**Package ‘approxMapR’**

**10/21/19**

**Note:** This document is not an exhaustive list of functions available within approxMap (as seen in ls(“package:approxmapR”)), but summarizes the used functions needed for creating the summary clusters and corresponding output.

**Version** N/A

**Title** Create Charts Showing Sequences within Longitudinal Data

**Description** Approxmap in an algorithm used for exploratory data analysis of sequential data. When we have longitudinal data and we want to find out the underlying patterns, we use approxmap. ‘approxmapR’ aims to provide a consistent and tidy api for using the algorithm in R.

**Imports**

**Suggests**

**License**

**Repository** GitHub

**Date/Publication** N/A

aggregate\_sequences

**Description**

Allows to go from dates to aggregations (called a period) in the package.

**Usage**

aggregate\_sequences(unaggregated\_data, format = "%m-%d-%Y", calendar = FALSE, unit = "week", n\_units = 1, anchor\_table = NA, anchor\_vector = NA, base\_date = NA, occurence = min, multiset = FALSE, include\_date = FALSE, summary\_stats = TRUE)

**Arguments**

unaggregated\_data unaggregated dataset

format date format

unit unit of aggregation; day, month, week

n\_units number of units to aggregate (ie, if the unit is “week” and

n\_units is 4, 4 weeks becomes the unit of aggregation

calendar

anchor\_table beta

anchor\_vector beta

base\_base beta

occurance beta

multiset beta

include\_date beta

summary\_stats Logical controlling printing of summary statistics regarding

aggregation.

**Value**

Aggregated data frame that has sequence id, itemset (period) and event (item)

**Examples**

mvad %>%

aggregate\_sequences(format = "%Y-%m-%d", unit = "month", n\_units = 6)

pre\_aggregated

**Description**

Used whenever the df to be analyzed in preaggregated i.e. instead of dates, we have periods (corresponding to itemsets)

**Usage**

pre\_aggregated(df, summary\_stats = TRUE)

**Arguments**

df pre-aggregated data frame

**Value**

Returns a data frame that has the properly class data frame.

**Examples**

pre\_agg\_demo %>% pre\_aggregated()

cluster\_knn

**Description**

Used to calculate the clusters in the data using the k-nearest neighbor algorithm.

**Usage**

cluster\_knn(df\_aggregated, k, use\_cache = TRUE)

**Arguments**

df\_aggregated pre-aggregated data frame

k number of nearest neighbors to consider while clustering

caching use cache to store results

**Value**

Returns a data frame with clusters and weights.

**Examples**

cluster %>% cluster\_knn(k=1)

filter\_pattern

**Description**

Calculates weighted sequence for each cluster and formats them with the required threshold.

**Usage**

filter\_pattern(threshold, pattern\_name = “consensus”)

**Arguments**

threshold required weighting threshold for all sequences

pattern\_name name of pattern

**Value**

Data frame containing pattern variation.

**Examples**

temp %>% filter\_pattern(threshold = 0.3, pattern\_name = “variation”) %>%

filter\_pattern(threshold = 0.4, pattern\_name = “consensus”)

format\_sequence

**Description**

A generic function that is used to display formatted results.

**Usage**

format\_sequence(x, ...)

**Arguments**

x A sequence, w\_sequence, w\_sequence\_list, w\_sequence\_pattern, or a w\_sequence\_dataframe object.

..... Additional parameters to the function that is invoked

**Value**

Formatted output from the sequence analysis.

**Examples**

results %>% format\_sequence()

generate\_reports

**Description**

Formats the output of the sequence analysis into two folders.

**Usage**

generate\_reports(w\_sequence\_dataframe, html\_format = TRUE, output\_directory = “~”)

**Arguments**

w\_sequence\_dataframe data frame formatted from the filter\_pattern

html\_format outputs html formats

output\_directory path for output files

**Value**

Folders “private” and “public” each containing a \*.csv file and two formatted \*.html files each containing a summary of the sequences.

**Examples**

temp\_f %>% generate\_reports(output\_directory = “//Users/”)