INTERFACULTY CENTER
DATA PROCESSING & STATISTICS
CORE FACILITY

Data Representation
general principles and pointers

Wilfried Cools & Lara Stas

Compiled May 25, 2020

Current draft aims to introduce researchers to the key ideas in data representation that would help to prepare their data for data analysis.

Our target audience is primarily the research community at VUB / UZ Brussel, those who might apply for data analysis at ICDS in particular.

We invite you to help improve this document by sending us feedback
wilfried.cools@vub.be or anonymously at icds.be/consulting (right side, bottom)

Key message on data representation

In preparation of data analysis, it is wise to think carefully about how to represent your data. The key ideas are listed first, and will be explained and exemplified in more detail throughout current draft.

- represent data so that
  - you and fellow researchers understand it, now but also in the future,
  - statistical algorithms understand it,
  - the gap researcher - algorithm is minimized (efficient processing)
    - allows for straightforward data manipulation, modeling, visualization.

- table formats combine rows and columns in cells:
  - cells contain one and only one piece of information,
  - rows relate cells to a research unit, could be a patient, a mouse, a center, … ,
  - columns relate cells to a property,
  - cells offer information for specific research unit - property combinations.

- ideally, data are TIDY, with meaning appropriately mapped into structure:
  - each row an observation as research unit,
  - each column a variable as property,
  - each cell a value,
  - note: data can be split into multiple tables.

- check data by
  - eye-balling to ensure a correct and unambiguous interpretation of cell values,
  - descriptive analysis to detect anomalies from frequency tables and summary statistics (eg., mean, median, minimum-maximum).

Challenge

Test yourself: create a data file for the following 4 participants (assuming many more), ready for analysis. Read through this draft and if necessary alter your solution.
A possible solution is included at the end.

- Enid Charles, age 43,
  - visual score 16, mathematical score 2.4,
  - suggested methods A and B,
  - performance score at first time point 101 and second time point 105.

- Gertrude Mary Cox, age 34,
  - visual score 26, mathematical score 1.4,
  - suggested methods A,
  - performance score at first time point missing and second time point 115.

- Helen Berg, age 53,
  - visual score 20, mathematical score missing,
  - suggested methods none (not A, nor B, nor C),
  - performance score at first time point 111 and second time point 110.

- Grace Wahba, age 50,
  - visual score 30, mathematical score above cut-off 10,
  - suggested methods A,
  - performance score at first time point 91 and second time point 115.

wilfried.cools@vub.be

Outline

Current draft addresses data representation with the following outline:

- a challenge: it is not always clear how (see above)
- errors and inconveniences
- common problems and solutions

In following drafts, data manipulation, modeling and visualization are considered. Typically, all are more straightforward when data are more tidy.

Errors and inconveniences

To avoid problems and frustration in your data analysis, it may be worthwhile to consider the checklist below. It points at various issues that have been encountered in actual data at ICDS and that are easy to avoid. In general most data offered by researchers whom did not attempt to do their own analysis, or at least the preliminary descriptives, is full with issues like the ones highlighted in this section.
In summary:

- inconsistencies
- ambiguities / incompleteness
- inconveniences for either software or user

Error: inconsistent specification of cell values

When labeling or scoring properties for research units (cells), avoid typo's, inconsistent labeling, inconsistent scoring, . . .

Often observed problems:

- typing errors in values or labels, eg., `man` - `women` - `womem` or `likely` - `likly` - `Likely`,
- inconsistent use of capital letters, eg., `man` - `Man` - `woman`. Most statistical software is case sensitive (eg., R),
- inconsistent use of spaces (_), eg., `man__` - `man` - `_woman` - `woman`,
- inconsistent use of decimal indicators, eg., `4.2` - `5,3` - `5,9`. A comma is often used locally, a dot is used internationally (scientifically),
- inconsistent use of missing value indicators: _ - NA - 99. Software differ in their default, but consistency is key !

Advice: frequency tables often suffice to detect most of these errors, or a summary for numeric values.

Note that the average score for the table on the left appears to be 3.65, do you see what went wrong ?

Error: ambiguous and incomplete specification of cell values

When labeling or scoring properties for research units (cells), avoid ambiguity and incompleteness.

Table 1: inconsistencies

id	gender	score
id1	man	4.2
id2	Man	5,3
id3	man	5,9
id4	woman	3.1
id5	woman	7,2

Table 2: frequencies of gender variable

man	1
Man	1
man	1
woman	2

Often observed problems within cells:

- empty cells not implying missing values
  - eg., those that imply the label above (eg., Excel showcase below with empty field meaning `group 1`),
  - eg., those implying either `missing` or `none`, no answer is different from the answer 0 or "" (eg., `types` variable in ambiguous - incomplete below),
- combined numerical and non-numerical values, eg., `3.9` combined with `>10` (eg., `score` variable in ambiguous - incomplete below),
- combined information within a cell, eg., `A:B`, `A:C`, `B` to signal treatments received (none or A, B, and/or C) (eg., `types` variable in ambiguous - incomplete below).

Each cell should best be fully interpretable on its own, with reference to both row and column only. A codebook, discussed below, serves to alleviate any possible discrepancy between the data representation and the actual data.

Often observed problems combining cells:

- multiple line headers (eg., Excel showcase `blood volume` for both `baseline and after treatment`),
- merged cells (eg., Excel showcase `baseline measurement`).

Inconvenience: use of special characters and numbers

When labeling or scoring, or when specifying a variable name, avoid characters that may not be understood properly. Note that some characters call for specific operations in certain statistical software.

Often observed inconveniences follow from using:

- special characters and spaces (eg., `$, %, #, ", ',`),
- use of names starting with numbers (eg., 1st).

Advice: keep columns with text, not part of the statistical analysis, in a separate file.

Table 3: ambiguous - incomplete

id	types	score
id1	A:B	4.2
id2	A	
id3	B	5.9
id4	A:B	>10
id5		7.2

Table 4: special characters

id	type	score
id1	% use	4.2
id2	% use	5,3
id3	'run'	5,9
id4	'run'	3.1
id5	% use	7,2

Inconvenience: complex and lengthy labels and values

When labeling variables or values, strike a balance between meaningful and simple. This is especially important when requesting help from data analysts who typically program their analysis and often do not understand your line of research. Some analysts may even prefer all values as numeric, (eg., 0 vs. 1) while others prefer short alphanumeric values (eg., male vs female).

Advice: To keep meaningful but long and complex headers, use a second line with simple headers to read in for the analysis. Maybe use `patientID` and `id1` instead of `patient_identifiers_of_first_block` and `patient_number_1`.

Table 5: lengthy - complex

patient_identifiers_of_first_block	my type	%mg rating
patient identity number 1	condition with extra air	4.2 mg/s
patient identity number 2	condition without extra air	5,3 mg/s
patient identity number 3	condition with extra air	5,9 mg/s
patient identity number 4	condition with extra air (stopped early)	3.1 mg/s
patient identity number 5	condition without extra air	7,2 mg/s

Advice: To ensure a correct interpretation, now and later, the researcher could make the following distinction,

- use numbers when values could be interpreted on a continuous scale,
- use text with clear order like `notAgree - neutral - agree`,
- use text postfixed with numbers with unclear order like `r1 - r2 - r3` for ordinal scale not to be used as continuous,
- use text for all remaining labels.

Table 6: appropriate labeling

id	type	intensity	score	rank
id1	black	low	4.2	rnk1
id2	black	medium	5.3	rnk4
id3	red	low	5.9	rnk3
id4	yellow	high	3.1	rnk3
id5	black	low	7.2	rnk2

A codebook could address the relation between labels and their interpretation as well.

Inconvenience: irrelevant data

When starting the analysis, or offering data to third parties, retain only the data of interest for the analysis. Store the remainder of the data in a secure place with an appropriate link.

Advice: remove

- information that could jeopardize GDPR, like names of patients (important),

- comments of participants, and other textual information not relevant for analysis,
- variables that are registered insufficiently, or erroneously,
- variables that are well understood transformations from other variables (eg., averages or log-transformations),
- anything that is not part of the main table, like figures and supporting tables.

Table 7: irrelevant

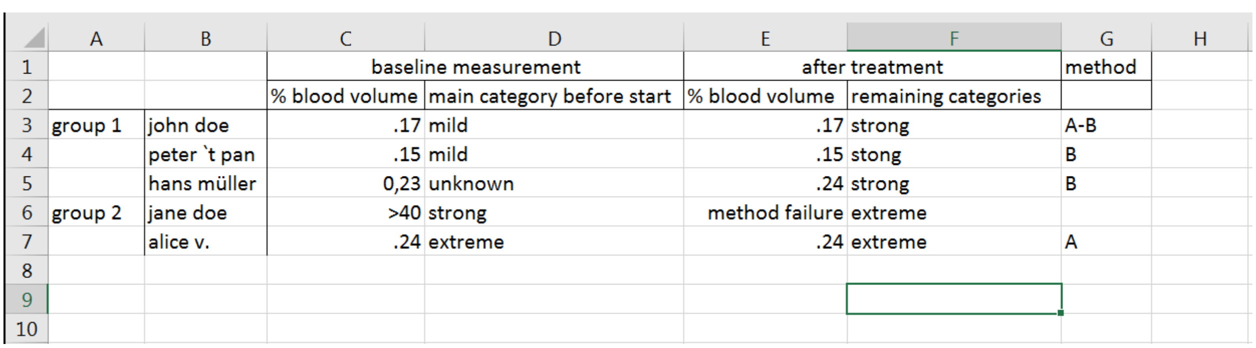
name	score1	score2	sumscore	comments
Enid Charles	3	4	7	some problems at the start
Gertrude Mary Cox	3	3	6	
Helen Berg	4	0	4	patient showed no interest
Grace Wahba	4	4	8	

Error: spreadsheets for human interpretation only

Spreadsheets are convenient for representing data because their base structure is a table, with rows and columns, which you need for most statistical analysis, and because they allow for straightforward manipulations of data.

Manually constructed spreadsheets, Excel or other, unfortunately, promote the use of implicit information rather than the required explicit information. For example, cells are left empty because it is, at least for a human, clear from the context what the value should be (eg., Excel showcase, empty field meaning group 1 or 2).

- incompleteness due to implicit information
- use of merged cells, not understood by algorithms



	A	B	C	D	E	F	G	H
1			baseline measurement		after treatment		method	
2			% blood volume	main category before start	% blood volume	remaining categories		
3	group 1	john doe	.17	mild	.17	strong	A-B	
4		peter `t pan	.15	mild	.15	stong	B	
5		hans müller	0,23	unknown	.24	strong	B	
6	group 2	jane doe	>40	strong	method failure	extreme		
7		alice v.	.24	extreme	.24	extreme	A	
8								
9								
10								

Figure 1: Excel showcase

Excel deserves special attention. Understandably very popular, it often does more than expected and can cause serious problems.

Often observed problems:

- inappropriate cell types (eg., numeric values read in as if they are dates),
- inappropriate dimensions (eg., activated cells outside the data-frame or hidden columns),

Advice: A safe way to store data, once fully ready, could be a tab-delimited text file. While inconvenient to manipulate, risks for unwanted behavior are eliminated. It is straightforward to convert one into the other.

Common problems and solutions

For data analysis data is most often represented in one or more tables. It is repeated that:

- Tables combine rows and columns into cells (see key message):
  - with rows that relate cells within a research unit (eg., a row contains all information about a particular observation),
  - with columns that relate cells to a property (eg., a column contains all information about a particular variable)
  - with cells that contain values which offer one and only one piece of information, combining a research unit and a property.
- Tables for different but related research units are linked by identifiers (eg., table for observation information, table for participant information).

A bad bad exemplary case, using R to turn it around

While it is best to avoid a bad data table from the start, it is in many cases not impossible to convert tables into more appropriate forms.

Purely for illustration purposes, `R` code is included using the `tidyverse` package to show a possible data transformation starting from a bad example turning it into another data representation. In current draft the focus is on data representation, not on changing it. More details on how to manipulate, visualize and model data are offered in future drafts.

Consider this monstrous dataset, showing various features that are common in data offered for analysis.

Table 8: bad bad example

id	young	old	stat	condA_time0	condA_time1	condA_time2	condB_time0	condB_time1	condB_time2	subst
person1	TRUE	FALSE	min	NA	-10	NA	NA	NA	NA	s1,s2
person1	TRUE	FALSE	max	NA	20	NA	NA	NA	NA	s1,s2
person1	TRUE	FALSE	min	NA	NA	NA	NA	NA	0	
person1	TRUE	FALSE	max	NA	NA	NA	NA	NA	25	
person2	FALSE	TRUE	min	NA	NA	NA	5	NA	NA	s2
person2	FALSE	TRUE	max	NA	NA	NA	15	NA	NA	s2
person2	FALSE	TRUE	min	NA	NA	0	NA	NA	NA	s1
person2	FALSE	TRUE	max	NA	NA	10	NA	NA	NA	s1

Apparently, substances (`subst`) can be `s1`, `s2`, both or none. So, having `s1,s2` is partly overlapping with `s1`, but how does the algorithm know ? Lets turn this multiple selection item into multiple columns. Apparently, young and old are two variables, which makes no sense because you are either young or old, so lets remove one of them.

```
badExample <- tBadBad %>%
    mutate(s1=ifelse(grepl('s1', subst),T,F),s2=ifelse(grepl('s2',subst),T,F)) %>%
    select(-subst,-old)
```

Table 9: split combined information

id	young	stat	condA_time0	condA_time1	condA_time2	condB_time0	condB_time1	condB_time2	s1	s2
person1	TRUE	min	NA	-10	NA	NA	NA	NA	TRUE	TRUE
person1	TRUE	max	NA	20	NA	NA	NA	NA	TRUE	TRUE
person1	TRUE	min	NA	NA	NA	NA	NA	0	FALSE	FALSE
person1	TRUE	max	NA	NA	NA	NA	NA	25	FALSE	FALSE
person2	FALSE	min	NA	NA	NA	5	NA	NA	FALSE	TRUE
person2	FALSE	max	NA	NA	NA	15	NA	NA	FALSE	TRUE
person2	FALSE	min	NA	NA	0	NA	NA	NA	TRUE	FALSE
person2	FALSE	max	NA	NA	10	NA	NA	NA	TRUE	FALSE

Apparently, various columns contain variable values (consider 4th to 9th column). As the variable names suggest, observations are obtained under certain conditions, A or B, and at various time points, time 0, 1 or 2. In this example `example condA_time1` partly overlaps with `condA_time2` with which it shares a method, and partly overlaps with `condB_time1` with which it shares a time point. Let's turn these columns into values first, and at the same time simply ignore the missing values.

Observe that the names of the columns turn into values in a column names `messystuff`, making the dataframe less wide and more long.

```
badExample <- badExample %>%
    pivot_longer(names_to="messyStuff",values_to="scores",-c(id,young,stat,s1,s2)) %>%
    filter(!is.na(scores))
```

Table 10: from wide to long form

id	young	stat	s1	s2	messyStuff	scores
person1	TRUE	min	TRUE	TRUE	condA_time1	-10
person1	TRUE	max	TRUE	TRUE	condA_time1	20
person1	TRUE	min	FALSE	FALSE	condB_time2	0
person1	TRUE	max	FALSE	FALSE	condB_time2	25
person2	FALSE	min	FALSE	TRUE	condB_time0	5
person2	FALSE	max	FALSE	TRUE	condB_time0	15
person2	FALSE	min	TRUE	FALSE	condA_time2	0
person2	FALSE	max	TRUE	FALSE	condA_time2	10

The new column still combines two types of information, condition and time. The column should be split into two columns.

```
badExample <- badExample %>%
    separate(messyStuff,c('cond','time'))
```

Much better. A last issue here is that the minimum and maximum could be variables and not values. No hard rules here, but often it is intuitively clear. So, let's turn these values into variables to represent two types of observation.

```
goodExample <- badExample %>%
    pivot_wider(names_from=stat,values_from=scores)
```

Table 11: separate combined information

id	young	stat	s1	s2	cond	time	scores
person1	TRUE	min	TRUE	TRUE	condA	time1	-10
person1	TRUE	max	TRUE	TRUE	condA	time1	20
person1	TRUE	min	FALSE	FALSE	condB	time2	0
person1	TRUE	max	FALSE	FALSE	condB	time2	25
person2	FALSE	min	FALSE	TRUE	condB	time0	5
person2	FALSE	max	FALSE	TRUE	condB	time0	15
person2	FALSE	min	TRUE	FALSE	condA	time2	0
person2	FALSE	max	TRUE	FALSE	condA	time2	10

Table 12: from long to wide

id	young	s1	s2	cond	time	min	max
person1	TRUE	TRUE	TRUE	condA	time1	-10	20
person1	TRUE	FALSE	FALSE	condB	time2	0	25
person2	FALSE	FALSE	TRUE	condB	time0	5	15
person2	FALSE	TRUE	FALSE	condA	time2	0	10

While not convenient here, if there are many variables it may be interesting to split the table into different tables. Each table is research unit specific. So, let's create a persons file and an observations file, and merge them together again afterwards.

```
persons <- goodExample %>% select(id,young) %>% distinct()
observations <- goodExample %>% select(-young)
combinedAgain <- observations %>% full_join(persons)
```

Table 13: simple persons table

id	young
person1	TRUE
person2	FALSE

Table 14: simple observations table

id	s1	s2	cond	time	min	max
person1	TRUE	TRUE	condA	time1	-10	20
person1	FALSE	FALSE	condB	time2	0	25
person2	FALSE	TRUE	condB	time0	5	15
person2	TRUE	FALSE	condA	time2	0	10

Table 15: merged again using person as identifier

id	s1	s2	cond	time	min	max	young
person1	TRUE	TRUE	condA	time1	-10	20	TRUE
person1	FALSE	FALSE	condB	time2	0	25	TRUE
person2	FALSE	TRUE	condB	time0	5	15	FALSE
person2	TRUE	FALSE	condA	time2	0	10	FALSE

Various issues were highlighted, and will be discussed in more detail below.

- The two most important points are
  - a long form (univariate) data representation is more flexible compared to a wide form (multivariate) one
  - additional columns can help isolate information in cells

Long form representation

If within a research unit several scores are obtained, they can be represented within a row but often it is better or even necessary to unfold them into multiple rows that are identified with an indicator variable.

For example, consider a repeated measurements datafile, with multiple observations for each participant. The observations within a patient could be represented on a patient specific row (wide) with an identifier column for the participant, or one below the other covering several rows (long) with an indicator variable for both the participant (includes multiple rows) and the time of observation.

Table 16: simple wide form

id	s1	s2
id1	7	6
id2	2	3
id3	4	3
id4	6	7
id5	8	7

Table 17: simple long form

id	type	score
id1	s1	7
id1	s2	6
id2	s1	2
id2	s2	3
id3	s1	4
id3	s2	3
id4	s1	6
id4	s2	7
id5	s1	8
id5	s2	7

Note: the switch between both representations is easy. In Excel use `pivot` tables, in `R` many functions exist, for example the pivot_wider or pivot_longer in `tidyr`. Knowing how to transform data between wide and long form is very convenient and worth the effort learning about it.

Research unit specific tables

It may be appropriate to split up a table into different tables, as is done with relational databases, in order to combine all information in research unit specific tables. Different tables can be combined when of interest using key variables. This is particularly interesting as datafiles get bigger and as values are constant within blocks.

For example, a datafile could be split into a person datafile and an observation datafile. A person file only consists of person related properties that are constant for a particular person. An observation file consists of observation related properties that are constant for a particular observation. Note that the person providing the observation is represented once per observation.

For example, an additional table could be used to add item specific information about what the correct response is, how to score a particular response, or whether a score should be inverted when using it to summarize over an underlying scale. The main observation file includes the actual responses, not the scores.

Note: to split up and merge tables is easy. In Excel use `merge`, in `R` use join in `dplyr` for example. Knowing how to split and combine data can be convenient.

Table 18: information combined

id	type	score	gender
id1	s1	7	M
id1	s2	6	M
id2	s1	2	M
id2	s2	3	M
id3	s1	4	F
id3	s2	3	F
id4	s1	6	M
id4	s2	7	M
id5	s1	8	F
id5	s2	7	F

Table 19: a subset

id	gender
id1	M
id2	M
id3	F
id4	M
id5	F

Table 20: the other subset

id	type	score
id1	s1	7
id1	s2	6
id2	s1	2
id2	s2	3
id3	s1	4
id3	s2	3
id4	s1	6
id4	s2	7
id5	s1	8
id5	s2	7

Possible but never observed responses

A full data representation not only considers the actual data but also the possible data. The way to include this type of information is with additional tables that specify all possible outcomes. A codebook can also be used to provide this information in textual format.

For example, consider a question for which the response option `fully agree` was never selected, a separate table could include that option nevertheless.

For example, consider a question for which selecting none of the alternatives is a viable response, a separate table could include this.

Table 21: response file

item	option	quality
i1	o1	wrong
i1	o2	correct
i1	o3	wrong
i2	o1	correct
i2	o2	wrong
i2	o3	wrong

Table 22: item responses

id	item	response
id1	i1	o1
id1	i2	o1
id2	i1	o2
id2	i2	o1
id3	i1	o2
id3	i2	o3

Note: it is possible to add option specific information, for example a score or indication of correctness. This has the advantage that the score can easily be changed and the used scores are easy to determine.

Disentangling information: different situations

A main point of interest is to include only one piece of information within a cell, unambiguously interpretable. Typically this would involve brining in additional columns.

Different types of missingness

It could be of interest to distinguish between a missing value due to non-response, and a missing value by design. A full data registration can include an extra column for example, to signal for each missing value how to interpret it. A codebook can be an alternative in which codes are specified for different types of missing data.

Table 23: labels with numbers

id	score
id1	7
id2	not applicable
id3	4
id4	not responded
id5	8

Table 24: disentangled

id	score	typeNA
id1	7	
id2		irrelevant
id3	4	
id4		nonResponse
id5	8	

Numbers and ranges

Variables sometimes combine both values and ranges of values. A possible full data registration adds a column to identify the ranges, so that the original column only includes values.

Table 25: labels with numbers

id	score
id1	7
id2	2
id3	4
id4	>10
id5	8

Table 26: disentangled

id	score	lwrBound
id1	7	NA
id2	2	NA
id3	4	NA
id4	NA	10
id5	8	NA

Note: the original information is still available, but each variable contains only one type of information and cells have only numbers or (implied) ranges.

Collections

Values sometimes partially overlap so that they do not offer a single piece of information. A possible full data registration adds columns to isolate the different pieces of information.

Table 27: combined information

id	score
id1	A:B
id2	A
id3	
id4	B
id5	A:B

Table 28: disentangled

id	A	B
id1	TRUE	TRUE
id2	TRUE	FALSE
id3	FALSE	FALSE
id4	FALSE	TRUE
id5	TRUE	TRUE

Table 29: adding order information

id	A	B
id1	1	2
id2	1	NA
id3	NA	NA
id4	NA	1
id5	1	2

Note that this way the combination of A and B is correctly considered as a combination of two constituting parts that were neither of them necessary. The original information is again easily retrieved from the available variables.

Note: the original information is still available, but each variable contains only one type of information and cells have only numbers or boolean values.

Codebook

It is best to let data be as self-explanatory as possible and ready for automated processing.
The information that is impossible or very impractical to include in the actual table(s) should be explained in a codebook. A codebook explains the discrepancy between the data as represented and its meaning.

- a codebook could include information on
  - meaning of variable / intended use
  - measurement scale
    - range / set of possible values
  - types of missingness and its coding
  - how data are collected
    - timing
    - tools of observations
  - . . .

wilfried.cools@vub.be

Solution

A possible solution to the challenge above is presented here. Other more simple solutions are possible.

Table 30: persons

idnr	age	vis	math	math10	A	B	C
1	43	16	2.4	FALSE	TRUE	TRUE	FALSE
2	34	26	1.4	FALSE	TRUE	FALSE	FALSE
3	53	20	NA	NA	FALSE	FALSE	FALSE
4	50	30	NA	TRUE	TRUE	FALSE	FALSE

Table 31: ids

idnr	id
1	Enid Charles
2	Gertrude Mary Cox
3	Helen Berg
4	Grace Wahba

Table 32: observations

idnr	time	score
1	0	101
1	1	105
2	0	NA
2	1	115
3	0	111
3	1	110
4	0	91
4	1	115

The logged file, with observations, and the persons file, with person specific observation excluding identifiers can be combined, especially if the data is not too large.

Table 33: a possible solution

idnr	time	score	age	vis	math	math10	A	B	C
1	0	101	43	16	2.4	FALSE	TRUE	TRUE	FALSE
1	1	105	43	16	2.4	FALSE	TRUE	TRUE	FALSE
2	0	NA	34	26	1.4	FALSE	TRUE	FALSE	FALSE
2	1	115	34	26	1.4	FALSE	TRUE	FALSE	FALSE
3	0	111	53	20	NA	NA	FALSE	FALSE	FALSE
3	1	110	53	20	NA	NA	FALSE	FALSE	FALSE
4	0	91	50	30	NA	TRUE	TRUE	FALSE	FALSE
4	1	115	50	30	NA	TRUE	TRUE	FALSE	FALSE

Methodological and statistical support to help make a difference

- ICDS provides complementary support in methodology and statistics to our research community, for both individual researchers and research groups, in order to get the best out of them
- ICDS aims to address all questions related to quantitative research, and to further enhance the quality of both the research and how it is communicated

website: https://www.icds.be/ includes information on who we serve, and how

booking: https://www.icds.be/consulting/ for individual consultations