Cecil Gilliard Crypto Trading bot

This is code to produce a BTC-USD ticker from Coinbase Sanbox API

My Youtube link: https://www.youtube.com/watch?v=JNWjZ6bOAk4

In [4]:
```
import pandas as pd
import cbpro
import time

# I found this cool tutorial on youtube and wanted to incude part of this g
# in order to have a ticker for the user to see realtime price feed of BTC-
# https://www.youtube.com/watch?v=a74pQbHgdXw
class TextWebsocketClient(cbpro.WebsocketClient):
    def on_open(self):
        self.url = 'wss://ws-feed-public.sandbox.pro.coinbase.com'
        self.message_count = 0

    def on_message(self,msg):
        self.message_count += 1
        msg_type = msg.get('type',None)
        if msg_type == 'ticker':
            time_val   = msg.get('time',('-'*10))
            price_val  = msg.get('price',None)
            if price_val is not None:
                price_val = float(price_val)
            product_id = msg.get('product_id',None)

            print(f"{time_val:30} \
                {price_val:.3f} \
                {msg_type}: \
                {product_id}")

    def on_close(self):
        print(f"<---Websocket connection closed--->\n\tTotal messages: {sel
```

In [5]:
```
# used to start the ticker
stream = TextWebsocketClient(products=['BTC-USD'],channels=['ticker'])
stream.start()
```

```
2021-12-06T17:01:21.883966Z                    49400.000
ticker:                    BTC-USD
2021-12-06T17:01:22.788104Z                    49522.010
ticker:                    BTC-USD
2021-12-06T17:01:22.987637Z                    50000.000
ticker:                    BTC-USD
2021-12-06T17:01:23.632466Z                    50000.000
ticker:                    BTC-USD
2021-12-06T17:01:24.044885Z                    50000.000
ticker:                    BTC-USD
2021-12-06T17:01:24.302470Z                    50000.000
ticker:                    BTC-USD
2021-12-06T17:01:24.302470Z                    49600.000
ticker:                    BTC-USD
2021-12-06T17:01:24.488460Z                    57564.000
ticker:                    BTC-USD
2021-12-06T17:01:24.643662Z                    49600.000
ticker:                    BTC-USD
2021-12-06T17:01:24.828967Z                    49624.110
ticker:                    BTC-USD
2021-12-06T17:01:25.027686Z                    57564.000
ticker:                    BTC-USD
2021-12-06T17:01:25.176231Z                    49600.000
ticker:                    BTC-USD
2021-12-06T17:01:25.176231Z                    49400.000
ticker:                    BTC-USD
2021-12-06T17:01:25.176231Z                    48663.630
ticker:                    BTC-USD
2021-12-06T17:01:25.434136Z                    48642.830
ticker:                    BTC-USD
2021-12-06T17:01:25.439923Z                    48224.700
ticker:                    BTC-USD
2021-12-06T17:01:25.475387Z                    48224.700
ticker:                    BTC-USD
2021-12-06T17:01:25.475387Z                    49625.530
ticker:                    BTC-USD
2021-12-06T17:01:25.696707Z                    49628.700
ticker:                    BTC-USD
2021-12-06T17:01:25.757313Z                    57564.000
ticker:                    BTC-USD
2021-12-06T17:01:25.783952Z                    57564.000
ticker:                    BTC-USD
2021-12-06T17:01:25.973210Z                    48669.840
ticker:                    BTC-USD
2021-12-06T17:01:25.973210Z                    48100.000
ticker:                    BTC-USD
2021-12-06T17:01:25.976405Z                    48100.000
ticker:                    BTC-USD
2021-12-06T17:01:25.976405Z                    47900.000
ticker:                    BTC-USD
2021-12-06T17:01:26.219137Z                    57564.000
ticker:                    BTC-USD
2021-12-06T17:01:26.261605Z                    57564.000
```

```
ticker:                  BTC-USD
2021-12-06T17:01:26.296278Z                    57564.000
ticker:                  BTC-USD
2021-12-06T17:01:26.405882Z                    48648.490
ticker:                  BTC-USD
2021-12-06T17:01:26.405882Z                    48300.000
ticker:                  BTC-USD
2021-12-06T17:01:26.467153Z                    48300.000
ticker:                  BTC-USD
2021-12-06T17:01:26.467153Z                    47900.000
ticker:                  BTC-USD
2021-12-06T17:01:26.531377Z                    47900.000
ticker:                  BTC-USD
2021-12-06T17:01:26.587076Z                    57564.000
ticker:                  BTC-USD
2021-12-06T17:01:26.651762Z                    49603.440
ticker:                  BTC-USD
2021-12-06T17:01:26.714083Z                    48621.190
ticker:                  BTC-USD
2021-12-06T17:01:26.720513Z                    57564.000
ticker:                  BTC-USD
2021-12-06T17:01:26.772745Z                    57564.000
ticker:                  BTC-USD
2021-12-06T17:01:26.862098Z                    47900.000
ticker:                  BTC-USD
```

In [6]:
```
# used to stop the ticker
stream.close()
```

```
2021-12-06T17:01:26.862098Z                    47800.000
ticker:                  BTC-USD
2021-12-06T17:01:27.010472Z                    48644.610
ticker:                  BTC-USD
<---Websocket connection closed--->
        Total messages: 42
```

This is a simple program used for buying and selling BTC manually through the Coinbase Sandbox API

In [7]:
```
import cbpro
import time
#  created my api key and saved secret, key, and passphrase on seperate fil
from cb_authy import (api_secret,
                      api_key,
                      api_pass)

# Create authenticatedClient instance
# this allows for buys/sells deposits/withdrawels from account
# https://pypi.org/project/cbpro/

url='https://api-public.sandbox.pro.coinbase.com'

client = cbpro.AuthenticatedClient(
    api_key,
    api_secret,
    api_pass,
    api_url=url
)
pub_client = cbpro.PublicClient()

# the method one would use to deposit into account
# I've already deposited so commented out
# payment_methods = client.get_payment_methods()

# for method in payment_methods:
#     currency  = method.get('currency',None)
#     if currency.upper() == 'USD':
#         method_id = method.get('id',None)
#     elif currency is None:
#         continue

# client.deposit(
#     amount=10000,
#     currency=currency,
#     payment_method_id=method_id
# )


# set spot price to variable
price = float(client.get_product_ticker(product_id='BTC-USD')['price'])

buy_sell = input("Would you like to buy or sell BTC? ").lower()


if (buy_sell == "buy"):
    print(f'The current price of BTC-USD is {price} USD.')
    buy_price = input("At what price in USD would you like to buy BTC? ")
    if (float(buy_price) >= price):
        buy_amt = input("How much BTC would you like to purchase? 'USD': ")
        print("Buying BTC!!!")
        print(client.buy(funds=buy_amt, order_type="market", product_id="BT
    elif (float(buy_price) <= price):
        buy_sze = input("How much BTC would you like to purchase? 'BTC': ")
        print("Setting limit order!!")
        print(client.buy(price=buy_price, size=buy_sze, order_type="limit",
```

```
elif (buy_sell == "sell"):
    print(f'The current price of BTC-USD is {price} USD.')
    sell_price = input("At what price in USD would you like to sell BTC? ")
    if (float(sell_price) <= price):
        sell_amt = input("How much BTC would you like to sell? USD: ")
        print("Selling BTC!!!")
        print(client.sell(funds=sell_amt, order_type="market", product_id="
    elif (float(sell_price) >= price):
        sell_sze = input("How much BTC would you like to sell? 'BTC': ")
        print("Setting limit order!!")
        print(client.sell(price=buy_price, size=sell_sze, order_type="limit

else:
    print("Error: Must select 'buy or sell'.")


cancel_orders = input("Would you like to cancel all BTC-USD limit orders? '


if (cancel_orders == "Y"):
    u_sure = input("Are you sure you would like to cancel all BTC-USD limit
    if (u_sure == 'Y'):
        print(client.cancel_all(product_id="BTC-USD"))
        print("All orders cancelled!!")
else:
    pass
```

```
Would you like to buy or sell BTC?buy
The current price of BTC-USD is 49741.2 USD.
At what price in USD would you like to buy BTC? 49741.2
How much BTC would you like to purchase? 'USD': 100
Buying BTC!!!
{'id': '534d9ea9-8011-4318-8a59-89a3527af864', 'product_id': 'BTC-USD',
'side': 'buy', 'stp': 'dc', 'funds': '99.50248756', 'specified_funds': '1
00', 'type': 'market', 'post_only': False, 'created_at': '2021-12-06T17:0
2:54.075147Z', 'fill_fees': '0', 'filled_size': '0', 'executed_value':
'0', 'status': 'pending', 'settled': False}
Would you like to cancel all BTC-USD limit orders? 'Y/N'N
```

Trading bot for BTC-USD using the Coinbase Sandbox API

In [1]:
```
import cbpro
import time
#  created my api key and saved secret, key, and passphrase on seperate fil
from cb_authy import (api_secret,
                      api_key,
                      api_pass)

# Create authenticatedClient instance
# this allows for buys/sells deposits/withdrawels from account
# https://pypi.org/project/cbpro/

url='https://api-public.sandbox.pro.coinbase.com'

client = cbpro.AuthenticatedClient(
    api_key,
    api_secret,
    api_pass,
    api_url=url
)
pub_client = cbpro.PublicClient()
# set spot price to variable
price = float(client.get_product_ticker(product_id='BTC-USD')['price'])
# get the opening price of BTC
stats = client.get_product_24hr_stats('BTC-USD')
open_price = float(stats['open'])

# function to decide percentage change between two values
# https://stackoverflow.com/questions/30926840/how-to-check-change-between-
def percentage_change(current, previous):
    """returns percent difference between 2 numbers"""
    if previous != 0 :
        return round(float(current - previous) / abs(previous) * 100, 2)
    else:
        return "undefined"

buy_amt = 100
sell_amt = 500

# bot to buy or sell btc based on percentage change between opening and cur
# obviously this is a horrible strategy; but it shows the endless ways a pr
# implement strategies to buy/sell cryptocurrency without the user's manual
while (True):
    # aribitrary percentenge change numbers, can make whatever you'd like
    if percentage_change(price, open_price) >= -2:
        print("BUY BUY BUY!!!")
        # actual code to buy
#         print(client.buy(funds=buy_amt, order_type="market", product_id="
        # i have the breaks in here just to stop the infinite loop
#         break
    elif percentage_change(price, open_price) >= 2:
        print("SELL SELL SELL!!!")
        # actual code to sell
#         print(client.sell(funds=sell_amt, order_type="market", product_id
#         break
    else:
        print("C'mon, do something BTC!!!")
```

```
#         break
    # timer to set bot to trading timeframe in seconds
    time.sleep(5)

# print(open_price)
# print(price)
```

```
C'mon, do something BTC!!!
C'mon, do something BTC!!!
C'mon, do something BTC!!!

---------------------------------------------------------------------------
--
KeyboardInterrupt                         Traceback (most recent call las
t)
<ipython-input-1-700f973e445b> in <module>
     57 #         break
     58     # timer to set bot to trading timeframe in seconds
---> 59     time.sleep(5)
     60
     61 # print(open_price)

KeyboardInterrupt:
```

In [ ]: