Latent class growth analysis with the latents R package

Mohammed Saqr, University of Eastern Finland
Sonsoles López-Pernas, University of Eastern Finland

26 November 2026

Contents

The data 2
Fitting and choosing the number of classes 3
Seeing the classes 5
Curved trajectories 9
Outcomes that are not scores 10
Distal outcomes 14
Choices and warnings 15
References 15

Latent class growth analysis (LCGA), also known as group-based trajectory modelling, is a person-centred approach used to identify distinct subgroups of individuals who exhibit different patterns of change over time. When the same individuals are measured repeatedly, LCGA assumes that the population consists of a finite number of latent classes, each characterized by a specific developmental trajectory. These trajectories describe how the outcome of interest changes over time within each subgroup. Individuals are assigned to latent classes according to their posterior probabilities of membership, which indicate the extent to which their observed measurement patterns correspond to each estimated trajectory (Nagin, 2005).

Similar to latent profile analysis, LCGA focuses on identifying meaningful subgroups within a population, but rather than distinguishing individuals according to their patterns of responses at a particular point in time, it differentiates them according to their longitudinal patterns of development (Scrucca et al., 2024; Saqr et al., 2024). A defining feature of LCGA is the assumption that individuals within the same class share an identical underlying growth trajectory. Accordingly, within-class variation in individual growth parameters is constrained to zero, and deviations from the estimated trajectory are attributed to random measurement error. This assumption distinguishes LCGA from growth mixture modelling, which allows individuals within the same class to vary in their growth parameters and, consequently, in their individual developmental trajectories.

LCGA is a special case of mixture regression (see the mixture regression article). Time is the predictor, and the class is shared by all measurements of a person rather than assigned to each measurement separately. In latents it is therefore fitted with `mixture_regression()`, with time on the right of the formula and `class_level = "group"`; adding random effects through `random` turns the same call into a growth mixture model.

The assumption of identical trajectories within a class is both the strength and the limitation of the model:

- **Strength.** The model is simple, fast to estimate and, as a mixture regression, accommodates every outcome family of `mixture_regression()`: continuous scores, binary outcomes, counts and ordered ratings.
- **Limitation.** Individuals of the same kind rarely follow one trajectory exactly. When they vary around it, the model can represent that variation only by adding classes, and it therefore tends to identify more classes than there are kinds of individuals (Bauer & Curran, 2003). The growth mixture model adds random effects for that variation; it is the subject of the growth mixture article.

Fitting both models and comparing them with `compare_models()`, as the growth mixture article does, is therefore advisable.

Role	Argument	Here
Outcome, measured repeatedly	left of `~`	`score`
Time (any shape)	right of `~`	`wave`, `wave + I(wave^2)`
Person	`id`, with `class_level = "group"`	`student`
Kind of outcome	`family`	`"gaussian"`, `"binomial"`, `"poisson"`, `"negative_binomial"`, `"ordinal"`
What predicts the class	`membership`	`"motivation"`

The data

`growth_scores` is simulated so the answer is known: 300 students tested on up to seven waves, of three kinds (improving, stable, declining), each student varying a little around their kind's trajectory, with motivation predicting the kind. Missed tests (about 12%) need no imputation.

```r
library(latents)
head(growth_scores)
#>   student wave score motivation trajectory
#> 1       1    0  60.2      -0.72  declining
#> 2       1    2  49.7      -0.72  declining
#> 3       1    3  49.2      -0.72  declining
#> 4       1    4  44.9      -0.72  declining
#> 5       1    5  44.4      -0.72  declining
#> 6       1    6  41.5      -0.72  declining
```

Fitting and choosing the number of classes

Persons are the classified units, so `id` names them and `class_level = "group"` keeps all of a person's measurements in one class.

```r
enumerate_regressions(score ~ wave, growth_scores, n_classes = 1:5,
                      id = "student", class_level = "group",
                      n_starts = 5, seed = 1)
#> Mixture regression class enumeration
#>
#>  n_classes n_group_classes log_likelihood n_parameters   bic   icl entropy smallest_share
#>          1               1          -6629            3 13276 13276      NA         1.0000
#>          2               1          -6080            7 12199 12210  0.9725         0.3263
#>          3               1          -5890           11 11842 11897  0.9156         0.3178
#>          4               1          -5778           15 11641 11689  0.9419         0.1376
#>          5               1          -5658           19 11425 11487  0.9359         0.1248
#>  converged best_bic
#>       TRUE    FALSE
#>       TRUE    FALSE
#>       TRUE    FALSE
#>       TRUE    FALSE
#>       TRUE     TRUE
#>
#> One model: get_results(x, "model", n_classes = ).
```

The BIC keeps falling up to five classes (from 13,276 with one class to 11,425 with five), although only three kinds of student generated the data: the students' own spread around their kind's trajectory is absorbed by additional classes. With these data the growth mixture model, which models that spread, picks three. Here, to keep the three kinds, the three-class model is read:

```r
fit <- mixture_regression(score ~ wave, growth_scores, n_classes = 3,
                          id = "student", class_level = "group",
                          membership = "motivation", n_starts = 5, seed = 1)
fit
#> Mixture of gaussian regressions: 3 classes (one class per `student`)
#> score ~ wave
#> 1844 rows in 300 groups | log likelihood -5848.9651 | BIC 11772.08 | entropy 0.917
#> Converged: TRUE | iterations: 22 | best likelihood reached by 6 of 6 completed starts (6 run)
#>
#> Classes
#>
#> Class    Share  Expected  Groups assigned  Avg. posterior  Residual SD
```

```
#> -------  -----  --------  ----------------  --------------  -----------
#> Class 1  0.345    103.43               104           0.945         5.16
#> Class 2  0.336    100.81               101           0.989         5.16
#> Class 3  0.319     95.76                95           0.949         4.71
#>
#> Regression coefficients (95% CI)
#>
#> Class    Term       Estimate          95% CI      p
#> -------  ---------  --------  ---------------  -----
#> Class 1  Intercept     61.00  [60.00, 62.01]  <.001
#> Class 1  wave           1.05  [ 0.80,  1.31]  <.001
#> Class 2  Intercept     57.87  [57.12, 58.63]  <.001
#> Class 2  wave          -2.35  [-2.56, -2.14]  <.001
#> Class 3  Intercept     47.56  [46.60, 48.52]  <.001
#> Class 3  wave           3.37  [ 3.14,  3.61]  <.001
#>
#> Every table: get_results(x, what = ), e.g. "classes", "membership", "fit", "assignments".
```

Every table is a data frame from `get_results()`:

```
get_results(fit, "classes")
#> Classes
#>
#> Class    Share  Expected  Groups assigned  Avg. posterior  Residual SD
#> -------  -----  --------  ----------------  --------------  -----------
#> Class 1  0.345    103.43               104           0.945         5.16
#> Class 2  0.336    100.81               101           0.989         5.16
#> Class 3  0.319     95.76                95           0.949         4.71
get_results(fit, "membership")
#> Class membership (log odds, 95% CI)
#>
#> Class    Term        Estimate          95% CI      p  Odds ratio
#> -------  ----------  --------  ---------------  -----  ----------
#> Class 2  Intercept      -0.28  [-0.62,  0.06]   .109
#> Class 2  motivation     -1.01  [-1.39, -0.62]  <.001        0.36
#> Class 3  Intercept      -0.30  [-0.69,  0.09]   .127
#> Class 3  motivation      0.62  [ 0.26,  0.99]  <.001        1.87
get_results(fit, "recovery", data = growth_scores, truth = "trajectory")
#> Recovery of a known classification
#>
#> Assigned  trajectory  Groups  Share of assigned
#> --------  ----------  ------  ------------------
#> Class 1   improving       26              0.250
#> Class 2   improving        1              0.010
#> Class 3   improving       88              0.926
#> Class 1   stable          77              0.740
#> Class 2   stable           4              0.040
#> Class 3   stable           7              0.074
#> Class 1   declining        1              0.010
#> Class 2   declining       96              0.950
#> Class 3   declining        0              0.000
```

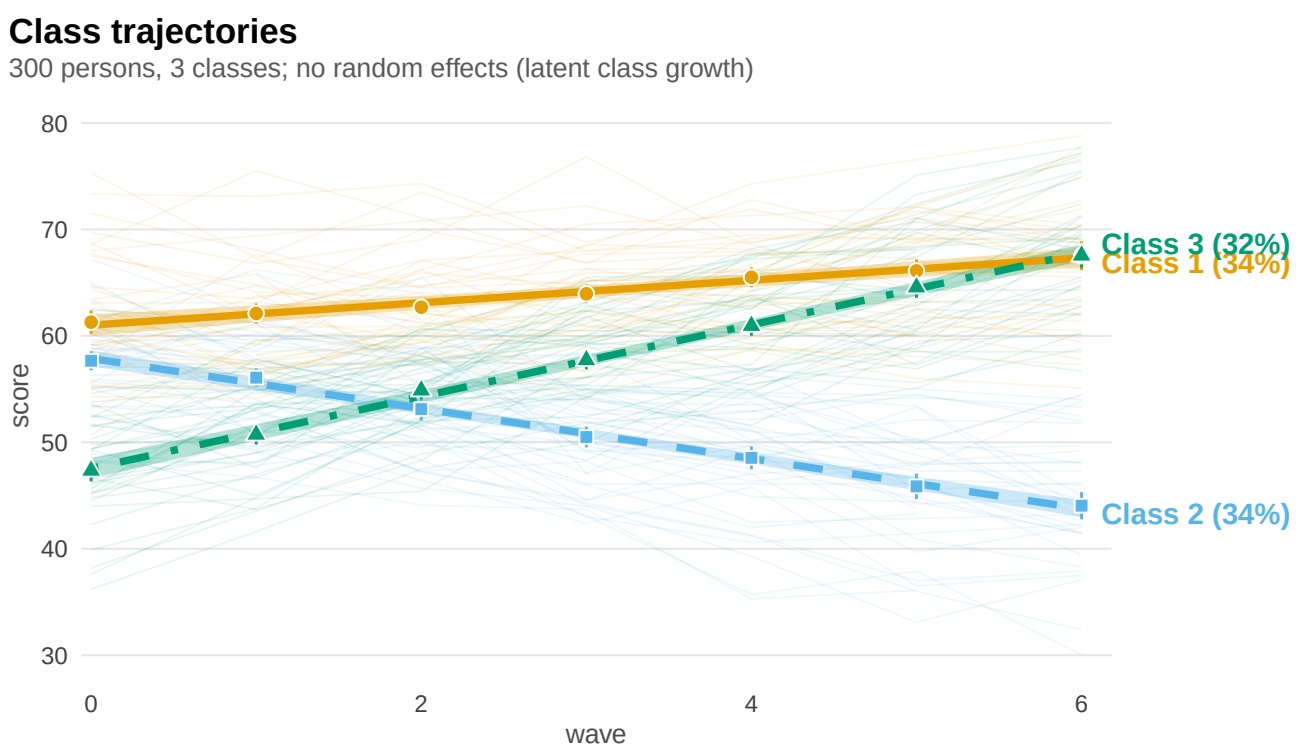
The three classes are an improving trajectory (class 3, 31.9% of the students, gaining 3.37 points per wave from 47.56), a declining trajectory (class 2, 33.6%, losing 2.35 points per wave from 57.87) and a stable

trajectory (class 1, 34.5%, gaining 1.05 points per wave from 61.00). Higher motivation lowers the odds of the declining class relative to the stable one (odds ratio 0.36) and raises the odds of the improving class (odds ratio 1.87). Of the students assigned to the improving and declining classes, 92.6% and 95.0% were generated with those trajectories, whereas a quarter of the stable class (26 of 104 students) are improving students, whose spread around their trajectory the model does not represent.

Seeing the classes

Every view of a trajectory fit is one `plot()` call. Each class's trajectory with its confidence band, the observed class means, and a sample of students' observed trajectories as thin lines (a band for the spread of students around their class needs random effects, so it appears only in the growth mixture model):

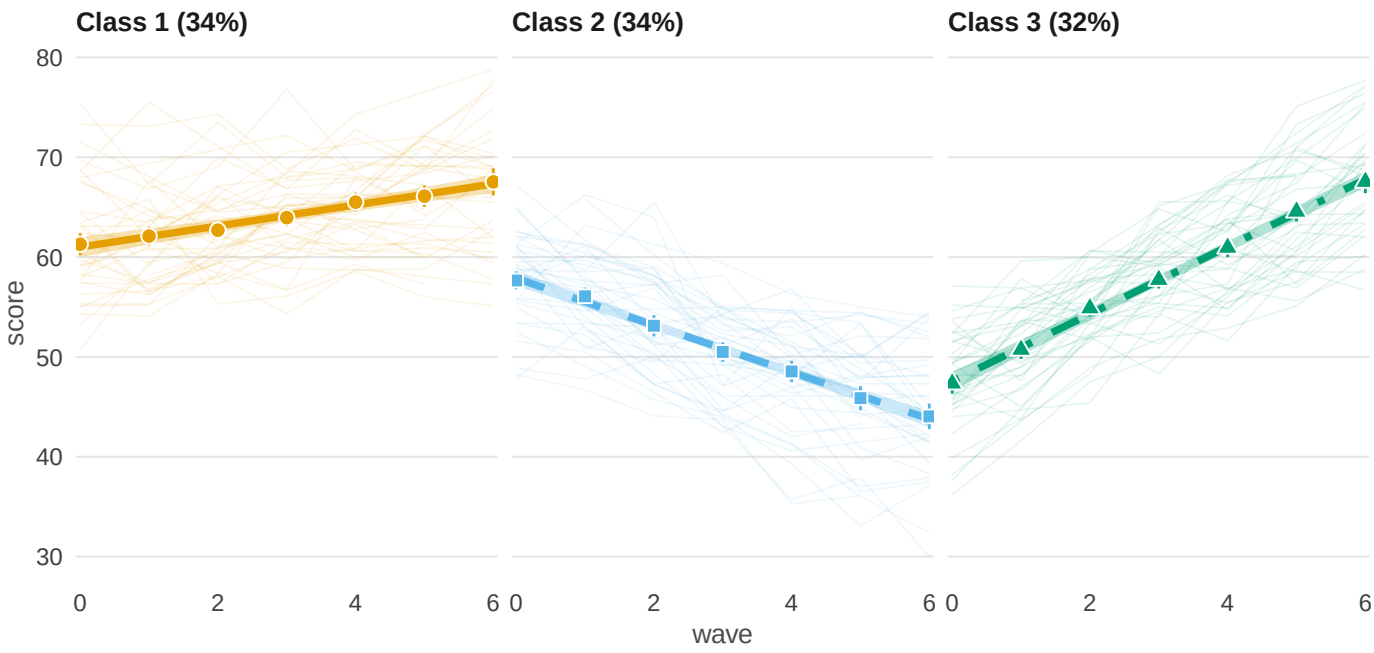
```
plot(fit)
```



Line and dark band: class mean trajectory and its 95% confidence band.
Points: observed class means (posterior−weighted) with intervals.
Thin lines: persons, coloured by their most likely class.

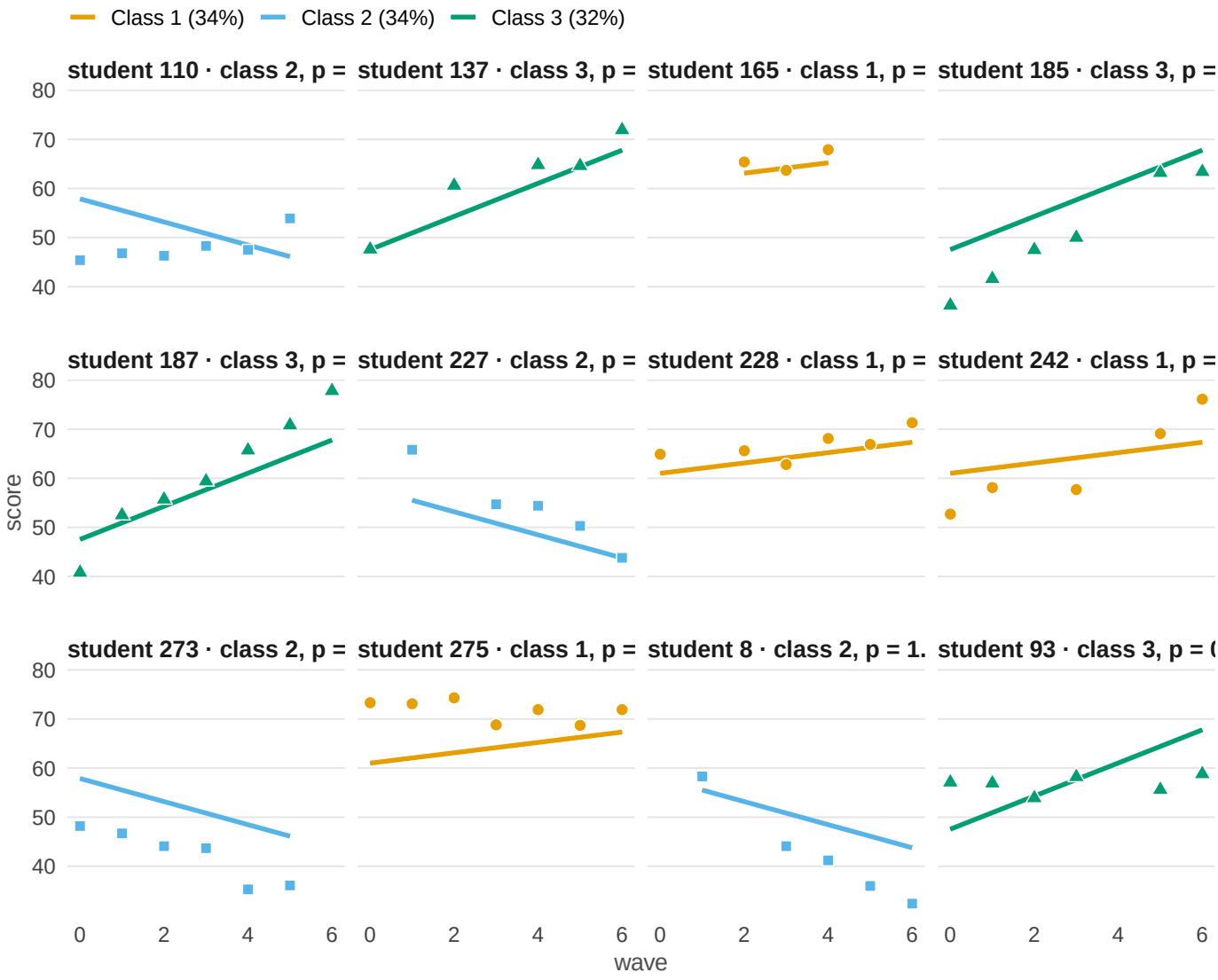
The same with one panel per class:

```
plot(fit, facet = TRUE)
```



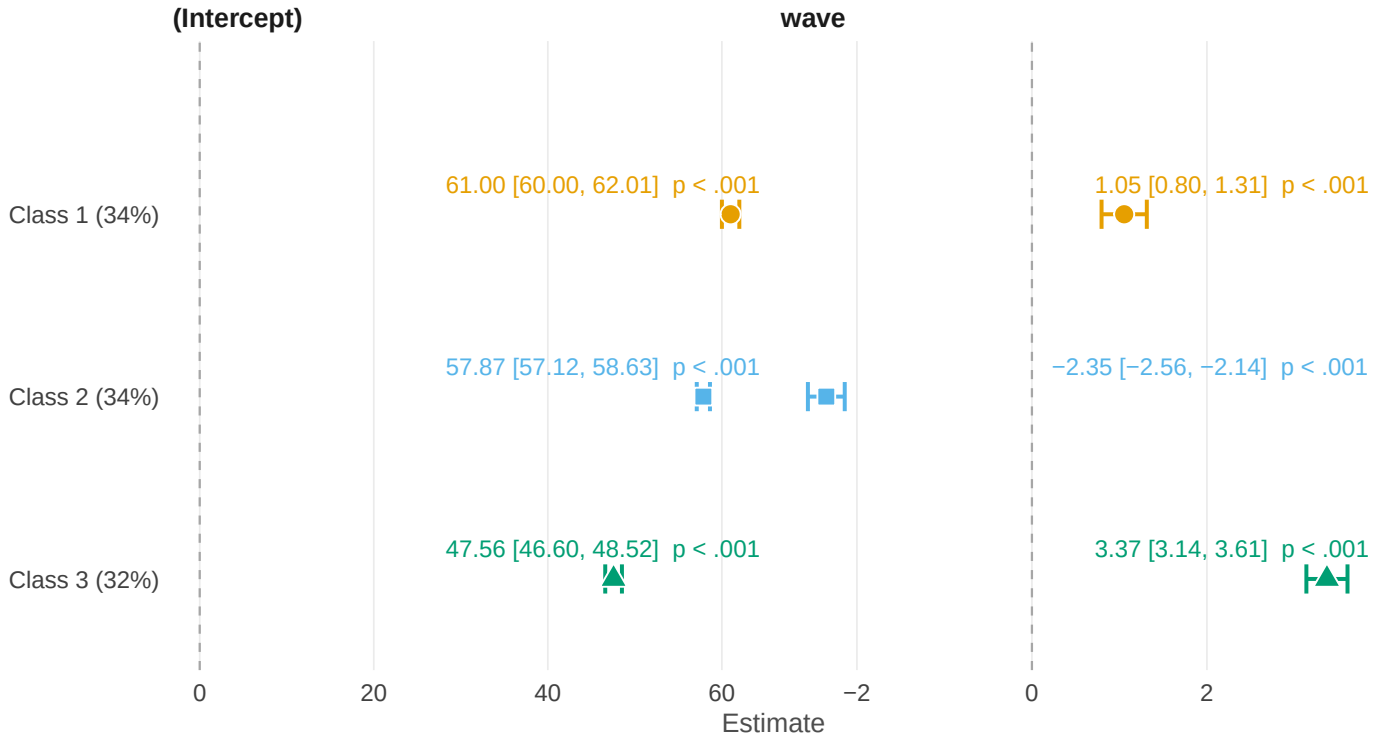
Individual students, chosen across each class's range of classification certainty: their observations and their class trajectory.

```
plot(fit, what = "individuals")
```



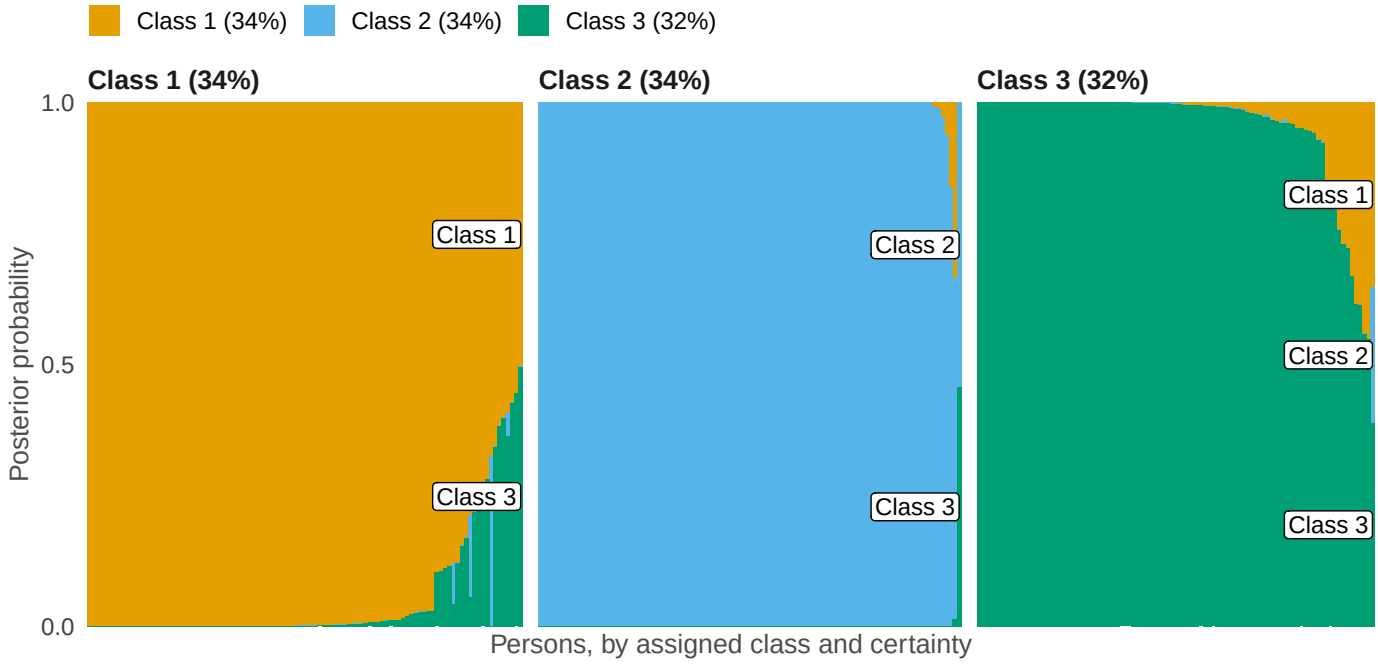
The coefficients with their intervals and p-values:

```
plot(fit, what = "coefficients")
```



How confidently each student is classified, one bar per student:

```
plot(fit, what = "classification")
```



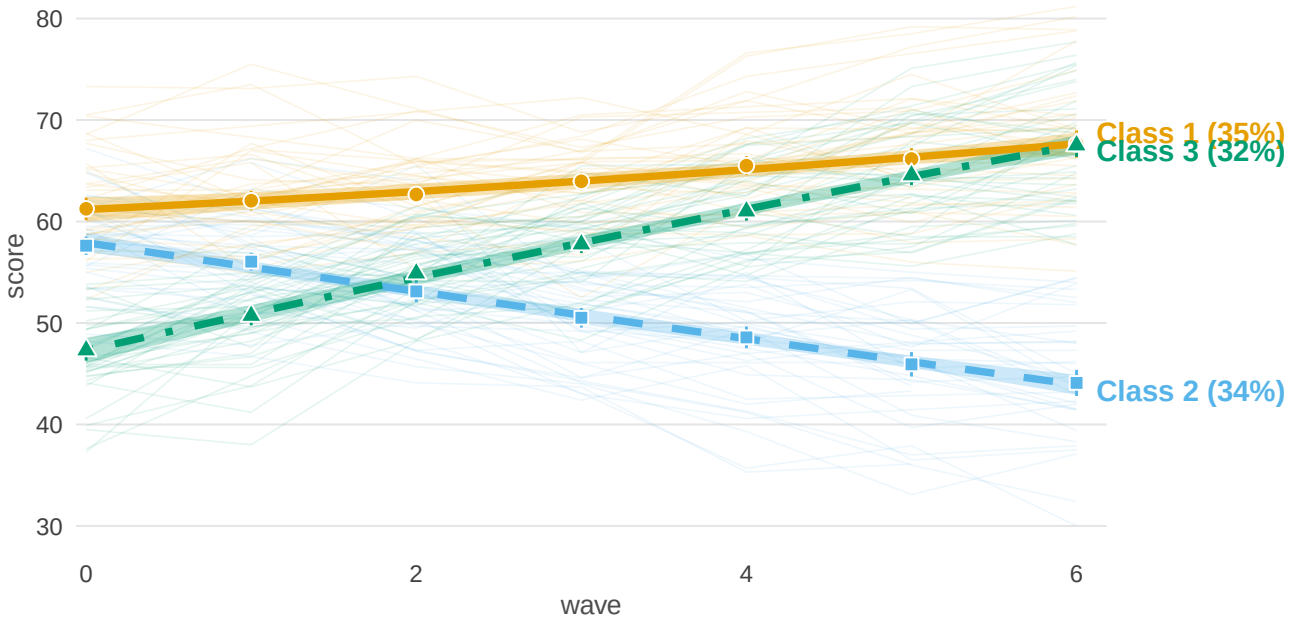
Curved trajectories

Any formula works. A quadratic lets each class bend; compare it with the straight model fitted the same way (without `membership`):

```
straight <- mixture_regression(score ~ wave, growth_scores, n_classes = 3,
                               id = "student", class_level = "group",
                               n_starts = 5, seed = 1)
curved <- mixture_regression(score ~ wave + I(wave^2), growth_scores,
                             n_classes = 3, id = "student", class_level = "group",
                             n_starts = 5, seed = 1)
compare_models(straight = straight, curved = curved)
#> Model comparison
#>
#> Model     Classes  Random   k    LogLik       BIC   ΔBIC  Weight  Entropy
#> --------  -------  ------  --  --------  --------  -----  ------  -------
#> straight        3  none    11  -5889.55  11841.85   0.00   1.000     0.92
#> curved          3  none    14  -5888.78  11857.41  15.56   0.000     0.92
```

These trajectories were generated as straight lines, and the BIC of the curved model is 15.6 points higher: the additional curvature is not supported.

```
plot(curved)
```



Line and dark band: class mean trajectory and its 95% confidence band.
Points: observed class means (posterior−weighted) with intervals.
Thin lines: persons, coloured by their most likely class.

Outcomes that are not scores

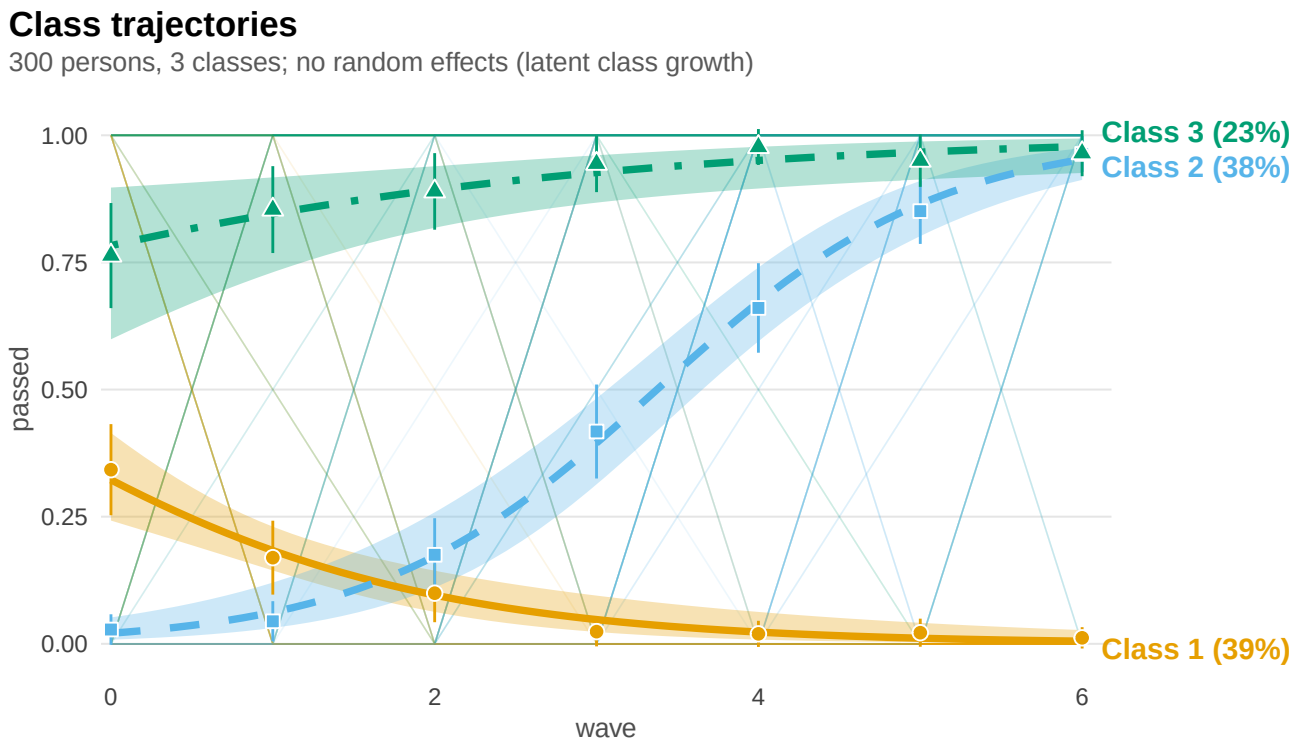
For outcomes that are not continuous scores, latent class growth analysis is the available model: random effects for a binary, count or ordered outcome require numerical integration, which the growth mixture model in latents does not perform. An alternative for trajectories of categorical states is to identify the state of each person at each occasion and model the resulting sequences of states (López-Pernas & Saqr, 2023, 2024).

Pass or fail. Whether each test reached 60 points, with a logistic trajectory per class:

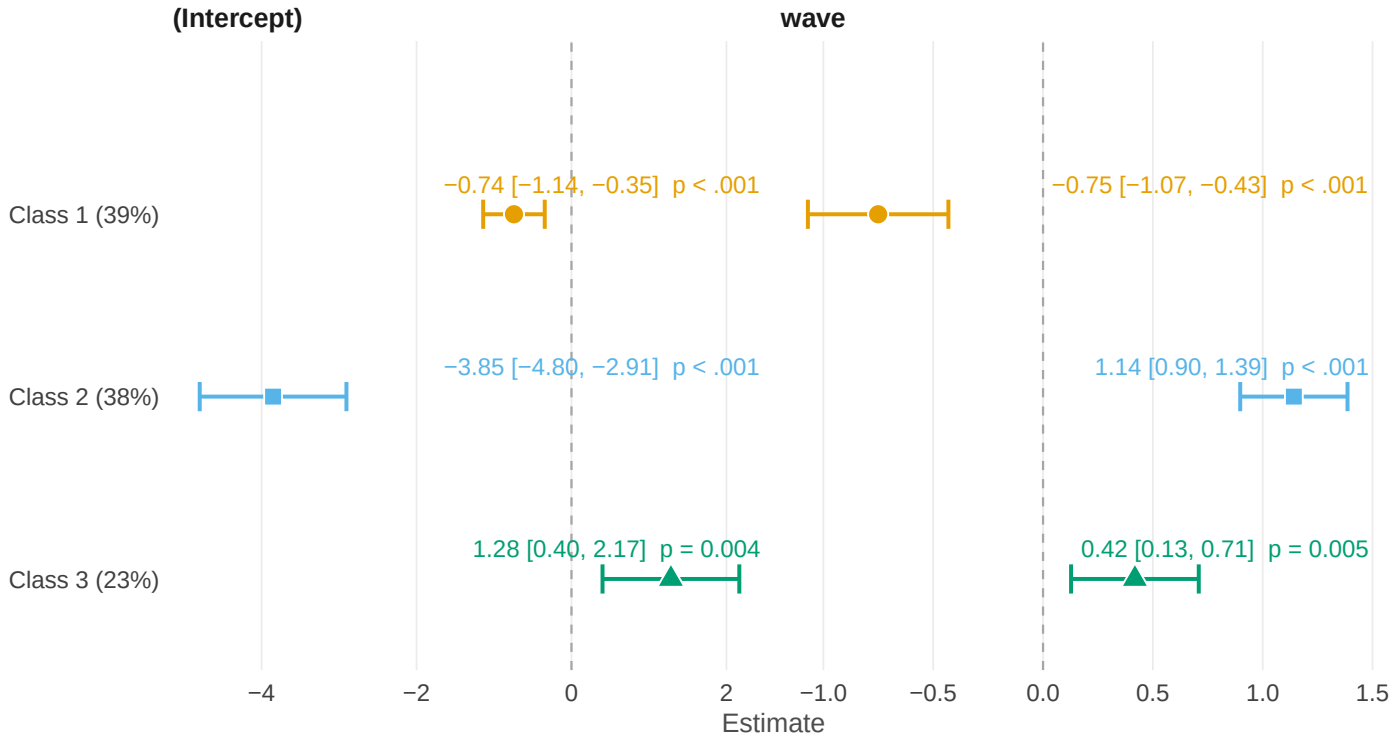
```
passing <- transform(growth_scores, passed = as.integer(score >= 60))
pass_fit <- mixture_regression(passed ~ wave, passing, n_classes = 3,
                               family = "binomial", id = "student",
                               class_level = "group", n_starts = 5, seed = 1)
get_results(pass_fit, "coefficients")
#> Regression coefficients (95% CI)
#>
#> Class    Term       Estimate          95% CI      p  Odds ratio
#> -------  ---------  --------  ---------------  -----  ----------
#> Class 1  Intercept     -0.74  [-1.14, -0.35]  <.001
#> Class 1  wave          -0.75  [-1.07, -0.43]  <.001        0.47
#> Class 2  Intercept     -3.85  [-4.80, -2.91]  <.001
#> Class 2  wave           1.14  [ 0.90,  1.39]  <.001        3.13
#> Class 3  Intercept      1.28  [ 0.40,  2.17]   .004
#> Class 3  wave           0.42  [ 0.13,  0.71]   .005        1.52
```

Each class's probability of passing over the waves, and the coefficients:

```
plot(pass_fit)
```




```
plot(pass_fit, what = "coefficients")
```



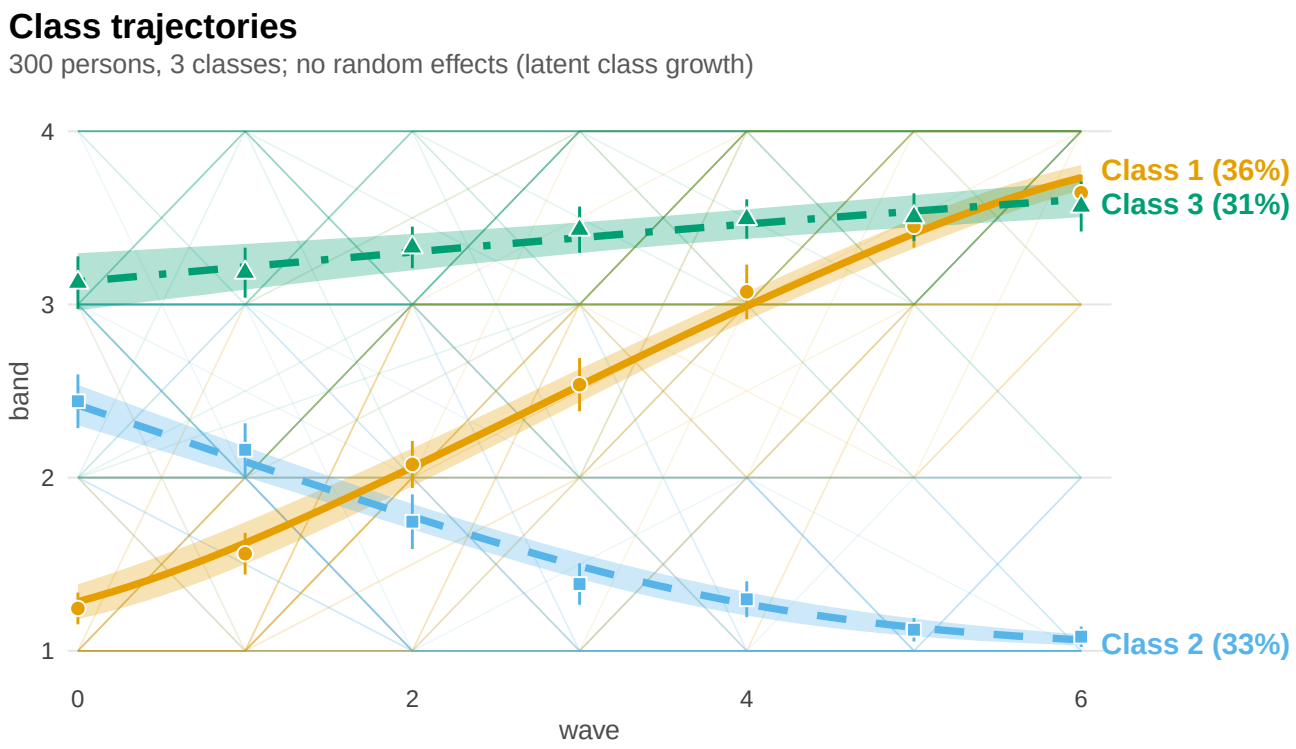
Ordered bands. The score in four ordered bands, with a cumulative-logit (proportional odds) trajectory per class (McCullagh, 1980): each class has its own thresholds between the bands and its own slope over the waves.

```
banded <- transform(growth_scores,
                    band = cut(score, c(-Inf, 52, 58, 64, Inf),
                               labels = c("low", "middle", "high", "top"),
                               ordered_result = TRUE))
band_fit <- mixture_regression(band ~ wave, banded, n_classes = 3,
                               family = "ordinal", id = "student",
                               class_level = "group", n_starts = 5, seed = 1)
get_results(band_fit, "coefficients")
#> Regression coefficients (95% CI)
#>
#> Class    Term                  Estimate            95% CI      p  Odds ratio
#> -------  --------------------  --------  -----------------  -----  ----------
#> Class 1  Threshold low|middle     1.107  [  0.663,  1.550]  <.001
#> Class 1  Threshold middle|high    3.396  [  2.781,  4.010]  <.001
#> Class 1  Threshold high|top       5.848  [  5.079,  6.617]  <.001
#> Class 1  wave                     1.168  [  1.012,  1.323]  <.001        3.21
#> Class 2  Threshold low|middle    -2.115  [ -2.484, -1.745]  <.001
#> Class 2  Threshold middle|high    0.100  [ -0.205,  0.404]   .520
#> Class 2  Threshold high|top       2.913  [  2.195,  3.630]  <.001
#> Class 2  wave                    -0.822  [ -0.951, -0.692]  <.001        0.44
#> Class 3  Threshold low|middle    -6.246  [-13.827,  1.335]   .106
```

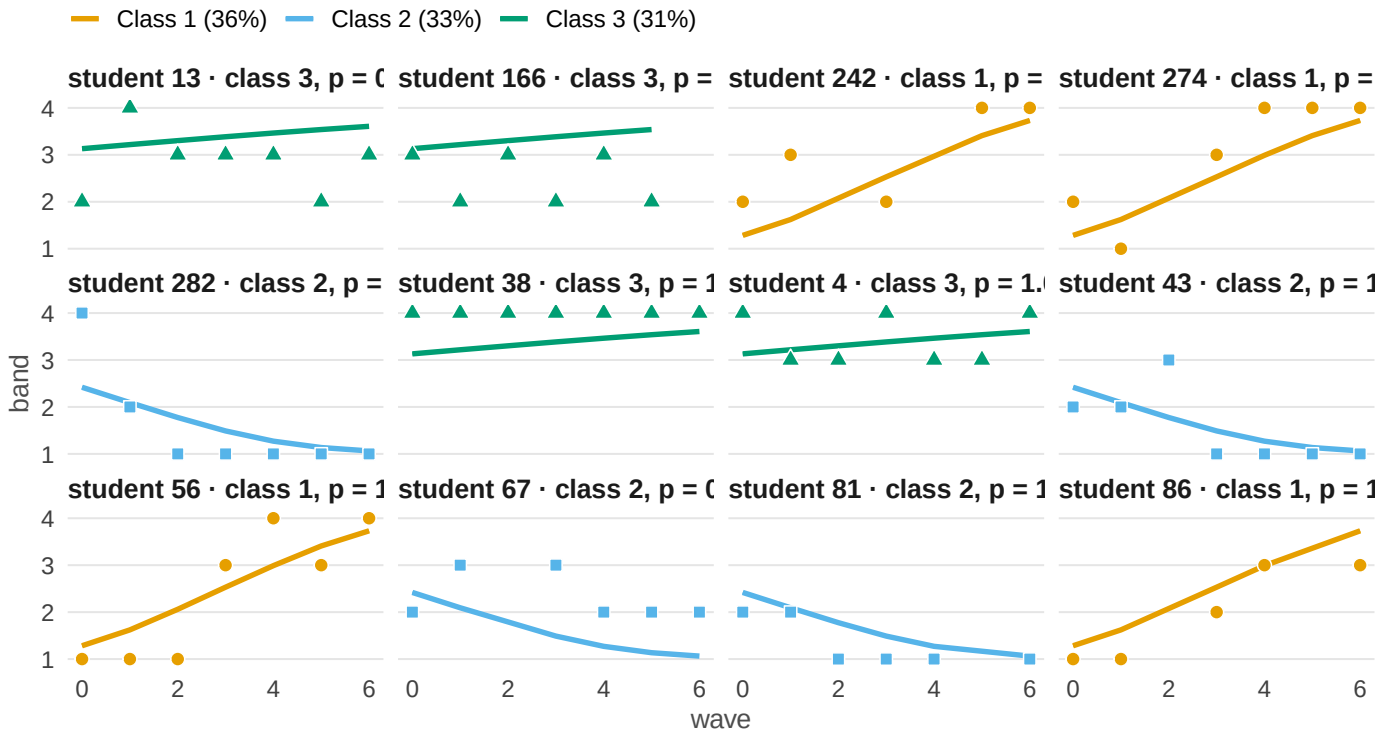
```
#> Class 3  Threshold middle|high    -1.538  [ -2.106, -0.971]  <.001
#> Class 3  Threshold high|top        0.802  [  0.403,  1.202]  <.001
#> Class 3  wave                      0.241  [  0.137,  0.346]  <.001        1.27
```

The plot shows each class's expected band (1 = low to 4 = top), and the students' own bands over the waves:

```
plot(band_fit)
```




```
plot(band_fit, what = "individuals")
```



Counts work the same way with `family = "poisson"`, or `family = "negative_binomial"` when the counts are more spread out than a Poisson allows.

Distal outcomes

`three_step()` compares the classes on a variable measured once per person, using the bias-corrected three-step method (Vermunt, 2010; Bakk & Vermunt, 2016), which accounts for persons being classified with error. The trajectory model must be fitted without `membership` for this correction:

```
three_step(straight, data = growth_scores, outcome = "motivation",
           contrast = "pairs")
#> Distal outcome: class differences (bch, 95% CI)
#>
#> Class  vs  Difference        95% CI      p  p (BH)
#> -----  --  ----------  ------------  -----  ------
#>     2   1        0.71  [0.46, 0.96]  <.001   <.001
#>     3   1        1.22  [0.96, 1.48]  <.001   <.001
#>     3   2        0.51  [0.24, 0.79]  <.001   <.001
```

All three pairs of classes differ in motivation (Benjamini-Hochberg adjusted p-values below .001).

Choices and warnings

- **Latent class growth analysis and growth mixture models.** When persons of the same kind vary around their trajectory, as they almost always do for scores, the growth mixture model is fitted as well and the two are compared with `compare_models()`; the growth mixture model is preferred when it fits better. Use latent class growth analysis for non-Gaussian outcomes, for a first look, and when within-class variation is not plausible.
- **Number of classes.** Read the BIC with the over-extraction above in mind; the bootstrap likelihood-ratio test (`enumerate_regressions(bootstrap =)`) simulates from the same model without random effects. It can also favour extra classes that absorb omitted within-class variation. Compare the random-effects model as well; the bootstrap does not repair misspecification of its null model.
- **Starts.** Several starts are used (`n_starts`), and the best likelihood should be reached more than once (the fit reports how many times).

References

Bakk, Z., & Vermunt, J. K. (2016). Robustness of stepwise latent class modeling with continuous distal outcomes. *Structural Equation Modeling*, 23, 20–31.

Bauer, D. J., & Curran, P. J. (2003). Distributional assumptions of growth mixture models: Implications for overextraction of latent trajectory classes. *Psychological Methods*, 8, 338–363.

López-Pernas, S., & Saqr, M. (2023). From variables to states to trajectories (VaSSTra): A method for modelling the longitudinal dynamics of learning and behaviour. In *Proceedings TEEM 2022: Tenth International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 1169–1178). Lecture Notes in Educational Technology. Springer. https://doi.org/10.1007/978-981-99-0942-1_123

López-Pernas, S., & Saqr, M. (2024). Modeling the dynamics of longitudinal processes in education: A tutorial with R for the VaSSTra method. In M. Saqr & S. López-Pernas (Eds.), *Learning analytics methods and tutorials: A practical guide using R* (pp. 355–379). Springer. https://doi.org/10.1007/978-3-031-54464-4_11

McCullagh, P. (1980). Regression models for ordinal data. *Journal of the Royal Statistical Society B*, 42, 109–142.

Nagin, D. S. (2005). *Group-based modeling of development*. Harvard University Press.

Saqr, M., Vogelsmeier, L. V. D. E., & López-Pernas, S. (2024). Capturing where the learning process takes place: A person-specific and person-centered primer. *Learning and Individual Differences*, 113, 102492. https://doi.org/10.1016/j.lindif.2024.102492

Scrucca, L., Saqr, M., López-Pernas, S., & Murphy, K. (2024). An introduction and R tutorial to model-based clustering in education via latent profile analysis. In M. Saqr & S. López-Pernas (Eds.), *Learning analytics methods and tutorials: A practical guide using R* (pp. 285–317). Springer. https://doi.org/10.1007/978-3-031-54464-4_9

Vermunt, J. K. (2010). Latent class modeling with covariates: Two improved three-step approaches. *Political Analysis*, 18, 450–469.