

# Package: throughyear (via r-universe)

October 9, 2026

**Title** Linking, Routing and Fairness for Through-Year Assessment

**Version** 0.1.0

**Description** Treats a through-year assessment system (interims during the year feeding a multistage summative) as the unit of analysis. Links interims to the summative scale with a latent multivariate normal model that carries each score's measurement error forward and handles missing interims, estimated by the EM algorithm (Dempster, Laird and Rubin, 1977, <[doi:10.1111/j.2517-6161.1977.tb01600.x](https://doi.org/10.1111/j.2517-6161.1977.tb01600.x)>) with SQUAREM acceleration (Varadhan and Roland, 2008, <[doi:10.1111/j.1467-9469.2007.00585.x](https://doi.org/10.1111/j.1467-9469.2007.00585.x)>); simulates cold-start versus prior-informed routing in a two-stage multistage test; checks whether priors disadvantage late enrollers, low scorers or fast growers; and evaluates decision accuracy and consistency of through-year scores against a single summative.

**License** MIT + file LICENSE

**URL** <https://github.com/edidasolutions/throughyear>,  
<https://edidasolutions.github.io/throughyear/>

**BugReports** <https://github.com/edidasolutions/throughyear/issues>

**Encoding** UTF-8

**Depends** R (>= 4.1)

**Imports** stats

**Suggests** knitr, markdown

**VignetteBuilder** knitr

**Config/roxygen2/version** 8.1.0

**Repository** <https://edidasolutions.r-universe.dev>

**Date/Publication** 2026-10-09 03:04:20 UTC

**RemoteUrl** <https://github.com/edidasolutions/throughyear>

**RemoteRef** HEAD

**RemoteSha** 3d87d1e6f1e6e21a949f784e2f0185d8339d95e3

Contents

|                           |           |
|---------------------------|-----------|
| predict.ty_link . . . . . | 2         |
| ty_administer . . . . .   | 3         |
| ty_decisions . . . . .    | 4         |
| ty_fairness . . . . .     | 5         |
| ty_link . . . . .         | 6         |
| ty_mst . . . . .          | 7         |
| ty_mst_default . . . . .  | 7         |
| ty_policies . . . . .     | 8         |
| ty_simulate . . . . .     | 9         |
| <b>Index</b>              | <b>11</b> |

---

|                 |                                                            |
|-----------------|------------------------------------------------------------|
| predict.ty_link | <i>Projected summative score (the prior) from interims</i> |
|-----------------|------------------------------------------------------------|

---

Description

Conditional distribution of the summative true score given a student’s observed interims and their SEs. Missing interims simply drop out, so late enrollers get wider priors rather than wrong ones.

Usage

```
## S3 method for class 'ty_link'
predict(object, newdata, suffix = "", ...)
```

Arguments

|         |                                                                                                                             |
|---------|-----------------------------------------------------------------------------------------------------------------------------|
| object  | A ‘ty_link’.                                                                                                                |
| newdata | Data frame with the interim columns and their SEs.                                                                          |
| suffix  | Optional suffix of alternative interim columns (e.g. “_r2” for a replicate set); SEs are still read from the ‘_se’ columns. |
| ...     | Unused.                                                                                                                     |

Value

Data frame: ‘mean’, ‘sd’, ‘n\_interims’.

Examples

```
sim <- ty_simulate(n_calibration = 300, n_operational = 300, seed = 1)
link <- ty_link(sim)
op <- sim[sim$cohort == "operational", ]
prior <- predict(link, op)
# late enrollers get wider priors
aggregate(prior$sd, list(late = op$late), mean)
```

---

ty\_administer

Administer a two-stage MST to simulated examinees

---

### Description

Routing uses the EAP after the (possibly shortened) routing module under ‘route\_prior’. With ‘routing\_n = 0’ the routing decision uses the prior alone. The final score is the EAP over all administered items under ‘score\_prior’. Priors are lists with ‘mean’ and ‘sd’ (scalars or one value per examinee).

### Usage

```
ty_administer(
  mst,
  theta,
  route_prior,
  score_prior,
  routing_n = NULL,
  grid = seq(-5, 5, by = 0.05),
  seed = NULL
)
```

### Arguments

|                          |                                     |
|--------------------------|-------------------------------------|
| mst                      | A ‘ty_mst’.                         |
| theta                    | True abilities.                     |
| route_prior, score_prior | Priors for routing and for scoring. |
| routing_n                | Routing items used (default: all).  |
| grid                     | Theta grid.                         |
| seed                     | Optional seed.                      |

### Value

Data frame: ‘module’, ‘correct\_module’ (most informative module at the true theta), ‘route\_eap’, ‘theta\_hat’, ‘se’, ‘n\_items’.

### Examples

```
mst <- ty_mst_default()
pop <- list(mean = 0, sd = 1)
res <- ty_administer(mst, theta = rnorm(500), route_prior = pop, score_prior = pop, seed = 1)
mean(res$module == res$correct_module) # routing accuracy
```

ty\_decisions

*Decision accuracy and consistency: through-year vs single summative***Description**

Compares three ways of making a proficiency decision at ‘cut’ (theta scale):

**summative** Single summative (cold MST, population prior).

**through\_year** Interim projection alone (prior mean from the link): the summative-replacement scenario.

**combined** Summative scored with the interim prior (interims and summative both count).

Accuracy is agreement with the true decision. Consistency is agreement between two independent replications: two MST administrations, and two independent sets of interim scores (‘prior’ and ‘prior\_r2’).

**Usage**

```
ty_decisions(
  mst,
  theta,
  prior,
  prior_r2,
  cut,
  population = NULL,
  groups = NULL,
  seed = NULL
)
```

**Arguments**

|                 |                                                           |
|-----------------|-----------------------------------------------------------|
| mst             | A ‘ty_mst’.                                               |
| theta           | True summative abilities.                                 |
| prior, prior_r2 | Student priors from two independent interim sets.         |
| cut             | Proficiency cut on the summative theta scale.             |
| population      | Population prior ‘c(mean, sd)’.                           |
| groups          | Optional named list of logical vectors for subgroup rows. |
| seed            | Optional seed.                                            |

**Value**

Data frame: ‘method’, ‘group’, ‘accuracy’, ‘consistency’, ‘false\_proficient’, ‘false\_not\_proficient’.

## Examples

```
sim <- ty_simulate(n_calibration = 300, n_operational = 300, seed = 1)
op <- sim[sim$cohort == "operational", ]
link <- ty_link(sim)
ty_decisions(ty_mst_default(), op$theta_S, predict(link, op),
             predict(link, op, suffix = "_r2"), cut = 0.3,
             groups = list(fast = op$fast), seed = 1)
```

---

ty\_fairness

*Fairness diagnostics for prior-informed routing and scoring*


---

## Description

For each policy and group, reports routing accuracy, the rate of being routed to an easier module than the one most informative at the student's true ability (the "locked into an easier path" concern), and bias and RMSE of the reported score. Bias for a group that the prior systematically under-predicts (late bloomers, students whose growth accelerated after the last interim) is the central fairness signal.

## Usage

```
ty_fairness(policies, groups)
```

## Arguments

|          |                                                           |
|----------|-----------------------------------------------------------|
| policies | A 'ty_policies' object.                                   |
| groups   | Named list of logical vectors (one element per examinee). |

## Value

Data frame: 'policy', 'group', and the metrics of 'summary()'.

## Examples

```
sim <- ty_simulate(n_calibration = 300, n_operational = 300, seed = 1)
op <- sim[sim$cohort == "operational", ]
prior <- predict(ty_link(sim), op)
pol <- ty_policies(ty_mst_default(), op$theta_S, prior, seed = 1)
fair <- ty_fairness(pol, list(late = op$late, fast = op$fast))
fair[fair$group == "fast", c("policy", "routed_too_easy", "bias")]
```

ty\_link

*Link interims to the summative scale***Description**

Latent-variable linking. The true scores on all occasions (interims on their own reporting scales, and the summative) are jointly multivariate normal,  $\tau \sim \text{MVN}(\mu, \Sigma)$ . Each observed score equals its true score plus error with the reported (known) SE, and any score may be missing.  $\mu$  and  $\Sigma$  are estimated by EM from all students; only the calibration cohort needs summative scores. Because measurement error is modeled rather than ignored, the regression of summative on interims is not attenuated, and a student's projection carries their own interim precision forward.

**Usage**

```
ty_link(
  data,
  interims = attr(data, "interims"),
  summative = "S",
  se_suffix = "_se",
  max_iter = 1000,
  tol = 1e-06
)
```

**Arguments**

|               |                                                                                                                      |
|---------------|----------------------------------------------------------------------------------------------------------------------|
| data          | Data frame with interim scores, the summative score, and their SEs.                                                  |
| interims      | Interim score columns, in time order.                                                                                |
| summative     | Summative score column (NA for students without one yet).                                                            |
| se_suffix     | Suffix of the SE columns.                                                                                            |
| max_iter, tol | EM controls; 'tol' is relative to the largest parameter in $\Sigma$ , so it does not depend on the reporting scales. |

**Value**

A 'ty\_link' object with  $\mu$ ,  $\Sigma$ ,  $\text{vars}$ ,  $\text{summative}$ ,  $\text{converged}$ ,  $\text{iterations}$  (EM step evaluations),  $\text{loglik}$ .

**Examples**

```
sim <- ty_simulate(n_calibration = 300, n_operational = 300, seed = 1)
link <- ty_link(sim)
link
```

---

|        |                                           |
|--------|-------------------------------------------|
| ty_mst | <i>Define a two-stage multistage test</i> |
|--------|-------------------------------------------|

---

**Description**

Define a two-stage multistage test

**Usage**

```
ty_mst(routing_b, modules, cuts = NULL)
```

**Arguments**

|           |                                                                                                                          |
|-----------|--------------------------------------------------------------------------------------------------------------------------|
| routing_b | Rasch difficulties of the routing module, in the order items would be dropped from the end when the module is shortened. |
| modules   | Named list of second-stage modules (difficulty vectors), ordered from easiest to hardest.                                |
| cuts      | Routing cuts on theta; by default the points where adjacent modules' information functions cross.                        |

**Value**

A 'ty\_mst'.

**Examples**

```
mst <- ty_mst(routing_b = seq(-1.5, 1.5, length.out = 10),
              modules = list(easy = rnorm(20, -1, 0.5), hard = rnorm(20, 1, 0.5)))
mst$cuts # where the two modules' information functions cross
```

---

|                |                                                                            |
|----------------|----------------------------------------------------------------------------|
| ty_mst_default | <i>A default 1-3 MST: 12 routing items, easy/medium/hard modules of 24</i> |
|----------------|----------------------------------------------------------------------------|

---

**Description**

A default 1-3 MST: 12 routing items, easy/medium/hard modules of 24

**Usage**

```
ty_mst_default(center = 0)
```

**Arguments**

|        |                                 |
|--------|---------------------------------|
| center | Center of the difficulty range. |
|--------|---------------------------------|

**Value**

A ‘ty\_mst’.

**Examples**

```
ty_mst_default()$cuts
```

---

ty\_policies

---

*Compare routing and scoring policies*


---

**Description**

Administers the MST under five policies to the same examinees:

**cold** Full routing module, population prior for routing and scoring.

**prior\_route** Full routing module, student’s interim prior for routing only; population prior for the reported score.

**prior\_both** Interim prior for routing and for the reported score.

**prior\_short** Interim prior for routing with a shortened routing module (‘short\_n’ items); population prior for scoring.

**prior\_only** Route on the interim prior alone (no routing module); population prior for scoring.

**Usage**

```
ty_policies(mst, theta, prior, population = NULL, short_n = 6, seed = NULL)
```

**Arguments**

|            |                                                                                           |
|------------|-------------------------------------------------------------------------------------------|
| mst        | A ‘ty_mst’.                                                                               |
| theta      | True summative abilities.                                                                 |
| prior      | Student priors (‘mean’, ‘sd’) from ‘predict()’ on a ‘ty_link’.                            |
| population | Population prior ‘c(mean, sd)’; default: moments of the student priors’ implied marginal. |
| short_n    | Routing items for ‘prior_short’.                                                          |
| seed       | Optional seed (each policy gets its own stream).                                          |

**Value**

A ‘ty\_policies’ object: named list of [ty\_administer()] results, plus ‘theta’.

**Examples**

```
sim <- ty_simulate(n_calibration = 300, n_operational = 300, seed = 1)
op <- sim[sim$cohort == "operational", ]
prior <- predict(ty_link(sim), op)
pol <- ty_policies(ty_mst_default(), op$theta_S, prior, seed = 1)
summary(pol)
```

ty\_simulate

*Simulate a through-year system with known true growth***Description**

Students grow linearly,  $\theta(t) = \theta_0 + g * t$ , and take interims at ‘times’ (reported on their own scale,  $\text{scale}[1] + \text{scale}[2] * \theta$ , with error from a Rasch form of ‘interim\_items’ items) and the summative at  $t = 1$ . Late enrollers miss the first ‘late\_missing’ interims. "Fast growers" gain an extra ‘fast\_extra’ logits after the last interim (e.g. a spring intervention), which interims cannot reveal. They are the hardest case for prior-informed scoring.

**Usage**

```
ty_simulate(
  n_calibration = 3000,
  n_operational = 3000,
  times = c(0.2, 0.5, 0.8),
  theta0_mean = -0.6,
  theta0_sd = 1,
  growth_mean = 0.6,
  growth_sd = 0.25,
  p_fast = 0.1,
  fast_extra = 0.6,
  p_late = 0.1,
  late_missing = 2,
  scale = c(200, 10),
  interim_items = 30,
  summative_se = 0.3,
  seed = NULL
)
```

**Arguments**

|                                                |                                                  |
|------------------------------------------------|--------------------------------------------------|
| n_calibration, n_operational                   | Cohort sizes.                                    |
| times                                          | Interim occasions as fractions of the year.      |
| theta0_mean, theta0_sd, growth_mean, growth_sd | True-score model.                                |
| p_fast, fast_extra                             | Share of fast growers and their extra growth.    |
| p_late, late_missing                           | Share of late enrollers and interims they miss.  |
| scale                                          | Interim reporting scale: intercept and slope.    |
| interim_items                                  | Items per interim form (sets measurement error). |
| summative_se                                   | SE of the calibration cohort’s summative scores. |
| seed                                           | Optional seed.                                   |

**Details**

Two cohorts: ‘calibration’ (last year: summative observed, used to link) and ‘operational’ (this year: summative not yet taken). A second, independent set of interim scores (‘\*\_r2’) supports decision-consistency analyses.

**Value**

A ‘ty\_sim’ data frame, one row per student.

**Examples**

```
sim <- ty_simulate(n_calibration = 300, n_operational = 300, seed = 1)
head(sim[c("id", "cohort", "late", "fast", "I1", "I2", "I3", "S")])
```

# Index

`predict.ty_link`, [2](#)

`ty_administer`, [3](#)

`ty_decisions`, [4](#)

`ty_fairness`, [5](#)

`ty_link`, [6](#)

`ty_mst`, [7](#)

`ty_mst_default`, [7](#)

`ty_policies`, [8](#)

`ty_simulate`, [9](#)