**Требования и рекомендации по установке**

Для выполнения задания использовались следующие библиотеки:

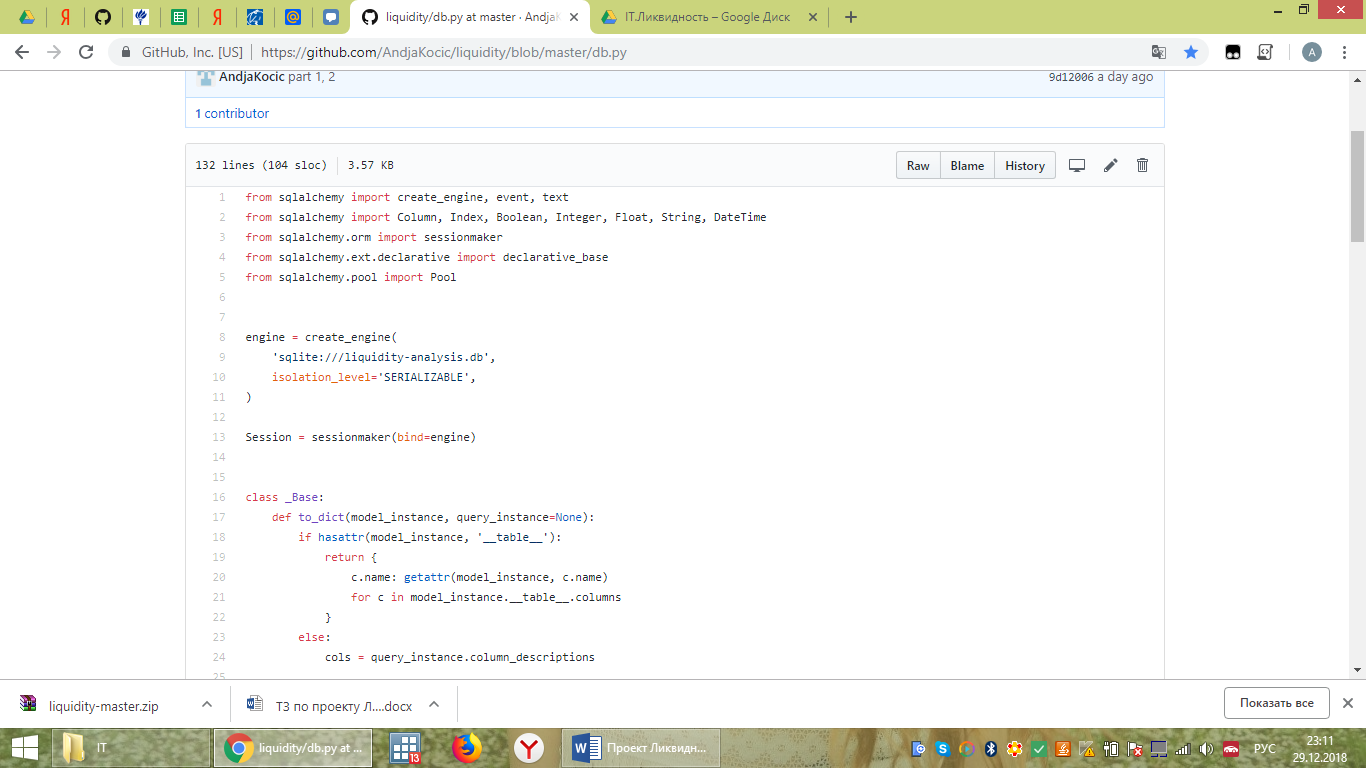
* Pandas (<https://www.youtube.com/watch?v=khwJY8EeJtU> – как установить)
* Matplotlib (<https://www.youtube.com/watch?v=Z6jgBcRiYK4> – рекомендации по установке)
* Sqlalchemy (<https://www.sqlalchemy.org/download.html> -ссылка на скачивание)
* python-dateutil (<http://qaru.site/questions/136233/installing-python-dateutil> - устранение ошибки)

**Блок 1: автоматический импорт в базу данных неструктурированных данных.**

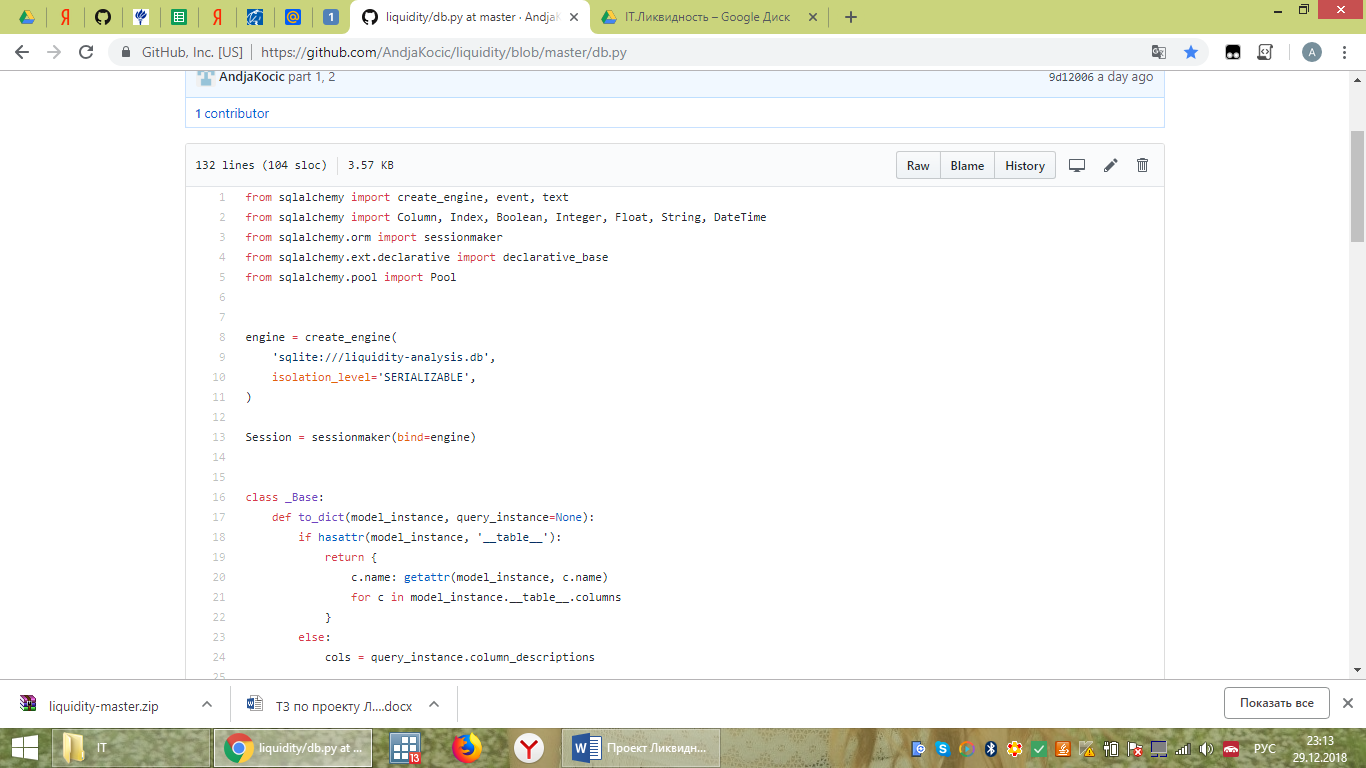
В рамках осуществления данного блока необходимо исполнить команды из файлов db.py и main.py.

Подготовительные процедуры:

Для успешного импорта данных в базу данных используется SQLAlchemy, позволяющее описывать структуры баз данных и способы взаимодействия с ними прямо на языке [Python](https://ru.wikibooks.org/wiki/Python" \o "Python).



Путем использования механизма engine создается база данных liquidity-analysis.db.

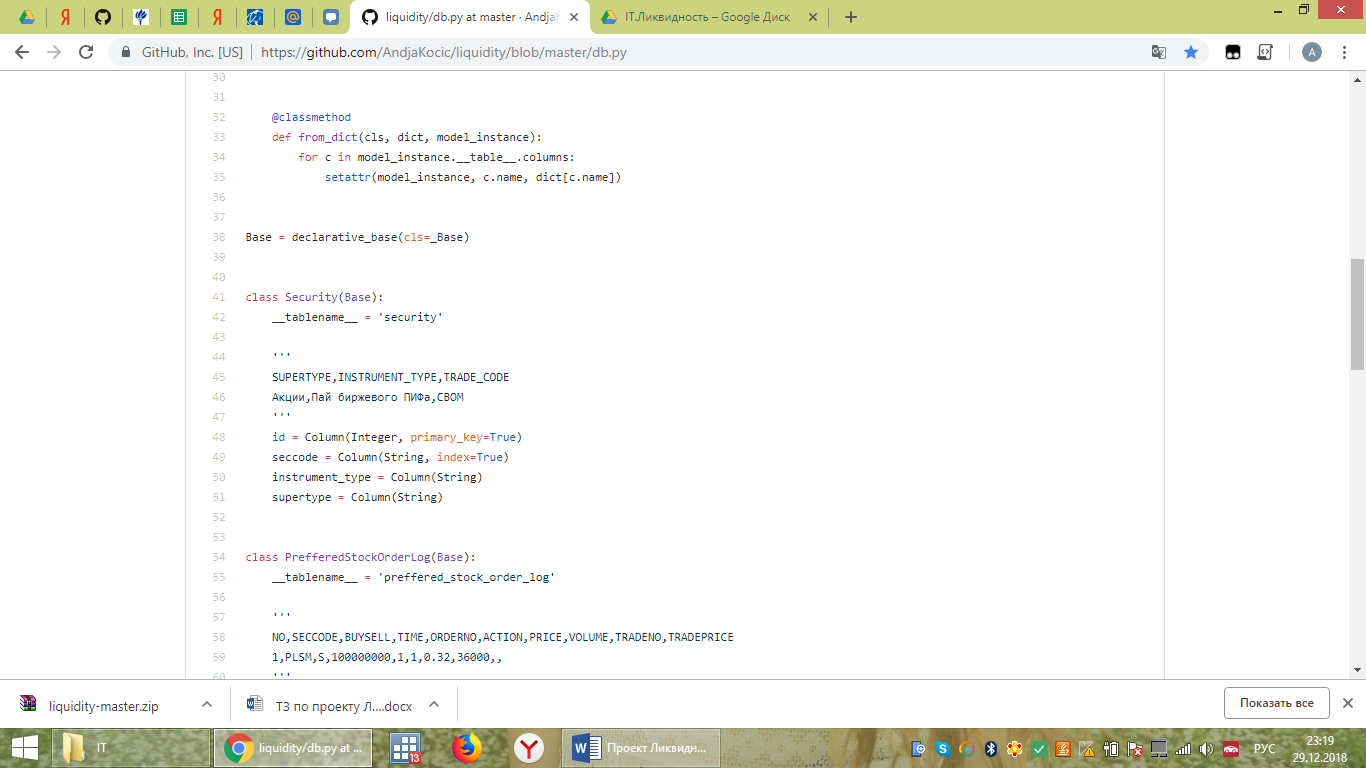


Пользовательские параметры загрузки:

Загрузка данных осуществляется через библиотеку. Используется модуль os.path, с помощью которого прописан путь к файлу классификатора ценных бумаг.

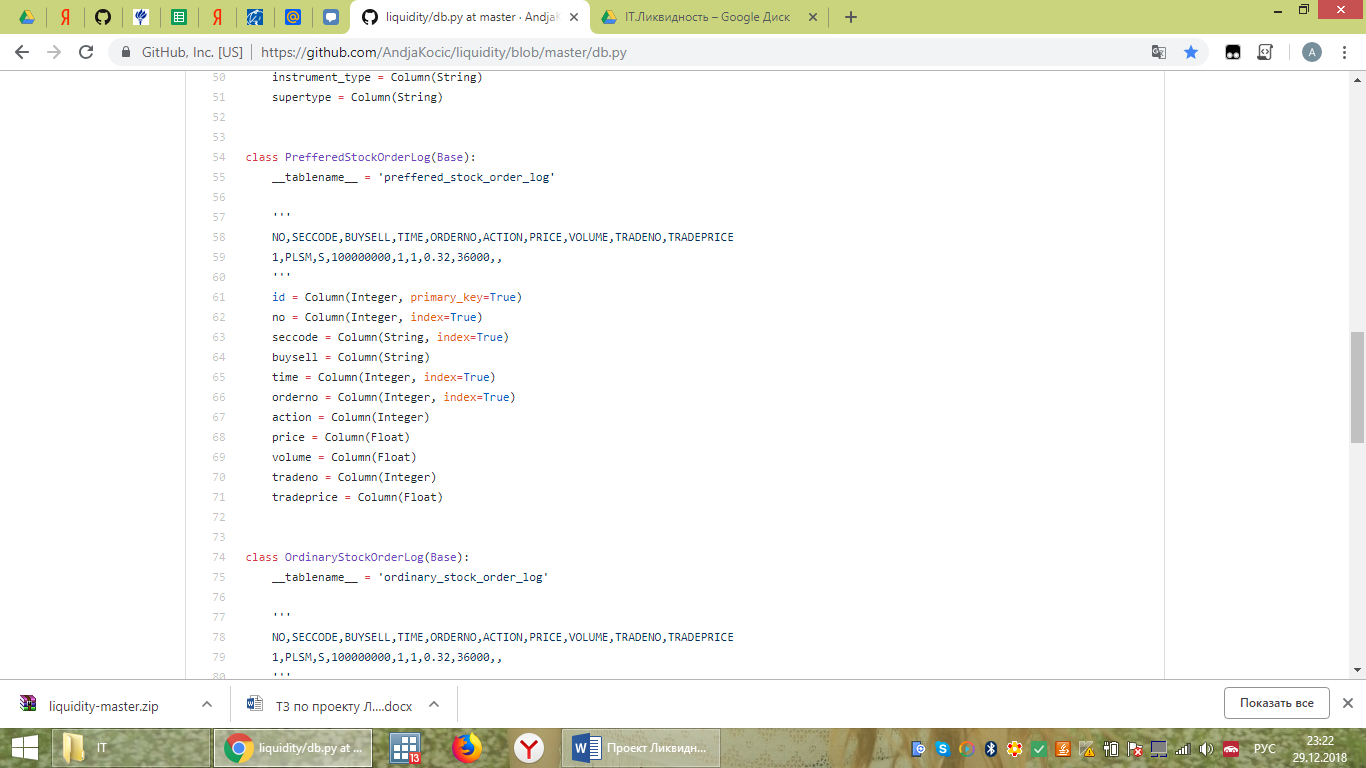
Таблица-классификатор:

Классификатор ценных бумаг по типам строится по классификации с сайта Московской Биржи.

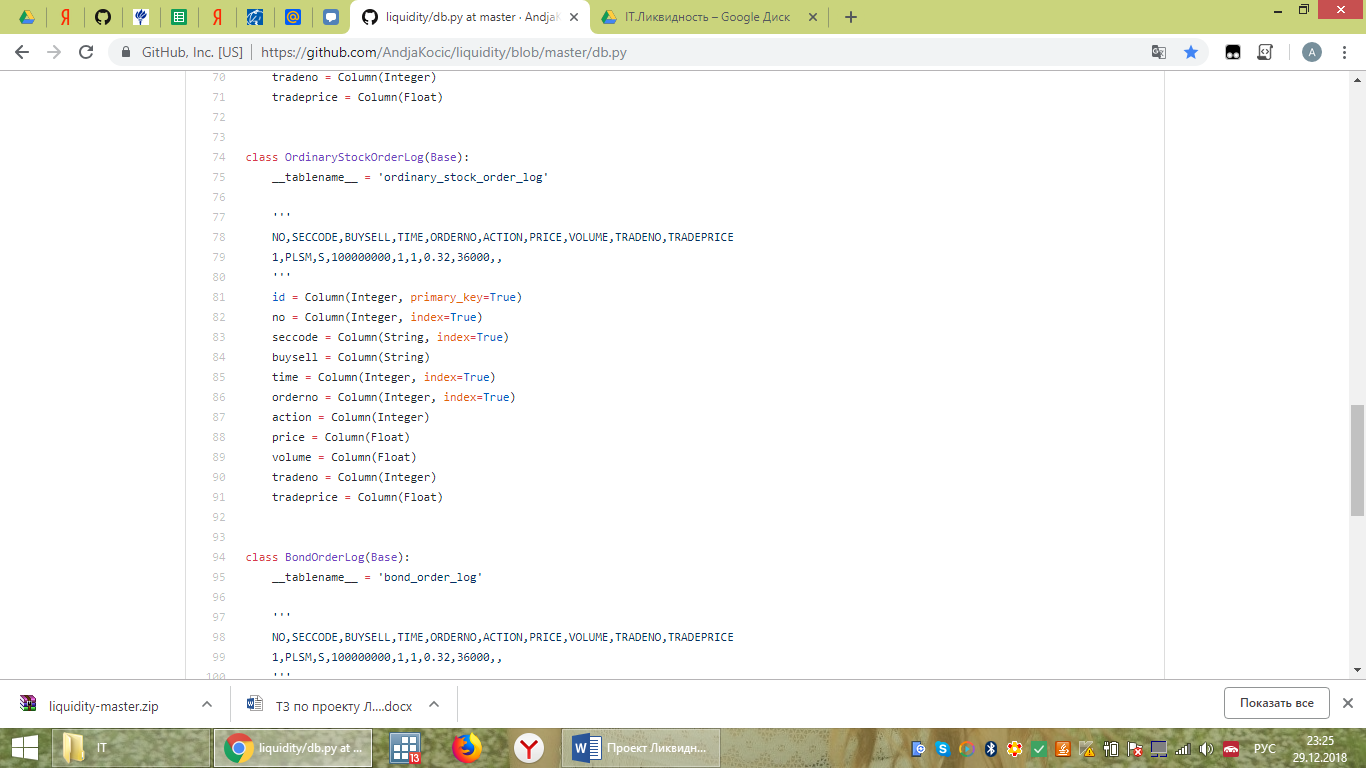


Далее согласно созданной таблице-классификатору создаются три таблицы (по типам ценных бумаг):

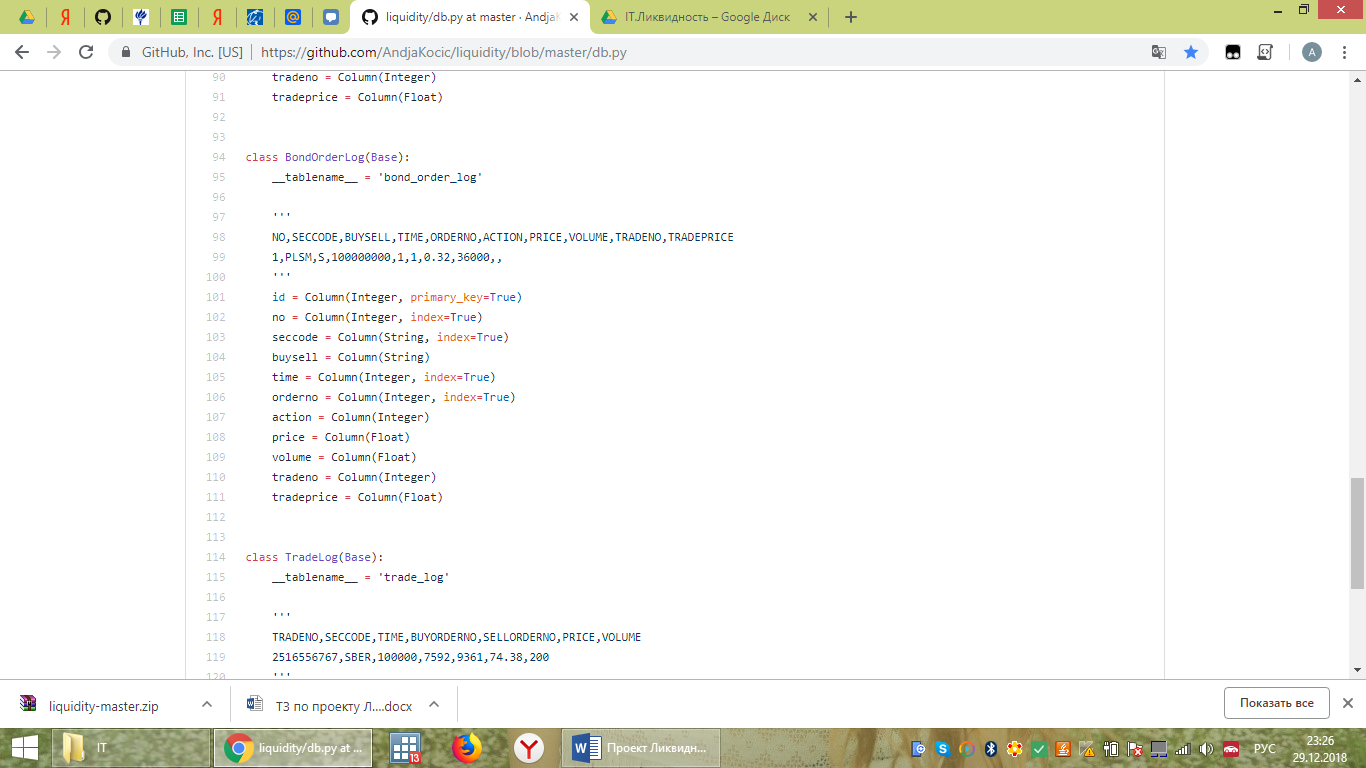
А) Поток заявок по привилегированным акциям:



Б) Поток заявок по обыкновенным акциям:

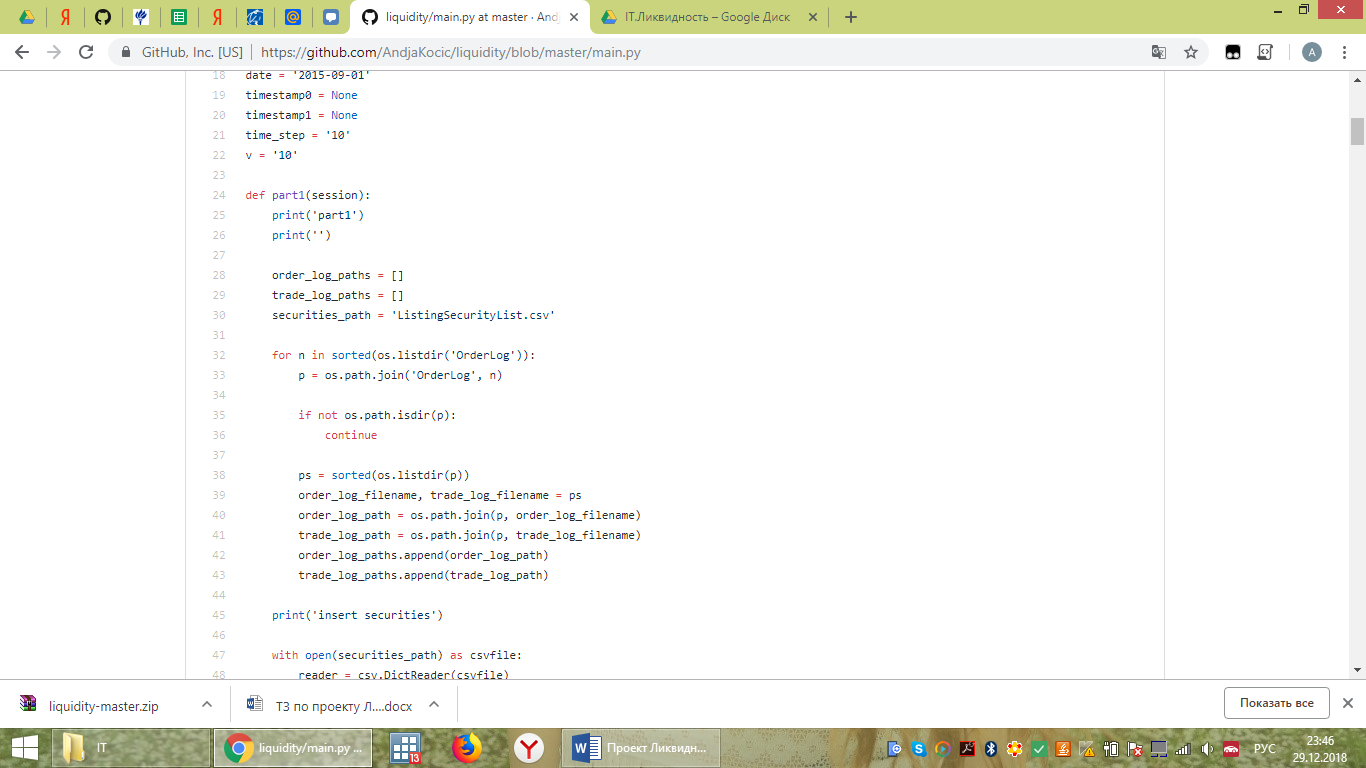


В) Поток заявок по облигациям:

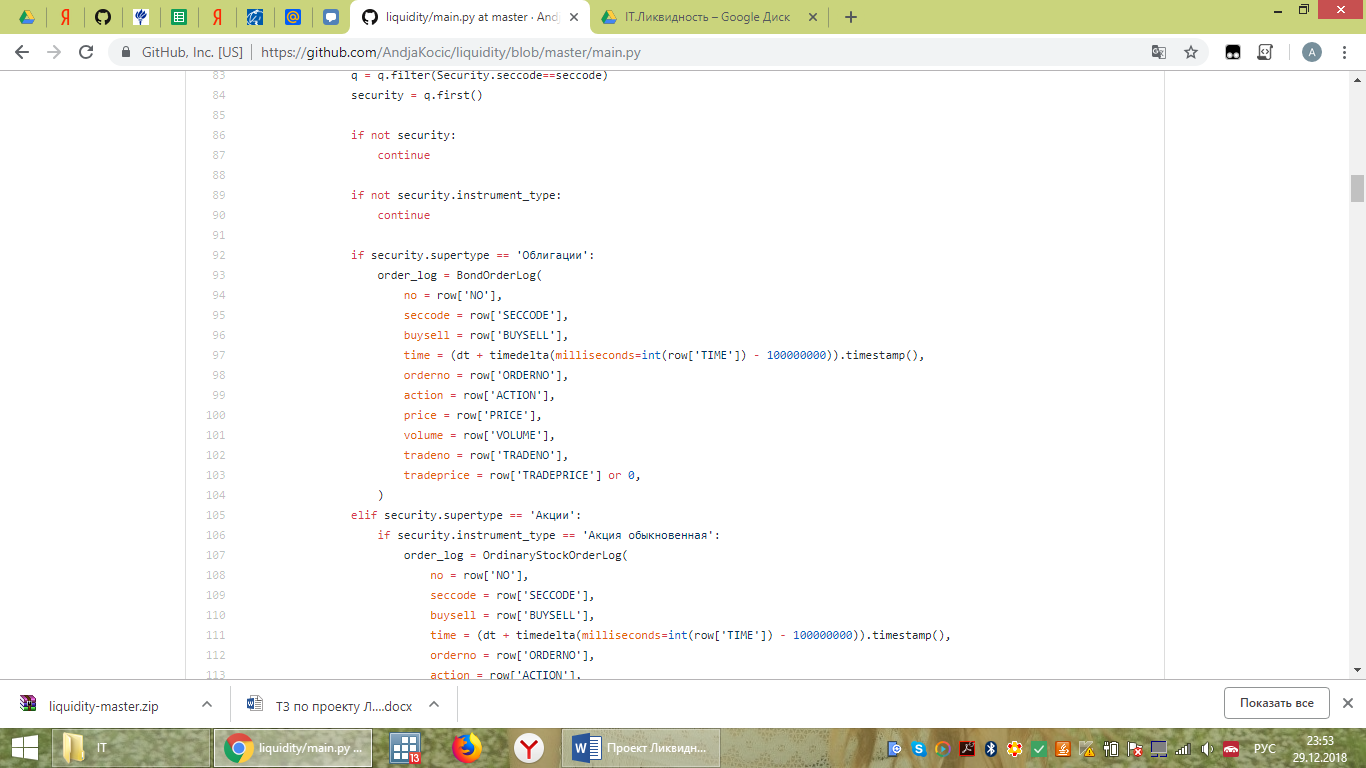


Далее производится заполнение созданных таблиц данными из исходного файла.

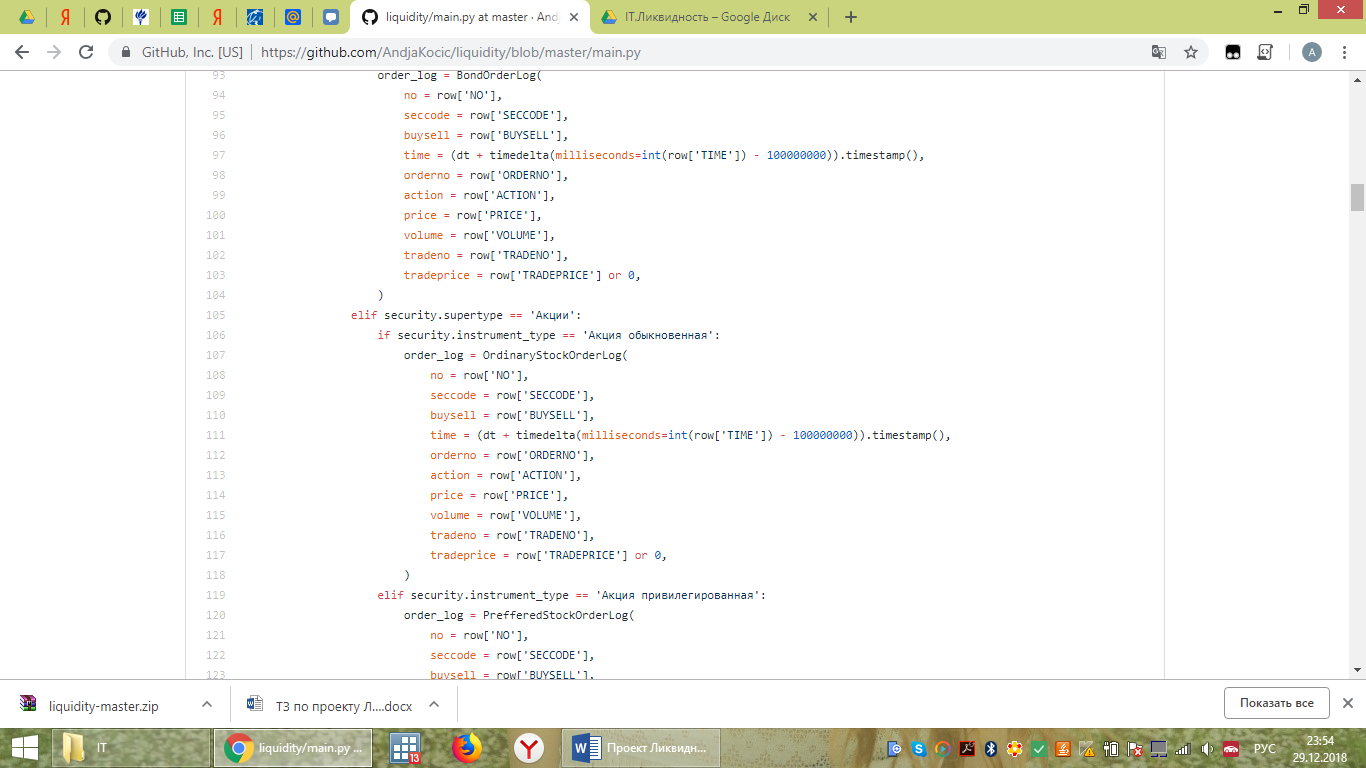
1. Заполнение классификатора:



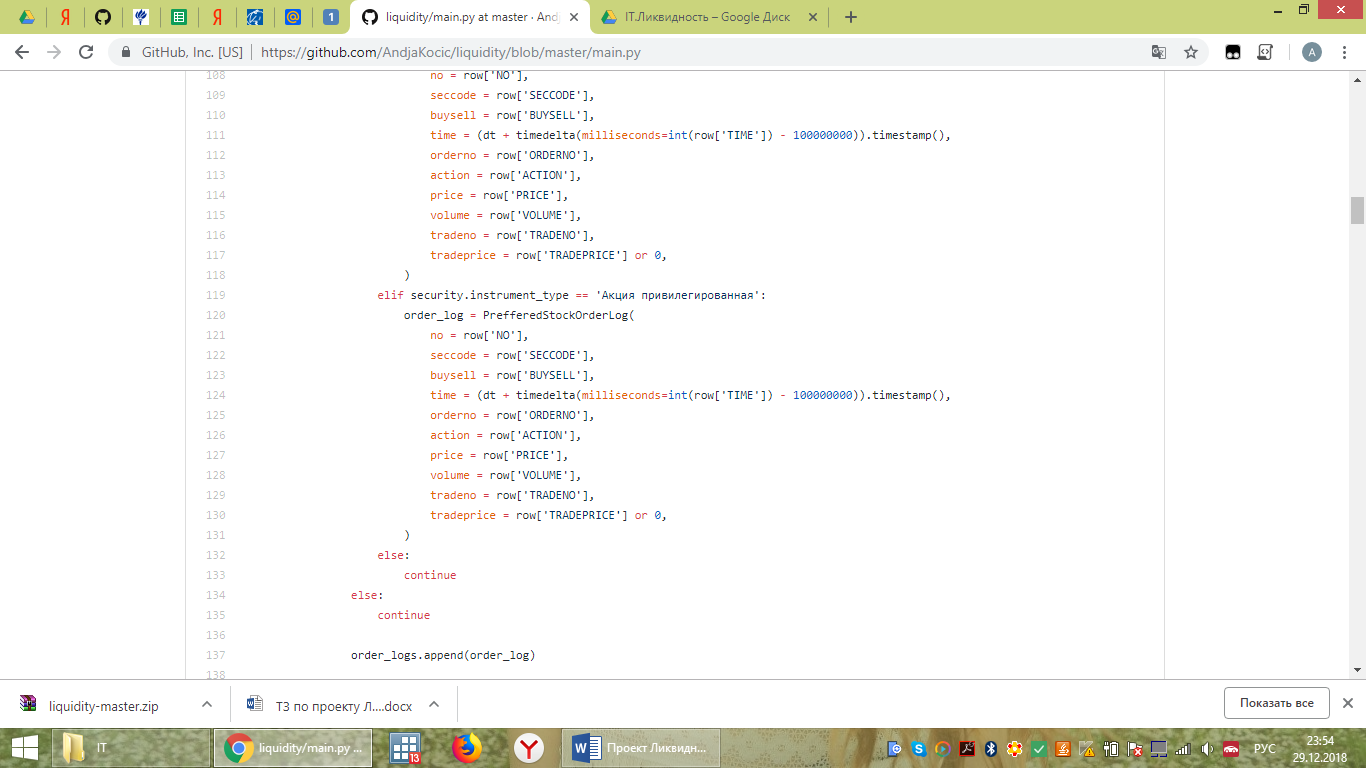
1. Соединение облигаций:



1. Соединение обыкновенных акций:



1. Соединение привилегированных акций:



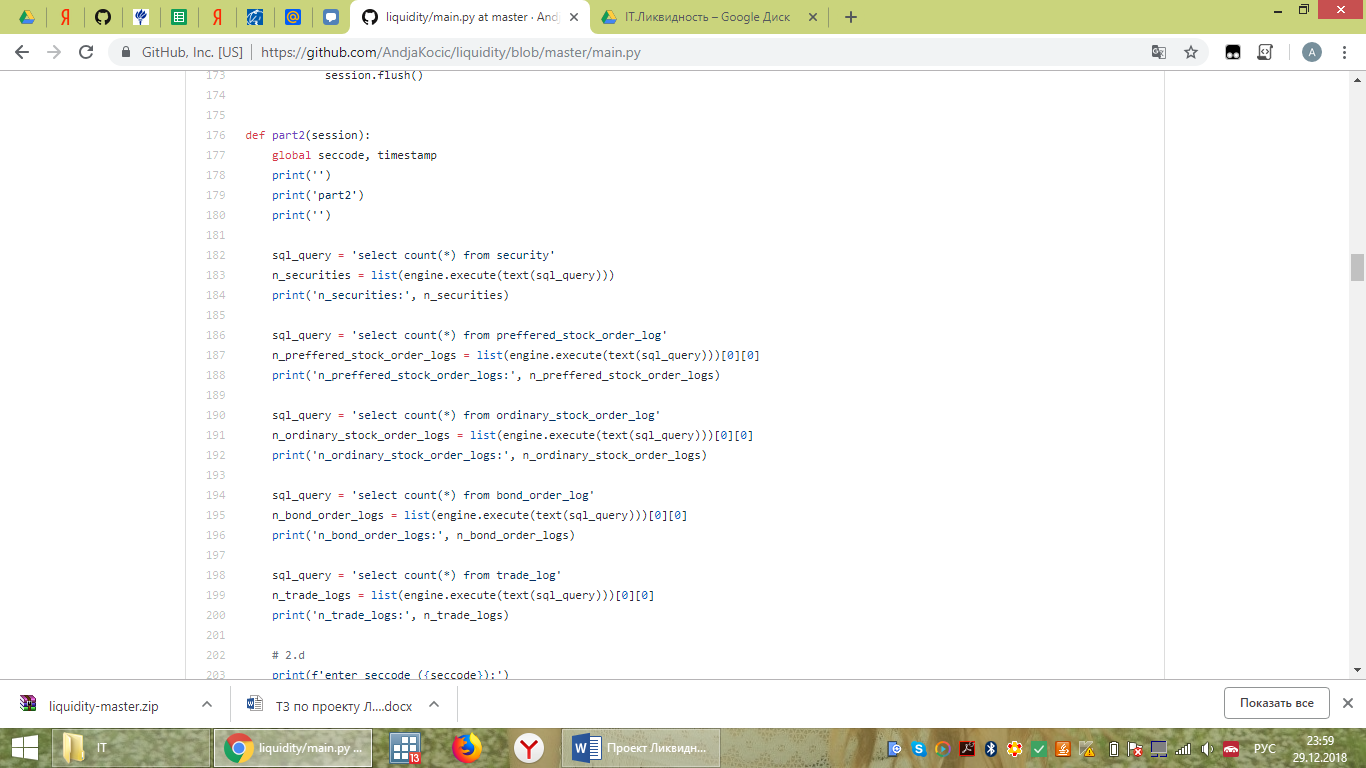
Следует отметить, что сессия прерывается, если длина ордерлога более 500000.

Для файлов типа .csv используется csv.DictReader.

**Блок 2: Анализ и переработка данных.**

В данном блоке посредством исполнения того же скрипта выполняется загрузка данных, выявление пропусков данных (пустых значений полей) и вывод описательной статистики.

Выявление и исправление ошибок выполняется посредством реализации следующих команд:

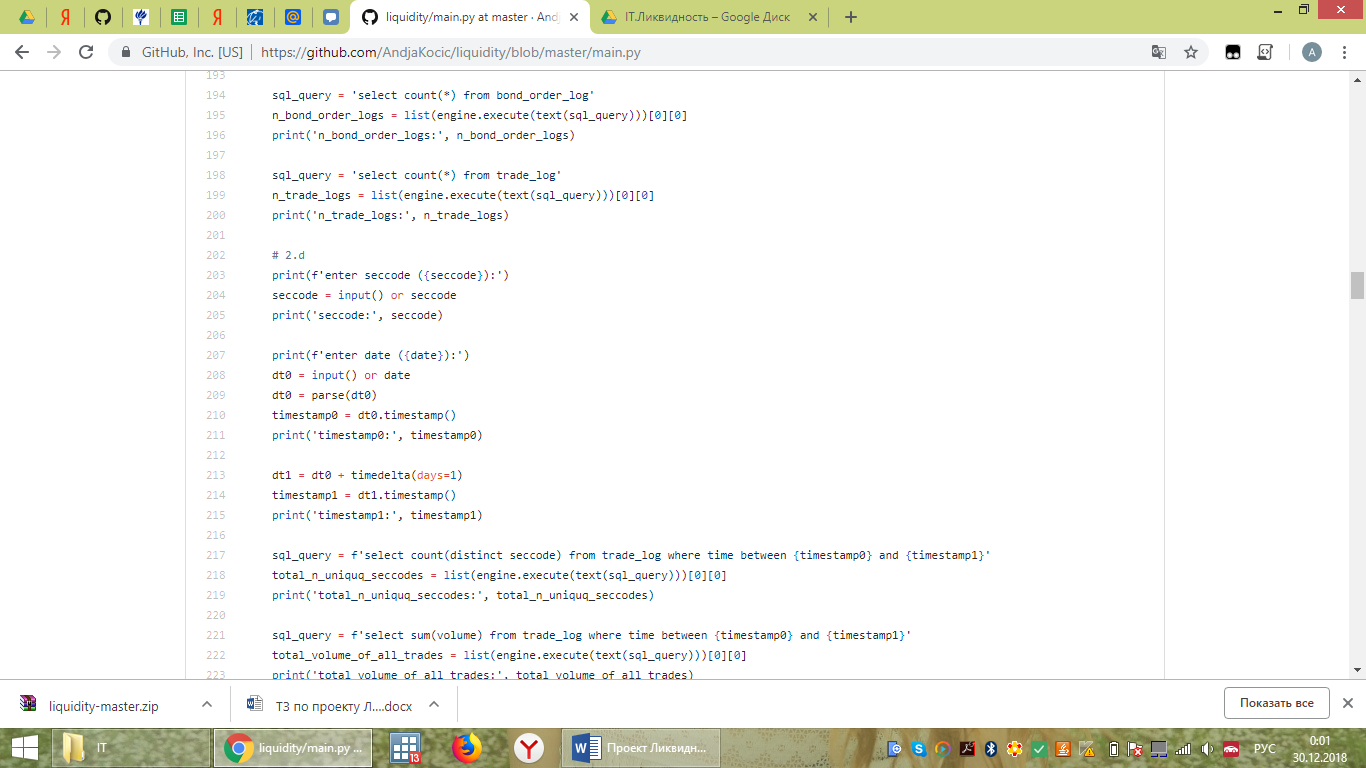


Выявление и исправление ошибок осуществляется полуавтоматически: программа выявляет нетипичные значения и предлагает пользователю для проверки; пользователь имеет возможность либо исправить ошибку, либо признать значение прощенным.

Следует отметить, что ошибкой признается значение, нарушающее формат поля или находящееся вне области определения.

Пользователь имеет возможность внести сделанные исправления в БД. Информация о пропусках сохраняется на жесткий диск с указанием номера строки из БД и имени поля, по которому найден пропуск/ошибка.

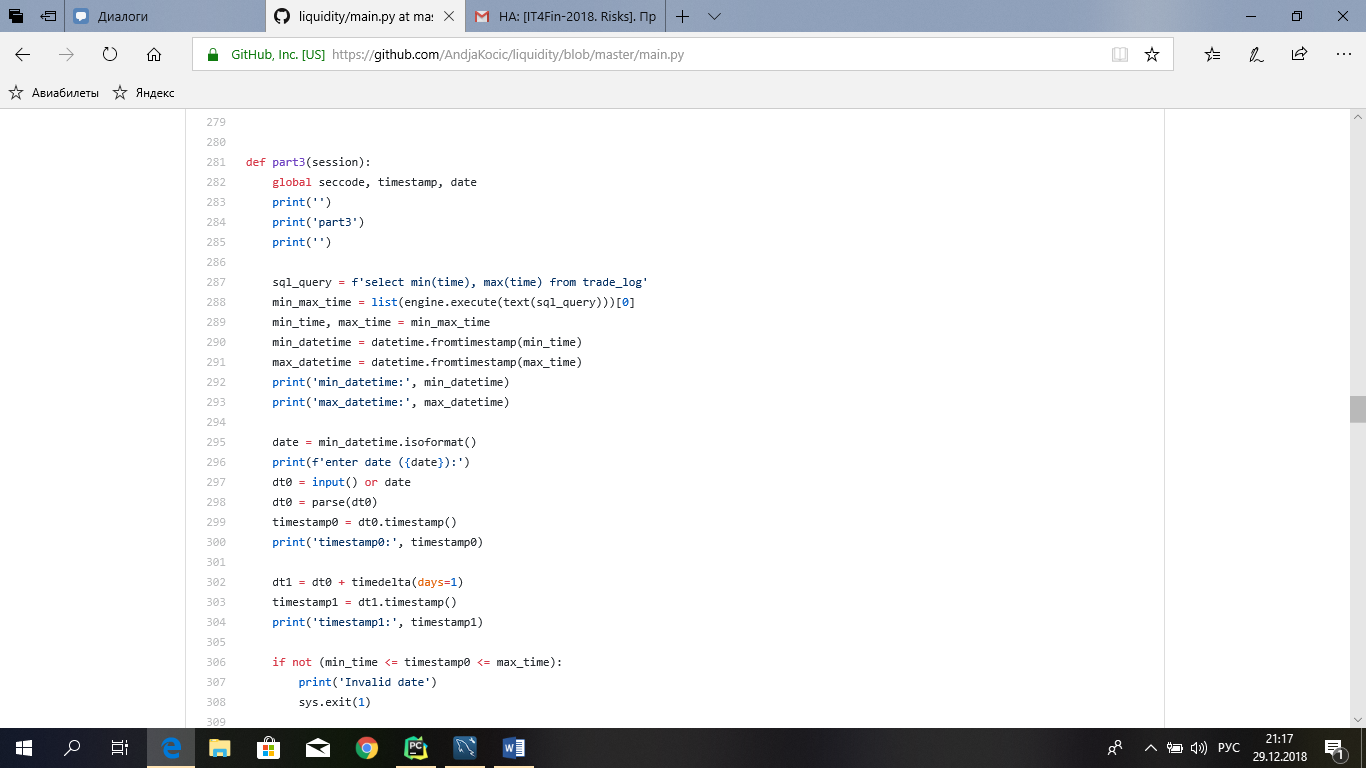
Описательные статистики выполняются посредством реализации следующих команд:



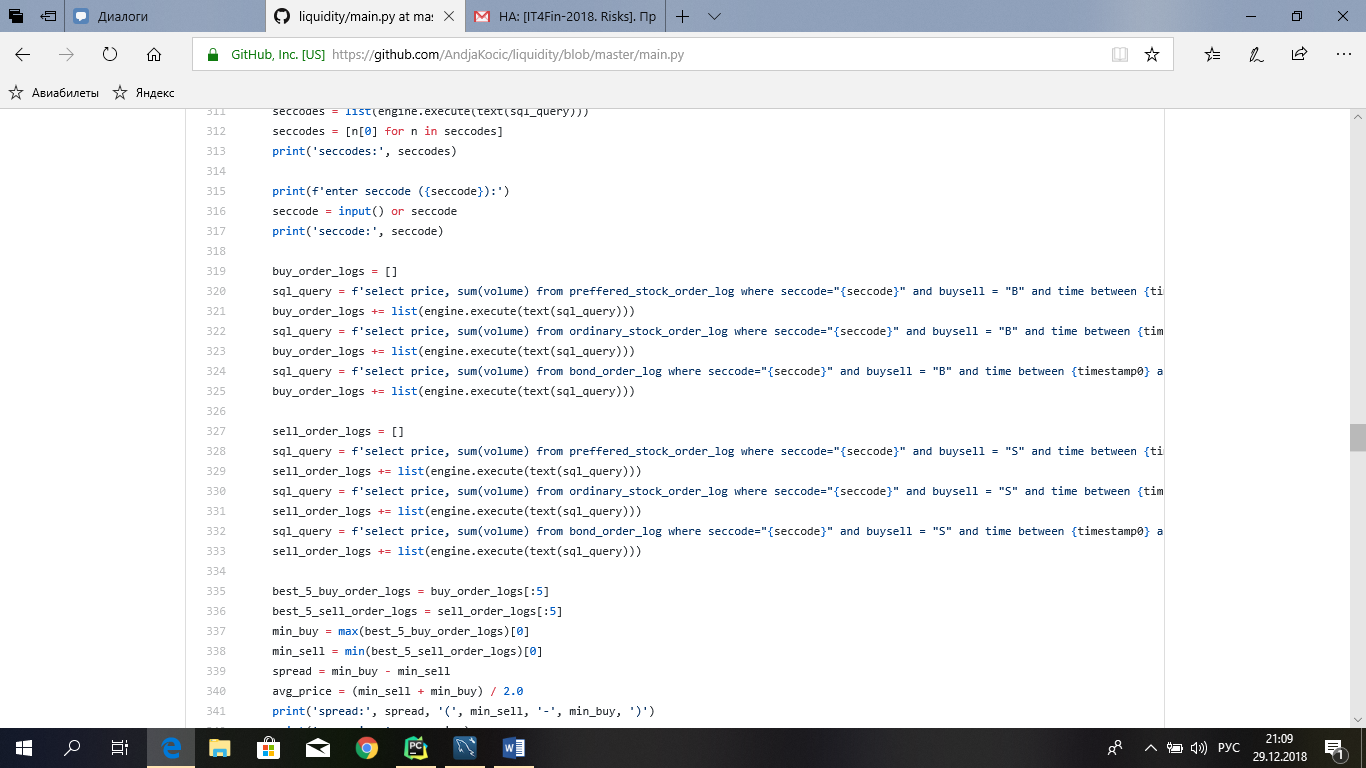
Далее осуществляется экспорт данных в файловую систему. Пользователь имеет возможность экспортировать результаты в файловую систему в общепринятых форматах (например, csv, xls, txt, png и т.д.).

**Блок 3 и 4: Построение книги лимитированных заявок, и оценка риска ликвидности.**

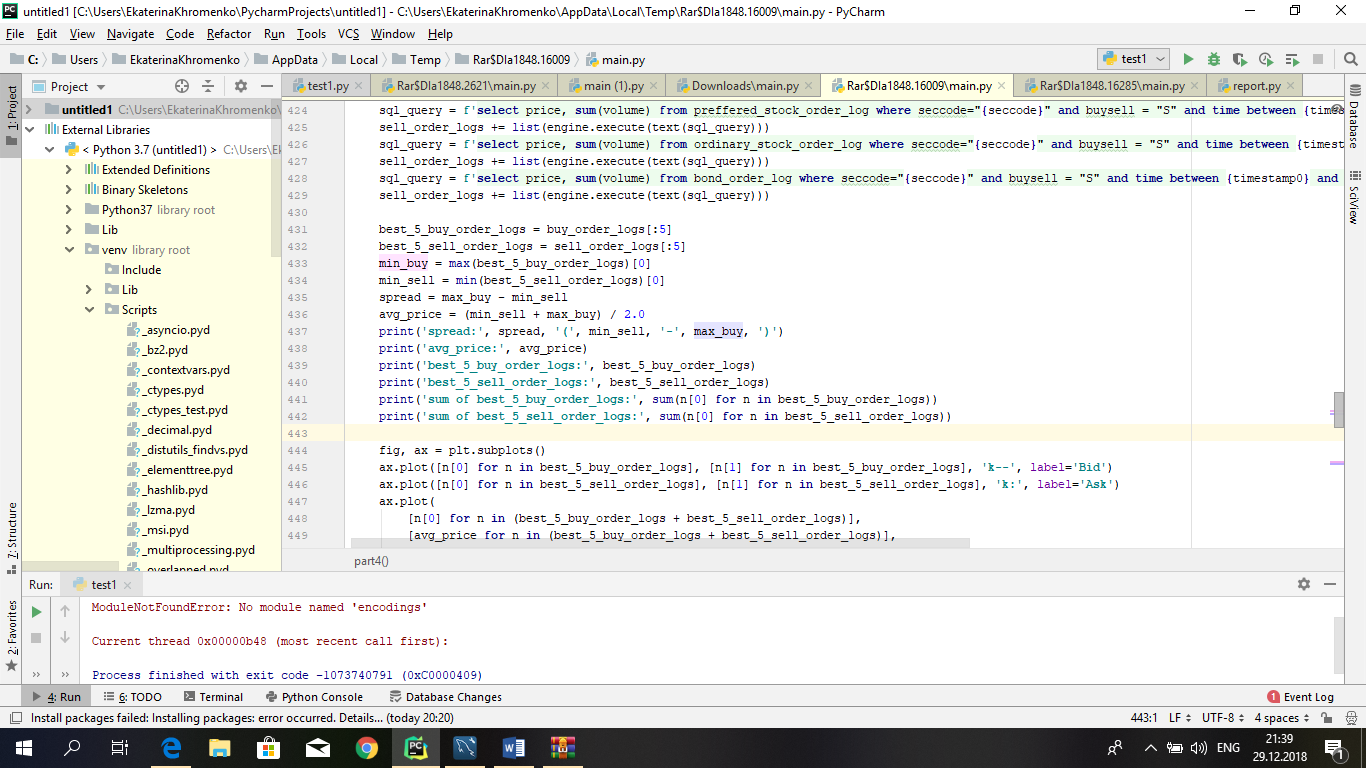
Данный фрагмент кода выбирает временной интервал, в рамках которого будут осуществляться запросы. В данном фрагменте задается диапазон времени, дельта, которая представляет собой интервал в 1 день.



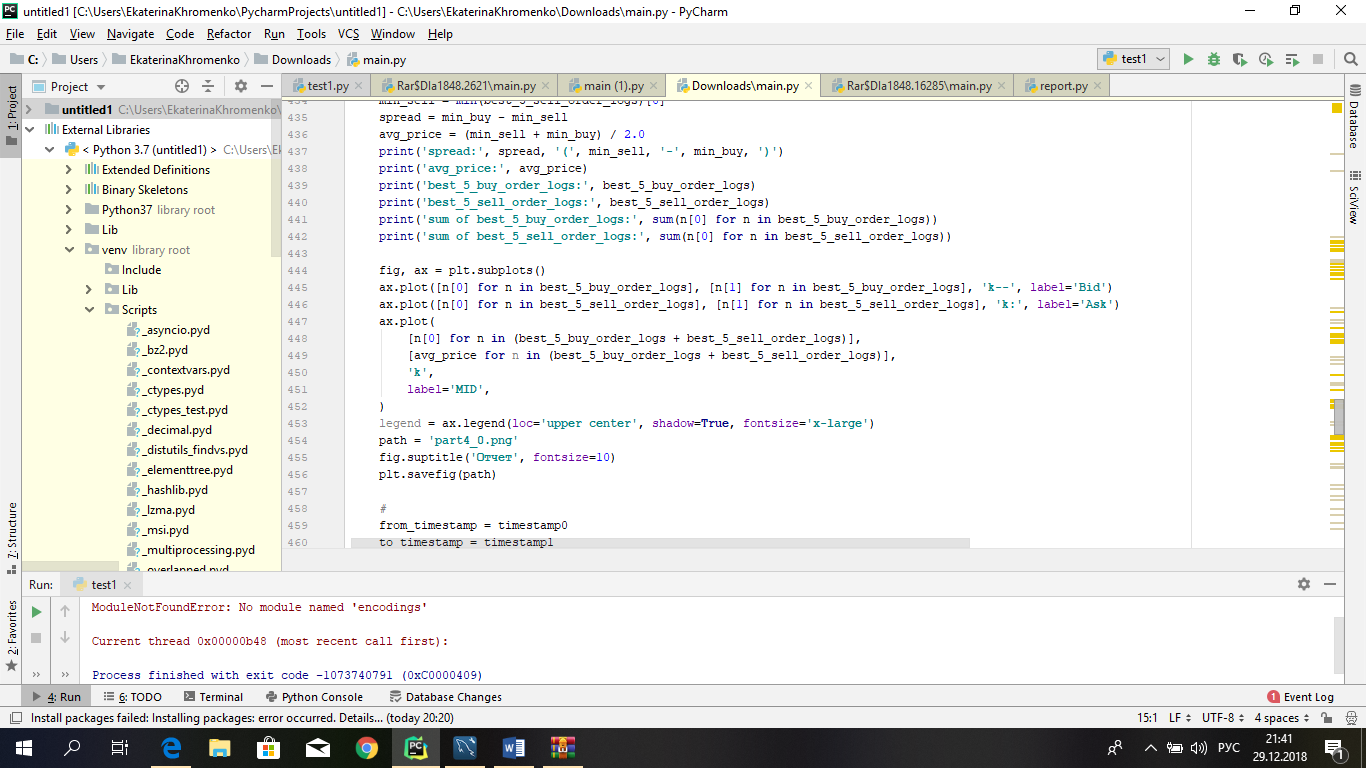
Данный фрагмент кода выбирает из таблиц привилегированных, обыкновенных акций и облигаций цену и объемы и выдает результат списком. Сначала запросы прописаны для заявок на стороне Buy, далее для заявок на стороне Sell. Заявки в запросах ранжируются по убыванию цены.



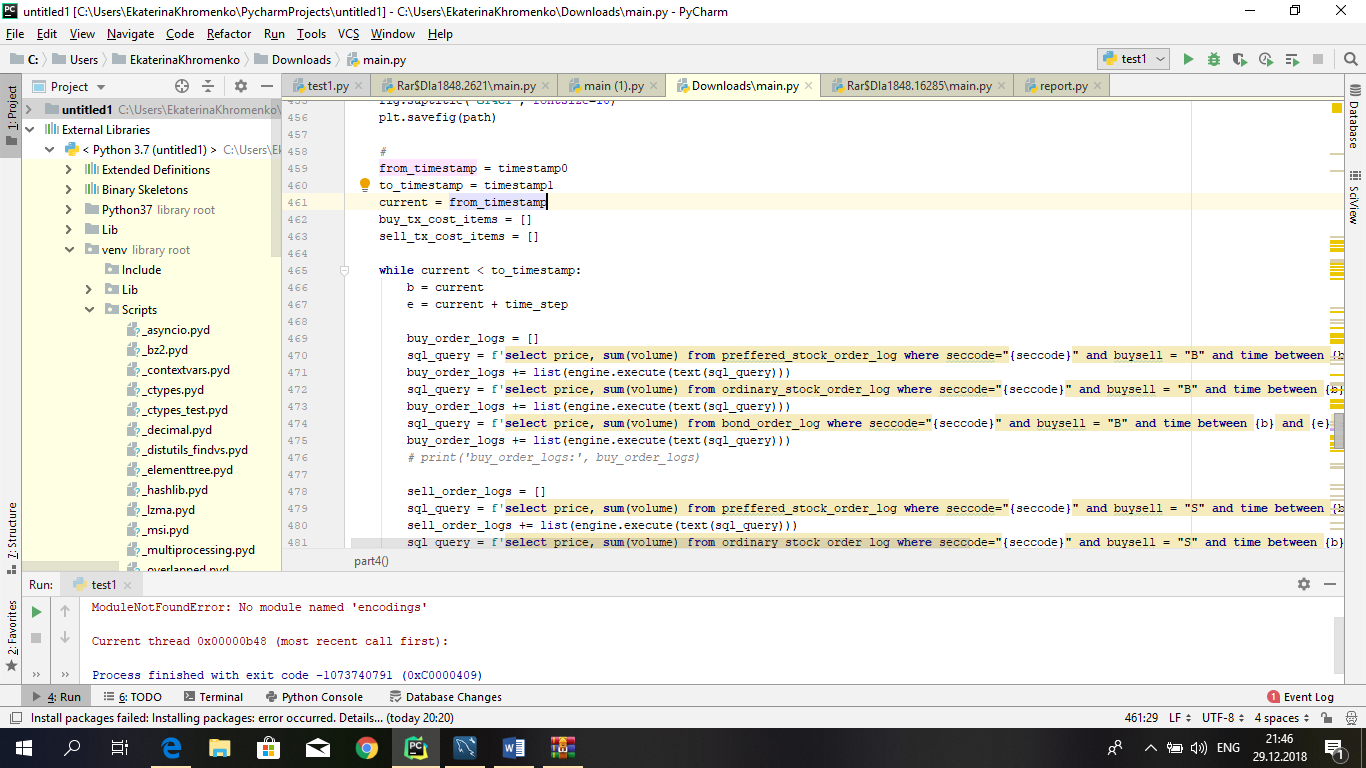
Далее выбираются наилучшие 5 заявок на стороне Buy и на стороне Sell. Наилучшими заявками Buy являются заявки с максимальными ценами. Наилучшими заявками Sell являются заявки с минимальными ценами. Спред вычисляется, как разница между максимальной ценой Buy и минимальной ценой Sell. Средняя цена вычисляется, как среднее между максимальной ценой покупки и минимальной ценой продажи.



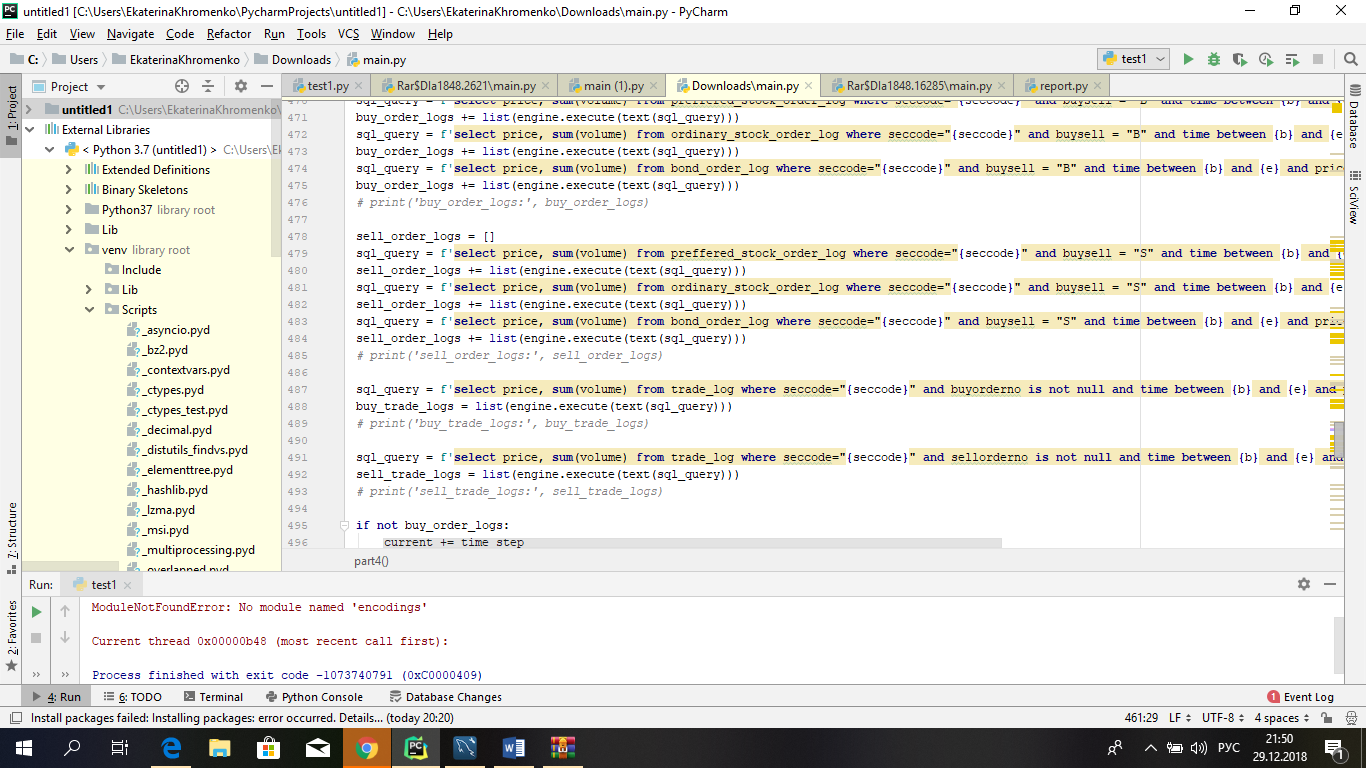
Далее визуализируются объемы на лучших ценах Bid и Ask.

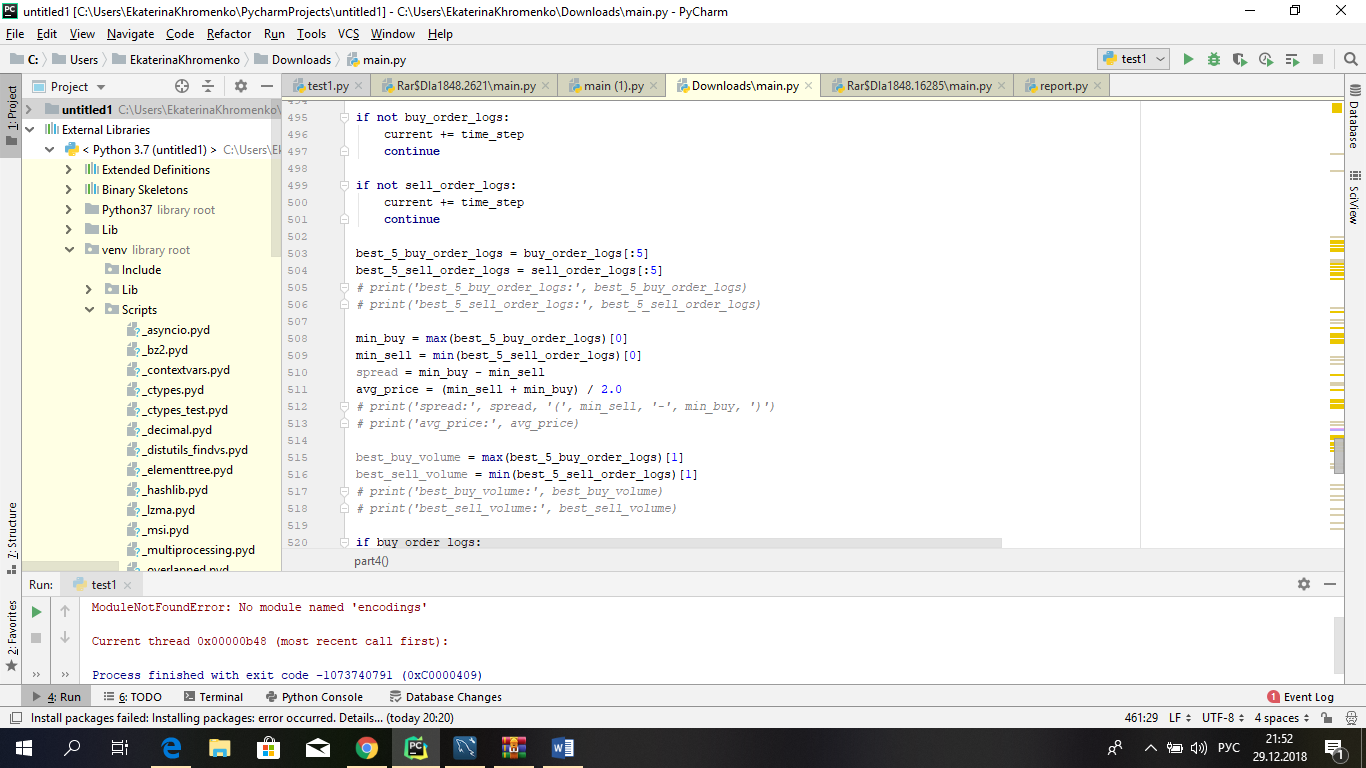


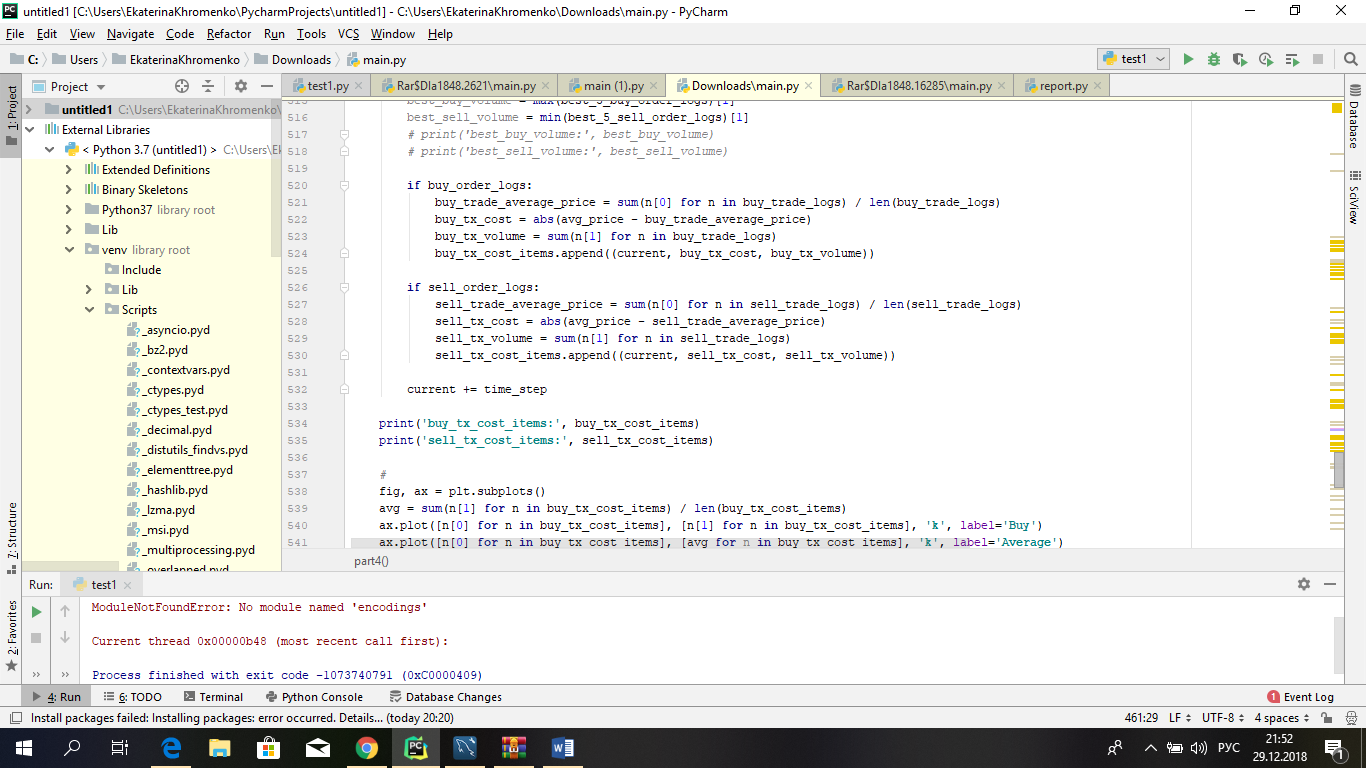
Далее выбирается временной интервал, в рамках которого будут строиться временные ряды. Запросом MySQL отбираются заявки со стороны Buy по указанному тикеру и в рамках данного периода времени с заданным шагом time\_step.



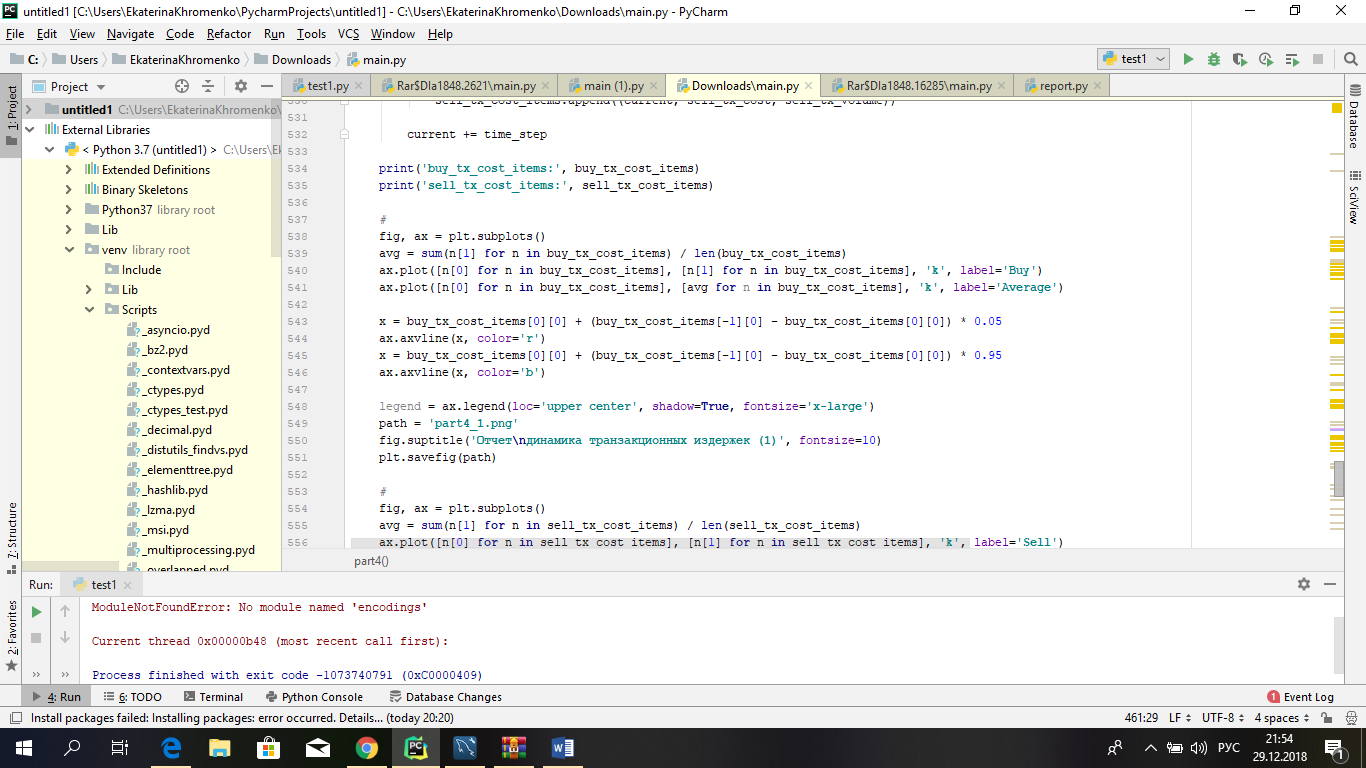
Запросом MySQL отбираются заявки со стороны Sell по указанному тикеру и в рамках данного периода времени с заданным шагом time\_step.



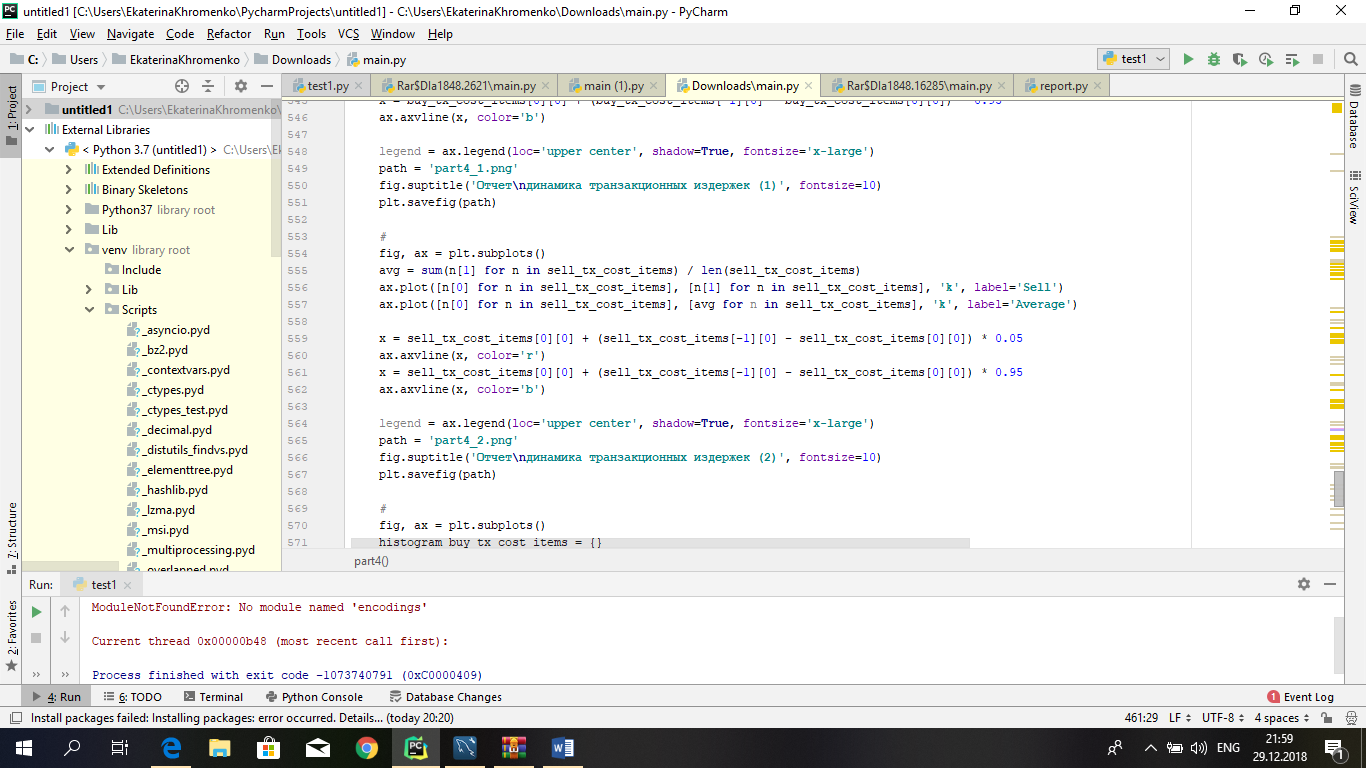




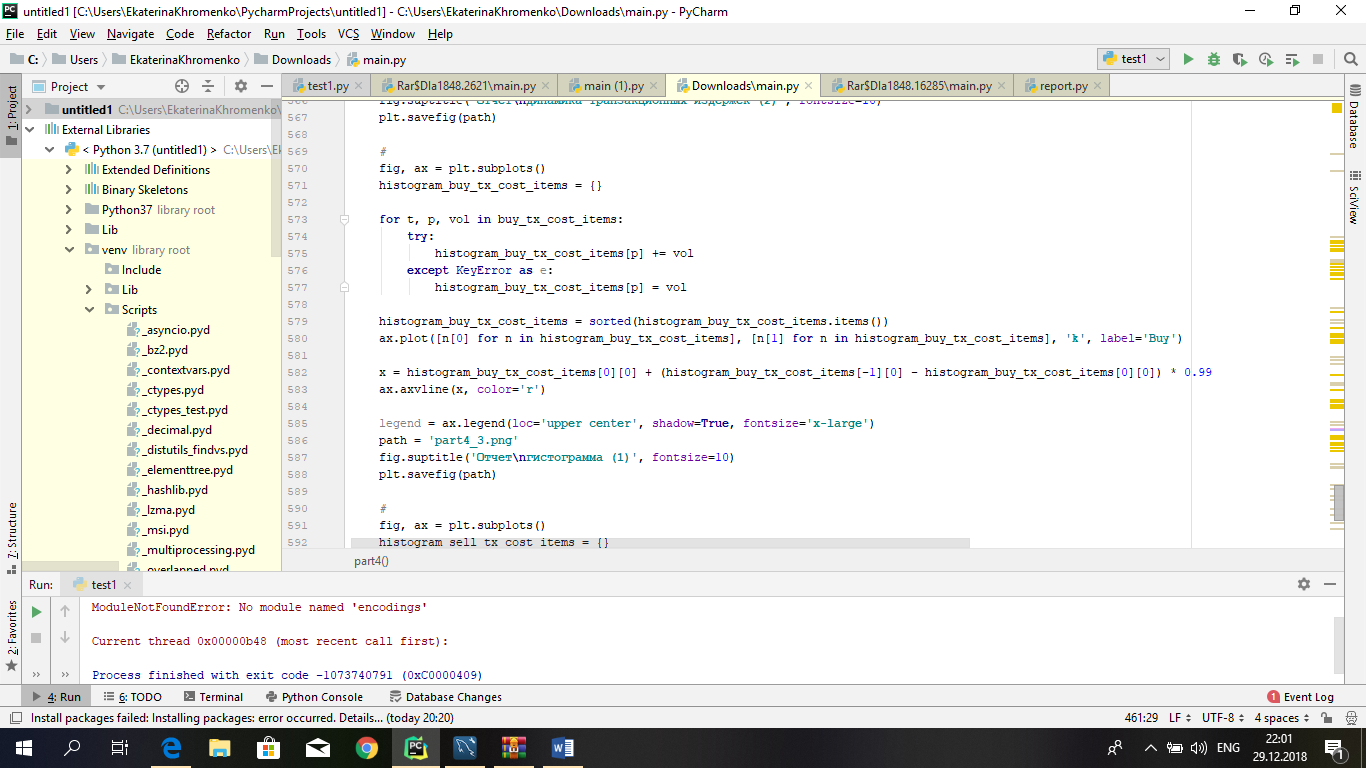
Строится временной ряд для трансакционных издержек с заданным шагом со стороны Buy. На графике должны быть отмечены 5% и 95% квантили, среднее и медиана ряда. Трансакционные издержки представляют собой разность между средней ценой MID и средней ценой, по которой объем был реализован.



Строится временной ряд для трансакционных издержек с заданным шагом со стороны Sell.



Строится гистограмма ряда со стороны Buy (на графике указывается 99% процентная квантиль)



Строится гистограмма ряда со стороны Sell (на графике указывается 99% процентная квантиль)

