

# Package ‘traj’

January 22, 2023

**Title** Trajectory Analysis

**Version** 1.3

**Description** Implements the three-step procedure proposed by Leffondree et al. (2004) to identify clusters of individual longitudinal trajectories. The procedure involves (1) calculating 24 measures describing the features of the trajectories; (2) using factor analysis to select a subset of the 24 measures and (3) using cluster analysis to identify clusters of trajectories, and classify each individual trajectory in one of the clusters.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Depends** R (>= 3.5.0)

**LazyData** true

**Imports** cluster, GPArotation, graphics, grDevices, NbClust, pastecs,  
psych, stats, utils

**Repository** CRAN

**Suggests** rmarkdown, knitr, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**URL** <https://github.com/tchouangue/traj>

**BugReports** <https://github.com/tchouangue/traj/issues>

**NeedsCompilation** no

**Author** Marie-Pierre Sylvestre [aut],  
Dan Vatnik [aut],  
Gillis Delmas TCHOUANGUE DINKOU [cre]

**Maintainer** Gillis Delmas TCHOUANGUE DINKOU <gillisdla2@gmail.com>

**Date/Publication** 2023-01-21 23:20:06 UTC

## R topics documented:

example.data . . . . .	2
plotBoxplotTraj . . . . .	3
plotCombTraj . . . . .	4
plotMeanTraj . . . . .	5
plotMedTraj . . . . .	6
step1measures . . . . .	8
step2factors . . . . .	10
step3clusters . . . . .	12
wrapperTraj . . . . .	14

<b>Index</b>	<b>16</b>
--------------	-----------

---

example.data	<i>Example data</i>
--------------	---------------------

---

### Description

Example data used to display the features of the traj package.

### Usage

```
example.data
```

### Format

List of 2 data.frames:

\$ data: 130 obs. of 7 variables. X1 to X6 correspond to the 6 measurements obtained on the 130 subjects:

```
ID: num  1 2 3 4 5 6 7 8 9 10 ...
```

```
X1: num  5.66 23.59 15.47 7.31 12.84 ...
```

```
X2: num  9.34 11.75 8.76 11.69 11.09 ...
```

```
X3: num  3.77 7.68 6.49 12.48 7.65 ...
```

```
X4: num  17.36 12.83 11.26 8.89 10.27 ...
```

```
X5: num  8.82 13 10.42 6.52 12.45 ...
```

```
X6: num  9.28 9.66 17.41 7.7 11.56 ...
```

\$ time: 130 subjects with 7 variables. time1 to time6 correspond to the measurement times for the variables X1 to X6:

```

ID      : num  1 2 3 4 5 6 7 8 9 10 ...
time.1: num  1 1 1 1 1 1 1 1 1 1 ...
time.2: num  2 2 2 2 2 2 2 2 2 2 ...
time.3: num  3 3 3 3 3 3 3 3 3 3 ...
time.4: num  4 4 4 4 4 4 4 4 4 4 ...
time.5: num  5 5 5 5 5 5 5 5 5 5 ...
time.6: num  6 6 6 6 6 6 6 6 6 6 ...

```

**Author(s)**

Marie-Pierre Sylvestre, Dan Vatnik  
marie-pierre.sylvestre@umontreal.ca

**Examples**

```

## Not run:
# data and time
data = example.data$data
time = example.data$time

## End(Not run)

```

---

plotBoxplotTraj      *Plot Boxplot of traj Object*

---

**Description**

Produce a boxplot of the values of the trajectories from each cluster at every time points.

**Usage**

```
plotBoxplotTraj(x, clust.num = NULL, ...)
```

**Arguments**

x	traj object.
clust.num	Integer indicating the cluster number to plot.NULL to print all clusters. Defaults to NULL.
...	Arguments to be passed to plot.

**Details**

The function plots a boxplot of values of the trajectories in a cluster at each time point.

**Author(s)**

Marie-Pierre Sylvestre, Dan Vatnik  
marie-pierre.sylvestre@umontreal.ca

**See Also**

[boxplot](#)

**Examples**

```
## Not run:
# Setup data
data = example.data$data

# Run step1measures, step2factors and step3clusters with
#a predetermined number of clusters
s1 = step1measures(data, ID=TRUE)
s2 = step2factors(s1)
s3.4clusters = step3clusters(s2, nclust = 4)

# Plot boxplots
plotBoxplotTraj(s3.4clusters)

## End(Not run)
```

---

plotCombTraj	<i>Plot Cluster-Specific Mean or Median Trajectories provided by a traj object</i>
--------------	--

---

**Description**

Plot cluster-specific mean or median trajectories.

**Usage**

```
plotCombTraj(x, stat.type = "mean", colored = FALSE, ...)
```

**Arguments**

x	traj object.
stat.type	Choice between "mean" or "median". The mean or the median calculated at each time point for a cluster-specific set of trajectories will be plotted. Defaults to "mean."

colored	Boolean in dictating if the plot should use colors. If not, the trajectory lines will be distinctively patterned. Defaults to FALSE for patterns.
...	Any extra parameter used by the plot function.

### Details

The function plots the mean or the median cluster-specific trajectory, calculated at each time point. A legend is generated in the top left corner of the plot. Other plotting parameter(s) can be added to the function with the use of ...

### Author(s)

Marie-Pierre Sylvestre, Dan Vatnik  
marie-pierre.sylvestre@umontreal.ca

### See Also

[mean](#) [median](#)

### Examples

```
## Not run:  
# Setup data  
data = example.data$data  
  
# Run step1measures, step2factors and step3clusters with a predetermined number of clusters  
s1 = step1measures(data, ID=TRUE)  
s2 = step2factors(s1)  
s3.4clusters = step3clusters(s2, nclust = 4)  
  
# Plot mean combination trajectories  
plotCombTraj(s3.4clusters)  
  
## End(Not run)
```

---

plotMeanTraj

*Plot Mean Trajectory*

---

### Description

Plot cluster-specific mean trajectory for one or all clusters provided by a traj object.

### Usage

```
plotMeanTraj(x, clust.num = NULL, ...)
```

### Arguments

x	traj object.
clust.num	Integer indicating the cluster number to plot.NULL to print all clusters. Defaults to NULL.
...	Arguments to be passed to plot.

### Details

The function plots cluster specific mean trajectory calculated at each time point. By setting the `clust.num` argument to an integer corresponding to a cluster number, one can plot the mean trajectory of that cluster only. Any other plotting arguments can be added to the function.

### Author(s)

Marie-Pierre Sylvestre, Dan Vatnik  
marie-pierre.sylvestre@umontreal.ca

### Examples

```
## Not run:  
# Setup data  
data = example.data$data  
  
# Run step1measures, step2factors and step3clusters with a predetermined number of clusters  
s1 = step1measures(data, ID=TRUE)  
s2 = step2factors(s1)  
s3.4clusters = step3clusters(s2, nclust = 4)  
  
# Plot mean trajectories  
plotMeanTraj(s3.4clusters)  
  
## End(Not run)
```

---

plotMedTraj

*Plot Median Trajectory of traj Object*

---

### Description

Plot cluster-specific median trajectory for one or all clusters provided by a traj object.

### Usage

```
plotMedTraj(  
  x,  
  clust.num = NULL,  
  plot.percentile = TRUE,  
  low.percentile = 0.1,
```

```

    high.percentile = 0.9,
    ...
)

```

### Arguments

x	traj object.
clust.num	Integer indicating the cluster number to plot. NULL to print all clusters. Defaults to NULL.
plot.percentile	Value indicating if the function should plot percentiles. Defaults to TRUE.
low.percentile	Value of the lower percentile to be plotted. Must be between 0 and 1. Defaults to 0.1.
high.percentile	Value of the high percentile to be plotted. Must be between 0 and 1. Defaults to 0.9.
...	Extra parameters used in the plot function.

### Details

The function plots cluster specific median trajectory calculated at each time point, in addition to 10th and 90th percentiles. By setting the `clust.num` argument to an integer corresponding to a cluster number, one can plot the median trajectory of that cluster only. Any other plotting arguments can be added to the function.

### Author(s)

Marie-Pierre Sylvestre, Dan Vatnik  
 marie-pierre.sylvestre@umontreal.ca

### See Also

[plot median quantile](#)

### Examples

```

## Not run:
# Setup data and time
data = example.data$data
time = example.data$time

# Run step1measures, step2factors and step3clusters with a predetermined number of clusters
s1 = step1measures(data,time, ID=TRUE)
s2 = step2factors(s1)
s3.4clusters = step3clusters(s2, nclust = 4)

# Plot median trajectories
plotMedTraj(s3.4clusters)

```

```
## End(Not run)
```

---

step1measures	<i>Compute 24 Measures Describing the Features of the Trajectories</i>
---------------	--

---

### Description

Compute 24 measures for each of the Trajectories. See details for the list of measures.

### Usage

```
step1measures(Data, ID = FALSE, verbose = TRUE)
```

### Arguments

Data	A $n$ by $m$ matrix or data frame containing the values of each individual trajectory. Each row corresponds to one of the $n$ trajectories, while the $m$ columns correspond to the ordered values of a given trajectory. See details.
ID	Logical. Set to TRUE if the first column of Data corresponds to an ID variable. Defaults to FALSE.
verbose	Logical. Set to TRUE to print information on screen. Defaults to TRUE.

### Details

There must be a minimum of 4 observations for each trajectory or the trajectory will be omitted from the analysis. The trajectories do not need to have the same number of observations, nor the same values of Time.

When ID is set to FALSE, a generic ID variable is created and appended as the first column of both the Data and Time data.frames. The 24 measures are:

1. Range
2. Mean-over-time\*
3. Standard deviation (SD)
4. Coefficient of variation (CV)
5. Change
6. Mean change per unit time
7. Change relative to the first score



8. Change relative to the mean over time
9. Slope of the linear model\*
10.  $R^2$ : Proportion of variance explained by the linear model
11. Maximum of the first differences
12. SD of the first differences
13. SD of the first differences per time unit
14. Mean of the absolute first differences\*
15. Maximum of the absolute first differences
16. Ratio of the maximum absolute difference to the mean-over-time
17. Ratio of the maximum absolute first difference to the slope
18. Ratio of the SD of the first differences to the slope
19. Mean of the second differences
20. Mean of the absolute second differences
21. Maximum of the absolute second differences
22. Ration of the maximum absolute second difference to the mean-over-time
23. Ratio of the maximum absolute second difference to mean absolute first difference
24. Ratio of the mean absolute second difference to the mean absolute first difference

- If a measure is equal to zero, it will be set to the smallest, non-zero value of the same measure across the sample during further calculations. If  $Y_1$ , the first observation of the trajectory of an individual, is equal to zero, it will also be replaced.

For the exact equations of the measures, please go to "User guides, package vignettes and other documentation" section of the "traj" package.

**Value**

trajMeasures, object containing the data used for the calculations and the 24 measures.

**Author(s)**

Marie-Pierre Sylvestre, Dan Vatnik  
marie-pierre.sylvestre@umontreal.ca

**Examples**

```
## Not run:
# Setup data
data = example.data$data

# Run step1measures
s1 = step1measures(data, ID=TRUE)

# Display measures
head(s1$measurments)

# Plot mean trajectory of all individuals
plot(s1$measurments$ID, s1$measurments$m5)

# The next step would be to run "step2factors"

## End(Not run)
```

---

step2factors

*Performs Factor Analysis to Select a Subset of the 24 Measures*


---

**Description**

Performs a factor analysis to reduce the set of 24 measures into a smaller set of measures that captures the main features of the trajectories.

**Usage**

```
step2factors(
  trajMeasures,
  num.factors = NULL,
  discard = NULL,
  verbose = TRUE,
  ...
)
```

**Arguments**

trajMeasures	List generated by step1mesures. Contains original data, original time and 24 measures.
num.factors	Numerical value specifying the number of factors to choose. Defaults to NULL. See details.Defaults to NULL.

discard	Vector containing names or numerical positions of measures to discard during factor analysis.
verbose	Logical indicating if the function should print information on screen. Defaults to TRUE.
...	Arguments to be passed to <code>principal</code> . See details.

### Details

If `num.factor` is NULL, the function will select the number of factors as the number of eigenvalues greater than 1.

The `principal` function is used in order to choose the measure that will represent each factor. `varimax` is used to rotate the data during the execution of the `principal` function. Any other parameter can be passed through `...` in order to further control the `principal` function.

If any measures that happen to be extremely correlated among themselves ( $\text{corr.} \geq 0.95$ ), one of them will have to be removed. Such measures are flagged by `step1measures`. These values can be removed with `discard` or they will be automatically removed by the function.

### Value

`trajFactors` Object containing the measures chosen as factors, the eigenvalues of the correlation matrix of the 24 measures, the list generated by the `principal` function used for the factor analysis and the data stored in the `trajMeasures` object.

### Author(s)

Marie-Pierre Sylvestre, Dan Vatnik  
marie-pierre.sylvestre@umontreal.ca

### See Also

[principal](#) [step1measures](#)

### Examples

```
## Not run:
# Setup data
data = example.data$data

# Run step1measures and step2factors
s1 = step1measures(data, ID=TRUE)
s2 = step2factors(s1)

# Display factors
head(s2$factors)

# The next step would be to run "step3clusters"

## End(Not run)
```

---

step3clusters	<i>Cluster Trajectories According to the Subset of Measures Selected Previously</i>
---------------	---

---

### Description

Classify trajectories based on the factors identified in `step2factors`.

### Usage

```
step3clusters(
  trajFactors,
  nclusters = NULL,
  nstart = 50,
  criteria = "ccc",
  forced.factors = NULL
)
```

### Arguments

<code>trajFactors</code>	Object generated by <code>step2factors</code> . Contains data factors, eigenvalues, principal factors as well as the original data.
<code>nclusters</code>	Integer number indicating the number of clusters to use in order to classify the trajectories. If <code>NULL</code> , the function selects the number of clusters based on an automated criteria specified by <code>index</code> . Defaults to <code>NULL</code> .
<code>nstart</code>	Integer number designating the number of seedings that <code>kmeans</code> should do in order to cluster the
<code>criteria</code>	String indicating the criteria to select the number of clusters. Defaults to <code>ccc</code> (Cubic clustering criterion).
<code>forced.factors</code>	(Optional) Vector containing the names of the measures calculated in <code>step1measures</code> to force as factors for the clustering. This vector will override the factors selected by <code>step2factors</code> . Available options: "m1", "m2", "m3", ..., "m23" and "m24". Defaults to <code>NULL</code> . See details.

### Details

If `nclusters` is set to `NULL`, the function will use the `NbClust` function to select the optimal number of clusters. The `NbClust` function uses `kmeans` as the cluster analysis method. The measures are standardized within the `step3clusters` function prior to clustering. The criteria to be computed can be chosen by the `criteria` argument. The list of available methods and criteria can be found in the `NbClust` help page. Criteria compatible with `step3clusters` are: "ch", "kl", "ccc", "hartigan", "scott", "trcovw", "tracew" and "friedman". It is important to note that some of these criteria will not always yield the same number of clusters when run multiple times. Increasing `nstart` will generally stabilize the results.

The function then uses `kmeans` in order to cluster the trajectories in the required number of clusters. If `nclusters` is set to `NULL`, then the number of clusters is computed by then the data will be

classified into that number of clusters. `kmeans` uses the `nstart` argument in order to select how many random sets should be run during its execution. If the function does not converge, increasing `nstart` can improve the result. Please consult the [kmeans](#) help page for more information.

When `forced.factors` is set to `NULL`, the function will select the factors identified by `step2factors` in order to cluster the trajectories. When the parameter is set to a vector, it must contain at least one measure name such as: "m1", "m2", "m3", ..., "m23" and "m24". The function will then cluster the trajectories using the stated measures. These measures are generated by `step1measures`. They range from "m1" to "m24". All of these measures are found in the `trajMeasures` object.

When the plot function is run without changing the default values, only a `traj` object is required. The function will generate a multiplot of all the clusters. In each plot, 10 randomly selected trajectories will be traced. The same number of trajectories for each cluster will be plotted. If the function is rerun, the plots will not look the same because the trajectories are randomly sampled. Seeding is required in order replicate a plot.

If `color.vect` is `NULL`, the function will randomly assign a color to each trajectory. The same colors will be used for all the trajectories in each plot. If specific colors are chosen, there must be as many colors in the vector as there are trajectories to be plotted or an error will be thrown.

If `clust.num` is set to an integer, the cluster associated with that integer will be plotted. Only that one will be displayed among the available clusters.

The `print` function displays the number of observations used in the computation of `traj`, the number of clusters as well as the number of observations in each one and the measures set as factors. These factors are used to cluster the data. The number of decimal places is defaulted to 2, it can be changed in the arguments of [step3clusters](#).

The `summary` function displays the number of observations analysed as well as the total number of clusters into which the data was classified. Prints the eigenvalues used to determine the number of factors to be selected in [step2factors](#). Prints summary statistics of each of the factors by cluster. The number of decimal places is defaulted to 2, it can be changed in the parameters of [step3clusters](#).

## Value

The function returns a `traj` object that contains objects carried through steps 1 and 2 which includes the original data, measures and factors.

Furthermore, it includes a `data.frame` containing the ID corresponding to each trajectory, and the cluster number in which the trajectory was classified. This is stored in the `clusters` field of the `traj` object. It also contains the cluster distribution of the observations.

Methods to plot the output of `step3clusters` include:

<code>plot</code>	plots a 10 person sample from every cluster
<code>plotMedTraj</code>	plots the median trajectory of the clusters
<code>plotMeanTraj</code>	plots the mean trajectory of the clusters
<code>plotBoxplotTraj</code>	produce a boxplot of trajectories of every cluster

**Author(s)**

Marie-Pierre Sylvestre, Dan Vatnik  
marie-pierre.sylvestre@umontreal.ca

**See Also**

[NbClust](#) [kmeans](#) [step1measures](#) [step2factors](#) [plot](#)

**Examples**

```
## Not run:
# Setup data
data = example.data$data

# Run step1measures, step2factors and step3clusters
s1 = step1measures(data, ID=TRUE)
s2 = step2factors(s1)
s3 = step3clusters(s2)

# Print and plot 'traj' object
s3
plot(s3)

# Run step3clusters with predetermined number of clusters
s3.4clusters = step3clusters(s2, nclusters=4)

# Display 'traj' object s3.4clusters
summary(s3.4clusters)
plot(s3.4clusters)

s3$cluster[1:10,]

## End(Not run)
```

---

wrapperTraj

*Wrapper Function to Perform Trajectory Analysis*

---

**Description**

Run three steps of trajectory analysis with default parameters.

**Usage**

```
wrapperTraj(Data, ID = FALSE)
```

**Arguments**

Data	Data frame containing trajectory data. Each line should contain sequential observations. See details and help file for the <a href="#">step1measures</a> function.
ID	Logical. Set to FALSE if the first column of Data corresponds to an ID variable. Defaults to FALSE.

**Details**

The function runs the full three step trajectory analysis and returns a traj object. It will execute [step1measures](#), [step2factors](#) and [step3clusters](#) sequentially with their default parameters. The result of [step3clusters](#) will be returned. Details regarding the data and the time arguments are found in the 'Details' section of [step1measures](#).

**Value**

The result is a traj object. Details can be found in the 'Value' section of [step3clusters](#).

**Author(s)**

Marie-Pierre Sylvestre, Dan Vatnik  
marie-pierre.sylvestre@umontreal.ca

**Examples**

```
## Not run:  
# Setup data  
data = example.data$data  
  
# Run clustering wrapper function  
wt = wrapperTraj(data, ID = TRUE)  
  
# Display and plot "traj" object  
wt  
  
summary(wt)  
  
plot(wt)  
  
## End(Not run)
```

# Index

- \* **datasets**
  - example.data, 2
- boxplot, 4
- example.data, 2
- kmeans, 12–14
- mean, 5
- median, 5, 7
- NbClust, 12, 14
- plot, 7, 14
- plotBoxplotTraj, 3, 13
- plotCombTraj, 4
- plotMeanTraj, 5, 13
- plotMedTraj, 6, 13
- principal, 11
- quantile, 7
- step1measures, 8, 11, 12, 14, 15
- step2factors, 10, 12–15
- step3clusters, 12, 12, 13, 15
- wrapperTraj, 14