Necessary Condition Analysis (NCA) with R (Version 3.3.3) A Quick Start Guide

5 September 2023

Jan Dul

Erasmus University Rotterdam (EUR) – Rotterdam School of Management

What are the changes in versions 3.3.3?

Version 3.3.3 of the NCA software in R includes several small changes and bug fixes in the `nca_analysis` function: fix bug in method peers, layout changes in test , and test.p_threshold = 0.05 is set as default.

If you install the package for the first time: see below from the section 'Abstract'. If you have installed an older version of the NCA package you can update the package as follows:

```
update.packages("NCA")
```

The NCA package uses other packages. It is possible that these packages that are not installed on your computer. In that case the following error message may appear: `There is no package called …` , where … corresponds to the name of the missing package. You then need to install the specific packages first.

```
install.packages("…")
```

Abstract

Necessary Condition Analysis (NCA) is an approach and data analysis technique for identifying necessary conditions in datasets. It can complement traditional regression-based data analysis as well as methods like QCA (see then NCA website www.erim.nl/nca for more information on NCA). This guide helps a novice user without knowledge of R or NCA to install the free R and NCA software on the user's computer and to perform an NCA analysis within 15 minutes. The main instructions are:

I. Install R
II. Install NCA
III. Load data
IV. Run NCA.

Details of the method can be found in:

- Dul, J. (2016) Necessary Condition Analysis (NCA). Logic and methodology of 'Necessary but not Sufficient' causality. *Organizational Research Methods* 19(1), 10-52. (https://journals.sagepub.com/doi/pdf/10.1177/1094428115584005)

- Dul, J. (2020), *Conducting Necessary Condition Analysis*, Sage Publications, ISBN: 9781526460141. (https://uk.sagepub.com/en-gb/eur/conducting-necessary-condition-analysis-for-business-and-management-students/book262898)
- Dul, J., van der Laan, E., & Kuik, R. (2020). A statistical significance test for Necessary Condition Analysis. *Organizational Research Methods*, 23(2), 385-395. (https://journals.sagepub.com/doi/10.1177/1094428118795272)

I. Install R (for new users)

1. What is R?

R is an open source programming language that is increasingly used for data analysis in different scientific fields, including the social sciences. It contains many statistical, mathematical and graphical functions that are also part of commercial statistical software such as SPSS and SAS. Additionally, R can run specific user-defined functions ("packages"). One such package is NCA. Only some basic knowledge about R (presented in this guide) is needed to run NCA with R.

2. How can I install R (for new users)?

R can be installed (downloaded) on your computer from the central R-website (see below). You need to have administration rights on your computer to install this software. The version of R that you must download depends on the platform of your computer: Windows or OS X (Mac). There is also a version for Linux.
For Windows users:

- Go to http://cran.r-project.org/bin/windows/base/
- Download "R x.y.z for Windows", where x,y,z, is the latest version number.
- Open the downloaded file and follow the instructions (accept all defaults).

For OS X (Mac) users:

- Go to http://cran.r-project.org/bin/macosx/
- Download the correct version for your OS X.
- Open the downloaded file and follow the instructions (accept all defaults).

Additionally we recommend to install RStudio, which is a user-friendly environment from which you can work with R. There are RStudio versions for Windows, OS X and Linux.

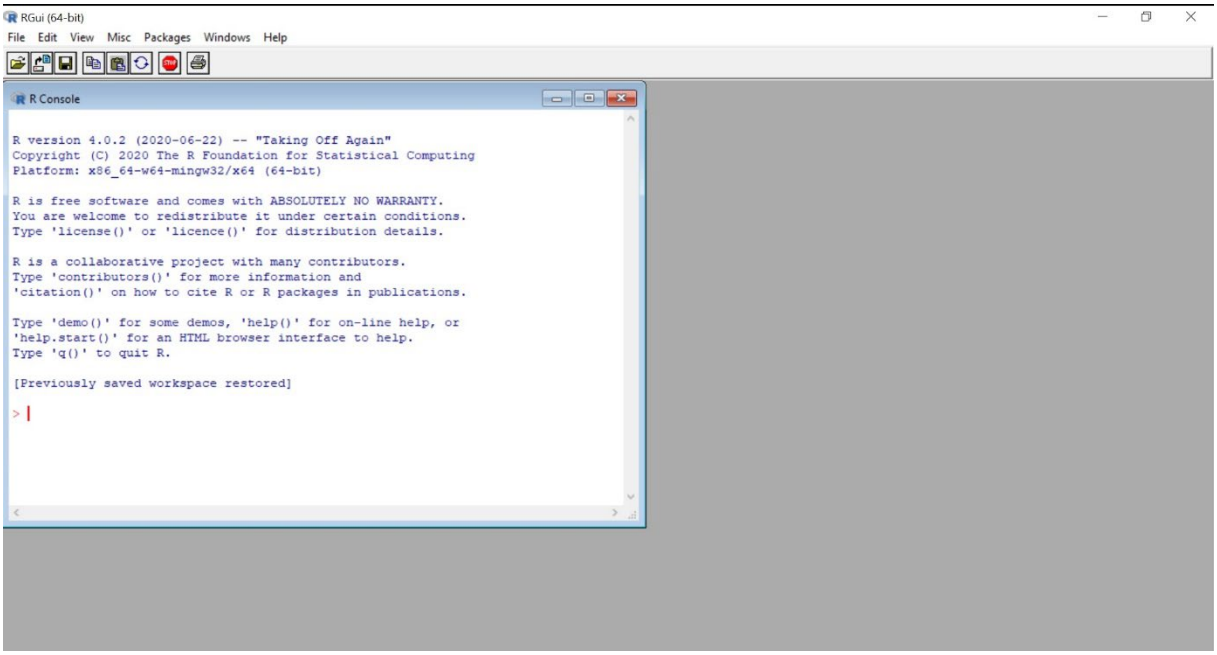
- Go to https://www.rstudio.com/products/rstudio/download/
- Select RStudio Desktop (open source license)
- Download the appropriate installer of RStudio x.y.z. for your platform (Windows, OS X, etc.).
- Follow the instructions (accept all defaults).

3. How can I start R?

R can be started in two ways: by opening R's interface (RGui), or by using Rstudio.

RGui:
Clicking on the R shortcut on the desktop will give you R's Graphical user interface (RGui).

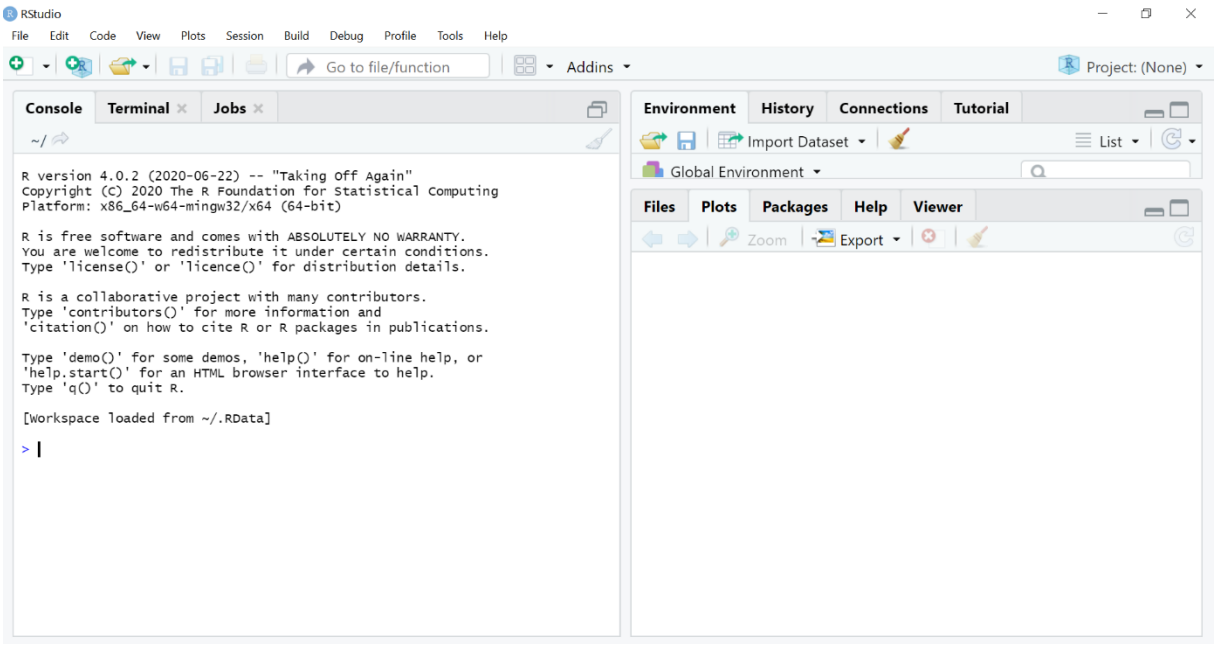
Opening R with RGui screen (console)



RGui opens with the window called “R console”. Here you can find some basic information about R. In the R console you can type instructions after the ">" prompt[1]. Each instruction must be followed by <enter>. In this guide the instructions for R are shown in `courier` font (“typewriter characters”). These instructions can be typed (or copy-paste) after the prompt in the console. Numerical output of NCA will be displayed in the console. Graphical output of NCA will be displayed in new windows that open in the RGui screen.

RStudio:
Go to the folder RStudio in the program files, open this folder, and click on the RStudio executable file. This will open the RStudio screen.

[1] For certain instructions you can also use the pull-down menu at the top of the RGui or RStudio page. In this guide menu instructions are printed in calibri font and successive steps are connected by an arrow “→”.



Opening of R with RStudio (console on the left and two other windows on the right) opens two other windows. The upper right window contains tabs with Environment, History, Connection and Tutorial, and the lower right window contains tabs with Files, Plots, Packages, Help, and Viewer. The Help tab displays the manuals for the packages that are installed on your computer, including the manual for NCA. This manual provides details of all instructions and options that can be used in the NCA package. The Files tab shows the folder structure on your computer. For the purpose of this quick start guide we primarily use the Plot and Viewer tabs in the lower-right window that displays plots produced by NCA. Further information about RStudio can be found on internet. A free introduction course can be obtained from for example DataCamp.

Click on the Plot tab in the lower right window. In the remainder of this guide this window will be called “plots window”. Numerical output of NCA will be printed in the console. Just like for the console in RGui, in RStudio’s console you can type instructions after the ">" prompt. However, we recommend using the “script window” that can be started with the pull down menu in RStudio as follows: File → New File → R Script. In the script window the instructions can be typed and edited. The script can be saved by using File → Save (or using the save button), or File → Save As. The stored script is a file with the extention “.R”, for example “Myscript.R”. You can load (“source”) an existing script by using File → Open file.

You can run the script in the script window by successively executing each instruction line. You can also select several instructions and run this set of instructions at once.
We advise to type instructions in the script window, so you can keep track of the instructions and store them in an R file for later replication.

4. How can I set my “working directory”?

The “working directory” is a folder on your computer where (by default) R searches your data and stores output file(s). You can check your current working directory by typing `getwd()` in the lower left window (console) after the “>” followed by a return (enter), or (preferably) in the upper left window (script window) after the line number, followed by pushing on the “Run button”:

```
getwd()
```

You can change your working directory by typing:

```
setwd("...\\MyWorkingDirectory")
```

In this example the working directory is named “MyWorkingDirectory” but you can use any name for the working directory folder. Note that R uses "\\" in the directory tree. Alternatively, you can use one forward slash “/”, but not one backward slash “\”.

You can also use the Files tab of the lower-right window of RStudio to select the working directory. Tick the square next to the folder that you want to select as working directory, select “More”, and select “Set as Working Directory”.

The working directory needs to be specified in R each time that you start R.

II. Install NCA

5. What is NCA software?

NCA is a free package for R. The reference to the NCA software is:
Dul, J. 2018. Necessary Condition Analysis. R Package Version 3.0. URL: http://cran.r-project.org/package=NCA.

6. How can I install NCA for R?

Installation of the NCA package is possible from R-version 3.0.1. In the console or script window type the following instruction**:**

```
install.packages("NCA", dependencies = TRUE)
```

Select the location nearest to you.[2]

This will install package NCA and all other R packages ("dependencies") that are used by NCA on your computer.

7. How can I install new versions of NCA?

A new version of NCA and of other installed packages can be obtained by:

[2] Alternatively NCA can be installed by using the file “NCA_x.y.z.tar.gz”, where x,y,z, is a version number . This file contains the NCA software to be copied to your computer.
Then you can install the package from this file by typing (or copy-paste) in the console the following instructions (note that in the final instruction (…\\ ) is the path to the NCA_x.y.z.tar.gz file on your computer, which must be specified by you; also note that R uses “\\”or "/" in the directory tree):
install.packages("…\\NCA_x.y.z.tar.gz", repos=NULL, type="source")
Note that R cannot handle too long path names when installing the package this way. Then R needs to be re-installed on your computer closer to the root.
You can also install the package from the pull down menu of RStudio: Tools → Install Packages → Install from: Package Archive File (.zip; tar.gz) → Browse to the location of the NCA package on your computer → Install
It may be necessary to install also the packages that are used by the NCA package (dependencies). Missing packages are mentioned in error messages while installing NCA.

```
update.packages()
```

8. How can I load NCA?

After the NCA package is installed (downloaded) on your computer, it must be loaded (activated) in R (NCA must be loaded each time you start R):

```
library(NCA)
```

Some basic information about NCA is displayed in the console:

```
Please cite the NCA package as:

  Dul, J. 2021.
  Necessary Condition Analysis.
  R Package Version 3.1.0.
  URL: http://cran.r-project.org/web/packages/NCA/

This package is based on:
  Dul, J. (2016) "Necessary Condition Analysis (NCA):
  Logic and Methodology of 'Necessary but Not Sufficient' Causality."
  Organizational Research Methods 19(1), 10-52.
  https://journals.sagepub.com/doi/pdf/10.1177/1094428115584005
and
  Dul, J. (2020) "Conducting Necessary Condition Analysis"
  SAGE Publications, ISBN: 9781526460141
  https://uk.sagepub.com/en-gb/eur/conducting-necessary-condition-analysis-
  for-business-and-management-students/book262898
and
  Dul, J., van der Laan, E., & Kuik, R. (2020).
  A statistical significance test for Necessary Condition Analysis."
  Organizational Research Methods, 23(2), 385-395.
  https://journals.sagepub.com/doi/10.1177/1094428118795272

A BibTeX entry is provided by:
  citation('NCA')

A quick start guide can be found here:
  http://repub.eur.nl/pub/78323/
  or
  https://ssrn.com/abstract=2624981

For general information about NCA see :
  http://www.erim.nl/nca
```

If you get a warning message that the NCA package was built under version x.x.x. (for Windows) you have an older R version. Then it is advised to update your R package, otherwise some NCA functions may not work properly. A simple way to update your R package is by installing the package "installr". It is advised to leave RStudio and to update R from Rgui as follows:

```
install.packages("installr")
library(installr)
updateR()
```

During the installation process you can press "next", "OK", and "Yes" on everything. Note that this process (in particular copying of files and updating of packages), may take several minutes.

III. Load data

9. How can I load the example dataset?

NCA comes with an example dataset of N= 28 countries with two independent variables or conditions (x_1= Individualism, x_2=Risk taking) and one dependent variable or outcome. (y=Innovation performance). You can load (activate) the example data in your R session as follows:

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

You can rename the data as "data".

```
data <- nca.example
```

The combination of symbols "<" and "−" is the "assignment operator" of R, which connects two objects (in this case "data" and "nca.example"). Usually the "=" symbol could be used as assignment operator, but there are exceptions. Therefore "<−" is used in this guide.

After this instruction the example data is a data object known as "data".

The data are shown on the screen in the console by typing the data name:

```
data
```

The first column on the screen contains the row names of cases: in this example "countries". The first row on the screen is the header, which contains the names of the variables. There are three data columns. In this example the first two data columns are the two independent variables, and the last column is the dependent variable.

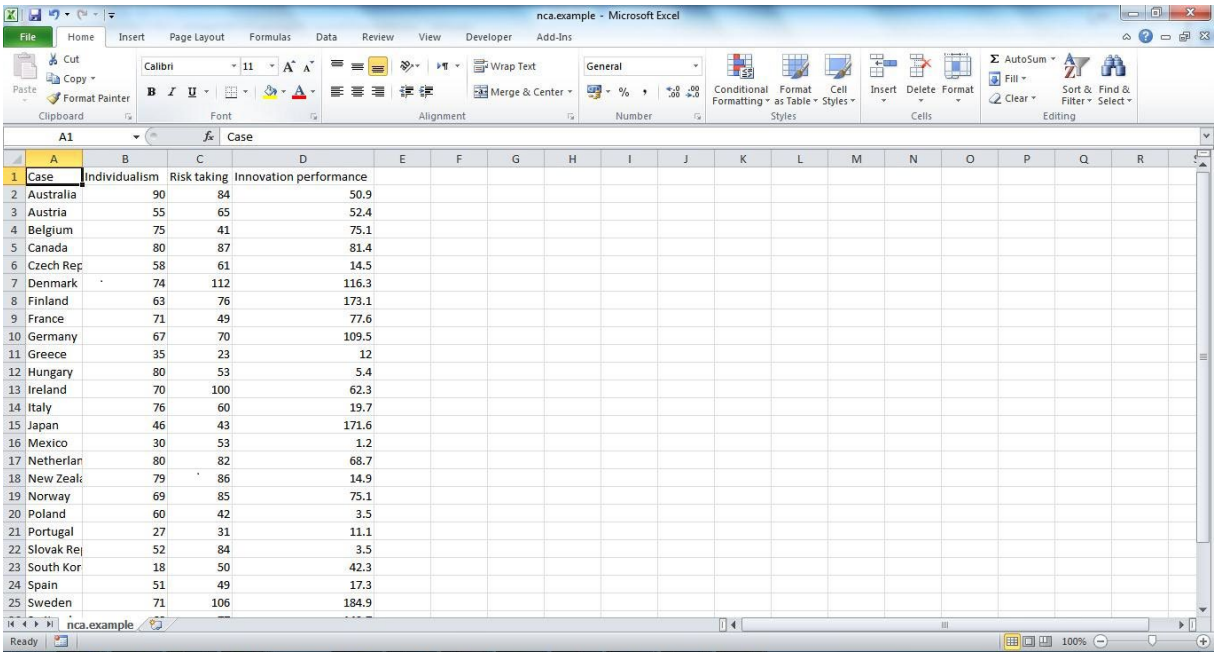
By using the upper arrow on the keyboard you can get back previous instructions that you typed.

10. How should I prepare my own data file?

NCA presumes that the data in your data file (input file) are organized in a similar way as is commonly used in data files, for example SPSS data files. Rows correspond to cases (except for the first row, which can be a header with variable names; these names will appear in the plots and other NCA output). Columns correspond to variables (except for the first column, which can be row names). All variable values must be numbers (no letters).
A common data file type for R is .csv (e.g., an Excel file saved as .csv[3]). Missing data in a .csv file must be an empty cell (do not use NA, 999 or other symbols). Other data file types than .csv are possible as well. Examples include SPSS (.sav), Stata (.dta), and SAS (.xpt). See

[3] Depending on the region and language settings of your computer, your Excel program uses decimal points or decimal commas, and the separator in the csv file uses a comma or a semi-colon, respectively. In this guide it is presumed that you have decimal points and comma separators.

a general R-manual for instructions about how to import these other types of data files in R (many R manuals can be found on internet).
Data file of nca.example (.csv)



Case	Individualism	Risk taking	Innovation performance
Australia	90	84	50.9
Austria	55	65	52.4
Belgium	75	41	75.1
Canada	80	87	81.4
Czech Rep	58	61	14.5
Denmark	74	112	116.3
Finland	63	76	173.1
France	71	49	77.6
Germany	67	70	109.5
Greece	35	23	12
Hungary	80	53	5.4
Ireland	70	100	62.3
Italy	76	60	19.7
Japan	46	43	171.6
Mexico	30	53	1.2
Netherlan	80	82	68.7
New Zeala	79	86	14.9
Norway	69	85	75.1
Poland	60	42	3.5
Portugal	27	31	11.1
Slovak Rep	52	84	3.5
South Kor	18	50	42.3
Spain	51	49	17.3
Sweden	71	106	184.9

11. How can I load my data file?

In RStudio you can load your data from the upper right window by clicking on Import DataSet and subsequently select the file location and the characteristics of your dataset. The imported data are known in R by the name of the file (e.g., mydata). You can change the name of the data as follows:

```
data <- mydata
```

Alternatively, you can load your data by giving instructions in the script window or console.

Load your .csv data file (if it contains a header and row names) as follows:

```
data <- read.csv("mydata.csv", row.names = 1)
```

(If your file uses decimal commas instead of decimal points, you can try using `read.csv2` , instead of `read.csv`, see also footnote 3).

After this instruction your data is a data object known as “data”.

If your dataset has no row names you type:

```
data <- read.csv("mydata.csv")
```

and the rows will be identified on screen with a number from 1 to N.

If your dataset has no header (and no row names) you can type:

```
data <- read.csv("mydata.csv", header = FALSE)
```

and the columns will be identified on screen with a number from 1 to the total number of variables.

For loading your dataset you can also use the Import Dataset tab in the upper-right window of RStudio.

For loading an SPSS data file (.sav) (no header) you can type in the R console :

```
library("foreign")
data <- read.spss("mydata.sav", to.data.frame = TRUE)
```

For more information on loading SPSS data:

```
?read.spss
```

IV. Run NCA

12. How can I run an NCA analysis?

After the data are loaded you can run NCA by specifying the name of the data object (e.g., "data" if you have named your data as such) followed by the specification of one or more x-variables (the condition(s) or independent variable(s)) and one y variable (the outcome or dependent variable). A variable can be specified by its column numbers (index) or variable name (column name). For the nca.example data file the first and second columns are the independent variables "Individualism" and "Risk taking", respectively. The third column is the dependent variable "Innovation performance". You can run NCA for Individualism (x_1) and Innovation performance (y) by specifying the dataset (here the dataset nca.example is renamed as "data"), and the column numbers of the variables. Usually the analysis is given a name, for example "model":

```
model <- nca_analysis(data,1,3)
```

`nca_analysis` is the core function of NCA.

Instead of using the column numbers to identify the variables you can also use the variable names, enclosed by quotation marks (`"`) :

```
nca_analysis(data, "Individualism", "Innovation
performance")
```

The analysis does not show yet the results. The print the basic results you give the name of the analysis:

```
model
```

As a result the following NCA output is printed on the console:

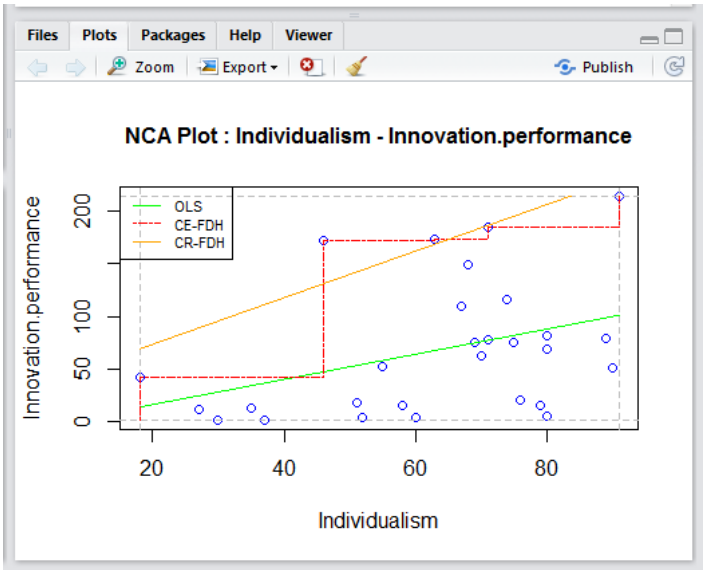
```
---------------------------------------------------------------
Effect size(s):
             ce_fdh cr_fdh
Individualism  0.416  0.307
---------------------------------------------------------------
```

The printed output shows per independent variable (here only the output for Individualism is shown) the necessary condition effect size for two different ceiling line techniques. These are

the default ceiling lines: the step function CE-FDH (Ceiling Envelopment – Free Disposal Hull) and the straight line CR-FDH (Ceiling Regression – Free Disposal Hull). The step ceiling line can be used when the data and underlying phenomenon are discrete with limited number of levels, and a straight ceiling line can be selected when the data and underlying phenomenon are discrete with a large number of levels, or continuous.

More output can be obtained with the `nca_output` function. For example, the following instruction displays the scatter plot (for Individualism and Innovation performance) in the plots window (and avoids summary output being printed in the console) .

```
nca_output (model, plots=TRUE, summaries = FALSE)
```



Note that the name Innovation performance is changed into Innovation.performance. After loading a dataset, spaces in names of columns or rows are changed into a dot ".". For example, the third column name in the csv data file is "Innovation performance", whereas the name after this file is loaded in R is changed into "Innovation.performance". In the remainder of this guide we use the names as loaded in R (hence with a dot). In the NCA package and in this guide, when variable names or other object names created by the user have separate words, a dot "." connects the words (e.g., "Innovation.performance" is a user defined object). An underscore ("_") is used to connect words in functions that are part for the package (e.g., "nca_analysis" is a function of the package).

The scatter plot shows the selected ceiling lines: the two default ceiling lines (CE-FDH in red, and CR-FDH in orange), and the OLS regression line (green) through the middle of the data (as a reference). If the effect size is greater than zero, there is an empty area in the upper-left corner of the scatter plot, which is an indication of the presence of a necessary condition. The necessary condition effect size (d) is the proportion of the scope above the ceiling: $d = C/S$. It ranges from 0 to 1 ($0 \leq d \leq 1$). The effect size indicates to what extent the condition is necessary for the outcome. In other words: to what extent the condition constrains the outcome, and the outcome is constrained by the condition. Hence, the effect size is the size of the empty zone relative to the total xy-zone where data can be expected (scope).

13. What is a general benchmark for the effect size?

According to Dul (2016, p.30) "An effect size can be valued as important or not, depending on the context. A given effect size can be small in one context and large in another. General qualifications for the size of an effect as 'small,' 'medium,' or 'large' are therefore disputable. If, nevertheless, a researcher wishes to have a general benchmark for necessary condition effect size, I would offer $0 < d < 0.1$ as a 'small effect,' $0.1 \leq d < 0.3$ as a 'medium effect,' $0.3 \leq d < 0.5$ as a 'large effect,' and $d \geq 0.5$ as a 'very large effect'."

14. How can I run a basic NCA analysis with multiple conditions?

You can perform an NCA analysis with two or more conditions (x_i), but always with only one outcome (y) at the same time. This is the multiple NCA. In the NCA instruction for multiple NCA the conditions (x variables) are specified as a vector (a list of variables) by using R's symbol for a vector which is a "c" (combine). For example, when running a multiple NCA with Individualism (x_1) in the first column of the dataset, Risk.taking (x_2) in the second column, and Innovation.performance (y) in the third column, the NCA instruction is:

```
model <- nca_analysis(data,c(1,2),3)
```

Alternatively, you can use the variable names, enclosed by quotation marks ("):

```
model <- nca_analysis(data,c("Individualism","Risk
taking"),"Innovation performance")
```

Yet another alternative is to identify the range of successive columns of conditions

```
model <- nca_analysis (data,c(1:2),3)

model
```

Now the following NCA output is printed on the console:

```
---------------------------------------------------------------
Effect size(s):
              ce_fdh cr_fdh
Individualism  0.416  0.307
Risk.taking    0.309  0.282
---------------------------------------------------------------
```

The output in the plots window now consists of two scatter plots, one for Individualism, and one for Risk.taking. You can switch between the scatter plots by using the arrows in the plots window.

15. How can I run a basic NCA analysis with different ceiling lines?

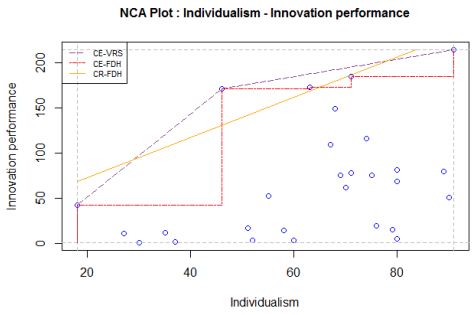
The default ceiling lines are CE-FDH (step function) and CR-FDH (straight line). With the option "ceilings" other ceiling lines can be selected. For example, the ceiling line CE-VRS can be selected, together with the two default ceiling lines (using a vector with the names of the ceiling lines) as follows:

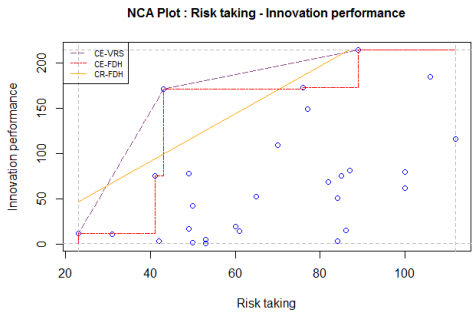
```
model <- nca_analysis(data,c(1:2),3,ceilings=c("ce_fdh",
"cr_fdh", "ce_vrs"))

model

---------------------------------------------------------------
Effect size(s):
              ce_vrs ce_fdh cr_fdh
```

```
Individualism 0.255  0.416  0.307
Risk taking   0.181  0.309  0.282
-----------------------------------------------------------------
```

The scatterplots show the three ceiling lines

```
nca_output(model, plots=TRUE, summaries = FALSE)
```





A plot can be saved with the Export tab in the plots window. The size of the plot can be changed with by moving the border of the plots window and Re- running the nca_output instruction (re-running latter also ensures a proper positioning of the legend after changing the borders).

16. How can I add the OLS line in the scatter plot?

Note that the above scatter plots do not show the OLS regression line. Although this line is not a ceiling line it can be included in the above scatter plots as a reference line (no NCA parameters are calculated for it) using the instruction:

```
nca_analysis(data,c(1:2),3,ceilings=c("ce_fdh", "cr_fdh",
"ce_vrs", "ols"))
```

17. How can I obtain an interactive scatter plot?

You can obtain an interactive scatter plot (plotly) with the following instruction:

```
nca_output(model, plotly=TRUE, summaries = FALSE)
```

The plot opens in the Viewer tab of the plots window. The plot identifies the 'peers' in red. Peers are observations near the ceiling line that are used to draw the default ceiling lines. They are called peers because they have the highest outcome for a given condition. When moving the pointer over the observations, the names and coordinates of the observation are displayed. When moving the pointer over the top of the plot, a toolbar pops up that has some additional functions, e.g. download, zoom, and selection. Subgroups of observations can be labeled. For example, the continents of the countries in the nca.example dataset can be named as follows (in the order of the rows in the dataset):

```
labels <- c("Australia", "Europe", "Europe", "North America",
"Europe", "Europe", "Europe", "Europe", "Europe", "Europe",
"Europe", "Europe", "Europe", "Asia", "North America",
"Europe", "Australia", "Europe", "Europe", "Europe", "Europe",
"Asia", "Europe", "Europe", "Europe", "Europe", "Europe",
"North America")
```

```
nca_output(model, summaries = FALSE, plotly = labels)
```

18. How can I perform a statistical significance test of the effect size?

In the NCA analysis a statistical significance test of the effect size can be performed with the argument `test.rep`. With `test.rep` a large number of random samples is created (e.g., 10,000) to obtain a distribution of effect sizes when the null-hypothesis is true (X and Y are not related). This distribution is used for comparison with the observed effect size and for calculating the p-value.

```
model <- nca_analysis(data,1:2,3,ceilings = "ce_fdh",
test.rep = 10000)

model
```

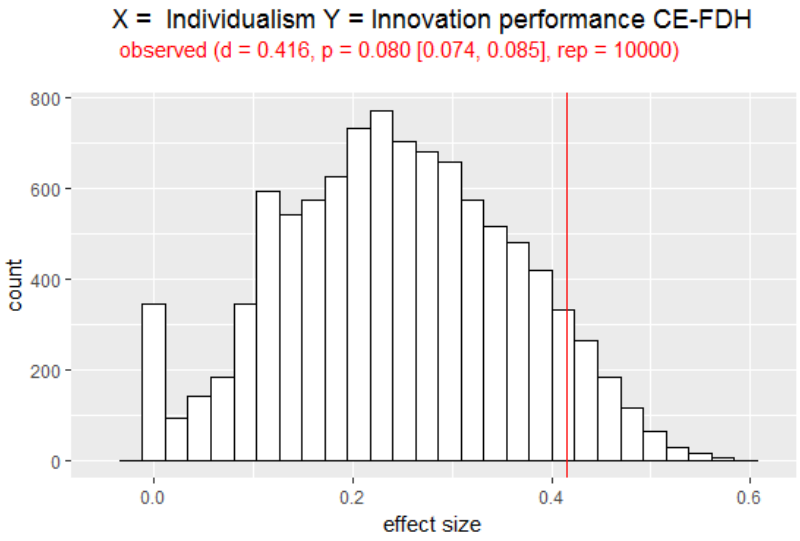
If `test.rep` is > 0, the calculation may take time some time and after the test is done a message the message "Done test for : …" is displayed and the results are shown in the console window.

```
---------------------------------------------------------------
Effect size(s):
               ce_fdh p
Individualism 0.416  0.080
Risk taking   0.309  0.101
---------------------------------------------------------------
```

The p-values may differ somewhat for each rest. If the number of resamples is large the p-values are stable and accurate.

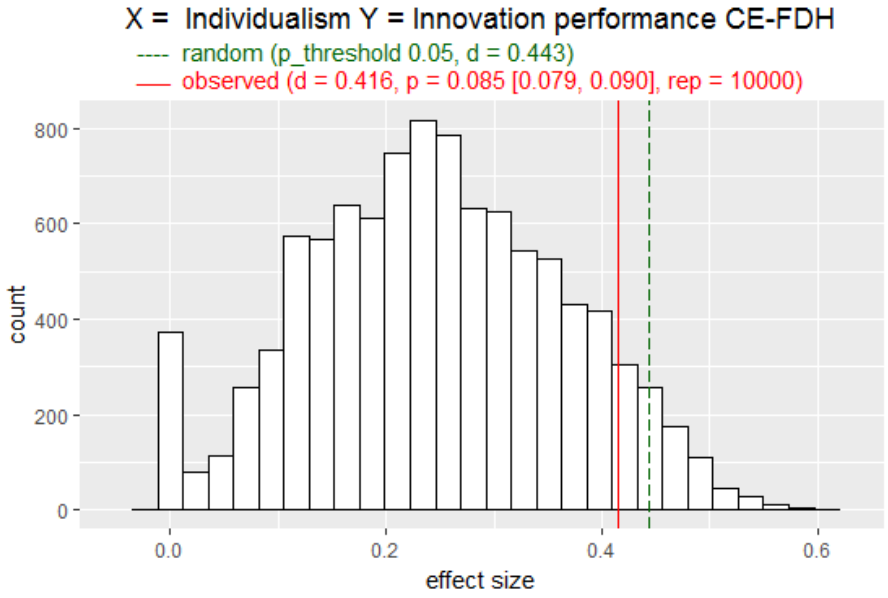
If the argument "test=TRUE" is added in the `nca_output` command the distribution of random effect sizes, the observed effect size and its p-value, are displayed in the plots window for each ceiling line.

```
nca_output(model, summaries = FALSE, test = TRUE)
```



If the test.p_threshold argument is used in `nca_analysis` (see below) the selected threshold value of the p-value is also displayed in the plots window.

```
model <- nca_analysis(data,1,3,ceilings = "ce_fdh",
test.rep = 10000, test.p_threshold=0.05)

nca_output(model, summaries = FALSE, test = TRUE)
```



19. How can I get detailed output from an NCA analysis?

With the argument `summaries = TRUE` in the `nca_output` function the user can get detailed output of an NCA analysis, for example:

```
model <- nca_analysis(data,1,3,ceilings = "cr_fdh",
test.rep=10000)
```

```
nca_output(model, summaries = TRUE)
```

The output consists of two parts:

```
--------------------------------------------------------------------------
NCA Parameters : Individualism - Innovation performance
--------------------------------------------------------------------------

Number of observations    28
Scope                  15563.6
Xmin                      18.0
Xmax                      91.0
Ymin                       1.2
Ymax                     214.4

                   cr_fdh
Ceiling zone     4772.541
Effect size         0.307
# above             2
c-accuracy          92.9%
Fit                 73.8%
p-value             0.165
p-accuracy          0.007

Slope               2.230
Intercept          28.353
Abs. ineff.      6018.517
Rel. ineff.        38.670
Condition ineff.   10.383
Outcome ineff.     31.565
```

The first part ("global") provides 6 lines of basic information about the dataset ("Number of observations", "Scope", "Xmin", "Xmax", "Ymin", and "Ymax"). "Scope" refers to the empirical area of possible X-Y combinations, given the minimum and maximum observed X and Y values. The next part ("param") consists of 13 lines of information about NCA parameters for each of the selected ceiling techniques (in this case the ceiling technique CE-FDH). The parameters are "Ceiling zone", which is the size of the "empty" area in the upper-left corner, "Effect size", which is the ceiling zone divided by the scope, "# above", which is the number of observations that are above the ceiling line, and hence in the "empty" ceiling zone, "c-accuracy", which is the number of observations on or below the ceiling line divided by the total number of observations and multiplied by 100%, "Fit", which relates to the "closeness" of the selected ceiling line to the CE-FDH ceiling line.

If in `nca_analysis`, the argument `test.rep > 0`, the p-value which is displayed in the summary. The corresponding parameters are the estimated p-value and the p-accuracy which is the estimated error of the p-value, such that the exact p-value lies between p-value minus p-accuracy and p-value plus p-accuracy. The p-accuracy improves when the number of resamples (`test.rep`) increases. The computation time also increases with number of resamples.

Further NCA parameters are "Slope" and "Intercept" of the straight ceiling line (no values are printed if the ceiling line is not a straight line, but a step function like CE-FDH), "Abs. ineff.", which is the total xy-space where x does not constrain y, and y is not constrained by x, "Rel. ineff.", which is the total XY space where X does not constrain Y, and Y is not constrained by X as percentage of the scope, "Condition ineff.", which is the condition

inefficiency that indicates for which range of X (as a percentage of the total range) X does not constrain Y (i.e., there is no ceiling line in that X-range), and "Outcome ineff.", which is the outcome efficiency that indicates for which range of Y (as a percentage of the total range of Y) Y is not constrained by X (i.e., there is no ceiling line in that Y-range).

20. How can I get the "bottlenecks" output from the NCA analysis?

The bottleneck table can be shown with `nca_output` as follows:

```
nca_output(model, bottlenecks=TRUE ,summaries = FALSE)
```

There is one bottleneck table for each ceiling line (only the bottleneck table for the CR-FDH ceiling is displayed below):

```
------------------------------------------------------------
Bottleneck CR-FDH (cutoff = 0)
Y Innovation.performance (percentage.range)
1 Individualism          (percentage.range)
------------------------------------------------------------
Y    1
0   NN
10  NN
20  NN
30  NN
40  11.0
50  24.1
60  37.2
70  50.3
80  63.4
90  76.5
100 89.6
```

The bottleneck table shows for which level of Y, which level of X is necessary. This is another interpretation of the ceiling line. The bottleneck table is particularly useful in multiple NCA (see below) for finding bottleneck levels of X's (the conditions) for a given level of Y (the outcome). For example, for a model with two necessary conditions:

```
model <- nca_analysis (data,c(1,2),3)
nca_output(model, summaries=FALSE, bottlenecks=TRUE)
```

This results in the following output on the console (for the CR-FDH ceiling line only):

```
------------------------------------------------------------
Bottleneck CR-FDH (cutoff = 0)
Y Innovation.performance (percentage.range)
1 Individualism          (percentage.range)
2 Risk.taking            (percentage.range)
------------------------------------------------------------
Y    1    2
0   NN   NN
10  NN   NN
20  NN   NN
30  NN   8.0
40  11.0 17.1
50  24.1 26.2
60  37.2 35.2
70  50.3 44.3
80  63.4 53.4
90  76.5 62.4
100 89.6 71.5
```

By default the Y values in the bottleneck table (first column) are expressed as percentage of the range of (observed) values (0%= lowest observed value, 100% is highest observed, 50% is in the middle of the lowest and highest observed values). The other columns are the corresponding values of the independent variables according to the ceiling line (also expressed as percentage of the range). The bottleneck table can be read horizontally (by row) as follows. For a given (desired) value of the dependent variable (in the first column) it shows the minimum required values of the independent variables (in the next columns). Hence, in nca.example according to the CR-FDH ceiling line, for an Innovation.performance level of 80%, the necessary level of Individualism is 63.4% and the necessary level of Risk.taking is 53.4%. At 30% for Y, only X_2 is necessary and at 20% none of the independent variables is necessary (NN=Not Necessary). Usually, when the dependent variable increases from 0% to 100%, more independent variables become necessary, and required levels of the independent variables become higher. The values of the Y and X variables in the bottleneck table can be also be expressed as "actual values" or as "percentages of the maximum values", by changing the defaults setting of `nca_analysis` (see below).

21. How can I select conditions for the output?

With the `selection` argument of `nca_output` it is possible to select conditions to be included in the plotted and printed output, for example:

```
nca_output(model, plots = T, plotly = T,
bottlenecks=TRUE, selection = "Individualism")
```

22. How can I obtain output as pdf files?

Three types of pdf files of the output of an NCA analysis can be generated and stored in the working directory as follows:

```
nca_output(model, plots=TRUE, summaries=TRUE,
bottlenecks=TRUE, pdf=TRUE)
```

The files "summary.Individualism-Innovation_performance" and "summary.Risk.taking-Innovation.performance.pdf" contains the output of the summaries for each condition.

The file: "bottlenecks.Innovation.performance.pdf" contains the bottleneck tables for the outcome Innovation.performance.

The files "plot.Individualism-Innovation.performance.pdf" and "plot.Risk.taking-Innovation.performance.pdf" contains the output of the plots for each condition.

The pdf files are placed in the Working Directory. The pdf output can also be directed to another existing folder (e.g., MyNCA) by providing the folder name and the path to that folder, as follows:

```
nca_output(model, plots=TRUE, summaries=TRUE,
bottlenecks=TRUE, pdf=TRUE, path="C:/Data/MyNCA")
```

23. How can I change the default settings of nca_analysis?

The default setting of the NCA analysis can be changed by changing the arguments below. For instructions in the R manual for NCA (type `?nca_analysis` in R).

```
nca_analysis(data, x, y, ceilings=c("ols", "ce_fdh",
"cr_fdh"), corner=0, 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)
```

`corner` allows the analysis of empty spaces in other corners of the XY scatter plot than the upper left corner. `corner = 1` (upper left corner) tests the necessity of the *presence or high level of X* for the presence or high level of Y. `corner = 2` (upper-right corner) testst *the absence or low level of X* for the presence or high level of Y. `corner = 3` (lower-left corner) tests the necessity of the presence or high level of X for the *absence or low level of Y*. `corner = 4` (lower-right corner) tests the necessity of the absence or low level of X for the absence or low level of Y. The `corner` argument can also be a vector representing the selected corner for each condition separately. For example: `corner = c(2,1)` selects the second corner for the first condition and the first corner for the second condition. If the `corner` argument is used, the `flip.x` and `flip.y` arguments are disabled. With `corner = 0` the `flip.x` and `flip.y` arguments can be used to flip the X and Y axes separately. With `flip` always the upper left corner is analysed. With `effect_aggregation` the effect sizes of different corners can be added. With `qr.tau` the quantile in the quantile regression ceiling technique can be set. `qr.tau = 0.95` approximates a ceiling line and `qr.tau = 0.05` approximates a floor line for `corner = 3`. With the `scope` argument a theoretical rather than empirical scope can be selected. With `steps`, `step.size` and `cutoff` the bottleneck table can be customized. The arguments `test.rep`, `test.p_confidence` and `test.p_threshold` can be used to calculate p-values in statistical significance testing for NCA for the number of samples, the confidence level for calculating p-accuracy, and the threshold significance level.

24. How can I select a specific parameter for further analysis?

The output of `nca_analysis` is stored in three lists of data frames (plots, summaries and bottlenecks). "plots" can be used for customizing plots (see below) and from the "summaries" data frames several NCA parameters can be selected. The summaries data frame for Individualism can be printed on the console as follows:

```
model$summaries[["Individualism"]]
```

This data frame consists of a vector "global" with the descriptive data of the dataset, a matrix "params" with the NCA parameters for two ceiling techniques, and a vector "names" with the names of the x and y variables. For example, the scope can be selected by:

```
model$summaries[["Individualism"]]$global[2]
```

The name of the y-variable can be selected by:

```
model$summaries[["Individualism"]]$names[2]
```

The value of the ceiling zone for the CR-FDH ceiling techniques can be selected by:

```
model$summaries[["Individualism"]]$params[2,2]
```

Further analysis of a value is possible by giving a name to the output. For example, the value of the ceiling zone for the CR-FDH ceiling techniques can be selected by:

```
effect<- model$summaries[["Individualism"]]$params[2,2]
```

and the result can be printed as follows:

```
effect
```

Having a name connected to an outcome allows for further analysis, for example:

```
half.effect<- effect/2

half.effect
```

25. How can I change the NCA output plot?

Before running `nca_output`, plots can be customized by changing the point type, line types, line colors (for each ceiling line separately) and line width (for all ceiling lines). For instance, this will change the line color for the CE-FDH line to blue:

```
line.colors["ce_fdh"] <- "blue"
```

You can run the output command with just the plots to see the effect:

```
nca_output(model, plots=TRUE, bottlenecks=FALSE,
summaries=FALSE)
```

Reset one line color to default type:

```
line.colors["ce_fdh"] <- NULL
```

Reset all line colors to default type:

```
line.colors <- NULL
```

If you want to change the point type you can type for example:

```
point.type <- 22

nca_output(model, plots=TRUE, bottlenecks=FALSE,
summaries=FALSE)
```

For all options see line.colors, line.types, line.width and point.type in the NCA manual or type:

```
?point.type
?line.colors
?line.types
?line.width
```

26. How can I further customize the NCA output plot? (for advanced users)

You may further want change the NCA output plot (the scatter plot with the ceiling line) to fit it to your personal preferences, or to conform to specific publication standards (e.g., black-

white, thicker lines). More advanced R users can get more control over the plot by downloading a script from:

https://stash.ict.eur.nl/projects/NCA/repos/public/browse/display_plot.R?raw

Save the file to display_plot.R and adjust to your liking.

Adjust and source the script, and then plot the output for the first independent variable:

```
source('display_plot.R')

display_plot(model$plots[[1]])
```

And for the second independent variable:

```
display_plot(model$plots[[2]])
```

27. How can I save an NCA plot?

A produced plot can be saved by using the "Export" in the plot window. A better alternative is to export the plot as a pdf file or png file to the working directory.

```
pdf("nca_example.pdf",5,5)
nca(data,1,3)
dev.off()

png("nca_example.png",units="cm", 15,15, res=300)
nca(data,1,3)
dev.off()

png("nca_example significance test.png",units="cm", 15,15,
res=300)
model<- nca_analysis(data,1,3, ceilings="ce_fdh",
test.rep=10000)
nca_output(model, test = TRUE)
dev.off()
```

28. How can I complement NCA with QCA

See Appendix 1. See also https://www.erim.eur.nl/necessary-condition-analysis/about-nca/faq/nca-and-other-data-analysis-methods/nca-and-qca/

29. Where can I get more information about the NCA methodology?

General information about NCA can be found here: http://www.erim.nl/nca

Details about the NCA methodology can be found in:

- Dul, J. (2016) Necessary Condition Analysis (NCA). Logic and Methodology of 'Necessary but not Sufficient' causality. *Organizational Research Methods* 19(1), 10-52 (https://journals.sagepub.com/doi/pdf/10.1177/1094428115584005)
- Dul, J. (2020) "Conducting Necessary Condition Analysis. SAGE Publications, ISBN: 9781526460141 (https://uk.sagepub.com/en-gb/eur/conducting-necessary-condition-analysis-for-business-and-management-students/book262898)

- Dul, J., van der Laan, E., & Kuik, R. (2020). A statistical significance test for Necessary Condition Analysis. Organizational Research Methods, 23(2), 385-395 (https://journals.sagepub.com/doi/10.1177/1094428118795272)

30. Where can I get more information about the NCA R package?

The latest version of this quick start guide for the Package 'NCA' – R can be found here: http://repub.eur.nl/pub/78323/ or http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2624981

The technical user manual for Package 'NCA' – R can be downloaded from: http://cran.r-project.org/web/packages/NCA/NCA.pdf

More information about NCA and its R package can be obtained from the author (jdul@rsm.nl) and the maintainer (buijs@rsm.nl). Comments and suggestions about NCA, the R Package or this Quick Start guide are very welcome.

APPENDIX 1: How to use NCA with QCA

1. How can NCA complement fsQCA?

Fuzzy Set Qualitative Comparative Analysis (fsQCA) is an approach and data analysis technique to identify sufficient but not necessary configurations (a configuration is a group of single conditions). QCA normally starts with the identification of single necessary conditions, because single necessary conditions must be part of any sufficient configuration, otherwise the configuration does not produce the outcome. However, fsQCA barely finds single necessary conditions. NCA is more refined to do so. For complementing QCA, NCA can be applied to QCA's data set of "membership scores" (values representing the extent to which a case is a member of a set, e.g., the set of countries with high innovation performance). For that purpose the data set of original scores ("raw scores" in QCA-language) must be transformed ("calibrated" in QCA language) to set membership scores.

2. How QCA transforms original data into membership scores?

Each variable must be transformed into set membership. QCA uses two steps for this transformation:

- Selecting three values of the variable ("anchor points", or "thresholds") that represent (1) membership that is "fully out of the set", (2) membership that is at the "cross-over point", and (3) membership that is "fully in of the set".
- Selecting a membership function for each variable to transform variable values into set membership scores.

Several techniques exist to select the thresholds. Data-driven calibration techniques are based on the empirical distribution of the data. Scale-driven techniques are based on anchor points of the measurement scale. Qualitative calibration techniques are based on theory and the qualitative knowledge of the researcher. Also for selecting the form of the membership function there are several alternatives, including logistic, quadratic, linear, etc.

The effect of calibration on the necessity outcomes of a QCA analysis can be evaluated with the QCA calibration evaluation tool described on the NCA website https://www.erim.eur.nl/necessary-condition-analysis/about-nca/faq/nca-and-other-data-analysis-methods/nca-and-qca/ and available here: https://r.erim.eur.nl/r-apps/qca/.

3. How to install and load QCA in R for performing transformation?

Transformation can be done by using R packages for QCA, for example QCA URL:http://cran.r-project.org/package=QCA). QCA can be installed and loaded (activated) as follows:

```
install.packages("QCA", dependencies=TRUE)

library ("QCA")
```

Information on the package can be obtained via:

```
?QCA
```

For details on the transformation type:

```
?calibrate
```

4. How to perform data-driven transformation select?

A simple way to perform the first step of the transformation of the data is to use data-driven transformation based on the empirical distribution of the variable values. For example for all three variables the 10th percentile, the 50th percentile and the 90th percentile of the variable can be selected as the threshold values for "fully out of the set", the "cross-over point", and "fully in the set", respectively. This applies to the example dataset as follows:

```
data(nca.example)
data <- nca.example
thx1 <- quantile(data[,1], c(0.10,0.50,0.90))
thx2 <- quantile(data[,2], c(0.10,0.50,0.90))
thy <- quantile(data[,3], c(0.10,0.50,0.90))
```

5. How to perform the logistic transformation of the data?

The transformation of the variables with a logistic function is the most commonly used transformation (because it is embedded in Ragin's software "fsQCA"). The logistic transformation with the above thresholds can be performed as follows:

```
x1T <- calibrate(data[,1],type="fuzzy", thresholds=
c(thx1[1], thx1[2], thx1[3]),logistic = TRUE, idm = 0.953)
x2T <- calibrate(data[,2], type="fuzzy", thresholds =
c(thx2[1], thx2[2], thx2[3]), logistic = TRUE, idm =
0.953)
yT <- calibrate(data[,3], type="fuzzy", thresholds =
c(thy[1], thy[2], thy[3]), logistic = TRUE, idm = 0.953)
```

6. How to construct a logistic transformed dataset?

The transformed dataset can be constructed and stored in the working directory as follows:

```
dataT <- cbind (x1T,x2T,yT)
rownames(dataT) <- rownames(data)
colnames(dataT) <- colnames(data)
dataT <- as.data.frame(dataT)
```

7. How to show the results of the logistic transformation of data?

The transformation can be shown on screen as a plot of the membership function for each variable:

```
plot(data[,1], dataT[,1], ylab="Membership score",
xlab="Original score", main="x1T")
plot(data[,2], dataT[,2], ylab="Membership score",
xlab="Original score", main="x2T")
plot(data[,3], dataT[,3], ylab="Membership score",
xlab="Original score", main="yT")
```

8. How to run NCA with the logistic transformed dataset?

Run the NCA with the logistic transformed dataset as follows:

```
nca(dataT,c(1,2),3)
```

```
Effect size(s):
             ce_fdh cr_fdh
Individualism 0.108  0.140
```

```
Risk taking   0.070  0.197
```

The effect of this (and similar) transformations of the original data set is that in the XY scatter plot the observations (cases) are moved from the middle to the corners. Then effect sizes of necessary conditions are reduced because the "empty" zone in the upper left corner of the scatter plot is filled with more cases. The choice of the logistic function is one of the reasons of the move of scores from the middle to the corners. The use of the logistic function is an arbitrary choice, and other membership functions could be selected as well. Normally, with a linear transformation (such as the "standardized" transformation, see below) higher effect sizes and therefore more necessary conditions may be found than with a logistic transformation.

9. How to run NCA with a "standardized" (minimally transformed) dataset?

If the original data are valid, the variables can be minimally transformed to obtain membership scores. Then NCA (and QCA) can be done with a "standardized" dataset. For a "standardized" transformation of a variable the thresholds are selected on the basis of the observed lowest value ("fully out of the set") and observed highest value ("fully in of the set"), with the cross-over point in the middle between these values. Then a linear membership function is selected such that original scores are standardized in the range between 0 and 1, corresponding to membership scores.

10. How to perform the standardized transformation?

```
x1S <- 1-((max(data[,1])- data[,1]) / (max(data[,1])-
min(data[,1])))
x2S <- 1-((max(data[,2])- data[,2]) / (max(data[,2])-
min(data[,2])))
yS <- 1-((max(data[,3])- data[,3]) / (max(data[,3])-
min(data[,3])))
```

11. How to construct a standardized transformed dataset?

```
dataS <- cbind(x1S,x2S,yS)
rownames(dataS) <- rownames(data)
colnames(dataS) <- colnames(data)
dataS <- as.data.frame(dataS)
```

12. How to show the results of the standardized transformation of data?

```
plot(data[,1], dataS[,1], ylab="Membership score",
xlab="Original score", main="x1S")
plot(data[,2], dataS[,2], ylab="Membership score",
xlab="Original score", main="x2S")
plot(data[,3], dataS[,3], ylab="Membership score",
xlab="Original score", main="yS")
```

13. How to compare the standard transformation with the logistic transformation?

```
plot(data[,1], dataS[,1], ylab="Membership score",
xlab="Original score", main="x1")
points(data[,1], dataT[,1])
plot(data[,2], dataS[,2], ylab="Membership score",
xlab="Original score", main="x2")
```

```
points(data[,2], dataT[,2])
plot(data[,3], dataS[,3], ylab="Membership score",
xlab="Original score", main="y")
points(data[,3], dataT[,3])
```

14. How to run NCA with the standardized transformed dataset?

```
nca(dataS,c(1,2),3)
```

```
Effect size(s):
             ce_fdh cr_fdh
Individualism  0.416  0.307
Risk.taking    0.309  0.282
```

15. How to run QCA necessity analysis with the logistic transformed dataset?

For running the necessity analysis in QCA the condition names should be short. For the nca.example: I = Individualism, R=Risk taking, P=Innovation performance. In the QCA output incl. is the necessity consistency level. According to QCA a condition can be considered necessary of the necessity consistency level is at least 0.85.

```
colnames(dataT)<- c("I","R","P")
pofind(dataT, outcome = "P")
```

```
       inclN  RoN    covN
-------------------------
1 ~I   0.501  0.657  0.457
2  I   0.746  0.736  0.655
3 ~R   0.452  0.610  0.394
4  R   0.750  0.773  0.691
-------------------------
```

This output shows consistency levels (incl.) of I and R (as well as for absence of of I ("~I") and R("~R")). With the logistic transformation and a consistency level of 0.85 QCA does not find that I and R are necessary.

16. How to run QCA necessity analysis with the standardized transformed dataset?

```
colnames(dataS)<- c("I","R","P")
pofind(dataS, outcome = "P")
```

```
       inclN  RoN    covN
-------------------------
1 ~I   0.510  0.731  0.412
2  I   0.891  0.532  0.450
3 ~R   0.614  0.627  0.387
4  R   0.840  0.667  0.515
-------------------------
```

With the standard transformation the necessity consistency levels are higher than for logistic transformation and with a threshold of 0.85, QCA finds that I is necessary and R is not necessary.