Using `yelt`

Aggregates from year loss tables with the yelt clause

Stephen J. Mildenhall

2026-10-30

A year loss table (YELT) lists simulated years y, each with its annual loss s_y and optionally a probability p_y, otherwise $1/n$ for n years. Each row is already a whole annual outcome, so there is no compounding: the aggregate is the discrete law $\Pr(S = s) = \sum_y p_y 1\{s_y = s\}$, built as `dfreq [1] dsev` over the annual losses, with population moments $\mathsf{P}S^r = \sum_y p_y s_y^r$.

A row of a `yelt` table is a loss alone (YELT1, a flat vector, years numbered by position), `[year loss]` (YELT2), or `[year loss prob]` (YELT3). Zero-loss years must be listed, since they set the probability of no loss. Supplied probabilities must sum to one. Units, declared with `unit NAME` as in Using elt, are aligned by year and summed within each year, which keeps whatever dependence the simulation produced; every unit must supply the same complete set of years. An annual loss does not identify its occurrences, so a YELT takes aggregate reinsurance only.

The DecL clause `yelt`[1] builds an aggregate from a YELT. The reports `elt_description`, `elt_explanation`, and `elt_df`, described in Using elt, apply to a YELT too. File columns are matched by name as in Using elt with files: `year`, `trial` or `simulation_id` for the year, `loss` or `annual_loss` for the loss, and `prob` or `probability` for the probability.

```
agg Annual yelt [0 0 100 300]                        # YELT1: losses only
agg Paired yelt [[1 0] [2 0] [3 100] [4 300]]        # YELT2: [year loss]
agg Weighted yelt [[1 0 .5] [2 100 .3] [3 300 .2]] aggregate net of 150 xs 50
agg Sim yelt "sim.parquet"
```

```
import tempfile
from pathlib import Path
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
```

A year loss table of four equally likely years

The table `yelt [0 0 100 300]` lists four years, each with probability 1/4, and gives the same aggregate as `dfreq [1] dsev [0 0 100 300]`. The probability of zero is one half, the mean is 100, and the variance is $(100^2 + 300^2)/4 - 100^2 = 15{,}000$, the population moment rather than the sample variance of 20,000.

[1] From aggregate 1.20.0

```
a = build('agg Annual yelt [0 0 100 300]')
qd(a)
b = build('agg Same dfreq [1] dsev [0 0 100 300]')
print(a.pmf(0), a.est_m, a.est_var, b.est_m, b.est_var)
```

```
Aggregate object: Annual. Frequency distribution empirical. Severity
dhistogram, [0, 300], bounded; atoms [0 100 300]. Updated with bucket
size 100 and log2 = 2. Validation: not unreasonable.

      Mean     SD     CV   Skew P01 Median P99
X
Freq     1      0      0
Sev    100 122.47 1.2247 0.8165   0      0 300
Agg    100 122.47 1.2247 0.8165   0      0 300
0.5 100.0 15000.0 100.0 15000.0
```

Supplied probabilities and an aggregate treaty

The aggregate `Weighted` is a YELT3 whose three years have probabilities 0.5, 0.3, and 0.2. Its `elt_df` lists each year of positive probability with `prob` and `loss`. The treaty `aggregate net of 150 xs 50` takes the outcomes 0, 100, and 300 to 0, 50, and 150, so the net mean is $0.3(50) + 0.2(150) = 45$ against a gross mean of 90.

```
w = build('agg Weighted yelt [[1 0 .5] [2 100 .3] [3 300 .2]] aggregate net of 150 xs 50')
wprint(w.elt_description)
display(GT(w.elt_df, float_format=',.4g'))
print(w.est_m)
```

```
loss table              YELT3, 3 years, 1 unit, supplied year probabilities
annual outcomes         3 outcomes, probability of zero 0.5, mean annual loss
                        90
grid                    bs 100, log2 2
checks                  11 pass
```

year	prob	loss
1	0.5	0
2	0.3	100
3	0.2	300

```
45.0
```

Units aligned by year

The aggregate `Years` has two units, A and B, and the table of unit B lists its years in a different order from unit A. The rows are matched by year and summed, so year 3 is $300 + 50 = 350$. The treaty `aggregate net of 150 xs 50` takes the totals 10, 100, and 350 to 10, 50, and 200, a net mean of 60.

```
y = build('''agg Years yelt unit A [[1 0 .5] [2 100 .3] [3 300 .2]]
                        yelt unit B [[3 50 .2] [1 10 .5] [2 0 .3]]
                        aggregate net of 150 xs 50''')
display(GT(y.elt_df, float_format=',.4g'))
print(y.est_m)
```

year	prob	loss
1	0.5	10
2	0.3	100
3	0.2	350

```
60.0
```

Two units from one CSV file

A long CSV file with a `Unit` column can hold several units. The file here holds the two units of the aggregate `Years`, under the headers `Trial`, `Annual Loss`, and `Probability`. The code writes it to a temporary directory and builds the aggregate there; the directory is removed when the `with` block ends, and the aggregate keeps what was read. The net mean is 60, the same as for the inline table. The provenance paragraph of `elt_explanation` lists the path, size, and SHA-256 digest of the file.

```
years = '''Trial,Unit,Annual Loss,Probability
1,A,0,.5
2,A,100,.3
3,A,300,.2
3,B,50,.2
1,B,10,.5
2,B,0,.3
'''
with tempfile.TemporaryDirectory() as d:
    path = Path(d, 'years.csv')
    path.write_text(years)
    f = build(f'agg YearsFile yelt "{path.as_posix()}" aggregate net of 150 xs 50')
print(f.est_m, y.est_m)
wprint(f.elt_explanation.split('\n\n')[1])
```

```
60.0 60.0
Source 1, unit A: file <tmp>/tmp89r4hn72/years.csv (102 bytes, modified
2026-10-30T12:37:20, SHA-256 ab4649389ba4), YELT3, 3 rows. Source 1, unit B:
file <tmp>/tmp89r4hn72/years.csv (102 bytes, modified 2026-10-30T12:37:20,
SHA-256 ab4649389ba4), YELT3, 3 rows.
```

Ten thousand simulated years from a Parquet file

The usual source of a YELT is the output of a simulation. Here a loss occurs in about 30% of 10,000 simulated years, lognormal when it occurs, and the years are saved to a Parquet file with `year` and `loss` columns. The aggregate is the empirical law of the years, so under `aggregate net of 1000 xs 500` its gross and net means equal the sample means of the gross and net losses, 47.87 and 39.69.

```
rng = np.random.default_rng(2026)
n = 10_000
sim = pd.DataFrame({'year': np.arange(1, n + 1),
                    'loss': np.round(rng.lognormal(4, 1.5, n) * (rng.random(n) < 0.3))})
with tempfile.TemporaryDirectory() as d:
    path = Path(d, 'sim.parquet')
    sim.to_parquet(path, index=False)
    s = build(f'agg Sim yelt "{path.as_posix()}" aggregate net of 1000 xs 500')
wprint(s.elt_description)
net = sim.loss - (sim.loss - 500).clip(0, 1000)
print(f'{sim.loss.mean():.4f} {net.mean():.4f} {s.est_m:.4f}')
```

```
loss table               YELT2, 10,000 years, 1 unit, equally likely years
annual outcomes          577 outcomes, probability of zero 0.6982, mean annual
                         loss 47.8732
grid                     bs 1, log2 14
checks                   11 pass
47.8732 39.6932 39.6932
```

Refused programs

Occurrence terms on a YELT are refused, because an annual loss does not identify its occurrences. Probabilities that do not sum to one, a unit missing a year, a year listed twice, and two units giving a year different probabilities are refused too. Each refusal is a `ValueError`, and the table checks raise its subclass `LossTableError`.

```
refuse('agg R1 yelt [0 0 100 300] occurrence net of 50 xs 50',
       'agg R2 yelt [[1 0 .5] [2 100 .3] [3 300 .1]]',
       'agg R3 yelt unit A [0 100 300] yelt unit B [10 20]',
       'agg R4 yelt [[1 0] [1 100] [2 300]]',
       'agg R5 yelt unit A [[1 0 .5] [2 100 .5]] yelt unit B [[1 0 .4] [2 100 .6]]')
```

```
DecL: a yelt holds annual outcomes, so occurrence reinsurance cannot apply: an
annual total does not identify its occurrences. Write annual terms as 'aggregate
net of ...', or declare the events with elt for occurrence terms.
R2: yelt source 1 (unit Total): the probabilities sum to 0.9, not one. Each row
is a year with its probability; counts, rates and weights are not probabilities.
R3: every unit supplies the same complete set of years, zero-loss years
included; yelt source 2 (unit B) has 2 years against 3 in yelt source 1 (unit A)
(missing years 3).
R4: yelt source 1 (unit Total): a year appears more than once; see row 1 (id 1),
row 2 (id 1).
R5: yelt source 2 (unit B) gives a year a different probability from yelt source
1 (unit A); a year has one probability, whatever its units.
```