PHENOMENOLOGY AND MODELLING OF FINANCIAL MARKETS

Petite Classe n. 3

Propagator Models

ECO 586 - J. Moran
M. Benzaquen

January 23, 2020

Introduction

In this exercise session we will work with real trade data on the E-mini S&P 500 futures contract.

Part 1 : Preliminaries and data cleaning

1. To begin, we are going to do very basic data cleaning : download the `spmini_trades.csv` file and put it in the root folder of your project (or adapt the cell below), run the following cells and read the explanations (also available in the notebook) :
   — We first load the csv file, putting appropriate names on the columns. `T_rec` contains the timestamp at which the information was received, `T_market` contains the timestamp corresponding to the time of execution at the exchange, `evt_seq_num` is a sequence of numbers used to know which order came first in case they have the same timestamp, `price` is the price of execution, `qty` is the volume, `side` is bid or ask, and `type` only concerns us when its value is `I`, meaning invalid.
   — We drop columns we're not going to use here, `T_rec` and `evt_seq_num`.
   — We drop invalid trades, and drop rows for which no side (bid or ask) is given.
   — We convert the `T_market` column into a datetime column, then transform the timezone from UTC+1 (Paris) to US/Eastern, corresponding to New York City.
   — Next, we want to check how the volume of trades evolves during the day. We do this by first grouping all the data by hour (between 0 and 23) and minutes (between 0 and 60), with the latter being divided by 5 to get 5 minute bins. We then compute the sum of the quantity traded in that 5-minute bin.
   — The `get_timestamp` function is artificial, and is only there to provide an artificial `DateTime` index with the right hour of day, but on 1st of Jan 2017 (for plotting purposes).
   — Last, we add a time column that we set as the index, drop the bin information and rename `qty` into `traded_volume`
2. After the cleaning procedure, we want to check how the volume of trades evolves during the day. We do this by first grouping all the data by hour (between 0 and 23) and minutes (between 0 and 60), with the latter being divided by 5 to get 5 minute bins. We then compute the sum of the quantity traded in that 5-minute bin, and plot it throughout the day. What do you see ?

Part 2 : Get your hands dirty ! Mid prices, signs and returns Since the E-mini is a very liquid asset, in absence of information on the mid-price we can define the mid-price as

$$m_t = p_t \pm \delta/2, \tag{1}$$

where p_t is the price of the current transaction, $\delta = 0.25$ is the tick-size and we take $+$ for bid and $-$ for ask.

1. Define a column `df['mid']` containing the mid-price using this. (Hint : use `np.where()`)

2. Define a column `df['r']` containing the returns,

$$r_t = m_{t+1} - m_t. \tag{2}$$

3. We expect to see an U-shape for the volatility throughout the day. Adapt the code given in the notebook, where the volume was binned in 5-minutes, to compute the volatility throughout the day. The 5-minute volatility is defined as the standard deviation of the returns inside a 5-minute bin. Plot the evolution of the volatility throughout the day.
4. All of this motivates us to only work with data between 10 a.m. and 3 p.m. US/Eastern. Call the two following lines to do the trick by first setting `T_market` as the index :

```
df.set_index('T_market', inplace=True)
df = df.between_time('10:00','15:00').copy()
```

5. You still need to fix the returns, if $r_t = m_{t+1} - m_t$ then the last value of each day doesn't make sense. Use `np.where()` to fill `df['r']` with an `np.nan` (Not a Number, which Pandas will ignore), whenever the day for time t is not the same that the one at $t+1$, or else leaves the value unoutched. (Hint : `df.index.dt.day` accesses the value of the current day).
6. Next you need to define the sign of a transaction : create a column `df['sign']` equal to $\varepsilon_t = +1$ for an ask order, and to $\varepsilon_t = -1$ for a bid order.

Part 3 : Sign correlation function and variogram For all of these questions, the Python functions to compute correlators and variograms are given.

1. Define the sign correlation function,

$$C(l) = \langle \varepsilon_t \varepsilon_{t+l} \rangle \tag{3}$$

where the average is first computed for each day (use `df.groupby(df.index.day).apply(function)` for a suitably defined `function`) in Jan. 2018 for $0 \leq l < 2000$, obtaining thus an array of size 2000 per day, and then averaged over all days to obtain a single array. Plot it in log-log scale. What do you notice ?

2. Next, compute the price variogram

$$V(\tau) = \langle (m_{t+\tau} - m_t)^2 \rangle_c, \tag{4}$$

again computing first for each day and then averaging over all days in January. Compute for $0 \leq \tau < 2000$ and plot in log-log scale, then plot the volatility signature plot $V(\tau)/\tau$.

3. How can you interpret previous figures ?

Part 4 : The propagator model We recall the propagator model, where the return dynamics are given by

$$r_t = \sum_{t' \leq t} \mathscr{G}(t-t')\varepsilon_{t'} + \eta_t, \tag{5}$$

where η_t is a white noise of 0 mean and variance $\langle \eta_t \eta_{t'} \rangle = 2\sigma_0^2 \delta(t-t')$ and where $\mathscr{G}(t) = G(t+1) - G(t)$ is the discrete derivative of the propagator. In this case, the mid price reads

$$m_t = m_0 + \sum_{t'<t} G(t-t')\varepsilon_{t'} + \sum_{t'<t} \eta_{t'}. \tag{6}$$

1. We are interested first in the response function of the returns to sign fluctuations, namely

$$S(l) := \langle r_{t+l} \varepsilon_t \rangle, \tag{7}$$

show that it verifies

$$S(l) = \sum_{n \geq 0} \mathscr{G}(n) C(n-l) \tag{8}$$

(Note : the average must be understood to run over t).

2. Compute the function S for $0 \leq l < 2000$, first over each day and then averaging over the whole month. Plot it with a log-scale in y.
3. Another correlation function with a "physical meaning" is the function R defined by

$$R(l) := \langle (m_{t+l} - m_t) \cdot \varepsilon_t \rangle. \tag{9}$$

   Show that

$$R(l) = \sum_{0 \leq i < l} S(i), \tag{10}$$

   and compute $R(l)$, using first its definition from S and then empirically for $0 \leq l < 5000$.
4. Solve for $\mathscr{G}$ using eq. (8). (Hint : define a matrix $A_{ln} = C(l-n)$ minding the fact that C is symmetric). Compute G from $\mathscr{G}$ and plot it. Comments ?