

Aluno: Rafael dos Santos

Lista: Exercício 2 - 29/Mar até 9h30

- Utilizar os dados de PETR3 **intradia** no Blackboard
- Usar o modelo de simulação acima usando a biblioteca backtesting
- Adaptar a estratégia Martingale:
 - Diminuir a posição para 1 quando o preço for 3% acima do preço médio
 - Quando fizer o stop gain, estabeleça o novo preço médio como o preço de venda
 - Se o preço cair, deixe a estratégia agir, dobrando a posição
 - Interprete o resultado
- Montar um notebook próprio, não reutilizar o da aula.
- Entregar um **PDF** com o código da estratégia
- Prazo: 29/Mar até 9h30 via Blackboard (Após esse prazo será considerado atrasado)
- Estritamente individual

```
In [ ]: from backtesting import evaluateIntr, Strategy, Order
import pandas as pd
from glob import glob
```

```
In [ ]: from glob import glob
# read data from all csvs in a folder named PETR4-Dataset and join them
data = pd.concat([pd.read_csv(f, sep=";", decimal=",")
                  for f in sorted(glob('PETR4-Dataset/*.csv'))])
# set the index to date
data.set_index('Dates', inplace=True)
# save data to csv
data.to_csv('PETR4.csv', sep=';', decimal=',')

# data['Close'].plot();
```

```
In [ ]: class Estrategi(Strategy):

    def __init__(self):
        self.trades = []
        self.size = 0
        self.average = 1

    def push(self, event):
        price = event.price[3] # Captura o preço atual vindo do evento

        orders = []

        ret = price/self.average - 1

        if self.size == 0: # Se está zerado
            orders.append(Order(event.instrument, 1, 0))
            self.size = 1
            self.average = price

        # stop loss
        elif ret <= -0.03: # Se está caindo 3% do preço médio

            # dobra a posição
```

```
orders.append(Order(event.instrument, self.size, 0))
self.size *= 2
# novo médio
self.average += price
self.average /= 2
# print(f"{price} {self.average} {self.size}")

# stop gain
elif ret >= 0.03: # Se está subindo 3% do preço médio

    # reseta a posição para 1
    orders.append(Order(event.instrument, 1 - self.size, 0))

    self.size = 1
    # reseta o médio
    self.average = price
    # print(f"{price} {self.average} {self.size}")

self.submit(self.id, orders)

print(evaluateIntr(Estrategi(), {"PETR4": "PETR4.csv"}))
```

Number of trades: 1
Gross P&L: 442.25
Gross Accumulated return: 3.12%
Gross Average Return: 3.12%
Net P&L: 43544.29
Hitting ratio: 100.0%