The Yield Curve Arbitrage Strategy
in Fixed Income Markets

Group A
Ryan Hou | Nigel Li | Jesse Price
December 9, 2025

I. Introduction
   A. Idea

Alpha comes from taking DV01-neutral Treasury butterflies with temporary richness/cheapness in the "belly", hedged with the "wings", often around macro shocks or demand/supply imbalances, and then the position capitalises on a roll-down while the curvature pricing mean-reverts.

could be a little clearer, but not bad.

Although Yield Curve-Arb was implemented years prior, Duarte, Longstaff & Yu (2007) was a seminal paper that shed lights on the alpha, where the trade aims to find "rich/cheap points", and long-short positions to hedge out the risk, wherein members of our group were inspired by this paper after taking the Fixed Income Portfolio Management Course in the Fall.

hm, work for this course was not supposed to overlap with any other course. this was laid out in the presentation area survey, but perhaps i did not make it clear enough that this extended to the implementation, so i will not take points for it.

   B. Highlights

      1. Strategy Overview

Economic Intuition

All yield curves, but notably, the US Treasury Yield Curve, exhibits persistent structures in level, slope, and curvature, but the relative pricing of the belly versus the wings is frequently distorted by macro news, auction supply, hedging flows, and risk-off/risk-on positioning. These distortions manifest as either "rich" or "cheap" bellies relative to the other tenors, and it tends to mean-revert as positioning normalises as the market re-anchors to fundamentals.
Once, level risk is neutralised by taking long-short positions, the return is expected to come from two sources:

1) Convergence of the belly relative to the wings mispricings, where the belly re-prices towards the wings, as temporary richness or cheapness fades

2) Carry and roll-down (CRD), where over a short holding horizon, each leg "ages" along the curve; when the belly's expected carry and roll-down dominates the DV01-weighted wings (net of costs), the fly has positive expected return even if the overall curve level does not move.

i dont remember that we discussed this; this statement depends on the direction of the fly and hence is not correct as stated

However, the fundamental premise is that the curve shape is reasonably stable over the short holding window.

Signal Generation

For each potential candidate combination (e.g., 2s5s10s, 5s10s30s), systematic entry and exit signals are computed for each country, by estimating the potential carry-and-roll-down CRD based off the spot yield curve, and then standardising the signal using a rolling z-score, such that the "rich/cheap" curvature is measured relative to recent history, in this case, 128 days, rather than a fixed-term long-run normalisation (which was an issue earlier). Trades are then entered when the absolute z-score exceeds a fixed entry band and exited when it mean-reverts inside a tighter exit threshold, with additional government-vs-IRS confirmation filter wherever the swap data is present. Positions are structured to be DV01-neutral, such that there is no level risk, extrapolated further in PC. As entry and exit is primarily predicated on Z-Score signals, it theoretically should be implemented systematically, although it is hybrid in the manner that it will need continuous refinement.

interesting. sounds like the hedge " package" would be defined first then

^^ spec and highlights not to be changed during research; fixes can only be done or explained in refinement

^^ and what else. 2 degrees freedom

Portfolio Construction

Portfolio Construction is risk-controlled in such a way that any time, four concurrent butterflies can be implemented, and implemented in such a manner that at any time, no position is more than 25% of the portfolio, so there is some diversity in exposure. A butterfly position is sized with a shorter tenor, and a longer tenor relative to the "belly", such that is DV01 = 0, and therefore "level hedged". As such, it is "equal weighted", in the sense that the highest-conviction opportunities are selected based off the best |Z| scores (provided it fulfils the entry conditions), and then the unused capacity is allocated to a cash proxy, namely SOFR, with implementation expressed through Sovereign US Treasury Futures, and other liquid sovereign futures where available.[1]

ok

ok then we need to measure sharpe not IR

2. Performance Estimate

Period	Strategy	Ann. return (%)	Ann. vol (%)	Sharpe	Max DD (% cap)	Calmar	Sortino
IS	Baseline (US only, fixed z)	5.919	8.239	0.718	-20.569	0.288	0.659
IS	R1 (roll z + MA)	5.600	10.309	0.543	-30.308	0.185	0.650
IS	R2 (multi-cty + futures + costs)	8.515	6.942	1.227	-8.147	1.045	1.939
IS	R3 (R2 + IRS filter)	7.657	6.632	1.155	-8.498	0.901	1.823
OOS	**Final (R3 / E3)**	**3.309**	**4.931**	**0.671**	**-4.859**	**0.681**	**0.971**

*See more detailed statistics in the Appendix, and further breakdown of results below

i dont see the actual estimate, only the results. 17.8

II. Specification

A. Analysis

1. Qualitative

As a recap, the yield curve "butterfly" (fly) is a three-tenor (can be many more in reality), rel-value structure intended to isolate the curvature by neutralizing first-order exposure, in other words yield shifts. We intend to evaluate the Economic Intuition Hypothesis, as detailed later.

[1] For instance, Australian 30y do not have futures and Italian Rates do not have readily liquid futures.

Positions are structured in a way that DV01-neutral weighing ensures that the fly's mark-to-market changes are driven by belly-vs-wings movements rather than the overall level of rates, otherwise it would be more of a fundamental strategy. This decomposition is economically natural because the yield curve's co-movement is well-approximated by a small number of factors (level/slope/curvature), with classic principal-component evidence showing that most variation is captured by a few components, as found by Litterman & Scheinkman (1991).

This term structure marks flies a natural venue for curve arbitrage, wherein temporary dislocations arising from flows related to Treasury auctions, refinancing activity (rolling the treasuries etc.), and asset-liability management (ALM) rebalancing can drive convergence opportunities as markets re-attach itself to fundamentals. For instance, scheduled US T auctions are documented to create temporary price pressures, and subsequent reversals closer to the auction windows, as it is related to potential dealer's limited risk-taking capacities, where Lou, Yan & Zhang (2013) have estimated that the implied issuance-cost magnitude can range from the order of single-digit to high-teen basis points in this context.

ok

Two implementation considerations motivate the design choices. Firstly, as interest-rate regimes and curve-shape behaviour varies over time, rolling standardisation (i.e. Z-scores), makes the rich/cheapness assessment adaptive to regime changes rather than pinned to a long-term normalization that becomes stale, as shown with preliminary implementations. Secondly, government curves reflect sovereign-specific technicalities like repo/specialness, or even supply, while swap curves add dimensionality and adding a layer of security with the IRS as a confirmation filter, serving as a theoretical cross-instrument check. We seek to evaluate whether the following economic hypothesis is the backbone of the Economic Intuition, which is in turn the theoretical bedrock for this project. More in Appendix 1. Overall, these hypotheses motivate a systematic strategy that identifies curvature dislocations via rolling Z-scores, and trades are manifested as DV01-neutral flies. Adding in the optionality of cross-instrument confirmation for the signal (covered in greater detail in signal generation), can enhance control implementation. Evidence that predictable supply and demand events can systematically distort yields, further support the economic intuition of why curvature can temporarily deviate from "market normals", and then reverse.

again, changes do not belong in spec.

2. Quantitative

We intend to build look at the associated quant statistics of:

Statistic	Definition / interpretation (why)
Annualised return, volatility, Sharpe	Excess return divided by the risk which is useful for measuring performance of strategy.
Maximum drawdown (Max DD)	The largest drop from a given high to a low. Measures the worst possible performance of a given strategy.

Calmar ratio	Calmar ratio is similar to sharpe and sortino ratio in that it measures return versus risk. This time the risk is the maximum drawdown. Another way of interpreting performance of a strategy.
Correlation vs major asset classes	Asset proxies like SPY for US equities or HYBL for HY, will be correlated with the trade, to help inform potential allocators.
Hit rate & average win/loss	Ratio detailing how many working trades divided by not working trades. Helps determine how good a strategy is.
P&L attribution (carry/roll vs shape)	A plan which involves dividing P&L into different categories allowing us to understand where the P&L actually came from.
Value at Risk (VaR)	Measures which deal with the evaluated risk of a strategy. VaR tells you how much you could at most lose with a certain amount of certainty.
Expected shortfall (ES)	Expected Shortfall deals with the average amount you would lose if you were to enter that VaR range

*There are more Statistics below computed as shown, but this table was minimised for brevity

good.
more? great

B. Data

1. Universe

The core universe lies in the G7 sovereign government curves where i) reliable daily curve data exists, curve shape (curvature) can meaningfully move through alternations in flows and other macro factors, and ii) actually implementable via spot bonds (which we extrapolate as STRIPS) or liquid futures. Note the latter was only available for US, Germany and the UK.

Countries (Sovereign Curves): United States, Germany, UK, Italy, Japan, Australia, Canada
Core Fly tenors: 2s5s10s and 5s10s30s (where available), i.e., focusing on the 2–30y sector

good

Instruments used to implement:

- Curve tenors used to form flies - from 2 - 30 years, but usually 2s5s10s
- Instruments used to reflect the strategy:
  - Spot Bonds implementation is based on cash government bonds at the fly tenors (e.g., 2y/5y/10y and 5y/10y/30y)
  - Futures sleeve (deep/liquid where available):
    - US: TU (2y), FV (5y), TY (10y), US (30y)
    - Germany: Schatz (2y), Bobl (5y), Bund (10y), Buxl (30y)
    - UK: short/medium/long Gilt futures mapped to 2/5/10 to 30y equivalents
  - Cash proxy for unused risk budget: SOFR (risk-free cash benchmark).
  - Extra: Asset proxies for commodities, HYFI, IG FI, US Equities via YFinance

The universe was chosen, because while such a strategy can technically be implemented for any rate or credit universes, these markets are (a) structurally “fly-friendly” (curvature is a persistent curve feature), (b) deep enough for systematic RV implementation (especially US/DE/UK via futures), and then also c) diversified enough across independent rate regimes for enough opportunities.

ok

2. Data Sets

To run signals, size the DV01-neutral flies, execute and mark theoretical PnL, the data required:

1) Sovereign Curve Data
- Par yields across the curve, with the fully fitted curve for roll/carry calculations across the universe sovereigns

2) IRS Curve Data (daily, by tenor)
- Used for dual-curve confirmation (trade only when government-curve and swap-curve signals agree in sign). In a practical note, IRS history can be shorter (in our pipeline, reliably available from 2009 onward on Bloomberg).

3) Futures Data (daily)
- Settlement prices, plus contract metadata needed for risk mapping (DV01 notionals, tick value, etc.). The latter is generally consistent, and is used to express the fly trades for 3 main countries

4) Funding/Cash Benchmark
- SOFR time series to represent the return on unused capital parked in cash.

5) FX Series for spots
- As we are aggregating multi-country positions into USD as the single base currency PnL, daily DX is needed to convert returns (we assume is already FX hedged - which is represented in the transaction costs later for round-trip)

ok

3. Data Sources

Primary Sources (all strategy inputs and implementation) - from Bloomberg Terminal
- Sovereign yield curves (daily curve snapshots / fitted curves)
- IRS term structures by currency/tenor
- Futures prices (daily settlement), rolls, and identifiers
- FX series (to convert for PnL)
- Contract reference data where needed

ok

Sanity Checks:
- FRED US Treasury constant-maturity yields (used to verify Bloomberg US curve levels in prior weeks): 2y/5y/10y/30y series are available as DGS2/DGS5/DGS10/DGS30.
- CME (US Treasury futures), Eurex (German bond futures), ICE Futures Europe (UK gilt futures) referenced for product/venue context.

sanity checks, great! give some +

Data Costs:

- Bloomberg - Institutionally licensed with costs borne by institution ($20k + pa)
- FRED/ NY Fed / Exchange reference pages - mostly free; CME data needs subscription, generally Category A1 Enterprise at $3,521/month

ok, this is meant as your cost and is intended for supplemental data, but gotcha

1. Date Range

Jan 2004-Nov 2025 data was used to capture multiple rate regimes like pre-GFC, GFC, QE, rate hikes, COVID shock, and tightening, to ensure that there are enough observations for fly-style mean-reversion strategies. Training data is not needed, as no models are trained - realistically, if one were to maximise backtest results, grid search for the parameters (z-scores, etc), would be done, and then have regime-blocked cross-validation.

- In-sample (**calibration / development from 2004-2020)**: Used to develop and tune the framework (signal construction choices like smoothing, rolling z-score window, and entry/exit thresholds; and portfolio rules like DV01 budgets and max-fly cap). This long window is intentional because curvature behaviour and CRD opportunities are regime-dependent, so we want calibration to see many environments. Realistically there should have been some parameter search, but not to the extent of grid search or Bayesian Optimisation, as that will literally be data-mining.[2]
- Out-of-sample (**forward validation - 2020-2025**): Held out for forward testing of robustness with frozen parameters. *Note that in the previous implementation (Week 11) - it was implemented until 2024, with 1 year as holdout. That has now been rectified.*

good

C. Strategy

1. Signal Generation

On a daily basis, a systematic curvature signal is computed for each country, for each potential combination fly with the tenors of 2s-5s-10s-30s. The pipeline converts observed yield curves into a CRD-based dislocation ,or "Z-Score", which directly informs entries and exits.

Step 0: For relevant tenors, (L, B, R) weights are chosen w_L, w_R such that the fly is approximately DV01-neutral:

$$DV01_{B,t} - w_L DV01_{L,t} - w_R DV01_{R,t} = 0, \qquad w_L + w_R = 1$$

ok cash neutral. should have mentioend earlier

Step 1: Let $CRD_{i,t}$ be the estimated carry and rolldown for individual legs of the trade for L, B, R, over a short horizon, of 1-3 months, computed from the spot curve under the "unchanged curve" assumption. Then, we define the fly-level CRD as a weighted combination such that:

[2] The dual-curve (Govt + IRS) confirmation is only applied for the subperiod where IRS curve history is available/clean in Bloomberg; for earlier years in the in-sample, the strategy uses government curve signals only (so the 2004–2022 IS is still valid, but the "B2 refinement" is partially active within it).

$$CRD_{i,t}^{fly} = CRD_{B,t} - w_L CRD_{L,t} - w_R CRD_{R,t}$$

And then the expected "edge" is the expected carry or roll-down, ex the DV01 weighed wings (which removes the DV01 first-factor PC risks).

ok. i dont feel like 1-3 months captures enough. i dont remember this discussion in class, i would have pushed on that. guess we had many other things to discuss.

Step 2: Smooth the signal (reduce noises in the data): $CRD_t = \frac{1}{5}\sum_{k=0}^{4} CRD_{t-k}$

Step 3: A rolling mean and standard deviation are computed over the last L = 126 trading days such that we have:

$$\mu_t = avg(CRD_{(t-L:t-1)}), \quad \sigma_t = \mathrm{std}(CRD_{(t-L:t-1)})$$

$$z_t = \frac{CRD - \mu_t}{\sigma_t}$$

Step 4: Initial implementations used fixed holding periods, but now has been replaced with explicit mean-reverting rules. We used a band-trading design for mean reversion, which mimics traditional extreme-then-revert rules in stat arb and the mean-reversion timing literature, with most stating that the benchmark for "large deviations" is roughly 2 standard deviations, but as we are dealing with far more liquid assets, a lower entry is justifiable such that:

changes in refinement, not spec

- Enter a position when $|z_t^{sovereign,c}| \geq z_{entry}$
- Exit a position when $|z_t^{sovereign,c}| \leq z_{exit}$

The baselines are set as[3]: < you are saying these papers suggested these non-round entries and exits? even if its in a paper, i need justification. those are also different asset classes and strategies.

$$z_{entry} = 1.6, \; z_{exit} = 0.2$$

Directionally:

- If z_t is sufficiently positive → take the "positive-signal" fly direction
- If z_t is sufficient negative → take the opposite direction

Sizing and ranking of these individual trades will be established more in PC.

Step 5: When IRS tenor data is available, we compute the same signal on the swap curve, to obtain as z_t^{irs}. Then, the confirmation rule is as follows where entries requires directional agreement:

$$sign(z_t^{sovereign,c}) = sign(z_t^{irs,\,c})$$

[3] Gatev, E., Goetzmann, W. N., & Rouwenhorst, K. G. (2006). *Pairs trading: Performance of a relative-value arbitrage rule.* **Review of Financial Studies, 19**(3), 797–827; Avellaneda, M., & Lee, J.-H. (2010). *Statistical arbitrage in the U.S. equities market.* **Quantitative Finance, 10**(7), 761–782; and Leung, T., & Li, X. (2015). *Optimal mean reversion trading with transaction costs and stop-loss exit* (arXiv:1411.5062). arXiv. (Refer to Github for the full list of sources considered to justify these numbers, but in this paper, they mention that opening when the spread is about 2stdev, and close when it is close to the mean)

Then, the exit signal is predicated on either signal mean-reverting at the exit band. The idea of this is that it is designed to reduce Type I errors (false positives) driven by idiosyncratic movements.

2. Portfolio Construction

Trading Frequency

The trading frequency is predicated on a signal evaluation on a daily basis, and the trading is event-driven such that trades are only entered when z-score rules trigger an entry or exit, otherwise there is just hold. At any time, when there are 4 open butterfly positions, the signal while still running, will not trigger and recalibrations. Only when there are spare capacities, will the highest signal that is not currently trading be traded. Resizing is on a needs basis, to keep each fly DV01-neutral as durations drift.

ok

Sizing

The positions are sized using a simple equal-weighting to diversify the risk budget, and the idea is that each fly will therefore carry similar interest-rate risk.

- Max concurrent trades will stay at 4 at any time, with no more than 25% allocated to each position.
- Selection will be predicated on eligible candidates by absolute Z-Score and the top 4 will be taken.
- Risk Budget per fly - The overall
- Scaling for DV01-neutrality, where for each selected fly, the trade is scaled so that the wing legs counterbalance the belly's level risk.
- Idle Capacity: If fewer than 4 flies or trades reach the 1.6 entry threshold, the remaining capacity is parked into a cash proxy, rather than force weaker trades that may not even cover transaction costs.

sounds like some potentially large unforced turnovers. dont love this methodology but ok

Hedging

Then, the exit signal is predicated on either signal mean-reverting at the exit band. The idea of this is that it is designed to reduce Type I errors (false positives) driven by idiosyncratic movements. By design, each position is intended to be DV01-neutral, such that the following is fulfilled (mentioned prior):

$$DV01_{B,t} - w_L DV01_{L,t} - w_R DV01_{R,t} = 0, \qquad w_L + w_R = 1$$

As a result, upon entry, exposure to level risk is approximately 0, and practically, the belly is the main leg, and the two wings are short (or long) in fixed priorities thus the DV01 risk is mitigated (like hedging delta for equities). There will **NOT** be a daily DV01 re-hedging, as the transaction costs will be high enough to basically wipe out all gains.

ok. that high?? the hedging should be small

Money Management

Then, the exit signal is predicated on signal mean-reverting at the exit band. The integration of the IRS curve idea into entry/exit conditions of this is that it is designed to reduce Type I errors (false positives) driven by idiosyncratic movements. A stop-loss was not implemented, and the “measures” taken for risk management is predicated on controlling downside with DV01 budgeting, a four-fly capacity and the signal-based entry/exits, as the idea in that mean-everting strategies can force liquidations near peak divergences and increase costs in the long-term.[4] Empirical evidence derived from literature reviews suggests that stop-losses in the context of mean-reversion strategies achieve the opposite effect, that is, it liquidates a position, that under the Economic Intuition, should mean-revert - therefore this directly contracts the main hypothesis of the trade/study that PnL comes from the convergence of the positions.[5] The main portfolio constraint in this case is that we cannot hold more than 4 flies simultaneously, such that it does limit concentration and control aggregate turnover. Once again, the exit remains signal based, even if the “stop-loss” has not been hit.

ok

3. Trade Execution

Signals are formed using information available through date *t* (close). To avoid look-ahead, trades are assumed to be executed on date *t+1* for next available prices (for futures, that is next-day settlement, for curve-based positions, next-day curve implied prices).

ok

Prices (timing vs input Data)

Signals are formed using information available through date *t* (close). To avoid look-ahead, trades are assumed to be executed on date t+1 for next available prices (for futures, that is next-day settlement, for curve-based positions, next-day curve implied prices). Positions are held until an exit event (signal re-enters the exit band) or until replaced by a stronger candidate due to the portfolio fly-cap rule. (25% max exposure to one wing)

good

Instruments:

We implement the strategy with two main options:

1) Curve

The US fly legs are implemented via the ZCB components of Treasury cash flows, or the US Treasury Separate Trading of Registered Interest and Principal Securities (i.e. ZCB legs correspond directly to STRIPS). For non-US sovereigns, DE/UK/IT/JP/AU/CA, where a STRIPS-like dataset is not available point-by-point, we use curve-implied zero-coupon prices as a consistent mark-to-market proxy for the underlying “strip” cash flows. This is standard term-structure work because coupon bonds are priced as discounted cash flows using discount factors, and coupon-bond exposures can be decomposed into portfolios of zeros, as introduced also in MATH5010. We assume execution at t+1 using next-day curve-implied prices at the required tenors. The idea is that each day’s fitted curve implies a set of prices, and coupon-bond exposures can be represented as a portfolio of bonds, based on no-arb[6]. We marked the Daily Pnl as: ok

[4] **Kaminski, K. M., & Lo, A. W. (2014).** *When do stop-loss rules stop losses?* **Journal of Financial Markets, 18**, 234–254. https://doi.org/10.1016/j.finmar.2013.07.001
(Shows stop-loss value depends on the return-generating process; under non-momentum settings a stop-loss can reduce expected return, supporting “no mechanical stop-loss” for mean-reversion.)

[5] **Leung, T., & Li, X. (2014).** *Optimal mean reversion trading with transaction costs and stop-loss exit* (arXiv:1411.5062). arXiv.
(Models the spread as an OU process and shows that adding a stop-loss constraint changes the optimal entry/exit regions, so stop-loss design is non-trivial and should be model-calibrated, not bolted on. The idea is that traditional X% drawdown → and therefore exits, is incompatible for such a strategy)

[6] Bank for International Settlements. (2005). Zero-coupon yield curves: Technical documentation. BIS Papers. (Discussion of discount factors / curve construction and bond pricing via the term structure.)

Daily PnL ≈ (belly weight × ΔP_belly) − (left-wing weight × ΔP_left) − (right-wing weight × ΔP_right)

2) Futures execution

To express the same curvature trades with deep liquidities, flies are implemented using standard government bond futures mapped to the nearest tenors, with the following:

- United States: TU (2y), FV (5y), TY (10y), US (30y)
- Germany: Schatz (2y), Bobl (5y), Bund (10y), Buxl (30y)
- United Kingdom**:** Short / Medium / Long Gilt futures (mapped to approx. 2/5/10 to 30y equivalents)

Note that other futures like Aussie futures are not included as it literally lacks 30Y futs, and Italian rate futures are not as liquid. For the futures, each tenor is mapped directly to the closest contract and sized, once again to maintain DV01-neutrality. This is documented later in implementations and refinements further. As for Trade/M2M price, we assume execution at t+1 using the next-day settlement price of the mapped contract (TU/FV/TY/US; Schatz/Bobl/Bund/Buxl; Short/Medium/Long Gilt). We marked the daily Pnl, computed from changes in settlement price and mark to market

futures get much more complaex bc of the ctd etc

Daily PnL = (# contracts) × (contract multiplier) × (Settle[t] − Settle[t−1])

As for roll handling, the backtest uses the same contiguous generic futures convention as the input data. Rolls are therefore embedded in the px series, and roll effects show up naturally in the PnL

Instrument Venues:

well you have to make sure you are not holding any futures across a roll date. i dont see you adjusting for that here

Actual trading venues:

- U.S. Treasury futures (TU/FV/TY/US): CME
- German government bond futures (Schatz/Bobl/Bund/Buxl): Eurex
- UK Gilt futures: ICE Futures Europe
- Cash/STRIPS legs are treated as OTC cash government markets, while our curve-based legs are marked via the fitted curve.

Transaction Cost models:

ok

We initially attempted to attempt a nonlinear cost model of the form "fixed & Linear in traded risk + quadratic market-impact term" as per Almgren (2000), but the quadratic-cost approach went awry, as DV01 changes jump discretely, and small unit mismatches caused the quadratic term to dominate and caused extremely unrealistic and unpredictable cost spikes. In practice, this model was extremely numerically unstable in the backtest pipeline. See Appendix for more details on the initial plan.

Resultantly, adoption of a simpler country-level round-trip bps model that is stable and transparent, and is more meaningful for comparison. The following country-level assumptions, show US 0.12 bps; DE 0.12 bps; UK 0.18 bps; JP 0.08 bps; AU 0.35 bps; CA 0.25 bps; IT 0.50 bps (round-trip per fly). These are applied uniformly across fly types and across both curve and futures sleeves, unless otherwise stated in the Appendix.

JP is cheaper than US? i guess so for best, but it is also much thinner. so here depends on your notional assumptiosn.

youve missed a couple important details. you havent clarified whether you are quoting in price or yield. price is traditional and more stable for this purpose; however based on your #s i believe you are quoting yield (else way to aggressive, esp for RT.

2nd, you havent clarified whether its two sided for the fly. assuming its in yield, whether you are one sided or two sided will change whether you are aggressive or conservative here. i wont be able to solidly judge later results, and can only take a little for the unclarity here.

good SUPP in the appendix

A simpler, more robust country-level round-trip cost assumption in bps per fly round trip by country was calibrated to expected bid-ask and spillage levels, and applied to each entry and exit. In this case, we let:

slippage?

$$Cost = (round - trip\ bps\ for\ the\ country) * |DV01\ traded|$$

SUM

$$wherein\ this\ case\ bps\ is\ interpreted\ as\ the\ yield - bps\ of\ slippage\ paid\ per\ unit\ of\ DV01$$

ok so it is yield

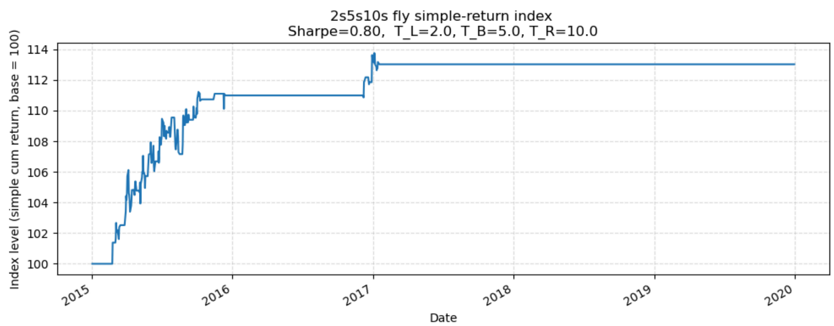
Do note that EoD Signal, next-day execution at settlement (futures) or net-day curve mark, with no intraday timing issues, and orders are assumed to be small enough to be filled at/near settlement proxy.

III. Implementation

19.3 not perfect but gave some back for a lot of good extras here

A. Results

1. PnL Graph



Our first backtest implements a single-country, single-structure rates relative-value strategy on the U.S. 2s5s10s. Basically, this only implements Steps 1, 2, 4 of the signal in the trading implementation. This backtest was deeply problematic, as it used a 10-year look-back rolling average for standardization. This is naturally problematic, as the trade literally did not execute from 2017 onwards, as the signal was flatlined, and not triggering. The fly construction does appear to be able to neutralised levered long-duration exposure, but the combination of conservative filters, and the thresholding led to long idle periods. Further refinements (and the above finalised methodology + signal generation), has contributed to better entry/exit design. Please note that this backtest was from 2015 onwards as we used **10 years of MA.** Further details in the appendix.

yeah 10 years is definitely too much.

nothing on performance?

also looks a bit too good in 2015. hard for me to be sure whether that is bc your trx cost was too aggressive or not

2. Summary Statistics

Ann. Return	Ann. Vol	Sharpe	Max DD (% TN)	Avg Hold (days)	Total Trx Cost (% TNl)
2.63%	3.29%	0.80	2.72%	8.33	0.00045%

this looks like IR not sharpe. not right with your cash management rules.

3. Differences From Expectation

~~wait, youre missing everything from the Spec, Quant Analysis. think i gave you guys extra pts for that~~

Relative to our proposal, the realised baseline results are directionally consistent, but operationally disappointing. It did attain low vol and reasonable Sharpe, but the strategy spent long stretches being inactive due to the stale standardization. Compared with RV literature, our outcome is consistent with the idea that RV is regime-dependent, where the post-2017 period showed entry signal being unreachable. This guides and motivates later refinements, as detailed below.

also missing stats analysis

ok found them in appendix, but still missing corr and attribution and not well documented here.

ok

slight penalty for OOS problem.

18.2

B. Difficulties

1. Sparse trading and long flat periods

The initial implementation with the first-pass entry/exit logic based on quantile thresholds and regime filtering, along with special events blocking **filtered out most days.** On the other hand, there was no MA, and a fixed average for the last 10 years. This means, it is not MA to the recent regime, and instead is a long-term average MA. Therefore it does not execute, so the backtest spent long stretches flat. That mechanically depresses calendar-day hit rate and makes the PnL look like it "stops," even if the conditional performance while invested is reasonable. ***The workaround implemented (as detailed above in Signal construction) involved entry/exit bands like 1.6/2.0 and a 126-day MA, so it trades comparably across regimes rather than being predicated on a static historical distribution.***

ok

2. Limited diversification (single country)

A single country, but the main execution (we iterated across tenors) is predicated on the US 2s5s10s fly is extremely regime-dependent, and lack of mean-reversion, means that either A0 the trade goes dormant, or B) There are concentrated drawdowns. It was mainly A. ***The workaround was to expand the universe for multi-country cures and multiples flies, and also expand to futures as another sleeve with a PC rule of max 4 concurrent flies, with unused risk in SOFR. This resolves the one-trade world problem and reduces reliance on literally one country's regime.***

thats not a difficulty, thats your own design

3. False positives from sovereign-specific distortions

In the baseline scenario, entries were triggered solely from the government CRD Z-score, and as such, sovereign curves can for reasons move for reasons that are not true "curvature mispricing", for example, auction supply or settlement technicals etc. and can generate *false positives*, and indeed, where the fly looks cheap on the curve, but does on mean-revert on the expected horizon, producing avoidable drawdowns and churn once the position is entered in. ***The workaround for this was the introduction of the IRS tenors in parallel with the government-curve signal, and if the two are in disagreement, the entry is skipped. Operationally, this will improve selectivity, at the cost of lower frequencies, but entry issue should be resolved with the first "workaround".***

again, more of a difficulty for your strategy rather than yourself. but ok

IV. Refinements

A. Implemented

1. Catalog

Following the baseline results, and guided by lecturer feedback, we selected three refinements that directly addressed the main issues observed in the strategy:

1) Rolling standardisation + smoothing of the CRD signal (fixes the "stale mean" problem)

The long,fixed historical normalisation was replaced with a rolling 126 trading day Z-score, and applied a 5-day MA to the raw signal. As mentioned above, the stale signal prevented trade, as the baseline mean was dominated by older rate regimes, so the signal rarely ever reached the signal. Refer to Appendix 5 for more information. It was implemented with the expected roll-down, and smoothed with 5-day MA and then, standardised it for entry/exit signal (refer to Signal Generation Step 2-4). Hard entry/exit rules, with enter at 1.6 (direction given by the sign),

yes, this is a necessary fix

and exit, with a mean-reversion to 0.2 For dual curve confirmation, refer to above. *Reason, why this is bundled together is because this completely revamped the stale signal that persisted. This was initially presented as two refinements.*

yeah i thought confirmation was separate. this is not perfect but if your idea is to simplify the paper, its perhaps understandable

2) Portfolio construction & Universal Expansion

universe

Expanded with single US fly, to multi-country, multi-fly implementation with futures for liquid countries. Implemented a capacity constraint of max 4 concurrent flies, chosen by largest absolute Z-scores, and sized such that no more than 25% exposure to each individual position of portfolio, and unused risk is parked in SOFR. this was already discussed in spec. so confusing.

The nature of expanding the universe means that transaction costs cannot remain at a low rate, as was the case with the US as the sole universe, as it has the most liquidity. We had accounted for different transaction costs. Refer to the appendix for more information, and above for how this was computed. *This will build on the above R1*, as the signal is a major refinement.

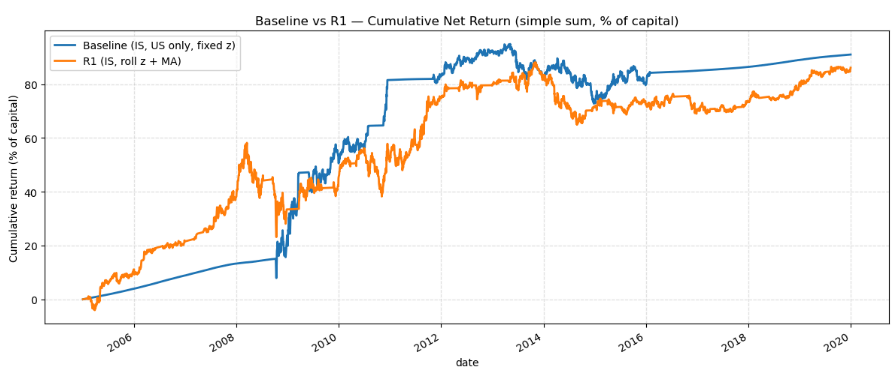
ok

3) Accounting for the IRS curve on top of the refinements

Intended to reduce false positives as mentioned prior. Sourced from BBG. Initially looked at OIS too.

i thought this was in 1? confusing

2. Results
    a. Rolling standardisation + smoothing of the CRD signal

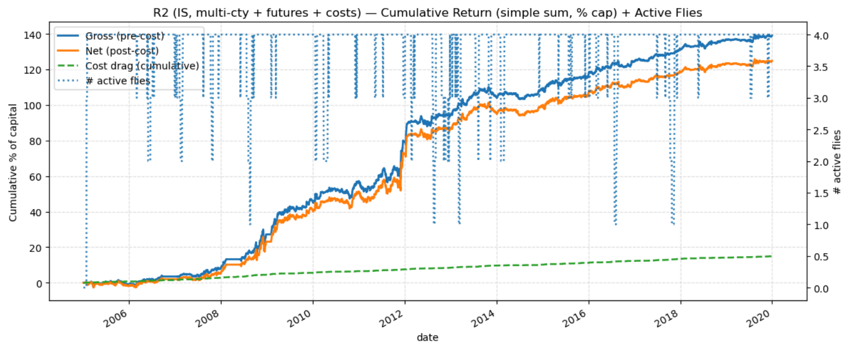


R1 eliminates the stale mean problem and increases trades (refer to Appendix for full results), but the higher turnover also increase realised drawdowns, and actually reduces risk-adjusted performance (note that here for "baseline" - when there are no positions, we are in SOFR, as that makes more senses for apples v apples comp). Noting the Sharpe is actually lowered, and in the PnL plot, it is more active, but less efficient. **Note the summary stat table for refinements are in the appendix.** ^^^ aha.

ok

needs stats and analysis here

b. Portfolio construction & Universal Expansion (including prior fix of signal)

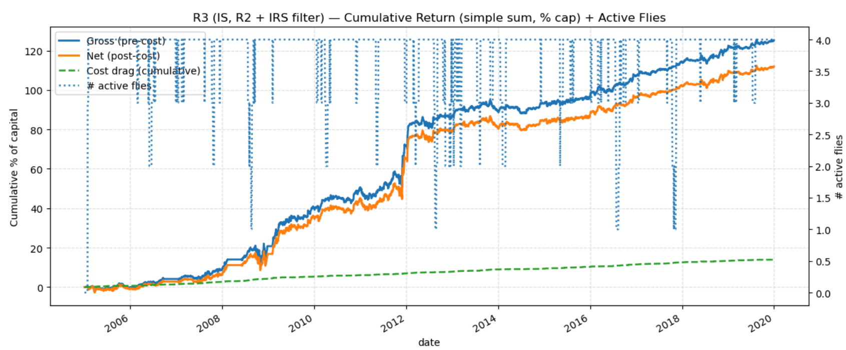


this looks a bit too good to be true. most likely low trx costs in other countries, or assumptions on liquidity/slippage

R2 integration of a single US fly to multiple sovereign opportunities universe, with futures for more liquid markets and 25% PC limit, with ranked signals, it is the first refinement that makes the strategy actually look reasonable, with materially improved results, with reasonable annual returns, good Sharpe, and extremely low drawdowns. The PnL path is far smoother, and consistently upwards, despite some meaningful transaction costs - although the transaction model might be more conservative that reality. This edge implies it is large enough to survive realistic executions. Refer to Appendix for comparison of results. **Note the summary stat table for refinements are in the appendix.**

i also have my previous concerns about your future handling

c. IRS Curve integration to reduce false positives



combinations here, without universe expansion?

R3 integration of an IRS curve confirmation filter, slightly reduces performance,and does have a similar max DD, while cutting out some churn, with 217 vs 234 round trips, and lowered total aggregate costs. In the plot it tracks R2 closely, but with marginally slower compounding, as it relies on higher-conviction signals.

ok

B. Proposals

1. Vol-Targeted DV01 sizing (to ensure risk parity across the positions)

The intention of this is to add another layer of risk control that scales each positon's DV01 notional down when recently fly PnL vol is slightly high, and scale up during calmer regimes. The same entry/exit logic will apply. The motivation of this is that the current drawdowns are largely regime-linked via rapid curve repricings etc. and vol targeting is a standard manner to stabilise realised risks without changing the underlying alpha source.[7] The generally hypothesis of this is similar average returns over the long-term, lower drawdowns, and better Sharpe/Sortino ratios. This was not implemented, as it requires a trade-relevant vol estimate. It is difficult when positions are sparse, and in this case the Pnl is what we can call "lumpy". Time was also an issue.

ok

2. Model-based entry/exit/stop-loss bands with integrated transaction costs

Instead of fixed z-score entries and exits, an estimated mean-reversion model (OU-style) for the fly signal, in fact, OU-Bridges have been suggested by Dr Graeme Baker from the Columbia Statistics faculty[8]. The motivation of this is that they are most cost-aware,that explicitly accounts for time to revert. It also removes the false positives and churn when the dislocation is technical or persistent. The expectation of this is again, fewer but higher-quality trades, reduced turnover, and a cleaner net PnL profile.

ok

18.1

V. Conclusion

A. Final Selection

R3 was selected as it has the best theoretical underpinning with expanded universe, IRS confirmation and fixed signal generation, while accounting for realistic implementability where liquidity exists with cross-instrument agreements.

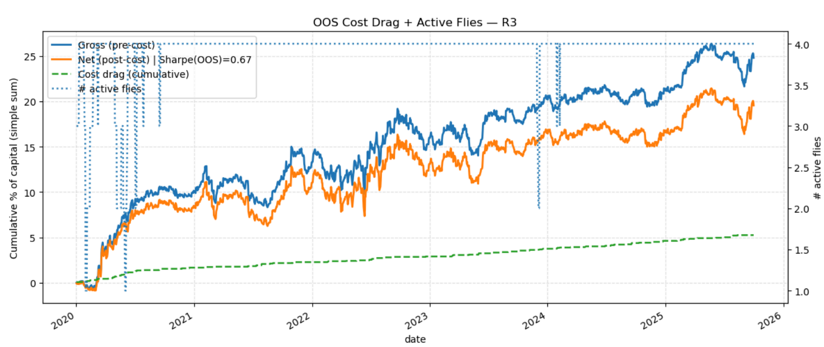
ok

B. Out of Sample Test(s)

1. PnL Graph

TA's said there was backtesting before selection, which is penalized.

The OOS test shows that the R3 remains profitable with a positive Sharpe and controlled drawdowns with costs evidently reducing performance, but the IRS filter maintains a higher trade hit rate and longer holding periods, which indicates consistencies with slower convergence trades rather than and short-term timings.



all those stats are to be discussed later, not in the graph analysis

2. Summary Statistics

Strategy	Ann. Return (%)	Ann. Vol (%)	Sharpe	Max DD (% Target Notional)	Avg Hold (days)	Total Trx Cost (% Target Notional)
Final: E3 OOS	3.309	4.931	0.671	-4.859	79.239	0.889

sharpe is not amazing but i still find this suspiciously good given everything.

[7] **Moreira, A., & Muir, T. (2016).** *Volatility-managed portfolios* (NBER Working Paper No. 22208). National Bureau of Economic Research. *(Motivates volatility targeting to stabilise realised risk)*

[8] **Liu and Longstaff (2004)** develop a dynamic portfolio framework showing that even in the presence of textbook arbitrage opportunities, optimal strategies may underinvest due to risk aversion and funding constraints, leading to interim losses prior to convergence. *(idea is that entry and exits may not be timed perfectly. OU-bridge is more mathematically rigorous)*

full table -- at least say, if its in appendix

Notably the net annual return drops to 3.309% from 7.657%, and sharpe actually declines, but volatility notably does improve. Max drawdown has improved substantively from 8.5% approx to about 4.9%. The holding period is notably longer at about 79 days. So on a cumulative basis, it does appear that the IRS cross-instrument confirmation does result in a better risk control, but risk-adjusted returns have slightly diminished, and does generalise well. As shown OOS delivers a lower edge.

good theres analysis here this time

C. Additional Considerations

1. Other Investment Concerns

Based on the return, it is primarily predicted off the carry down and roll down capture, and mean-reversion in the curvature. In terms of tax benefits, rates futures may qualify as Section 1256 contracts, with gains/losses typically treated under the 60% long-term and 40% short-term Capital gain split, as per the IRs. As for the spot bonds, especially when trading STRIPs, the practical issue is Original issue discount, or what is called ‘phantom income”, even when no interest is received at the current moment.

In terms of risk, regime risk is a major issue, as curvature can stay rich/cheap for prolonged periods, such as with policy shocks or QE etc, and drawdowns can cluster. On the other hand, residual factor exposure is definitely an issue, as DV01-neutral does not actually guarantee zero exposure to level/slope PCs, under potential curve reshapes and convexity effects. Hedging errors can definitely persist. Liquidity risks, while likely not an issue for the current futures or spot bonds, gradual expansion to other sovereigns, especially on the long-end can be materially less liquid, therefore widening spreads and potential slippages, which is literally a transaction cost. Funding risks is a natural risk of margins, sharp moves can force margin calls at extremely bad times, even if the trade is technically profitable, that backtest was predicated on 100% notional, rather than just putting in the maintenance margin, as there are likely times, where it will require a margin replenishment. In terms of operations, the natural issues of sizing and “fat fingering” persist for every strategy. Also, model risk - the notion of a z-score can overfit and there is definitely risk of performance decay, as shown in the OOS.

the more i look at this, the more i think its just not a YCA strategy. this is just a carry strategy. i guess that in and of itself isnt wrong, but its been rather misleading the whole paper.

2. Business Concerns

From a business concern POV, operationally, the data dependency on Bloomberg on industry standard, and as a fixed cost, if the strategy scales, this is quite minimal. But however, there does need to be daily checks for curve availability, and regular monitoring for potential decay of the signal and further RnD. Perhaps, there also may also be considerations as to diversifications of data vendors, like Wind.

In terms of legal/compliance perspective, exchange-traded futures reduce counterparty risks, in fact CME is basically credit-risk free, but standard controls like approved limits, and documentation will be needed. As for the future of this strategy, as an RV strategy, it can get crowded and the edge can decay in the long-run. Capacity is the best in the futures that we backtested, i.e. US/DE/UK futures, but capacity is substantially weaker in smaller sovereign markets like Italy, where long-end instruments are far less liquid.

In terms of the tooling environment, the data pipeline does work. Marketing-wise, it will be difficult to communicate the relative value of such an active strategy, especially with the astronomic performance of passives in the last decade, where the “two and twenty” for this strategy will likely be a hard sell. It has to be framed in a way where it is immune to regime changes, as shown in the PnL, with relatively stable performance and limited correlations with other asset class proxies.

ok. i dont agree w everything but the point is to put thought into it.

Trading Recommendation

1. Recommendation:

We recommend a measured, phased deployment rather than an immediate full-scale rollout. The strategy is economically well-founded, the signal has been materially improved through rolling standardization, smoothing, and cross-instrument confirmation, and the instruments used are liquid and operationally straightforward. But, the selected final version, although it does showcase reasonable OOS performance, with controlled drawdowns relative to previous versions, the Sharpe is not so high that regime nad implementation risk can be ignored.

ok. it seems, no concerns about the development methodology and any other possible model risks (eg trx cost).

2. Further Research:

Before scaling, additional research should focus on stress-testing performance across historical stress periods (e.g., 2008, 2013, 2020, 2022), validating the joint signals across other sovereigns. It may be worth exploring residual PC1/PC2 exposures and convexity across time, and set hard limits to these exposures. Notably, for position managements, vol-targeting for DV01 sizing, and turnover management and hard “circuit breaker” rules during macro event clusters may be beneficial.

ok

3. Concluding Thoughts:

On an aggregate basis, this is a credible relative value framework that is implementable and diversified across curves, with the refinements showing a demonstrable track record of improved stability and performance. The remaining challenge is not signal intuition, it is regime robustness (although ostensibly fine for events in the past 20 years), which can be tested through monte-carlo simulations potentially; but also, implementation realism in terms of mapping, slippage and further risk controls. R3 should be treated as a “core” or foundation, and iterations on execution and risk overlays has potential for it to be an attractive and alpha-generating strategy.

i feel like its more carry than RV, but i see there can be some component there.

17.5

D. Appendix

90.9 this paper is strong technically, but there were a few research framework points that were not followed and i didnt always agree with all the choices or conclusions. some parts were also imperfectly documented and made the paper hard to follow. still, this score is very high historically. great job!

1. References:

Almgren, Robert, and Neil Chriss. 2000. “Optimal Execution of Portfolio Transactions.” *Journal of Risk* 3 (2): 5–39. https://doi.org/10.21314/JOR.2001.041.

ps i am also trusting there was no work done here that overlapped any assignments/projects from FI.

Avellaneda, Marco, and Jeong-Hyun Lee. 2010. “Statistical Arbitrage in the US Equities Market.” *Quantitative Finance* 10 (7): 761–782. https://doi.org/10.1080/14697680903124632

Bank for International Settlements. 2005. *Zero-Coupon Yield Curves: Technical Documentation*. BIS Papers no. 25. Basel: Bank for International Settlements.

CME Group. 2024. “The Basics of U.S. Treasury Futures.” Accessed December 12, 2025.

Duarte, Jefferson, Francis A. Longstaff, and Fan Yu. 2007. “Risk and Return in Fixed-Income Arbitrage: Nickels in Front of a Steamroller?” *Review of Financial Studies* 20 (3): 769–811. https://doi.org/10.1093/rfs/hhl026.

Eurex. 2025. “Euro-Bund Futures (FGBL).” Accessed December 12, 2025.

Federal Reserve Bank of St. Louis. 2025. “FRED: Market Yield on U.S. Treasury Securities at 2-Year Constant Maturity (DGS2)” (and related DGS5/DGS10/DGS30 series). Accessed December 13, 2025.

Gatev, Evan, William N. Goetzmann, and K. Geert Rouwenhorst. 2006. “Pairs Trading: Performance of a Relative-Value Arbitrage Rule.” *Review of Financial Studies* 19 (3): 797–827. https://doi.org/10.1093/rfs/hhj020.

Intercontinental Exchange. 2025. “Short Gilt Future” and “Long Gilt Future.” Accessed December 13, 2025.

Kaminski, Kathryn M., and Andrew W. Lo. 2014. “When Do Stop-Loss Rules Stop Losses?” *Journal of Financial Markets* 18: 234–254. https://doi.org/10.1016/j.finmar.2013.07.001.

Leung, Tim, and Xin Li. 2015. “Optimal Mean Reversion Trading with Transaction Costs and Stop-Loss Exit.” *International Journal of Theoretical and Applied Finance* 18 (3): 1550020. https://doi.org/10.1142/S021902491550020X.

Leung, Tim, and Xin Li. 2014. “Optimal Mean Reversion Trading with Transaction Costs and Stop-Loss Exit.” *arXiv* (arXiv:1411.5062). Accessed December 14, 2025.

Litterman, Robert, and José Scheinkman. 1991. “Common Factors Affecting Bond Returns.” *Journal of Fixed Income* 1 (1): 54–61. https://doi.org/10.3905/jfi.1991.692347.

Liu, Jun, and Francis A. Longstaff. 2004. “Losing Money on Arbitrage: Optimal Dynamic Portfolio Choice in Markets with Arbitrage Opportunities.” *Review of Financial Studies* 17 (3): 611–641. https://doi.org/10.1093/rfs/hhg029.

Lou, Dong, Hongjun Yan, and Jinfan Zhang. 2013. “Anticipated and Repeated Shocks in Liquid Markets.” *Review of Financial Studies* 26 (8): 1891–1912. https://doi.org/10.1093/rfs/hht034.

Moreira, Alan, and Tyler Muir. 2016. *Volatility-Managed Portfolios*. NBER Working Paper no. 22208. Cambridge, MA: National Bureau of Economic Research. https://doi.org/10.3386/w22208.

ok

2. Allocation

Student-by-student listing and description of contributions.

a. Student 1 :Nigel Li

Nigel led the overall research direction and methodological design, conducted the core theoretical modelling and literature synthesis, and took the main responsibility for writing and structuring the paper. He also played a central role in interpreting the empirical results and refining the final arguments

b. Student 2 :Ryan Hou

Ryan contributed to the implementation and empirical validation of the proposed strategy, including backtesting, robustness analysis, and risk diagnostics. He also assisted in refining the trading rules and contributed to the methodology and results sections through revisions and feedback.

c. Student 3:Jesse Price

Jesse contributed by reviewing code and result visualisations. He participated in discussions on model design and provided feedback during the drafting process.

ok

3. Code

```
## Assignment 12: Finalised codebase showcasing all relevant results

# This Notebook
1) loads & aligns gov/IRS/futures panels,
2) builds fly CRD signals (baseline fixed-z + rolling-z + IRS-confirmed z),
3) runs IS backtests for Baseline/R1/R2/R3,
4) produces IS plots + stats + correlation heatmaps,
5) runs OOS backtest for R3 only (signals computed on full history; OOS slice from 2020
onwards),
6) produces OOS plots + stats for R3.
import numpy as np
import pandas as pd
from pathlib import Path
import matplotlib.pyplot as plt


# IS & OOS windwos:
IN_SAMPLE_START = pd.Timestamp("2004-01-01")
IN_SAMPLE_END   = pd.Timestamp("2019-12-31")
```



```
ANNUAL_DAYS = 252
DT = 1.0 / ANNUAL_DAYS

H_MONTH = 1
H = H_MONTH / 12.0

ROLL_WIN_6M = 126
MA_WIN = 5

ENTRY_Z = 1.6
EXIT_Z = 0.2

DV01_BUDGET_TOTAL = 100_000
DV01_LIMIT_FLY = 0.25 * DV01_BUDGET_TOTAL
CAPITAL_PER_DV01 = 100.0
MAX_FLY_POSITIONS = 4

# Baseline stale normalisation - just include to show its appalling performance
BASELINE_FIXED_WIN = 2520  # ~10y trading days

xlsx_path = Path("Yield curve arb.xlsx")

CURVE_COLS = {
    "US": ["USG2YR", "USG5YR", "US10GYR", "US30GYR"],
    "DE": ["GDBR2", "GDBR5", "GDBR10", "GDBR30"],
    "UK": ["GUKG2", "GUKG5", "GUKG10", "GUKG30"],
    "IT": ["GBTPGR2", "GBTPGR5", "GBTPGR10", "GBTPGR30"],
    "JP": ["JGBS2", "JGBS5", "JGBS10", "JGBS30"],
    "AU": ["GTAUD2Y", "GTAUD5Y", "GTAUD10Y", "GTAUD30Y"],
    "CA": ["GTCAD2Y", "GTCAD5Y", "GTCAD10Y", "GTCAD30Y"],
}

TENOR_MAP = {
    "USG2YR": 2.0, "USG5YR": 5.0, "US10GYR": 10.0, "US30GYR": 30.0,
    "GDBR2": 2.0, "GDBR5": 5.0, "GDBR10": 10.0, "GDBR30": 30.0,
    "GUKG2": 2.0, "GUKG5": 5.0, "GUKG10": 10.0, "GUKG30": 30.0,
    "GBTPGR2": 2.0, "GBTPGR5": 5.0, "GBTPGR10": 10.0, "GBTPGR30": 30.0,
    "JGBS2": 2.0, "JGBS5": 5.0, "JGBS10": 10.0, "JGBS30": 30.0,
    "GTAUD2Y": 2.0, "GTAUD5Y": 5.0, "GTAUD10Y": 10.0, "GTAUD30Y": 30.0,
    "GTCAD2Y": 2.0, "GTCAD5Y": 5.0, "GTCAD10Y": 10.0, "GTCAD30Y": 30.0,
}

IRS_CURVE_COLS = {
    "US": ["USSW2", "USSW5", "USSW10", "USSW30"],
    "DE": ["EUSA2", "EUSA5", "EUSA10", "EUSA30"],
```



```
    "UK": ["BPSW2", "BPSW5", "BPSW10", "BPSW30"],
    "JP": ["JYSW2", "JYSW5", "JYSW10", "JYSW30"],
    "CA": ["CDSW2", "CDSW5", "CDSW10", "CDSW30"],
    "AU": ["ADSW2", "ADSW5", "ADSW10", "ADSW30"],
}

IRS_TENOR_MAP = {
    "USSW2": 2.0, "USSW5": 5.0, "USSW10": 10.0, "USSW30": 30.0,
    "EUSA2": 2.0, "EUSA5": 5.0, "EUSA10": 10.0, "EUSA30": 30.0,
    "BPSW2": 2.0, "BPSW5": 5.0, "BPSW10": 10.0, "BPSW30": 30.0,
    "JYSW2": 2.0, "JYSW5": 5.0, "JYSW10": 10.0, "JYSW30": 30.0,
    "CDSW2": 2.0, "CDSW5": 5.0, "CDSW10": 10.0, "CDSW30": 30.0,
    "ADSW2": 2.0, "ADSW5": 5.0, "ADSW10": 10.0, "ADSW30": 30.0,
}

FLIES = [(2.0, 5.0, 10.0), (5.0, 10.0, 30.0)]

FUT_MAP = {
    "US": {2.0: "TU1 Comdty", 5.0: "FV1 Comdty", 10.0: "TY1 Comdty", 30.0: "US1 Comdty"},
    "DE": {2.0: "DU1 Comdty", 5.0: "OE1 Comdty", 10.0: "RX1 Comdty", 30.0: "UB1 
Comdty"},
    "UK": {2.0: "WB1 Comdty", 5.0: "WX1 Comdty", 10.0: "G 1 Comdty", 30.0: "UGL1 
Comdty"},
}
CONTRACT_MULT = 1000.0

COST_ROUNDTRIP_BP_COUNTRY = {
    "US": 0.12, "DE": 0.12, "UK": 0.18, "JP": 0.08, "AU": 0.35, "CA": 0.25, "IT": 0.50,
}
FLY_LIQ_MULT = {"2s5s10s": 1.0, "5s10s30s": 1.3}

# Load data - Align the excel to panels etc
def _read_pairs(sheet_name: str, pairs, dayfirst=False, div100=False) -> pd.DataFrame:
    raw = pd.read_excel(xlsx_path, sheet_name=sheet_name)
    series_list = []
    for date_col, val_col in pairs:
        df = raw[[date_col, val_col]].copy()
        df.columns = ["Date", val_col]
        df = df.dropna(how="all")
        df["Date"] = pd.to_datetime(df["Date"], errors="coerce", dayfirst=dayfirst)
        df = df.dropna(subset=["Date"]).set_index("Date").sort_index()
        df = df[~df.index.duplicated(keep="last")]
        series_list.append(df)
    out = pd.concat(series_list, axis=1).sort_index().ffill()
    return (out / 100.0) if div100 else out
```



```
def load_sovereign_yields() -> pd.DataFrame:
    yc_pairs = [
        ("Date", "USG2YR"), ("Date.1", "USG5YR"), ("Date.2", "US10GYR"), ("Date.3",
"US30GYR"),
        ("Date.4", "GDBR2"), ("Date.5", "GDBR5"), ("Date.6", "GDBR10"), ("Date.7",
"GDBR30"),
        ("Date.8", "GUKG2"), ("Date.9", "GUKG5"), ("Date.10", "GUKG10"), ("Date.11",
"GUKG30"),
        ("Date.12", "GBTPGR2"), ("Date.13", "GBTPGR5"), ("Date.14", "GBTPGR10"),
("Date.15", "GBTPGR30"),
        ("Date.16", "JGBS2"), ("Date.17", "JGBS5"), ("Date.18", "JGBS10"), ("Date.19",
"JGBS30"),
        ("Date.20", "GTAUD2Y"), ("Date.21", "GTAUD5Y"), ("Date.22", "GTAUD10Y"),
("Date.23", "GTAUD30Y"),
        ("Date.24", "GTCAD2Y"), ("Date.25", "GTCAD5Y"), ("Date.26", "GTCAD10Y"),
("Date.27", "GTCAD30Y"),
    ]
    return _read_pairs("Yield Signals", yc_pairs, dayfirst=False, div100=True)


def load_futures_prices() -> pd.DataFrame:
    fut_pairs = [
        ("Date", "TU1 Comdty"), ("Date.1", "FV1 Comdty"), ("Date.2", "TY1 Comdty"), ("Date.3",
"US1 Comdty"),
        ("Date.4", "DU1 Comdty"), ("Date.5", "OE1 Comdty"), ("Date.6", "RX1 Comdty"),
("Date.7", "UB1 Comdty"),
        ("Date.8", "WB1 Comdty"), ("Date.9", "WX1 Comdty"), ("Date.10", "G 1 Comdty"),
("Date.11", "UGL1 Comdty"),
    ]
    return _read_pairs("Futs", fut_pairs, dayfirst=True, div100=False)


def load_irs_yields() -> pd.DataFrame:
    irs_pairs = [
        ("Date", "USSW2"), ("Date.1", "USSW5"), ("Date.2", "USSW10"), ("Date.3", "USSW30"),
        ("Date.4", "EUSA2"), ("Date.5", "EUSA5"), ("Date.6", "EUSA10"), ("Date.7", "EUSA30"),
        ("Date.8", "BPSW2"), ("Date.9", "BPSW5"), ("Date.10", "BPSW10"), ("Date.11",
"BPSW30"),
        ("Date.12", "JYSW2"), ("Date.13", "JYSW5"), ("Date.14", "JYSW10"), ("Date.15",
"JYSW30"),
        ("Date.16", "CDSW2"), ("Date.17", "CDSW5"), ("Date.18", "CDSW10"), ("Date.19",
"CDSW30"),
        ("Date.20", "ADSW2"), ("Date.21", "ADSW5"), ("Date.22", "ADSW10"), ("Date.23",
"ADSW30"),
    ]
```



```
    return _read_pairs("IRS", irs_pairs, dayfirst=True, div100=True)


def build_curve_df_for_country(panel: pd.DataFrame, country: str, cols_map: dict, tenor_map: 
dict) -> pd.DataFrame:
    cols = cols_map[country]
    sub = panel[cols].copy()
    sub.columns = [tenor_map[c] for c in cols]
    return sub[sorted(sub.columns)]

# Loading data, and enforce the in-sample
yields = load_sovereign_yields()
futures_px = load_futures_prices()
irs = load_irs_yields()

curve_by_cty = {cty: build_curve_df_for_country(yields, cty, CURVE_COLS, TENOR_MAP) 
for cty in CURVE_COLS}
irs_curve_by_cty = {cty: build_curve_df_for_country(irs, cty, IRS_CURVE_COLS, 
IRS_TENOR_MAP) for cty in IRS_CURVE_COLS}

# aligning dates across gov + futures + IRS
common = None
for cty, df in curve_by_cty.items():
    common = df.index if common is None else common.intersection(df.index)
common = common.intersection(futures_px.index)
for cty, df in irs_curve_by_cty.items():
    common = common.intersection(df.index)
common = common.sort_values()

for cty in curve_by_cty:
    curve_by_cty[cty] = curve_by_cty[cty].reindex(common).ffill()
for cty in irs_curve_by_cty:
    irs_curve_by_cty[cty] = irs_curve_by_cty[cty].reindex(common).ffill()
futures_px = futures_px.reindex(common).ffill()

# Esnure the insample
ins_mask = (common >= IN_SAMPLE_START) & (common <= IN_SAMPLE_END)
ins_dates = common[ins_mask]

for cty in curve_by_cty:
    curve_by_cty[cty] = curve_by_cty[cty].reindex(ins_dates).ffill()
for cty in irs_curve_by_cty:
    irs_curve_by_cty[cty] = irs_curve_by_cty[cty].reindex(ins_dates).ffill()
futures_px = futures_px.reindex(ins_dates).ffill()

print("IN-SAMPLE date range:", ins_dates.min(), "→", ins_dates.max(), "| N =", len(ins_dates))
```

```
# rebi9old the rqe objects if missing
needed = ["curve_by_cty", "irs_curve_by_cty", "futures_px", "ins_dates"]
missing = [x for x in needed if x not in globals()]

if missing:
    print("Rebuilding missing objects:", missing)

    # load the raw panels
    yields = load_sovereign_yields()
    futures_px = load_futures_prices()
    irs = load_irs_yields()

    # build countryt curves
    curve_by_cty = {cty: build_curve_df_for_country(yields, cty, CURVE_COLS,
TENOR_MAP) for cty in CURVE_COLS}
    irs_curve_by_cty = {cty: build_curve_df_for_country(irs, cty, IRS_CURVE_COLS,
IRS_TENOR_MAP) for cty in IRS_CURVE_COLS}

    # Align dates across gov + futures + IRS
    common = None
    for cty, df in curve_by_cty.items():
        common = df.index if common is None else common.intersection(df.index)
    common = common.intersection(futures_px.index)
    for cty, df in irs_curve_by_cty.items():
        common = common.intersection(df.index)
    common = common.sort_values()

    for cty in curve_by_cty:
        curve_by_cty[cty] = curve_by_cty[cty].reindex(common).ffill()
    for cty in irs_curve_by_cty:
        irs_curve_by_cty[cty] = irs_curve_by_cty[cty].reindex(common).ffill()
    futures_px = futures_px.reindex(common).ffill()

    # Enforce In-Sample - please dont break it again
    ins_mask = (common >= IN_SAMPLE_START) & (common <= IN_SAMPLE_END)
    ins_dates = common[ins_mask]

    for cty in curve_by_cty:
        curve_by_cty[cty] = curve_by_cty[cty].reindex(ins_dates).ffill()
    for cty in irs_curve_by_cty:
        irs_curve_by_cty[cty] = irs_curve_by_cty[cty].reindex(ins_dates).ffill()
    futures_px = futures_px.reindex(ins_dates).ffill()

    print("IN-SAMPLE date range:", ins_dates.min(), "→", ins_dates.max(), "| N =",
len(ins_dates))
```



```
else:
    print("All required objects already exist. Proceeding...")

# Helpers for bond math and the signal engine (baseline and the refinement)
```

- including in the baseline for this final?

```
def zero_coupon(y: float, T: float):
    P = np.exp(-y * T)
    dv01 = P * T * 1e-4
    return P, dv01

def interp_yield(curve_row: pd.Series, T: float) -> float:
    xs = curve_row.index.values.astype(float)
    ys = curve_row.values.astype(float)
    o = np.argsort(xs)
    xs, ys = xs[o], ys[o]
    return float(np.interp(T, xs, ys))

def roll_down_dy(curve_row: pd.Series, T: float, H: float) -> float:
    if T - H <= 0:
        return 0.0
    return interp_yield(curve_row, T - H) - interp_yield(curve_row, T)

def crd_single_leg(curve_row: pd.Series, T: float, H: float, fund_rate: float):
    y = interp_yield(curve_row, T)
    P, dv01 = zero_coupon(y, T)
    dy_roll = roll_down_dy(curve_row, T, H)
    roll_pnl = -dv01 * (dy_roll * 1e4)
    carry = (y - fund_rate) * H * P
    crd_ret = (roll_pnl + carry) / P
    return crd_ret, P, dv01

def fly_weight_self_financing_dv01_neutral(dv01_L, dv01_B, dv01_R):
    A = np.array([[dv01_L, dv01_R],
                  [1.0,    1.0]])
    b = np.array([-dv01_B, -1.0])
    wL, wR = np.linalg.solve(A, b)
    return float(wL), 1.0, float(wR)

def compute_signal_for_fly(curve_df: pd.DataFrame, tenors: tuple, H: float,
                           ma_win: int, roll_win: int, fixed_win: int):
    T_L, T_B, T_R = tenors
    rows = []
    for date, curve_row in curve_df.iterrows():
        if curve_row.isna().any():
            rows.append((date, np.nan, np.nan, np.nan, np.nan, np.nan))
```

```
            continue
        fund_rate = float(curve_row.loc[2.0]) if 2.0 in curve_row.index else float(curve_row.iloc[0])

        crd_L, _, dv01_L = crd_single_leg(curve_row, T_L, H, fund_rate)
        crd_B, _, dv01_B = crd_single_leg(curve_row, T_B, H, fund_rate)
        crd_R, _, dv01_R = crd_single_leg(curve_row, T_R, H, fund_rate)

        wL, wB, wR = fly_weight_self_financing_dv01_neutral(dv01_L, dv01_B, dv01_R)
        fly_crd = wL * crd_L + wB * crd_B + wR * crd_R
        fly_dv01_abs = abs(dv01_L * wL) + abs(dv01_B * wB) + abs(dv01_R * wR)

        rows.append((date, fly_crd, wL, wB, wR, fly_dv01_abs))

    sig = pd.DataFrame(rows,
columns=["Date","sig_raw","wL","wB","wR","fly_dv01_abs"]).set_index("Date")

    sig["sig_smooth"] = sig["sig_raw"].rolling(ma_win, min_periods=1).mean()
    roll = sig["sig_smooth"].rolling(roll_win, min_periods=20)
    sig["z_roll"] = (sig["sig_smooth"] - roll.mean()) / roll.std(ddof=1)

    # baseline "stale" z-score: freeze mean/std from early sample window - terrible idea, but
included for comparison (the one before broke, cant fix the codebase)
    s = sig["sig_smooth"]
    mu0 = s.rolling(fixed_win, min_periods=fixed_win).mean()
    sd0 = s.rolling(fixed_win, min_periods=fixed_win).std(ddof=1)

    first_mu = mu0.dropna().iloc[0] if len(mu0.dropna()) else np.nan
    first_sd = sd0.dropna().iloc[0] if len(sd0.dropna()) else np.nan

    sig["z_fixed"] = (s - first_mu) / first_sd
    return sig

def build_signals(curve_by_cty: dict, flies: list[tuple]) -> dict:
    sig_by_key = {}
    for cty, cdf in curve_by_cty.items():
        for tenors in flies:
            if not all(t in cdf.columns for t in tenors):
                continue
            fly_name = f"{int(tenors[0])}s{int(tenors[1])}s{int(tenors[2])}s"
            sig_df = compute_signal_for_fly(
                cdf, tenors, H,
                ma_win=MA_WIN, roll_win=ROLL_WIN_6M,
fixed_win=BASELINE_FIXED_WIN
            )
            sig_by_key[(cty, fly_name)] = {"tenors": tenors, "sig_df": sig_df}
    return sig_by_key
```



```
# Adding the IRS confirmation for the R3
def apply_irs_confirmation(sig_gov: dict, sig_irs: dict, z_col: str = "z_roll") -> dict:
    for key, obj in sig_gov.items():
        g = obj["sig_df"]

        if key not in sig_irs:
            g["z_irs"] = np.nan
            g["filter_ok"] = True
            g["z_filtered"] = g[z_col]
            continue

        i = sig_irs[key]["sig_df"].reindex(g.index)
        g["z_irs"] = i[z_col]

        z_g = g[z_col]
        z_i = g["z_irs"]

        irs_missing = z_i.isna()
        both_valid = z_g.notna() & z_i.notna()
        same_sign = (np.sign(z_g) * np.sign(z_i) > 0) & both_valid

        g["filter_ok"] = (irs_missing & z_g.notna()) | same_sign
        g["z_filtered"] = np.where(g["filter_ok"], z_g, np.nan)

    return sig_gov

# Building the backtest componenets - accounting for costs and other tx costs
def zero_price_from_curve(curve_row: pd.Series, T: float) -> float:
    y = interp_yield(curve_row, T)
    P, _ = zero_coupon(y, T)
    return P

def run_portfolio_backtest(curve_by_cty, sig_by_key, futures_px,
                           z_use, universe_keys, max_positions,
                           dv01_limit_fly, entry_z, exit_z,
                           use_futures, use_country_costs,
                           cash_proxy_country="US"):
    CAPITAL_TOTAL = max_positions * dv01_limit_fly * CAPITAL_PER_DV01
    CAP_PER_FLY = dv01_limit_fly * CAPITAL_PER_DV01
    FUT_ENTRY_Z = entry_z + 1.0

    common_dates = None
    for cty, cdf in curve_by_cty.items():
        common_dates = cdf.index if common_dates is None else
common_dates.intersection(cdf.index)
```



```
    common_dates = common_dates.sort_values()
    if futures_px is not None:
        common_dates = common_dates.intersection(futures_px.index).sort_values()

    positions = {
        key: dict(pos_dir=0, inst_type="none", wL=0.0, wB=0.0, wR=0.0,
                  scale=0.0, prev_curve=None, nL=0.0, nB=0.0, nR=0.0,
                  hold_days=0, trade_pnl=0.0, entry_date=None)
        for key in universe_keys
    }

    portfolio_rows, trade_log = [], []

    for t_idx, date in enumerate(common_dates):
        date_prev = None if t_idx == 0 else common_dates[t_idx - 1]
        daily_pnl_gross, trade_cost_today = 0.0, 0.0

        # We need mark to market for the sharpe later
        for key, st in positions.items():
            cty, fly_name = key
            curve_today = curve_by_cty[cty].loc[date]
            prev_curve = st["prev_curve"]
            pnl = 0.0

            if st["inst_type"] == "bond" and st["pos_dir"] != 0 and prev_curve is not None and
st["scale"] > 0:
                T_L, T_B, T_R = sig_by_key[key]["tenors"]
                wL, wB, wR = st["wL"], st["wB"], st["wR"]

                P_L0 = zero_price_from_curve(prev_curve, T_L)
                P_B0 = zero_price_from_curve(prev_curve, T_B)
                P_R0 = zero_price_from_curve(prev_curve, T_R)

                P_L1 = zero_price_from_curve(curve_today, T_L)
                P_B1 = zero_price_from_curve(curve_today, T_B)
                P_R1 = zero_price_from_curve(curve_today, T_R)

                fly_leg_pnl = wL*(P_L1-P_L0) + wB*(P_B1-P_B0) + wR*(P_R1-P_R0)

                fund_prev = float(prev_curve.iloc[0])
                y_L_prev = interp_yield(prev_curve, T_L)
                y_B_prev = interp_yield(prev_curve, T_B)
                y_R_prev = interp_yield(prev_curve, T_R)

                carry_L = (y_L_prev - fund_prev) * DT * P_L0
                carry_B = (y_B_prev - fund_prev) * DT * P_B0
```



```
            carry_R = (y_R_prev - fund_prev) * DT * P_R0
            fly_carry = wL*carry_L + wB*carry_B + wR*carry_R

            pnl = (fly_leg_pnl + fly_carry) * st["scale"] * st["pos_dir"]

        if st["inst_type"] == "fut" and st["pos_dir"] != 0 and futures_px is not None and
date_prev is not None:
            if cty in FUT_MAP:
                T_L, T_B, T_R = sig_by_key[key]["tenors"]
                tick_L = FUT_MAP[cty].get(T_L)
                tick_B = FUT_MAP[cty].get(T_B)
                tick_R = FUT_MAP[cty].get(T_R)
                if all(t in futures_px.columns for t in [tick_L, tick_B, tick_R]):
                    dPL = futures_px.loc[date, tick_L] - futures_px.loc[date_prev, tick_L]
                    dPB = futures_px.loc[date, tick_B] - futures_px.loc[date_prev, tick_B]
                    dPR = futures_px.loc[date, tick_R] - futures_px.loc[date_prev, tick_R]
                    pnl += CONTRACT_MULT * (st["nL"]*dPL + st["nB"]*dPB + st["nR"]*dPR)

        daily_pnl_gross += pnl
        if st["pos_dir"] != 0:
            st["trade_pnl"] += pnl
        st["prev_curve"] = curve_today

    # Cash proxy
    active = sum(1 for st in positions.values() if st["pos_dir"] != 0)
    cap_used = active * CAP_PER_FLY
    cap_unused = max(CAPITAL_TOTAL - cap_used, 0.0)

    cash_pnl = 0.0
    y2 = curve_by_cty[cash_proxy_country].loc[date, 2.0]
    cash_pnl = cap_unused * y2 * DT
    daily_pnl_gross += cash_pnl

    # Here define the exits
    for key, st in positions.items():
        if st["pos_dir"] == 0:
            continue
        z = sig_by_key[key]["sig_df"].loc[date, z_use]
        if (not np.isnan(z)) and abs(z) < exit_z:
            cty, fly_name = key
            if use_country_costs:
                base_rt = COST_ROUNDTRIP_BP_COUNTRY.get(cty, 0.15)
                mult = FLY_LIQ_MULT.get(fly_name, 1.0)
                trade_cost_today += (base_rt * mult / 2.0) * dv01_limit_fly

            trade_log.append(dict(
```



```
                country=cty, fly=fly_name,
                entry_date=st["entry_date"], exit_date=date,
                holding_days=st["hold_days"], side=st["pos_dir"],
                pnl=st["trade_pnl"], inst_type=st["inst_type"],
            ))

            st.update(pos_dir=0, inst_type="none", scale=0.0,
                      wL=0.0, wB=0.0, wR=0.0,
                      nL=0.0, nB=0.0, nR=0.0,
                      hold_days=0, trade_pnl=0.0, entry_date=None)

    # --- Entries (top-|z| up to capacity) ---
    active = sum(1 for st in positions.values() if st["pos_dir"] != 0)
    capacity = max_positions - active

    if capacity > 0:
        candidates = []
        for key, st in positions.items():
            if st["pos_dir"] != 0:
                continue
            row = sig_by_key[key]["sig_df"].loc[date]
            z = row[z_use]
            if np.isnan(z) or abs(z) <= entry_z:
                continue
            candidates.append((abs(z), z, key, row))
        candidates.sort(key=lambda x: x[0], reverse=True)

        for abs_z, z, key, row in candidates[:capacity]:
            cty, fly_name = key
            if use_country_costs:
                base_rt = COST_ROUNDTRIP_BP_COUNTRY.get(cty, 0.15)
                mult = FLY_LIQ_MULT.get(fly_name, 1.0)
                trade_cost_today += (base_rt * mult / 2.0) * dv01_limit_fly

            pos_dir = 1 if z > 0 else -1
            st = positions[key]
            st["pos_dir"] = pos_dir
            st["hold_days"] = 0
            st["trade_pnl"] = 0.0
            st["entry_date"] = date
            st["wL"], st["wB"], st["wR"] = row["wL"], row["wB"], row["wR"]

            use_fut = (use_futures and futures_px is not None and cty in FUT_MAP and abs_z >=
FUT_ENTRY_Z)

            if use_fut:
```



```
            T_L, T_B, T_R = sig_by_key[key]["tenors"]
            tick_L = FUT_MAP[cty].get(T_L)
            tick_B = FUT_MAP[cty].get(T_B)
            tick_R = FUT_MAP[cty].get(T_R)
            if not all(t in futures_px.columns for t in [tick_L, tick_B, tick_R]):
                use_fut = False

        if use_fut:
            st["inst_type"] = "fut"
            pL = futures_px.loc[date, tick_L]
            pB = futures_px.loc[date, tick_B]
            pR = futures_px.loc[date, tick_R]
            wL, wB, wR = st["wL"], st["wB"], st["wR"]
            wabs = abs(wL) + abs(wB) + abs(wR)
            wabs = 1.0 if wabs == 0 else wabs
            cap_L = CAP_PER_FLY * abs(wL) / wabs
            cap_B = CAP_PER_FLY * abs(wB) / wabs
            cap_R = CAP_PER_FLY * abs(wR) / wabs
            st["nL"] = pos_dir * np.sign(wL) * cap_L / (pL * CONTRACT_MULT)
            st["nB"] = pos_dir * np.sign(wB) * cap_B / (pB * CONTRACT_MULT)
            st["nR"] = pos_dir * np.sign(wR) * cap_R / (pR * CONTRACT_MULT)
        else:
            st["inst_type"] = "bond"
            st["scale"] = dv01_limit_fly / max(row["fly_dv01_abs"], 1e-10)

    for st in positions.values():
        if st["pos_dir"] != 0:
            st["hold_days"] += 1

    pnl_net = daily_pnl_gross - trade_cost_today
    portfolio_rows.append(dict(
        date=date, pnl_gross=daily_pnl_gross, pnl_net=pnl_net,
        cash_pnl=cash_pnl, trade_cost=trade_cost_today,
        num_active_positions=sum(1 for st in positions.values() if st["pos_dir"] != 0),
    ))

  port = pd.DataFrame(portfolio_rows).set_index("date")
  trades = pd.DataFrame(trade_log)

  port["ret_net"] = port["pnl_net"] / (max_positions * dv01_limit_fly * CAPITAL_PER_DV01)
  port["cum_ret_net"] = (1 + port["ret_net"].fillna(0)).cumprod() - 1
  port["cum_pnl_net"] = port["pnl_net"].cumsum()
  return port, trades

# running it
# Build signals (in-sample only because curves were filtered already)
```



```
sig_gov = build_signals(curve_by_cty, FLIES)
sig_irs = build_signals(irs_curve_by_cty, FLIES)
sig_gov = apply_irs_confirmation(sig_gov, sig_irs, z_col="z_roll")

US_ONLY = sorted([k for k in sig_gov.keys() if k[0] == "US"])
print("US_ONLY universe:", US_ONLY)

FULL_UNIVERSE = sorted(list(sig_gov.keys()))

results = {}

# Baseline performance - this is the same as prev ass
port0, tr0 = run_portfolio_backtest(
    curve_by_cty, sig_gov, futures_px=None,
    z_use="z_fixed",
    universe_keys=US_ONLY,
    max_positions=1,
    dv01_limit_fly=DV01_LIMIT_FLY,
    entry_z=ENTRY_Z, exit_z=EXIT_Z,
    use_futures=False, use_country_costs=False
)
results["Baseline (IS, US only, fixed z)"] = (port0, tr0)

# R1
port1, tr1 = run_portfolio_backtest(
    curve_by_cty, sig_gov, futures_px=None,
    z_use="z_roll",
    universe_keys=US_ONLY,
    max_positions=1,
    dv01_limit_fly=DV01_LIMIT_FLY,
    entry_z=ENTRY_Z, exit_z=EXIT_Z,
    use_futures=False, use_country_costs=False
)
results["R1 (IS, roll z + MA)"] = (port1, tr1)

# R2
port2, tr2 = run_portfolio_backtest(
    curve_by_cty, sig_gov, futures_px=futures_px,
    z_use="z_roll",
    universe_keys=FULL_UNIVERSE,
    max_positions=MAX_FLY_POSITIONS,
    dv01_limit_fly=DV01_LIMIT_FLY,
    entry_z=ENTRY_Z, exit_z=EXIT_Z,
    use_futures=True, use_country_costs=True
)
results["R2 (IS, multi-cty + futures + costs)"] = (port2, tr2)
```



```
# Re3
port3, tr3 = run_portfolio_backtest(
    curve_by_cty, sig_gov, futures_px=futures_px,
    z_use="z_filtered",
    universe_keys=FULL_UNIVERSE,
    max_positions=MAX_FLY_POSITIONS,
    dv01_limit_fly=DV01_LIMIT_FLY,
    entry_z=ENTRY_Z, exit_z=EXIT_Z,
    use_futures=True, use_country_costs=True
)
results["R3 (IS, R2 + IRS filter)"] = (port3, tr3)

# Non-Cum Pnl
fig, ax = plt.subplots(figsize=(12, 4))
for name, (p, _) in results.items():
    p["pnl_net"].plot(ax=ax, linewidth=1.0, alpha=0.85, label=name)
ax.set_title("IN-SAMPLE Daily Net PnL (non-cumulative): Baseline vs Refinements")
ax.set_ylabel("Daily net PnL ($)")
ax.grid(True, linestyle="--", alpha=0.35)
ax.legend()
plt.tight_layout()
plt.show()

print({k: len(v[1]) for k, v in results.items()})
```

ASS 12 Meat and bone

- This has the plots and the correl analysis

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

results_meta = {
    "Baseline (IS, US only, fixed z)": dict(max_positions=1,  has_costs=False, z_col="z_fixed"),
    "R1 (IS, roll z + MA)"           : dict(max_positions=1,  has_costs=False, z_col="z_roll"),
    "R2 (IS, multi-cty + futures + costs)": dict(max_positions=MAX_FLY_POSITIONS,
has_costs=True,  z_col="z_roll"),
    "R3 (IS, R2 + IRS filter)"           : dict(max_positions=MAX_FLY_POSITIONS,
has_costs=True,  z_col="z_filtered"),
}

def _capital_total(max_positions: int) -> float:
    return max_positions * DV01_LIMIT_FLY * CAPITAL_PER_DV01
```



```
def _safe_series(x: pd.Series) -> pd.Series:
    return x.replace([np.inf, -np.inf], np.nan).dropna()

def _max_drawdown_from_simple_cumret(ret_simple: pd.Series) -> float:
    """
    ret_simple = daily returns (as fraction of capital), NOT compounded.
    Equity curve for DD: 1 + cumsum(ret_simple).
    """
    r = _safe_series(ret_simple)
    if r.empty:
        return np.nan
    eq = 1.0 + r.cumsum()
    dd = eq / eq.cummax() - 1.0
    return float(dd.min())

def compute_summary_stats(port: pd.DataFrame, trades: pd.DataFrame, capital_total: float) ->
pd.Series:
    out = {}
    if port is None or port.empty:
        return pd.Series(out)

    # daily returns as a % of current capital
    ret_net   = _safe_series(port["pnl_net"]   / capital_total)
    ret_gross = _safe_series(port["pnl_gross"] / capital_total)

    def ann_stats(r: pd.Series):
        if len(r) < 2:
            return np.nan, np.nan, np.nan
        mu = r.mean()
        vol = r.std(ddof=1)
        ann_ret = mu * ANNUAL_DAYS
        ann_vol = vol * np.sqrt(ANNUAL_DAYS)
        sharpe = ann_ret / ann_vol if ann_vol > 0 else np.nan
        return float(ann_ret), float(ann_vol), float(sharpe)

    ann_ret_net, ann_vol_net, sharpe_net = ann_stats(ret_net)
    ann_ret_gross, ann_vol_gross, sharpe_gross = ann_stats(ret_gross)

    max_dd = _max_drawdown_from_simple_cumret(ret_net)
    calmar = ann_ret_net / abs(max_dd) if (not np.isnan(max_dd) and max_dd < 0) else np.nan

    downside = ret_net[ret_net < 0]
    if len(downside) >= 2:
        down_std = downside.std(ddof=1) * np.sqrt(ANNUAL_DAYS)
        sortino = ann_ret_net / down_std if down_std > 0 else np.nan
```



```
else:
    sortino = np.nan

def var_es(r: pd.Series, alpha=0.99):
    r = _safe_series(r)
    if len(r) < 50:
        return np.nan, np.nan
    s = np.sort(r.values)
    idx = int((1 - alpha) * len(s))
    idx = int(np.clip(idx, 0, len(s) - 1))
    var = float(s[idx])
    es = float(s[:idx + 1].mean()) if idx >= 0 else np.nan
    return var, es

var99, es99 = var_es(ret_net, 0.99)
var95, es95 = var_es(ret_net, 0.95)

if trades is not None and not trades.empty:
    pnl_tr = trades["pnl"].dropna()
    wins = pnl_tr[pnl_tr > 0]
    losses = pnl_tr[pnl_tr < 0]
    hit = len(wins) / len(pnl_tr) if len(pnl_tr) else np.nan
    avg_win = wins.mean() if len(wins) else np.nan
    avg_loss = losses.mean() if len(losses) else np.nan
    avg_hold = trades["holding_days"].mean() if "holding_days" in trades.columns else np.nan
    n_trades = len(pnl_tr)

    n_years = len(ret_net) / ANNUAL_DAYS if len(ret_net) else np.nan
    turnover = n_trades / n_years if (n_years and n_years > 0) else np.nan
else:
    hit = avg_win = avg_loss = avg_hold = np.nan
    n_trades = 0
    turnover = np.nan

total_cost = float(port["trade_cost"].sum()) if "trade_cost" in port.columns else np.nan
avg_daily_cost = float(port["trade_cost"].mean()) if "trade_cost" in port.columns else np.nan

out.update({
    "Ann return net (%)": 100 * ann_ret_net,
    "Ann vol net (%)": 100 * ann_vol_net,
    "Sharpe net": sharpe_net,
    "Ann return gross (%)": 100 * ann_ret_gross,
    "Ann vol gross (%)": 100 * ann_vol_gross,
    "Sharpe gross": sharpe_gross,
    "Max DD (net, %)": 100 * max_dd,
    "Calmar (net)": calmar,
```



```python
        "Sortino (net)": sortino,
        "VaR 99% (daily, %cap)": 100 * var99,
        "ES 99% (daily, %cap)": 100 * es99,
        "VaR 95% (daily, %cap)": 100 * var95,
        "ES 95% (daily, %cap)": 100 * es95,
        "# round-trips": n_trades,
        "Hit rate (%)": 100 * hit if not np.isnan(hit) else np.nan,
        "Avg win ($)": avg_win,
        "Avg loss ($)": avg_loss,
        "Avg hold (days)": avg_hold,
        "Trades per year": turnover,
        "Total tx cost ($)": total_cost,
        "Avg daily tx cost ($)": avg_daily_cost,
    })
    return pd.Series(out)

def _heatmap(ax, mat: np.ndarray, xlabels, ylabels, title: str,
             vmin=-1.0, vmax=1.0, annotate=True, fmt="{:.2f}"):
    im = ax.imshow(mat, aspect="auto", vmin=vmin, vmax=vmax)
    ax.set_title(title)
    ax.set_xticks(np.arange(len(xlabels)))
    ax.set_yticks(np.arange(len(ylabels)))
    ax.set_xticklabels(xlabels, rotation=45, ha="right")
    ax.set_yticklabels(ylabels)

    # Maybe make it columbia theme do we haveea a theme guys?
    ax.set_xticks(np.arange(-0.5, len(xlabels), 1), minor=True)
    ax.set_yticks(np.arange(-0.5, len(ylabels), 1), minor=True)
    ax.grid(which="minor", linestyle="--", linewidth=0.5)
    ax.tick_params(which="minor", bottom=False, left=False)

    if annotate:
        for i in range(mat.shape[0]):
            for j in range(mat.shape[1]):
                val = mat[i, j]
                if np.isfinite(val):
                    ax.text(j, i, fmt.format(val), ha="center", va="center", fontsize=8)
    return im

# pnl plot
fig, ax = plt.subplots(figsize=(12, 4))
for name, (p, _) in results.items():
    p["pnl_net"].fillna(0.0).plot(ax=ax, linewidth=1.0, alpha=0.85, label=name)
ax.set_title("IN-SAMPLE Daily Net PnL (non-cumulative)")
ax.set_ylabel("Daily net PnL ($)")
ax.grid(True, linestyle="--", alpha=0.35)
```



```
ax.legend()
plt.tight_layout()
plt.show()

# cum plot
fig, ax = plt.subplots(figsize=(12, 5))
for name, (p, _) in results.items():
    cap = _capital_total(results_meta[name]["max_positions"])
    daily_ret = (p["pnl_net"].fillna(0.0) / cap)
    cum_simple = daily_ret.cumsum()
    (cum_simple * 100).plot(ax=ax, linewidth=2, alpha=0.9, label=name)
ax.set_title("IN-SAMPLE Cumulative Return (net, simple sum, capital-normalised)")
ax.set_ylabel("Cumulative return (% of capital)")
ax.grid(True, linestyle="--", alpha=0.35)
ax.legend()
plt.tight_layout()
plt.show()

# Pre vs post - thi was prior feedback
def plot_cost_drag_simple(name: str):
    p, _ = results[name]
    cap = _capital_total(results_meta[name]["max_positions"])

    cum_gross = (p["pnl_gross"].fillna(0.0) / cap).cumsum()
    cum_net   = (p["pnl_net"].fillna(0.0)   / cap).cumsum()
    cum_cost  = (p["trade_cost"].fillna(0.0) / cap).cumsum()

    fig, ax = plt.subplots(figsize=(12, 5))
    (cum_gross * 100).plot(ax=ax, linewidth=2, label=f"{name}: Gross (pre-cost)")
    (cum_net   * 100).plot(ax=ax, linewidth=2, label=f"{name}: Net (post-cost)")
    (cum_cost  * 100).plot(ax=ax, linewidth=1.8, linestyle="--", label="Cumulative cost drag")
    ax.set_title(f"Cumulative Return: Gross vs Net vs Cost Drag — {name} (simple sum)")
    ax.set_ylabel("Cumulative % of capital")
    ax.grid(True, linestyle="--", alpha=0.35)
    ax.legend()
    plt.tight_layout()
    plt.show()

for nm in results:
    if results_meta[nm]["has_costs"]:
        plot_cost_drag_simple(nm)

# Add the active flies so we can see when active when not
def plot_active_overlay_pct(name: str):
    p, _ = results[name]
    cap = _capital_total(results_meta[name]["max_positions"])
```



```
    fig, ax1 = plt.subplots(figsize=(12, 5))
    cum_net_pct = (p["pnl_net"].fillna(0.0).cumsum() / cap) * 100
    cum_net_pct.plot(ax=ax1, linewidth=2, label="Cumulative net PnL (% of capital)")
    ax1.set_title(f"Cumulative Net PnL (% cap) and Active Flies — {name}")
    ax1.set_ylabel("Cumulative net PnL (% of capital)")
    ax1.grid(True, linestyle="--", alpha=0.35)

    ax2 = ax1.twinx()
    p["num_active_positions"].fillna(0).plot(ax=ax2, linestyle="--", alpha=0.8, label="# active
flies")
    ax2.set_ylabel("# active flies")

    h1, l1 = ax1.get_legend_handles_labels()
    h2, l2 = ax2.get_legend_handles_labels()
    ax1.legend(h1 + h2, l1 + l2, loc="upper left")
    plt.tight_layout()
    plt.show()

for nm in results:
    plot_active_overlay_pct(nm)

# Call table form before
stats_rows = []
for name, (p, tr) in results.items():
    cap = _capital_total(results_meta[name]["max_positions"])
    s = compute_summary_stats(p, tr, cap)
    s.name = name
    stats_rows.append(s)

stats_table = pd.DataFrame(stats_rows)
display(stats_table.round(3))

# Corr matrix
ret_panel = {}
for name, (p, _) in results.items():
    cap = _capital_total(results_meta[name]["max_positions"])
    ret_panel[name] = (p["pnl_net"] / cap)

ret_panel = pd.DataFrame(ret_panel).replace([np.inf, -np.inf], np.nan).dropna(how="any")
corr_variants = ret_panel.corr()

fig, ax = plt.subplots(figsize=(9, 7))
im = _heatmap(
    ax,
    corr_variants.values,
```



```
    xlabels=corr_variants.columns.tolist(),
    ylabels=corr_variants.index.tolist(),
    title="Correlation Matrix: Daily Net Returns (Baseline vs Refinements)",
    vmin=-1.0, vmax=1.0, annotate=True
)
fig.colorbar(im, ax=ax, fraction=0.046, pad=0.04)
plt.tight_layout()
plt.show()

# Corr matrix
bench_cols = [c for c in ["TY1 Comdty", "RX1 Comdty", "G 1 Comdty", "US1 Comdty"] if c in
futures_px.columns]
if bench_cols:
    fut_ret = futures_px[bench_cols].pct_change()

    corr_vs_fut = pd.DataFrame(index=list(results.keys()), columns=bench_cols, dtype=float)
    nobs_vs_fut = pd.DataFrame(index=list(results.keys()), columns=bench_cols, dtype=float)

    for name, (p, _) in results.items():
        cap = _capital_total(results_meta[name]["max_positions"])
        strat = (p["pnl_net"] / cap).rename("Strategy")
        merged = pd.concat([strat, fut_ret], axis=1).replace([np.inf, -np.inf], np.nan).dropna()

        if merged.shape[0] < 50:
            continue

        c = merged.corr().loc["Strategy", bench_cols]
        corr_vs_fut.loc[name, bench_cols] = c.values
        nobs_vs_fut.loc[name, bench_cols] = merged.shape[0]

    fig, ax = plt.subplots(figsize=(9, 5))
    im = _heatmap(
        ax,
        corr_vs_fut.values.astype(float),
        xlabels=bench_cols,
        ylabels=corr_vs_fut.index.tolist(),
        title="Correlation Matrix: Strategy Variants vs Rates Futures (daily returns)",
        vmin=-1.0, vmax=1.0, annotate=True
    )
    fig.colorbar(im, ax=ax, fraction=0.046, pad=0.04)
    plt.tight_layout()
    plt.show()

    print("Obs count used per strategy (rows) vs futures (cols):")
    display(nobs_vs_fut)
else:
```



```
    print("No benchmark futures columns found in futures_px for correlation matrix.")

# Use yfinance to get the data -
try:
    import yfinance as yf

    yahoo_tickers = {
        "SPY": "US Equities (SPY)",
        "EEM": "EM Equities (EEM)",
        "TLT": "US Long Treasuries (TLT)",
        "IEF": "US 7–10Y Treasuries (IEF)",
        "LQD": "US IG Credit (LQD)",
        "HYG": "US HY Credit (HYG)",
        "GLD": "Gold (GLD)",
        "DBC": "Broad Commodities (DBC)",
        "VNQ": "US REITs (VNQ)",
    }

    pick_name = "R3 (IS, R2 + IRS filter)"
    cap = _capital_total(results_meta[pick_name]["max_positions"])
    strategy_ret = (results[pick_name][0]["pnl_net"] / cap).dropna().rename("Strategy")

    start = strategy_ret.index.min().strftime("%Y-%m-%d")
    end   = strategy_ret.index.max().strftime("%Y-%m-%d")

    raw = yf.download(list(yahoo_tickers.keys()), start=start, end=end, group_by="ticker",
auto_adjust=False)

    # Extract Adj Close if possible
    if isinstance(raw.columns, pd.MultiIndex):
        prices = {}
        for tic in yahoo_tickers.keys():
            if tic in raw.columns.get_level_values(0):
                sub = raw[tic]
                col = "Adj Close" if "Adj Close" in sub.columns else ("Close" if "Close" in
sub.columns else None)
                if col:
                    prices[tic] = sub[col]
        prices = pd.DataFrame(prices)
    else:
        col = "Adj Close" if "Adj Close" in raw.columns else ("Close" if "Close" in raw.columns
else None)
        prices = raw[[col]].rename(columns={col: list(yahoo_tickers.keys())[0]})

    asset_ret = prices.pct_change()
```



```
    combined = pd.concat([strategy_ret, asset_ret], axis=1).replace([np.inf, -np.inf],
np.nan).dropna()
    combined = combined.rename(columns={k: v for k, v in yahoo_tickers.items() if k in
combined.columns})

    # just in case it dies later on other machines
    if "US Long Treasuries (TLT)" in combined.columns and "US 7–10Y Treasuries (IEF)" in
combined.columns:
        combined["FI RV proxy (TLT–IEF)"] = combined["US Long Treasuries (TLT)"] -
combined["US 7–10Y Treasuries (IEF)"]

    corr = combined.corr()

    fig, ax = plt.subplots(figsize=(10, 8))
    im = _heatmap(
        ax,
        corr.values,
        xlabels=corr.columns.tolist(),
        ylabels=corr.index.tolist(),
        title=f"Correlation Matrix: {pick_name} + Asset-Class Proxies (daily returns)",
        vmin=-1.0, vmax=1.0, annotate=True
    )
    fig.colorbar(im, ax=ax, fraction=0.046, pad=0.04)
    plt.tight_layout()
    plt.show()


    strat_corr = corr.loc["Strategy"].drop("Strategy").sort_values()
    display(strat_corr.to_frame("Corr w/ Strategy").round(4))

except Exception as e:
    print("Yahoo proxy block skipped (yfinance/internet issue). Error:", repr(e))

def _cap_for(name: str) -> float:
    return _capital_total(results_meta[name]["max_positions"])

def _cum_simple_pct(p: pd.DataFrame, cap: float, col: str) -> pd.Series:
    return (p[col].fillna(0.0).cumsum() / cap) * 100.0

# Ass12 req for the baseline v r1
base_name = "Baseline (IS, US only, fixed z)"
r1_name   = "R1 (IS, roll z + MA)"

p0, _ = results[base_name]
p1, _ = results[r1_name]
```

```
cap0 = _cap_for(base_name)
cap1 = _cap_for(r1_name)

fig, ax = plt.subplots(figsize=(12, 5))
_cum_simple_pct(p0, cap0, "pnl_net").plot(ax=ax, linewidth=2, label=base_name)
_cum_simple_pct(p1, cap1, "pnl_net").plot(ax=ax, linewidth=2, label=r1_name)

ax.set_title("Baseline vs R1 — Cumulative Net Return (simple sum, % of capital)")
ax.set_ylabel("Cumulative return (% of capital)")
ax.grid(True, linestyle="--", alpha=0.35)
ax.legend()
plt.tight_layout()
plt.show()


# the newer imrpovements running hte strates
def plot_cost_drag_and_flies(name: str):
    p, _ = results[name]
    cap = _cap_for(name)

    cum_gross = _cum_simple_pct(p, cap, "pnl_gross")
    cum_net   = _cum_simple_pct(p, cap, "pnl_net")
    cum_cost  = (p["trade_cost"].fillna(0.0).cumsum() / cap) * 100.0

    fig, ax1 = plt.subplots(figsize=(12, 5))

    cum_gross.plot(ax=ax1, linewidth=2, label="Gross (pre-cost)")
    cum_net.plot(ax=ax1, linewidth=2, label="Net (post-cost)")
    cum_cost.plot(ax=ax1, linewidth=1.8, linestyle="--", label="Cost drag (cumulative)")

    ax1.set_title(f"{name} — Cumulative Return (simple sum, % cap) + Active Flies")
    ax1.set_ylabel("Cumulative % of capital")
    ax1.grid(True, linestyle="--", alpha=0.35)

    ax2 = ax1.twinx()
    p["num_active_positions"].fillna(0).plot(
        ax=ax2, linestyle=":", linewidth=1.8, alpha=0.9, label="# active flies"
    )
    ax2.set_ylabel("# active flies")

    h1, l1 = ax1.get_legend_handles_labels()
    h2, l2 = ax2.get_legend_handles_labels()
    ax1.legend(h1 + h2, l1 + l2, loc="upper left")

    plt.tight_layout()
    plt.show()
```



```
plot_cost_drag_and_flies("R2 (IS, multi-cty + futures + costs)")
plot_cost_drag_and_flies("R3 (IS, R2 + IRS filter)")
## OOS Run for 2020 onwards
IS_R3_KEY = "R3 (IS, R2 + IRS filter)"
if "IN_SAMPLE_END" in globals():
    IS_END = pd.Timestamp(IN_SAMPLE_END)
elif "results" in globals() and (IS_R3_KEY in results) and (results[IS_R3_KEY][0] is not None)
and (not results[IS_R3_KEY][0].empty):
    IS_END = pd.Timestamp(results[IS_R3_KEY][0].index.max())
else:
    raise RuntimeError(
        "lol no go. go back and fix it ."
    )

OOS_START = IS_END + pd.Timedelta(days=1)
OOS_END   = None  # waht was last dae again
print("Requested OOS window:", OOS_START.date(), "→", ("END" if OOS_END is None else
pd.Timestamp(OOS_END).date()))



# load the panels
yields_full  = load_sovereign_yields()
futs_full    = load_futures_prices()
irs_full     = load_irs_yields()

curve_by_cty_full = {cty: build_curve_df_for_country(yields_full, cty, CURVE_COLS,
TENOR_MAP) for cty in CURVE_COLS}
irs_by_cty_full   = {cty: build_curve_df_for_country(irs_full,    cty, IRS_CURVE_COLS,
IRS_TENOR_MAP) for cty in IRS_CURVE_COLS}

# Align the daes for the OOS
common = None
for _, df in curve_by_cty_full.items():
    common = df.index if common is None else common.intersection(df.index)
common = common.intersection(futs_full.index)
for _, df in irs_by_cty_full.items():
    common = common.intersection(df.index)
common = common.sort_values()

for cty in curve_by_cty_full:
    curve_by_cty_full[cty] = curve_by_cty_full[cty].reindex(common).ffill()
for cty in irs_by_cty_full:
    irs_by_cty_full[cty] = irs_by_cty_full[cty].reindex(common).ffill()
futs_full = futs_full.reindex(common).ffill()
```



```
# Cut it up
oos_end_eff = common.max() if OOS_END is None else pd.Timestamp(OOS_END)
oos_dates = common[(common >= OOS_START) & (common <= oos_end_eff)]

if len(oos_dates) == 0:
    raise RuntimeError(
        f"No OOS dates available after {OOS_START.date()} in the aligned dataset.
common.max()={common.max().date()}"
    )

print("Aligned FULL range:", common.min().date(), "→", common.max().date(), "| N =",
len(common))
print("Aligned OOS  range :", oos_dates.min().date(), "→", oos_dates.max().date(), "| N =",
len(oos_dates))

# build the signals on the full OOS datset
sig_gov_full = build_signals(curve_by_cty_full, FLIES)
sig_irs_full = build_signals(irs_by_cty_full,   FLIES)
sig_gov_full = apply_irs_confirmation(sig_gov_full, sig_irs_full, z_col="z_roll")

def slice_signals(sig_by_key: dict, dates: pd.DatetimeIndex) -> dict:
    out = {}
    for k, obj in sig_by_key.items():
        out[k] = {"tenors": obj["tenors"], "sig_df": obj["sig_df"].reindex(dates)}
    return out

curve_by_cty_oos = {cty: df.reindex(oos_dates).ffill() for cty, df in curve_by_cty_full.items()}
futs_oos         = futs_full.reindex(oos_dates).ffill()
sig_gov_oos      = slice_signals(sig_gov_full, oos_dates)

# run the r3 oos backtest
FULL_UNIVERSE = sorted(list(sig_gov_oos.keys()))
p3, t3 = run_portfolio_backtest(
    curve_by_cty_oos, sig_gov_oos, futures_px=futs_oos,
    z_use="z_filtered",
    universe_keys=FULL_UNIVERSE,
    max_positions=MAX_FLY_POSITIONS,
    dv01_limit_fly=DV01_LIMIT_FLY,
    entry_z=ENTRY_Z, exit_z=EXIT_Z,
    use_futures=True, use_country_costs=True
)

print("OOS # trades (R3):", (0 if t3 is None else len(t3)))

# Overall stats
```

```
cap_oos = MAX_FLY_POSITIONS * DV01_LIMIT_FLY * CAPITAL_PER_DV01
stats_oos_r3 = compute_summary_stats(p3, t3, cap_oos)
display(pd.DataFrame([stats_oos_r3], index=["R3 (OOS, R2 + IRS filter)"]).round(3))

# Plot 1 - PnL
fig, ax = plt.subplots(figsize=(12, 4))
p3["pnl_net"].fillna(0.0).plot(ax=ax, linewidth=1.2, alpha=0.9)
ax.set_title("OOS — R3 Daily Net PnL ($)")
ax.set_ylabel("Daily net PnL ($)")
ax.grid(True, linestyle="--", alpha=0.35)
plt.tight_layout()
plt.show()

# Cum returns
fig, ax = plt.subplots(figsize=(12, 5))
_cum_simple_pct(p3, cap_oos, "pnl_net").plot(ax=ax, linewidth=2.2)
ax.set_title("OOS — R3 Cumulative Net Return (simple sum, % of capital)")
ax.set_ylabel("Cumulative return (% of capital)")
ax.grid(True, linestyle="--", alpha=0.35)
plt.tight_layout()
plt.show()

# Gross vs ex transaction costs
cum_gross = _cum_simple_pct(p3, cap_oos, "pnl_gross")
cum_net   = _cum_simple_pct(p3, cap_oos, "pnl_net")
cum_cost  = (p3["trade_cost"].fillna(0.0).cumsum() / cap_oos) * 100.0

fig, ax1 = plt.subplots(figsize=(12, 5))
cum_gross.plot(ax=ax1, linewidth=2, label="Gross (pre-cost)")
cum_net.plot(ax=ax1, linewidth=2, label="Net (post-cost)")
cum_cost.plot(ax=ax1, linewidth=1.8, linestyle="--", label="Cost drag (cumulative)")
ax1.set_title("OOS — R3: % cap + Active Flies")
ax1.set_ylabel("Cumulative % of capital")
ax1.grid(True, linestyle="--", alpha=0.35)

ax2 = ax1.twinx()
p3["num_active_positions"].fillna(0).plot(ax=ax2, linestyle=":", linewidth=1.8, alpha=0.9,
label="# active flies")
ax2.set_ylabel("# active flies")

h1, l1 = ax1.get_legend_handles_labels()
h2, l2 = ax2.get_legend_handles_labels()
ax1.legend(h1 + h2, l1 + l2, loc="upper left")
plt.tight_layout()
plt.show()
```

ok

4. Appendices - Other details

Appendix 1: Economic Hypothesis that is the bedrock of Economic Intuition

We evaluate the following Economic Intuitions in this project.

A rolling Z-score is then defined (Covered further in Signal Generation):

$$z_t = \frac{Fly_t - \mu_t}{\sigma_t}$$

Wherein the μ_t and σ_t is evaluated over the 126-day rolling window. With this, the following hypotheses are tested:

H1) Mean reversion in curvature - Wherein Extreme deviations in the Fly_t relative to recent history tends to mean-revert over horizons consistent with the entry and exit rules

H2) Regime-adaptive Normalisation improves the signal efficacy Rolling standardisation yields a more stable trade frequency and more robust signals than fixed long-horizon normalization across changing interest rate environments. Theoretically, the only country that long-term normalization may work is Japan, with perennially low rates until recently.

H3) Dual curve confirmation reducing false positives - Agreements between the government and IRS-curve signals (more details later), reduces trades driven by specific idiosyncratic instrument distortions, given this, entry quality should be enhanced, and churn is lowered at the cost of reduced frequencies.

ok

Appendix 2: Proposed Quadratic Transaction Costings:

$$C_t = c_{\text{fixed}} \cdot \mathbf{1}_{\Delta\text{pos}\neq 0} + c_1 \cdot |\Delta \text{DV01}_t^{\text{fly}}| + c_2 \cdot (|\Delta \text{DV01}_t^{\text{fly}}|)^2$$

For the linear term c1, it is corresponding to the position size, for the quadratic term c2, it stands for the market impact or slippage, the larger our position, the higher the impact and thus lead to more slippage. For c1, we set it based on bid-ask spread(eg. 0.1 bps for one leg), for the c2, we set it based on historical slippage (eg. 0.05 bps for one leg). However, we can also select a threshold, say, 100M, if our open position is larger than it, we should consider higher c2 for more precise estimation.

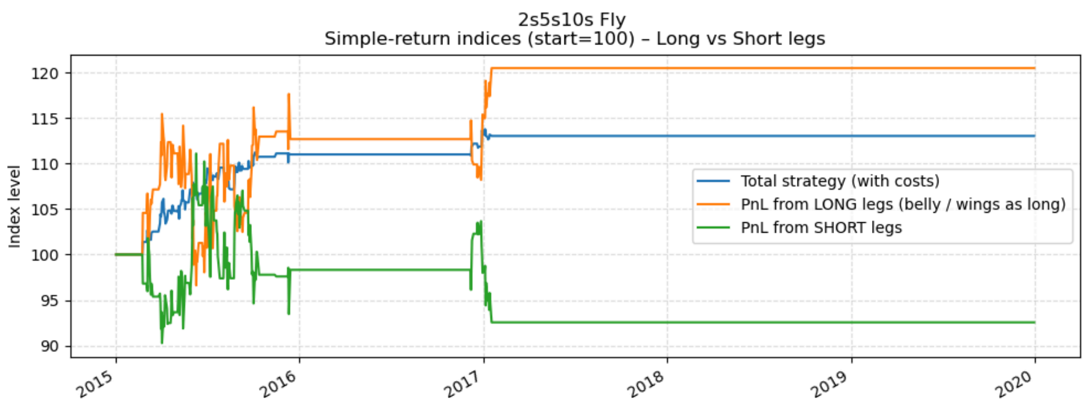
ok

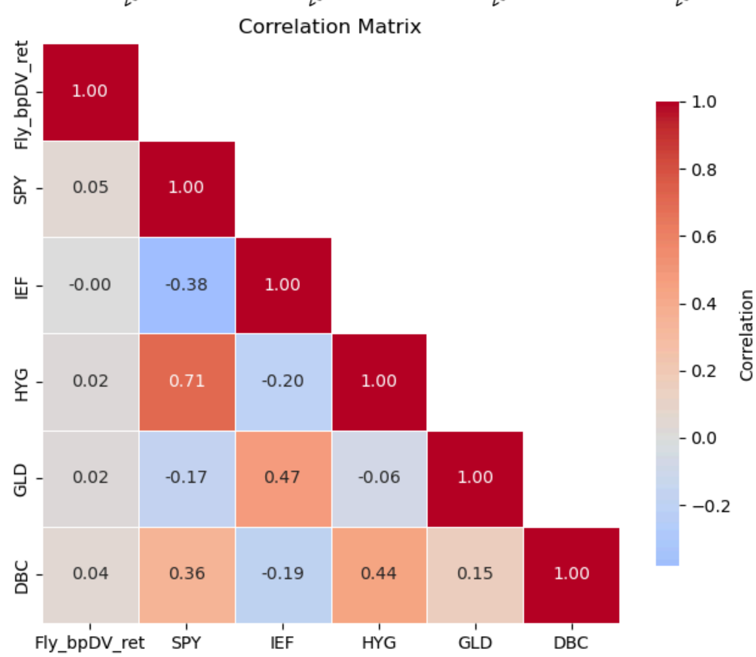
Appendix 3: Full-run costs:

```
COST_ROUNDTRIP_BP_COUNTRY = {
    "US": 0.12,   # UST fly via futures, very tight
    "DE": 0.12,   # Bund – Germany
    "UK": 0.18,   # Gilts a bit wider – wider than above, esp in recent markets – refer to marketwatch
    "JP": 0.08,   # JGBs – generally very very liquid
    "AU": 0.35,   # ACGBs – Aussies
    "CA": 0.25,   # Canadas – wider than US/DE/JP, generally not that liquid, but more liquid than AUS
    "IT": 0.50,   # BTPs widest – LOL
}
```

ok

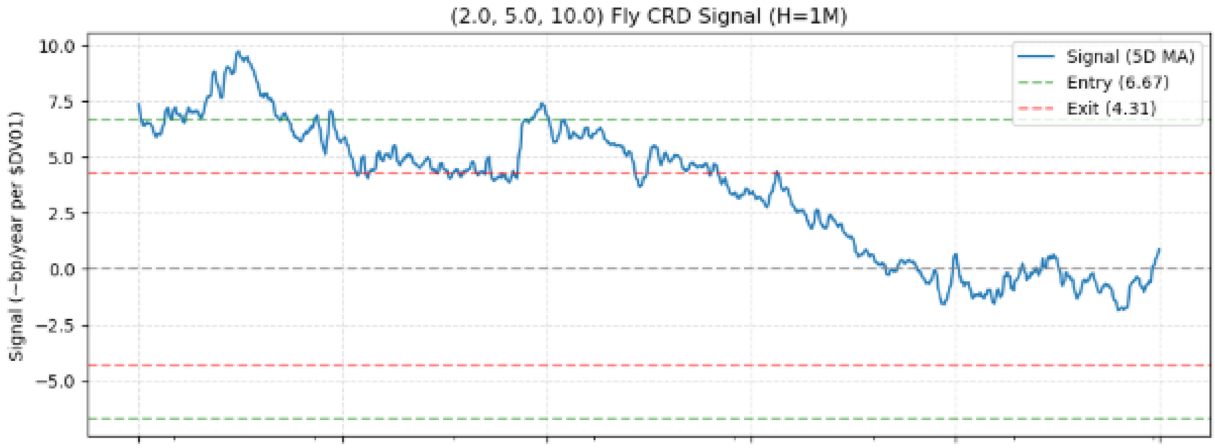
Appendix 4: Diagrams, and other information for first backtesting.





ok

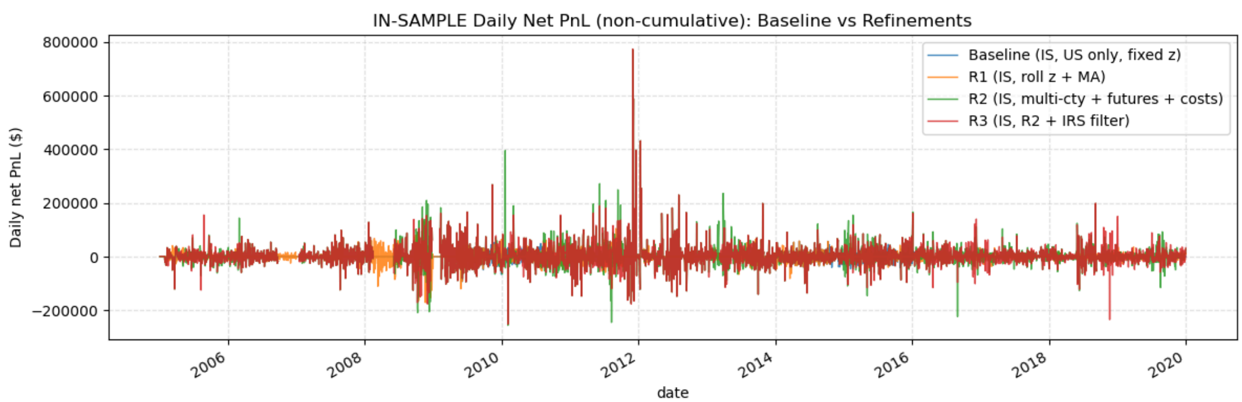
Appendix 5: Signal deprecating, and therefore trade not executing. Hence the inspiration for further refinements.



ok

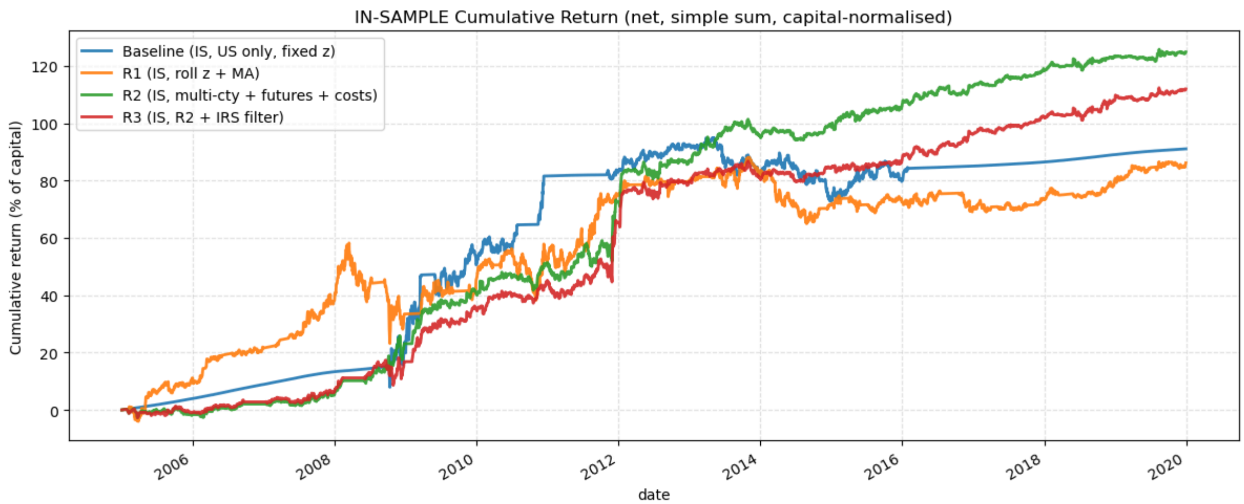
Appendix 6: Comps for PnL Plot:

Commentary: As we can see, the IRS, especially with the IRS confirmation filter, this basically minimises losses, and higher peaks.



ok

Appendix 7: Comparisons of the baseline strategy and the cumulative return



ok

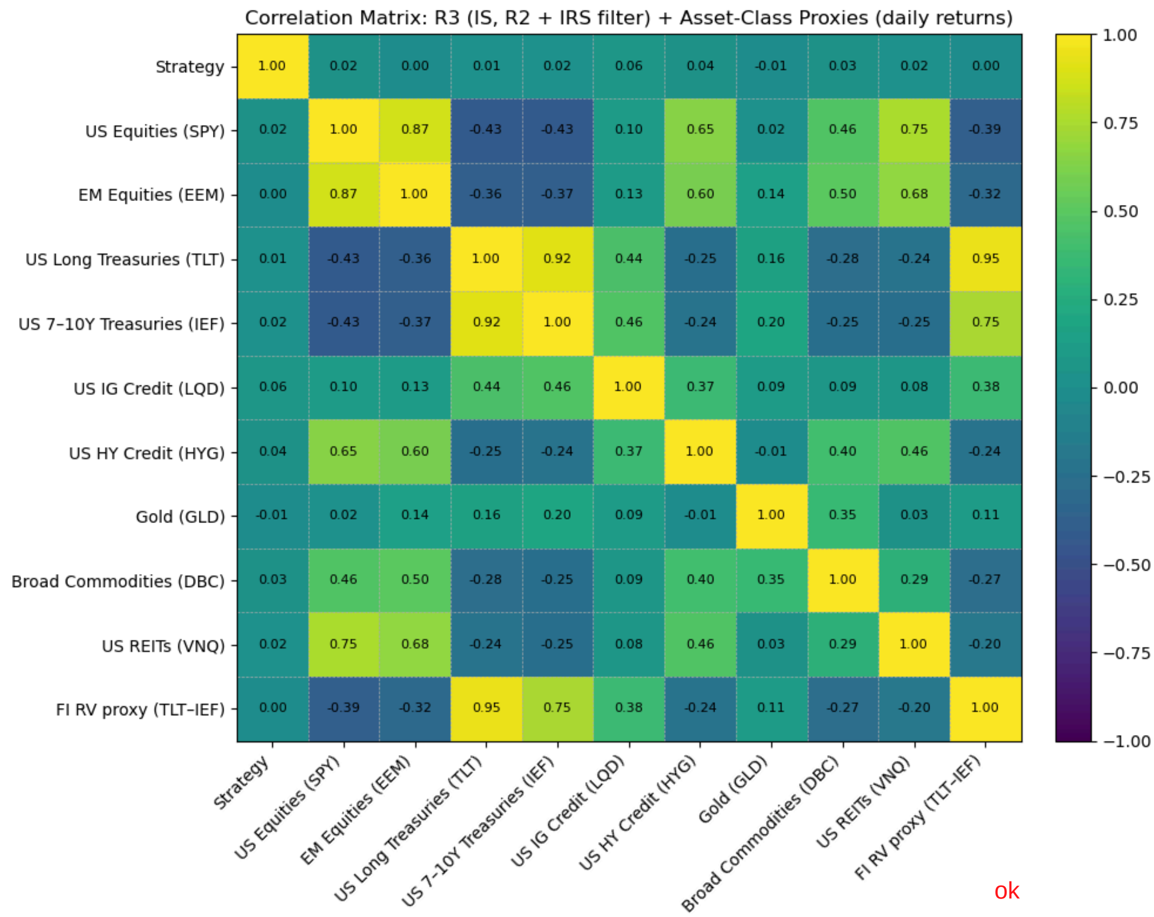
Appendix 8: IS Full Comps (in this case for baseline, we also ensure that spare risk capacity is in SOFR)

	Ann return net (%)	Ann vol net (%)	Sharpe net	Ann return gross (%)	Ann vol gross (%)	Sharpe gross	Max DD (net, %)	Calmar (net)	Sortino (net)	VaR 99% (daily, %cap)	...	VaR 95% (daily, %cap)	ES 95% (daily, %cap)	# round-trips	Hit rate (%)	Avg win ($)	Avg loss ($)	Avg hold (days)	Trades per year
Baseline (IS, US only, fixed z)	5.919	8.239	0.718	5.919	8.239	0.718	-20.569	0.288	0.659	-1.531	...	-0.722	-1.260	5.0	80.000	435533.079	-31786.670	306.600	0.325
R1 (IS, roll z + MA)	5.600	10.309	0.543	5.600	10.309	0.543	-30.308	0.185	0.650	-1.849	...	-0.973	-1.563	47.0	74.468	98432.202	-118230.559	64.128	3.056
R2 (IS, multi-cty + futures + costs)	8.515	6.942	1.227	9.484	6.944	1.366	-8.147	1.045	1.939	-1.069	...	-0.510	-0.856	234.0	73.077	110132.318	-62415.764	61.992	15.963
R3 (IS, R2 + IRS filter)	7.657	6.632	1.155	8.569	6.633	1.292	-8.498	0.901	1.823	-1.068	...	-0.499	-0.839	217.0	72.811	108860.282	-64968.032	65.566	14.848

ok

i dont see asset class correlations or pnl attributions

Appendix 9: Comps for PnL Plot (IS):



ok

Appendix 10: OOS Summary Stats table

```
Requested OOS window: 2020-01-01 → END
Aligned FULL range: 2005-01-03 → 2025-10-01 | N = 5365
Aligned OOS  range : 2020-01-02 → 2025-10-01 | N = 1489
OOS # trades (R3): 71
```

	Ann return net (%)	Ann vol net (%)	Sharpe net	Ann return gross (%)	Ann vol gross (%)	Sharpe gross	Max DD (net, %)	Calmar (net)	Sortino (net)	VaR 99% (daily, %cap)	...	VaR 95% (daily, %cap)	ES 95% (daily, %cap)	# round-trips	Hit rate (%)	Avg win ($)	Avg loss ($)	Avg hold (days)	Trades per year	Total tx cost ($)	Avg daily tx cost ($)
R3 (OOS, R2 + IRS filter)	3.309	4.931	0.671	4.198	4.915	0.854	-4.859	0.681	0.971	-0.937	...	-0.453	-0.707	71.0	71.831	79467.782	-71003.611	79.239	12.016	525112.5	352.661

ok

