

# Package ‘tbrf’

November 15, 2019

**Type** Package

**Title** Time-Based Rolling Functions

**Version** 0.1.3

**Maintainer** Michael Schramm <michael.schramm@ag.tamu.edu>

**Description** Provides rolling statistical functions based  
on date and time windows instead of n-lagged observations.

**URL** <https://mps9506.github.io/tbrf/>

**BugReports** <https://github.com/mps9506/tbrf/issues>

**License** GPL-3 | file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**Depends** R (>= 2.10)

**Imports** boot, dplyr, lubridate, purrr, rlang, tibble, tidyr

**Suggests** covr, ggalt, ggplot2, testthat, knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Michael Schramm [aut, cre] (<<https://orcid.org/0000-0003-1876-6592>>),  
Frank Harrell [ctb]

**Repository** CRAN

**Date/Publication** 2019-11-15 21:30:02 UTC

## R topics documented:

Dissolved_Oxygen . . . . .	2
tbr_binom . . . . .	3
tbr_gmean . . . . .	4
tbr_mean . . . . .	5
tbr_median . . . . .	6

tbr_misc . . . . .	7
tbr_sd . . . . .	7
tbr_sum . . . . .	8

<b>Index</b>	<b>10</b>
--------------	-----------

---

Dissolved_Oxygen	<i>Dissolved oxygen measurements from the Tres Palacios rivers</i>
------------------	--

---

### Description

Data from the Texas Commission on Environmental Quality Surface Water Quality Monitoring Information System. The AverageDO field is the mean of dissolved oxygen concentrations (mg/L) measured at a field site at that day. The MinDO is the minimum dissolved oxygen concentration measured at that site on that day.

### Usage

```
data(Dissolved_Oxygen)
```

### Format

A data frame with 236 rows and 6 variables:

**Station\_ID** unique water quality monitoring station identifier

**Date** sampling date in yyyy-mm-dd format

**Param\_Code** unique parameter code

**Param\_Desc** parameter description with units

**Average\_DO** mean of dissolved oxygen measurement, in mg/L

**Min\_DO** minimum of dissolved oxygen measurement, in mg/L

### Source

<https://www80.tceq.texas.gov/SwqmisPublic/public/default.htm>

---

`tbr_binom`*Time-Based Rolling Binomial Probability*

---

**Description**

Produces a a rolling time-window based vector of binomial probability and confidence intervals.

**Usage**

```
tbr_binom(.tbl, x, tcolumn, unit = "years", n, alpha = 0.05)
```

**Arguments**

<code>.tbl</code>	dataframe with two variables.
<code>x</code>	indicates the variable column containing "success" and "failure" observations coded as 1 or 0.
<code>tcolumn</code>	indicates the variable column containing Date or Date-Time values.
<code>unit</code>	character, one of "years", "months", "weeks", "days", "hours", "minutes", "seconds"
<code>n</code>	numeric, describing the length of the time window in the selected units.
<code>alpha</code>	numeric, probability of a type 1 error, so confidence coefficient = 1-alpha

**Value**

tibble with binomial point estimate and confidence intervals.

**See Also**

[binom\\_ci](#)

**Examples**

```
## Generate Sample Data
df <- tibble::data_frame(
  date = sample(seq(as.Date('2000-01-01'), as.Date('2015/12/30')), by = "day"), 100),
  value = rbinom(100, 1, 0.25)
)

## Run Function
tbr_binom(df, x = value,
  tcolumn = date, unit = "years", n = 5,
  alpha = 0.1)
```

---

`tbr_gmean`*Time-Based Rolling Geometric Mean*

---

**Description**

Produces a a rolling time-window based vector of geometric means and confidence intervals.

**Usage**

```
tbr_gmean(.tbl, x, tcolumn, unit = "years", n, ...)
```

**Arguments**

<code>.tbl</code>	a data frame with at least two variables; time column formatted as date, date/time and value column.
<code>x</code>	column containing the values to calculate the geometric mean.
<code>tcolumn</code>	formatted time column.
<code>unit</code>	character, one of "years", "months", "weeks", "days", "hours", "minutes", "seconds"
<code>n</code>	numeric, describing the length of the time window.
<code>...</code>	additional arguments passed to <a href="#">gm_mean_ci</a>

**Value**

tibble with columns for the rolling geometric mean and upper and lower confidence levels.

**See Also**

[gm\\_mean\\_ci](#)

**Examples**

```
## Return a tibble with new rolling geometric mean column
tbr_gmean(Dissolved_Oxygen, x = Average_DO, tcolumn = Date, unit = "years", n = 5)

## Not run:
## Return a tibble with rolling geometric mean and 95% CI
tbr_gmean(Dissolved_Oxygen, x = Average_DO, tcolumn = Date, unit = "years", n = 5, conf = .95)
## End(Not run)
```

---

tbr_mean	<i>Time-Based Rolling Mean</i>
----------	--------------------------------

---

## Description

Produces a a rolling time-window based vector of means and confidence intervals.

## Usage

```
tbr_mean(.tbl, x, tcolumn, unit = "years", n, ...)
```

## Arguments

.tbl	a data frame with at least two variables; time column formatted as date, date/time and value column.
x	column containing the numeric values to calculate the mean.
tcolumn	formatted time column.
unit	character, one of "years", "months", "weeks", "days", "hours", "minutes", "seconds"
n	numeric, describing the length of the time window.
...	additional arguments passed to <a href="#">mean_ci</a> .

## Value

tibble with columns for the rolling mean and upper and lower confidence intervals.

## See Also

[mean\\_ci](#)

## Examples

```
## Return a tibble with new rolling mean column
tbr_mean(Dissolved_Oxygen, x = Average_DO, tcolumn = Date, unit = "years", n = 5)

## Not run:
## Return a tibble with rolling mean and 95% CI
tbr_mean(Dissolved_Oxygen, x = Average_DO, tcolumn = Date, unit = "years", n = 5, conf = .95)
## End(Not run)
```

---

`tbr_median`*Time-Based Rolling Median*

---

**Description**

Produces a a rolling time-window based vector of medians and confidence intervals.

**Usage**

```
tbr_median(.tbl, x, tcolumn, unit = "years", n, ...)
```

**Arguments**

<code>.tbl</code>	a data frame with at least two variables; time column formatted as date, date/time and value column.
<code>x</code>	column containing the numeric values to calculate the mean.
<code>tcolumn</code>	formatted time column.
<code>unit</code>	character, one of "years", "months", "weeks", "days", "hours", "minutes", "seconds"
<code>n</code>	numeric, describing the length of the time window.
<code>...</code>	additional arguments passed to <a href="#">median_ci</a>

**Value**

tibble with columns for the rolling median and upper and lower confidence intervals.

**See Also**

[median\\_ci](#)

**Examples**

```
## Return a tibble with new rolling median column
tbr_median(Dissolved_Oxygen, x = Average_DO, tcolumn = Date, unit = "years",
n = 5)

## Not run:
## Return a tibble with rolling median and 95% CI
tbr_median(Dissolved_Oxygen, x = Average_DO, tcolumn = Date, unit = "years", n = 5, conf = .95)
## End(Not run)
```

---

`tbr_misc`*Use Generic Functions with Time Windows*

---

**Description**

Use Generic Functions with Time Windows

**Usage**`tbr_misc(.tbl, x, tcolumn, unit = "years", n, func, ...)`**Arguments**

<code>.tbl</code>	a data frame with at least two variables; time column formatted as date, date/time and value column.
<code>x</code>	column containing the values the function is applied to.
<code>tcolumn</code>	formatted time column.
<code>unit</code>	character, one of "years", "months", "weeks", "days", "hours", "minutes", "seconds"
<code>n</code>	numeric, describing the length of the time window.
<code>func</code>	specified function
<code>...</code>	optional additional arguments passed to function <code>func</code>

**Value**

tibble

**Examples**`tbr_misc(Dissolved_Oxygen, x = Average_DO, tcolumn = Date, unit = "years", n = 5, func = mean)`

---

`tbr_sd`*Time-Based Rolling Standard Deviation*

---

**Description**

Time-Based Rolling Standard Deviation

**Usage**`tbr_sd(.tbl, x, tcolumn, unit = "years", n, na.rm = FALSE)`

**Arguments**

.tbl	a data frame with at least two variables; time column formatted as date, date/time and value column.
x	column containing the values to calculate the standard deviation.
tcolumn	formatted time column.
unit	character, one of "years", "months", "weeks", "days", "hours", "minutes", "seconds"
n	numeric, describing the length of the time window.
na.rm	logical. Should missing values be removed?

**Value**

tibble with column for the rolling sd.

**See Also**

[sd](#)

**Examples**

```
tbr_sd(Dissolved_Oxygen, x = Average_D0, tcolumn = Date, unit = "years", n = 5)
```

---

tbr_sum	<i>Time-Based Rolling Sum</i>
---------	-------------------------------

---

**Description**

Time-Based Rolling Sum

**Usage**

```
tbr_sum(.tbl, x, tcolumn, unit = "years", n, na.rm = FALSE)
```

**Arguments**

.tbl	a data frame with at least two variables; time column formatted as date, date/time and value column.
x	column containing the values to calculate the sum.
tcolumn	formatted time column.
unit	character, one of "years", "months", "weeks", "days", "hours", "minutes", "seconds"
n	numeric, describing the length of the time window.
na.rm	logical. Should missing values be removed?



**Value**

dataframe with column for the rolling sum.

**See Also**

[sum](#)

**Examples**

```
tbr_sum(Dissolved_Oxygen, x = Average_DO, tcolumn = Date, unit = "years", n = 5)
```

# Index

## \*Topic **datasets**

Dissolved\_Oxygen, [2](#)

binom\_ci, [3](#)

Dissolved\_Oxygen, [2](#)

gm\_mean\_ci, [4](#)

mean\_ci, [5](#)

median\_ci, [6](#)

sd, [8](#)

sum, [9](#)

tbr\_binom, [3](#)

tbr\_gmean, [4](#)

tbr\_mean, [5](#)

tbr\_median, [6](#)

tbr\_misc, [7](#)

tbr\_sd, [7](#)

tbr\_sum, [8](#)