

Для удобной работы с исходными данными был импортирован pandas. Таблица с исходными данными была названа `old_df`. К сожалению, исходные колонки читались неправильно, вместо нормальных кириллических символов были кракозябры, по этой причине потребовалось переименовать колонки у этой таблицы.

Во-первых, предварительно с тем, что есть, можно сделать следующее: объединить в одну группу активы и пассивы, у которых одинаковая ставка процента и одинаковый срок погашения, просуммировав активы и пассивы там. У нас в итоге 116 групп. Так была создана `new_df`.

```
new_df = old_df.groupby(by=['Ставка, %', 'Срок погашения, дней'], as_index=False).sum()
```

Если посчитать RWA по тому, что имеется, то будет несколько больше 2 млрд. Как выяснится ниже, можно получить меньшее значение, хотя уже неплохо.

Для удобства можно создать такую штуку, как чистый актив, взяв разницу активов и пассивов везде, так был сделан столбец `s`.

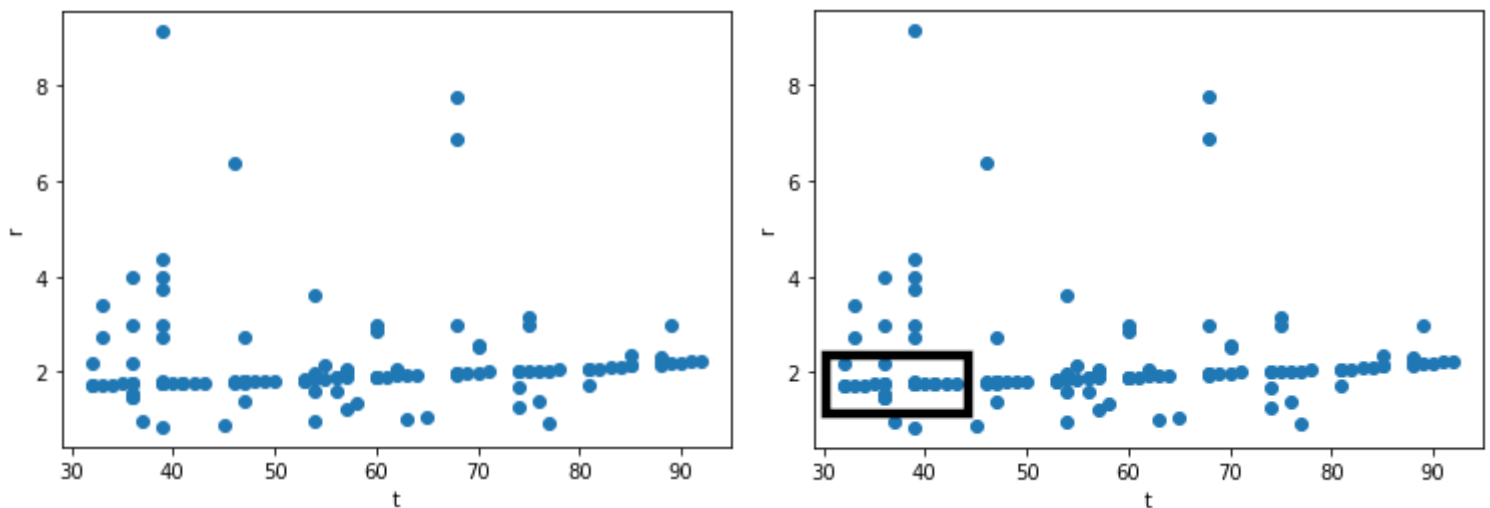
Дальше для удобства был сделан словарь, а затем кортеж `d`. Он состоит из `r`, `t`, `s`. Как несложно догадаться, это, соответственно, процентная ставка, срок погашения и чистый актив.

Далее имеются две важные функции. `rwa(t, a, p) -> float` берёт на вход срок погашения, актив, пассив, а возвращает `rwa` по формуле: $(a + p) \cdot w_{gross} + |a - p| \cdot w_{net}$

Далее основная часть программы, которая занесена в функцию `main`., на вход она берет кортеж `d`.

Суть программы следующая. Наши 116 групп в зависимости от `r` и `t` выглядят так:

Что с этим делать дальше? Можно как-то разбить их по прямоугольникам с указанными в задаче соотношениями: $\Delta r = 0.15$; $\Delta t = 30$. В этих прямоугольниках будут находиться как раз группы



неттирования. Иллюстрация на втором рисунке. Чтобы такой прямоугольник сделать, надо выбрать какую-то исходную точку. Как это сделать? В принципе, можно брать некоторую точку, делать от неё отступы влево, вправо, вверх, вниз. Тогда те точки, которые входят в

допустимые расстояния, будут расположены внутри прямоугольника. Пусть исходная точка будет в центре, тогда прямоугольник для группы неттирования будет выглядеть так:



Дальше будем пробегаться по кортежу `d`. Я выделил чистые активы и чистые пассивы для того, чтобы считать RWA (`sA`, `sP`). Если некоторая точка лежит в некотором прямоугольнике, центром которого был выбран центр некоторой другой точки, то она добавляется в группу неттирования.

Также надо запоминать, какие точки мы уже добавили в группы неттирования, для этого есть список `indexes`. По ходу дела также обновляется `new_df` с ответами на номера группы. Для того, чтобы подсчитать `rwa`, был создан словарь `groups`, устроенный так:

номер группы : (исходный `t`, исходная `г`, всего активов, всего пассивов). Дальше всю необходимую информацию можно вытащить из него и посчитать RWA. У меня получилось 1_392_201_363.599. Групп получилось 38. Промежуточное разбиение для 116 групп, объединенных по ставкам и срокам, записаны в файл `промежуточный_результат.csv`

Если экспериментировать с разным порядком выбора исходных точек, то можно добиться немного лучшего результата, однако код будет работать дольше и будет более громоздким. Остаётся только соотнести исходные данные с полученными группами неттирования. Они все заносились в `new_df`, где исходные данные объединены по процентным ставкам и срокам. Из `new_df` для каждой из 116 групп достаётся ставка, срок и номер группы. В `old_df` где ставка и срок такие же, изменяется номер группы.

Затем полученная таблица записывается в `answer.csv`.

Важно заметить, что новые таблицы csv имеют разделителем запятую, а не точку с запятой.