

Package ‘salty’

October 14, 2022

Type Package

Title Turn Clean Data into Messy Data

Version 0.1.0

Description Take real or simulated data and salt it with errors commonly found in the wild, such as pseudo-OCR errors, Unicode problems, numeric fields with nonsensical punctuation, bad dates, etc.

License MIT + file LICENSE

Depends R (>= 2.10)

Imports assertthat, purrr, stringr

Suggests charlatan, testthat (>= 2.0.0), tibble, covr

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

URL <https://github.com/mdlincoln/salty>

BugReports <https://github.com/mdlincoln/salty/issues>

NeedsCompilation no

Author Matthew Lincoln [aut, cre] (<<https://orcid.org/0000-0002-4387-3384>>)

Maintainer Matthew Lincoln <matthew.d.lincoln@gmail.com>

Repository CRAN

Date/Publication 2018-09-17 11:40:03 UTC

R topics documented:

inspect_shaker	2
p_indices	2
salt	3
salty	4
salt_delete	5
salt_insert	5

Value

An integer vector of indices.

salt

Salt vectors with common data problems

Description

These are easy-to-use wrapper functions that call either [salt_insert](#) (for including new characters) or [salt_replace](#) (for salting that requires replacement of specific characters) with sane defaults.

Usage

```
salt_punctuation(x, p = 0.2, n = 1)
```

```
salt_letters(x, p = 0.2, n = 1)
```

```
salt_whitespace(x, p = 0.2, n = 1)
```

```
salt_digits(x, p = 0.2, n = 1)
```

```
salt_ocr(x, p = 0.2, rep_p = 0.1)
```

```
salt_capitalization(x, p = 0.1, rep_p = 0.1)
```

```
salt_decimal_commas(x, p = 0.1, rep_p = 0.1)
```

Arguments

x	A vector. This will always be coerced to character during salting.
p	A number between 0 and 1. Percent of values in x that should be salted.
n	A positive integer. Number of times to add new values from insertions into selected values in x manually supply your own list of characters.
rep_p	A number between 0 and 1. Probability that a given match should be replaced in one of the selected values.

Details

For a more fine-grained control over how characters are added and whether , see the documentation for [salt_insert](#), [salt_substitute](#), [salt_replace](#), and [salt_delete](#).

Functions

- `salt_punctuation`: Punctuation characters
- `salt_letters`: Upper- and lower-case letters
- `salt_whitespace`: Spaces
- `salt_digits`: 0-9
- `salt_ocr`: Replace some substrings with common OCR problems
- `salt_capitalization`: Flip capitalization of letters
- `salt_decimal_commas`: Flip decimals to commas and vice versa

salty

salty: Turn Clean Data Into Messy Data

Description

Insert, delete, replace, and substitute bits of your data with messy values.

Details

Convenient wrappers such as [salt_punctuation](#) are provided for quick access to this package's functionality with simple defaults. For more fine-grained control, use one of the underlying `salt_` functions:

- [salt_insert](#) will insert new characters into some of the values of `x`. All the original characters of the original values will be maintained.
- [salt_substitute](#) will substitute some characters in some of the values of `x` in place of some of the original characters.
- [salt_replace](#) will replace some characters in some of the values of `x`. Unlike [salt_substitute](#), [salt_replace](#) does conditional replacement dependent on the original values of `x`, such as changing capitalization or simulating OCR errors based on certain character combinations.
- [salt_delete](#) will remove some characters in the values of `x`
- [salt_na](#) and [salt_empty](#) will replace some values of `x` with NA or with empty strings.
- [salt_swap](#) replaces entire values of `x` with new strings

salt_delete	<i>Delete some characters from some values</i>
-------------	--

Description

Delete some characters from some values

Usage

```
salt_delete(x, p = 0.2, n = 1)
```

Arguments

x	A vector. This will always be coerced to character during salting.
p	A number between 0 and 1. Percent of values in x that should be salted.
n	A positive integer. Number of times to add new values from insertions into selected values in x manually supply your own list of characters.

Value

A character vector the same length as x

Examples

```
x <- c("Lorem ipsum dolor sit amet, consectetur adipiscing elit.",  
      "Nunc finibus tortor a elit eleifend interdum.",  
      "Maecenas aliquam augue sit amet ultricies placerat.")  
  
salt_delete(x, p = 0.5, n = 5)  
  
salt_empty(x, p = 0.5)  
  
salt_na(x, p = 0.5)
```

salt_insert	<i>Insert new characters into some values in a vector</i>
-------------	---

Description

Inserts a selection of characters into a percentage of values in the supplied vector.

Usage

```
salt_insert(x, insertions, p = 0.2, n = 1)
```

Arguments

x	A vector. This will always be coerced to character during salting.
insertions	A shaker function, or a character vector.
p	A number between 0 and 1. Percent of values in x that should be salted.
n	A positive integer. Number of times to add new values from insertions into selected values in x manually supply your own list of characters.

Value

A character vector the same length as x

salt_na	<i>Remove entire values from a vector</i>
---------	---

Description

Remove entire values from a vector

Usage

```
salt_na(x, p = 0.2)
```

```
salt_empty(x, p = 0.2)
```

Arguments

x	A vector
p	A number between 0 and 1. Proportion of values to edit.

Value

A vector the same length as x

salt_replace	<i>Replace certain patterns into some values in a vector</i>
--------------	--

Description

Inserts a selection of characters into some values of `x`. Pair `salt_replace` with the named vectors in `replacement_shaker`, or supply your own named vector of replacements. The convenience functions `salt_ocr` and `salt_capitalization` are light wrappers around `salt_replace`.

Usage

```
salt_replace(x, replacements, p = 0.1, rep_p = 0.5)
```

Arguments

<code>x</code>	A vector. This will always be coerced to character during salting.
<code>replacements</code>	A <code>replacement_shaker</code> function, or a named character vector of patterns and replacements.
<code>p</code>	A number between 0 and 1. Percent of values in <code>x</code> that should be salted.
<code>rep_p</code>	A number between 0 and 1. Probability that a given match should be replaced in one of the selected values.

Value

A character vector the same length as `x`

Examples

```
x <- c("Lorem ipsum dolor sit amet, consectetur adipiscing elit.",
      "Nunc finibus tortor a elit eleifend interdum.",
      "Maecenas aliquam augue sit amet ultricies placerat.")

salt_replace(x, replacement_shaker$capitalization, p = 0.5, rep_p = 0.2)

salt_ocr(x, p = 1, rep_p = 0.5)
```

salt_substitute *Substitute certain characters in a vector*

Description

Substitute certain characters in a vector

Usage

```
salt_substitute(x, substitutions, p = 0.2, n = 1)
```

Arguments

x	A vector. This will always be coerced to character during salting.
substitutions	Values to be substituted in
p	A number between 0 and 1. Percent of values in x that should be salted.
n	A positive integer. Number of times to add new values from insertions into selected values in x manually supply your own list of characters.

Value

A character vector the same length as x

Examples

```
x <- c("Lorem ipsum dolor sit amet, consectetur adipiscing elit.",  
      "Nunc finibus tortor a elit eleifend interdum.",  
      "Maecenas aliquam augue sit amet ultricies placerat.")  
  
salt_substitute(x, shaker$digits, p = 0.5, n = 5)
```

salt_swap *Randomly swap out entire values in a vector*

Description

Because swaps can be provided by either a character vector or a function that returns a character vector, `salt_swap` can be fruitfully used in conjunction with the [charlatan::charlatan](#) package to intersperse real data with simulated data.

Usage

```
salt_swap(x, swaps, p = 0.2)
```


Arguments

x A vector. This will always be coerced to character during salting.
swaps Values to be swapped out
p A number between 0 and 1. Percent of values in x that should be salted.

Value

A character vector the same length as x

Examples

```
x <- c("Lorem ipsum dolor sit amet, consectetur adipiscing elit.",  
      "Nunc finibus tortor a elit eleifend interdum.",  
      "Maecenas aliquam augue sit amet ultricies placerat.")  
  
new_values <- c("foo", "bar", "baz")  
  
salt_swap(x, swaps = new_values, p = 0.5)
```

shaker

Get a set of values to use in salt_functions

Description

[shaker](#) contains various character sets to be added to your data using [salt_insert](#) and [salt_substitute](#). [replacement_shaker](#) is for [salt_replace](#), and contains pairlists that replace matched patterns in your data.

Usage

```
shaker  
  
replacement_shaker  
  
available_shakers()
```

Format

An object of class list of length 6.

Value

A sampling function that will be called by [salt_insert](#), [salt_substitute](#), or [salt_replace](#).

Examples

```
salt_insert(letters, shaker$punctuation)  
available_shakers()
```

Index

* datasets

shaker, 9

available_shakers (shaker), 9

charlatan::charlatan, 8

inspect_shaker, 2

p_indices, 2

replacement_shaker, 7, 9

replacement_shaker (shaker), 9

salt, 3

salt_capitalization, 7

salt_capitalization (salt), 3

salt_decimal_commas (salt), 3

salt_delete, 3, 4, 5

salt_digits (salt), 3

salt_empty, 4

salt_empty (salt_na), 6

salt_insert, 3, 4, 5, 9

salt_letters (salt), 3

salt_na, 4, 6

salt_ocr, 7

salt_ocr (salt), 3

salt_punctuation, 4

salt_punctuation (salt), 3

salt_replace, 3, 4, 7, 7, 9

salt_substitute, 3, 4, 8, 9

salt_swap, 4, 8

salt_whitespace (salt), 3

salty, 4

salty-package (salty), 4

shaker, 2, 6, 9, 9