Using `aad` and `agglimit`

Annual aggregate deductible and aggregate limit on an occurrence layer (aad, agglimit)

Stephen J. Mildenhall

2026-10-30

The recovery from an occurrence layer `y xs a` on the claims $X_1, \dots, X_N$ of a year is $R = \sum_i \min((X_i - a)^+, y)$. An annual aggregate deductible d and an aggregate limit m apply to that annual recovery, and we call them the **annual terms**. Under them the reinsurer pays

$$A(R) = \min((R - d)^+, m)$$

and the cedant keeps $L - A(R)$, where L is the gross annual loss. The amounts are for 100% of the layer: under a share s the reinsurer pays $s\min((R_{100} - d)^+, m) = \min((R - sd)^+, sm)$ on the shared recovery $R = sR_{100}$.

The annual terms are the DecL clauses `aad d` and `agglimit m`[1], written directly after the occurrence layer and before any premium clause. A `pnl` or `xpnl` books them[2]. The reinsurer's side is a function of R alone. The cedant's net is a function of the pair (L, R), not of $L-R$. Neither of L and R determines the other, because the claim count decouples them.

A plain `agg` therefore applies the annual terms after `occurrence ceded to`, as the aggregate stage `aggregate ceded to sm xs sd`, and drops them, with a warning, after `occurrence net of`. A `pnl` statement computes the profit and loss of a program, premium less losses, and an `xpnl` computes the same account in steps from gross to net. Both compute the joint distribution of (L, R) and book the cedant's net.

```
occurrence ceded to 750 xs 250 aad 50
occurrence ceded to 750 xs 250 agglimit 500
occurrence net of 90% po 750 xs 250 aad 50 agglimit 500 deposit 100 cede 10%
```

```
import warnings
import numpy as np
import pandas as pd
from aggregate import build, qd, __version__
from style import wprint, GT, refuse, figures
figures()
print(__version__)
```

```
1.43.1
```

[1] From aggregate 1.8.0
[2] From aggregate 1.9.0

The reinsurer's payment under aad and agglimit

The reinsurer's payment $A(R)$ is computed below for an example book with ten claims a year on average, from a lognormal severity with mean 100 and coefficient of variation 2, limited to 1,000. The book cedes a 750 xs 250 occurrence layer under each combination of the annual terms. Each ceded aggregate is the distribution of $A(R)$. The attribute `agg_reins` lists its aggregate stage as (share, limit, attachment).

```
book = 'agg {} 10 claims 1000 xs 0 sev lognorm 100 cv 2 occurrence {} poisson'
ceded = {t: build(book.format('Ceded', f'ceded to 750 xs 250 {t}'))
         for t in ['', 'aad 50', 'agglimit 500', 'aad 50 agglimit 500']}
df = pd.DataFrame.from_dict(
    {t or 'none': [str(c.agg_reins), c.est_m, c.est_sd,
                   c.density_df.p_total.iloc[0]] for t, c in ceded.items()},
    orient='index', columns=['agg_reins', 'mean', 'SD', 'Pr(A = 0)'])
df.index.name = 'terms'
GT(df)
```

terms	agg_reins	mean	SD	Pr(A = 0)
none	None	199.0162	300.7350	0.4173
aad 50	[(1.0, inf, 50.0)]	172.2558	286.0968	0.5087
agglimit 500	[(1.0, 500.0, 0.0)]	152.5279	190.7864	0.4173
aad 50 agglimit 500	[(1.0, 500.0, 50.0)]	133.0354	187.0483	0.5087

The deductible raises the probability of no recovery from 0.417 to 0.509, and the limit cuts the standard deviation (SD) from 301 to 191. The aggregate stage is exactly the map A applied to the annual recovery of the layer. Pushing the distribution of R, the first row of the table, through A with $d = 50$ and $m = 500$ reproduces the mean of the last row, 133.035.

```
plain, both = ceded[''], ceded['aad 50 agglimit 500']
r = plain.density_df
A = np.minimum(np.maximum(r.loss - 50, 0), 500)
print(f'{(A * r.p_total).sum():.6f} {both.est_m:.6f}')
```

```
133.035368 133.035368
```

Under a 90% share the aggregate stage is 450 xs 45 on the shared recovery, *sm* xs *sd* with $s = 0.9$. The reinsurance description lists the two stages in order.

```
c = build(book.format('Share',
                      'ceded to 90% po 750 xs 250 aad 50 agglimit 500'))
print(c.agg_reins)
wprint(c.reins_description)
```

```
[(1.0, 450.0, 45.0)]
Ceded to 90% share of 750 xs 250 per occurrence then ceded to 100% share of 450
xs 45 in the aggregate.
```

The cedant's net under aad and agglimit

A plain `agg` cannot compute the cedant's net under the annual terms. After `occurrence net of` it drops the annual terms (and the deposit) with an `IgnoredDecLClauseWarning` and returns the net

of the plain layer, $L - R$: the gross mean of 952.70 less the layer mean of 199.02. The stored program keeps the full declaration.

```
with warnings.catch_warnings(record=True) as rec:
    warnings.simplefilter('always')
    n = build(book.format(
        'Net', 'net of 750 xs 250 aad 50 agglimit 500 deposit 150'))
gross = n.reins_summary_df.loc[('occ', 'gross', 'agg'), 'EX']
print(rec[0].category.__name__)
print(f'net {n.est_m:.2f} = gross {gross:.2f} less layer {plain.est_m:.2f}')
print(n.pprogram)
```

```
IgnoredDecLClauseWarning
net 753.68 = gross 952.70 less layer 199.02
agg Net
  10 claims
  1000 xs 0
  sev lognorm 100 cv 2
  occurrence net of
    750 xs 250 aad 50 agglimit 500 deposit 150
  poisson
```

A `pnl` or `xpnl` that refers to the stored program as `agg.Net` books the annual terms. The `xpnl` has one step per reinsurance tier, and the step for the layer books its premium and the recovery $A(R)$. The table lists the obligations with the cedant's signs: a loss is negative, a recovery positive.

```
x = build('xpnl NetX 1500 premium less agg.Net')
e = x.economic_df.xs('Obligation', level='Side')[['EX', 'SD']]
e.index = pd.Index(['gross loss', 'layer recovery', 'net loss'], name='leg')
GT(e, float_format=',.2f')
```

leg	EX	SD
gross loss	-952.70	551.50
layer recovery	133.04	187.05
net loss	-819.66	416.48

The recovery, 133.043, agrees with the ceded mean of 133.035 in the first table to the accuracy of the grid of the joint distribution. The net loss, 819.66, is the gross loss less the recovery. The consolidated `pnl` gives the same net margin, 530.34.

```
p = build('pnl NetP 1500 premium less agg.Net')
print(f'recovery {e.iloc[1, 0]:.4f} against {both.est_m:.4f}')
print(f'net margin: xpnl {x.est_m:.4f}, pnl {p.est_m:.4f}')
```

```
recovery 133.0430 against 133.0354
net margin: xpnl 530.3440, pnl 530.3440
```

An aggregate limit of two occurrence limits is the same contract as one free reinstatement: under both the cedant recovers $\min(R, 1500)$. The two accounts agree, with net margin 595.50 and SD 340.35.

```
for layer in ['agglimit 1500 deposit 150', 'deposit 150 reinstatements 1 free']:
    q = build('xpnl R 1500 premium less '
              + book.format('RB', f'net of 750 xs 250 {layer}'))
    print(f'{layer:36s} {q.est_m:.4f} {q.est_sd:.4f}')
```

```
agglimit 1500 deposit 150            595.5028 340.3454
deposit 150 reinstatements 1 free    595.5028 340.3454
```

Refused programs

The annual terms apply to a single occurrence layer, never to an aggregate layer, and not together with reinstatements; the deductible must be non-negative and the limit positive and finite. On `occurrence ceded to` the annual terms become the aggregate stage, so the program takes no `aggregate` clause.

```
pre = 'agg T 10 claims 1000 xs 0 sev lognorm 100 cv 2 '
refuse(*[pre + s for s in [
    'poisson aggregate net of 750 xs 250 aad 50',
    'occurrence net of 750 xs 250 aad 50 and 1000 xs 1000 poisson',
    'occurrence net of 750 xs 250 aad 50 deposit 100 '
    'reinstatements 1 free poisson',
    'occurrence net of 750 xs 250 aad -5 poisson',
    'occurrence net of 750 xs 250 agglimit 0 poisson',
    'occurrence ceded to 750 xs 250 aad 50 poisson '
    'aggregate ceded to 300 xs 0']])
```

```
DecL: 'aad' / 'agglimit' decorate an occurrence layer, not aggregate
reinsurance. An aggregate layer is itself the annual deductible and limit: write
'm xs d'.
DecL: 'aad' / 'agglimit' require a single occurrence layer; found 2. The annual
terms cannot share the occurrence tier with other layers (the engine ceiling is
the 2-D (L, R) joint).
DecL: 'aad' / 'agglimit' cannot combine with 'reinstatements' on the same layer;
both set the layer's annual recovery. Write the annual limit one way or the
other.
DecL: 'aad' must be finite and >= 0, got -5.
DecL: 'agglimit' must be finite and > 0, got 0. Omit the clause for an unlimited
annual recovery.
T: annual terms (aad / agglimit) on an 'occurrence ceded to' program become its
aggregate stage, so the program cannot also carry an 'aggregate' clause. Write
the annual terms into the aggregate clause instead: 'aggregate ceded to inf xs
50' at this share, composed with the existing aggregate layers.
```