

# Package ‘clusterWebApp’

May 8, 2026

**Title** Universal Clustering Analysis Platform

**Version** 0.1.3

**Description** An interactive platform for clustering analysis and teaching based on the 'shiny' web application framework. Supports multiple popular clustering algorithms including k-means, hierarchical clustering, DBSCAN (Density-Based Spatial Clustering of Applications with Noise), PAM (Partitioning Around Medoids), GMM (Gaussian Mixture Model), and spectral clustering. Users can upload datasets or use built-in ones, visualize clustering results using dimensionality reduction methods such as Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE), evaluate clustering quality via silhouette plots, and explore method-specific visualizations and guides. For details on implemented methods, see: Reynolds (2009, ISBN:9781598296975) for GMM; Luxburg (2007) <[doi:10.1007/s11222-007-9033-z](https://doi.org/10.1007/s11222-007-9033-z)> for spectral clustering.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** shiny, shinythemes, shinycssloaders, cluster, factoextra, datasets, ggplot2, dbscan, mclust, kernlab, Rtsne, DT, dplyr, tidyr, mlbench, magrittr

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Yijin Zhou [aut, cre]

**Maintainer** Yijin Zhou <[yijin\\_zhou1116@163.com](mailto:yijin_zhou1116@163.com)>

**Repository** CRAN

**Date/Publication** 2025-07-14 16:50:02 UTC

## Contents

compute_silhouette . . . . .	2
plot_elbow . . . . .	3

plot_radar . . . . .	3
plot_silhouette . . . . .	4
prepare_data . . . . .	5
run_app . . . . .	6
run_clustering . . . . .	6
<b>Index</b>	<b>8</b>

---

compute_silhouette	<i>Compute Average Silhouette Width</i>
--------------------	---

---

## Description

Calculates the average silhouette coefficient from a silhouette object.

## Usage

```
compute_silhouette(sil)
```

## Arguments

`sil` A silhouette object as returned by [silhouette](#).

## Value

A numeric value indicating the average silhouette width, or NA if input is NULL.

## Examples

```
data <- scale(iris[, 1:4])
cl <- kmeans(data, 3)$cluster
sil <- cluster::silhouette(cl, dist(data))
if (interactive()) {
  compute_silhouette(sil)
}
```

---

plot_elbow	<i>Plot Elbow Method for KMeans</i>
------------	-------------------------------------

---

**Description**

Uses within-cluster sum of squares (WSS) to help determine the optimal number of clusters.

**Usage**

```
plot_elbow(data)
```

**Arguments**

data            A numeric matrix or data frame for clustering.

**Value**

A ggplot object showing the elbow plot.

**Examples**

```
data <- scale(iris[, 1:4])
if (interactive()) {
  plot_elbow(data)
}
```

---

plot_radar	<i>Plot Radar Chart for PAM Cluster Centers</i>
------------	---

---

**Description**

Displays the medoids of each PAM cluster using a polar radar chart.

**Usage**

```
plot_radar(data, clusters)
```

**Arguments**

data            A numeric matrix or data frame for clustering.

clusters        An integer indicating the number of clusters.

**Value**

A ggplot object showing the radar chart of cluster medoids.

## Examples

```
data <- scale(iris[, 1:4])
if (interactive()) {
  plot_radar(data, clusters = 3)
}
```

---

plot_silhouette	<i>Plot Silhouette Diagram</i>
-----------------	--------------------------------

---

## Description

Plots the silhouette diagram for a given clustering result.

## Usage

```
plot_silhouette(sil)
```

## Arguments

`sil` A silhouette object as returned by [silhouette](#).

## Value

A silhouette plot if input is not NULL, otherwise a placeholder text.

## Examples

```
data <- scale(iris[, 1:4])
cl <- kmeans(data, 3)$cluster
sil <- cluster::silhouette(cl, dist(data))
if (interactive()) {
  plot_silhouette(sil)
}
```

---

prepare_data	<i>Prepare Built-in Datasets for Clustering</i>
--------------	---

---

### Description

Loads and preprocesses a built-in dataset for clustering analysis. Depending on the dataset name provided, different cleaning steps are applied.

### Usage

```
prepare_data(dataset)
```

### Arguments

**dataset** A string specifying the dataset name. Options are: "iris", "USArrests", "mtcars", "CO2", "swiss", "Moons".

### Details

**iris** The classic iris dataset, excluding the species column.

**USArrests** State-wise arrest data. Missing values are removed.

**mtcars** Motor trend car data set. No transformation applied.

**CO2** CO2 uptake in grass plants. Only numeric columns are selected and rows with missing values are removed.

**swiss** Swiss fertility and socio-economic indicators. Used as-is.

**Moons** Synthetic non-linear dataset generated by `mlbench::mlbench.smiley()`.

### Value

A cleaned `data.frame` containing only numeric variables and no missing values.

### Examples

```
data <- prepare_data("iris")
head(data)
```

---

run_app	<i>Launch the Shiny Clustering Web App</i>
---------	--

---

**Description**

This function launches the Shiny web application located in the `inst/app` directory of the installed package. The application provides an interactive interface for clustering analysis.

**Usage**

```
run_app()
```

**Value**

No return value. This function is called for its side effect (launching the app).

**Examples**

```
if (interactive()) {  
  run_app()  
}
```

---

run_clustering	<i>Perform clustering analysis</i>
----------------	------------------------------------

---

**Description**

This function performs clustering on a numeric matrix using one of six common clustering methods: KMeans, Hierarchical, DBSCAN, PAM, Gaussian Mixture Model (GMM), or Spectral Clustering.

**Usage**

```
run_clustering(data, method, k = 3, eps = 0.5, minPts = 5)
```

**Arguments**

<code>data</code>	A numeric matrix or data frame, typically standardized, to be clustered.
<code>method</code>	A string indicating the clustering method to use. Options are: "KMeans", "Hierarchical", "DBSCAN", "PAM", "GMM", "Spectral".
<code>k</code>	An integer specifying the number of clusters. Required for KMeans, Hierarchical, PAM, GMM, and Spectral.
<code>eps</code>	A numeric value specifying the epsilon parameter for DBSCAN. Default is 0.5.
<code>minPts</code>	An integer specifying the minimum number of points for DBSCAN. Default is 5.

**Value**

A list containing two elements:

**cluster** A vector of cluster labels assigned to each observation.

**silhouette** An object of class `silhouette` representing silhouette widths.

**Examples**

```
data(iris)
result <- run_clustering(scale(iris[, 1:4]), method = "KMeans", k = 3)
print(result$cluster)
if (interactive()) {
  plot(result$silhouette)
}
```

# Index

`compute_silhouette`, [2](#)

`plot_elbow`, [3](#)

`plot_radar`, [3](#)

`plot_silhouette`, [4](#)

`prepare_data`, [5](#)

`run_app`, [6](#)

`run_clustering`, [6](#)

`silhouette`, [2, 4](#)