Using `elt` with files

Event loss tables read from CSV and Parquet files with the elt clause

Stephen J. Mildenhall

2026-10-30

An event loss table (ELT) from a catastrophe model usually arrives as a file. The `elt` clause takes a quoted path to one[1] in place of an inline table. The model is the one in Using elt: the annual loss is compound Poisson with expected count $\Lambda = \sum_e \lambda_e$, where λ_e is the rate of event e, and severity the mixture of the event losses with weights λ_e/Λ. Rows sharing an event identifier across units are one occurrence.

CSV, TSV, Parquet, and Feather files are read; the last two need `pyarrow`, and Excel is not read. Columns are matched by name after ignoring case, spaces, and underscores, through one alias table: `Event ID` is `event_id`, a rate may be `rate`, `frequency`, `freq`, or `annual_rate`, and the mean loss (`mean`, `mean_loss`, or `loss`) is the mean loss given the event, not the rate times it. A `unit` column splits one long file into units. The columns present set the form: ELT3 (identifier, rate, and mean loss), ELT4 (adding a maximum), or ELT6 (adding the independent and correlated standard deviations). Any other columns are ignored and listed. Identifiers are read as text, so they stay exact. A relative path resolves against the directory of the declaring `.agg` file, or the working directory when the statement is built.

The file is read once, when the aggregate is built, and its resolved path, size, modification time, and SHA-256 digest are recorded. Year loss tables (YELT) are read from files in the same way, as Using yelt shows.

```
agg Cat elt "cat.csv"                                  # units from a unit column
agg Split elt unit Personal "personal.parquet"
          elt unit Commercial "commercial.parquet"
agg Net elt "cat.csv" occurrence net of 300 xs 200
```

```
import io
import tempfile
from pathlib import Path
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 1.13.0

A long CSV file with a unit column

The CSV file below has one row per event and unit, headers as a vendor might write them, and an extra `Region` column. The code writes it to a temporary directory and builds the aggregate there, and the directory is removed when the `with` block ends.

```
csv = '''Event ID,Unit,Rate,Mean Loss,Max Loss,Region
1,Personal,0.10,300,3000,FL
2,Personal,0.10,100,1000,TX
1,Commercial,0.10,200,1000,FL
3,Commercial,0.25,50,1000,CA
'''
with tempfile.TemporaryDirectory() as d:
    path = Path(d, 'cat.csv')
    path.write_text(csv)
    a = build(f'agg CatFile elt "{path.as_posix()}" occurrence net of 300 xs 200')
qd(a)
```

```
Aggregate object: CatFile. Frequency distribution poisson. Severity
dhistogram, [50, 500], bounded; atoms [50 100 500]. Updated with bucket
size 50 and log2 = 6. Validation: reinsurance; subject not unreasonable.

       Mean      SD      CV    Skew P01 Median P99
X
Freq   0.45 0.67082  1.4907  1.4907
Sev  94.444  59.835 0.63355 0.99251  50     50 200
Agg    42.5      75  1.7647  2.2074   0      0 300
```

Event 1 has a row in each unit, so it is one occurrence of 500 with maximum 4,000. The table is ELT4, with a gross mean annual loss of $0.1(500+100)+0.25(50)=72.5$. The occurrence treaty 300 xs 200 cedes 300 of the 500, for a net mean of 42.5.

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

```
loss table              ELT4, 3 events from 4 rows, 2 units: Personal,
                        Commercial
event rate              0.45, mean annual loss 72.5
grid                    bs 50, log2 6
checks                  13 pass
```

event	rate	mean	max	units
1	0.1	500	4,000	2
2	0.1	100	1,000	1
3	0.25	50	1,000	1

Provenance of a file source

The second paragraph of `elt_explanation` lists the provenance of each source: the resolved path, size, modification time, the leading digits of the SHA-256 digest, and the ignored columns. The temporary directory is already gone, and the aggregate is unaffected.

```
wprint(a.elt_explanation.split('\n\n')[1])
print(path.exists(), f'{a.est_m:.4f}')
```

```
Source 1, unit Personal: file <tmp>/tmpu61cer9t/cat.csv (165 bytes, modified
2026-10-30T12:29:29, SHA-256 f3df7660ae67), ELT4, 2 rows, ignoring columns
Region. Source 1, unit Commercial: file <tmp>/tmpu61cer9t/cat.csv (165 bytes,
modified 2026-10-30T12:29:29, SHA-256 f3df7660ae67), ELT4, 2 rows, ignoring
columns Region.
False 42.5000
```

The stored program holds the absolute path of the file, so a rebuild reads the file again. With the directory gone, the rebuild fails.

```
wprint(a.program)
refuse(a.program)
```

```
agg CatFile elt "<tmp>/tmpu61cer9t/cat.csv" occurrence net of 300 xs 200
elt file "<tmp>/tmpu61cer9t/cat.csv": cannot read <tmp>\tmpu61cer9t\cat.csv
([Errno 2] No such file or directory:
'C:\\Users\\steve\\AppData\\Local\\Temp\\tmpu61cer9t\\cat.csv').
```

One Parquet file per unit

Two Parquet files, one per unit, can hold the same four rows as the CSV file of the first example, without its `Unit` column. With one `elt unit` clause per file they give the same aggregate, a net mean of 42.5. In the Parquet files pandas stores the identifiers as integers, and the clause converts each to its decimal text, so the events are `'1'`, `'2'`, and `'3'`, the same as from the CSV file.

```
cat = pd.read_csv(io.StringIO(csv))
with tempfile.TemporaryDirectory() as d:
    sources = []
    for unit, rows in cat.groupby('Unit', sort=False):
        rows.drop(columns='Unit').to_parquet(Path(d, f'{unit}.parquet'), index=False)
        sources.append(f'elt unit {unit} "{Path(d, unit).as_posix()}.parquet"')
    b = build(f'agg CatParquet {" ".join(sources)} occurrence net of 300 xs 200')
wprint(b.elt_description)
print(f'{a.est_m:.4f} {b.est_m:.4f}', list(b.elt_df.index))
```

```
loss table              ELT4, 3 events from 4 rows, 2 units: Personal,
                        Commercial
event rate              0.45, mean annual loss 72.5
grid                    bs 50, log2 6
checks                  13 pass
42.5000 42.5000 ['1', '2', '3']
```

Refused files

A Parquet identifier column of floating point numbers is refused, since its digits may already be lost; in a CSV file the same digits are text and are kept as written. Two columns with one meaning, an Excel file, and a `unit` clause that does not match the unit column of the file (the clause does not select rows) each raise `LossTableError`, a subclass of `ValueError`.

```
with tempfile.TemporaryDirectory() as d:
    q = lambda name: f'"{Path(d, name).as_posix()}"'
    floats = pd.DataFrame({'id': [1.0, 2.0], 'rate': [.1, .1], 'mean': [100, 50]})
    floats.to_parquet(Path(d, 'float.parquet'))
    Path(d, 'twice.csv').write_text('id,rate,freq,mean\n1,.1,.1,100\n')
    Path(d, 'cat.csv').write_text(csv)
```

```
    refuse(f'agg F1 elt {q("float.parquet")}',
           f'agg F2 elt {q("twice.csv")}',
           f'agg F3 elt {q("cat.xlsx")}',
           f'agg F4 elt unit Personal {q("cat.csv")}')
```

```
elt file "<tmp>/tmpui1_wi9_/float.parquet": the identifier column holds floating
point numbers, which may already have lost digits; store it as integers or text.
elt file "<tmp>/tmpui1_wi9_/twice.csv": columns 'rate' and 'freq' both mean
rate.
elt file "<tmp>/tmpui1_wi9_/cat.xlsx": unsupported file type '.xlsx'; use .csv,
.tsv, .parquet, .feather.
elt file "<tmp>/tmpui1_wi9_/cat.csv": the clause says unit Personal but the unit
column also holds Commercial.
```