

# Package: tidyna (via r-universe)

May 29, 2026

**Title** NA-Aware Defaults for Common R Functions

**Version** 0.5.0

**Description** Provides drop-in replacements for common R functions (mean(), sum(), sd(), min(), etc.) that default to 'na.rm = TRUE' and issue warnings when missing values are removed. It handles some special cases. The table() default is set to 'useNA = ifany'.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Config/roxygen2/markdown** TRUE

**Imports** cli (>= 3.0.0)

**Suggests** testthat (>= 3.0.0), withr

**Config/testthat/edition** 3

**URL** <https://statzhero.github.io/tidyna/>,  
<https://github.com/statzhero/tidyna>

**BugReports** <https://github.com/statzhero/tidyna/issues>

**Config/roxygen2/version** 8.0.0

**Config/Needs/website** pkgdown

**Repository** <https://statzhero.r-universe.dev>

**Date/Publication** 2026-05-29 16:14:03 UTC

**RemoteUrl** <https://github.com/statzhero/tidyna>

**RemoteRef** HEAD

**RemoteSha** 7d433622da513e8fe327fa96b4f09eb54d0c84

## Contents

correlation-functions . . . . .	2
extrema-functions . . . . .	3
logical-functions . . . . .	4

row-functions . . . . .	5
summary-functions . . . . .	6
table-functions . . . . .	7
weighted.mean . . . . .	7
<b>Index</b>	<b>9</b>

---

correlation-functions *NA-aware Correlation Function*

---

## Description

Drop-in replacement for `cor()` that defaults to use = "pairwise.complete.obs".

## Usage

```
cor(
  x,
  y = NULL,
  use = "pairwise.complete.obs",
  method = c("pearson", "kendall", "spearman"),
  ...
)
```

## Arguments

<code>x</code>	A numeric vector, matrix, or data frame.
<code>y</code>	Optional. A numeric vector, matrix, or data frame.
<code>use</code>	Method for handling missing values. Default "pairwise.complete.obs".
<code>method</code>	Correlation method: "pearson", "kendall", or "spearman".
<code>...</code>	Additional arguments passed to <code>stats::cor()</code> .

## Value

A correlation matrix or single correlation coefficient.

## Examples

```
x <- c(1, 2, NA, 4)
y <- c(2, 4, 6, 8)
cor(x, y)
```

**Description**

Drop-in replacements for `min()`, `max()`, `range()`, `pmax()`, and `pmin()` that default to `na.rm = TRUE`.

**Usage**

```
min(..., na.rm = TRUE, all_na = NULL)
max(..., na.rm = TRUE, all_na = NULL)
range(..., na.rm = TRUE, all_na = NULL, finite = FALSE)
pmax(..., na.rm = TRUE, all_na = NULL)
pmin(..., na.rm = TRUE, all_na = NULL)
```

**Arguments**

<code>...</code>	Numeric or character arguments.
<code>na.rm</code>	Logical. Should missing values be removed? Default <code>TRUE</code> .
<code>all_na</code>	Character. What to do when all values are NA: "error" (default) throws an error, "base" returns what base R does with <code>na.rm = TRUE</code> (e.g., <code>Inf</code> for <code>min()</code> , <code>-Inf</code> for <code>max()</code> ), "na" returns NA. If <code>NULL</code> , uses <code>getOption("tidyna.all_na", "error")</code> .
<code>finite</code>	Logical. If <code>TRUE</code> , removes all non-finite values (NA, NaN, <code>Inf</code> , <code>-Inf</code> ). Only applies to <code>range()</code> . Default <code>FALSE</code> .

**Value**

For `min()` and `max()`, a length-one vector. For `range()`, a length-two vector containing the minimum and maximum. For `pmax()` and `pmin()`, a vector of length equal to the longest input.

**Examples**

```
x <- c(1, NA, 5, 3)
min(x)
max(x)
range(x)

# Multiple arguments
min(c(5, NA), c(1, 2))

# Parallel max/min
```

```
pmax(c(1, 5, 3), c(2, 1, 4))
pmin(c(1, NA, 3), c(NA, NA, 1))

# range with infinite values
y <- c(1, Inf, 3, -Inf)
range(y)
range(y, finite = TRUE)
```

---

logical-functions      *NA-aware Logical Functions*

---

## Description

Drop-in replacements for `any()` and `all()` that default to `na.rm = TRUE`.

## Usage

```
any(x, ..., na.rm = TRUE, all_na = NULL)
```

```
all(x, ..., na.rm = TRUE, all_na = NULL)
```

## Arguments

<code>x</code>	A logical vector.
<code>...</code>	Additional arguments passed to the base function.
<code>na.rm</code>	Logical. Should missing values be removed? Default <code>TRUE</code> .
<code>all_na</code>	Character. What to do when all values are NA: "error" (default) throws an error, "base" returns what base R does with <code>na.rm = TRUE</code> ( <code>FALSE</code> for <code>any()</code> , <code>TRUE</code> for <code>all()</code> ), "na" returns NA. If <code>NULL</code> , uses <code>getOption("tidyna.all_na", "error")</code> .

## Value

A single logical value.

## Examples

```
x <- c(TRUE, NA, FALSE)
any(x)
all(x)
```

---

row-functions	<i>NA-aware Row-wise Functions</i>
---------------	------------------------------------

---

**Description**

Drop-in replacements for `rowMeans()` and `rowSums()` that default to `na.rm = TRUE`. Both return NA for rows where ALL values are missing (base `rowMeans()` returns NaN, base `rowSums()` returns 0).

**Usage**

```
rowMeans(x, na.rm = TRUE, all_na = NULL, dims = 1L, ...)
```

```
rowSums(x, na.rm = TRUE, all_na = NULL, dims = 1L, ...)
```

**Arguments**

<code>x</code>	A numeric matrix or data frame.
<code>na.rm</code>	Logical. Should missing values be removed? Default TRUE.
<code>all_na</code>	Character. What to do when all values are NA: "error" (default) throws an error, "base" returns what base R does with <code>na.rm = TRUE</code> (NaN for <code>rowMeans()</code> , 0 for <code>rowSums()</code> ), "na" returns NA. If NULL, uses <code>getOption("tidyna.all_na", "error")</code> .
<code>dims</code>	Integer. Number of dimensions to treat as rows.
<code>...</code>	Additional arguments passed to the base function.

**Value**

A numeric or complex array of suitable size, or a vector if the result is one-dimensional.

**Examples**

```
mat <- matrix(c(1, NA, 3, NA, NA, NA), nrow = 2, byrow = TRUE)
rowSums(mat)

# Compare to base R:
base::rowSums(mat, na.rm = TRUE)
```

---

summary-functions      *NA-aware Summary Functions*

---

### Description

Drop-in replacements for summary functions that default to `na.rm = TRUE` and warn when missing values are removed.

### Usage

```
mean(x, ..., na.rm = TRUE, all_na = NULL)
sum(x, ..., na.rm = TRUE, all_na = NULL)
prod(x, ..., na.rm = TRUE, all_na = NULL)
sd(x, ..., na.rm = TRUE, all_na = NULL)
var(x, ..., na.rm = TRUE, all_na = NULL)
median(x, ..., na.rm = TRUE, all_na = NULL)
quantile(x, ..., na.rm = TRUE, all_na = NULL)
```

### Arguments

<code>x</code>	A numeric vector.
<code>...</code>	Additional arguments passed to the base function.
<code>na.rm</code>	Logical. Should missing values be removed? Default <code>TRUE</code> .
<code>all_na</code>	Character. What to do when all values are NA: "error" (default) throws an error, "base" returns what base R does with <code>na.rm = TRUE</code> (e.g., <code>NaN</code> for <code>mean()</code> , <code>0</code> for <code>sum()</code> ), "na" returns NA. If <code>NULL</code> , uses <code>getOption("tidyna.all_na", "error")</code> .

### Value

The computed summary statistic.

### Examples

```
x <- c(1, 2, NA, 4)
mean(x)

# Suppress warnings
options(tidyna.warn = FALSE)
mean(x)
options(tidyna.warn = TRUE)
```

```
# Control all-NA behavior
mean(c(NA, NA), all_na = "na")
```

---

table-functions	<i>NA-aware Table Function</i>
-----------------	--------------------------------

---

### Description

Drop-in replacement for `table()` that defaults to `useNA = "ifany"`, showing NA counts when present.

### Usage

```
table(..., useNA = "ifany")
```

### Arguments

<code>...</code>	Objects to cross-tabulate.
<code>useNA</code>	Whether to include NA values. Default "ifany".

### Value

A contingency table of class `table`.

### Examples

```
x <- c("a", "b", NA, "a", NA)
table(x)
```

---

weighted.mean	<i>NA-aware weighted mean</i>
---------------	-------------------------------

---

### Description

Drop-in replacement for `stats::weighted.mean()` that defaults to `na.rm = TRUE` and warns when missing values are removed. Unlike base R, missing values in either `x` or `w` cause the corresponding pair to be removed.

### Usage

```
weighted.mean(x, w, ..., na.rm = TRUE, all_na = NULL)
```

**Arguments**

<code>x</code>	A numeric vector of values.
<code>w</code>	A numeric vector of weights the same length as <code>x</code> .
<code>...</code>	Additional arguments passed to <code>stats::weighted.mean()</code> .
<code>na.rm</code>	Logical. Should missing values be removed? Default TRUE.
<code>all_na</code>	Character. What to do when all values are NA: "error" (default) throws an error, "base" returns what base R does with <code>na.rm = TRUE</code> , "na" returns NA. If NULL, uses <code>getOption("tidyna.all_na", "error")</code> .

**Value**

A length-one numeric vector.

**Examples**

```
x <- c(1, 2, NA, 4)
w <- c(1, 1, 1, 1)
weighted.mean(x, w)
```

# Index

all (logical-functions), 4  
any (logical-functions), 4  
  
cor (correlation-functions), 2  
correlation-functions, 2  
  
extrema-functions, 3  
  
logical-functions, 4  
  
max (extrema-functions), 3  
mean (summary-functions), 6  
median (summary-functions), 6  
min (extrema-functions), 3  
  
pmax (extrema-functions), 3  
pmin (extrema-functions), 3  
prod (summary-functions), 6  
  
quantile (summary-functions), 6  
  
range (extrema-functions), 3  
row-functions, 5  
rowMeans (row-functions), 5  
rowSums (row-functions), 5  
  
sd (summary-functions), 6  
stats::weighted.mean(), 7, 8  
sum (summary-functions), 6  
summary-functions, 6  
  
table (table-functions), 7  
table-functions, 7  
  
var (summary-functions), 6  
  
weighted.mean, 7