



Анализ МОЕХ Задание №3

Оглавление

1. Данные
2. Архитектура решения
3. Применение для бизнеса
4. Аннотация к скрипту

1. Данные

В нашем задании мы выгружаем исторические данные по активам. Котировки ежедневные. "TRADEDATE" (дата), "OPEN" (цена открытия), "LOW" (мин. цена за день), "HIGH" (макс. цена за день), "CLOSE" (цена закрытия), "VOLUME" (объем сделок)

Данные для запроса

/engines/(trade_engine_name)
– торговая платформа
/markets/(market_name) –
наименование рынка
/boards/(boardid) – класс
актива
/boardgroups/(board_group_id)
– вид актива
/securities/{secid} –
наименование актива

Данные по котировке

"BOARDID", "TRADEDATE",
"SHORTNAME", "SECID",
"NUMTRADES", "VALUE",
"OPEN", "LOW", "HIGH",
"LEGALCLOSEPRICE",
"WAPRICE", "CLOSE",
"VOLUME",
"MARKETPRICE2",
"MARKETPRICE3",
"ADMITTEDQUOTE",
"MP2VALTRD",
"MARKETPRICE3TRADESVALUE",
"ADMITTEDVALUE",
"WAVAL"

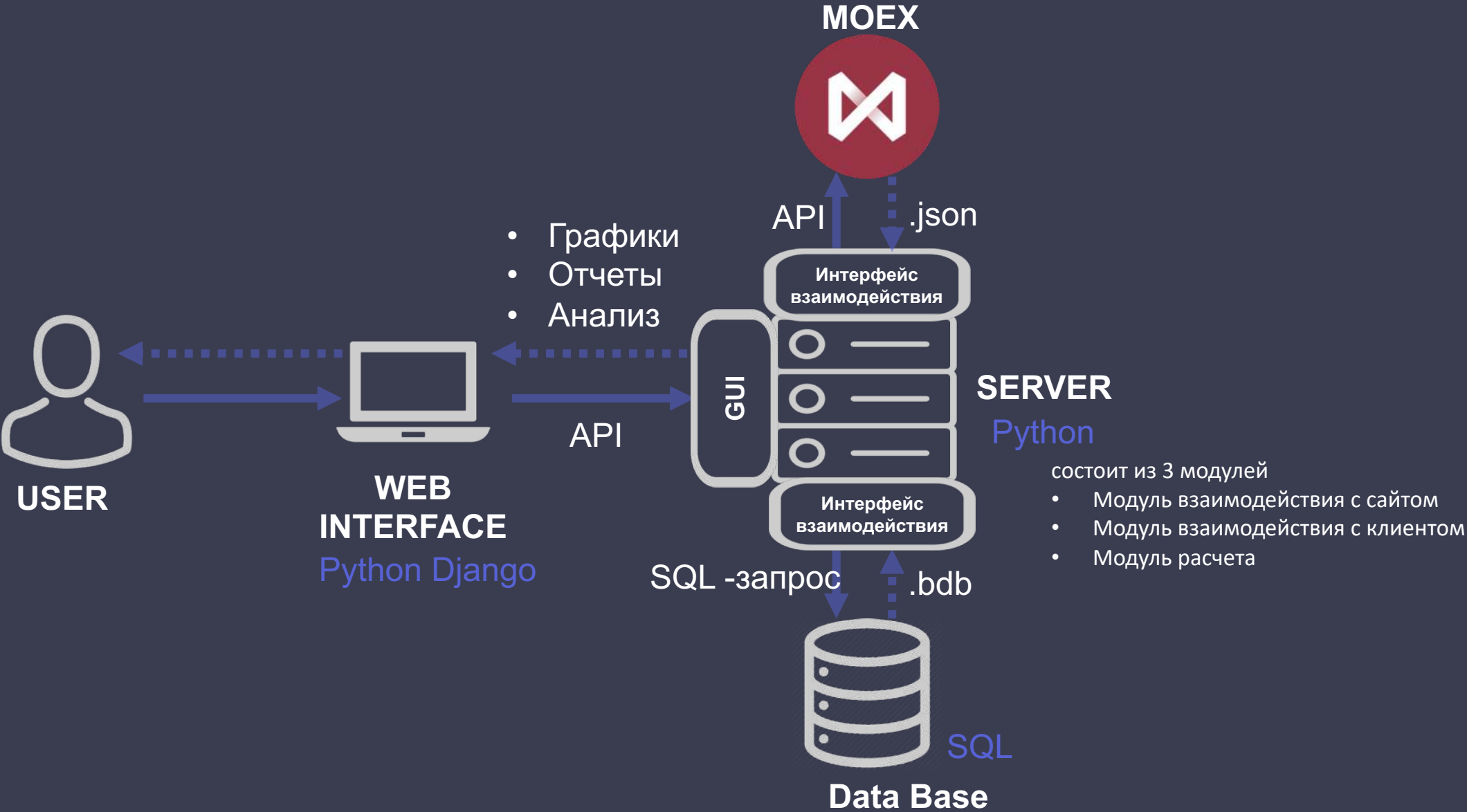
Сущность 1

котировка:
- id (индекс)
- торговая платформа
- наименования рынка
- класс актива
- вид актива
- наименование актива
(ключ)

Сущность 2

показатели котировки:
- наименование актива
(ключ)
- дата (индекс)
- цена открытия
- цена закрытия
- мин цена за день
- макс цена за день

Архитектура решения*



WEB INTERFACE отсутствует, но предполагается в финальном варианте решения

Предложения по автоматизации

Предположим, что наш пользователь это либо исследователь, либо инвестор фондового рынка

1. Оценка рынка

Перед использование функционала сервиса, дать пользователю короткий обзор на фондовый рынок и основных показателей стран (макро показатели, товарный рынок, облигации, акции, драг. металлы) и дать рекомендацию относительно потенциальных действий

3. Рекомендательная система по метрикам

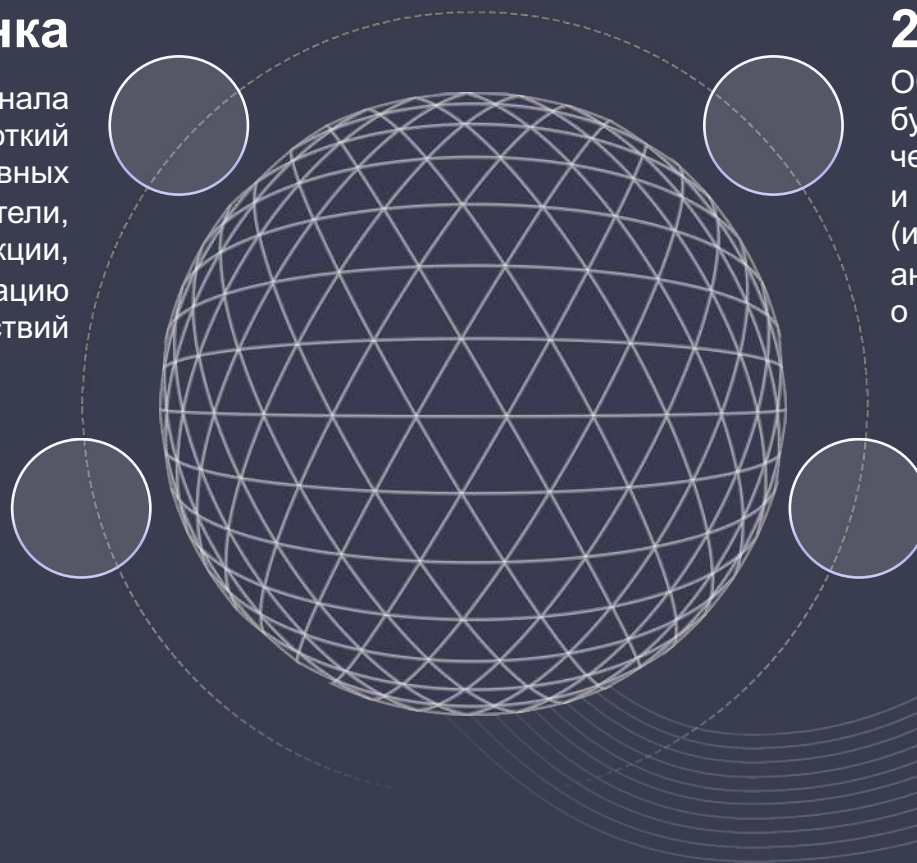
Сделать систему рекомендацию по построению графиков или выдачи метрик в зависимости от целей пользователя. исходя из а) истории использования экспертов, б) особенностей временных рядов котировок. в) количества имеющейся информации по бумаге

2. Актуализация информации

Обновление информации о доступных бумагах и котировках на московской бирже через использование официальных ресурсов и запросов api. Сохранение данных о всех (исключенных, не исключённых для исторического анализа). Предоставление комментария о недавнем включении или исключении компаний

4. Автоматизированная интерпретация

Основываясь на расчетах параметров и исходных данных, выделить критические точки, важные корреляции или пересечение исх. данных с расчетами. Предоставить пользователю интерпретацию на основе проведенного анализа



BPMN 2.0 process

Комментарий:

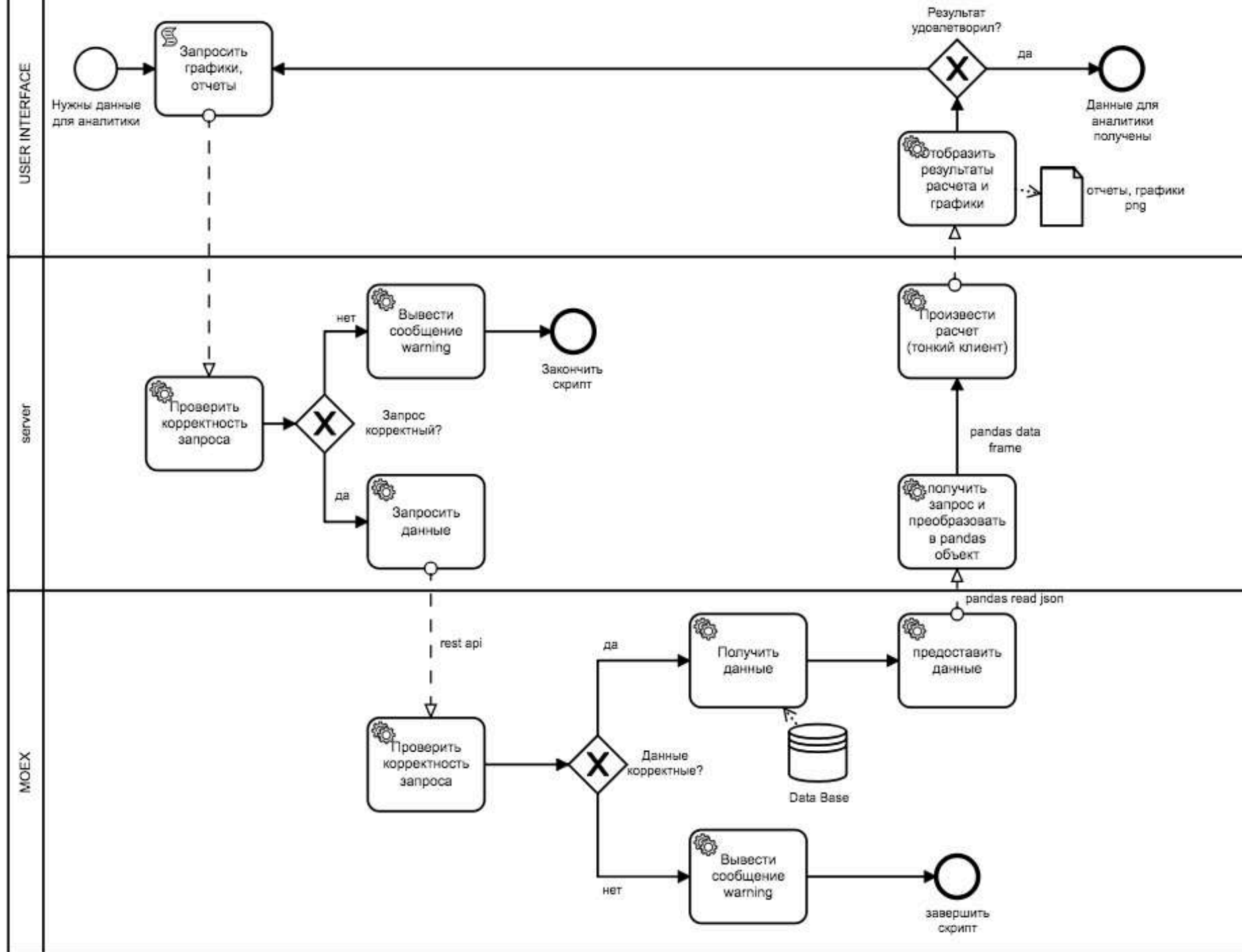
Начинается процесс с потребности пользователя получить данные для дальнейшей аналитики

Таски, кроме запроса выполняются удаленно

Сторона MOEX является не нашей стороной исполнения, поэтому представление их процесса было упрощено

Jupyter обладает **server-client** структурой, поэтому здесь была добавлена дорожка сервера

Конечная цель процесса – получить данные для аналитики



2. Применение для бизнеса

Основные

1. Обработка данных

2. Формирование отчетов

3. Анализ результатов

Остальные

- 1) Построение графиков для дальнейшей интерпретации
- 2) Анализ исторических данных по различным метрикам для оценки активов (ликвидность, недооцененность/переоцененность, сезонность, тренды)
- 3) Построение прогнозов
- 4) Классификация, кластеризация активов для дальнейшего изучения групп активов
- 5) Глубокое обучение для нахождения новых взаимосвязей между активами
- 6) Идентификация фрод-поведения на рынке и спекуляций по поведению временных рядов

Аннотация к скрипту

Описание функций:

Главное улучшение над всем решением – это построение класса с функциями с различными уровнями скрытости данных.

1) choose_stock_numb()

входные переменные – start (стартовая дата), end (конечная дата), engines (торговая платформа), markets (рынок), boards (вид актива), sec_name (название актива)

Данная функция была создана для возможности скачивания названия бумаги по индексу в массиве. Пользователь вряд ли будет помнить название каждой бумаги. Я изначально загрузил все доступные акции (а именно сокращенные названия бумаг) биржи на конец декабря.

улучшение – определение ресурса, откуда можно было бы получать актуальную информацию о доступных бумагах (security name) на бирже, вместо хранения специального файла

2) get_data_request()

входные переменные – engines (торговая платформа), markets (рынок), boards (вид актива), sec_name (название актива)

Данная функция делает запрос на биржу, получает данные в формате .json, далее распаковывает их, в итоге возвращает pandas объект. Также в этой функции учтено, что запрос может вернуть None, в этом случае выведется строка warning – информации нет.

улучшение – добавить взаимодействие с пользователем и с помощью GUI запрашивать информацию, нажимая на кнопки.

Аннотация к скрипту

3) get_available_dates()

входные переменные – title (название графика), x_lab (название оси икс), y_lab (название оси игрек), x_row (данные для оси икс), y_row (данные для оси игрек), line_names (название линий), SwitchSideOfY=0 (переместить ли ось игрек слева-направо)*

*нужно для корректного одновременного отображения графика объема продаж и скользящей средней.

Данная функция возвращает сразу 2 значения, первое это – с какой даты доступна запрашиваемая бумага, второе – по какую дату. Тут также есть проверка на ошибку, в случае ошибки вернется warning.

улучшение – добавить GUI, кнопки. Сделать отдельную функцию, которая будет делать запрос общего образца, и его можно будет модифицировать в зависимости от характера запроса.

4) plot_any()

входные переменные – title (название графика), x_lab (название оси икс), y_lab (название оси игрек), x_row (данные для оси икс), y_row (данные для оси игрек), line_names (название линий), SwitchSideOfY=0 (переместить ли ось игрек слева-направо)

Данная функция была сделана для построения графика общего типа, т.к. далее по заданию я буду делать это неоднократно. Это облегчило остальные функции по построению отдельных видов графиков.

улучшение – упрощение логики с отображением легенды

5) plot_bar()

входные переменные – x_row (данные для оси икс), y_row (данные для оси игрек)

Функция строит столбчатый график, поэтому его реализацию я выделил отдельно. В нашем задании этот график использовался для отображения объема сделок.

Аннотация к скрипту

6) plot_low_high()

входные переменные – data (данные по котировке формата пандас)

Данная функция строит график минимального и максимального значения за день.

7) plot_open_close()

входные переменные – data (данные по котировке формата пандас)

Данная функция строит график цены открытия и закрытия.

8) plot_mov_av()

входные переменные – data (данные по котировке формата пандас), window (окно), volume (объем)

Данная функция строит график скользящей средней и объема торгов (по запросу).

9) plot_candle()

входные переменные – data (данные по котировке формата пандас)

Данная функция строит график скользящей средней.

10) plot_RSI()

входные переменные – data (данные по котировке формата пандас), n (количество периодов)

Данная функция рассчитывает RSI и рисует график.

11) rescale()

входные переменные – values (значения)

Данная функция пересчитывает массив приводя его к виду от 0 до 100. Данная функция используется в этой работе для изменения массива RSI.

Аннотация к скрипту

12) make_data_for_corr()

входные переменные – name (название котировки)

Данная функция собирает набор котировок, ориентируясь на выбранную бумагу (ее самая ранняя дата доступная котировка с 2013-01-01). Остальные будут набираться с одинаковым стартом. Поскольку корреляционный анализ временных рядов требует, чтобы периоды совпадали. Конец я не обрезал, т.к. ограничение по скачиванию данных по запросу 100.

улучшение – обрезка конца, сопоставление каждой даты у котировок.

13) form_data_for_plot(portf)

входные переменные – portf (условный портфель из бумаг)

Данная функция предназначена для того, чтобы подготовить данные для корреляционного анализа на графике

14) plot_corr(portf_names)

входные переменные – portf_names (условный портфель из бумаг, в которых есть названия)

Финальная функция, которая рисует график множественного корреляционного анализа.