

IMPORTANDO BIBLIOTECAS

```
In [ ]: import pandas as pd
import sgs
import numpy as np
import yfinance as yf

%matplotlib inline

import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from tqdm import tqdm
import random
plt.style.use('dark_background')

import warnings
warnings.filterwarnings("ignore")
```

BAIXANDO BASE DE DADOS NECESSÁRIAS

```
In [ ]: tickers = ['MSFT34.SA', 'RADL3.SA', 'EGIE3.SA', 'BBAS3.SA']
df = yf.download(
    tickers=tickers,
    interval='1d',
    start='2015-01-01',
    end='2024-06-11')['Adj Close']

df_bova = yf.download(
    tickers=('BOVA11.SA'),
    interval='1d',
    start='2015-01-01',
    end='2024-06-11')['Adj Close']

SELIC_CODE = 11
df_selic = sgs.time_series(SELIC_CODE, start='02/01/2015', end='10/06/2024')

df_retorno = pd.DataFrame()

for i in df.columns:
    df_retorno[f'ln_{i}'] = np.log(df[i].shift(-1) / df[i])
```

```
df_retorno.dropna(inplace=True)
```

```
[*****100%*****] 4 of 4 completed  
[*****100%*****] 1 of 1 completed
```

Risco / Retorno dos ativos

```
In [ ]: risco_ativos = df_retorno.std()  
risco_ativos
```

```
Out[ ]: ln_BBAS3.SA    0.026209  
ln_EGIE3.SA    0.014697  
ln_MSFT34.SA    0.018971  
ln_RADL3.SA    0.019195  
dtype: float64
```

```
In [ ]: retorno_ativos = df_retorno.mean()  
retorno_ativos
```

```
Out[ ]: ln_BBAS3.SA    0.000606  
ln_EGIE3.SA    0.000457  
ln_MSFT34.SA    0.001299  
ln_RADL3.SA    0.000745  
dtype: float64
```

DEFININDO O PESO DA CARTEIRA SEGUINDO MARKOWITZ/SHARPE

```
In [ ]: pesos = [0.05,0.40,0.30,0.25]
```

Covariância entre os ativos

```
In [ ]: cov_retornos = df_retorno.cov()  
cov_retornos
```

Out []:

	In_BBAS3.SA	In_EGIE3.SA	In_MSFT34.SA	In_RADL3.SA
In_BBAS3.SA	0.000687	0.000192	-0.000022	0.000157
In_EGIE3.SA	0.000192	0.000216	0.000009	0.000097
In_MSFT34.SA	-0.000022	0.000009	0.000360	0.000023
In_RADL3.SA	0.000157	0.000097	0.000023	0.000368

Estimando o Retorno do Portfólio

```
In [ ]: retorno_portfolio = (retorno_ativos*pesos).sum()  
retorno_portfolio
```

Out []: 0.0007889918612724181

Estimando o Risco do Portfólio

```
In [ ]: df_retorno
```

Out []:

	In_BBAS3.SA	In_EGIE3.SA	In_MSFT34.SA	In_RADL3.SA
Date				
2015-01-02	-0.020969	0.001226	0.005143	-0.006826
2015-01-05	0.013880	0.020012	0.099820	0.028596
2015-01-06	0.043078	0.024905	-0.111650	0.017081
2015-01-07	0.003401	0.010197	0.015847	0.002691
2015-01-08	-0.044259	-0.079922	-0.014809	-0.015474
...
2024-06-03	0.002942	0.001589	0.019476	0.000000
2024-06-04	0.002201	-0.006372	0.018997	-0.009176
2024-06-05	0.013105	0.004783	-0.006005	0.007188
2024-06-06	-0.017144	-0.012116	0.011869	0.007532
2024-06-07	0.001103	0.000000	0.014667	-0.008727

2345 rows × 4 columns

In []:

```
df_risco_retorno = pd.concat([risco_ativos, retorno_ativos], axis=1)
df_risco_retorno.rename(columns={0: 'Risco', 1: 'Retorno'}, inplace=True)
df_risco_retorno['W'] = pesos
df_risco_retorno['W²'] = np.square(pesos)
df_risco_retorno['sigma²'] = np.square(df_risco_retorno['Retorno'])
df_risco_retorno['W² x sigma²'] = df_risco_retorno['W²'] * df_risco_retorno['sigma²']
df_risco_retorno
```

Out []:

	Risco	Retorno	W	W²	sigma²	W² x sigma²
In_BBAS3.SA	0.026209	0.000606	0.05	0.0025	3.666729e-07	9.166823e-10
In_EGIE3.SA	0.014697	0.000457	0.40	0.1600	2.088951e-07	3.342321e-08
In_MSFT34.SA	0.018971	0.001299	0.30	0.0900	1.687826e-06	1.519043e-07
In_RADL3.SA	0.019195	0.000745	0.25	0.0625	5.544046e-07	3.465029e-08

In []:

cov_retornos

Out []:

	In_BBAS3.SA	In_EGIE3.SA	In_MSFT34.SA	In_RADL3.SA
In_BBAS3.SA	0.000687	0.000192	-0.000022	0.000157
In_EGIE3.SA	0.000192	0.000216	0.000009	0.000097
In_MSFT34.SA	-0.000022	0.000009	0.000360	0.000023
In_RADL3.SA	0.000157	0.000097	0.000023	0.000368

```
In [ ]: # somatorio de (W^2 x sigma^2)
# 2 x W[BBAS3.SA] x W[EGIE3.SA] x covariancia [BBAS3.SA - EGIE3.SA]
# 2 x W[BBAS3.SA] x W[MSFT34.SA] x covariancia [BBAS3.SA - MSFT34.SA]
# 2 x W[BBAS3.SA] x W[RADL3.SA] x covariancia [BBAS3.SA - RADL3.SA]

# 2 x W[EGIE3.SA] x W[MSFT34.SA] x covariancia [EGIE3.SA - MSFT34.SA]
# 2 x W[EGIE3.SA] x W[RADL3.SA] x covariancia [EGIE3.SA - RADL3.SA]

# 2 x W[MSFT34.SA] x W[RADL3.SA] x covariancia [MSFT34.SA - RADL3.SA]

risco_portfolio = np.sqrt(
    df_risco_retorno['W^2 x sigma^2'].sum()
    + (2 * df_risco_retorno['W'][0] * df_risco_retorno['W'][1] * cov_retornos[f'ln_{tickers[0]}'][1])
    + (2 * df_risco_retorno['W'][0] * df_risco_retorno['W'][2] * cov_retornos[f'ln_{tickers[0]}'][2])
    + (2 * df_risco_retorno['W'][0] * df_risco_retorno['W'][3] * cov_retornos[f'ln_{tickers[0]}'][3])
    + (2 * df_risco_retorno['W'][1] * df_risco_retorno['W'][2] * cov_retornos[f'ln_{tickers[1]}'][2])
    + (2 * df_risco_retorno['W'][1] * df_risco_retorno['W'][3] * cov_retornos[f'ln_{tickers[1]}'][3])
    + (2 * df_risco_retorno['W'][2] * df_risco_retorno['W'][3] * cov_retornos[f'ln_{tickers[2]}'][3])
)
risco_portfolio
```

Out []: 0.01028428717665578

Retorno do Portfólio

```
In [ ]: print(f'O retorno do portfólio foi de {round(retorno_portfolio*100,2)}% a.d.')
```

O retorno do portfólio foi de 0.08% a.d.

Risco do Portfólio

```
In [ ]: print(f'O risco do portfólio foi de {round(risco_portfolio*100,2)}% a.d.')
```

O risco do portfólio foi de 1.03% a.d.

Calculando o alfa e o beta da sua carteira por meio do CAPM

Retorno da carteira (Rp):

```
In [ ]: rp = retorno_portfolio
```

Retorno do mercado (Rm)

```
In [ ]: retorno_bova = np.log(df_bova.shift(-1) / df_bova)
rm = retorno_bova.mean()
rm
```

```
Out [ ]: 0.0003923626717217679
```

Taxa livre de risco (Rf)

```
In [ ]: retorno_selic = np.log(df_selic.shift(-1) / df_selic)
rf = retorno_selic.mean()
rf
```

```
Out [ ]: -4.5591913717772645e-05
```

Covariância entre os retornos da carteira e do mercado

```
In [ ]: carteira = df_retorno*pesos
carteira['carteira'] = carteira[f'ln_{tickers[0]}']+carteira[f'ln_{tickers[1]}']+carteira[f'ln_{tickers[2]}']+carteira[f'ln_{tickers[3]}']
df_cov_portfolio_mercado = pd.concat([carteira['carteira'],retorno_bova], axis=1)
df_cov_portfolio_mercado.rename(columns={'Adj Close':'mercado'}, inplace=True)
df_cov_portfolio_mercado = df_cov_portfolio_mercado.cov()
cov_portfolio_mercado = df_cov_portfolio_mercado['mercado'][1]
cov_portfolio_mercado
```

```
Out [ ]: 0.00024889933404975754
```

Variância dos retornos do mercado

```
In [ ]: var_retornos_mercado = retorno_bova.var()
var_retornos_mercado
```

```
Out [ ]: 0.0002488993340497582
```

Beta (b)

```
In [ ]: beta = cov_portfolio_mercado/var_retornos_mercado  
beta
```

```
Out[ ]: 0.9999999999999973
```

Alpha (a)

```
In [ ]: alpha = rp - (rf + beta*(rm - rf))  
alpha
```

```
Out[ ]: 0.00039662918955065145
```

```
In [ ]:
```