Using FFT windows

The FFT grid window chosen by the bucket sizer, shown by bs_window_df

Stephen J. Mildenhall

2026-10-30

An aggregate is computed on $n = 2^\ell$ buckets of size b, at the points $x_k = x_0 + kb$ for $k = 0, \dots, n-1$, so on the **window** $[x_0, x_0 + W)$ of width $W = nb$. In code the bucket size is `bs` and ℓ is `log2`. Convolution by FFT is circular: it returns the law of S reduced modulo W. Provided one severity fits in $[0, W)$, any window of width W that holds essentially all the mass therefore gives the right answer once the result is relabeled onto it, and the origin x_0 is free.

The bucket sizer uses that freedom. For an ordinary non-negative law $x_0 = 0$. A **concentrated** law, one whose coefficient of variation (CV) is below 0.1, has its mass far above zero, and the window is placed around it, so the same n gives a finer bucket. For a signed severity the window straddles zero and $x_0 < 0$. Nothing is declared: the sizer selects the window and `density_df` is indexed on it. The table `bs_window_df` lists the candidate windows, one row per sizing method, with the width `W` of each and the probability `coverage` it is sized to hold.

A concentrated law is computed on a window away from zero[1]. The attributes `bs_window_df`, `bs_description`, and `bs_explanation` describe the choice of window. The tail class of the severity also enters that choice, as described in Using tail classes. The examples are three aggregates, one of each kind.

```
agg Ord    10 claims sev gamma 50 cv 0.5 poisson       # window [0, W)
agg Conc   10000 claims sev gamma 10 cv 0.5 poisson    # concentrated, far from 0
agg Signed 10 claims ssev 50 * norm + 10 poisson       # signed, straddles 0
```

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

```
1.43.1
```

The window of an ordinary aggregate

For a non-negative aggregate that is not concentrated the window starts at 0. Each row of `bs_window_df` is a window $[x_{min}, x_{max}]$ with its bucket. The `moment` row is the window of a three-moment fit, `sbj` is the single big jump floor on the window, described in Using tail classes, and `used` is the realized window, of width $2^{16}/16 = 4{,}096$. The `selected` column marks the method whose bucket was taken.

[1] From aggregate 1.0

```
cols = ['applies', 'selected', 'x_min', 'x_max', 'W', 'bs', 'log2', 'coverage']
fmt = {'x_min': ',.0f', 'x_max': ',.0f', 'W': ',.0f', 'bs': '.4g'}
a = build('agg Ord 10 claims sev gamma 50 cv 0.5 poisson')
wprint(a.bs_description)
GT(a.bs_window_df[cols], formatters=fmt)
```

```
moment grid: bs=1/16, log2=16, window [0, 4,096] of width 4,096
```

method	applies	selected	x_min	x_max	W	bs	log2	coverage
moment	True	True	0	2,573	2,573	0.0625	16	1-1e-12
sbj	True	False	0	1,166	1,166	0.03125	16	E[N]-adj 1-1e-12
used	True	False	0	4,096	4,096	0.0625	16	1-1e-12

The window of a concentrated aggregate

The aggregate `Conc` has ten thousand claims, mean 100,000, and CV 0.011. The `moment` window $[0, 107{,}987]$ would need $b = 2$. The `windowed` method holds the mass in a band of width 13,248 and takes $b = 1/4$, eight times finer on the same n. The `used` row is that band padded out to 2^{16} buckets, with the slack split on either side. The coverage differs by edge. For a loss the right tail, which determines the risk measures, is covered to $1 - 10^{-12}$, and the left only to $1 - 10^{-6}$.

```
c = build('agg Conc 10000 claims sev gamma 10 cv 0.5 poisson')
wprint(c.bs_description)
GT(c.bs_window_df[cols], formatters=fmt)
```

```
windowed grid: bs=1/4, log2=16, window [93,484, 109,868] of width 16,384
```

method	applies	selected	x_min	x_max	W	bs	log2	coverage
moment	True	False	0	107,987	107,987	2	16	1-1e-12
windowed	True	True	94,739	107,987	13,248	0.25	16	lo 1-1e-6 / hi 1-1e-12
sbj	True	False	0	100,148	100,148	2	16	E[N]-adj 1-1e-12
used	True	False	93,484	109,868	16,384	0.25	16	lo 1-1e-6 / hi 1-1e-12

For a concentrated aggregate `density_df` and `xs` start at the window, not at 0, and quantiles and moments are computed from them as usual. The first buckets hold probabilities of order 10^{-12}. The median is 99,997.5, the mean is 100,000, and the aggregate passes validation.

```
print(c.xs[0], c.xs[-1], len(c.xs))
print(c.density_df[['p_total', 'F']].head(3))
print(c.q(0.5), c.est_m, c.validation_description)
```

```
93484.5 109868.25 65536
              p_total             F
loss
93484.50  2.478809e-12  2.478809e-12
93484.75  2.482170e-12  4.960979e-12
93485.00  2.485535e-12  7.446514e-12
99997.5 100000.00003263868 not unreasonable
```

The string `bs_explanation` gives the same choice in prose. It lists the width of each candidate window, the winning method, and the recommended window.

```
wprint(c.bs_explanation)
```

```
The aggregate is severity-driven, [0, inf), exponential right tail. Log2 is 16.
Of the methods that applied (moment 107,987, windowed 13,248, sbj 100,148), the
windowed method won (window width 13,248) -- a two-sided window far from 0
(benign FFT wrap). It has a natural lower support bound of 0. The distribution
is concentrated with a CV of 0.0111803. The recommended x_min is 93,484 and the
x_max that follows is 109,868, so the window is [93,484, 109,868] of width
16,384.
```

Passing `x_min=0` to `update` forces a window starting at 0. The sizer then takes the coarse `moment` bucket, $b = 2$, and the severity and aggregate CVs fail validation.

```
c0 = build('agg Conc 10000 claims sev gamma 10 cv 0.5 poisson')
c0.update(x_min=0)
wprint(c0.bs_description)
print(c0.validation_description)
```

```
moment grid: bs=2, log2=16, window [0, 131,072] of width 131,072
fails sev cv, agg cv
```

The window of a signed severity

An `ssev` severity takes both signs (Using `ssev`), so the `moment` window is two-sided and starts below zero, at -826.75. The `used` row keeps that left edge and pads the window to 2^{16} buckets of size $1/16$ on the right.

```
s = build('agg Signed 10 claims ssev 50 * norm + 10 poisson')
wprint(s.bs_description)
display(GT(s.bs_window_df[cols], formatters=fmt))
print(s.xs[0], s.xs[-1])
```

```
moment grid: bs=1/16, log2=16, window [-826.75, 3,269] of width 4,096
```

method	applies	selected	x_min	x_max	W	bs	log2	coverage
moment	True	True	-827	1,504	2,331	0.0625	16	1-1e-12
sbj	True	False	-267	467	735	0.01562	16	E[N]-adj 1-1e-12
used	True	False	-827	3,269	4,096	0.0625	16	1-1e-12

```
-826.75 3269.1875
```

cdf and sf outside the window

Below the window of a non-negative law that starts at 0, the method `cdf` returns 0 and the method `sf` returns 1. Below a window placed above the origin, or below the window of a signed law, the mass there was trimmed or never computed, so both are `nan`. Above the window both come from the mass on the grid, so far above it `sf` returns the shortfall. For the aggregates `Conc` and `Signed` the shortfall, 1.8×10^{-9} and 6.2×10^{-8}, is the left tail below the window. For `Signed` it exceeds the 10^{-12} in the `coverage` column, which is the target of the three-moment fit, not a measured probability.

```python
for ob, lo in [(a, -1), (c, 1000), (s, -2000)]:
    print(f'{ob.name:7s} cdf({lo}) = {ob.cdf(lo)}, sf({lo}) = {ob.sf(lo)}, '
          f'sf(1e6) = {ob.sf(1e6):.2e}')
```

```
Ord     cdf(-1) = 0.0, sf(-1) = 1.0, sf(1e6) = 2.12e-14
Conc    cdf(1000) = nan, sf(1000) = nan, sf(1e6) = 1.78e-09
Signed  cdf(-2000) = nan, sf(-2000) = nan, sf(1e6) = 6.22e-08
```