

Package: elicitr (via r-universe)

October 14, 2026

Title Processing and Aggregation of Expert Elicitation Data

Version 0.2.0

Description Provides tools to import, validate, visualise and aggregate expert estimates for continuous and categorical variables. Analysis of continuous variables supports one-, three- and four-point elicitations, confidence-based rescaling and sampling from PERT. Analysis of categorical variables supports Dirichlet distributions, and weighted aggregation. You provide the data, and elicitr transforms them into easily readable graphics. Implements workflows adapted from the IDEA protocol described by Hemming et al. (2018) [doi:10.1111/2041-210X.12857](https://doi.org/10.1111/2041-210X.12857) and categorical elicitation described by Vernet et al. (2024) [doi:10.1007/s10530-024-03341-2](https://doi.org/10.1007/s10530-024-03341-2).

License GPL (>= 3)

URL <https://crewdecisions.github.io/elicitr/>

BugReports <https://github.com/CREWdecisions/elicitr/issues>

Encoding UTF-8

Language en-GB

Roxygen list(markdown = TRUE)

Imports cli, digest, dplyr (>= 1.1.0), extraDistr, ggplot2 (>= 4.0.0), googlesheets4, mc2d, miceadds, openxlsx, rlang, scales, stats, stringr (>= 1.5.0), tibble, tidyr, tools, utils, ggbeeswarm, lme4, stringi

Suggests knitr, lifecycle, lintr, randomNames, rmarkdown, Surrogate, testthat (>= 3.0.0), withr (>= 3.0.1)

Config/testthat/edition 3

Depends R (>= 4.2.0)

LazyData true

VignetteBuilder knitr

Config/roxygen2/version 8.1.0

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev libx11-dev zlib1g-dev
Repository https://crewdecisions.r-universe.dev
Date/Publication 2026-10-14 17:13:19 UTC
RemoteUrl https://github.com/CREWdecisions/elicitr
RemoteRef HEAD
RemoteSha 0c4a6ead165e48958d30f0d46aac291e6e7e6a30

Contents

cat_add_data	2
cat_data	6
cat_get_data	7
cat_sample_data	8
cat_start	10
cont_add_data	12
cont_data	15
cont_get_data	16
cont_sample_data	18
cont_start	21
elic_cat	23
elic_cont	24
plot.cat_sample	24
plot.cont_sample	26
plot.elic_cont	28
summary.cat_sample	31
summary.cont_sample	32
Index	34

cat_add_data	<i>Add data</i>
--------------	-----------------

Description

[Stable]
cat_add_data() adds data to an [elic_cat](#) object from different sources.

Usage

```
cat_add_data(  
  x,  
  data_source,  
  topic,  
  ...,  
  sep = ",",
```

```

    sheet = 1,
    verbose = TRUE,
    anonymise = TRUE
  )

```

Arguments

x	an object of class <code>elic_cat</code> .
data_source	either a <code>data.frame</code> or <code>tibble</code> , a string with the path to a <i>csv</i> or <i>xlsx</i> file, or anything accepted by the <code>read_sheet</code> function.
topic	character string that indicates the mechanism to which the data belongs.
...	Unused arguments, included only for future extensions of the function.
sep	character used as field separator, used only when <code>data_source</code> is a path to a <i>csv</i> file.
sheet	integer or character to select the sheet. The sheet can be referenced by its position with a number or by its name with a string. Used only when <code>data_source</code> is a path to a <i>xlsx</i> file or when data are imported from Google Sheets.
verbose	logical, if <code>TRUE</code> it prints informative messages.
anonymise	logical, if <code>TRUE</code> expert names are anonymised before adding the data to the <code>elic_cat</code> object.

Value

The provided object of class `elic_cat` updated with the data.

Data format

For each topic, data are expected to have five columns, built as follows:

- The first column of the data should hold the names of the experts. The name of each expert should be repeated as many times as the number of categories and options. (i.e. each expert should appear *number of categories · number of options* times).
- The second column should hold the names of the options considered in the study. The name of each option should be repeated as many times as the number of categories considered. If you do not use multiple options in your study, please input 1 option for all elements.
- The third column should be the names of the categories considered in the elicitation. Each block of categories should be repeated as many times as the number of options considered.
- The fourth column should be the experts confidence in their own estimates (given in percent). Experts should estimate how confident they are in their estimates for each block of categories and for each option. Therefore, expert confidence estimates should be repeated as many times as the number of categories of impact considered for each option.
- The final column should be the estimates of each expert for each option and category. These estimates should sum up to 1 (probabilities) (or 100 (percentages)) for each expert and option.

The name of the columns is not important, `cat_add_data()` will overwrite them according to the following convention:

The first column will be renamed `id`, the second column `option`, the third column `category`, the fourth column `confidence`, and the fifth column `estimate`.

Here is an example of data correctly formatted for an elicitation with five categories and two options (only one expert is shown):

name	option	category	confidence	estimate
expert 1	option 1	category 1	55	0.08
expert 1	option 1	category 2	55	0
expert 1	option 1	category 3	55	0.84
expert 1	option 1	category 4	55	0.02
expert 1	option 1	category 5	55	0.06
expert 1	option 2	category 1	75	0.02
expert 1	option 2	category 2	75	0.11
expert 1	option 2	category 3	75	0.19
expert 1	option 2	category 4	75	0.02
expert 1	option 2	category 5	75	0.66

Data cleaning

When data are added to the `elic_cat` object, first names are standardised by converting capital letters to lower case, and by removing any whitespaces and punctuation. Then, data are anonymised by converting names to short sha1 hashes. In this way, sensible information collected during the elicitation process never reaches the `elic_cat` object.

Author(s)

Sergio Vignali and Maude Vernet

See Also

Other cat data helpers: `cat_get_data()`, `cat_sample_data()`, `cat_start()`, `summary.cat_sample()`

Examples

```
# Create the elic_cat object for an elicitation process with three topics,
# four options, five categories and a maximum of six experts per topic
my_topics <- c("topic_1", "topic_2", "topic_3")
my_categories <- c("category_1", "category_2", "category_3",
                  "category_4", "category_5")
my_options <- c("option_1", "option_2", "option_3", "option_4")
x <- cat_start(topics = my_topics,
               options = my_options,
               categories = my_categories,
               experts = 6)

# Add data for the three topics from a data.frame. Notice that the three
# commands can be piped
```

```

my_elicit <- cat_add_data(x,
                          data_source = topic_1,
                          topic = "topic_1") |>
  cat_add_data(data_source = topic_2, topic = "topic_2") |>
  cat_add_data(data_source = topic_3, topic = "topic_3")
my_elicit

# Add data for the first and second round from a csv file
files <- list.files(path = system.file("extdata", package = "elicitr"),
                    pattern = "topic_",
                    full.names = TRUE)
my_elicit <- cat_add_data(x,
                          data_source = files[1],
                          topic = "topic_1") |>
  cat_add_data(data_source = files[2], topic = "topic_2") |>
  cat_add_data(data_source = files[3], topic = "topic_3")
my_elicit

# Add data for the first and second round from a xlsx file with three sheets
file <- list.files(path = system.file("extdata", package = "elicitr"),
                  pattern = "topics",
                  full.names = TRUE)
# Using the sheet index
my_elicit <- cat_add_data(x,
                          data_source = file,
                          sheet = 1,
                          topic = "topic_1") |>
  cat_add_data(data_source = file,
                sheet = 2,
                topic = "topic_2") |>
  cat_add_data(data_source = file,
                sheet = 3,
                topic = "topic_3")
my_elicit
# Using the sheet name
my_elicit <- cat_add_data(x,
                          data_source = file,
                          sheet = "Topic 1",
                          topic = "topic_1") |>
  cat_add_data(data_source = file,
                sheet = "Topic 2",
                topic = "topic_2") |>
  cat_add_data(data_source = file,
                sheet = "Topic 3",
                topic = "topic_3")
my_elicit

# Add data for the first and second round from Google Sheets
googlesheets4::gs4_deauth()
gs <- "18VHeHB89P1s-6banaVoq0P-ggFmQZYx-z_31nMffAb8"

# Using the sheet index

```

```

my_elicit <- cat_add_data(x,
                          data_source = gs,
                          sheet = 1,
                          topic = "topic_1") |>
cat_add_data(data_source = gs,
              sheet = 2,
              topic = "topic_2") |>
cat_add_data(data_source = gs,
              sheet = 3,
              topic = "topic_3")
my_elicit

# (You can also do this using the sheet name)
## Not run:
my_elicit <- cat_add_data(x, data_source = gs,
                          sheet = "Topic 1",
                          topic = "topic_1") |>
cat_add_data(data_source = gs,
              sheet = "Topic 2",
              topic = "topic_2") |>
cat_add_data(data_source = gs,
              sheet = "Topic 3",
              topic = "topic_3")
my_elicit

## End(Not run)

```

cat_data

Elicitation data for categorical variables

Description

Simulated data for three topics investigated in an elicitation process.

Usage

```

topic_1

topic_2

topic_3

```

Format

A data frame with:

expert Name of the experts (randomly generated).

category The name of the categories. There are 5 different categories per option.

option The name of the options. Topic 1 and 2 have 4 options, while topic 3 has 3 options.

confidence The confidence of the experts given in percentages. One confidence value is given for each option.

estimate Expert estimates given in probabilities.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 120 rows and 5 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 100 rows and 5 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 90 rows and 5 columns.

Source

Randomly generated numbers and names.

`cat_get_data`*Get data*

Description

[Stable]

Get data from an [elic_cat](#) object.

Usage

```
cat_get_data(x, topic, ..., option = "all")
```

Arguments

<code>x</code>	an object of class elic_cat .
<code>topic</code>	character string that indicates the mechanism to which the data belongs.
<code>...</code>	Unused arguments, included only for future extensions of the function.
<code>option</code>	character string with the name of the option or character vector with the options that you want to extract from the data. Use <code>all</code> for all options.

Value

A [tibble](#) with the extracted data.

Author(s)

Sergio Vignali and Maude Vernet

See Also

Other cat data helpers: [cat_add_data\(\)](#), [cat_sample_data\(\)](#), [cat_start\(\)](#), [summary.cat_sample\(\)](#)

Examples

```
# Create the elic_cat object for an elicitation process with three topics,
# four options, five categories and a maximum of six experts per topic
my_topics <- c("topic_1", "topic_2", "topic_3")
my_options <- c("option_1", "option_2", "option_3", "option_4")
my_categories <- c("category_1", "category_2", "category_3",
                  "category_4", "category_5")
my_elicit <- cat_start(topics = my_topics,
                      options = my_options,
                      categories = my_categories,
                      experts = 6) |>
  cat_add_data(data_source = topic_1, topic = "topic_1") |>
  cat_add_data(data_source = topic_2, topic = "topic_2") |>
  cat_add_data(data_source = topic_3, topic = "topic_3")

# Get all data from Topic 1
cat_get_data(my_elicit, topic = "topic_1")

# Get data by option name----
# Get data for option_1 from Topic 2
cat_get_data(my_elicit, topic = "topic_2", option = "option_1")

# Get data for option_1 and option_3 from Topic 3
cat_get_data(my_elicit,
             topic = "topic_3",
             option = c("option_1", "option_3"))
```

cat_sample_data	<i>Sample categorical data</i>
-----------------	--------------------------------

Description

[Stable]

cat_sample_data() samples data based on expert estimates stored in the [elic_cat](#) object.

Usage

```
cat_sample_data(
  x,
  method,
  topic,
  ...,
  n_votes = 100,
  option = "all",
  verbose = TRUE
)
```

Arguments

x	an object of class <code>elic_cat</code> .
method	character string with the name of the method to sample the data. The available methods are: <i>unweighted</i> and <i>weighted</i> , see Methods below.
topic	character string that indicates the mechanism to which the data belongs.
...	Unused arguments, included only for future extensions of the function.
n_votes	numeric indicating the number of votes to consider.
option	character string with the name of the option or character vector with the options that you want to extract from the data. Use <code>all</code> for all options.
verbose	logical, if TRUE it prints informative messages.

Details

Weighted allocations distribute the full budget among answering experts. Thus, the number of row in the sampled data will $n_experts * n_votes$. If an expert doesn't provide data for the sampled variable, a placeholder row is added. In such case, row counts can exceed $n_experts * n_votes$.

Value

An `tibble` with the sampled data. This object has the additional class `cat_sample` used to implement the plotting method.

Methods

Two methods are implemented. These methods are explained in Vernet et al. (2024), see references below.

- *unweighted*: This method samples data based on the expert estimates without accounting for their confidence. Values are sampled from a Dirichlet distribution using the expert estimates as parameters. When only one estimate is provided, i.e. 100 % for one category, the method assigns 100 % to all votes for this category.
- *weighted*: This method samples data based on the expert estimates accounting for their confidence. The confidence is used to weight the number of votes assigned to each expert. The method samples data from a Dirichlet distribution using the expert estimates as parameters. When only one estimate is provided, i.e. 100 % for one category, the method assigns 100 % to all votes for this category.

Author(s)

Sergio Vignali and Maude Vernet

References

Vernet, M., Trask, A.E., Andrews, C.E., Ewen, J.E., Medina, S., Moehrensclager, A. & Canessa, S. (2024) Assessing invasion risks using EICAT-based expert elicitation: application to a conservation translocation. *Biological Invasions*, 26(8), 2707–2721. <https://doi.org/10.1007/s10530-024-03341-2>

See Also

Other cat data helpers: [cat_add_data\(\)](#), [cat_get_data\(\)](#), [cat_start\(\)](#), [summary.cat_sample\(\)](#)

Examples

```
# Create the elic_cat object for an elicitation process with three topics,
# four options, five categories and a maximum of six experts per topic
my_categories <- c("category_1", "category_2", "category_3",
                  "category_4", "category_5")
my_options <- c("option_1", "option_2", "option_3", "option_4")
my_topics <- c("topic_1", "topic_2", "topic_3")
my_elicit <- cat_start(categories = my_categories,
                      options = my_options,
                      experts = 6,
                      topics = my_topics) |>
  cat_add_data(data_source = topic_1, topic = "topic_1") |>
  cat_add_data(data_source = topic_2, topic = "topic_2") |>
  cat_add_data(data_source = topic_3, topic = "topic_3")

# Sample data from Topic 1 for all options using the unweighted method
samp <- cat_sample_data(my_elicit,
                       method = "unweighted",
                       topic = "topic_1")

# Sample data from Topic 2 for option 1 and 3 using the weighted method
samp <- cat_sample_data(my_elicit,
                       method = "weighted",
                       topic = "topic_2",
                       option = c("option_1", "option_3"))
```

cat_start

Start elicitation

Description**[Stable]**

`cat_start()` initialises an [elic_cat](#) object which stores important metadata for the data collected during the elicitation process of categorical data.

Usage

```
cat_start(
  topics,
  options,
  categories,
  experts,
  ...,
  title = "Elicitation",
  verbose = TRUE
)
```

Arguments

topics	character vector with the names of the topics.
options	character vector with the names of all options investigated. See Options for more.
categories	character vector with the names of the categories. See Categories for more.
experts	numeric, indicating the maximum number of experts participating in the elicitation process for one topic. See Experts for more.
...	Unused arguments, included only for future extensions of the function.
title	character, used to bind a name to the object.
verbose	logical, if TRUE it prints informative messages.

Value

An object of class `elic_cat` binding metadata related to the elicitation process. These metadata are used by other functions to validate the correctness of the provided data.

Experts

The expert parameter is a number indicating the maximum number of experts taking part in the elicitation of one of the investigated topics. The number and IDs of experts can differ between the topics.

Options

The option parameter is a character vector containing the names of all the options investigated in the elicitation. However, not all options have to be investigated in every topic. If you do not use multiple options in your study, please still input 1 option which will apply to all topics, categories and experts (e.g., `options = "none"`).

Categories

Categories are inherited between topics. A minimum of two categories are needed. If only one category is investigated, please refer to the functions for the elicitation of continuous data (e.g. `cont_start`).

Author(s)

Sergio Vignali and Maude Vernet

References

Hemming, V., Burgman, M. A., Hanea, A. M., McBride, M. F., & Wintle, B. C. (2018). A practical guide to structured expert elicitation using the IDEA protocol. *Methods in Ecology and Evolution*, 9(1), 169–180. <https://doi.org/10.1111/2041-210X.12857> Vernet, M., Trask, A.E., Andrews, C.E., Ewen, J.E., Medina, S., Moehrenschrager, A. & Canessa, S. (2024) Assessing invasion risks using EICAT-based expert elicitation: application to a conservation translocation. *Biological Invasions*, 26(8), 2707–2721. <https://doi.org/10.1007/s10530-024-03341-2>

See Also

Other cat data helpers: [cat_add_data\(\)](#), [cat_get_data\(\)](#), [cat_sample_data\(\)](#), [summary.cat_sample\(\)](#)

Examples

```
# Create the elic_cat object for an elicitation process over 2 topics, 3
# options, 3 categories per options, and a maximum number of 8 experts per
# topic
my_categories <- c("category_1", "category_2", "category_3")
my_elicit <- cat_start(topics = c("topic_1", "topic_2"),
                      options = c("option_1", "option_2", "option_3"),
                      categories = my_categories,
                      experts = 8)

my_elicit

# A title can be added to bind a name to the object:
my_elicit <- cat_start(topics = c("topic_1", "topic_2"),
                      options = c("option_1", "option_2", "option_3"),
                      categories = my_categories,
                      experts = 8,
                      title = "My elicitation")

my_elicit
```

cont_add_data

Add data

Description

[Stable]

cont_add_data() adds data to an [elic_cont](#) object from different sources.

Usage

```
cont_add_data(
  x,
  data_source,
  round,
  ...,
  sep = ",",
  sheet = 1,
  overwrite = FALSE,
  verbose = TRUE,
  anonymise = TRUE
)
```

Arguments

x	an object of class <code>elic_cont</code> .
data_source	either a <code>data.frame</code> or <code>tibble</code> , a string with the path to a <i>csv</i> or <i>xlsx</i> file, or anything accepted by the <code>read_sheet</code> function.
round	integer indicating if the data belongs to the first or second elicitation round.
...	Unused arguments, included only for future extensions of the function.
sep	character used as field separator, used only when <code>data_source</code> is a path to a <i>csv</i> file.
sheet	integer or character to select the sheet. The sheet can be referenced by its position with a number or by its name with a string. Used only when <code>data_source</code> is a path to a <i>xlsx</i> file or when data are imported from <i>Google Sheets</i> .
overwrite	logical, whether to overwrite existing data already added to the <code>elic_cont</code> object.
verbose	logical, if TRUE it prints informative messages.
anonymise	logical, if TRUE expert names are anonymised before adding the data to the <code>elic_cont</code> object.

Value

The provided object of class `elic_cont` updated with the data.

Data format

Data are expected to have the name of the expert always as first column. The only exception is for data coming from *Google Sheet* which can have an additional column with a timestamp. This column is automatically removed before the data are added to the `elic_cont` object (see "Data cleaning"). After the name there should be one or more blocks which follow the specifications below:

One point elicitation:

- `var_best`: best estimate for the variable

Three points elicitation:

- `var_min`: minimum estimate for the variable
- `var_max`: maximum estimate for the variable
- `var_best`: best estimate for the variable

Four points elicitation:

- `var_min`: minimum estimate for the variable
- `var_max`: maximum estimate for the variable
- `var_best`: best estimate for the variable
- `var_conf`: confidence for the estimate

The column with names is unique, the other columns are a block and can be repeated for each variable.

Column and variable names

Moreover, the name of the columns is not important, `cont_add_data()` will overwrite it according to the following convention:

variable_name_suffix

with *suffix* being one of *min*, *max*, *best*, or *conf*. The information to build the column names is taken from the metadata available in the `elic_cont` object.

`var_conf`, given as percentage, can be any number in the range (50, 100]. Any value smaller or equal to 50 would imply that the accuracy of the estimates is only due to chance).

NAs

It is possible that some experts did not provide estimates for all variables. In this case, the missing values should be recorded as NA. However, experts should always answer completely or not at all. If estimates are only partially given for a variable (example *min* and *max* provided but not *best* for 3p elicitation), an error is raised.

Data cleaning

When data are added to the `elic_cont` object, first names are standardised by converting capital letters to lower case, and by removing any whitespaces and punctuation. Then, data are anonymised by converting names to short sha1 hashes. In this way, sensible information collected during the elicitation process never reaches the `elic_cont` object. For three and four points elicitation processes, the order of the values is checked for each expert. If it is not *min-max-best*, the values are swapped accordingly and an informative warn is raised.

If the data are imported from *Google Sheets*, `cont_add_data()` performs additional data cleaning operations. This is relevant when data are collected with Google Forms because, for example, there could be multiple submission by the same expert or a different decimal separator could be used. When data are collected with Google Form, a column with the date and time is recorded. First, the function checks for multiple submissions and if present, only the last submission is retained. Second, the function removes the column with the timestamp. Then it checks for consistency of the decimal separator, i.e. commas , are replaced with periods .. Finally, all columns but the first one (which contains the names) are forced to numeric.

Author(s)

Sergio Vignali and Maude Vernet

See Also

Other cont data helpers: `cont_get_data()`, `cont_sample_data()`, `cont_start()`, `summary.cont_sample()`

Examples

```
# Create the elic_cont object for an elicitation process that estimates 3
# variables, the first for a one point estimation of a positive integer, the
# second for three points estimation of a negative real, and the last for a
# four point estimation of a probability
x <- cont_start(var_names = c("var1", "var2", "var3"),
```

```

var_types = "ZNp",
elic_types = "134",
experts = 6)

# Add data for the first and second round from a data.frame. Notice that the
# two commands can be piped
my_elicit <- cont_add_data(x, data_source = round_1, round = 1) |>
  cont_add_data(data_source = round_2, round = 2)
my_elicit

# Add data for the first and second round from a csv file
files <- list.files(path = system.file("extdata", package = "elicitr"),
  pattern = "round_",
  full.names = TRUE)
my_elicit <- cont_add_data(x, data_source = files[1], round = 1) |>
  cont_add_data(data_source = files[2], round = 2)
my_elicit

# Add data for the first and second round from a xlsx file with two sheets
file <- list.files(path = system.file("extdata", package = "elicitr"),
  pattern = "rounds",
  full.names = TRUE)
# Using the sheet index
my_elicit <- cont_add_data(x, data_source = file, sheet = 1, round = 1) |>
  cont_add_data(data_source = file, sheet = 2, round = 2)
my_elicit
# Using the sheet name
my_elicit <- cont_add_data(x, data_source = file,
  sheet = "Round 1", round = 1) |>
  cont_add_data(data_source = file, sheet = "Round 2", round = 2)
my_elicit

# Add data for the first and second round from Google Sheets
googlesheets4::gs4_deauth()
gs1 <- "12lGIPa-jJ0h3f0gUDaERmkf04pVpPu9i8S1oL2jAdqc"
gs2 <- "1wImcfJYnC9a423jlxZiU_BFKXZpTZ7AIsZSxFtEsBQw"
my_elicit <- cont_add_data(x, data_source = gs1, round = 1) |>
  cont_add_data(data_source = gs2, round = 2)
my_elicit

```

cont_data

Elicitation data for continuous variables

Description

Simulated data for the first and second round of an elicitation process estimating three variables with three different elicitation types.

Usage

```
round_1

round_2
```

Format

A data frame with 6 rows and 9 columns:

name Name of the experts (randomly generated).

var1_best Best estimate of var1. The estimate contains integer numbers referring to the one point elicitation method.

var2_min, var2_max, var2_best Minimum, maximum, and best guess estimates of var2. The estimates contain positive integer numbers referring to the three points elicitation method.

var3_min, var3_max, var3_best Minimum, maximum and best guess estimates of var3. The estimates contain probabilities referring to the four points elicitation method.

var3_conf Expert confidence for var3, expressed as a percentage higher than 50%.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 6 rows and 9 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 6 rows and 9 columns.

Source

Randomly generated numbers and names.

cont_get_data	<i>Get data</i>
---------------	-----------------

Description

[Stable]

`cont_get_data()` gets data from an [elic_cont](#) object.

Usage

```
cont_get_data(
  x,
  round,
  ...,
  var = "all",
  var_types = "all",
  elic_types = "all"
)
```

Arguments

x	an object of class elic_cont .
round	integer indicating if the data belongs to the first or second elicitation round.
...	Unused arguments, included only for future extensions of the function.
var	character string with the name of the variable or character vector with more variable names that you want to extract from the data. Use all for all variables.
var_types	character string with short codes indicating the variable type. See Variable types for more.
elic_types	character string with the short codes codes indicating the elicitation type. Use all for all elicitation types. See Elicitation Types for more.

Details

One optional argument can be specified. If more than one is provided, the first of the following will be used: var, var_types, or elic_types.

Value

A [tibble](#) with the extracted data.

Variable types

Variable types must be provided as a single string containing short codes, e.g. "pZN".

Valid short codes are:

- Z: *integers*, when the estimate must be an integer number in the interval $(-\infty, \infty)$.
- N: *positive integers*, when the estimate must be an integer number in the interval $[0, \infty)$.
- z: *negative integers*, when the estimate must be an integer number in the interval $(-\infty, 0)$.
- R: *reals*, when the estimate must be a real number in the interval $(-\infty, \infty)$.
- s: *positive reals*, when the estimate must be a real number in the interval $[0, \infty)$.
- r: *negative reals*, when the estimate must be a real number in the interval $(-\infty, 0)$.
- p: *probability*, when the estimate must be a real number in the interval $[0, 1]$.

Elicitation types

Elicitation types must be provided as a single string containing short codes, e.g. "134".

Valid short codes are:

- 1: *one point elicitation*, when only the best estimate is provided.
- 3: *three points elicitation*, when the minimum, maximum, and best estimates are provided.
- 4: *four points elicitation*, when the minimum, maximum, best, and confidence estimates are provided.

Author(s)

Sergio Vignali and Maude Vernet

See Also

Other cont data helpers: [cont_add_data\(\)](#), [cont_sample_data\(\)](#), [cont_start\(\)](#), [summary.cont_sample\(\)](#)

Examples

```
# Create the elicit object and add data for the first and second round from a
# data.frame.
my_elicit <- cont_start(var_names = c("var1", "var2", "var3"),
                        var_types = "ZNp",
                        elic_types = "134",
                        experts = 6) |>
  cont_add_data(data_source = round_1, round = 1) |>
  cont_add_data(data_source = round_2, round = 2)

# Get all data from round 1
cont_get_data(my_elicit, round = 1)

# Get data by variable name----
# Get data for var3 from round 2
cont_get_data(my_elicit, round = 2, var = "var3")

# Get data for var1 and var2 from round 1
cont_get_data(my_elicit, round = 1, var = c("var1", "var2"))

# Get data by variable type----
# Get data for variables containing integer numbers
cont_get_data(my_elicit, round = 2, var_types = "Z")
# Get data for variables containing positive integers and probabilities
cont_get_data(my_elicit, round = 2, var_types = "Np")

# Get data by elicitation type----
# Get data for three points estimates
cont_get_data(my_elicit, round = 2, elic_types = "3")
# Get data for one and four points estimates
cont_get_data(my_elicit, round = 2, elic_types = "14")
```

cont_sample_data

Sample continuous data

Description

[Stable]

cont_sample_data() samples data based on expert estimates stored in the [elic_cont](#) object.

Usage

```
cont_sample_data(
  x,
  round,
```

```

    ...,
    method = "PERT",
    var = "all",
    n_votes = 1000,
    weights = NULL,
    scale_conf = 100,
    verbose = TRUE
  )

```

Arguments

x	an object of class <code>elic_cont</code> .
round	integer indicating if the data belongs to the first or second elicitation round.
...	Unused arguments, included only for future extensions of the function.
method	character string with the name of the method to sample the data, only the <i>PERT</i> Method is implemented, see Method below.
var	character string with the name of the variable or character vector with more variable names that you want to extract from the data. Use <code>all</code> for all variables.
n_votes	numeric indicating the number of votes to consider.
weights	numeric vector with the weights to apply to the estimates. If equal to 1, each experts get <code>n_votes</code> votes, see Weights below.
scale_conf	numeric, the scale factor for the confidence interval.
verbose	logical, if TRUE it prints informative messages.

Details

Weighted allocations distribute the full budget among answering experts. Thus, the number of row in the sampled data will `n_experts * n_votes`. If an expert doesn't provide data for the sampled variable, a placeholder row is added. In such case, row counts can exceed `n_experts * n_votes`.

Value

A `tibble` with the sampled data. This object has an additional class `cont_sample` used to implement the plotting method.

Weights

To provide a different number of votes to each expert use the `weights` argument. The length of the vector must be equal to the number of experts or have a length of one. Default is `NULL`, when no weight is provided.

If provided when the elicitation type is a *four points elicitation*, the weight values overwrite the confidence estimates.

Method

The function samples the data differently depending on the elicitation type:

- *one point elicitation*: the estimate provided by each expert is repeated `n_votes` number of times. The `weights` argument can be used to weight the estimates of each expert (give a certain number of vote to each expert) in the overall distribution.
- *three points elicitation*: the minimum, best, and maximum estimates of each expert are used as scaling parameters of the PERT distribution from which the data are sampled. The `weights` argument can be used to weight the estimates of each expert (give a certain number of vote to each expert) in the overall distribution.
- *four points elicitation*: the minimum, best, and maximum estimates of each expert are rescaled according to their confidence and used as scaling parameters of the PERT distribution from which the data are sampled. Furthermore, their confidence is used as the `weights` argument to weight the estimates of each expert (give a number of vote to each expert) in the overall distribution.

scale_conf

If the variable plotted is the result of a four points elicitation where expert confidence is provided, the minimum and maximum values provided by each expert are rescaled using their provided confidence categories. Users can choose how they want to rescale minimum and maximum values by providing a value for the `scale_conf` argument. If no argument is provided, a default value of 100 is used for `scale_conf`.

The scaled minimum and maximum values are obtained with:

$$\begin{aligned} \text{minimum} &= \text{best guess} - (\text{best guess} - \text{minimum}) \frac{\text{scale_conf}}{\text{confidence}} \\ \text{maximum} &= \text{best guess} + (\text{maximum} - \text{best guess}) \frac{\text{scale_conf}}{\text{confidence}} \end{aligned}$$

Author(s)

Sergio Vignali and Maude Vernet

See Also

Other cont data helpers: [cont_add_data\(\)](#), [cont_get_data\(\)](#), [cont_start\(\)](#), [summary.cont_sample\(\)](#)

Examples

```
# Create the elicit object and add data for the first and second round from a
# data.frame.
my_elicit <- cont_start(var_names = c("var1", "var2", "var3"),
                        var_types = "ZNp",
                        elic_types = "134",
                        experts = 6) |>
  cont_add_data(data_source = round_1, round = 1) |>
  cont_add_data(data_source = round_2, round = 2)

# Sample data for the second round for all variables
samp <- cont_sample_data(my_elicit, round = 2)
```

```
samp

# Sample data for the first round for the variable `var1` and `var2`
samp <- cont_sample_data(my_elicit, round = 1, var = c("var1", "var2"))

# Sample data for the second round for the variable `var3`. Notice that the
# data are rescaled using the expert confidence before sampling.
samp <- cont_sample_data(my_elicit, round = 2, var = "var3")

# Sample data for the first round for the variable `var3` providing the
# weights. Notice that the weights overwrite the confidence estimates and
# that some values are constrained to be between 0 and 1 after rescaling.
samp <- cont_sample_data(my_elicit, round = 1, var = "var3",
                        weights = c(0.8, 0.7, 0.9, 0.7, 0.6, 0.9))
```

cont_start

Start elicitation

Description

[Stable]

cont_start() initialises an [elic_cont](#) object which stores important metadata for the data collected during the elicitation process of continuous variables.

Usage

```
cont_start(
  var_names,
  var_types,
  elic_types,
  experts,
  ...,
  title = "Elicitation",
  verbose = TRUE
)
```

Arguments

var_names	character vector with the name of the estimated variables.
var_types	character string with short codes indicating the variable type. If only one var_type is provided, its value is recycled for all variables. See Variable types for more.
elic_types	character string with short codes indicating the elicitation type. If only one elic_type is provided, its value is recycled for all variables. See Elicitation Types for more.
experts	numeric indicating the number of experts participating in the elicitation process.
...	Unused arguments, included only for future extensions of the function.
title	character, used to bind a name to the object.
verbose	logical, if TRUE it prints informative messages.

Value

An object of class `elic_cont` binding metadata related to the elicitation process. These metadata are used by other functions to validate the correctness of the provided data.

Variable types

Variable types must be provided as a single string containing short codes, e.g. "pZN".

Valid short codes are:

- Z: *integers*, when the estimate must be an integer number in the interval $(-\text{Inf}, \text{Inf})$.
- N: *positive integers*, when the estimate must be an integer number in the interval $[0, \text{Inf})$.
- z: *negative integers*, when the estimate must be an integer number in the interval $(-\text{Inf}, 0)$.
- R: *reals*, when the estimate must be a real number in the interval $(-\text{Inf}, \text{Inf})$.
- s: *positive reals*, when the estimate must be a real number in the interval $[0, \text{Inf})$.
- r: *negative reals*, when the estimate must be a real number in the interval $(-\text{Inf}, 0)$.
- p: *probability*, when the estimate must be a real number in the interval $[0, 1]$.

Elicitation types

Elicitation types must be provided as a single string containing short codes, e.g. "134".

Valid short codes are:

- 1: *one point elicitation*, when only the best estimate is provided.
- 3: *three points elicitation*, when the minimum, maximum, and best estimates are provided.
- 4: *four points elicitation*, when the minimum, maximum, best, and confidence estimates are provided.

Author(s)

Sergio Vignali

References

Hemming, V., Burgman, M. A., Hanea, A. M., McBride, M. F., & Wintle, B. C. (2018). A practical guide to structured expert elicitation using the IDEA protocol. *Methods in Ecology and Evolution*, 9(1), 169–180. <https://doi.org/10.1111/2041-210X.12857>

See Also

Other cont data helpers: `cont_add_data()`, `cont_get_data()`, `cont_sample_data()`, `summary.cont_sample()`

Examples

```
# Create the elic_cont object for an elicitation process that estimates 3
# variables, the first for a one point estimation of a positive integer, the
# second for three points estimation of a negative real, and the last for a
# four point estimation of a probability
my_elicit <- cont_start(var_names = c("var1", "var2", "var3"),
                        var_types = "Nrp",
                        elic_types = "134",
                        experts = 4)

my_elicit

# A title can be added to bind a name to the object:
my_elicit <- cont_start(var_names = c("var1", "var2", "var3"),
                        var_types = "Nrp",
                        elic_types = "134",
                        experts = 4,
                        title = "My elicitation")

my_elicit
# Notice that if var_types and elic_types are provided as single character,
# their value is recycled and applied to all variables. In the following
# example all three variables will be considered for a four point estimation
# to estimate a probability:
my_elicit <- cont_start(var_names = c("var1", "var2", "var3"),
                        var_types = "p",
                        elic_types = "4",
                        experts = 4)

my_elicit
```

elic_cat

elic_cat class

Description

The `elic_cat` class is a container for the elicitation data. It is used to store the data collected during the elicitation process and their metadata.

Object elements

There are 6 elements in the `elic_cat` object:

- `categories`: character vector with the names of the categories.
- `options`: character vector with the names of the options investigated.
- `experts`: numeric, indicating the maximum number of experts participating in the elicitation process for one topic.
- `topics`: character vector with the names of the topics investigated.
- `data`: list with the data collected during the elicitation process. The list has multiple elements, corresponding to the topics investigated.

Moreover, the object has a `title` attribute that binds a name to the object.

elic_cont

elic_cont class

Description

The `elic_cont` class is a container for the elicitation data. It is used to store the data collected during the elicitation process and their metadata.

Object elements

There are 5 elements in the `elic_cont` object:

- `var_names`: character vector with the name of the estimated variables.
- `var_types`: character string with short codes indicating the variable type.
- `elic_types`: character string with short codes indicating the elicitation type.
- `experts`: numeric indicating the number of experts participating in the elicitation process.
- `data`: list with the data collected during the elicitation process. The list has two elements, `round_1` and `round_2`, which store the data collected during the first and second round of the elicitation process, respectively.

Moreover, the object has a `title` attribute that binds a name to the object.

plot.cat_sample

Plot categorical samples

Description

[Stable]

`plot()` aggregates and plots categorical samples as violin plot.

Usage

```
## S3 method for class 'cat_sample'
plot(
  x,
  type = "violin",
  ...,
  option = "all",
  title = NULL,
  ylab = "Probability",
  colours = NULL,
  family = "sans",
  theme = NULL,
  beeswarm_cex = 0.6,
  beeswarm_corral = "none"
)
```

Arguments

x	an object of class <code>cat_sample</code> created by the function <code>cat_sample_data</code> .
type	character string with the type of plot, either <i>beeswarm</i> or <i>violin</i> .
...	Unused arguments, included only for future extensions of the function.
option	character string with the name of the option or character vector with the options that you want to extract from the data. Use <code>all</code> for all options.
title	character string with the title of the plot. If <code>NULL</code> , the title will be the topic name.
ylab	character string with the label of the y-axis.
colours	vector of colours to use for the categories.
family	character string with the font family to use in the plot.
theme	a theme object to use in the plot, such as <code>ggplot2::theme_minimal()</code> .
beeswarm_cex	numeric, the space between points in the beeswarm plot.
beeswarm_corral	character string, the wrapping corral for the beeswarm plot. Anything accepted by the geom_beeswarm function.

Details

If a theme is provided, the family argument is ignored.

Value

A [ggplot](#) object.

Author(s)

Sergio Vignali and Maude Vernet

See Also

Other plot helpers: [plot.cont_sample\(\)](#), [plot.elic_cont\(\)](#)

Examples

```
# Create the elic_cat object for an elicitation process with three topics,
# four options, five categories and a maximum of six experts per topic
my_categories <- c("category_1", "category_2", "category_3",
                  "category_4", "category_5")
my_options <- c("option_1", "option_2", "option_3", "option_4")
my_topics <- c("topic_1", "topic_2", "topic_3")
my_elicit <- cat_start(categories = my_categories,
                      options = my_options,
                      experts = 6,
                      topics = my_topics) |>
  cat_add_data(data_source = topic_1, topic = "topic_1") |>
  cat_add_data(data_source = topic_2, topic = "topic_2") |>
  cat_add_data(data_source = topic_3, topic = "topic_3")
```

```

# Sample data from Topic 1 for all options using the unweighted method
samp <- cat_sample_data(my_elicit,
                        method = "unweighted",
                        topic = "topic_1")

# Plot the sampled data for all options
plot(samp)

# Plot the sampled data as beeswarm plot
plot(samp, type = "beeswarm", beeswarm_corral = "wrap")

# Plot the sampled data for option 1
plot(samp, option = "option_1")

# Plot the sampled data for option 1 and 3
plot(samp, option = c("option_1", "option_3"))

# Provide custom colours
plot(samp, colours = c("steelblue4", "darkcyan", "chocolate1",
                      "chocolate3", "orangered4"))

# Overwrite the default theme
plot(samp, theme = ggplot2::theme_minimal())

```

plot.cont_sample	<i>Plot continuous samples</i>
------------------	--------------------------------

Description

[Stable]

plot() aggregates and plots continuous samples as violin or density plot.

Usage

```

## S3 method for class 'cont_sample'
plot(
  x,
  var = NULL,
  ...,
  group = FALSE,
  type = "violin",
  title = NULL,
  xlab = "",
  ylab = "",
  colours = NULL,
  line_width = 0.7,
  family = "sans",
  expert_names = NULL,

```

```

    theme = NULL,
    beeswarm_cex = 0.6,
    beeswarm_corral = "none"
  )

```

Arguments

x	n object of class <code>cont_sample</code> created by the function cont_sample_data .
var	character string with the name of the variable to be plotted.
...	Unused arguments, included only for future extensions of the function.
group	logical, if TRUE data are aggregated by expert.
type	character string with the type of plot, either <i>beeswarm</i> or <i>violin</i> or <i>density</i> .
title	character string with the title of the plot.
xlab	character string with the x-axis label.
ylab	character string with the y-axis label.
colours	character vector with the colours to be used in the plot.
line_width	numeric with the width of the lines in the density plot.
family	character string with the font family to be used in the plot.
expert_names	numeric or character, the labels for the experts.
theme	a theme object to use in the plot, such as <code>ggplot2::theme_minimal()</code> .
beeswarm_cex	numeric, the space between points in the beeswarm plot.
beeswarm_corral	character string, the wrapping corral for the beeswarm plot. Anything accepted by the geom_beeswarm function.

Details

If a theme is provided, the family argument is ignored.

Value

A [ggplot](#) object.

Author(s)

Sergio Vignali and Maude Vernet

See Also

Other plot helpers: [plot.cat_sample\(\)](#), [plot.elic_cont\(\)](#)

Examples

```
# Create the elic object and add data for the first and second round from a
# data.frame.
my_elicit <- cont_start(var_names = c("var1", "var2", "var3"),
                        var_types = "ZNp",
                        elic_types = "134",
                        experts = 6) |>
  cont_add_data(data_source = round_1, round = 1) |>
  cont_add_data(data_source = round_2, round = 2)

# Sample data for the first round for all variables
samp <- cont_sample_data(my_elicit, round = 1)

# Plot the sampled data for the variable `var3` as violin plot
plot(samp, var = "var3", type = "violin")

# Plot the sampled data for the variable `var1` as beeswarm plot
plot(samp, var = "var1", type = "beeswarm")

# Plot the sampled data for the variable `var2` as density plot
plot(samp, var = "var2", type = "density")

# Plot the sampled data for the variable `var1` as density plot for the group
plot(samp, var = "var1", group = TRUE, type = "density")

# Plot the sampled data for the variable `var3` as violin plot passing the
# colours
plot(samp, var = "var3", type = "violin",
     colours = c("steelblue4", "darkcyan", "chocolate1",
                 "chocolate3", "orangered4", "royalblue1"))
```

plot.elic_cont

Plot elicitation data

Description

[Stable]

plot() plots elicitation data for a specific round and variable.

Usage

```
## S3 method for class 'elic_cont'
plot(
  x,
  round,
  var,
  ...,
  scale_conf = 100,
  group = FALSE,
```

```

    truth = NULL,
    colour = "purple",
    group_colour = "orange",
    truth_colour = "red",
    point_size = 4,
    line_width = 1.5,
    title = paste("Round", round),
    xlab = var,
    ylab = "Experts",
    family = "sans",
    expert_names = NULL,
    theme = NULL,
    verbose = TRUE
  )

```

Arguments

x	an object of class elic_cont .
round	integer indicating if the data belongs to the first or second elicitation round.
var	character string, the variable to be plotted.
...	Unused arguments, included only for future extensions of the function.
scale_conf	numeric, the scale factor for the confidence interval.
group	logical, whether to plot the group mean, see Details for more.
truth	list, the true value of the variable, see Details for more.
colour	character string, the colour of estimated values.
group_colour	character string, the colour of the group mean.
truth_colour	character string, the colour of the true value.
point_size	numeric, the size of the points.
line_width	numeric, the width of the lines.
title	character, the title of the plot.
xlab	character, the title of the x axis.
ylab	character, the title of the y axis.
family	character, the font family.
expert_names	numeric or character, the labels for the experts.
theme	a theme object to use in the plot, such as <code>ggplot2::theme_minimal()</code> . <code>theme</code> .
verbose	logical, if TRUE it prints informative messages.

Details

If a 1-point elicitation is plotted, the `group` argument will show the mean and 2.5th–97.5th percentile interval of expert estimates.

The `truth` argument is useful when the elicitation process is part of a workshop and is used for demonstration. In this case the true value is known and can be added to the plot. This argument

must be a list with the following elements: min, max, best, and conf. When var refers to a *one point elicitation* estimate, only the best element is required. When var refers to a *three points elicitation*.estimate, the min and max elements are also required. Finally, when var refers to a *four points elicitation* estimate, the conf element is also required. The conf element is used to rescale the min and max values.

If a theme is provided, the family argument is ignored.

Value

A `ggplot` object.

scale_conf

If the variable plotted is the result of a four points elicitation where expert confidence is provided, the minimum and maximum values provided by each expert are rescaled using their provided confidence categories. Users can choose how they want to rescale minimum and maximum values by providing a value for the `scale_conf` argument. If no argument is provided, a default value of 100 is used for `scale_conf`.

The scaled minimum and maximum values are obtained with:

$$\text{minimum} = \text{best guess} - (\text{best guess} - \text{minimum}) \frac{\text{scale_conf}}{\text{confidence}}$$

$$\text{maximum} = \text{best guess} + (\text{maximum} - \text{best guess}) \frac{\text{scale_conf}}{\text{confidence}}$$

Author(s)

Sergio Vignali, Maude Vernet and Stefano Canessa

See Also

Other plot helpers: `plot.cat_sample()`, `plot.cont_sample()`

Examples

```
# Create the elicit object and add data for the first and second round from a
# data.frame.
my_elicit <- cont_start(var_names = c("var1", "var2", "var3"),
                        var_types = "ZNp",
                        elic_types = "134",
                        experts = 6) |>
  cont_add_data(data_source = round_1, round = 1) |>
  cont_add_data(data_source = round_2, round = 2)

# Plot the elicitation data for the first round and the variable var1 (only
# the best estimate)
plot(my_elicit, round = 1, var = "var1")

# Plot the elicitation data for the first round and the variable var2 (best
# estimate with min and max errors)
plot(my_elicit, round = 1, var = "var2")

# Plot the elicitation data for the first round and the variable var3 (best
```

```
# estimate with min and max errors rescaled to the confidence value)
plot(my_elicit, round = 1, var = "var3")

# Add the group mean
plot(my_elicit, round = 1, var = "var3", group = TRUE)

# Add the true value
plot(my_elicit, round = 1, var = "var3",
      truth = list(min = 0.6, max = 0.85, best = 0.75, conf = 100))

# Overwrite the default theme
plot(my_elicit, round = 1, var = "var3",
      theme = ggplot2::theme_classic())
```

summary.cat_sample	<i>Summarise samples of categorical data</i>
--------------------	--

Description

[Stable]

summary() summarises the sampled data and provides the minimum, first quartile, median, mean, third quartile, and maximum values for each category.

Usage

```
## S3 method for class 'cat_sample'
summary(object, option = "all", ...)
```

Arguments

object	an object of class cat_sample created by the function cat_sample_data .
option	character string with the name of the option(s). If option = "all", all options are summarised.
...	Unused arguments, included only for future extensions of the function.

Value

A [tibble](#) with the summary statistics.

Author(s)

Sergio Vignali

See Also

Other cat data helpers: [cat_add_data\(\)](#), [cat_get_data\(\)](#), [cat_sample_data\(\)](#), [cat_start\(\)](#)

Examples

```
# Create the elic_cat object for an elicitation process with three topics,
# four options, five categories and a maximum of six experts per topic
my_categories <- c("category_1", "category_2", "category_3",
                  "category_4", "category_5")
my_options <- c("option_1", "option_2", "option_3", "option_4")
my_topics <- c("topic_1", "topic_2", "topic_3")
my_elicit <- cat_start(categories = my_categories,
                      options = my_options,
                      experts = 6,
                      topics = my_topics) |>
  cat_add_data(data_source = topic_1, topic = "topic_1") |>
  cat_add_data(data_source = topic_2, topic = "topic_2") |>
  cat_add_data(data_source = topic_3, topic = "topic_3")

# Sample data from Topic 1 for all options using the unweighted method
samp <- cat_sample_data(my_elicit,
                      method = "unweighted",
                      topic = "topic_1")

# Summarise the sampled data
summary(samp, option = "option_1")
```

summary.cont_sample	<i>Summarise samples of continuous data</i>
---------------------	---

Description

[Stable]

summary() summarises the sampled data and provides the minimum, first quartile, median, mean, third quartile, and maximum values for each variable.

Usage

```
## S3 method for class 'cont_sample'
summary(object, ..., var = "all")
```

Arguments

object	an object of class cont_sample created by the function cont_sample_data .
...	Unused arguments, included only for future extensions of the function.
var	character vector with the names of the variables to summarise. If var = "all", all variables are summarised.

Value

A [tibble](#) with the summary statistics.

Author(s)

Sergio Vignali

See AlsoOther cont data helpers: [cont_add_data\(\)](#), [cont_get_data\(\)](#), [cont_sample_data\(\)](#), [cont_start\(\)](#)**Examples**

```
# Create the elicit object and add data for the first and second round from a
# data.frame.
my_elicit <- cont_start(var_names = c("var1", "var2", "var3"),
                        var_types = "ZNp",
                        elic_types = "134",
                        experts = 6) |>
  cont_add_data(data_source = round_1, round = 1) |>
  cont_add_data(data_source = round_2, round = 2)

# Sample data for the second round for all variables
samp <- cont_sample_data(my_elicit, round = 2)

# Summarise the sampled data for all variables
summary(samp)

# Summarise the sampled data for the variable "var1"
summary(samp, var = "var1")

# Summarise the sampled data for the variables "var1" and "var3"
summary(samp, var = c("var1", "var3"))
```

Index

* cat data helpers

- cat_add_data, 2
- cat_get_data, 7
- cat_sample_data, 8
- cat_start, 10
- summary.cat_sample, 31

* cont data helpers

- cont_add_data, 12
- cont_get_data, 16
- cont_sample_data, 18
- cont_start, 21
- summary.cont_sample, 32

* datasets

- cat_data, 6
- cont_data, 15

* plot helpers

- plot.cat_sample, 24
- plot.cont_sample, 26
- plot.elic_cont, 28

- cat_add_data, 2
- cat_add_data(), 7, 10, 12, 31
- cat_data, 6
- cat_get_data, 7
- cat_get_data(), 4, 10, 12, 31
- cat_sample_data, 8, 25, 31
- cat_sample_data(), 4, 7, 12, 31
- cat_start, 10
- cat_start(), 4, 7, 10, 31
- cont_add_data, 12
- cont_add_data(), 18, 20, 22, 33
- cont_data, 15
- cont_get_data, 16
- cont_get_data(), 14, 20, 22, 33
- cont_sample_data, 18, 27, 32
- cont_sample_data(), 14, 18, 22, 33
- cont_start, 11, 21
- cont_start(), 14, 18, 20, 33

- data.frame, 3, 13

- elic_cat, 2–4, 7–11, 23

- elic_cont, 12–14, 16–19, 21, 22, 24, 29

- geom_beeswarm, 25, 27

- ggplot, 25, 27, 30

- plot.cat_sample, 24

- plot.cat_sample(), 27, 30

- plot.cont_sample, 26

- plot.cont_sample(), 25, 30

- plot.elic_cont, 28

- plot.elic_cont(), 25, 27

- read_sheet, 3, 13

- round_1 (cont_data), 15

- round_2 (cont_data), 15

- summary.cat_sample, 31

- summary.cat_sample(), 4, 7, 10, 12

- summary.cont_sample, 32

- summary.cont_sample(), 14, 18, 20, 22

- theme object, 25, 27, 29

- tibble, 3, 7, 9, 13, 17, 19, 31, 32

- topic_1 (cat_data), 6

- topic_2 (cat_data), 6

- topic_3 (cat_data), 6