

Trabalho 3 de Econofísica

April 27, 2021

Edelson Luis Pinheiro Sezerotto Júnior, 288739

0.1 Introdução

Neste trabalho iremos criar uma carteira diversificada de ativos com o objetivo de maximizar o retorno esperado do nosso investimento para um dado risco assumido. Os ativos serão escolhidos usando o Capital Asset Pricing Model (CAPM) e os pesos dados para cada investimento do nosso portfólio serão determinados usando a Teoria de Markowitz. As séries temporais para todos os ativos serão baixadas do site Yahoo Finance (referência [1]).

Abaixo são importadas as bibliotecas necessárias e é definida uma lista de ativos candidatos a entrar em nossa carteira. Esses ativos (todos de empresas brasileiras) foram escolhidos arbitrariamente, e a decisão de colocá-los ou não na carteira será dada, conforme mencionado, pelo modelo CAPM.

```
[29]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import yfinance as yf

ativos = ['PBR', 'VALE', 'ITSA4.SA', 'MGLU3.SA', 'BBD', 'BDORY', 'CIG', 'CSNA3.
↪SA', 'BRF', 'CSAB4.SA', 'CIEL3.SA', 'ABEV3.SA'] #ativos candidatos
```

Abaixo é definida uma função que baixa a série temporal de um dado ativo e retorna uma lista com seus retornos lineares. Neste trabalho as séries temporais serão diárias e compreenderão o período de 24 de Abril de 2016 até 24 de Abril de 2021, totalizando um intervalo de 5 anos.

```
[30]: def obter_retornos(a):
    #gerando o data frame
    df = yf.download(a, start='2016-04-24', end='2021-04-24', progress=False)
    df = df.dropna() #remove valores nulos

    #pegando valores de interesse
    valores = df['Adj Close'].tolist()
    N = len(valores)

    #calculando os retornos
    rm = [(valores[i+1]-valores[i])/valores[i] for i in range(N-1)] #retorno
    ↪linear
```

```

    rm = [np.log10(valores[i+1])-np.log10(valores[i]) for i in range(N-1)]
    ↪ #logaritmico ###
    return rm

```

Agora definimos uma outra função, que nos fornece uma lista de retornos para um dado conjunto de ativos, fazendo também o gráfico de suas séries temporais:

```

[31]: def comparar_acoes(acoes):
    l = []
    for i in acoes:
        rm = obter_retornos(i)
        l.append(rm)

    ll = l.copy()
    for i in range(len(l)): l[i] = len(l[i])
    Min = min(l)
    for i in range(len(ll)):
        while len(ll[i]) > Min: del ll[i][0]

    for i in range(len(ll)): plt.plot(ll[i],label=acoes[i],alpha=0.4)
    plt.xlabel('Tempo')
    plt.ylabel('Retorno linear')
    plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
    plt.show()
    return ll

```

Tendo feito essas definições iniciais, vamos olhar para alguns conceitos teóricos.

0.2 Conceitos teóricos

0.2.1 Modelo CAPM

O modelo CAPM serve para nos auxiliar a determinar quais ativos devem ou não entrar na carteira. Para isso, partimos da equação de primeiro grau abaixo:

$$\overline{R}_i = r_0 + \beta_i(\overline{R}_M - r_0) \quad (1)$$

onde \overline{R}_i é o retorno esperado para um certo ativo i ao longo de algum intervalo de tempo, que aqui está sendo usado como 5 anos; r_0 é uma taxa livre de riscos, que no Brasil corresponde à taxa Selic; \overline{R}_M é o retorno médio de um mercado de referência, que no Brasil é o Bovespa; e β_i é um valor associado a cada ativo, sendo dado pela equação abaixo:

$$\beta_i = \frac{Cov(\overline{R}_i, \overline{R}_M)}{Var(\overline{R}_M)} \quad (2)$$

ou seja, β_i é igual à covariância entre os retornos do ativo i e do mercado, dividida pela variância dos retornos do mercado.

A equação (1) nos fornece o retorno mínimo esperado para um ativo dado o seu β . Se um ativo candidato a entrar em nossa carteira tiver um retorno menor do que o mínimo esperado, ele será deixado de fora. Quanto maior for o retorno médio comparado ao retorno mínimo, maior é a preferência do ativo para entrar em nosso portfólio. Neste trabalho serão selecionados até 9 dos melhores ativos, desde que seus retornos passem pelo critério estabelecido.

0.2.2 Teoria de Markowitz

Uma vez que selecionamos quais ativos irão compor nossa carteira, precisamos definir qual porcentagem de nosso investimento será destinada a cada ativo. Nosso investimento terá retornos ao longo do tempo que terão um certo valor médio α_0 e uma certa volatilidade σ , e dependendo da forma como montamos a carteira, poderemos ter valores diferentes de α_0 para um mesmo σ . A Teoria de Markowitz serve para descobrirmos qual o maior α_0 possível de ser obtido dado um certo σ . Vamos agora desenvolver as ferramentas necessárias para isso.

Inicialmente, vamos definir um vetor \vec{R} que armazena um conjunto de vetores \vec{R}_i representando as séries temporais dos retornos lineares dos N ativos da carteira:

$$\vec{R} = \begin{pmatrix} \vec{R}_1 \\ \vec{R}_2 \\ \vdots \\ \vec{R}_N \end{pmatrix} \quad (3)$$

Tomamos agora os valores médios de cada \vec{R}_i e os colocamos em um vetor $\vec{\alpha}$:

$$\vec{\alpha} = \begin{bmatrix} \overline{\vec{R}_1} \\ \overline{\vec{R}_2} \\ \vdots \\ \overline{\vec{R}_N} \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix} \quad (4)$$

Em seguida definiremos uma matriz S que conterà as covariâncias entre cada par de retornos (\vec{R}_i, \vec{R}_j) :

$$S = \overline{(\vec{R} - \vec{\alpha})(\vec{R} - \vec{\alpha})^T} \quad (5)$$

Vamos fazer uma análise cuidadosa de S para entendê-la melhor. Para isso, vamos considerar o caso $N = 3$ e ver qual a forma explícita de S nesse caso:

$$\begin{aligned} \vec{R} - \vec{\alpha} &= \begin{bmatrix} \vec{R}_1 - \alpha_1 \\ \vec{R}_2 - \alpha_2 \\ \vec{R}_3 - \alpha_3 \end{bmatrix} = \begin{bmatrix} [R_{11} - \alpha_1, R_{12} - \alpha_1, \dots, R_{1n} - \alpha_1]^T \\ [R_{21} - \alpha_2, R_{22} - \alpha_2, \dots, R_{2n} - \alpha_2]^T \\ [R_{31} - \alpha_3, R_{32} - \alpha_3, \dots, R_{3n} - \alpha_3]^T \end{bmatrix} \\ (\vec{R} - \vec{\alpha})(\vec{R} - \vec{\alpha})^T &= \begin{bmatrix} \vec{R}_1 - \alpha_1 \\ \vec{R}_2 - \alpha_2 \\ \vec{R}_3 - \alpha_3 \end{bmatrix} \begin{bmatrix} \vec{R}_1 - \alpha_1 & \vec{R}_2 - \alpha_2 & \vec{R}_3 - \alpha_3 \end{bmatrix} \end{aligned}$$

A multiplicação do vetor coluna pelo vetor linha acima nos dará uma matriz 3 x 3. O elemento na posição (0,0) é mostrado explicitamente abaixo:

$$(\vec{R}_1 - \alpha_1) \cdot (\vec{R}_1 - \alpha_1) = (R_{11} - \alpha_1)(R_{11} - \alpha_1) + \dots + (R_{1n} - \alpha_1)(R_{1n} - \alpha_1) = n \cdot Cov(\vec{R}_1, \vec{R}_1)$$

Mais genericamente temos:

$$(\vec{R}_i - \alpha_i) \cdot (\vec{R}_j - \alpha_j) = n \cdot Cov(\vec{R}_i, \vec{R}_j) \rightarrow \overline{(\vec{R}_i - \alpha_i) \cdot (\vec{R}_j - \alpha_j)} = Cov(\vec{R}_i, \vec{R}_j)$$

onde estamos supondo que todas as séries de retornos tem o mesmo tamanho n (isso será forçado a acontecer no nosso código). Assim, S terá o seguinte formato:

$$S = \begin{bmatrix} Cov(\vec{R}_1, \vec{R}_1) & Cov(\vec{R}_1, \vec{R}_2) & Cov(\vec{R}_1, \vec{R}_3) \\ Cov(\vec{R}_2, \vec{R}_1) & Cov(\vec{R}_2, \vec{R}_2) & Cov(\vec{R}_2, \vec{R}_3) \\ Cov(\vec{R}_3, \vec{R}_1) & Cov(\vec{R}_3, \vec{R}_2) & Cov(\vec{R}_3, \vec{R}_3) \end{bmatrix}$$

A utilidade de colocar S no formato acima é que podemos notar que, além de todos os seus valores serem reais, ela é simétrica em relação à sua diagonal principal, sendo portanto inversível - mais à frente necessitaremos da matriz inversa de S e precisávamos conferir se ela existia. Agora definimos um vetor \vec{W} que conterá os pesos para cada ativo (queremos determinar quais são os pesos que maximizam α_0 para um certo σ):

$$\vec{W} = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_N \end{bmatrix} \leftrightarrow \vec{W} \cdot \vec{e} = 1 \quad (6)$$

onde \vec{e} é um vetor de N elementos iguais a 1 (afinal a soma de todos os pesos da carteira deve valer $1 = 100\%$ do investimento). Feitas essas definições básicas, vejamos como montar a carteira. Se queremos minimizar σ para um certo α , podemos utilizar uma função de Lagrange que depende de σ e de algumas restrições:

$$\Lambda = \frac{1}{2} \vec{W}^T S \vec{W} + \lambda(1 - \vec{W} \cdot \vec{e}) + \mu(\alpha_0 - \vec{W} \cdot \vec{\alpha}) \quad (7)$$

onde λ e μ são multiplicadores de Lagrange; $\vec{W}^T S \vec{W} = \sigma_W$ e $\vec{W} \cdot \vec{\alpha} = \alpha_W$, ou seja, o desvio padrão e o retorno esperado da carteira, respectivamente. O fator $\frac{1}{2}$ serve para simplificar as contas, mas não interfere no resultado final. Repare que a menos desse fator, idealmente teremos que $\Lambda = \sigma_W$, portanto minimizar Λ é equivalente a minimizar σ . Minimizando Λ em relação a \vec{W} obtemos a seguinte expressão:

$$\vec{W}_0 = \lambda S^{-1} \vec{e} + \mu S^{-1} \vec{\alpha} \quad (8)$$

Para encontrarmos λ e μ devemos minimizar Λ também em relação a eles. Fazendo isso, chegamos na seguinte expressão:

$$\begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} \vec{e} \cdot S^{-1} \vec{e} & \vec{\alpha} \cdot S^{-1} \vec{e} \\ \vec{e} \cdot S^{-1} \vec{\alpha} & \vec{\alpha} \cdot S^{-1} \vec{\alpha} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ \alpha_0 \end{bmatrix} = M^{-1} \begin{bmatrix} 1 \\ \alpha_0 \end{bmatrix} \quad (9)$$

Para obtermos \vec{W}_0 , falta ainda determinarmos α_0 . Ele pode ser escolhido de forma arbitrária, desde que respeite a relação abaixo:

$$\sigma_W^2 = \begin{bmatrix} 1 & \alpha_0 \end{bmatrix} M^{-1} \begin{bmatrix} 1 \\ \alpha_0 \end{bmatrix} \quad (10)$$

Vamos expandir a relação (10) para entendê-la melhor. Usando $M^{-1} = \omega$ temos que:

$$\sigma_W^2 = \begin{bmatrix} 1 & \alpha_0 \end{bmatrix} \begin{bmatrix} \omega_{00} & \omega_{01} \\ \omega_{10} & \omega_{11} \end{bmatrix} \begin{bmatrix} 1 \\ \alpha_0 \end{bmatrix} = \begin{bmatrix} 1 & \alpha_0 \end{bmatrix} \begin{bmatrix} \omega_{00} + \omega_{01}\alpha_0 \\ \omega_{10} + \omega_{11}\alpha_0 \end{bmatrix} = \omega_{00} + \omega_{01}\alpha_0 + \omega_{10}\alpha_0 + \omega_{11}\alpha_0^2$$

Reorganizando os termos da expressão acima obtemos a seguinte equação quadrática:

$$\omega_{11}\alpha_0^2 + (\omega_{01} + \omega_{10})\alpha_0 + \omega_{00} - \sigma_W^2 = 0 \quad (11)$$

Quanto maior o valor de σ_W^2 , maior a volatilidade dos nossos retornos e maior o risco que estamos assumindo. Uma vez que aceitamos um certo risco escolhendo um valor para σ_W^2 , determinamos α_0 usando a expressão (11) - considerando apenas a raiz positiva (caso exista), pois queremos um retorno positivo; daí então obtemos λ e μ usando a relação (9) e finalmente determinamos \vec{W}_0 com a expressão (8). Aqui devemos tomar um cuidado: \vec{W}_0 pode conter termos negativos. Esses termos satisfazem as relações matemáticas apresentadas mas não fazem sentido no contexto de investimentos. Precisamos forçar os pesos W_n a serem nulos ou positivos, e isso pode ser feito com a expressão abaixo:

$$W_n^* = \frac{MAX(0, W_n)}{\sum_m MAX(0, W_m)} \quad (12)$$

onde $\{W_n^*\}$ serão os novos pesos. Com isso concluímos a exposição dos conceitos teóricos. Na próxima seção é implementado o código para montar a carteira.

0.3 Implementação e resultados

0.3.1 Selecionando os ativos

Primeiramente, precisamos definir quais dentre os ativos candidatos entrarão na carteira, o que é determinado pela expressão (1). O trecho abaixo define uma função para calcular o β de um certo ativo:

```
[32]: def calcular_beta(a):
    #obtendo os retornos
    r_ativo = obter_retornos(a)

    #deixando as duas listas de retorno com os mesmos tamanhos
    Min = min(len(r_ibovespa), len(r_ativo))
    while len(r_ibovespa) > Min: del r_ibovespa[-1]
```

```

while len(r_ativo) > Min: del r_ativo[-1]

#calculando beta
cov_am = pd.Series(r_ibovespa).cov(pd.Series(r_ativo)) #covariancia entre
↪ativo e mercado
beta = cov_am/var_m
return(beta)

```

O trecho abaixo nos dá a lista de retornos diários do Bovespa:

```

[33]: r_ibovespa = obter_retornos('~BVSP')
var_m = np.var(r_ibovespa) #variancia do mercado
Rm = np.mean(r_ibovespa) #retorno medio do mercado

```

Agora, com o trecho abaixo calculamos o excedente do retorno esperado de cada ativo em relação ao mínimo esperado. Vamos lembrar que esse excedente precisa ser positivo, e no máximo 9 ativos com os maiores excedentes serão os selecionados (serão menos caso não hajam 9 ativos com excedentes positivos):

```

[34]: lucros = []
for ativo in ativos:
    #obtendo beta
    beta = calcular_beta(ativo)

    #obtendo a taxa selic diaria
    dfs = pd.read_csv('selic.csv',delimiter=';')
    x = dfs[dfs.columns[1]].tolist()
    del x[0]
    for i in range(len(x)): x[i] = float(x[i].replace(',', '.'))
    r0 = np.mean(x)/(100*365)

    #calculando o lucro em relacao ao valor minimo esperado
    Ri = np.mean(r_ibovespa) #retorno medio do ativo
    Re = r0+beta*(Rm-r0) #retorno minimo esperado
    lucros.append([ativo,round(100*(Ri/Re-1),2)])

```

No trecho acima, a taxa Selic T_S foi determinada da seguinte maneira: foi obtida, da referência [2], a série temporal das porcentagens da taxa Selic ao longo dos últimos 5 anos e tomou-se uma média P_a desses valores. Note que essas porcentagens são anuais, e estamos interessados em uma taxa diária, já que nossos ativos contêm retornos diários. Uma forma de obter isso é dividindo P_a por 365, o que nos dá $T_S = \frac{T_a}{100 \cdot 365}$. Finalmente, o trecho abaixo seleciona os ativos:

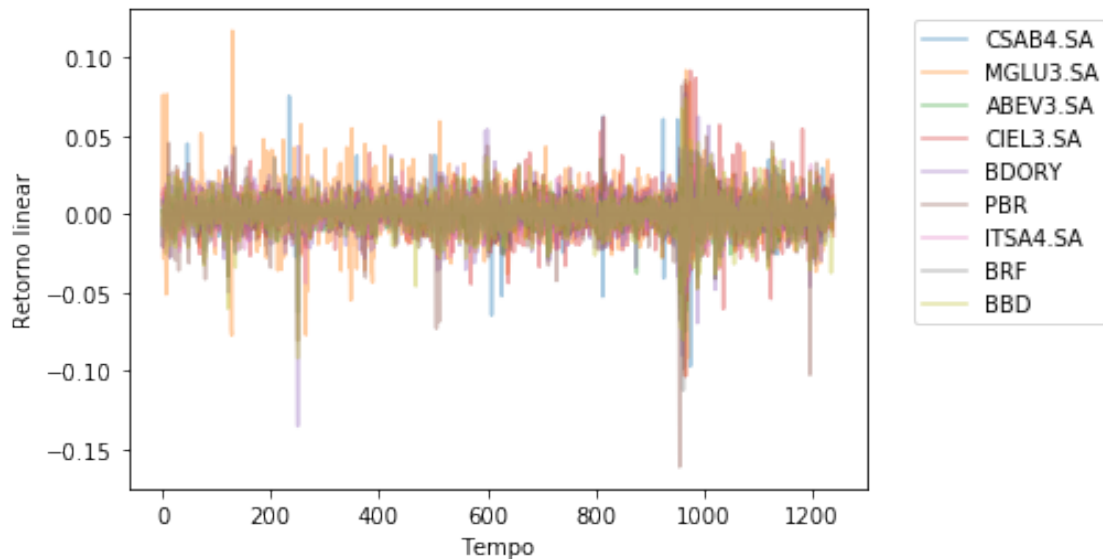
```

[35]: dfa = pd.DataFrame(lucros,columns=['ativo','lucro']) #dataframe dos ativos
dfa = dfa.sort_values('lucro',ascending=False).reset_index(drop=True)
dfa = dfa.head(9) #vou selecionar ate 9 dos melhores ativos
dfa = dfa.drop(dfa[dfa.lucro < 0].index) #remove ativos com lucro menor que o
↪mínimo
ativos = dfa['ativo'].tolist() #seleciona os melhores ativos

```

Foram selecionados 9 ativos, cujas séries de retornos podem ser visualizadas com o trecho abaixo:

```
[36]: R = comparar_acoes(ativos)
qr = len(R[0]) #qtd de retornos
for i in range(len(R)): R[i] = np.array(R[i])
```



Vamos agora compor nossa carteira usando a Teoria de Markowitz.

0.3.2 Compondo a carteira

Já determinamos \vec{R} com o trecho cima. Vamos agora obter $\vec{\alpha}$, S e M :

```
[37]: #obtendo o vetor alpha
alpha = []
for i in R: alpha.append(np.mean(i))
alpha = np.array(alpha)

#obtendo a matriz de covariancias
R_alpha = []
for i in range(len(R)): R_alpha.append(R[i]-alpha[i])
S = np.matmul(R_alpha,np.transpose(R_alpha))
S = S/qr

#obtendo a matriz M
Si = np.linalg.inv(S) #matriz inversa de S
e = np.array([1]*len(alpha)) #vetor de unidades
M = np.matrix([[e.dot(Si.dot(e)),alpha.dot(Si.dot(e))],[e.dot(Si.
    dot(alpha)),alpha.dot(Si.dot(alpha))]])
```

O próximo passo é determinar α_0 . Aqui a abordagem utilizada foi a seguinte: verificamos qual

o ativo de maior retorno dentro da carteira, tomamos a variância desse ativo e a colocamos na expressão (11) para calcular α_0 . Ou seja, queremos montar nossa carteira de forma que seu σ seja igual ao σ do ativo de maior retorno. Com isso poderemos verificar a vantagem de fazer investimentos usando uma carteira diversificada: veremos que, assumindo o mesmo risco que assumiríamos com o ativo de maior retorno (caso fôssemos investir apenas nele), poderemos obter um retorno ainda maior investindo em ativos diversificados. O trecho abaixo calcula α_0 :

```
[38]: #obtendo alpha0
Mi = np.linalg.inv(M) #matriz inversa de M
lrd = [] #lista de retornos e desvios
for i in range(len(R)): lrd.append([np.std(R[i]),np.mean(R[i]),ativos[i]])
dfrd = pd.DataFrame(lrd,columns=['risco','retorno','ativo']) #desvio padrao,
    ↳retorno, ativo
aMret = dfrd.sort_values('retorno',ascending=False).reset_index(drop=True).
    ↳values.tolist()[0] #ativo de maior retorno
sigma = aMret[0] #risco do ativo de maior retorno

W = Mi.tolist()
alpha0 = max(np.roots([W[1][1],W[0][1]+W[1][0],W[0][0]-sigma**2])).tolist()
```

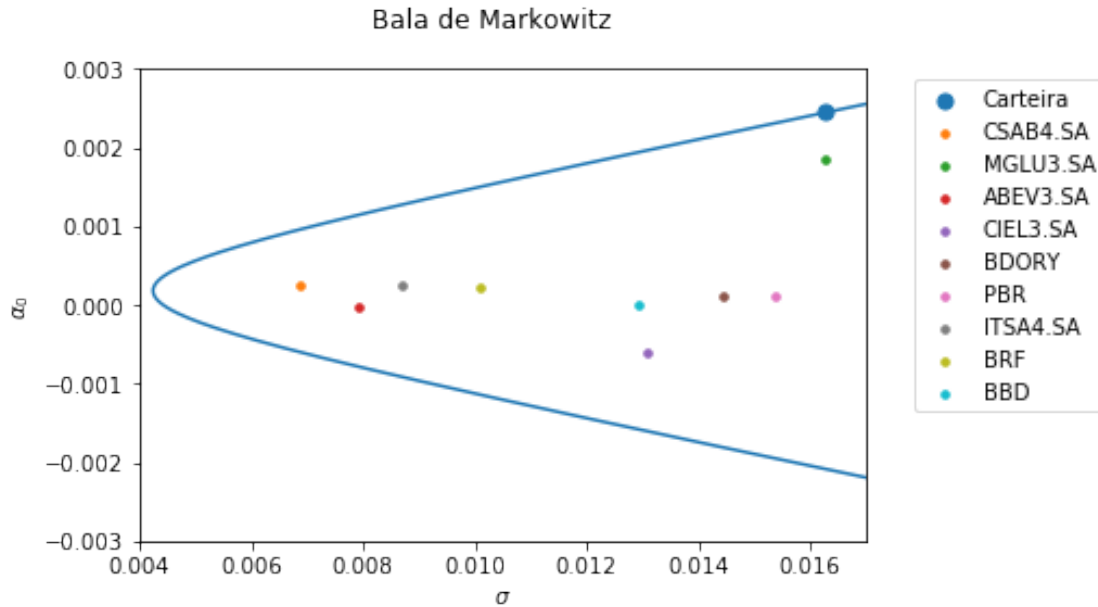
Aqui cabe fazermos uma inspeção visual das relações entre σ e α dos nossos ativos e da nossa carteira. Usando a expressão (11) obtemos uma curva α_0 vs σ , chamada de Bala de Markowitz:

```
[39]: lalpha0 = np.linspace(-0.1,0.1,10000).tolist() #lista de alpha0's
lsigma = [] #lista de sigma's

for i in lalpha0: lsigma.append((np.array([1,i]).dot(np.transpose(Mi.dot(np.
    ↳array([1,i])))).tolist()[0][0])**0.5)

plt.suptitle('Bala de Markowitz')
plt.xlabel(r'$\sigma$')
plt.ylabel(r'$\alpha_0$')
plt.xlim([0.004,0.017])
plt.ylim([-0.003,0.003])
plt.plot(lsigma,lalpha0)
plt.scatter(sigma,alpha0,s=50,label='Carteira')
for i in range(len(R)): plt.scatter(np.std(R[i]),np.
    ↳mean(R[i]),s=12,label=ativos[i])
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
```

```
[39]: <matplotlib.legend.Legend at 0x7fe1f86dce20>
```

No gráfico acima, temos pontos associados a cada ativo individual e um ponto maior associado à carteira. A Bala de Markowitz delimita as relações possíveis entre retornos e variâncias: para qualquer carteira, todos os pontos associados a seus ativos individuais sempre estarão dentro da convexidade da Bala, e os pontos que compõem a Bala em si mostram o maior retorno possível de se obter para cada σ (considerando apenas a curva de retornos positivos). Tendo selecionado o retorno desejado, podemos finalmente determinar os pesos de cada ativo:

```
[42]: #obtendo os pesos da carteira
mL = Mi.dot(np.array([1,alpha0])).tolist() #multiplicadores de Lagrange
lanbda, mu = mL[0][0], mL[0][1]
w0 = lanbda*(Si.dot(e))+mu*(Si.dot(alpha)) #pesos nao normalizados
w0 = w0.tolist()
for i in range(len(w0)): w0[i] = max(0,w0[i])
soma = sum(w0)
for i in range(len(w0)): w0[i] /= soma
dfc = pd.DataFrame(w0,columns=['peso']) #data frame da carteira
dfc['ativo'] = ativos
dfc = dfc.drop(dfc[dfc.peso == 0].index) #remove ativos sem peso
dfc = dfc.reset_index(drop=True)
dfc['peso'] = round(dfc['peso']*100,2)
dfc
```

```
[42]:      peso      ativo
0   19.17  CSAB4.SA
1   35.30  MGLU3.SA
2   11.71  ITSA4.SA
3   33.82      BRF
```

Na tabela acima, os pesos estão expressos em porcentagem. Percebemos que a maior parte de nossos investimentos serão destinadas à empresa Magazine Luiza (sigla MGLU3.SA), o que já era de se esperar, visto que ela é a empresa de maior retorno individual. Notamos ainda que, na prática, dos 9 ativos selecionados pelo método CAPM, apenas 4 deles irão efetivamente compor a carteira (os outros possuem peso 0 e foram eliminados automaticamente pelo código). Vamos verificar a razão entre α_0 e o retorno da Magazine Luiza (obtido da variável “dfrd”):

```
[41]: round(alpha0/0.00186163,2)*100
```

```
[41]: 131.0
```

Ou seja, usando os métodos descritos aqui, conseguimos obter um retorno 31% maior do que teríamos para o ativo de maior retorno individual, assumindo o mesmo risco.

0.4 Referências

[1] <https://finance.yahoo.com/>

[2] <https://www.bcb.gov.br/htms/SELIC/SELICdiarios.asp?frame=1>