

Package ‘SemNetCleaner’

October 16, 2026

Title An Automated Cleaning Tool for Semantic and Linguistic Data

Version 2.0.0

Date 2026-10-14

Maintainer Alexander P. Christensen <alexpaulchristensen@gmail.com>

Description Implements several functions that automates the cleaning and spell-checking of text data. Also converges, finalizes, removes plurals and continuous strings, and puts text data in binary format for semantic network analysis. Uses the 'SemNet-Dictionaries' package to make the cleaning process more accurate, efficient, and reproducible.

License GPL (>= 3.0)

URL <https://github.com/AlexChristensen/SemNetCleaner>

BugReports <https://github.com/AlexChristensen/SemNetCleaner/issues>

NeedsCompilation no

Encoding UTF-8

Language en-US

LazyData true

Depends R (>= 4.0.1), SemNetDictionaries (>= 0.3.0)

Imports future, future.apply, hunspell, phonics, progressr, searcher, stringi, stringdist

Suggests DT, foreign, promises, R.matlab, readxl, rstudioapi, shiny (>= 1.8.0), tcltk, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Config/roxygen2/version 8.1.1

Author Alexander P. Christensen [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-9798-7037>>)

Repository CRAN

Date/Publication 2026-10-16 13:30:22 UTC

Contents

SemNetCleaner-package	2
bad_response	3
best_guess	4
binary_to_responses	5
convert2snafu	6
letter.freq	7
open.animals	8
open.clean	8
open.preprocess	9
pluralize	10
qwerty.dist	11
read.data	12
resp2bin	14
singularize	15
textcleaner	16
Index	21

SemNetCleaner-package	<i>SemNetCleaner-package</i>
-----------------------	------------------------------

Description

Implements several functions that automates the cleaning and spell-checking of text data. Also converges, finalizes, removes plurals and continuous strings, and puts text data in binary format for semantic network analysis. Uses the SemNetDictionaries package to make the cleaning process more accurate, efficient, and reproducible.

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

See Also

Useful links:

- <https://github.com/AlexChristensen/SemNetCleaner>
- Report bugs at <https://github.com/AlexChristensen/SemNetCleaner/issues>

bad_response	<i>Changes Bad Responses to NA</i>
--------------	------------------------------------

Description

Determines whether responses are good or bad. Bad responses are replaced with missing (NA). Good responses are returned

Usage

```
bad_response(word, ...)
```

```
bad.response(word, ...)
```

Arguments

word	Character. A word to be tested for whether it is bad
...	Vector. Additional responses to be considered bad

Value

If response is bad, then returns NA. If response is valid, then returns the response

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

Examples

```
# Bad response
bad_response(word = " ")

# Good response
bad_response(word = "hello")

# Make a good response bad
bad_response(word = "hello", "hello")

# Add additional bad responses
bad_response(word = "hello", c("hello", "world"))
```

best_guess

Makes Best Guess for Spelling Correction

Description

A wrapper function for the best guess of a spelling mistake based on the letters, the ordering of those letters, and the potential for letters to be interchanged. The **Damerau-Levenshtein distance** is used to guide inferences into what word the participant was trying to spell from a dictionary (see SemNetDictionaries)

Usage

```
best_guess(word, full_dictionary, dictionary = NULL, tolerance = 1)
```

```
best.guess(word, full.dictionary, dictionary = NULL, tolerance = 1)
```

Arguments

word	Character. A word to get best guess spelling options from dictionary
full_dictionary	Character vector. The dictionary to search for best guesses in. See SemNetDictionaries
dictionary	Character. A dictionary from SemNetDictionaries for monikers (enhances guessing)
tolerance	Numeric. The distance tolerance set for automatic spell-correction purposes. This function uses the function <code>stringdist</code> to compute the Damerau-Levenshtein distance, which is used to determine potential best guesses Unique words (i.e., $n = 1$) that are within the (distance) tolerance are automatically output as best guess responses. This default is based on Damerau's (1964) proclamation that more than 80% of all human misspellings can be expressed by a single error (e.g., insertion, deletion, substitution, and transposition). If there is more than one word that is within or below the distance tolerance, then these will be provided as potential options. The recommended and default distance tolerance is <code>tolerance = 1</code>, which only spell corrects a word if there is only one word with a DL distance of 1.
full.dictionary	Legacy name for 'full_dictionary'.

Value

The best guess(es) of the word

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

References

Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7, 171-176.

Philips, L. (1990). Hanging on the metaphone. *Computer Language*, 7(12), 39-43.

Examples

```
# Misspelled "bombay"
best_guess("bomba", full_dictionary = SemNetDictionaries::animals.dictionary)
```

binary_to_responses	<i>Binary Responses to Character Responses</i>
---------------------	--

Description

Converts a binary response matrix into characters for each participant

Usage

```
binary_to_responses(binary_matrix, to_data_frame = FALSE)

bin2resp(rmat, to.data.frame = FALSE)
```

Arguments

binary_matrix	Matrix. A binarized response matrix of verbal fluency or linguistic data
to_data_frame	Boolean. Should output be a data frame where participants are columns? Defaults to FALSE. Set to TRUE to convert output to data frame
rmat	Legacy name for 'binary_matrix'.
to.data.frame	Legacy name for 'to_data_frame'.

Value

A list containing objects for each participant and their responses

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

Examples

```
# Binary matrix of the first 10 participants of the cleaned data
binary_matrix <- responses_to_binary(open.clean[1:10, ])

# Each participant's responses, as a list
binary_to_responses(binary_matrix)

# As a participants by responses matrix
binary_to_responses(binary_matrix, to_data_frame = TRUE)
```

convert2snafu	<i>Converts textcleaner object to a SNAFU GUI format</i>
---------------	--

Description

Converts [textcleaner](#) object to a SNAFU GUI format (only works for fluency data)

Usage

```
convert2snafu(..., category, path, reader = readline)
```

Arguments

...	Matrix or data frame. A clean response matrices
category	Character. Category of verbal fluency data
path	Character. Directory to save the SNAFU-formatted .csv file to
reader	Function. The input reader used to prompt for the output file name. Defaults to readline. Exposed so this can be driven programmatically (e.g. in automated tests) instead of real console input

Details

The format of the file has 7 columns:

- id — Defaults to the row names of the inputted data
- listnum — The list number for the fluency category. Defaults to 0. Future implementations will allow more lists
- category — The verbal fluency category that is input into the category argument
- item — The verbal fluency responses for every participant
- RT — Response time. Currently not implemented. Defaults to 0
- RTstart — Start of response time. Currently not implemented. Defaults to 0
- group — Names of groups. Defaults to the names of the objects input into the function (...)

Value

A .csv file formatted for SNAFU

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

References

For SNAFU, see: Zemla, J. C., Cao, K., Mueller, K. D., & Austerweil, J. L. (2020). SNAFU: The Semantic Network and Fluency Utility. *Behavior Research Methods*, 1-19. <https://doi.org/10.3758/s13428-019-01343-w>

Examples

```
# Convert data to SNAFU (\code{reader} supplies the file name)
convert2snafu(
  open.clean[1:5, ], category = "animals", path = tempdir(),
  reader = function(prompt = "") "snafu_example"
)

if(interactive())
{
  # Asks for the file name
  convert2snafu(open.clean, category = "animals", path = tempdir())
}
```

letter.freq

Letter Frequencies Based on 40,000 Words

Description

A vector corresponding the frequency of letters across 40,000 words. Retrieved from: <http://pi.math.cornell.edu/~mec/2003-2004/cryptography/subs/frequencies.html>

Usage

```
data(letter.freq)
```

Format

letter.freq (26-element numeric vector)

Examples

```
data("letter.freq")
```

open.animals	<i>Openness and Verbal Fluency</i>
--------------	------------------------------------

Description

Raw Animals verbal fluency data ($n = 516$) from Christensen et al. (2018).

Usage

```
data(open.animals)
```

Format

open.animals (matrix 516 x 38)

Details

First column is a grouping variable ("Group") with 1 corresponding to low openness to experience and 2 to high openness to experience

Second column is the latent variable of openness to experience with Intellect items removed (see Christensen et al., 2018 for more details).

Third column is the ID variable for each participant.

Columns 4-38 are raw fluency data.

References

Christensen, A. P., Kenett, Y. N., Cotter, K. N., Beaty, R. E., & Silvia, P. J. (2018). Remotely close associations: Openness to experience and semantic memory structure. *European Journal of Personality*, 32, 480-492.

Examples

```
data("open.animals")
```

open.clean	<i>Cleaned Response Matrices (Openness and Verbal Fluency)</i>
------------	--

Description

Cleaned response matrices for the Animals verbal fluency data ($n = 516$) from Christensen et al. (2018).

Usage

```
data(open.clean)
```

Format

open.clean (matrix, 516 x 35)

References

Christensen, A. P., Kenett, Y. N., Cotter, K. N., Beaty, R. E., & Silvia, P. J. (2018). Remotely close associations: Openness to experience and semantic memory structure. *European Journal of Personality*, 32, 480-492.

Examples

```
data("open.clean")
```

open.preprocess	Preprocessed textcleaner Object (Openness and Verbal Fluency)
-----------------	---

Description

Preprocessed [textcleaner](#) object for the Animals verbal fluency data ($n = 516$) from Christensen and Kenett (2020).

Usage

```
data(open.preprocess)
```

Format

open.preprocess (list, length = 4)

References

Christensen, A. P., & Kenett, Y. N. (2020). Semantic network analysis (SemNA): A tutorial on preprocessing, estimating, and analyzing semantic networks. *PsyArxiv*.

Examples

```
data("open.preprocess")
```

pluralize*Converts Words to their Plural Form*

Description

A function to change words to their plural form. The rules for converting words to their plural forms are based on the grammar rules.

This function handles most special cases and some irregular cases (see examples) but caution is necessary. If no plural form is identified, then the original word is returned.

Usage

```
pluralize(word)
```

Arguments

word	A word
------	--------

Value

Returns the word in plural form, unless a plural form could not be found (then the original word is returned)

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

Examples

```
# Handles any prototypical cases
"dogs"
pluralize("dog")

"foxes"
pluralize("fox")

"wolves"
pluralize("wolf")

"octopi"
pluralize("octopus")

"taxa"
pluralize("taxon")

# And most special cases:
"wives"
pluralize("wife")
```

```
"roofs"
pluralize("roof")

"photos"
pluralize("photo")

# And some irregular cases:
"children"
pluralize("child")

"teeth"
pluralize("tooth")

"mice"
pluralize("mouse")
```

qwerty.dist

QWERTY Distance for Same Length Words

Description

Computes QWERTY distance for words that have the same number of characters. Distance is computed based on the number of keys a character is away from another character on a QWERTY keyboard

Usage

```
qwerty.dist(wordA, wordB)

qwerty_distance(word_a, word_b)
```

Arguments

wordA	Legacy name for 'word_a'.
wordB	Legacy name for 'word_b'.
word_a	Character. Word to be compared
word_b	Character. Word to be compared

Value

Numeric value for distance between word_a and word_b

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

Examples

```
# Identical values for Damerau-Levenshtein
stringdist::stringdist("big", "pig", method = "dl")

stringdist::stringdist("big", "bug", method = "dl")

# Different distances for QWERTY
qwerty_distance("big", "pig")

qwerty_distance("big", "bug") # Probably meant to type "bug"
```

read.data

Read in Common Data File Extensions

Description

A single function to read in common data file extensions. Note that this function is specialized for reading in text data in the format necessary for functions in SemNetCleaner

File extensions supported:

- .Rdata
- .rds
- .csv
- .xlsx
- .xls
- .sav
- .txt
- .mat
- .dat

Usage

```
read.data(file = file.choose(), header = TRUE, sep = ",", ...)
```

```
read_data(file = file.choose(), header = TRUE, sep = ",", ...)
```

Arguments

file	Character. A path to the file to load. Defaults to interactive file selection using file.choose
header	Boolean. A logical value indicating whether the file contains the names of the variables as its first line. If missing, the value is determined from the file format: header is set to TRUE if and only if the first row contains one fewer field than the number of columns

sep Character. The field separator character. Values on each line of the file are separated by this character. If sep = "" (the default for `read.table`) the separator is a 'white space', that is one or more spaces, tabs, newlines or carriage returns

... Additional arguments. Allows for additional arguments to be passed onto the respective read functions. See documentation in the list below:

- .Rdata [load](#)
- .rds [readRDS](#)
- .csv [read.table](#)
- .xlsx [read_excel](#)
- .xls [read_excel](#)
- .sav [read.spss](#)
- .txt [read.table](#)
- .mat [readMat](#)
- .dat [read.table](#)

Value

A data frame containing a representation of the data in the file. If file extension is ".Rdata", then data will be read to the global environment

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

References

R Core Team

R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

readxl

Hadley Wickham and Jennifer Bryan (2019). readxl: Read Excel Files. R package version 1.3.1. <https://CRAN.R-project.org/package=readxl>

R.matlab

Henrik Bengtsson (2018). R.matlab: Read and Write MAT Files and Call MATLAB from Within R. R package version 3.6.2. <https://CRAN.R-project.org/package=R.matlab>

Examples

```
# Use this example for your data
if(interactive())
{read_data()}

# Example for CRAN tests
## Create test data
test1 <- c(1:5, "6,7", "8,9,10")

## Path to temporary file
```

```
tf <- tempfile()

## Create test file
writeLines(test1, tf)

## Read in data
read_data(tf)

# See documentation of respective R functions for specific examples
```

resp2bin

Converts a Response Matrix to a Binary Response Matrix

Description

Converts a response matrix (verbal fluency or other linguistic data, one participant per row) into a binary (participant x vocabulary) response matrix

Usage

```
resp2bin(resp)

responses_to_binary(responses)
```

Arguments

resp	Legacy name for ‘responses’.
responses	Matrix or data frame. A response matrix of verbal fluency or linguistic data

Value

A binary matrix with one row per participant and one column per unique response observed across the whole sample

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

Examples

```
# First 10 participants of the cleaned data
binary_matrix <- responses_to_binary(open.clean[1:10, ])

# Participants by responses (0 = not given, 1 = given)
binary_matrix[, 1:10]
```

singularize	<i>Converts Words to their Singular Form</i>
-------------	--

Description

A function to change words to their singular form. The rules for converting words to their singular forms are based on the *inverse* of the grammar rules. This function handles most special cases and some irregular cases (see examples) but caution is necessary. If no singular form is identified, then the original word is returned.

Usage

```
singularize(word, dictionary = TRUE)
```

Arguments

word	Character. A word
dictionary	Boolean. Should dictionary be used to verify word exists? Default to TRUE

Value

Returns the word in singular form, unless a singular form could not be found (then the original word is returned)

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

Examples

```
# Handles any prototypical cases
# "dog"
singularize("dogs")

# "fox"
singularize("foxes")

# "wolf"
singularize("wolves")

# "octopus"
singularize("octopi")

# "taxon"
singularize("taxa")

# And most special cases:
# "wife"
singularize("wives")
```

```
# "fez"
singularize("fezzes")

# "roof"
singularize("roofs")

# "photo"
singularize("photos")

# And some irregular cases:
# "child"
singularize("children")

# "tooth"
singularize("teeth")

# "mouse"
singularize("mice")
```

textcleaner

Text Cleaner

Description

An automated cleaning function for spell-checking, de-pluralizing, removing duplicates, and binarizing text data

Usage

```
textcleaner(
  data = NULL,
  type = c("fluency", "free"),
  miss = 99,
  partBY = c("row", "col"),
  dictionary = NULL,
  spelling = c("UK", "US"),
  add.path = NULL,
  keepStrings = FALSE,
  allowPunctuations,
  allowNumbers = FALSE,
  keepLength = NULL,
  keepCue = FALSE,
  keepRepeats = FALSE,
  continue = NULL,
  reader = readline,
  search_fn = searcher::search_site,
  cores = 1,
```

```
    edit_fn = automated_edit
  )
```

Arguments

data Matrix or data frame.
For type = "fluency", data are expected to follow wide formatting (IDs are the row names and are **not** a column in the matrix or data frame; alternatively, a column named "ID", "subject", "participant", "part" or "sub" is used as the IDs and removed from the responses; failing that, a single column whose values are all different and contain digits (e.g., 1, 2, 3 or P01, P02) is taken to be the IDs; with none of these, rows are numbered):

row.names	Response 1	Response 2	Response n
ID_1	1	2	n
ID_2	1	2	n
ID_n	1	2	n

For type = "free", data are expected to follow long formatting:

ID	Cue	Response
1	1	1
1	1	2
1	1	n
1	2	1
1	2	2
1	2	n
1	n	1
1	n	2
1	n	n
2	1	1
2	1	2
2	1	n
2	2	1
2	2	2
2	2	n
2	n	1
2	n	2
2	n	n
n	1	1
n	1	2
n	1	n
n	2	1
n	2	2
n	2	n
n	n	1
n	n	2
n	n	n

type	Character vector. Type of task to be preprocessed. • "fluency" — Verbal fluency data (e.g., categories, phonological, synonyms) • "free" — Free association data (e.g., cue terms or words)
miss	Numeric or character. Value for missing data. Defaults to 99
partBY	Character. Are participants by row or column? Set to "row" for by row. Set to "col" for by column (the data are transposed first, so the column names become the participants' IDs, exactly as the row names are when participants are by row)
dictionary	Character vector. The name(s) of the dictionary (or dictionaries) to spell-check against, e.g., "animals", or a custom dictionary saved as <name>.dictionary.rds in the working directory or in add.path. It can also be a vector of the words themselves, such as a corpus or result\$dictionary from an earlier run; as in SemNetDictionaries::load.dictionaries(), a vector of MORE than 10 elements is taken to be the words, and 10 or fewer are taken to be names. For type = "free", the spell-check is always against the cocaspell word list; dictionary then only supplies its list of common misspellings Dictionary to be used for more efficient text cleaning. Defaults to NULL, which will use general.dictionary Use dictionaries() or find.dictionaries() for more options (See SemNetDictionaries for more details)
spelling	Character vector. English spelling to be used. • "UK" — For British spelling (e.g., colour, grey, programme, theatre) • "US" — For American spelling (e.g., color, gray, program, theater)
add.path	Character. Path to additional dictionaries to be found. DOES NOT search recursively (through all folders in path) to avoid time intensive search. Set to "choose" to open an interactive directory explorer
keepStrings	Boolean. Should strings be retained or separated? Defaults to FALSE. Set to TRUE to retain strings as strings. For fluency data, a string whose words are all in the dictionary (after correction) is still split into them, unless the string itself is a dictionary entry ("sea slug" stays "sea slug"; "marmoset otter" becomes "marmoset" and "otter")
allowPunctuations	Character vector. Allows punctuation characters to be included in responses. Defaults to "-". Set to "all" to keep all punctuation characters
allowNumbers	Boolean. Defaults to FALSE, which removes stand-alone numbers (e.g., list numbering such as "1. "). Words that mix letters and digits (e.g., "d0g") are kept as written so that the spell-check can correct or flag them. Set to TRUE to keep all numbers in text
keepLength	Numeric. Maximum number of words allowed in a response. Defaults to NULL. Set a number to keep responses with words less than or equal to the number (e.g., 3 will keep responses with three or less words)
keepCue	Boolean. Should cue words be retained in the responses? Defaults to FALSE. Set to TRUE to allow cue words to be retained
keepRepeats	Boolean. Should repeated responses (perseverations) within a participant be retained in \$responses\$clean? Only applies to type = "fluency". Defaults

	to FALSE, which keeps only the first occurrence of each response. Set to TRUE to keep every occurrence (e.g., so that a later analysis, such as SNAFU, can count them). The binary response matrix is unaffected, and repeats are counted in \$behavioral\$Perseverations either way
continue	List. A result previously unfinished that still needs to be completed. Allows you to continue to manually spell-check your data after you've exited early (X) or it stopped on an error: those are the cases in which textcleaner() returns an unfinished result instead of the final output. A finished result (one that already holds \$responses) cannot be passed back, because there is nothing left to continue. The result of a manual spell-check stage that has already finished (as the Shiny app hands over) is accepted too, to skip straight to finalizing the output. Defaults to NULL
reader	Function. The input reader used for every interactive prompt in the manual spell-check wizard. Defaults to readline. Exposed so the whole wizard can be driven programmatically (e.g. in automated tests) with a pre-scripted queue of answers instead of real console input
search_fn	Function. The function used to open a browser search for the "GOOGLE WORD"/"GOOGLE STRING" manual spell-check options. Defaults to searcher::search_site
cores	Numeric. Number of computer cores to use in parallelized computations (the "ad hoc" moniker re-check after manual spell-check). Defaults to 1 (sequential). With more than one core the workers load the installed package, so SemNetCleaner must be installed (not loaded with devtools::load_all())
edit_fn	Function. The spreadsheet-editor function used to review automated spell-check changes. Defaults to automated_edit, a shiny/DT-based review app (requires those packages to be installed). Pass edit_fn = edit to use base R's lightweight edit() instead, with no extra dependencies. Exposed so this step can be driven programmatically instead of opening a real interactive editor

Details

Responses are cleaned before they are spell-checked: they are lowercased, accented and typographic characters are transliterated to ASCII ("caf\u00e9" becomes "cafe"), and characters with no ASCII equivalent (e.g., Chinese or Cyrillic text) are removed, so a response made only of such characters is treated as missing. The dictionaries supplied by **SemNetDictionaries** are English.

Value

A list of class "textcleaner". For type = "fluency":

responses	A list of three matrices: - original — The responses as given, after basic preparation: missing values (miss) and blank cells converted to NA; punctuation and stand-alone numbers removed (see allowPunctuations and allowNumbers); accented letters converted to plain ones (e.g., an accented "e" to a plain "e"); surrounding and repeated white space removed; and letters lowercased - clean — The spell-checked, de-pluralized responses with repeats removed (unless keepRepeats = TRUE); one row per participant, labeled by ID. This can be used as a final dataset for analyses (e.g., fluency of responses)
-----------	---

	• binary — A matrix where each row is a participant and each column a unique response: 1 if the participant gave the response and 0 if not
behavioral	A data frame, one row per participant, with the number of Perseverations (repeated responses), Intrusions (responses marked bad), and Appropriate (responses kept)
spellcheck	A list of three matrices: correspondence — every unique response (from) and what it became (to_1, to_2, ...); automated — the rows changed by the automated spell-check; manual — the rows that went through the manual wizard
moniker	The manual rows of spellcheck: the corrections made by hand. Consider submitting them to <code>SemNetDictionaries</code>
dictionary, added_words	Only present if words were added to the dictionary and "In my results" was chosen when asked where to save them (or a file could not be written). dictionary is the whole dictionary, the one used plus the added words, so it can be passed back as <code>dictionary = result\$dictionary</code> . added_words is only the words added during the session

For `type = "free"`, `data` holds the original and clean long-format data, `responses$clean` is a response-by-cue frequency matrix, and `behavioral`, `spellcheck` and the `dictionary` objects are as above. If the manual spell-check is exited early, the object instead holds the saved state and can be passed back with `continue =`

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

References

- Christensen, A. P., & Kenett, Y. N. (2021). Semantic network analysis (SemNA): A tutorial on preprocessing, estimating, and analyzing semantic networks. *Psychological Methods*. doi:10.1037/met0000463
- Hornik, K., & Murdoch, D. (2010). Watch Your Spelling!. *The R Journal*, 3, 22-28.

Examples

```
if(interactive())
{
  # Full example with the manual spell-check
  clean <- textcleaner(open.animals[, -c(1, 2)], partBY = "row", dictionary = "animals")
}
```

Index

* datasets

- letter.freq, [7](#)
- open.animals, [8](#)
- open.clean, [8](#)
- open.preprocess, [9](#)

- bad.response (bad_response), [3](#)
- bad_response, [3](#)
- best.guess (best_guess), [4](#)
- best_guess, [4](#)
- bin2resp (binary_to_responses), [5](#)
- binary_to_responses, [5](#)

- convert2snafu, [6](#)

- file.choose, [12](#)

- general.dictionary, [18](#)

- letter.freq, [7](#)
- load, [13](#)

- open.animals, [8](#)
- open.clean, [8](#)
- open.preprocess, [9](#)

- pluralize, [10](#)

- qwerty.dist, [11](#)
- qwerty_distance (qwerty.dist), [11](#)

- read.data, [12](#)
- read.spss, [13](#)
- read.table, [13](#)
- read_data (read.data), [12](#)
- read_excel, [13](#)
- readMat, [13](#)
- readRDS, [13](#)
- resp2bin, [14](#)
- responses_to_binary (resp2bin), [14](#)

- SemNetCleaner (SemNetCleaner-package), [2](#)

- SemNetCleaner-package, [2](#)

- singularize, [15](#)

- stringdist, [4](#)

- textcleaner, [6](#), [9](#), [16](#)