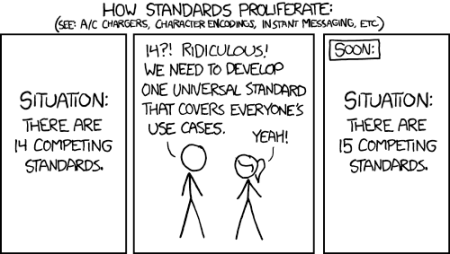
The R ecosystem knows a vast number of time series classes: ts, xts, zoo, tsibble, tibbletime or timeSeries. The plethora of standards causes confusion. As different packages rely on different classes, it is hard to use them in the same analysis. tsbox provides a set of tools that make it easy to switch between these classes. It also allows the user to treat time series as plain data frames, facilitating the use with tools that assume rectangular data.



The tsbox package is built around a set of functions that convert time series of different classes to each other. They are frequency-agnostic, and allow the user to combine multiple non-standard and irregular frequencies. Because coercion works reliably, it is easy to write functions that work identically for all classes. So whether we want to smooth, scale, differentiate, chain-link, forecast, regularize or seasonally adjust a time series, we can use the same tsbox-command for any time series class.

This blog gives a short overview of the changes introduced in 0.1.

**Keeping explicit missing values**

Version 0.1, now on CRAN, brings a large number of bug fixes and improvements. A substantial change involves the treatment of NA values in data frames. Previously, all NAs in data frames were treated as implicit, and were only made explicit by a call to ts\_regular.

This has changed now. If you convert a ts object to a data frame, all NA values will be preserved. To replicate previous behaviour, apply the ts\_na\_omit function:

library(tsbox)

x.ts <- ts\_c(mdeaths, austres)

x.ts

ts\_df(x.ts)

ts\_na\_omit(ts\_df(x.ts))

**ts\_span extends outside of series span**

This lays the groundwork for ts\_span to be extensible. With extend = TRUE, ts\_span extends a regular series with NA values, up to the specified limits, similar to base window. Like all functions in tsbox, this is frequency-agnostic. For example, in the following, the monthly series mdeaths is extended by monthly NA values, while the quarterly series austres is extended by quarterly NA values.

x.df <- ts\_df(ts\_c(mdeaths, austres))

ts\_span(x.df, end = "1999-12-01", extend = TRUE)

**ts\_default standardizes column names in a data frame**

In rectangular data structures, i.e., in a data.frame, a data.table, or a tibble, tsbox stores one or multiple time series in the ‘long’ format. By default, tsbox detects a *value*, a *time* and zero, one or several *id*columns. Alternatively, the time column and the value column can be explicitly named time and value. If explicit names are used, the column order will be ignored.

While automatic column name detection is useful in interactive mode, it produces unnecessary overhead in longer workflows. The helper function ts\_default detects and renames the time and the value column, so that auto-detection will be turned off in subsequent steps (note that the names of the id columns are not affected):

x.df <- ts\_df(ts\_c(mdeaths, austres))

names(x.df) <- c("a fancy id name", "date", "count")

ts\_plot(x.df) # tsbox is fine with that

ts\_default(x.df)

**ts\_summary summarizes time series**

ts\_summary provides a frequency agnostic summary of a ts-boxable object:

ts\_summary(ts\_c(mdeaths, austres))

#> id obs diff freq start end

#> 1 mdeaths 72 1 month 12 1974-01-01 1979-12-01

#> 2 austres 89 3 month 4 1971-04-01 1993-04-01

ts\_summary returns a plain data frame that can be used for any purpose. It is also recommended for the extraction of various time series properties, such as start, freq or id:

ts\_summary(austres)$id

#> [1] "austres"

ts\_summary(austres)$start

#> [1] "1971-04-01"