

# Package ‘neatR’

March 23, 2023

**Type** Package

**Title** Neat Data for Presentation

**Version** 0.2.0

**Description** Utilities for unambiguous, neat and legible representation of data (date, time stamp, numbers, percentages and strings) for presentation of analysis , aiming for elegance and consistency. The purpose of this package is to format data, that is better for presentation and any automation jobs that reports numbers.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** data.table, magrittr, tools

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

**RoxygenNote** 7.2.3

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Shivaprakash Suresh [aut, cre, cph]

**Maintainer** Shivaprakash Suresh <dswithai@gmail.com>

**Repository** CRAN

**Date/Publication** 2023-03-23 04:30:02 UTC

## R topics documented:

ndate . . . . .	2
nday . . . . .	2
nnumber . . . . .	3
npercent . . . . .	4
nstring . . . . .	5
ntimestamp . . . . .	6
<b>Index</b>	<b>8</b>

---

ndate *neat representation of dates*

---

### Description

neat representation of dates

### Usage

```
ndate(date, display.weekday = TRUE, is.month = FALSE)
```

### Arguments

date a Date or POSIX time stamp

display.weekday a Boolean. Whether the weekday of the date to be included.

is.month a Boolean variable representing if the date represents month. If this set to TRUE, the function returns 'MMMM'YY' as the output which is a neater representation of month.

### Value

String representation of the date

### Examples

```
# Neat representation of current date
x <- Sys.Date()
ndate(x)
# Neat representation of current date with day of week.
ndate(x, display.weekday = FALSE)
# Neat representation of current date with only month and year
ndate(x, display.weekday = FALSE, is.month = TRUE)
```

---

nday *neat alias of the week day with reference based on current date*

---

### Description

neat alias of the week day with reference based on current date

### Usage

```
nday(date, reference.alias = FALSE)
```

**Arguments**

date                    a Date or POSIX time stamp  
 reference.alias                    a Boolean. If set to TRUE, a reference alias of week day is shown based on current date such as Today/Yesterday/Tomorrow/Last/Coming.

**Value**

week day of the date in a readable format with reference alias based on current date

**Examples**

```
# Get day of the week of current date without reference alias
x <- Sys.Date()
nday(x, reference.alias = FALSE)
# Get day of the week with reference alias
nday(x, reference.alias = TRUE)
```

---

nnumber                    *neat representation of numbers*

---

**Description**

neat representation of numbers

**Usage**

```
nnumber(
  number,
  digits = 1,
  unit = "custom",
  unit.labels = list(thousand = "K", million = "Mn", billion = "Bn", trillion = "Tn"),
  prefix = "",
  suffix = "",
  thousand.separator = ", "
)
```

**Arguments**

number                    an integer or double.  
 digits                    number of digits to round-off. Default value is 1.  
 unit                      unit to which the number to be converted. See examples below.  
 unit.labels                a vector of strings (optional) that gives the unit label for thousand, million, billion and trillion.  
 prefix                    a string (optional) that can be prepended to the formatted number.  
 suffix                    a string (optional) that can be appended at the end of the formatted number.

thousand.separator

a character (optional) that can be used to chunk thousands to display large numbers. Default is set as comma, dot, comma or underscore can be used.

### Value

String representation of numbers with suffix denoting K for thousands, Mn for millions, Bn for billions, Tn for trillions. A number lower than thousand is represented as it is.

### Examples

```
x <- c(10, 100, 1000, 10000, 100000, 1000000, 10000000, 100000000, 1000000000)
nnumber(x)
nnumber(123456789.123456, digits = 1)
nnumber(123456789.123456, digits = 1, unit = 'Mn', prefix = '$')
```

---

npercent

*neat representation of percentage*

---

### Description

neat representation of percentage

### Usage

```
npercent(
  percent,
  is.decimal = TRUE,
  digits = 1,
  plus.sign = TRUE,
  factor.out = FALSE
)
```

### Arguments

percent	an integer or double representing percentage
is.decimal	a Boolean variable. If the percent is raw, the value to set as TRUE. See examples below. If the percent variable is already pre-multiplied by 100 then the value to be set as FALSE.
digits	number of digits to round-off
plus.sign	a Boolean variable. If the percent is positive then setting plus_sign = TRUE, includes an explicit + sign before the percent
factor.out	an optional Boolean variable. If the percent exceeds 1100 readable factors. See examples below.

### Value

String representation of the percentages.

## Examples

```
# Formatting 22.3%
npercent(0.223, is.decimal = TRUE, digits = 1)
npercent(22.3, is.decimal = FALSE, digits = 1)
# Formatting percentages with growth factors
npercent(c(-4.01, 2.56), is.decimal = TRUE, factor.out = TRUE)
```

---

nstring	<i>neat representation of string</i>
---------	--------------------------------------

---

## Description

neat representation of string

## Usage

```
nstring(  
  string,  
  case = NULL,  
  remove.specials = FALSE,  
  whitelist.specials = "",  
  en.only = FALSE  
)
```

## Arguments

string	a string / character
case	an optional parameter to convert the string variable to specific case. By default the case of the string is kept as it is. The available case conversions are lower, upper, title, start and initcap case.
remove.specials	an optional boolean. To remove special characters including any punctuation to be removed from the string, set this to TRUE.
whitelist.specials	an optional vector of strings. If any special characters to be retained while remove.specials is set to TRUE. See examples below.
en.only	an optional parameter taking boolean values, if set to TRUE, only english alphabets (and numbers) are kept in the string. Non english characters are removed.

## Value

White space cleaned and optionally formatted by case conversion and removal of special characters of the input string.

## See Also

Refer to [https://en.wikipedia.org/wiki/Letter\\_case#Stylistic\\_or\\_specialised\\_usage](https://en.wikipedia.org/wiki/Letter_case#Stylistic_or_specialised_usage) for more information about the different cases of text/string.

**Examples**

```
nstring(' All MOdels are wrong. some ARE useful!!! ', case = 'title', remove.specials = TRUE)
nstring("all Models are Wrong some are Useful", case = 'start', remove.specials = TRUE)
nstring('variable_123!!', remove.specials = TRUE, whitelist.specials = c('_'))
```

---

<code>ntimestamp</code>	<i>neat representation of time stamp</i>
-------------------------	--

---

**Description**

neat representation of time stamp

**Usage**

```
ntimestamp(
  timestamp,
  display.weekday = TRUE,
  include.date = TRUE,
  include.hours = TRUE,
  include.minutes = TRUE,
  include.seconds = TRUE,
  include.timezone = TRUE
)
```

**Arguments**

<code>timestamp</code>	a POSIX time stamp
<code>display.weekday</code>	a Boolean representing if the weekday of the timestamp to be included. By default it is set to TRUE
<code>include.date</code>	a Boolean representing if the date of time stamp to be included. By default it is set to TRUE.
<code>include.hours</code>	a Boolean representing if the hours to be included. By default it is set to TRUE
<code>include.minutes</code>	a Boolean representing if the minutes to be included. By default it is set to TRUE
<code>include.seconds</code>	a Boolean representing if the seconds to be included. By default it is set to TRUE
<code>include.timezone</code>	a Boolean variable representing if the timezone of the date variable to be included. By default it is set to TRUE.

**Value**

String representation of time stamp

**Examples**

```
# Neat representation of time stamp
x <- Sys.time()
ntimestamp(x)
# Neat representation of time from a time stamp
ntimestamp(x, include.date = FALSE, include.seconds = FALSE,
include.timezone = FALSE)
```

# Index

ndate, 2  
nday, 2  
nnumber, 3  
npercent, 4  
nstring, 5  
ntimestamp, 6