Class enumeration for finite mixture models with the latents R package

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

26 November 2026

Contents

Data . . . 3
Fitting the candidate models . . . 4
The table of candidates . . . 5
Information criteria . . . 8
Change with each added class . . . 11
Classification quality . . . 15
Checks of the estimation . . . 17
Comparing candidates . . . 19
Testing neighbouring candidates . . . 22
The choice . . . 23
Analysis of the chosen candidate . . . 23
Other model types . . . 47
Functions . . . 52
References . . . 53

Class enumeration refers to the process of determining the number of latent classes to be retained in an analysis. In latent profile analysis (LPA), these classes are referred to as *profiles*. The number of profiles is not estimated directly by the model; rather, it is specified a priori by the analyst. The model is then fitted to the data under the specified number of profiles, yielding a solution containing exactly that number. Class enumeration therefore involves fitting a sequence of models with progressively increasing numbers of profiles, typically beginning with a one-profile model, and evaluating their relative adequacy to determine which solution best represents the underlying structure of the data.

The selection of the number of profiles is consequential because it determines the structure and interpretation of the entire subsequent analysis. A model with too few profiles may combine substantively distinct groups of observations into a single profile, resulting in estimated means that adequately characterize none of the underlying groups. Conversely, a model with too many profiles may divide a genuine profile into smaller, less distinct groups that reflect sampling variability rather than meaningful heterogeneity. Additional profiles may also capture features of the measurement structure, such as residual correlations among indicators, rather than substantively distinct patterns of observations. Consequently, the number of profiles affects their estimated means, relative proportions, the assignment of observations to profiles, and the conclusions of subsequent analyses based on profile membership. In the data examined in this article, the two-profile and four-profile solutions provide different representations of the same enrolments. However, only one of these solutions corresponds to the data-generating structure, illustrating that a model with more profiles does not necessarily provide a more accurate representation of the underlying population.

In a two-level latent profile analysis, class enumeration concerns three distinct specifications, each of which must be determined by the analyst:

- **The number of profiles.** Profiles represent distinct patterns of observations at the lower level, in this article course enrolments, and are defined by their means on the indicators.
- **The number of group classes.** Group classes represent distinct types of groups at the upper level, in this article students, and are defined by the probability with which the observations of their groups belong to each profile. Because group classes are defined in terms of the profiles, a model with a single profile cannot contain more than one group class.
- **The covariance structure.** The covariance structure determines whether the indicators are allowed to be correlated within a profile and whether profiles may differ in the size, shape and orientation of their distributions (`EEI`, `VVI`, `EEE` and `VVV`; Celeux & Govaert, 1995).

Class enumeration involves fitting and comparing models across alternative specifications of the number of profiles, the number of group classes, and the model structures under consideration. The objective is to identify the model that provides the most adequate representation of the data while accounting for model complexity and the substantive characteristics of the resulting profiles and group classes.

Conventional likelihood ratio tests are generally not appropriate for comparing finite mixture models that differ in the number of profiles or group classes. In such comparisons, the model with fewer classes can be viewed as a constrained version of the more complex model in which the probability of an additional class is fixed at zero. This restriction places the parameter on the boundary of the parameter space, violating the regularity conditions required for the usual chi-squared approximation of the likelihood ratio statistic (McLachlan, 1987). Alternative procedures include the parametric bootstrap likelihood ratio test (McLachlan, 1987) and the Lo-Mendell-Rubin approximate likelihood ratio test (Lo et al., 2001), available for supported models through `bootstrap_lrt()` and `get_results(x, "lmr")`, respectively. These tests require repeated model estimation and, consequently, careful evaluation of convergence and estimation stability.

Information criteria provide a complementary approach to model comparison by balancing model fit, typically represented by the log likelihood, against a penalty for model complexity. The Akaike information criterion (AIC; Akaike, 1974), Bayesian information criterion (BIC; Schwarz, 1978), sample-size adjusted BIC (Sclove, 1987), consistent AIC (CAIC; Bozdogan, 1987), and Kullback information criterion (KIC; Cavanaugh, 1999) incorporate penalties related to the number of estimated parameters. In contrast, the integrated classification likelihood (ICL; Biernacki et al., 2000), approximate weight of evidence (AWE; Banfield & Raftery, 1993), and classification likelihood criterion (CLC; Biernacki & Govaert, 1997) additionally account for classification

uncertainty. These criteria therefore tend to favour solutions in which profiles and group classes are more clearly separated and observations can be assigned with greater certainty.

The various enumeration procedures do not necessarily identify the same model, and their relative performance depends on the characteristics of the model and the data. A simulation study by Nylund, Asparouhov, and Muthén (2007) found that the BIC and the bootstrap likelihood ratio test performed particularly well among the methods examined. Subsequent simulation research indicated that the AIC and relative entropy perform poorly in identifying the number of classes and that no criterion achieves adequate power when classes are weakly separated (Tein et al., 2013). Moreover, implementation of the Lo-Mendell-Rubin test differs between Mplus and Latent GOLD (Vermunt, 2024), highlighting the importance of considering software-specific procedures when interpreting test results.

Current recommendations therefore emphasize evaluating a range of candidate models, reporting the available fit indices and statistical tests, and integrating these results with classification diagnostics and substantive interpretability (Nylund-Gibson & Choi, 2018; Spurk et al., 2020; Weller et al., 2020; Sorgente et al., 2025). Entropy should be interpreted as a measure of classification quality for a given model rather than as a standalone criterion for determining the number of classes. The final selection should reflect the collective evidence from model fit, classification uncertainty, estimation stability, and the substantive meaningfulness of the resulting profiles and group classes. Houle et al. (2026) provide corresponding guidance for class enumeration in multilevel latent profile analysis.

These three specifications are not independent of one another and should therefore be evaluated jointly rather than sequentially. A diagonal covariance structure assumes that the indicators are uncorrelated within each profile. When this assumption is violated, the model can represent the residual association among the indicators only by introducing additional profiles, and the enumeration consequently overstates the number of profiles. In a similar manner, dependence among the observations of a group that the model does not represent, such as change across a sequence of courses, may be absorbed by additional group classes. A criterion that continues to decrease throughout the candidate range may reflect such departures from the model rather than a substantively meaningful set of profiles. For the number of group classes, the sample-size convention of the BIC is also consequential: the package reports criteria based on both the number of groups and the number of observations, and Lukočienė et al. (2010) recommend basing the choice of the number of group classes on the number of groups.

This vignette uses `enumerate_classes()` to fit a grid of two-level latent profile models over numbers of profiles, numbers of group classes and covariance structures. It examines information criteria, changes between adjacent candidates, descriptive curve summaries, classification diagnostics, and estimation diagnostics. Selected models are compared and a fitted candidate is extracted for further analysis. No single statistic is treated as a conclusive estimate of the number of classes.

```
library(latents)
```

Data

The examples use `course_engagement`, a simulated data frame in long format with one row per course enrolment. It contains 1,422 enrolments of 106 students. Five measures of activity in each course serve as indicators: browsing of the course site, lecture viewing, reading of the forum, posting to the forum and attendance. Each measure is standardized within its course, so a value of zero is the course average. The column `student` identifies the student to whom each enrolment belongs. The simulation is calibrated to the engagement data of students followed through a full study programme (Saqr & López-Pernas, 2021; Saqr et al., 2023). It was generated from two profiles and two kinds of students, and `attendance` is generated from the four click measures, so the indicators are correlated within profiles (`?course_engagement`).

```
head(course_engagement)
#>   student course sequence browse lectures forum_read forum_post attendance previous_grade
#> 1       1     23        1   0.73    -0.08       0.30       0.67       0.72           0.54
```

```
#> 2       1     19        2   0.68     0.53      -0.02      -0.55       0.70         -0.71
#> 3       1      4        3  -1.51    -0.51      -0.91      -0.80      -0.97          1.98
#> 4       1     18        4   0.64    -1.02      -1.62      -1.20      -1.26         -1.42
#> 5       1     16        5  -0.94    -0.37      -0.71      -0.28      -1.52         -0.67
#> 6       1     29        6  -0.36    -0.46      -0.77      -0.43      -1.34         -0.40
#>   engagement student_type
#> 1    engaged     wavering
#> 2    engaged     wavering
#> 3 disengaged     wavering
#> 4 disengaged     wavering
#> 5 disengaged     wavering
#> 6 disengaged     wavering
```

Enrolments are nested in students. The profiles therefore describe patterns of activity in a single course, and the group classes describe students by how their enrolments are distributed across those patterns.

Fitting the candidate models

The candidates are fitted with `enumerate_classes()` from the data, the indicators in `vars` and the group identifier in `id`. The arguments `n_profiles` and `n_group_classes` give the numbers of classes to try, and one model is fitted for every combination of them. Here the numbers of profiles and of group classes both run from one to four. The argument `model` sets the covariance structure of the profiles; without it, as here, the four basic structures `EEI`, `VVI`, `EEE` and `VVV` are fitted for every combination.

Mixture likelihoods can have multiple local maxima, so each candidate is estimated from several starting values. The `n_starts` argument controls the number of fully fitted starts and `seed` controls reproducibility. This example uses five starts per candidate to limit computation. An applied analysis should assess whether increasing the number of starts changes the selected likelihood or substantive conclusions.

Every estimation argument, such as `n_starts`, applies identically to all candidates, and a grid is re-estimated in its entirety when more starts are required, so that every candidate has the same opportunity to reach its maximum. When many starts are used, a candidate with more profiles than the data support can reach a degenerate solution in which one profile collapses on a single observation with every variance at its lower bound; its likelihood, and hence its criteria, then depend on `min_variance` rather than on the data. A profile can also be spurious without reaching the bound, when a few nearly coincident observations give it variances close to zero (McLachlan & Peel, 2000, section 3.10).

A Gaussian component can concentrate around very few observations and approach a variance bound. The `boundary = "exclude"` option excludes starts with variances at the configured bound; it does not detect every narrowly concentrated component. The `latents_thin_profiles` warning flags limited posterior support relative to the number of profile-specific Gaussian parameters. This is a diagnostic heuristic rather than an identification test. Inspect component sizes and covariance diagnostics before interpreting such a profile.

The default search depends on the covariance specification. EII, EEI and EEE models use emEM screening: five times `n_starts` starts are screened by a short EM run each, and the first start and the best of the others, `n_starts` in all, are continued to convergence (Biernacki et al., 2003). The other structures start from k-means, because with profile-specific variances emEM reaches spurious maxima more often. The `search` argument can override these defaults. The `cores` argument permits parallel candidate fitting.

```
activity <- c("browse", "lectures", "forum_read", "forum_post", "attendance")
models <- enumerate_classes(course_engagement, activity, "student",
                            n_profiles = 1:4, n_group_classes = 1:4,
                            n_starts = 5, seed = 1)
#> 7 of 52 candidates reached their best log likelihood from one start only (profiles4_groups4_EEE, profiles4_g
```

The result records 64 requested candidates. Each candidate is named by its numbers of profiles and group classes and its covariance structure, so that `profiles2_groups4_VVV` has two profiles, four group classes and a full covariance per profile; the name identifies the candidate in every function that follows. The 12 candidates with one profile and several group classes could not be estimated. Group classes are defined by differences in the distribution of profiles, and a model with a single profile has no such differences. These candidates remain in the table with the error that stopped them.

The table of candidates

The table of candidates has one row per candidate and is returned by `as.data.frame()`; printing `models` shows the same table with a summary of failed and unreplicated candidates. The 16 candidates with the `VVV` structure:

```
subset(as.data.frame(models), model == "VVV")
#>            candidate n_profiles n_group_classes model log_likelihood n_parameters  aic  kic
#> 49 profiles1_groups1_VVV        1              1   VVV        -8710         20 17460 17483
#> 50 profiles2_groups1_VVV        2              1   VVV        -8490         41 17061 17105
#> 51 profiles3_groups1_VVV        3              1   VVV        -8472         62 17067 17132
#> 52 profiles4_groups1_VVV        4              1   VVV        -8434         83 17033 17119
#> 53 profiles1_groups2_VVV        1              2   VVV           NA         NA    NA    NA
#> 54 profiles2_groups2_VVV        2              2   VVV        -8313         43 16712 16758
#> 55 profiles3_groups2_VVV        3              2   VVV        -8289         65 16708 16776
#> 56 profiles4_groups2_VVV        4              2   VVV        -8261         87 16696 16786
#> 57 profiles1_groups3_VVV        1              3   VVV           NA         NA    NA    NA
#> 58 profiles2_groups3_VVV        2              3   VVV        -8295         45 16681 16729
#> 59 profiles3_groups3_VVV        3              3   VVV        -8275         68 16686 16757
#> 60 profiles4_groups3_VVV        4              3   VVV        -8237         91 16656 16750
#> 61 profiles1_groups4_VVV        1              4   VVV           NA         NA    NA    NA
#> 62 profiles2_groups4_VVV        2              4   VVV        -8289         47 16672 16722
#> 63 profiles3_groups4_VVV        3              4   VVV        -8262         71 16666 16740
#> 64 profiles4_groups4_VVV        4              4   VVV        -8231         95 16652 16750
#>    bic_groups bic_individual sabic_groups sabic_individual caic_groups caic_individual awe_groups
#> 49      17514          17565       17450           17502       17534          17585       17667
#> 50      17170          17277       17041           17147       17211          17318       17732
#> 51      17232          17393       17036           17196       17294          17455       18344
#> 52      17254          17470       16992           17206       17337          17553       18315
#> 53         NA             NA          NA              NA          NA             NA          NA
#> 54      16827          16938       16691           16802       16870          16981       17346
#> 55      16881          17050       16676           16844       16946          17115       17593
#> 56      16927          17153       16652           16877       17014          17240       18366
#> 57         NA             NA          NA              NA          NA             NA          NA
#> 58      16801          16917       16658           16775       16846          16962       17360
#> 59      16867          17044       16652           16828       16935          17112       17958
#> 60      16899          17135       16611           16846       16990          17226       18195
#> 61         NA             NA          NA              NA          NA             NA          NA
#> 62      16798          16920       16649           16770       16845          16967       17380
#> 63      16855          17040       16631           16814       16926          17111       17636
#> 64      16905          17151       16604           16849       17000          17246       17910
#>    awe_individual icl_groups icl_individual clc_groups clc_individual profile_entropy group_entropy
#> 49          17771      17514          17565      17420          17420             NA            NA
#> 50          17945      17418          17524      17226          17226          0.875            NA
#> 51          18666      17869          18030      17580          17580          0.796            NA
#> 52          18746      17679          17894      17292          17292          0.892            NA
```

```
#> 53           NA       NA           NA       NA           NA           NA           NA
#> 54        17570    17017        17128    16816        16816        0.906        0.942
#> 55        17931    17095        17264    16792        16792        0.933        0.943
#> 56        18818    17699        17925    17294        17294        0.805        0.946
#> 57           NA       NA           NA       NA           NA           NA           NA
#> 58        17593    17015        17132    16805        16805        0.909        0.816
#> 59        18312    17437        17614    17120        17120        0.828        0.824
#> 60        18667    17497        17734    17073        17073        0.855        0.837
#> 61           NA       NA           NA       NA           NA           NA           NA
#> 62        17624    17020        17142    16801        16801        0.912        0.797
#> 63        18005    17092        17276    16761        16761        0.940        0.801
#> 64        18403    17182        17429    16739        16739        0.936        0.830
#>   avepp_min avepp_max avepp_mean avepp_below group_avepp_min group_avepp_max group_avepp_mean
#> 49        NA        NA         NA          NA              NA              NA               NA
#> 50     0.959     0.971      0.965           0              NA              NA               NA
#> 51     0.665     0.956      0.833           1              NA              NA               NA
#> 52     0.630     0.958      0.837           1              NA              NA               NA
#> 53        NA        NA         NA          NA              NA              NA               NA
#> 54     0.973     0.978      0.976           0           0.971           0.993            0.982
#> 55     0.796     0.976      0.915           1           0.979           0.990            0.985
#> 56     0.802     0.974      0.856           0           0.962           0.995            0.979
#> 57        NA        NA         NA          NA              NA              NA               NA
#> 58     0.974     0.978      0.976           0           0.903           0.960            0.923
#> 59     0.650     0.973      0.841           1           0.878           0.955            0.920
#> 60     0.649     0.967      0.838           1           0.881           0.976            0.928
#> 61        NA        NA         NA          NA              NA              NA               NA
#> 62     0.977     0.978      0.977           0           0.830           0.925            0.874
#> 63     0.830     0.979      0.928           0           0.851           0.910            0.879
#> 64     0.785     0.976      0.887           1           0.868           0.961            0.921
#>    group_avepp_below converged boundary min_standardized_eigenvalue min_gaussian_mass
#> 49                NA      TRUE    FALSE                     0.279287           1422.0
#> 50                NA      TRUE    FALSE                     0.269614            577.4
#> 51                NA      TRUE    FALSE                     0.120317            112.7
#> 52                NA      TRUE    FALSE                     0.000276             26.1
#> 53                NA     FALSE       NA                           NA               NA
#> 54                 0      TRUE    FALSE                     0.271042            583.2
#> 55                 0      TRUE    FALSE                     0.011064             18.6
#> 56                 0      TRUE    FALSE                     0.001958             26.3
#> 57                NA     FALSE       NA                           NA               NA
#> 58                 0      TRUE    FALSE                     0.272051            589.5
#> 59                 0      TRUE    FALSE                     0.120469            116.5
#> 60                 0      TRUE    FALSE                     0.002424             26.9
#> 61                NA     FALSE       NA                           NA               NA
#> 62                 0      TRUE    FALSE                     0.271143            590.0
#> 63                 0      TRUE    FALSE                     0.003469             17.2
#> 64                 0      TRUE    FALSE                     0.013442             31.2
#>    n_best_replicated
#> 49                 5
#> 50                 5
#> 51                 4
#> 52                 1
#> 53                NA
#> 54                 5
```

```
#> 55                  1
#> 56                  1
#> 57                 NA
#> 58                  5
#> 59                  4
#> 60                  1
#> 61                 NA
#> 62                  5
#> 63                  1
#> 64                  1
#>
#> 49
#> 50
#> 51
#> 52
#> 53
#> 54
#> 55 Profile 2: 18.6 Gaussian membership mass for 20 free means and variances. This is a conservative diagnosti
#> 56
#> 57
#> 58
#> 59
#> 60
#> 61
#> 62
#> 63 Profile 2: 17.2 Gaussian membership mass for 20 free means and variances. This is a conservative diagnosti
#> 64
#>                                                                                                      error
#> 49                                                                                                    <NA>
#> 50                                                                                                    <NA>
#> 51                                                                                                    <NA>
#> 52                                                                                                    <NA>
#> 53 Multiple group classes are not identifiable with one profile or only singleton groups.
#> 54                                                                                                    <NA>
#> 55                                                                                                    <NA>
#> 56                                                                                                    <NA>
#> 57 Multiple group classes are not identifiable with one profile or only singleton groups.
#> 58                                                                                                    <NA>
#> 59                                                                                                    <NA>
#> 60                                                                                                    <NA>
#> 61 Multiple group classes are not identifiable with one profile or only singleton groups.
#> 62                                                                                                    <NA>
#> 63                                                                                                    <NA>
#> 64                                                                                                    <NA>
```

The columns fall into groups.

Columns	Content
`candidate`, `n_profiles`, `n_group_classes`, `model`	Identity of the candidate
`log_likelihood`, `n_parameters`	Fit and size
`aic`, `kic`, `bic_*`, `sabic_*`, `caic_*`, `awe_*`, `icl_*`, `clc_*`	Information criteria
`profile_entropy`, `group_entropy`	Relative entropy at each level

Columns	Content
`avepp_min`, `avepp_max`, `avepp_mean`, `avepp_below`	Average posterior probability of the profiles
`group_avepp_min`, `group_avepp_max`, `group_avepp_mean`, `group_avepp_below`	Average posterior probability of the group classes
`converged`, `boundary`, `n_best_replicated`, `warnings`, `error`	Checks of the estimation

Information criteria

Many information criteria combine deviance (minus twice the maximized log likelihood) with a penalty for the number of estimated parameters. Some criteria also penalize classification uncertainty. Lower values are preferred under the chosen criterion. In a two-level model, several parameter penalties depend on the sample-size convention. Criteria ending in `_groups` count the 106 students as the independent units, and criteria ending in `_individual` count the 1,422 enrolments. An analysis uses one of the two conventions throughout.

ICL and AWE include uncertainty about both the students' group classes and their enrolments' profiles under either convention. Changing the sample-size convention changes the parameter penalty, not which classification uncertainty is included.

The candidate at the minimum of each criterion:

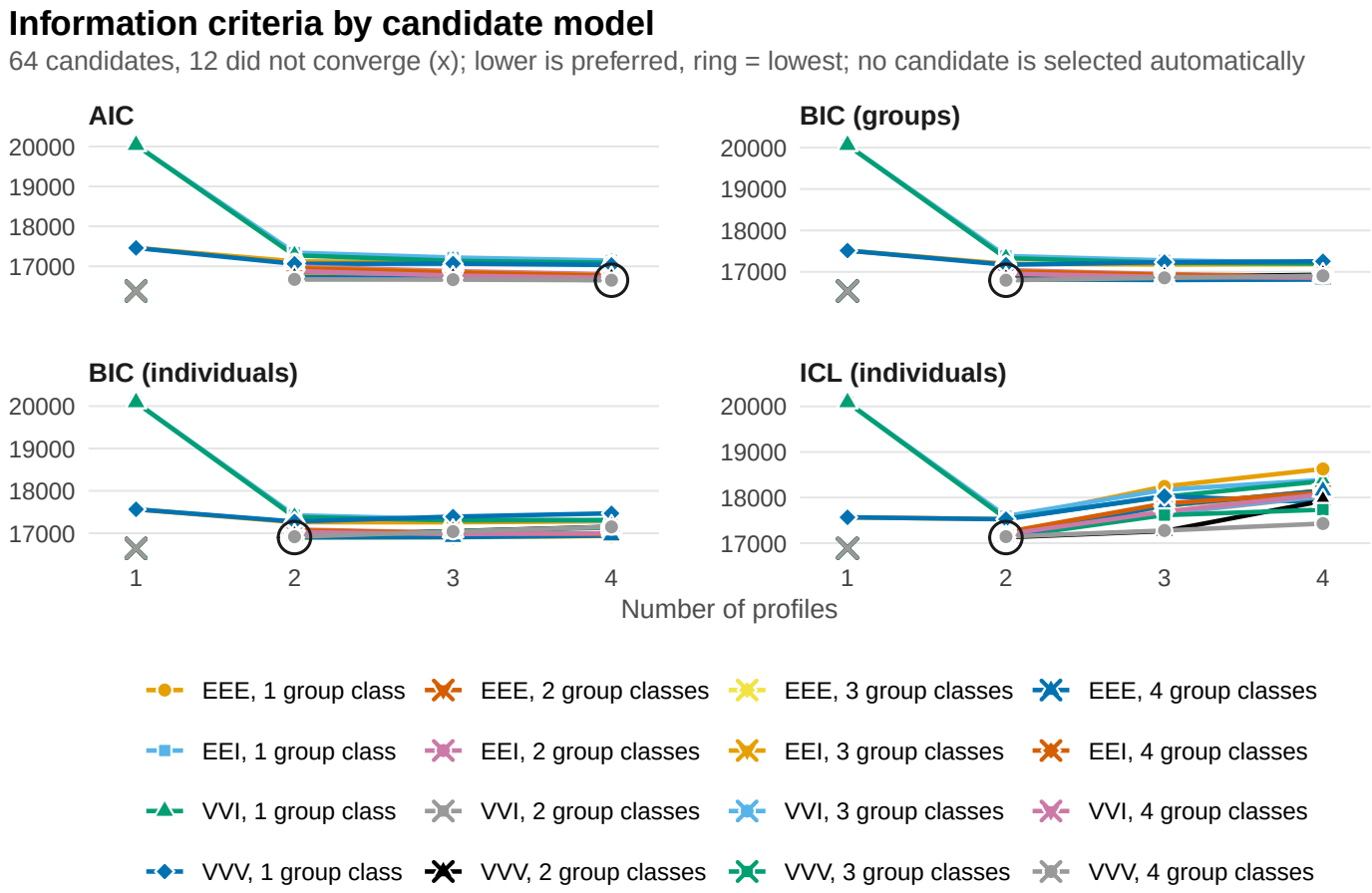
```
get_results(models, "best")
#>    criterion  convention             candidate n_profiles n_group_classes model value
#> 1        aic        <NA> profiles4_groups4_VVV          4               4   VVV 16652
#> 2        kic        <NA> profiles2_groups4_VVV          2               4   VVV 16722
#> 3        bic      groups profiles2_groups4_VVV          2               4   VVV 16798
#> 4        bic individuals profiles2_groups3_EEE          2               3   EEE 16900
#> 5      sabic      groups profiles4_groups4_VVV          4               4   VVV 16604
#> 6      sabic individuals profiles2_groups4_VVV          2               4   VVV 16770
#> 7       caic      groups profiles3_groups3_EEE          3               3   EEE 16844
#> 8       caic individuals profiles2_groups3_EEE          2               3   EEE 16930
#> 9        awe      groups profiles2_groups2_EEE          2               2   EEE 17267
#> 10       awe individuals profiles2_groups2_VVI          2               2   VVI 17412
#> 11       icl      groups profiles2_groups3_VVV          2               3   VVV 17015
#> 12       icl individuals profiles2_groups3_EEE          2               3   EEE 17122
#> 13       clc      groups profiles4_groups4_VVV          4               4   VVV 16739
#> 14       clc individuals profiles4_groups4_VVV          4               4   VVV 16739
```

Most criteria select two profiles; the AIC, the CLC and the SABIC with students as the units, whose parameter penalties are weakest or absent, select four, and the CAIC with students as the units three. They agree that the indicators are correlated within profiles: all but one select a structure with covariances (`EEE` or `VVV`). They disagree on the number of group classes. With students as the units, the BIC is lowest for profiles2_groups4_VVV, the ICL for profiles2_groups3_VVV and the AWE for profiles2_groups2_EEE; with enrolments as the units, the BIC is lowest for profiles2_groups3_EEE. The AIC is lowest for profiles4_groups4_VVV, the largest candidate fitted.

The covariance structure matters for the number of profiles. Among the diagonal `VVI` candidates alone, the BIC with students as the units is lowest for profiles4_groups3_VVI (BIC 16,852): with the within-profile correlation of attendance excluded, two further profiles absorb it. Two profiles with free covariances and two group classes fit better (BIC 16,827).

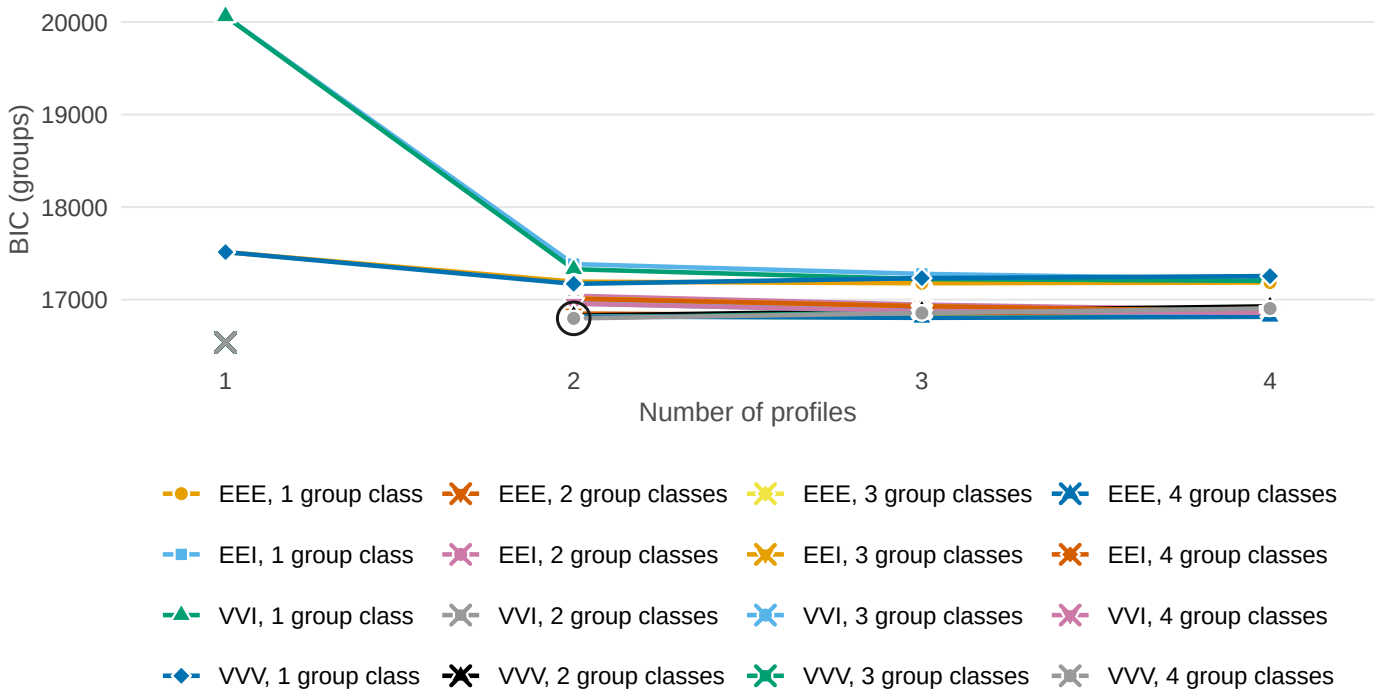
The criteria are plotted against the number of profiles, one line per number of group classes, for every structure, with the lowest value of each criterion ringed.

```
plot_enumeration(models)
```



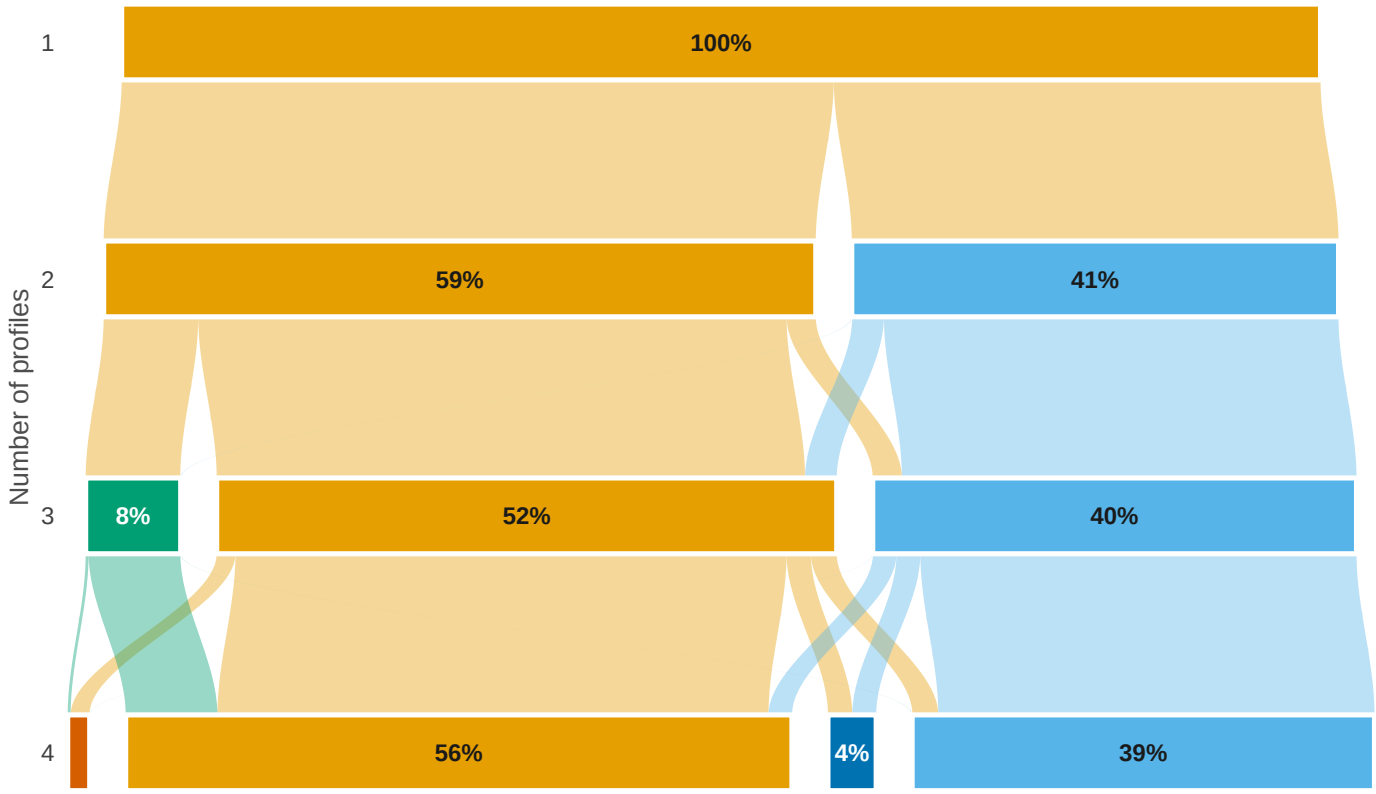
A minimum at the largest candidate leaves uncertainty about models beyond the fitted range. Consider a wider, scientifically plausible range alongside alternative measurement and covariance specifications. An interior minimum is not guaranteed. Any criterion can be plotted by name; `what = "all"` displays every criterion.

```
plot_enumeration(models, what = "bic_groups")
```



The tree summarizes correspondence between profiles across independently fitted models (Zappia & Oshlack, 2018). Links are weighted by shared posterior membership mass, computed from products of the models' membership probabilities, rather than by assuming that profiles form nested partitions. Each number of group classes and structure has its own tree; `n_group_classes` and `model` select one, and `plot_tree(models)` returns them all, one plot each. The guide *Tree plot* compares solutions at one number of profiles and plots the tree of group classes.

```
plot_tree(models, n_group_classes = 1, model = "VVV")
```



A band that forks: a class split in two (nested, stable solutions).
Bands that merge or cross: units shared between classes, by reassignment or by uncertain classification.

Change with each added class

When a criterion decreases across the whole candidate range, successive changes help describe how much each additional class improves that criterion. For every candidate, the table reports the decrease in BIC from the candidate with one profile fewer and the same number of group classes and structure, and the decrease from the candidate with one group class fewer. A positive value means that the added class lowered the criterion. The argument `using` selects another criterion.

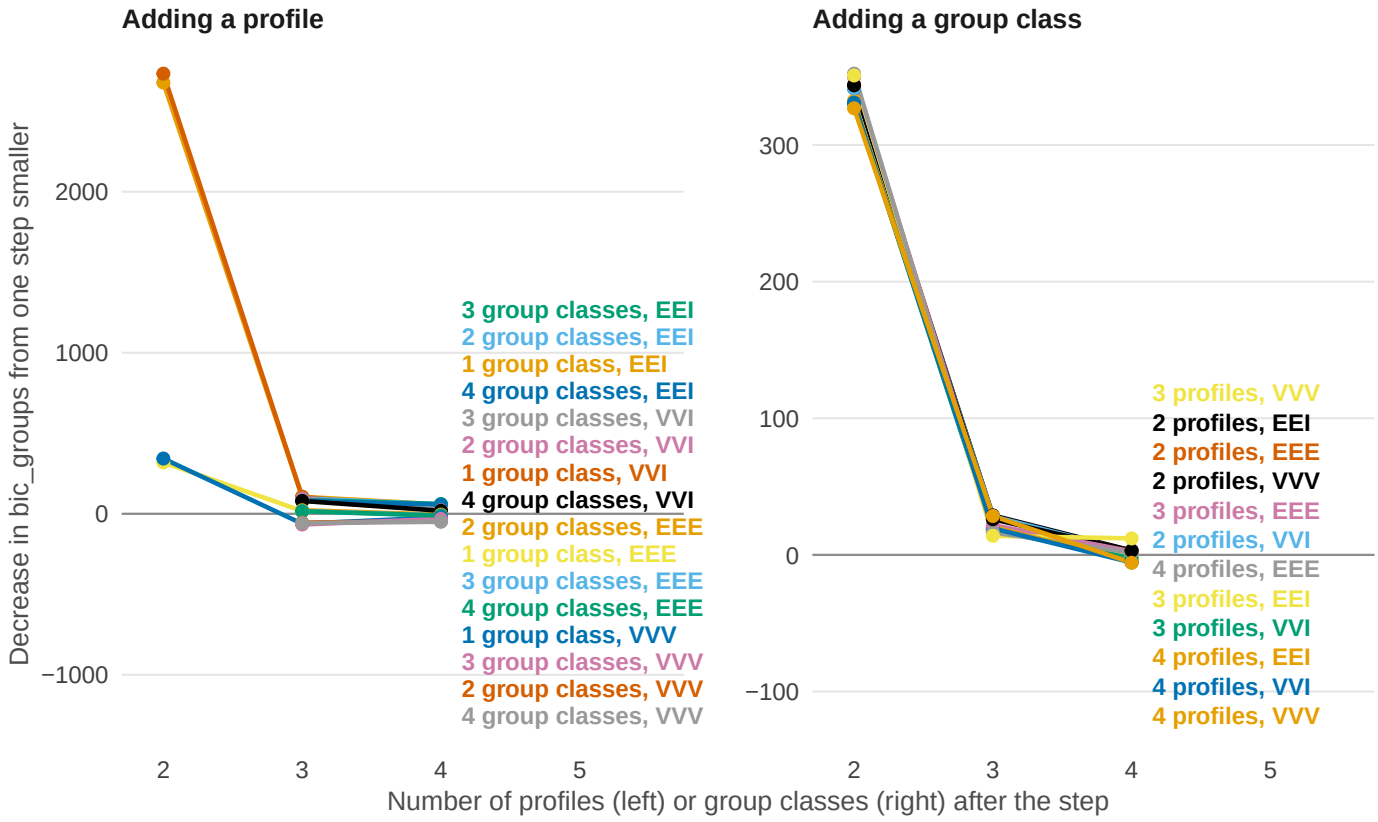
The rows of the `VVV` candidates:

```
subset(get_results(models, "deltas"), model == "VVV")
#>                candidate n_profiles n_group_classes model  criterion value delta_profile
#> 49 profiles1_groups1_VVV          1               1   VVV bic_groups 17514            NA
#> 50 profiles2_groups1_VVV          2               1   VVV bic_groups 17170         343.1
#> 51 profiles3_groups1_VVV          3               1   VVV bic_groups 17232         -61.7
#> 52 profiles4_groups1_VVV          4               1   VVV bic_groups 17254         -22.0
```

```
#> 53 profiles1_groups2_VVV          1               2   VVV bic_groups    NA            NA
#> 54 profiles2_groups2_VVV          2               2   VVV bic_groups 16827            NA
#> 55 profiles3_groups2_VVV          3               2   VVV bic_groups 16881         -54.5
#> 56 profiles4_groups2_VVV          4               2   VVV bic_groups 16927         -46.0
#> 57 profiles1_groups3_VVV          1               3   VVV bic_groups    NA            NA
#> 58 profiles2_groups3_VVV          2               3   VVV bic_groups 16801            NA
#> 59 profiles3_groups3_VVV          3               3   VVV bic_groups 16867         -66.6
#> 60 profiles4_groups3_VVV          4               3   VVV bic_groups 16899         -31.5
#> 61 profiles1_groups4_VVV          1               4   VVV bic_groups    NA            NA
#> 62 profiles2_groups4_VVV          2               4   VVV bic_groups 16798            NA
#> 63 profiles3_groups4_VVV          3               4   VVV bic_groups 16855         -57.6
#> 64 profiles4_groups4_VVV          4               4   VVV bic_groups 16905         -49.3
#>             from_profile delta_group_class      from_group_class
#> 49                 <NA>                NA                  <NA>
#> 50 profiles1_groups1_VVV                NA                  <NA>
#> 51 profiles2_groups1_VVV                NA                  <NA>
#> 52 profiles3_groups1_VVV                NA                  <NA>
#> 53                 <NA>                NA                  <NA>
#> 54                 <NA>            343.67 profiles2_groups1_VVV
#> 55 profiles2_groups2_VVV            350.88 profiles3_groups1_VVV
#> 56 profiles3_groups2_VVV            326.86 profiles4_groups1_VVV
#> 57                 <NA>                NA                  <NA>
#> 58                 <NA>             26.13 profiles2_groups2_VVV
#> 59 profiles2_groups3_VVV             13.99 profiles3_groups2_VVV
#> 60 profiles3_groups3_VVV             28.46 profiles4_groups2_VVV
#> 61                 <NA>                NA                  <NA>
#> 62                 <NA>              3.06 profiles2_groups3_VVV
#> 63 profiles2_groups4_VVV             12.06 profiles3_groups3_VVV
#> 64 profiles3_groups4_VVV             -5.74 profiles4_groups3_VVV
```

For the VVV structure with one group class, a second profile lowers the BIC by 343 and a third raises it by 62, which appears in the table as a negative decrease. With two profiles, a second group class lowers the BIC by 344, a third by 26 and a fourth by 3. The decreases are plotted for both directions, along the number of profiles on the left and along the number of group classes on the right.

```
plot_selection(models, what = "deltas")
```



The knee summary scales both axes of a curve to the unit interval and finds the candidate with the largest perpendicular distance below the straight line from the first point to the last. This chord-distance rule is inspired by Kneedle (Satopää et al., 2011), without its smoothing or sensitivity thresholds. A nearly linear curve can have a very small maximum distance, so the reported distance describes the strength of the bend as well as its location. The knee is computed along the number of profiles for each number of group classes, along the number of group classes for each number of profiles, and across all candidates ordered by their number of parameters, separately for each structure.

The lines of the `VVV` candidates:

```
subset(get_results(models, "knees"), endsWith(line, "VVV"))
#>        direction                line  criterion                 from                   to
#> 13     profiles   1 group class, VVV bic_groups profiles1_groups1_VVV profiles4_groups1_VVV
#> 14     profiles 2 group classes, VVV bic_groups profiles2_groups2_VVV profiles4_groups2_VVV
#> 15     profiles 3 group classes, VVV bic_groups profiles2_groups3_VVV profiles4_groups3_VVV
#> 16     profiles 4 group classes, VVV bic_groups profiles2_groups4_VVV profiles4_groups4_VVV
#> 29 group_classes      1 profile, VVV bic_groups profiles1_groups1_VVV profiles1_groups1_VVV
#> 30 group_classes     2 profiles, VVV bic_groups profiles2_groups1_VVV profiles2_groups4_VVV
#> 31 group_classes     3 profiles, VVV bic_groups profiles3_groups1_VVV profiles3_groups4_VVV
#> 32 group_classes     4 profiles, VVV bic_groups profiles4_groups1_VVV profiles4_groups4_VVV
#> 36   parameters  all candidates, VVV bic_groups profiles1_groups1_VVV profiles4_groups4_VVV
#>    n_points        knee_candidate distance  knee_without_largest stable
#> 13        4 profiles2_groups1_VVV    0.597 profiles2_groups1_VVV   TRUE
```

```
#> 14        3                   <NA>       NA                   <NA>     NA
#> 15        3                   <NA>       NA                   <NA>     NA
#> 16        3                   <NA>       NA                   <NA>     NA
#> 29        1                   <NA>       NA                   <NA>     NA
#> 30        4 profiles2_groups2_VVV    0.416 profiles2_groups2_VVV   TRUE
#> 31        4 profiles3_groups2_VVV    0.423 profiles3_groups2_VVV   TRUE
#> 32        4 profiles4_groups2_VVV    0.422 profiles4_groups2_VVV   TRUE
#> 36       13 profiles2_groups3_VVV    0.542 profiles2_groups3_VVV   TRUE
#>                                          note
#> 13
#> 14 the criterion does not fall along this line
#> 15 the criterion does not fall along this line
#> 16 the criterion does not fall along this line
#> 29                 fewer than three candidates
#> 30
#> 31
#> 32
#> 36
```

Along the number of group classes, the knee lies at two group classes for every number of profiles from two to four and every structure; with two profiles and `VVV` it is profiles2_groups2_VVV, and it does not move when the largest candidate is removed (`stable`). Along the number of profiles, the BIC of the `VVV` candidates with two or more group classes does not fall beyond two profiles, so those lines have no knee (`note`).

The heatmap displays each candidate's criterion value relative to the minimum across the fitted grid:

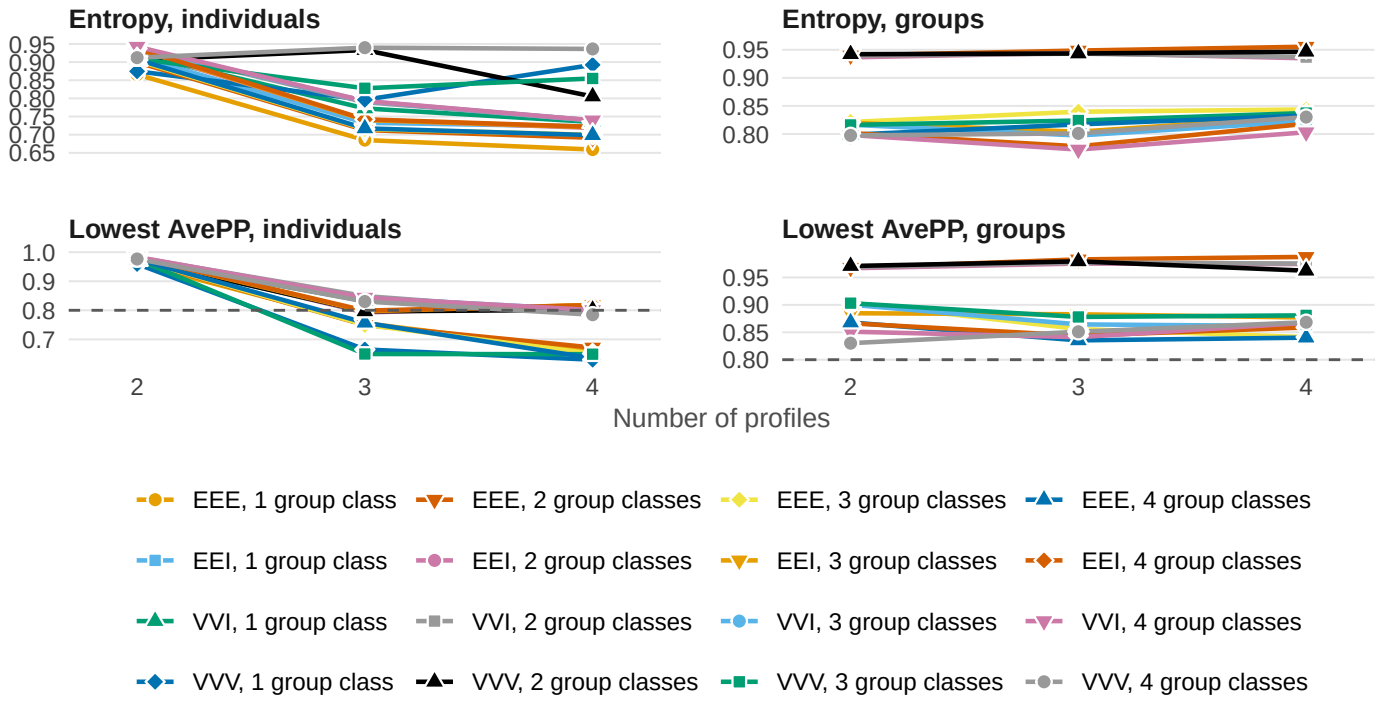
```
plot_selection(models, what = "heatmap")
```



Classification quality

Every observation has a posterior probability of membership in each class. Relative entropy summarizes these probabilities over all observations, from 0 when every observation is equally likely to belong to every class, to 1 when every observation belongs to one class with certainty (Ramaswamy et al., 1993). The average posterior probability (AvePP) of a class is the mean posterior probability of that class among the observations assigned to it. Higher AvePP values indicate greater posterior certainty within the assigned class. The candidate table reports the lowest, highest and mean AvePP at each level and the number of classes below a user-selected cutoff (`cutoff`, default 0.8). This threshold is a descriptive screening choice, not a validated criterion for selecting the number of classes.

```
plot_selection(models, what = "diagnostics")
```



`get_results(models, "classification")` returns the AvePP of every class of every candidate, with its size and the relative entropy of its level; for the `VVV` candidates with two profiles:

```
subset(get_results(models, "classification"), n_profiles == 2 & model == "VVV")
#>                 candidate n_profiles n_group_classes model       level class n_modal
#> 190 profiles2_groups1_VVV          2               1   VVV individuals     1     846
#> 191 profiles2_groups1_VVV          2               1   VVV individuals     2     576
#> 199 profiles2_groups2_VVV          2               2   VVV individuals     1     842
#> 200 profiles2_groups2_VVV          2               2   VVV individuals     2     580
#> 201 profiles2_groups2_VVV          2               2   VVV      groups     1      35
#> 202 profiles2_groups2_VVV          2               2   VVV      groups     2      71
#> 214 profiles2_groups3_VVV          2               3   VVV individuals     1     836
#> 215 profiles2_groups3_VVV          2               3   VVV individuals     2     586
#> 216 profiles2_groups3_VVV          2               3   VVV      groups     1      31
#> 217 profiles2_groups3_VVV          2               3   VVV      groups     2      25
#> 218 profiles2_groups3_VVV          2               3   VVV      groups     3      50
#> 232 profiles2_groups4_VVV          2               4   VVV individuals     1     839
#> 233 profiles2_groups4_VVV          2               4   VVV individuals     2     583
#> 234 profiles2_groups4_VVV          2               4   VVV      groups     1      14
#> 235 profiles2_groups4_VVV          2               4   VVV      groups     2      25
#> 236 profiles2_groups4_VVV          2               4   VVV      groups     3      21
#> 237 profiles2_groups4_VVV          2               4   VVV      groups     4      46
#>     proportion_modal estimated_n estimated_proportion average_posterior odds_correct_classification
#> 190            0.595      844.6                0.594             0.971                       22.7
#> 191            0.405      577.4                0.406             0.959                       34.6
#> 199            0.592      838.8                0.590             0.978                       30.6
```

```
#> 200          0.408      583.2              0.410          0.973              52.3
#> 201          0.330       34.5              0.325          0.971              69.2
#> 202          0.670       71.5              0.675          0.993              68.6
#> 214          0.588      832.5              0.585          0.978              31.2
#> 215          0.412      589.5              0.415          0.974              53.6
#> 216          0.292       30.8              0.290          0.960              58.4
#> 217          0.236       26.4              0.249          0.905              28.5
#> 218          0.472       48.8              0.460          0.903              10.9
#> 232          0.590      832.0              0.585          0.977              29.6
#> 233          0.410      590.0              0.415          0.978              63.7
#> 234          0.132       12.2              0.115          0.830              37.5
#> 235          0.236       24.1              0.228          0.862              21.2
#> 236          0.198       21.7              0.204          0.878              27.9
#> 237          0.434       48.0              0.453          0.925              15.0
#>     relative_entropy
#> 190            0.875
#> 191            0.875
#> 199            0.906
#> 200            0.906
#> 201            0.942
#> 202            0.942
#> 214            0.909
#> 215            0.909
#> 216            0.816
#> 217            0.816
#> 218            0.816
#> 232            0.912
#> 233            0.912
#> 234            0.797
#> 235            0.797
#> 236            0.797
#> 237            0.797
```

Separation of the students declines as group classes are added. With two profiles and `VVV`, the group-level relative entropy is 0.942 with two group classes, 0.816 with three and 0.797 with four, and the lowest group-class AvePP falls from 0.971 to 0.83.

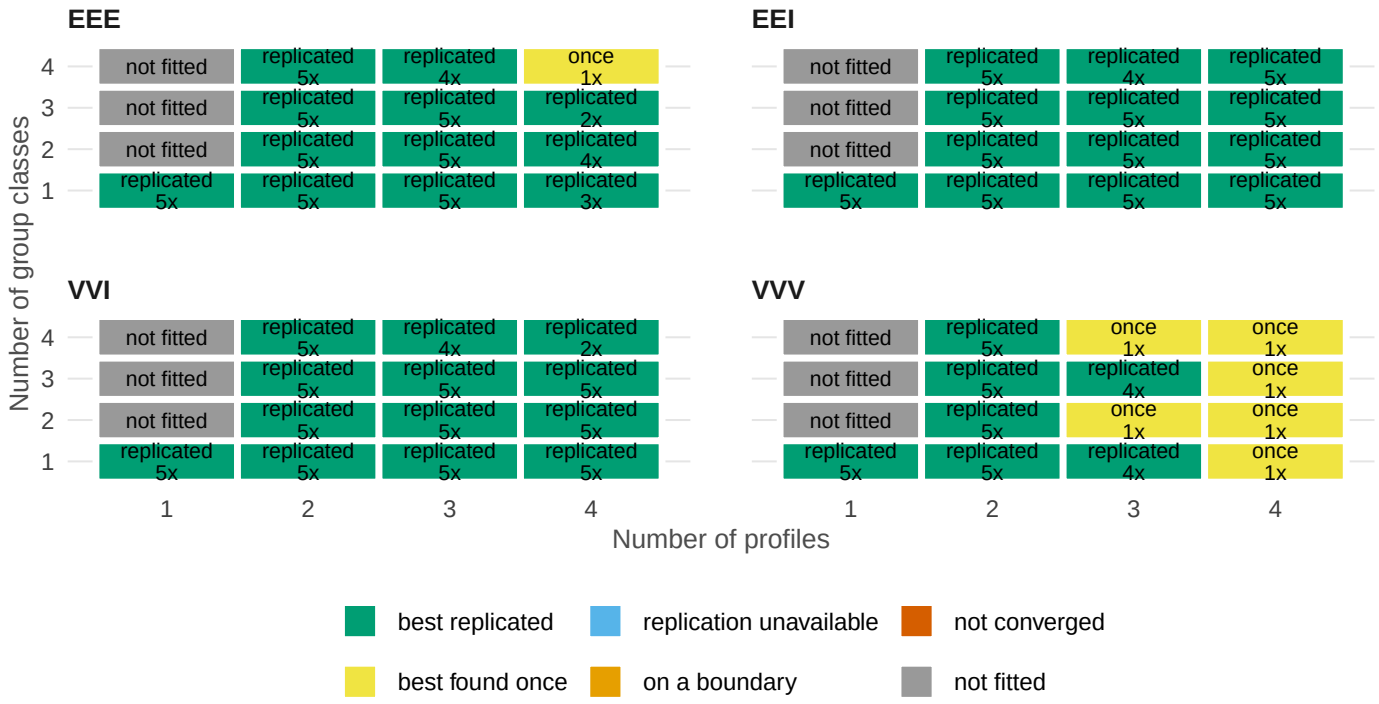
Checks of the estimation

Replication of the selected likelihood across starts provides evidence of numerical stability, but cannot establish that the global maximum has been found. Boundary estimates and small profiles warrant investigation because they can indicate a solution driven by limited support or a restrictive parameter bound. Their interpretation depends on the fitted model and data. Refitting with `boundary = "exclude"` selects among starts whose continuous variance parameters are above their lower bounds. This option changes start selection and does not by itself establish that the retained profiles are substantively meaningful. The following plots summarize these diagnostics.

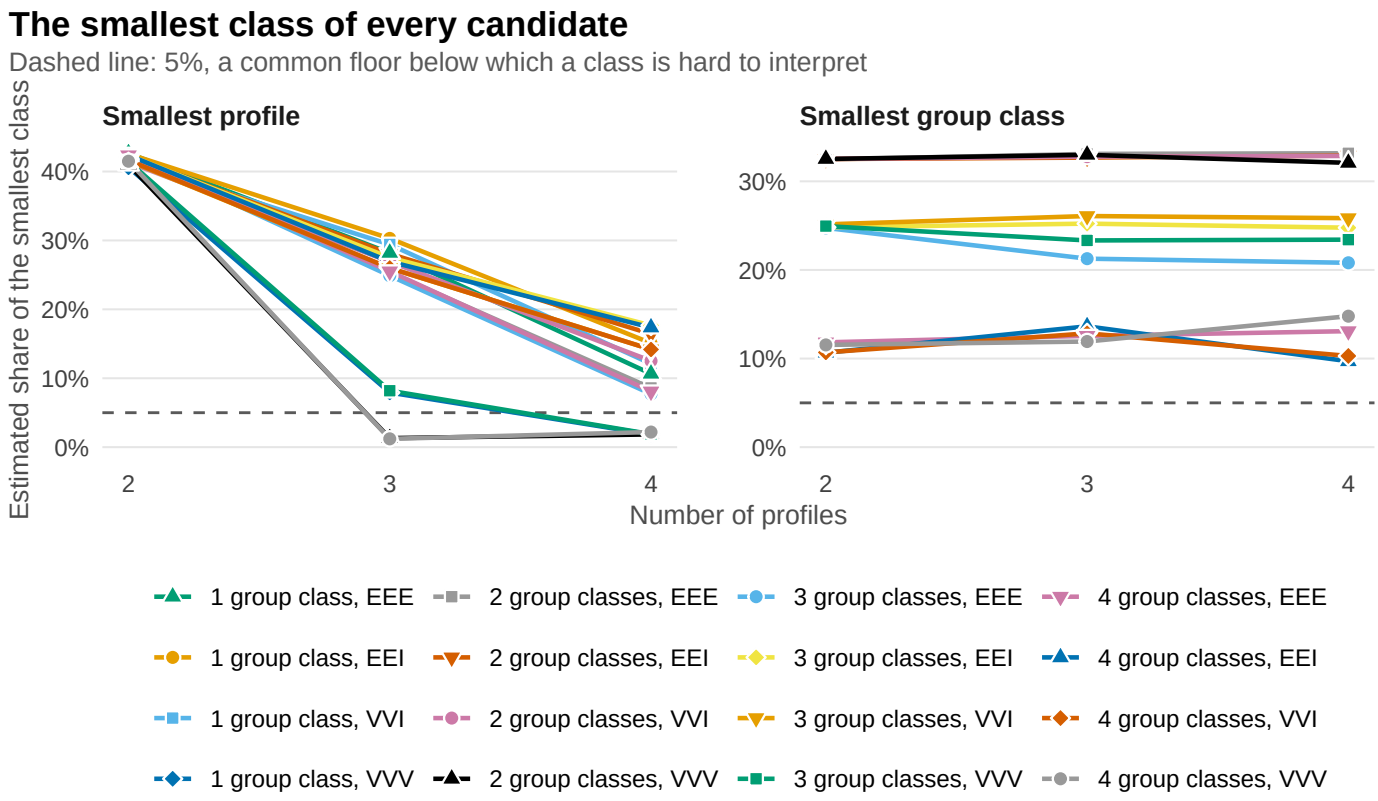
```
plot_selection(models, what = "trust")
```

Convergence and repeated maxima

Tile: the status, and how many random starts reached the best log likelihood. Replication does not prove a global maxim




```
plot_selection(models, what = "smallest")
```



All of the plots for choosing the number of classes are produced together with `plot_selection(models)`.

A candidate on a boundary whose smallest profile holds about one observation is a degenerate solution, not a profile: a Gaussian profile centred on one observation raises the likelihood without limit as its variances shrink, so the reported likelihood is fixed by `min_variance`. The default, `boundary = "allow"`, reports it, as Mplus does, with a warning. `boundary = "exclude"` retains for each candidate the best start with no variance at its lower bound, and is advisable when many starts are used, since each additional start increases the chance of reaching such a solution. Categorical indicators are unaffected.

```
excluded <- enumerate_classes(course_engagement, activity, "student",
                              n_profiles = 1:4, n_group_classes = 1:4,
                              n_starts = 200, boundary = "exclude", seed = 1)
```

Comparing candidates

Selected candidates are placed side by side with `compare_models()`. Candidates are named as in the table or selected by a number of profiles or group classes. The comparison adds the difference in BIC from the lowest of the candidates compared and the BIC weight. The weight is `exp(-delta_bic / 2)` normalized over the compared candidates (Wagenmakers & Farrell, 2004). It summarizes relative support under this BIC approximation and depends on the candidate set; it should not be interpreted as a calibrated probability that a model is correct.

The candidates preferred by BIC, ICL and AWE with students as the units:

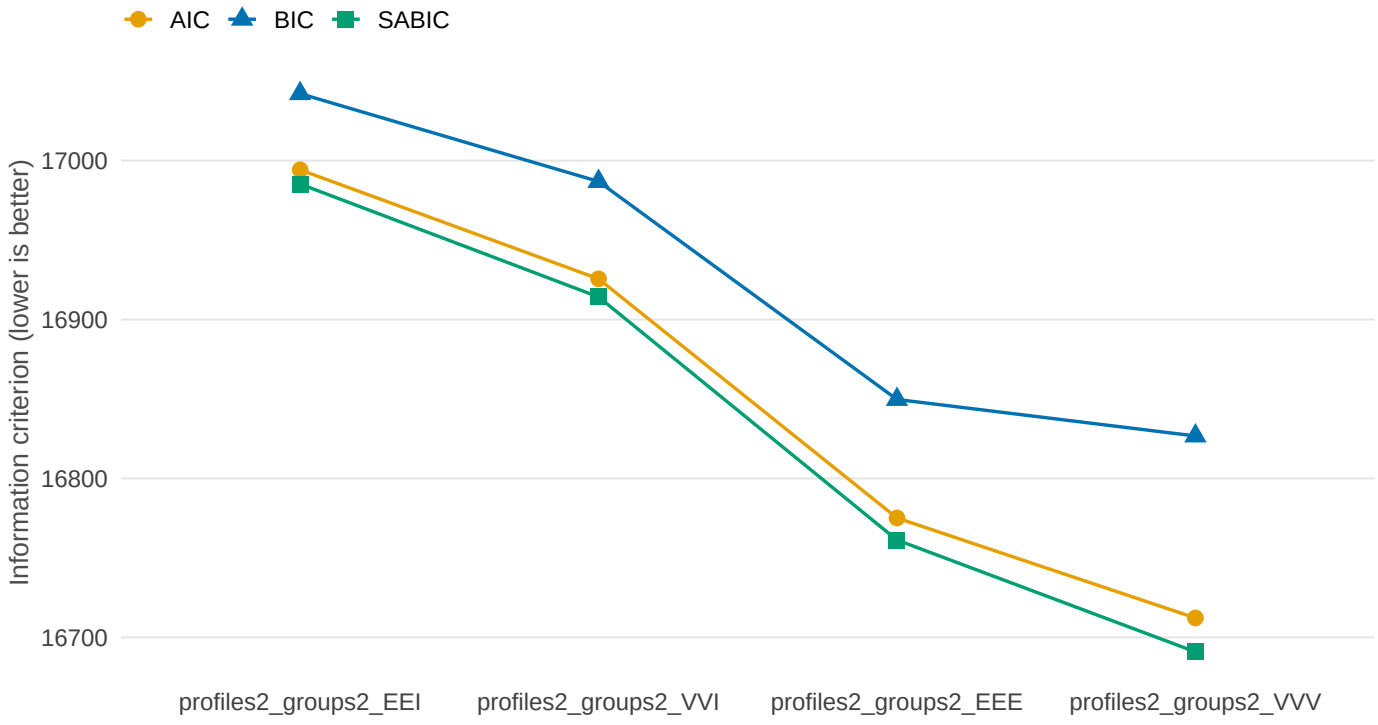
```
compare_models(models, "profiles2_groups4_VVV", "profiles2_groups3_VVV",
               "profiles2_groups2_EEE")
#> Model comparison
#>
#> Model            Profiles  Group classes  k   LogLik     BIC  ΔBIC  Weight     ICL  Entropy  Min AvePP  < 0.8
#> -------------------- -------- ------------- -- -------- -------- ----- ------ -------- ------- ----
#> profiles2_groups4_VVV      2         4 47 -8289.19 16797.56  0.00  0.822 17142.10    0.91     0.98    0
#> profiles2_groups3_VVV      2         3 45 -8295.38 16800.62  3.06  0.178 17131.51    0.91     0.97    0
#> profiles2_groups2_EEE      2         2 28 -8359.58 16849.73 52.17  0.000 17125.20    0.90     0.97    0
```

The same comparison with labels, and with the candidate analysed below:

```
compare_models(models, bic = "profiles2_groups4_VVV",
               icl = "profiles2_groups3_VVV", awe = "profiles2_groups2_EEE",
               chosen = "profiles2_groups2_VVV")
#> Model comparison
#>
#> Model  Profiles  Group classes  k   LogLik     BIC  ΔBIC  Weight     ICL  Entropy  Min AvePP  < 0.8  Group ent
#> ------ -------- ------------- -- -------- -------- ----- ------ -------- ------- --------- ----- ---
#> bic        2         4 47 -8289.19 16797.56  0.00  0.822 17142.10    0.91     0.98    0        0.80
#> icl        2         3 45 -8295.38 16800.62  3.06  0.178 17131.51    0.91     0.97    0        0.82
#> awe        2         2 28 -8359.58 16849.73 52.17  0.000 17125.20    0.90     0.97    0        0.94
#> chosen      2         2 43 -8313.11 16826.75 29.19  0.000 17128.35    0.91     0.97    0        0.94
```

The four structures with two profiles and two group classes, which shows what the covariance structure adds:

```
comparison <- compare_models(models, n_profiles = 2, n_group_classes = 2)
comparison
#> Model comparison
#>
#> Model            Profiles  Group classes  k   LogLik     BIC   ΔBIC  Weight     ICL  Entropy  Min AvePP  < 0.
#> -------------------- -------- ------------- -- -------- -------- ------ ------ -------- ------- ---
#> profiles2_groups2_EEI      2         2 18 -8479.12 17042.18 215.42  0.000 17228.54    0.93     0.98
#> profiles2_groups2_VVI      2         2 23 -8439.82 16986.89 160.14  0.000 17175.69    0.94     0.98
#> profiles2_groups2_EEE      2         2 28 -8359.58 16849.73  22.98  0.000 17125.20    0.90     0.97
#> profiles2_groups2_VVV      2         2 43 -8313.11 16826.75   0.00  1.000 17128.35    0.91     0.97
plot(comparison)
```



The comparison keeps the fitted models. The AvePP of every class of every compared model is returned by `get_results()`, and the model behind a row by `get_candidate()`.

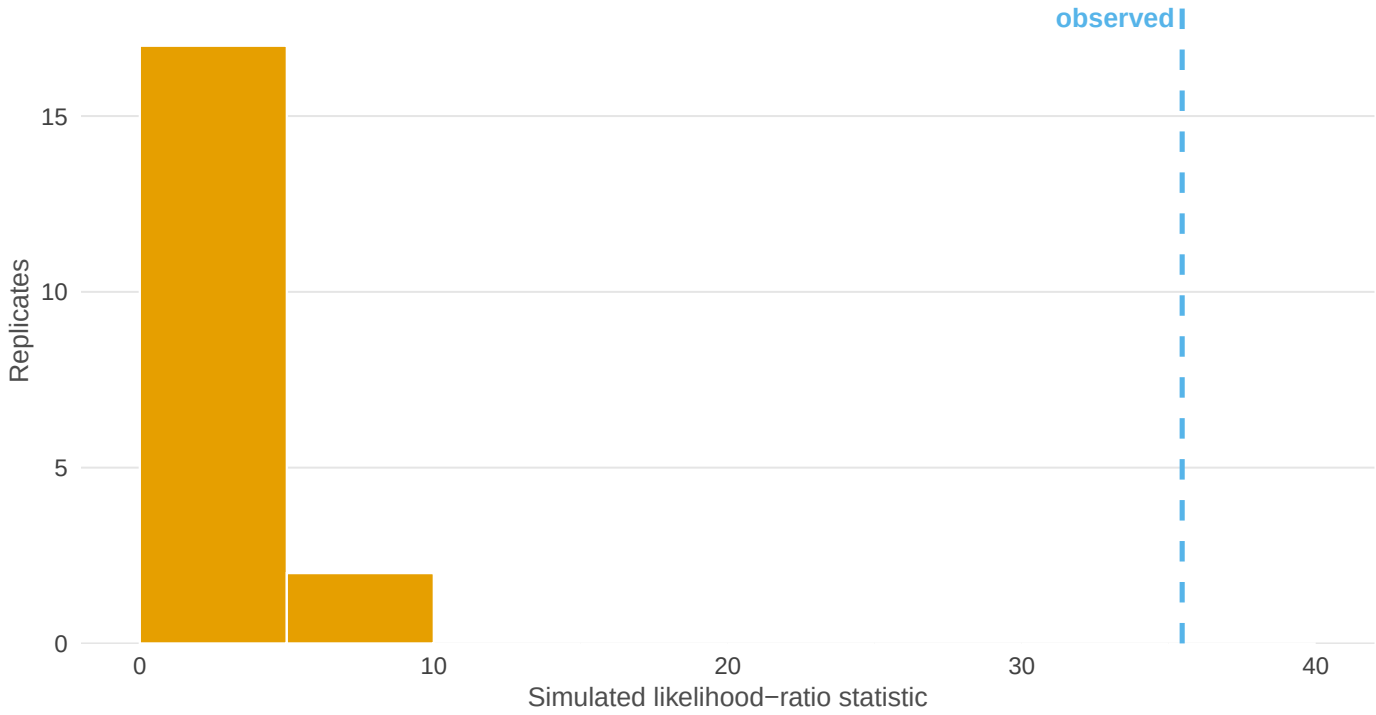
```
get_results(comparison, "classification")
#>                    model       level class n_modal proportion_modal estimated_n
#> 1  profiles2_groups2_EEI individuals     1     825            0.580       821.2
#> 2  profiles2_groups2_EEI individuals     2     597            0.420       600.8
#> 3  profiles2_groups2_EEI      groups     1      35            0.330        34.5
#> 4  profiles2_groups2_EEI      groups     2      71            0.670        71.5
#> 5  profiles2_groups2_VVI individuals     1     836            0.588       834.1
#> 6  profiles2_groups2_VVI individuals     2     586            0.412       587.9
#> 7  profiles2_groups2_VVI      groups     1      35            0.330        34.5
#> 8  profiles2_groups2_VVI      groups     2      71            0.670        71.5
#> 9  profiles2_groups2_EEE individuals     1     826            0.581       822.3
#> 10 profiles2_groups2_EEE individuals     2     596            0.419       599.7
#> 11 profiles2_groups2_EEE      groups     1      35            0.330        34.4
#> 12 profiles2_groups2_EEE      groups     2      71            0.670        71.6
#> 13 profiles2_groups2_VVV individuals     1     842            0.592       838.8
#> 14 profiles2_groups2_VVV individuals     2     580            0.408       583.2
#> 15 profiles2_groups2_VVV      groups     1      35            0.330        34.5
#> 16 profiles2_groups2_VVV      groups     2      71            0.670        71.5
#>    estimated_proportion average_posterior odds_correct_classification relative_entropy
#> 1                 0.578             0.982                        40.3            0.933
#> 2                 0.422             0.982                        73.3            0.933
#> 3                 0.325             0.967                        60.5            0.937
#> 4                 0.675             0.991                        53.8            0.937
```

```
#> 5                 0.587            0.985                        46.1            0.938
#> 6                 0.413            0.982                        76.4            0.938
#> 7                 0.325            0.969                        64.8            0.940
#> 8                 0.675            0.992                        59.9            0.940
#> 9                 0.578            0.975                        28.7            0.900
#> 10                0.422            0.972                        47.3            0.900
#> 11                0.325            0.968                        62.4            0.940
#> 12                0.675            0.992                        61.0            0.940
#> 13                0.590            0.978                        30.6            0.906
#> 14                0.410            0.973                        52.3            0.906
#> 15                0.325            0.971                        69.2            0.942
#> 16                0.675            0.993                        68.6            0.942
```

Testing neighbouring candidates

The bootstrap likelihood ratio test (McLachlan, 1987) compares two candidates that differ by one class. Data sets are simulated from the smaller candidate and both candidates are re-estimated on each. Of `B` requested data sets, let `V` pass the estimation checks and `b` of those yield a likelihood ratio at least as large as the observed one. The p-value is `(b + 1) / (V + 1)` (Davison & Hinkley, 1997); the additional count includes the observed statistic and precludes a p-value of zero. Invalid replicates are reported, and the p-value is withheld when fewer than half the requested replicates are valid. The test below compares two and three group classes with two profiles and the `VVV` structure, the decision on which the criteria disagree. Nineteen data sets keep the example brief; if all pass, the smallest attainable p-value is $1/20 = 0.05$. Applied analyses should use at least 99 and inspect the valid count.

```r
smaller <- get_candidate(models, "profiles2_groups2_VVV")
larger <- get_candidate(models, "profiles2_groups3_VVV")
test <- bootstrap_lrt(smaller, larger, data = course_engagement,
                      iter = 19, n_starts = 2, seed = 1)
test
#> Parametric bootstrap likelihood-ratio comparison
#> Null: 2 profiles, 2 group classes; alternative: 2 profiles, 3 group classes
#> Observed statistic: 35.457791
#> p-value: 0.05 (Monte Carlo SE 0.0487) from 19 of 19 valid replicates (0 invalid)
plot(test)
```



All 19 simulated data sets are valid, and none yields a likelihood ratio as large as the observed one, so the p-value is 0.05, the smallest attainable with this number of data sets. The result therefore favours three group classes over two, in agreement with the BIC, at the smallest p-value that 19 bootstrap samples can give.

The choice

The evidence points to two profiles with free within-profile covariances. For the group classes, the bootstrap test and the BIC indicate that the data contain structure beyond two group classes. The magnitude of that structure is small: the second group class accounts for almost all of the improvement in the BIC (344 against 26 and 3 for the third and fourth), the knee lies at two group classes for every structure, and further group classes separate the students less clearly. The analysis therefore continues with profiles2_groups2_VVV, whose BIC is 29 above the lowest. The small improvements from further group classes have an identifiable source: a student's engagement changes along the course sequence, a dependence between the enrolments of a student that the multilevel model does not represent and that additional group classes partly absorb. A latent transition model represents it directly (see the latent transition analysis article).

Analysis of the chosen candidate

The candidate is taken from the fitted candidates by name with `get_candidate()`. The numbers of classes with the structure give the same model, as in `get_candidate(models, 2, 2, model = "VVV")`.

```r
chosen <- get_candidate(models, "profiles2_groups2_VVV")
chosen
#> Two-level latent profile analysis: 2 profiles, 2 group classes
#> 1422 individuals in 106 groups; varying full residual covariance (VVV)
#> Log likelihood: -8313.111554 | AIC: 16712.223 | BIC (groups): 16826.751
#> Converged: TRUE | iterations: 12 | best log likelihood reached by 5 of 5 starts (5 converged)
#> Smallest standardized Gaussian eigenvalue: 0.271 (profiles, profile_2); Gaussian mass: 583.2; weight effect
#>
#> Class sizes   profile_1 59.0% (839)   profile_2 41.0% (583)
#> Group classes group_class_1 32.5% (34)   group_class_2 67.5% (72)
#>
#> Means
#>   indicator profile_1 profile_2
#>      browse     0.531    -0.764
#>    lectures     0.422    -0.607
#>  forum_read     0.613    -0.881
#>  forum_post     0.527    -0.758
#>  attendance     0.632    -0.908
#>
#> Variances and standard errors: get_results(x, "profiles").
#> Every other table: get_results(x, what = ), or get_results(x, "all").
```

`chosen` is the same kind of object that `multilpa()` returns. Its summary reports the estimates with their standard errors; `rows` limits the rows printed from each table.

```r
print(summary(chosen), rows = 3)
#> Two-level latent profile analysis: 2 profiles, 2 group classes
#> Individuals: 1422; groups: 106; parameters: 43; converged: TRUE
#> Log likelihood: -8313.111554; AIC: 16712.223
#> BIC (groups): 16826.751; BIC (individuals): 16938.395
#> Best likelihood replicated in 5/5 starts (absolute tolerance 0.00711).
#>
#> -- profiles ------------------------------------------------------
#>  profile  indicator  mean variance standard_deviation mean_standard_error variance_standard_error
#>        1     browse 0.5313   0.6012           0.7754           0.02804              0.03084
#>        1   lectures 0.4221   0.8459           0.9197           0.03310              0.04256
#>        1 forum_read 0.6128   0.4921           0.7015           0.02592              0.02578
#>    ... 7 more rows.  get_results(x, what = "profiles")
#>
#> -- responses -----------------------------------------------------
#>    (no rows)
#>
#> -- covariances ---------------------------------------------------
#>  profile  indicator indicator_2 covariance
#>        1     browse      browse   0.601215
#>        1   lectures      browse   0.001435
#>        1 forum_read      browse   0.026721
#>    ... 47 more rows.  get_results(x, what = "covariances")
#>
#> -- gaussian_diagnostics ------------------------------------------
#>   context   profile gaussian_mass positive_rows weight_effective_n min_variance_ratio
#>  profiles profile_1         838.8          1422             865.7             0.4184
#>  profiles profile_2         583.2          1422             610.5             0.3675
```

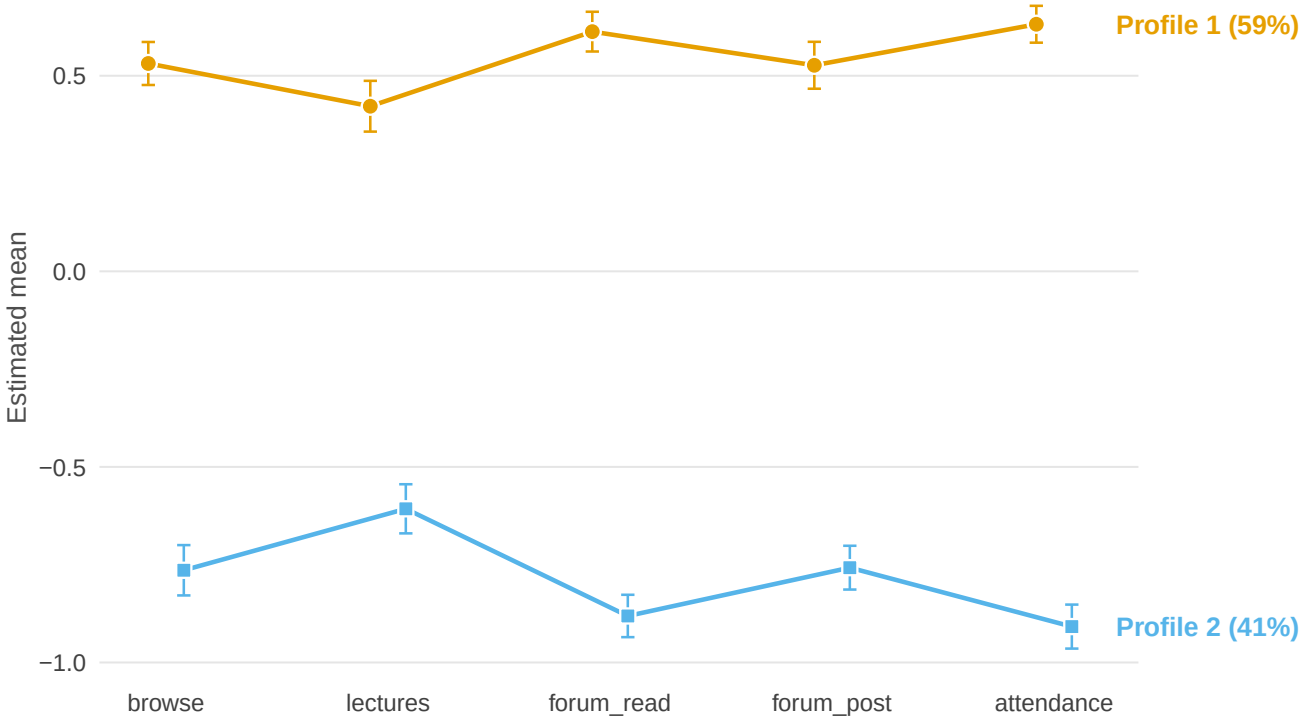
```
#>  min_standardized_eigenvalue boundary
#>                        0.2767    FALSE
#>                        0.2710    FALSE
#>
#> -- profile_probabilities ------------------------------------------
#>  group_class profile probability group_class_probability
#>            1       1      0.1644                  0.3252
#>            1       2      0.8356                  0.3252
#>            2       1      0.7934                  0.6748
#>    ... 1 more rows.  get_results(x, what = "profile_probabilities")
#>
#> -- counts ---------------------------------------------------------
#>        level class effective_count effective_proportion
#>  individuals     1          838.78               0.5899
#>  individuals     2          583.22               0.4101
#>       groups     1           34.48               0.3252
#>    ... 1 more rows.  get_results(x, what = "counts")
#>
#> -- posteriors -----------------------------------------------------
#>  row group profile posterior modal
#>    1     1       1 9.898e-01  TRUE
#>    2     1       1 8.523e-01  TRUE
#>    3     1       1 6.103e-05 FALSE
#>    ... 2841 more rows.  get_results(x, what = "posteriors")
#>
#> -- group_posteriors -----------------------------------------------
#>  group group_size log_likelihood group_class posterior modal
#>      1         13         -61.99           1 1.000e+00  TRUE
#>      2         13         -70.35           1 1.000e+00  TRUE
#>      3         14         -77.34           1 1.699e-09 FALSE
#>    ... 209 more rows.  get_results(x, what = "group_posteriors")
#>
#> -- assignments ----------------------------------------------------
#>  student browse lectures forum_read forum_post attendance profile group_class uncertainty
#>        1   0.73    -0.08       0.30       0.67       0.72       1           1   1.024e-02
#>        1   0.68     0.53      -0.02      -0.55       0.70       1           1   1.477e-01
#>        1  -1.51    -0.51      -0.91      -0.80      -0.97       2           1   6.103e-05
#>  posterior_profile_1 posterior_profile_2
#>            9.898e-01             0.01024
#>            8.523e-01             0.14772
#>            6.103e-05             0.99994
#>    ... 1419 more rows.  get_results(x, what = "assignments")
#>
#> -- classification -------------------------------------------------
#>     level class n_modal proportion_modal estimated_n estimated_proportion average_posterior
#>  individuals     1     842          0.5921     838.78               0.5899           0.9778
#>  individuals     2     580          0.4079     583.22               0.4101           0.9733
#>       groups     1      35          0.3302      34.48               0.3252           0.9709
#>  odds_correct_classification
#>                        30.56
#>                        52.34
#>                        69.15
#>    ... 1 more rows.  get_results(x, what = "classification")
```

```
#>
#> -- average_posteriors -------------------------------------------------
#>        level assigned_class class n_assigned average_posterior
#>  individuals              1     1        842           0.97775
#>  individuals              1     2        842           0.02225
#>  individuals              2     1        580           0.02674
#>    ... 5 more rows.  get_results(x, what = "average_posteriors")
#>
#> -- classification_errors ----------------------------------------------
#>        level true_class assigned_class probability
#>  individuals          1              1     0.98151
#>  individuals          1              2     0.01849
#>  individuals          2              1     0.03212
#>    ... 5 more rows.  get_results(x, what = "classification_errors")
#>
#> -- bch_weights --------------------------------------------------------
#>        level unit assigned_class class   weight
#>  individuals    1              1     1  1.01948
#>  individuals    1              1     2 -0.01948
#>  individuals    2              1     1  1.01948
#>    ... 3053 more rows.  get_results(x, what = "bch_weights")
#>
#> -- entropy ------------------------------------------------------------
#>        level n_classes n_units entropy_sum relative_entropy
#>  individuals         2    1422       92.36           0.9063
#>       groups         2     106        4.25           0.9422
#>
#> -- replication --------------------------------------------------------
#>  n_starts n_screened n_converged n_failed n_best_replicated share tolerance best_start
#>         5          0           5        0                 5     1   0.00711          1
#>  log_likelihood
#>           -8313
#>
#> -- residuals ----------------------------------------------------------
#>   profile indicator_1 indicator_2    kind observed expected  residual effective_n statistic df
#> profile_2  forum_post  attendance gaussian 0.124466 0.124466 2.296e-08     583.2 5.617e-07 NA
#> profile_2    browse  forum_post gaussian 0.006572 0.006572 1.549e-08     583.2 3.732e-07 NA
#> profile_2    browse  attendance gaussian 0.218610 0.218610 1.254e-08     583.2 3.173e-07 NA
#>  p_value p_adjusted
#>        1          1
#>        1          1
#>        1          1
#>    ... 17 more rows.  get_results(x, what = "residuals")
#>
#> -- information_criteria -----------------------------------------------
#> log_likelihood n_parameters  aic  kic bic_groups bic_individual sabic_groups sabic_individual
#>          -8313          43 16712 16758      16827         16938       16691           16802
#> caic_groups caic_individual awe_groups awe_individual icl_groups icl_individual clc_groups
#>       16870           16981      17346          17570      17017          17128      16816
#>  clc_individual
#>           16816
#>
#> -- model --------------------------------------------------------------
```

```
#>  n_observations n_informative n_groups n_profiles n_group_classes centering scaling
#>            1422          1422      106          2               2      none    none
#>  covariance_structure n_parameters n_parameters_with_measurement log_likelihood  aic bic_groups
#>                 VVV          43                          43         -8313 16712     16827
#>  bic_individual converged iterations boundary small_classes best_start n_best_replicated weights
#>          16938      TRUE        12    FALSE        FALSE         1                5    <NA>
#>
#> -- stages ---------------------------------------------------------
#>  stage group_classes fixed log_likelihood parameters parameters_with_measurement converged
#>  joint             2  <NA>          -8313         43                          43      TRUE
#>
#> -- starts ---------------------------------------------------------
#>  start log_likelihood converged iterations error boundary
#>      1          -8313      TRUE         12  <NA>    FALSE
#>      2          -8313      TRUE         15  <NA>    FALSE
#>      3          -8313      TRUE         12  <NA>    FALSE
#>    ... 2 more rows.  get_results(x, what = "starts")
#>
#> -- data -----------------------------------------------------------
#>  student browse lectures forum_read forum_post attendance
#>        1   0.73    -0.08       0.30       0.67       0.72
#>        1   0.68     0.53      -0.02      -0.55       0.70
#>        1  -1.51    -0.51      -0.91      -0.80      -0.97
#>    ... 1419 more rows.  get_results(x, what = "data")
#>
#> 21 tables above, truncated to fit. get_results(x, what = ) returns any
#> of them whole, and get_results(x, what = "all") returns every one.
```

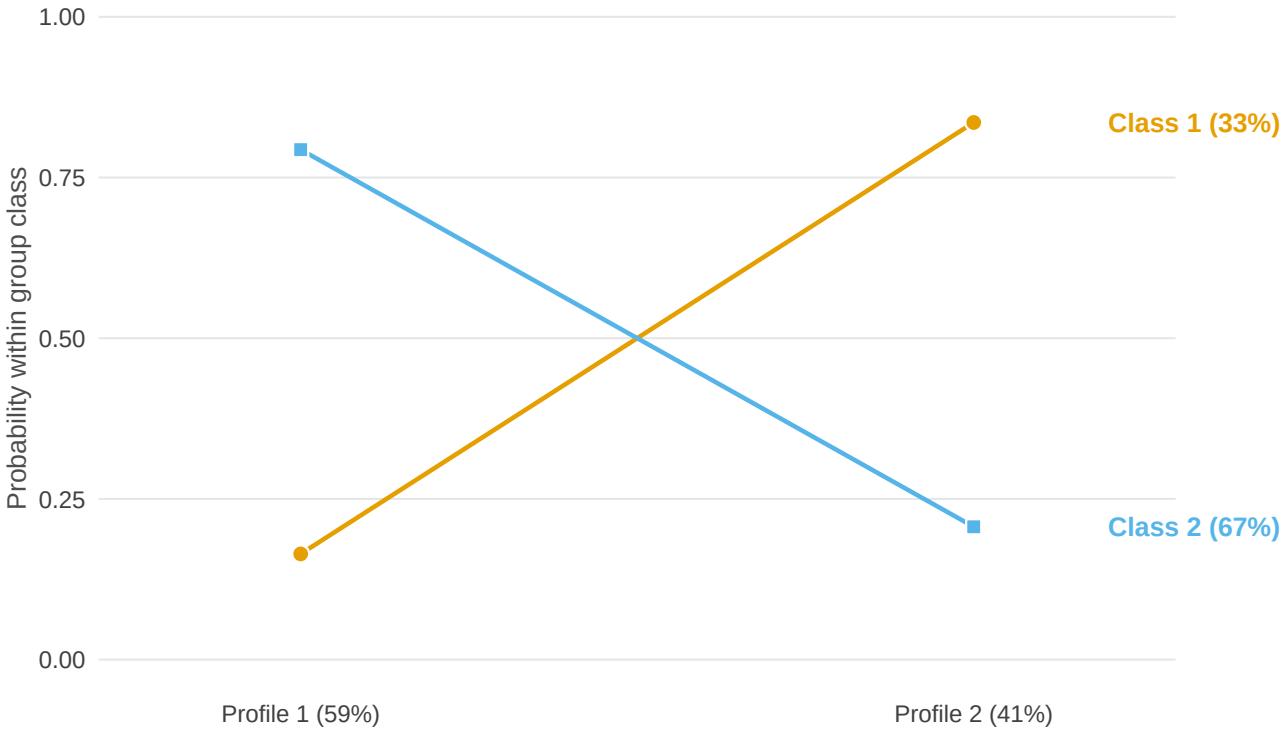
The profiles are plotted as their means on the five indicators.

```
plot_profiles(chosen)
```



Each group class of students has its own distribution of the two profiles.

```
plot_prevalence(chosen)
```

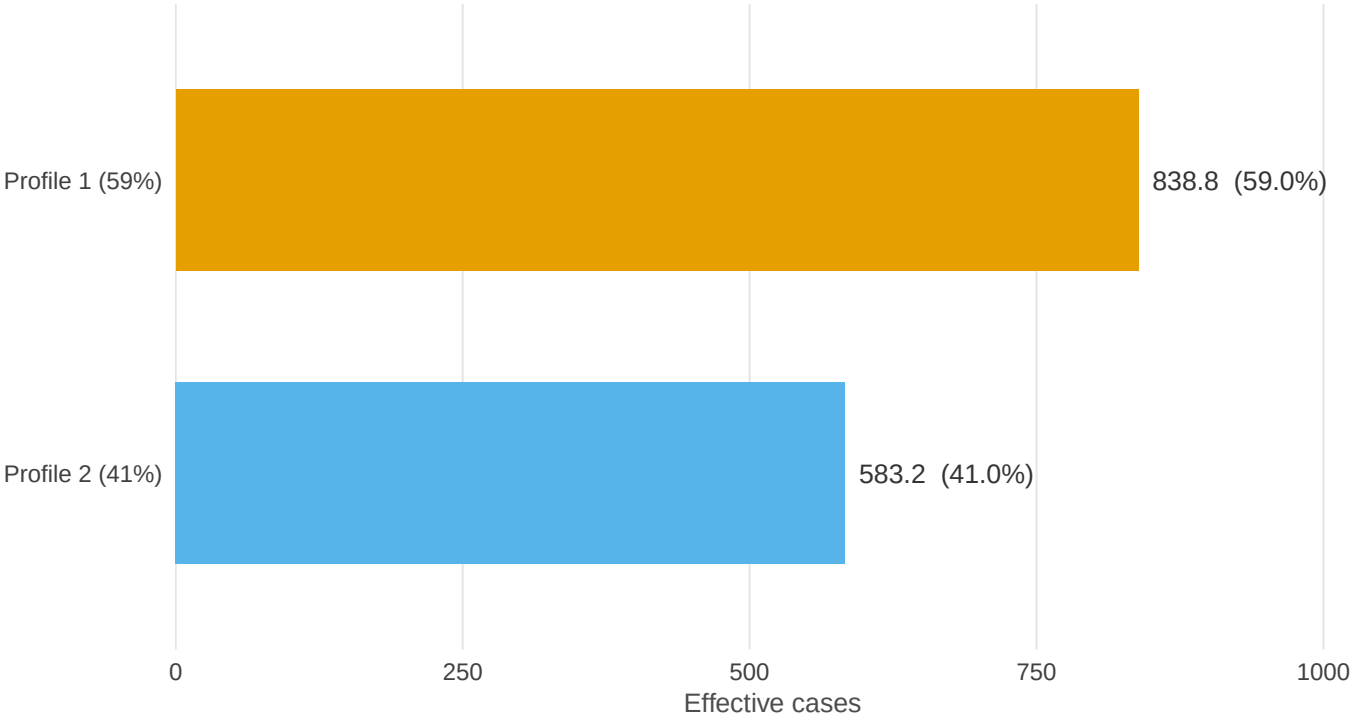


The classification diagnostics of the chosen model:

```
diagnostics(chosen)
```

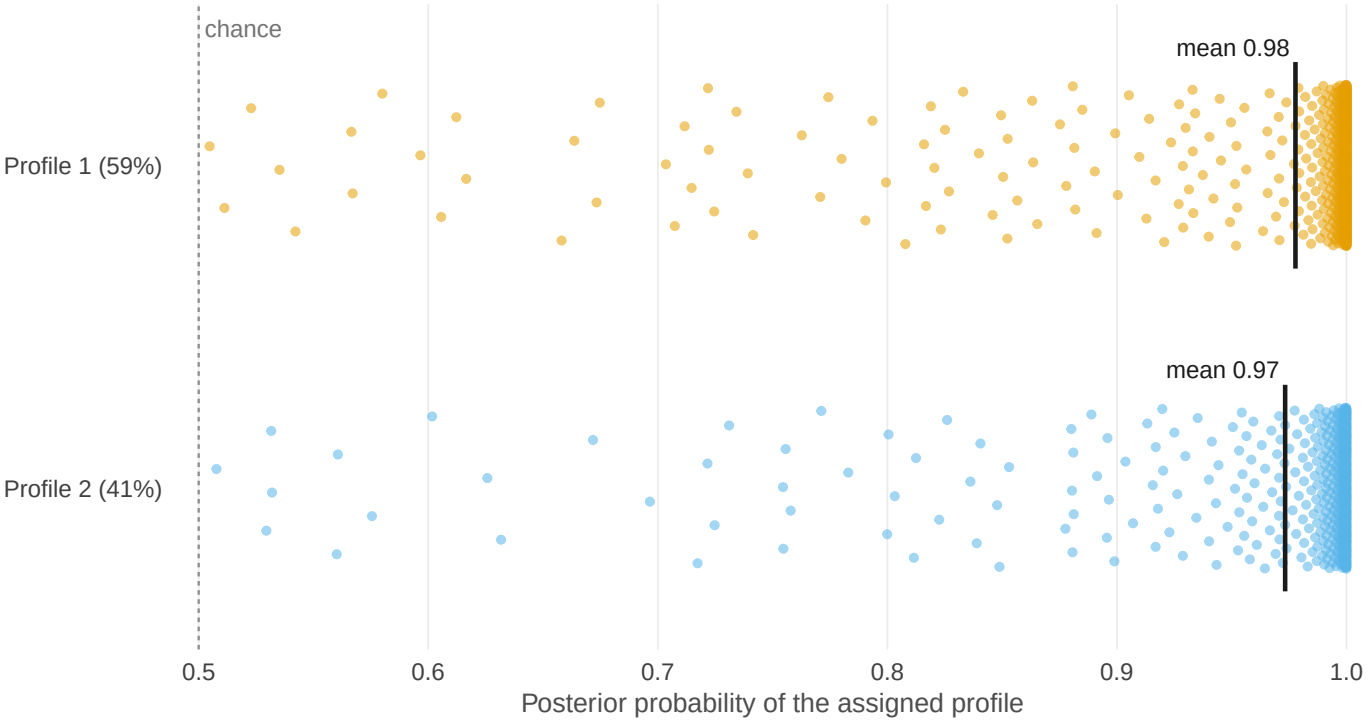
Profile sizes

Effective number of cases (sum of posterior probabilities); 1,422 in total



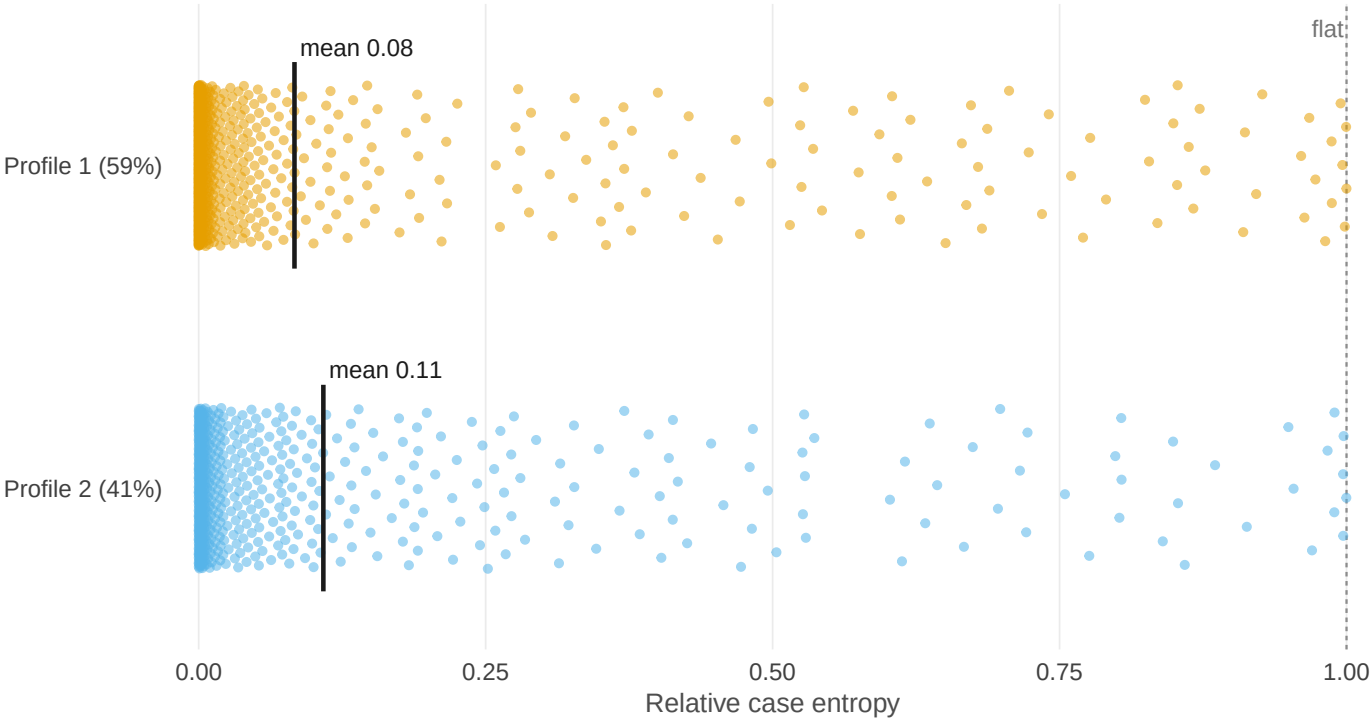
Classification certainty

Posterior probability of each case's assigned profile; the profile mean is its average posterior probability



Classification uncertainty

Each case's entropy relative to a flat posterior (0 = certain, 1 = flat); relative entropy = 1 − mean = 0.906



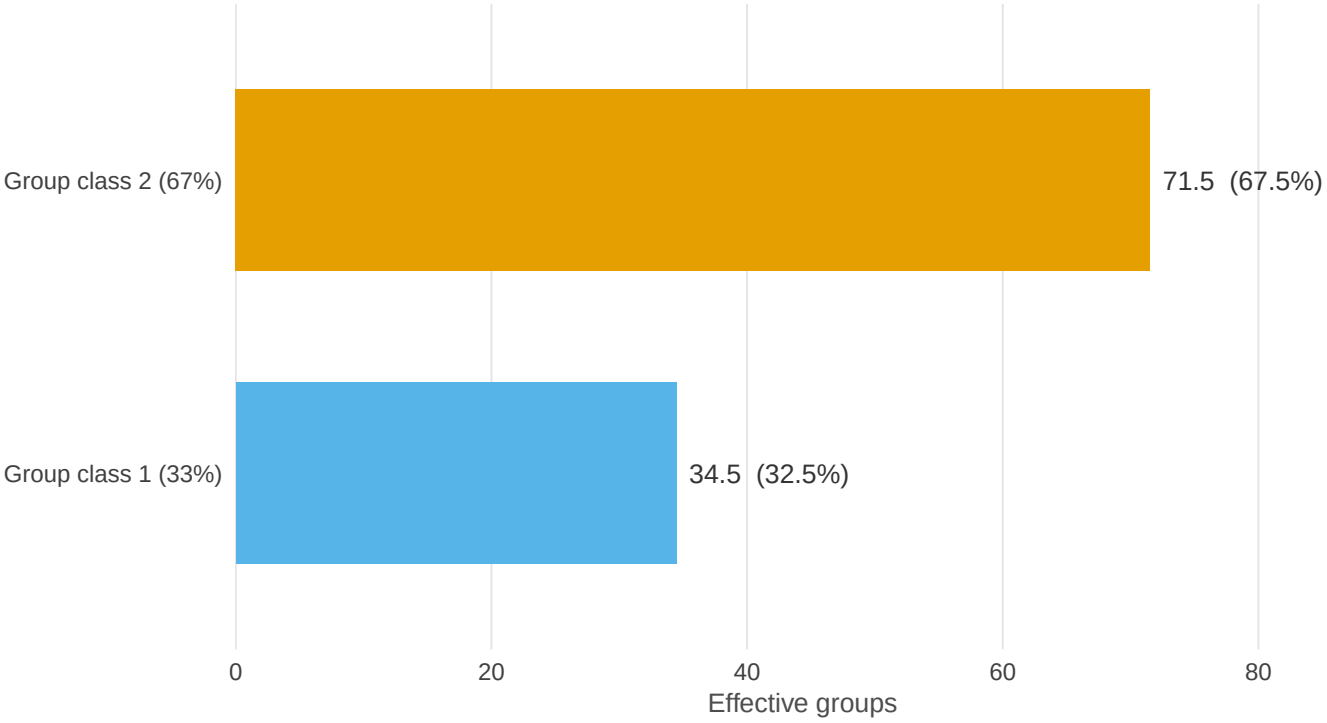
Average posterior probability

Mean posterior each assigned group places on each profile; the outlined diagonal is classification accuracy



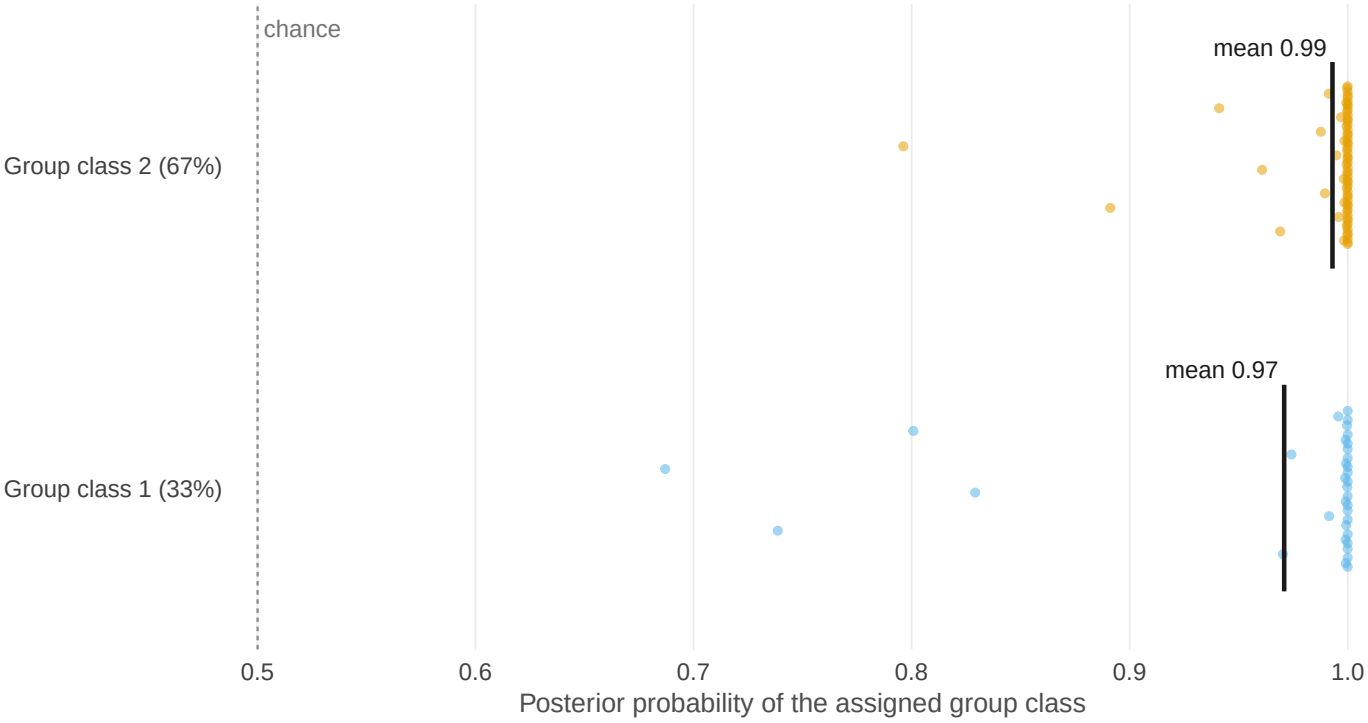
Group class sizes

Effective number of groups (sum of posterior probabilities); 106 in total



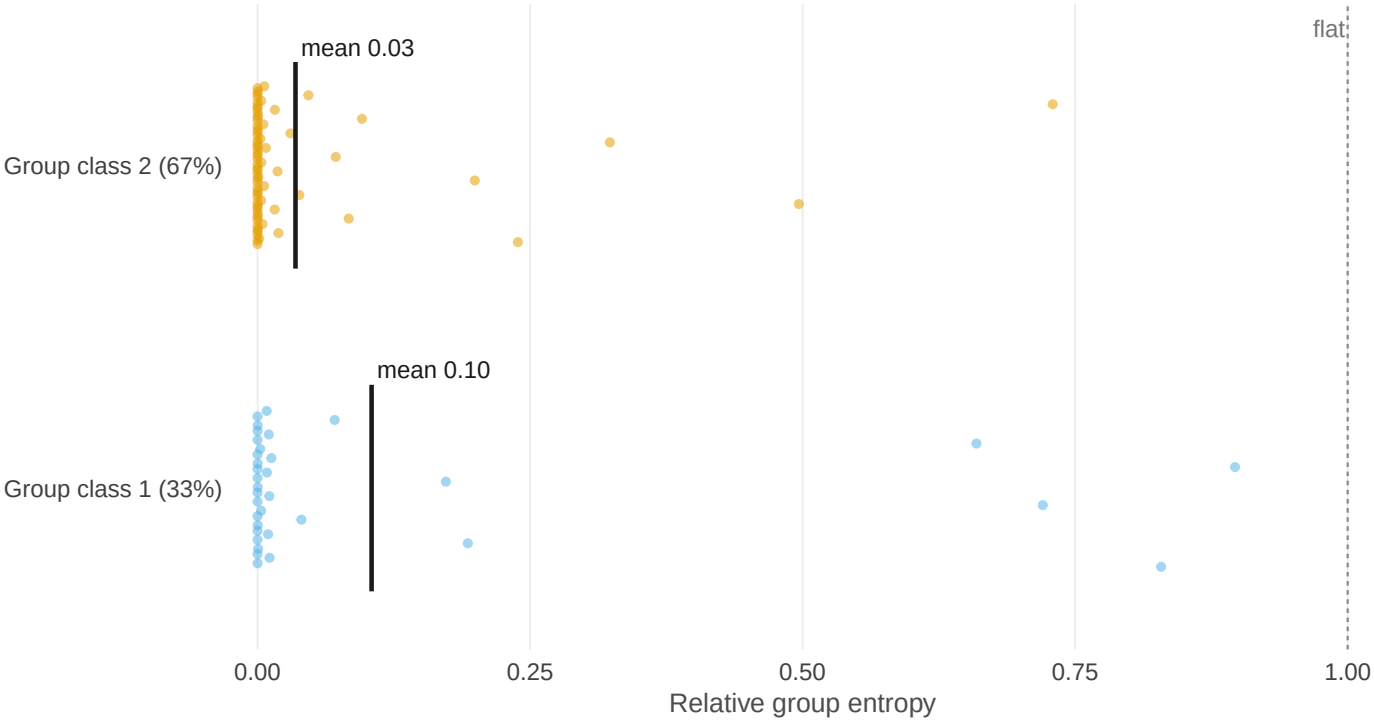
Group classification certainty

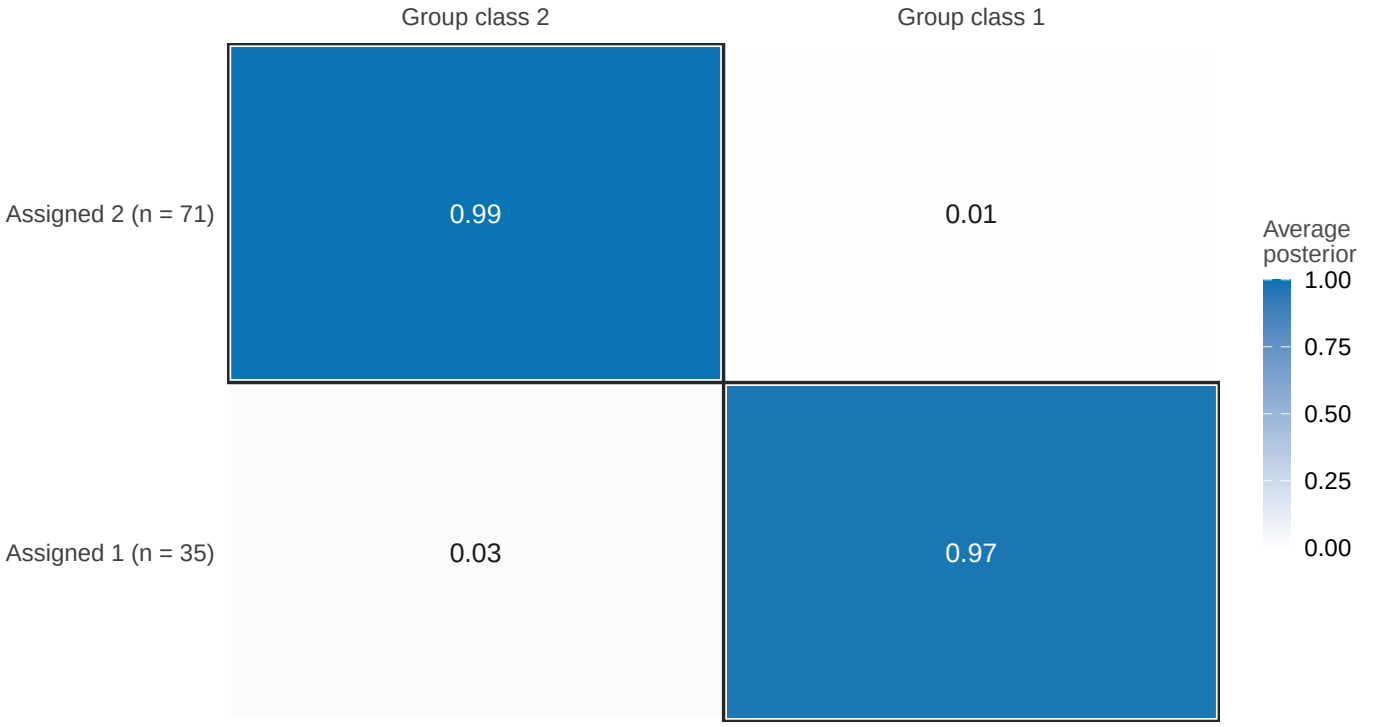
Posterior probability of each group's assigned group class; the group class mean is its average posterior probability



Group classification uncertainty

Each group's entropy relative to a flat posterior (0 = certain, 1 = flat); relative entropy = 1 − mean = 0.942






```
#> Diagnostics: Two-level latent profile analysis, 2 profiles, 2 group classes
#> No check needs attention.
#>
#> Estimation
#>   ok     best start converged                                yes
#>   info   best log likelihood reached by                      5 of 5 starts
#>   ok     starts failed or unfinished                         0 of 5
#>   ok     estimates at a bound                                no
#>
#> Class size
#>   info   smallest class                       individuals  41.0% (n = 583, class 2)
#>   info   modal vs estimated proportions       individuals  largest gap 0.002
#>   ok     classes with no assigned unit        individuals  0 of 2
#>   info   smallest class                       groups       32.5% (n = 34, class 1)
#>   info   modal vs estimated proportions       groups       largest gap 0.005
#>   ok     classes with no assigned group       groups       0 of 2
#>
#> Classification
#>   info   relative entropy                     individuals  0.906
#>   info   lowest average posterior             individuals  0.973 (class 2)
#>   info   lowest odds of correct classification individuals  30.6 (class 1)
#>   info   relative entropy                     groups       0.942
#>   info   lowest average posterior             groups       0.971 (class 1)
#>   info   lowest odds of correct classification groups       68.6 (class 2)
#>
#> Local independence
```

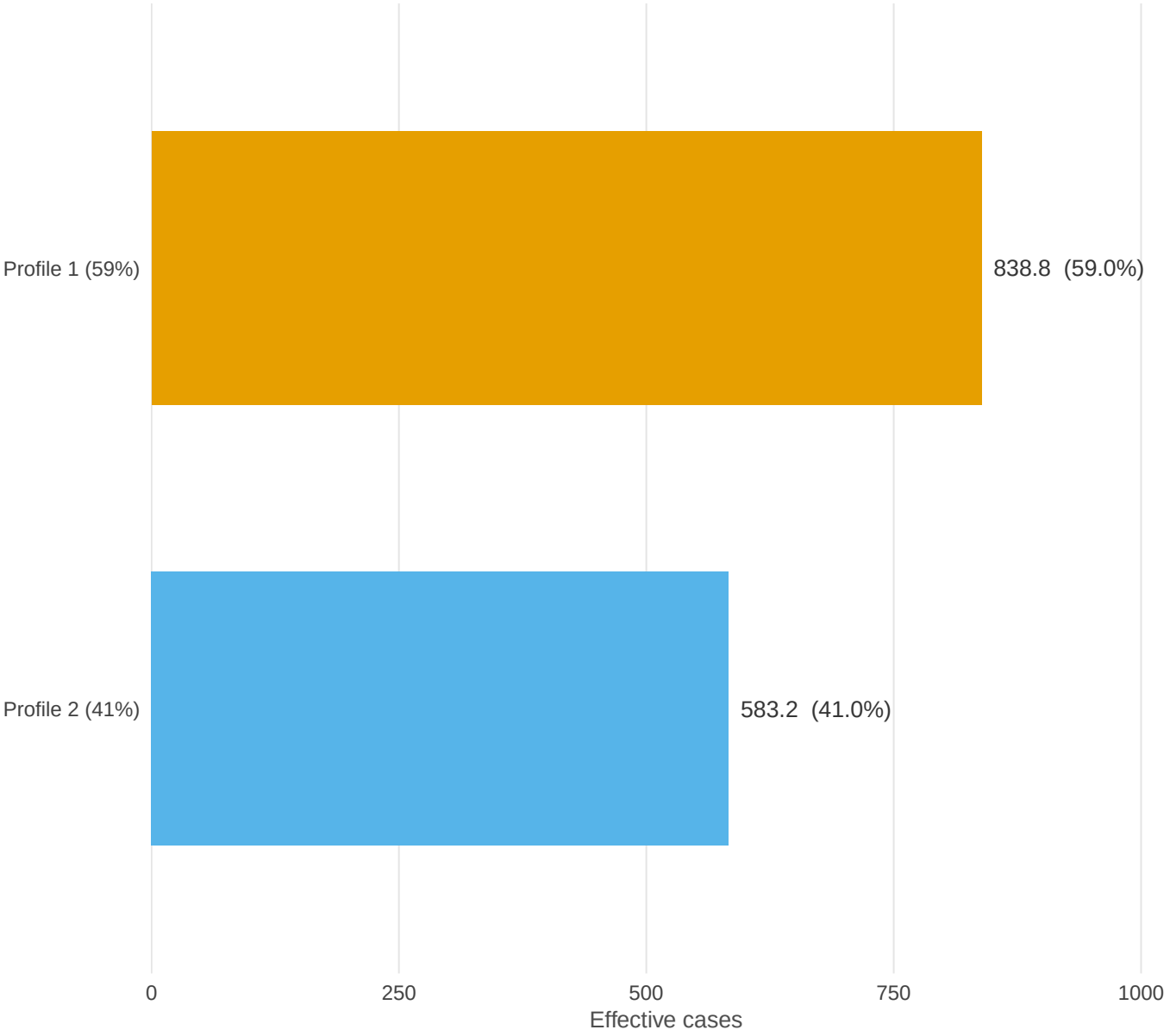
```
#>   ok     pairs dependent within a class                      0 of 20 tested pairs     BH-
adjusted p < 0.05
#>       smallest adjusted p: forum_post x attendance in profile_2, residual 0.000, adjusted p 1
#>
#> Information
#>   info   log likelihood, parameters, BIC                     -8313.11, 43, 16826.75
#>
#> Every check with its reference: get_results(x, "checks"). Other tables: get_results(x, what = ).
```

The relative entropy of the chosen model is 0.906 for the profiles and 0.942 for the group classes; the lowest AvePP is 0.973 among the profiles and 0.971 among the group classes.

```
plot_diagnostics(chosen)
```

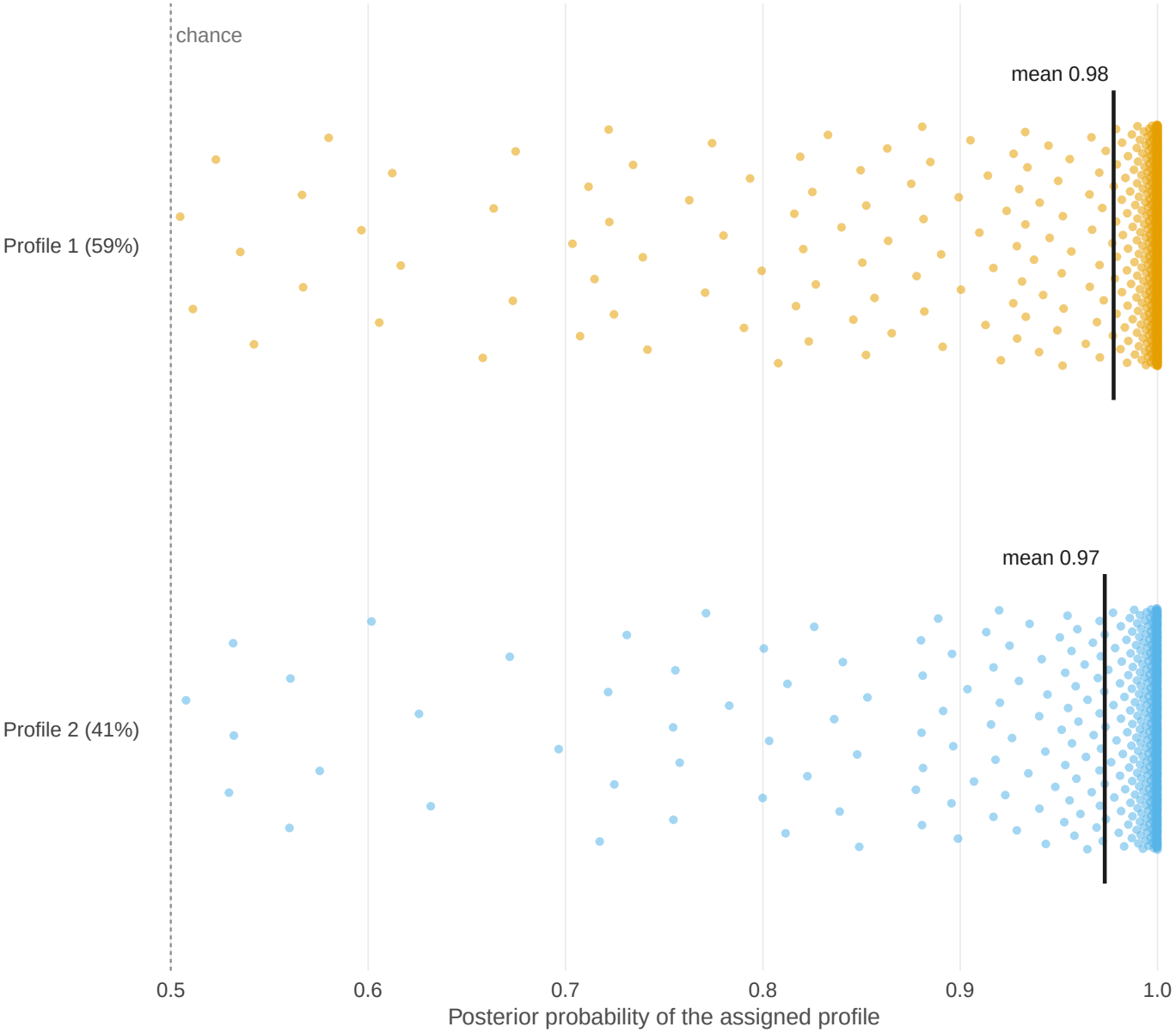
Profile sizes

Effective number of cases (sum of posterior probabilities); 1,422 in total



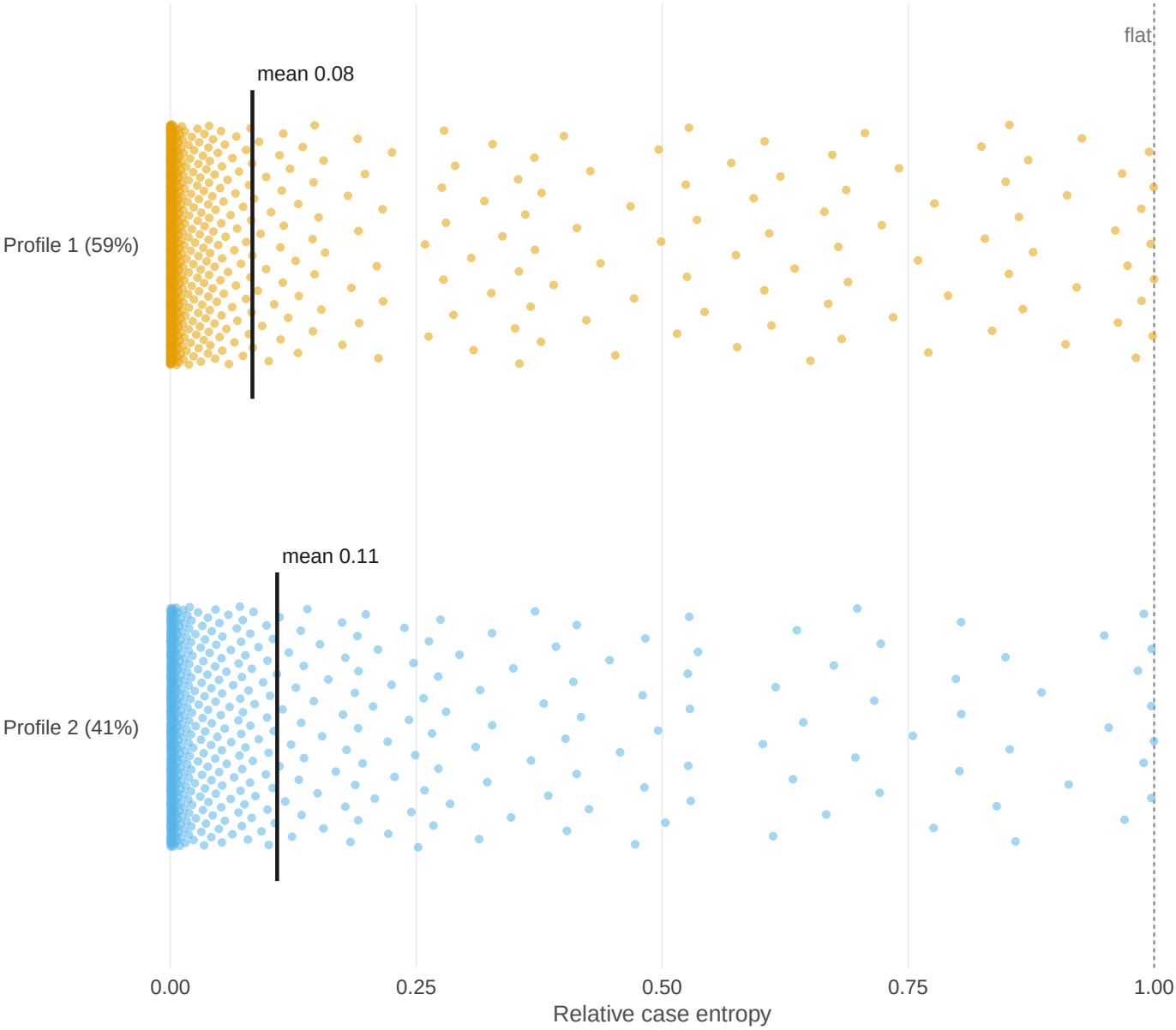
Classification certainty

Posterior probability of each case's assigned profile; the profile mean is its average posterior probability



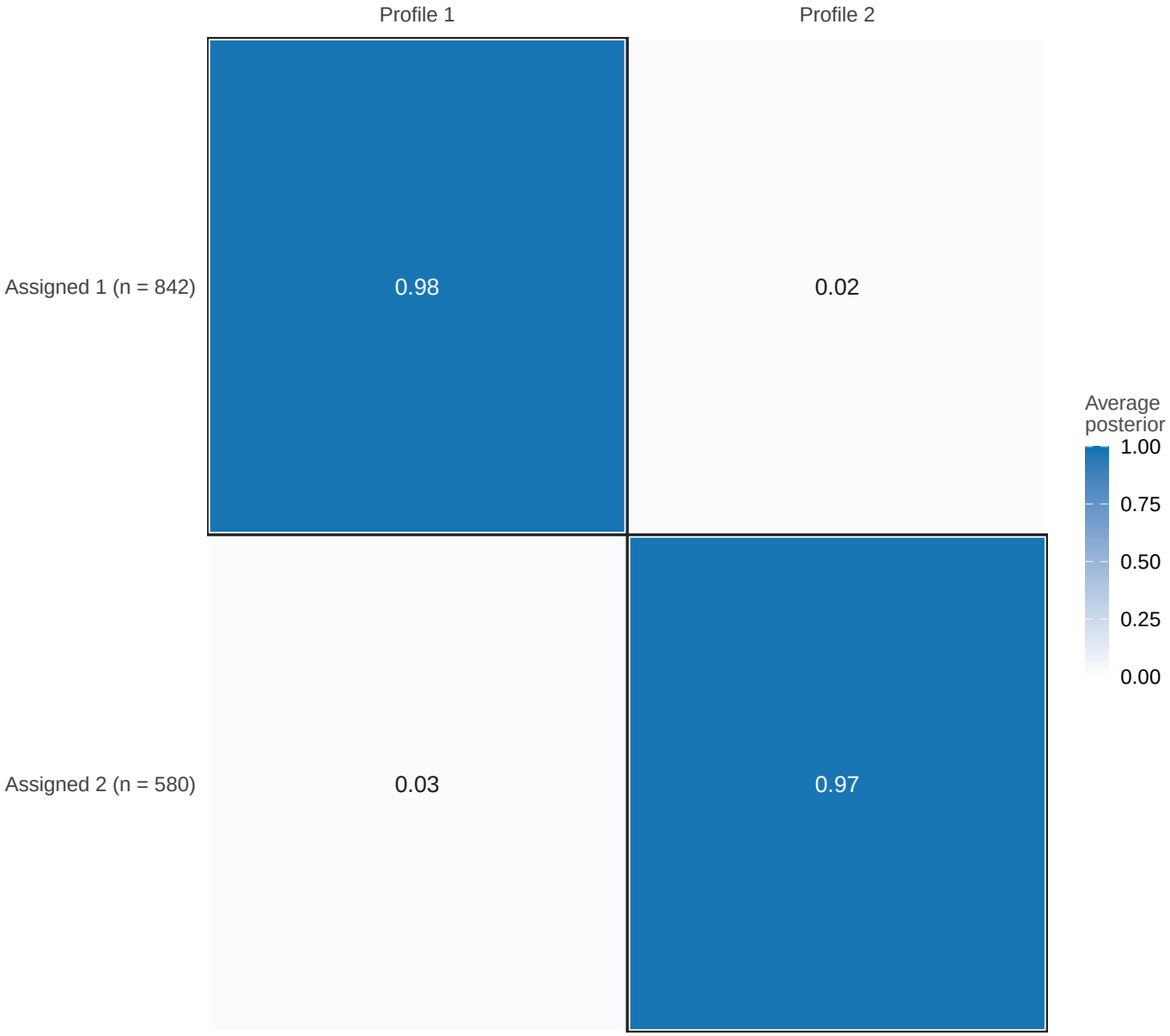
Classification uncertainty

Each case's entropy relative to a flat posterior (0 = certain, 1 = flat); relative entropy = 1 − mean = 0.906



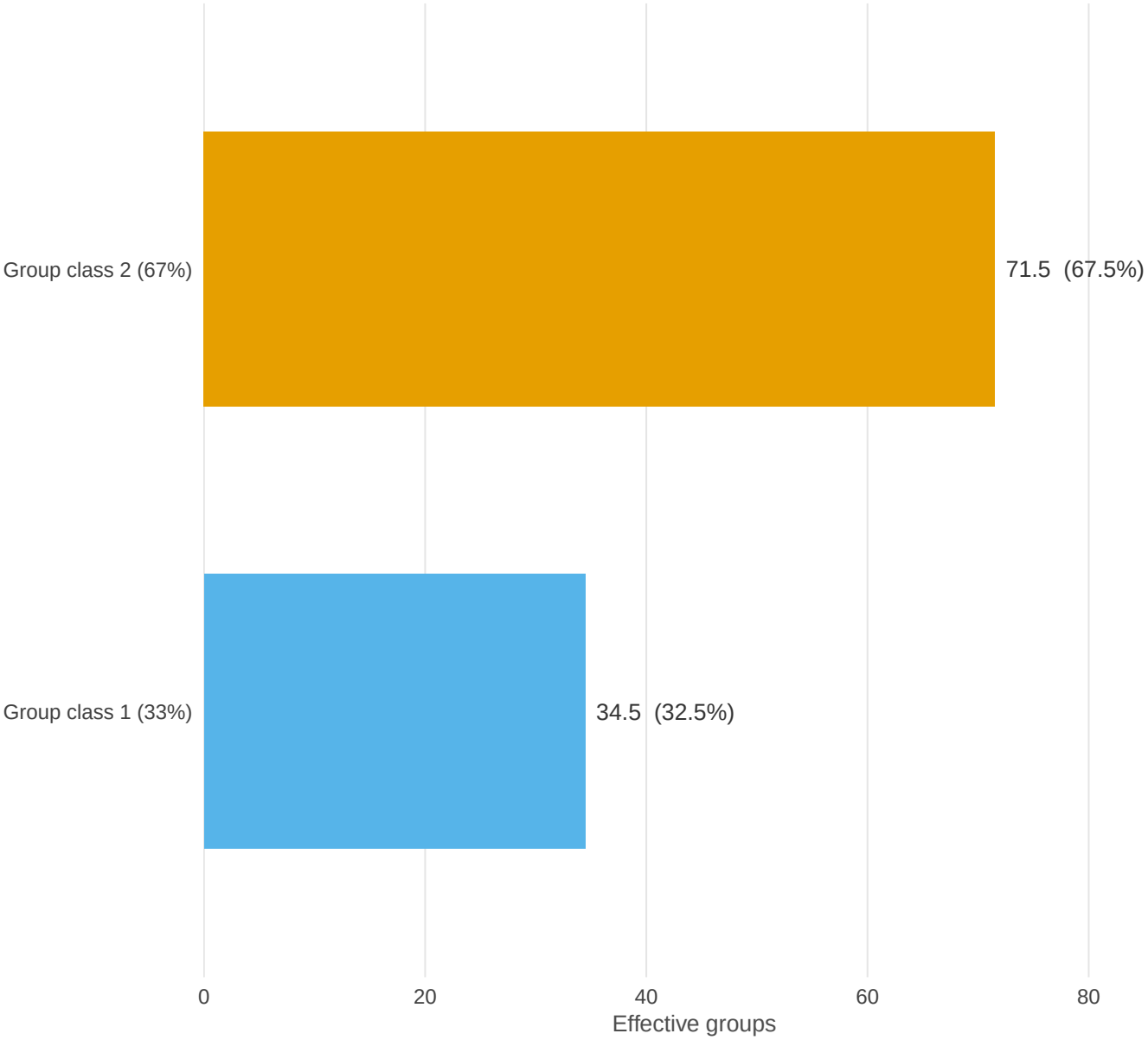
Average posterior probability

Mean posterior each assigned group places on each profile; the outlined diagonal is classification accuracy



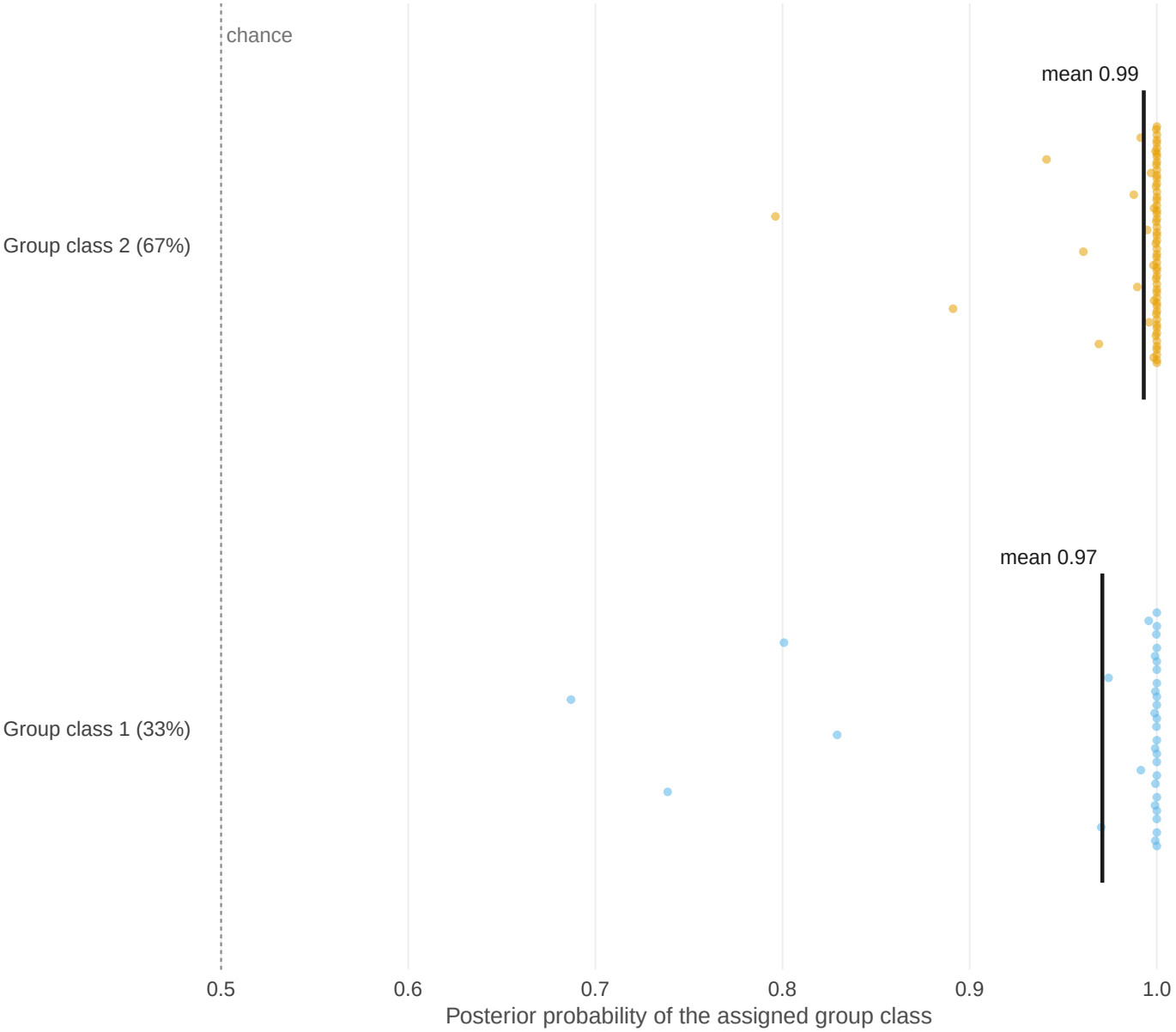
Group class sizes

Effective number of groups (sum of posterior probabilities); 106 in total



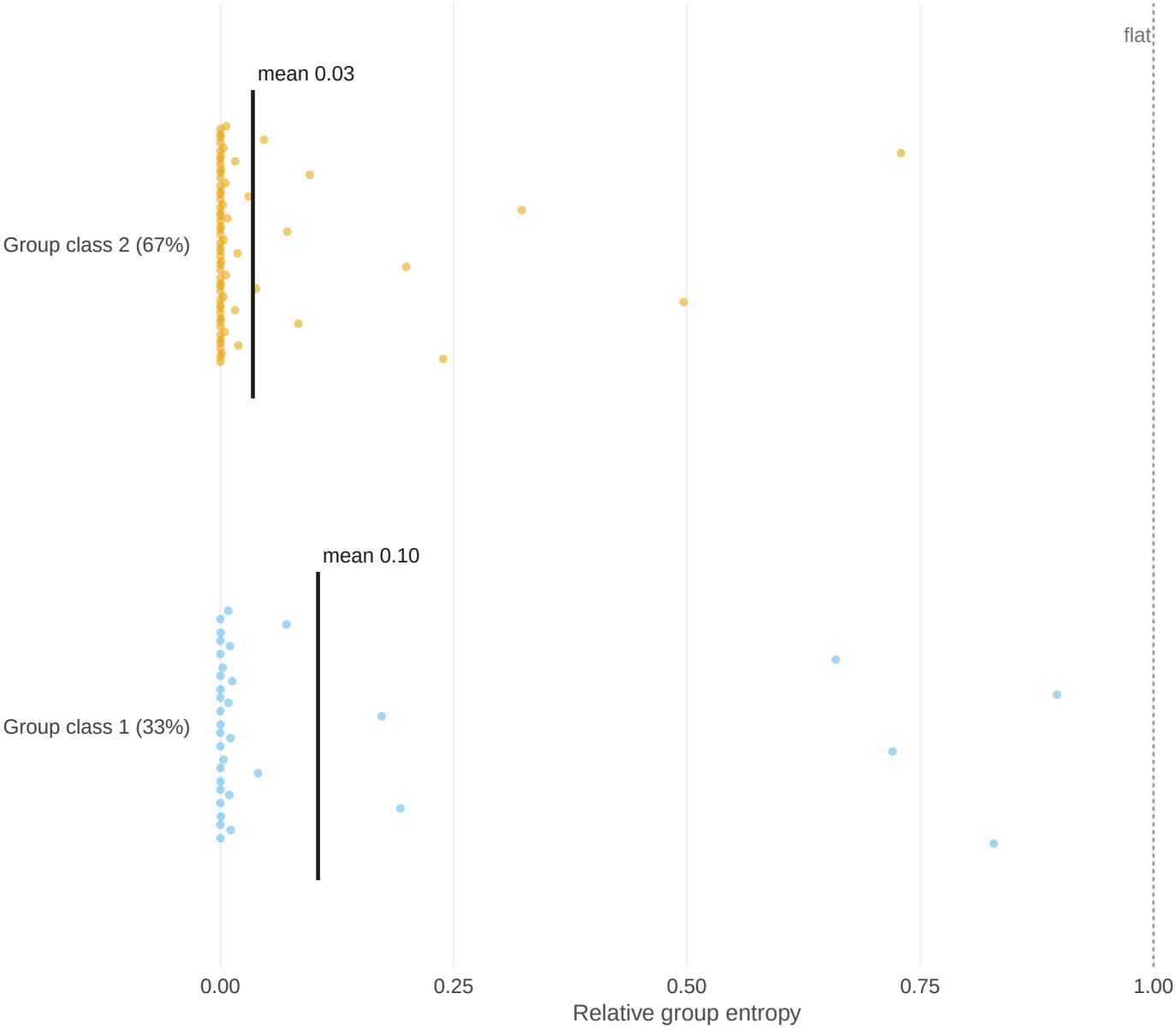
Group classification certainty

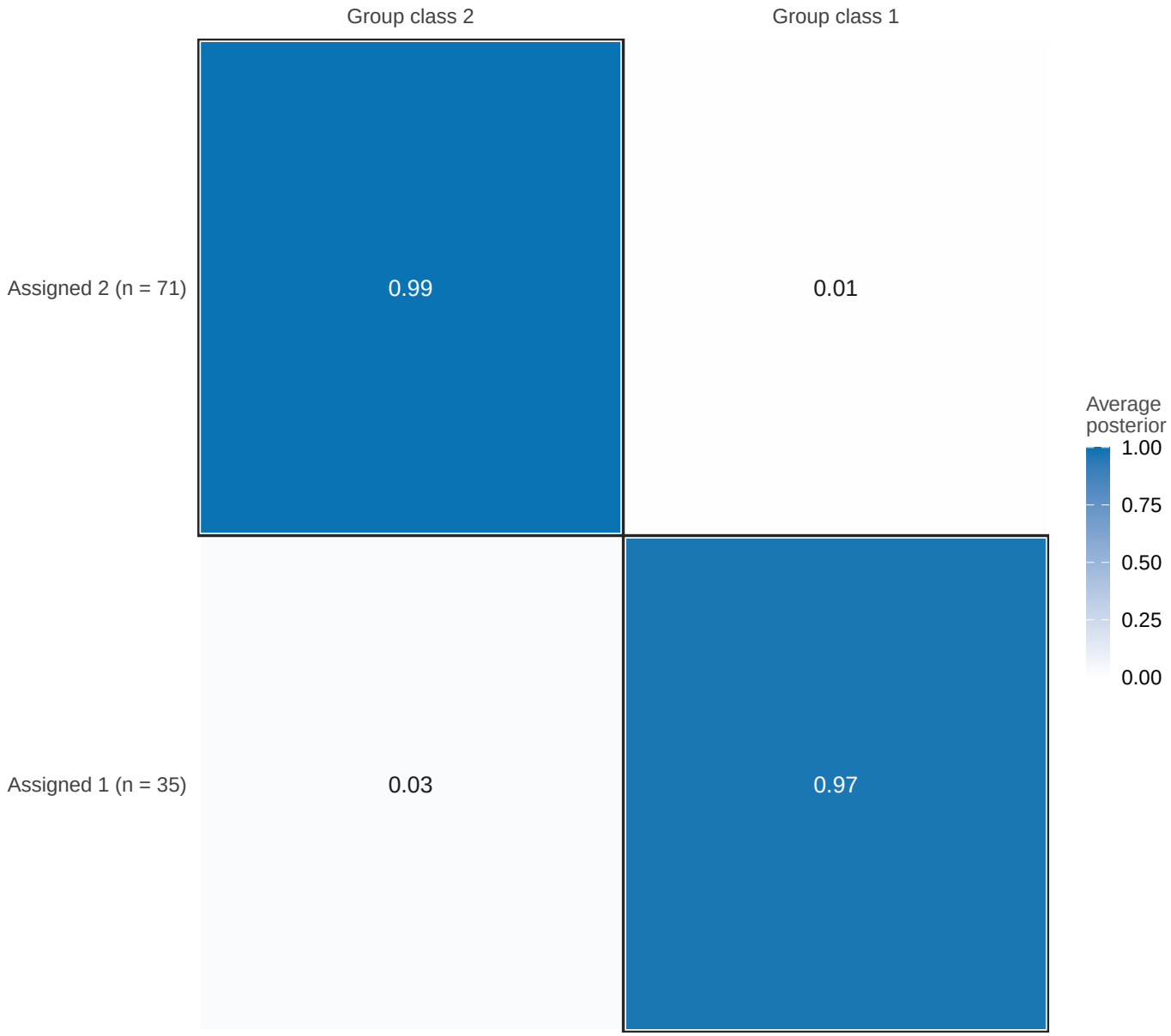
Posterior probability of each group's assigned group class; the group class mean is its average posterior probability



Group classification uncertainty

Each group's entropy relative to a flat posterior (0 = certain, 1 = flat); relative entropy = 1 − mean = 0.942





Each enrolment with its profile, its student's group class and its posterior probabilities:

```
head(get_results(chosen, "assignments", data = course_engagement))
#>   student course sequence browse lectures forum_read forum_post attendance previous_grade
#> 1       1     23        1   0.73    -0.08       0.30       0.67       0.72           0.54
#> 2       1     19        2   0.68     0.53      -0.02      -0.55       0.70          -0.71
#> 3       1      4        3  -1.51    -0.51      -0.91      -0.80      -0.97           1.98
#> 4       1     18        4   0.64    -1.02      -1.62      -1.20      -1.26          -1.42
#> 5       1     16        5  -0.94    -0.37      -0.71      -0.28      -1.52          -0.67
#> 6       1     29        6  -0.36    -0.46      -0.77      -0.43      -1.34          -0.40
#>  engagement student_type profile group_class uncertainty posterior_profile_1 posterior_profile_2
```

```
#> 1    engaged    wavering      1         1   0.010237          0.989763          0.0102
#> 2    engaged    wavering      1         1   0.147719          0.852281          0.1477
#> 3 disengaged    wavering      2         1   0.000061          0.000061          0.9999
#> 4 disengaged    wavering      2         1   0.000193          0.000193          0.9998
#> 5 disengaged    wavering      2         1   0.000224          0.000224          0.9998
#> 6 disengaged    wavering      2         1   0.000563          0.000563          0.9994
```

The profiles can also be related to an external variable. The following analysis uses `three_step()` to estimate profile-specific means of `previous_grade`, with the BCH correction for classification error (Bolck et al., 2004; Bakk & Vermunt, 2016). The variable precedes the course enrolment, so these comparisons describe associations rather than effects of profile membership on grades.

```
three_step(chosen, data = course_engagement, outcome = "previous_grade")
#> Distal outcome by class (bch, 95% CI)
#>
#> Class   Mean          95% CI     SE  Effective n
#> -----  -----  --------------  -----  -----------
#>     1   0.23  [ 0.16,  0.30]  0.035       803.34
#>     2  -0.33  [-0.40, -0.27]  0.035       548.42
```

The previous grade averages 0.23 standard deviations above the mean before enrolments of the first profile and 0.33 below it before enrolments of the second, with 95% intervals that do not overlap.

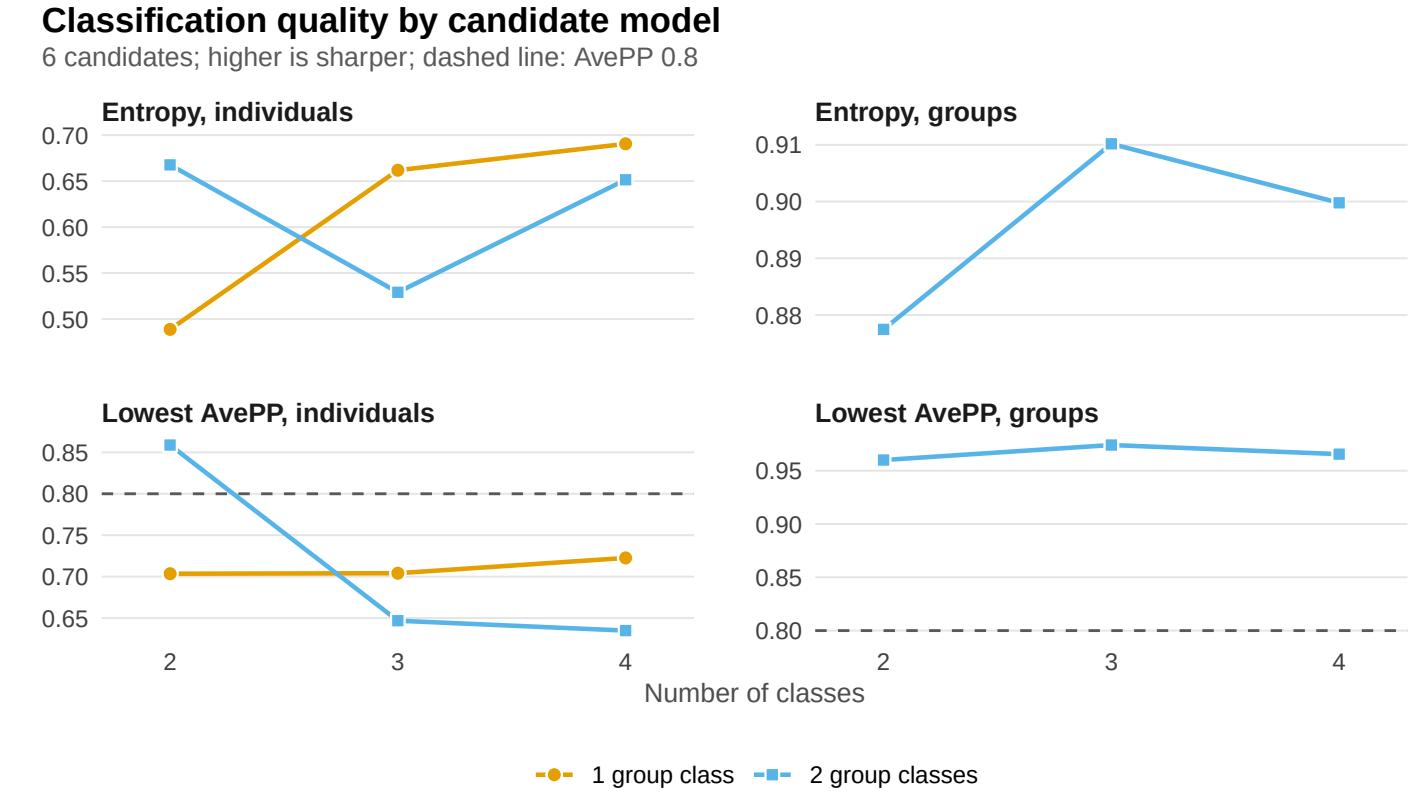
Other model types

Latent class models

`student_esm` contains 2,582 momentary reports from students, with eight activities recorded as present or absent. Declaring the indicators in `categorical` makes every candidate a latent class model, and the candidates are named with `classes` in place of `profiles`.

```
activities <- c("time_with_friends", "on_social_media", "tv_video_games",
                "listened_music", "sports", "walking", "reading",
                "part_time_job")
lca_models <- enumerate_classes(student_esm, activities, "student",
                                n_profiles = 2:4, n_group_classes = 1:2,
                                categorical = activities, n_starts = 5,
                                seed = 1)
#> 3 of 6 candidates reached their best log likelihood from one start only (classes4_groups1, classes3_groups2,
lca_models
#> Class enumeration: 6 candidate models (LCA)
#>         candidate n_profiles n_group_classes log_likelihood n_parameters   aic bic_groups
#>  classes2_groups1          2               1          -7892           17 15819      15863
#>  classes3_groups1          3               1          -7848           26 15749      15817
#>  classes4_groups1          4               1          -7811           35 15692      15783
#>  classes2_groups2          2               2          -7740           19 15518      15567
#>  classes3_groups2          3               2          -7695           29 15448      15524
#>  classes4_groups2          4               2          -7653           39 15383      15485
#>  bic_individual icl_individual profile_entropy group_entropy avepp_min avepp_max avepp_mean
#>           15918          17748           0.489            NA     0.704     0.941      0.822
#>           15901          17820           0.662            NA     0.704     0.952      0.863
#>           15897          18112           0.691            NA     0.723     0.997      0.814
```

```
#>          15629          16763          0.668         0.877     0.859     0.926      0.892
#>          15618          18204          0.529         0.910     0.647     0.846      0.778
#>          15612          18012          0.651         0.900     0.635     0.883      0.793
#>  avepp_below group_avepp_min group_avepp_max group_avepp_mean group_avepp_below converged boundary
#>          1             NA             NA             NA               NA      TRUE    FALSE
#>          1             NA             NA             NA               NA      TRUE     TRUE
#>          3             NA             NA             NA               NA      TRUE     TRUE
#>          0          0.960          0.980          0.970                0      TRUE    FALSE
#>          1          0.974          0.984          0.979                0      TRUE     TRUE
#>          1          0.966          0.976          0.971                0      TRUE     TRUE
#>   n_best_replicated
#>                   5
#>                   2
#>                   1
#>                   4
#>                   1
#>                   1
#> Every column, unrounded: as.data.frame(x).
#> No candidate is selected automatically. Compare one convention consistently.
plot_selection(lca_models, what = "diagnostics")
```




```
compare_models(lca_models, n_profiles = 3)
#> Model comparison
#>
```

```
#> Model         Classes  Group classes  k   LogLik     BIC   ΔBIC  Weight     ICL  Entropy  Min AvePP  < 0.8  Gr
#> ---------------- ------- ------------- -- -------- -------- ------ ------ -------- -------  ---------
#> classes3_groups1     3          1  26 -7848.45 15816.64 293.10  0.000 17819.69    0.66     0.70    1
#> classes3_groups2     3          2  29 -7695.00 15523.55   0.00  1.000 18203.66    0.53     0.65    1
```

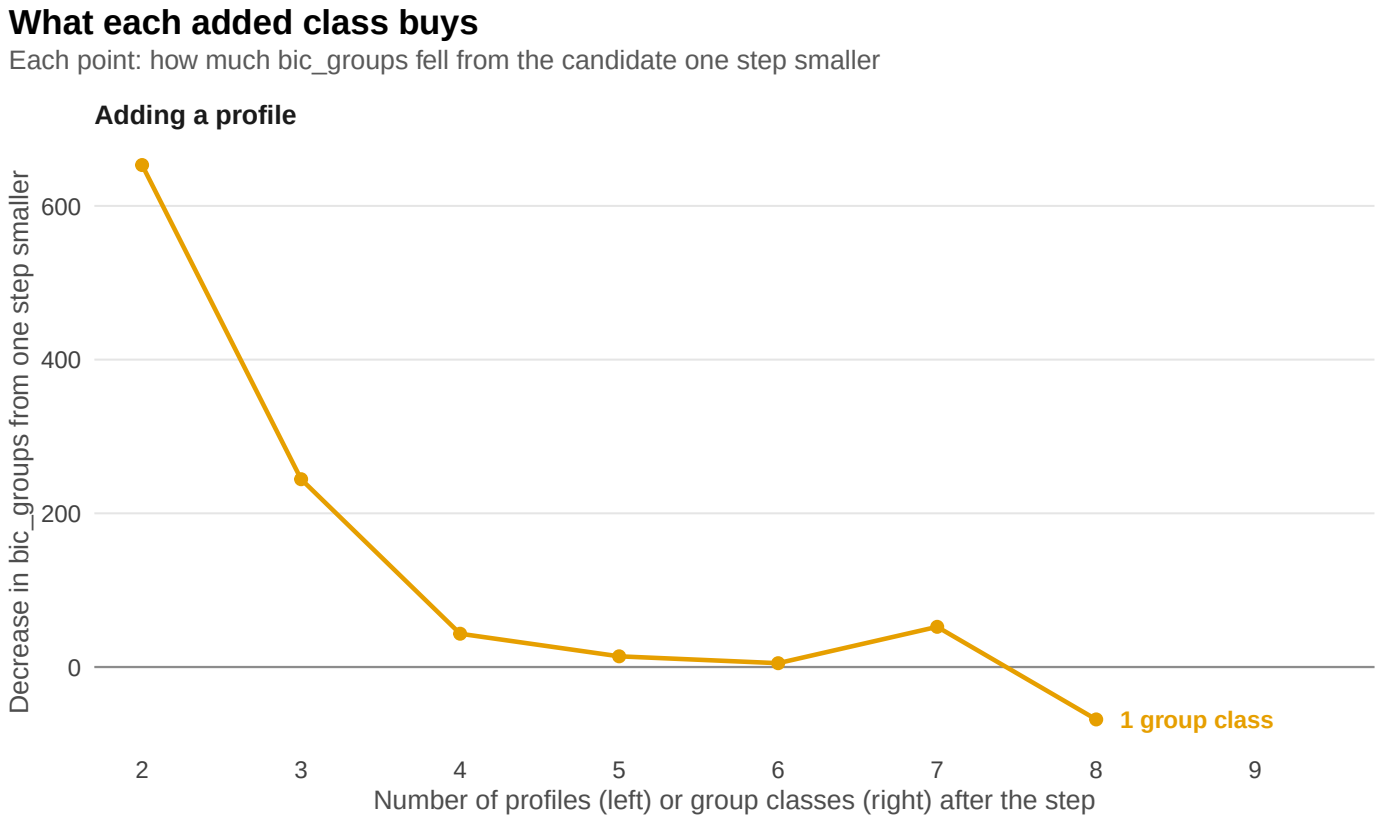
Among the three-class models, the model with two group classes has the lower BIC. Its relative entropy is 0.53, and its least distinct class has an AvePP of 0.65. This solution was reached by a single start and lies below the maximum found with more starts in the lca vignette (log likelihood -7,695.0 against -7,678.1), so its overlapping classes reflect the search rather than the model; the two-level latent class analysis vignette analyses these data with more starts.

Single-level models

Observations without a grouping structure are modelled with `enumerate_lpa()` for continuous indicators and `enumerate_lca()` for categorical indicators. The candidates are named by their numbers of profiles or classes alone. The example below enumerates the `srl` data under the diagonal `VVI` structure, which shows how an unsuitable structure leads to degenerate candidates; the latent profile analysis workflow analyses the same data with the full `EEE` structure.

```
srl_models <- enumerate_lpa(srl, c("cognitive_strategies", "intrinsic_value",
                                   "self_efficacy", "self_regulation",
                                   "test_anxiety"),
                            n_profiles = 1:8, model = "VVI", n_starts = 10,
                            seed = 1)
#> Warning: Profile 4: 9.8 Gaussian membership mass for 10 free means and variances. This is a
#> conservative diagnostic, not an identification test; inspect the Gaussian covariance diagnostics.
#> Warning: A variance reached min_variance; this is a bound-active constrained fit.
#> Warning: Profile 3: 3.0 Gaussian membership mass for 10 free means and variances. This is a
#> conservative diagnostic, not an identification test; inspect the Gaussian covariance diagnostics.
#> Warning: A variance reached min_variance; this is a bound-active constrained fit.
#> Warning: Profiles 3, 4: 1.0, 2.0 Gaussian membership mass for 10 free means and variances each.
#> This is a conservative diagnostic, not an identification test; inspect the Gaussian covariance
#> diagnostics.
#> 2 of 8 candidates reached their best log likelihood from one start only (profiles7, profiles8); refit those y
srl_models
#> Class enumeration: 8 candidate models (LPA)
#>  candidate n_profiles model log_likelihood n_parameters  aic bic_groups icl_individual
#>  profiles1          1   VVI          -1247           10 2515       2552           2552
#>  profiles2          2   VVI           -889           21 1821       1898           1940
#>  profiles3          3   VVI           -736           32 1535       1654           1714
#>  profiles4          4   VVI           -683           43 1451       1611           1728
#>  profiles5          5   VVI           -644           54 1397       1597           1684
#>  profiles6          6   VVI           -611           65 1351       1592           1708
#>  profiles7          7   VVI           -553           76 1258       1539           1648
#>  profiles8          8   VVI           -556           87 1285       1608           1736
#>  profile_entropy avepp_min avepp_max avepp_mean avepp_below converged boundary n_best_replicated
#>               NA        NA        NA         NA          NA      TRUE    FALSE               10
#>            0.899     0.969     0.979      0.974           0      TRUE    FALSE               10
#>            0.909     0.936     0.975      0.959           0      TRUE    FALSE               10
#>            0.859     0.885     0.954      0.917           0      TRUE    FALSE                8
#>            0.909     0.921     0.949      0.938           0      TRUE    FALSE                7
#>            0.892     0.886     0.959      0.934           0      TRUE    FALSE                5
```

```
#>          0.907    0.904    0.995     0.949         0     TRUE     TRUE              1
#>          0.897    0.896    1.000     0.937         0     TRUE     TRUE              1
#> Every column, unrounded: as.data.frame(x).
#> No candidate is selected automatically. Compare one convention consistently.
compare_models(srl_models, "profiles5", "profiles6", "profiles7")
#> Model comparison
#>
#> Model      Profiles   k   LogLik      BIC   ΔBIC  Weight      ICL  Entropy  Min AvePP  < 0.8
#> ---------  --------  --  -------  -------  -----  ------  -------  -------  ---------  -----
#> profiles5         5  54  -644.38  1596.76  57.27   0.000  1684.32     0.91       0.92      0
#> profiles6         6  65  -610.52  1591.78  52.30   0.000  1707.76     0.89       0.89      0
#> profiles7         7  76  -553.00  1539.48   0.00   1.000  1648.49     0.91       0.90      0
plot_selection(srl_models, what = "deltas")
```



BIC is minimized by profiles7, but that candidate is a boundary fit: a variance reached its lower bound, and one profile holds about 3 observations. Its low BIC reflects a nearly degenerate component, not a better description of the data. The six-profile candidate is the best fit that is not on the boundary, but it too has a profile of about 10 observations, too few to determine its own ten means and variances. The enumeration reports both with a `latents_thin_profiles` warning; such a profile may be a spurious maximum, and the choice should be weighed against the five-profile candidate, which `compare_models()` shows beside them. The eight-profile candidate is also on the boundary, with two nearly empty profiles, which marks the limit of what 300 observations can support. Every warning issued during the enumeration is retained in the `warnings` column of the table.

Missing values and scaled indicators

Two arguments alter the data to which the candidates are fitted and, like `n_starts`, apply to all candidates. With `missing = "fiml"`, rows with some indicators missing contribute through the indicators observed (full information maximum likelihood); rows with every indicator missing are excluded from the individual sample size, entropy and AvePP, since they carry no information on class membership. `centering` subtracts and `scaling` divides by a mean or standard deviation computed over the sample (`"grand"`) or within each group (`"person"`); combined, they yield grand or within-person z-scores. Information criteria are comparable only among candidates fitted under the same settings. Model selection describes both with examples.

Multiple covariance structures in single-level models

Specifying several structures in `model` multiplies the candidates, and the structure is appended to each name. Because the numbers of classes then match several candidates, a candidate must be specified by its full name.

```
crossed_models <- enumerate_lpa(srl, c("cognitive_strategies",
                                       "intrinsic_value", "self_efficacy",
                                       "self_regulation", "test_anxiety"),
                                n_profiles = 2:4, model = c("VVI", "EEE"),
                                n_starts = 5, seed = 1)
#> 1 of 6 candidates reached their best log likelihood from one start only (profiles4_EEE); refit those you cons
crossed_models
#> Class enumeration: 6 candidate models (LPA)
#>      candidate n_profiles model log_likelihood n_parameters  aic bic_groups icl_individual
#>  profiles2_VVI          2   VVI           -889           21 1821       1898           1940
#>  profiles3_VVI          3   VVI           -736           32 1535       1654           1714
#>  profiles4_VVI          4   VVI           -683           43 1451       1611           1728
#>  profiles2_EEE          2   EEE           -601           26 1254       1350           1457
#>  profiles3_EEE          3   EEE           -593           32 1250       1369           1609
#>  profiles4_EEE          4   EEE           -582           38 1240       1381           1574
#>  profile_entropy avepp_min avepp_max avepp_mean avepp_below converged boundary n_best_replicated
#>            0.899     0.969     0.979      0.974          0      TRUE    FALSE                5
#>            0.909     0.936     0.975      0.959          0      TRUE    FALSE                5
#>            0.859     0.885     0.954      0.917          0      TRUE    FALSE                3
#>            0.744     0.880     0.954      0.917          0      TRUE    FALSE                5
#>            0.635     0.796     0.881      0.834          1      TRUE    FALSE                2
#>            0.769     0.818     0.918      0.878          0      TRUE    FALSE                1
#> Every column, unrounded: as.data.frame(x).
#> No candidate is selected automatically. Compare one convention consistently.
compare_models(crossed_models, "profiles3_VVI", "profiles3_EEE")
#> Model comparison
#>
#> Model        Profiles   k   LogLik     BIC    ΔBIC  Weight     ICL  Entropy  Min AvePP  < 0.8
#> ------------ -------- -- -------- -------- ------ ------ -------- -------- --------- -----
#> profiles3_VVI       3  32 -735.74  1654.00 285.32   0.000  1714.03    0.91      0.94     0
#> profiles3_EEE       3  32 -593.08  1368.68   0.00   1.000  1609.46    0.63      0.80     1
```

Latent transition models

With `time`, every candidate is a latent transition model over the occasions, here the order of each student's enrolments in `sequence`. As in the two-level grid above, the within-profile covariances are estimated (`covariance_model = "full"`).

```r
transition_models <- enumerate_classes(course_engagement, activity, "student",
                                       time = "sequence",
                                       covariance_model = "full",
                                       n_profiles = 2:4, n_group_classes = 1,
                                       n_starts = 3, seed = 1)
#> 2 of 3 candidates reached their best log likelihood from one start only (profiles3_groups1, profiles4_groups
transition_models
#> Transition-model enumeration: 3 candidates (LTA)
#>       candidate n_profiles n_group_classes model log_likelihood n_parameters  bic  aic entropy
#>  profiles2_groups1        2              1  <NA>          -8206          43 16613 16499  0.9434
#>  profiles3_groups1        3              1  <NA>          -8172          68 16662 16481  0.9462
#>  profiles4_groups1        4              1  <NA>          -8143          95 16729 16476  0.9585
#>   converged boundary
#>        TRUE    FALSE
#>        TRUE     TRUE
#>        TRUE     TRUE
#> Pick one by its name: get_candidate(x, "<candidate>").
compare_models(transition_models)
#> Model comparison
#>
#> Model          Profiles  k   LogLik     BIC   ΔBIC  Weight     ICL  Entropy  Min AvePP  < 0.8
#> ---------------- -------- -- -------- -------- ------ ------ -------- -------- ---------- -----
#> profiles2_groups1      2 43 -8206.28 16613.08   0.00  1.000 16836.37    0.94     0.98    0
#> profiles3_groups1      3 68 -8172.26 16661.64  48.56  0.000 17006.36    0.95     0.78    1
#> profiles4_groups1      4 95 -8142.89 16728.81 115.74  0.000 17139.21    0.96     0.85    0
```

The BIC selects two profiles, the generating number, and every candidate converges. The latent transition analysis article enumerates profiles and group classes together.

Functions

Call	Result
enumerate_classes(data, vars, id, n_profiles, n_group_classes, model)	The fitted candidate models
enumerate_classes(…, n_starts = , missing = “fiml”, centering = , scaling =)	The same starts, missing-data handling, centring and scaling for every candidate
enumerate_classes(…, boundary = “exclude”)	Starts with a variance at its lower bound set aside for every candidate
enumerate_classes(…, time =), enumerate_lpa(), enumerate_lca()	Transition models, single-level profile and class models
bootstrap_lrt(smaller, larger, data = , iter =)	Bootstrap likelihood ratio test of two neighbouring candidates
as.data.frame(models)	One row per candidate with every column
get_results(models, “best”)	The candidate at the minimum of each criterion
get_results(models, “classification”)	Size, AvePP and entropy of every class of every candidate
get_results(models, “candidates”, cutoff = 0.9)	The table of candidates with another AvePP cutoff
get_results(models, “deltas”, using =)	Decrease in a criterion from one profile fewer and one group class fewer

Call	Result
get_results(models, “knees”, using =)	Maximum chord-distance knee of each line, its range and its stability
plot_enumeration(models), plot_enumeration(models, what = “bic_groups”), what = “all”	Information criteria across the candidates
plot_tree(models, n_group_classes = , model =)	Division of profiles as more are added
plot_enumeration(models, what = “entropy”, “avepp” or “diagnostics”, cutoff =)	Entropy and lowest AvePP across the candidates
plot_selection(models, what = “deltas”, “heatmap”, “smallest” or “trust”, using =)	Decreases, distance from the lowest criterion, smallest class, estimation checks
plot_selection(models)	All of the plots for choosing the number of classes
get_candidate(models, “profiles2_groups2_VVV”), get_candidate(models, 2, 2, model = “VVV”)	One fitted candidate
compare_models(models, “name”, label = “name”, …)	Named candidates side by side
compare_models(models, n_profiles = , n_group_classes =)	All candidates with the given numbers of classes
compare_models(name = fit, name = fit)	Fitted models side by side, on the same data
get_candidate(comparison, “name”)	The model behind a row of a comparison
get_results(comparison, “classification”)	Every class of every compared model

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19, 716–723.

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

Banfield, J. D., & Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49, 803–821.

Biernacki, C., & Govaert, G. (1997). Using the classification likelihood to choose the number of clusters. *Computing Science and Statistics*, 29, 451–457.

Biernacki, C., Celeux, G., & Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 719–725.

Biernacki, C., Celeux, G., & Govaert, G. (2003). Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Computational Statistics & Data Analysis*, 41, 561–575.

Bolck, A., Croon, M., & Hagenaars, J. (2004). Estimating latent structure models with categorical variables: One-step versus three-step estimators. *Political Analysis*, 12(1), 3–27.

Bozdogan, H. (1987). Model selection and Akaike’s information criterion (AIC): The general theory and its analytical extensions. *Psychometrika*, 52, 345–370.

Cavanaugh, J. E. (1999). A large-sample model selection criterion based on Kullback’s symmetric divergence. *Statistics and Probability Letters*, 42, 333–343.

Celeux, G., & Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5), 781–793.

Davison, A. C., & Hinkley, D. V. (1997). *Bootstrap methods and their application*. Cambridge University Press.

Houle, S. A., Morin, A. J. S., & Harvey, J.-F. (2026). Multilevel latent profile analyses: A comprehensive guide. *Organizational Research Methods*. https://doi.org/10.1177/10944281261469432

Lo, Y., Mendell, N. R., & Rubin, D. B. (2001). Testing the number of components in a normal mixture. *Biometrika*, 88, 767–778.

Lukočienė, O., Varriale, R., & Vermunt, J. K. (2010). The simultaneous decision(s) about the number of lower- and higher-level classes in multilevel latent class analysis. *Sociological Methodology*, 40, 247–283.

McLachlan, G. J. (1987). On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture. *Applied Statistics*, 36, 318–324.

McLachlan, G. J., & Peel, D. (2000). *Finite mixture models*. Wiley.

Nylund, K. L., Asparouhov, T., & Muthén, B. O. (2007). Deciding on the number of classes in latent class analysis and growth mixture modeling: A Monte Carlo simulation study. *Structural Equation Modeling*, 14, 535–569.

Nylund-Gibson, K., & Choi, A. Y. (2018). Ten frequently asked questions about latent class analysis. *Translational Issues in Psychological Science*, 4(4), 440–461. https://doi.org/10.1037/tps0000176

Ramaswamy, V., DeSarbo, W. S., Reibstein, D. J., & Robinson, W. T. (1993). An empirical pooling approach for estimating marketing mix elasticities with PIMS data. *Marketing Science*, 12, 103–124.

Saqr, M., & López-Pernas, S. (2021). The longitudinal trajectories of online engagement over a full program. *Computers & Education*, 175, 104325. https://doi.org/10.1016/j.compedu.2021.104325

Saqr, M., López-Pernas, S., Helske, S., & Hrastinski, S. (2023). The longitudinal association between engagement and achievement varies by time, students' profiles, and achievement state: A full program study. *Computers & Education*, 199, 104787. https://doi.org/10.1016/j.compedu.2023.104787

Satopää, V., Albrecht, J., Irwin, D., & Raghavan, B. (2011). Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *31st International Conference on Distributed Computing Systems Workshops* (pp. 166–171). IEEE.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6, 461–464.

Sclove, S. L. (1987). Application of model-selection criteria to some problems in multivariate analysis. *Psychometrika*, 52, 333–343.

Sorgente, A., Caliciuri, R., Robba, M., Lanz, M., & Zumbo, B. D. (2025). A systematic review of latent class analysis in psychology: Examining the gap between guidelines and research practice. *Behavior Research Methods*, 57, 301. https://doi.org/10.3758/s13428-025-02812-1

Spurk, D., Hirschi, A., Wang, M., Valero, D., & Kauffeld, S. (2020). Latent profile analysis: A review and "how to" guide of its application within vocational behavior research. *Journal of Vocational Behavior*, 120, 103445. https://doi.org/10.1016/j.jvb.2020.103445

Tein, J.-Y., Coxe, S., & Cham, H. (2013). Statistical power to detect the correct number of classes in latent profile analysis. *Structural Equation Modeling*, 20(4), 640–657. https://doi.org/10.1080/10705511.2013.824781

Vermunt, J. K. (2024). The Vuong-Lo-Mendell-Rubin test for latent class and latent profile analysis: A note on the different implementations in Mplus and LatentGOLD. *Methodology*, 20(1), 72–83. https://doi.org/10.5964/meth.12467

Wagenmakers, E.-J., & Farrell, S. (2004). AIC model selection using Akaike weights. *Psychonomic Bulletin & Review*, 11, 192–196.

Weller, B. E., Bowen, N. K., & Faubert, S. J. (2020). Latent class analysis: A guide to best practice. *Journal of Black Psychology*, 46(4), 287–311. https://doi.org/10.1177/0095798420930932

Zappia, L., & Oshlack, A. (2018). Clustering trees: A visualization for evaluating clusterings at multiple resolutions. *GigaScience*, 7(7), giy083.