

Package ‘phonics’

August 23, 2026

Type Package

Title Phonetic Spelling Algorithms

Version 1.4.0

Date 2026-08-09

Encoding UTF-8

URL <https://phonics.jameshoward.us>,
<https://github.com/k3jph/phonics-in-r>

BugReports <https://github.com/k3jph/phonics-in-r/issues>

Description Provides a collection of phonetic algorithms including Soundex, Metaphone, NYSIIS, Caverphone, and others. The package is documented in <[doi:10.18637/jss.v095.i08](https://doi.org/10.18637/jss.v095.i08)>.

License BSD_2_clause + file LICENSE

Imports Rcpp (>= 0.12.1), data.table

Suggests testthat (>= 3.0.0), knitr, rmarkdown, roxygen2

LinkingTo Rcpp

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation yes

Author James Howard [aut, cre] (ORCID:
<<https://orcid.org/0000-0003-4530-1547>>),
Kyle Haynes [ctb],
Amanda Hood [ctb],
Os Keyes [ctb]

Maintainer James Howard <jh@jameshoward.us>

Repository CRAN

Date/Publication 2026-08-23 11:20:08 UTC

Contents

caverphone	2
cologne	3
lein	4
metaphone	6
mra_encode	7
nysiis	8
onca	9
phonex	10
phonics	12
rogerroot	13
soundex	14
statcan	15
Index	17

caverphone	<i>Caverphone</i>
------------	-------------------

Description

The Caverphone family of phonetic algorithms

Usage

```
caverphone(word, maxCodeLen = NULL, modified = FALSE, clean = TRUE)
```

Arguments

word	string or vector of strings to encode
maxCodeLen	maximum length of the resulting encodings, in characters
modified	if TRUE, use the Caverphone 2 algorithm
clean	if TRUE, return NA for unknown alphabetical characters

Details

The variable maxCodeLen is the limit on how long the returned Caverphone code should be. The default is 6, unless modified is set to TRUE, then the default is 10.

The variable modified directs caverphone to use the Caverphone2 method, instead of the original.

The caverphone algorithm is only defined for inputs over the standard English alphabet, *i.e.*, "A-Z.". Non-alphabetical characters are removed from the string in a locale-dependent fashion. This strips spaces, hyphens, and numbers. Other letters, such as "Ü," may be permissible in the current locale but are unknown to caverphone. For inputs outside of its known range, the output is undefined and NA is returned and a warning is issued. If clean is FALSE, caverphone attempts to process the strings. The default is TRUE.

Value

the Caverphone encoded character vector

References

David Hood, "Caverphone: Phonetic matching algorithm," Technical Paper CTP060902, University of Otago, New Zealand, 2002.

David Hood, "Caverphone Revisited," Technical Paper CTP150804 University of Otago, New Zealand, 2004.

James P. Howard, II, "Phonetic Spelling Algorithm Implementations for R," *Journal of Statistical Software*, vol. 95, no. 8, (2020), p. 1–21, <10.18637/jss.v095.i08>.

See Also

Other phonics: [cologne\(\)](#), [lein\(\)](#), [metaphone\(\)](#), [mra_encode\(\)](#), [nysiis\(\)](#), [onca\(\)](#), [phonex\(\)](#), [phonics\(\)](#), [rogerroot\(\)](#), [soundex\(\)](#), [statcan\(\)](#)

Examples

```
caverphone("William")
caverphone(c("Peter", "Peady"), modified = TRUE)
caverphone("Stevenson", maxCodeLen = 4)
```

cologne	<i>Cologne Phonetic Name Coding</i>
---------	-------------------------------------

Description

The Cologne phonetic name coding procedure.

Usage

```
cologne(word, maxCodeLen = NULL, clean = TRUE)
```

Arguments

- | | |
|------------|--|
| word | string or vector of strings to encode |
| maxCodeLen | maximum length of the resulting encodings, in characters |
| clean | if TRUE, return NA for unknown alphabetical characters |

Details

`maxCodeLen = NULL` preserves the algorithm's historical unbounded output. An explicit nonnegative whole-number value truncates each code to at most that many characters.

The cologne algorithm is only defined for inputs over the standard English alphabet, *i.e.*, "A-Z," "Ä," "Ö," "Ü," and "ß." Non-alphabetical characters are removed from the string in a locale-dependent fashion. This strips spaces, hyphens, and numbers. Other letters, such as "ç," may be permissible in the current locale but are unknown to cologne. For inputs outside of its known range, the output is undefined and NA is returned and a warning is issued. If `clean` is `FALSE`, cologne attempts to process the strings. The default is `TRUE`.

Value

the Cologne encoded character vector

References

James P. Howard, II, "Phonetic Spelling Algorithm Implementations for R," *Journal of Statistical Software*, vol. 95, no. 8, (2020), p. 1–21, <10.18637/jss.v095.i08>.

Hans Joachim Postel. "Die Koelner Phonetik. Ein Verfahren zur Identifizierung von Personennamen auf der Grundlage der Gestaltanalyse." *IBM-Nachrichten* 19. Jahrgang, 1969, p. 925-931.

See Also

Other phonics: [caverphone\(\)](#), [lein\(\)](#), [metaphone\(\)](#), [mra_encode\(\)](#), [nysiis\(\)](#), [onca\(\)](#), [phonex\(\)](#), [phonics\(\)](#), [rogerroot\(\)](#), [soundex\(\)](#), [statcan\(\)](#)

Examples

```
cologne("William")
cologne(c("Peter", "Peady"))
cologne("Stevenson", maxCodeLen = 8)
```

lein

Lein Name Coding

Description

The Lein name coding procedure.

Usage

```
lein(word, maxCodeLen = 4, clean = TRUE)
```

Arguments

word	string or vector of strings to encode
maxCodeLen	maximum length of the resulting encodings, in characters
clean	if TRUE, return NA for unknown alphabetical characters

Details

The variable word is the name to be encoded. The variable maxCodeLen is the limit on how long the returned name code should be. The default is 4.

The lein algorithm is only defined for inputs over the standard English alphabet, *i.e.*, "A-Z.". Non-alphabetical characters are removed from the string in a locale-dependent fashion. This strips spaces, hyphens, and numbers. Other letters, such as "Ü," may be permissible in the current locale but are unknown to lein. For inputs outside of its known range, the output is undefined and NA is returned and a warning is issued. If clean is FALSE, lein attempts to process the strings. The default is TRUE.

Value

the Lein encoded character vector

References

James P. Howard, II, "Phonetic Spelling Algorithm Implementations for R," *Journal of Statistical Software*, vol. 95, no. 8, (2020), p. 1–21, <10.18637/jss.v095.i08>.

Billy T. Lynch and William L. Arends. "Selection of surname coding procedure for the SRS record linkage system." United States Department of Agriculture, Sample Survey Research Branch, Research Division, Washington, 1977.

See Also

Other phonics: [caverphone\(\)](#), [cologne\(\)](#), [metaphone\(\)](#), [mra_encode\(\)](#), [nysiis\(\)](#), [onca\(\)](#), [phonex\(\)](#), [phonics\(\)](#), [rogerroot\(\)](#), [soundex\(\)](#), [statcan\(\)](#)

Examples

```
lein("William")
lein(c("Peter", "Peedy"))
lein("Stevenson", maxCodeLen = 8)
```

metaphone

*Generate phonetic versions of strings with Metaphone***Description**

The function `metaphone` phonetically encodes the given string using the metaphone algorithm.

Usage

```
metaphone(word, maxCodeLen = 10L, clean = TRUE)
```

Arguments

<code>word</code>	string or vector of strings to encode
<code>maxCodeLen</code>	maximum length of the resulting encodings, in characters
<code>clean</code>	if TRUE, return NA for unknown alphabetical characters

Details

Metaphone has accumulated incompatible implementation variants. This function follows the traditional rules from Philips' 1990 algorithm; it is therefore not expected to reproduce PHP or Apache Commons Codec for every input.

The variable `maxCodeLen` is the limit on how long the returned metaphone should be.

The metaphone algorithm is only defined for inputs over the standard English alphabet, *i.e.*, "A-Z.". Non-alphabetical characters are removed from the string in a locale-dependent fashion. This strips spaces, hyphens, and numbers. Other letters, such as "Ü," may be permissible in the current locale but are unknown to metaphone. For inputs outside of its known range, the output is undefined and NA is returned and a warning this thrown. If `clean` is FALSE, metaphone attempts to process the strings. The default is TRUE.

Value

a character vector containing the metaphones of `word`, or an NA if the word value is NA

References

James P. Howard, II, "Phonetic Spelling Algorithm Implementations for R," *Journal of Statistical Software*, vol. 95, no. 8, (2020), p. 1–21, <10.18637/jss.v095.i08>.

Lawrence Philips, "Hanging on the Metaphone," *Computer Language*, vol. 7, no. 12 (1990), p. 38–44.

See Also

Other phonics: [caverphone\(\)](#), [cologne\(\)](#), [lein\(\)](#), [mra_encode\(\)](#), [nysiis\(\)](#), [onca\(\)](#), [phonex\(\)](#), [phonics\(\)](#), [rogerroot\(\)](#), [soundex\(\)](#), [statcan\(\)](#)

Examples

```
metaphone("wheel")  
metaphone(c("school", "benji"))
```

mra_encode

Match Rating Approach Encoder

Description

The Western Airlines matching rating approach name encoder

Usage

```
mra_encode(word, clean = TRUE)  
  
mra_compare(x, y)
```

Arguments

word	string or vector of strings to encode
clean	if TRUE, return NA for unknown alphabetical characters
x	MRA-encoded character vector
y	MRA-encoded character vector

Details

The variable `word` is the name to be encoded. The variable `maxCodeLen` is *not* supported in this algorithm encoder because the algorithm itself is dependent upon its six-character length. The variables `x` and `y` are MRA-encoded and are compared to each other using the MRA comparison specification.

The `mra_encode` algorithm is only defined for inputs over the standard English alphabet, *i.e.*, "A-Z.". Non-alphabetical characters are removed from the string in a locale-dependent fashion. This strips spaces, hyphens, and numbers. Other letters, such as "Ü," may be permissible in the current locale but are unknown to `mra_encode`. For inputs outside of its known range, the output is undefined and NA is returned and a warning is issued. If `clean` is FALSE, `mra_encode` attempts to process the strings. The default is TRUE.

Value

The `mra_encode` function returns match rating approach encoded character vector. The `mra_compare` returns a boolean vector which is TRUE if `x` and `y` pass the MRA comparison test.

References

James P. Howard, II, "Phonetic Spelling Algorithm Implementations for R," *Journal of Statistical Software*, vol. 95, no. 8, (2020), p. 1–21, <10.18637/jss.v095.i08>.

G.B. Moore, J.L. Kuhns, J.L. Treffz, and C.A. Montgomery, *Accessing Individual Records from Personal Data Files Using Nonunique Identifiers*, US National Institute of Standards and Technology, SP-500-2 (1977), p. 17.

See Also

Other phonics: [caverphone\(\)](#), [cologne\(\)](#), [lein\(\)](#), [metaphone\(\)](#), [nysiis\(\)](#), [onca\(\)](#), [phonex\(\)](#), [phonics\(\)](#), [rogerroot\(\)](#), [soundex\(\)](#), [statcan\(\)](#)

Examples

```
mra_encode("William")
mra_encode(c("Peter", "Peady"))
mra_encode("Stevenson")
```

nysiis

New York State Identification and Intelligence System

Description

The NYSIIS phonetic algorithm

Usage

```
nysiis(word, maxCodeLen = 6, modified = FALSE, clean = TRUE)
```

Arguments

word	string or vector of strings to encode
maxCodeLen	maximum length of the resulting encodings, in characters
modified	if TRUE, use the modified NYSIIS algorithm
clean	if TRUE, return NA for unknown alphabetical characters

Details

The `nysiis` function implements the original NYSIIS character scan described by Taft and the USDA modified scan documented by Lynch and Arends.

The variable `maxCodeLen` is the limit on how long the returned NYSIIS code should be. The default is 6.

Set `modified = TRUE` for the USDA variant. That publication classifies names ending in JR or SR as errors; this function warns and returns NA for those inputs.

The `nysiis` algorithm is only defined for inputs over the standard English alphabet, *i.e.*, "A-Z". Non-alphabetical characters are removed from the string in a locale-dependent fashion. This strips

spaces, hyphens, and numbers. Other letters, such as "Ü," may be permissible in the current locale but are unknown to nysiis. For inputs outside of its known range, the output is undefined and NA is returned and a warning is issued. If clean is FALSE, nysiis attempts to process the strings. The default is TRUE.

Value

the NYSIIS encoded character vector

References

James P. Howard, II, "Phonetic Spelling Algorithm Implementations for R," *Journal of Statistical Software*, vol. 95, no. 8, (2020), p. 1–21, <10.18637/jss.v095.i08>.

Robert L. Taft, *Name search techniques*, Bureau of Systems Development, Albany, New York, 1970.

Billy T. Lynch and William L. Arends, *Selection of a Surname Coding Procedure for the SRS Record Linkage System*, United States Department of Agriculture, 1977, Appendix B.

See Also

Other phonics: [caverphone\(\)](#), [cologne\(\)](#), [lein\(\)](#), [metaphone\(\)](#), [mra_encode\(\)](#), [onca\(\)](#), [phonex\(\)](#), [phonics\(\)](#), [rogerroot\(\)](#), [soundex\(\)](#), [statcan\(\)](#)

Examples

```
nysiis("Robert")
nysiis("rupert")
nysiis(c("Alabama", "Alaska"), modified = TRUE)
nysiis("mississippi", 4)
```

onca	<i>Oxford Name Compression Algorithm</i>
------	--

Description

The Oxford Name Compression Algorithm name coding procedure

Usage

```
onca(word, maxCodeLen = 4, clean = TRUE, modified = FALSE, refined = FALSE)
```

Arguments

word	string or vector of strings to encode
maxCodeLen	maximum length of the resulting encodings, in characters
clean	if TRUE, return NA for unknown alphabetical characters
modified	if TRUE, use the modified nysiis function
refined	if TRUE, use the refinedSoundex function

Details

ONCA applies NYSIIS and then Soundex. Gill describes the first stage as an "anglicised" NYSIIS but does not publish its complete rule set; this implementation uses the documented standard NYSIIS scan. The modified and refined combinations are package extensions. `maxCodeLen` bounds the final code and defaults to 4.

The `onca` algorithm is only defined for inputs over the standard English alphabet, *i.e.*, "A-Z". Non-alphabetical characters are removed from the string in a locale-dependent fashion. This strips spaces, hyphens, and numbers. Other letters, such as "Ü," may be permissible in the current locale but are unknown to `onca`. For inputs outside of its known range, the output is undefined and NA is returned and a warning is issued. If `clean` is FALSE, `onca` attempts to process the strings. The default is TRUE.

Value

the ONCA encoded character vector

References

Gill, L. E. (1997). "OX-LINK: The Oxford Medical Record Linkage System." In *Record Linkage Techniques—1997*, p. 15–33.

James P. Howard, II, "Phonetic Spelling Algorithm Implementations for R," *Journal of Statistical Software*, vol. 95, no. 8, (2020), p. 1–21, <10.18637/jss.v095.i08>.

See Also

Other phonics: [caverphone\(\)](#), [cologne\(\)](#), [lein\(\)](#), [metaphone\(\)](#), [mra_encode\(\)](#), [nysiis\(\)](#), [phonex\(\)](#), [phonics\(\)](#), [rogerroot\(\)](#), [soundex\(\)](#), [statcan\(\)](#)

Examples

```
onca("William")
onca(c("Peter", "Peedy"))
onca("Stevenson", maxCodeLen = 8)
```

phonex

Phonex Name Coding

Description

The Phonex name coding procedure.

Usage

```
phonex(word, maxCodeLen = 4, clean = TRUE)
```

Arguments

<code>word</code>	string or vector of strings to encode
<code>maxCodeLen</code>	maximum length of the resulting encodings, in characters
<code>clean</code>	if TRUE, return NA for unknown alphabetical characters

Details

The variable `word` is the name to be encoded. The variable `maxCodeLen` is the limit on how long the returned name code should be. The default is 4.

The phonex algorithm is only defined for inputs over the standard English alphabet, *i.e.*, "A-Z," "Ä," "Ö," "Ü," and "ß." Non-alphabetical characters are removed from the string in a locale-dependent fashion. This strips spaces, hyphens, and numbers. Other letters, such as "ç," may be permissible in the current locale but are unknown to phonex. For inputs outside of its known range, the output is undefined and NA is returned and a warning is issued. If `clean` is FALSE, phonex attempts to process the strings. The default is TRUE.

Value

the Phonex encoded character vector

References

James P. Howard, II, "Phonetic Spelling Algorithm Implementations for R," *Journal of Statistical Software*, vol. 95, no. 8, (2020), p. 1–21, <10.18637/jss.v095.i08>.

A.J. Lait and Brian Randell. "An assessment of name matching algorithms." Technical Report Series-University of Newcastle Upon Tyne Computing Science (1996).

See Also

Other phonics: [caverphone\(\)](#), [cologne\(\)](#), [lein\(\)](#), [metaphone\(\)](#), [mra_encode\(\)](#), [nysiis\(\)](#), [onca\(\)](#), [phonics\(\)](#), [rogerroot\(\)](#), [soundex\(\)](#), [statcan\(\)](#)

Examples

```
phonex("William")
phonex(c("Peter", "Peedy"))
phonex("Stevenson", maxCodeLen = 8)
```

phonics

Phonetic Spelling Algorithms

Description

Apply one or more phonetic encoders to a character vector and return their codes together in a data frame.

Usage

```
phonics(word, method, clean = TRUE)
```

Arguments

word	A character vector to encode.
method	A character vector naming the methods to apply.
clean	If TRUE, warn and return NA for inputs containing characters outside an encoder's supported alphabet. If FALSE, unsupported characters are discarded before encoding.

Details

Available methods are "caverphone", "caverphone.modified", "cologne", "lein", "metaphone", "nysiis", "nysiis.modified", "onca", "onca.modified", "onca.refined", "onca.modified.refined", "phonex", "rogerroot", "soundex", "soundex.refined", and "statcan". Unknown method names produce a warning and no result column.

Value

A data frame whose first column is word, followed by one character column for each recognized method.

References

Howard, J. P., II (2020). "Phonetic Spelling Algorithm Implementations for R." *Journal of Statistical Software*, 95(8), 1–21. doi:10.18637/jss.v095.i08.

See Also

Other phonics: [caverphone\(\)](#), [cologne\(\)](#), [lein\(\)](#), [metaphone\(\)](#), [mra_encode\(\)](#), [nysiis\(\)](#), [onca\(\)](#), [phonex\(\)](#), [rogerroot\(\)](#), [soundex\(\)](#), [statcan\(\)](#)

Examples

```
phonics(c("Peter", "Peedy"), c("soundex", "soundex.refined"))
```

rogerrootRoger Root Name Coding Procedure

Description

Provides the Roger Root name coding system

Usage

```
rogerroot(word, maxCodeLen = 5, clean = TRUE)
```

Arguments

word	string or vector of strings to encode
maxCodeLen	maximum length of the resulting encodings, in characters
clean	if TRUE, return NA for unknown alphabetical characters

Details

The `rogerroot` function phonetically encodes the given string using the Roger Root algorithm. The variable `word` is a string or vector of strings to encode.

The variable `maxCodeLen` is the limit on how long the returned code should be. The default is 5.

The `rogerroot` algorithm is only defined for inputs over the standard English alphabet, *i.e.*, "A-Z.". Non-alphabetical characters are removed from the string in a locale-dependent fashion. This strips spaces, hyphens, and numbers. Other letters, such as "Ü," may be permissible in the current locale but are unknown to `rogerroot`. For inputs outside of its known range, the output is undefined and NA is returned and a warning this thrown. If `clean` is FALSE, `rogerroot` attempts to process the strings. The default is TRUE.

Value

the Roger Root encoded character vector

References

James P. Howard, II, "Phonetic Spelling Algorithm Implementations for R," *Journal of Statistical Software*, vol. 95, no. 8, (2020), p. 1–21, <10.18637/jss.v095.i08>.

Billy T. Lynch and William L. Arends, *Selection of a Surname Coding Procedure for the SRS Record Linkage System*, United States Department of Agriculture, 1977, Appendix B.

See Also

Other phonics: [caverphone\(\)](#), [cologne\(\)](#), [lein\(\)](#), [metaphone\(\)](#), [mra_encode\(\)](#), [nysiis\(\)](#), [onca\(\)](#), [phonex\(\)](#), [phonics\(\)](#), [soundex\(\)](#), [statcan\(\)](#)

Examples

```

rogerroot("William")
rogerroot(c("Peter", "Peady"))
rogerroot("Stevenson")

```

soundex

Soundex

Description

The Soundex phonetic algorithms

Usage

```

soundex(word, maxCodeLen = 4L, clean = TRUE)

refinedSoundex(word, maxCodeLen = 10L, clean = TRUE)

```

Arguments

word	string or vector of strings to encode
maxCodeLen	maximum length of the resulting encodings, in characters
clean	if TRUE, return NA for unknown alphabetical characters

Details

The function `soundex` phonetically encodes the given string using the U.S. census Soundex rules. The function `refinedSoundex` implements the U.S.-English mapping and duplicate handling from Apache Commons Codec's Refined Soundex.

The variable `maxCodeLen` is the limit on how long the returned soundex should be.

The `soundex` and `refinedSoundex` algorithms are only defined for inputs over the standard English alphabet, *i.e.*, "A-Z." Non-alphabetical characters are removed from the string in a locale-dependent fashion. This strips spaces, hyphens, and numbers. Other letters, such as "Ü," may be permissible in the current locale but are unknown to `soundex` and `refinedSoundex`. For inputs outside of its known range, the output is undefined and NA is returned and a warning is issued. If `clean` is FALSE, `soundex` and `refinedSoundex` attempts to process the strings. The default is TRUE.

Value

soundex encoded character vector

Caveats

The `soundex` and `refinedSoundex` algorithms are only defined for inputs over the standard English alphabet, *i.e.*, "A-Z." For inputs outside this range, the output is undefined.

References

Charles P. Bourne and Donald F. Ford, "A study of methods for systematically abbreviating English words and names," *Journal of the ACM*, vol. 8, no. 4 (1961), p. 538-552.

James P. Howard, II, "Phonetic Spelling Algorithm Implementations for R," *Journal of Statistical Software*, vol. 95, no. 8, (2020), p. 1–21, <10.18637/jss.v095.i08>.

Howard B. Newcombe, James M. Kennedy, "Record linkage: making maximum use of the discriminating power of identifying information," *Communications of the ACM*, vol. 5, no. 11 (1962), p. 563-566.

United States National Archives and Records Administration, "The Soundex Indexing System."

Apache Software Foundation, *Apache Commons Codec: RefinedSoundex*.

See Also

Other phonics: [caverphone\(\)](#), [cologne\(\)](#), [lein\(\)](#), [metaphone\(\)](#), [mra_encode\(\)](#), [nysiis\(\)](#), [onca\(\)](#), [phonex\(\)](#), [phonics\(\)](#), [rogerroot\(\)](#), [statcan\(\)](#)

Examples

```
soundex("wheel")
soundex(c("school", "benji"))
```

statcan

Statistics Canada Name Coding

Description

The modified Statistics Canada name coding procedure

Usage

```
statcan(word, maxCodeLen = 4, clean = TRUE)
```

Arguments

word	string or vector of strings to encode
maxCodeLen	maximum length of the resulting encodings, in characters
clean	if TRUE, return NA for unknown alphabetical characters

Details

The variable `word` is the name to be encoded. The variable `maxCodeLen` is the limit on how long the returned name code should be. The default is 4.

The `statcan` algorithm is only defined for inputs over the standard French alphabet. Non-alphabetical characters are removed from the string in a locale-dependent fashion. This strips spaces, hyphens, and numbers. Other letters, such as "Ü," may be permissible in the current locale but are unknown to `statcan`. For inputs outside of its known range, the output is undefined and NA is returned and a warning is issued. If `clean` is FALSE, `statcan` attempts to process the strings. The default is TRUE.

Value

the Statistics Canada encoded character vector

References

James P. Howard, II, "Phonetic Spelling Algorithm Implementations for R," *Journal of Statistical Software*, vol. 95, no. 8, (2020), p. 1–21, <10.18637/jss.v095.i08>.

Billy T. Lynch and William L. Arends. "Selection of surname coding procedure for the SRS record linkage system." United States Department of Agriculture, Sample Survey Research Branch, Research Division, Washington, 1977.

See Also

Other phonics: [caverphone\(\)](#), [cologne\(\)](#), [lein\(\)](#), [metaphone\(\)](#), [mra_encode\(\)](#), [nysiis\(\)](#), [onca\(\)](#), [phonex\(\)](#), [phonics\(\)](#), [rogerroot\(\)](#), [soundex\(\)](#)

Examples

```
statcan("William")
statcan(c("Peter", "Peady"))
statcan("Stevenson", maxCodeLen = 8)
```


Index

* **phonics**

- caverphone, [2](#)
- cologne, [3](#)
- lein, [4](#)
- metaphone, [6](#)
- mra_encode, [7](#)
- nysiis, [8](#)
- onca, [9](#)
- phonex, [10](#)
- phonics, [12](#)
- rogerroot, [13](#)
- soundex, [14](#)
- statcan, [15](#)

caverphone, [2](#)
caverphone(), [4–6](#), [8–13](#), [15](#), [16](#)
cologne, [3](#)
cologne(), [3](#), [5](#), [6](#), [8–13](#), [15](#), [16](#)

lein, [4](#)
lein(), [3](#), [4](#), [6](#), [8–13](#), [15](#), [16](#)

metaphone, [6](#)
metaphone(), [3–5](#), [8–13](#), [15](#), [16](#)
mra_compare (mra_encode), [7](#)
mra_encode, [7](#)
mra_encode(), [3–6](#), [9–13](#), [15](#), [16](#)

nysiis, [8](#)
nysiis(), [3–6](#), [8](#), [10–13](#), [15](#), [16](#)

onca, [9](#)
onca(), [3–6](#), [8](#), [9](#), [11–13](#), [15](#), [16](#)

phonex, [10](#)
phonex(), [3–6](#), [8–10](#), [12](#), [13](#), [15](#), [16](#)
phonics, [12](#)
phonics(), [3–6](#), [8–11](#), [13](#), [15](#), [16](#)

refinedSoundex (soundex), [14](#)
rogerroot, [13](#)

rogerroot(), [3–6](#), [8–12](#), [15](#), [16](#)

soundex, [14](#)
soundex(), [3–6](#), [8–13](#), [16](#)
statcan, [15](#)
statcan(), [3–6](#), [8–13](#), [15](#)