

Forecast Evaluation and Combinations

Module 7 · Scoring, testing, and averaging forecasts

Gary Cornwall

Econ 6376 · The George Washington University

Module 6 recap: a forecast is a point, an interval and a density

Point	$\tilde{y}_{t+h t} = \mathbb{E}[y_{t+h} \mathcal{F}_t]$	the headline number
Interval	$\tilde{y}_{t+h t} \pm c_h$	with a stated coverage
Density	$f(y_{t+h} \mathcal{F}_t)$	the whole distribution; the fan chart draws it

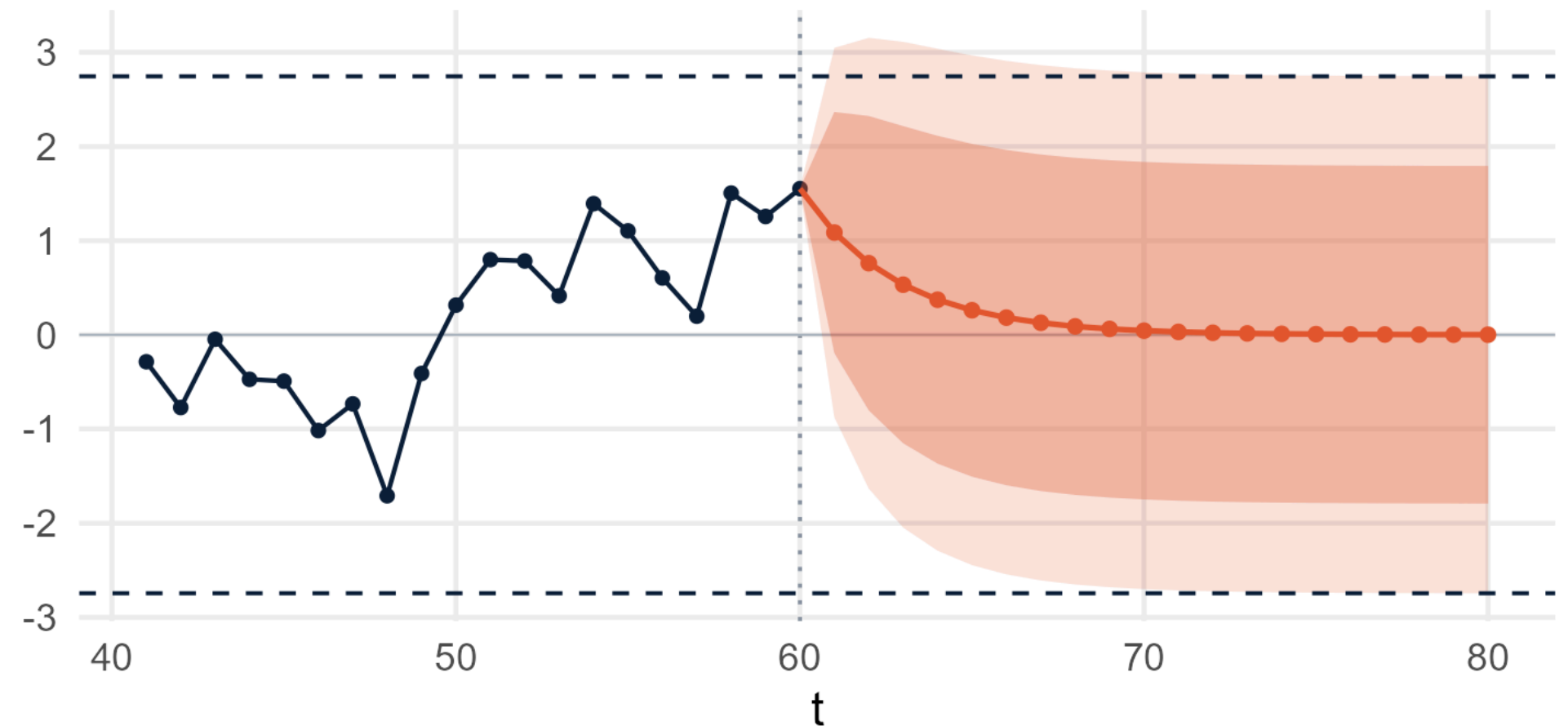
Uncertainty grows with the horizon.

$$\text{Var}(e_{t+h|t}) = \sigma^2 \sum_{i=0}^{h-1} \psi_i^2$$

$h = 1: \sigma^2$. Stationary: $\rightarrow \gamma_0$. Unit root: keeps growing.

AR(1), $\varphi = 0.7$, $\sigma^2 = 1$: the fan

80% and 95% bands; dashed: $\pm 1.96\sqrt{\gamma_0}$, the unconditional limit



Module 6 recap: the loss picks the target

Under MSE, the best point forecast is the **conditional mean**

Under MAE **conditional median**

Under asymmetric linear loss with weight α **the α -quantile**

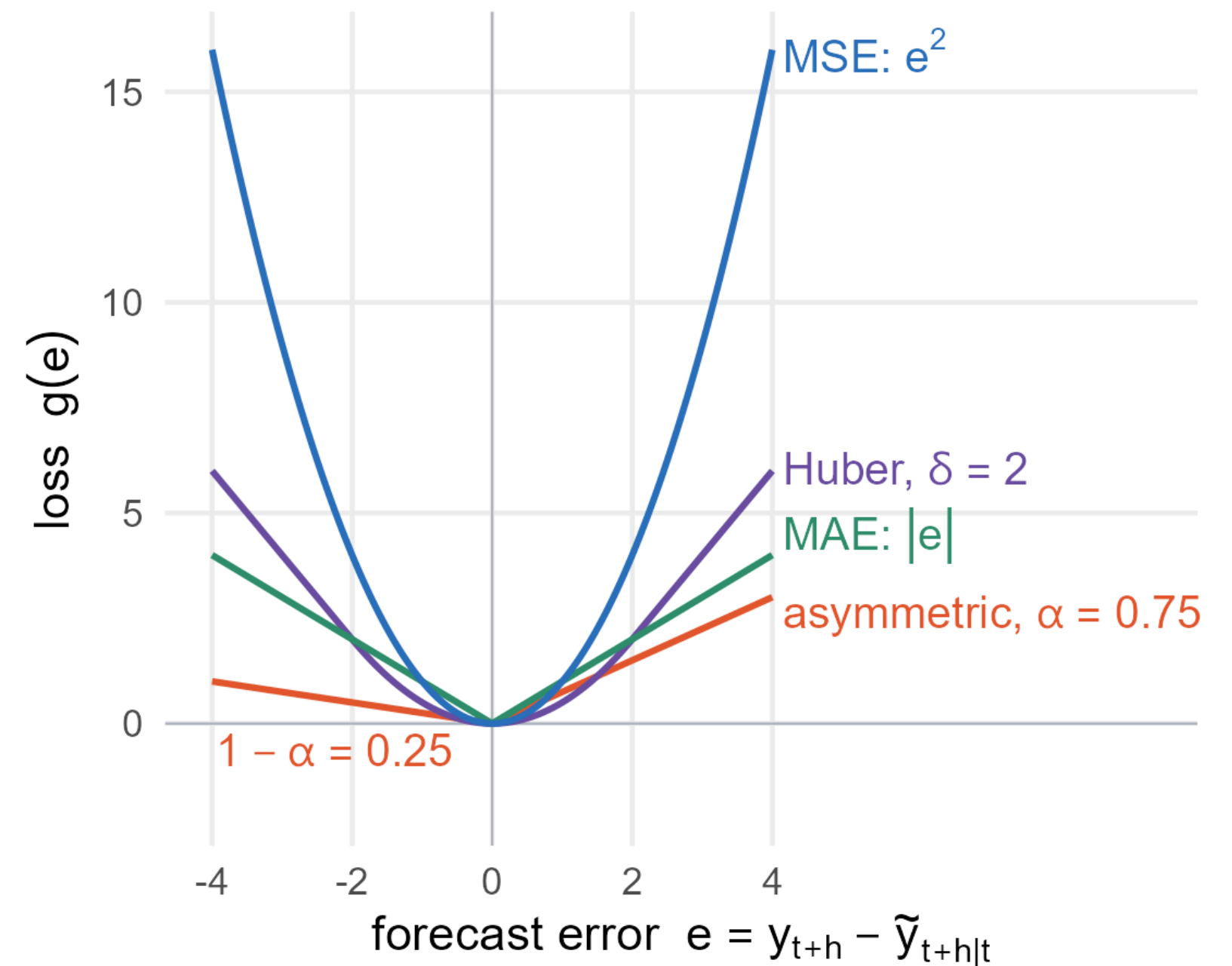
Huber and MAPE are losses too; this course scores with them rather than deriving a target. MAPE is scale-free but undefined at $y_t = 0$.

Choose the loss before you see the results; compare within a loss, never between: same loss, same y , same test observations N .

Today's DM test takes a loss as an input. Which one, and who chooses?

Four loss functions, one axis

zero at a perfect forecast, larger as the miss grows



Module 6 recap: train first, then test, in time order

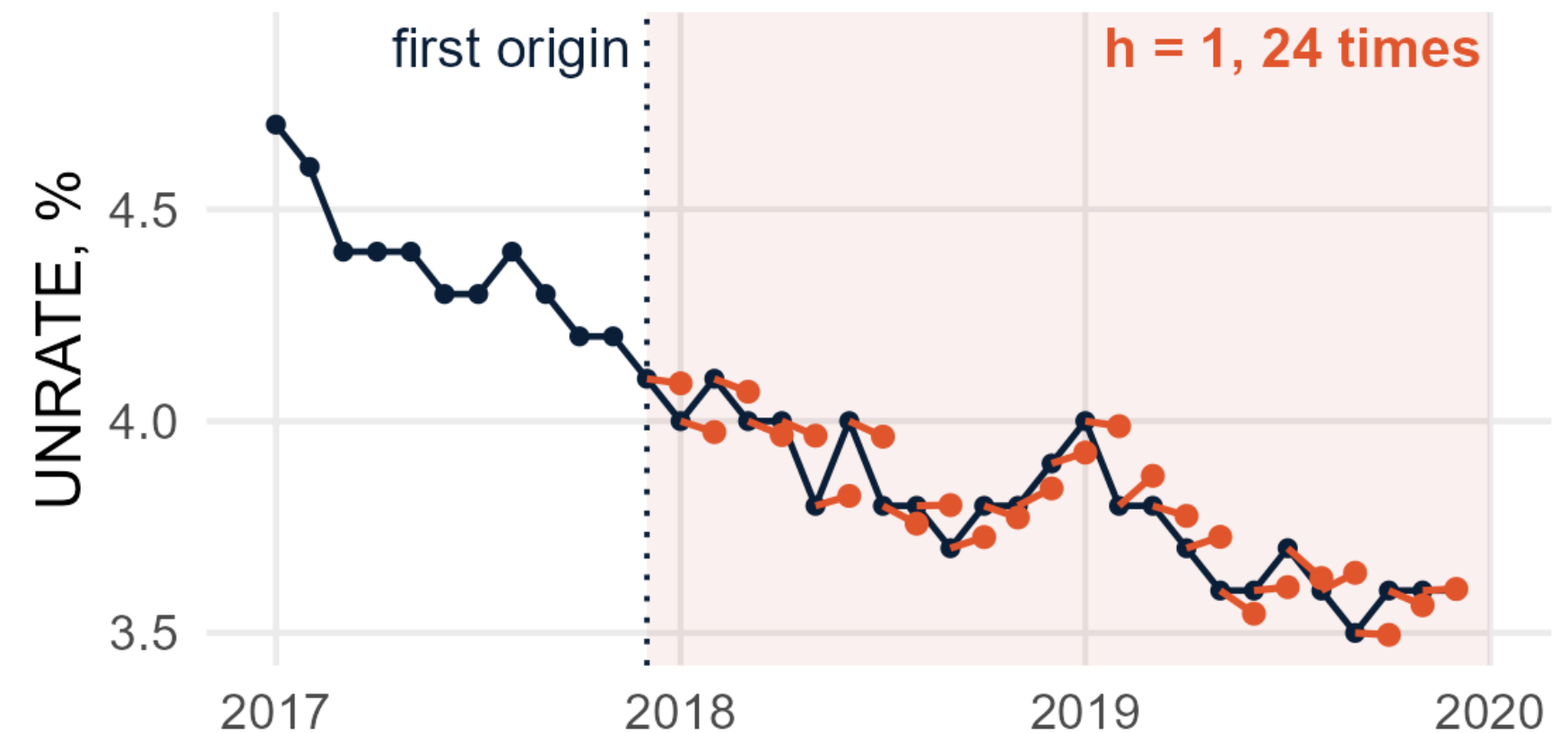
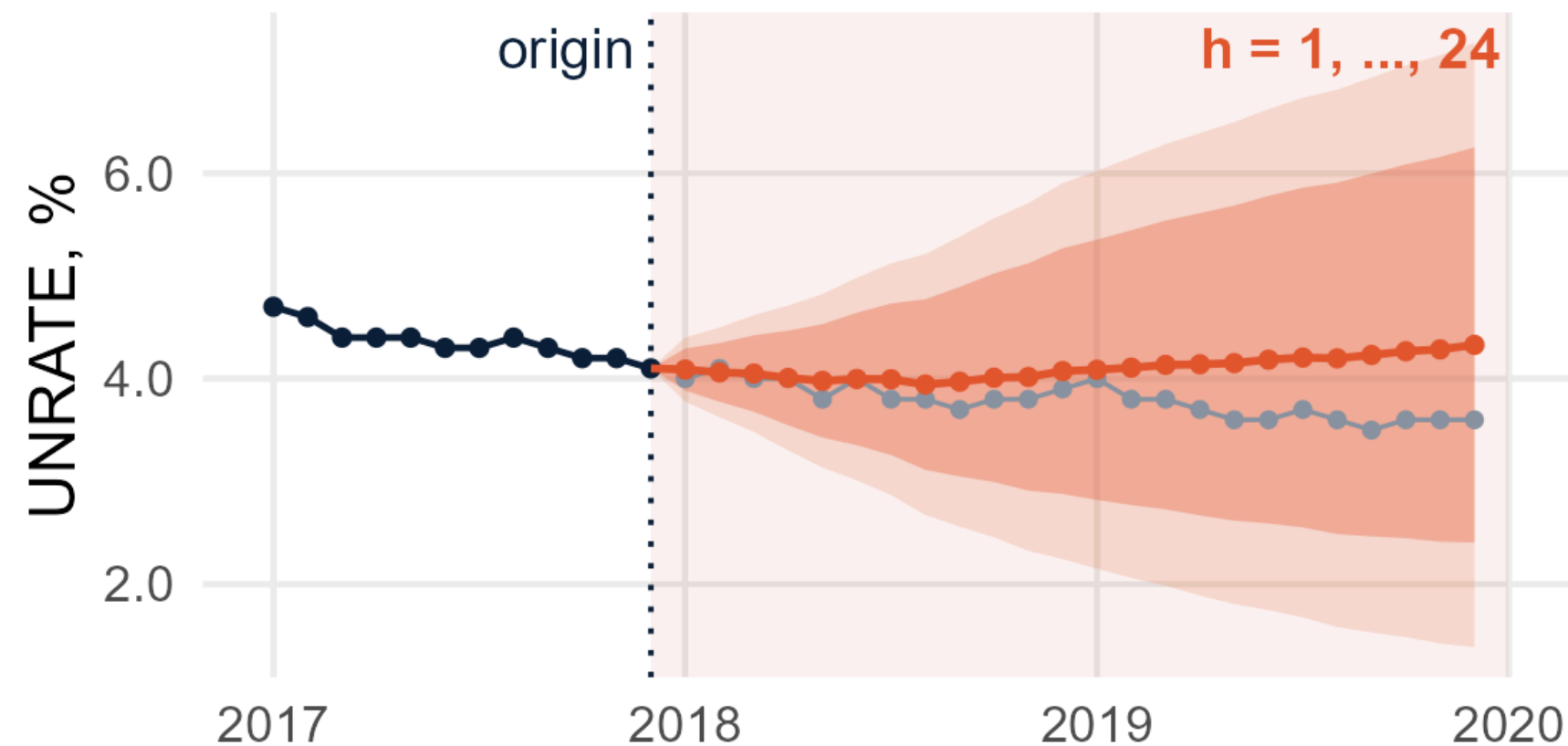
Train: estimate here, once
UNRATE 1960-01 ... 2017-12

Test
24
months

1960

origin: December 2017 → 2019

Never a random split. Shuffling leaks the future into training: Biff with the sports almanac.



Module 6: one origin, many horizons. One fit, one path, $h = 1, \dots, 24$.

Module 7: many origins, one horizon. Fixed coefficients, rolling data.

Today: many origins, one horizon. Same parameters, the information set rolls.

Why Choose Only One?

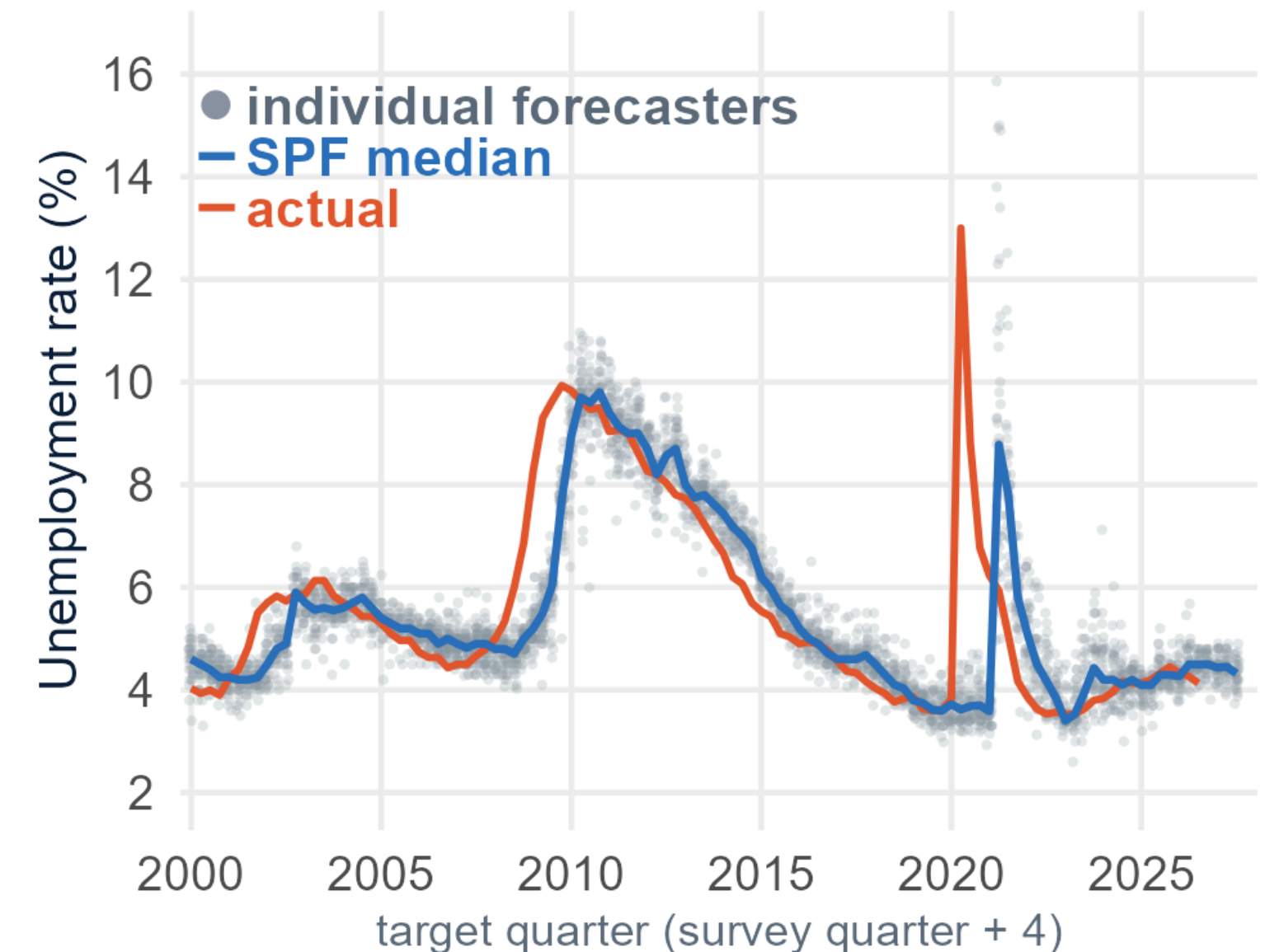
Open Release Date Radar

Debrief: the crowd average is a forecast combination

- Your guess of the release date is a forecast based on a mental model.
- Each of your models has partially overlapping information sets: you can exploit this!
- Each of you made a forecast $\tilde{y}^{(m)}$ of the release year, $m = 1, \dots, M$. The crowd mean is

$$\tilde{y}^{(c)} = \frac{1}{M} \sum_{m=1}^M \tilde{y}^{(m)}$$

the **simple average**: every weight $w_m = 1/M$, nothing estimated.



Source: Philadelphia Fed SPF (individual responses, 4-quarter-ahead); BLS via FRED.

The SPF's published forecast is the median across ~37 professional forecasters: forecast combination as standard practice.

Where we are: nothing new on the mixing board

$$y_t = \alpha + \delta t + \sum_{j=1}^p \phi_j y_{t-j} + \sum_{l=1}^q \theta_l \epsilon_{t-l} + \epsilon_t$$

Every slider is on, seasonal block included. Module 6 projected one fitted equation forward; Module 7 puts several on the table and referees between them.

Term	L4	L5	L6	L7
$\alpha, \delta t$	On	On	On	On
$\phi_j y_{t-j}$ (all p)	On	On	On	On
$\theta_l \epsilon_{t-l}$ (all q)	On	On	On	On
Seasonal block $(P, D, Q)_s$	Off	On	On	On
What we do with it	Select	Diagnose	Forecast	Evaluate and combine M forecasts

Today’s arc, three questions:

1. How do I score several forecasts?

U_2 , MASE
2. Is the difference real?

Diebold-Mariano
3. Should I even pick one?

combinations

The PS3 design: many origins, one horizon

Train: estimate every model here, once

$y_1, \dots, y_T, T = N_{\text{total}} - 24$

Weights window

$T + 1, \dots, T + 12$

Holdout

$T + 13, \dots, T + 24$

UNRATE 1960-01

train to 2017-12 · weights 2018 · holdout 2019

Parameters are estimated once, on $y_1, \dots, y_T$ never again

The information set rolls. $\tilde{y}_{t|t-1}$ uses actuals up to $h = 1$ throughout y_{t-1}

Weights window, $T_w = 12$ where combination weights are estimated dates

Holdout, $N = 12$ dates where everything is scored

One fit per model; at each date, the training coefficients meet the newest data:

```
1 T_origin <- length(unrate) - 24 # Dec 2017
2 train_un <- upto(unrate, T_origin)
3 fit1 <- Arima(train_un, order = c(1, 1, 2),
4   seasonal = list(order = c(1, 0, 1),
5     period = 12))
6 # one-step forecasts, coefficients frozen:
7 fitted(Arima(upto(unrate, T_origin + 24),
8   model = fit1))
```

Subset a `ts` with `window()` (here `upto()`), never `unrate[1:T_origin]`: square brackets drop the monthly frequency.

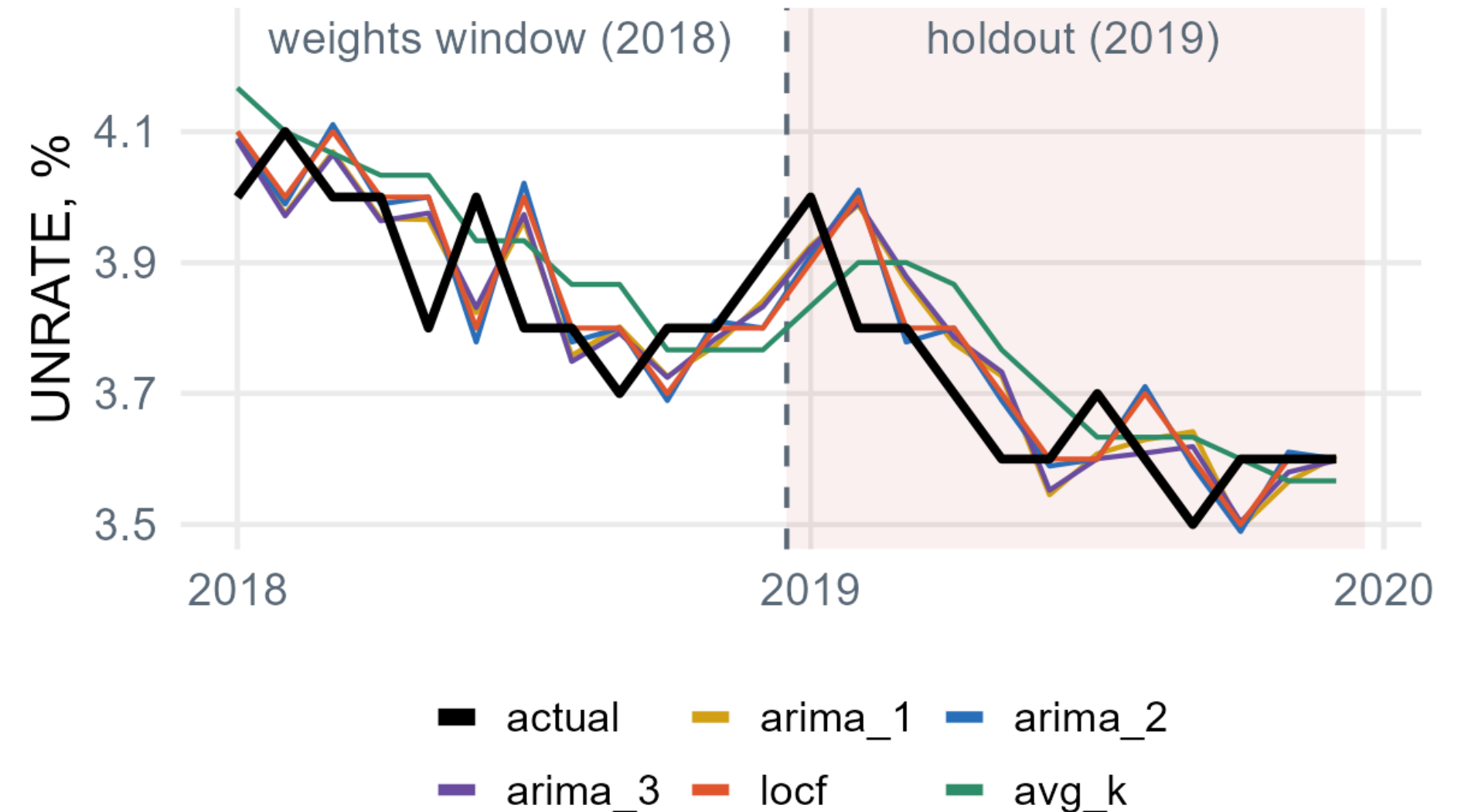
Six forecast tracks

Track	On UNRATE	In the pool as
<code>arima_1</code>	ARIMA(1,1,2)(1,0,1) ₁₂	the Module 5/6 model
<code>arima_2</code>	ARIMA(1,1,0)	a simpler alternative
<code>arima_3</code>	<code>auto.arima()</code> on train	the automatic pick
<code>locf</code>	$k = 1$	the naive benchmark
<code>avg_k</code>	$k = 3$	smooths sampling noise
<code>meanf</code>	$k = t - 1$	the long-run mean

$$\tilde{y}_{t|t-1} = \frac{1}{k} \sum_{j=1}^k y_{t-j}$$

k -period average forecasts, never “moving averages”: in this course that is the MA(q) block.

Error: $e_{m,t} = y_t - \tilde{y}_{t|t-1}^{(m)}$. Out of sample and observable: not ϵ_t , not a residual.



UNRATE 2018-2019, actuals in black. `meanf` is off the chart, near 6.0%.

Why $k = 3$ for UNRATE, and not $k = 24$?

The course simulation: one track is genuinely better

$$(1 - 0.7L) \left(1 - 0.5L^{12}\right) y_t = \alpha + \epsilon_t, \quad \epsilon_t \sim N(0, 1), \quad \alpha = 5(1 - 0.7)(1 - 0.5) = 0.75, \quad \mu = 5$$

Track	In the simulation	
arima_1	ARIMA(1,0,0)(1,0,0) ₁₂ , mean	true specification
arima_2	ARIMA(1,0,0), mean	misses the season
arima_3	ARIMA(0,1,1)	a needless difference
locf , avg_k , meanf	$k = 1, 3, t - 1$	as on UNRATE

Each ARIMA estimated once, on 240 training months; then 12 weights-window and 12 holdout dates.

Built so the optimal weights are far from $1/M$: a method that finds [arima_1](#) should beat the simple average.

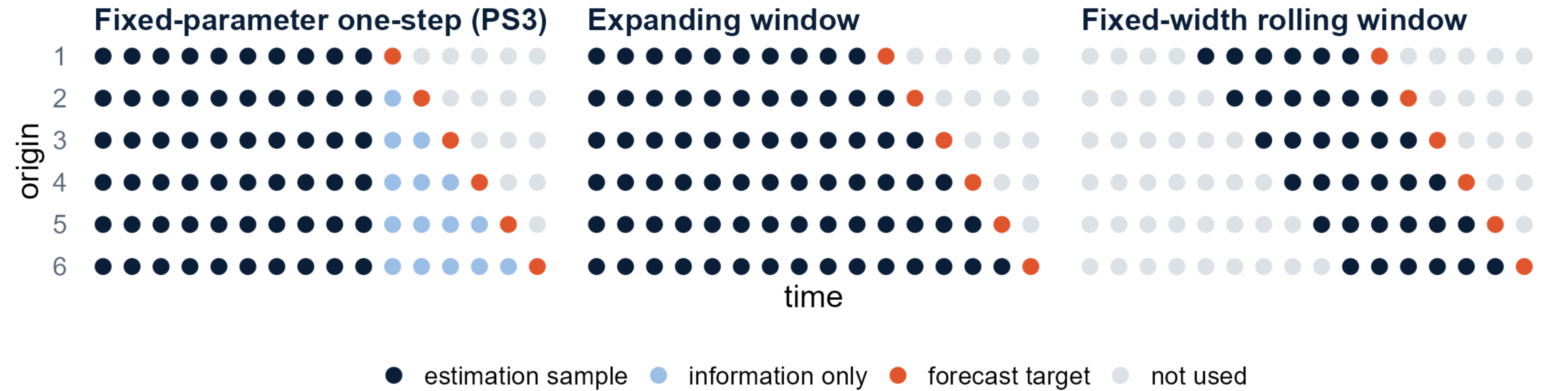


The course realization (seed 6376): the end of training, then the weights window and the holdout.

Three ways to run an evaluation

Scheme	Parameters	Estimation window at origin t	Fits per model
Fixed-parameter one-step (PS3)	estimated once	$1:T$; information rolls to $t - 1$	1
Expanding window	re-estimated at every origin	$1:(t - 1)$	one per origin
Fixed-width rolling window	re-estimated at every origin	$(t - W):(t - 1)$	one per origin

Module 7 builds only the first row. The other two re-estimate at every origin.



Re-estimation buys robustness to change at the cost of noisier parameters. Fixed-width windows forget old regimes on purpose. Which scheme would you use if you suspected a break in 2008?

U_2 and MASE answer different questions

RMSE and MAE are in units of y ; MAPE is a percentage. Is 0.1 points good? Two relative scores, two different questions.

U_2 : a race. Did model m beat **locf** on these holdout dates?

$$U_2^{(m)} = \frac{\text{RMSE}^{(m)}}{\text{RMSE}^{(\text{locf})}}$$

Both RMSEs over the same holdout dates, $t = T + 13, \dots, T + 24$.

$U_2 < 1$ beat **locf** here

$U_2 > 1$ lost to it here

MASE (mean absolute scaled error): a yardstick. Model m 's holdout miss, relative to the naive's typical one-step miss in training.

$$\text{MASE}^{(m)} = \frac{\text{MAE}_{\text{holdout}}^{(m)}}{\text{MAE}_{\text{train}}^{(\text{locf})}}$$

$$\text{MAE}_{\text{holdout}}^{(m)} = \frac{1}{N} \sum_{t=T+13}^{T+24} |y_t - \tilde{y}_{t|t-1}^{(m)}|$$

$$\text{MAE}_{\text{train}}^{(\text{locf})} = \frac{1}{T-1} \sum_{t=2}^T |y_t - \tilde{y}_{t|t-1}^{(\text{locf})}|$$

with $\tilde{y}_{t|t-1}^{(\text{locf})} = y_{t-1}$. Seasonal variant: y_{t-s} replaces y_{t-1} , over $t = s + 1, \dots, T$, divided by $T - s$.

MASE = 0.6. Did your model beat the naive in the holdout?

You cannot tell: the naive's holdout errors never enter MASE. MASE < 1 does not mean you beat the naive; $U_2 < 1$ does.

Helpers `theil_u2(actual, forecast, benchmark)` and `mase(actual, forecast, train, m = 1)`: the helper's `m` is the seasonal lag s . `accuracy()` differs: a seasonal MASE scale by default, and a different "Theil's U".

The scorecard on UNRATE: small errors, no win

```

1 metrics_table <- function(actual, Fmat, train) {
2   bench <- Fmat[, "locf"]
3   t(apply(Fmat, 2, function(f) c(
4     RMSE = rmse(actual, f),
5     MAE   = mae(actual, f),
6     MAPE  = mape(actual, f),
7     MASE  = mase(actual, f, train),
8     U2    = theil_u2(actual, f, bench))))
9 }
10 metrics_table(y_h, F_h, train_un) # 2019

```

Every credible track has MASE below 1. Did any of them beat locf?

Track	RMSE	MAE	MAPE	MASE	U_2
arima_1	0.097	0.083	2.258	0.642	1.011
arima_2	0.097	0.079	2.127	0.607	1.016
arima_3	0.096	0.080	2.168	0.618	0.999
locf	0.096	0.075	2.030	0.580	1.000
avg_k	0.108	0.092	2.478	0.709	1.124
meanf	2.315	2.312	63.103	17.875	24.182

No meaningful win: `arima_3`'s 0.999 is a tie with `locf`. MASE 0.58 to 0.71, but U_2 0.999 to 1.124 (`locf` is 1 by construction). 2019 was easy, for everyone, the naive included.

| `meanf`: an $I(1)$ series has no μ to converge to (Module 2).

Scoring is not testing: the taste test

A metrics table is a ranking, not a verdict. Is the gap bigger than what the noise in twelve forecast errors would produce by chance?

Two coffee shops, one sip each	no verdict
Twelve mornings, one sip from each	a tally of the difference
Is the average difference zero?	a paired <i>t</i> -test

Shop A won one sip. How many paired sips before you would bet on A?

The Diebold-Mariano test is a *t*-test on the loss differentials.

Diebold-Mariano: a t -test on loss differentials

$$d_t = g(e_{1,t}) - g(e_{2,t}), \quad H_0: \mathbb{E}[d_t] = 0$$

$$\bar{d} = \frac{1}{N} \sum_t d_t, \quad \text{DM} = \frac{\bar{d}}{\widehat{\sigma}_{\bar{d}}} \stackrel{a}{\sim} N(0, 1)$$

$\widehat{\sigma}_{\bar{d}}$: a standard error that **allows for serial correlation in d_t** . At $h = 1$, `dm.test()` uses lag 0 only, $\widehat{\sigma}_d/\sqrt{N}$ (after the small-sample correction): it **assumes d_t is serially uncorrelated**, a condition on d_t , not merely on the errors.

Out-of-sample errors only never `residuals(fit)`

Same dates, same loss both on the same N

The sign: $\text{DM} < 0$ forecast 1 had the smaller loss

On the course simulation, the correctly specified model against the naive:

```
1 es_h <- ys_h - Fs_h           # N x M holdout errors
2 dm.test(es_h[, "arima_1"], es_h[, "locf"],
3         h = 1, power = 2)    # power 2: squared
4 d_sim <- es_h[, "arima_1"]^2 - es_h[, "locf"]^2
5 t.test(d_sim)$statistic
```

`dm.test()` **DM = -0.90, p = 0.390**

`t.test(d_sim)` **t = -0.90**

The same statistic: `dm.test()` applies the Harvey-Leybourne-Newbold correction and a t_{N-1} reference. Named, not derived.

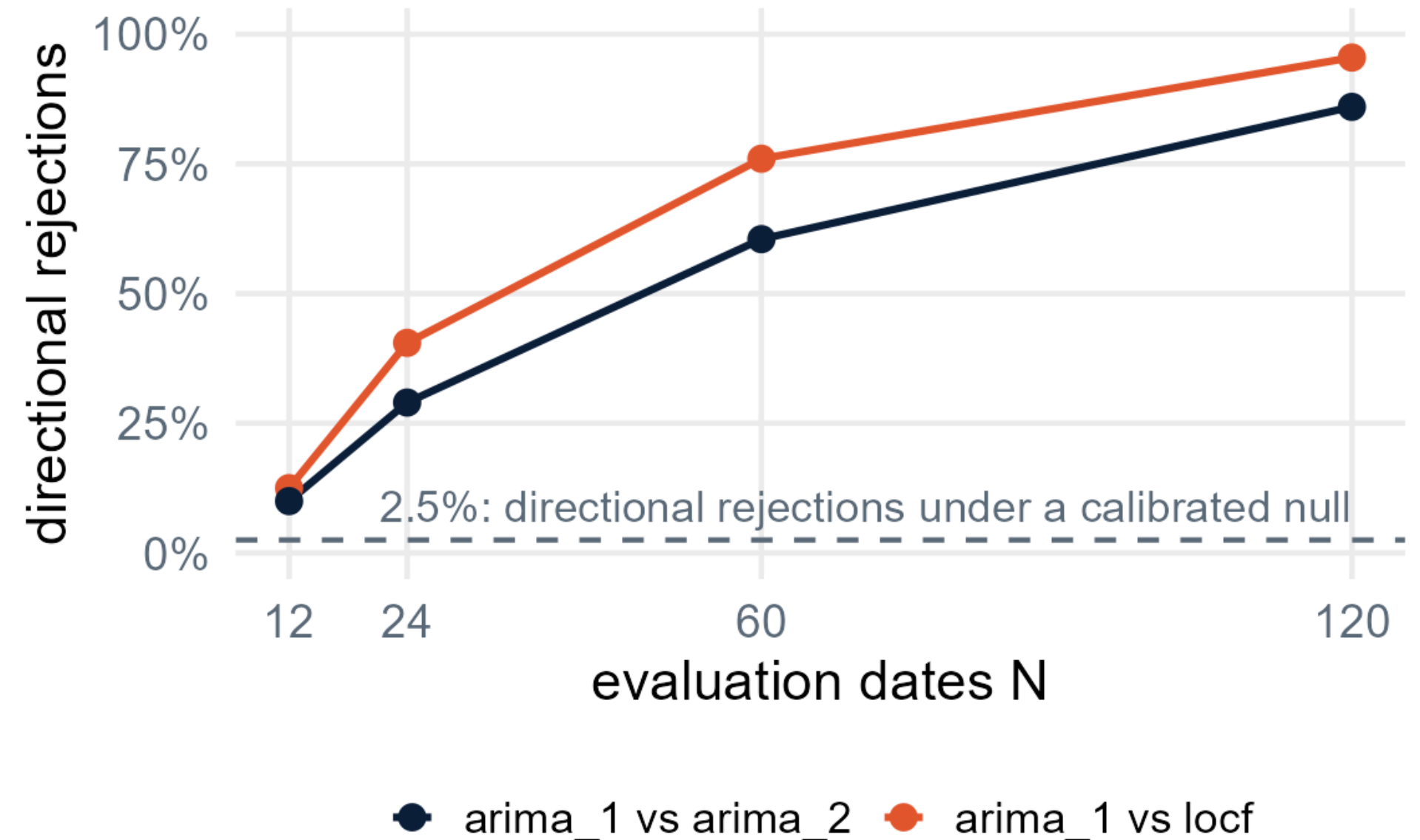
Shrink the sample: DM power from $N = 12$ to 120

`arima_1`, correctly specified and estimated, has an expected-loss advantage over `locf` and `arima_2` on average. How often does DM detect it?

$R = 200$ replications: a fresh series, the ARIMAs estimated once on 240 months, one-step tracks over the next 120, DM on the first N dates.

At $N = 12$, what share of replications reject in favour of `arima_1`: about 90%, 50%, or 20%?

Rejecting for <code>arima_1</code>	N=12	N=24	N=60	N=120
<code>arima_1</code> vs <code>locf</code>	12.5%	40.5%	76.0%	95.5%
<code>arima_1</code> vs <code>arima_2</code>	10.0%	29.0%	60.5%	86.0%



Mainly power: DM rejects when the gap is large relative to the variability of d_t . “Failed to reject” never means “equally good”.

Caveat: these are rates of the $h = 1$ test, calibrated only if d_t is serially uncorrelated. Under a calibrated null, 2.5% would reject in this direction (dashed line).

Module 6's question, answered

Module 6: on 2018-2019 the naive random walk beat the Module 5 SARIMA by a few hundredths. Does that margin mean anything?

Asked in the PS3 one-step design, on the 2019 holdout:

```
1 e_h <- y_h - F_h           # N x M holdout errors
2 dm.test(e_h[, "arima_1"], e_h[, "locf"],
3         h = 1, power = 2)
```

arima_1 vs locf

DM = 0.13, p = 0.90

Bartlett variance, $h = 2:4$

p = 0.88 to 0.89

No: the test does not reject on these twelve months.

What it does not claim.

It does not test Module 6's 24-step path. Errors from one origin share the early shocks and their spread grows with h : not the sequence DM assumes.

It does not say the two are equally good. Twelve dates rarely separate even a correctly specified estimated model from a misspecified one.

It is not assumption-free: $h = 1$ assumes d_t is serially uncorrelated. A variance that allows for serial correlation gives the same verdict (a low-power check).

If you can't pick a winner, why pick?

On **UNRATE** the five credible tracks are descriptively close, holdout RMSE 0.096 to 0.108, and the one tested comparison, **arima_1** vs **locf**, does not reject. Picking the winner of a twelve-month race is mostly picking noise.

Each forecast carries some information, and the loser may carry information the winner lacks.

The portfolio analogy. Forecasts are assets; forecast errors are their risks. Imperfectly correlated errors diversify, as imperfectly correlated returns do.

Bates and Granger (1969)	weight by accuracy
--------------------------	--------------------

Granger and Ramanathan (1984)	weights as a regression
-------------------------------	-------------------------

Stock and Watson (2004)	name the puzzle
-------------------------	-----------------

Did the crowd mean beat most of you? Why might it?

A combined forecast is a weighted sum of forecasts

$$\tilde{y}_{t | t - 1}^{(c)} = w_0 + \sum_{m = 1}^M w_m \tilde{y}_{t | t - 1}^{(m)} = w_0 + \mathbf{w}'\tilde{\mathbf{y}}_t$$

$\tilde{\mathbf{y}}_t$	the $M \times 1$ vector of the M forecasts of y_t
$w_m, \mathbf{w}$	the weight on forecaster m ; the $M \times 1$ weight vector
w_0	an intercept: zero for every method but one
$e_{c,t} = y_t - \tilde{y}_{t t - 1}^{(c)}$	the combined error; c is a label, not an index

Hats, stars and nothing , the DGP-versus-estimation rule applied to weights:	
w	generic, or a fixed rule ($1/M$)
$\widehat{w}$	estimated on the weights window
w^*	the population optimum

Weights are always w , never α or β (taken). Enders writes $\alpha_0, \dots, \alpha_n$.

Two forecasts: the combined error variance

Weights w and $1 - w$ sum to one, so the combined error is the same mix of the two errors:

$$e_{c,t} = w e_{1,t} + (1 - w) e_{2,t}$$

$$\begin{aligned} \text{Var}(e_{c,t}) &= w^2 \text{Var}(e_{1,t}) + (1 - w)^2 \text{Var}(e_{2,t}) \\ &\quad + 2w(1 - w) \text{Cov}(e_{1,t}, e_{2,t}) \end{aligned}$$

Minimise over w (Enders 2.73):

$$w^* = \frac{\text{Var}(e_{2,t}) - \text{Cov}(e_{1,t}, e_{2,t})}{\text{Var}(e_{1,t}) + \text{Var}(e_{2,t}) - 2 \text{Cov}(e_{1,t}, e_{2,t})}$$

With zero covariance, the **inverse-variance weight**:

$$w^* = \frac{1/\text{Var}(e_{1,t})}{1/\text{Var}(e_{1,t}) + 1/\text{Var}(e_{2,t})}$$

Error correlation (two indices: not the lag- k ρ_k):

$$\rho_{1,2} = \frac{\text{Cov}(e_{1,t}, e_{2,t})}{\sqrt{\text{Var}(e_{1,t}) \text{Var}(e_{2,t})}}$$

Variance ratio $\text{Var}(e_{2,t})/\text{Var}(e_{1,t}) \geq 1$: forecast 1 is the better one.

Diversification: equal variances V , zero covariance: $w^* = 0.5$ and $\text{Var}(e_{c,t}) = 0.5V$. Averaging halves the error variance.

Correlation eats the gain. Among weights in $[0, 1]$, the best mix beats forecast 1 alone until $\text{Cov}(e_{1,t}, e_{2,t}) = \text{Var}(e_{1,t})$, that is, $\rho_{1,2} = \sqrt{\text{Var}(e_{1,t})/\text{Var}(e_{2,t})}$.

Past that threshold, $w^* > 1$: the formula **shorts** the worse forecast. Unconstrained regression weights do exactly this (Method 3).

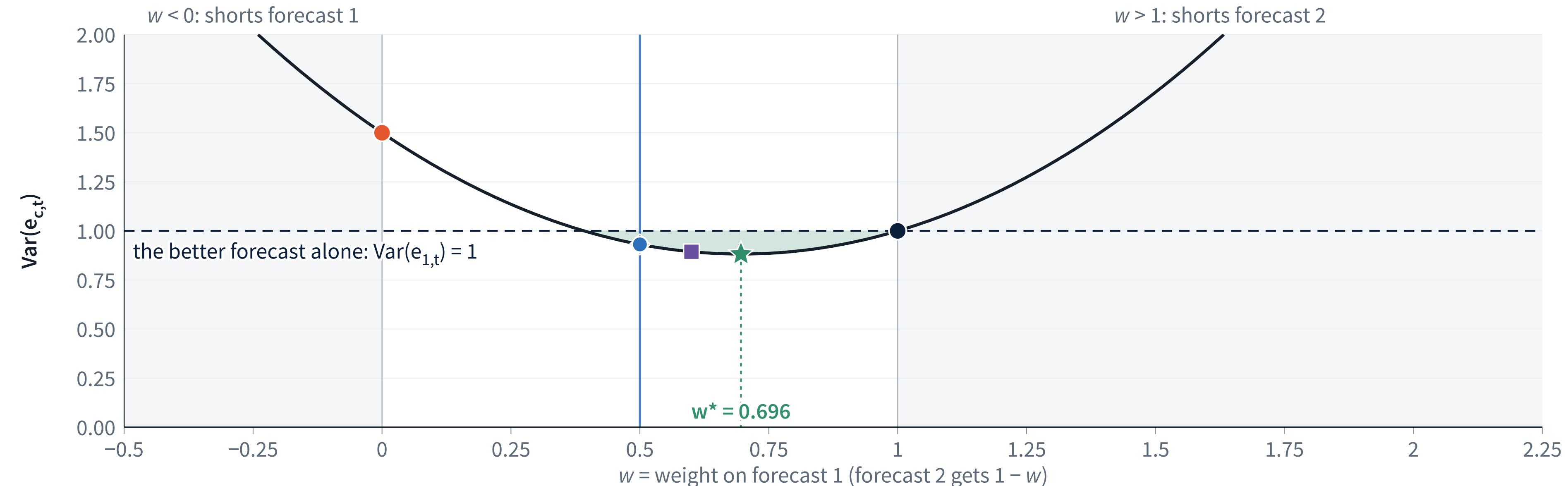
Drag the weight, the correlation, the variance ratio

$w =$ **0.50** $\rho_{1,2} =$ **0.50** $\text{Var}(e_{2,t}) / \text{Var}(e_{1,t}) =$ **1.50**

"/>

diversification pays $w^* = 0.696$ mixes both forecasts; the best mix cuts $\text{Var}(e_{c,t})$ 11.8% below the better forecast alone. ($\text{Cov}(e_{1,t}, e_{2,t}) = 0.612 < \text{Var}(e_{1,t}) = 1$)

$\text{Var}(e_{c,t})$ at $w = 0.50$: **0.931** (−6.9% vs better) ⓘ · $w^* = \mathbf{0.696}$ ⓘ · at w^* : **0.882** (−11.8% vs better) ⓘ · threshold $\rho_{1,2} = 0.816$ ⓘ



● forecast 1 alone ($w = 1$) ⓘ ● forecast 2 alone ($w = 0$) ⓘ ◆ simple average ($w = \frac{1}{2}$) ⓘ ■ inverse-variance ⓘ ★ w^* (Bates-Granger) ⓘ ● your w ⓘ
 --- better forecast ⓘ ■ beats it ⓘ

Method 1: the simple average

$$w_0 = 0, \quad w_m = \frac{1}{M}$$

Nothing is estimated, so the weights carry no hat. That is the whole point: the zero-estimation-error benchmark that every other method has to beat.

It is the radar's crowd mean: every forecaster gets one vote, $1/M$ of the result.

Every method fits on the weights window ($T_w = 12$ actuals and the $T_w \times M$ forecasts of them) and is then applied, unchanged, to the holdout forecasts.

```
1 comb_simple <- function(y, F) {
2   M <- ncol(F)    # y unused: common interface
3   list(intercept = 0,
4         weights = setNames(rep(1 / M, M),
5                             colnames(F)))
6 }
```

Method 2: inverse-MSE weights

$$\widehat{w}_m = \frac{1/\text{MSE}_m}{\sum_{j=1}^M 1/\text{MSE}_j}, \quad \text{MSE}_m = \frac{1}{T_w} \sum_{t \in \text{weights}} e_{m,t}^2, \quad \widehat{w}_0 = 0$$

The M -forecast version of the zero-covariance w^* (Bates and Granger 1969; Enders 2.74).

M numbers estimated, each from its own column. Non-negative and summing to one by construction.

MSE, not the error variance, so a biased forecaster pays for its bias too.

```
1 comb_inv_mse <- function(y, F) {
2   mse <- colMeans((y - F)^2)
3   list(intercept = 0,
4         weights = (1 / mse) / sum(1 / mse))
5 }
```

Method 3: Granger-Ramanathan

$$y_t = w_0 + \sum_{m=1}^M w_m \tilde{y}_{t|t-1}^{(m)} + v_t,$$

$t \in \text{weights window}$

Unconstrained least squares with an intercept. OLS on the weights window delivers $\widehat{w}_0, \widehat{w}_1, \dots, \widehat{w}_M$.

The intercept absorbs any bias; weights can be negative and need not sum to one.

With $M = 6$ and $T_w = 12$: seven numbers from twelve observations.

v_t is the regression's disturbance: not an innovation ϵ_t , not a forecast error $e_{m,t}$.

```
1 comb_gr <- function(y, F) {
2   b <- coef(lm(y ~ F))
3   w <- setNames(b[-1], colnames(F))
4   w[is.na(w)] <- 0 # exactly collinear track
5   list(intercept = unname(b[1]), weights = w)
6 }
```

Module 4 asked the same question of extra AR lags. Do parameters that fit the sample forecast?

Method 4: constrained least squares

$$\widehat{\mathbf{w}} = \arg \min_{\mathbf{w}} \sum_{t \in \text{weights}} (y_t - \mathbf{w}' \tilde{\mathbf{y}}_t)^2 \quad \text{subject to} \quad w_m \geq 0, \quad \sum_{m=1}^M w_m = 1, \quad \widehat{w}_0 = 0$$

A quadratic program. No intercept, weights in a box: non-negative and summing to one.

The constraints do model selection: several weights land exactly on zero, and the rest read as shares.

Stable, interpretable, cannot short a forecaster.

No separate intercept for bias adjustment; reweighting can reduce bias, but cannot eliminate a common same-direction bias shared by every forecast.

```

1 comb_cls <- function(y, F) {
2   M <- ncol(F)
3   Amat <- cbind(rep(1, M), diag(M))
4   bvec <- c(1, rep(0, M)) # sum = 1; w >= 0
5   w <- quadprog::solve.QP(crossprod(F),
6                           crossprod(F, y), Amat, bvec,
7                           meq = 1)$solution
8   w <- pmax(w, 0); w <- w / sum(w)
9   list(intercept = 0,
10        weights = setNames(w, colnames(F)))
11 }

```

Four methods, one dial

Method	w_0	$\sum w_m = 1$	$w_m \geq 0$	Numbers estimated from the window	Estimation risk
Simple average (SA)	0	yes	yes	0	none
Inverse-MSE	0	yes	yes	M , one column at a time	low
Constrained LS (CLS)	0	yes	yes	M , jointly, boxed in	moderate
Granger-Ramanathan (GR)	free	no	no	$M + 1$, jointly, unrestricted	high

The four, fit on the course simulation’s weights window:

	simple avg	inverse-MSE	constrained LS	Granger-Ramanathan
$\widehat{w}_0$	0.00	0.00	0.00	261.70
arima_1	0.17	0.27	0.66	0.78
arima_2	0.17	0.14	0.00	1.88
arima_3	0.17	0.11	0.00	−2.04
locf	0.17	0.09	0.00	0.00
avg_k	0.17	0.16	0.00	−0.83
meanf	0.17	0.23	0.34	−50.83

Read it as a dial. The simple average estimates nothing; inverse-MSE a little; CLS more, inside a box; Granger-Ramanathan everything.

CLS finds [arima_1](#). Granger-Ramanathan’s intercept is 262 on a series whose mean is 5.

[Which end of the dial wins on data the weights have not seen?](#)

A combination method is an estimator

Theory says Granger-Ramanathan should win. It minimises the weights-window sum of squared errors over all intercepts and weights: no other weights can fit that window better.

But the fit is in-sample, and its weights are estimates from twelve observations. Estimates carry error.

The assumption we break: that you know the weights.

The experiment is fair. The course DGP has a dominant track, `arima_1`, so the optimal weights are far from $1/M$.

A method that finds `arima_1` should beat the simple average. Can it, with $T_w = 12$?

Stage 1: weights that win the window they were fit on

RMSE	Weights window	Holdout
arima_1	0.706	1.120
arima_2	0.979	1.208
arima_3	1.115	1.372
locf	1.208	1.301
avg_k	0.922	1.793
meanf	0.769	1.701
simple avg	0.877	1.284
inverse-MSE	0.804	1.281
constrained LS	0.681	1.237
Granger-Ramanathan	0.282	3.009

One realization (seed 6376), $T_w = 12$: the \$7.6 weights, scored on the window they were fit on and on the holdout.

Granger-Ramanathan, weights window **0.282**

simple average; [arima_1](#) **0.877; 0.706**

Granger-Ramanathan, holdout **3.009**

Best in the window, worst on the holdout: almost as many weights as observations, fitting the noise.

The other three sit close together; CLS, which found [arima_1](#), is the best of them.

Scoring a method: holdout RMSE relative to the simple average

$$\text{relative RMSE}_{\text{method}} = \frac{\text{RMSE of the method's combination on the holdout}}{\text{RMSE of the simple average on the same holdout}}$$

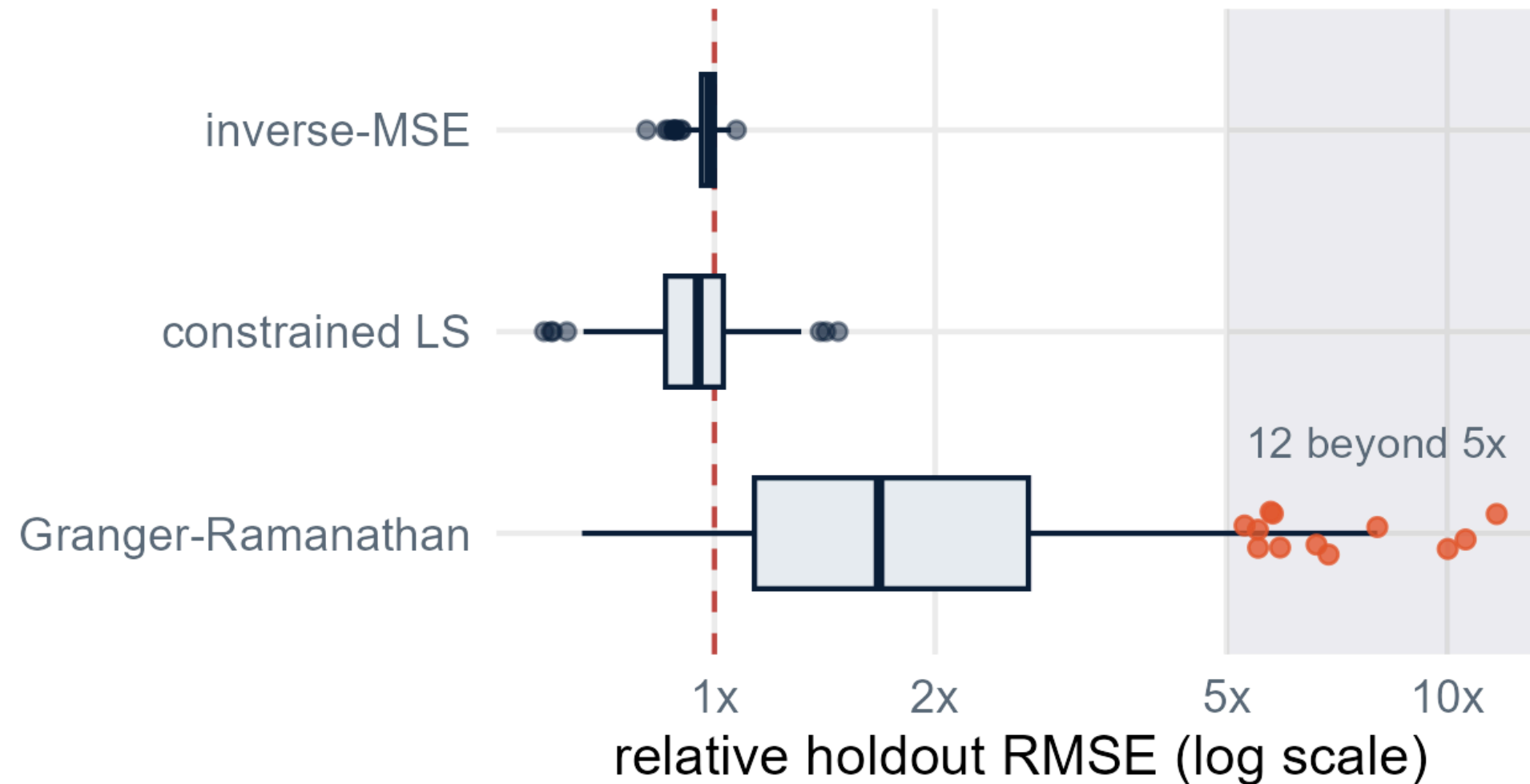
Below 1	beat the simple average	Over $R = 200$ simulated series, one ratio per replication. Three summaries:
Equal to 1	a tie	
Above 1	lost to it	
Weights fit on the weights window; scored on the 12 holdout dates the weights never saw.		Median the typical replication
		90th percentile the bad case: the right tail
		Share beating the simple average fraction of replications below 1

Not U_2 . U_2 divides by **locf**'s RMSE; this ratio divides by the simple average's. Same dates, a different benchmark.

Stage 2: 200 replications at a 12-date weights window

Each replication (seed 6378): a fresh series, the ARIMAs estimated once, six tracks, four methods fit on $T_w = 12$, scored on the 12-date holdout relative to the simple average.

Granger-Ramanathan beats the simple average in about 80%, 50%, or 15% of replications?



	Median	90th pct	Share < 1
inverse-MSE	0.99	1.01	74.5%
constrained LS	0.95	1.12	64.0%
Granger-Ramanathan	1.68	4.07	16.0%

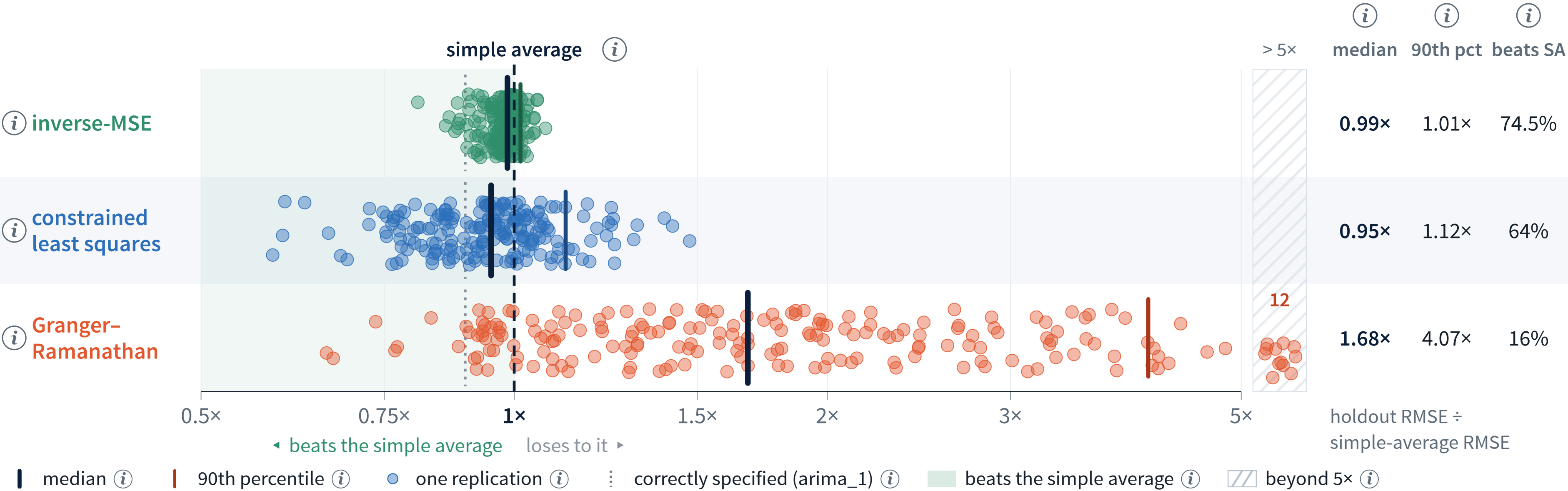
The long right tail is the puzzle in one picture. Inverse-MSE is the simple average with a tilt (median 0.99); CLS beats the simple average at the median (0.95).

Every ratio above 5× is drawn (orange) in the shaded band: 12 of Granger-Ramanathan's 200, the largest 11.7×; in the widget they fill the hatched lane. The summaries use every replication.

Stage 3: how long a window do estimated weights need?

weights window $T_w =$ **12** 24 60 120

R = 200 replications · holdout N = 12 · seed 6378



$T_w = 12$: **Granger-Ramanathan's estimated weights overfit**: at the median its holdout RMSE is 1.68× the simple average's (90th percentile 4.07×). At the median the restricted methods already beat the simple average: constrained least squares 0.95×, inverse-MSE 0.99×.

Each dot is one simulated series from the course DGP, the same 200 series at every T_w , so the dots slide as the window grows; the axis is on a log scale. Hover, focus or tap any item marked ⓘ for a definition. Notes §7.7, Stages 2–3.

The forecast combination puzzle

Theory says optimally weighted combinations should dominate. In practice, the simple average is very hard to beat. [Stock and Watson \(2004\)](#)

Three lessons from the simulation.

1. **A combination method is an estimator.** Holdout performance = the population gain from its weights minus the estimation error in them.
2. **The puzzle is a comparison: attainable gain against estimation cost.** It bites when the gain over equal weights is small relative to the cost of estimating the weights: near-equal optimal weights, or redundant forecasts. Twelve weights-window dates make the cost high, as in most applied work, and PS3.
3. **Restrictions are a dial.** Moving down from $1/M$ pays only when the window is long or the gain is large.

The index fund. Active managers (estimated weights) should beat the index (equal weights): they pick the winners. In practice fees eat the edge. Here the fee is estimation error.

But an index fund that holds a bankrupt stock at full weight is not a good index. Equal weights are robust to estimation error, not to a broken member of the pool.

UNRATE: weights fit on 2018

```

1 ws_un <- list(SA = comb_simple(y_w, F_w),
2               IMSE = comb_inv_mse(y_w, F_w),
3               CLS = comb_cls(y_w, F_w),
4               GR = comb_gr(y_w, F_w))
5 sapply(ws_un, function(w)
6   c(w0 = w$intercept, w$weights))
7 kappa(cbind(1, F_w), exact = TRUE)

```

Condition number, GR design matrix **41,620**

Before the holdout: which of the four will the simple average beat on 2019?

	simple avg	inverse-MSE	constrained LS	Granger- Ramanathan
$\widehat{w}_0$	0.000	0.000	0.000	-141.450
arima_1	0.167	0.235	0.690	1.224
arima_2	0.167	0.158	0.000	-5.825
arima_3	0.167	0.229	0.000	1.088
locf	0.167	0.179	0.000	5.098
avg_k	0.167	0.198	0.310	-2.605
meanf	0.167	0.001	0.000	24.803

Over twelve months the expanding mean of 696+ observations barely moves: the `meanf` column is almost a constant, almost collinear with the intercept.

UNRATE: scored on 2019

2019	Holdout RMSE
arima_1	0.097
arima_2	0.097
arima_3	0.096
locf	0.096
avg_k	0.108
meanf	2.315
simple avg	0.416
inverse-MSE	0.093
constrained LS	0.094
Granger-Ramanathan	0.463

Simple average	0.416
credible single tracks	0.096 to 0.108
Inverse-MSE; CLS	0.093; 0.094
Granger-Ramanathan	0.463

Of the four combinations, the simple average beat only Granger-Ramanathan.

Inverse-MSE and CLS match or edge out the best single tracks; the simple average is roughly four times their RMSE.

UNRATE: each estimated-weight combination against the simple average

Three tests, declared in advance, all reported.

```
1 comb_h <- sapply(ws_un, function(w)      # N x 4
2   w$intercept + F_h %*% w$weights)
3 ec_h <- y_h - comb_h      # combined errors
4 for (m in c("IMSE", "CLS", "GR"))
5   print(dm.test(ec_h[, m], ec_h[, "SA"],
6               h = 1, power = 2))
```

Never pick the holdout winner and then test it on the same twelve dates: that chooses the hypothesis after seeing the data.

Against the simple average	DM	p
Inverse-MSE	-9.1	1.9×10^{-6}
CLS	-9.1	1.8×10^{-6}
Granger-Ramanathan	0.87	0.40

Inverse-MSE and CLS: a clear rejection at $N = 12$, the power lesson from the other side. Granger-Ramanathan: no detectable difference.

The $h = 1$ check, all four UNRATE tests: Ljung-Box $p = 0.58$ to 0.97 on d_t (little power at $N = 12$); Bartlett variance, $h = 2:4$: no verdict changes.

The honest reading: a broken member, not a lucky estimate

Here the attainable gain is large. `meanf` has no business on an $I(1)$ series, and $1/M$ gives it a sixth of the vote: enough to drag the average about 0.39 points off the actuals every month.

Inverse-MSE and CLS won by down-weighting it, which is exactly what estimated weights are for: that gain pays for estimating the restricted weights.

Granger-Ramanathan lost for the Stage 1 reason, amplified by near-collinearity: the intercept and the nearly constant `meanf` column are almost the same regressor.

Caveats: one window, a calm glide with no turning point; the $h = 1$ verdicts survive a low-power dependence check, so robust, not definitive. One realization is one draw; the Monte Carlo is the distribution.

For Problem Set 3: look at your six tracks before you combine them. If one is broken on your series, say so, explain why from the series' properties, and expect it to drag the simple average.

Equal weights are robust to estimation error.
What are they not robust to?

A broken member of the pool.

Key takeaways + next time

1. **Many origins, one horizon.** Estimate once; roll the information set.
2. U_2 : did you beat the naive here? **MASE**: how big was the miss against the naive's training miss?
3. **A metrics table is a ranking, not a verdict.** DM is a t -test on d_t ($h = 1$: assumes serially uncorrelated d_t); twelve dates buy little power. UNRATE: five tracks descriptively close; `arima_1` vs `locf` does not reject.
4. **Combine because errors diversify.** Zero covariance: weights $\propto$ inverse error variance.
5. **A combination method is an estimator.** With $T_w = 12$, estimate as little as you can.
6. **Equal weights are the benchmark to beat, when the pool is sound.** The puzzle: estimation cost outweighs the attainable gain.
7. Every forecast here used only the series' own past.

Next time, Module 8: ADL models.

What if an exogenous regressor could improve the forecast?

$$y_t = \alpha + \beta x_t + \sum_{j=1}^p \phi_j y_{t-j} + \epsilon_t$$

A new slider on the mixing board: βx_t and its lags. Combinations only reshuffle the information already in the pool.