.gov.br/apex/f?p=2031:2:0:::>>, os dados obtidos foram transcritos diretamente para listas (ltns_2017, ltns_2018, ltns_2019)

Utilizamos os arquivos LTN dos anos 2017, 2018 e 2019

Todos os valores usados levam em conta emissão em 01 de Janeiro do ano corrente, e maturidade em 01 de Janeiro do ano de maturidade, ano este que varia para cada ano de emissão.

2.0.1 Estrutura a termo de juros

Os juros foram calculados utilizando o modelo para estrutura a termo fornecido no livro texto:

$$y_m = \left(\frac{V_f}{B_{0,m}} \right)^{\frac{1}{m}} - 1$$

Onde:

y_m é o rendimento/juros anual para o título.

m é o tempo restante para maturidade, em anos.

$B_{0,m}$ é o valor em tempo 0 do título com maturidade em tempo m

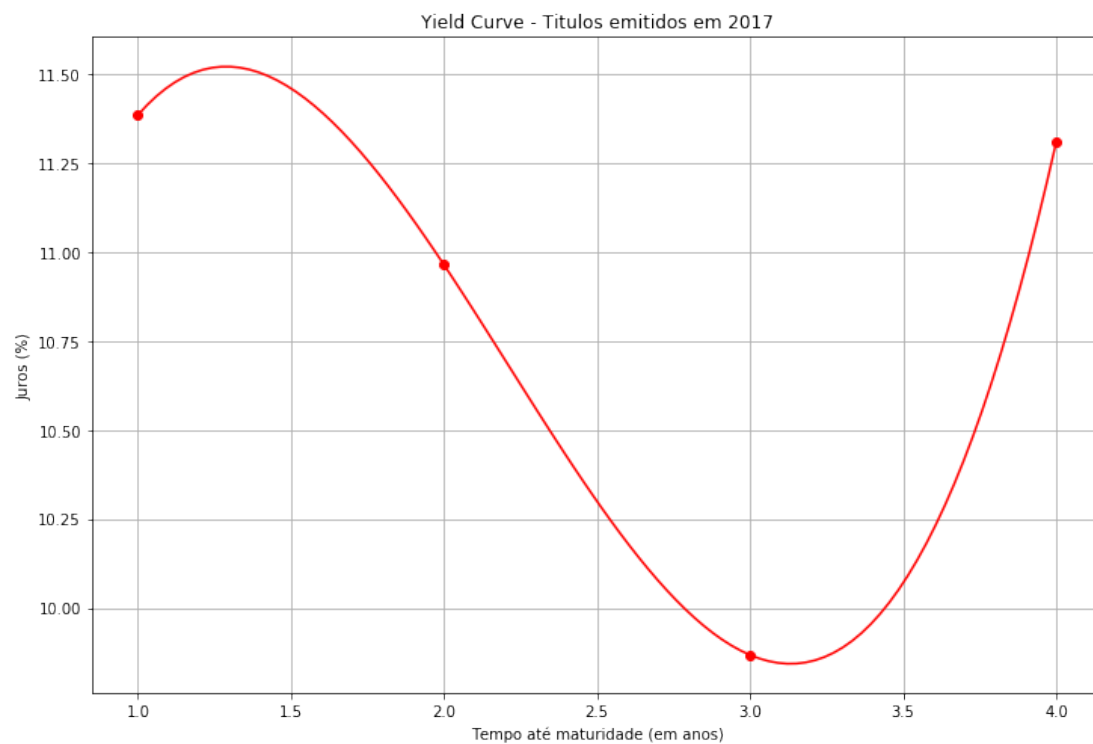
V_f é o valor de face, que no caso das LTNs: $V_f = 1000$

Com y_m calculados, plotamos a Curva de Rendimento (Yield Curve), ou estrutura a termo de juros.

O Gráfico é feito com tempo para maturidade no eixo horizontal e rendimento no eixo vertical, os pontos são interpolados com Spline do pacote SciPy.

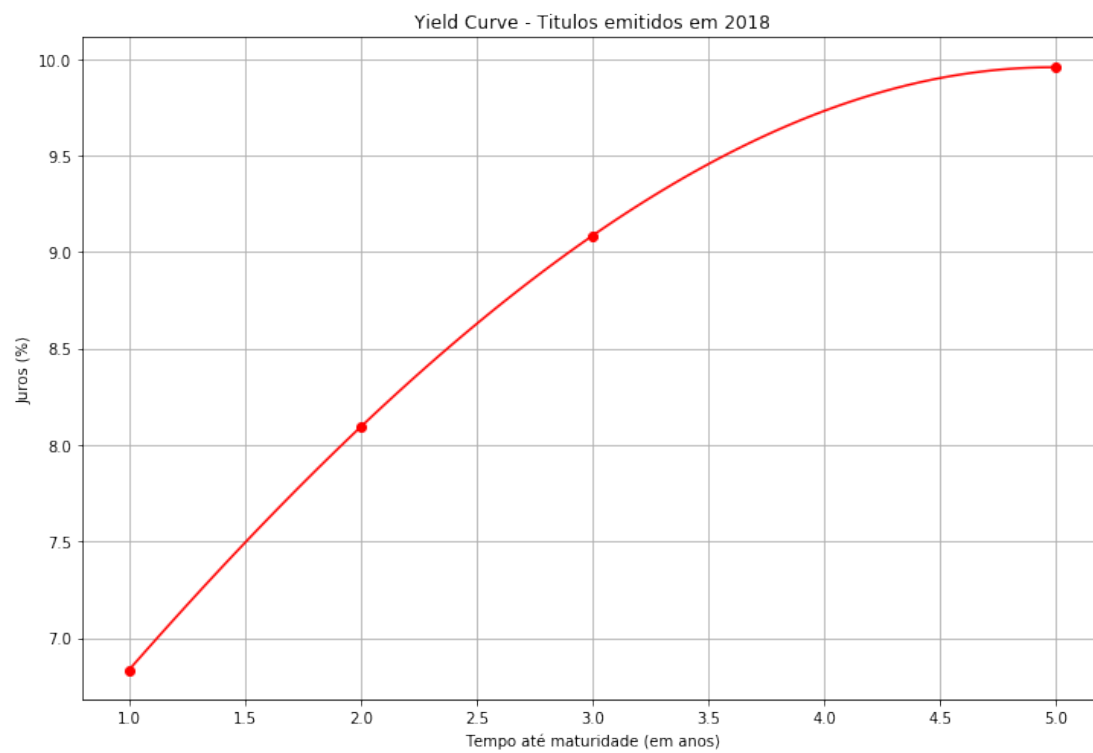
2.0.2 Estrutura a Termo para Título emitido em 2017

Ano de Maturidade	Valor em tempo 0
2018	R\$ 897.78
2019	R\$ 812.14
2020	R\$ 754.04
2021	R\$ 651.41



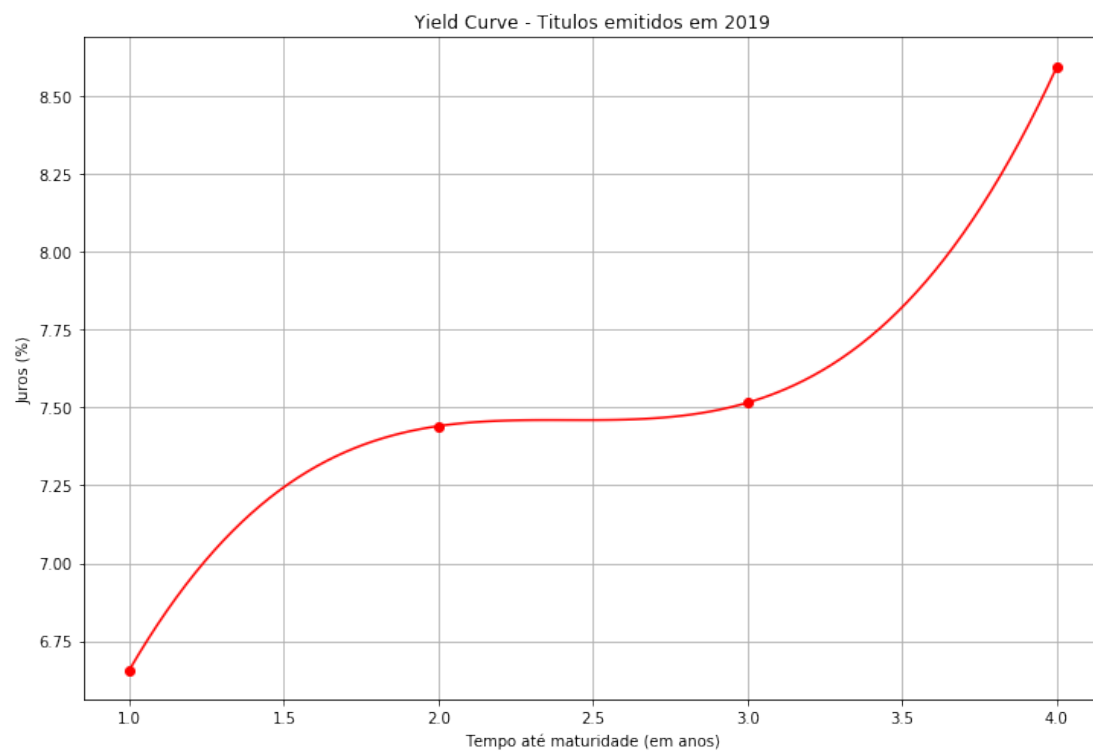
2.0.3 Estrutura a Termo para Título emitido em 2018

Ano de Maturidade	Valor em tempo 0
2019	R\$ 936.03
2020	R\$ 855.85
2021	R\$ 770.38
2023	R\$ 622.14



2.0.4 Estrutura a Termo para Título emitido em 2019

Ano de Maturidade	Valor em tempo 0
2018	R\$ 897.78
2019	R\$ 812.14
2020	R\$ 754.04
2021	R\$ 651.41



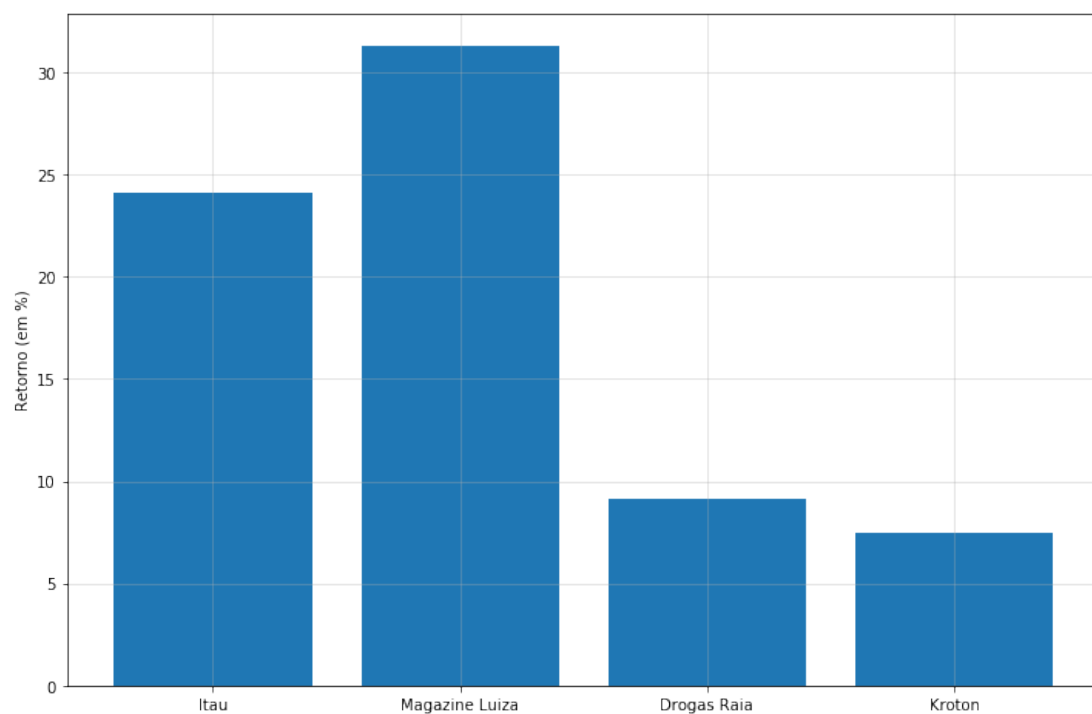
3 Atividade 3

Usamos como fonte de dados para valor das ações o histórico fornecido pelo site Investing.com: <<https://br.investing.com/equities/>>, os dados foram salvos em um arquivo .csv e importado com a biblioteca Pandas.

As empresas e ações utilizadas foram:

- Itaú Unibanco - ITUB4
- Kroton Educacional - KROT3
- Magazine Luiza - MGLU3
- Drogas Raia - RADL3

3.0.1 Retorno esperado



3.0.2 Índice Sharpe

