

Bayesian Modelling In Real Life

Developing a Bayesian model to solve a problem.



Lessons from the Trenches

07 October 2026

❖ We have a problem!

How I approached the problem

How I failed

How I came up with a better way

How I learned the best way

The academic approach was mathematically rigorous — but didn't deliver what the business needed.

Realised how to work starting from practice to get applicable usable results.

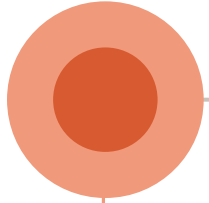
MARKEL



Today's Story

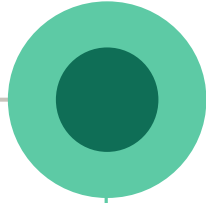
From challenge to insight — the journey of a Bayesian model

The Challenge



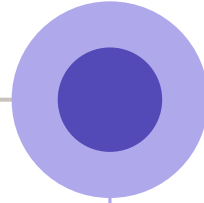
Forecast Policies in Force (PIF) using occurrence-based data
split by New Business and Renewals

Method



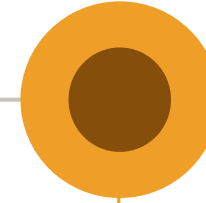
Bayesian hierarchical modelling in R (brms)
Multiple formula families tested

Data



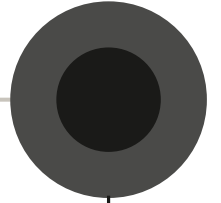
Occurrence-based, stepped, volatile, non-stationary
Regular spikes create modelling headaches

Key Insights



Compression is key — raw data resists mathematical fitting
Sigmoid (cumsum) far more tractable

Conclusion



Formula model overpredicts over time
Business-informed model is accurate & operationally grounded

The Goal: Forecasting Policies in Force

Data is drawn from an anonymised company segment — all identifying and quantified features have been removed.

- 1 The business question is simple: how many policies will be in force next year?
- 2 PIF is a valuable leading indicator — it informs exposure estimates and signals portfolio growth or contraction.
- 3 Policies split into two types: New Business (NB) and Renewals (R) — each with distinct dynamics.
- 4 Can we build a model that honours this structure and produces reliable, interpretable forecasts?

Key Takeaways

Approach 1: Formula-Driven

A1

- Sought the mathematically correct model from first principles
- Difficulties with raw occurrence data — volatile, noisy, non-stationary
- Tried cosine functions, splines, polynomials and development percentiles
- Result: usable but flawed — prone to overprediction over time

vs

Approach 2: Business-Informed

A2

- Asked how the segment actually works — not what the maths looks like
- Used operational knowledge as genuine Bayesian prior information
- Reframed: based on last year's data, what next year?
- Result: accurate, interpretable, operationally grounded

What Actually Drives the Forecast

↻ Prior year renewals

80–90% of the book

+ New business volume

Likely linked to total PIF

↗ YoY trend

Growth or contraction

🎯 Management intent

Grow, hold, or shrink

*The best model came from understanding the **business first**, the **mathematics second**.*

First Look:

The Raw Data

We start simply — plotting raw daily policy counts as a time series, year by year.

Each year looks something like this:

~

No clear trend

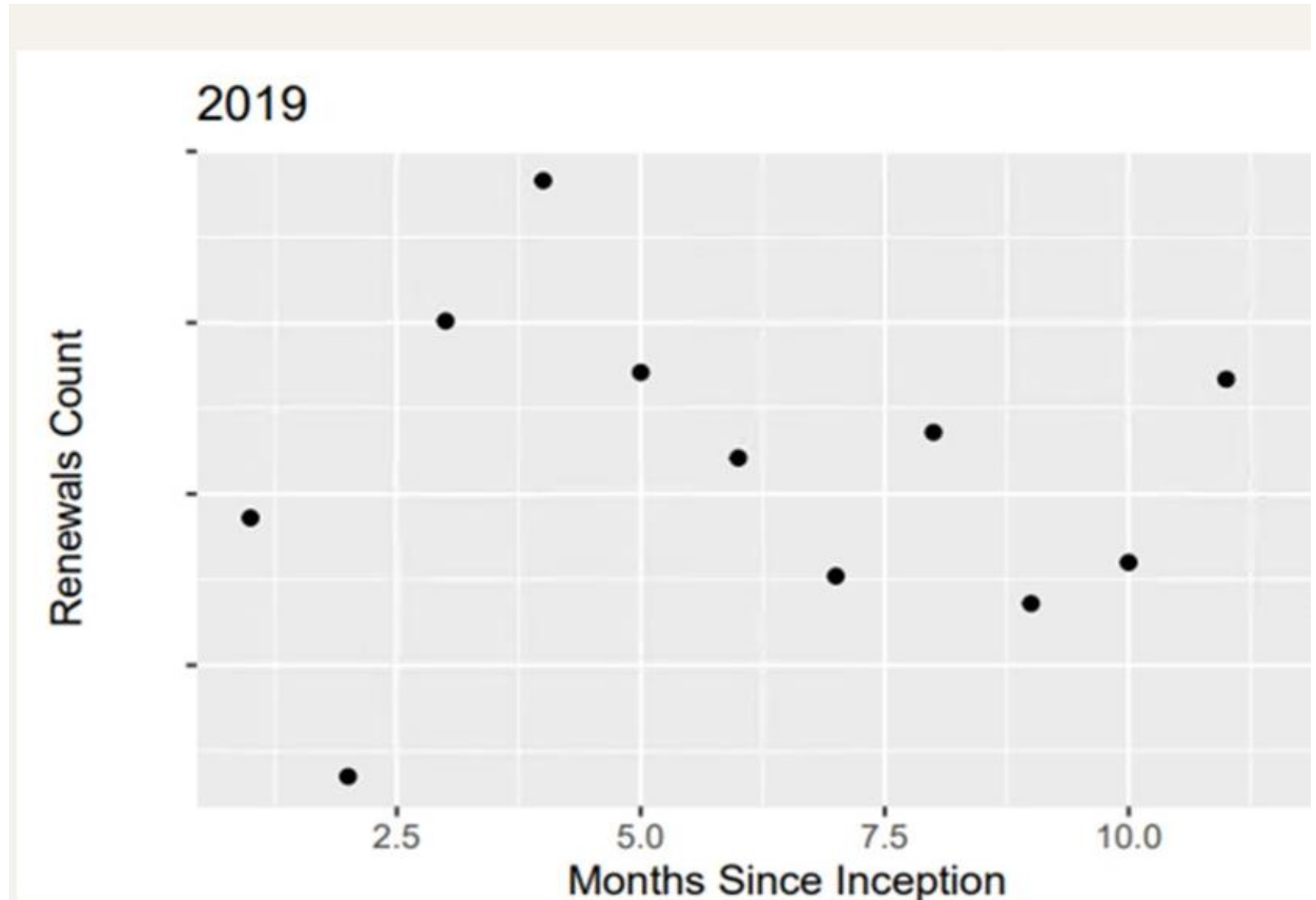
↕

High month-to-month variance

?

Seasonality unclear

Raw Policy Counts · 2019 · Monthly by Inception



Each dot = one month's renewal count. Spread is wide — motivating the need for a proper model.

The Periodicity Problem

Mean daily policy counts across all years
— regular, jagged spikes throughout.

This is the core modelling challenge.

~Monthly spikes

Recurring at regular intervals

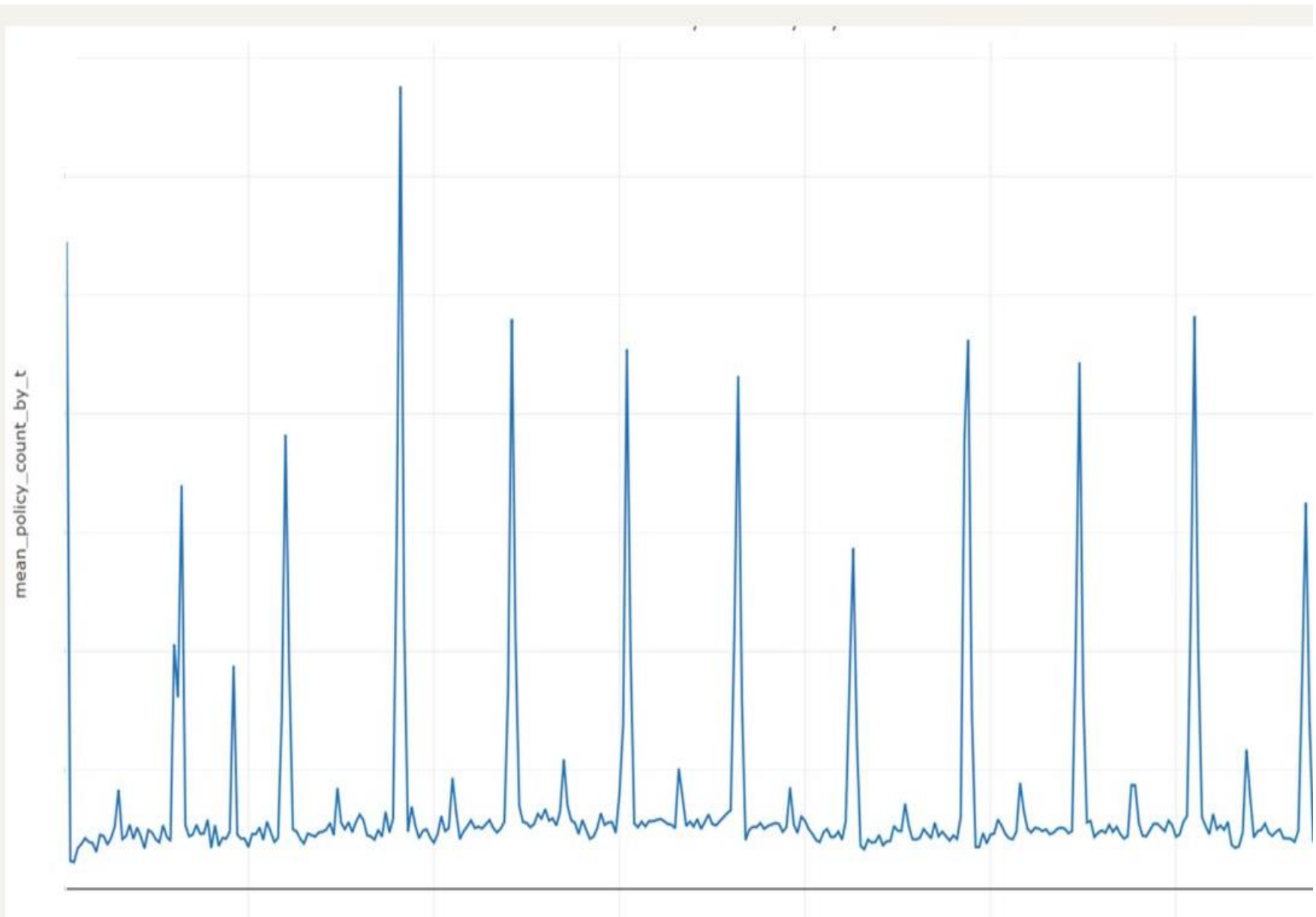
Baseline low noise

Quiet periods between spikes

Spike amplitude varies

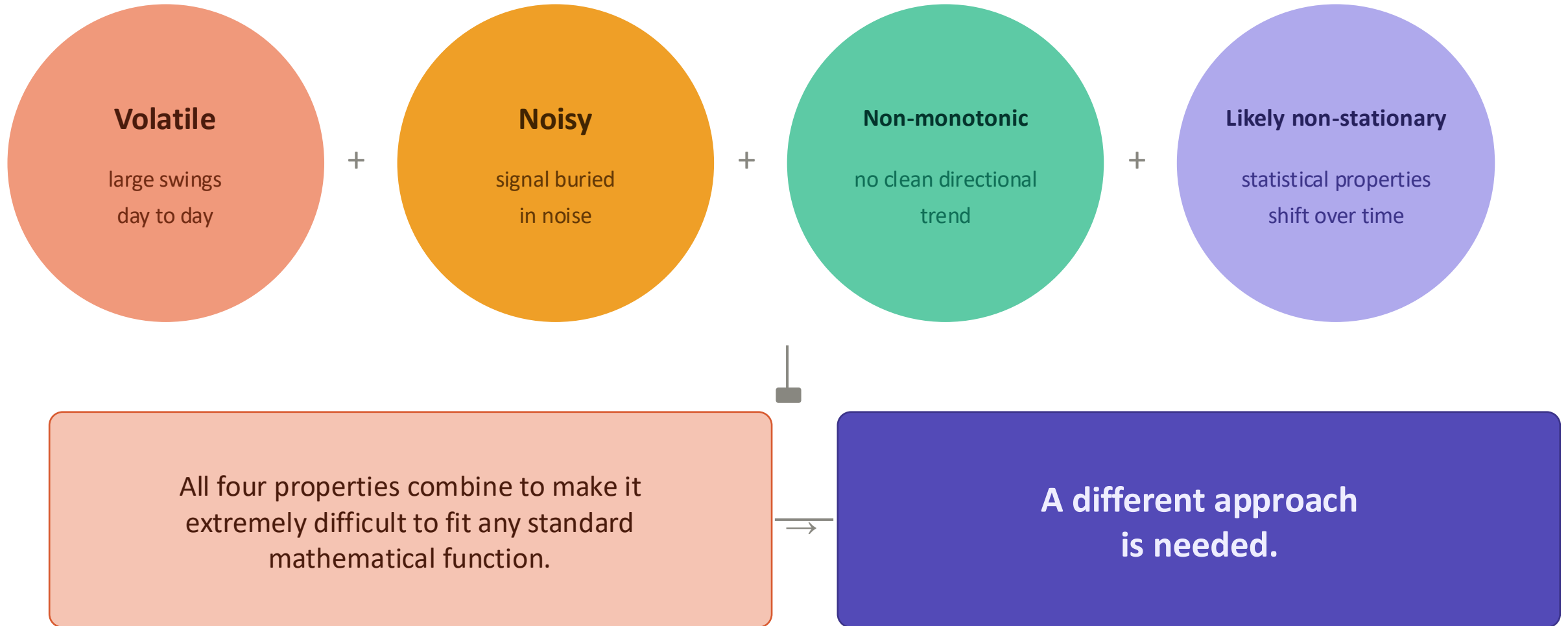
Not perfectly regular in height

Mean Daily Policy Counts · All Years · Day of Year (1–365)



Spikes repeat $\approx$ every 30 days — a clear monthly renewal pattern that any model must capture.

Why Raw Occurrence Data Is Hard to Model



The Insight

Model the Cumulative Sum

Practitioners quickly reach for the cumulative sum — and for good reason.

Taking it by timepoint forms a sigmoid curve that resolves nearly all modelling problems at once.

No free lunch: the cumsum trades short-term accuracy for long-run tractability.

01

Monotonic

Always increasing — trend is clear, no reversals to confuse the model.

TREND

02

Far less volatile

Noise averages out as counts accumulate — signal becomes dominant.

NOISE

03

More likely stationary

Better statistical behaviour — standard modelling assumptions hold more readily.

STABILITY

04

Well-understood forms

Sigmoid and logistic functions can describe it directly — no exotic machinery needed.

MATHS

05

The downside

Worse at short-term prediction — particularly when the horizon is shorter than your aggregated time period.

CAVEAT

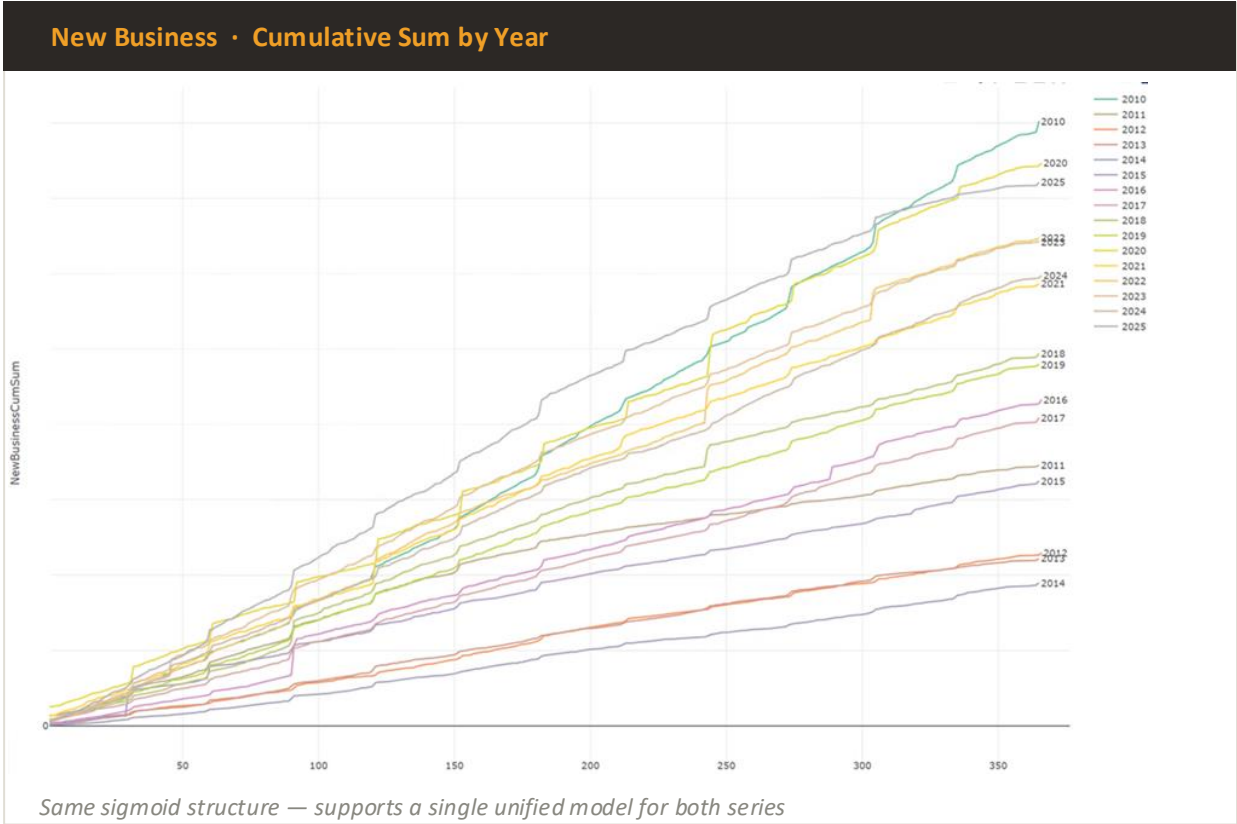
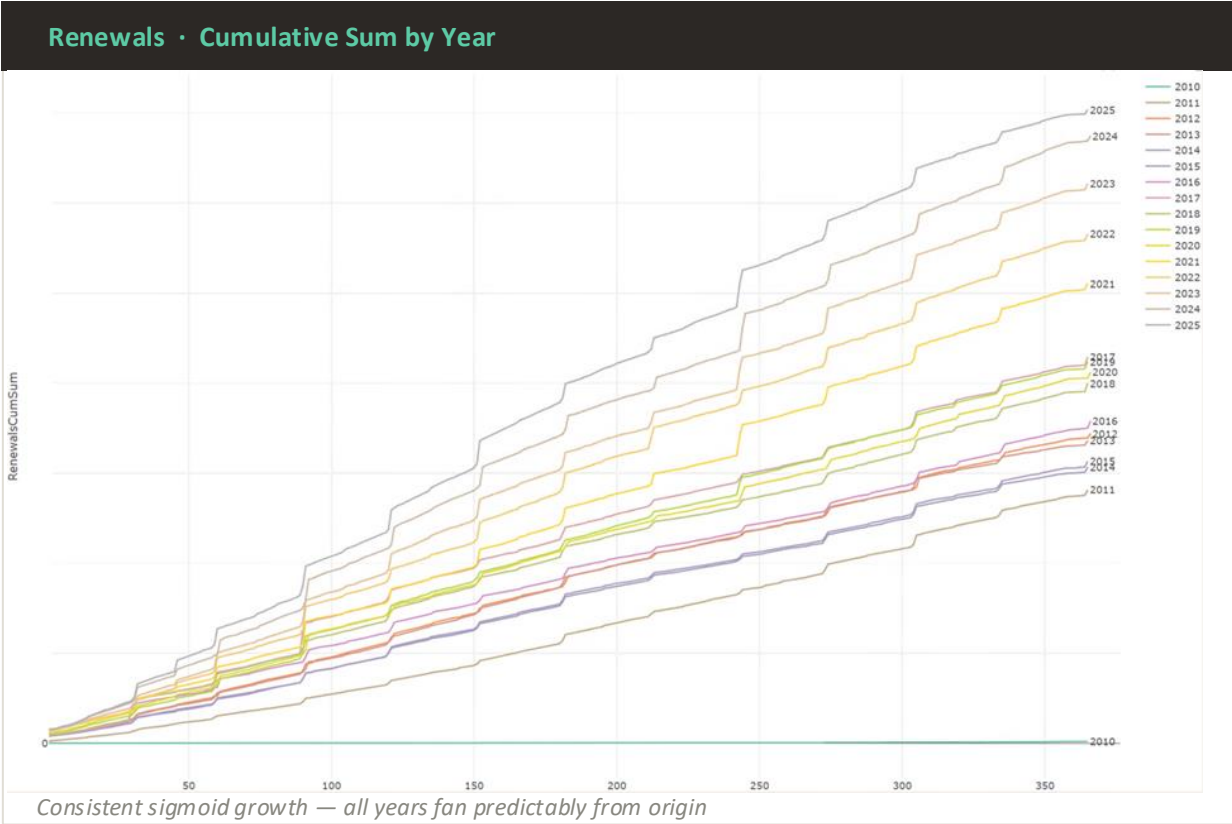
Summary Data

Both New Business and Renewals show the same sigmoid-shaped cumulative pattern — encouraging news for a unified modelling approach.

Same shape,
both series

Fan-out
across years

One unified
model viable



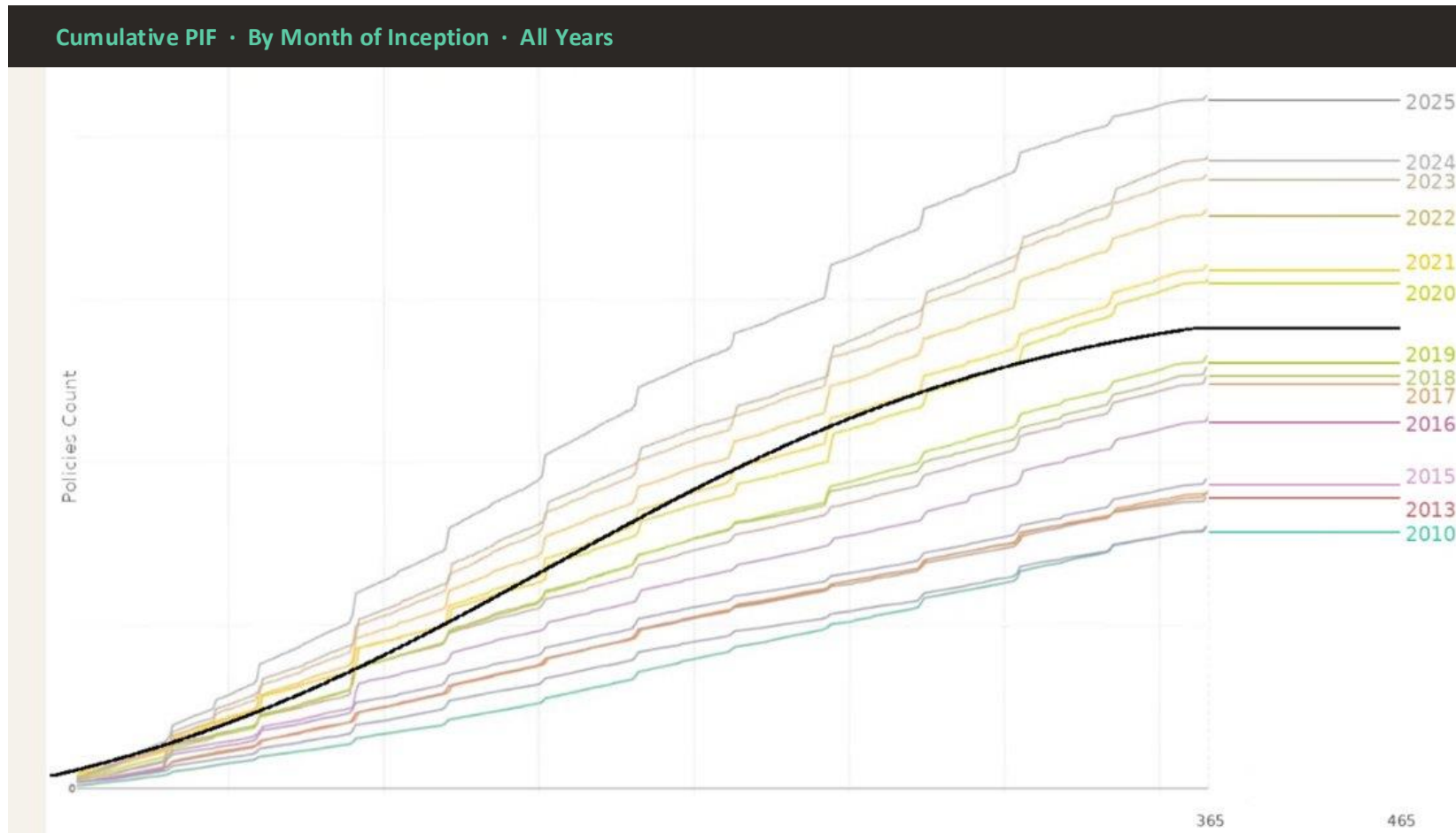
Summary Data

12

Cumulative PIF by month of inception — the sigmoid shape is usable for modelling;
it becomes more visible when we extend the YOA lines.

Sigmoid shape
visible

Extends beyond
365 days

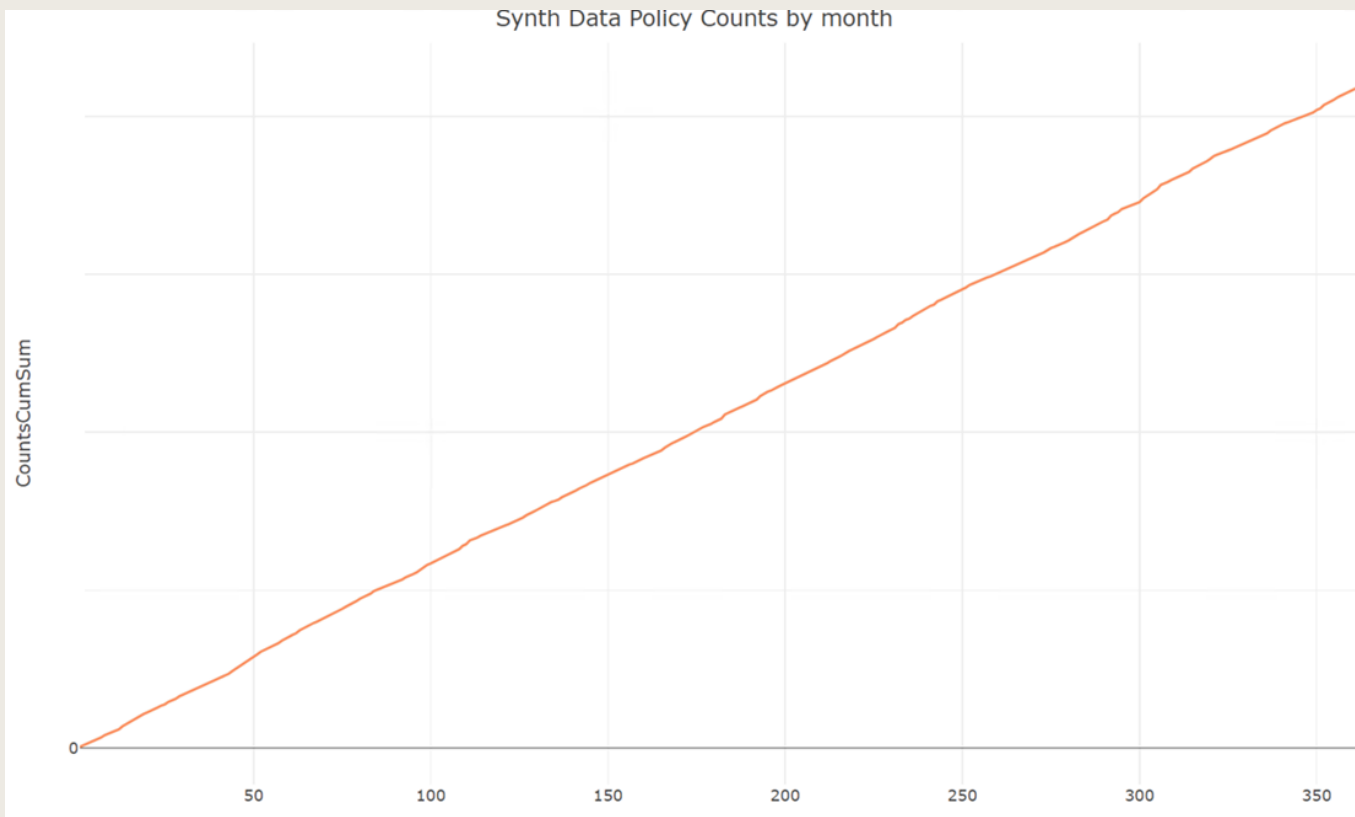


Black line = mean trajectory across years — sigmoid acceleration is the key modellable signal

Attempt: Synthetic Data

Generating realistic training data — what worked, and what didn't.

Synth Data: Policy Counts Cumulative Sum by Month



01 Simulate parameters

Drew model parameters from Normal distributions to mimic realistic variation.

02 Draw counts

Per-timepoint counts drawn from a Poisson distribution.

03 Assess realism

Overall shape looked plausible — but real-world periodicity never emerged.

⚙️ Workaround: splines used in trial models to approximate the periodic structure.

Synthetic data captured the **shape** but not the **periodicity** — the characteristic spikes that make real occurrence data so hard to model.

Two Paths Forward: Which Data Structure to Model?

At this point, two candidate data structures existed — each with a different philosophy.

Time Series

PATH CHOSEN

- Daily occurrence counts
- Volatile & noisy signal
- Non-monotonic
- Non-stationary
- Hard to fit directly

"There's value in understanding why the harder path is harder."

The Decision

Cumsum was clearly recommended.

Raw occurrences chosen anyway — to learn & understand why.

Sigmoid Cumsum

RECOMMENDED PATH

- Monotonic — always rising
- Far less volatile
- More likely stationary
- Much lower noise
- Well-understood math forms

Sidesteps nearly all data problems identified earlier.

Fitting a sigmoid formula directly (rather than letting the cumsum emerge naturally) also felt like skipping a step.



Null Model

Starting Simple: The Null Model

01

Philosophy

The Baseline Principle

Before building anything complex, establish a sensible baseline.

The simplest possible prediction: what happened before is what will happen again.

02

Data Choice

Starting with Renewals

Renewals were chosen first — a larger, more consistent dataset.

A natural starting point before tackling more variable New Business.

03

Prediction

The Null Prediction

The mean of all fully-developed historical years, applied to the forecast year.

Refinements like rolling means over 2–3 years are straightforward extensions.

The goal here is just a yardstick — something to beat. All subsequent models are judged against this baseline.

Null Model

2024/25 forecast using the mean of all fully-developed years to date, expressed at monthly granularity.

Dots = actuals

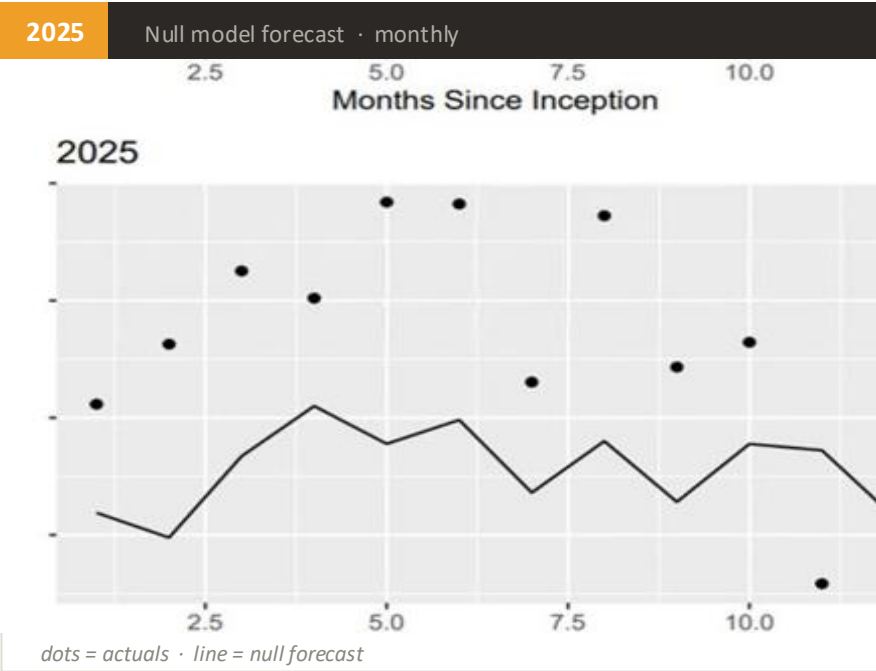
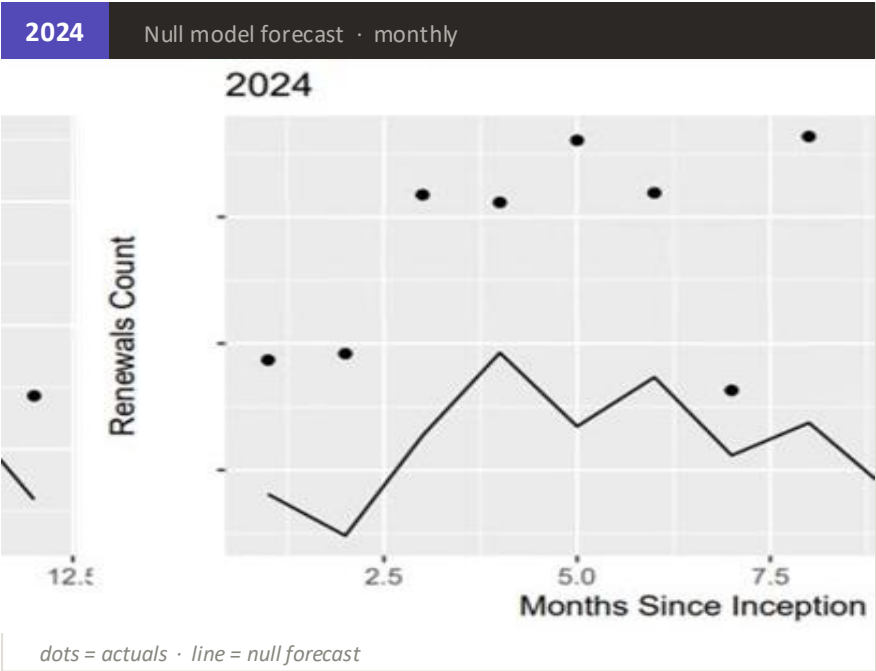
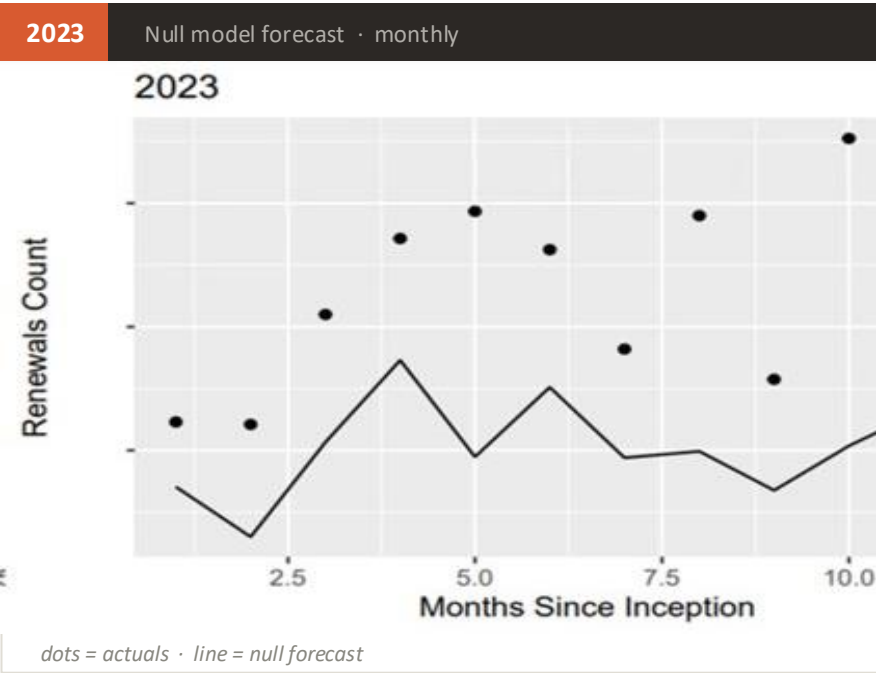
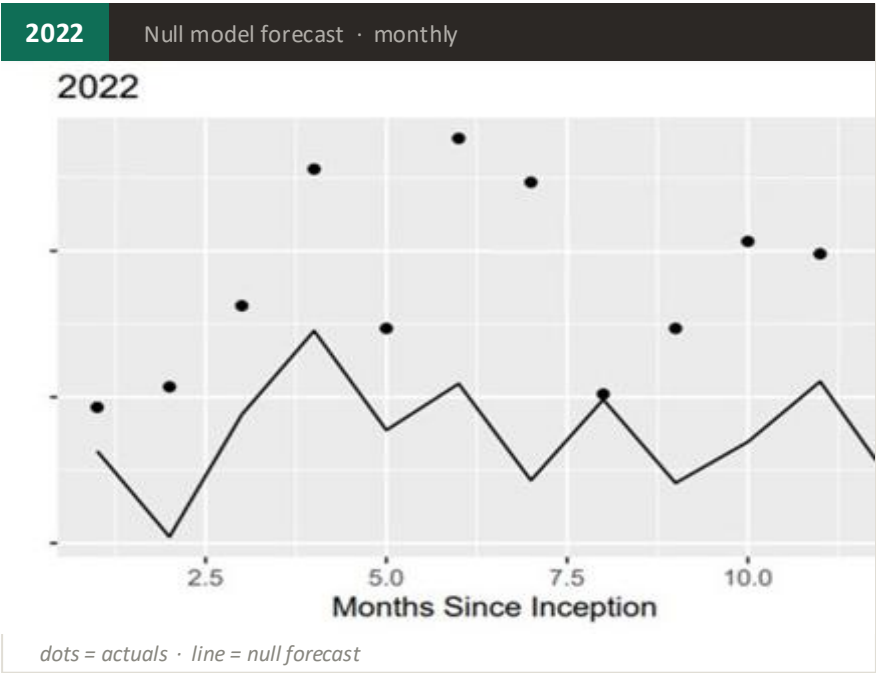
Observed monthly counts

Line = null forecast

Mean of prior full years

Consistent pattern

Model misses most peaks

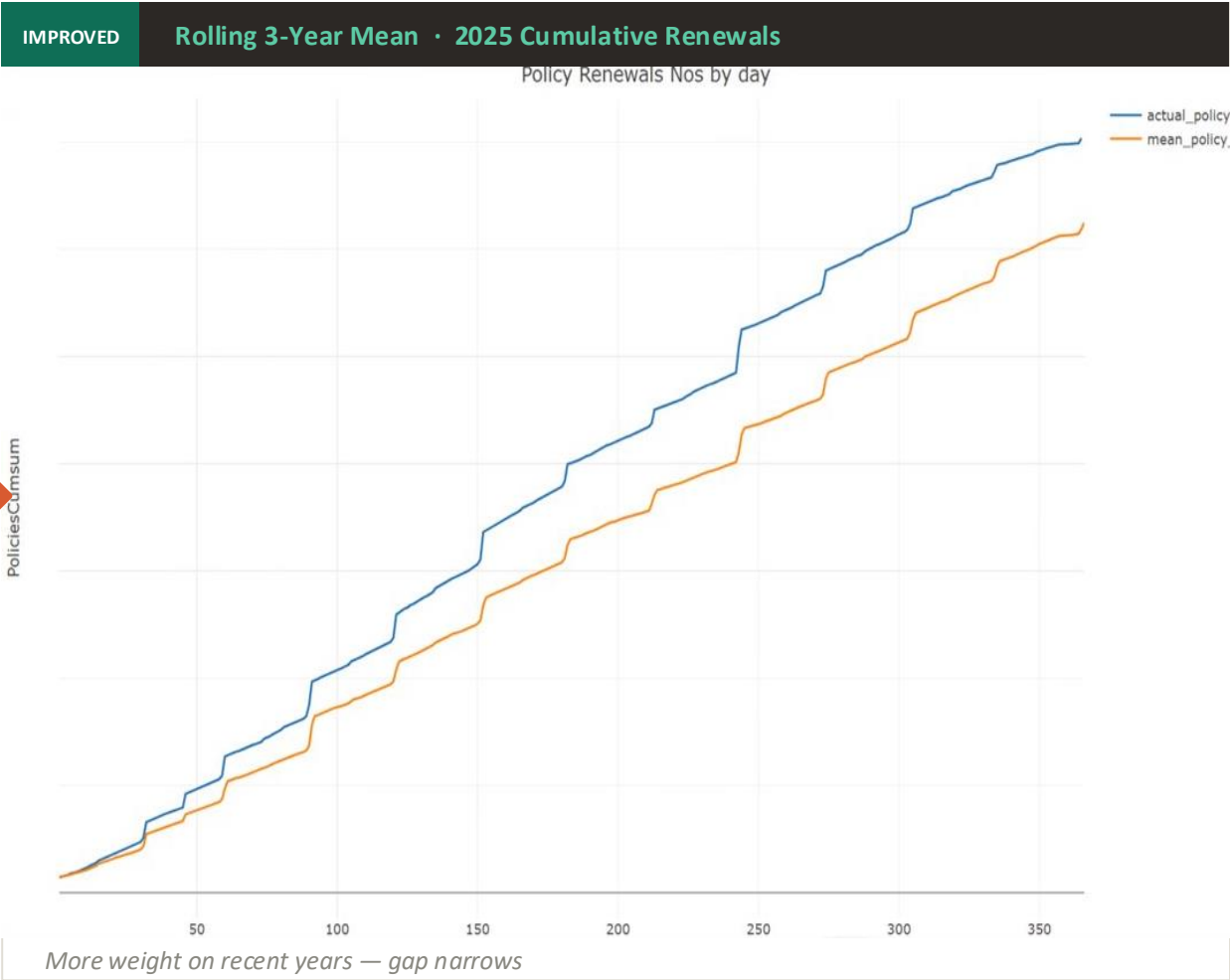
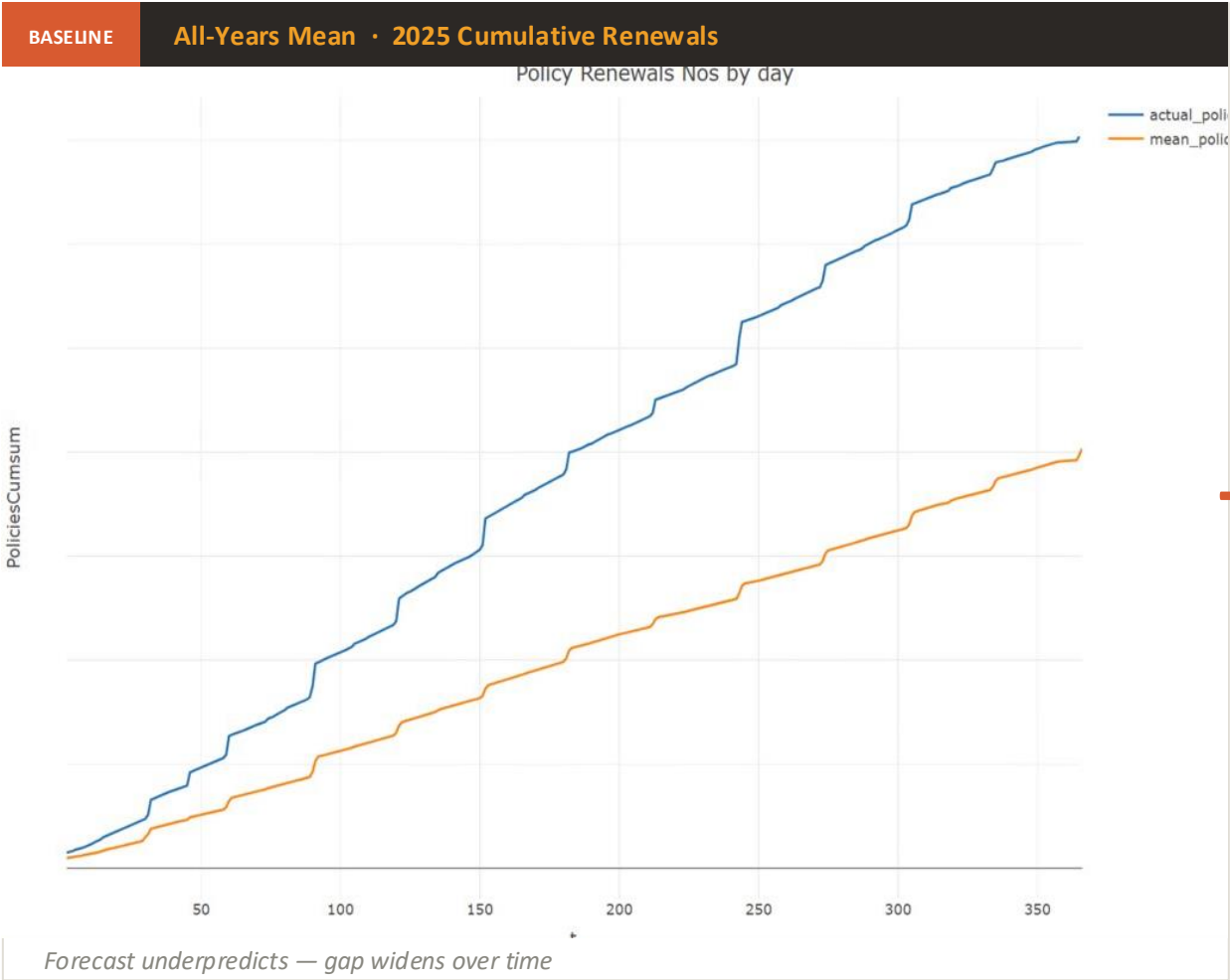


Null Model

Cumulative Renewals forecast for 2025 — comparing all-years mean (left) vs rolling 3-year mean (right).
Using the all-years mean gives a simple but informative baseline.
Replacing the all-years mean with a rolling 3-year mean gives more weight to recent experience.

All-years mean
underpredicts

3-yr rolling mean
closer to actuals





Technical Trap: The brms Non-Linear Syntax Pitfall

 The Issue & What Happened

The Issue

In **brms**, non-linear parameters must be explicitly declared using `nl = TRUE`
Exception: reserved parameter names like **mu** bypass this — silently.

```
# Must declare explicitly:  
brm(y ~ x, nl = TRUE, ...)  
  
# Silent trap — DO NOT rely on:  
brm(mu ~ x, ...)
```

What Happened

1

Used non-linear syntax without `nl = TRUE`
— was trying to write a linear model

2

Used reserved name `mu` — so the model
compiled without any errors

3

`brms` silently passed formula through
`terms.formula()`, treating it as non-linear

A silent error — no warnings, no failures, just wrong results waiting to be found.



Technical Trap: The brms Non-Linear Syntax Pitfall

 *Why This Is Dangerous*

Why This Is Dangerous



No errors thrown

The model compiled cleanly. Nothing in the output suggested anything was wrong.



Appeared valid

Diagnostics looked plausible — it seemed like a legitimate linear model.



Structure was wrong

Internally, brms treated it as non-linear. The parameter estimates were incorrect.



Silent — the worst kind

In statistical modelling, a silent error is far more dangerous than a noisy failure.

The model ran, looked fine, and was wrong. That combination is uniquely hard to catch.



Technical Trap: The brms Non-Linear Syntax Pitfall

 *How It Was Caught & Takeaway*

How It Was Caught

1 Extended the model

Adding new features to the model triggered unexpected compilation errors.



2 Traced the failures

Investigating the errors revealed the formula had been silently misspecified from the start.

Takeaway

Always declare non-linear models explicitly with `nl = TRUE`

Do not rely on reserved parameter behaviour — it can mask serious misspecification.

Avoid reserved parameter names unless you know exactly how brms handles them

When in doubt, be explicit. Explicit code is safe code.

Explicit is always safer than implicit in statistical model specification.

Bayesian GLM: The Naïve Starting Point

The first GLM formula — in ShinyStan, diagnostics looked deceptively good.

The Model Formula

```
renewals_glm <- brm(  
  formula =  
    PoliciesCount~PolicyYearOfAccount  
  family = poisson(link = "log")  
  data = renewals_data)
```

R

✓ Good chain mixing

Low R-hats, chains
converged

✓ MCMC space explored

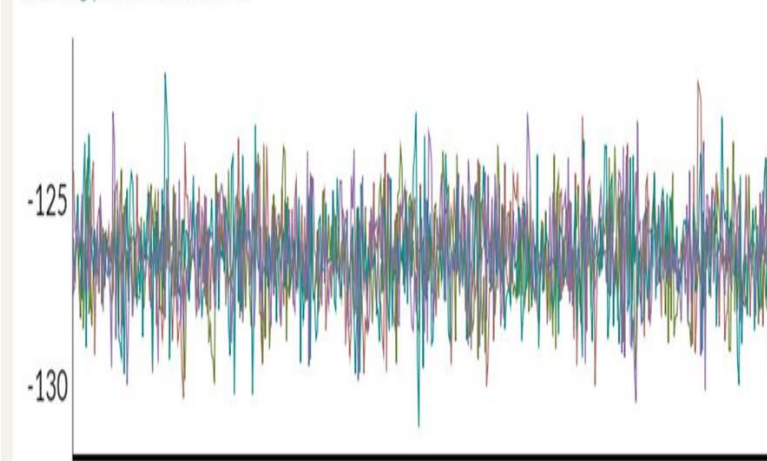
No multimodal
parameters

⚠ Deceptively good

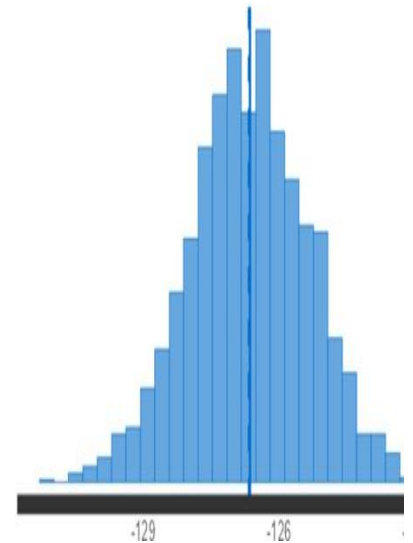
Out-of-sample tells a
different story

ShinyStan Diagnostics

Use your mouse to select a range in the traceplot to zoom into. The other plots on the screen will update accordingly. Double-click to reset.



Lines are mean (solid) and median (dashed)



Traceplot (left): chains mix well, no divergence. Posterior density (right): clean unimodal distribution for $b_Intercept$.

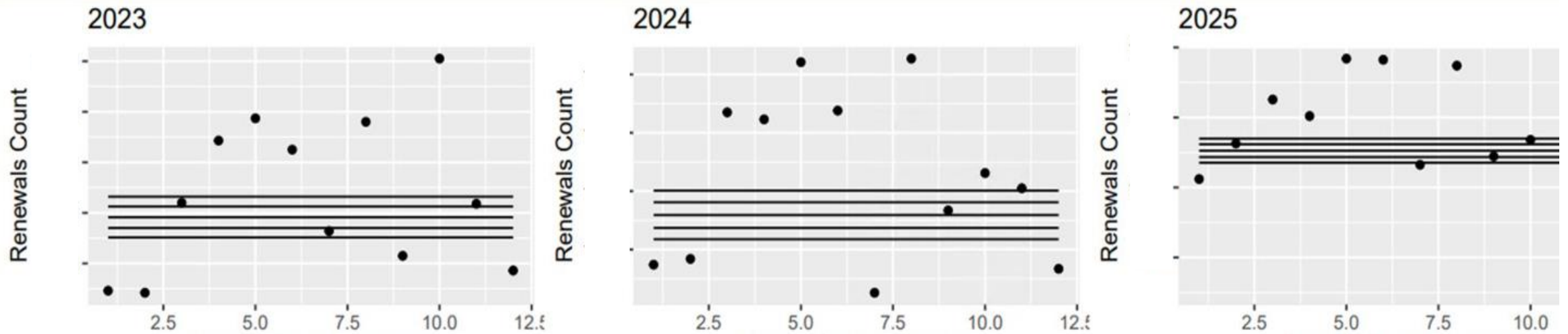
Diagnostics looked promising — but in-sample performance doesn't tell us how the model predicts out-of-sample.

Bayesian GLM: The Naïve Starting Point

Actual vs. Expected — within sample. In-sample performance can be misleading.

In-Sample Fit: 2023 · 2024 · 2025

IN SAMPLE



Each panel = one inception year. Dots = actual monthly Renewals Count. Horizontal bands = model posterior predictive intervals.

✓ Bands are centred

Posterior intervals track the data reasonably in-sample.

~ Wide intervals

Predictive uncertainty is large — model isn't confident.

⚠ Don't be deceived

Good in-sample fit does not guarantee good forecasts.

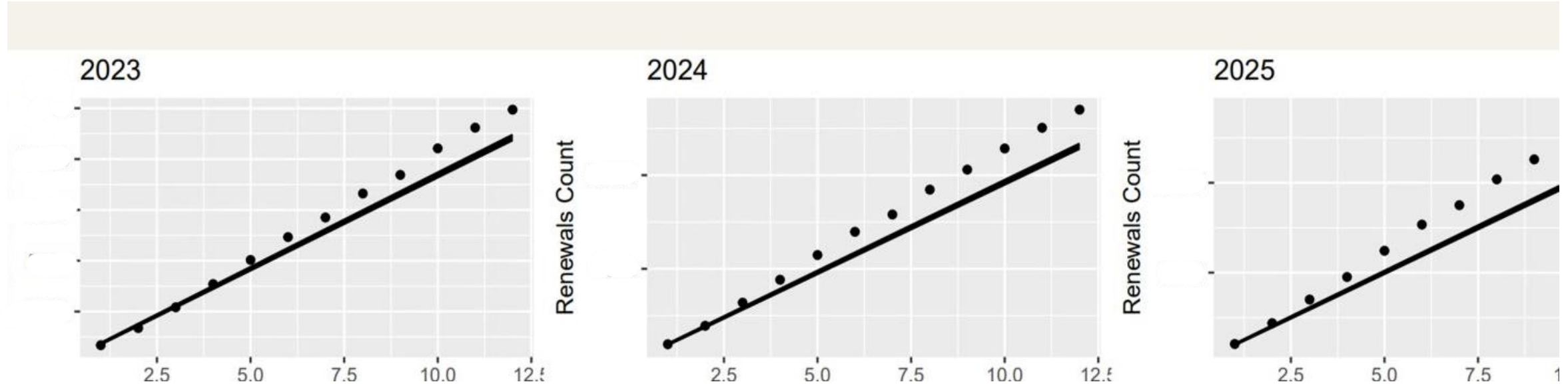
The real test is out-of-sample: how well does the model predict years it has never seen?

Bayesian GLM: The Naïve Starting Point

Cumulative Actual vs. Expected with 2024/25 held out — the out-of-sample performance tells the real story.

Out-of-Sample Fit: 2023 (in) · 2024 (out) · 2025 (out)

OUT OF SAMPLE



Cumulative counts by month. Solid line = model prediction. Dots = actual. 2024 and 2025 were not used in training.

✓ 2023 underpredicts but not too badly

In-sample year tracks best — as expected.

✗ 2024 diverges

The model undershoots as the year progresses.

✗ 2025 worse still

Prediction gap widens — the naïve GLM can't forecast reliably.

Verdict: the naïve GLM is not good enough. A better model is needed.

Bayesian GLM

One of the best-performing GLM specifications.

✓ Good chain mixing

Low R values

✓ Adequate MCMC exploration

No multimodal parameters

✓ ShinyStan validated

Genuine posterior exploration

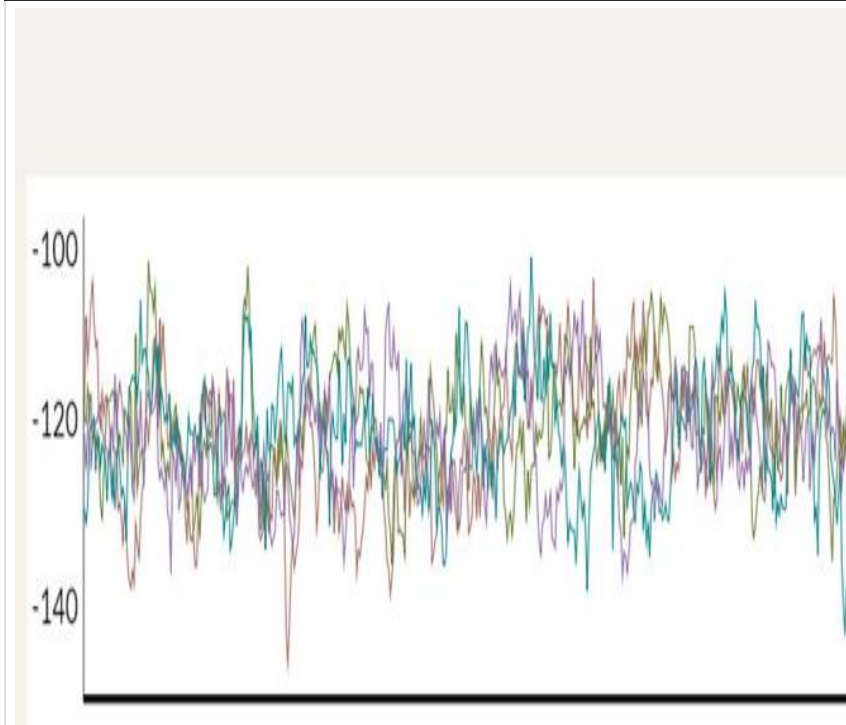
*A simple distribution swap.
A meaningful improvement.*

R

```
renewals_glm <-  
  
PoliciesCountByYearMonth ~  
  
  PolicyYearOfAccount +  
  
  PolicyMonthOfAccount +  
  
  (1 | PolicyYearOfAccount:PolicyMonthOfAccount)
```

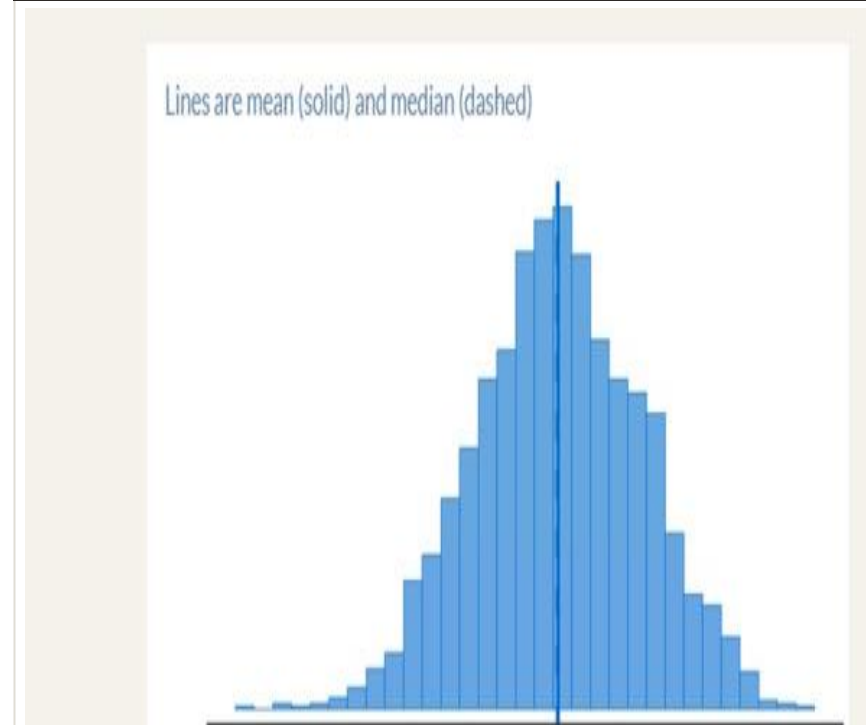
↳ Random intercept per year-month — no slope, partial pooling across groups

MCMC Trace Plot · b_Intercept



Chains mix well — no divergence, stable exploration

Posterior Distribution · b_Intercept



Unimodal, well-centred — mean $\approx$ median, no pathologies

Bayesian GLM

Actual vs. Expected — in-sample fit.

Good in-sample capture

Posterior bands track actuals

Posterior uncertainty

Fan of credible interval lines

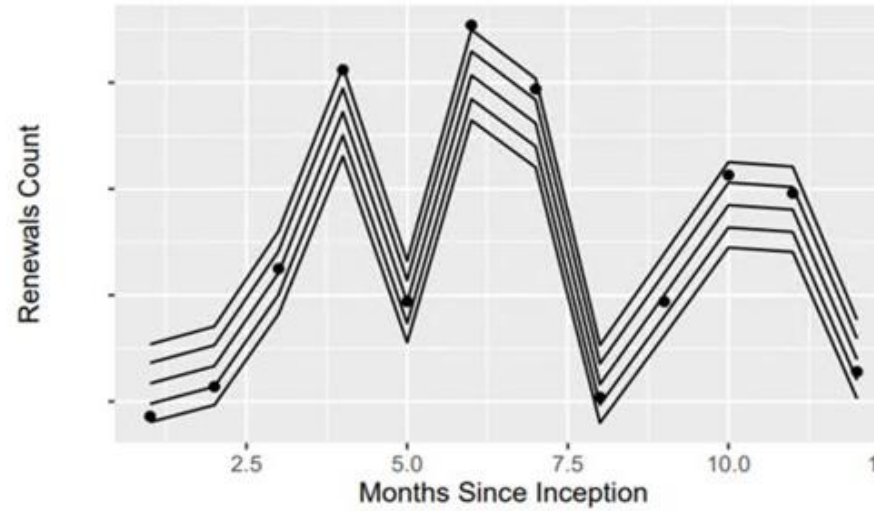
2025 extrapolation risk

Spike at month 12 — watch this

2022

in-sample

2022

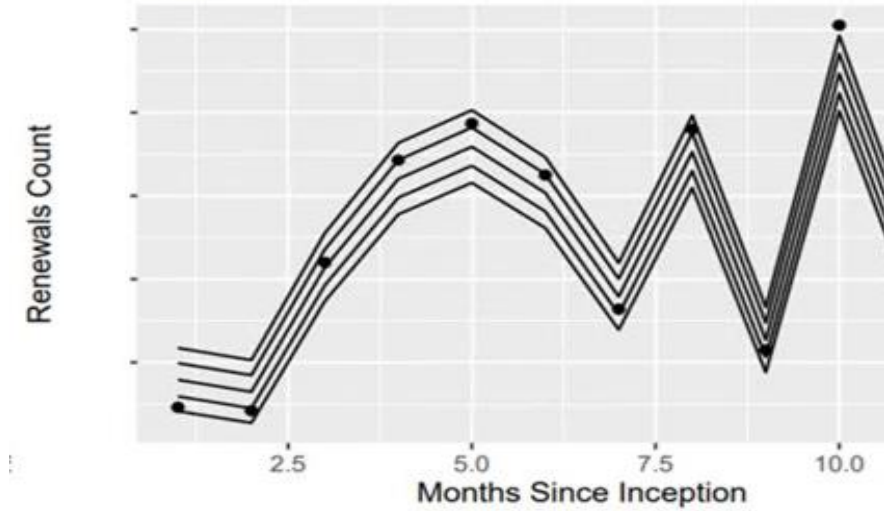


actuals vs posterior

2023

in-sample

2023

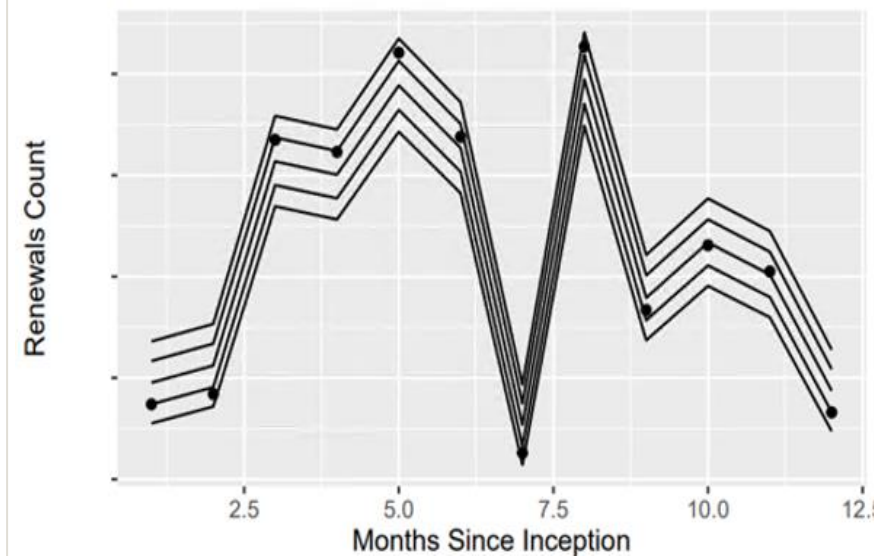


actuals vs posterior

2024

in-sample

2024

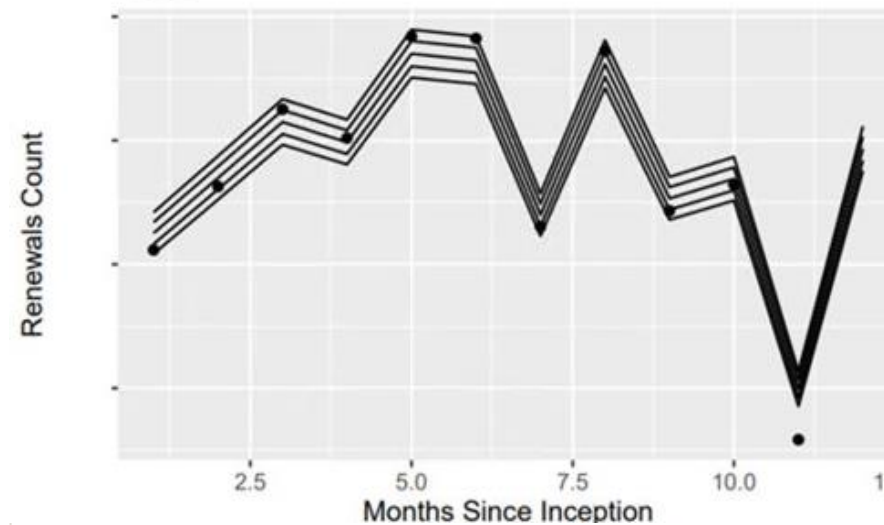


actuals vs posterior

2025

in-sample

2025



actuals vs posterior

Bayesian GLM

Both 2024 and 2025 held out — a more demanding test of predictive performance.

2022–23 fit retained

In-sample years still well-fit

2024 flat prediction

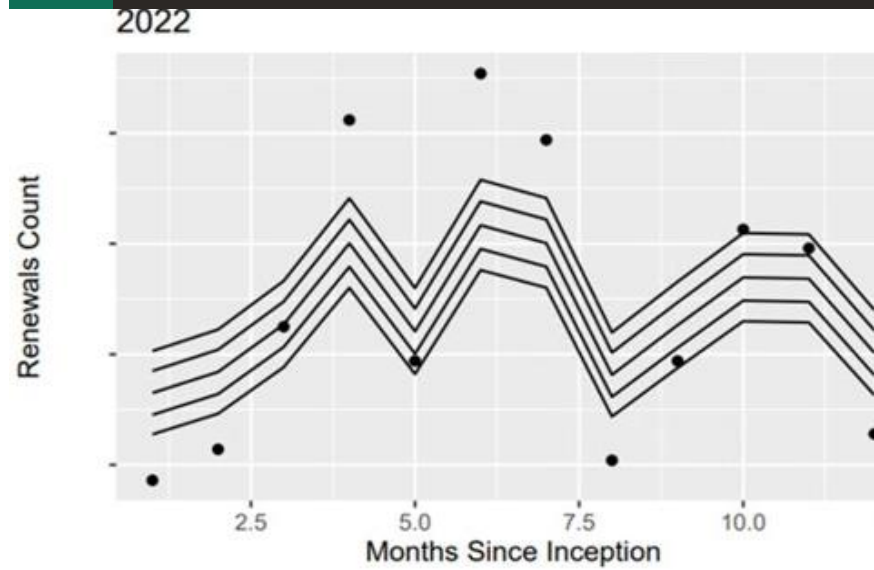
Model struggles out-of-sample

2025 flat prediction

Now the bands are even narrower

2022

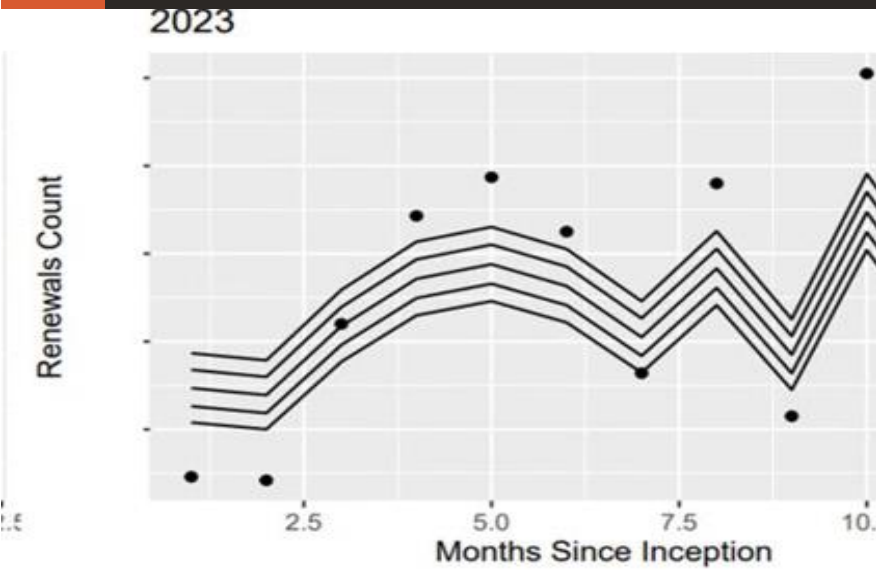
In-sample



actuals vs predicted

2023

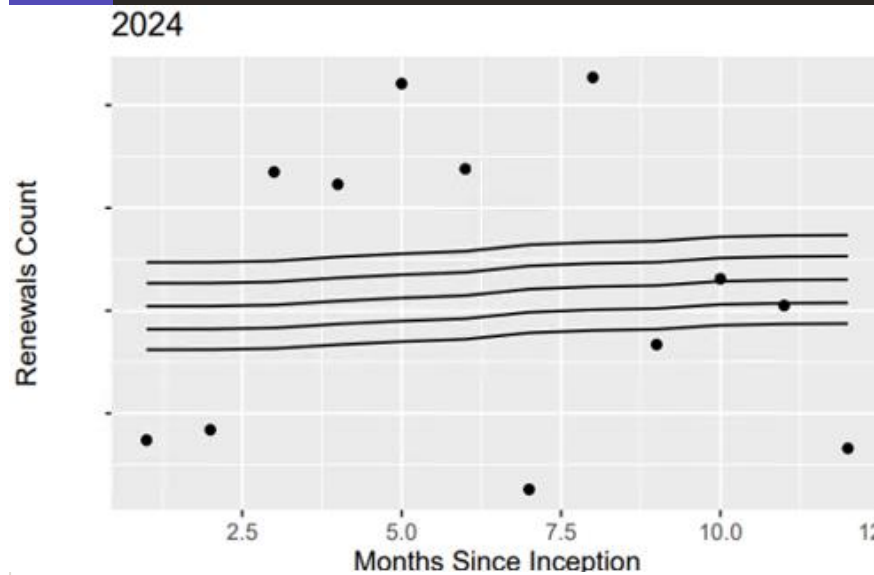
In-sample



actuals vs predicted

2024

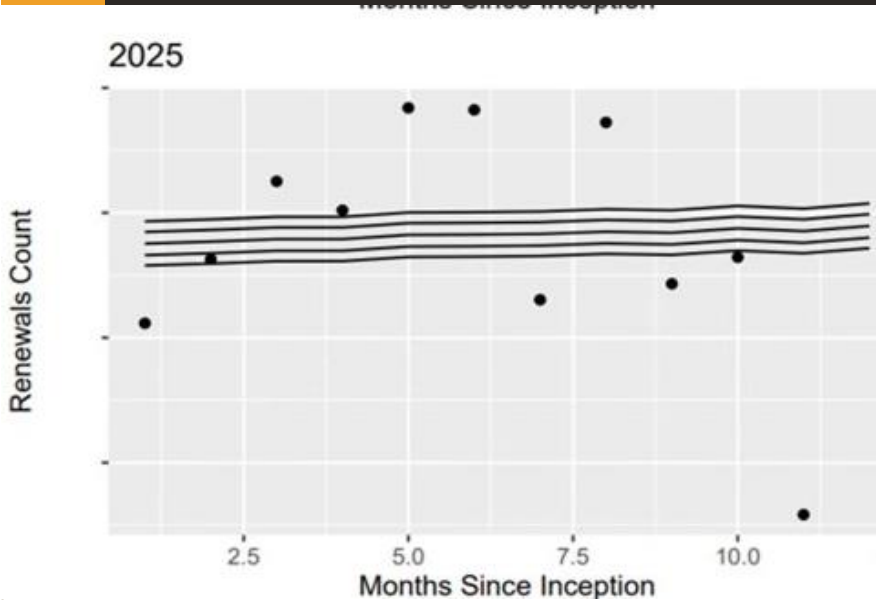
Out-of-sample



actuals vs predicted

2025

Out-of-sample



actuals vs predicted

Bayesian GLM

Cumulative Actual vs. Expected for 2025
(out-of-sample).

No worse than null model

Matched baseline performance

No better than null model

Didn't beat a simple mean

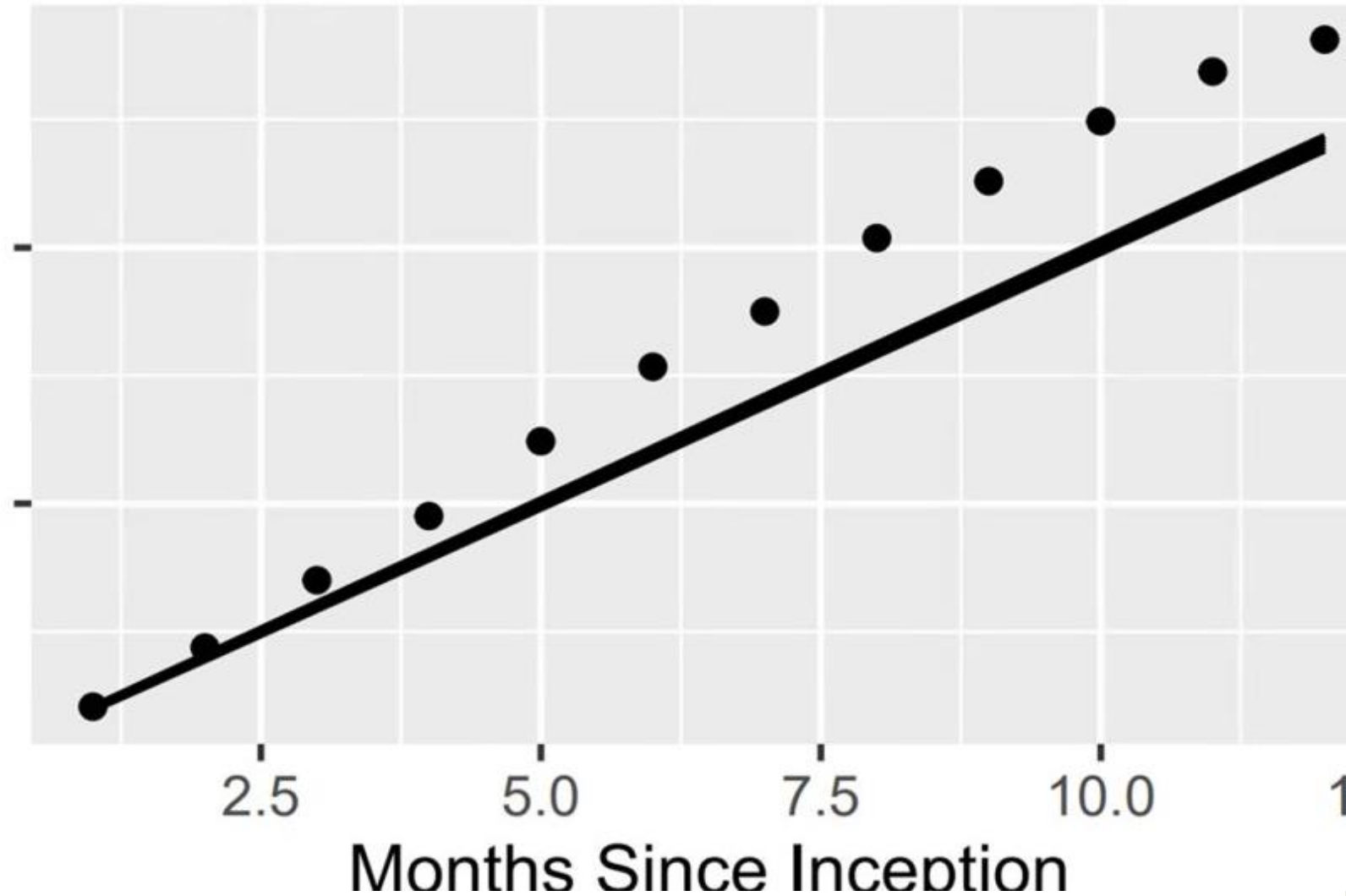
Narrow prediction bands

But no improvement over mean

2025

Cumulative Actual vs. Expected · Out-of-sample

2025



Actuals (dots) vs predicted line — close tracking but no gain over null model

The GLM Verdict: Back to the Drawing Board



The Search Begins

Bayesian Non-Linear Models

INITIAL APPROACH

Systematic search across model families
Starting point: **simple non-linear Poisson**, no hierarchical effects

Three families tried. Same conclusion each time.

01

Power Functions

POWER MODELS

Multiple functional forms tested across the power family.

✗ Consistently poor performance — failed to fit the data pattern.

02

Cosine Formula

COSINE MODELS

Attempted to capture the stepped periodicity using cosine-based functional forms.

✗ Systematic, material underprediction.

03

Spline Models

SPLINE MODELS

Successfully captured the stepped pattern — the most structurally promising family.

~ Still: systematic, material underprediction out of sample.



Structural fit improved with complexity — but none predicted reliably **out of sample**.
The problem runs deeper than **formula choice**.

Non-Linear Model

Power Function

R

```
renewals_model <- bf(  
  PoliciesCount ~  
    a - b*(Month - c)^2,  
  a+b+c ~ (1+1|Year:Month),  
  nl = TRUE,  
  family = poisson("log")  
)
```

2024/25 held out

Out-of-sample test

Power fn non-linear

Partial pooling by year-month

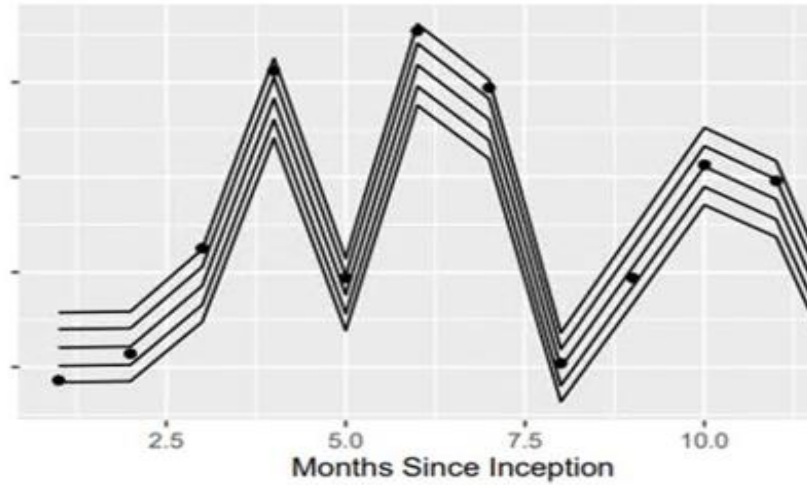
2024 flat prediction

Model underfits held-out years

2022

In-sample

2022

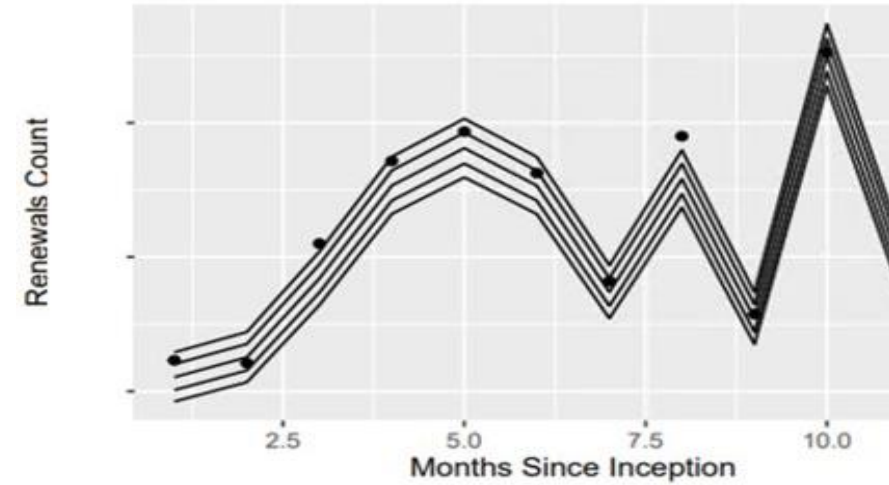


actuals vs posterior predictive

2023

In-sample

2023

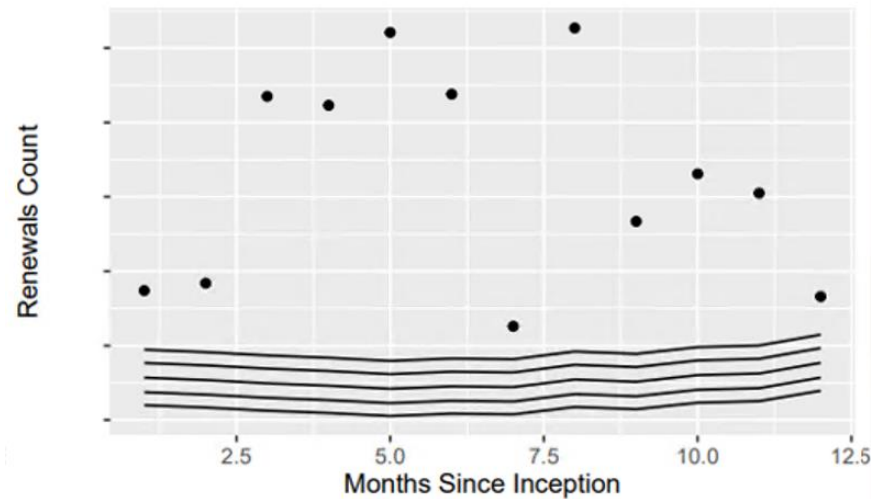


actuals vs posterior predictive

2024

Out-of-sample

2024

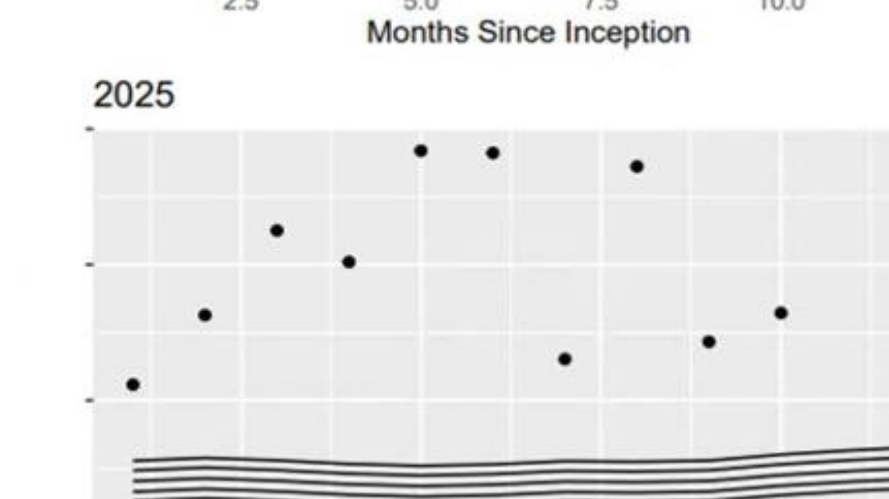


actuals vs posterior predictive

2025

Out-of-sample

2025



actuals vs posterior predictive



Non-Linear Model

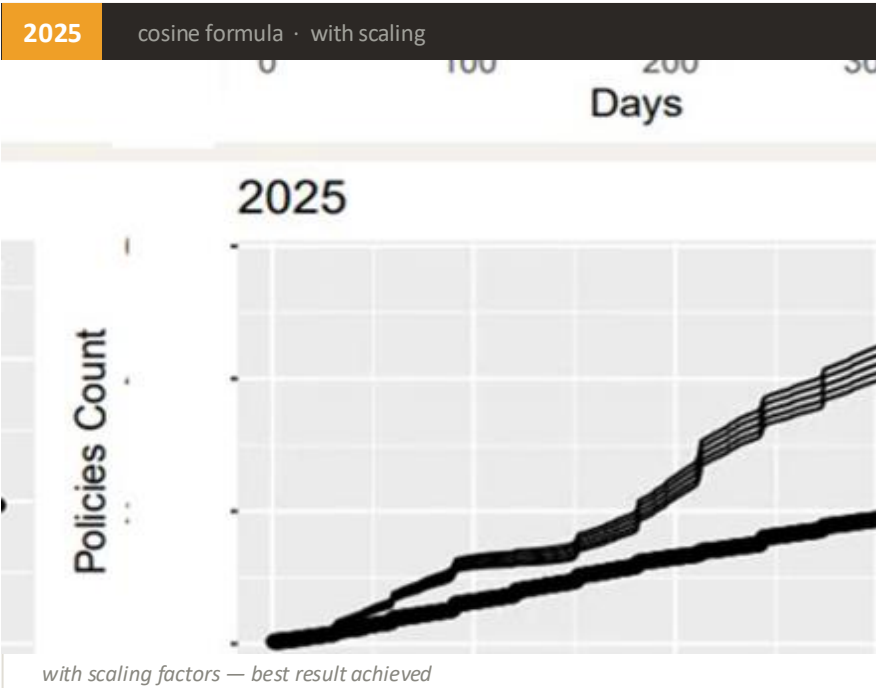
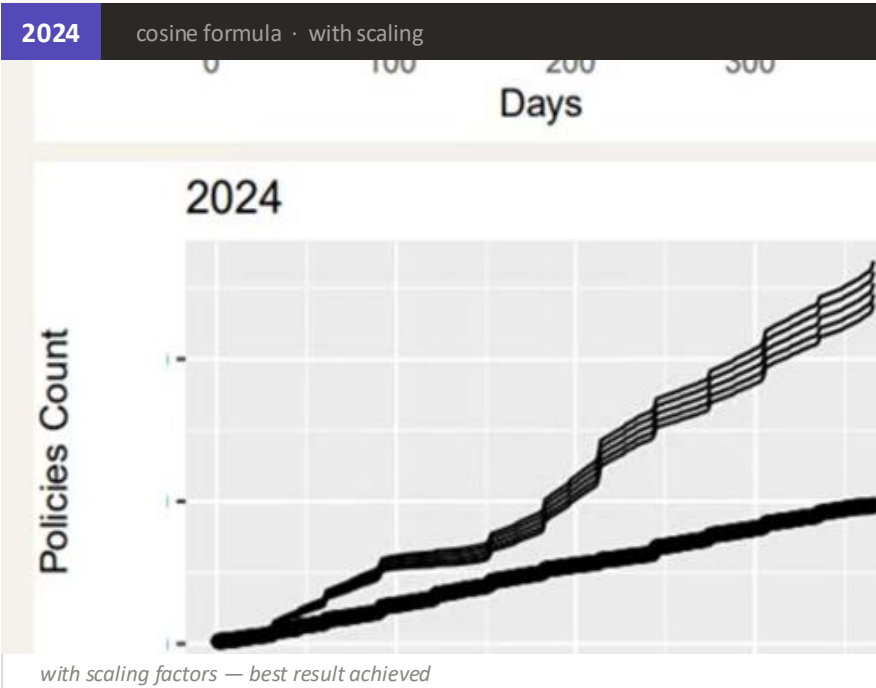
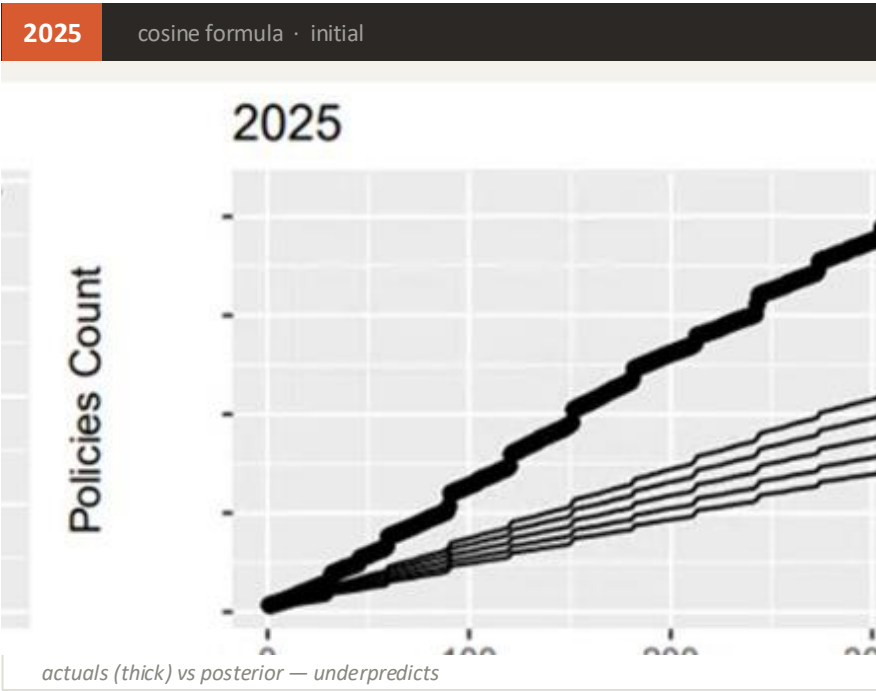
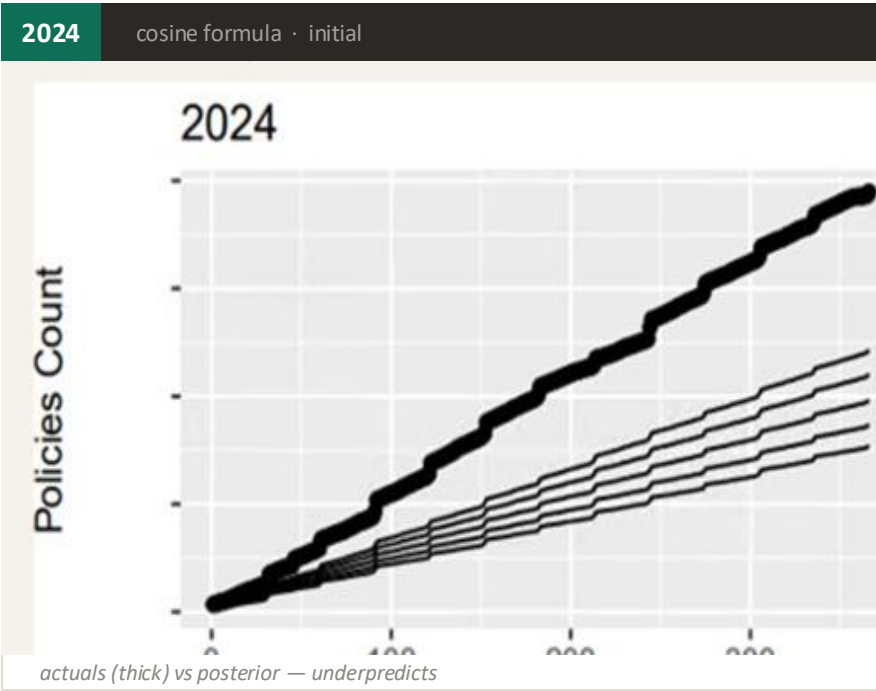
Cosine Formula

R

```
model_formula <- bf(
  CountOfPolicies ~
    policycount +
    #everyday base
    mufifteen*fmax(
      cos((15/36)*t*x), 0) +
    #15th-day spike
    muthirty*fmax(
      cos((30/146)*t*y), 0),
    #30th-day spike
    mufifteen + muthirty ~
      (1+1|day:RecordType),
    policycount ~ 1 + normyear
      + (normyear|RecordType),
    x+y ~ (1+normyear|RecType),
    nl=TRUE, family=poisson("log")
)
```

In-sample underpredicts
Gap visible even without holdout

Scaling factors helped
Bottom row = now overpredicts!



Non-Linear Model

Cosine Formula

Out-of-sample performance after applying a correction fix (discussed shortly).

Actuals above bands

Upper dashed line = actuals 2025

Wide 95% interval

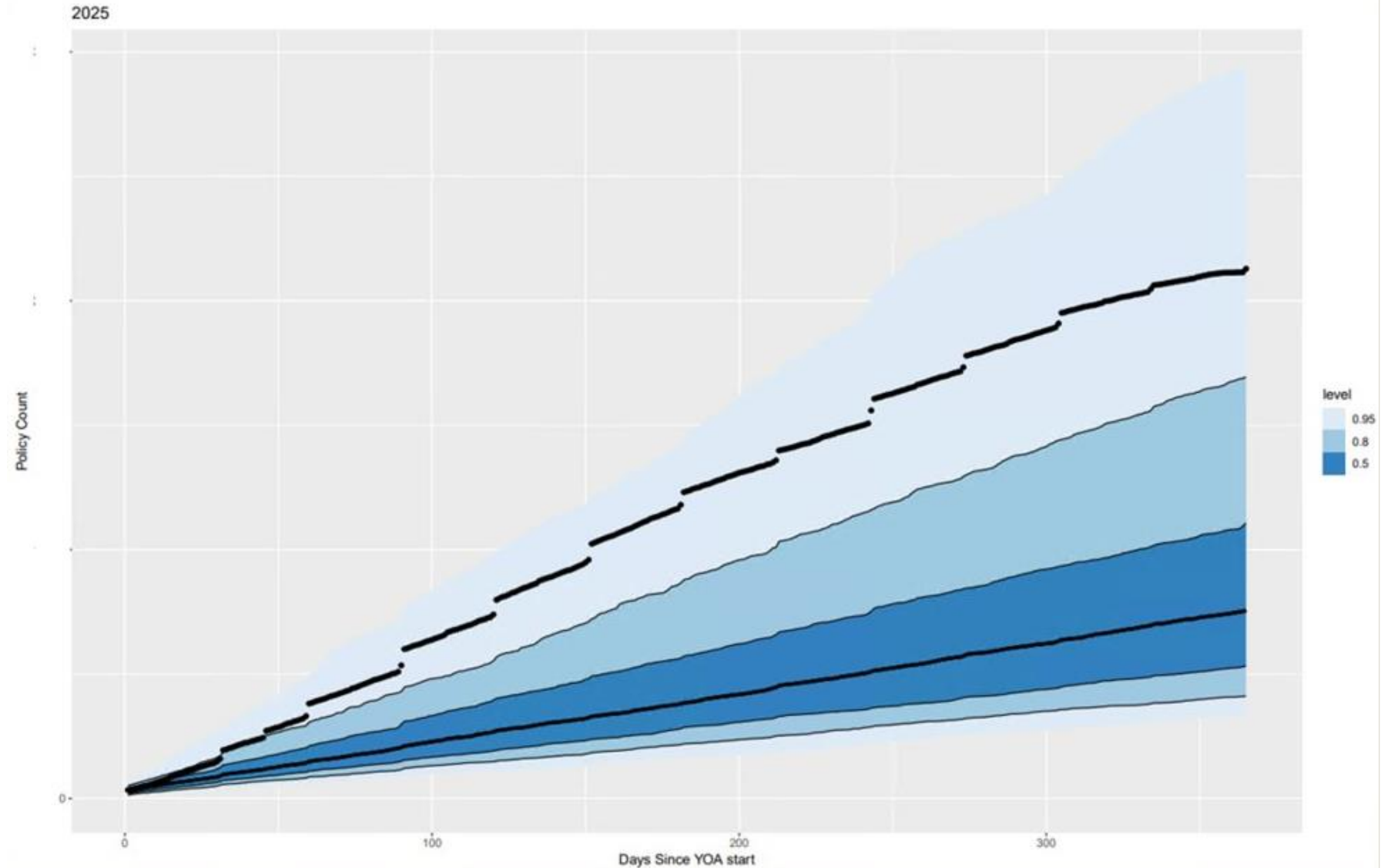
Model highly uncertain out-of-sample

Correction fix applied

Improves the gap but is still clearly under

2025

Out-of-sample · Cosine formula with correction fix · Credible intervals: 50% / 80% / 95%



Dashed line (top) = actuals — sits above 95% band, indicating systematic underprediction remains



Non-Linear Model

Spline

R

```
renewals_model2 <- bf(  
  PoliciesCount ~  
    (100 + YearOfAccount/3)  
    * myear +  
    MonthOfAccount * varmonth,  
  myear ~ (1+1|Year:Month),  
  nlf(varmonth ~  
    200 - (0.9*Month*mumonth  
    - 12)^2),  
  mumonth ~ (1|Month) + s(day),  
  nl = TRUE,  
  family = poisson("log")  
)
```

Spline on day-of-year

s(day) captures periodicity

Still underpredicts

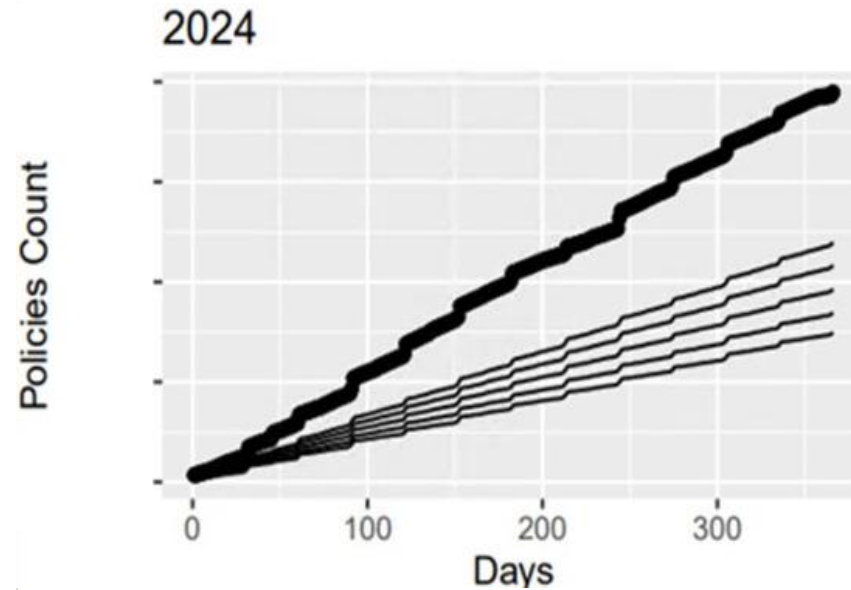
Actuals above posterior bands

2024/25 held out

Out-of-sample performance test

2024

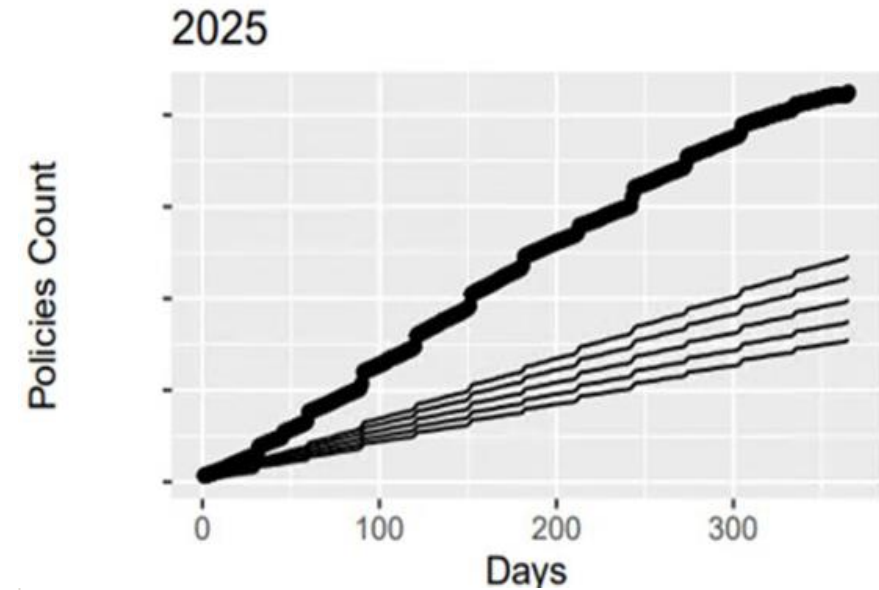
Spline model · out-of-sample · cumulative policy count



actuals (thick) vs posterior bands — gap widens

2025

Spline model · out-of-sample · cumulative policy count



actuals (thick) vs posterior bands — persistent underprediction

Why Did None of These Work?

01

DATA QUALITY

No Compression

- Volatile — large swings day to day
- Noisy — signal buried in noise
- Non-monotonic — no clean directional trend
- Non-stationary — properties shift over time

Raw occurrence data resists any standard mathematical function.

02

METHODOLOGY

Formula-Chasing

- Increasingly complex formulas with too many free parameters
- No principled way to judge variable importance

Complexity without constraint leads to overfit, not insight.

03

PRIOR KNOWLEDGE

Ignoring Context

- No sensible baseline to anchor predictions
- Significant time spent on formulas that business context could have ruled out early

The business already knew what mattered — the model didn't.

!

*Simply attempting to recreate the data with a mathematical formula — without **compression**, **constraints**, or **business context** — will fail.*

Summary: None of the Non-Linear Models Were Good Enough



A New Direction

Adapting the FNOL Model

BREAKTHROUGH

Adapting the existing FNOL prediction model to the PIF problem outperformed everything else tried — by a meaningful margin.

- Compresses to development percentiles

- Removes stepped periodicity

- Occurrence-based structure

Still not quite there: decent in-sample fit but systematic out-of-sample underprediction.

The Model Formula

R

```
model_formula <-  
  bf(  
    CountOfPolicies ~  
      ultimatepolicyrate +  
      log_sum_exp(  
        logdifffunction(t, tprev, exp(loglambda))  
        + log(1 - inv_logit(percentdonebyinflection))  
      ),  
      log(fmin(t, 365) - fmin(tprev, 365))  
      + log(inv_logit(percentdonebyinflection))  
    ),  
    ultimatepolicyrate ~ 1 + normyear + (normyear | RecordType),  
    # global linear normyear trend; normyear has its own RecordType split  
    percentdonebyinflection ~ (1 + 1 | RecordType),  
    # single constant mean + random noise; no covariates, intercept only  
    loglambda ~ (normyear | RecordType),  
    # no fixed predictors; random intercept per year, partial pooling by  
    RecordType  
    nl = TRUE,  
    family = poisson(link = "log")  
  )
```

⚠ nl = TRUE — explicit!

Adapted Model

Initial Results

2025 forecast — comparing in-sample fit (left) against out-of-sample predictive performance (right).

In-sample: fans out

Posterior lines bracket actuals

Out-of-sample: tight

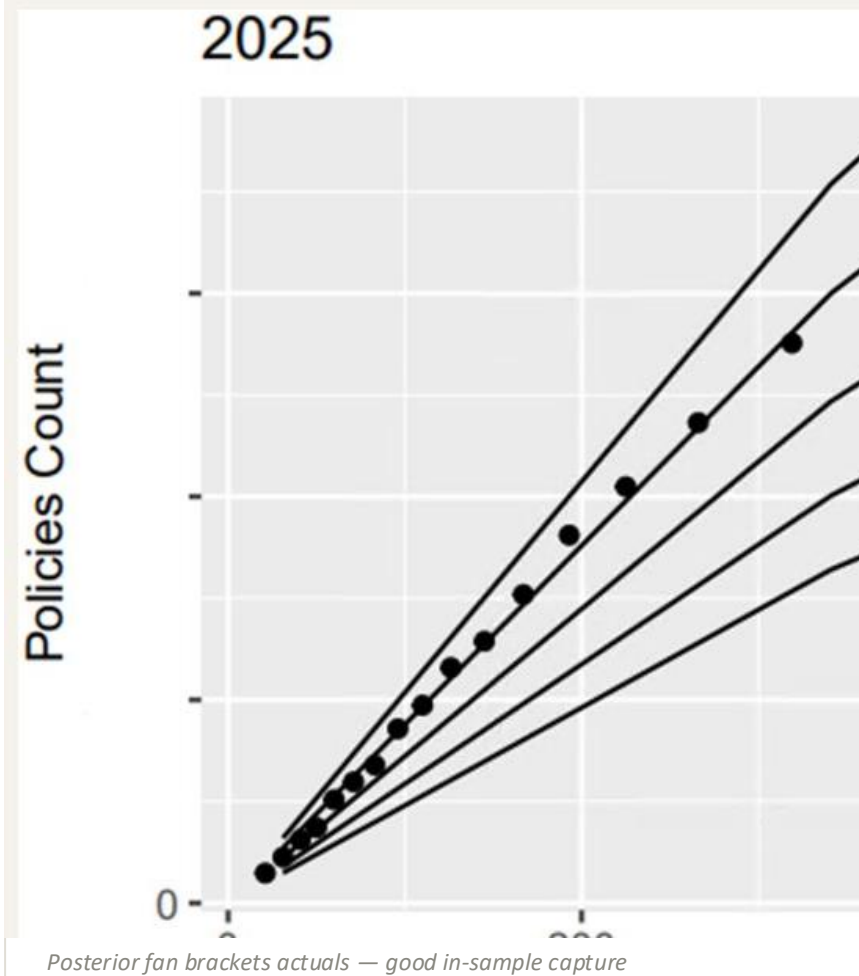
Bands narrow, actuals tracked

Promising initial fit

Model adapts well to 2025 data

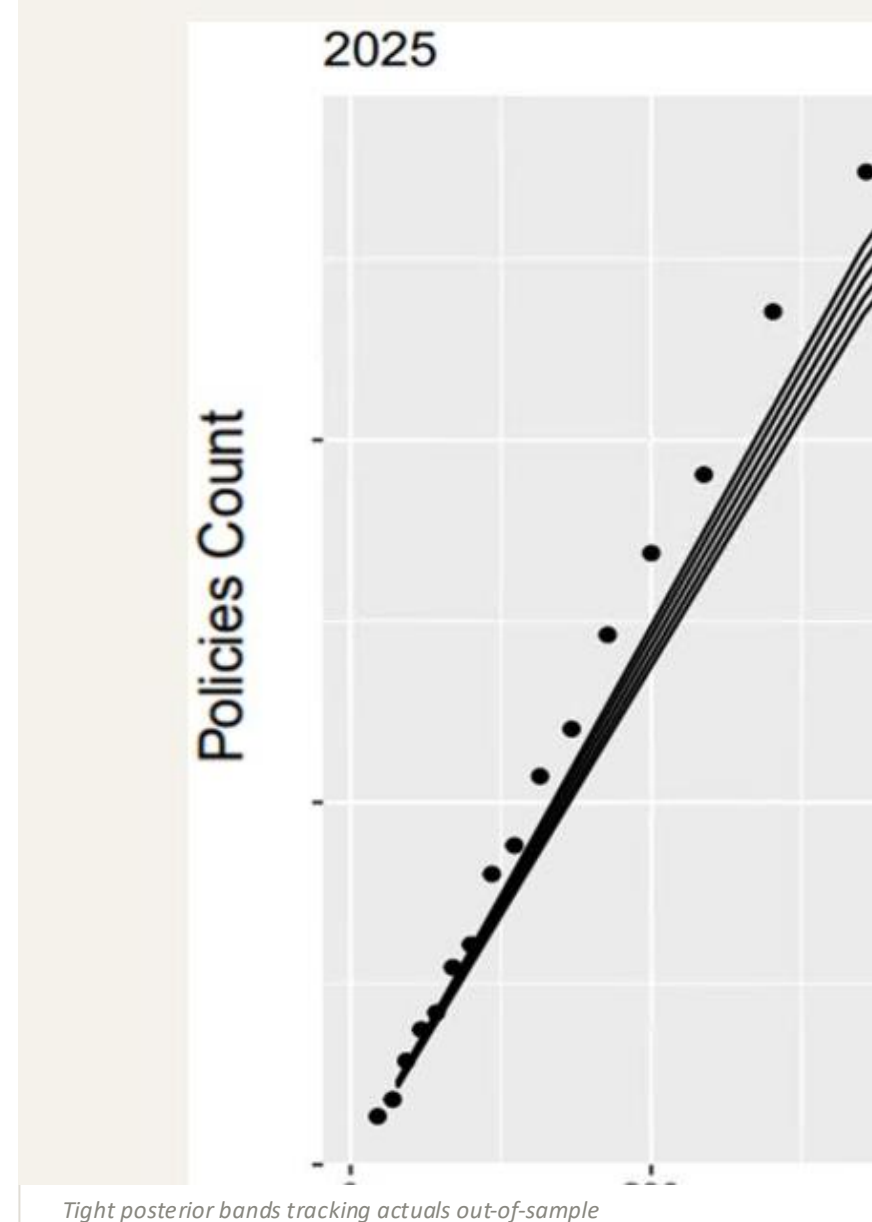
In-sample

2025 · Adapted model · Policies Count vs Days



Out-of-sample

2025 · Adapted model · Policies Count vs Days



Summary: None of the Non-Linear Models Were Good Enough



Diagnosing the Underprediction

Two hypotheses investigated — and systematically ruled out.

H1

PRIOR REGULARISATION

Priors Too Strong

HYPOTHESIS

Tight priors on some features could bias estimates toward lower rates.

TEST

Tested progressively looser priors — including completely uninformative Uniform priors for all features.

RESULT

Negligible difference. Prior strength was not the cause.

~~X RULED OUT~~

H2

HIERARCHICAL STRUCTURE

Partial Pooling Shrinkage

HYPOTHESIS

Partial pooling shrinks group-level estimates toward the grand mean — a plausible culprit given its widespread use in the model.

TEST

Added fixed effects on year to counteract shrinkage.

RESULT

Made little difference and sometimes degraded performance.

~~X RULED OUT~~

&



Both hypotheses ruled out. The real cause was a **distributional mismatch** — Poisson assumes variance = mean, but insurance count data almost always shows **overdispersion**.

Diagnosing the Under-prediction

Hypothesis 3

Hard development cap — a creative hack

Extend the development window to 2 years by padding with zeros before compressing.

Quantiles extend >1yr

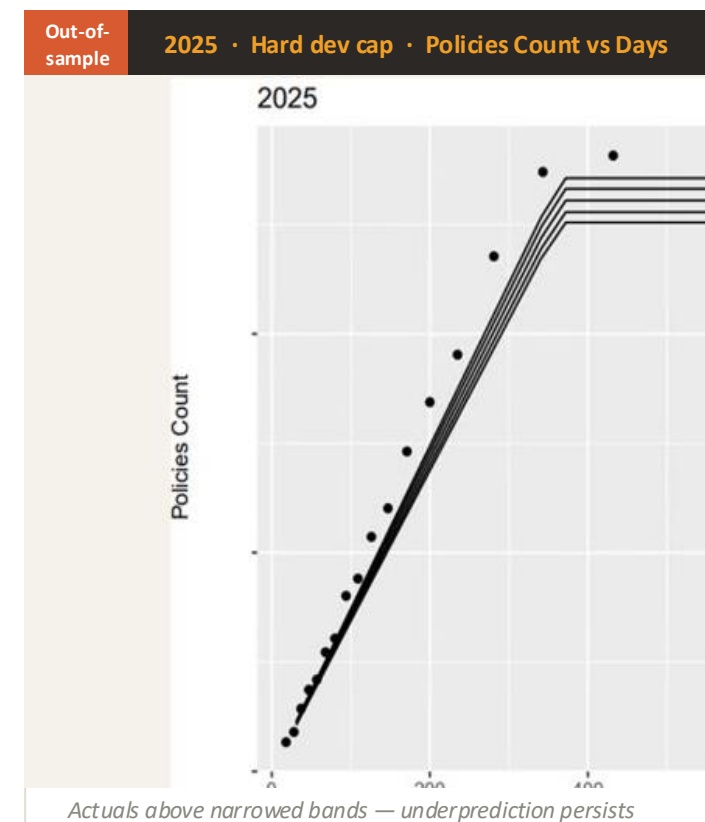
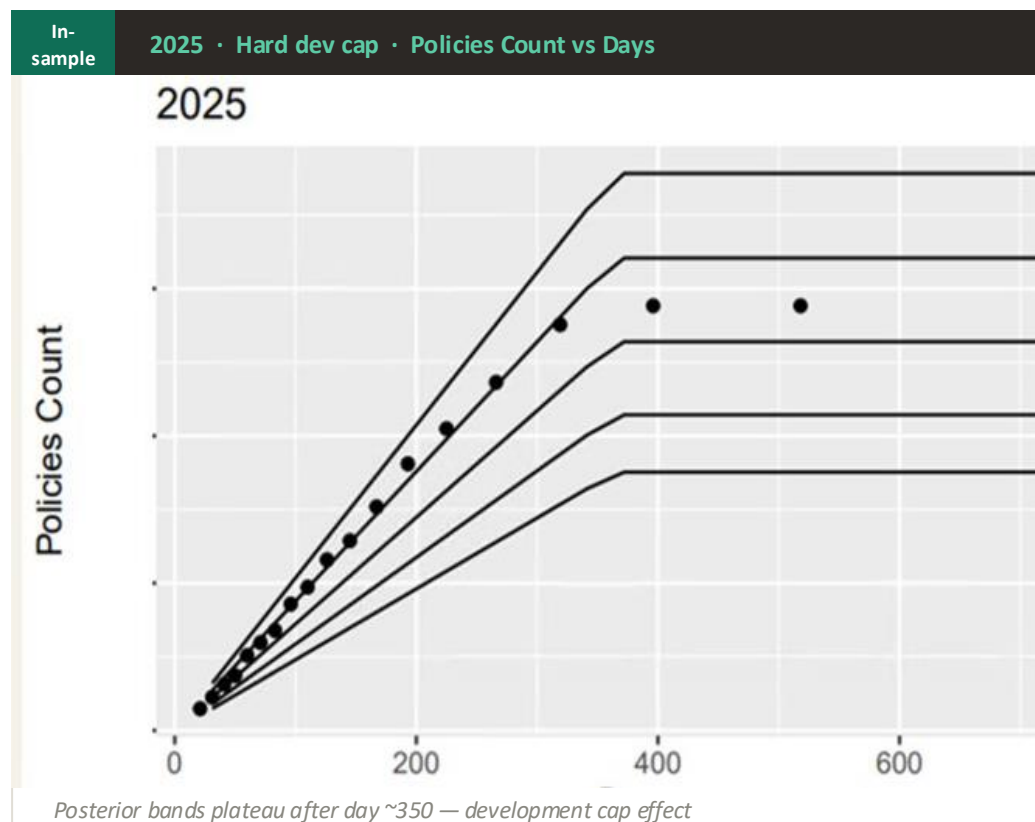
Padding effect visible in bands

In-sample: bands flatten

Cap causes plateau at ~day 350

Underprediction remains

Creative hack didn't close the gap



Diagnosing the Under-prediction

Poisson log-link bias

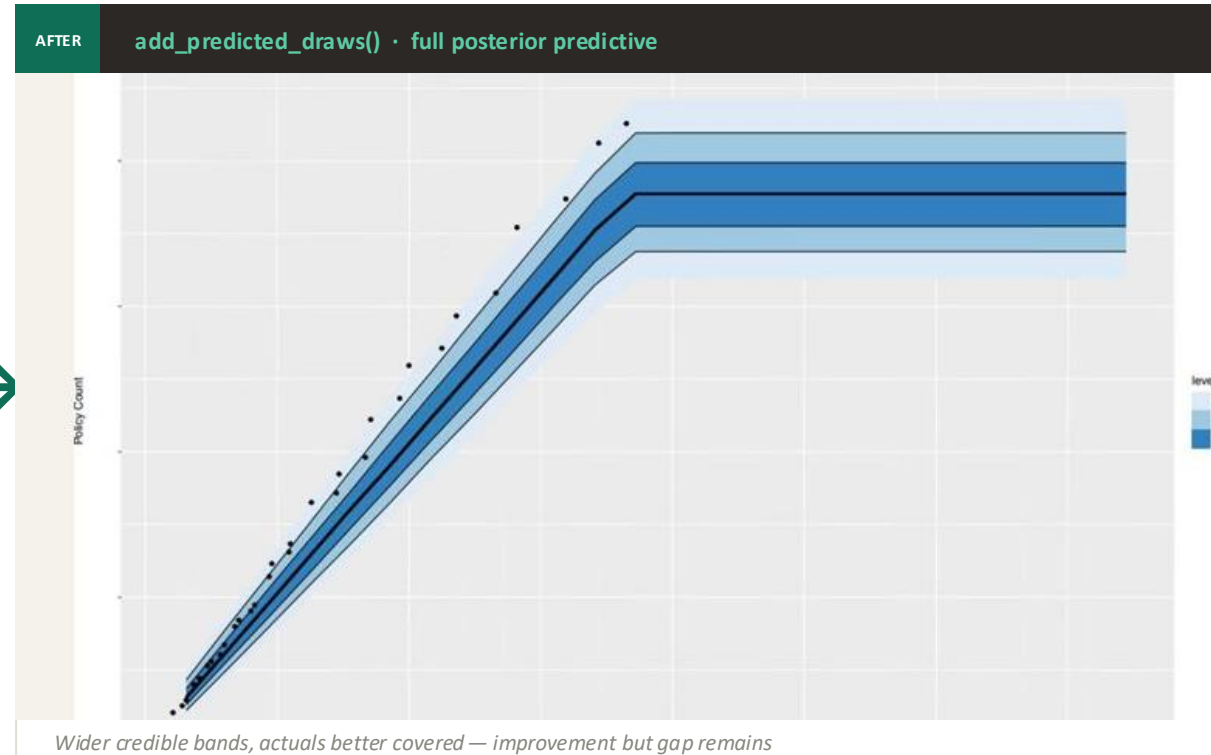
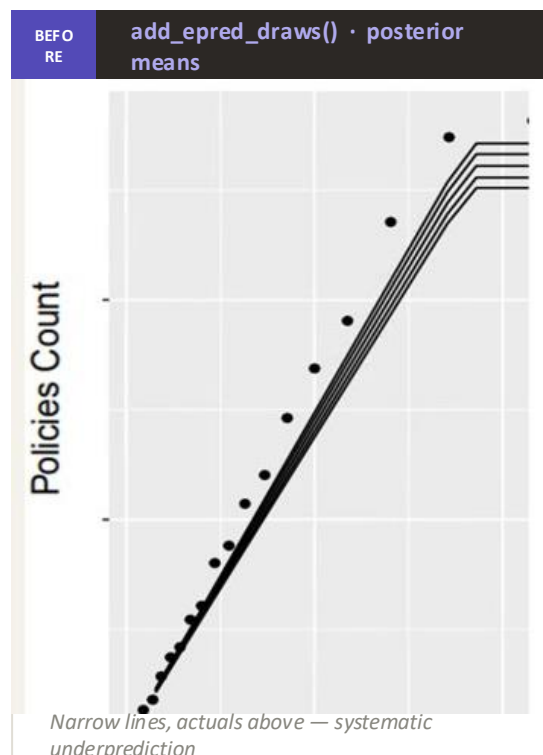
Jensen's Inequality

$$E[\exp(X)] \geq \exp(E[X])$$

Using posterior means (point estimates) to predict will systematically underpredict the mean count.

Fix: switch from posterior means to full posterior predictive draws.

● This helped — but not enough



The Fix

Switching to Negative Binomial

Poisson works when
variance = mean.

This looked exactly
like my problem —
and it's well-known
to experienced
modelers.

*A simple distribution swap.
A meaningful improvement.*

Overdispersion

PROBLEM

1

$$\text{Var}(Y) = E[Y]$$

← Poisson assumption

Insurance policy counts almost never satisfy this. Even slight overdispersion causes:

✓ *Good in-sample fit* ✗ *Underprediction out-of-sample*

The Negative Binomial Generalises Poisson

FIX

2

Poisson

$$\text{Var}(Y) = E[Y]$$

Special case —
forces mean = variance



Negative Binomial

$$\text{Var}(Y) = E[Y] + \phi \cdot E[Y]^2$$

General — allows
variance > mean

I went back to basics and tried the simplest possible fix:



swap Poisson for Negative Binomial.

Common in insurance; well-known to experienced modelers.

The Fix: Switching to Negative Binomial

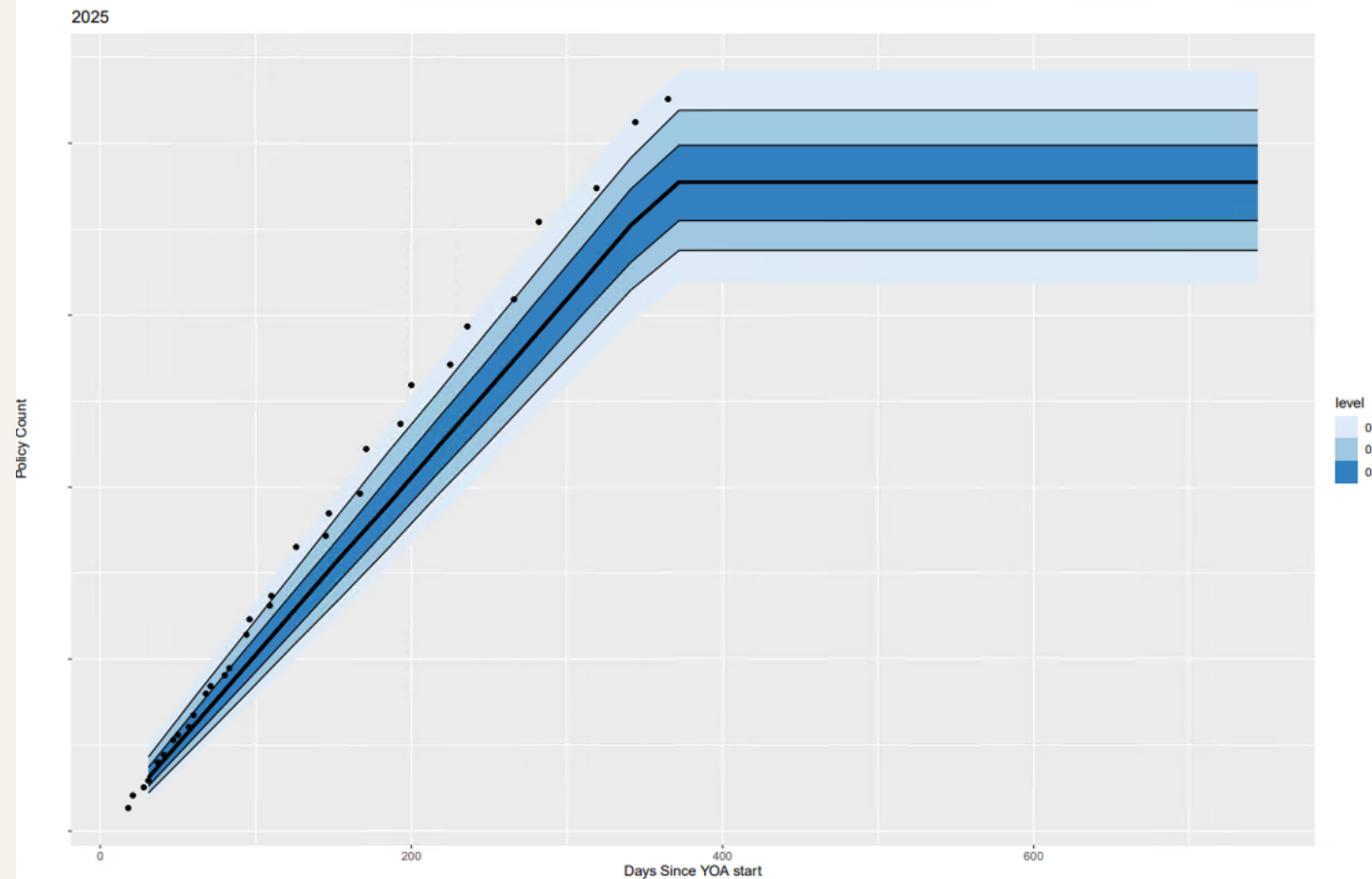
Model diagnostics

RecordType	PolicyYearOfAccount	mupos	sigmapos	mu	sigma
<chr>	<db 1>	<db 1>	<db 1>	<db 1>	<db 1>
N	2019	4.82	1.13	6.57	11.5
N	2020	4.82	1.31	10.2	32.1
N	2021	4.78	1.19	8.04	17.3
N	2022	4.92	0.995	8.86	24.9
N	2023	4.82	1.04	8.80	15.6
N	2024	4.88	1.03	8.16	13.3
N	2025	4.80	0.977	9.88	20.1
R	2019	4.83	1.21	11.6	27.2
R	2020	4.80	1.26	11.3	24.9
R	2021	4.84	1.26	14.0	40.4
R	2022	4.84	1.21	15.5	41.5
R	2023	4.87	1.16	17.0	46.5
R	2024	4.85	1.16	18.4	47.7
R	2025	4.80	1.14	19.3	49.5

Why Negative Binomial?

- Poisson is a special case of Neg Binom

Out-of-sample predictions



The Fix

Switching to Negative Binomial

2025 out-of-sample results for both Renewals and New Business.

Renewals: bands above actuals

Model overpredicts slightly at the start

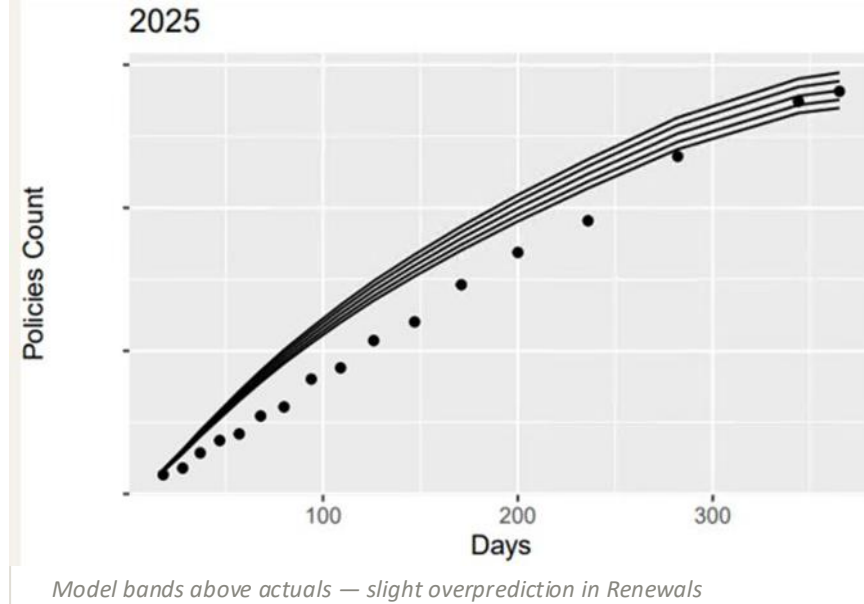
New Business: tracks well

Actuals sit within posterior

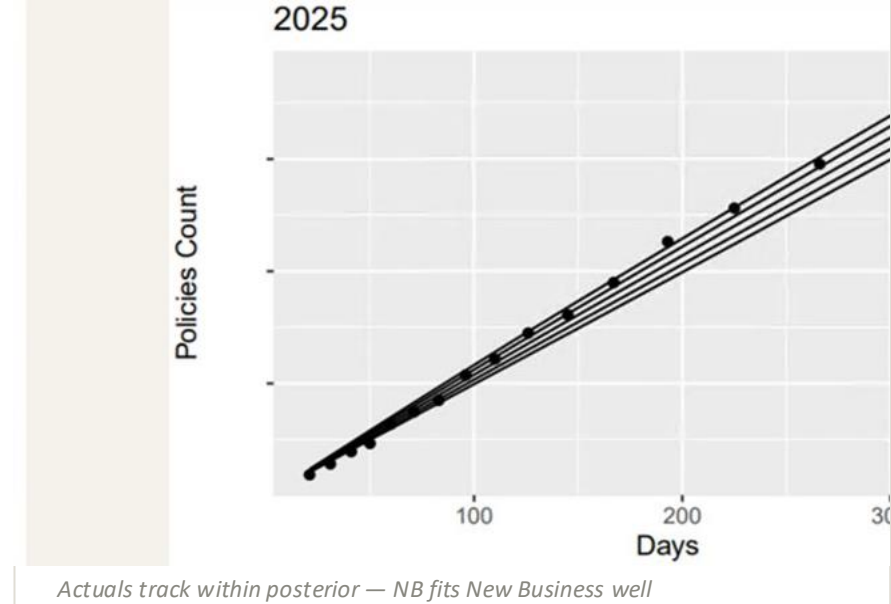
NB variance helps both

Overdispersion now accounted for

Renewals 2025 · Negative Binomial · Out-of-sample



New Business 2025 · Negative Binomial · Out-of-sample



The Fix

Switching to Negative Binomial

A clear improvement — but a new issue has emerged: modest but consistent overprediction.

Bands above actuals

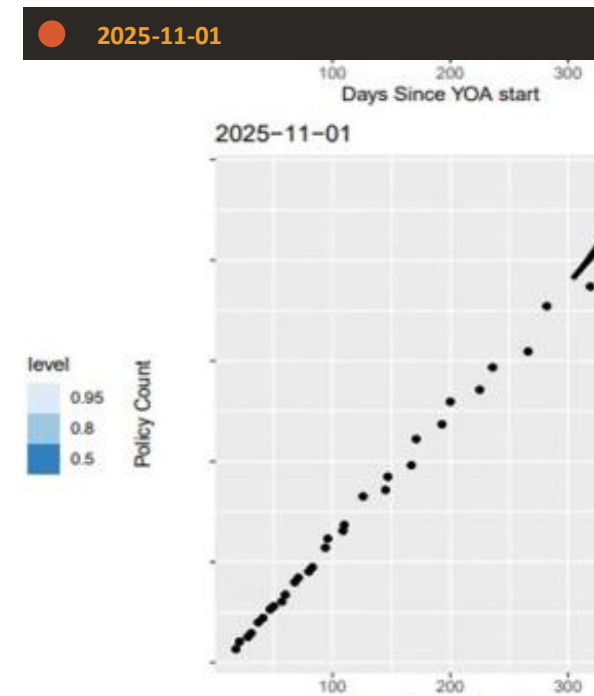
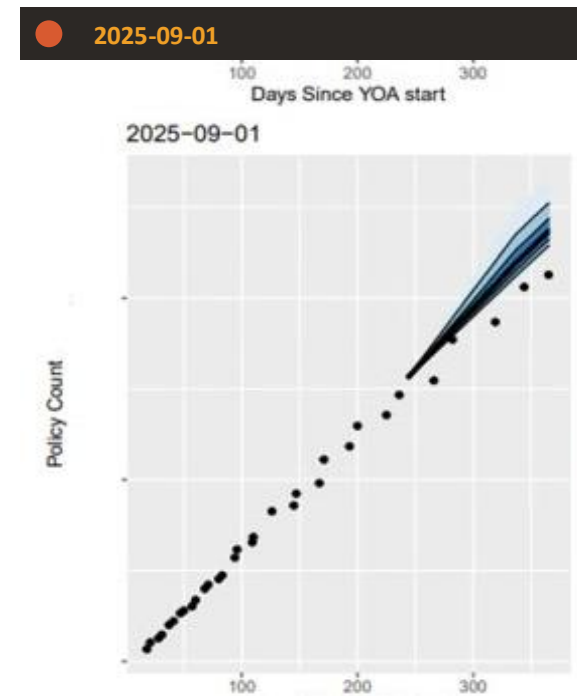
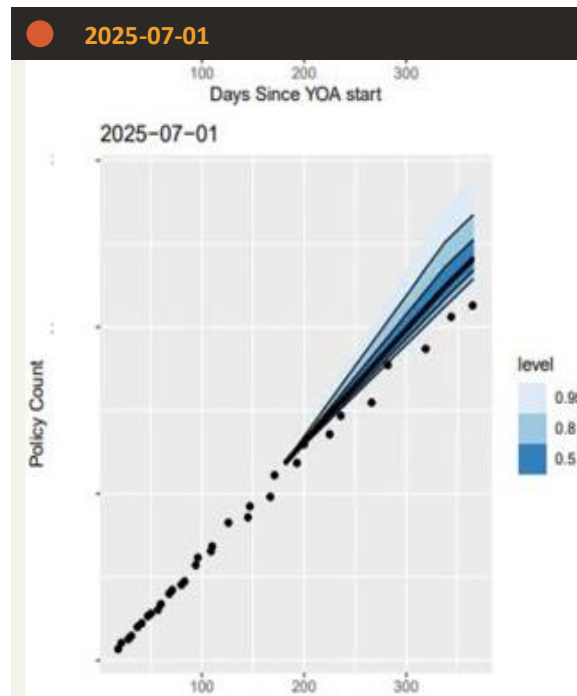
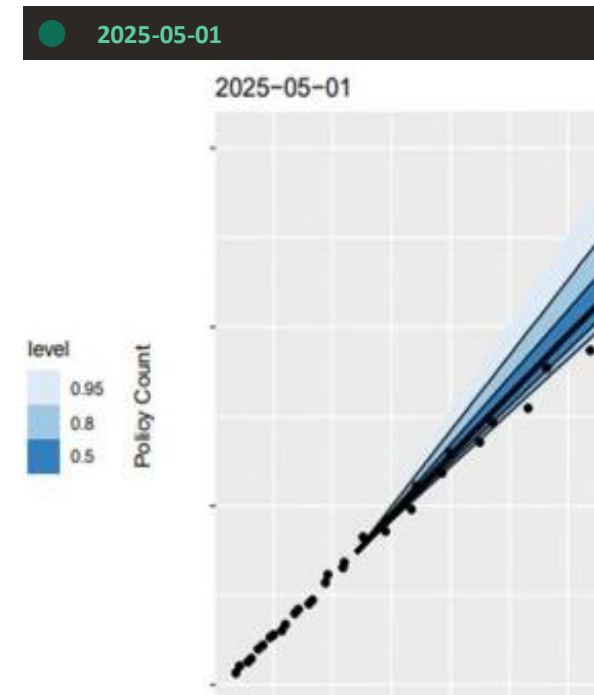
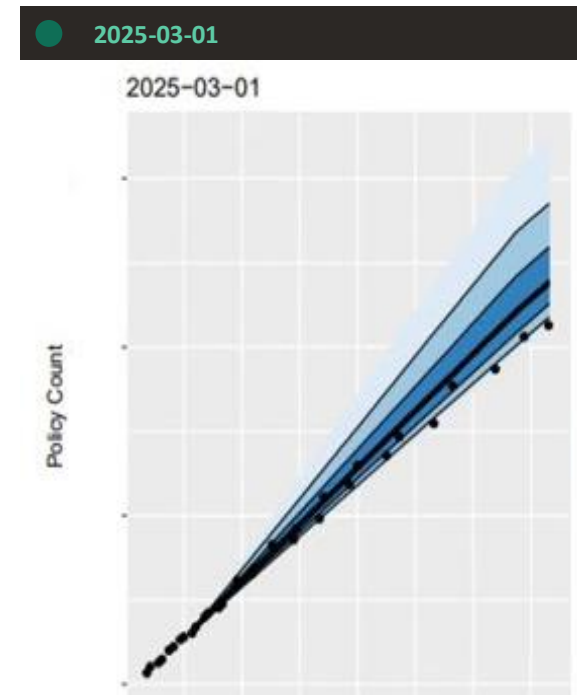
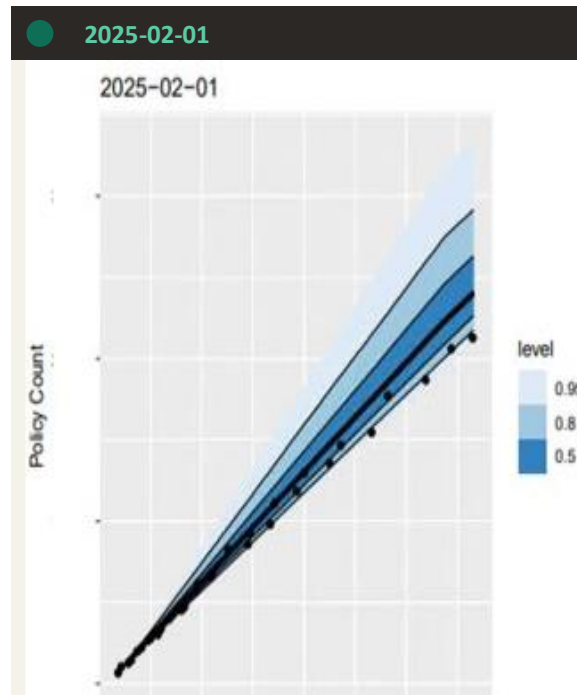
Systematic overprediction across months

Effect worsens later

Gap grows for Sep–Nov cohorts

NB clearly better

vs Poisson — but not yet solved



The Breakthrough: Taking Prior Information Seriously

The Key Insight

80–90% of this segment's policies are renewals — last year's policy count is a legitimate, powerful prior for next year.

Why Negative Binomial?

Insurance count data almost always shows overdispersion (variance > mean). Neg Binomial adds a dispersion parameter — Poisson is just a special case.

What the Formula Captures

- Last year's count (lagged, log-transformed)
- Status interaction (NB vs Renewals behave differently)
- Rolling 3-month trend
- Seasonal month-of-year effect
- Smooth time spline by status
- Random intercept per policy year (partial pooling)

The Model Formula

R

```
brm(  
  formula =  
    CountOfPolicies ~  
      1 + # intercept  
      status + # NB vs Renewals  
      log1p(Count_lag12) + # lag: last year  
      status:log1p(Count_lag12) + # interaction  
      log1p(roll3) + # 3-month trend  
      factor(month) + # seasonality  
      s(t, by = status, k = 8) + # time spline  
      (1 | PolicyYearOfAccount), # partial pool  
  family = negbinomial()  
)
```

*The winning insight: encode what you know about the **business as priors**. The best model was not the most sophisticated — it was the most **honest** about what drives the data.*

The Business-Informed Model

Results

Actual vs. Expected — this model finally delivers reliable out-of-sample performance.

In-sample: tight bands

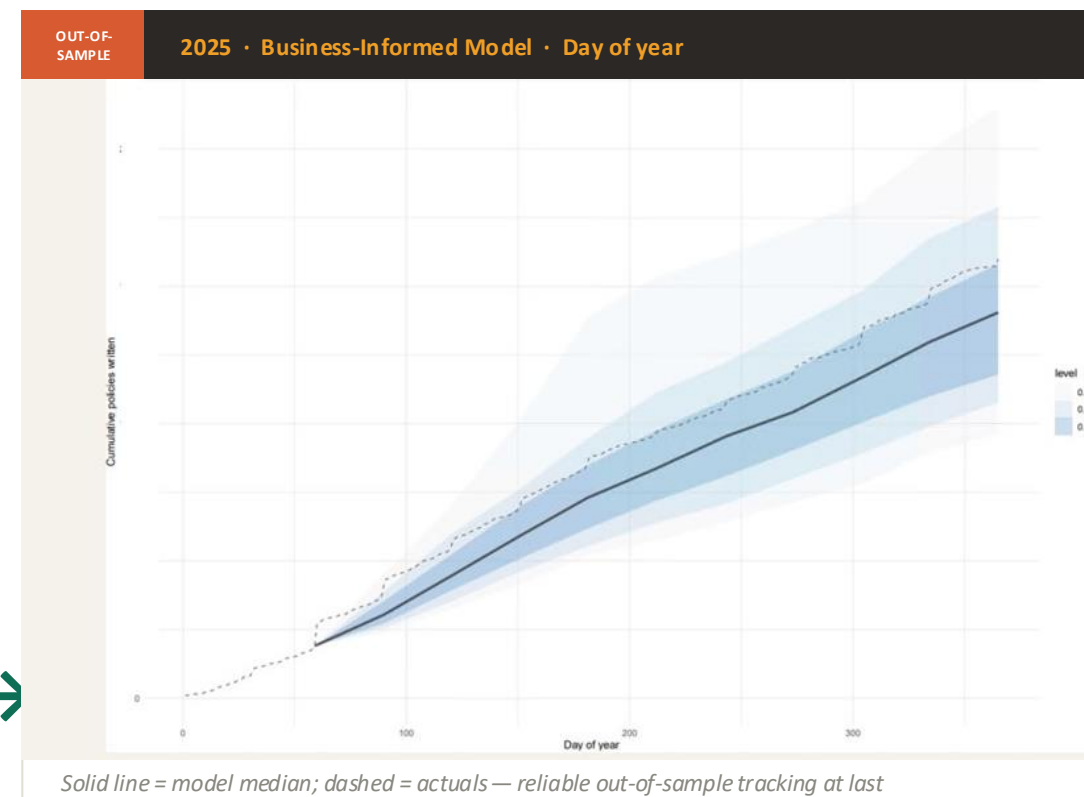
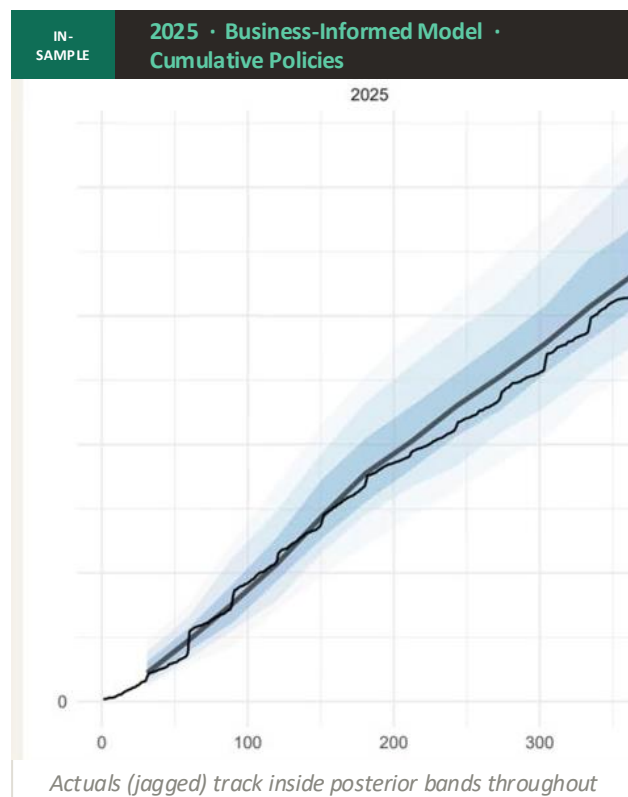
Actuals well within posterior

Out-of-sample: reliable

First model to track OOS consistently

Business prior works

Renewals structure captures the data



Two Mindsets for Bayesian Modelling

Where do you start?

MINDSET 1 Starting from the Mathematics

- What are the statistical properties of the segment?
- What is the development period?
- How do patterns change over time?
- What correlations exist in the data?
- What distributions do the features follow?
- How can I express all of this in a mathematical formula?

vs

MINDSET 2 Starting from the Business

- How does this segment actually operate?
- What is the data-generating process in reality?
- What drives policy counts from one period to the next?
- What drives year-on-year changes?
↳ 80–90% of the book is renewals — last year's data is a strong prior.
- How can real-world business logic be encoded as genuine prior information?

APPROACH 1

Formula-Driven · Academic

- Started from mathematical first principles
- Hit fundamental difficulties with raw occurrence data
- Explored cosine functions, splines, development percentiles

⚠ Usable but flawed — prone to overprediction

APPROACH 2

Business-Informed · Practical

- Asked how the segment actually works — not what the maths looks like
- Used operational knowledge as genuine prior information

“Based on last year’s data, what should we expect next year?”

✓ *Best-performing model*

What Actually Drives the Forecast:

Last year's renewals
80–90% of book

New business volume
linked to total PIF

YoY growth / contraction
observed trend

Management intent
grow · hold · shrink

“

The best model came from understanding the business first,
the mathematics second.