

# Package ‘tstools’

February 7, 2018

**Type** Package

**Version** 0.3.5.1

**Title** A Time Series Toolbox for Official Statistics

**Description** Plot official statistics' time series conveniently: automatic legends, highlight windows, stacked bar charts with positive and negative contributions, sum-as-line option, two y-axes with automatic horizontal grids that fit both axes and other popular chart types. 'tstools' comes with a plethora of defaults to let you plot without setting an abundance of parameters first, but gives you the flexibility to tweak the defaults. In addition to charts, 'tstools' provides a super fast, 'data.table' backed time series I/O that allows the user to export / import long format, wide format and transposed wide format data to various file types.

**Depends** R ( $\geq 3.0.0$ ), zoo ( $\geq 1.7-12$ ),

**Imports** xts, stats, graphics, jsonlite, data.table,

**Suggests** knitr, rmarkdown, testthat, reshape2, openxlsx

**VignetteBuilder** knitr

**Date** 2018-02-06

**URL** <https://github.com/mbannert/tstools>

**BugReports** <https://github.com/mbannert/tstools/issues>

**License** GPL-2

**LazyData** true

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Author** Matthias Bannert [aut, cre],  
Severin Thoeni [aut]

**Maintainer** Matthias Bannert <bannert@kof.ethz.ch>

**Repository** CRAN

**Date/Publication** 2018-02-07 13:08:21 UTC

**R topics documented:**

compute_decimal_time . . . . .	2
concat_ts . . . . .	3
create_dummy_ts . . . . .	3
df_to_reg_ts . . . . .	4
fill_year_with_nas . . . . .	5
init_tsplot_theme . . . . .	5
KOF . . . . .	9
m_to_q . . . . .	9
overlap_sorted_ts_lists . . . . .	10
overlap_ts_lists_by_name . . . . .	10
read_ts . . . . .	11
resolve_ts_overlap . . . . .	11
start_ts_after_internal_nas . . . . .	12
strip_ts_of_leading_nas . . . . .	13
tsplot . . . . .	13
tsqm . . . . .	14
tstools-deprecated . . . . .	15
write_ts . . . . .	15

<b>Index</b>	<b>17</b>
--------------	-----------

---

compute\_decimal\_time *Compute Decimal Time from a ts Period Vector*

---

**Description**

Standard ts object use a vector of length two to store a period. E.g. 2010,1 means first quarter of 2010, if the series was quarterly and first month if the series was monthly etc.

**Usage**

```
compute_decimal_time(v, f)
```

**Arguments**

v	integer vector denoting a period in time
f	frequency

---

concat_ts	<i>Concatenate to Non-Overlapping Time Series</i>
-----------	---

---

**Description**

Append one time series to another. This only works for non-overlapping time series of the same frequency. For overlapping time series please see [resolveOverlap](#).

**Usage**

```
concat_ts(ts1, ts2)
```

**Arguments**

ts1	object of class ts1, typically the older of two time series.
ts2	object of class ts1, typically the younger of two time series.

---

create_dummy_ts	<i>Flexible Function to Create Time Series Dummy Variables</i>
-----------------	--

---

**Description**

Generate time series with a default value that is changed within a certain subperiod. The function allows for additional convenience when specifying single period dummies and dummies that go from a certain point in time to the end of the series.

**Usage**

```
create_dummy_ts(end_basic, dummy_start, dummy_end = NULL, sp = T,
  start_basic = c(1980, 1), basic_value = 0, dummy_value = 1,
  frequency = 4)
```

**Arguments**

end_basic	numeric vector of form c(yyyy,p) defining the end of the time series.
dummy_start	numeric vector of form c(yyyy,p) defining the beginning of the period with different value.
dummy_end	numeric vector of form c(yyyy,p) defining the end of the period with different value. Defaults to NULL, using the end_date of the series.
sp	logical should NULL value for dummy_end lead to a single period dummy (TRUE) or to alternative values until the end.
start_basic	numeric vector of form c(yyyy,p) defining the start of the time series. Defaults to c(1980,1)
basic_value	default value of the time series, defaults to 0.
dummy_value	the alternative value, defaults to 1.
frequency	integer frequency of the regular time series, defaults to 4 (quarterly).

**Author(s)**

Matthias Bannert

df\_to\_reg\_ts

*Turn data.frame to Regular Monthly or Quarterly Time Series***Description**

Turn a data.frame with date columns to a regular time series object if possible. Design to work with quarterly and monthly data.

**Usage**

```
df_to_reg_ts(dframe, var_cols, year_col = "year", period_col = "month",
  freq = 12, return_ts = T, by = NULL)
```

**Arguments**

dframe	data.frame input
var_cols	columns that contain variables as opposed to date index.
year_col	integer, logical or character vector indicating the year position within the data.frame.
period_col	integer, logical or character vector indicating the period position within the data.frame.
freq	integer indicating the frequency of new time series.
return_ts	logical should a (list of) time series be returned? Defaults to TRUE. FALSE returns data.frame.
by	character overwrite automatically detected (from freq) by parameter. e.g. '1 day'. Defaults to NULL.

**Examples**

```
start_m <- as.Date("2017-01-01")
df_missing <- data.frame(
  date = seq(start_m, by='2 months', length=6),
  value = 1:6,
  another_value = letters[1:6],
  yet_another_col = letters[6:1]
)
df_to_reg_ts(df_missing, c("value", "another_value"))
df_to_reg_ts(df_missing, c("value", "another_value"), return_ts = FALSE)
```

---

fill\_year\_with\_nas      *Fill Up a Time Series with NAs*

---

### Description

When plotting a time series you might want set the range of the plot a little wider than just the start and end date of the original series. This function add fills up the current period (typically year) with NA.

### Usage

```
fill_year_with_nas(x, add_periods = 1, fill_up_start = FALSE)
```

### Arguments

x                      object of class ts  
 add\_periods          integer periods to add.  
 fill\_up\_start        logical should start year be filled up? Defaults to FALSE.

---

init\_tsplot\_theme      *Initiate Default Theme*

---

### Description

The `tsplot` methods provide a theme argument which is used to pass on a plethora of useful defaults. These defaults are essentially stored in a list. Sometimes the user may want to tweak some of these defaults while keeping most of them. Hence the `init_tsplot_theme` function create a fresh list object containing default values for lot of different layout parameters etc. By replacing single elements of the list and passing the entire list to the plot function, single aspects can be tweaked while keeping most defaults. `Init defaultTheme` does not need any parameters.

### Usage

```
init_tsplot_theme(margins = c(5, 4, 3, 3) + 0.1, fillYearWithNAs = TRUE,
  line_colors = c(ETH8_100 = "#007a92", ETH4_100 = "#72791c", ETH8_20 =
  "#cce5eb", ETH5_60 = "#cc67a7", ETH8_60 = "#66b0c2", ETH5_100 = "#91056a",
  ETH4_60 = "#a9af66"), line_to_middle = TRUE, lwd = c(2, 3, 1, 4, 2, 4),
  lty = 1, xaxs = "i", yaxs = "i", bar_border = "#000000",
  total_bar_margin_pct = 0.2, bar_fill_color = c(ETH8 = "#007A92", ETH8_60 =
  "#66b0c2", ETH8_30 = "#b3d7e0", ETH8_20 = "#cce5eb", ETH5 = "#91056a", ETH5_60 =
  "#cc67a7", ETH5_30 = "#e6b3d3"), sum_as_line = FALSE, sum_line_lty = 1,
  sum_line_lwd = 3, sum_line_color = c(ETH8_100 = "#007a92", ETH4_100 =
  "#72791c", ETH8_20 = "#cce5eb", ETH5_60 = "#cc67a7", ETH8_60 = "#66b0c2",
  ETH5_100 = "#91056a", ETH4_60 = "#a9af66"), highlight_window = FALSE,
  highlight_window_freq = 4, highlight_window_start = NA,
```

```

highlight_window_end = NA, highlight_color = "#e9e9e9", use_box = FALSE,
y_las = 2, lwd_ticks_1 = 1.5, lwd_ticks_2 = 1, yearly_ticks = TRUE,
quarterly_ticks = TRUE, monthly_ticks = FALSE,
tcl_quarterly_tick_tcl = -0.5, tcl_yearly_tick = -0.75,
lwd_yearly_ticks = 1.5, lwd_quarterly_ticks = 1, label_pos = "mid",
show_left_y_axis = TRUE, show_right_y_axis = TRUE, y_grid_count = c(5,
6, 8, 10), show_y_grids = TRUE, y_grid_color = "#CCCCCC",
y_grid_count_strict = FALSE, y_tick_margin = 0.15,
preferred_y_gap_sizes = c(25, 20, 15, 10, 5, 2.5, 1, 0.5),
y_range_min_size = NULL, legend_col = 3, title_outer = TRUE,
title_adj = 0, title_line = 0.8, title_cex.main = 1,
title_transform = NA, subtitle_adj = 0, subtitle_outer = TRUE,
subtitle_line = -0.6, subtitle_cex.main = 1,
subtitle_transform = "toupper", subtitle_adj_r = 0.9,
legend_intersp_x = 1, legend_intersp_y = 1,
range_must_not_cross_zero = FALSE)

```

### Arguments

<code>margins</code>	integer vector defaults to <code>c(5, 4, 3, 3) + 0.1</code> ,
<code>fillYearWithNAs</code>	logical should year be filled up with missing in order to plot the entire year on the axis. Defaults to TRUE,
<code>line_colors</code>	character vector of hex colors for 6 lines.
<code>line_to_middle</code>	logical try to put a line into the middle of the plot. defaults to TRUE.
<code>lwd</code>	integer vector line width, defaults to <code>c(2,3,1,4,2,4)</code> .
<code>lty</code>	integer vector line type defaults to 1.
<code>xaxs</code>	character axis definition as in base plot, defaults to "i".
<code>yaxs</code>	character axis definition as in base plot, defaults to "i".
<code>bar_border</code>	character hex colors for the border around bars in bar charts.
<code>total_bar_margin_pct</code>	numeric definition as in base plot, defaults to "i", defaults to .2,
<code>bar_fill_color</code>	character vector of hex colors for 6 time series.
<code>sum_as_line</code>	logical should the sum of stacked time series be displayed as a line on top of stacked bar charts. defaults to FALSE,
<code>sum_line_lty</code>	integer line type of <code>sum_as_line</code> , defaults to 1.
<code>sum_line_lwd</code>	integer line width of <code>sum_as_line</code> , defaults to 3.
<code>sum_line_color</code>	character hex color of of <code>sum_as_line</code> , defaults "#91056a".
<code>highlight_window</code>	logical should a particular time span be highlighted by different background color. Defaults to FALSE.
<code>highlight_window_freq</code>	integer frequency of the highlight window definition, defaults to 4.
<code>highlight_window_start</code>	integer vector highlight window start position, defaults to NA.

highlight_window_end	integer vector highlight window start position, defaults to NA.,
highlight_color	character hex color code of highlight background, defaults to "#e9e9e9".
use_box	logical use a box around the plot.
y_las	integer, same as base <code>plot</code> parameter defaults to 2.
lwd_ticks_1	numeric width of type 1 ticks, defaults to 1.5.
lwd_ticks_2	numeric width of type 1 ticks, defaults to 1.
yearly_ticks	logical, should yearly ticks be shown. Defaults to TRUE.
quarterly_ticks	logical, should quarterly ticks be shown. Defaults to TRUE.
monthly_ticks	logical, should monthly ticks be shown. Defaults to FALSE.
tcl_quarterly_tick_tcl	numeric, same as base <code>plot</code> tcl parameter defaults to -.5,
tcl_yearly_tick	numeric, same as base <code>plot</code> tcl parameter defaults to -.75,
lwd_yearly_ticks	numeric, width of yearly ticks, defaults to 1.5.
lwd_quarterly_ticks	numeric, width of yearly ticks, defaults to 1.
label_pos	character, currently undocumented. sorry. defaults to "mid".
show_left_y_axis	logical: should left y axis be shown, defaults to TRUE.
show_right_y_axis	logical: should left y axis be shown, defaults to TRUE.
y_grid_count	integer vector preferred y grid counts c(5,6,8,10).
show_y_grids	logical should y_grids by shown at all, defaults to TRUE.
y_grid_color	character hex color of grids. Defaults to gray "#CCCCCC".
y_grid_count_strict	logical should we strictly stick to preferred y grid count? Defaults to FALSE.
y_tick_margin	numeric, minimal percentage of horizontal grid that needs to be clean, i.e., without lines or bars. Defaults to 0.15 (15 percent).
preferred_y_gap_sizes	numeric c(25, 20, 15, 10, 5, 2.5, 1, 0.5),
y_range_min_size	= NULL ,
legend_col	integer number of columns for the legend, defaults to 3.
title_outer	logical, currently undocumented. Defaults to TRUE.
title_adj	numeric, same as base <code>plot</code> parameter, defaults to 0.
title_line	numeric same as base <code>plot</code> parameter, defaults to .8.
title_cex.main	numeric, same as base <code>plot</code> parameter defaults to 1

**title\_transform** function to transform the title, defaults to NA.  
**subtitle\_adj** numeric same as base `plot` parameter, defaults to 0.  
**subtitle\_outer** numeric same as base `plot` parameter, defaults to TRUE  
**subtitle\_line** numeric same as base `plot` parameter, defaults to -.6  
**subtitle\_cex.main** numeric same as base `plot` parameter, defaults to 1  
**subtitle\_transform** function to transform the subtitle, defaults to "toupper",  
**subtitle\_adj\_r** numeric same as base `plot` parameter, defaults to .9  
**legend\_intersp\_x** numeric same as base `plot` parameter, defaults to 1  
**legend\_intersp\_y** numeric same as base `plot` parameter, defaults to 1  
**range\_must\_not\_cross\_zero** logical automatic range finders are forced to do not find ranges below zero. Defaults to FALSE.

## Details

Themes are essentially list that contain `par` parameters. Below all items are listed, some of them with comments. I will try to write comments on all params soon. The list contains the following elements:

## Author(s)

Matthias Bannert

## Examples

```

# create a list
data(KOF)
tt <- init_tsplot_theme()
# adjust a single element
tt$highlight_window <- TRUE
# pass the list to tsplot
tsplot(KOF$kofbarometer, theme = tt)
# for more theme examples check the vignette
vignette("tstools")

```



---

KOF	<i>KOF Barometer - Swiss Business Cycle Indicator</i>
-----	---

---

**Description**

A list of time series containing two time series the KOF Barometer and the growth of Swiss GDP over time. KOF Barometer is a monthly business cycle indicator computed by the KOF Swiss Economic Institute. The GDP growth rate is used as a reference series to the Barometer.

**Usage**

KOF

**Format**

A list of two time series of class `ts`

**kofbarometer** KOF Barometer Indicator

**reference** Reference series to KOF Barometer, change in Swiss GDP compared to previous month

...

**Source**

<https://www.kof.ethz.ch/en/forecasts-and-indicators/indicators/kof-economic-barometer.html>

---

<code>m_to_q</code>	<i>Turn monthly series with regular NAs to quarter</i>
---------------------	--

---

**Description**

Monthly series with NAs in non-quarter months are turned to quarterly series. Series without NAs are just returned.

**Usage**

`m_to_q(series)`

**Arguments**

`series` an object of class `ts` with monthly frequency

---

`overlap_sorted_ts_lists`*Concat Time Series list wise*

---

**Description**

Concat overlapping time series list wise. List needs to be of same length. Takes names of list B.

**Usage**

```
overlap_sorted_ts_lists(listA, listB)
```

**Arguments**

<code>listA</code>	list of time series
<code>listB</code>	list of time series

---

`overlap_ts_lists_by_name`*Resolve Overlap Listwise, helpful with SA*

---

**Description**

Resolve Overlap Listwise, helpful with SA

**Usage**

```
overlap_ts_lists_by_name(listA, listB, chunkA = "_f4", chunkB = "_f12")
```

**Arguments**

<code>listA</code>	list of time series often of lower frequency
<code>listB</code>	list of time series often of higher frequency
<code>chunkA</code>	character chunk representing frequencies, defaults to <code>_f4</code> .
<code>chunkB</code>	character chunk representing frequencies, defaults to <code>_f12</code> .

---

read_ts	<i>Import time series data from a file.</i>
---------	---

---

### Description

If importing from a zip file, the archive should contain a single file with the extension .csv, .xlsx or .json.

### Usage

```
read_ts(file, format = c("csv", "xlsx", "json", "zip"), sep = ",")
```

### Arguments

file	Path to the file to be read
format	Which file format is the data stored in? If no format is supplied, read_ts will attempt to guess from the file extension.
sep	character separator for csv files. defaults to ','.

### Value

A named list of ts objects

---

resolve_ts_overlap	<i>Concatenate Time Series and Resolve Overlap Automatically</i>
--------------------	--

---

### Description

Append time series to each other. Resolve overlap determines which of two ts class time series is reaching further and arranges the two series into first and second series accordingly. Both time series are concatenated to one if both series had the same frequency. Typically this function is used concatenate two series that have a certain overlap, but one series clearly starts earlier while the other lasts longer. If one series starts earlier and stops later, all elements of the shorter series will be inserted into the larger series, i.e. elements of the smaller series will replace the elements of the longer series. Usually ts2 is kept.

### Usage

```
resolve_ts_overlap(ts1, ts2, keep_ts2 = T, tolerance = 0.001)
```

**Arguments**

ts1	ts time series, typically the older series
ts2	ts time series, typically the younger series
keep_ts2	logical should ts2 be kept? Defaults to TRUE.
tolerance	numeric when comparing min and max values with a index vector of a time series R runs in to trouble with precision handling, thus a tolerance needs to be set. Typically this does not need to be adjusted. E.g. 2010 != 2010.000. With the help of the tolerance parameter these two are equal.

**Examples**

```
ts1 <- ts(rnorm(100),start = c(1990,1),frequency = 4)
ts2 <- ts(1:18,start = c(2000,1),frequency = 4)
resolve_ts_overlap(ts1,ts2)

# automatical detection of correction sequence!
ts1 <- ts(rnorm(90),start = c(1990,1),frequency = 4)
ts2 <- ts(1:60,start = c(2000,1),frequency = 4)
resolve_ts_overlap(ts1,ts2)

# both series are of the same length use sequence of arguments.
ts1 <- ts(rnorm(100),start = c(1990,1),frequency = 4)
ts2 <- ts(1:48,start = c(2003,1),frequency = 4)
resolve_ts_overlap(ts1,ts2)
ts1 <- ts(rnorm(101),start = c(1990,1),frequency = 4)
ts2 <- ts(1:61,start = c(2000,1),frequency = 4)
resolve_ts_overlap(ts1,ts2)
#' clearly dominatn ts2 series
ts1 <- ts(rnorm(50),start = c(1990,1),frequency = 4)
ts2 <- ts(1:100,start = c(1990,1),frequency = 4)
resolve_ts_overlap(ts1,ts2)
```

---

```
start_ts_after_internal_nas
```

*Start a Time Series after the Last Internal NA*

---

**Description**

Internal NAs can cause trouble for time series operations such as X-13-ARIMA SEATS seasonal adjustment. Often, internal NAs only occur at at the beginning of a time series. Thus an easy solution to the problem is to discard the initial part of the data which contains the NA values. This way only a small part of the information is lost as opposed to not being able to seasonally adjust an entire series.

**Usage**

```
start_ts_after_internal_nas(series)
```

**Arguments**

series                    on object of class ts

**See Also**

[stripLeadingNAsFromTs](#), [stripTrailingNAsFromTs](#)

**Examples**

```
ts1 <- 1:30
ts1[c(3,6)] <- NA
ts1 <- ts(ts1,start=c(2000,1),frequency = 4)
start_ts_after_internal_nas(ts1)
```

---

strip\_ts\_of\_leading\_nas

*Strip Leading / Trailing NAs from a Time Series Object*

---

**Description**

Removes NAs to begin with and starts time series index at the first non-NA value.

**Usage**

```
strip_ts_of_leading_nas(s)
strip_ts_of_trailing_nas(s)
```

**Arguments**

s                        an object of class ts.

---

tsplot

*Plot Time Series*

---

**Description**

Conveniently plot time series.

**Usage**

```
tsplot(..., tsr = NULL, left_as_bar = FALSE, group_bar_chart = NULL,
plot_title = NULL, plot_subtitle = NULL, plot_subtitle_r = NULL,
find_ticks_function = "findTicks", fill_up_start = FALSE,
overall_xlim = NULL, overall_ylim = NULL, manual_date_ticks = NULL,
manual_value_ticks_l = NULL, manual_value_ticks_r = NULL, theme = NULL,
quiet = TRUE, auto_legend = TRUE)
```

**Arguments**

...	multiple objects of class ts or a list of time series. All objects passed through the ... parameter relate to the standard left y-axis.
tsr	list of time series objects of class ts.
left_as_bar	logical should the series that relate to the left bar be drawn as (stacked) bar charts?
group_bar_chart	logical should a bar chart be grouped instead of stacked?
plot_title	character title to be added to the plot
plot_subtitle	character subtitle to be added to the plot
plot_subtitle_r	character second subtitle to be added at the top right
find_ticks_function	function to compute ticks.
fill_up_start	logical should the start year be filled up?
overall_xlim	integer overall x-axis limits, defaults to NULL.
overall_ylim	integer overall y-axis limits, defaults to NULL.
manual_date_ticks	character vector of manual date ticks.
manual_value_ticks_l	numeric vector, forcing ticks to the left y-axis
manual_value_ticks_r	numeric vector, forcing ticks to the right y-axis
theme	list of default plot output parameters. Defaults to NULL, which leads to <a href="#">init_tsplot_theme</a> being called. Please see the vignette for details about tweaking themes.
quiet	logical suppress output, defaults to TRUE.
auto_legend	logical should legends be printed automatically, defaults to TRUE.

---

 tsqm

---

*Interpolate quarterly time series into monthly*


---

**Description**

Repeat quarterly variables two times to generate a monthly variable.

**Usage**

```
tsqm(qts)
```

**Arguments**

qts                    quarterly time series

**Examples**

```
tsq <- ts(1:20,start=c(1990,1),frequency = 4)
tsqm(tsq)
```

---

tstools-deprecated      *Deprecated function(s) in tstools*

---

**Description**

These functions are provided for compatibility with older version of the tstools package. They may eventually be completely removed.

**Arguments**

...                      Parameters to be passed to the modern version of the function

**Details**

computeDecimalTime	now a synonym for <a href="#">compute_decimal_time</a>
concatTs	now a synonym for <a href="#">concat_ts</a>
fillupYearWithNAs	now a synonym for <a href="#">fill_year_with_nas</a>
importTimeSeries	now a synonym for <a href="#">read_ts</a>
init_tsplot_theme	now a synonym for <a href="#">init_tsplot_theme</a>
overlapSortedLists	now a synonym for <a href="#">overlap_sorted_ts_lists</a>
overlapTslByName	now a synonym for <a href="#">overlap_ts_lists_by_name</a>
resolveOverlap	now a synonym for <a href="#">resolve_ts_overlap</a>
stripLeadingNAsFromTs	now a synonym for <a href="#">strip_ts_of_leading_nas</a>
stripTrailingNAsFromTs	now a synonym for <a href="#">strip_ts_of_trailing_nas</a>
writeTimeSeries	now a synonym for <a href="#">write_ts</a>

---

write\_ts                      *Export a list of time series to a file.*

---

**Description**

Export a list of time series to a file.

**Usage**

```
write_ts(tl, fname = NULL, format = "csv", date_format = NULL,
  timestamp_to_fn = FALSE, round_digits = NULL, rdata_varname = "tslist",
  ...)
```

**Arguments**

t1	list of time series
fname	character file name. Defaults to NULL, displaying output on console. Set a file name without file extension in order to store a file. Default file names / location are not CRAN compliant which is why the file name defaults to NULL.
format	character denotes export formats. Defaults to .csv. "csv", "xlsx", "json", "rdata" are available. Spreadsheet formats like csv allow for further optional parameters.
date_format	character denotes the date format. Defaults to NULL. If set to null the default is used: Jan 2010.
timestamp_to_fn	If TRUE, the current date will be appended to the file name. Defaults to FALSE.
round_digits	integer, precision in digits.
rdata_varname	character name of the list of time series within the store RData. Defaults to "tlist".
...	additional arguments used by specific formats.

**Details**

Additional arguments covered by ...

<b>Name</b>	<b>Effect</b>	<b>Format(s)</b>
wide	Export data in a wide format (one column per series)	CSV, XLSX
transpose	Transpose exported data (one row per series)	CSV, XLSX, only if wide = TRUE
zip	If set to TRUE, the file is compressed into a zip archive after export	any



# Index

## \*Topic **datasets**

KOF, 9

compute\_decimal\_time, 2, 15

computeDecimalTime  
(tstools-deprecated), 15

concat\_ts, 3, 15

concatTs (tstools-deprecated), 15

create\_dummy\_ts, 3

df\_to\_reg\_ts, 4

fill\_year\_with\_nas, 5, 15

fillupYearWithNAs (tstools-deprecated),  
15

importTimeSeries (tstools-deprecated),  
15

init\_tsplot\_theme, 5, 14, 15

initDefaultTheme (tstools-deprecated),  
15

KOF, 9

m\_to\_q, 9

overlap\_sorted\_ts\_lists, 10, 15

overlap\_ts\_lists\_by\_name, 10, 15

overlapSortedLists  
(tstools-deprecated), 15

overlapTslByName (tstools-deprecated),  
15

par, 8

plot, 7, 8

read\_ts, 11, 15

resolve\_ts\_overlap, 11, 15

resolveOverlap, 3

resolveOverlap (tstools-deprecated), 15

start\_ts\_after\_internal\_nas, 12

strip\_ts\_of\_leading\_nas, 13, 15

strip\_ts\_of\_trailing\_nas, 15

strip\_ts\_of\_trailing\_nas  
(strip\_ts\_of\_leading\_nas), 13

stripLeadingNAsFromTs, 13

stripLeadingNAsFromTs  
(tstools-deprecated), 15

stripTrailingNAsFromTs, 13

stripTrailingNAsFromTs  
(tstools-deprecated), 15

tsplot, 5, 13

tsqm, 14

tstools-deprecated, 15

tstools-deprecated-package  
(tstools-deprecated), 15

write\_ts, 15, 15

writeTimeSeries (tstools-deprecated), 15