

# Quantitative Development проект для Центра Математических Финансов

Часов Никита

## 1 Введение

В данном документе представлено подробное объяснение двух функций: `create_candles` и `simulator`, предназначенных для обработки торговых данных и симуляции торговли на основе заданных стратегий.

## 2 Функция: `create_candles`

### 2.1 Описание

Функция `create_candles` предназначена для создания свечей (candlestick data) на основе данных о сделках. Свечи представляют собой агрегированные данные за определённые временные интервалы.

### 2.2 Параметры

Функция принимает следующие параметры:

- `T` — длина временного окна в миллисекундах. По умолчанию 10.
- `trades` — `DataFrame`, содержащий данные о сделках.
- `day`, `hour`, `min`, `sec` — булевы параметры, позволяющие задавать временное окно в днях, часах, минутах или секундах соответственно.

### 2.3 Работа функции

- В начале функции определяются константы, соответствующие количеству миллисекунд в различных единицах времени:
  - `ms_per_day` — 86400000 (количество миллисекунд в дне).
  - `ms_per_hour` — 3600000 (количество миллисекунд в часе).
  - `ms_per_min` — 60000 (количество миллисекунд в минуте).
  - `ms_per_sec` — 1000 (количество миллисекунд в секунде).

- Если параметры `day`, `hour`, `min` или `sec` установлены в `True`, значение `T` пересчитывается в миллисекунды.
- Далее функция находит минимальное время из столбца `local_timestamp` и создает новую колонку `group`, которая позволяет группировать сделки по временным окнам. Этот процесс выполняется с помощью выражения:
 
$$\text{trades}["group"] = (\text{trades}["local\_timestamp"] - \text{min\_time}) // (1000 \times T)$$
- Затем данные группируются по колонке `group` и для каждой группы рассчитываются следующие параметры:
  - `open` — первая цена в группе.
  - `high` — максимальная цена в группе.
  - `low` — минимальная цена в группе.
  - `close` — последняя цена в группе.
- Далее, отдельно для покупателей и продавцов, рассчитываются средние цены и объемы сделок. Это достигается с помощью дополнительных группировок:
  - Для покупателей — расчет средней цены покупки и общего объема.
  - Для продавцов — расчет средней цены продажи и общего объема.
- Наконец, значения `NaN` в результирующем `DataFrame` заменяются на `0`, и функция возвращает готовый `DataFrame` со свечами.

## 3 Функция: `simulator`

### 3.1 Описание

Функция `simulator` отвечает за симуляцию торговли на основе свечей и применения различных торговых стратегий.

### 3.2 Параметры

Функция принимает следующие параметры:

- `T` — длина временного окна.
- `candles_dict` — список кортежей, каждый из которых содержит `DataFrame` со свечами и матрицу стратегий.
- `mode` — режим расчета цены (`'close'` или `'mean'`).

### 3.3 Работа функции

- Функция начинается с инициализации переменной `money_market`, представляющей начальный баланс (100 единиц).
- Затем идет цикл по каждому инструменту в `candles_dict`.

Для каждого инструмента:

- Если режим установлен на `'close'`, производится поэлементное умножение матрицы стратегий на цены закрытия. Если режим `'mean'`, производится умножение на среднюю цену покупки и продажи.
- Рассчитываются массивы для статистик, таких как доходность, Sharpe ratio, Sortino ratio и др.
- Используется функция `np.cumsum` для расчета кумулятивного баланса и кумулятивного количества акций, что позволяет отслеживать общий счет на каждом этапе.
- Также рассчитываются показатели, такие как максимальная просадка и среднее время удержания позиции, а также количество переходов через ноль.
- В конце для каждого инструмента создается итоговый `DataFrame` со всеми рассчитанными метриками, который добавляется в общий список `pairs`.