Date Axis Formats

Dave Lorenz

October 24, 2016

Abstract

These examples demonstrate the various date and time axis formats for graphs created using the `timePlot` function. **NOTE:** to use any high-level graphics function in the smwrGraphs package, you must first call a function to set up the graphics environment like `setPage` or `setPDF`, but these functions are not shown in these examples because `setSweave` is used in vignettes.

Contents

1 Introduction

The `timePlot` function controls the formatting of the date- (x-) axis with the `xlabels` argument. The default value for `xlabels` is "Auto" but may be any of "hours," "days," "months," "years," or "water years" that will select a specific format. The value for the `xlabels` argument also may be a list of the arguments to the `datePretty` function for more control. The examples in this vignette use primarily the "Auto" option, but at least one example shows one of the other options.

All of these examples focus on the date-axis formatting when the x-axis data are class "Date" or "POSIXt." These examples plot no data and simplify the y-axis labeling. The x-axis range is set using the `xaxis.range` argument for these graphs, but the range also may be set by the range in the `x` argument to `timePlot` as done in the first example.

```
> # Load the smwrGraphs package
> library(smwrGraphs)
> # Generate the data to plot
> DD <- as.POSIXct(c("2009-07-02 10:30", "2009-07-02 12:30"))
> YY <- c(0.14, 0.82)
```

2 Hour Formats

Hour formats are suitable for date ranges less than about 3 days. The minimum time increment for axis formats is hours. Minor ticks are minutes by default when the major ticks and labels are hours.

This example demonstrates the hour format for a short timeframe, less than 1 day. This example also demonstrates how to add minor ticks to the graph. The `ticks` argument for the hours axis format can be computed as 60/min - 1, where min is the number of minutes desired between minor ticks.

```
> # Set up the graphics environment, the equivalent call for an on screen
> #  device could be setPage("square")
> setSweave("graph01", 6, 1.5)
> #
> AA.pl <- timePlot(DD, YY, Plot=list(what="none"), ylabels=c(0,1), ytitle="")
> # add 15-minute minor ticks (3 minor ticks in 1 hour)
> addMinorTicks("bottom", AA.pl, 3)
> addMinorTicks("top", AA.pl, 3)
> graphics.off()
```
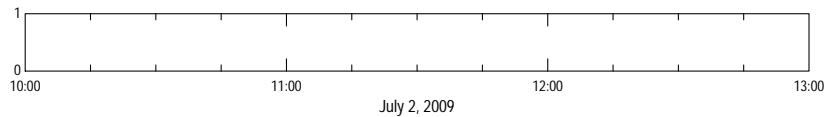

Figure 1. Hour format with 15-minute minor ticks.

The next example demonstrates the hour format suitable for a timeframe between about 1 and 3 days. Note that the upper limit for the x-axis range is the day after the day being plotted. The upper limit requirement is required because the class "Date" data are always interpreted as midnight of the day. Minor ticks at hourly intervals are added in this example.

```
> # Set up the graphics environment, the equivalent call for an on screen
> #  device could be setPage("square")
> setSweave("graph02", 6, 1.5)
> #
> AA.pl <- timePlot(DD, YY, Plot=list(what="none"), ylabels=c(0,1), ytitle="",
+                   xaxis.range=as.Date(c("2009-07-02", "2009-07-03")))
> # add hourly minor ticks (5 minor ticks in 6 hours)
> addMinorTicks("bottom", AA.pl, 5)
> addMinorTicks("top", AA.pl, 5)
> graphics.off()
```
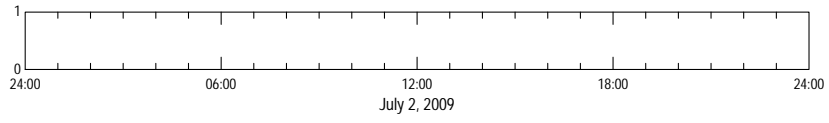
3

Figure 2. The default hour format for a timespan of 1 day.

3 Day Formats

Day formats are suitable for date ranges between 1 and several days, although generally less than 2 months. The format of the displayed day changes from the full date to just the day number as the length of the date range increases. Minor ticks are by default hours when the major ticks and labels are days.

This example replicates figure 2, but forces the format to be days. The default day labels are placed at the ticks in figure 3, but forced to be between the ticks in figure 4. Hourly ticks are added in both figures; a value for the `ticks` argument is not needed—all minor ticks are drawn.

```
> # Set up the graphics environment, the equivalent call for an on screen
> #  device could be setPage("square")
> setSweave("graph03", 6, 1.5)
> #
> AA.pl <- timePlot(DD, YY, Plot=list(what="none"), ylabels=c(0,1), ytitle="",
+                   xaxis.range=as.Date(c("2009-07-02", "2009-07-03")),
+                   xlabels="days")
> # add hourly minor ticks (5 minor ticks in 6 hours)
> addMinorTicks("bottom", AA.pl, 23)
> addMinorTicks("top", AA.pl, 23)
> graphics.off()
```
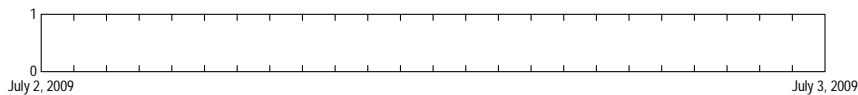

Figure 3. The day format for a timespan of 1 day.

```
> # Set up the graphics environment, the equivalent call for an on screen
> #  device could be setPage("square")
> setSweave("graph04", 6, 1.5)
> #
> AA.pl <- timePlot(DD, YY, Plot=list(what="none"), ylabels=c(0,1), ytitle="",
+                   xaxis.range=as.Date(c("2009-07-02", "2009-07-03")),
+                   xlabels=list(major="days", style="between"))
> # add hourly minor ticks
> addMinorTicks("bottom", AA.pl)
> addMinorTicks("top", AA.pl)
> graphics.off()
```
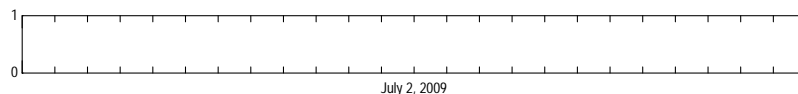
Figure 4. The format for a timespan of 1 day with the day label placed between the ticks.

The default x-axis format for timeframes longer than 3 days, but shorter than 2 months is days. Figures 5, 6, and 7 show how the format for displaying the day changes as the timeframe increases. No ticks are drawn in any of those figures.

```
> # Set up the graphics environment, the equivalent call for an on screen
> #  device could be setPage("square")
> setSweave("graph05", 6, 1.5)
> #
> AA.pl <- timePlot(DD, YY, Plot=list(what="none"), ylabels=c(0,1), ytitle="",
+                   xaxis.range=as.Date(c("2009-07-01", "2009-07-05")))
> graphics.off()
```
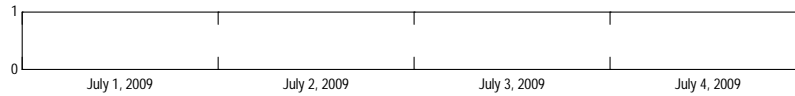

Figure 5. The default format for a timespan of 4 days.

```
> # Set up the graphics environment, the equivalent call for an on screen
> #  device could be setPage("square")
> setSweave("graph06", 6, 1.5)
> #
> AA.pl <- timePlot(DD, YY, Plot=list(what="none"), ylabels=c(0,1), ytitle="",
+                   xaxis.range=as.Date(c("2009-06-28", "2009-07-05")))
> graphics.off()
```
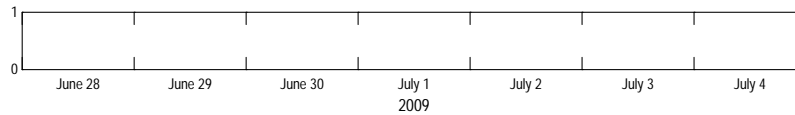

Figure 6. The default format for a timespan of 7 days.

```
> # Set up the graphics environment, the equivalent call for an on screen
> #  device could be setPage("square")
> setSweave("graph07", 6, 1.5)
> #
> AA.pl <- timePlot(DD, YY, Plot=list(what="none"), ylabels=c(0,1), ytitle="",
+                   xaxis.range=as.Date(c("2009-06-20", "2009-07-18")))
> graphics.off()
```
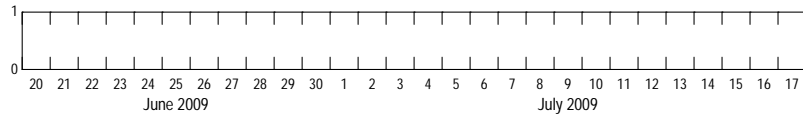
Figure 7. The default format for a timespan of 28 days.

4 Month Formats

Just as the format of the day changes as the timespan changes, the format of the month name changes from the full month name to the abbreviated month name to single letters. Single letter abbreviations are not acceptable for U.S. Geological Survey publication standards (U.S. Geological Survey, written commun., October 2015), but are included as an option for quick plot review, editing by an illustrator, or others who wish to use them. Minor ticks are days by default when the major ticks and labels are months. To draw minor ticks at specific day intervals requires including an argument set up like `xlabels=list(major="months", minor="5 days")` in the call to `timePlot`.

Figures 8, 9, and 10 show the change in month format as the timespan increases from 2 months to 18 months. Minor ticks indicating days are shown only for figure 8. Note that the format for the month abbreviations in figure 10 can be changed to the standard abbreviation by including `xlabels=list(major="months", label.abbr=2)` in the call to `timePlot`.

```
> # Set up the graphics environment, the equivalent call for an on screen
> #  device could be setPage("square")
> setSweave("graph08", 6, 1.5)
> #
> AA.pl <- timePlot(DD, YY, Plot=list(what="none"), ylabels=c(0,1), ytitle="",
+                 xaxis.range=as.Date(c("2009-06-01", "2009-08-01")))
> # add daily minor ticks
> addMinorTicks("bottom", AA.pl)
> addMinorTicks("top", AA.pl)
> graphics.off()
```
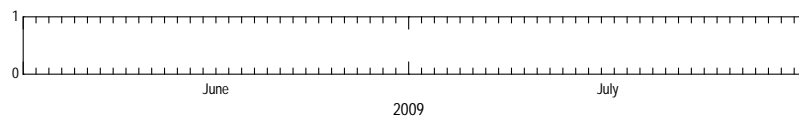

Figure 8. The default format for a timespan of 2 months.

```
> # Set up the graphics environment, the equivalent call for an on screen
> #  device could be setPage("square")
> setSweave("graph09", 6, 1.5)
> #
> AA.pl <- timePlot(DD, YY, Plot=list(what="none"), ylabels=c(0,1), ytitle="",
+                 xaxis.range=as.Date(c("2009-06-01", "2010-06-01")))
> graphics.off()
```
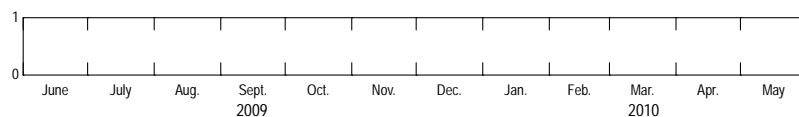
Figure 9. The default format for a timespan of 1 year.

```
> # Set up the graphics environment, the equivalent call for an on screen
> #  device could be setPage("square")
> setSweave("graph10", 6, 1.5)
> #
> AA.pl <- timePlot(DD, YY, Plot=list(what="none"), ylabels=c(0,1), ytitle="",
+                   xaxis.range=as.Date(c("2009-01-01", "2010-06-01")))
> graphics.off()
```
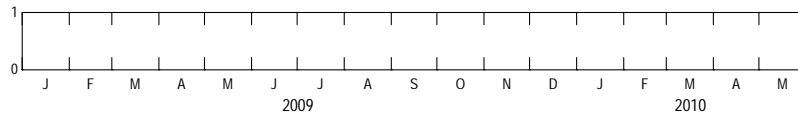

Figure 10. The default format for a timespan of 18 months.

5 Year Formats

Year formats are suitable for timespans from 1 to many years, although the month format may be used for timespans of a single year. In contrast to the previous formats, the year is always displayed in the 4-digit format, but longer timespans will force the displayed year to be displayed in multiples of 2, 5, 10, or longer. Minor ticks are by default months when the major ticks and labels are years.

The year-axis formats are shown for timespans of 2 years (fig. 11), where the individual years are shown, and 14 years (fig. 12), which shows the multiple year format. The default format for 2 years is months; figure 11 forces the format to be year. Minor ticks are drawn for months in figure 11 and the intervening year in figure 12.

```
> # Set up the graphics environment, the equivalent call for an on screen
> #  device could be setPage("square")
> setSweave("graph11", 6, 1.5)
> #
> AA.pl <- timePlot(DD, YY, Plot=list(what="none"), ylabels=c(0,1), ytitle="",
+                   xaxis.range=as.Date(c("2009-01-01", "2011-01-01")),
+                   xlabels="year")
> # add monthly minor ticks
> addMinorTicks("bottom", AA.pl)
> addMinorTicks("top", AA.pl)
> graphics.off()
```
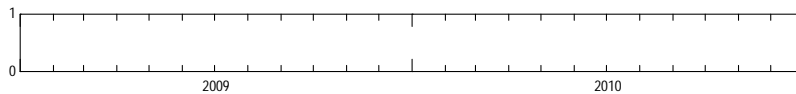

Figure 11. The years format for a timespan of 2 years.

```
> # Set up the graphics environment, the equivalent call for an on screen
> #  device could be setPage("square")
> setSweave("graph12", 6, 1.5)
> #
> AA.pl <- timePlot(DD, YY, Plot=list(what="none"), ylabels=c(0,1), ytitle="",
+                   xaxis.range=as.Date(c("2000-01-01", "2013-06-01")))
> # add intervening year minor ticks
> addMinorTicks("bottom", AA.pl, 1)
> addMinorTicks("top", AA.pl, 1)
> graphics.off()
```

Figure 12. The default format for a timespan of 14 years.

6 Water Year Formats

The water-year begins on October 1 of the previous year and ends on September 30 of that year. Water year formats are suitable for timespans from 1 to several years, although the month format may be used for timespans of a single year. Long-term water-year summary data can be displayed using a numeric axis with axis titles, or captions, indicating that the unit of time is the water year. The water-year format is never a default, it must always be specified by the `xlabels` argument.

Figures 13 and 14 show the water year format for 1 and 5 water years, respectively. The format changes after a timespan of 2 water years. Minor ticks are drawn only in figure 13.

```
> # Set up the graphics environment, the equivalent call for an on screen
> #  device could be setPage("square")
> setSweave("graph13", 6, 1.5)
> #
> AA.pl <- timePlot(DD, YY, Plot=list(what="none"), ylabels=c(0,1), ytitle="",
+                   xaxis.range=as.Date(c("2008-10-01", "2009-10-01")),
+                   xlabels="water year")
> # add intervening year minor ticks
> addMinorTicks("bottom", AA.pl)
> addMinorTicks("top", AA.pl)
> graphics.off()
```
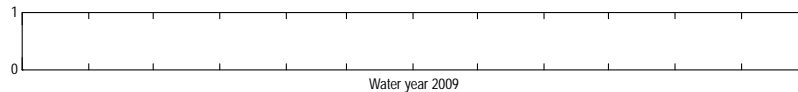

Figure 13. The format for a timespan of 1 water year.

```
> # Set up the graphics environment, the equivalent call for an on screen
> #  device could be setPage("square")
> setSweave("graph14", 6, 1.5)
> #
> AA.pl <- timePlot(DD, YY, Plot=list(what="none"), ylabels=c(0,1), ytitle="",
+                   xaxis.range=as.Date(c("2005-10-01", "2010-10-01")),
+                   xlabels="water year")
> graphics.off()
```
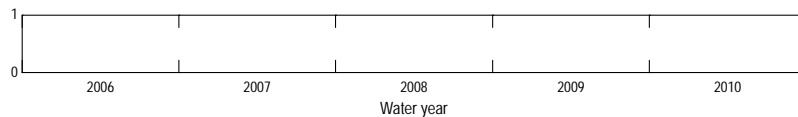

Figure 14. The format for a timespan of 5 water years.