Using `elt`

Aggregates from event loss tables with the elt clause

Stephen J. Mildenhall

2026-11-07

An event loss table (ELT), the usual output of a catastrophe model, lists events e, each with an annual rate λ_e and a loss X_e given that the event occurs. Each row contains the mean loss μ_e and optionally a maximum M_e, an independent standard deviation (SD) σ_I, and a correlated SD σ_C. **Secondary uncertainty** is the variability of X_e about μ_e, and the two SDs measure it. Events arrive as independent Poisson processes, so the annual loss is compound Poisson with expected count $\Lambda = \sum_e \lambda_e$ and severity the mixture of the X_e with weights λ_e/Λ. Its cumulants are

$$\kappa_r = \sum_e \lambda_e \mathsf{P}X_e^r,$$

and without secondary uncertainty $X_e = \mu_e$ is an atom.

The `elt` clause takes the place of the exposure, severity and frequency clauses of an `agg`, and occurrence and aggregate reinsurance follow as usual. A row is `[id rate mean]` (ELT3), `[id rate mean max]` (ELT4), or `[id rate mean max sdi sdc]` (ELT6). An identifier is an unsigned integer or a quoted string, compared exactly, so `7` and `"7"` are one event and `"007"` another. Each `elt` clause is one source, and `unit NAME [as LABEL]` assigns it to a unit (`Total` when omitted). Rows sharing an identifier across units are one occurrence.

The `elt` clause[1] builds an aggregate from an inline table, as in the programs below. Tables read from CSV and Parquet files are in Using elt with files, and year loss tables (YELT), which list simulated annual losses rather than events, are in Using yelt.

```
agg Basic elt [[1 .1 100] [2 .1 40] [3 .1 80]]
agg Shared elt unit Personal as "Personal lines" [[1 .1 300] [2 .1 100]]
           elt unit Commercial [[1 .1 200] [3 .1 200]]
           occurrence net of 300 xs 200
agg Cat elt no2u [[1 .1 300 3000 400 300] [2 .1 100 1000 371 267]]
```

```
from aggregate import build, qd, __version__
from style import wprint, GT, refuse, figures
figures()
print(__version__)
```

```
2.1.0a6
```

[1] From aggregate 1.12.0

An event loss table of three events

The aggregate `Basic` has three events, each with rate 0.1, so $\Lambda = 0.3$. The mean annual loss is $0.1(100 + 40 + 80) = 22$ and the variance is $0.1(100^2 + 40^2 + 80^2) = 1{,}800$, an SD of 42.43. The severity is three equally weighted atoms, at 40, 80, and 100.

```
a = build('agg Basic elt [[1 .1 100] [2 .1 40] [3 .1 80]]')
qd(a)
```

```
Aggregate object: Basic. Frequency distribution poisson. Severity
dhistogram, [40, 100], bounded; atoms [40 80 100]. Updated with bucket
size 20 and log2 = 5. Validation: not unreasonable.

       Mean      SD      CV    Skew P01 Median P99
X
Freq    0.3 0.54772  1.8257  1.8257
Sev  73.333  24.944 0.34015 -0.3818  40     80 100
Agg      22  42.426  1.9285  2.0637   0      0 180
```

Reports on a loss table aggregate

An aggregate built from a loss table has three reports: `elt_description`, `elt_explanation`, and `elt_df`. The first, `elt_description`, is a few aligned lines: the table, the event rate and mean annual loss, the grid, and the checks counted by status.

```
wprint(a.elt_description)
```

```
loss table              ELT3, 3 events from 3 rows, 1 unit
event rate              0.3, mean annual loss 22
grid                    bs 20, log2 5
checks                  13 pass
```

The frame `elt_df` is the table the model uses, one row per event with a positive rate. The `units` column counts the input rows behind each event.

```
GT(a.elt_df, float_format=',.4g')
```

event	rate	mean	units
1	0.1	100	1
2	0.1	40	1
3	0.1	80	1

The string `elt_explanation` is three paragraphs: the model in words, each source with its provenance, and every check with its value, tolerance, and status, warnings first. The checks compare the moments of the table, the model, and the discretized distribution, and all of them pass here.

```
wprint(a.elt_explanation)
```

```
The annual loss is compound Poisson with event rate 0.3 and severity the
rate-weighted mixture of the event totals, so the mean annual loss is 22. An
event's unit losses are summed into one event total before any occurrence
treaty, and equal totals are combined: the 3 events with a positive rate give 3
severity atoms.

Source 1, unit Total: inline, ELT3, 3 rows.

13 checks: 13 pass. Form projection: pass, 0; 0 ELT6 source(s) read as ELT4
under no2u; combined form ELT3. One row per event and unit: pass, 0 against 0;
no event repeats within a unit. Event rate agreement: pass, 0 against 1e-12;
unit rows whose rate differs from the first row of the event, within the
relative tolerance. Zero-rate events: pass, 0; excluded from the count and the
moments. Severity mass: pass, 0 against 1e-12; lowered severity weight against
the event rate. Severity mean, table against model: pass, 0 against 0.0001;
independent sums before and after coalescing. Annual mean, table against model:
pass, 2.22e-16 against 0.0001; independent sums before and after coalescing.
Severity mean estimate: pass, 0 against 0.0001; discretized density against the
table. Severity cv estimate: pass, 0 against 0.0001; discretized density against
the table. Severity skew estimate: pass, -3.93e-14 against 0.0001; discretized
density against the table. Annual mean estimate: pass, -1.07e-09 against 0.0001;
discretized density against the table. Annual cv estimate: pass, -7.31e-09
against 0.0001; discretized density against the table. Annual skew estimate:
pass, -1.59e-07 against 0.0001; discretized density against the table.
```

Units and shared events

The aggregate `Shared` has two units, Personal (labeled "Personal lines") and Commercial, and event 1 has a row in each. Its losses add to one occurrence of 500 before the occurrence treaty 300 xs 200 applies, and `elt_df` lists the event once, with `units` 2. The mean annual loss is $0.1(500+100+200) = 80$.

```
s = build('''agg Shared elt unit Personal as "Personal lines" [[1 .1 300] [2 .1 100]]
                        elt unit Commercial [[1 .1 200] [3 .1 200]]
                        occurrence net of 300 xs 200''')
wprint(s.elt_description)
GT(s.elt_df, float_format=',.4g')
```

```
loss table              ELT3, 3 events from 4 rows, 2 units: Personal (Personal
                        lines), Commercial
event rate              0.3, mean annual loss 80
grid                    bs 100, log2 5
checks                  13 pass
```

event	rate	mean	units
1	0.1	500	2
2	0.1	100	1
3	0.1	200	1

Giving the Commercial row of event 1 a new identifier, 4, makes two occurrences, of 300 and 200, in place of one of 500. The gross mean stays 80, and the net mean moves from 50 to 70: the treaty cedes 100 of the 300 rather than 300 of the 500.

```
t = build('''agg Separate elt unit Personal [[1 .1 300] [2 .1 100]]
                          elt unit Commercial [[4 .1 200] [3 .1 200]]
                          occurrence net of 300 xs 200''')
print(f'{s.est_m:.2f} {t.est_m:.2f}')
```

```
50.00 70.00
```

Secondary uncertainty and the no2u flag

With secondary uncertainty, an ELT6 row makes X_e a beta distribution on $[0, M_e]$ with mean μ_e and SD $\sigma_I + \sigma_C$. When the variance $(\sigma_I + \sigma_C)^2$ is inadmissible for a law on $[0, M_e]$ with mean μ_e, that is at least $\mu_e(M_e - \mu_e)$, the row becomes the two-point law on $\{0, M_e\}$ with the same mean. The eltq package (`pip install aggregate[elt]`) computes the moments and quantizes the event laws on the grid. For example, the row `[1 .1 50 100 10 10]` defines a beta(2.625, 2.625) severity on $[0, 100]$, with annual mean 5 and variance $0.1(20^2 + 50^2) = 290$, an SD of 17.03.

```
b = build('agg Beta elt [[1 .1 50 100 10 10]]')
qd(b)
wprint(b.elt_description)
```

```
Aggregate object: Beta. Frequency distribution poisson. Severity elt,
[0, 100], bounded. Updated with bucket size 1/256 and log2 = 16.
Validation: not unreasonable.

      Mean      SD     CV       Skew    P01 Median    P99
X
Freq   0.1 0.31623 3.1623     3.1623
Sev     50      20    0.4 1.9824e-07 8.9336     50 91.066
Agg      5  17.029 3.4059     3.7458      0      0 80.055
loss table               ELT6, 1 event from 1 row, 1 unit
event rate               0.1, mean annual loss 5
secondary uncertainty    1 beta, 0 two-point, 0 atom rows
grid                     bs 1/256, log2 16, severity edge 255.998
checks                   19 pass
```

The severity functions of such an aggregate, `b.sev.sf`, `cdf`, `pdf`, `ppf`, and `isf`, are exact: eltq computes them from the rows, not from the grid.[2] The 99th percentile `b.sev.ppf(0.99)` of the beta(2.625, 2.625) law on $[0, 100]$ is 91.0665, and the quantile `q_sev(0.99)` on the grid, with bucket size 1/256, is 91.0664. Occurrence PMLs follow from them with `oep` (see Using oep).

```
print(f'{b.sev.ppf(0.99):.6f} {b.q_sev(0.99):.6f} {b.bs}')
```

```
91.066484 91.066406 0.00390625
```

The flag `no2u` (no secondary uncertainty) replaces each ELT6 row by an atom at its mean, which leaves the mean annual loss unchanged. The table below has one two-point row (event 2) and three betas. Secondary uncertainty doubles the SD, from 152 to 306, and nearly triples the 99.9% quantile, from 1,000 to 2,976.

[2] From aggregate 2.1.0a6

```
table = '''[[1 .10 300 3000 400 300] [2 .10 100 1000 371 267]
           [3 .25  50 1000 100 100] [4 .05 500 1000 120 100]]'''
for flag in ['no2u', '']:
    c = build(f'agg Cat elt {flag} {table}')
    print(f'{flag or "2U":5s} {c.est_m:8.2f} {c.est_sd:8.2f}',
          f'{c.q(0.99):8.0f} {c.q(0.999):8.0f}')
wprint(c.elt_description)
```

```
no2u     77.50   152.07      600     1000
2U       77.50   305.85     1473     2976
loss table               ELT6, 4 events from 4 rows, 1 unit
event rate               0.5, mean annual loss 77.5
secondary uncertainty    3 beta, 1 two-point, 0 atom rows
grid                     bs 1/2, log2 16, severity edge 32,768
checks                   19 pass
```

When an ELT6 event has rows in two units, the rows are rolled up into one before modeling: means, maximums and correlated SDs add, and independent SDs add in quadrature. In the aggregate `Rolled` below, event 1 has maximum $3{,}000 + 1{,}000 = 4{,}000$, independent SD $(400^2 + 300^2)^{1/2} = 500$ and correlated SD $300 + 100 = 400$.

```
r = build('''agg Rolled elt unit Personal [[1 .1 300 3000 400 300] [2 .1 100 1000 50 50]]
                        elt unit Commercial [[1 .1 200 1000 300 100]]''')
GT(r.elt_df, float_format=',.4g')
```

event	rate	mean	max	sdi	sdc	units
1	0.1	500	4,000	500	400	2
2	0.1	100	1,000	50	50	1

Refused programs

An event repeated within a unit, an event with different rates in different units, an ELT6 with secondary uncertainty beside a mean-only source, and a mean above its maximum each raise `LossTableError`, a subclass of `ValueError`. A row of the wrong length, and a loss table combined with `claims`, `sev`, `dsev`, or `approximate`, are refused when the program is parsed.

```
refuse('agg E1 elt [[1 .1 100] [1 .1 40]]',
       'agg E2 elt unit A [[1 .1 100]] elt unit B [[1 .2 40]]',
       'agg E3 elt [[1 .1 100 1000 10 10]] elt unit B [[2 .2 40]]',
       'agg E4 elt [[1 .1 100 50]]')
```

```
E1: an event appears more than once for one unit: elt source 1 (unit Total) row
1 (event 1), elt source 1 (unit Total) row 2 (event 1).
E2: an event's unit rows have different rates (an event has one rate, whatever
its units): elt source 2 (unit B) row 1 (event 1).
E3: an ELT6 source with secondary uncertainty cannot be combined with ELT3 or
ELT4 sources, including an ELT6 converted by no2u. Add no2u to elt source 1
(unit Total) for the mean-only model.
E4: elt source 1 (unit Total): mean exceeds max; see row 1 (id 1).
```