1. Instalação e importação de bibliotecas

```
!pip install python-binance
!pip instal plotly
```

```
Collecting python-binance
  Downloading python_binance-1.0.15-py2.py3-none-any.whl (63 kB)
     |████████████████████████████████| 63 kB 2.4 MB/s
Collecting ujson
  Downloading ujson-5.1.0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (43 kB)
     |████████████████████████████████| 43 kB 2.2 MB/s
Collecting dateparser
  Downloading dateparser-1.1.0-py2.py3-none-any.whl (288 kB)
     |████████████████████████████████| 288 kB 45.7 MB/s
Collecting aiohttp
  Downloading aiohttp-3.8.1-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_64.whl (1.1 MB)
     |████████████████████████████████| 1.1 MB 50.0 MB/s
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from python-binance) (1.15.0)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from python-binance) (2.23.0)
Collecting websockets==9.1
  Downloading websockets-9.1-cp37-cp37m-manylinux2010_x86_64.whl (103 kB)
     |████████████████████████████████| 103 kB 59.5 MB/s
Collecting asynctest==0.13.0
  Downloading asynctest-0.13.0-py3-none-any.whl (26 kB)
Collecting frozenlist>=1.1.1
  Downloading frozenlist-1.3.0-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (144 kB)
     |████████████████████████████████| 144 kB 58.4 MB/s
Collecting multidict<7.0,>=4.5
  Downloading multidict-6.0.2-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (94 kB)
     |████████████████████████████████| 94 kB 2.8 MB/s
Collecting yarl<2.0,>=1.0
  Downloading yarl-1.7.2-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_64.whl (271 kB)
     |████████████████████████████████| 271 kB 48.3 MB/s
Collecting async-timeout<5.0,>=4.0.0a3
  Downloading async_timeout-4.0.2-py3-none-any.whl (5.8 kB)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.7/dist-packages (from aiohttp->python-binance) (21.4.0)
Collecting aiosignal>=1.1.2
  Downloading aiosignal-1.2.0-py3-none-any.whl (8.2 kB)
```

```
Requirement already satisfied: typing-extensions>=3.7.4 in /usr/local/lib/python3.7/dist-packages (from aiohttp->python-binance) (3.10.0.
Requirement already satisfied: charset-normalizer<3.0,>=2.0 in /usr/local/lib/python3.7/dist-packages (from aiohttp->python-binance) (2.0
Requirement already satisfied: idna>=2.0 in /usr/local/lib/python3.7/dist-packages (from yarl<2.0,>=1.0->aiohttp->python-binance) (2.10)
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from dateparser->python-binance) (2018.9)
Requirement already satisfied: regex!=2019.02.19,!=2021.8.27 in /usr/local/lib/python3.7/dist-packages (from dateparser->python-binance)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.7/dist-packages (from dateparser->python-binance) (2.8.2)
Requirement already satisfied: tzlocal in /usr/local/lib/python3.7/dist-packages (from dateparser->python-binance) (1.5.1)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests->python-binance) (2021.10.8)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests->python-b
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests->python-binance) (3.0.4)
Installing collected packages: multidict, frozenlist, yarl, asynctest, async-timeout, aiosignal, websockets, ujson, dateparser, aiohttp,
Successfully installed aiohttp-3.8.1 aiosignal-1.2.0 async-timeout-4.0.2 asynctest-0.13.0 dateparser-1.1.0 frozenlist-1.3.0 multidict-6.0
ERROR: unknown command "instal" - maybe you meant "install"
```

```
import pandas as pd
from binance.client import Client
import plotly.graph_objects as go
from·plotly.subplots·import·make_subplots
```

```
client = Client()
```

```
btcbrl = client.get_historical_klines(symbol = "BTCBRL", start_str = "2021-01-01", end_str = "2022-02-08", interval = Client.KLINE_INTERVAL_1DA
```

```
type(btcbrl)
```

```
list
```

```
btc_brl = pd.DataFrame(btcbrl)
```

```
btc_brl.head()
```

	0	1	2	3	4	5	6	
0	1609459200000	152056.00000000	154167.00000000	150911.00000000	153480.00000000	58.94861700	1609545599999	9013538.827905
1	1609545600000	153478.00000000	173900.00000000	152171.00000000	168784.00000000	187.81836700	1609631999999	31072559.509199
2	1609632000000	168885.00000000	180925.00000000	167897.00000000	173503.00000000	172.84791800	1609718399999	30309745.064700
3	1609718400000	173504.00000000	175981.00000000	150000.00000000	169820.00000000	245.30932800	1609804799999	40581533.735731
4	1609804800000	169825.00000000	182966.00000000	159769.00000000	179972.00000000	229.90616900	1609891199999	39920823.605106

```
# Renomear as colunas
btc_brl.rename(columns={0:'date',1:'open',2:'high',3:'low',4:'close'},inplace=True)

# Configurar o indice do dataframe como sendo a data
btc_brl.set_index('date', inplace=True)

# Mudar o formato da data para padrão
btc_brl.index = pd.to_datetime(btc_brl.index,unit='ms')

# Dropar todas as colunas da 4 pra frente
btc_brl = btc_brl.drop(btc_brl.columns[4:],axis=1)
```

```
btc_brl.head()
```

date	open	high	low	close
2021-01-01	152056.00000000	154167.00000000	150911.00000000	153480.00000000
2021-01-02	153478.00000000	173900.00000000	152171.00000000	168784.00000000
2021-01-03	168885.00000000	180925.00000000	167897.00000000	173503.00000000
2021-01-04	173504.00000000	175981.00000000	150000.00000000	169820.00000000
2021-01-05	169825.00000000	182966.00000000	159769.00000000	179972.00000000

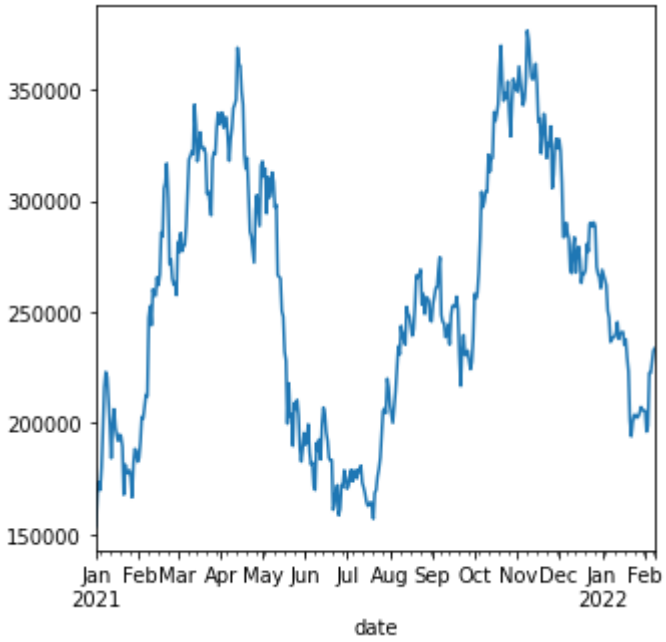
```
btc_brl.dtypes
```

```
open     object
high     object
low      object
close    object
dtype: object
```

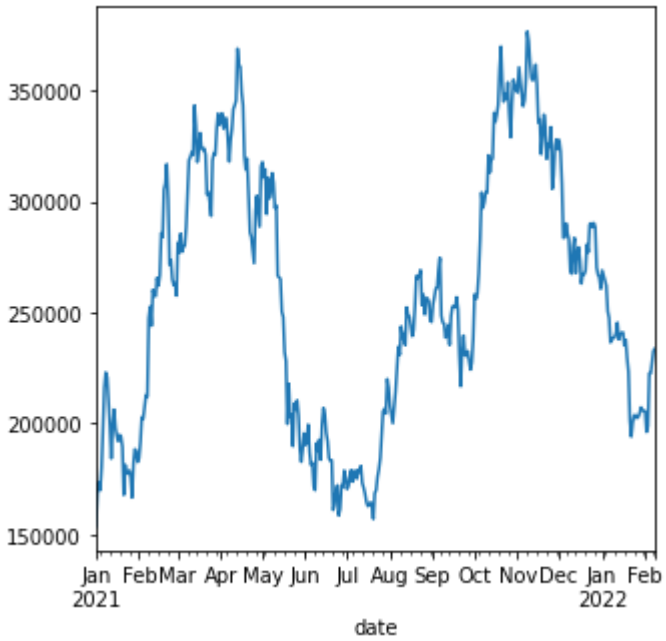
```
btc_brl['open'] = pd.to_numeric(btc_brl['open'])
btc_brl['high'] = pd.to_numeric(btc_brl['high'])
btc_brl['low'] = pd.to_numeric(btc_brl['low'])
btc_brl['close'] = pd.to_numeric(btc_brl['close'])
```

2. Análise Quantitativa

```
btc_brl['close'].plot(figsize = (5,5));
```




```
btc_brl.close.plot(figsize = (5,5));
```




```
btc_brl['retornos'] = btc_brl['close'].pct_change()
```

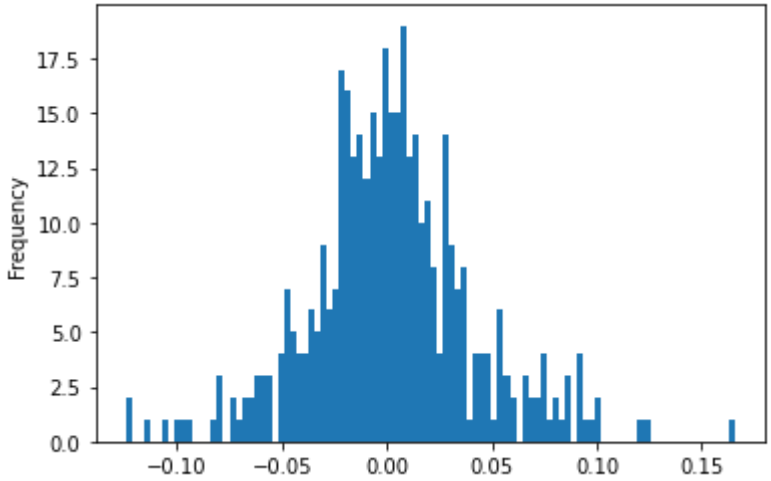
```
btc_brl.head()
```

date	open	high	low	close	retornos
2021-01-01	152056.0	154167.0	150911.0	153480.0	NaN
2021-01-02	153478.0	173900.0	152171.0	168784.0	0.099713
2021-01-03	168885.0	180925.0	167897.0	173503.0	0.027959
2021-01-04	173504.0	175981.0	150000.0	169820.0	-0.021227
2021-01-05	169825.0	182966.0	159769.0	179972.0	0.059781

```
btc_brl['retornos'].describe()
```

```
count    403.000000
mean       0.001815
std        0.039452
min       -0.124411
25%       -0.019981
50%        0.000319
75%        0.021763
max        0.165940
Name: retornos, dtype: float64
```

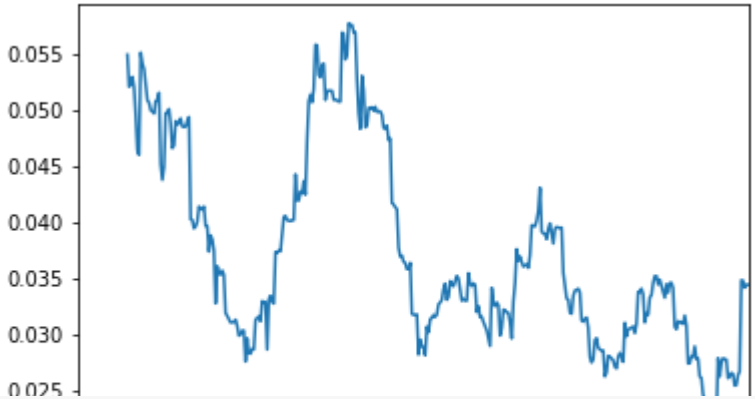
```
# Distribuição dos retornos do BTC

btc_brl['retornos'].plot.hist(bins = 100);
```




```
# Janela de volatilidade

vol_30 = btc_brl['retornos'].rolling(window = 30).std()
```

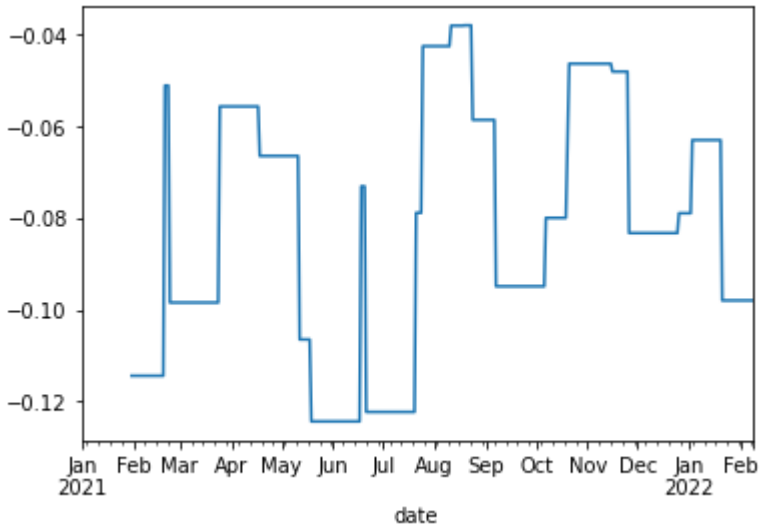
```
vol_30.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff8125fb5d0>
```




```
# Drawdown em janela de 30 dias
# Perda máxima em um determinado período - distância entre ganho máximo e a mínima

dd_30 = btc_brl['retornos'].rolling(window = 30).min()
```

```
dd_30.plot();
```



3. Criação dos sinais do trading e backtesting

3.1. Definição

```
import IPython
url = 'https://i1.wp.com/escolatrader.net/wp-content/uploads/2013/08/petr4pond.jpg'
IPython.display.Image(url, width = 500)
```

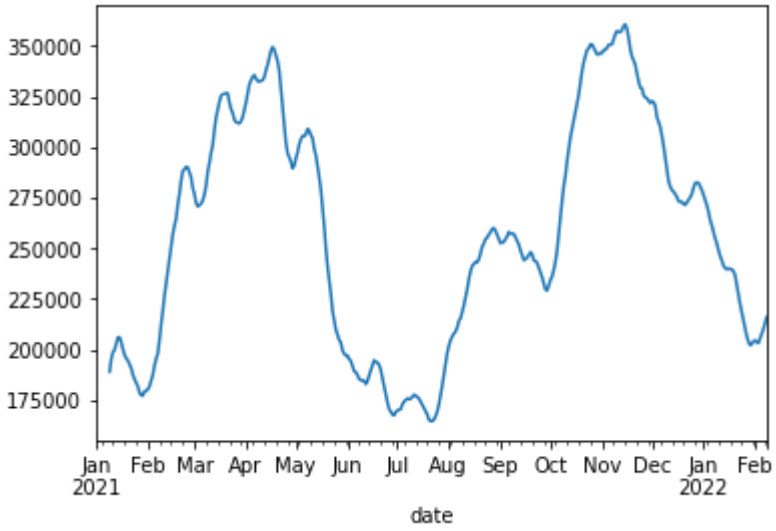

```
# Lembrete: procure parametrizar tudo o que você faz

## Variação mínima do preço || Ações = R$0.01 || WINFUT = 5pts || INDFUT= 1pt

tick_min = 1 # variação minínima para ações no preço
MM_periodo = 9
```

3.2. Construção do sinal de COMPRA

Regra 1: Preço Fech > MM

```
## Cálculo Média Móvel de 9 períodos
MMA = btc_brl['close'].rolling(window=MM_periodo).mean()
```

```
MMA.plot();
```

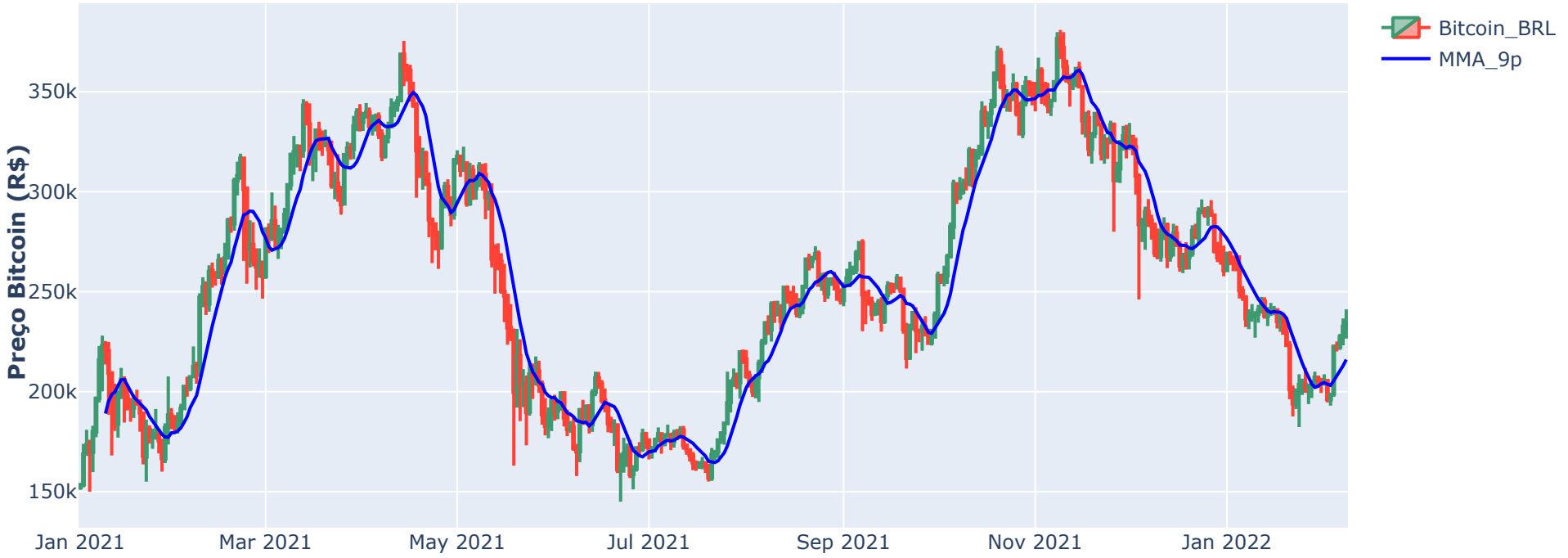



```
btc_brl['MMA'] = MMA
btc_brl.head(20)
```

date	open	high	low	close	retornos	MMA
2021-01-01	152056.0	154167.0	150911.0	153480.0	NaN	NaN
2021-01-02	153478.0	173900.0	152171.0	168784.0	0.099713	NaN
2021-01-03	168885.0	180925.0	167897.0	173503.0	0.027959	NaN
2021-01-04	173504.0	175981.0	150000.0	169820.0	-0.021227	NaN
2021-01-05	169825.0	182966.0	159769.0	179972.0	0.059781	NaN
2021-01-06	179990.0	197555.0	177210.0	196635.0	0.092587	NaN
2021-01-07	196752.0	222813.0	194600.0	215874.0	0.097841	NaN
2021-01-08	216160.0	228045.0	200150.0	222915.0	0.032616	NaN
2021-01-09	223113.0	225333.0	213593.0	220612.0	-0.010331	189066.111111
2021-01-10	220504.0	224900.0	188500.0	209390.0	-0.050868	195278.333333
2021-01-11	209415.0	209822.0	168112.0	199020.0	-0.049525	198637.888889
2021-01-12	199160.0	203999.0	180375.0	184137.0	-0.074781	199819.444444
2021-01-13	183868.0	200600.0	177000.0	198819.0	0.079734	203041.555556
2021-01-14	198827.0	211899.0	196299.0	206134.0	0.036792	205948.444444
2021-01-15	205900.0	207684.0	184563.0	197976.0	-0.039576	206097.444444

```
#GRÁFICO MÉDIA MOVEL
#cria figura
fig = make_subplots(rows=1, cols=1)
#Adiciona o gráfico de candlestick com os parametros OHLC no eixo y e no eixo x o index com as datas
fig.add_trace(go.Candlestick(name='Bitcoin_BRL', x=btc_brl.index, open=btc_brl['open'], high = btc_brl['high'], low=btc_brl['low'], close=btc_b
# cria o gráfico de MMA com o mesmo racional de eixo de cima
fig.add_trace(go.Scatter(name='MMA_9p', x=btc_brl.index, y=btc_brl.MMA, marker_color='blue'),row=1,col=1)
#Personalizar o eixo Y
fig.update_yaxes(title_text="<b> Preço Bitcoin (R$)", row=1, col=1)
#Personalizar aspectos da figura em geral
fig.update_layout(xaxis_rangeslider_visible=False,width=1000 ,height=500)
```

```
#Mostra a figura
fig.show()
```



Regra 2: Preço candle anterior < MMA9p