

Package ‘jaciii2025’

January 23, 2025

Title Viewing Intention Analysis using VAS and FCM
Version 0.1.0
Description This package provides functions for analyzing Japanese audience data, including clustering, trend analysis, and viewing intention modeling.
License MIT
Encoding UTF-8
Roxygen list(markdown = TRUE)
RoxygenNote 7.3.2
Imports tidyverse,
ggbeeswarm,
cluster,
ggplot2,
dplyr,
purrr,
tibble

Contents

analyze_viewing_intention	2
analyze_viewing_intention	3
apply_common_theme	4
calculate_basic_stats	4
create_cluster_vas_plots	5
create_coefficient_plot	6
create_correlation_plot	7
create_sample_data	8
create_vas_plot	9
create_vas_plots_comparison	10
get_error_message	10
get_translation	11
initialize_environment	12
load_and_preprocess_data	14
main_basic_trend	15

main_fuzzy_cmeans	16
main_viewing_intention	18
perform_clustering	19
perform_country_ttest	20
save_results	22
set_language	23
Index	25

analyze_viewing_intention
<i>Statistical Analysis of Viewing Intention Factors</i>

Description

Analyzes the influence of evaluation items (Q1-Q4) on viewing intention (Q5) using movie impression evaluation data.

Usage

```
analyze_viewing_intention(data, subset_country = NULL, lang = "en")
```

Arguments

data	Analysis data frame containing columns Q1-Q5
subset_country	Target country name for country-specific analysis
lang	Output language setting ("ja": Japanese, "en": English)

Details

- Analysis steps:
1. Data validation
 2. Preprocessing (missing value removal, scaling)
 3. Multiple regression analysis execution
 4. Standardized coefficient calculation
 5. Correlation analysis
 6. VIF (multicollinearity) check

Value

- list A list containing analysis results:
- influence_summary** Summary of factor influences
 - model_fit** Model fit indicators
 - model** Regression analysis model
 - correlation_matrix** Correlation matrix

analyze_viewing_intention
<i>Statistical Analysis of Movie Viewing Intention Factors</i>

Description

Analyzes the influence of evaluation items (Q1-Q4) on viewing intention (Q5) using movie impression evaluation data.

Usage

```
analyze_viewing_intention(data, subset_country = NULL, lang = "en")
```

Arguments

data	Analysis data frame (containing columns Q1-Q5)
subset_country	Target country name for country-specific analysis
lang	Output language setting ("ja": Japanese, "en": English)

Details

Analysis steps:

- 1. Data validation
- 2. Preprocessing (removing missing values, scaling)
- 3. Executing multiple regression analysis
- 4. Calculating standardized coefficients
- 5. Correlation analysis
- 6. VIF (multicollinearity) check

Value

list A list containing the following analysis results:

- influence_summary** Summary of factor influences
- model_fit** Model fit indicators
- model** Regression analysis model
- correlation_matrix** Correlation matrix

`apply_common_theme` *Apply Common Theme to Plots*

Description

Applies unified theme settings to a ggplot object to maintain consistent visual appearance across all plots in the analysis.

Usage

```
apply_common_theme(base_size = 14, rotate_x_labels = FALSE)
```

Arguments

<code>base_size</code>	Base font size (default: 14)
<code>rotate_x_labels</code>	Whether to rotate X-axis labels by 45 degrees (default: FALSE)
<code>p</code>	ggplot object

Details

Theme settings include:

- Standardized font sizes
- Consistent axis label formatting
- Uniform grid line settings
- Optional X-axis label rotation

Value

ggplot object with applied theme

`calculate_basic_stats`
Optimized Basic Statistics Calculation

Description

Calculates basic statistical measures for the specified country's data

Usage

```
calculate_basic_stats(data, lang = "ja", country_filter = NULL)
```

Arguments

data	Long-format data
lang	Language code
country_filter	Country name for filtering specific country data (optional)

Details

Calculated statistics include:

- Mean
- Standard deviation
- Median
- Quartiles
- Sample size

The function performs optimized calculations using vectorized operations for improved performance with large datasets.

Value

Data frame containing basic statistics

```
create_cluster_vas_plots
```

Generate VAS Plots for Each Cluster

Description

Visualizes Visual Analog Scale (VAS) evaluations based on clustering results

Usage

```
create_cluster_vas_plots(data, cluster_info, plots_dir, lang = "ja", k = 4)
```

Arguments

data	Long-format data containing question and value columns for VAS evaluations
cluster_info	Data frame with cluster information including ID, cluster, and membership columns
plots_dir	Directory for saving plots (both PDF and SVG formats)
lang	Language code for output messages and labels (e.g., 'ja' for Japanese)
k	Number of clusters (default: 4) specifying the data partition count

Details

The function creates comprehensive visualizations that combine multiple plot types:

- Violin plots show the probability density of the data
- Box plots display quartiles and outliers
- Jitter plots reveal the distribution of individual responses
- Color coding distinguishes between clusters
- Automated plot saving in both PDF and SVG formats

Value

List of ggplot objects containing:

- Violin plots for each cluster
- Box plots showing distribution
- Jitter plots displaying individual data points

```
create_coefficient_plot
```

Generate Coefficient Plot

Description

Generates a visualization plot of standardized coefficients from the regression analysis results. The plot displays coefficient magnitudes and confidence intervals to facilitate comparison of factor influences.

Usage

```
create_coefficient_plot(analysis_results, lang = "ja")
```

Arguments

<code>analysis_results</code>	Result list from <code>analyze_viewing_intention</code>
<code>lang</code>	Language setting ("ja": Japanese, "en": English)

Details

Plot features:

- Point plot of standardized coefficients
- Display of 95% confidence intervals
- Translation-enabled factor names

Value

ggplot object (coefficient plot)

```
create_correlation_plot
```

Generate Correlation Plot

Description

Generates a heatmap visualization of variable correlations. The plot represents correlation coefficients through color intensity, providing an intuitive understanding of positive and negative relationships between variables.

Usage

```
create_correlation_plot(analysis_results, lang = "ja")
```

Arguments

<code>analysis_results</code>	Analysis results list returned from <code>analyze_viewing_intention()</code>
<code>lang</code>	Language setting ("ja": Japanese, "en": English)

Details

Plot specifications:

- Correlation coefficient range: -1 to 1
- Color scheme:
 - Positive correlation: Red (darker for stronger)
 - Negative correlation: Blue (darker for stronger)
 - No correlation: White
- Variable names use translations defined in `translation_config.R`

Value

ggplot object with:

- Correlation coefficients displayed as a heatmap
- Numeric values shown in each cell
- Axis labels in the specified language

See Also

- [analyze_viewing_intention](#) - Generate analysis results
- [get_translation](#) - Get translated strings

Examples

```
## Not run:  
# Generate correlation plot in Japanese  
plot <- create_correlation_plot(analysis_results, lang = "ja")  
print(plot)  
  
# Generate correlation plot in English  
plot <- create_correlation_plot(analysis_results, lang = "en")  
print(plot)  
  
## End(Not run)
```

create_sample_data *Generate Sample Data*

Description

Generates sample data that reflects clustering patterns and realistic response distributions based on empirical observations.

Usage

```
create_sample_data(n_samples_per_country = 30)
```

Arguments

n_samples_per_country
Number of samples per country (default: 30)

Details

Data generation process:

1. Creates balanced samples for each country
2. Generates correlated evaluation scores
3. Adds realistic noise to simulate actual response patterns
4. Includes temporal variation in responses

Data characteristics:

- VAS scores follow expected correlation patterns
- Country-specific response tendencies are simulated
- Temporal patterns reflect typical data collection periods

Value

Data frame containing generated sample data with:

- ID: Unique identifier for each response
- Country: Country indicator (e.g., "ja", "sg")
- Q1-Q5: Evaluation scores on Visual Analog Scale (0.00-1.00)
- Timestamp: Data collection timestamp

create_vas_plot	Create Optimized VAS Plot
-----------------	---------------------------

Description

Creates an optimized visualization of Visual Analog Scale (VAS) data with enhanced readability and statistical information.

Usage

```
create_vas_plot(data, lang = "ja", country_filter = NULL)
```

Arguments

data	Long-format data
lang	Language code
country_filter	Country name for filtering specific country data

Details

Plot features:

- Combined violin and box plots for distribution visualization
- Jittered individual data points for transparency
- Consistent color scheme for easy interpretation
- Optimized label placement and spacing
- Statistical annotations (mean, median, quartiles)
- Language-specific labels based on configuration

Performance optimizations:

- Efficient data processing for large datasets
- Optimized memory usage for plot rendering
- Cached computations for repeated elements

Value

ggplot object with optimized Visual Analog Scale visualization

```
create_vas_plots_comparison
```

Create Country-wise VAS Plots

Description

Creates Visual Analogue Scale (VAS) plots comparing data between countries

Usage

```
create_vas_plots_comparison(data, lang = "ja")
```

Arguments

data	Long-format data
lang	Language code

Value

ggplot object

```
get_error_message
```

Retrieve Error Messages

Description

Retrieves localized error messages based on the specified language setting

Usage

```
get_error_message(error_key, lang = getOption("movie_analysis.lang", "ja"))
```

Arguments

error_key	Error key identifier
lang	Language code for message localization

Details

This function provides internationalization support for error messages:

- Uses a centralized message configuration system
- Supports multiple languages through translation mappings
- Falls back to English if translation is not available
- Allows runtime language switching

Configuration:

- Messages are defined in translation_config.R
- Default language can be set via options()
- Supports dynamic message formatting

Example error keys:

- "file_not_found"
- "invalid_data_format"
- "missing_required_column"
- "invalid_parameter"

Value

character Localized error message

get_translation	<i>Retrieve Translated Text</i>
-----------------	---------------------------------

Description

Retrieves translated text corresponding to the specified language, category, and key.

Usage

```
get_translation(key, category, lang = getOption("movie_analysis.lang", "ja"))
```

Arguments

key	Translation key (e.g., "Q1", "main_title", etc.)
category	Category identifier (e.g., "questions", "plot_labels", etc.)
lang	Language code ("ja" for Japanese or "en" for English)

Details

Translation system features:

- Hierarchical organization by categories
- Fallback mechanism for missing translations
- Runtime language switching
- Support for dynamic text formatting

Available categories:

- questions: Survey question texts
- plot_labels: Visualization labels
- messages: System messages
- descriptions: Analysis descriptions

Value

character Translated text string

See Also

- [get_error_message](#) - For error message translations

Examples

```
# Get question 1 in Japanese
text_ja <- get_translation("Q1", "questions", "ja")

# Get main title in English
text_en <- get_translation("main_title", "plot_labels", "en")
```

initialize_environment

Initialize Environment Settings

Description

Initializes system environment settings, locale configuration, and graphics device setup

Usage

```
initialize_environment(  
  verbose = FALSE,  
  install_packages = TRUE,  
  force_cairo = FALSE  
)
```

Arguments

verbose	Flag for displaying debug information
install_packages	Flag for automatic installation of missing packages
force_cairo	Flag to force usage of Cairo graphics device

Details

Initialization process includes:

1. System locale configuration
 - Sets appropriate character encoding
 - Configures language settings
 - Handles regional formatting
2. Package dependency management
 - Checks required packages
 - Installs missing dependencies
 - Verifies package versions
3. Graphics device setup
 - Configures appropriate device for platform
 - Handles font settings
 - Optimizes for high-resolution output
4. Error handling
 - Provides detailed error messages
 - Implements fallback options
 - Logs configuration issues

The function ensures consistent behavior across different platforms and provides detailed feedback for troubleshooting configuration issues.

Value

list A list containing configuration results:

success	bool Overall success status
locale	character Configured locale settings
graphics_device	character Configured graphics device
messages	character vector Processing messages

`load_and_preprocess_data`*Enhanced Data Loading and Preprocessing*

Description

Enhanced version of data loading and preprocessing pipeline with improved error handling and data validation.

Usage

```
load_and_preprocess_data(file_path, lang = "ja")
```

Arguments

<code>file_path</code>	Path to the data file
<code>lang</code>	Language code for messages and labels

Details

Preprocessing steps:

1. Data Loading
 - File existence check
 - Encoding detection
 - CSV format validation
2. Data Cleaning
 - Missing value handling
 - Data type conversion
 - Column name standardization
3. Data Validation
 - Range checks for VAS scores (0-100)
 - Country code validation
 - Timestamp format verification
4. Data Transformation
 - Wide to long format conversion
 - Column ordering
 - Factor level setting

Error Handling:

- Detailed error messages for each processing step
- Graceful handling of missing or invalid data
- Logging of preprocessing operations
- Warning messages for potential data issues

Value

- List of data frames containing:
- raw: Original data in wide format
 - long: Transformed data in long format for analysis

main_basic_trend	<i>Enhanced Main Execution Function for Basic Trend Analysis</i>
------------------	--

Description

Enhanced version of the main execution function for basic trend analysis with improved error handling and performance monitoring.

Usage

```
main_basic_trend(  
  input_file,  
  output_dir,  
  lang = "ja",  
  verbose = FALSE,  
  benchmark = FALSE  
)
```

Arguments

input_file	Input file path
output_dir	Output directory for results
lang	Language code for output localization
verbose	Flag for displaying debug information
benchmark	Flag for performance benchmarking

Details

- Analysis workflow:
1. Data Processing
 - Data loading and validation
 - Missing value handling
 - Data transformation
 2. Statistical Analysis
 - Basic statistics calculation
 - Distribution analysis
 - Trend identification

- 3. Visualization Generation
 - VAS plots
 - Distribution comparisons
 - Summary visualizations
- 4. Results Output
 - CSV data files
 - PDF/SVG plots
 - Summary reports

Performance Features:

- Optimized data processing
- Progress monitoring
- Execution time tracking
- Memory usage optimization

Output Structure:

- data/: Statistical results and processed data
- plots/: Generated visualizations
- logs/: Processing logs and benchmarks

main_fuzzy_cmeans *Execute Fuzzy C-means Clustering Analysis and Visualize Results*

Description

Executes the complete workflow from data loading through preprocessing, clustering, and result visualization using Fuzzy C-means algorithm.

Usage

```
main_fuzzy_cmeans (
    input_file,
    output_dir,
    lang = "ja",
    verbose = FALSE,
    benchmark = FALSE
)
```

Arguments

input_file	Input file path specifying the data file for analysis
output_dir	Output directory where results and plots will be saved
lang	Language code for output messages and labels (e.g., 'ja' for Japanese)
verbose	Debug information display flag. If TRUE, outputs detailed processing information
benchmark	Benchmark execution flag. If TRUE, measures processing time

Details

Analysis workflow:

1. Data Preparation
 - Data loading and validation
 - Feature scaling and normalization
 - Missing value handling
2. Fuzzy C-means Clustering
 - Optimal cluster number determination
 - Membership degree calculation
 - Cluster centroid computation
 - Convergence monitoring
3. Result Analysis
 - Cluster characteristic analysis
 - Membership degree evaluation
 - Validity index calculation
4. Visualization
 - Cluster distribution plots
 - Membership degree heatmaps
 - Characteristic VAS plots per cluster

Algorithm Parameters:

- Fuzzifier (m): Controls clustering fuzziness
- Epsilon: Convergence threshold
- Maximum iterations: Iteration limit
- Distance metric: Euclidean distance

Output Files:

- cluster_info.csv: Detailed clustering results
- clustering.pdf: Main clustering visualization
- vas_cluster_*.pdf: Individual cluster analyses

Value

NULL. Results are saved in the specified directory:

- Clustering results CSV file
- Clustering result scatter plots (PDF/SVG)
- VAS plots for each cluster (PDF/SVG)

`main_viewing_intention`*Main Execution Function for Movie Viewing Intention Analysis*

Description

Reads movie impression evaluation data from a CSV file and analyzes factors affecting viewing intention. Executes environment initialization, data preprocessing, and statistical analysis as a sequential workflow.

Usage

```
main_viewing_intention(  
    input_file,  
    output_dir,  
    lang = "ja",  
    verbose = FALSE,  
    benchmark = FALSE  
)
```

Arguments

<code>input_file</code>	Path to input CSV file
<code>output_dir</code>	Directory for analysis output
<code>lang</code>	Language setting ("ja": Japanese, "en": English)
<code>verbose</code>	Flag for detailed debug information
<code>benchmark</code>	Flag for processing time measurement

Details

Analysis is executed in the following steps:

1. Environment initialization (fonts, locale, etc.)
2. Creation of output directory
3. Data loading and preprocessing
4. Analysis of viewing intention factors

Error handling:

- Environment initialization failure
- Data file reading failure
- Analysis execution errors

perform_clustering *Execute and Visualize Fuzzy C-means Clustering*

Description

Executes Fuzzy C-means clustering algorithm and visualizes the results with comprehensive analysis outputs.

Usage

```
perform_clustering(data, k = 4, m = 4, max_iter = 1000, lang = "ja")
```

Arguments

data	Raw survey data frame containing Q columns for analysis
k	Number of clusters (default: 4). Specifies target number of groups for data partitioning
m	Fuzzifier parameter. Values closer to 1 produce crisper clustering, larger values produce fuzzier clustering
max_iter	Maximum number of iterations. Specifies when to terminate if algorithm doesn't converge
lang	Language code for output messages and labels (e.g., 'ja' for Japanese)

Details

Algorithm Implementation:

1. Initialization Phase
 - Random initialization of membership matrix
 - Data normalization and scaling
 - Parameter validation
2. Iterative Process
 - Update cluster centroids
 - Recalculate membership degrees
 - Check convergence criteria
3. Results Processing
 - Cluster assignment
 - Membership degree calculation
 - Cluster characteristic analysis

Parameter Selection:

- k (cluster number): Based on silhouette analysis
- m (fuzzifier): Typically between 1.5 and 3.0

- Convergence criteria: Based on membership change
- Distance metric: Euclidean distance

Visualization Features:

- Cluster distribution plots
- Membership degree visualization
- Centroid location indicators
- Confidence ellipses

Value

A list containing:

- clusters - Clustering results (membership matrix, centroids, etc.)
- plot - Scatter plot of clustering results
- data - Processed data for plotting
- centers - Coordinates of cluster centroids
- labels - Cluster labels

```
perform_country_ttest
```

Perform Country-wise t-tests

Description

Performs independent two-sample t-tests comparing responses between countries with comprehensive statistical analysis.

Usage

```
perform_country_ttest(data, lang = "ja")
```

Arguments

data	Long-format data
lang	Language code for output localization

Details

Statistical Analysis Process:

1. Data Preparation
 - Normality testing
 - Variance homogeneity check
 - Outlier detection
2. T-test Execution
 - Welch's t-test for unequal variances
 - Equal variance t-test when appropriate
 - Multiple comparison adjustment
3. Effect Size Calculation
 - Cohen's d computation
 - Confidence interval estimation
 - Standardized mean differences

Statistical Assumptions:

- Independence of observations
- Normal distribution within groups
- Homogeneity of variances (when applicable)

Output Interpretation:

- Significance levels: * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$
- Effect size interpretation guidelines
- Confidence interval interpretation

Value

Data frame containing t-test results with:

- t-statistics
- p-values
- Confidence intervals
- Effect sizes (Cohen's d)
- Group means and standard deviations

save_results	Save Analysis Results
--------------	-----------------------

Description

Integrated function for saving data frames and plots in various formats with enhanced error handling and format validation.

Usage

```
save_results(data, file_path, type = NULL)
```

Arguments

data	Data to save (data frame or ggplot object)
file_path	File path for saving the output
type	Optional output format specification ("csv", "pdf", "svg"). If not specified, determined from file extension

Details

Supported File Formats:

- Data Frames:
 - CSV (comma-separated values)
 - RDS (R binary format)
- Plots:
 - PDF (vector graphics)
 - SVG (scalable vector graphics)
 - PNG (bitmap graphics)

File Handling Features:

- Automatic directory creation
- File overwrite protection
- Format validation
- Error logging

Output Specifications:

- CSV: UTF-8 encoding with BOM
- PDF: High-resolution vector graphics
- SVG: W3C standard compliant
- PNG: 300 DPI resolution

Value

invisible(TRUE) if successful, invisible(FALSE) if failed

Examples

```
# Save data frame as CSV
df <- data.frame(x = 1:3, y = letters[1:3])
save_results(df, "output/data.csv")

# Save ggplot as PDF
p <- ggplot(df, aes(x, y)) +
  geom_point()
save_results(p, "output/plot.pdf")
```

set_language	<i>Change Language Settings</i>
--------------	---------------------------------

Description

Changes the display language for the entire system with comprehensive internationalization support.

Usage

```
set_language(lang = "ja")
```

Arguments

lang	Language code ("ja" for Japanese or "en" for English)
------	---

Details

Language System Features:

- Global language configuration
- Runtime language switching
- Cascading settings application
- Fallback handling for missing translations

Implementation Details:

1. Sets global option 'movie_analysis.lang'
2. Updates active translation mappings
3. Refreshes display components
4. Validates language selection

Supported Languages:

- Japanese (ja): Default language

- English (en): Alternative language
- Fallback to English for unsupported languages

Configuration Impact:

- Plot labels and titles
- Error messages
- Statistical output
- Documentation strings

See Also

- [get_translation](#) - For retrieving translated strings
- [get_error_message](#) - For localized error messages

Examples

```
# Set to Japanese
set_language("ja")

# Set to English
set_language("en")
```


Index

- * **VAS analysis**
 - create_vas_plot, 9
 - * **cluster analysis**
 - create_cluster_vas_plots, 5
 - * **clustering analysis**
 - main_fuzzy_cmeans, 16
 - * **clustering**
 - perform_clustering, 19
 - * **comparative analysis**
 - perform_country_ttest, 20
 - * **configuration**
 - set_language, 23
 - * **correlation analysis**
 - create_correlation_plot, 7
 - * **data analysis**
 - perform_clustering, 19
 - * **data export**
 - save_results, 22
 - * **data generation**
 - create_sample_data, 8
 - * **data loading**
 - load_and_preprocess_data, 14
 - * **data preprocessing**
 - load_and_preprocess_data, 14
 - * **error handling**
 - get_error_message, 10
 - * **file output**
 - save_results, 22
 - * **fuzzy logic**
 - main_fuzzy_cmeans, 16
 - perform_clustering, 19
 - * **initialization**
 - initialize_environment, 12
 - * **internationalization**
 - get_error_message, 10
 - get_translation, 11
 - set_language, 23
 - * **main function**
 - main_basic_trend, 15
 - main_viewing_intention, 18
 - * **sample data**
 - create_sample_data, 8
 - * **statistical analysis**
 - analyze_viewing_intention, 2, 3
 - calculate_basic_stats, 4
 - * **statistical testing**
 - perform_country_ttest, 20
 - * **system configuration**
 - initialize_environment, 12
 - * **text processing**
 - get_translation, 11
 - * **trend analysis**
 - main_basic_trend, 15
 - * **visualization**
 - apply_common_theme, 4
 - create_cluster_vas_plots, 5
 - create_coefficient_plot, 6
 - create_correlation_plot, 7
 - create_vas_plot, 9
 - create_vas_plots_comparison, 10
- analyze_viewing_intention, 2, 3, 7
- apply_common_theme, 4
- calculate_basic_stats, 4
- create_cluster_vas_plots, 5
- create_coefficient_plot, 6
- create_correlation_plot, 7
- create_sample_data, 8
- create_vas_plot, 9
- create_vas_plots_comparison, 10
- get_error_message, 10, 12, 24
- get_translation, 7, 11, 24
- initialize_environment, 12
- load_and_preprocess_data, 14

main_basic_trend, [15](#)
main_fuzzy_cmeans, [16](#)
main_viewing_intention, [18](#)

perform_clustering, [19](#)
perform_country_ttest, [20](#)

save_results, [22](#)
set_language, [23](#)