Preface

Stephen J. Mildenhall

2026-10-22 19:58 commit b7145b5+

A recitation is the class that sits beside a lecture, where an instructor works problems from the assigned text. This book is a recitation for actuarial textbooks and examinations. It works their examples and exercises with `aggregate`, a Python package for compound probability distributions with its own specification language, DecL. Every number in it is computed when the book is rendered, never transcribed, so an answer here can be set beside the one printed in the text.

The book is written for teachers and students. A teacher can use a chapter to show how a problem posed for hand calculation becomes a short program, and how the same program answers every variation on the question. A student can use it to check their own work, and to see the whole distribution where the text asks only for a mean or a single probability. We assume the source text is open beside this one: each chapter is named for the book it follows, and each section for the example or exercise it works.

The problems fall into three categories. A **translation problem** states a frequency, a severity, and a coverage, and asks for a moment or a probability; most of *Loss Models* Chapter 9 and *Distributions for Actuaries* Chapter 4 fall into this category. The work is writing the DecL, and the text is the authority: the program should reproduce its answer. The value to a student is everything else that comes with the answer, because the program computes the whole distribution and not just the requested quantity.

An **approximation problem** asks for a normal, lognormal, gamma, or compound Poisson approximation to an aggregate distribution. The texts teach approximations because historically the exact distribution was expensive to compute. Here the roles reverse: `aggregate` computes the exact answer, and the approximation becomes the quantity being checked.

An **application problem** uses the distribution to make a decision: increased limit factors and risk loads, aggregate limits and deductible credits, a reinsurance program, an allocation of capital. These problems carry the most arithmetic outside the distribution itself, and they show how far a short program reaches into actuarial practice.

Companion volumes

Three volumes divide the material. *Recitations* holds worked examples and exercises from textbooks, study texts, and examinations. *Reproductions* holds results from published research papers, rebuilt with `aggregate` and checked against the papers' own tables and figures. The monograph, *Working with Aggregate Probability Distributions*, holds original teaching material and the author's own analysis, including analysis built on top of a published example. A reader new to aggregate distributions should start with the monograph's two student chapters, the Student guide and the Actuarial Student guide. They are original material, with no published answer to check against, so they live in the monograph rather than here.

Running the programs

The aggregate Loss Lab, aLL for short, runs the library in the browser with no installation, and is the quickest way to try a program from this book. The same programs run through `build` in any Python session with `aggregate` installed. The appendix collects every program the book builds, chapter by chapter, and `recitations.agg`, beside the book's source, holds the same programs as one file that `build.load` reads, each named for the text and noted with the example or exercise it works.

About this edition

This book reports the environment that computed it. Every chapter is recomputed whenever the library or any other installed package changes, so all the numbers in a given copy come from the one environment described below.

```
Exception reporting mode: Minimal
Rendered 2026-10-22 19:58:26 GMT Daylight Time (UTC+0100)
----------------------------------------------------------
Python           | 3.14.7 on Windows-11-10.0.26200-SP0
aggregate        | 1.7.1, source at git commit 0036744
greater_tables   | 8.1.0
```