Group Project : Phase 2

심명보, 박주얼

코드를 만들며 교수님의 모든 함수를 이해하기 어려워서 book-delta, book-imbalance의 정의에 맞게 버전 2가지를 만들어 피쳐를 구했습니다(v1은 units_traded로 구함, v2는 price*units_traded로 구함). Mid price 이외에도 mean price도 현재 시장참여자의 매매 의향을 나타낸다고 생각하여 피쳐에 추가했습니다.

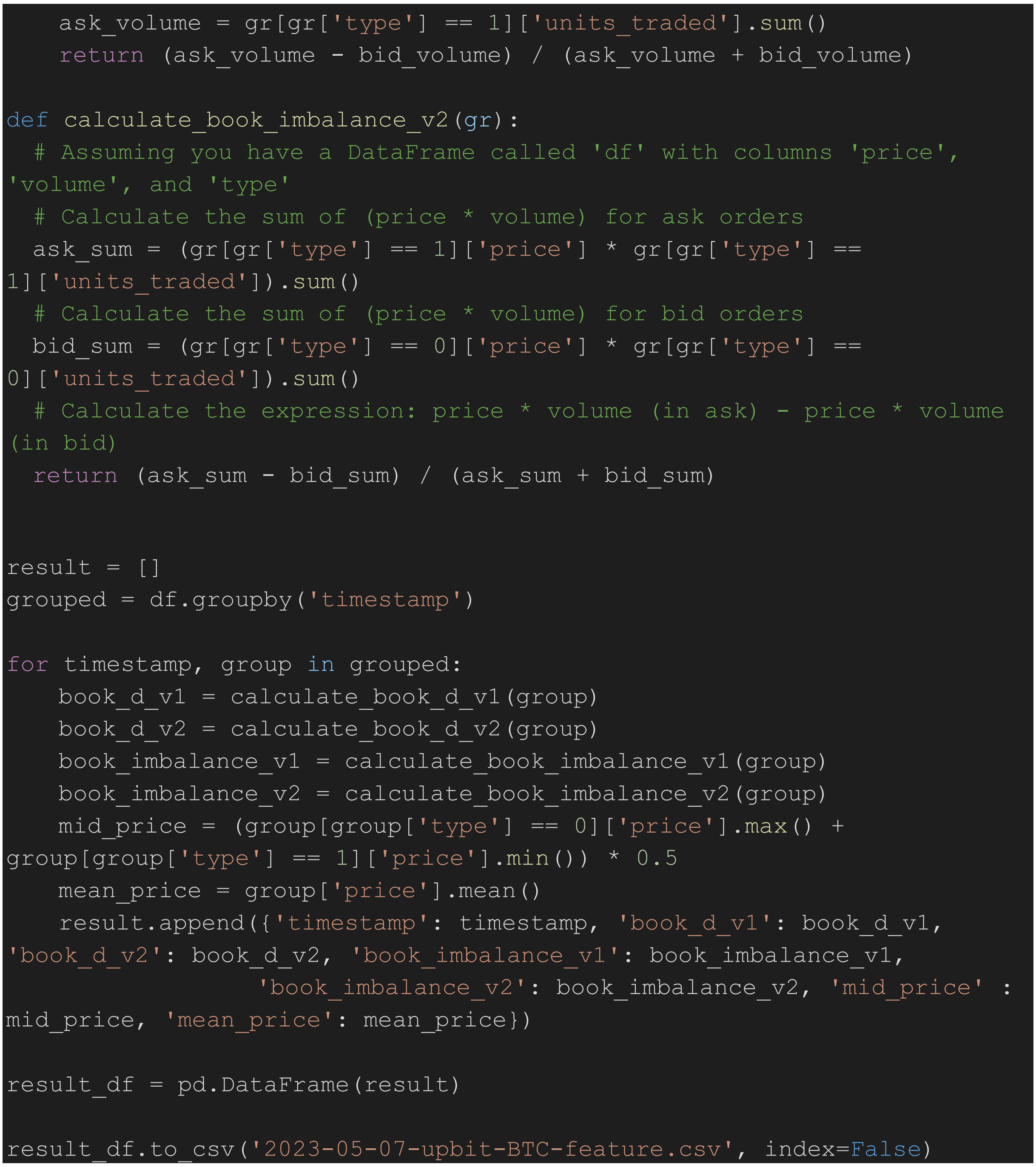
```
import requests
import datetime
import pandas as pd
import time
from google.colab import files
import itertools
import numpy as np

# 오더북 데이터 파일 읽어오기
# 데이터 이름 size -> units_traded, time -> timestamp
uploaded = files.upload()
df = pd.read_csv('2023-05-07-upbit-btc-
orderbook.csv').apply(pd.to_numeric,errors='ignore')

def calculate_book_d_v1(gr):
    bid_volume = gr[gr['type'] == 0]['units_traded'].sum()
    ask_volume = gr[gr['type'] == 1]['units_traded'].sum()
    return ask_volume - bid_volume

def calculate_book_d_v2(gr):
  # Assuming you have a DataFrame called 'df' with columns 'price',
'volume', and 'type'
  # Calculate the sum of (price * volume) for ask orders
  ask_sum = (gr[gr['type'] == 1]['price'] * gr[gr['type'] ==
1]['units_traded']).sum()
  # Calculate the sum of (price * volume) for bid orders
  bid_sum = (gr[gr['type'] == 0]['price'] * gr[gr['type'] ==
0]['units_traded']).sum()
  # Calculate the expression: price * volume (in ask) - price * volume
(in bid)
  return ask_sum - bid_sum

def calculate_book_imbalance_v1(gr):
    bid_volume = gr[gr['type'] == 0]['units_traded'].sum()
```



```
    ask_volume = gr[gr['type'] == 1]['units_traded'].sum()
    return (ask_volume - bid_volume) / (ask_volume + bid_volume)

def calculate_book_imbalance_v2(gr):
  # Assuming you have a DataFrame called 'df' with columns 'price',
'volume', and 'type'
  # Calculate the sum of (price * volume) for ask orders
  ask_sum = (gr[gr['type'] == 1]['price'] * gr[gr['type'] ==
1]['units_traded']).sum()
  # Calculate the sum of (price * volume) for bid orders
  bid_sum = (gr[gr['type'] == 0]['price'] * gr[gr['type'] ==
0]['units_traded']).sum()
  # Calculate the expression: price * volume (in ask) - price * volume
(in bid)
  return (ask_sum - bid_sum) / (ask_sum + bid_sum)


result = []
grouped = df.groupby('timestamp')

for timestamp, group in grouped:
    book_d_v1 = calculate_book_d_v1(group)
    book_d_v2 = calculate_book_d_v2(group)
    book_imbalance_v1 = calculate_book_imbalance_v1(group)
    book_imbalance_v2 = calculate_book_imbalance_v2(group)
    mid_price = (group[group['type'] == 0]['price'].max() +
group[group['type'] == 1]['price'].min()) * 0.5
    mean_price = group['price'].mean()
    result.append({'timestamp': timestamp, 'book_d_v1': book_d_v1,
'book_d_v2': book_d_v2, 'book_imbalance_v1': book_imbalance_v1,
                   'book_imbalance_v2': book_imbalance_v2, 'mid_price' :
mid_price, 'mean_price': mean_price})

result_df = pd.DataFrame(result)

result_df.to_csv('2023-05-07-upbit-BTC-feature.csv', index=False)
```