

Package: VaSTra (via r-universe)

October 12, 2026

Title Simple Variables to States to Sequences to Trajectories

Version 0.4.0

Description A small, pipe-friendly implementation of the VaSTra workflow for longitudinal multivariate data. It provides explicit steps to turn variables into states, states into aligned sequences, and sequences into trajectory groups, followed by tidy trajectory descriptions. A single call runs the complete analysis with transparent, overridable automation of data roles and cluster counts, and an evaluation verb compares the fitted clustering against its alternatives. 'Nestimate' provides the default sequence-clustering backend and all sequence plots. No 'TraMineR' dependency is required.

License MIT + file LICENSE

URL <https://sonsoles.me/vasstra/>,
<https://github.com/sonsoleslp/vasstra>,
<https://lamethods.org/book1/chapters/ch11-vasstra/ch11-vasstra.html>

BugReports <https://github.com/sonsoleslp/vasstra/issues>

Encoding UTF-8

Language en-US

LazyData true

Roxygen list(markdown = TRUE)

Depends R (>= 4.1.0)

Imports cluster, graphics, grDevices, mclust (>= 6.1.0), Nestimate (>= 0.8.0), stats, utils

Suggests bslib, cograph (>= 2.4.0), DT, knitr, rmarkdown, shiny, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

Repository <https://mohsaqr.r-universe.dev>

Date/Publication 2026-08-25 11:22:49 UTC
RemoteUrl https://github.com/sonsoleslp/vasstra
RemoteRef HEAD
RemoteSha 04490e71993b7ea835d954189d525e951e5675aa

Contents

as.data.frame.vasstra	2
engagement	3
evaluate	4
fit_indices	5
fit_state_choice	6
fit_trajectory_choice	7
flow_plot	8
launch_app	10
plot.vasstra	11
plot.vasstra_evaluation	12
plot.vasstra_evaluations	13
plot.vasstra_sequences	14
plot.vasstra_state_choices	15
plot.vasstra_states	16
plot.vasstra_trajectories	17
plot.vasstra_trajectory_choices	18
set_labels	19
state_choices	20
state_descriptives	22
step1_states	23
step2_sequences	25
step3_trajectories	27
step4_describe	28
trajectory_choices	29
transition_centrality	31
transition_plot	32
transition_table	34
vasstra	35
Index	39

as.data.frame.vasstra	<i>Convert a VaSSTra Analysis to a Tidy Table</i>
-----------------------	---

Description

Convert a VaSSTra Analysis to a Tidy Table

Usage

```
## S3 method for class 'vasstra'
as.data.frame(
  x,
  row.names = NULL,
  optional = FALSE,
  ...,
  unit = c("subject", "observation", "state_profile", "trajectory")
)
```

Arguments

x	A vasstra analysis.
row.names, optional	Passed to <code>base::as.data.frame()</code> .
...	Additional arguments passed to <code>base::as.data.frame()</code> .
unit	Analysis unit to return: one row per "subject" (default), "observation", "state_profile", or "trajectory".

Value

A data frame at the requested analysis unit.

Examples

```
data("engagement", package = "VaSSTra")
fit <- vasstra(engagement, n_states = 3, n_trajectories = 3)
head(as.data.frame(fit))
as.data.frame(fit, unit = "trajectory")
```

engagement

Longitudinal Student Engagement

Description

Analysis-ready data from the VaSSTra chapter. The object preserves the chapter's eight raw state indicators and adds `_z` versions standardized within `course_id`. It also carries VaSSTra role meta-data, so `id`, `time`, `variables`, `standardization`, and `indicator missingness` do not need to be repeated in common package calls.

Usage

```
engagement
```

Format

A data frame with 1,136 observations and 19 variables:

user_id Pseudonymous student identifier.

course_id Pseudonymous course identifier used for standardization.

sequence_position Course position from 1 through 8.

course_view_count Raw course-view frequency.

forum_consume_count Raw forum-consumption frequency.

forum_contribute_count Raw forum-contribution frequency.

lecture_view_count Raw lecture-view frequency.

course_view_regularity Raw course-view regularity.

session_count Raw session count.

total_duration Raw total activity duration.

active_days Raw number of active days.

course_view_count_z, forum_consume_count_z, forum_contribute_count_z, lecture_view_count_z, course_view_regularity_z

The eight indicators standardized within course_id using the sample standard deviation.

Three singleton courses create 24 intentional missing values; VaSSTra's attached metadata requests within-time median imputation.

Source

[Learning Analytics Methods, Chapter 11](#) and the [lamethods data repository](#).

Examples

```
data(engagement)
str(engagement)
states <- step1_states(engagement, n_states = 3)
```

evaluate

Evaluate a Fitted Clustering

Description

Compares the fitted number of states or trajectories against the alternative counts for the same clustering method and reports per-cluster quality. The result contains one tidy candidate row per compared count, with the fitted and recommended solutions marked, and one tidy row per fitted cluster with its size and mean silhouette width.

Usage

```

evaluate(x, ...)

## S3 method for class 'vasstra_states'
evaluate(x, n_states = NULL, ...)

## S3 method for class 'vasstra_trajectories'
evaluate(x, n_trajectories = NULL, ...)

## S3 method for class 'vasstra'
evaluate(x, ...)

```

Arguments

x	A vasstra_states, vasstra_trajectories, or complete vasstra object.
...	Method-specific arguments.
n_states	Candidate state counts to compare. Defaults to the stored automatic comparison when one exists, otherwise 2 through 6.
n_trajectories	Candidate trajectory counts to compare. Defaults to the stored automatic comparison when one exists, otherwise 2 through 6.

Value

A vasstra_evaluation object (or a vasstra_evaluations pair for a complete fit). `as.data.frame()` returns the candidate comparison; `summary()` returns the per-cluster quality table.

Examples

```

set.seed(1)
data <- expand.grid(student = 1:12, course = 1:3)
level <- rep(c(2, 8, 16), length.out = nrow(data))
data$views <- level + rnorm(nrow(data), sd = 0.4)
data$duration <- level * 3 + rnorm(nrow(data), sd = 0.4)
states <- step1_states(data, n_states = 3)
evaluate(states)

```

Description

Returns the fit statistics of the selected clustering as one tidy row, or one row per compared candidate with `compare = TRUE`. LPA solutions report the complete information-criterion family (log-likelihood, AIC, BIC, SABIC, CAIC, AWE, CLC, KIC, ICL — all on the conventional lower-is-better scale), normalized entropy, and the smallest and largest average posterior class probabilities. Hard clustering methods report their own objectives (total within-cluster sum of squares, PAM objective). Silhouette and group sizes are always included, and columns that do not apply to the fitted method are dropped.

Usage

```
fit_indices(x, ...)

## S3 method for class 'vasstra_states'
fit_indices(x, compare = FALSE, n_states = NULL, ...)

## S3 method for class 'vasstra_trajectories'
fit_indices(x, compare = FALSE, n_trajectories = NULL, ...)

## S3 method for class 'vasstra'
fit_indices(x, step = c("states", "trajectories"), compare = FALSE, ...)
```

Arguments

x	A vasstra_states, vasstra_trajectories, or complete vasstra object.
...	Method-specific arguments.
compare	Return one row per compared candidate instead of only the fitted solution.
n_states	Candidate state counts used when compare = TRUE. Defaults to the stored automatic comparison when one exists, otherwise 2 through 6.
n_trajectories	Candidate trajectory counts used when compare = TRUE. Defaults to the stored automatic comparison when one exists, otherwise 2 through 6.
step	Which clustering of a complete fit to summarize: "states" (default) or "trajectories".

Value

A vasstra_fit_indices data frame: one row for the fitted solution, or one row per candidate (with best and fitted markers) when compare = TRUE.

Examples

```
set.seed(1)
data <- expand.grid(student = 1:12, course = 1:3)
level <- rep(c(2, 8, 16), length.out = nrow(data))
data$views <- level + rnorm(nrow(data), sd = 0.4)
data$duration <- level * 3 + rnorm(nrow(data), sd = 0.4)
states <- step1_states(data, n_states = 3)
fit_indices(states)
fit_indices(states, compare = TRUE)
```

fit_state_choice

Fit One State Choice

Description

Fits a candidate from `state_choices()`, selected by candidate_id, by any combination of n_states, method, and lpa_model, or — when nothing is specified — the recommended candidate.

Usage

```
fit_state_choice(
  choices,
  candidate_id = NULL,
  labels = NULL,
  state = "state",
  n_states = NULL,
  method = NULL,
  lpa_model = NULL
)
```

Arguments

choices	A <code>vasstra_state_choices</code> object from <code>state_choices()</code> .
candidate_id	Optional explicit candidate number to fit.
labels	Optional state labels ordered from low to high profile.
state	Name of the state column created in the returned data.
n_states	Optional state count used to select the candidate.
method	Optional clustering method used to select the candidate.
lpa_model	Optional LPA covariance model used to select the candidate.

Value

A `vasstra_states` object from `step1_states()` with the selected candidate and complete comparison table recorded in diagnostics.

Examples

```
data <- expand.grid(student = 1:12, course = 1:3)
group <- rep(rep(1:3, each = 4), times = 3)
data$views <- group * 5 + data$course * 0.01
data$duration <- group * 10 - data$course * 0.01
choices <- state_choices(
  data, "student", "course", c("views", "duration"),
  n_states = 2:3, method = "kmeans"
)
fit_state_choice(choices)           # the recommended candidate
fit_state_choice(choices, n_states = 3)
```

fit_trajectory_choice *Fit One Trajectory Choice*

Description

Fits a candidate from `trajectory_choices()`, selected by `candidate_id`, by any combination of `n_trajectories`, `dissimilarity`, and `method`, or — when nothing is specified — the recommended candidate.

Usage

```
fit_trajectory_choice(
  choices,
  candidate_id = NULL,
  labels = NULL,
  n_trajectories = NULL,
  dissimilarity = NULL,
  method = NULL
)
```

Arguments

choices	A <code>vasstra_trajectory_choices</code> object.
candidate_id	Optional explicit candidate number to fit.
labels	Optional trajectory labels.
n_trajectories	Optional trajectory count used to select the candidate.
dissimilarity	Optional sequence distance used to select the candidate.
method	Optional clustering method used to select the candidate.

Value

A `vasstra_trajectories` object with the selected candidate and complete comparison table recorded in diagnostics.

Examples

```
sequences <- step2_sequences(
  data.frame(
    id = rep(1:6, each = 3),
    time = rep(1:3, 6),
    state = rep(c("A", "A", "A", "B", "B", "B"), each = 3)
  ),
  "id", "time", "state"
)
choices <- trajectory_choices(
  sequences, n_trajectories = 2, dissimilarity = "hamming",
  method = "pam"
)
fit_trajectory_choice(choices)
```

Description

Draws how subjects move between states from one time point to the next, using `cograph::plot_alluvial()` for aggregated bands and `cograph::plot_trajectories()` for individually tracked lines. Sequence index, distribution, and heatmap views remain with `plot()` and `Nestimate`; `flow_plot()` answers the complementary question of *where movement goes*, which a sequence plot ordered by similarity cannot show.

Usage

```
flow_plot(x, ...)

## S3 method for class 'vasstra_sequences'
flow_plot(
  x,
  type = c("alluvial", "individual"),
  colors = NULL,
  main = NULL,
  color_by = NULL,
  bundle = "auto",
  bundle_max = 50,
  ...
)

## S3 method for class 'vasstra_trajectories'
flow_plot(
  x,
  type = c("alluvial", "individual"),
  colors = NULL,
  main = NULL,
  color_by = NULL,
  bundle = "auto",
  bundle_max = 50,
  group = NULL,
  ...
)

## S3 method for class 'vasstra'
flow_plot(x, ...)
```

Arguments

<code>x</code>	A <code>vasstra_sequences</code> , <code>vasstra_trajectories</code> , or <code>vasstra</code> object.
<code>...</code>	Additional arguments passed to <code>cograph::plot_alluvial()</code> or <code>cograph::plot_trajectories()</code> .
<code>type</code>	"alluvial" (default) for aggregated flow bands whose width is the number of subjects making that move, or "individual" for one line per subject.
<code>colors</code>	Optional colors, one per state, in state order. Defaults to the shared <code>VaSSTra</code> palette.

main	Optional plot title. A type-specific title is used by default.
color_by	State that gives a flow its color: "source" (default for "alluvial") or "destination". Individual lines also accept "first" (the default, coloring each subject by the state they start in) and "last".
bundle	Line bundling for type = "individual". "auto" (default) bundles only when subjects outnumber bundle_max lines, FALSE draws every subject, and a number sets the subjects represented by one line. Ignored by type = "alluvial".
bundle_max	Largest number of lines drawn before "auto" bundling starts. Default 50.
group	Optional trajectory label. Restricts the plot to the subjects of one trajectory, which is how a single group's movement is inspected; flow plots draw one panel and cannot be faceted.

Details

State colors, state order, and time labels are taken from the fitted object, so a flow plot is directly comparable with the sequence heatmap and the state profiles.

Value

A ggplot object. Unlike the base-graphics `plot()` methods, which draw immediately and return their tidy data invisibly, this is returned visibly so that it prints at the console.

See Also

`plot.vasstra_sequences()` for the Nestimate sequence views.

Examples

```
if (requireNamespace("cograph", quietly = TRUE)) {
  data("engagement", package = "VaSSTra")
  fit <- vasstra(
    engagement,
    state_labels = c("Disengaged", "Average", "Active")
  )
  flow_plot(fit)
  flow_plot(fit, type = "individual")
  flow_plot(fit, group = "Trajectory 1")
}
```

launch_app

Launch the Interactive VaSSTra App

Description

Opens a Shiny application that runs the complete VaSSTra workflow interactively: load data, map the subject, time, and indicator roles, fit states and trajectories with automated or explicit counts, inspect evaluation panels and tidy fit indices, rename groups, and export every tidy table. The app requires the suggested shiny and DT packages.

Usage

```
launch_app(...)
```

Arguments

... Passed to [shiny::runApp\(\)](#).

Value

Called for its side effect of running the app.

Examples

```
if (interactive()) {
  launch_app()
}
```

plot.vasstra

Plot a Complete VaSSTra Analysis

Description

Plot a Complete VaSSTra Analysis

Usage

```
## S3 method for class 'vasstra'
plot(x, which = c("trajectories", "states", "sequences"), ...)
```

Arguments

x A vasstra object.

which One of "trajectories", "states", or "sequences".

... Passed to the selected step's plot method. Sequence and trajectory plots are rendered by Nestimate. For the trajectory step, type accepts a vector of views (e.g. type = c("transition", "index", "distribution")) to draw a grid with one row per trajectory; see [plot.vasstra_trajectories\(\)](#).

Value

The selected plot method's invisible result.

Examples

```
data("engagement", package = "VaSSTra")
fit <- vasstra(engagement, n_states = 3, n_trajectories = 3)
plot(fit)
plot(fit, which = "states", type = "profile")
plot(fit, which = "sequences")

if (requireNamespace("cograph", quietly = TRUE)) {
  plot(fit, type = c("transition", "index", "distribution"))
}
```

```
plot.vasstra_evaluation
```

Plot a Clustering Evaluation

Description

Draws the selection curve, per-cluster silhouette widths, and cluster sizes for one `evaluate()` result. The default "summary" layout places the three panels side by side.

Usage

```
## S3 method for class 'vasstra_evaluation'
plot(
  x,
  type = c("summary", "curve", "silhouette", "sizes"),
  colors = NULL,
  main = NULL,
  ...
)
```

Arguments

<code>x</code>	A <code>vasstra_evaluation</code> object from <code>evaluate()</code> .
<code>type</code>	One of "summary" (default), "curve", "silhouette", or "sizes".
<code>colors</code>	Optional colors, one per fitted cluster. The selection curve uses the first color.
<code>main</code>	Optional plot title. A panel-specific title is used by default.
<code>...</code>	Additional graphical arguments passed to the panel functions.

Value

The tidy data drawn by the requested panel, invisibly. The "summary" layout returns both the candidate and cluster tables.

Examples

```
data("engagement", package = "VaSSTra")
fit <- vasstra(engagement, n_states = 3, n_trajectories = 3)
evaluation <- evaluate(fit$trajectories)
plot(evaluation)
plot(evaluation, type = "silhouette")
```

```
plot.vasstra_evaluations
```

Plot the Evaluations of a Complete Fit

Description

Draws one row of evaluation panels per clustering step: the state clustering on top and the trajectory clustering below.

Usage

```
## S3 method for class 'vasstra_evaluations'
plot(x, colors = NULL, ...)
```

Arguments

<code>x</code>	A <code>vasstra_evaluations</code> object from <code>evaluate()</code> on a complete <code>vasstra</code> fit.
<code>colors</code>	Optional colors recycled within each evaluation row. By default the state row reuses the palette pinned on the fit (<code>state_colors</code>), so it matches every other state plot, while the trajectory row keeps a distinct palette so the two groupings are not read as corresponding.
<code>...</code>	Additional graphical arguments passed to the panel functions.

Value

The evaluated candidate and cluster tables, invisibly.

Examples

```
data("engagement", package = "VaSSTra")
fit <- vasstra(engagement, n_states = 3, n_trajectories = 3)
plot(evaluate(fit))
```

plot.vasstra_sequences

Plot VaSSTra State Sequences with Nestimate

Description

Delegates sequence index, distribution, and heatmap rendering to [Nestimate::sequence_plot\(\)](#).

Usage

```
## S3 method for class 'vasstra_sequences'
plot(
  x,
  type = c("heatmap", "distribution", "index"),
  colors = NULL,
  main = NULL,
  sort = "lcs",
  na = x$diagnostics$n_missing > 0L,
  ...
)
```

Arguments

x	A vasstra_sequences object.
type	One of "heatmap" (default), "distribution", or "index". The heatmap shows every aligned sequence at full resolution and is the most informative single view of the complete cohort.
colors	Optional colors, one per state, passed to Nestimate as state_colors.
main	Plot title. A type-specific title is used by default.
sort	Sequence ordering passed to Nestimate::sequence_plot() .
na	Show missing cells as a separate distribution band. By default, this is TRUE only when the sequence data contain NA.
...	Additional arguments passed to Nestimate::sequence_plot() .

Value

The value returned by [Nestimate::sequence_plot\(\)](#).

See Also

[flow_plot\(\)](#) for alluvial and individual flow views.

Examples

```
data("engagement", package = "VaSTra")
fit <- vasstra(engagement, n_states = 3, n_trajectories = 3)
plot(fit$sequences)
plot(fit$sequences, type = "distribution")
```

```
plot.vasstra_state_choices
```

Plot State-Clustering Choices

Description

Draws a compact model-selection plot from `state_choices()`. Each line is one clustering method (and LPA covariance model). A ring marks the optimal eligible value of the plotted metric within each group; hollow points mark candidates that fail the requested size constraints.

Usage

```
## S3 method for class 'vasstra_state_choices'
plot(
  x,
  metric = c("silhouette", "bic", "aic", "classification_entropy", "mean_uncertainty",
    "min_size", "size_ratio", "bic_native", "icl_native"),
  colors = NULL,
  main = NULL,
  groups = NULL,
  show_legend = TRUE,
  ...
)
```

Arguments

<code>x</code>	A <code>vasstra_state_choices</code> object.
<code>metric</code>	Candidate metric to plot.
<code>colors</code>	Optional colors, one per displayed method/model.
<code>main</code>	Optional plot title.
<code>groups</code>	Optional character vector of method/model labels to display, such as "kmeans" or "lpa/EEI".
<code>show_legend</code>	Show the group and marker legends.
<code>...</code>	Additional arguments passed to <code>graphics::plot()</code> .

Value

The tidy candidate data used in the plot, invisibly.

Examples

```
data("engagement", package = "VaSSTra")
options <- state_choices(engagement, n_states = 2:4, method = "kmeans")
plot(options)
plot(options, metric = "silhouette")
```

plot.vasstra_states *Plot Estimated VaSSTra States*

Description

Plot Estimated VaSSTra States

Usage

```
## S3 method for class 'vasstra_states'
plot(
  x,
  colors = NULL,
  main = NULL,
  ...,
  type = c("profile", "heatmap", "bars", "sizes", "variables", "all"),
  scale = c("standardized", "original"),
  by_time = TRUE
)
```

Arguments

x	A vasstra_states object.
colors	Optional colors, one per state. Used by profile, bar, and size plots; heatmaps use a value-based sequential or diverging palette.
main	Optional plot title. A type-specific title is used by default.
...	Additional graphical arguments passed to <code>graphics::matplot()</code> , <code>graphics::image()</code> , <code>graphics::barplot()</code> , or <code>graphics::boxplot()</code> .
type	One of "profile" (default), "heatmap", "bars", "sizes", "variables", or "all" for a two-by-two overview of the first four. "variables" draws grouped boxplots of the clustering indicators, one panel per time point and one box per state within each indicator, so it shows whether each indicator keeps separating the states at every point rather than only on average.
scale	Show state profiles on the "standardized" (default) or "original" indicator scale. Ignored by type = "sizes" and type = "variables" (the latter always uses the scale the states were clustered on).
by_time	Used only by type = "variables": draw one panel per time point (TRUE, default) or pool the time points into a single panel (FALSE).

Value

For profile, bar, heatmap, and overview plots, the plotted indicator-by-state matrix, invisibly. For a size plot, a tidy state-size data frame. For type = "variables", a tidy time-by-variable-by-state table of medians, invisibly.

Examples

```
data("engagement", package = "VaSSTra")
states <- step1_states(engagement, n_states = 3)
plot(states)
plot(states, type = "sizes")
plot(states, type = "variables")
plot(states, type = "variables", by_time = FALSE)
plot(states, type = "all")
```

```
plot.vasstra_trajectories
```

Plot VaSSTra Trajectories with Nestimate

Description

Delegates grouped sequence index, distribution, and heatmap rendering to [Nestimate::sequence_plot\(\)](#) and per-trajectory transition networks to [transition_plot\(\)](#).

Usage

```
## S3 method for class 'vasstra_trajectories'
plot(
  x,
  type = c("index", "distribution", "heatmap", "transition"),
  colors = NULL,
  main = NULL,
  sort = "start",
  na = x$source$diagnostics$n_missing > 0L,
  ...
)
```

Arguments

x	A vasstra_trajectories object.
type	One or more of "index" (the default), "distribution", "heatmap", and "transition". A character vector requests several views and switches to the per-trajectory grid described above; for example type = c("transition", "index", "distribution").
colors	Optional colors, one per state, passed to Nestimate as state_colors and to transition_plot() as node fills.
main	Plot title for the single faceted view. Ignored by the grid, where each row is titled with its trajectory label.

sort	Sequence ordering passed to <code>Nestimate::sequence_plot()</code> . The default is "start" for robust within-trajectory index plots, including trajectories containing one sequence.
na	Show missing cells as a separate distribution band. By default, this is TRUE only when the sequence data contain NA.
...	Additional arguments passed to the sequence panels (<code>Nestimate::sequence_plot()</code>). In the grid, <code>legend = "none"</code> or <code>legend = FALSE</code> suppresses the shared bottom legend.

Details

A single view (the default) is drawn as one faceted figure with a panel per trajectory. Passing several views, or the "transition" network, instead lays out a grid with **one row per trajectory and one column per requested view** (in the order given), which reproduces the familiar per-cluster VaSSTra figure of a transition network beside its sequence index and state-distribution plots. A single shared legend is drawn along the bottom; pass `legend = "none"` (or `legend = FALSE`) to omit it.

Value

For a single view, the value returned by `Nestimate::sequence_plot()`. For the grid, NULL invisibly.

See Also

`flow_plot()` for alluvial and individual flow views, `transition_plot()` for a single transition network.

Examples

```
data("engagement", package = "VaSSTra")
fit <- vasstra(engagement, n_states = 3, n_trajectories = 3)
plot(fit$trajectories)
plot(fit$trajectories, type = "distribution")

if (requireNamespace("cograph", quietly = TRUE)) {
  # One row per trajectory; columns are the requested views.
  plot(fit$trajectories, type = c("transition", "index", "distribution"))
}
```

plot.vasstra_trajectory_choices

Plot Trajectory-Clustering Choices

Description

Draws a compact Nestimate model-selection plot from `trajectory_choices()`. Each line is one distance-method combination. A ring marks the optimal eligible value of the plotted metric within each group; hollow points mark candidates that fail the requested size constraints.

Usage

```
## S3 method for class 'vasstra_trajectory_choices'
plot(
  x,
  metric = c("silhouette", "mean_within_distance", "min_size", "size_ratio"),
  colors = NULL,
  main = NULL,
  groups = NULL,
  show_legend = TRUE,
  ...
)
```

Arguments

x	A vasstra_trajectory_choices object.
metric	One of "silhouette", "mean_within_distance", "min_size", or "size_ratio".
colors	Optional colors, one per displayed distance-method pair.
main	Optional plot title.
groups	Optional character vector of distance-method labels to display, such as "lcs + ward.D2".
show_legend	Show the group and marker legends.
...	Additional arguments passed to <code>graphics::plot()</code> .

Value

The tidy candidate data used in the plot, invisibly.

Examples

```
data("engagement", package = "VaSSTra")
sequences <- step2_sequences(step1_states(engagement, n_states = 3))
options <- trajectory_choices(sequences, n_trajectories = 2:4)
plot(options)
```

set_labels

Relabel Fitted States or Trajectories

Description

Renames the groups of an already-fitted object and propagates the new labels through every derived table. Supply either a complete label vector ordered from the first to the last group, or named renames such as `c("State 1" = "Disengaged")` to change only some labels.

Usage

```

set_labels(x, ...)

## S3 method for class 'vasstra_states'
set_labels(x, labels, ...)

## S3 method for class 'vasstra_trajectories'
set_labels(x, labels, ...)

## S3 method for class 'vasstra'
set_labels(x, states = NULL, trajectories = NULL, ...)

```

Arguments

x	A vasstra_states, vasstra_trajectories, or complete vasstra object.
...	Method-specific arguments.
labels	New labels: a complete vector or named partial renames.
states	New state labels for a complete fit (complete vector or named renames). The rename is propagated in place through the fitted states, sequences, trajectories, and description — including the recorded positive and negative states — without changing any fitted value.
trajectories	New trajectory labels for a complete fit.

Value

The relabeled object, with the same class as x.

Examples

```

set.seed(1)
data <- expand.grid(student = 1:12, course = 1:3)
level <- rep(c(2, 8, 16), length.out = nrow(data))
data$views <- level + rnorm(nrow(data), sd = 0.4)
data$duration <- level * 3 + rnorm(nrow(data), sd = 0.4)
states <- step1_states(data, n_states = 3)
states <- set_labels(states, c("Low", "Average", "High"))
summary(states)

```

state_choices

Compare Choices for the Number and Method of States

Description

Fits a tidy grid of state solutions. No solution is silently selected. Hard-clustering recommendations maximize silhouette within each method; LPA recommendations use the explicit `lpa_criterion`, subject to the requested size constraints. Use `fit_state_choice()` with a `candidate_id` after inspecting the table.

Usage

```
state_choices(
  data,
  id = NULL,
  time = NULL,
  variables = NULL,
  n_states = 2:6,
  method = c("lpa", "kmeans", "pam", "ward.D2"),
  lpa_model = "EEI",
  lpa_criterion = c("bic", "aic", "silhouette", "icl_native"),
  standardize = NULL,
  missing = NULL,
  time_levels = NULL,
  n_start = 25L,
  seed = 123L,
  minimum_size = 2L,
  minimum_proportion = 0,
  maximum_size_ratio = Inf
)
```

Arguments

<code>data</code>	A data frame with one row per subject and time point.
<code>id</code>	Name of the subject identifier column. May be omitted when the data carry VaSSTra role metadata.
<code>time</code>	Name of the time or ordering column. May be omitted when the data carry VaSSTra role metadata.
<code>variables</code>	Character vector naming numeric state indicators. May be omitted when the data carry VaSSTra role metadata.
<code>n_states</code>	Whole-number candidate state counts.
<code>method</code>	State-clustering methods to compare. See step1_states() .
<code>lpa_model</code>	Optional mclust covariance models evaluated when method includes "lpa". "EEI" corresponds to tidyLPA model 1.
<code>lpa_criterion</code>	Criterion used only to mark LPA recommendations: conventional "bic" (default) or "aic" (smaller is better), "silhouette" (larger is better), or native mclust "icl_native" (larger is better).
<code>standardize, missing, time_levels, n_start, seed</code>	Passed to step1_states() .
<code>minimum_size</code>	Minimum acceptable number of observations in every state.
<code>minimum_proportion</code>	Minimum acceptable proportion in every state.
<code>maximum_size_ratio</code>	Maximum acceptable largest-to-smallest state-size ratio.

Value

A `vasstra_state_choices` object. `as.data.frame()` returns one tidy row per candidate; `$recommendations` contains one transparent recommendation per method and LPA model.

Examples

```
data <- expand.grid(student = 1:12, course = 1:3)
group <- rep(rep(1:3, each = 4), times = 3)
data$views <- group * 5 + data$course * 0.01
data$duration <- group * 10 - data$course * 0.01
choices <- state_choices(
  data,
  id = "student",
  time = "course",
  variables = c("views", "duration"),
  n_states = 2:4,
  method = c("kmeans", "pam")
)
choices
states <- fit_state_choice(
  choices,
  candidate_id = choices$recommendations$candidate_id[[1L]]
)
```

state_descriptives	<i>Descriptive Statistics per State</i>
--------------------	---

Description

Summarises each clustering indicator within each state, the numeric companion to `plot(x, which = "states", type = "variables")`. By default the indicators are pooled over time; `by_time = TRUE` reports one block per time point.

Usage

```
state_descriptives(x, ...)

## S3 method for class 'vasstra_states'
state_descriptives(
  x,
  scale = c("standardized", "original"),
  by_time = FALSE,
  ...
)

## S3 method for class 'vasstra'
state_descriptives(x, ...)
```

Arguments

x	A vasstra or vasstra_states object.
...	Not used.
scale	Indicator scale: "standardized" (default, the scale the states were clustered on) or "original" (the raw indicators, where a trailing "_z" resolves to its untransformed column).
by_time	Break the summary down per time point (TRUE) or pool over time (FALSE, default).

Value

A tidy data frame with one row per state and indicator (and time point when `by_time = TRUE`): the grouping columns followed by `n`, `mean`, `sd`, `median`, `min`, and `max`. States are ordered by the fit's state order and indicators by their clustering order.

See Also

`plot.vasstra_states()` with `type = "variables"` for the matching boxplots.

Examples

```
data("engagement", package = "VaSTra")
fit <- vasstra(engagement, n_states = 3)
state_descriptives(fit)
state_descriptives(fit, scale = "original")
state_descriptives(fit, by_time = TRUE)
```

step1_states

Step 1: Turn Variables into States

Description

Standardizes numeric indicators and clusters each subject-time observation. The default method is Gaussian-mixture latent profile analysis (LPA); reproducible k-means, PAM, and hierarchical clustering are available through `method`. Clusters are ordered by their average standardized indicator level, then optionally given user-supplied labels.

Usage

```
step1_states(
  data,
  id = NULL,
  time = NULL,
  variables = NULL,
  n_states = "auto",
  labels = NULL,
```

```

state_order = NULL,
state_colors = NULL,
state = "state",
standardize = NULL,
missing = NULL,
time_levels = NULL,
n_start = 25L,
seed = 123L,
method = c("lpa", "kmeans", "pam", "clara", "fanny", "diana", "ward.D2", "ward.D",
  "complete", "average", "single", "mcquitty", "median", "centroid"),
lpa_model = "EEI"
)

```

Arguments

<code>data</code>	A data frame with one row per subject and time point.
<code>id</code>	Name of the subject identifier column. May be omitted when the data carry VaSSTra role metadata.
<code>time</code>	Name of the time or ordering column. May be omitted when the data carry VaSSTra role metadata.
<code>variables</code>	Character vector naming numeric state indicators. May be omitted when the data carry VaSSTra role metadata.
<code>n_states</code>	Number of states to estimate. One number fits exactly that count, several numbers (for example 2:4) compare those candidates with <code>state_choices()</code> and fit the recommended count, and "auto" (default) compares 2 through 6 — or simply matches labels when labels are supplied. Automatic comparison never selects a solution whose smallest state holds under 5 percent of the observations. Compared candidates are kept in <code>diagnostics\$selection</code> and every automated choice is reported with a message.
<code>labels</code>	Optional unique labels, ordered from the lowest to the highest average standardized profile.
<code>state_order</code>	Optional character vector giving the order the states should appear in every plot. It must list the same labels rearranged; the state column is stored as a factor in this order. Defaults to the profile (discovery) order.
<code>state_colors</code>	Optional state palette stored on the result and reused by every plot. A named vector (matched by state) or one colour per state in state order.
<code>state</code>	Name of the state column created in the returned data.
<code>standardize</code>	How to z-score the indicators before clustering: "time" (default for ordinary data) standardizes within each time point, "subject" within each subject (removing every person's own mean and scale, so states reflect deviations from their baseline), "global" across the whole sample, and "none" leaves the raw values. Analysis-ready package data may provide its own default through metadata.
<code>missing</code>	One of "error" (default for ordinary data) or "median". Analysis-ready package data may provide its own default through metadata. Median imputation is performed within time point, with the global variable median as a fallback for an entirely missing time point.

time_levels	Explicit chronological time values. Required for character or unordered-factor time columns.
n_start	Number of random k-means starts.
seed	Reproducible random seed.
method	State-clustering method. Choose "lpa" (default, Gaussian-mixture latent profile analysis via mclust), a partitioning method — "kmeans", "pam", "clara" (a sampling-based PAM that scales to many observations), or "fanny" (fuzzy clustering, hardened to its nearest partition) — or a hierarchical method: "diana" (divisive) and the agglomerative linkages "ward.D2", "ward.D", "complete", "average", "single", "mcquitty", "median", and "centroid". The partitioning and hierarchical methods are selected by average silhouette; only "lpa" uses BIC.
lpa_model	Covariance model used only by method = "lpa". The default "EEI" corresponds to tidyLPA model 1: equal variances and zero covariances.

Value

A `vasstra_states` object. Its data element is the original data with a factor state column; profiles is a tidy state-by-indicator table.

Examples

```
set.seed(1)
data <- expand.grid(student = 1:12, course = 1:3)
level <- rep(c(2, 8, 16), length.out = nrow(data))
data$views <- level + rnorm(nrow(data), sd = 0.4)
data$duration <- level * 3 + rnorm(nrow(data), sd = 0.4)
states <- step1_states(
  data,
  id = "student",
  time = "course",
  variables = c("views", "duration"),
  labels = c("Low", "Average", "High")
)
states
```

step2_sequences

Step 2: Turn States into Sequences

Description

Aligns every subject on the same observed time axis and returns wide sequences plus tidy long data, state distributions, and transitions.

Usage

```
step2_sequences(
  data,
  id = NULL,
  time = NULL,
  state = NULL,
  state_order = NULL,
  state_colors = NULL,
  time_levels = NULL,
  missing = c("error", "explicit", "keep"),
  missing_label = "Missing"
)
```

Arguments

<code>data</code>	A <code>vasstra_states</code> object or a data frame containing states.
<code>id</code>	Subject identifier column. Inferred from <code>vasstra_states</code> , attached role meta-data, or a common identifier name.
<code>time</code>	Time or ordering column. Inferred from <code>vasstra_states</code> , attached role meta-data, or a common time name.
<code>state</code>	State column. Inferred from <code>vasstra_states</code> , or detected as the single categorical non-role column of a plain data frame.
<code>state_order</code>	Optional character vector giving the order the states should appear in every plot. It must list exactly the observed states, rearranged. Defaults to the state factor's levels, or alphabetical order for a character state column.
<code>state_colors</code>	Optional state palette stored on the result and reused by every plot. A named vector (matched by state) or one colour per state in state order. Inherited from a <code>vasstra_states</code> input when not given.
<code>time_levels</code>	Explicit chronological values. Required for character or unordered-factor time when data is a plain data frame.
<code>missing</code>	Structural-gap policy: "error" (default), "explicit" to create a state for missing subject-time cells, or "keep" to preserve NA cells for advanced use.
<code>missing_label</code>	Label used when <code>missing = "explicit"</code> .

Value

A `vasstra_sequences` object with data (wide state sequences), `long_data`, `meta_data`, `distribution`, and `transitions`.

Examples

```
states <- data.frame(
  student = rep(1:3, each = 3),
  course = rep(1:3, 3),
  engagement = c("Low", "Average", "High",
                 "Average", "Average", "High",
                 "Low", "Low", "Average")
)
```

```

)
sequences <- step2_sequences(
  states,
  id = "student",
  time = "course",
  state = "engagement"
)
sequences

```

step3_trajectories *Step 3: Turn Sequences into Trajectories*

Description

Clusters aligned state sequences with Nestimate sequence distances and clustering methods. The default Hamming plus PAM combination follows the Nestimate and Carm sequence-clustering interface. Set `dissimilarity = "lcs"` and `method = "ward.D2"` to follow the worked VaSSTra chapter.

Usage

```

step3_trajectories(
  data,
  n_trajectories = "auto",
  dissimilarity = c("hamming", "osa", "lv", "dl", "lcs", "qgram", "cosine", "jaccard",
    "jw"),
  method = c("pam", "ward.D2", "ward.D", "complete", "average", "single", "mcquitty",
    "median", "centroid"),
  backend = c("Nestimate", "base"),
  labels = NULL,
  seed = 123L
)

```

Arguments

<code>data</code>	A <code>vasstra_sequences</code> object.
<code>n_trajectories</code>	Number of trajectory groups. One number fits exactly that count, several numbers (for example 2:4) compare those candidates with trajectory_choices() and fit the recommended count, and "auto" (default) compares 2 through 6 — or simply matches labels when labels are supplied. Automatic comparison never selects a solution whose smallest group holds under 5 percent of the sequences. Compared candidates are kept in <code>diagnostics\$selection</code> and every automated choice is reported with a message.
<code>dissimilarity</code>	One of "hamming", "osa", "lv", "dl", "lcs", "qgram", "cosine", "jaccard", or "jw".
<code>method</code>	One of "pam", "ward.D2", "ward.D", "complete", "average", "single", "mcquitty", "median", or "centroid".

backend	"Nestimate" (default) uses <code>Nestimate::build_clusters()</code> . Use "base" for the small internal equivalence backend, which supports only Hamming/LCS distances.
labels	Optional unique trajectory labels.
seed	Reproducible random seed passed to the clustering backend.

Value

A `vasstra_trajectories` object with tidy membership, cluster sizes, silhouette, and the complete distance matrix.

Examples

```
sequences <- step2_sequences(
  data.frame(
    id = rep(1:6, each = 3),
    time = rep(1:3, 6),
    state = c(
      "A", "A", "A", "A", "A", "B",
      "B", "B", "B", "B", "B", "A",
      "C", "C", "C", "C", "C", "B"
    )
  ),
  id = "id",
  time = "time",
  state = "state"
)
trajectories <- step3_trajectories(sequences, n_trajectories = 3)
trajectories
```

step4_describe

Step 4: Describe the Trajectories

Description

Produces tidy per-subject sequence indices and trajectory-level state, distribution, and transition summaries. Entropy is normalized Shannon entropy. Complexity is the geometric mean of entropy and transition rate, matching the common sequence complexity formulation. Integrative potential and negative exposure are time-weighted state proportions.

Usage

```
step4_describe(data, positive_states = NULL, negative_states = NULL, omega = 1)
```

Arguments

data	A <code>vasstra_trajectories</code> object.
positive_states	Optional states considered positive.
negative_states	Optional states considered negative.
omega	Positive exponent controlling how strongly later time points are weighted for integrative potential and negative exposure.

Value

A `vasstra_description` object with tidy indices, `trajectory_summary`, `mean_time`, `distribution`, and `transitions`.

Examples

```
long <- data.frame(
  id = rep(1:6, each = 3),
  time = rep(1:3, 6),
  state = c(
    "Low", "Low", "Low", "Low", "Low", "Average",
    "Average", "Average", "Average", "Average", "Average", "Low",
    "High", "High", "High", "High", "High", "Average"
  )
)
trajectories <- long |>
  step2_sequences(id = "id", time = "time", state = "state") |>
  step3_trajectories(n_trajectories = 3)
description <- step4_describe(
  trajectories,
  positive_states = "High",
  negative_states = "Low"
)
description
```

trajectory_choices	<i>Compare Choices for the Number and Method of Trajectories</i>
--------------------	--

Description

Uses `Nestimate::cluster_choice()` to compare a tidy grid of sequence distances, clustering methods, and trajectory counts. Recommendations are made within each distance-method pair; no distance or algorithm is silently chosen for the user.

Usage

```
trajectory_choices(
  data,
  n_trajectories = 2:6,
  dissimilarity = c("hamming", "lcs"),
  method = c("pam", "ward.D2"),
  seed = 123L,
  minimum_size = 2L,
  minimum_proportion = 0,
  maximum_size_ratio = Inf
)
```

Arguments

<code>data</code>	A <code>vasstra_sequences</code> object.
<code>n_trajectories</code>	Whole-number candidate trajectory counts.
<code>dissimilarity</code>	Sequence distances to compare.
<code>method</code>	Clustering methods to compare.
<code>seed</code>	Reproducible seed passed to <code>Nestimate</code> .
<code>minimum_size</code>	Minimum acceptable number of sequences in every group.
<code>minimum_proportion</code>	Minimum acceptable proportion in every group.
<code>maximum_size_ratio</code>	Maximum acceptable largest-to-smallest group-size ratio.

Value

A `vasstra_trajectory_choices` object with one tidy candidate row per `Nestimate` fit and transparent recommendations within method-distance combinations.

Examples

```
sequences <- step2_sequences(
  data.frame(
    id = rep(1:6, each = 3),
    time = rep(1:3, 6),
    state = rep(c("A", "A", "A", "B", "B", "B"), each = 3)
  ),
  "id", "time", "state"
)
choices <- trajectory_choices(
  sequences,
  n_trajectories = 2:3,
  dissimilarity = "hamming",
  method = c("pam", "ward.D2")
)
choices
```

transition_centrality *State Transition Network Centralities*

Description

Tidy centralities of the state transition network, one row per state. The network is built by `Nestimate::build_tna()` and the measures come from `Nestimate::net_centrality()`, so they match `tna::centralities()`.

Usage

```
transition_centrality(x, ...)

## S3 method for class 'vasstra_sequences'
transition_centrality(
  x,
  measures = c("InStrength", "OutStrength"),
  weights = c("probability", "count"),
  loops = FALSE,
  ...
)

## S3 method for class 'vasstra_trajectories'
transition_centrality(
  x,
  measures = c("InStrength", "OutStrength"),
  weights = c("probability", "count"),
  loops = FALSE,
  group = NULL,
  ...
)

## S3 method for class 'vasstra'
transition_centrality(x, ...)
```

Arguments

x	A <code>vasstra_sequences</code> , <code>vasstra_trajectories</code> , or <code>vasstra</code> object.
...	Not used.
measures	Centrality measures to compute, passed to <code>Nestimate::net_centrality()</code> . Defaults to <code>c("InStrength", "OutStrength")</code> ; "all" returns every built-in measure.
weights	"probability" (default) uses row-normalized transition probabilities from <code>Nestimate::build_tna()</code> ; "count" uses raw transition counts from <code>Nestimate::build_ftna()</code> .
loops	Include self-transitions in the computation. Default FALSE, matching <code>Nestimate::net_centrality()</code> .
group	Optional trajectory label restricting the network to one trajectory's subjects.

Details

In-strength is the total incoming transition weight, so the state with the largest in-strength is the one the cohort most often moves *into*. Self-transitions are excluded by default: with `loops = TRUE` a persistent state scores highly merely because its members stay put, which is a different claim from attracting movement.

Value

A tidy data frame with one row per state and one column per requested measure.

See Also

[transition_plot\(\)](#) to draw the network.

Examples

```
data("engagement", package = "VaSTra")
fit <- vasstra(engagement, n_states = 3, n_trajectories = 3)
transition_centrality(fit)
transition_centrality(fit, weights = "count")
```

transition_plot

Plot the State Transition Network

Description

Draws states as nodes and transitions as directed edges, with node size encoding a centrality of the transition network. The network comes from [Nestimate::build_tna\(\)](#) and the centralities from [Nestimate::net_centrality\(\)](#); [cograph::splot\(\)](#) draws it, recognising the Nestimate object and applying TNA styling, node labels, and initial-probability donuts automatically.

Usage

```
transition_plot(x, ...)

## S3 method for class 'vasstra_sequences'
transition_plot(
  x,
  size = "InStrength",
  weights = c("probability", "count"),
  loops = FALSE,
  size_range = c(8, 18),
  sequences = FALSE,
  colors = NULL,
  main = NULL,
  ...
)
```

```
## S3 method for class 'vasstra_trajectories'
transition_plot(
  x,
  size = "InStrength",
  weights = c("probability", "count"),
  loops = FALSE,
  size_range = c(8, 18),
  sequences = FALSE,
  colors = NULL,
  main = NULL,
  group = NULL,
  ...
)

## S3 method for class 'vasstra'
transition_plot(x, ...)
```

Arguments

x	A <code>vasstra_sequences</code> , <code>vasstra_trajectories</code> , or <code>vasstra</code> object.
...	Additional arguments passed to <code>cograph::splot()</code> , such as <code>layout</code> , <code>threshold</code> , or <code>edge_label_size</code> (which defaults to a value slightly larger than <code>cograph</code> 's, so the transition probabilities are easy to read).
size	Centrality measure that sets node size, or "none" to draw every node at one size. Default "InStrength".
weights	"probability" (default) uses row-normalized transition probabilities from <code>Nestimate::build_tna()</code> ; "count" uses raw transition counts from <code>Nestimate::build_ftna()</code> .
loops	Include self-transitions in the computation. Default FALSE, matching <code>Nestimate::net centrality()</code> .
size_range	Smallest and largest node size. The measure is mapped onto this range anchored at zero, so a state with half the in-strength of the largest draws halfway up the range and sizes stay comparable between plots.
sequences	Draw the state sequences beside the network, the conventional pairing in which the sequences show the raw data and the network summarises its movement. FALSE (default) draws the network alone; TRUE adds an index plot; "index", "heatmap", or "distribution" choose the sequence view. Both panels are drawn on one device, so a state has the same colour in each.
colors	Optional colors, one per state, in state order. Defaults to the shared VaSSTra palette, so the network matches the sequence and state plots. Colors are matched to states by name, so a trajectory that never reaches a state still colours the rest correctly.
main	Optional plot title.
group	Optional trajectory label restricting the network to one trajectory's subjects.

Details

Where `flow_plot()` shows movement resolved over time, `transition_plot()` collapses every time step into one network and asks which states attract movement overall.

Value

A tidy data frame of the plotted centrality and node size, one row per state, invisibly.

See Also

`transition_centrality()` for the numbers without a plot, and `flow_plot()` for time-resolved movement.

Examples

```
if (requireNamespace("cograph", quietly = TRUE)) {
  data("engagement", package = "VaSSTra")
  fit <- vasstra(engagement, n_states = 3, n_trajectories = 3)
  transition_plot(fit)
  transition_plot(fit, weights = "count", size = "OutStrength")
  transition_plot(fit, sequences = TRUE)
}
```

transition_table	<i>State Transition Table</i>
------------------	-------------------------------

Description

The transition counts behind the flow and network views, as a tidy data frame. By default every consecutive move is aggregated over time (the numbers `transition_plot()` draws); `by_time = TRUE` instead returns one block per consecutive time pair, matching the alluvial `flow_plot()`.

Usage

```
transition_table(x, ...)

## S3 method for class 'vasstra_sequences'
transition_table(x, by_time = FALSE, ...)

## S3 method for class 'vasstra_trajectories'
transition_table(x, by_time = FALSE, group = NULL, ...)

## S3 method for class 'vasstra'
transition_table(x, ...)
```

Arguments

<code>x</code>	A <code>vasstra_sequences</code> , <code>vasstra_trajectories</code> , or <code>vasstra</code> object.
<code>...</code>	Not used.
<code>by_time</code>	Return per-time-step transitions (TRUE) or the time-aggregated table (FALSE, default).
<code>group</code>	Optional trajectory label restricting the table to one trajectory's subjects.

Value

A tidy data frame. Aggregated: from, to, count, and probability (row-normalized within from). With `by_time = TRUE`, also from_time and to_time, with probability normalized within each (from_time, from).

See Also

[transition_plot\(\)](#) and [flow_plot\(\)](#) to draw these movements, [transition_centrality\(\)](#) for network centralities.

Examples

```
data("engagement", package = "VaSTra")
fit <- vasstra(engagement, n_states = 3, n_trajectories = 3)
transition_table(fit)
transition_table(fit, by_time = TRUE)
```

vasstra

Run the Complete VaSTra Workflow

Description

A one-call wrapper around [step1_states\(\)](#), [step2_sequences\(\)](#), [step3_trajectories\(\)](#), and [step4_describe\(\)](#). Every argument has a working default, so `vasstra(data)` runs a complete analysis: subject, time, and indicator roles come from explicit arguments, attached role metadata, or common column names, and the numbers of states and trajectories are selected automatically unless given. Every automated decision is reported with a message and recorded in the fit.

Usage

```
vasstra(
  data,
  id = NULL,
  time = NULL,
  variables = NULL,
  state = NULL,
  n_states = "auto",
  n_trajectories = "auto",
```

```

state_labels = NULL,
state_order = NULL,
state_colors = NULL,
trajectory_labels = NULL,
state_name = "state",
standardize = NULL,
indicator_missing = NULL,
sequence_missing = c("error", "explicit", "keep"),
missing_label = "Missing",
time_levels = NULL,
dissimilarity = c("hamming", "osa", "lv", "dl", "lcs", "qgram", "cosine", "jaccard",
  "jw"),
cluster_method = c("pam", "ward.D2", "ward.D", "complete", "average", "single",
  "mcquitty", "median", "centroid"),
backend = c("Nestimate", "base"),
positive_states = NULL,
negative_states = NULL,
omega = 1,
n_start = 25L,
seed = 123L,
state_method = c("lpa", "kmeans", "pam", "clara", "fanny", "diana", "ward.D2",
  "ward.D", "complete", "average", "single", "mcquitty", "median", "centroid"),
lpa_model = "EEI"
)

```

Arguments

<code>data</code>	A longitudinal data frame.
<code>id</code>	Subject identifier column. May be omitted when the data carry VaSSTra role metadata or use a common identifier name.
<code>time</code>	Time or ordering column. May be omitted when the data carry VaSSTra role metadata or use a common time name.
<code>variables</code>	Numeric state indicators. Use either <code>variables</code> or <code>state</code> , but not both. When both are omitted, the numeric non-role columns are used (columns ending in <code>_z</code> are preferred when present).
<code>state</code>	Existing state column. Use either <code>state</code> or <code>variables</code> .
<code>n_states</code>	Number of states when <code>variables</code> is supplied: one number, a candidate vector such as 2:4 to compare and fit the recommended count, or "auto" (default). Supplied <code>state_labels</code> determine the count under "auto".
<code>n_trajectories</code>	Number of trajectory groups: one number, a candidate vector to compare, or "auto" (default). Supplied <code>trajectory_labels</code> determine the count under "auto".
<code>state_labels</code>	Optional labels ordered from low to high profile.
<code>state_order</code>	Optional character vector giving the order the states should appear in every plot (stacking, legends, transition nodes, flow nodes). It must list exactly the same

	states as <code>state_labels</code> (or the observed states, for precomputed input), rearranged. When supplied, the state column becomes a factor in this order. Useful when the cluster discovery order does not match the order that reads best.
<code>state_colors</code>	Optional state palette stored on the fit and reused by every plot without repeating colors. Either a named vector (<code>c(Low = "grey", High = "red", ...)</code>), matched by state name and robust to <code>state_order</code> and renaming) or one colour per state in state order. An explicit <code>colors</code> argument to a plot still overrides it.
<code>trajectory_labels</code>	Optional labels for stable trajectory groups.
<code>state_name</code>	Output column name when states are estimated.
<code>standardize</code>	State-indicator standardization: "time" (within each time point), "subject" (within each subject, relative to their own baseline), "global", or "none". Analysis-ready package data may provide its own default.
<code>indicator_missing</code>	Indicator missingness policy passed to step 1. Analysis-ready package data may provide its own default.
<code>sequence_missing</code>	Structural sequence-gap policy passed to step 2.
<code>missing_label</code>	Explicit missing-state label.
<code>time_levels</code>	Explicit chronological values when needed.
<code>dissimilarity</code>	Sequence distance passed to <code>step3_trajectories()</code> .
<code>cluster_method</code>	Trajectory clustering method.
<code>backend</code>	Clustering backend: "Nestimate" (default) or "base".
<code>positive_states</code>	Optional states considered positive.
<code>negative_states</code>	Optional states considered negative.
<code>omega</code>	Later-time weighting exponent for step 4.
<code>n_start</code>	Number of k-means starts.
<code>seed</code>	Reproducible base seed.
<code>state_method</code>	State-clustering method passed to <code>step1_states()</code> . The default "lpa" estimates Gaussian-mixture latent profiles.
<code>lpa_model</code>	mclust covariance model used when <code>state_method = "lpa"</code> . The default "EEI" is tidyLPA model 1.

Value

A `vasstra` object containing all four fitted step objects.

Examples

```
set.seed(1)
data <- expand.grid(student = 1:15, course = 1:4)
latent <- ceiling(data$student / 5)
```

```
data$views <- latent * 5 + rnorm(nrow(data), sd = 0.4)
data$duration <- latent * 10 + rnorm(nrow(data), sd = 0.4)

# Fully automated: roles are detected and the counts are selected.
fit <- vasstra(data)
fit

# Labels imply the counts; other choices named explicitly.
fit <- vasstra(
  data,
  state_labels = c("Low", "Average", "High"),
  positive_states = "High",
  negative_states = "Low"
)
fit
```

Index

- * **datasets**
 - engagement, [3](#)
- as.data.frame.vasstra, [2](#)
- base::as.data.frame(), [3](#)
- cograph::plot_alluvial(), [9](#)
- cograph::plot_trajectories(), [9](#)
- cograph::splot(), [32](#), [33](#)
- engagement, [3](#)
- evaluate, [4](#)
- evaluate(), [12](#), [13](#)
- fit_indices, [5](#)
- fit_state_choice, [6](#)
- fit_state_choice(), [20](#)
- fit_trajectory_choice, [7](#)
- flow_plot, [8](#)
- flow_plot(), [14](#), [18](#), [34](#), [35](#)
- graphics::barplot(), [16](#)
- graphics::boxplot(), [16](#)
- graphics::image(), [16](#)
- graphics::matplot(), [16](#)
- graphics::plot(), [15](#), [19](#)
- launch_app, [10](#)
- Nestimate::build_clusters(), [28](#)
- Nestimate::build_ftna(), [31](#), [33](#)
- Nestimate::build_tna(), [31–33](#)
- Nestimate::cluster_choice(), [29](#)
- Nestimate::net_centrality(), [31–33](#)
- Nestimate::sequence_plot(), [14](#), [17](#), [18](#)
- plot(), [9](#), [10](#)
- plot.vasstra, [11](#)
- plot.vasstra_evaluation, [12](#)
- plot.vasstra_evaluations, [13](#)
- plot.vasstra_sequences, [14](#)
- plot.vasstra_sequences(), [10](#)
- plot.vasstra_state_choices, [15](#)
- plot.vasstra_states, [16](#)
- plot.vasstra_states(), [23](#)
- plot.vasstra_trajectories, [17](#)
- plot.vasstra_trajectories(), [11](#)
- plot.vasstra_trajectory_choices, [18](#)
- set_labels, [19](#)
- shiny::runApp(), [11](#)
- state_choices, [20](#)
- state_choices(), [6](#), [7](#), [15](#), [24](#)
- state_descriptives, [22](#)
- step1_states, [23](#)
- step1_states(), [7](#), [21](#), [35](#), [37](#)
- step2_sequences, [25](#)
- step2_sequences(), [35](#)
- step3_trajectories, [27](#)
- step3_trajectories(), [35](#), [37](#)
- step4_describe, [28](#)
- step4_describe(), [35](#)
- trajectory_choices, [29](#)
- trajectory_choices(), [7](#), [18](#), [27](#)
- transition_centrality, [31](#)
- transition_centrality(), [34](#), [35](#)
- transition_plot, [32](#)
- transition_plot(), [17](#), [18](#), [32](#), [34](#), [35](#)
- transition_table, [34](#)
- vasstra, [35](#)