Математическое моделирование

**СУЩНОСТЬ RWA**

Активами, взвешенными с учетом риска или RWA (risk-weighted assets) называются балансовые и вне балансовые активы банка, умноженные на определенный риск-вес. Этот вид расчета активов используется при определении коэффициента достаточности капитала для финансового учреждения.

Обычно все активы подразделяются на несколько групп, каждой из которой в свою очередь присвоен определенный коэффициент, на который умножаются все активы данной группы перед тем, как получится итоговая сумма. Чтобы попасть в определенную группу, активы должны удовлетворять требованиям, присущим именно этой группе.

Плюсами этого подхода является, во-первых, то, что он позволяет стандартизировать и упростить подход к оценке и сравнению банков из различных географических регионов. Во-вторых, данный подход позволяет сравнительно легко учитывать вне балансовые активы при подсчете норматива.

Величина необходимого капитала для покрытия рыночного риска складывается из процентного риска для долговых инструментов, фондового риска для акций и валютного риска. Подход для оценки величины процентного риска основан на коэффициентах риска: разбитые по срокам до погашения чистые позиции по ценным бумагам умножаются на соответствующие срокам коэффициенты риска. Подход для фондового риска также сводится к произведению позиций по акциям и коэффициентов риска, а в качестве валютного риска используется размер открытой валютной позиции.

Таким образом мы видим, что RWA является инструментом, благодаря которому банки могут объективно оценивать риски в прибыли активов и пассивов банка, тем самым оценивая будущий капитал предприятия, перспективы на будущее и примерную оценку прибыли банка как в долгосрочной, так и в краткосрочной перспективе.

использование подхода с учетом веса риска является предпочтительной методологией, которую банки должны использовать для расчета капитала:

Некоторые активы, такие как долговые обязательства, имеют более высокий риск, чем другие, такие как денежные средства или государственные ценные бумаги/облигации. Поскольку различные типы активов имеют различные профили риска, взвешивание активов в соответствии с их уровнем риска в первую очередь корректируется для менее рискованных активов, позволяя банкам дисконтировать активы с более низким риском. В самом простом применении, государственный долг допускается 0% «взвешивание риска» - то есть, они вычитаются из общих активов для целей расчета автомобиля.

В 1988 году Базельским комитетом по банковскому надзору был подготовлен документ, который рекомендует определенные стандарты и правила для банков. Это было названо Базелем I, и комитет выступил с пересмотренной структурой, известной как Базель II. Основная рекомендация этого документа заключается в том, что банки должны иметь достаточный капитал, равный по крайней мере 8% от своих активов, взвешенных по риску. Совсем недавно комитет опубликовал еще одну пересмотренную структуру, известную как Базель III. расчет суммы активов, взвешенных с учетом риска, зависит от того, какой пересмотр Базельского соглашения следует за финансовым учреждением. Большинство стран внедрили ту или иную версию этого положения.

**ХОД РАБОТЫ**

В процессе создания программы наша команда столкнулась с некоторыми трудностями в написании алгоритма: изначально были проблемы, связанные со временем исполнения программы- алгоритм был настроен на огромное количество операций, что привело бы к бесконечному исполнению программы, но нам удалось это исправить за счет изменения алгоритма.

На первых этапах работы наша команда тщательно ознакомилась с исходными данными, затем открыв таблицы в EXCEL мы увидели список данных, с которыми предстоит в дальнейшем работать. Следующей важной проблемой оказалось умение считывать данный с таблицы в EXCEL:

Мы использовали язык python а в нём библиотеки под названием xlrd, xlwt.

**АЛГОРИТМ РАБОТЫ ПРОГРАММЫ**

В пункте (а)

* Сначала мы считываем данные из таблицы excel

import xlrd

import xlwt

import xlsxwriter

rb = xlrd.open\_workbook('dannye\_mamont\_3mes.xlsx')

sheet = rb.sheet\_by\_index(0)

wb = xlwt.Workbook()

ws = wb.add\_sheet('RWA')

val = sheet.row\_values(0)[0]

vals = [[rownum] + sheet.row\_values(rownum) for rownum in range(sheet.nrows)][1:]

* сортируем их по процентной ставке

vals = sorted(vals, key=lambda x: float(x[5]))

 мы разбиваем их на группы, так чтобы разница между максимальными минимальным элементами была не больше, чем 0,15%.

* Считаем пассивы и активы в каждой группе, затем вычитаем пассивы из активов- если разность больше 0 тогда в активы этой группы записываем данную разность, а пассивы обнуляем, если меньше 0, то в пассивы этой группы записываем модуль разности, а активы обнуляем.

group = 1

min = vals[0][5]

Active, Pasive = 0, 0

nowA, nowP = 0, 0

for i in vals:

ws.write(i[0], 0, i[1])

ws.write(i[0], 1, i[2])

ws.write(i[0], 2, i[3])

ws.write(i[0], 3, i[4])

ws.write(i[0], 4, i[5])

if abs(min - float(i[5])) > 0.15:

group += 1

ws.write(i[0], 5, group)

min = float(i[5])

if nowA > nowP:

Active += (nowA - nowP)

else:

Pasive += (nowP - nowA)

nowA, nowP = 0, 0

else:

ws.write(i[0], 5, group)

nowA += float(i[1])

nowP += float(i[2])

* Складываем активы и пассивы всех групп и подставляем их в формулу RWA:

**RWA = (∑Актив + ∑Пассив) \*Wgross + |∑Актив- ∑Пассив|\*Wnet**

Таким образом исходя из исчислений в первом пункте RWA равен 441794826.3495747

В пункте (б)

* Считываем данные из таблицы EXCEL

import xlrd

import xlwt

import xlsxwriter

rb = xlrd.open\_workbook('dannye\_mamont\_3mes.xlsx')

sheet = rb.sheet\_by\_index(0)

wb = xlwt.Workbook()

ws = wb.add\_sheet('RWA')

val = sheet.row\_values(0)[0]

vals = [[rownum] + sheet.row\_values(rownum) for rownum in range(sheet.nrows)][1:]

* Сортируем их по процентной ставке и по дате погашения

vals = sorted(vals, key=lambda x: (float(x[5]), int(x[3])))

* Далее начиная с первого элемента разбиваем их по группам, так что разница в процентной ставке между максимальным и минимальным элементами не превосходит 0,15%, а разница между сроками погашения не превосходит 30 дней и неттируем пассивы и активы каждой группы

group = 1

min = vals[0][5]

minDay = vals[0][3]

Active, Pasive = 0, 0

nowA, nowP = 0, 0

for i in vals:

ws.write(i[0], 0, i[1])

ws.write(i[0], 1, i[2])

ws.write(i[0], 2, i[3])

ws.write(i[0], 3, i[4])

ws.write(i[0], 4, i[5])

if abs(min - float(i[5])) > 0.15 or (int(i[3]) - int(minDay)) > 30:

group += 1

ws.write(i[0], 5, group)

min = float(i[5])

minDay = int(i[3])

if nowA > nowP:

Active += (nowA - nowP)

else:

Pasive += (nowP - nowA)

nowA, nowP = 0, 0

else:

ws.write(i[0], 5, group)

nowA += float(i[1])
nowP += float(i[2])

 Вычисляем значение RWA по формуле:

print((Active + Pasive) \* 0.1 + abs(Active - Pasive) \* 0.4)

**RWA = (∑Актив + ∑Пассив) \*Wgross + |∑Актив- ∑Пассив|\*WЗnet**

Таким образом исходя из исчислений во втором пункте RWA равен 1121651548.93469

За используемые константы мы взяли значения Wgross и Wnet

|  |  |  |
| --- | --- | --- |
| Временная группа | Wgross | Wnet |
| До 1 месяца | 0,05 | 0,2 |
| От 1 до 3 месяцев | 0,1 | 0,4 |

В своих расчётах мы так же использовались значения активов и пассивов(данных в таблице)

Наши алгоритмы можно назвать эффективными , так как они работают относительно быстро, мы не можем обосновать, что наша программа самая оптимальная, но наши рассуждения об эффекивности работы этого алгоритма основывались на том, что мы можем каждое число записать сколько в несколько групп, но заметим, что для каждой группы неттировавния RWA равно модулю разности активов и пассивов, а знчит разницы между тем,как разбивать на группы нет.

Положительные аспекты программы

* Быстрая работа программы- меньше чем за секундунд
* В нем множество перспектив для последующего развития
* Облегчать восприятие кода и использование программы. Этому способствуют логичное именование и хороший стиль интерфейса и реализации.
* Удобный процесс внесения дальнейших изменений. Чем лучше оптимизирована структура, тем проще изменять код, добавлять новые свойства, повышать быстродействие и изменять архитектуру.
* Обеспечивать устойчивость программы. При внесении изменений или возможных неполадках можно легко внести исправления. А правильная обработка ошибок значительно облегчает эксплуатацию программного продукта.

**ССЫЛКИ НА ИСТОЧНИКИ ИНФОРМАЦИИ:**

* <https://habr.com/ru/company/geekbrains/blog/270001/>
* <https://dic.academic.ru/dic.nsf/ruwiki/1701093>
* <https://en.wikipedia.org/wiki/Risk-weighted_asset>
* <https://2016.report-sberbank.ru/ru/management-of-risks/significant-risks>
* <https://studbooks.net/2026719/bankovskoe_delo/aktivy_vzveshennye_uchetom_riska>
* <https://www.sberbank.com/common/img/uploaded/ir/docs/sberbank_analyst_day_2018_ru.pdf>