Package 'MemoryDecay'

April 15, 2025

Title Fitting Collective Memory Decay Curves

Version 0.1.0

Author Cristian Candia [aut, cre]

Maintainer Cristian Candia <crcandiav@gmail.com>

Description Provides functions to fit and visualize forgetting curves using nonlinear least squares (via nlsLM), LOESS smoothing, and generalized additive models (GAM). Supports biexponential, exponential, and log-normal modulated power-law decay forms. Includes tools for grouped and stratified model fitting, critical time estimation, and model comparison using AIC, BIC, and pseudo-R^2. Offers publication-ready plotting functions for raw, smoothed, and fitted data. Designed for analyzing collective memory and attention decay in cross-sectional, time-series, and survey-based datasets. The package is grounded in recent research on collective memory dynamics. If you use this package, please cite: Candia et al. (2019) <doi:10.1038/s41562-018-0474-5>; Candia & Uzzi (2021) <doi:10.1037/amp0000863>.

Depends R (>= 4.0.0)

Imports dplyr,
ggplot2,
ggtext,
grDevices,
gridExtra,
magrittr,
mgcv,
minpack.lm,
purrr,
reshape2,
rlang,
scales,
stats,
stringi,
stringr,
tibble,
tidyr

Suggests knitr,
rmarkdown

VignetteBuilder knitr

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

R topics documented:

`add_critical_time` *Add critical time to model parameters from biexponential fit*

Description

Computes the critical time t_c from the parameters p, r, and q obtained via the biexponential model. This is the point in time at which the contributions of communicative and cultural memory become equal.

Usage

```
add_critical_time(parameters)
```

Arguments

`parameters`	A data frame containing columns `p`, `r`, and `q`, typically returned by `fit_biexponential_model()`.

Details

The formula used is:

$$t_c = \frac{1}{p + r - q} \log\left(\frac{(p + r)(p - q)}{rq}\right)$$

This function is designed to work with the ‘params‘ output of `fit_biexponential_model`.

Value

A data frame with a new column:

critical_time The estimated time t_c at which cultural memory begins to dominate.

See Also

`fit_biexponential_model`

`aggregate_mean_response` *Aggregate Mean Response by Age and Grouping Variables*

Description

This function computes the average of a response variable (e.g., recall accuracy, citation count) over time or historical age. It is particularly useful for preparing clean, aggregated data for plotting or model fitting in memory decay studies.

Usage

```
aggregate_mean_response(
  data,
  age_var,
  response_var,
  group_var = NULL,
  group_var2 = NULL,
  filter_age = Inf
)
```

Arguments

`data`	A data frame containing memory or attention data (e.g., from surveys, citations, or popularity time series).
`age_var`	Name of the column representing age or time (e.g., years since event or birth).
`response_var`	Name of the numeric response variable to average (e.g., correct recall rate, citations).
`group_var`	(Optional) First grouping variable name (e.g., "same_country", "decay_bin").
`group_var2`	(Optional) Second grouping variable name (e.g., "attention_level", "demographic").
`filter_age`	(Optional) Maximum value of age to include in the output (default: `Inf` = keep all).

Details

You can optionally provide up to two grouping variables to stratify the results: - ‘group_var‘: A primary categorical dimension (e.g., geographic match, decay bin). - ‘group_var2‘: A secondary categorical dimension (e.g., attention level, quantile group).

Both grouping variables are preserved in the output using their original column names. If no grouping is provided, the function returns the global mean response at each time point.

The output always contains a column named ‘age_metric‘ to standardize time/age representation across functions in the package.

Value

A data frame with one row per combination of:

age_metric Numeric value representing time or age.

mean_response The average of the response variable within each group/age.

group_var (Optional) Column named as passed by user.

group_var2 (Optional) Column named as passed by user.

Examples

```
# Aggregate recall by age and location
aggregated <- aggregate_mean_response(
  data = survey_data,
  age_var = "age_metric",
  response_var = "performance_score",
  group_var = "location_flag"
)
```

`assign_decay_bins` *Assign Decay Bins to Cumulative Attention Values (Log-Scaled)*

Description

Assigns each cumulative attention value (e.g., cumulative citations) to a discrete bin for stratified modeling of memory decay. Bins can be defined either: - Manually using custom breakpoints (‘breaks‘), or - Automatically using log-scaled quantization (‘n_bins‘).

Usage

```
assign_decay_bins(
  cumulative_df,
  breaks = NULL,
  n_bins = 3,
  return_labels = TRUE
)
```

Arguments

cumulative_df	A data frame of cumulative values (from compute_cumulative_matrix).
breaks	Optional numeric vector of manual breakpoints (strictly increasing). If provided, n_bins is ignored.
n_bins	Integer. Number of log-spaced bins to compute automatically (default: 3).
return_labels	Logical. If TRUE, returns bins as factor labels (1 = highest); if FALSE, returns numeric bin index.

Details

The input should be a wide-format matrix of cumulative values over time, typically produced by compute_cumulative_matrix. Columns represent items (e.g., papers), and rows represent time steps.

Treatment of Zeros

Items with **zero cumulative attention** are excluded from bin breakpoint computation: - Log-scale binning is undefined at zero (log10(0) = -Inf). - Including zeros would skew binning and reduce stratification resolution.

In this function, **zero values are intentionally excluded** from the binning computation. This is because log-scale binning requires strictly positive values and because zero cumulative attention reflects complete forgetting — which is not comparable in magnitude to any non-zero value. Therefore, items with zero cumulative attention are assigned 'NA' as their decay bin. These zero entries are retained in the output but marked as NA to indicate that they are unbinned. This allows separate modeling of completely forgotten items.

Rationale: Controlling for Preferential Attachment

As described by Candia et al. (2019, *Nature Human Behaviour*), binning items by cumulative attention allows researchers to control for preferential attachment— the tendency of popular items to accumulate even more attention. This enables fair comparisons of memory decay across items with similar total attention.

Value

A data frame with the same shape as cumulative_df, containing bin assignments. Zero-valued entries are returned as NA.

See Also

merge_bins_with_original, reshape_citation_timeseries, process_time_series_bins

Examples

```
mat <- data.frame(A = c(0, 2, 10, 100), B = c(0, 1, 5, 50))
assign_decay_bins(mat, n_bins = 3)
```

compare_model_fits *Plot AIC or BIC Comparison Across Forgetting Models*

Description

Generates a comparative bar chart showing how different forgetting models (biexponential, exponential, log-normal) perform based on the selected information criterion (AIC or BIC).

Usage

```
compare_model_fits(model_comparison, metric = "AIC")
```

Arguments

model_comparison	A data frame returned by `fit_all_models_log()` in the element `$model_comparison`, containing AIC/BIC scores and group identifiers (if applicable).
metric	String. The information criterion to plot: either `"AIC"` or `"BIC"` (default: "AIC").

Details

This function is typically used after fitting all models with `fit_all_models_log`, and helps visually identify the model with the best fit for each group or condition.

Value

A `ggplot2` bar chart comparing model fit quality across groups or levels.

See Also

`fit_all_models_log`, `plot_all_models`, `plot_fitted_decay_for_publication`

compute_cumulative_matrix *Compute Cumulative Attention Over Time*

Description

This function calculates the cumulative attention (e.g., citations) received by each item over time. It takes as input a wide-format time-series matrix — typically generated using `reshape_citation_timeseries` — and returns the element-wise cumulative sum for each item across successive time points.

Usage

```
compute_cumulative_matrix(wide_data)
```

Arguments

`wide_data`	A data frame in wide format, with:
	`time` A numeric column representing the temporal axis (e.g., year or semester).
	Other columns Named after unique item identifiers (e.g., DOIs) containing the attention values (e.g., citation counts).

Details

Each column (except `time`) represents a unique item (e.g., paper or cultural artifact), and each row corresponds to a discrete time point (e.g., year or semester). Missing values (NAs) are replaced with zeros before computing cumulative sums, assuming that missing data corresponds to zero observed attention at that time.

This step is essential before assigning decay bins using `assign_decay_bins`.

Value

A data frame with the same structure (excluding `time`) where each cell contains the cumulative value of attention received by an item up to that time point.

See Also

`reshape_citation_timeseries`, `assign_decay_bins`, `merge_bins_with_original`

Examples

```
# Example input matrix
wide_df <- data.frame(
  time = 1980:1982,
  value.A = c(3, 4, 2),
  value.B = c(5, 6, 1)
)

cum_df <- compute_cumulative_matrix(wide_df)
head(cum_df)
```

`cross_section_data` *Example dataset: Cross-sectional song popularity data (Billboard-based)*

Description

This dataset contains cross-sectional data on the popularity of songs and artists, measured on October 29th, 2016. Each row corresponds to a song that entered the Billboard ranking at some point prior to that date.

Usage

```
cross_section_data
```

Format

A data frame with multiple rows and the following variables:

Song Character. Title of the song.

Artist Character. Name of the artist.

Date Date. The date when the song entered the Billboard ranking (i.e., date of accomplishment).

CurrentPopularity Numeric. Popularity of the song as of October 29th, 2016, measured by streaming volume, standardized and in linear scale.

age Numeric. Time since the song entered the ranking, measured in years.

Control1 Numeric. Control variable reflecting initial popularity at the time of Billboard entry.

Control2 Numeric. Another control for initial attention at Billboard entry.

Details

The data is used to illustrate the modeling of forgetting curves in collective attention.

Source

Derived from Billboard song rankings and streaming data as of 2016-10-29.

Examples

```
data("cross_section_data")
head(cross_section_data)
```

---

fit_all_models_log *Fit All Log-Transformed Decay Models: Biexponential, Log-Normal, Exponential*

Description

Fits three theoretical memory decay models to time-dependent attention or recall data:

Usage

```
fit_all_models_log(
  data,
  age_var = "age",
  observed_col = "CurrentPopularity",
  group_var = NULL,
  group_var2 = NULL,
  N_ref = NULL,
  weight_early_points = TRUE
)
```

Arguments

`data`	A data frame with time-series or cross-sectional data (e.g., memory or attention scores).
`age_var`	Name of the column representing age (e.g., time since event or figure).
`observed_col`	Name of the observed response variable to be modeled (e.g., "loess_correct").
`group_var`	Optional: first grouping variable (e.g., "same_country", "decay_bin").
`group_var2`	Optional: second grouping variable (e.g., attention level, quantiles).
`N_ref`	Optional: fixed value for parameter N in the biexponential model. If `NULL`, it will be estimated.
`weight_early_points`	Logical. If `TRUE`, assigns higher weight to early observations to prioritize early decay (default: `TRUE`).

Details

- **Biexponential Decay:**

$$S(t) = N \left[\exp(-(p+r)t) + \frac{r}{p+r-q} \left(\exp(-qt) - \exp(-(p+r)t) \right) \right]$$

- **Log-Normal Modulated Power Law:**

$$S(t) = \exp(b) \cdot t^{b_1} \cdot \exp\left(-b_2 (\log t)^2\right)$$

- **Exponential Decay:**

$$S(t) = c \cdot \exp(-qt)$$

All models are fitted in log-transformed form using nonlinear least squares (NLS). Grouped and stratified fitting is supported through one or two categorical variables.

Value

A named list with the following components:

biexponential A data frame of fitted values and parameters for the biexponential model.

lognormal Fitted results for the log-normal modulated power-law model.

exponential Fitted results for the exponential decay model.

model_comparison Model comparison table including AIC and BIC for each fitted model.

See Also

`fit_biexponential_model`, `fit_lognormal_log_model`, `fit_exponential_log_model`, `plot_all_models`

`fit_biexponential_model` *Fit biexponential model to memory data (grouped or not)*

Description

This function fits a biexponential forgetting curve to memory-related data, optionally by group and/or attention level. It allows the user to specify a reference value for N and tries a wide range of parameter initializations to improve model convergence.

Usage

```
fit_biexponential_model(
  data,
  age_var = "age",
  observed_col = "CurrentPopularity",
  group_var = NULL,
  group_var2 = NULL,
  N_ref = NULL,
  weight_early_points = FALSE
)
```

Arguments

`data`	A data frame containing memory data.
`age_var`	Name of the age column (default: "age").
`observed_col`	Name of the observed popularity/attention column (default: "CurrentPopularity").
`group_var`	(Optional) Name of a grouping column (e.g., same_country).
`group_var2`	(Optional) Name of attention-level group column (e.g., accumulated_advantage_level).
`N_ref`	Optional value for N to use in all starting points (default: NULL).
`weight_early_points`	Logical. Whether to give more weight to early time points (default: FALSE).

Details

Biexponential Decay Model:

The model estimated (on the log scale) is:

$$\log S(t) = \log\left[N\left(e^{-(p+r)t} + \frac{r}{p+r-q}\left(e^{-qt} - e^{-(p+r)t}\right)\right)\right]$$

where:

- $S(t)$: attention/popularity at time t
- N: initial popularity level
- p: communicative decay rate
- r: communicative-to-cultural transfer rate

- q: cultural decay rate

Why use the log-transformed model?

Working in the log scale provides several practical and statistical advantages:

- Stabilizes the variance across time, improving homoscedasticity.
- Enhances numerical stability during optimization.
- Prevents invalid predictions (e.g., negative popularity).
- Aligns with likelihood-based estimation for proportion/attention data.

Critical Time:

The *critical time* t_c is the point at which communicative and cultural memory contributions are equal:

$$t_c = \frac{1}{p + r - q} \log\left(\frac{(p + r)(p - q)}{rq}\right)$$

The function also estimates the standard error of t_c using the delta method, propagating uncertainty via the gradient and the parameter variance-covariance matrix.

Input Requirements:

- ‘age_var‘: numeric column indicating the age (e.g., time since event)
- ‘observed_col‘: numeric column with observed popularity or attention values
- ‘group_var‘ (optional): grouping variable for stratified fitting
- ‘group_var2‘ (optional): secondary grouping variable (e.g., attention level)
- ‘accumulated_attention‘ (optional): used to estimate critical time error

Value

A list with:

fitted A data frame with fitted values and predicted popularity.

params A data frame of estimated parameters: N, p, r, q, AIC, BIC, R^2, t_c, and its standard error.

See Also

`plot_fitted_decay`, `fit_all_models_log`

`fit_exponential_log_model`

Fit Exponential Decay Model (log-transformed)

Description

Fits a log-transformed exponential decay model to collective memory or attention data. The model assumes attention decays continuously over time without a peak.

Usage

```
fit_exponential_log_model(
  data,
  age_var = "age",
  observed_col = "CurrentPopularity",
  group_var = NULL,
  group_var2 = NULL,
  weight_early_points = FALSE
)
```

Arguments

`data`	A data frame with attention or recall values.
`age_var`	Name of the column representing age (e.g., time since birth or publication).
`observed_col`	Name of the observed response variable (e.g., "loess_correct").
`group_var`	Optional: primary grouping column (e.g., "same_country").
`group_var2`	Optional: secondary grouping column (e.g., "attention_level").
`weight_early_points`	Logical. Currently ignored in this model. Included for API consistency.

Value

A list containing:

fitted A data frame including predicted values.

params Parameter estimates c and q, along with AIC, BIC, and pseudo-R^2.

Exponential Decay Model

The model is expressed in log-transformed form as:

$$\log S(t) = \log c - q \cdot t$$

which is equivalent to:

$$S(t) = c \cdot e^{-qt}$$

where:

- $S(t)$ is the observed popularity or recall at time t,
- c is the initial attention (scaling factor),
- q is the exponential decay rate.

Fitting in log-space has several advantages:

- It linearizes the decay process for stable nonlinear least squares (NLS) estimation.
- Reduces the influence of high-value outliers.
- Emphasizes early decay, typical in forgetting processes.
- Improves variance stability in heteroscedastic data.

This model assumes **monotonic decay**, and may not be appropriate if the data exhibit a peak or burst pattern followed by decline.

See Also

`fit_biexponential_model`, `fit_lognormal_log_model`, `fit_all_models_log`

`fit_lognormal_log_model` *Fit Log-Normal Modulated Power-Law to Memory Data (Grouped or Not)*

Description

Fits a log-normal-inspired forgetting curve to memory or attention data, optionally stratified by grouping variables. The model is estimated using a quadratic regression on the log-log scale.

Usage

```
fit_lognormal_log_model(
  data,
  age_var = "age",
  observed_col = "CurrentPopularity",
  group_var = NULL,
  group_var2 = NULL,
  weight_early_points = FALSE
)
```

Arguments

`data`	A data frame containing memory or attention data.
`age_var`	Name of the age column (e.g., time since event or publication).
`observed_col`	Name of the observed attention/popularity column.
`group_var`	Optional: primary grouping column (e.g., "same_country").
`group_var2`	Optional: secondary grouping column (e.g., "attention_level").
`weight_early_points`	Logical. Whether to give more weight to early time points (default: FALSE).

Value

A list containing:

fitted A data frame with predicted values by age and group.

params Estimated parameters b, b_1, b_2, and model metrics (AIC, BIC, R^2).

Model Specification

The model is estimated as:

$$\log S(t) = b + b_1 \log(t) - b_2(\log(t))^2$$

which is equivalent to:

$$S(t) = \exp(b) \cdot t^{b_1} \cdot \exp\left(-b_2(\log t)^2\right)$$

where:

- $S(t)$ is the observed popularity or recall at time t,

- b, b_1, b_2 are parameters controlling early growth and long-tail decay.

This model captures:

- Initial attention growth via the power-law term t^{b_1},
- Long-tail memory decay via the log-normal term $\exp(-b_2(\log t)^2)$.

Why log-log transformation?

- Improves numerical stability during model fitting.
- Enables interpretable parameters in logarithmic space.
- Captures the heavy-tailed nature of attention decay.
- Linearizes a nonlinear decay relationship for fitting.

See Also

`fit_biexponential_model`, `fit_exponential_log_model`, `fit_all_models_log`

`merge_bins_with_original` *Merge Decay Bins with Original Time-Series Data*

Description

This function combines the original wide-format time-series matrix with its corresponding decay bin matrix (e.g., from `assign_decay_bins`), converting both into long format and joining them for downstream modeling or visualization.

Usage

```
merge_bins_with_original(wide_data, bins_df, replace_na_with_zero = TRUE)
```

Arguments

`wide_data`	A data frame in wide format, with a `time` column and one column per entity (e.g., paper or product).
`bins_df`	A data frame of decay bins with the same number of rows as `wide_data` and one column per entity (excluding time).
`replace_na_with_zero`	Logical. If `TRUE` (default), missing values in the attention matrix are replaced with 0.

Details

Each entry in the resulting dataset represents the attention (e.g., citations, views, mentions) received by an item at a specific time, along with its corresponding decay bin. This is especially useful for stratified modeling of attention decay across different levels of cumulative popularity.

Value

A long-format data frame with the following columns:

time Time point (e.g., year or semester).

doi Unique identifier for each entity (e.g., paper ID).

value Observed attention value at that time (e.g., number of citations).

decay_bin Decay bin assigned to that time/entity pair (may be NA for zero attention).

Treatment of Zero Attention

As emphasized in Candia et al. (2019, *Nature Human Behaviour*), attention values of zero are not missing data — they indicate genuine lack of attention or forgetting. Therefore, by default, this function:

- Converts all NA values in the attention matrix to 0.
- Ensures that zeroes contribute to aggregated averages (e.g., $\Delta c(t)$), making them reflective of the true collective state.

This behavior can be disabled by setting `replace_na_with_zero = FALSE`.

In this function, **NA values are replaced by 0** to represent *genuine lack of attention* (e.g., zero citations in a given time window). This ensures that zeros are **included in averages and decay modeling**, as described in Candia et al. (2019). Including zeros is important for measuring the true dynamics of collective forgetting.

See Also

`assign_decay_bins`, `process_time_series_bins`, `reshape_citation_timeseries`

Examples

```
# See full pipeline: reshape_citation_timeseries() → compute_cumulative_matrix() → assign_decay_bins()
# Then use merge_bins_with_original() to finalize the long-format dataset.
```

---

normalize_string *Normalize a String for Consistent Text Matching*

Description

This utility function standardizes character strings by applying the following transformations:

- Converts all characters to lowercase.
- Removes accents and diacritics.
- Trims leading and trailing whitespace.
- Removes all non-alphanumeric characters (except underscores).

Usage

```
normalize_string(x)
```

Arguments

x	A character vector to normalize.

Details

It is useful for text preprocessing tasks such as:

- Comparing entity names or labels across datasets,
- Deduplicating responses in surveys,
- Cleaning category labels before grouping or joining.

Value

A character vector of normalized strings.

Examples

```
normalize_string(c(" García ", "garcia", "GARCÍA!", "Garcia_1"))
# Returns: "garcia", "garcia", "garcia", "garcia_1"
```

---

plot_all_models *Plot All Fitted Forgetting Curves (Biexponential, Exponential, Log-normal)*

Description

This function overlays the fitted forgetting curves from three competing models:

- **Biexponential**: captures short- and long-term decay via two interacting memory systems.
- **Exponential**: simple continuous decay process.
- **Log-normal modulated power-law**: captures early attention rise followed by long-tail decay.

Usage

```
plot_all_models(
  model_outputs,
  age_var = "age",
  observed_col = "CurrentPopularity",
  group_var = NULL,
  group_var2 = NULL,
  log_y = TRUE,
  log_x = FALSE
)
```

Arguments

model_outputs	A list returned by fit_all_models_log, containing fitted values for each model and a merged data frame in $fitted.
age_var	Name of the column indicating age or time since the event (default: "age").
observed_col	Name of the observed attention or recall variable (default: "CurrentPopularity").
group_var	Optional. Primary grouping variable for line color and point shape (e.g., "same_country", "decay_bin").
group_var2	Optional. Secondary grouping variable used for faceting (e.g., attention level or demographic stratum).
log_y	Logical. If TRUE, apply log10 transformation to the Y axis. Default is TRUE.
log_x	Logical. If TRUE, apply log10 transformation to the X axis. Default is FALSE.

Details

It compares each model's predictions against observed attention or recall values. The function supports grouping (color/style by category) and faceting for stratified comparison across populations or experimental conditions.

Value

A ggplot2 object with layered curves and observed values, styled and faceted according to grouping variables.

See Also

fit_biexponential_model, fit_lognormal_log_model, fit_exponential_log_model, compare_model_fits

---

plot_fitted_decay *Plot Fitted Memory Decay Curves Against Observed Data*

Description

Visualizes the fitted forgetting curve against observed attention or popularity values. This function supports comparisons across groups and stratification by attention levels.

Usage

```
plot_fitted_decay(
  model_output,
  observed_col = "CurrentPopularity",
  fitted_col = "fitted_correct",
  age_var = "age",
  group_var = NULL,
  group_var2 = NULL,
  log_y = TRUE,
  log_x = FALSE
)
```

Arguments

model_output	Output from a model fitting function (e.g., fit_biexponential_model).
observed_col	Name of the column containing observed values (default: "CurrentPopularity").
fitted_col	Name of the column containing model predictions (default: "fitted_correct").
age_var	Name of the age/time column (default: "age").
group_var	Optional column for group-level coloring.
group_var2	Optional column for faceting (e.g., stratification by attention).
log_y	Logical. Use log10 scale on the y-axis? (default: TRUE).
log_x	Logical. Use log10 scale on the x-axis? (default: FALSE).

Value

A ggplot object showing observed vs fitted decay curves.

Required Columns in the Fitted Data

- age_var: Numeric variable indicating time since the event or subject (e.g., years).
- observed_col: Observed attention/popularity (e.g., survey recall, citations).
- fitted_col: Predicted values from a decay model.

Optional Columns

- group_var: Used to color lines/points by groups (e.g., same vs. different country).
- group_var2: Used to facet the plot by stratification variable (e.g., attention level).

Supports log-log visualization to capture long-tail decay or exponential behavior.

See Also

plot_all_models, fit_all_models_log, plot_fitted_decay_for_publication

---

plot_fitted_decay_for_publication
Plot Fitted Decay Curve for Publication

Description

Creates a high-quality plot comparing observed and fitted memory decay curves for inclusion in publications. The plot highlights temporal decay patterns in attention or popularity and supports grouping and faceting by categorical variables.

Usage

```
plot_fitted_decay_for_publication(
  model_output,
  observed_col = "CurrentPopularity",
  fitted_col = "fitted_correct",
  age_var = "age",
  group_var = NULL,
  group_var2 = NULL,
  export_path = NULL,
  x_breaks = NULL,
  x_limits = NULL,
  y_breaks = NULL,
  y_limits = NULL,
  log_y = FALSE,
  log_x = FALSE
)
```

Arguments

model_output	A list returned by a model fitting function (e.g., fit_biexponential_model), containing a data frame named fitted with predicted values.
observed_col	Name of the column containing observed attention/popularity values (default: "CurrentPopularity").
fitted_col	Name of the column containing fitted model predictions (default: "fitted_correct").
age_var	Name of the column representing age or time since event (default: "age").
group_var	Optional. A grouping variable to color lines and points (e.g., "same_country").
group_var2	Optional. A second grouping variable for faceting (e.g., attention level or quantile).
export_path	Optional file path (without extension). If provided, saves both PDF and SVG outputs.
x_breaks	Optional numeric vector of X-axis breaks.
x_limits	Optional numeric vector of length 2 to set X-axis limits.
y_breaks	Optional numeric vector of Y-axis breaks.
y_limits	Optional numeric vector of length 2 to set Y-axis limits.
log_y	Logical. If TRUE, uses log10 scale on Y-axis. Default is FALSE.
log_x	Logical. If TRUE, uses log10 scale on X-axis. Default is FALSE.

Details

The function supports flexible axis scaling (log or linear), custom axis breaks and limits, and exports to both PDF and SVG formats if desired.

Value

A ggplot2 object suitable for academic publications.

Requirements

This function requires the following packages:

- ggplot2
- ggtext
- scales

See Also

fit_all_models_log, plot_all_models, plot_raw_memory_decay

---

plot_raw_memory_decay *Plot Raw and Aggregated Memory Decay Data*

Description

Creates a flexible and publication-ready visualization of memory or attention decay over time. It overlays raw observations (e.g., from survey or time-series data) with smoothed or aggregated trends (e.g., LOESS, GAM, or grouped means).

Usage

```
plot_raw_memory_decay(
  raw_df = NULL,
  aggregated_df = NULL,
  age_var,
  response_var_raw,
  response_var_agg = NULL,
  group_var = NULL,
  group_var2 = NULL,
  log_y = TRUE,
  log_x = FALSE,
  xlim_vals = NULL,
  ylim_vals = NULL
)
```

Arguments

raw_df	(Optional) Raw individual-level dataset. Each row should represent a single observation (e.g., a survey recall response). Requires columns age_var and response_var_raw.
aggregated_df	(Optional) Output from smooth_survey_decay or manual aggregation. Should include age_var and response_var_agg.
age_var	String. Name of the column representing historical age or time since event (must be numeric).
response_var_raw	String. Name of the response variable in raw_df (e.g., "correct", "recall").
response_var_agg	String. Name of the aggregated response variable in aggregated_df (e.g., "mean_response", "loess_correct").

group_var	String (optional). First grouping variable (e.g., "same_country", "decay_bin"). Used for coloring lines and points.
group_var2	String (optional). Second grouping variable for faceting (e.g., attention level, stratification).
log_y	Logical. Whether to apply log10 transformation to the Y axis (default: TRUE).
log_x	Logical. Whether to apply log10 transformation to the X axis (default: FALSE).
xlim_vals	Optional numeric vector of length 2. Manual X-axis limits.
ylim_vals	Optional numeric vector of length 2. Manual Y-axis limits. If using log10 and no valid values are detected, the function will request this explicitly.

Details

The function is designed to support grouped comparisons and faceting by secondary grouping variables (e.g., attention level or demographic strata). Both linear and log-log scales are supported.

Value

A ggplot2 object showing raw data (if provided), aggregated trends, and stratified group patterns (if applicable).

Input Requirements

You must provide at least one of the following:

- raw_df: a data frame of raw individual-level responses.
- aggregated_df: a data frame with aggregated responses over age.

See Also

smooth_survey_decay, aggregate_mean_response, plot_fitted_decay

---

process_data *Process and Clean Survey-Based Memory Data for Decay Curve Modeling*

Description

This function prepares noisy memory or recall survey data for forgetting curve analysis. It includes filtering, outlier removal, and optional grouping or stratification steps. It is tailored for use in cross-sectional or recall-based studies (e.g., cultural memory, icon recall).

Usage

```
process_data(
  data,
  id_col = "slug",
  age_var = "age_metric",
  group_var = NULL,
  replies_col = NULL,
  group_var2 = NULL,
```

```
  quantile = 1,
  percentile = 0.99,
  filter_n = 3
)
```

Arguments

data	A data frame with raw survey-based memory or attention responses.
id_col	Name of the unique identifier column (e.g., "slug", "person_id"). Default: "slug".
age_var	Name of the age or time-since-event variable. Default: "age_metric".
group_var	Optional. First grouping variable for stratified modeling (e.g., "same_country").
replies_col	Optional. Column with number of responses per item (e.g., "n_replies") used for filtering.
group_var2	Optional. Column representing accumulated attention (e.g., "global") used for outlier removal and quantile binning.
quantile	Integer. Number of quantile bins to create from group_var2 (default: 1 = no bins).
percentile	Numeric [0, 1]. Threshold to remove top outliers in group_var2 (default: 0.99).
filter_n	Minimum number of replies required to keep an item (default: 3).

Details

Specifically, the function:

- Filters out IDs with too few responses (e.g., less than 3).
- Removes outliers in accumulated attention using a top percentile threshold.
- Optionally classifies items into quantiles based on attention/popularity (e.g., deciles).
- Keeps grouping variables for later stratified model fitting.

This is often a first preprocessing step before applying forgetting models such as fit_biexponential_model, fit_lognormal_log_model, or fit_exponential_log_model.

Value

A cleaned and annotated data frame ready for forgetting curve modeling. Columns may include:

- Filtered items with sufficient responses.
- Age variable standardized to age_metric.
- Optional columns for group, attention level, and outlier-stripped popularity.

Examples

```
# Load and preprocess a dataset
data(survey_data)
cleaned <- process_data(
  survey_data,
  id_col = "entity_id",
  age_var = "age_metric",
  replies_col = "reply_count",
  group_var = "location_flag",
  group_var2 = "performance_score",
```

```
  quantile = 3
)
```

`process_time_series_bins` *Process Time-Series Citation Data and Assign Decay Bins*

Description

This function provides a full pipeline to prepare time-series attention or citation data for decay analysis. It reshapes the data from long to wide format, computes cumulative attention over time, assigns each item to a decay bin using log-scaled thresholds, and merges the result back to a long-format data frame for modeling or visualization.

Usage

```
process_time_series_bins(data)
```

Arguments

`data`	A long-format citation dataset with at least two columns: `time` (e.g., semester, year) and `doi` (unique item ID).

Details

The typical use case involves datasets where each row represents a citation event, with columns such as ‘time‘ and ‘doi‘, and the goal is to understand forgetting or attention decay dynamics across items of different cumulative popularity.

Steps Performed:

1. Reshape long-format citation data into wide format using `reshape_citation_timeseries`.
2. Compute cumulative citation counts over time using `compute_cumulative_matrix`.
3. Assign each cumulative value to a decay bin using `assign_decay_bins`, based on log-scaled binning.
4. Merge decay bin labels back to the original data using `merge_bins_with_original`.

Treatment of Zero Values:

- When computing decay bins, items with **zero cumulative attention** are excluded from bin computation because log-scaling is undefined at zero. These entries are assigned `NA` in the decay_bin column.

- However, when merging bins back to the time-series matrix, `merge_bins_with_original` replaces missing values (NAs) in the attention matrix with `0`, treating them as informative "no-attention" events. This is consistent with the logic in Candia et al. (2019, *Nature Human Behaviour*).

Value

A long-format data frame with the following columns:

time Time point of the observation (e.g., year or semester).

doi Unique identifier for each paper or entity.

value Number of citations or attention units at that time.

decay_bin Decay bin assigned based on cumulative attention (can be NA for uncited items).

See Also

assign_decay_bins, merge_bins_with_original, plot_raw_memory_decay

---

reshape_citation_timeseries

Reshape Citation Time Series from Long to Wide Format

Description

This function transforms a long-format citation dataset into wide format, where each row represents a time point (e.g., year or semester) and each column corresponds to a unique item identifier (typically a paper DOI).

Usage

```
reshape_citation_timeseries(data)
```

Arguments

data	A data frame in long format. Must include: time Numeric variable indicating the time point (e.g., year or semester). doi Unique identifier for each paper or entity (e.g., "10.1103/PhysRevLett.1.1"). value Citation count or attention metric at each time point.

Details

The resulting matrix is suitable for computing cumulative attention, decay curves, and memory stratification. Each cell in the matrix represents the number of citations (or another attention metric) that a specific item received at a given time point. If a citation is missing for a given time-item pair, the corresponding cell will contain NA.

This reshaping is typically used as the first step in the 'MemoryDecay' pipeline for time-series data, prior to applying compute_cumulative_matrix and assign_decay_bins.

Value

A wide-format data frame where:

- Each row is a unique time point.
- Each column is a paper/item ID prefixed with value., containing the attention received at that time.
- The first column is time, ordered in ascending order.

See Also

compute_cumulative_matrix, assign_decay_bins, merge_bins_with_original

Examples

```
# Sample long-format data
citation_data <- data.frame(
  time = rep(1980:1982, each = 2),
  doi = rep(c("A", "B"), 3),
  value = c(3, 5, 4, 6, 2, 1)
)

wide_df <- reshape_citation_timeseries(citation_data)
head(wide_df)
```

---

smooth_survey_decay *Smooth and Aggregate Noisy Survey-Based Memory Decay Data*

Description

This function is designed to smooth and aggregate memory or attention decay data collected from cross-sectional **survey instruments**. It is particularly suited for analyzing recall accuracy or attention scores over time (e.g., the age of historical or cultural figures).

Usage

```
smooth_survey_decay(
  data,
  age_var,
  response_var,
  group_var = NULL,
  group_var2 = NULL,
  filter_age = Inf
)
```

Arguments

data	A data frame with survey-based memory or attention data.
age_var	String. Name of the age variable (e.g., "age_metric"). Must be numeric.
response_var	String. Name of the response variable (e.g., "performance_score").
group_var	(Optional) First grouping variable (default: NULL).
group_var2	(Optional) Second grouping variable for faceting or stratification (default: NULL).
filter_age	Numeric. Maximum age to retain (default: Inf = include all).

Details

It computes:

- **Mean response** per age and group
- **LOESS-smoothed** trend using local polynomial regression
- **GAM-smoothed** trend using penalized cubic splines

The function supports one or two grouping variables:

- `group_var`: Primary grouping (e.g., same vs. different country)
- `group_var2`: Secondary grouping (e.g., attention decile, demographic strata)

These allow stratified smoothing and facilitate visual comparisons across subpopulations.

Value

A data frame with the following columns:

`age_metric` Age variable in numeric form.

`response` Raw response values (copied from `response_var`).

`mean_response` Mean response for each age/group combination.

`loess_correct` LOESS-smoothed response over age.

`gam_correct` GAM-smoothed response over age.

`group_var, group_var2` Preserved grouping columns (if provided).

See Also

`plot_raw_memory_decay`, `fit_biexponential_model`

Examples

```
data(survey_data)
smooth_df <- smooth_survey_decay(
  data = survey_data,
  age_var = "age_metric",
  response_var = "performance_score",
  group_var = "location_flag"
)
```

---

survey_data *Anonymized Cross-Cultural Icon Recall Survey*

Description

`survey_data` is an anonymized dataset from a cross-cultural survey investigating the accuracy and depth of respondents' recall of prominent historical and cultural figures. All sensitive details have been masked, ensuring participant and icon anonymity.

Usage

```
data(survey_data)
```

Format

A data frame with the following columns:

`entity_id` An anonymous identifier for each cultural icon (e.g., "ID1").

`age_metric` A numeric measure approximating the historical or temporal context of the icon.

`location_flag` A binary indicator (0/1) indicating whether the respondent resides in the same region as the icon.

`performance_score` A numeric score reflecting how accurately participants recalled details about the icon.

`reply_count` The number of relevant responses or mentions in the survey.

Details

The survey was administered online across multiple world regions, featuring participants who answered questions on various iconic historical or cultural figures. The data has been approved for release under institutional ethics (IRB) protocols to protect personally identifiable information (PII). Only non-sensitive attributes are included in this dataset.

The primary goals of the survey were to:

1. Measure the extent of familiarity with globally recognized icons,
2. Examine potential geographic or demographic biases in icon recall, and
3. Assess how accurately participants recalled key details about these icons.

This dataset contains **5 columns** and *n* observations (rows). The variables included are described below.

Examples

```
# Load the dataset
data(survey_data)

# Preview the first few rows
head(survey_data)
```

---

theme_scl *Custom ggplot2 theme for decay plots*

Description

Custom ggplot2 theme for decay plots

Usage

```
theme_scl()
```

`time_series_data` *Anonymized Citation Time Series from a Scientific Journal (1980–2003)*

Description

`time_series_data` is an anonymized time series dataset containing citation activity for scientific papers published in a major (anonymized) journal, observed at a semiannual frequency between 1980 and 2003. It captures how many citations each paper received over time, with inflation adjustments that correct for changing publication volume.

Usage

```
data(time_series_data)
```

Format

A data frame with the following columns:

`doi` An anonymized identifier for the cited paper.

`N_cit_Infla` Inflation-adjusted number of citations received during the semester.

`time` Publication time window, expressed in semiannual steps (e.g., 1984.0, 1984.5).

Details

Each row represents the number of citations received by a single paper in a specific semester. The dataset allows tracking of longitudinal citation dynamics at the individual paper level, and is designed for modeling memory decay, preferential attachment, or inflation-normalized impact.

The variable `N_cit_Infla` refers to the number of citations received by the paper in that time window, adjusted by an inflation factor derived from the total number of publications in the journal. The time variable is numeric and increases in 0.5-year increments (i.e., Jan–Jun = ".0", Jul–Dec = ".5").

All identifiers (e.g., DOIs, journal IDs) have been anonymized. The dataset was constructed using SQL queries over a MonetDB instance connected to a large-scale citation graph from the APS corpus.

This dataset contains **3 columns** and *n* rows (observations by paper-semester).

Examples

```
# Load the dataset
data(time_series_data)

# Plot citation trajectory of a single paper
library(ggplot2)
ggplot(subset(time_series_data, doi == 643244), aes(x = time, y = N_cit_Infla)) +
  geom_line() +
  labs(title = "Citation Trajectory (Inflation-Adjusted)",
       x = "Time (Year)", y = "Citations (adjusted)") +
  theme_minimal()
```

Index