© 2026 Yiguedan Philippe Andy Yaro



Sealed Forecast: Grading a Company's Own Rule

Yiguedan Philippe Andy Yaro

View GitHub Repository: github.com/andyyaro/Longaeva-Hackathon

Let's connect! Email: yyaro@gmu.edu

LinkedIn: linkedin.com/in/andy-yaro

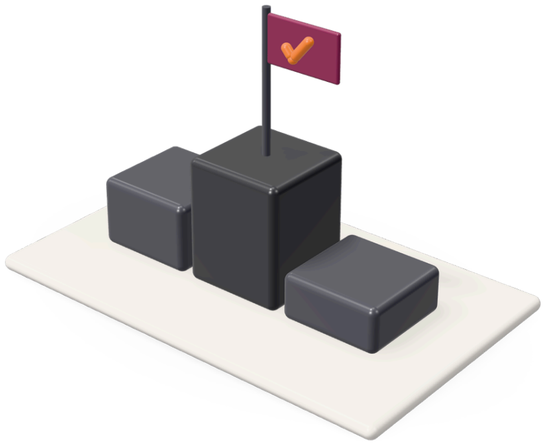


In September and October 2026 I entered the Longaeva AI Hackathon, “Businesses, Not Numbers”, run by Longaeva Partners, part of Balyasny Asset Management. My entry models one mechanism at one company: the yield Interactive Brokers (IBKR) pays on customer cash, against its stated rule: the federal funds rate, the overnight rate banks charge each other, minus 0.50%.

Four model families forecast that yield a quarter ahead on a 12,200-line library. On 8 September I committed and sealed a forecast for the third quarter of 2026. In Stage 2, the second round, a program screened 13 SEC filings, reports to the U.S. securities regulator, for such rules. All data is public.

On 3 November 2026 this deck reruns the tests and the Stage 2 screen offline, checks the seal, and scores it against the 2.31% IBKR reported on 15 October. I built it with AI assistance and made every commit. I am Yiguedan Philippe Andy Yaro.

A. The Challenge

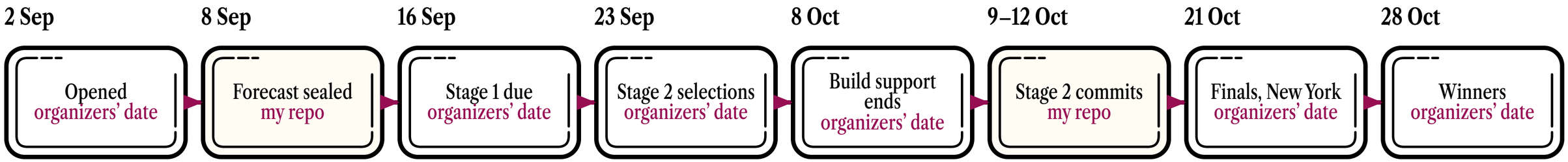


Longaeva Partners, part of the hedge fund Balyasny Asset Management, ran the hackathon for students: build a working simulation of a company that can be run forward under scenarios.

The rules asked entrants to show which signals moved their outputs, and why. Stage 1 was a written proposal; three to five finalists would present in New York.

I used public data only: filings with the SEC, the U.S. securities regulator; the Federal Reserve's rate series; and Interactive Brokers' website and monthly releases. The monthly panel's exact source is not recorded.

1. The Timeline, and What Was Sent



Stage 1, Built 16 September

An 8-page proposal, "Grading the Mechanism: Scoring What Companies Say About Their Own Economics".

Appendices on method, backtest, a 25-page claims ledger and the propriety of the scoring rules, plus a corrections log.

Word and PDF copies, for a Box file request.

Stage 2, Built 12 October

A 6-page report, a 30-page claims ledger and 6 pages of corrections.

An offline copy of the results page, and a code archive of about 40 MB kept out of git.

Not in the Repo

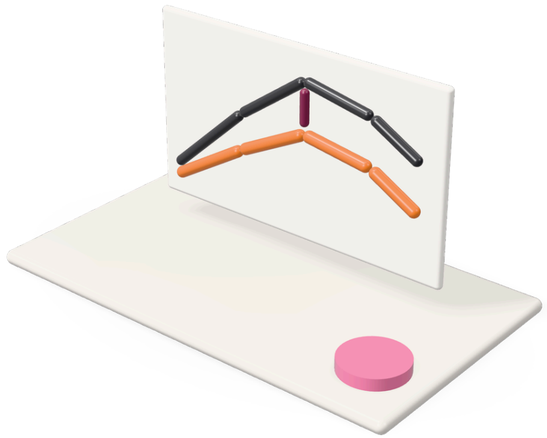
Any upload receipt, Stage 2 selection, finals or result.

Any NDA or data from Longaeva: every input is public.

Why the Stage 2 commits fall after 8 October.

Competition dates as the repo, my project's git repository, records them from the organizers' email; the seal and commit dates are from git. A commit is a saved version; a claims ledger lists each number with its data; a proper scoring rule rewards honest forecasts; NDA = non-disclosure agreement; Box shares files.

B. One Mechanism



Interactive Brokers' (IBKR's) quarterly report says "any rate cuts are passed through to our customers": its published rate on customer cash is the federal funds rate, the overnight rate banks charge each other, minus 0.50%.

Its own earnings releases show the yield it actually paid sits below that rule. The gap is the mechanism I model.

I did not model IBKR's whole business, only this one rule and what it is worth.

1. The Rule and the Wedge

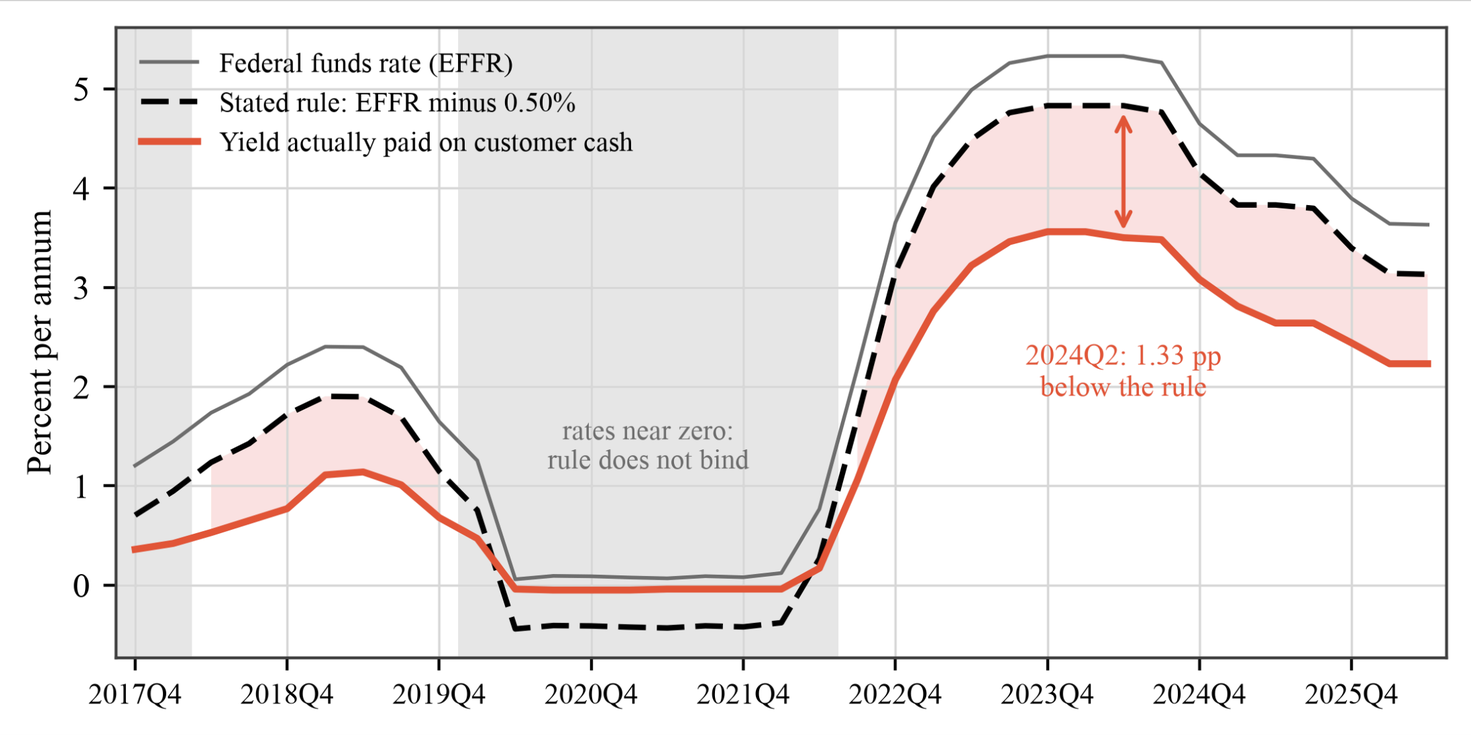


Figure from the Stage 2 report: IBKR earnings releases and the Federal Reserve's daily rate. The grayed quarters, where the rule is 1.00% or less, are excluded.

paid: the yield IBKR actually paid on average customer cash, from each quarter's earnings release.

rule: EFFR − 0.50, the published rate for IBKR Pro accounts.

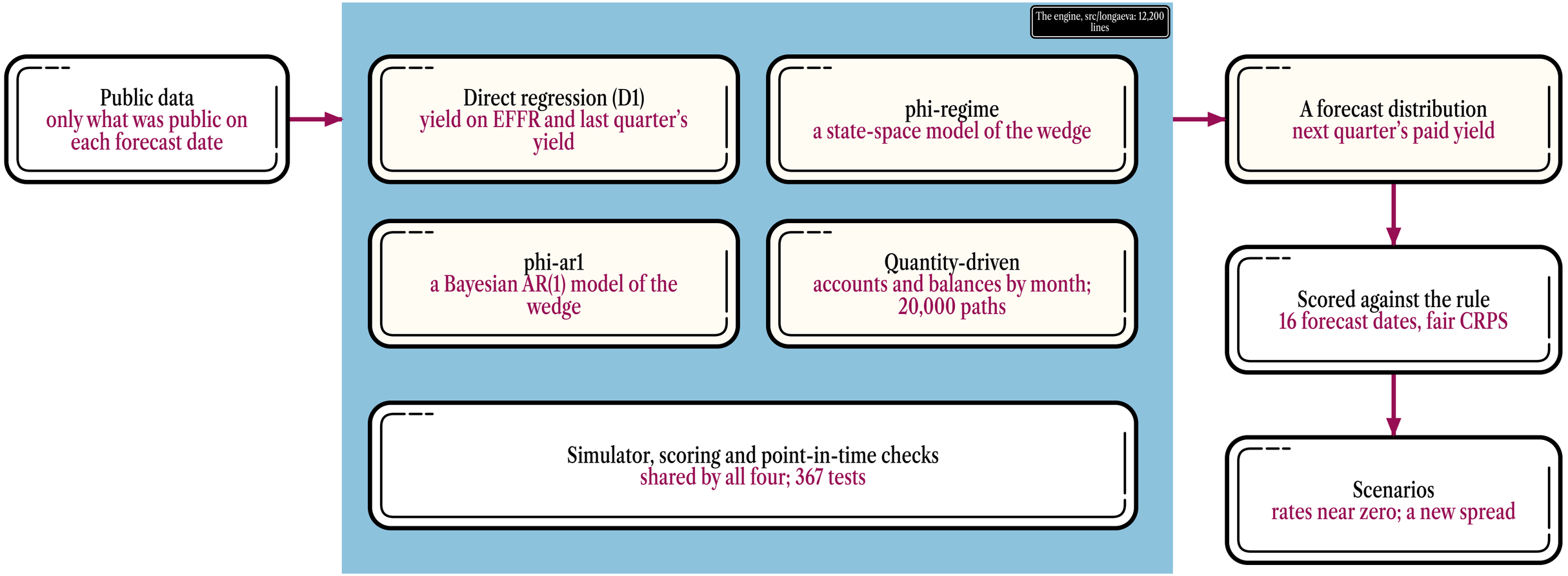
wedge φ = 1 − paid ÷ rule: the share of customer cash the rule does not reach. True by definition, not estimated.

Quarters where the rule was 1.00% or less, which include the near-zero years, are dropped rather than forced to a value.

IBKR names three reasons for the gap; the data cannot split it among them.

EFFR = the effective federal funds rate, the overnight rate banks charge each other
φ (phi) = the Greek letter used for the wedge
IBKR Pro = IBKR's main account type
earnings release = the quarterly results IBKR files as an 8-K
8-K = a current report filed with the SEC, the U.S. securities regulator

2. Four Model Families, One Engine

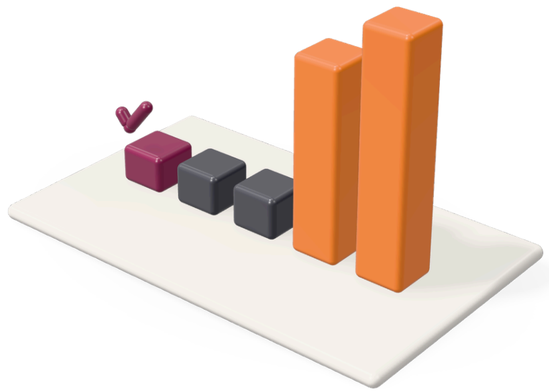


The four families were built separately, by what the docs call four "teams", and ranked by an adjudication step: an AI-assisted workflow. The quantity-driven family uses a monthly data panel whose source is not recorded.

: My engine : Data or results

regression = a fitted straight-line relation
state-space = a hidden level that moves over time
AR(1) = each value depends on the last one
paths = simulated futures
point-in-time = no data dated after the forecast
adjudication = scoring and ranking the families

C. The Backtest



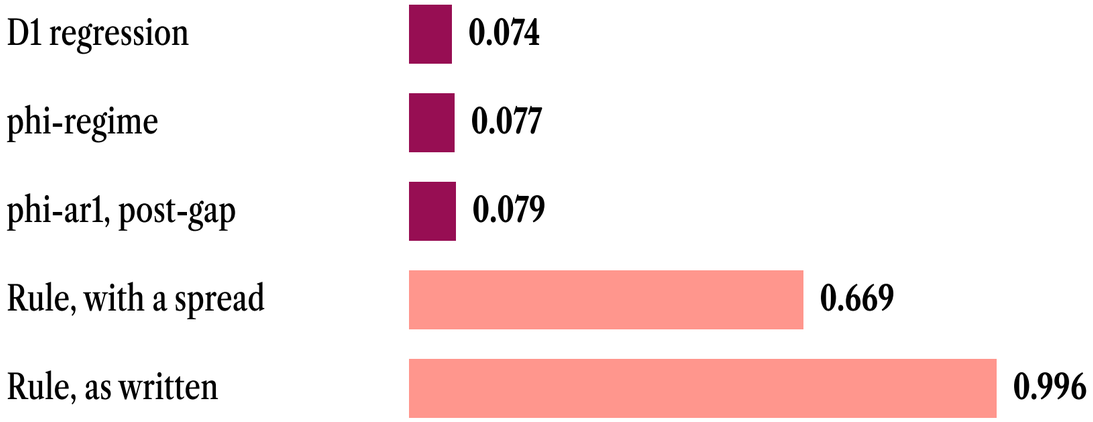
In this backtest, each model forecast the next quarter's yield from 16 past dates, using only data public at the time, and was scored against what Interactive Brokers (IBKR) then paid.

The company's rule scored about nine times worse than the best model. My structural models, built on the wedge, did not beat a plain regression.

Both results are in the Stage 1 package, the proposal and its backtest appendix, with the tests behind them.

1. Scores on 16 Forecast Dates

Fair CRPS, 16 forecast dates (lower is better)



The rule states a single number, so it was scored two ways: as written, and with a spread added around it. D1 had the lowest score of 36 scored versions, including the rule's readings and simple baselines.

Why the Rule Loses

Diebold-Mariano tests, which compare two forecasts' accuracy, reject equal accuracy: p of 0.00000001 or less.

The rule ranks quarters as well as the models, as any rate-based forecast must; it loses on calibration, by sitting too high.

A bet that D1 beats the rule grew from $1 to $51.97 at a bound of 2, past the $20 threshold; at 5 or more, none of 24 such comparisons reaches $20. I fixed 2 for Stage 2 after seeing this.

Honest Negatives

Structural against regression: p = 0.72. The regression's small lead, an effect size of 0.11, is far below the 0.75 this sample could detect.

The effective sample is about 6 to 7 quarters, not 16, and a constant wedge is firmly rejected.

On held-out quarters, not forecasts: near-zero rates, structural error 0.044 against 0.626 for the regression; mid-range rates, it loses, 0.236 against 0.105.

fair CRPS = scores a whole forecast distribution; next slide
p = chance of a result this strong if there were no real difference
error = mean absolute error, in percentage points
calibration = being right on average
bet, bound = the e-process and its fixed cap on each quarter's loss
effect size = a difference in standard units

2. How the Forecasts Were Scored

1. **CRPS, continuous ranked probability score**
Scores a whole forecast distribution against the outcome. For a single number it equals the absolute error; lower is better.

2. **Fair CRPS**
With a finite set of simulated futures, or paths, plain CRPS favors forecasts that are too narrow. The fair version, from Ferro (2014), removes that bias.

3. **Isotonic decomposition**
Splits the score into miscalibration (bias), minus discrimination (ranking skill), plus uncertainty (the outcomes' spread), each zero or more, using plain CRPS.

4. **Diebold-Mariano test**
Tests whether two forecasts are equally accurate, with the Harvey-Leybourne-Newbold correction for small samples.

5. **E-process**
A bet against a claim that starts at $1 and is sized each quarter from past quarters only. Reaching $20 rejects the claim at the 5% level, whenever betting stops.

6. **Leakage harness**
Replaces all data after each forecast date with markers and deliberately corrupts one quarter. It caught a model reading future data two quarters out.

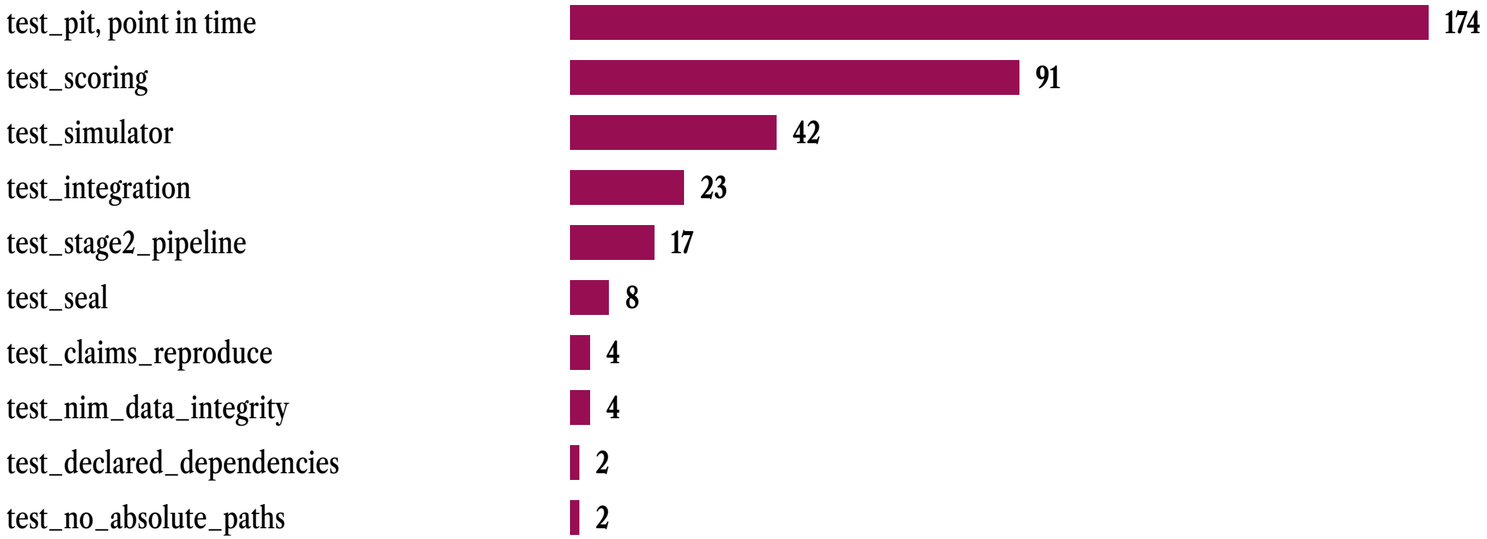
Definitions from the engine's README and the backtest appendix. The harness: 1,392 of 1,457 model-and-date runs reacted to the corrupted quarter, and all 1,457 came back clean after the fix, which changed no ranking.

3. 367 Tests, 66 Claims, 22 Figure Numbers

```
$ unshare -rn python3 -I -m pytest -q -p no:cacheprovider 2>&1 | tail -1
367 passed, 2 warnings in 47.14s
$ unshare -rn python3 -I ../checks/run.py scripts/verify_claims.py | tail -1
all 66 derivable claims reproduce from committed data
$ unshare -rn python3 -I ../checks/run.py figures/src/verify_figures.py | tail -1
22 of 22 figure numbers re-derive from committed data.
```

1. 367 tests pass offline in 47 seconds; the project's README still says about 25.
2. All 66 numbers in the claims ledger re-derive from committed data.
3. All 22 numbers drawn on the 7 figures match the ledger.

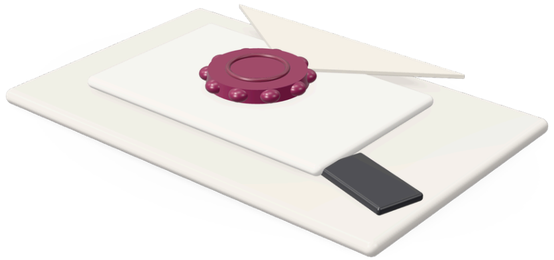
Tests per file



Run on 3 November 2026 in a fresh copy of the repo with no network. verify_claims re-derives the ledger from committed data; it does not re-run the backtests, whose 38 MB cache is not in git.

ledger = each number with its source
pytest = Python's test runner

D. The Sealed Forecast



On 8 September 2026, eight days before Stage 1 was due, I committed to git four forecast files, one per model family, for Interactive Brokers' (IBKR's) customer-cash yield in the third quarter.

Two days later I committed their SHA-256 digests, fingerprints that change if a file does. IBKR reported 2.31% on 15 October.

The repo never scored the forecast. This deck does, from the four files as committed.

1. What Was Sealed, and How

1. 8 September, 17:37 US Eastern: all four files enter in one commit, each with a median and an 80% interval.
2. The federal funds rate forecast already had 65 days of the quarter observed; the rule's forecast, 3.13%, is the same in all four files.
3. 10 September: a file of SHA-256 digests is committed; 8 tests guard the four files.
4. The Stage 1 proposal quoted 2.17% for the best model and 3.13% for the rule; if its upload reached the organizers, that dates both numbers.
5. The target: the quarterly yield in IBKR's earnings release, due mid-October.

What the Seal Cannot Prove

Commit dates are set by the author; the repo records no outside timestamp.

The proposal says the digests were "committed and emailed"; the repo shows only an unsent draft.

"Pre-registered", my notes' word, overstates it: nothing outside git records the seal.

My notes cite monthly data; the target is quarterly.

i

median = the middle of the forecast
80% interval = 10th to 90th percentile
commit = a saved version in git
SHA-256 digest = a standard fingerprint of a file
pre-registered = a plan recorded with an outside party before the data exist
earnings release = IBKR's quarterly results, filed as an 8-K

2. Unchanged Files; Three of Four Cover 2.31

```
$ git log --date=iso --format='%h %ad %s' -- experiments/direct-yield/live_forecast.json \
    experiments/phi-ar1/live_forecast_2026Q3.json \
    experiments/quantity-driven/live_forecast_2026Q3.json \
    experiments/phi-regime/results/live_2026Q3.json
9b281e8 2026-09-08 17:37:42 -0400 Fix CorrelatedShocks silently returning an unrequested correlation
$ sha256sum -c experiments/SEALED-2026Q3.sha256
experiments/direct-yield/live_forecast.json: OK
experiments/phi-ar1/live_forecast_2026Q3.json: OK
experiments/quantity-driven/live_forecast_2026Q3.json: OK
experiments/phi-regime/results/live_2026Q3.json: OK
$ python3 -I ../checks/score_sealed.py 2.31
outcome: 2.31%
sealed model                      median     80% interval    error  outcome
D1 regression (best backtest)      2.173   2.070 to 2.272   0.137  above
phi-regime (structural)            2.227   2.105 to 2.345   0.083  inside
phi-ar1, post-gap                  2.241   2.067 to 2.418   0.069  inside
quantity-driven, Q-shrunk          2.229   2.129 to 2.318   0.081  inside
company rule, EFFR - 0.50          3.129      no interval   0.819  (median in the 4 files: 3.1297, 3.1293, 3.1287, 3.1287)
```

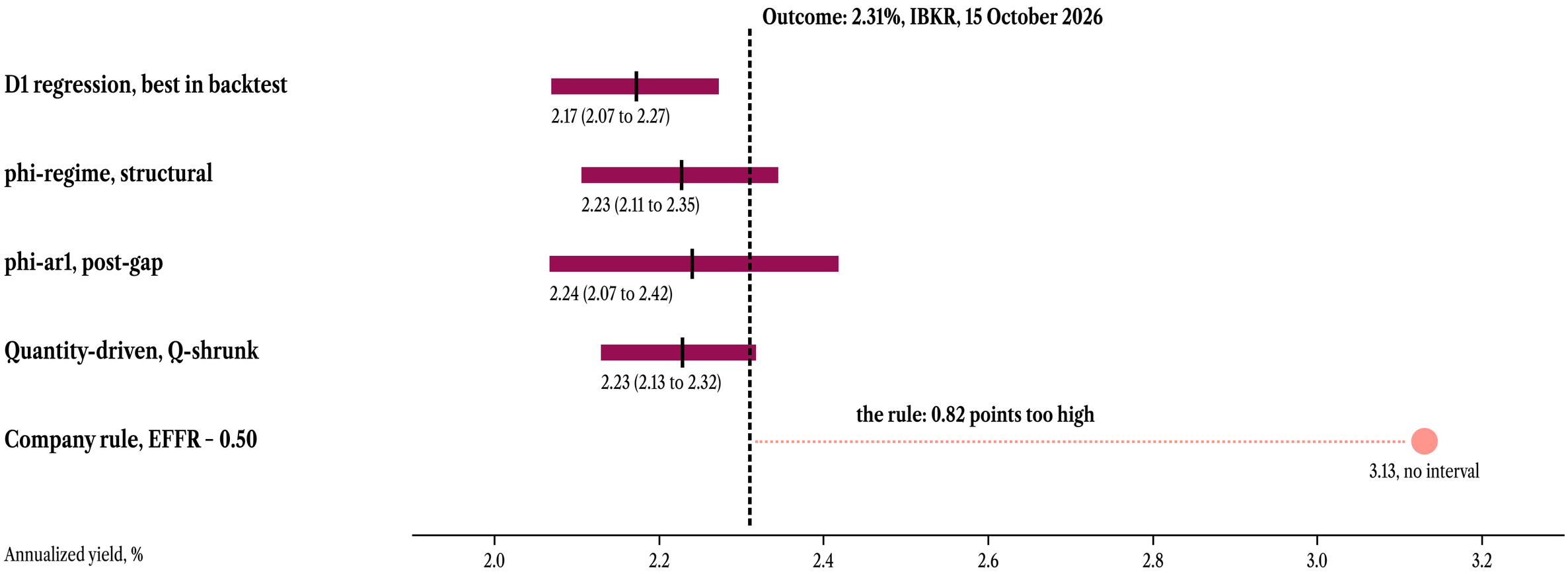
1. One commit, on 8 September 2026, is the only one that touches the four files.
2. Their digests still match: unchanged since, within a history I control.
3. D1, best in the backtest, missed: 2.31 is above its interval. The other three hold it; the rule is 0.82 too high.

Run on 3 November 2026. score_sealed.py, written for this deck, reads each file's median and 10th and 90th percentiles and takes the outcome as an argument: 2.31%, the annualized yield on customer credit balances in IBKR's third-quarter 2026 earnings release, 8-K Exhibit 99.1, 15 October 2026. The rule states no spread, so only its median is scored. The sealing commit's message describes an unrelated code fix.

i

git log = lists the commits that changed a file
sha256sum -c = rechecks files against stored digests
error = distance from the median to 2.31
D1, phi-regime, phi-ar1, quantity-driven = the four model families
8-K Exhibit 99.1 = the earnings release attached to IBKR's 8-K

3. How the Sealed Forecast Turned Out



D1, the best model on the backtest, missed: its median was 0.14 too low, and the top of its interval 0.04 below the outcome. The other three families' intervals held 2.31. The rule missed by 0.82 points: the wedge persisted. One quarter proves nothing; the backtest's effective sample is about 6 to 7 quarters.

Medians as ticks and 10th-to-90th percentiles as bars, the 80% intervals, from the four files sealed on 8 September 2026; outcome from IBKR's 8-K of 15 October 2026. EFFR = effective federal funds rate; the wedge = the share of customer cash the rule does not reach.

E. Stage 2: A Filing Screen

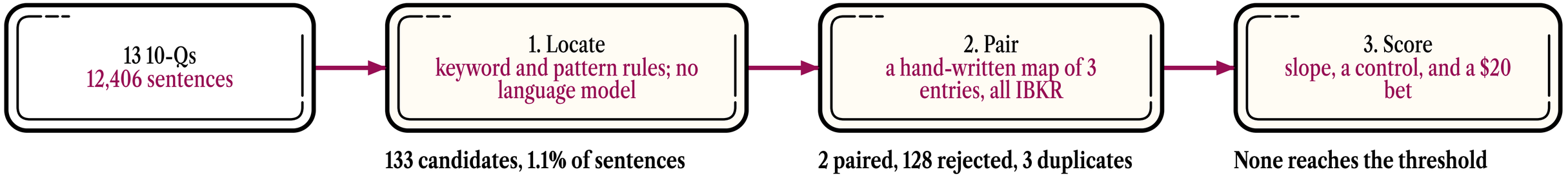


In Stage 2 I tried to widen the test: find rules like Interactive Brokers' (IBKR's) in other companies' filings, pair each with public data, and grade it.

A three-step screen, a program that filters text, read 13 quarterly reports, 10-Qs, for the period ending 30 June 2026. It found 133 candidate rules and could grade two.

It also re-tested IBKR's rule on quarter-to-quarter changes, and I withdrew five earlier claims, three of them from the Stage 1 proposal.

1. Three Steps Over 13 Filings



Recall

8 of 8 hand-labeled claims in IBKR's 10-Q, the filing the rules were tuned on.

0 of 3 labeled later, in Capital One's and Ally's filings.

Rate Grid

IBKR's website pricing grid, not a filing: 170 rows, 99 tied to a named benchmark rate.

Schwab's page gave no table; Robinhood's was not found.

Why Only Two

127 candidates had no public data series mapped to them.

Both pairs are IBKR's own, and neither matches its claim's population.

i

10-Q = a company's quarterly report to the SEC, the U.S. securities regulator
recall = the share of known claims found
benchmark = a reference rate, such as the federal funds rate
population = which customers and currencies a number covers
tuned = adjusted until it worked on that filing
$20 bet = the e-process threshold for the 5% level; slide C2

2. The Screen, Rebuilt Offline

```
$ unshare -rn python3 -I ../checks/run.py stage2/run_all.py | grep -E 'documents, 12406|candidates \
    \(|recall (in|out)|MISSED|paired,|changes: beta|e-process \(c|-> RULE'
  13 documents, 12406 sentences                                                          (1)
  133 candidates (1.1% of sentences)
  recall in-sample      8 of 8
  recall out-of-sample  0 of 3                                                           (2)
    MISSED COF   cof_deposit_beta
    MISSED COF   cof_deposit_beta_definition
    MISSED ALLY  ally_deposits_reprice
  2 paired, 128 rejected, 3 duplicate restatements                                       (3)
    changes: beta 0.7113, n 21, t vs 1 -7.44 (OLS -7.44, Newey-West -13.67), p < 1e-6
    e-process (c = 2): max 2.35, anytime p 0.426, does not clear 5%; common population no -> does not pass the Stage 1 bar
    -> RULE REJECTED
    changes: beta 0.8570, n 21, t vs 1 -3.32 (OLS -3.32, Newey-West -6.98), p 0.0036
    e-process (c = 2): max 1.7, anytime p 0.59, does not clear 5%; common population no -> does not pass the Stage 1 bar
    -> RULE REJECTED
$ git status --short stage2/ | wc -l
0                                                                                        (4)
```

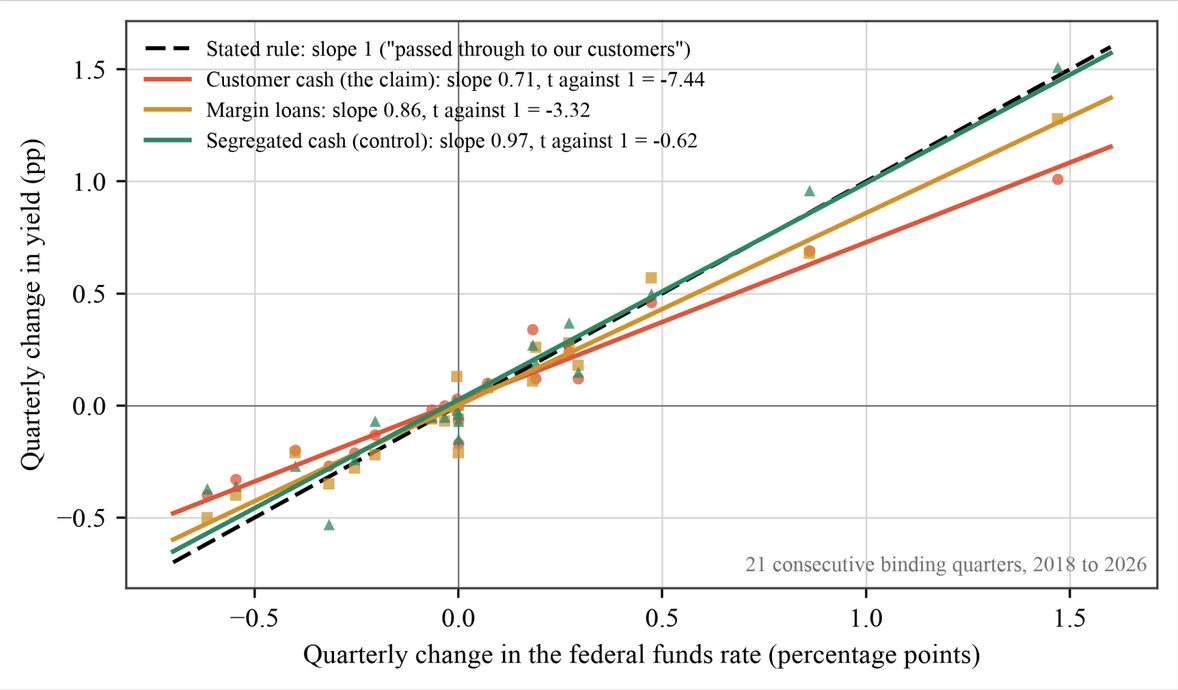
Run on 3 November 2026 in a fresh copy of the repo, offline, on the committed filings. run_all.py prints “RULE REJECTED” from the t-test. The evidence against the claim still does not meet the stricter Stage 1 standard: the bet must reach $20 and the data must cover the same customers as the claim. It reaches $2.35, and the populations differ.

1. 12,406 sentences in, 133 candidate rules out.
2. All 8 hand-labeled IBKR claims, in the filing the rules were tuned on; none of the 3 labeled later elsewhere.
3. 0.71 of each rate move reaches customers. A t-test rejects 1.00, but the bet set in advance peaks at $2.35 of the $20 threshold.
4. Rebuilding the screen changes no file under stage2/.

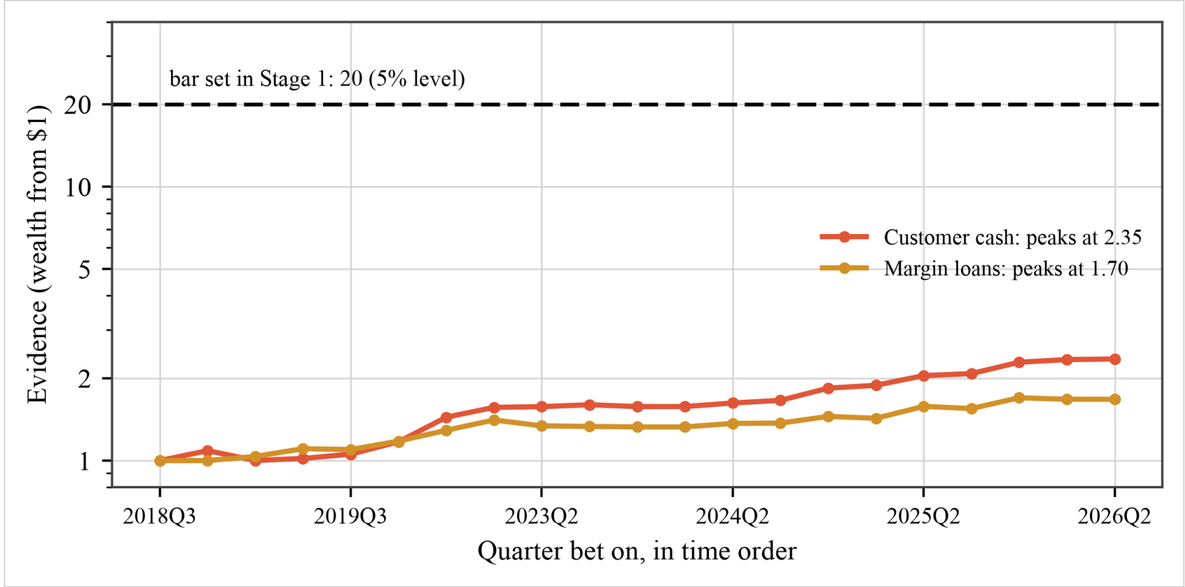
i

beta = the slope: yield change per one-point rate change
t vs 1, t-test = how many standard errors below 1.00
Newey-West, OLS = two ways to compute standard errors
n 21 = 21 quarter-to-quarter changes
e-process = a bet against the claim, sized from past quarters
recall = the share of known claims found

3. What the Re-Test Found



Only 0.71 of each rate move reaches customer cash. The control, segregated cash, which IBKR places in the market with no pricing rule of its own, moves 0.97, so the test is not rejecting everything.



The bet, the e-process, needs $20; it peaks at $2.35. Small rate moves over 21 changes do not build enough evidence under the bound, a fixed cap on each quarter's loss, and I did not loosen it.

What the Gap Is Worth

About $496 million a year for each one-point rate move, on $171.7 billion of cash; about $250 million a year net of the margin-loan gap, and about $53 million of that to holders of IBKR's listed shares, after its holding structure and taxes. IBKR's own, broader figure: $81 million a year of net interest income for each 0.25-point move in U.S. dollar rates.

4. Withdrawn Claims, and Two Side Tools

Withdrawn in Stage 2

"$344m a year" as a mispricing the market missed: IBKR publishes how its income moves with rates.

A symmetry prediction: any roughly linear rule gives a symmetric answer.

A binomial (sign-test) p-value for the yield sitting below the rule every quarter: those quarters are not independent.

IBKR's published rate grid as a floor: accounts under $100,000 are paid less.

Margin loans as a true negative, a case with nothing to find: in quarterly changes their slope, 0.86, is rejected too.

13F Analysis

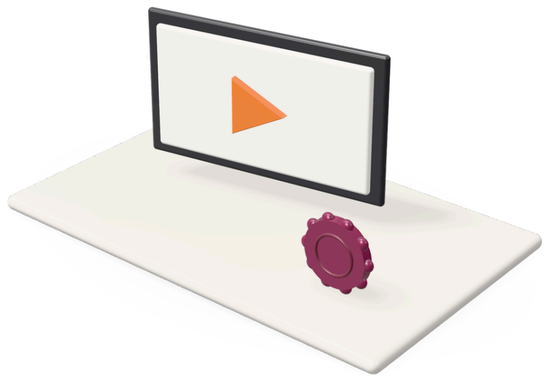
Longaeva's own public 13F filings, 2025 to 2026: in 2026Q2, 326 stock names worth $3.46 billion; the $4.58 billion reported total counts $1.11 billion of options at the value of their shares.

PDF Hidden-Content Scan

Checks the organizer's two legal PDFs for invisible text and active content such as scripts, and recovers their text. Both clean; its first run wrongly reported no text.

From the Stage 2 report, the corrections log, which holds 34 numbered corrections, and the two scripts' output. One correction: pass-through is now measured in quarterly changes, 0.7113, not levels, 0.7998. A 13F is a fund manager's quarterly list of U.S. stock holdings; the two tools were background research on the organizer, not part of the model.

F. History and the Film



88 commits from 7 September to 2 November 2026, all mine, 80 of them on five days.

The work was AI-assisted: the docs log machine research passes, model-building teams and an adjudicator, and the film's commit, a saved version in git, credits Claude.

On 1 and 2 November I added a logo and a 68-second launch film.

1. 88 Commits, One Author, AI-Assisted

```
$ git log --date=format:%Y-%m-%d --format=%ad | sort | uniq -c
      1 2026-09-07
     25 2026-09-08
      9 2026-09-09
     17 2026-09-10
     15 2026-09-16
     14 2026-10-09
      5 2026-10-12
      1 2026-11-01
      1 2026-11-02
$ git log --format=%an | sort | uniq -c
     88 Andy Yaro
$ git log --format=%B | grep -c 'Co-Authored-By: Claude'
1
$ git tag
film-v1
```

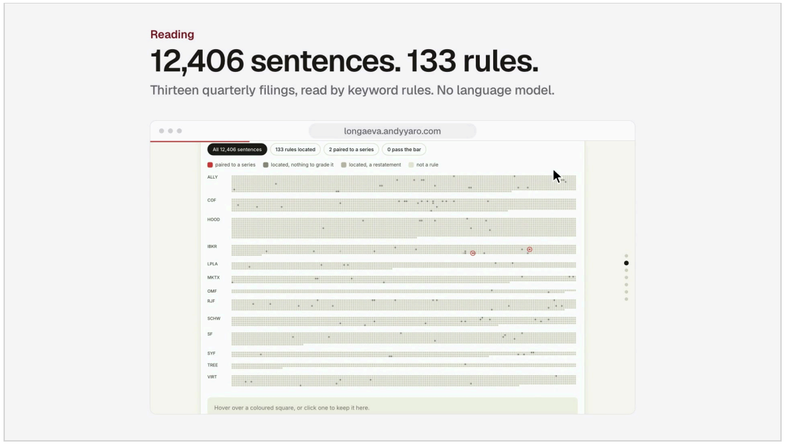
Run on 3 November 2026 on a copy of the repo. Dates are as recorded, US Eastern. A Co-Authored-By line credits a co-author; film-v1 is the tag, a named version, for the film's release.

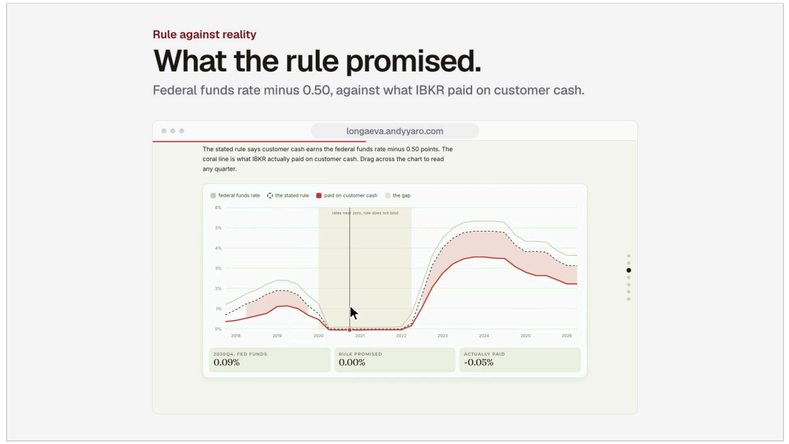
1. 7 to 16 September: 67 commits, up to the Stage 1 package.
2. 9 and 12 October: 19 commits for Stage 2, after the 8 October date the organizers gave; the repo does not say why.
3. All 88 commits are mine.
4. One commit, the film, credits Claude Opus 5.5 as co-author.

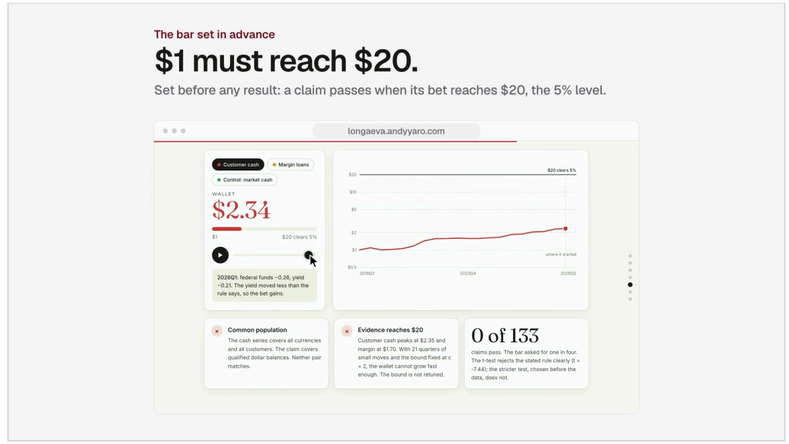
How the Work Was Done

On 7 September, 31 research and verification "passes" logged 272 findings and 285 judgments; 33 more followed on 8 September. Later docs name ranking "lenses", "critics", four model-building "teams" and an "adjudicator". They also cite an "independent verifier" and an "external review" without saying who.

2. The Launch Film and the Results Page







The Film

67.6 seconds, 15.3 MB, with 12 caption lines; a file attached to a release of the private repo, so not public.

Recorded from the live results page, composed with HyperFrames and GSAP, narrated by the Kokoro-82M voice model.

Made in a Claude Code session, Anthropic's coding agent; on screen it says "Not scored in the repo yet".

The Results Page

longaeva.andyyaro.com loaded on 3 November 2026: 120,050 bytes, identical to stage2/ui/index.html in the repo.

One square per sentence, the rule quarter by quarter, a rate slider, the test set in advance, the dollars and the withdrawn claims.

HyperFrames renders web pages to video; GSAP is a JavaScript animation library. The music and sound effects are licensed for use inside the film only.

G. Proof and Limits



Checked on 3 November 2026: the tests, the claims ledger, the figures, the Stage 2 screen, the seal and its score, the history and the live page.

Three of the four sealed families held the outcome; the company's rule did not.

The last slides set my notes against the repo, list risks and fixes, and say what the work shows.

1. My Claims Against the Repo

My notes said	The repo shows	So I now say
A company-outcome model anchored on IBKR	It models one mechanism: the yield paid on customer cash	A model of one IBKR mechanism, against its own rule
LPL Financial as the transfer case	Named on 8 September; no model was ever run on it	LPL was planned as a second company, not built
A sealed, pre-registered forecast from monthly data	A quarterly target; nothing outside git records the seal	A forecast committed on 8 September and hashed on 10 September
Fair CRPS, isotonic decomposition, Shapley sensitivity	Shapley ran only on synthetic data	Fair CRPS, isotonic decomposition, Diebold-Mariano, e-process
A filing screen: locator, recall, rate grid, pairing	All four run; recall 0 of 3 out of sample; the grid is a web page	133 candidate rules located, 2 paired, none reached the threshold
128 gapless months of operating metrics	True, but the panel's source is not recorded	As stated, source unverified
13F analysis; a hidden-content PDF scan	Both run; both PDFs clean	As stated
86 commits, solo	88 commits, all mine; AI-assisted	88 commits, sole author, AI-assisted
Results page at longaeva.andyyaro.com	Loaded on 3 November, identical to the repo	As stated
(not in my notes)	Scored for this deck against IBKR's 8-K: D1 missed; three of four held 2.31	Add the scored result

From my career-hub notes, checked against the repo and IBKR's 8-K on 3 November 2026. IBKR = Interactive Brokers; LPL = LPL Financial; transfer case = a second company to run the model on; pre-registered = recorded with an outside party in advance; Shapley sensitivity splits a forecast's variance among its inputs.

2. Risks Found in This Review

1. **Third parties' personal data**
6,615 FINRA BrokerCheck records of named advisers are committed; the repo's own note finds no bulk-download license.

2. **Profiles of real people**
Research notes profile named judges and organizer staff, and name university library staff.

3. **The organizer's documents**
Copies of its legal PDFs, which bar republishing site content, and an email marked confidential are committed.

4. **The seal's date is self-reported**
Commit dates are mine to set, and nothing shows the digest email was sent.

5. **Unverified inputs**
The monthly panel and 19 of 20 rate-sensitivity rows have no committed source.

6. **Withdrawn figures in the repo**
Two Stage 1 figures carry withdrawn claims, unmarked; a README says ~25 s for tests that take ~45, another 330 tests.

7. **No record of the outcome**
No upload receipt, selection or result; Stage 2 commits are dated 9 to 12 October, after support ended on 8 October.

8. **History details**
A personal email on every commit, a local file path and a leftover Word file in history; tests rewrite a file in the repo.

Found in this review. FINRA = Financial Industry Regulatory Authority; BrokerCheck is its public lookup of brokers and advisers. The repo is private, so none of this is public, but the Stage 2 code archive built for the organizers includes the BrokerCheck records and a library staff name.

3. What to Fix Next

1. **Remove the BrokerCheck data**
   Delete it, remove it from history and rebuild the Stage 2 archive; it served only a dropped plan.

2. **Remove the people profiles**
   Keep my own rule from Stage 1: name no individuals.

3. **Link, do not copy, the organizer's files**
   Replace the PDFs and the email with links and a one-line summary.

4. **Score the seal in the repo**
   Commit IBKR's 8-K exhibit of 15 October and a scoring script; record that D1 missed.

5. **Record the next seal outside git**
   Send or post the digests before the outcome is published, so the date is not mine alone.

6. **Source or drop the panel claims**
   Commit the code that extracted the monthly panel and sensitivity rows, or retire the claims.

7. **Mark what was withdrawn**
   Rename or remove the two Stage 1 figures; update the READMEs and status table.

8. **Record the outcome; clean history**
   Add receipts and the result, use GitHub's no-reply email, and run tests on a copy.

Roughly in order of risk: other people's data first, then the seal's credibility, then upkeep.

4. What It Shows, and What It Does Not

It Shows

Forecasts held to point-in-time data, with a leakage check that caught a model reading future data.

Negative results reported, and five claims withdrawn in writing.

A forecast sealed before its outcome; scored now, three of four families held it.

A tested codebase: 367 tests, and every number in the claims ledger re-derives.

It Does Not Show

That the wedge-based models beat a plain regression.

That the screen finds known rules in other filings: 0 of the 3 labeled later.

That the model transfers to a second company.

A competition result, or a seal anyone else can date.

Five weeks of work, with commits on seven days, as sole author and AI-assisted, that graded one company's own rule and logged its own misses. Next: score the seal in the repo, remove other people's data, and build the language-model extractor Stage 1 promised.