

Strategie de la Moyenne Mobile sur l'indice BRVM-Services-Public

importation des bibliotheques

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from print_color import print
import locale
from datetime import datetime
locale.setlocale(locale.LC_TIME, 'fr_FR.UTF-8')

'fr_FR.UTF-8'

BRVM = pd.read_csv("../data/BRVM-Public-Services.csv", index_col="Date")

debut = -365
fin = len(BRVM)
brvm = BRVM.iloc[debut:]
```

Calcul des moyennes mobiles longues et courtes

```
# creation de de colonne de moyenne mobile

# moyenne mobile rapide / court terme
brvm["MA10"] = brvm["Close"].rolling(6).mean() #10

# moyenne mobile lente / long terme
brvm["MA50"] = brvm["Close"].rolling(10).mean() #40

brvm = brvm.dropna() ;

# creation de la liste position
condition = [brvm["MA10"]>brvm["MA50"] , brvm["MA10"]<brvm["MA50"]]
# creation d'une liste choix de deux element
choix = [1,0]
#
brvm['P'] = np.select(condition, choix)

brvm
```

	Dernier	Ouv.	Plus Haut	Plus Bas	Variation %	Close	High	\
Date								
14/12/2021	443,01	436,88	443,01	436,88	1,40%	443.01	443.01	
15/12/2021	435,24	443,01	443,57	435,24	-1,75%	435.24	443.57	
16/12/2021	436,14	435,24	436,14	435,24	0,21%	436.14	436.14	
17/12/2021	434,41	436,14	436,14	434,41	-0,40%	434.41	436.14	

20/12/2021	436,12	434,41	438,06	434,41	0,39%	436.12	438.06
...
12/05/2023	489,95	486,77	490,69	483,95	0,65%	489.95	490.69
15/05/2023	468,96	489,95	489,95	459,94	-4,28%	468.96	489.95
16/05/2023	468,02	468,96	469,30	464,42	-0,20%	468.02	469.30
17/05/2023	467,68	468,02	469,67	462,53	-0,07%	467.68	469.67
19/05/2023	465,82	467,68	468,15	459,58	-0,40%	465.82	468.15

Date	Low	Open	MA10	MA50	P
14/12/2021	436.88	436.88	443.201667	445.231	0
15/12/2021	435.24	443.01	441.356667	443.788	0
16/12/2021	435.24	435.24	440.021667	442.452	0
17/12/2021	434.41	436.14	438.063333	441.227	0
20/12/2021	434.41	434.41	436.966667	440.112	0
...
12/05/2023	483.95	486.77	483.473333	481.239	1
15/05/2023	459.94	489.95	481.510000	480.598	1
16/05/2023	464.42	468.96	479.605000	479.758	0
17/05/2023	462.53	468.02	477.275000	478.623	0
19/05/2023	459.58	467.68	474.533333	477.132	0

[356 rows x 12 columns]

Vérification des conditions d'achat et de vente du marché

A l'entré sur le marché nous ne pocédons pas de stock. Au premier signal d'achat nous achetons des actifs. Si il y a d'autre signal d'achat nous n'achetons pas car nous avons déjà des actifs. De meme nous ne vendons pas si nous n'avons pas de stock en cas de signal de vente

```

condition = []
signal_achat = []
signal_vente = []
date_1 = brvm.index[0]
stock = 0 # permet d'éviter d'acheter quand on pas de stock au debut

for date in brvm.index :
    if brvm["MA10"].loc[date] > brvm["MA50"].loc[date] :
        condition.append(1)
    else :
        condition.append(0)

    if( brvm["MA10"].loc[date] > brvm["MA50"].loc[date] ) and (brvm["MA10"].loc[date_1] < brvm["MA50"].loc[date_1]) :
        signal_achat.append(brvm["Close"].loc[date])
        stock = 1

```

```

else :
    signal_achat.append(np.nan)

if( brvm["MA10"].loc[date] < brvm["MA50"].loc[date] ) and (brvm["MA10"].loc[date_1] > brvm["MA50"].loc[date_1]) :
    signal_vente.append(brvm["Close"].loc[date])
    stock = 0
else :
    signal_vente.append(np.nan)

date_1 = date

brvm["Position"] = condition

for i in range(len(brvm)-1 , 0 , -1 ) :
    if signal_achat[i] > 0 :
        signal_achat[i] = np.nan
        break

achat_vente = []
stock = 0
for i in range( len(brvm) ) :
    if signal_achat[i] > 0 and stock != 1:
        achat_vente.append('acheter')
        stock = 1
    elif signal_vente[i] > 0 and stock != 0 :
        achat_vente.append('vendre' )
        stock=0
    else :
        achat_vente.append(np.nan)

brvm['achat_vente'] = achat_vente

fig = plt.figure(figsize=(14,7), dpi=150)

axe1 = fig.add_axes([0,1,1,0.9])

axe1 = brvm["Close"].plot(label='Prix de cloture')
axe1 = brvm["MA10"].plot(label='Moyenne mobile rapide')
axe1 = brvm["MA50"].plot(label='Moyenne mobile lente')

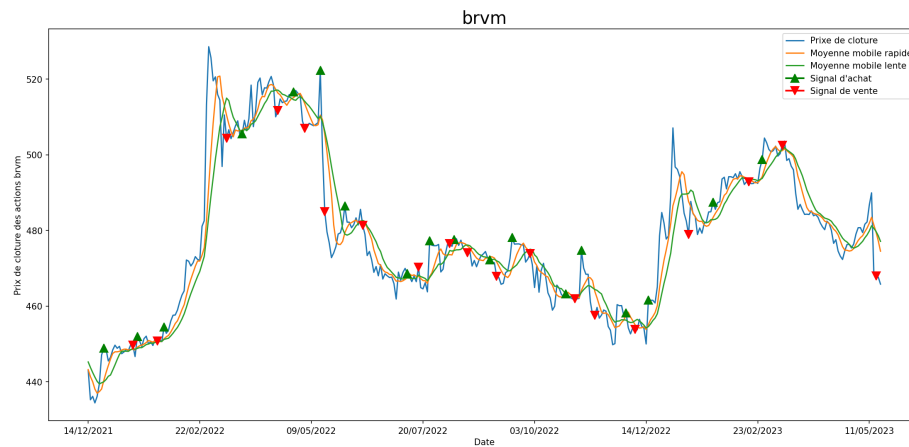
axe1.plot(brvm.index , signal_achat , marker='^',color='green',markersize=10,label = "Signal d'achat")
axe1.plot(brvm.index , signal_vente , marker='v',color='red',markersize=10,label = "Signal de vente")

```

```

axe1.legend()
axe1.set_title("brvm" , fontsize=20)
axe1.set_ylabel("Prix de cloture des actions brvm")
Text(0, 0.5, 'Prix de cloture des actions brvm')

```



Backtesting

```

benefice = 0
depart = 1000
d = depart
nombre = 0
total = 0

for i in range( len( brvm ) ) :
    if brvm['achat_vente'].iloc[i] == 'acheter' :

        nombre = depart/brvm['Close'].iloc[i]
        entrer = nombre*brvm['Close'].iloc[i]

    elif brvm['achat_vente'].iloc[i] == 'vendre' :

        sorti = (nombre*brvm['Close'].iloc[i])
        benefice += sorti-entrer

        total += sorti
        depart = sorti
    else :
        pass

pourcentage_befice = 100*(benefice/d)

```

Resulatats du backtesting

```
date_depart = datetime.strptime(BRVM.iloc[debut].name , "%d/%m/%Y" ).strftime("%A %d %B, %Y")
date_fin = datetime.strptime(BRVM.iloc[fin-1].name , "%d/%m/%Y" ).strftime("%A %d %B, %Y ")
```

```
print("Avec un capitale de depart de " ,end="")
print("{:,.2f} Fcfa".format(d) , color='green')
print("nous avons réalisé un benefice de ",end='')
print("{:,.2f} %".format(pourcentage_befice) , color='green')
print("Soit un benefice total de ",end='')
print("{:,.2f} Fcfa".format(benefice) , color='green')
print("Sur la periode allant du ",end='')
print(date_depart , color="blue" , end="") ; print(" au " ,end="")
print(date_fin , color="blue" , end="")
```

Avec un capitale de depart de 1,000.00 Fcfa

nous avons réalisé un benefice de -0.63 %

Soit un benefice total de -6.35 Fcfa

Sur la periode allant du mercredi 01 décembre, 2021 au vendredi 19 mai, 2023