

## Домашняя работа 2

Для выполнения задания ДЗ 2 использовать базу данных, созданную в результате ДЗ 1.

### Вариант 1. Кредитные рейтинги

1. Соединить Python с БД по API. (1 балл)
2. Из Python запросить информацию, необходимую для пункта 3. (2 балл)
3. Написать функцию, которая строит матричные характеристики рейтинговых переходов. Аргументы функции: вид рейтинга, границы периода наблюдения. На выходе функция должна выдавать не только матрицу, но и значения аргументов. Привести иллюстрационный пример использования. (2 балл)

**Примечания:**

- Ознакомиться с разделами 3.2.1 и 3.2.2.1 материала: <https://pdfs.semanticscholar.org/418e/c4f8208331ead58487be5e8bea1048561adc.pdf>
- Пользователь функции должен иметь возможность выбрать тип матрицы, которую построит функция:
  - i. матрица переходных вероятностей когортным методом (матрица P из раздела 3.2.1)
  - ii. матрица интенсивностей (генератор) методом дюраций (матрица  $\Lambda$  из раздела 3.2.2.1).
  - iii. функция обязана работать на рейтингах Эксперт РА; использовать данные о шкале рейтингов этого агентства из таблицы `scale_exp` из ДЗ 1.
- 4. (Бонусное задание) Предусмотреть исполнение SQL запроса из п.2 внутри функции из п. 3. Запрос должен храниться в выходных данных функции. (1 балл)

### Вариант 2. Облигации

1. Соединить Python с БД по API. (1 балл)
2. Из Python запросить информацию, необходимую для пункта 3. (2 балл)
3. Написать функцию, которая возвращает матрицу платежей облигаций заданного эмитента на заданную дату. Аргументы: код эмитента, дата матрицы платежей. Привести иллюстрационный пример использования. (2 балл)

**Примечания:**

- Конвенцию расчета дней для вычисления размера купонной выплаты игнорировать.
- Индикатор дефолта игнорировать.
- В матрице не должны учитывать уже погашенные облигации и не выпущенные на заданную дату.
- В матрице следует учесть облигации бескупонные, купонные с постоянным купоном и амортизационные.
- При наличии оферты в облигации, ее поток платежей вычислить до ближайшей оферты, в дату которой предположить погашение облигации.
- В результате должна получиться матрица платежей для инвестора, который купил по одной облигации из всех облигаций с детерминированным потоком платежей заданного эмитента, существовавших в заданную дату.

5. (Бонусное задание) Предусмотреть исполнение SQL запроса из п.2 внутри функции из п. 3. Запрос должен храниться в выходных данных функции. (1 балл)

### Материалы для сдачи

1. Краткая инструкция, содержащая:
  - а. подготовительные действия для работы с данными (где и в каком виде хранить данные, какой софт установить и т.д.);
  - б. краткое содержания процедур, заложенных в скриптах/запросах, а также порядок их запуска;
  - с. описание ожидаемых результатов каждого этапа.
2. Код сдается в виде одного \*.ipynb с комментариями и выполненными ячейками.
3. Все (кроме данных) необходимо разместить в репозиторий на github, в котором преподаватель является участником с правом редактирования.

### Требования к выполнению задания

1. Все процедуры и результаты должны быть воспроизводимы и описаны в мере, обеспечивающей использование третьими лицами.
2. Все пользовательские параметры, в том числе, параметры подключения к базе данных должны быть обособлены в отдельные переменные в начале скрипта.

### Правила и сроки выполнения

1. Домашняя работа является индивидуальной активностью студента.
2. Срок выполнения (первая итерация сдачи) Домашней работы 1: **9 декабря 23:59:59**.
3. Максимальная оценка за выполненное задание: **5 баллов + 1 бонусный**.
4. По результатам первой итерации студент получает фидбэк и предварительную оценку. Ориентировочно через **3 дня после сдачи работы**. Получив фидбэк, студент может исправить работу и сдать ее до **19 декабря**, но не позднее, чем другие домашние работы по курсу.
5. Студент пропустивший дедлайн первой итерации сдает работу на итоговую проверку **без возможности ее исправить**.
6. Студент выполнивший все пункты Домашней работы 1 (не зависимо от итерации) получает право проверки аналогичной работы сокурсника. Срок проведения peer-review: **4 дня с момента получения задания на проверку**. Продление сроков peer-review допускается по договоренности с преподавателем.
7. Peer-review проводится по следующим аспектам:
  - *Полнота* (по всем ли пунктам была предпринята попытка выполнения?)
  - *Понятность* (инструкция понятно написано? понятно ли прописаны требования к исходным данным? понятно ли, что делает код и каким результатам это приводит? пользовательские параметры обособлены и понятны?)
  - *Содержание* (сделано именно то, что требуется в задании? если нет, то в чем отличия того, что сделано? в чем это выражается и что Вы сделали, чтобы это увидеть?)
  - *Корректность* (разумно ли выбраны форматы поле таблиц? присутствуют ли ошибки исполнения кода или смысловые ошибки? в чем именно ошибка? как Вы это обнаружили? знаете ли Вы как это исправить? видите ли Вы заведомо абсурдные результаты?)
8. Максимальная оценка за peer-review: **4 балла**.