

Package ‘HMLET’

August 11, 2024

Type Package

Title What the Package Does (Title Case)

Version 2.0

Author Alireza Kazemi

Maintainer Alireza Kazemi <alireza.kzmi@gmail.com>

Description HMLET uses machine learning algorithm to analyze Eye-Tracking data.

License MIT + file LICENSE

Imports combinat,

dplyr,

ggplot2,

miceadds,

purrr,

reshape2,

rray,

stats,

tidyr,

utils,

vdiffr,

zoo

Suggests testthat (>= 3.0.0)

Author@R person("`Alireza", "`Kazemi", email = "`kazemi@gmail.com", role =
c("`aut", "`cre")), person("`Grisha", "`Bandodkar", email =
"`gbandodkar@ucdavis.edu", role = c("`ctb", "`com"))

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Contents

AddClusterInfotoData_HMLET	2
ClusterStats_HMLET	3
ComputeSamplingInterval_HMLET	3
ComputeSubjectLevelPermLabels_HMLET	4

ComputeTValues_HMLET	4
CreateTimeBinData_HMLET	5
ExportDataForGUI_HMLET	6
FilterClusterTimePoints_HMLET	7
FindClusters_HMLET	7
GenerateRandomData_HMLET	8
PermutationTestDataPrep_HMLET	9
PermutationTestSingleParticipant_HMLET	10
PermutationTest_HMLET	10
PlotNullDistribution_HMLET	11
PlotTimeSeries_HMLET	12
Preprocess_AOIExtraction_HMLET	13
Preprocess_CheckData_HMLET	14
Preprocess_FixSampling_HMLET	15
Preprocess_Interpolate_HMLET	17
RelativeViewingProportionLabels_HMLET	18
RemoveIncompleteTimePoints_HMLET	19
SubjectLevelPermutationTestWithin_HMLET	19
TrialLevelPermutationTestWithin_HMLET	20
UniquePermutations_HMLET	20

Index 21

AddClusterInfoData_HMLET

Add cluster information and stats to the corresponding time points in the initial dataframe.

Description

Add cluster information and stats to the corresponding time points in the initial dataframe.

Usage

```
AddClusterInfoData_HMLET(data, clusterInf)
```

Arguments

data	data frame with temporal data for permutation tests routine, output from PermutationTestDataPrep_HMLET.R.
clusterInf	data frame with cluster information, output from ClusterStats_HMLET or PermutationTest_HMLET

Value

data frame compatible for permutation tests and other HMLET modules

ClusterStats_HMLET	<i>Adds cluster statistics to t value dataframe, includes all info for all tests.</i>
--------------------	---

Description

Adds cluster statistics to t value dataframe, includes all info for all tests.

Usage

```
ClusterStats_HMLET(data, paired = T, detailed = F, threshold_t = NULL)
```

Arguments

data	data frame with temporal data for permutation tests routine, output from PermutationTestDataPrep_HMLET.R.
paired	optional boolean to control T-tests. True is for statistical comparisons in paired mode, False for a two-sample T-Test; defaults to True.
detailed	optional boolean to include cluster statistics and t value statistics within one dataframe. Defaults to False to populate dataframe only with cluster statistics.
threshold_t	optional probability threshold for statistical comparison computed based on number of subjects. alpha = 0.025.

Value

returns data frame with cluster statistics

ComputeSamplingInterval_HMLET	<i>Compute the sampling time interval</i>
-------------------------------	---

Description

This function estimates the sampling interval for each trial based on the time differences between each time point and uses mode to find the interval. Note that your dataset has to have a unique consistent time interval between all samples. In case, several time intervals is estimated for your dataset it use "Preprocess_FixSampling_HMLET" function and pass the highest sampling interval to fix the inconsistency.

Usage

```
ComputeSamplingInterval_HMLET(
  data,
  ID = "ID",
  trial,
  timePoint,
  appendFlag = T,
  userCallFlag = T
)
```

Arguments

data	long format dataframe containing temporal data.
ID	string for column name that represents IDs within data frame, defaults to "ID".
trial	string for column name that represents trials within data frame. Can be numerical or categorical.
timePoint	string for column name that represents time stamps of each sample.
appendFlag	optional flag specifies whether the interval(s) being appended to the data or returned separately, defaults to TRUE.
userCallFlag	optional flag to print out data when users are calling this function, defaults to TRUE.

ComputeSubjectLevelPermLabels_HMLET

Provides unique permutation indices within each subject.

Description

Provides unique permutation indices within each subject.

Usage

```
ComputeSubjectLevelPermLabels_HMLET(labels, n = 1)
```

Arguments

labels	labels from the data frame.
n	optional integer for number of permutations to resample data.

Value

labels of unique permutation indices to use for subject level permutation test.

ComputeTValues_HMLET *Compute t values for permutation tests.*

Description

Compute t values for permutation tests.

Usage

```
ComputeTValues_HMLET(respTime, paired = TRUE)
```

Arguments

respTime	dataframe for responses in temporal order.
paired	optional boolean to control T-tests. True is for statistical comparisons in paired mode, False for a two-sample T-Test; defaults to True.

Value

t values in the format of a data frame.

CreateTimeBinData_HMLET

Collapse time points in specified time bins.

Description

Collapse time points in specified time bins.

Usage

```
CreateTimeBinData_HMLET(
  data,
  groupingColumns = NULL,
  timeBinWidth = 250,
  timeMax = 3000,
  FixatedOn,
  timePoint = "timeStamp",
  AOIs = NULL,
  timeForward = T,
  aggregateFun = mean
)
```

Arguments

data	dataframe containing temporal data.
groupingColumns	optional array of strings for column names of variables to be grouped, defaults to NULL.
timeBinWidth	optional integer specified time interval for time bins, defaults to 250.
timeMax	optional integer for maximum time in temporal order, samples are left out if duration is longer. Defaults to 3000.
FixatedOn	string for column name of specific AOI in dataframe to be fixated on.
timePoint	optional string representing column name for time Stamps, defaults to "timeStamp".
AOIs	optional array of strings representing column names for areas of interest, defaults to NULL.
timeForward	optional boolean to sort timebins, defaults to True for ascending order.
aggregateFun	optional function for aggregation, defaults to mean.

Value

temporal data with sorted time points into time bins.

ExportDataForGUI_HMLET

ExportDataForGUI_HMLET Export samples of trial with maximum time point to CSV file for MATLAB GUI.

Description

ExportDataForGUI_HMLET Export samples of trial with maximum time point to CSV file for MATLAB GUI.

Usage

```
ExportDataForGUI_HMLET(
    data,
    ID = "ID",
    trial = "trial",
    timePoint = "timePoint",
    timeMax = NULL,
    dataPointDuration,
    response,
    fixation,
    condition,
    testName = NULL,
    gazeX,
    gazeY,
    gazeXRelative = NULL,
    gazeYRelative = NULL,
    miscVars = NULL,
    fileName = "HMLET_DataforGUI.csv",
    path = getwd()
)
```

Arguments

data	dataframe containing temporal data.
ID	optional string for column name that represents IDs within data frame, defaults to "ID."
trial	string for column name that represents trials within data frame, for example "TrialNum."
timePoint	string for column name that represents time intervals, for example "timeStamp."
timeMax	optional integer for maximum time in temporal order, samples are left out if duration is longer. Default is NULL
dataPointDuration	optional integer or character for specified duration of samples, used for time bins in CreateTimeBinData_HMLET.
response	the column in which participants' responses are stored e.g., HIT, Miss, FA, and CR.
fixation	data for specific AOI in dataframe to be fixated on. #TODO: ASK
condition	string for column name that specifies a condition within the data frame.

testName	optional string for name of data – used as condition name or test names to compare permutation test results between different tests/conditions later, defaults to NULL.
gazeX	integer X-coordinate of gaze point on the screen according to top left corner.
gazeY	integer Y-coordinate of gaze point on the screen according to top left corner.
gazeXRelative	optional integer X-coordinate of gaze point relative to an arbitrary center.
gazeYRelative	optional integer Y-coordinate of gaze point relative to an arbitrary center.
miscVars	optional list of strings containing column names of additional variables to be included in exported file
fileName	optional string for output csv file, defaults to "HMLET_DataforGUI.csv."
path	optional path to identify the destination location default: current working directory

FilterClusterstimePoints_HMLET
not being used.

Description

not being used.

Usage

```
FilterClusterstimePoints_HMLET(data, clusterInf = clusterInf)
```

FindClusters_HMLET	<i>Cluster all samples whose t value is greater than threshold value by temporal adjacency.</i>
--------------------	---

Description

Cluster all samples whose t value is greater than threshold value by temporal adjacency.

Usage

```
FindClusters_HMLET(tValues, threshold_t = threshold_t)
```

Arguments

tValues	dataframe of t values calculated from ComputeTValues_HMLET.
threshold_t	optional probability threshold for statistical comparison, defaults to NA and will be computed based on number of trials when "between trials permutation" is called or will be computed based on number of subjects when "between subjects permutation" is called. Alpha = 0.025.

Value

t value dataframe with appended cluster information.

GenerateRandomData_HMLET

Generate Data

Description

This function generates data to simulate a desired effect size at a specified time interval. This data is generated only for testing and evaluation of the HMLET package and is not valid for other purposes or research.

Usage

```
GenerateRandomData_HMLET(
  trialNum = 20,
  subjectNum = 20,
  timeMax = 1000,
  samplingInterval = 17,
  effectInterval = c(250, 350),
  effectSize = 0.2,
  sdCond1 = 1,
  sdCond2 = 1,
  smoothingPar = 0.4
)
```

Arguments

trialNum	Number of trials, defaults to 20.
subjectNum	Number of subjects, defaults to 20.
timeMax	Length of each trial in milliseconds, defaults to 1000.
samplingInterval	Specify the expected sampling interval in milliseconds, defaults to 17.
effectInterval	The expected time interval to simulate the effect, defaults to c(250,350).
effectSize	Size of the effect
sdCond1	standard deviation for condition 1, defaults to 1.
sdCond2	standard deviation for condition 2, defaults to 1.
smoothingPar	smoothing parameter to smooth the generated data based on spar parameter in smooth.spline(), defaults to 0.4

Details

Data has two within participants' conditions: Cond1 and Cond2. Trial level data for the number of participants will be generated with an effect appearing in the specified time interval in which the two conditions are different with the specified effect size

PermutationTestDataPrep_HMLET

Return a data frame compatible for permutation tests routine.

Description

Return a data frame compatible for permutation tests routine.

Usage

```
PermutationTestDataPrep_HMLET(
  data,
  ID = "ID",
  trial = NULL,
  timePoint,
  condition,
  gazeMeasure,
  conditionLevels = NULL,
  targetAOI = NULL,
  testName = NULL,
  timeBinName = NULL
)
```

Arguments

data	data frame with temporal data for permutation tests routine.
ID	optional string for column name that represents IDs within data frame, defaults to "ID".
trial	string for column name that represents trials within data frame.
timePoint	string for column name that includes time point or index values in numeric form.
condition	string for column name that specifies a condition within the data frame.
gazeMeasure	string for column name which includes numerical value of the gaze measure at each time point such as gaze proportion.
conditionLevels	optional array of string column names to denote all the values of conditions, defaults to NULL.
targetAOI	optional string specify the name of the target AOI which should be set as 1 in the gazeMeasure variable, defaults to NULL.
testName	optional string for name of data – used as condition name or test names to compare permutation test results between different tests/conditions later, defaults to NULL.
timeBinName	optional string for the column name that includes time bin names, defaults to NULL.

Value

data frame compatible for permutation tests and other HMLET modules

PermutationTestSingleParticipant_HMLET

Conduct trial level permutation test for a single participant.

Description

Conduct trial level permutation test for a single participant.

Usage

```
PermutationTestSingleParticipant_HMLET(data, samples = 2000, threshold_t = NA)
```

Arguments

data	data frame with temporal data for permutation tests routine, output from PermutationTestDataPrep_HMLET.R.
samples	optional integer for number of samples, defaults to 2000.
threshold_t	optional probability threshold for statistical comparison, defaults to NA and will be computed based on number of trials when "between trials permutation" is called or will be computed based on number of subjects when "between subjects permutation" is called. Alpha = 0.025.

Value

a list of permutation tests in which distribution of desired statistic under the null hypothesis is estimated in a large number of permutations of the original data

PermutationTest_HMLET *Permutation Tests General Routine.*

Description

Permutation Tests General Routine.

Usage

```
PermutationTest_HMLET(
  data,
  samples = 2000,
  paired = T,
  permuteTrialsWithinSubject = F,
  threshold_t = NA
)
```

Arguments

data	data frame with temporal data for permutation tests routine, output from PermutationTestDataPrep_HMLET.R.
samples	optional integer for number of samples, defaults to 2000.
paired	optional boolean to control T-tests. True is for statistical comparisons in paired mode, False for a two-sample T-Test; defaults to True.
permuteTrialsWithinSubject	optional boolean to run permutation test pipeline for conducting permutation tests between trials or between subjects, defaults to False.
threshold_t	optional probability threshold for statistical comparison, defaults to NA and will be computed based on number of trials when "between trials permutation" is called or will be computed based on number of subjects when "between subjects permutation" is called. Alpha = 0.025.

Value

a list of permutation tests in which distribution of desired statistic under the null hypothesis is estimated in a large number of permutations of the original data

PlotNullDistribution_HMLET

Plot Null Distribution.

Description

Plot Null Distribution.

Usage

```
PlotNullDistribution_HMLET(resultList, smoothingBandWidth = NULL)
```

Arguments

resultList	list of the dataframes that is already prepared by PrepareMLETData_HMLET or a list that is the result of PermutationTest_HMLET.
smoothingBandWidth	The smoothing bandwidth which determines the granularity of distribution estimation, defaults to sd/3

Value

null distribution plot handle.

PlotTimeSeries_HMLET *Plot temporal trends for time series.*

Description

Plot temporal trends for time series.

Usage

```
PlotTimeSeries_HMLET(
  resultList,
  showDataPointProp = T,
  showOverallMean = "none",
  gazePropRibbonAlpha = 0.1,
  clusterFillColor = "#CC9933",
  clusterFillAlpha = 0.5,
  pointSize = 1,
  pointAlpha = 0.7,
  pointFatten = 3,
  testNameOrder = NULL,
  conditionOrder = NULL,
  onlySignificantClusters = T,
  clusterData = NULL,
  yLabel = "Gaze Proportion",
  dataAxisLabel = "Available Observations (%)",
  xLabel = "Time (ms)",
  lineWidthGazeProp = 1,
  lineWidthDataPointProp = 1,
  lineWidthOverallMean = 1,
  lineTypeOverallMean = "dotted",
  alphaOverallMean = 0.5,
  shapeCodeOverallMean = 1,
  tickSizeOverallMean = NULL
)
```

Arguments

resultList	dataframe of the data that is already prepared by PrepareMLETData_HMLET or a list that is the result of PermutationTest_HMLET.
showDataPointProp	optional boolean to display data points on the axes, defaults to True.
showOverallMean	optional string to specify the overall mean. "Line" = to display a horizontal line on the y = average. "Point" to display overall average and range in a pointRange format. "none" to not display it, defaults to "none".
gazePropRibbonAlpha	optional float to change opacity of gaze ribbon, defaults to 0.1.
clusterFillColor	optional string to change fill color of clustered data points, defaults to "#CC9933".

clusterFillAlpha	optional float to change opacity of clustered data points, defaults to 0.5.
pointSize	optional integer to change size of points on plot, defaults to 1.
pointAlpha	optional float to change opacity of points on plot, defaults to 0.7.
pointFatten	optional integer to change size of lines on plot, defaults to 3.
testNameOrder	specify order of test name labels, defaults to NULL.
conditionOrder	specify order of condition levels, defaults to NULL.
onlySignificantClusters	optional Boolean to plot only the significant clusters, defaults to True.
clusterData	data containing all clusters generated from ClusterStats_HMLET to display on plot, defaults to NULL.
yLabel	Proportion" specify label for the y axis, defaults to "Gaze"
dataAxisLabel	Specify label for the second y axis on the right, defaults to "Data Points \(%\)".
xLabel	Specify label for the x axis, defaults to "Time \(\text{ms}\)"
lineWidthGazeProp	Specify line width for gaze proportion, defaults to 1.
lineWidthDataPointProp	Specify line width for data point numbers, defaults to 1.
lineWidthOverallMean	Specify line width for overall mean, defaults to 1.
lineTypeOverallMean	Specify line type for overall mean, defaults to "dotted".
alphaOverallMean	Specify alpha for overall mean, defaults to 0.5.
shapeCodeOverallMean	Specify the shape of the overall mean points when showOverallMean = "Point", defaults to 1 \(\text{circle}\).
tickSizeOverallMean	Optional to specify where to plot the overall means when showOverallMean = "Point", defaults to NULL.

Value

a plot handle that visualizes the data from PrepareMLETData_HMLET or the list from Permutation-Test_HMLET.

Preprocess_AOIExtraction_HMLET

AOI Extraction

Description

AOIs can be computed in rectangular or elliptic forms. This function uses gaze coordinates to categorize them into specified AOIs

Usage

```
Preprocess_AOIExtraction_HMLET(
  data,
  GazeX,
  GazeY,
  AOI_Names,
  AOIs_Center_X,
  AOIs_Center_Y,
  AOIs_Widths,
  AOIs_Heights,
  ContentLabel = "Content",
  EllipticAOI = FALSE
)
```

Arguments

data	long format dataframe containing temporal data.
GazeX	string for column name that includes X coordinate of gazepoints
GazeY	string for column name that includes Y coordinate of gazepoints
AOI_Names	List of Strings for AOIs names
AOIs_Center_X	List of numerical values for the horizontal coordinate of the center of all AOIs or list of strings indicating the column names in the input data that have these values
AOIs_Center_Y	List of numerical values for the vertical coordinate of the center of all AOIs or list of strings indicating the column names in the input data that have these values
ContentLabel	Label for gaze points that are outside all of the AOIs, defaults to "Content"
EllipticAOI	Binary value to identify whether elliptic AOIs should be used instead of rectangular ones, defaults to FALSE
AOIs_Radius_X	List of numerical values for width size/horizontal diameter of all AOIs or list of strings indicating the column names in the input data that have these values
AOIs_Radius_Y	List of numerical values for height size/vertical diameter of all AOIs or list of strings indicating the column names in the input data that have these values.

Details

The output is the same dataframe as input with one new column: AOIs: Includes the name of the AOI at each time point.

Preprocess_CheckData_HMLET

Check Data for Critical Issues

Description

Raw eye-tracking data comes with time stamps of each sample and Gaze Coordinates. These time stamps has to have consistent and unique time interval across all datapoints. Gaze Coordinates may have missing values due to blinks, head movement, etc. This function reports issues about inconsistent sampling intervals and missing Gaze coordinates.

Usage

```
Preprocess_CheckData_HMLET(
  data,
  ID = "ID",
  trial = "trial",
  timePoint = "timePoint",
  GazeX,
  GazeY,
  groupingColumns = NULL,
  missRateCap = 50,
  shortTrialsThreshold = 2
)
```

Arguments

data	long format dataframe containing temporal data.
ID	string for column name that represents IDs within data frame, defaults to "ID".
trial	string for column name that identify unique trials within data frame. Can be numerical or categorical.
timePoint	string for column name that represents time stamps of each sample.
GazeX	string for column name that includes X coordinate of gaze points
GazeY	string for column name that includes Y coordinate of gaze points
groupingColumns	optional array of strings for column names of variables to be grouped for the final miss Rate Summary report, defaults to NULL.
missRateCap	Specify the percentage of missing gaze points cap to be counted as high Miss Rate
shortTrialsThreshold	Specify the minimum number of samples that should be available within a trial to be counted as a valid trial. defaults to 2

Details

This function returns a dataframe with summary of missing gaze points. The summary dataframe has 4 columns: ID (with the original column name): indicate participant's ID; nTrials: number of trials; nTrialsMoreThanCap: number of trials in which, more than missRateCap (defaults to 50) averageMissRate: Average percentage of missing gaze points across all trials of each participant.

Data Cleaning Note: You may want to remove trials with more than missRateCap (defaults to 50) participants that most of their trials has high rate of missing gazepoints.

Description

Raw eye-tracking data comes with time stamps of each sample. These time stamps has to have consistent and unique time interval across all datapoints. If across participants there are different sampling intervals this function downsample all datapoint to the biggest sampling interval. This function also relabel all time stamps with a consistent time stamps to be comparable across trials and participants. If there is a gap between samples of a trial this function fill in the gap with NA values for Gaze coordinates. difference between each two consecutive samples within a trial. Note that your dataset has to have a unique consistent time interval between all samples. In case, several time intervals is estimated for your dataset it use "Preprocess_FixSampling_HMLET" function and pass the highest sampling interval to fix the inconsistency.

Usage

```
Preprocess_FixSampling_HMLET(
  data,
  ID = "ID",
  trial,
  timePoint,
  GazeX,
  GazeY,
  samplingInterval = NULL,
  fillGenratedRows = T,
  ignoreColumns = NULL,
  removeShortTrials = T,
  shortTrialsThreshold = 2
)
```

Arguments

data	long format dataframe containing temporal data.
ID	string for column name that represents IDs within data frame, defaults to "ID".
trial	string for column name that identify unique trials within data frame. Can be numerical or categorical.
timePoint	string for column name that represents time stamps of each sample.
GazeX	string for column name that includes X coordinate of gazepoints
GazeY	string for column name that includes Y coordinate of gazepoints
samplingInterval	Specify the expected sampling interval
fillGenratedRows	Optional flag to specify whether the new rows generated to fill temporal jumps should be filled or not. This function uses a 'down-up' order (each value filled with its preceding value; otherwise, with its following value) to fill these columns. Defaults to TRUE.
ignoreColumns	Optional string(s) to specify columns that shouldn't be filled, defaults to NULL.
removeShortTrials	Optional flag to specify whether short trials has to be removed or not, defaults to T.
shortTrialsThreshold	Specify the minimum number of samples that should be available within a trial to be counted as a valid trial. defaults to 2

Details

The output is the same dataframe as input with three new columns: "time": includes new time stamps restarting to (0+sampling interval) in each trial "sampleIdx": includes index for each sample restarting to 1 in each trial "interval": includes the unique sampling interval which is equal to the "time" Note: If your dataframe has columns with same names make sure to rename them before using this function.

Preprocess_Interpolate_HMLET

Interpolate missing Gaze Points

Description

Gaze Coordinates may have missing values due to blinks, head movement, etc. This function fill the NA Gaze points with linear interpolation. You can specify how many consecutive NAs should be interpolated by passing maxNAToFill parameter.

Usage

```
Preprocess_Interpolate_HMLET(
  data,
  ID = "ID",
  trial,
  timePoint,
  GazeX,
  GazeY,
  maxNAToFill = Inf
)
```

Arguments

data	long format dataframe containing temporal data.
ID	string for column name that represents IDs within data frame, defaults to "ID".
trial	string for column name that identify unique trials within data frame. Can be numerical or categorical.
timePoint	string for column name that represents time stamps of each sample.
GazeX	string for column name that includes X coordinate of gazepoints
GazeY	string for column name that includes Y coordinate of gazepoints
maxNAToFill	numeric value to denote the max number of consecutive NAs that should be interpolated, defaults to Inf

Details

The output is the same dataframe as input with three new columns: "GazeX_Interpolated": includes all original GazeX coordinates and interpolated values "GazeY_Interpolated": includes all original GazeY coordinates and interpolated values "Interpolated": is a flag that is 1 for samples that are interpolated and 0 otherwise. Note: If your dataframe has columns with same names make sure to rename them before using this function.

RelativeViewingProportionLabels_HMLET

Label trials whether participant was disproportionately or equally looking toward items

Description

Label trials whether participant was disproportionately or equally looking toward items

Usage

```
RelativeViewingProportionLabels_HMLET(
  data,
  ID = "ID",
  trial,
  AOIName,
  comparingAOI = c("Target", "Lure"),
  targetAOI = "Target",
  dispropotionCriterion = 10
)
```

Arguments

data	data frame with temporal data for permutation tests routine.
ID	optional string for column name that represents IDs within data frame, defaults to "ID".
trial	string for column name that represents trials within data frame.
AOIName	string for column name which includes name of the AOI that the subject is looking at each timePoint.
comparingAOI	names of the two AOIs that are their looking proportion should be compared.
targetAOI	names of the two AOIs that are their looking proportion should be compared.
dispropotionCriterion	optional string for column name to focus on specific AOI, defaults to NULL.
testName	Name of this data – can be used as condition name or test names to compare permutation test results between different tests/conditions later

Value

data frame as original data with an extracolumn with labels

RemoveIncompleteTimePoints_HMLET

Remove Time points with missing conditions. This function is currently only applicable for within participant manipulations.

Description

Remove Time points with missing conditions. This function is currently only applicable for within participant manipulations.

Usage

```
RemoveIncompleteTimePoints_HMLET(data)
```

Arguments

data dataframe containing temporal data.

Value

data with no missing conditions.

SubjectLevelPermutationTestWithin_HMLET

Conduct subject level permutation tests between trials.

Description

Conduct subject level permutation tests between trials.

Usage

```
SubjectLevelPermutationTestWithin_HMLET(
  data,
  samples = 2000,
  paired = T,
  threshold_t = NULL
)
```

Arguments

data data frame with temporal data for permutation tests routine, output from PermutationTestDataPrep_HMLET.R.

samples optional number of resampling data, how many permutations needed.

paired optional boolean to control T-tests. True is for statistical comparisons in paired mode, False for a two-sample T-Test; defaults to True.

threshold_t optional probability threshold for statistical comparison computed based on number of subjects. alpha = 0.025.

Value

returns t value distribution presented in a data frame.

TrialLevelPermutationTestWithin_HMLET

Conduct trial level permutation tests between trials.

Description

Conduct trial level permutation tests between trials.

Usage

```
TrialLevelPermutationTestWithin_HMLET(
  data,
  samples = 2000,
  paired = T,
  threshold_t = NULL
)
```

Arguments

data	data frame with temporal data for permutation tests routine, output from PermutationTestDataPrep_HMLET.R.
samples	optional number of resampling data, how many permutations needed.
paired	optional boolean to control T-tests. True is for statistical comparisons in paired mode, False for a two-sample T-Test; defaults to True.
threshold_t	optional probability threshold for statistical comparison computed based on number of trials. alpha = 0.025.

Value

returns t value distribution presented in a data frame.

UniquePermutations_HMLET

Generate unique random permutations.

Description

Generate unique random permutations.

Usage

```
UniquePermutations_HMLET(listInput, uniqueLabels, n = 1)
```

Arguments

listInput	Arbitrary list.
uniqueLabels	All possible Labels.
n	An integer number indicating the number of permutations.

Index

AddClusterInfoToData_HMLET, [2](#)

ClusterStats_HMLET, [3](#)

ComputeSamplingInterval_HMLET, [3](#)

ComputeSubjectLevelPermLabels_HMLET, [4](#)

ComputeTValues_HMLET, [4](#)

CreateTimeBinData_HMLET, [5](#)

ExportDataForGUI_HMLET, [6](#)

FilterClusterTimePoints_HMLET, [7](#)

FindClusters_HMLET, [7](#)

GenerateRandomData_HMLET, [8](#)

PermutationTest_HMLET, [10](#)

PermutationTestDataPrep_HMLET, [9](#)

PermutationTestSingleParticipant_HMLET, [10](#)

PlotNullDistribution_HMLET, [11](#)

PlotTimeSeries_HMLET, [12](#)

Preprocess_AOIExtraction_HMLET, [13](#)

Preprocess_CheckData_HMLET, [14](#)

Preprocess_FixSampling_HMLET, [15](#)

Preprocess_Interpolate_HMLET, [17](#)

RelativeViewingProportionLabels_HMLET, [18](#)

RemoveIncompleteTimePoints_HMLET, [19](#)

SubjectLevelPermutationTestWithin_HMLET, [19](#)

TrialLevelPermutationTestWithin_HMLET, [20](#)

UniquePermutations_HMLET, [20](#)