Package ‘LagSequential’

May 16, 2019

Type Package

Title Lag-Sequential Categorical Data Analysis

Version 0.1.1

Date 2019-05-15

Author Zakary A. Draper & Brian P. O'Connor

Maintainer Brian P. O'Connor <brian.oconnor@ubc.ca>

Depends R(>= 1.9.0)

LazyLoad yes

LazyData yes

Description Lag-sequential analysis is a method of assessing of patterns (what tends to follow what?) in sequences of codes. The codes are typically for discrete behaviors or states. The functions in this package read a stream of codes, or a frequency transition matrix, and produce a variety of lag sequential statistics, including transitional frequencies, expected transitional frequencies, transitional probabilities, z values, adjusted residuals, Yule's Q values, likelihood ratio tests of stationarity across time and homogeneity across groups or segments, transformed kappas for unidirectional dependence, bidirectional dependence, parallel and nonparallel dominance, and significance levels based on both parametric and randomization tests. The methods are described in Bakeman & Quera (2011) <doi:10.1017/CBO9781139017343>, O'Connor (1999) <doi:10.3758/BF03200753>, Wampold & Margolin (1982) <doi:10.1037/0033-2909.92.3.755>, and Wampold (1995, ISBN:0-89391-919-5).

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2019-05-16 15:50:11 UTC

R topics documented:

LagSequential-package ......................................................... 2
Description

This package provides functions for conducting lag sequential analyses of categorical data. The functions are R versions of the programs provided by O’Connor (1999). The functions read a stream of codes, or a frequency transition matrix, and produce a variety of lag sequential statistics, including transitional frequencies, expected transitional frequencies, transitional probabilities, z values, adjusted residuals, Yule’s Q values, likelihood ratio tests of stationarity across time and homogeneity across groups or segments, transformed kappas for unidirectional dependence, bidirectional dependence, parallel and nonparallel dominance, and significance levels based on both parametric and randomization tests.

When data is a frequency transition matrix, the code value that preceded the first code in the sequence, and the code value that followed the final code value, are usually unknown/unavailable. This missing information may cause slight inaccuracies in some of the provided statistics, most likely only at the second decimal place. The inaccuracies will be negligible in longer data sequences.

References

**Description**

Tests for bidirectional dependence between pairs of lag sequential transitions.

**Usage**

```r
bidirectional(data, labels = NULL, lag = 1, adjacent = TRUE,
               tailed = 1, permtest = FALSE, nperms = 10)
```

**Arguments**

- `data` A one-column dataframe, or a vector of code sequences, or a square frequency transition matrix. If data is not a frequency transition matrix, then data must be either (a) a series of string (non-numeric) code values, or (b) a series of integer codes with values ranging from "1" to whatever value the user specifies in the "ncodes" argument. There should be no code values with zero frequencies. Missing values are not permitted.

- `labels` Optional argument for providing labels to the code values. Accepts a list of string variables. If unspecified, codes will be labeled "Code1", "Code2", etc.

- `lag` The lag number for the analyses.

- `adjacent` Can adjacent values be coded the same? Enter "FALSE" if adjacent events can never be the same. Enter "TRUE" if adjacent events can always be the same.

- `tailed` Specify whether significance tests are one-tailed or two-tailed. Options are "1" or "2".

- `permtest` Do you want to run permutation tests of significance? Options are "FALSE" for no, or "TRUE" for yes. Warning: these computations can be time consuming.

- `nperms` The number of permutations per block.

**Details**

This function tests the bidirectional dependence of behaviors \(i\) to \(j\), and \(j\) to \(i\), an additive sequential pattern described by Wampold and Margolin (1982) and Wampold (1989, 1992). Bidirectional dependence suggests a reciprocal effect of behaviors. That is, behavior \(i\) influences behavior \(j\) and behavior \(j\) influences behavior \(i\). For example, if behavior \(i\) is a husband’s positive behavior, and behavior \(j\) is his wife’s positive behavior, a test of bidirectional dependence asks whether the husband reciprocates the wife’s positive behavior, and the wife reciprocates the husband’s positive behavior (See Margolin and Wampold, 1982). Bidirectional dependence is sometimes called a "circuit".
Value

A list with the following elements:

- freqs: The transitional frequency matrix
- bifreqs: The bidirectional frequencies
- expbifreqs: The expected bidirectional frequencies
- kappas: The bidirectional kappas
- z: The z values for the kappas
- pk: The p values (significance levels) for the kappas

Author(s)

Zakary A. Draper & Brian P. O’Connor

References


Examples

```r
bidirectional(data_Wampold_1982, 
  labels = c('HPos','HNeg','WPos','WNeg'), 
  permtest = TRUE, nperms = 100)
```

Description

A column vector of simulated data with 393 observations in 3 segments (which could, e.g., be groups or dyads).
**data_seqgroups_strings**

**Details**
A column vector of numeric data with 393 observations in 3 segments (which could, e.g., be groups or dyads). The beginning of each segment is indicated by a number greater than 999. The data set is provided as trial data for the seqgroups function. It is a numeric version of the data in data_seqgroups_strings.

**Examples**

```r
table(data_seqgroups_numeric)
```

**data_seqgroups_strings**

**Description**
A column vector of simulated data with 393 observations in 3 segments (which could, e.g., be groups or dyads).

**Details**
A column vector of string data with 393 observations in 3 segments (which could, e.g., be groups or dyads). The beginning of each segment is indicated by the word "segment". The data set is provided as trial data for the seqgroups function. It is a string/character version of the data in data_seqgroups_numeric.

**Examples**

```r
table(data_seqgroups_strings)
```

**data_sequential**

**Description**
A column vector of trial data for sequential analyses.

**Details**
A column vector with 122 observations (codes). The data are provided as trial data for the sequential, bidirectional, twocells, paradom, and nonparadom functions.

**Examples**

```r
table(data_sequential)
```
data_Wampold_1982  data_Wampold_1982

Description
A vector of code sequences that mimic the frequency transition matrix and the statistical results reported in Wampold & Margolin (1982).

Details
A column vector of 200 sequential codes. The data are provided as trial data for the paradom and nonparadom functions.

References

Examples
table(data_Wampold_1982)


data_Wampold_1984  data_Wampold_1984

Description
A vector of code sequences that mimic the frequency transition matrix and the statistical results reported in Wampold (1984).

Details
A column vector of 200 sequential codes. The data are provided as trial data for the paradom and nonparadom functions.

References

Examples
table(data_Wampold_1984)
Description

Tests for nonparallel dominance, a form of asymmetry in predictability, between i to j and k to L (Wampold, 1984, 1989, 1992, 1995).

Usage

```r
nonparadom(data, i, j, k, L, labels = NULL, lag = 1, adjacent = TRUE,
            tailed = 1, permtest = FALSE, nperms = 10)
```

Arguments

- **data**: A one-column dataframe, or a vector of code sequences, or a square frequency transition matrix. If data is not a frequency transition matrix, then data must be either (a) a series of string (non-numeric) code values, or (b) a series of integer codes with values ranging from "1" to whatever value the user specifies in the "ncodes" argument. There should be no code values with zero frequencies. Missing values are not permitted.
- **i**: Code value for i.
- **j**: Code value for j.
- **k**: Code value for k.
- **L**: Code value for L.
- **labels**: Optional argument for providing labels to the code values. Accepts a list of string variables. If unspecified, codes will be labeled "Code1", "Code2", etc.
- **lag**: The lag number for the analyses.
- **adjacent**: Can adjacent values be coded the same? Options are "TRUE" for yes or "FALSE" for no.
- **tailed**: Specify whether significance tests are one-tailed or two-tailed. Options are "1" or "2".
- **permtest**: Do you want to run permutation tests of significance? Options are "FALSE" for no, or "TRUE" for yes. Warning: these computations can be time consuming.
- **nperms**: The number of permutations per block.

Details

Tests for nonparallel dominance or asymmetry in predictability, which is the difference in predictability between i to j and k to L, as described by Wampold (1984, 1989, 1992, 1995). Parallel dominance (another function in this package) is the difference in predictability between i to j and j to i. In parallel dominance the i and j values across the two pairs of codes are the same. In nonparallel dominance, the i and j values across the two pairs of codes may vary, i.e., they do not have to be the same.
**Value**

Displays the transitional frequency matrix, expected frequencies, expected and observed nonparallel dominance frequencies, kappas, the z values for the kappas, and the significance levels.

Returns a list with the following elements:

- `freqs` The transitional frequency matrix
- `expfreqs` The expected frequencies
- `npdomfreqs` The nonparallel dominance frequencies
- `expnpdomfreqs` The expected nonparallel dominance frequencies
- `domtypes` There are 4 sequential dominance case types described by Wampold (1989). These cases describe the direction of the effect for i on j and j on i. The four cases are: (1) i increases j, and j increases i, (2) i decreases j, and j decreases i, (3) i increases j, and j decreases i, and (4) i decreases j, and j increases i. Each cell of this matrix indicates the case that applies to the transition indicated by the cell.
- `kappas` The nonparallel dominance kappas
- `z` The z values for the kappas
- `pk` The p-values for the kappas

**Author(s)**

Zakary A. Draper & Brian P. O’Connor

**References**


Examples

```r
nonparadom(data_Wampold_1984, i = 6, j = 1, k = 3, L = 4,
  labels = c('HPos','HNeu','HNeg','WPos','WNeu','WNeg'),
  permtest = TRUE, nperms = 1000)
```

Description

Tests for parallel dominance in lag sequential data.

Usage

```r
paradom(data, labels = NULL, lag = 1, adjacent = TRUE,
  tailed = 1, permtest = FALSE, nperms = 10)
```

Arguments

data  A one-column dataframe, or a vector of code sequences, or a square frequency transition matrix. If data is not a frequency transition matrix, then data must be either (a) a series of string (non-numeric) code values, or (b) a series of integer codes with values ranging from "1" to whatever value the user specifies in the "ncodes" argument. There should be no code values with zero frequencies. Missing values are not permitted.

labels Optional argument for providing labels to the code values. Accepts a list of string variables. If unspecified, codes will be labeled "Code1", "Code2", etc.

lag The lag number for the analyses.

adjacent Can adjacent values be coded the same? Options are "TRUE" for yes or "FALSE" for no.

tailed Specify whether significance tests are one-tailed or two-tailed. Options are "1" or "2".

permtest Do you want to run permutation tests of significance? Options are "FALSE" for no, or "TRUE" for yes. Warning: these computations can be time consuming.

nperms The number of permutations per block.

Details

Tests for parallel dominance or asymmetry in predictability, which is the difference in predictability between i to j and j to i (e.g., whether B’s behavior is more predictable from A’s behavior than vice versa), as described by Wampold (1984, 1989, 1992, 1995).
Value

Displays the transitional frequency matrix and matrices of expected frequencies, expected and observed parallel dominance frequencies, parallel dominance kappas, z values for the kappas, and significance levels. There are four possible cases, or kinds, of parallel dominance (see Wampold 1989, 1992, 1995), and the function returns a matrix indicating the kind of case for each cell in the transitional frequency matrix.

Returns a list with the following elements:

- freqs: The transitional frequency matrix
- expfreqs: The expected frequencies
- domfreqs: The parallel dominance frequencies
- expdomfreqs: The expected parallel dominance frequencies
- domtypes: There are 4 sequential dominance case types described by Wampold (1989). These cases describe the direction of the effect for i on j and j on i. The four cases are: (1) i increases j, and j increases i, (2) i decreases j, and j decreases i, (3) i increases j, and j decreases i, and (4) i decreases j, and j increases i. Each cell of this matrix indicates the case that applies to the transition indicated by the cell.
- kappas: The parallel dominance kappas
- z: The z values for the kappas
- pk: The p-values for the kappas

Author(s)

Zakary A. Draper & Brian P. O’Connor

References


Examples

```r
paradom(data_Wampold_1984,
  labels = c('HPos', 'HNeu', 'HNeg', 'WPos', 'WNeu', 'WNeg'),
  permtest = TRUE, nperms = 1000)
```

Description

Computes a variety of sequential analysis statistics for data that are in segments (e.g., for multiple dyads or groups).

Usage

```r
seqgroups(alldata, labels = NULL, lag = 1, adjacent = TRUE,
  onezero = NULL, tailed = 2, test = "homogeneity",
  output = "all")
```

Arguments

- **alldata**: A one-column dataframe, or a vector of code sequences, which can be numeric or strings. Missing values are not permitted.

  If alldata is numeric, then the integers must range from "1" to the total number of possible code values (which is not the total number of codes in a sequence), and a number greater than 999 must be used in alldata to separate the code sequences for different groups/dyads. See "data_seqgroups" for an example.

  If alldata consists of strings/characters, then the word "segment" must be used in alldata to separate the code sequences for different groups/dyads. See "data_seqgroups" for an example.

- **labels**: Optional argument for providing labels to the code values. Accepts a list of string variables. If unspecified, codes will be labeled "Code1", "Code2", etc.

- **lag**: The lag number for the analyses.

- **adjacent**: Can adjacent values be coded the same? Enter "FALSE" if adjacent events can never be the same. Enter "TRUE" if adjacent events can always be the same. Enter "TRUE" if some adjacent events can, and others cannot, be the same; then enter the appropriate onezero matrix for your data.

- **onezero**: Optional argument for specifying the one-zero matrix for the data. Useful when some adjacent events can, and others cannot, be the same. Accepts a square matrix of ones and zeros with length ncodes. A "1" indicates that the expected frequency for a given cell is to be estimated, whereas a "0" indicates that the expected frequency for the cell should NOT be estimated, typically because it is a structural zero (codes that cannot follow one another). By default, the matrix
that is created by the above commands has zeros along the main diagonal, and
ones everywhere else, which will be appropriate for most data sets. However,
if your data happen to involve structural zeros that occur in cells other than the
cells along the main diagonal, then you must create a onezero matrix with ones
and zeros that is appropriate for your data.

tailed Specify whether significance tests are one-tailed or two-tailed. Options are "1" or "2".

test Specify whether to run tests for homogeneity of homogeneity or stationarity. Homogeneity should be tested when groups in the data are actually different
groups, whereas stationarity should be tested when groups in the data are seg-
ments of a single stream of observations. Options are "homogeneity" or "sta-
tionarity".

output Specify the desired output. Options are "pooled" for pooled data only, or "all"
for all data sets.

Details
Computes a variety of sequential analysis statistics for data that are in segments (e.g., for multiple
dyads or groups. This is the same as the "sequential" function provided in this package, but allows
for the data to be segmented. Sequential statistics are calculated for each segment, as well as for the
data pooled across all segments.

Value
For each of the groups or segments and for the pooled data, displays the transitional frequency
matrix, expected frequencies, transitional probabilities, adjusted residuals and significance levels,
Yule's Q values, transformed Kappas (Wampold, 1989, 1992, 1995), z values for the kappas, and
significance levels.

Returns a list with the following elements:

freqs The transitional frequency matrix
expfreqs The expected frequencies
probs The transitional probabilities
chi The overall chi-square test of the difference between the observed and expected
transitional frequencies
adjres The adjusted residuals
p The statistical significance levels
YulesQ Yule's Q values, indicating the strength of the relationships between the an-
tecedent and the consequence transitions
kappas The nonparallel dominance kappas
z The z values for the kappas
pk The p-values for the kappas
output The requested output data
Author(s)
Zakary A. Draper & Brian P. O’Connor

References


Examples
seqgroupsHdata_seqgroups_stringsI

Description
Computes a variety of lag sequential analysis statistics for one series of codes.

Usage
sequential(data, labels = NULL, lag = 1, adjacent = TRUE, onezero = NULL, tailed = 2, permtest = FALSE, nperms = 10)

Arguments
data A one-column dataframe, or a vector of code sequences, or a square frequency transition matrix. If data is not a frequency transition matrix, then data must be either (a) a series of string (non-numeric) code values, or (b) a series of integer codes with values ranging from "1" to whatever value the user specifies in the “ncodes” argument. There should be no code values with zero frequencies. Missing values are not permitted.

labels Optional argument for providing labels to the code values. Accepts a list of string variables. If unspecified, codes will be labeled "Code1", "Code2", etc.

lag The lag number for the analyses.
adjacent  Can adjacent values be coded the same? Enter "FALSE" if adjacent events can never be the same. Enter "TRUE" if any adjacent events can be the same. If some adjacent events can, and others cannot, be the same, then enter the appropriate onezero matrix for your data using the onezero argument.

onezero  Optional argument for specifying the one-zero matrix for the data. Accepts a square matrix of ones and zeros with length ncodes. A “1” indicates that the expected frequency for a given cell is to be estimated, whereas a “0” indicates that the expected frequency for the cell should NOT be estimated, typically because it is a structural zero (codes that cannot follow one another). By default, the matrix that is created by the above commands has zeros along the main diagonal, and ones everywhere else, which will be appropriate for most data sets. However, if your data happen to involve structural zeros that occur in cells other than the cells along the main diagonal, then you must create a onezero matrix with ones and zeros that is appropriate for your data.

tailed  Specify whether significance tests are one-tailed or two-tailed. Options are "1" or "2".

permtest  Do you want to run permutation tests of significance? Options are "FALSE" for no, or "TRUE" for yes. Warning: these computations can be time consuming.

nperms  The number of permutations per block.

Details

Tests unidirectional dependence of states (codes). Specifically, this function tests the hypothesis that state i (the antecedent) follows state j (the consequence) with a greater than chance probability. Computes a variety of statistics including two indices of effect size with corresponding significance tests. The larger the effect the more like the consequence is to follow the antecedent.

Value

Displays the transitional frequency matrix, expected frequencies, transitional probabilities, adjusted residuals and significance levels, Yule’s Q values, transformed Kappas (Wampold, 1989, 1992, 1995), z values for the kappas, and significance levels.

Returns a list with the following elements:

freqs  The transitional frequency matrix
expfreqs  The expected frequencies
probs  The transitional probabilities
chi  The overall chi-square test of the difference between the observed and expected transitional frequencies
adjres  The adjusted residuals
p  The statistical significance levels
YulesQ  Yule’s Q values, indicating the strength of the relationships between the antecedent and the consequence transitions
kappas  The nonparallel dominance kappas
z  The z values for the kappas
pk  The p-values for the kappas
**Author(s)**

Zakary A. Draper & Brian P. O'Connor

**References**


**Examples**

```r
# data is a one-column dataframe of code sequences
sequential(data_sequential, permtest = TRUE, nperms = 100)

# in this case, data is the frequency transition matrix from
# interaction. In G. R. Patterson (Ed.), *Family social interaction: Content and methodology
freqs <- t(matrix(c(0, 0, 0, 0, 2, 2,
0, 0, 2, 5, 60, 20,
0, 9, 2, 1, 3, 0,
0, 3, 0, 1, 5, 0,
3, 54, 6, 2, 24, 8,
1, 24, 2, 1, 3, 12), 6, 6))

sequential(freqs, adjacent = 1,
    labels = c('H+', 'Ho', 'H-', 'W+', 'Wo', 'W-'))

# Data from p 159 of Bakeman & Quera (2011), Sequential Analysis and Observational
data_BQ2011 <- t(matrix(c(2,1,4,3,3,4,3,4,2,1,4,4,5,4,1,3,4,5,3,2,2,1,4,1,2,
5,2,1,2,3,1,4,4,1,4,1,3,3,1,5,2,1,1,3,1,4,1,2,
3,3,4,5,2,3,5,2,5,4,4,2,3,1,5,5,2,2,1,3,3,3,3))

sequential(data_BQ2011, labels=c('Chat','Write','Read','Ask','Attentive'),
    permtest = TRUE, nperms = 1000, tailed = 1)
```
**Description**

Simultaneously tests the unidirectional dependence of $i$ to $j$, and the unidirectional dependence of $k$ to $L$, an additive pattern described by Wampold and Margolin (1982) and Wampold (1989, 1992).

**Usage**

twocells(data, i, j, k, L, labels = NULL, lag = 1, adjacent = TRUE, tailed = 1, permtest = FALSE, nperms = 10)

**Arguments**

data A one-column dataframe, or a vector of code sequences, or a square frequency transition matrix. If data is not a frequency transition matrix, then data must be either (a) a series of string (non-numeric) code values, or (b) a series of integer codes with values ranging from "1" to whatever value the user specifies in the "ncodes" argument. There should be no code values with zero frequencies. Missing values are not permitted.

i Code value for $i$.

j Code value for $j$.

k Code value for $k$.

L Code value for $L$.

labels Optional argument for providing labels to the code values. Accepts a list of string variables. If unspecified, codes will be labeled "Code1", "Code2", etc.

lag The lag number for the analyses.

adjacent Can adjacent values be coded the same? Options are "TRUE" for yes, and "FALSE" for no.

tailed Specify whether significance tests are one-tailed or two-tailed. Options are "1" or "2".

permtest Do you want to run permutation tests of significance? Options are "FALSE" for no, or "TRUE" for yes. Warning: these computations can be time consuming.

nperms The number of permutations per block.

**Details**

This function simultaneously tests the unidirectional dependence of $i$ to $j$ and the unidirectional dependence of $k$ to $L$. The user specifies the code values used for $i$, $j$, $k$, and $L$ in the analyses. For example, Wampold and Margolin (1982) described a situation wherein a spouse responds to negative behaviors with something other than a negative behavior.
Value

Displays the transitional frequency matrix, observed and expected values for the two cell test, kappa, the z value for kappa, and the significance level.

Returns a list with the following elements:

- freqs: The transitional frequency matrix
- expfreqs: The expected frequencies
- twocellfreq: The observed number of transitions from i to j and from k to L.
- kappa: The twocells kappa
- z: The z value for the kappa
- pk: The p-value for the kappa

Author(s)

Zakary A. Draper & Brian P. O’Connor

References


Examples

twocells(data_Wampold_1982, i = 6, j = 1, k = 3, L = 4,
labels = c('HPos','HNeu','HNeg','WPos','WNeu','WNeg'),
permtest = TRUE, nperms = 100)
Index

*Topic Sequential Analysis
  bidirectional, 3
  nonparadom, 7
  paradom, 9
  seqgroups, 11
  sequential, 13
  twocells, 16
*Topic dataset
  data_seqgroups_numeric, 4
  data_seqgroups_strings, 5
  data_sequential, 5
  data_Wampold_1982, 6
  data_Wampold_1984, 6

bidirectional, 3

data_seqgroups_numeric, 4
data_seqgroups_strings, 5
data_sequential, 5
data_Wampold_1982, 6
data_Wampold_1984, 6

LagSequential-package, 2

nonparadom, 7

paradom, 9

seqgroups, 11
sequential, 13

twocells, 16