Package ‘NCA’

June 11, 2020

Type Package

Title Necessary Condition Analysis

Version 3.0.3

Date 2020-06-11

Author Jan Dul <jdul@rsm.nl>

Maintainer Govert Buijs <buijs@rsm.nl>

Description Performs a Necessary Condition Analysis (NCA). (Dul, J. 2016. Necessary Condition Analysis (NCA). "Logic and Methodology of 'Necessary but not Sufficient' causality." Organizational Research Methods 19(1), 10-52, <http://journals.sagepub.com/doi/abs/10.1177/1094428115584005>).
NCA identifies necessary (but not sufficient) conditions in datasets, where x causes (e.g. precedes) y. Instead of drawing a regression line "through the middle of the data" in an xy-plot, NCA draws the ceiling line. The ceiling line y = f(x) separates the area with observations from the area without observations.
(Nearly) all observations are below the ceiling line: y <= f(x). The empty zone is in the upper left hand corner of the xy-plot (with the convention that the x-axis is "horizontal" and the y-axis is "vertical" and that values increase "upwards" and "to the right"). The ceiling line is a (piecewise) linear non-decreasing line: a linear step function or a straight line. It indicates which level of x (e.g. an effort or input) is necessary but not sufficient for a (desired) level of y (e.g. good performance or output). A quick start guide for using this package can be found here: <http://repub.eur.nl/pub/78323/> or <https://ssrn.com/abstract=2624981>.

License GPL (>= 3)

Depends R (>= 3.0.1)

Imports gplots, quantreg, sfa, KernSmooth, lpSolve, ggplot2,
doParallel, foreach, iterators

NeedsCompilation no

Repository CRAN

Suggests testthat

Date/Publication 2020-06-11 09:10:02 UTC

R topics documented:

NCA-package *Necessary Condition Analysis*

Description

The NCA package implements Necessary Condition Analysis (NCA) as developed by Dul (2016). For running the NCA package a data file (e.g., mydata.csv, which contains the input data) must be available. An example data file (presented in above paper) is included in the package. The user must load the data and call the nca function.

Details

Package:	NCA
Type:	Package
Version:	3.0.3
Date:	2020-06-11
License:	GPL (>= 3)

Author(s)

Author: Jan Dul <jdul@rsm.nl>
Maintainer: Govert Buijs <buijs@rsm.nl>

References

Dul, J. 2016. Necessary Condition Analysis (NCA). “Logic and Methodology of ’Necessary but not Sufficient’ causality.” This paper is published in Organizational Research Methods 19(1), 10-52 (Sage publishers)
http://dx.doi.org/10.2139/ssrn.2588480

See Also

nca_analysis, nca_output

Examples

```
# A more detailed guide can be found here : http://repub.eur.nl/pub/78323/
# or https://ssrn.com/abstract=2624981

# Load data from a CSV file with header and row names:
data <- read.csv('mydata.csv', row.names=1)
# Or load the example dataset
data(nca.example)
data <- nca.example

# Run NCA with the dataset.
# Specify the independent (cause) and dependent (effect) variables
# More than 1 independent variables can be specified with a vector
model <- nca_analysis(data, c(1, 2), 3)

# A quick summary can be displayed by printing the model
model

# A full summary is shown by nca_output (see documentation for more options)
nca_output(model)

# The model is a list of 6 items :
# - plots (1 for each independent variable)
# - summaries (1 for each independent variable)
# - bottleneck tables (1 for each ceiling technique)
# - peers (1 dataframe for each independent variable)
# - tests (1 list for each independent variabl)
# - test.time (total time to run all tests)
names(model)

# The first item contains the graphical outputs for each independent variable
# This is not really useful to humans
model$plots[[1]]

# The seconds item contains a list with the summaries for the independent variables
model$summaries[[1]]

# The third item contains a list with the bottleneck tables, one for each ceiling technique
model$bottlenecks$cr_fdh

# The fourth item shows the peers, for each independent variable
model$peers$Individualism

# For the fifth and sixth item, the test.rep needs to be bigger than 0
# Optionally the p_confidence (default 0.95) and the p_threshold (default 0) can be set
model <- nca_analysis(data, c(1, 2), 3, test.rep=100)

# The fifth item shows the tests for each independent variable
# This is not really useful to humans
model$tests$Individualism

# The last item shows the total time needed to perform the tests
model$test.time
```

ceilings *a set of all available ceiling techniques*

Description

Ceilings to use for the nca or nca_analysis methods
> nca(data, c(1, 2), 3, ceilings=c('ols', 'ce_fdh', 'cr_fdh'))

Ceiling Technique	**Name**
lh	Low-High
cols	Corrected Ordinary Least Squares
qr	Quantile Regression
sfa	Stochastic Frontier Analysis
ce_vrs	Ceiling Envelopment with Varying Return to Scale
cr_vrs	Ceiling Regression with Varying Return to Scale
ce_fdh	Ceiling Envelopment with Free Disposal Hull
cr_fdh	Ceiling Regression with Free Disposal Hull

line.colors *a set defining the line colors for the plots*

Description

Set before calling nca_output
> line.colors['ce_fdh'] <- 'blue'

Reset one line color by setting it to NULL
> line.colors['ce_fdh'] <- NULL

Reset all line colors by setting line.colors to NULL
> line.colors <- NULL

Format

This is a list with line colors for each ceiling technique

ols	’green’	**lh**	’red3’
cols	’darkgreen’	**qr**	’blue’
ce_vrs	’orchid4’	**cr_vrs**	’violet’
ce_fdh	’red’	**cr_fdh**	’orange’
sfa	’darkgoldenrod’		

line.types *a set defining the line types for the plots*

Description

Set before calling nca_output
> line.types[’ce_fdh’] <- 1

Reset one line type by setting it to NULL
> line.types[’ce_fdh’] <- NULL

Reset all line types by setting line.types to NULL
> line.types <- NULL

Format

This is a list with line types for each ceiling technique

ols	1	**lh**	2
cols	3	**qr**	4
ce_vrs	5	**cr_vrs**	1
ce_fdh	6	**cr_fdh**	1
sfa	7		

line.width *parameter defining the width of the lines in the plots*

Description

This will be used for the **lwd** parameter of the plot, default is 1.5.
Set before calling nca_output
> line.width <- 5

nca *Run a basic NCA analyses on a data set*

Description

Run a basic NCA analyses on a data set

Usage

```
nca(data, x, y, ceilings=c('ols', 'ce_fdh', 'cr_fdh'))
```

Arguments

`data`	dataframe with columns of the variables
`x`	collection of the columns with the independent variables
`y`	index or name of the column with the dependent variable
`ceilings`	vector with the ceiling techniques to include in this analysis

Value

Returns a list with 3 items (see examples for further explanation):

`plots`	A list of plot-data for each x-y combination
`summaries`	A list of dataframes with the summaries for each x-y combination
`bottlenecks`	A list of dataframes with a bottleneck table for each ceiling technique

Examples

```
# Load the data
data(nca.example)
data <- nca.example

# Basic NCA analysis
# Independent variables in the first 2 columns, dependent variable in the third column
# This shows scatter plot(s) with the ceiling lines and the effect size(s) on the console
nca(data, c(1, 2), 3)

# Columns can be selected by name as well
nca(data, c('Individualism', 'Risk taking'), 'Innovation performance')

# Define the ceiling techniques via the ceilings parameter
nca(data, c(1, 2), 3, ceilings=c('ols', 'ce_vrs'))
# These are the available ceiling techniques
print(ceilings)
```

nca.example *NCA example data with 2 independent and 1 dependent variables*

Description

This data set has Individualism and Risk taking as independent variables, and Innovation performance as the dependent variable for 28 countries.

Usage

```
data(nca.example)
```

Format

A matrix containing 28 observations, incl. headers and row names.

nca_analysis *Run NCA analyses on a data set*

Description

Run multiple types of NCA analyses on a dataset

Usage

```
nca_analysis(data, x, y, ceilings=c('ols', 'ce_fdh', 'cr_fdh'),
             flip.x=FALSE, flip.y=FALSE, scope=NULL,
             bottleneck.x='percentage.range', bottleneck.y='percentage.range',
         steps=10, step.size=NULL, cutoff=0, qr.tau=0.95, effect_aggregation = c(1),
             test.rep=0, test.p_confidence=0.95, test.p_threshold=0)
```

Arguments

`data`	dataframe with columns of the variables
`x`	index or name (or a vector of those) with independent variable(s)
`y`	index or name of the column with the dependent variable
`ceilings`	vector with the ceiling techniques to include in this analysis
`flip.x`	reverse the direction of the independent variables use either a boolean for all independent variables, or a vector with the same length as x
`flip.y`	reverse the direction of the dependent variables, boolean
`scope`	a theoretical scope in list format : (x.low, x.high, y.low, y.high)

`bottleneck.x` options for displaying the independent variables in the bottleneck table
'percentage.range' to display the percentage of range between min(x) and max(x)
'percentage.max' to display the percentage of max(x)
'actual' to display the actual values
'percentile' to display the percentiles
Using percentage.max with negative values might yield counterintuitive results

`bottleneck.y` options for displaying the dependent variables in the bottleneck table,
see bottleneck.x

`steps` number of steps in the bottleneck table (only useful if step.size is not defined)

`step.size` define the step size in the bottleneck table
the user will be warned if the stepsize does not fit the Y range
defaults to null for using steps

`cutoff` display calculated x,y values that are lower/higher than lowest/highest observed x,y values in the bottleneck table as:
0 : NN (not necessary) and NA (not available)
1 : lowest/highest observed values
2 : calculated values

`qr.tau` define the qr tau (between 0 and 1), default 0.95

`effect_aggregation`
define the corners to aggregate into the effect size. 1 is upper-left and is always selected, 2 is upper-right, 3 is lower-left and 4 is lower-right

`test.rep` number of resamples in the approximate permutation test

`test.p_confidence`
confidence level of the estimated p-value
will be used to calculate the p-accuracy for a given number of samples (test.rep), default 0.95

`test.p_threshold`
define the threshold significance level in the returned plot of the approximate permutation test, default 0 (returns nothing)

Value

Returns a list of 6 items (see examples for further explanation):

`plots` A list of plot-data each x-y combination

`summaries` A list of dataframes with the summaries for each x-y combination

`bottlenecks` A list of dataframes with a bottleneck table for each ceiling technique

`peers` A list of peers for each independent variable

`tests` The results of the test for each independent variable, if test.rep is larger than 0 (not human friendly, use nca_output)

`test.time` The total time needed to run the tests for all independent variables

Examples

```
# Load the data
data(nca.example)
data <- nca.example

# Basic NCA analysis, with independent variables in the first 2 columns
# and the dependent variable in the third column
model <- nca_analysis(data, c(1, 2), 3)

# Use nca_output to show the summaries (see nca_output documentation for more options)
nca_output(model)

# Columns can be selected by name as well
model <- nca_analysis(data, c('Individualism', 'Risk taking'), 'Innovation performance')

# Define the ceiling techniques via the ceilings parameter, see 'ceilings' for all types
model <- nca_analysis(data, c(1, 2), 3, ceilings=c('ce_fdh', 'ce_vrs'))
# These are the available ceiling techniques
print(ceilings)

# For using the upper right corner(s), 'flip' the x variables
model <- nca_analysis(data, c(1, 2), 3, flip.x=TRUE)
# It is also possible to flip a single x variable
model <- nca_analysis(data, c(1, 2), 3, flip.x=c(TRUE, FALSE))
# Flip the y variable if the lower corners need analysing
model <- nca_analysis(data, c(1, 2), 3, flip.x=c(TRUE, FALSE), flip.y=TRUE)

# Use a theoretical scope instead of the (calculated) empirical scope
model <- nca_analysis(data, c(1, 2), 3, scope=c(0, 120, 0, 240))

# By default the bottleneck tables use percentages of the range for the x and y values.
# Using the percentage of the max value is also possible
model <- nca_analysis(data, c(1, 2), 3, bottleneck.y='percentage.max')
# Use the actual values, in this case the x-value
model <- nca_analysis(data, c(1, 2), 3, bottleneck.x='actual')
# Use percentile, in this case for the y-values
model <- nca_analysis(data, c(1, 2), 3, bottleneck.y='percentile')
# Any combination is possible
model <- nca_analysis(data, c(1, 2), 3, bottleneck.x='actual', bottleneck.y='percentile')

# The number of steps is adjustible via the steps parameter
model <- nca_analysis(data, c(1, 2), 3, steps=20)
# Or via the step.size parameter, this ignores the steps parameter
model <- nca_analysis(data, c(1, 2), 3, step.size=5)

# If the ceiling line crosses the X = Xmax line at a point C below Y = Ymax,
# for Y < Yc < Ymax, the corresponding X in the bottleneck table is displayed as 'NA'
# It is also possible to display them as Xmax
model <- nca_analysis(data, c(1, 2), 3, cutoff=1)
# or as the calculated value on the ceiling line
model <- nca_analysis(data, c(1, 2), 3, cutoff=2)
```

```
# To run tests, the test.rep needs to be larger than 0
# Optionally the p_confidence (default 0.95) and the p_threshold (default 0) can be set
model <- nca_analysis(data, c(1, 2), 3, test.rep=1000, test.p_confidence=0.9, test.p_threshold=0.05)


# The output of the tests can be shown via nca_output with test=TRUE
nca_output(model, test=TRUE)
```

nca_output *display the result of the NCA analysis*

Description

Show the plots, NCA summaries and bottleneck tables of a NCA analysis.

Usage

```
nca_output(model, plots=FALSE, bottlenecks=FALSE, summaries=TRUE,
                  test=FALSE, pdf=FALSE, path=NULL)
```

Arguments

model	the output of the nca or nca_analysis commands
plots	if true show the plot(s)
bottlenecks	if true show the bottleneck table(s)
summaries	if true show the summaries
test	if true show the result of the statistical sigificance test (if present)
pdf	if true export to pdf
path	optional path for the output file(s)

Examples

```
# Use the result of the nca command:
data(nca.example)
data <- nca.example
model <- nca_analysis(data, c(1, 2), 3)

# This shows the summaries in the console
nca_output(model)

# Suppress the summaries and display the plots
nca_output(model, plots=TRUE, summaries=FALSE)

# Suppress the summaries and display the bottlenecks
nca_output(model, bottlenecks=TRUE, summaries=FALSE)

# Show the results of the statistical significance test (p-value)
# Make sure to set test.rep in nca_analysis
nca_output(model, test=TRUE)

# Show all four
nca_output(model, plots=TRUE, bottlenecks=TRUE, test=TRUE)

# Per independent variable, export plots and summaries to PDF files,
# and export all the bottleneck tables to a single PDF file
nca_output(model, plots=TRUE, bottlenecks=TRUE, pdf=TRUE)

# Use the path option to export to an existing directory
outdir <- '/tmp'
nca_output(model, plots=TRUE, pdf=TRUE, path=outdir)
```

`point.color` *parameter defining the point color in the plots*

Description

This will be used for the **col** parameters of the plots, default is blue.
Set before calling nca_output
> point.color <- red

See http://www.statmethods.net/advgraphs/images/points.png for more symbols

`point.type` *parameter defining the plotting symbol in the plots*

Description

This will be used for the **pch** parameter of the plots, default is 21.
Set before calling nca_output
> point.type <- 22

See http://www.statmethods.net/advgraphs/images/points.png for more symbols

Index