Package ‘litteR’

July 26, 2019

Title Litter Analysis
Version 0.4.1
Date 2019-07-23

Description Data sets on various litter types like beach litter, riverain litter, floating litter, and seafloor litter are rapidly growing. This package offers a simple user interface to analyse these litter data in a consistent and reproducible way. It also provides functions to facilitate several kinds of litter analysis, e.g., trend analysis, power analysis, and baseline analysis. Under the hood, these functions are also used by the user interface. See Schulz et al. (2019) <doi:10.1016/j.envpol.2019.02.030> for details.

Depends R (>= 3.5.0)
Imports readr (>= 1.3.1), stringr (>= 1.4.0), dplyr (>= 0.8.3), tidyr (>= 0.8.3), fs (>= 1.3.1), ggplot2 (>= 3.2.0), purrr (>= 0.3.2), rlang (>= 0.4.0), yaml (>= 2.2.0), rmarkdown (>= 1.14), tcltk

Suggests knitr, kableExtra, testthat (>= 2.1.0)

License GPL (>= 3)

Encoding UTF-8
LazyData true

VignetteBuilder knitr

RoxygenNote 6.1.1

NeedsCompilation no

Author Dennis Walvoort [aut, cre, cph], Willem van Loon [aut, cph], Rijkswaterstaat - The Netherlands [cph, fnd, dtc]

Maintainer Dennis Walvoort <dennis.Walvoort@wur.nl>

Repository CRAN

Date/Publication 2019-07-26 11:00:02 UTC
# R topics documented:

litteR-package .................................................. 2
adj_boxplot_stats ............................................... 2
as_internal_format ............................................. 3
create_litter_project ......................................... 4
cv ................................................................. 4
enumerate ......................................................... 5
enumerate.character ........................................... 5
get_group_code .................................................. 6
get_signature ................................................... 7
get_type_code ................................................... 7
intercept .......................................................... 8
iod ................................................................. 9
is_date_format .................................................... 9
is_group_code ................................................... 10
is_natural_number .............................................. 10
is_type_code .................................................... 11
is_type_name .................................................... 11
litter .............................................................. 12
mann_kendall ..................................................... 13
medcouple ........................................................ 14
power .............................................................. 15
power.wilcoxon .................................................. 15
p_value ............................................................ 16
read_litter ......................................................... 17
read_litter_groups ............................................. 17
recdf ............................................................... 18
rmad ............................................................... 18
roll ................................................................. 19
sequenize ........................................................ 19
sequenize.integer .............................................. 20
slope ............................................................... 20
test_statistic ..................................................... 21
theil_sen .......................................................... 21
top ................................................................. 22
toString.sequenized .......................................... 23
trimean ............................................................. 23
validate ........................................................... 24
wilcoxon ........................................................... 25
litteR-package

Description

A tool for the analysis of various litter types, e.g., beach litter, riverain litter, floating litter, and seafloor litter.

Details

The easiest way to get convenient with litteR is to create an empty project directory and fill it with example files by calling the function `create_litter_project`. The workhorse function in litteR is called `litter`. This function will start a simple user interface and lets you select an input file (*.csv) and a settings file (*.yaml). It will produce an HTML-report with litter analysis results according to the selected options in the settings file. See the package vignette for more details.

Author(s)

Maintainer: Dennis Walvoort <dennis.Walvoort@wur.nl> [copyright holder]

Authors:

• Willem van Loon <willem.van.loon@rws.nl> [copyright holder]

Other contributors:

• Rijkswaterstaat - The Netherlands [copyright holder, funder, data contributor]

References


adj_boxplot_stats Adjusted Boxplot Statistics

Description

Adjusted boxplot statistics according to Hubert & Vandervieren (2008). The upper whisker extends from the hinge to the largest value no further than the upper fence. Similarly, the lower whisker extends from the hinge to the smallest value no further than the lower fence. See Hubert & Vandervieren (2008, p.5191, Eq.5).
Usage

adj_boxplot_stats(x, ...)

## Default S3 method:
adj_boxplot_stats(x, ...)

Arguments

x         numeric vector
...       further arguments passed to or from other methods.

Value

Numeric vector consisting of respectively the lower whisker/fence, the first quartile/hinge, the median, the third quartile/hinge, and the upper whisker/fence.

Methods (by class)

- default: Adjusted Boxplot Statistics

References


as_internal_format  Signature from Files

Description

Generic function for retrieving the file signature

Usage

as_internal_format(x, ...)

## S3 method for class 'ospar'
as_internal_format(x, ...)

## S3 method for class 'wide'
as_internal_format(x, ...)

## S3 method for class 'long'
as_internal_format(x, ...)

1https://doi.org/10.1016/j.csda.2007.11.008
create_litter_project

Arguments

\( x \) object of which the signature has to be determined

... further arguments passed to or from other methods.

Methods (by class)

- \texttt{ospar}: convert OSPAR data to internal representation.
- \texttt{wide}: convert wide data to internal representation.
- \texttt{long}: convert long data to internal representation.

---

create_litter_project

\textit{Create Project Directory}

---

Description

Fills an empty directory (\texttt{path}) with example files. If the \texttt{path} argument is missing or NULL, a Tcl/Tk dialogue will be started.

Usage

create_litter_project(path = NULL)

Arguments

\texttt{path} (Existing) directory name

---

cv

\textit{Coefficient of Variation}

---

Description

Coefficient of Variation

Usage

cv(x, na.rm = FALSE)

Arguments

\texttt{x} a numeric vector

\texttt{na.rm} logical. Should missing values be removed?
Value

coefficient of variation (numeric vector of length 1).

References


---

**enumerate**  

*Enumerate Objects*

**Description**

Generic function for enumerating objects

**Usage**

```r
classify(x, ...)
```

**Arguments**

x  
object to enumerate

...  
further arguments passed to or from other methods.

**See Also**

*enumerate.character*

---

**enumerate.character**  

*Enumerate Character Vector*

**Description**

Collapsing a character vector of length n, to a character vector of length 1.

**Usage**

```r
## S3 method for class 'character'
classify(x, ...)
```

**Arguments**

x  
character vector

...  
further arguments passed to or from other methods.
**get_group_code**

**Value**

character vector of length 1, with elements separated by a comma except for the last element which is prepended by "and".

**Examples**

```r
enumerate("apples")
enumerate(c("apples", "oranges"))
enumerate(c("apples", "oranges", "pears"))
```

---

**get_group_code Extract Litter Group Code**

**Description**

Extracts litter group code (i.e., a code in square brackets), from a character vector.

**Usage**

```r
group_code(x)
```

**Arguments**

- `x` character vector containing litter group codes

**Value**

character vector of litter group codes

**Examples**

```r
# valid litter type
stopifnot(get_group_code("[TA]") == "[TA]")
stopifnot(get_group_code("all kinds of plastic [PLASTIC]") == "[PLASTIC]")
stopifnot(is.na(get_group_code("all kinds of plastic [Plastic]")))
```
get_signature  

Detect File Signature

Description

Generic function for detecting the file signature.

Usage

get_signature(x, ...)

## S3 method for class 'character'
get_signature(x, ...)

## S3 method for class 'tbl'
get_signature(x, ...)

## S3 method for class 'fs_path'
get_signature(x, ...)

Arguments

x          object of which the signature has to be determined
...        further arguments passed to or from other methods.

Value

Detected file signature. Currently "OSPAR", "long", and "wide" are recognized. In case of an unknown file signature, "unknown" is returned (character vector of length 1).

Methods (by class)

- character: detect file signature from character.
- tbl: detect file signature from a tibble (tbl).
- fs_path: detect file signature from fs_path.

get_type_code  

Extract Litter Code

Description

Extract litter codes (ASCII characters in square brackets) from character vector x.

Usage

get_type_code(x)
intercept

Arguments

x character vector containing litter codes

Value

litter code (character vector).

Examples

# valid litter type
stopifnot(get_type_code("Plastic: Food [6]") == "[6]"
stopifnot(get_type_code(c("Plastic: Food [6]", "Plastic: Shoes [44]"))) == c("[6]", "[44]"))

# invalid litter type: additional punctuation : and + are not allowed
stopifnot(is.na(get_type_code("All cartons/tetrapaks [302:204+62+118]"))
stopifnot(is.na(get_type_code("[TA]")))

intercept Intercept

Description

Extract the intercept from object x.

Usage

intercept(x, ...)

Arguments

x object

... further arguments passed to or from other methods.

Value

estimate of the intercept (numeric vector of length 1).
iod  

*Index of Dispersion*

**Description**
A normalized measure of the dispersion of a probability distribution.

**Usage**

`iod(x, na.rm = FALSE)`

**Arguments**

- `x`: a numeric vector
- `na.rm`: logical. Should missing values be removed?

**Value**
index of dispersion (numeric vector of length 1).

**References**

---

is_date_format  

*Check Date Format*

**Description**
Checks if the data format `x` complies with `format`.

**Usage**

`is_date_format(x, format)`

**Arguments**

- `x`: object of class character or Date
- `format`: required date format (see `strptime`)

**Value**
TRUE if `x` complies with `format`, and FALSE otherwise.

**Examples**

`is_date_format("2019-05-14", "%Y-%m-%d")`
**is_group_code**  
*Test For Group Code*

---

**Description**
Test if \( x \) contains a valid litter group code.

**Usage**
\[
is\_group\_code(x)
\]

**Arguments**
\[
x \quad \text{character vector to test}
\]

**Value**
TRUE if test passes, FALSE otherwise.

---

**is_natural_number**  
*Test for Natural Numbers*

---

**Description**
Test for natural numbers according to ISO 80000-2, that is the set 0, 1, 2, ...

**Usage**
\[
is\_natural\_number(x)
\]

**Arguments**
\[
x \quad \text{numeric vector}
\]

**Value**
TRUE in case \( x \) is a natural number, FALSE otherwise.

**Examples**
\[
\text{stopifnot(!is\_natural\_number(3.1))}
\text{stopifnot(!is\_natural\_number(2.99))}
\text{stopifnot(is\_natural\_number(3))}
\text{stopifnot(all(is\_natural\_number(0:9)))}
\text{stopifnot(sum(is\_natural\_number(c(1, 2.5, 3))) == 2)}
\]
**is_type_code**  
*Test For Type Code*

**Description**
Test if \( x \) contains a valid litter type code.

**Usage**

\[
is_{\text{type}}(x)
\]

**Arguments**
\( x \) character vector to test

**Value**
TRUE if test passes, FALSE otherwise.

---

**is_type_name**  
*Test Litter Data by Name*

**Description**
Checks if litter names are OSPAR compliant. The OSPAR format consists of a litter category, a specification, and an integer code in the range 000-989 in square brackets. In addition the special code [TA] is allowed to specify total abundance.

**Usage**

\[
is_{\text{type}}(x)
\]

**Arguments**
\( x \) character vector to check

**Value**
TRUE if \( x \) complies with OSPAR, FALSE if not.
Examples

```r
# valid litter type
stopifnot(is_type_name("Plastic: Food [6]"))

# invalid litter type: additional punctuation : and + are not allowed
stopifnot(!is_type_name("All cartons/tetrapaks [302:204+62+118]"))

# invalid litter type: numeric litter code is missing
stopifnot(!is_type_name("no litter here"))

# invalid litter type: number greater than 989
stopifnot(!is_type_name("Survey: Remarks [999]"))
stopifnot(!is_type_name("[TA]"))
```

---

**litter**  
*Graphical User Interface to the litteR-package*

---

**Description**

Starts a graphical user interface for analysing litter data. A Tcl/Tk-dialogue will be started if one or more arguments are missing.

**Usage**

```r
litter(file_settings = NULL, file_input = NULL, dir_output = NULL)
```

**Arguments**

- `file_settings`: file containing settings
- `file_input`: file containing litter data
- `dir_output`: directory to store output

**Value**

An HTML-document in which all the litter analysis results (tables, figures, explanatory text) are reported.
mann_kendall  Mann Kendall

Description

Performs Mann-Kendall non-parametric trend test.

Usage

mann_kendall(x, type = c("both", "increasing", "decreasing"))

## S3 method for class 'mann_kendall'
 test_statistic(x, ...)

## S3 method for class 'mann_kendall'
 p_value(x, ...)

Arguments

x                 numeric vector representing a time-series.

 type               direction to test (both, increasing, or decreasing).

...               further arguments passed to or from other methods.

Value

object of class Mann-Kendall.

Methods (by generic)

• test_statistic: Extracts Mann Kendall tau
• p_value: Extract p-value

See Also

test_statistic, p_value, cor.test

Examples

# create mann_kendall object
mk <- mann_kendall(c(9, 4, 7, 5, 3), type = "decreasing")

# get test statistic tau
 test_statistic(mk)

# get p-value
p_value(mk)
**medcouple**  

---

**Description**

Robust statistic that quantifies the skewness of univariate distributions.

**Usage**

`medcouple(x, ...)`

## Default S3 method:
`medcouple(x, ...)`

**Arguments**

- `x` numeric vector
- `...` further arguments passed to or from other methods.

**Value**

`medcouple` (numeric vector of length 1).

**Methods (by class)**

- `default`: default method

**Note**

This is a naive, but robust en simple implementation. For a more efficient implementation see package robustbase\(^2\) and the references section below.

**References**


---

\(^2\)https://CRAN.R-project.org/package=robustbase  
\(^3\)https://doi.org/10.1198/106186004X12632
**power**  
*Statistical Power*

**Description**
Estimate the statistical power of a test.

**Usage**
```
power(x, ...)```

**Arguments**
- `x`  
  object  
- `...`  
  further arguments passed to or from other methods.

**Value**
power of a test (numeric vector of length 1).

**References**
https://en.wikipedia.org/wiki/Statistical_power

---

**power.wilcoxon**  
*Power of Wilcoxon Test*

**Description**
Estimates the power of a Wilcoxon test by means of Monte Carlo simulation.

**Usage**
```
## S3 method for class 'wilcoxon'
power(x, n = 10, alpha = 0.05, n_sim = 1000, ...)```

**Arguments**
- `x`  
  numeric vector  
- `n`  
  number of samples of `x`  
- `alpha`  
  significance level  
- `n_sim`  
  number of Monte Carlo samples  
- `...`  
  further arguments passed to or from other methods.
p_value

Value

p-value of a test (numeric vector of length 1).

References


Examples

# continuous variable
x <- rnorm(n = 25, mean = 1, sd = 2)
w <- wilcoxon(x, mu = 2, type = "less")
p <- power(w, n = 10, alpha = 0.05)

# discrete variable
x <- rpois(n = 10, lambda = 50)
w <- wilcoxon(x, mu = 60, type = "less")
p <- power(w, n = 10, alpha = 0.05)

Description

Extract p-value.

Usage

p_value(x, ...)

Arguments

x object
...

further arguments passed to or from other methods.

Value

p-value of a test (numeric vector of length 1).
### read_litter

**Read Litter Data**

**Description**

Reads litter data from various formats. Currently only the OSPAR data snapshot format, a long format, and a wide format are supported. See the package vignette for more details.

**Usage**

```r
read_litter(file)
```

**Arguments**

- `file` name of litter file

**Value**

- `tibble` with litter data in long format

### read_litter_groups

**Read Litter Groups**

**Description**

Read the file that links litter codes to litter groups. See the package vignette for more details.

**Usage**

```r
read_litter_groups(file = path_package("litteR", "extdata", "litter-groups.csv"))
```

**Arguments**

- `file` name of litter group file

**Value**

- `tibble` with look-up-table of litter groups
**recdf**  

*Sample From an ECDF*

**Description**
Type stable implementation of an Empirical Cumulative Distribution Function (ECDF) sampler.

**Usage**
```r
recdf(x, n)
```

**Arguments**
- `x`: numeric vector
- `n`: number of draws

**Value**
vector of `n` elements of the same type as `x`

**See Also**
ecdf

**Examples**
```r
recdf(1:5, 10)
```

---

**rmad**  

*Relative Median Absolute Deviation*

**Description**
This is the Median Absolute Deviation divided by the median and is similar to the coefficient of variation.

**Usage**
```r
rmad(x, na.rm = FALSE)
```

**Arguments**
- `x`: a numeric vector
- `na.rm`: logical. Should missing values be removed?
**Value**

Relative median absolute deviation (numeric vector of length 1).

**References**

https://en.wikipedia.org/wiki/Median_absolute_deviation

---

**roll**  
*Rolling Statistics*

---

**Description**

Applies function `fun` within a rolling (moving) window of size `w` to vector numeric vector `x`.

**Usage**

`roll(x, w = 3, fun = mean)`

**Arguments**

- `x`: numeric vector (time-series)
- `w`: width of moving window
- `fun`: function to be applied

**Value**

vector of length `length(x)-w`

---

**sequenize**  
*Sequenize Objects*

---

**Description**

Generic function for sequenizing objects

**Usage**

`sequenize(x, ...)`

**Arguments**

- `x`: object to sequenize
- `...`: further arguments passed to or from other methods.

**See Also**

`sequenize.integer`
sequenize.integer  Sequenize Integer Sequence

Description

Compression of integer sequences to 'start-end' notation. For instance c(1:5, 8:9) becomes "1-5, 8-9".

Usage

## S3 method for class 'integer'
sequenize(x, ...)

Arguments

x  vector of integers.
...

Value

object of class sequenized

Note

The elements of x should be unique and in ascending order.

See Also

toString

Examples

toString(sequenize(c(1:4, 8:9)))

slope  Slope

Description

Extract slope.

Usage

slope(x, ...)

test_statistic  Test Statistic

Description
Extract test_statistic.

Usage
test_statistic(x, ...)

Arguments
x  object
... further arguments passed to or from other methods.

Value
test statistic of a test (numeric vector of length 1).

See Also
test_statistic.wilcoxon, test_statistic.mann_kendall

theil_sen  Theil Sen Slope Estimator

Description
Theil Sen Slope Estimator

Usage
theil_sen(x, y, ...)

## S3 method for class 'theil_sen'
slope(x, ...)

## S3 method for class 'theil_sen'
intercept(x, ...)
Arguments

- `x`: time vector (numeric, or Date).
- `y`: numeric value.
- `...`: further arguments passed to or from other methods.

Value

Object of class `Theil_Sen`.

Methods (by generic)

- `slope`: Extract slope.
- `intercept`: Extract intercept.

Examples

```r
# create theil_sen object
ts <- theil_sen(1:5, c(1, 2, 3, 5, 9))

# get slope
slope(ts)

# get intercept
intercept(ts)
```

---

**top**  
*Select Most Abundant Litter Types*

Description

Select the fraction most abundant litter types.

Usage

```r
top(type, count, fraction = 0.8)
```

Arguments

- `type`: character vector of litter types
- `count`: counts for each litter type
- `fraction`: fraction of most abundant types (numeric value in [0, 1])

Value

Fraction of most abundant litter types in descending order (numeric vector of length 1).
**toString.sequenized**

*Convert Sequenized Output to Character String*

---

**Description**

Convert Sequenized Output to Character String

**Usage**

```r
## S3 method for class 'sequenized'
toString(x, ...)
```

**Arguments**

- `x`  
  object of class `sequenized`.
- `...`  
  further arguments passed to or from other methods.

**Value**

string representation (character vector of length 1) of a sequenized object

**See Also**

`sequenize.integer`

---

**trimean**  
*Tukey's Trimean*

---

**Description**

Robust centrality measure estimated as the weighted average of the three quartiles: \((Q_1 + 2Q_2 + Q_3)/4\), where \(Q_1, Q_2\) and \(Q_3\) are the first, second and third quartiles respectively.

**Usage**

```r
trimean(x, ...)
```

**Arguments**

- `x`  
  numeric vector
- `...`  
  further arguments passed to or from other methods.
validate

Value

timean (numeric value of length 1).

Methods (by class)

• default: Tukey’s timean

References

https://en.wikipedia.org/wiki/Timean

Examples

stopifnot(timean(0:100) == mean(0:100))
stopifnot(timean(0:100) == median(0:100))

Description

Generic function for validation of file formats.

Usage

validate(x, ...)

## S3 method for class 'ospar'
validate(x, ...)

## S3 method for class 'long'
validate(x, ...)

## S3 method for class 'wide'
validate(x, ...)

## S3 method for class 'litter_group'
validate(x, ...)

Arguments

x object to validate
...

further arguments passed to or from other methods.
Value

validated object of class ospar
validated object of class long
validated object of class wide
validated object of class litter_group

Methods (by class)

• ospar: validate OSPAR data.
• long: validate long format data.
• wide: validate wide format data.
• litter_group: validate litter_group file

wilcoxon

Wilcoxon Test

Description

Constructor for a Wilcoxon test (simple wrapper for wilcox.test).

Usage

wilcoxon(x, type = c("both", "greater", "less"), mu = 0)

## S3 method for class 'wilcoxon'
test_statistic(x, ...)

## S3 method for class 'wilcoxon'
p_value(x, ...)

Arguments

x numeric vector representing a time-series.
type direction to test (both, increasing, or decreasing).
mu baseline value (null hypothesis)
... further arguments passed to or from other methods.

Value

object of class wilcoxon.

Methods (by generic)

• test_statistic: Extract test statistic V
• p_value: Extract p-value
See Also

wilcox.test, p_value, test_statistic

Examples

# create wilcoxon object
w <- wilcoxon(c(9, 4, 7, 5, 3), type = "less")

# get test statistic V
test_statistic(w)

# get p-value
p_value(w)