

Package: segen (via r-universe)

October 7, 2026

Type Package

Title Sequence Generalization Through Similarity Network

Version 2.0.1

Description Proposes an application for sequence prediction
generalizing the similarity within the network of previous
sequences.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.6)

Imports stats, utils, graphics, grDevices, parallel

URL https://rpubs.com/giancarlo_vercellino/segen

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

RoxygenNote 7.3.3

NeedsCompilation no

Author Giancarlo Vercellino [aut, cre]

Maintainer Giancarlo Vercellino <giancarlo.vercellino@gmail.com>

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

Date/Publication 2026-09-07 17:40:02 UTC

RemoteUrl <https://github.com/cran/segen>

RemoteRef HEAD

RemoteSha 600cad57f4d602372ed4669bc0501af69bf7aff9

Contents

plot.segen_plot	2
segen	2
time_features	4
Index	6

plot.segen_plot	<i>Draw a segen forecast</i>
-----------------	------------------------------

Description

Draw a segen forecast

Usage

```
## S3 method for class 'segen_plot'
plot(x, ...)

## S3 method for class 'segen_plot'
print(x, ...)
```

Arguments

- x A forecast plot returned in best_model\$plots.
- ... Additional arguments passed to the base plot function.

Value

The plot object, invisibly.

segen	<i>segen</i>
-------	--------------

Description

Sequence Generalization Through Similarity Network

Usage

```
segen(
  df,
  seq_len = NULL,
  similarity = NULL,
  dist_method = NULL,
  rescale = NULL,
  smoother = FALSE,
  ci = 0.8,
  error_scale = "naive",
  error_benchmark = "naive",
  n_windows = 10,
  n_samp = 30,
  dates = NULL,
  seed = 42,
  use_parallel = FALSE,
  parallel_workers = NULL
)
```

Arguments

<code>df</code>	data.frame of time features (all numeric OR all categorical).
<code>seq_len</code>	integer, forecasting horizon. If NULL, auto-sampled.
<code>similarity</code>	numeric in (0,1), similarity quantile. If NULL, sampled.
<code>dist_method</code>	character. Options: "euclidean", "manhattan", "maximum", "minkowski", "correlation", "dtw". If NULL, sampled from the six internal distance methods.
<code>rescale</code>	logical, rescale weights before normalization.
<code>smoother</code>	logical, apply loess smoothing for numeric features.
<code>ci</code>	numeric in (0,1), confidence level.
<code>error_scale</code>	"naive" or "deviation".
<code>error_benchmark</code>	"naive" or "average".
<code>n_windows</code>	integer, rolling validation windows.
<code>n_samp</code>	integer, random search samples.
<code>dates</code>	Date vector aligned with rows of <code>df</code> (optional).
<code>seed</code>	integer, RNG seed.
<code>use_parallel</code>	logical, use standard-library PSOCK workers for parallel exploration.
<code>parallel_workers</code>	NULL or integer, number of workers when parallel.

Value

list with exploration, history, best_model, time_log.

This function returns a list including:

- exploration: list of all not-null models, complete with predictions and error metrics
- history: a table with the sampled models, hyper-parameters, validation errors
- best_model: results for the best selected model according to the weighted average rank, including:
 - predictions: for continuous variables, min, max, q25, q50, q75, quantiles at selected ci, mean, sd, mode, skewness, kurtosis, IQR to range, risk ratio, upside probability and divergence for each point fo predicted sequences; for factor variables, min, max, q25, q50, q75, quantiles at selected ci, proportions, difformity (deviation of proportions normalized over the maximum possible deviation), entropy, upgrade probability and divergence for each point fo predicted sequences
 - testing_errors: testing errors for each time feature for the best selected model (for continuous variables: me, mae, mse, rmsse, mpe, mape, rmae, rrmse, rame, mase, smse, sce, gmrae; for factor variables: czekanowski, tanimoto, cosine, hassebrook, jaccard, dice, canberra, gower, lorentzian, clark)
 - plots: standard plots with confidence interval for each time feature
- time_log

Author(s)

Giancarlo Vercellino <giancarlo.vercellino@gmail.com>

Maintainer: Giancarlo Vercellino <giancarlo.vercellino@gmail.com>

See Also

Useful links:

- https://rpubs.com/giancarlo_vercellino/segem

Examples

```
segem(time_features[, 1, drop = FALSE], seq_len = 30, similarity = 0.7, n_windows = 3, n_samp = 1)
```

time_features

time features example: IBM and Microsoft Close Prices

Description

A data frame with with daily with daily prices for IBM and Microsoft since April 2020

Usage

```
time_features
```

time_features

5

Format

A data frame with 2 columns and 500 rows.

Source

finance.yahoo.com

Index

* **datasets**

time_features, [4](#)

plot.segen_plot, [2](#)

print.segen_plot (plot.segen_plot), [2](#)

segn, [2](#)

segn-package (segn), [2](#)

time_features, [4](#)