

# timechart — A package for drawing chronological charts\*

Alan J. Cain

Released 2025-02-15

## Abstract

This package provides for the easy creation of chronological charts which show visually the relative historical positions of people and events. Each event or period can be specified by a single line of  $\LaTeX$  code comprising (possibly uncertain) start and finish dates and a label, and the package takes care of indicating the uncertainties and whether intervals extend beyond the specified bounds of the chart.

## Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Introduction</b>   | <b>3</b> |
| <b>2</b> | <b>Requirements</b>   | <b>3</b> |
| <b>3</b> | <b>Installation</b>   | <b>3</b> |
| <b>4</b> | <b>Getting started</b>  | <b>3</b> |
| <b>5</b> | <b>Example</b>  | <b>3</b> |
| <b>6</b> | <b>Configuration</b>  | <b>6</b> |
| <b>7</b> | <b>Specifying dates and date ranges</b>                                   | <b>7</b> |
| <b>8</b> | <b>timechart environment</b>  | <b>8</b> |
| 8.1      | General configuration of the <code>timechart</code> environment . . . . . | 8        |
| 8.2      | Grid configuration . . . . .  | 9        |
| 8.3      | Axis configuration . . . . .  | 9        |
| <b>9</b> | <b>Within the timechart environment</b>                                   | <b>9</b> |
| 9.1      | Intervals . . . . .   | 9        |
|          | Interval configuration . . . . .  | 10       |
| 9.2      | Text . . . . .  | 11       |
|          | Text configuration . . . . .  | 11       |
| 9.3      | Space . . . . .   | 12       |
| 9.4      | Positioning . . . . .   | 12       |
| 9.5      | Completion . . . . .  | 13       |
| 9.6      | Shortcut keys . . . . .   | 13       |

---

\*This file describes v0.51, last revised 2025-02-15.

|           |                                       |           |
|-----------|---------------------------------------|-----------|
| <b>10</b> | <b>Era indicators</b>                 | <b>13</b> |
| <b>11</b> | <b>Legend</b>                         | <b>13</b> |
| <b>12</b> | <b>Usage notes</b>                    | <b>14</b> |
| 12.1      | Additional TikZ code . . . . .        | 14        |
| 12.2      | ‘year zero’ . . . . .                 | 15        |
| 12.3      | dimension too large error . . . . .   | 15        |
| <b>13</b> | <b>Appendix: full example source</b>  | <b>15</b> |
| 13.1      | Setup source . . . . .                | 15        |
| 13.2      | Timechart source . . . . .            | 16        |
| 13.3      | Legend source . . . . .               | 19        |
| <b>14</b> | <b>Implementation</b>                 | <b>20</b> |
| 14.1      | Coding standard . . . . .             | 20        |
| 14.2      | Initial set-up . . . . .              | 20        |
| 14.3      | Load TikZ . . . . .                   | 20        |
| 14.4      | Scratch variables . . . . .           | 20        |
| 14.5      | Generic auxiliary functions . . . . . | 21        |
| 14.6      | PGF auxiliary functions . . . . .     | 21        |
| 14.7      | PGF keys . . . . .                    | 23        |
| 14.8      | Main environment . . . . .            | 26        |
| 14.9      | Grid drawing . . . . .                | 30        |
| 14.10     | Axis drawing . . . . .                | 30        |
| 14.11     | Bounding box . . . . .                | 35        |
| 14.12     | Positioning . . . . .                 | 35        |
| 14.13     | Bounds checking . . . . .             | 36        |
| 14.14     | Date and date range parsing . . . . . | 37        |
| 14.15     | Interval drawing . . . . .            | 42        |
| 14.15.1   | Preliminaries . . . . .               | 42        |
| 14.15.2   | Error message definition . . . . .    | 42        |
| 14.15.3   | Main macros . . . . .                 | 42        |
| 14.15.4   | Clipping . . . . .                    | 47        |
| 14.15.5   | Fading in/out . . . . .               | 49        |
| 14.15.6   | Slant in/out . . . . .                | 51        |
| 14.15.7   | Solid bar . . . . .                   | 53        |
| 14.15.8   | Marks . . . . .                       | 54        |
| 14.15.9   | Bar node . . . . .                    | 54        |
| 14.15.10  | Label . . . . .                       | 55        |
| 14.16     | Text . . . . .                        | 57        |
| 14.17     | Space . . . . .                       | 58        |
| 14.18     | Legends . . . . .                     | 59        |

## 1 Introduction

The `timechart` package provides a system for the easy creation of chronological charts — of the type pioneered by Joseph Priestley (1733–1804) in his ‘Chart of Biography’ and more famously in his ‘New Chart of History’<sup>1</sup> — which can show visually the relative historical positions of people and events. An example of what `timechart` can be used to produce is shown in [Figure 1](#) on page 4.

Essentially (and this was a design requirement), *only one line* of  $\LaTeX$  code is required for each interval (which, in the case of [Figure 1](#), are mostly lifetimes). The `timechart` package takes care of indicating ranges of possible dates by suitable fading from or to transparency. It also handles indicators to show that intervals continue outside the specified scope of the chart.

`timechart` was developed from, and replaced, a set of macros used to create the chronological charts in the author’s book *Form & Number: A History of Mathematical Beauty*, which is available on the Internet Archive under a Creative Commons licence.<sup>2</sup>

**Licence.** `timechart` is released under the LaTeX Project Public Licence v1.3c or later.<sup>3</sup>

**Acknowledgements.** The author thanks Tânia Paulista for reading and commenting on an earlier draft of the documentation.

## 2 Requirements

`timechart` requires PGF/TikZ and a  $\LaTeX$  kernel with `expl3` support (any kernel version since 2020-02-02 should suffice).

## 3 Installation

To install manually, run `tex timechart.ins` and copy the file `timechart.sty` to somewhere  $\LaTeX$  can find it.

## 4 Getting started

The `timechart` package is loaded as usual via `\usepackage{timechart}`. There are no package options.

The small example in [Section 5](#) illustrates the basic principles of `timechart`. [Section 13](#) shows the full code used to produce the large example in [Figure 1](#).

## 5 Example

This section illustrates how to create the small chart shown in [Figure 3](#) on page 6.

The basic environment is `timechart`, which includes the specification of the start and finish years. The start year 50 BCE is specified as `-50`, the finish year 75 CE as `75`. Each interval in the chart is specified using the `\timechartinterval` command, which

---

<sup>1</sup>URL: [https://commons.wikimedia.org/wiki/File:A\\_New\\_Chart\\_of\\_History\\_color.jpg](https://commons.wikimedia.org/wiki/File:A_New_Chart_of_History_color.jpg)

<sup>2</sup>URL: [https://archive.org/details/cain\\_formandnumber\\_ebook\\_large](https://archive.org/details/cain_formandnumber_ebook_large)

<sup>3</sup>URL: <https://www.latex-project.org/lppl.txt>

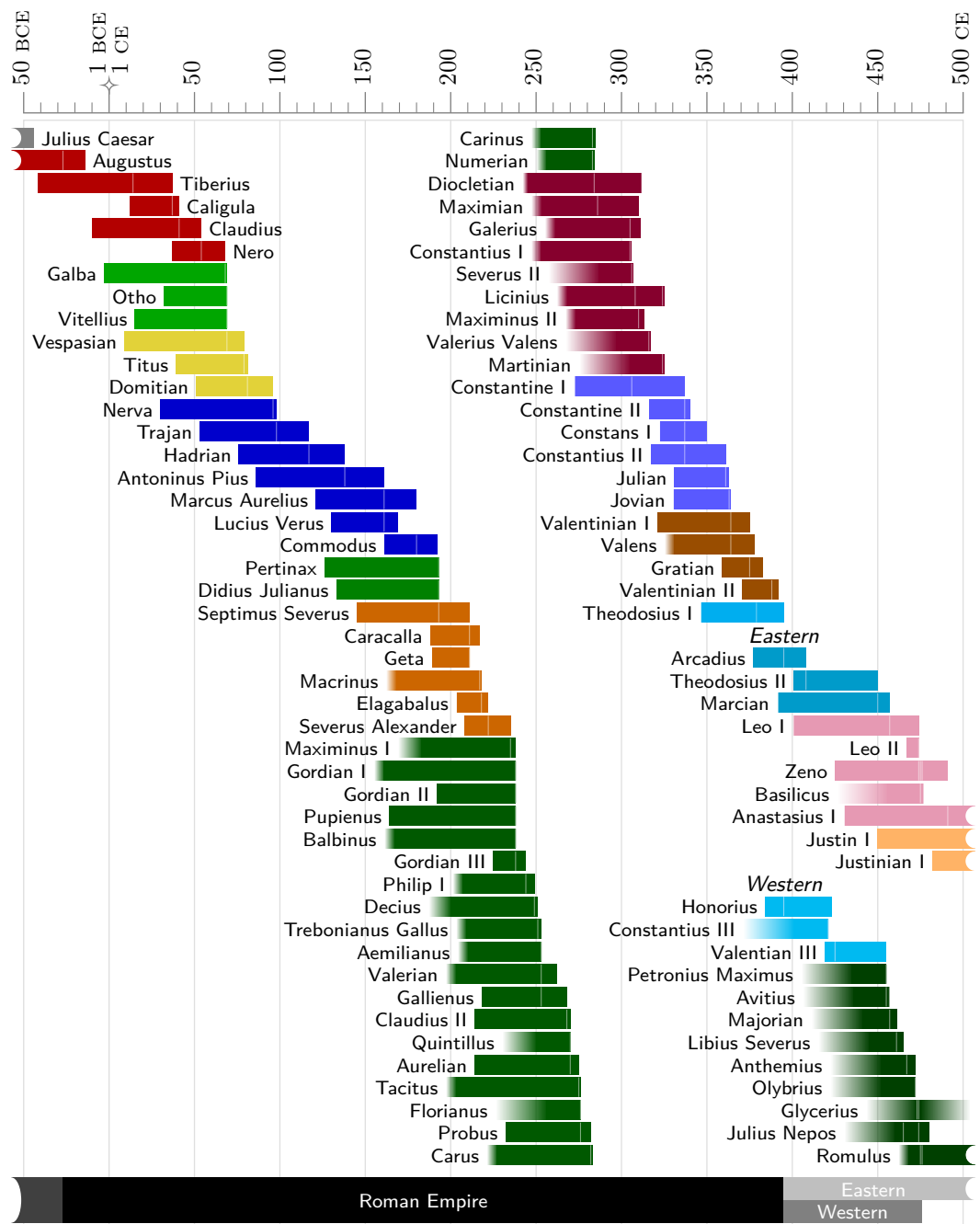


Figure 1: Timechart showing the lifetimes of Roman emperors from 50 BCE to 500 CE. Marks on each lifetime indicate the beginning (and, where relevant, the end) of that emperor's reign. Colours generally indicate dynasties, with shades of green indicating periods when the imperial power shifted between many short-reigning emperors.

takes three mandatory parameters: a start year, a finish year, and a label. The following code produces the flawed preliminary version shown in [Figure 2](#).

```

1 \begin{timechart}{-50}{75}
2   \timechartinterval{-63}{-14}{Augustus}
3   \timechartinterval{-42}{37}{Tiberius}
4   \timechartinterval{12}{41}{Caligula}
5   \timechartinterval{-10}{54}{Claudius}
6   \timechartinterval{37}{68}{Nero}
7 \end{timechart}

```

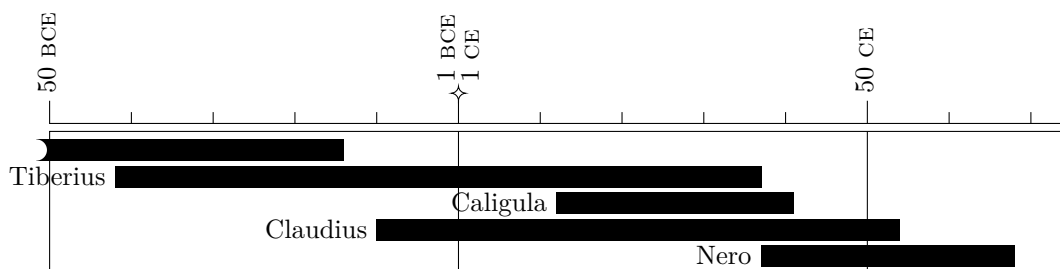


Figure 2: Flawed preliminary version of a chart showing the lifetimes of Roman emperors of the Julio-Claudian dynasty. (The final version is shown in [Figure 3](#).)

This first attempt result illustrates some of the principles of `timechart`. Each interval has been placed on its own line. More precisely, the  $y$  coordinate of the first interval is 0 and each use of `\timechartinterval` increments the ‘current  $y$  coordinate’ by a specified amount. (Various commands are available to set the  $y$  coordinate manually or to reset it automatically when it passes certain bounds; see [Subsection 9.4](#).) Horizontally, the chart starts and finishes at the specified years. The topmost interval, indicating Augustus’ life, has been truncated with an indicator that it begins before the specified start year of the chart. Vertically, the chart has been sized to fit around the specified intervals.

But this version is hardly satisfactory, for many reasons. The chart finishes between two minor ticks on the axis, because the intervals between major and minor ticks default respectively to 10 years and 50 years. The black intervals and text do not contrast well with the black axis and grid. The serif font is perhaps not best suited to label the intervals. And the label ‘Augustus’ has been lost, since labels are by default placed on the left of intervals. To rectify these problems, some changes are necessary, all of which can be made using key-value syntax in an optional parameter to the `timechart` environment or the `\timechartinterval` command:

1. Set the intervals between major and minor ticks to 5 years and 25 years respectively, by setting `minor tick interval=5` and `major tick interval=25`.
2. Change the colour of the grid to light grey by appending `draw=lightgray` to the `grid` style
3. Change the colour of the axis to grey by appending `draw=gray` to the `axis line`, `minor tick`, and `major tick` styles.
4. Change the font used for interval labels to small san-serif by appending `node font=\sfamily\small` to the `interval label` style

5. Change the colour of the intervals by defining a style `julioclaudian` that sets `interval bar color=red!80!black` and applying it to each interval via its optional argument. (While `interval bar color` could be set locally for each interval, it is better to define a style that corresponds to the semantic meaning of the colour: in this case, a single dynasty.)
6. Use the `right` key to place some labels on the right

The result is the following code, which produces [Figure 3](#).

```

1 \begin{timechart}[
2   minor tick interval=5,
3   major tick interval=25,
4   grid/.append style={ draw=lightgray },
5   axis line/.append style={ draw=gray },
6   minor tick/.append style={ draw=gray },
7   major tick/.append style={ draw=gray },
8   interval label/.append style={ node font=\sffamily\small },
9   julioclaudian/.style={ interval bar color=red!80!black },
10 ]{-50}{75}
11 \timechartinterval[right,julioclaudian]{-63}{-14}{Augustus}
12 \timechartinterval[right,julioclaudian]{-42}{37}{Tiberius}
13 \timechartinterval[julioclaudian]{12}{41}{Caligula}
14 \timechartinterval[julioclaudian]{-10}{54}{Claudius}
15 \timechartinterval[julioclaudian]{37}{68}{Nero}
16 \end{timechart}

```

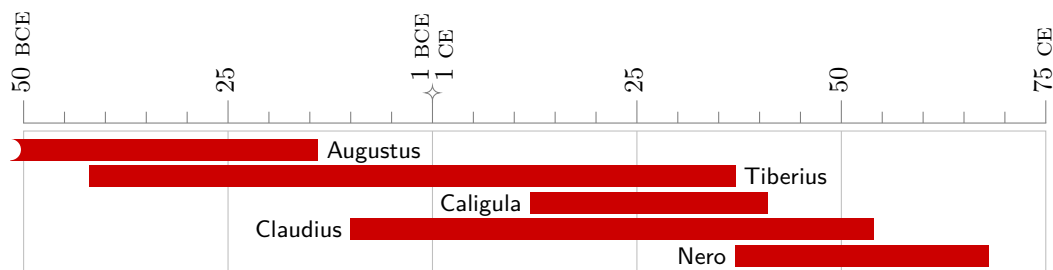


Figure 3: Chart showing the lifetimes of Roman emperors of the Julio-Claudian dynasty.

## 6 Configuration

All `timechart` configuration, global or local, is via PGF keys, so some familiarity with their use is beneficial; see the PGF/TikZ manual.

Configuration keys for `timechart` are contained in `/timechart/` in the PGF keys hierarchy. The `<options>` passed to the `timechart` environment or any of the commands `\timechartinterval`, `\timecharttext`, or `\timechartspace` are processed within `/timechart/` (since `/timechart/.cd` is executed before keys are processed).

The user may wish to define PGF styles for different kinds of interval within a chart. For example, one could define styles `science` and `art` that set a particular colour for the interval, and write `\timechartinterval[science]{<birth>}{<death>}{<name>}` or

`\timechartinterval[art]{\langle birth \rangle}{\langle death \rangle}{\langle name \rangle}` to distinguish visually the lifetimes of various scientists and artists.

## 7 Specifying dates and date ranges

Using `timechart` requires specification of dates and date ranges for the start and finish of each interval, both of which may be uncertain.

The basic specification of a date uses ISO 8601 format `YYYY-MM-DD`. This format specifies a date with day-level precision; use `YYYY-MM` and `YYYY` for month- and year-level precision. If the date is prefixed by `-`, it is treated as the corresponding date before the epoch. (This is a difference with ISO 8601, in which 0 represents 1 BCE, -1 represents 2 BCE, and so on.) So (assuming that one is using BCE/CE) one uses `-100` to indicate 100 BCE and `100` to indicate 100 CE. (The era indicators ‘BCE’ and ‘CE’ appear on the axis. Alternative era indicators — or a different epoch — can be specified; see [Section 10](#).)

A date can be prefixed with a `c` to indicate ‘circa’, such as `c-100` for ‘circa 100 BCE’ and `c100` for ‘circa 100 CE’. When an interval is drawn in a chart, ‘circa’ will be indicated by automatically creating (or extending) a range according to the value of the key `circa uncertainty` (see [Subsection 9.1](#)).

A date range comprises two dates (each with or without `c`) separated by a slash `/`, with the first date being earlier (or equal to) the second date. (The slash indicates a range of dates per ISO 8601.) A date range can be used to indicate a broader uncertainty than the default ‘circa’, or to indicate a definite range within which an interval starts or ends.

**Examples of correctly formatted dates and date ranges:** `-50`, `100`, `c-50`, `c100`, `-50/100`, `c-50 / +100`, `-50/c100`, `c-50/c100`, `-585-05-28`, `1947-12-01`, `1989-11`.

**Examples of incorrectly formatted date and date ranges:** `100?`, `100CE`, `100BCE`, `-50-100`, `100/-50`.

That is, the syntax for dates and date ranges is per the following (not-quite-formal) grammar:

|   |                  |  |
|---|------------------|--|
| <code>&lt; cdate-or-crangle &gt;</code> | <code>::=</code> | <code>&lt; cdate &gt;</code><br> <br><code>&lt; crangle &gt;</code>  |
| <code>&lt; cdate &gt;</code>            | <code>::=</code> | <code>&lt; date &gt;</code><br> <br><code>c &lt; date &gt;</code>  |
| <code>&lt; date &gt;</code>             | <code>::=</code> | <code>&lt; pdate &gt;</code><br> <br><code>- &lt; pdate &gt;</code>  |
| <code>&lt; pdate &gt;</code>            | <code>::=</code> | <code>&lt; year &gt;- &lt; month &gt;- &lt; day &gt;</code><br> <br><code>&lt; year &gt;- &lt; month &gt;</code><br> <br><code>&lt; year &gt;</code>   |
| <code>&lt; crangle &gt;</code>          | <code>::=</code> | <code>&lt; date<sub>1</sub> &gt;/ &lt; date<sub>2</sub> &gt;</code><br> <br><code>c &lt; date<sub>1</sub> &gt;/ &lt; date<sub>2</sub> &gt;</code><br> <br><code>&lt; date<sub>1</sub> &gt;/ c &lt; date<sub>2</sub> &gt;</code><br> <br><code>c &lt; date<sub>1</sub> &gt;/ c &lt; date<sub>2</sub> &gt;</code> (with <code>date<sub>1</sub> ≤ date<sub>2</sub></code> ) |

The bounds of the `timechart` environment (see [Section 8](#)) must satisfy `< cdate >` in this grammar (although only the `< year >` is used); the start and finish dates of an `\timechartinterval` command (see [Subsection 9.1](#)) must satisfy `< cdate-or-crangle >`; the parameter of an `\timecharttext` command (see [Subsection 9.2](#)) must satisfy `< date >`.

*Note.* For performance reasons, the date parser does only limited error-checking. Months outside the range from 01 to 12 or days outside the range of the specified month will be ignored. Otherwise malformed dates or date ranges may produce obscure error messages or unexpected results.

## 8 timechart environment

---

```
timechart \begin{timechart}[(options)]{(start)}{(finish)}
          <content>
          \end{timechart}
```

This is the main environment for creating a chronological chart. The mandatory arguments `<start>` and `<finish>` specify the first and last years of the chart. These can be dates with circa indicators (that is, they satisfy `<cdate>` in the grammar in [Section 7](#)), but the circa specifier has no effect and only the ‘year’ part of the date is used. The optional argument `<options>` supplies PGF keys that apply to the entire chart.

The `<content>` comprises commands like `\timechartinterval`, `\timecharttext`, `\timechartspace`, commands for positioning, as described in [Section 9](#), and the user’s own TikZ code.

### 8.1 General configuration of the timechart environment

- `/timechart/width=<dimension>` (default `\textwidth`)  
The width of the chart. This refers to the width of the grid and axis of the chart, not including intervals that pass beyond the specified limits of the chart, or axis labels that protrude beyond the width of the axis itself.
- `/timechart/tolerance=<dimension>` (default `5pt`)  
The length by which an interval is allowed to pass beyond the limits of the chart before it ‘counts’ as doing so and the appropriate indicator is drawn.
- `/timechart/beyond length=<dimension>` (default `5pt`)  
The length of the indicator that an interval passes beyond the limits of the chart.
- `/timechart/beyond x radius=<dimension>` (default `4pt`)  
The horizontal radius of the concave part of the indicator that an interval passes beyond the limits of the chart. (The vertical radius will be half the thickness of the bar.)
- `/timechart/ystep=<dimension>` (default `-10pt`)  
The default length (positive or negative) by which the current *y* coordinate is automatically adjusted after each interval, text, or space is placed.
- `/timechart/minor tick interval=<number>` (default `10`)  
The number of years (which must be positive) between each minor tick on the axis.
- `/timechart/major tick interval=<number>` (default `50`)  
The number of years (which must be positive) between each major tick on the axis and each vertical line in the grid.



## 8.2 Grid configuration

`/timechart/no grid`  
Do not draw the grid.

`/timechart/grid top ysep=<dimension>` (default 3pt)  
Distance between the top of the grid and the topmost interval or space.

`/timechart/grid bottom ysep=<dimension>` (default 3pt)  
Distance between the bottom of the grid and the bottommost interval or space.

`/timechart/grid` (style, initially empty)  
Style for drawing the grid.

## 8.3 Axis configuration

`/timechart/no axis`  
Do not draw the axis.

`/timechart/axis line` (style, initially `line cap=rect`)  
Style for drawing the axis line.

`/timechart/minor tick` (style, initially empty)  
Style for drawing minor ticks.

`/timechart/minor tick length=<dimension>` (default 1.5mm)  
Length of minor ticks.

`/timechart/major tick` (style, initially empty)  
Style for drawing major ticks.

`/timechart/major tick length=<dimension>` (default 3mm)  
Length of major ticks.

`/timechart/major tick label` (style, initially as described below)  
Style for labels on the major ticks on the axis. The initial style essentially sets `inner sep=0`, `outer sep=0`, `anchor=mid west`, `rotate=90`.

# 9 Within the timechart environment

## 9.1 Intervals

---

---

`\timechartinterval` `\timechartinterval[<options>]{<start>}{<finish>}{<label>}`

This command creates an interval in the chart at the current  $y$  coordinate between the specified `<start>` and `<finish>`, with the given `<label>`. These arguments are mandatory. Each of `<start>` and `<finish>` can be either a year or a range of years, possibly with circa markers. That is, each must satisfy `<year-or-crangle>` in the grammar in [Section 7](#).

The optional argument `<options>` specifies PGF keys within `/timechart/` that are applied locally to the interval.



The current  $y$  coordinate will be adjusted according to `/timechart/ystep` unless `/timechart/no autostep` has been set.



## Interval configuration

- `/timechart/no autostep`  
Do not automatically alter the current  $y$  coordinate by the amount specified in `/timechart/ystep`.
- `/timechart/ref=<label>` (default none)  
Make the interval label a hyperlink to the position labelled by `<label>`.
- `/timechart/mark=<comma-separated list of years>` (default empty)  
Draw marks in the interval at the years contained in the list. Each entry in the list must be a definite year (that is, must satisfy `<year>` in the grammar in [Section 7](#)). The colour of marks can be specified using `/timechart/interval mark color`. `/timechart/marks` is a synonym for this key.
- `/timechart/circa uncertainty=<number>` (default 3)  
Treat a circa indicator  $c$  as indicating an uncertainty of  $\pm<number>$ .
- `/timechart/interval minimum width=<dimension>` (default 1pt)  
Ensure that any interval has a width of at least `<dimension>`. This is useful to ensure that a single event is visible in the chart.  
If an interval is specified with start and finish ranges, and with `start range=fade` and `finish range=fade`, then the ‘certain’ portion of the interval will also have width at least `<dimension>`. (This restriction prevents a common rendering error where start and finish fadings around a ‘certain’ interval of length 0 would not quite meet.)
- `/timechart/interval bar color=<color>` (default black)  
Fill the interval bar with `<color>`.
- `/timechart/interval bar thickness=<dimension>` (default 8pt)  
Set the vertical thickness of the interval bar to `<dimension>`.
- `/timechart/interval bar node name=<string>` (default bar node)  
Set the name of the node containing the interval bar to `<string>`.
- `/timechart/interval mark color=<color>` (default gray)  
Draw marks using `<color>`.
- `/timechart/interval label` (style, initially empty)  
Style to apply to an interval label.
- `/timechart/interval label centered` (style, initially as below)  
Style to apply to an interval label placed centrally. Initially, this style executes the style `/timechart/interval label` and sets `text=white`. The reason for a separate style for centred labels is that often a contrasting colour will be required. For instance, labels positioned to the left and right may be black, but if the bar is black, a centred label should be a light colour.
- `/timechart/interval label centered background` (style, initially as below)  
Style to apply to the ‘background’ interval label placed centrally. Initially, this style executes `/timechart/interval label`. The ‘background’ interval label is simply the usual label and is placed in the same location, but on a layer behind the bar and, unlike the label itself, is not clipped. The reason for this style is that if the bar is narrow, part of the label text (such as ascenders and/or descenders) may naturally extend beyond the bar itself and it may be useful that these should appear in a different colour.
- `/timechart/interval label baseline=<dimension>` (default -3pt)  
Position the baseline of the interval label `<dimension>` below the current  $y$  coordinate (which is the midpoint of the interval bar).

`/timechart/interval label pos=<position>` (default `left`)  
 Specify where to place the label relative to the interval bar: `<position>` may be `left`, `center`, or `right`. The position `center` places the label at the midpoint of *the visible segment of the solid part* of the interval bar (that is, not including fading at the start or finish of the bar, and not including part of the bar that would extend beyond the bounds of the chart). Further, a centred label is clipped to the size of the bar and a unclipped ‘background’ copy of it is drawn behind the bar, so that the portion appearing ‘on’ and ‘outside’ the bar can have different styles. (See the keys `/timechart/interval label centered` and `/timechart/interval label centered background`.)

`/timechart/interval label node name=<string>` (default `label node`)  
 Set the name of the node containing the interval label to `<string>`.

`/timechart/start range=<range-type>` (default `fade`)  
 Type of indication of the range where an interval may start. `<range-type>` can be `fade`, which produces an indicator like , or `slant`, which produces .

`/timechart/finish range=<range-type>` (default `fade`)  
 Type of indication of the range where an interval may finish. `<range-type>` can be `fade`, which produces an indicator like , or `slant`, which produces .

`/timechart/fade minimum width=<dimension>` (default `0pt`)  
 No start or finish range indicator of type `fade` will be drawn if it is smaller than `<dimension>`. Using this key is sometimes necessary because, under certain limited circumstances, a very narrow fading can cause PGF/TikZ to produce a `dimension` too large error.

## 9.2 Text

---

`\timecharttext` `\timecharttext[<options>]{<year>}{<text>}`

Place `<text>` in the time chart at the current  $y$  coordinate and at the horizontal position of `<year>`, which must be a definite year (that is, must satisfy `<year>` in the grammar in [Section 7](#)).

The optional argument `<options>` specifies PGF keys within `/timechart/` that are applied locally.

The current  $y$  coordinate will be adjusted according to `/timechart/ystep` unless `/timechart/no autostep` has been set.

### Text configuration

`/timechart/text node name=<string>` (default `text node`)  
 Set the name of the node containing the text to `<string>`.

`/timechart/text` (style, initially empty)  
 Style to apply to the text.

`/timechart/text baseline=<dimension>` (default `-3pt`)  
 Position the baseline of the text `<dimension>` below the current  $y$  coordinate.

`/timechart/text pos=<position>` (default `left`)  
 Specify where to place the label relative to the given `<year>`: `<position>` may be `left`, `center`, or `right`.

### 9.3 Space

---

`\timechartspace` `\timechartspace[<options>]`

Create a space in the time chart at the current  $y$  coordinate, with the same effect on vertical spacing as an interval. More precisely, the current  $y$  coordinate will be adjusted according to `/timechart/ystep` unless `/timechart/no autostep` has been set.

The optional argument `<options>` specifies PGF keys within `/timechart/` that are applied locally.

### 9.4 Positioning

The commands `\timechartinterval`, `\timecharttext`, and `\timechartspace` all act at the current  $y$  coordinate and change its value according to `/timechart/ystep` (unless `/timechart/no autostep` is used). There are several functions to set the current  $y$  coordinate and to have it reset automatically.

---

`\timechartsety` `\timechartsety{<dimension>}`

Set the current  $y$  coordinate to `<dimension>`.

---

`\timechartsavey` `\timechartsavey`

Save the current  $y$  coordinate. If `\timechartresety` is used, the  $y$  coordinate resets to the last saved  $y$  coordinate. If the current  $y$  coordinate exceeds the minimum or maximum set by `\timechartsetyminimumautoreset` and `\timechartsetymaximumautoreset`, it will be reset to the last saved  $y$  coordinate.

---

`\timechartresety` `\timechartresety`

Reset the  $y$  coordinate to the last coordinate saved using `\timechartsavey`, or to 0 if there has been no use of `\timechartsavey` within the current `timechart` environment.

---

`\timechartsetyminimumautoreset` `\timechartsetyminimumautoreset{<min-coordinate>}`

`\timechartsetymaximumautoreset` `\timechartsetymaximumautoreset{<max-coordinate>}`

Set  $y$  coordinates that automatically trigger `\timechartresety` if the current  $y$  coordinate goes below `<min-coordinate>` or above `<max-coordinate>`.

---

`\timechartstepy` `\timechartstepy[<count>]`

Manually step the current  $y$  coordinate by `<count>` times the value of `/timechart/ystep`. The default value of `<count>` is 1. (The `/timechart/no autostep` does not affect `\timechartstepy`.)

## 9.5 Completion

---

`\timechartfinish` Signal that the chart is complete and that the grid and axis should be drawn (unless the keys `/timechart/no grid` and/or `/timechart/no axis` have been used). It is not necessary to use this command: if it is not given, the grid and axis will be drawn at the end of the `timechart` environment. But after this command, the TikZ nodes `grid` and `axis` are available, containing (respectively) the grid and the axis. These can be used in for further TikZ drawing.

Note that after `\timechartfinish`, none of the various `\timechart...` commands are available inside that `timechart` environment.

## 9.6 Shortcut keys

`/timechart/left`

Equivalent to setting `/timechart/interval label pos` and `/timechart/text pos` to `left`.

`/timechart/center`

Equivalent to setting `/timechart/interval label pos` and `/timechart/text pos` to `center`.

`/timechart/right`

Equivalent to setting `/timechart/interval label pos` and `/timechart/text posright` to `right`.

## 10 Era indicators

---

`\timechartmakebeforeyear` `\timechartmakebeforeyear{<number>}`  
`\timechartmakeafteryear` `\timechartmakeafteryear{<number>}`

Typeset `<number>` (which should be a positive whole number) as a year before or after the epoch. By default, `\timechartmakebeforeyear{<number>}` produces `<number>` BCE and `\timechartmakeafteryear{<number>}` produces `<number>` CE.

These commands are used for axis labels and can be re-defined by the user. For example, if BC/AD is preferred to BCE/CE, the user can define

```
1 \renewcommand*\timechartmakebeforeyear}[1]{#1-\textsc{bc}}
2 \renewcommand*\timechartmakeafteryear}[1]{\textsc{ad}~#1}
```

Similarly, if AH/BH is preferred, the user can define

```
1 \renewcommand*\timechartmakebeforeyear}[1]{#1-\textsc{bh}}
2 \renewcommand*\timechartmakeafteryear}[1]{\textsc{ah}~#1}
```

## 11 Legend

`timechart` supplies a number of auxiliary macros for creating a legend to explain, for example the significance of different colours of intervals. For example, [Figure 4](#) shows a suitable legend for [Figure 1](#).



Figure 4: Example legend for the timechart shown in Figure 1.

The `\timechartlength...` macros are *not* meant to be used inside a `timechart` environment, but in locations such as running text or a `tabular` environment.

---

`\timechartlegenditem` `\timechartlegenditem[options]`

Draw a bar suitable for use in a legend. *options* specifies PGF keys that are applied within `/timechart`. The same PGF keys that affect interval bars affect the drawn bar, as do the keys listed below.

---

`\timechartlegendstartrange` `\timechartlegendstartrange[options]`  
`\timechartlegendfinishrange` `\timechartlegendfinishrange[options]`

Draw a bar suitable for use in a legend, with a start or finish range. *options* specifies PGF keys that are applied within `/timechart/`. The same PGF keys that affect interval bars affect the drawn bar, as do the keys listed below.

`/timechart/legend item width=dimension` (default 9mm)

When using macros `\timechartlegenditem`, `\timechartlegendstartrange`, or `\timechartlegendfinishrange`, draw a bar of total width *dimension*.

`/timechart/legend item range width=dimension` (default 3mm)

When using `\timechartlegendstartrange` or `\timechartlegendfinishrange`, draw a bar with a range of width *dimension*.

## 12 Usage notes

### 12.1 Additional TikZ code

The `timechart` environment is a `tikzpicture`. The user can add any TikZ code before, between, or after content created using the `\timechart...` commands. Each use of `\timechartinterval` defines two nodes. One, by default named `bar node`, contains the interval bar; the other, by default named `label node`, contains the interval label. Similarly, text added using `\timecharttext` is contained in a node, by default

named `text node`. (The default names are re-used, but can be changed using the keys `/timechart/interval bar node name`, `/timechart/interval label node name`, and `/timechart/text node name`.) The user can use these nodes to position extra content.

If the `\timechartcomplete` command is used (after which the `\timechart...` commands are no longer available within the `timechart` environment) the nodes `grid` and `axis`, which contain the grid and the axis, are also available.

## 12.2 ‘year zero’

Although calendars typically do not admit a ‘year zero’ (for instance, 1 BCE is immediately followed by 1 CE, with no intervening ‘year zero’), `timechart` does allow 0 for the `<start>` or `<finish>` of the `timechart` environment or as the `<start>` or `<finish>` of `\timechartinterval` or the `<year>` of `\timecharttext`. But ‘year zero’ is indicated on the axis by a special epoch marker showing the last year before and first year after the epoch.

## 12.3 dimension too large error

Under certain limited circumstances, a very narrow fading can cause PGF/TikZ to produce a `dimension too large` error. This error can be triggered by a range of type `fade` at the start or end of an interval when the range is small compared to the width of the chart. In this case, the fading is unlikely to be visible, so one can simply set `/timechart/fade minimum width` to a suitable small positive value, which will stop the fading from being drawn and so prevent the error.

# 13 Appendix: full example source

This section contains the necessary source code to produce the example timechart and legend shown in [Figures 1 and 4](#) on pages [4 and 14](#).

## 13.1 Setup source

```
1 \pgfkeys{
2   /timechart/.cd,
3   julioclaudian/.style={
4     interval bar color=red!70!black,
5     interval mark color=red!70!black!50!white
6   },
7   fourempereors/.style={
8     interval bar color=green!65!black,
9     interval mark color=green!65!black!50!white
10  },
11  flavian/.style={
12    interval bar color=yellow!85!black,
13    interval mark color=yellow!85!black!50!white
14  },
15  nervaantonine/.style={
16    interval bar color=blue!80!black,
17    interval mark color=blue!80!black!50!white
18  },
19  fiveemperors/.style={
```

```

20     interval bar color=green!50!black,
21     interval mark color=green!50!black!50!white
22 },
23 severan/.style={
24     interval bar color=orange!80!black,
25     interval mark color=orange!80!black!50!white
26 },
27 thirdcentury/.style={
28     interval bar color=green!35!black,
29     interval mark color=green!35!black!50!white
30 },
31 tetrarchy/.style={
32     interval bar color=purple!70!black,
33     interval mark color=purple!70!black!50!white
34 },
35 constantinian/.style={
36     interval bar color=blue!65!white,
37     interval mark color=blue!65!white!50!white
38 },
39 valentinianic/.style={
40     interval bar color=orange!60!black,
41     interval mark color=orange!60!black!50!white
42 },
43 theodosian/.style={
44     interval bar color=cyan,
45     interval mark color=cyan!50!white
46 },
47 theodosian-east/.style={
48     interval bar color=cyan!80!black,
49     interval mark color=cyan!80!black!50!white
50 },
51 theodosian-west/.style={
52     interval bar color=cyan!80!white,
53     interval mark color=cyan!80!white!50!white
54 },
55 leonid/.style={
56     interval bar color=purple!40!white,
57     interval mark color=purple!40!white!50!white
58 },
59 justinian/.style={
60     interval bar color=orange!60!white,
61     interval mark color=orange!60!white!50!white
62 },
63 lastwest/.style={
64     interval bar color=green!25!black,
65     interval mark color=green!25!black!50!white
66 },
67 }

```

## 13.2 Timechart source

```

1 \begin{timechart}[
2     axis line/.append style={ draw=gray, line width=.5pt },
3     grid/.append style={ draw=lightgray!50!white, line width=.5pt },

```



```

4     minor tick/.append style={ draw=gray },
5     major tick/.append style={ line width=.5pt, draw=gray },
6     major tick label/.append style={ node font=\small },
7     interval label/.style={ node font=\sffamily\footnotesize },
8     text/.style={ node font=\sffamily\small\itshape, },
9     ystep=-3.25mm,
10    ]{-50}{500}
11
12    \pgfmathsetmacro{\mainlinecount}{46}
13
14    \timechartsetyminimumautoreset{-3.25mm*\mainlinecount+1mm}
15
16    \timechartinterval[interval bar color=gray,right,mark=-45]{-100}{-44}{Julius
17      Caesar}
18    \timechartinterval[julioclaudian,right,mark=-27]{-63}{-14}{Augustus}
19    \timechartinterval[julioclaudian,right,mark=14]{-42}{37}{Tiberius}
20    \timechartinterval[julioclaudian,right,mark=37]{12}{41}{Caligula}
21    \timechartinterval[julioclaudian,right,mark=41]{-10}{54}{Claudius}
22    \timechartinterval[julioclaudian,right,mark=54]{37}{68}{Nero}
23
24    \timechartinterval[foureemperors,mark=68]{-3}{69}{Galba}
25    \timechartinterval[foureemperors,mark=69]{32}{69}{Otho}
26    \timechartinterval[foureemperors,mark=69]{15}{69}{Vitellius}
27
28    \timechartinterval[flavian,mark=69]{9}{79}{Vespasian}
29    \timechartinterval[flavian,mark=79]{39}{81}{Titus}
30    \timechartinterval[flavian,mark=81]{51}{96}{Domitian}
31
32    \timechartinterval[nervaantonine,mark=96]{30}{98}{Nerva}
33    \timechartinterval[nervaantonine,mark=98]{53}{117}{Trajan}
34    \timechartinterval[nervaantonine,mark=117]{76}{138}{Hadrian}
35    \timechartinterval[nervaantonine,mark=138]{86}{161}{Antoninus Pius}
36    \timechartinterval[nervaantonine,mark=161]{121}{180}{Marcus Aurelius}
37    \timechartinterval[nervaantonine,mark=161]{130}{169}{Lucius Verus}
38    \timechartinterval[nervaantonine,mark=180]{161}{192}{Commodus}
39
40    \timechartinterval[fiveemperors,mark=193]{126}{193}{Pertinax}
41    \timechartinterval[fiveemperors,mark=193]{133}{193}{Didius Julianus}
42
43    \timechartinterval[severan,mark=193]{145}{211}{Septimus Severus}
44    \timechartinterval[severan,mark=211]{188}{217}{Caracalla}
45    \timechartinterval[severan,mark=211]{189}{211}{Geta}
46    \timechartinterval[severan,mark=217]{c165}{218}{Macrinus}
47    \timechartinterval[severan,mark=218]{203/204}{222}{Elagabalus}
48    \timechartinterval[severan,mark=222]{208}{235}{Severus Alexander}
49
50    \timechartinterval[thirdcentury,mark=235]{c172/c180}{238}{Maximinus I}
51    \timechartinterval[thirdcentury,mark=238]{c158}{238}{Gordian I}
52    \timechartinterval[thirdcentury,mark=238]{192}{238}{Gordian II}
53    \timechartinterval[thirdcentury,mark=238]{164}{238}{Pupienus}
54    \timechartinterval[thirdcentury,mark=238]{c164}{238}{Balbinus}
55    \timechartinterval[thirdcentury,mark=238]{225}{244}{Gordian III}
56    \timechartinterval[thirdcentury,mark=244]{c204}{249}{Philip I}
57    \timechartinterval[thirdcentury,mark=249]{c190/200}{251}{Decius}

```

57 \timechartinterval [thirdcentury,mark=251] {c206}{253}{Trebonianus Gallus}  
58 \timechartinterval [thirdcentury,mark=253] {c207}{253}{Aemilianus}  
59 \timechartinterval [thirdcentury,mark=253] {c200}{262}{Valerian}  
60 \timechartinterval [thirdcentury,mark=253] {218}{268}{Gallienus}  
61 \timechartinterval [thirdcentury,mark=268] {214}{270}{Claudius II}  
62 \timechartinterval [thirdcentury,mark=270] {230/250}{270}{Quintillus}  
63 \timechartinterval [thirdcentury,mark=270] {214}{275}{Aurelian}  
64 \timechartinterval [thirdcentury,mark=275] {c200}{276}{Tacitus}  
65 \timechartinterval [thirdcentury,mark=276] {226/256}{276}{Florianus}  
66 \timechartinterval [thirdcentury,mark=276] {232}{282}{Probus}  
67 \timechartinterval [thirdcentury,mark=282] {c224}{283}{Carus}  
68 \timechartinterval [thirdcentury,mark=283] {c250}{285}{Carinus}  
69 \timechartinterval [thirdcentury,mark=283] {c253}{284}{Numerian}  
70  
71 \timechartinterval [tetrarchy,mark=284] {242/245}{311/312}{Diocletian}  
72 \timechartinterval [tetrarchy,mark=286] {c250}{310}{Maximian}  
73 \timechartinterval [tetrarchy,mark=305] {c258}{311}{Galerius}  
74 \timechartinterval [tetrarchy,mark=305] {c250}{306}{Constantius I}  
75 \timechartinterval [tetrarchy,mark=306] {257/287}{307}{Severus II}  
76 \timechartinterval [tetrarchy,marks={308,324}] {c265}{325}{Licinius}  
77 \timechartinterval [tetrarchy,mark=310] {c270}{313}{Maximinus II}  
78 \timechartinterval [tetrarchy,mark=316] {267/297}{317}{Valerius Valens}  
79 \timechartinterval [tetrarchy,mark=324] {275/305}{325}{Martinian}  
80  
81 \timechartinterval [constantinian,mark=306] {272/273}{337}{Constantine I}  
82 \timechartinterval [constantinian,mark=337] {316}{340}{Constantine II}  
83 \timechartinterval [constantinian,mark=337] {322/323}{350}{Constans I}  
84 \timechartinterval [constantinian,mark=337] {317}{361}{Constantius II}  
85 \timechartinterval [constantinian,mark=361] {331}{363}{Julian}  
86 \timechartinterval [constantinian,mark=363] {330/331}{364}{Jovian}  
87  
88 \timechartinterval [valentinianic,mark=364] {321}{375}{Valentinian I}  
89 \timechartinterval [valentinianic,mark=364] {c328}{378}{Valens}  
90 \timechartinterval [valentinianic,mark=375] {359}{383}{Gratian}  
91 \timechartinterval [valentinianic,mark=388] {371}{392}{Valentinian II}  
92  
93 \timechartinterval [theodosian,mark=379] {346/347}{395}{Theodosius I}  
94 \timecharttext [center] {395}{Eastern}  
95 \timechartinterval [theodosian-east,mark=395] {377}{408}{Arcadius}  
96 \timechartinterval [theodosian-east,mark=408] {401}{450}{Theodosius II}  
97 \timechartinterval [theodosian-east,mark=450] {391/392}{457}{Marcian}  
98  
99 \timechartinterval [leonid,mark=457] {400/401}{474}{Leo I}  
100 \timechartinterval [leonid,mark=474] {467}{474}{Leo II}  
101 \timechartinterval [leonid,mark={474,475,476}] {425}{491}{Zeno}  
102 \timechartinterval [leonid,mark=475] {426/456}{476/477}{Basilicus}  
103 \timechartinterval [leonid,mark=491] {430/431}{518}{Anastasius I}  
104  
105 \timechartinterval [justinian,mark=518] {450}{527}{Justin I}  
106 \timechartinterval [justinian,mark=527] {482}{565}{Justinian I}  
107  
108 \timecharttext [center] {395}{Western}  
109 \timechartinterval [theodosian-west,mark=395] {384}{423}{Honorius}  
110 \timechartinterval [theodosian-west,mark=421] {371/401}{421}{Constantius III}

```

111 \timechartinterval[theodosian-west,mark=425]{419}{455}{Valentian III}
112
113 \timechartinterval[lastwest,mark=455]{405/435}{455}{Petronius Maximus}
114 \timechartinterval[lastwest,mark=455]{406/436}{456/457}{Avitius}
115 \timechartinterval[lastwest,mark=457]{411/441}{461}{Majorian}
116 \timechartinterval[lastwest,mark=461]{415/445}{465}{Libius Severus}
117 \timechartinterval[lastwest,mark=467]{422/452}{472}{Anthemius}
118 \timechartinterval[lastwest,mark=472]{422/452}{472}{Olybrius}
119 \timechartinterval[lastwest,mark={473,474}]{443/473}{474/504}{Glycerius}
120 \timechartinterval[lastwest,mark={474,465}]{430/460}{480}{Julius Nepos}
121 \timechartinterval[lastwest,mark={475,476}]{c465}{507/527}{Romulus}
122
123 \timechartsetyminimumautoreset{-\maxdimen}
124 \timechartsety{(-\mainlinecount-1)*3.25mm}
125
126 \timechartinterval[
127     center,
128     interval bar color=darkgray,
129     interval text/.append style={ text=white },
130     interval bar thickness=6.5mm
131 ]{-509}{-27}{}
132 \timechartstepy[-1]
133 \timechartinterval[
134     center,
135     interval bar color=black,
136     interval text/.append style={ node font=\sffamily\normalsize,text=white },
137     interval bar thickness=6.5mm
138 ]{-27}{395}{Roman Empire}
139 \timechartstepy[-1.5]
140 \timechartinterval[
141     center,
142     interval bar color=lightgray,
143     interval bar thickness=3.25mm
144 ]{395}{1453}{Eastern}
145 \timechartinterval[
146     center,
147     interval bar color=gray,
148     interval bar thickness=3.25mm
149 ]{395}{476}{Western}
150 \end{timechart}

```

### 13.3 Legend source

```

1 \begin{tabular}{rl}
2 \timechartlegenditem[julioclaudian] & Julio-Claudian dynasty & \\
3 \timechartlegenditem[fourempereors] & Year of four emperors & \\
4 \timechartlegenditem[flavian] & Flavian dynasty & \\
5 \timechartlegenditem[nervaantonine] & Nerva-Antonine dynasty & \\
6 \timechartlegenditem[fiveemperors] & Year of five emperors & \\
7 \timechartlegenditem[severan] & Severan dynasty & \\
8 \timechartlegenditem[thirdcentury] & Crisis of the third century & \\
9 \timechartlegenditem[tetrarchy] & Tetrarchy & \\
10 \timechartlegenditem[constantinian] & Constantinian dynasty & \\
11 \timechartlegenditem[valentinianic] & Valentinianic dynasty & \\

```

```

12 \timechartlegenditem[theodosian] & Theodosian dynasty \\
13 \timechartlegenditem[theodosian-east] & --- in the east \\
14 \timechartlegenditem[theodosian-west] & --- in the west \\
15 \timechartlegenditem[leonid] & Leonid dynasty \\
16 \timechartlegenditem[justinian] & Justinian dynasty \\
17 \timechartlegenditem[lastwest] & Last emperors in the west \\
18 \end{tabular}

```

## 14 Implementation

```

1 <*package>
2 <@@=timechart>

```

### 14.1 Coding standard

This package makes extensive use of `pgfmath` computations. The usual `expl3` standard of ending variables with a type indicator (`_bool`, `_int`, etc.) is therefore adapted as follows:

`_year` Stores a year, which could in principle be fractional.

`_pgf` Stores a length calculated by `pgfmath`. (Unlike `_dim`, there is no underlying dimension register.)

`_x` Stores a raw  $x$  coordinate (not in `TikZ`'s  $XY$ -coordinate system).

`_y` Stores a raw  $y$  coordinate (not in `TikZ`'s  $XY$ -coordinate system).

`_text` Stores text (not an `expl3` string).

### 14.2 Initial set-up

Package identification/version information.

```

3 \NeedsTeXFormat{LaTeX2e}[2020-02-02]
4 \ProvidesExplPackage{timechart}{2025-02-15}{0.51}
5 {Typesetting chronological charts}

```

### 14.3 Load TikZ

```

6 \RequirePackage{tikz}

```

In the remainder of the package, only a limited subset of `TikZ` is used, and `PGF` code is preferred. For `PGF` keys, it is necessary to use `~` in place of a space.

### 14.4 Scratch variables

```

\l__timechart_tmpa_bool Scratch boolean variables.
\l__timechart_tmpb_bool 7 \bool_new:N\l__timechart_tmpa_bool
\l__timechart_tmpc_bool 8 \bool_new:N\l__timechart_tmpb_bool
\l__timechart_tmpd_bool 9 \bool_new:N\l__timechart_tmpc_bool
10 \bool_new:N\l__timechart_tmpd_bool

```

*(End of definition for `\l__timechart_tmpa_bool` and others.)*

```

\l__timechart_tmpa_dim Scratch dimension variables, reusing \l_tmpa_dim and \l_tmpb_dim with uniform names.
\l__timechart_tmpb_dim 11 \cs_set_eq:NN\l__timechart_tmpa_dim\l_tmpa_dim
\l__timechart_tmpc_dim 12 \cs_set_eq:NN\l__timechart_tmpb_dim\l_tmpb_dim
\l__timechart_tmpd_dim 13 \dim_new:N\l__timechart_tmpc_dim
14 \dim_new:N\l__timechart_tmpd_dim

```

(End of definition for \l\_\_timechart\_tmpa\_dim and others.)

## 14.5 Generic auxiliary functions

`\__timechart_make_ref:NN` Make hyperreference from text, if the supplied target is non-empty.  
**#1** : Reference for hyperlink target, or empty.  
**#2** : Text.

```

15 \cs_new:Npn\__timechart_make_ref:NN #1#2
16 {
17   \str_if_empty:NTF #1
18     { #2 }
19     { \hyperref[#1]{#2} }
20 }

```

(End of definition for \\_\_timechart\_make\_ref:NN.)

## 14.6 PGF auxiliary functions

`__timechart_pgfmathsetbool:nn` Set an expl3 boolean variable to the outcome of a `pgfmath` comparison. This macro is simply a wrapper around `\pgfmathsetmacro` using `ifthenelse` and returning the boolean literal `true` or `false`.

```

21 \cs_new:Npn\__timechart_pgfmathsetbool:nn #1#2
22 {
23   \pgfmathsetmacro{#1}{ifthenelse(#2,"c_true_bool","c_false_bool")}
24 }

```

(End of definition for \_\_timechart\_pgfmathsetbool:nn.)

`\__timechart_if_equal:nnF` Use `pgfmath` to check whether `#1` and `#2` are equal. If not, execute `#3`.

```

25 \cs_new:Npn\__timechart_if_equal:nnF #1#2#3
26 {
27   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}{#1==#2}
28   \bool_if:NF\l__timechart_tmpa_bool{#3}
29 }

```

(End of definition for \\_\_timechart\_if\_equal:nnF.)

`\__timechart_pgfextractxy:nnn` Extract coordinates of `#3` (a PGF point) to dimension variables `#1` and `#2`. This macro simply combines the functionality of `\pgfextractx` and `\pgfextracty`.

```

30 \cs_new:Npn\__timechart_pgfextractxy:nnn #1#2#3
31 {
32   \pgf@process{#3}
33   #1=\pgf@x\relax
34   #2=\pgf@y\relax
35 }

```

(End of definition for \\_\_timechart\_pgfextractxy:nnn.)

`\_timechart_hsmash_pgfnode:nnnnn` Do the same as `\pgfnode` but only update the bounding box ‘vertically’.

```

36 \cs_new:Npn\_timechart_hsmash_pgfnode:nnnnn #1#2#3#4#5
37 {
38   \pgfinterruptboundingbox
39   \pgfnode{#1}{#2}{#3}{#4}{#5}
40   \pgfcoordinate
41     {\_timechart_tmpa_coord}
42     {\pgfpointanchor{current~bounding~box}{south}}
43   \pgfcoordinate
44     {\_timechart_tmpb_coord}
45     {\pgfpointanchor{current~bounding~box}{north}}
46   \endpgfinterruptboundingbox
47   \pgfextractx
48     {\l\_timechart_tmpa_dim}
49     {\pgfpointanchor{\_timechart_tmpa_coord}{center}}
50   \pgfextractx
51     {\l\_timechart_tmpb_dim}
52     {\pgfpointanchor{\_timechart_tmpb_coord}{center}}
53   \pgfpathmoveto{\pgfpoint{\l\_timechart_tmpa_dim}{0}}
54   \pgfpathmoveto{\pgfpoint{\l\_timechart_tmpb_dim}{0}}
55   \pgfusepath{discard}
56 }

```

*(End of definition for `\_timechart_hsmash_pgfnode:nnnnn`.)*

`\_timechart_make_rectangle_node:nnnn` Make a node with south west corner #1, north east corner #2, and name #3. #1 and #2 should be given as PGF points. #4 is a boolean literal indicating whether the path should be stroked.

```

57 \cs_new:Npn\_timechart_make_rectangle_node:nnnn #1#2#3#4
58 {
59   \group_begin:
60   \_timechart_pgfextractxy:nnn
61     {\l\_timechart_tmpa_dim}{\l\_timechart_tmpb_dim}{#1}
62   \_timechart_pgfextractxy:nnn
63     {\l\_timechart_tmpe_dim}{\l\_timechart_tmpe_dim}{#2}
64   \pgftransformshift{#1}
65   \pgfset{
66     minimum~width=\l\_timechart_tmpe_dim-\l\_timechart_tmpa_dim,
67     minimum~height=\l\_timechart_tmpe_dim-\l\_timechart_tmpe_dim,
68     inner~sep=0,
69     outer~sep=0,
70   }
71   \bool_if:NTF #4
72     { \pgfnode{rectangle}{south~west}{#3}{\pgfusepath{draw}} }
73     { \pgfnode{rectangle}{south~west}{#3}{\pgfusepath{discard}} }
74   \group_end:
75 }

```

*(End of definition for `\_timechart_make_rectangle_node:nnnn`.)*

`\l\_timechart_left_nonclip_x` `\l\_timechart_right_nonclip_x` In order to clip ‘only on one side’, define coordinates for specifying the ‘other side’ of the clipping path.

```

76 \pgfmathsetmacro{\l\_timechart_left_nonclip_x}{-16000pt}
77 \pgfmathsetmacro{\l\_timechart_right_nonclip_x}{16000pt}

```

(End of definition for `\l__timechart_left_nonclip_x` and `\l__timechart_right_nonclip_x`.)

`\__timechart_set_style_line_width:nn` Set macro #1 to be the line width set by the PGF style #2. Note that `\begingroup` and `\endgroup` are used here because of the definition of `\pgfmathsmuggle`.

```
78 \cs_new:Npn\__timechart_set_style_line_width:nn #1#2
79   {
80     \begingroup
81     \tikzset{#2}
82     \pgfmathsetlengthmacro{#1}{\pgflinewidth}
83     \pgfmathsmuggle #1
84     \endgroup
85   }
```

(End of definition for `\__timechart_set_style_line_width:nn`.)

## 14.7 PGF keys

All PGF keys for this package are under `/timechart/`.

```
86 \pgfkeys{
87   /timechart/.cd,
```

Keys applicable to whole chart.

```
88   width/.initial=\textwidth,
89   tolerance/.initial=5pt,
90   beyond~length/.initial=5pt,
91   beyond~x~radius/.initial=4pt,
92   ystep/.initial=-10pt,
93   minor~tick~interval/.initial=10,
94   major~tick~interval/.initial=50,
```

Keys applicable to the grid.

```
95   no~grid/.code = { \bool_set_false:N\l__timechart_grid_bool },
96   grid~top~ysep/.initial={3pt},
97   grid~bottom~ysep/.initial={3pt},
98   grid/.style={},
```

Keys applicable to the axis.

```
99   no~axis/.code = { \bool_set_false:N\l__timechart_axis_bool },
100   axis~line/.style={
101     line~cap=rect,
102   },
103   axis~ysep/.initial=3pt,
104   minor~tick/.style={},
105   minor~tick~length/.initial=1.5mm,
106   major~tick/.style={},
107   major~tick~length/.initial=3mm,
108   major~tick~label/.style={
109     inner~sep=0,
110     outer~sep=0,
111     anchor=mid~west,
112     rotate=90,
113   },
```

Keys applicable to intervals, texts, spaces, and legends.

```
114 no~autostep/.code = { \bool_set_false:Nz\l__timechart_autostep_bool },
115 ref/.initial={},
116 mark/.initial={},
117 marks/.forward~to=/timechart/mark,
118 circa~uncertainty/.initial=3,
119 interval~minimum~width/.initial=1pt,
120 interval~bar~color/.initial=black,
121 interval~bar~thickness/.initial=8pt,
122 interval~bar~node~name/.initial = {bar~node},
123 interval~mark~color/.initial=gray,
124 interval~label/.style={},
125 interval~label~centered/.style={/timechart/interval~label,text=white},
126 interval~label~centered~background/.style={/timechart/interval~label},
127 interval~label~baseline/.initial=-3pt,
128 interval~label~pos/.is~choice,
129 interval~label~pos/left/.code
130   = { \int_set:Nn\l__timechart_label_pos_int{0} },
131 interval~label~pos/center/.code
132   = { \int_set:Nn\l__timechart_label_pos_int{1} },
133 interval~label~pos/right/.code
134   = { \int_set:Nn\l__timechart_label_pos_int{2} },
135 interval~label~node~name/.initial = {label~node},
136 start~range/.is~choice,
137 start~range/fade/.code
138   = { \int_set:Nn\l__timechart_start_range_type_int{0} },
139 start~range/slant/.code
140   = { \int_set:Nn\l__timechart_start_range_type_int{1} },
141 finish~range/.is~choice,
142 finish~range/fade/.code
143   = { \int_set:Nn\l__timechart_finish_range_type_int{0} },
144 finish~range/slant/.code
145   = { \int_set:Nn\l__timechart_finish_range_type_int{1} },
146 fade~minimum~width/.initial = 0pt,
```

Keys applicable only to texts.

```
147 text~node~name/.initial = {text~node},
148 text/.style={},
149 text~baseline/.initial=-3pt,
150 text~pos/.is~choice,
151 text~pos/left/.code = { \int_set:Nn\l__timechart_text_pos_int{0} },
152 text~pos/center/.code = { \int_set:Nn\l__timechart_text_pos_int{1} },
153 text~pos/right/.code = { \int_set:Nn\l__timechart_text_pos_int{2} },
```

Keys applicable only to legends.

```
154 legend~item~width/.initial=9mm,
155 legend~item~range~width/.initial=3mm,
```

Shortcuts for positioning.

```
156 left/.code = {
157   \int_set:Nn\l__timechart_label_pos_int{0}
158   \int_set:Nn\l__timechart_text_pos_int{0}
159 },
160 center/.code = {
161   \int_set:Nn\l__timechart_label_pos_int{1}
```



```

162     \int_set:Nn\l__timechart_text_pos_int{1}
163   },
164   right/.code = {
165     \int_set:Nn\l__timechart_label_pos_int{2}
166     \int_set:Nn\l__timechart_text_pos_int{2}
167   },
168 }

```

`\l__timechart_grid_bool` Boolean indicating whether the grid will be drawn. This variable is by default true but can be set false via the `/timechart/no grid` PGF key.

```

169 \bool_new:N\l__timechart_grid_bool
170 \bool_set_true:N\l__timechart_grid_bool

```

*(End of definition for \l\_\_timechart\_grid\_bool.)*

`\l__timechart_axis_bool` Boolean indicating whether the axis will be drawn. This variable is by default true but can be set false via the `/timechart/no axis` PGF key.

```

171 \bool_new:N\l__timechart_axis_bool
172 \bool_set_true:N\l__timechart_axis_bool

```

*(End of definition for \l\_\_timechart\_axis\_bool.)*

`\l__timechart_autostep_bool` Boolean indicating whether to automatically step the  $y$  coordinate after an interval, text, or space. This variable is by default true but can be set false via the `/timechart/no autostep` PGF key.

```

173 \bool_new:N\l__timechart_autostep_bool
174 \bool_set_true:N\l__timechart_autostep_bool

```

*(End of definition for \l\_\_timechart\_autostep\_bool.)*

`\l__timechart_label_pos_int` An integer to hold the interval label position. This is set via the `/timechart/interval label pos` PGF key.

```

175 \int_new:N \l__timechart_label_pos_int

```

*(End of definition for \l\_\_timechart\_label\_pos\_int.)*

`\l__timechart_text_pos_int` An integer to hold the text position. This is set via the `/timechart/text pos` PGF key.

```

176 \int_new:N \l__timechart_text_pos_int

```

*(End of definition for \l\_\_timechart\_text\_pos\_int.)*

`\l__timechart_start_range_type_int` and `\l__timechart_finish_range_type_int` Integers to hold the type of the start/end ranges. These are set via the `/timechart/start range` and `/timechart/finish range` PGF keys.

```

177 \int_new:N \l__timechart_start_range_type_int
178 \int_new:N \l__timechart_finish_range_type_int

```

*(End of definition for \l\_\_timechart\_start\_range\_type\_int and \l\_\_timechart\_finish\_range\_type\_int.)*

## 14.8 Main environment

`timechart` The main environment.

**#1** : PGF keys to apply.  
**#2** : Start year.  
**#3** : End year.

```
179 \NewDocumentEnvironment{timechart}{0}{m m }
180 { \_timechart_main_begin:nnn{#1}{#2}{#3} }
181 { \_timechart_main_end: }
```

(End of definition for `timechart`. This function is documented on page 8.)

`\_timechart_main_begin:nnn` This command uses values specified by PGF keys to make some necessary calculations to begin the chart.

```
182 \cs_new:Npn \_timechart_main_begin:nnn #1#2#3
183 {
```

Process the supplied PGF keys and retrieve values that affect the chart as a whole.

```
184   \pgfkeys{
185     /timechart/.cd,
186     #1,
187     width/.get=\_timechart_width_pgf,
188     tolerance/.get=\_timechart_tolerance_pgf,
189     ystep/.get=\_timechart_ystep_pgf,
190     grid-top-ysep/.get=\_timechart_grid_top_ysep_pgf,
191     grid-bottom-ysep/.get=\_timechart_grid_bottom_ysep_pgf,
192     beyond-length/.get=\_timechart_beyond_length_pgf,
193     beyond-x-radius/.get=\_timechart_beyond_x_radius_pgf,
194     minor-tick-interval/.get=\_timechart_minor_tick_interval_year,
195     major-tick-interval/.get=\_timechart_major_tick_interval_year,
196   }
```

Start the TikZ picture and set up the necessary layers.

```
197   \tikzpicture
198   \pgfdeclarelayer{grid}
199   \pgfdeclarelayer{labelbg}
200   \pgfsetlayers{grid,labelbg,main}
```

Store the line width of the grid and axis, treating them as 0 pt if they are disabled.

```
201   \bool_if:NTF\_timechart_grid_bool
202   {
203     \_timechart_set_style_line_width:nn
204     {\_timechart_grid_line_width}
205     {/timechart/grid}
206   }
207   { \pgfmathsetlengthmacro{\_timechart_grid_line_width}{0} }
208   \bool_if:NTF\_timechart_axis_bool
209   {
210     \_timechart_set_style_line_width:nn
211     {\_timechart_axis_line_width}
212     {/timechart/axis~line}
213     \_timechart_set_style_line_width:nn
214     {\_timechart_major_tick_line_width}
215     {/timechart/major~tick}
216     \_timechart_set_style_line_width:nn
```

```

217         {\l__timechart_minor_tick_line_width}
218         {/timechart/minor-tick}
219     }
220     {
221         \pgfmathsetlengthmacro{\l__timechart_axis_line_width}{0}
222         \pgfmathsetlengthmacro{\l__timechart_major_tick_line_width}{0}
223         \pgfmathsetlengthmacro{\l__timechart_minor_tick_line_width}{0}
224     }

```

Store the start and finish years (ignoring circa, month, day), and then set up the conversion from years to  $x$  coordinates. `\l__timechart_x` is the  $x$ -distance corresponding to one year, and `yeartox` is the `pgfmath` function that does the conversion.

```

225     \l__timechart_parse_date:NNn\l_tmpa_bool\l__timechart_start_year{#2}
226     \l__timechart_parse_date:NNn\l_tmpa_bool\l__timechart_finish_year{#3}
227     \pgfmathsetmacro{\l__timechart_start_year}
228         {floor(\l__timechart_start_year)}
229     \pgfmathsetmacro{\l__timechart_finish_year}
230         {floor(\l__timechart_finish_year)}
231     \pgfmathsetmacro{\l__timechart_x}
232     {
233         (
234             \l__timechart_width_pgf
235             - max(
236                 \l__timechart_grid_line_width,
237                 \l__timechart_axis_line_width,
238                 \l__timechart_major_tick_line_width,
239                 \l__timechart_minor_tick_line_width
240             )
241         )/(\l__timechart_finish_year-\l__timechart_start_year)
242     }
243     \pgfkeys{
244         /pgf/declare-function={
245             yeartox(\n)=\l__timechart_x*(\n-\l__timechart_start_year);
246         },
247     }

```

Calculate the start and finish  $x$  coordinates.

```

248     \pgfmathsetmacro{\l__timechart_start_x}
249         {yeartox(\l__timechart_start_year)}
250     \pgfmathsetmacro{\l__timechart_finish_x}
251         {yeartox(\l__timechart_finish_year)}
252     \pgfmathsetmacro{\l__timechart_start_tolerance_x}{
253         \l__timechart_start_x-(\l__timechart_tolerance_pgf)
254     }
255     \pgfmathsetmacro{\l__timechart_finish_tolerance_x}{
256         \l__timechart_finish_x+(\l__timechart_tolerance_pgf)
257     }
258     \pgfmathsetmacro{\l__timechart_start_beyond_x}{
259         \l__timechart_start_x-(\l__timechart_beyond_length_pgf)
260     }
261     \pgfmathsetmacro{\l__timechart_finish_beyond_x}{
262         \l__timechart_finish_x+(\l__timechart_beyond_length_pgf)
263     }

```

Set up tracking of current  $y$  coordinate.

```

264 \pgfmathsetmacro{\l__timechart_current_y}{0}
265 \pgfmathsetmacro{\l__timechart_saved_y}{0}
266 \pgfmathsetmacro{\l__timechart_auto_reset_minimum_y}{-16000pt}
267 \pgfmathsetmacro{\l__timechart_auto_reset_maximum_y}{16000pt}

```

Calculate some years used in loops.

```

268 \pgfmathsetmacro{\l__timechart_start_plus_year}{
269   \l__timechart_start_year+\l__timechart_minor_tick_interval_year
270 }
271 \pgfmathsetmacro{\l__timechart_start_plusplus_year}{
272   \l__timechart_start_year+(2*\l__timechart_minor_tick_interval_year)
273 }
274 \pgfmathsetmacro{\l__timechart_end_minus_year}{
275   \l__timechart_finish_year-\l__timechart_minor_tick_interval_year
276 }

```

Begin a group and make available the user commands `\timechart...`. (The group will be ended by `\__timechart_main_end_user:`)

```

277 \group_begin:
278 \cs_set_eq:NN\timechartinterval\__timechart_interval_user:0mmm
279 \cs_set_eq:NN\timecharttext\__timechart_text_user:0mm
280 \cs_set_eq:NN\timechartspace\__timechart_space_user:0
281 \cs_set_eq:NN\timechartsety\__timechart_set_y_user:m
282 \cs_set_eq:NN\timechartsavey\__timechart_save_y_user:
283 \cs_set_eq:NN\timechartresety\__timechart_reset_y_user:
284 \cs_set_eq:NN\timechartsetyminimumautoreset
285   \__timechart_set_y_minimum_auto_reset_user:m
286 \cs_set_eq:NN\timechartsetymaximumautoreset
287   \__timechart_set_y_maximum_auto_reset_user:m
288 \cs_set_eq:NN\timechartstepy\__timechart_step_y_user:0
289 \cs_set_eq:NN\timechartfinish\__timechart_main_end_user:
290 }

```

(End of definition for `\__timechart_main_begin:nnn`.)

`\__timechart_main_end:` Make sure the chart is complete and end the TikZ picture. `\__timechart_main_end_user:` ends the group begun by `\__timechart_main_begin:nnn`, so whether the user has *not* called it (as `\timechartfinish`) is equivalent to it being equal to `\timechartfinish`.

```

291 \cs_new:Npn\__timechart_main_end:
292 {
293   \cs_if_eq:NNT\timechartfinish\__timechart_main_end_user:
294     { \__timechart_main_end_user: }
295   \endtikzpicture
296 }

```

(End of definition for `\__timechart_main_end:`.)

`\__timechart_main_end_user:` End the group begun by `\__timechart_main_begin:nnn`, draw the axis and grid, and set the bounding box. This macro is made available as `\timechartfinish` inside the `timechart` environment.

```

297 \cs_new:Npn\__timechart_main_end_user:
298 {

```

The aim here is to set the bounding box (1) to fit horizontally the axis *not* including labels and the grid and (2) to fit vertically the axis including labels and the grid . All the ‘horizontal’ data is already known, and the ‘vertical’ data is determined by the *current* bounding box. So extract the ‘vertical’ data and then reset the bounding box.

```

299   \pgfextracty{\l__timechart_tmpa_dim}
300   { \pgfpointanchor{current~bounding~box}{south} }
301   \pgfextracty{\l__timechart_tmpb_dim}
302   { \pgfpointanchor{current~bounding~box}{north} }
303   \pgfresetboundingbox

```

If the timechart is empty, then the extracted  $y$  coordinates of ‘north’ and ‘south’ anchors of the bounding box will be  $-16\,000$  pt and  $16\,000$  pt respectively. Test for this and treat them as both having  $y$  coordinate  $0$  pt in this case.

```

304   \dim_compare:nNnTF{\l__timechart_tmpa_dim}>{\l__timechart_tmpb_dim}
305   {
306     \pgfmathsetmacro{\l__timechart_content_bottom_y}{0pt}
307     \pgfmathsetmacro{\l__timechart_content_top_y}{0pt}
308   }
309   {
310     \pgfmathsetmacro{\l__timechart_content_bottom_y}
311     {\l__timechart_tmpa_dim}
312     \pgfmathsetmacro{\l__timechart_content_top_y}
313     {\l__timechart_tmpb_dim}
314   }

```

Now draw the grid and axis if necessary and set the bounding box if not.

```

315   \bool_if:NTF{\l__timechart_grid_bool}
316   {
317     \__timechart_grid_draw:
318     \pgfmathsetmacro{\l__timechart_axis_y}
319     {
320       \l__timechart_content_top_y
321       + \l__timechart_grid_top_ysep_pgf
322       + \pgfkeysvalueof{/timechart/axis~ysep}
323     }
324   }
325   {
326     \__timechart_nogrid_bounding_box_set:
327     \pgfmathsetmacro{\l__timechart_axis_y}
328     {
329       \l__timechart_content_top_y
330       + \pgfkeysvalueof{/timechart/axis~ysep}
331     }
332   }
333   \bool_if:NT{\l__timechart_axis_bool}
334   { \__timechart_axis_draw: }

```

Finally, end the group begun by `\__timechart_main_begin:nnn`.

```

335   \group_end:
336   }

```

(End of definition for `\__timechart_main_end_user:.`)

## 14.9 Grid drawing

`\__timechart_grid_draw:` Draw the grid of the chart, assuming that the  $y$  coordinates of the top and bottom of the content have been calculated and stored in `\l__timechart_content_top_y` and `\l__timechart_content_bottom_y`.

```

337 \cs_new:Npn\__timechart_grid_draw:
338   {
339     \pgfmathsetmacro{\l__timechart_grid_bottom_y}{
340       \l__timechart_content_bottom_y-\l__timechart_grid_bottom_ysep_pgf
341     }
342     \pgfmathsetmacro{\l__timechart_grid_top_y}{
343       \l__timechart_content_top_y+\l__timechart_grid_top_ysep_pgf
344     }
345     \pgfonlayer{ grid }
346     \scope[/timechart/grid]
347     \foreach \year in {
348       \l__timechart_start_plus_year,
349       \l__timechart_start_plusplus_year,
350       ...,
351       \l__timechart_end_minus_year
352     } {
353       \group_begin:

```

Only draw gridlines at major ticks.

```

354       \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
355       { Mod(\year,\l__timechart_major_tick_interval_year)==0 }
356       \bool_if:NT\l__timechart_tmpa_bool
357       {
358         \pgftransformshift{ \pgfpoint{yeartox(\year)}{0} }
359         \pgfpathmoveto{ \pgfpoint{0}{\l__timechart_grid_top_y} }
360         \pgfpathlineto{ \pgfpoint{0}{\l__timechart_grid_bottom_y} }
361         \pgfusepath{ draw }
362       }
363       \group_end:
364     }

```

Define and draw the grid node.

```

365     \__timechart_make_rectangle_node:nnnn
366     { \pgfpoint{\l__timechart_start_x}{\l__timechart_grid_bottom_y} }
367     { \pgfpoint{\l__timechart_finish_x}{\l__timechart_grid_top_y} }
368     { grid }
369     { \c_true_bool }
370     \endscope
371     \endpgfonlayer
372   }

```

*(End of definition for `\__timechart_grid_draw:`.)*

## 14.10 Axis drawing

`\__timechart_axis_draw:` Draw the axis, with large/small ticks and labels on appropriate years, assuming that the  $y$  coordinate has been calculated and stored in `\l__timechart_axis_y`.

```

373 \cs_new:Npn\__timechart_axis_draw:
374   {
375     \group_begin:

```

```

376 \pgfkeys{
377   /timechart/minor-tick-length/.get=\_timechart\_minor\_tick\_length\_pgf,
378   /timechart/major-tick-length/.get=\_timechart\_major\_tick\_length\_pgf,
379 }

```

Shift to the  $y$  coordinate of the axis line.

```

380 \pgftransformshift{ \pgfpoint{0}{\l\_timechart\_axis\_y} }

```

Work out the first and last years which will have a major tick (since these are marked with the era).

```

381 \pgfmathsetmacro{\_timechart\_start\_major\_tick\_year}
382 {
383   \l\_timechart\_start\_year
384   -Mod(
385     \l\_timechart\_start\_year,
386     \l\_timechart\_major\_tick\_interval\_year
387   )
388 }
389 \pgfmathsetmacro{\_timechart\_start\_major\_tick\_year}
390 {
391   ifthenelse(
392     \_timechart\_start\_major\_tick\_year<\l\_timechart\_start\_year,
393     \_timechart\_start\_major\_tick\_year
394     +\l\_timechart\_major\_tick\_interval\_year,
395     \_timechart\_start\_major\_tick\_year
396   )
397 }
398 \pgfmathsetmacro{\_timechart\_finish\_major\_tick\_year}
399 {
400   \l\_timechart\_finish\_year
401   -Mod(
402     \l\_timechart\_finish\_year,
403     \l\_timechart\_major\_tick\_interval\_year
404   )
405 }

```

Loop over years and draw the appropriate ticks.

```

406 \foreach \year in {
407   \l\_timechart\_start\_year,
408   \l\_timechart\_start\_plus\_year,
409   ...,
410   \l\_timechart\_finish\_year
411 } {
412   \pgfmathsetmacro{\x}{yeartox(\year)}
413   \_timechart\_pgfmathsetbool:nn
414     {\l\_timechart\_tmpa\_bool}
415     { Mod(\year,\l\_timechart\_major\_tick\_interval\_year)==0 }
416   \bool\_if:NTF\l\_timechart\_tmpa\_bool
417     { \_timechart\_axis\_draw\_labelled\_major\_tick:NN\x\year }
418     { \_timechart\_axis\_draw\_minor\_tick:N\x }
419 }
420 \_timechart\_draw\_axis\_line

```

Define the axis node.

```

421 \pgfextracty{\l\_timechart\_tmpa\_dim}
422 { \pgfpointanchor{current~bounding~box}{north} }

```

```

423   \__timechart_make_rectangle_node:nnnn
424   { \pgfpoint{\l__timechart_start_x}{0} }
425   { \pgfpoint{\l__timechart_finish_x}{\l__timechart_tmpa_dim} }
426   { axis }
427   { \c_false_bool }
428   \group_end:
429 }

```

(End of definition for `\__timechart_axis_draw:`)

All the remaining axis-related macros (which begin `\__timechart_axis_draw_`) assume that a transformation has been applied so that the axis line is at  $y = 0$ .

`\__timechart_axis_draw_minor_tick:N` Draw an unlabelled tick at  $x$  coordinate #1.

```

430 \cs_new:Npn\__timechart_axis_draw_minor_tick:N #1
431 {
432   \scope[/timechart/minor-tick]
433   \pgfpathmoveto{ \pgfpoint{#1}{0} }
434   \pgfpathlineto{ \pgfpoint{#1}{\__timechart_minor_tick_length_pgf} }
435   \pgfusepath{draw}
436   \endscope
437 }

```

(End of definition for `\__timechart_axis_draw_minor_tick:N`.)

`\__timechart_axis_draw_labelled_major_tick:NN` Draw a labelled major tick at  $x$  coordinate #1, with label for year #2, using the special epoch marker if the year is 0, and showing the era if and only if the year is for the first or last major tick. This macro assumes that `\__timechart_start_major_tick_year` and `\__timechart_finish_major_tick_year` have been calculated.

```

438 \cs_new:Npn\__timechart_axis_draw_labelled_major_tick:NN #1#2
439 {
440   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}{#2==0}
441   \bool_if:NTF \l__timechart_tmpa_bool
442   { \__timechart_axis_draw_zero_tick:N #1 }
443   {
444     \__timechart_pgfmathsetbool:nn{\l__timechart_tmpb_bool}
445     { #2==\__timechart_start_major_tick_year }
446     \__timechart_pgfmathsetbool:nn{\l__timechart_tmpc_bool}
447     { #2==\__timechart_finish_major_tick_year }
448     \bool_set:Nn\l__timechart_tmpd_bool
449     { \l__timechart_tmpb_bool || \l__timechart_tmpc_bool }
450     \__timechart_axis_draw_major_tick:N #1
451     \__timechart_axis_draw_year_label:nnnnnn
452     { #1 }
453     { #2 }
454     { \l__timechart_tmpd_bool }
455     { mid~west }
456     { 0 }
457     { \__timechart_major_tick_length_pgf+1mm }
458   }
459 }

```

(End of definition for `\__timechart_axis_draw_labelled_major_tick:NN`.)



```

\__timechart_axis_draw_major_tick:N Draw a major tick at  $x$  coordinate #1.
460 \cs_new:Npn\__timechart_axis_draw_major_tick:N #1
461 {
462   \scope[/timechart/major-tick]
463   \pgfpathmoveto{ \pgfpoint{#1}{0} }
464   \pgfpathlineto{ \pgfpoint{#1}{\__timechart_major_tick_length_pgf} }
465   \pgfusepath{draw}
466   \endscope
467 }

```

(End of definition for `\__timechart_axis_draw_major_tick:N`.)

```

\__timechart_axis_draw_zero_tick:N Draw a special (labelled) tick for year zero at  $x = #1$ .
468 \cs_new:Npn\__timechart_axis_draw_zero_tick:N #1
469 {
470   \group_begin:
471   \pgftransformshift{ \pgfpoint{#1}{0} }

```

The mark is a cross made up of four arcs and the miter joins between them. The parameter `\r` is the arc radius. The parameter `\a` is how many degrees should be trimmed from the start/end of a quarter-circle to form each arc. Thus the length of the miter is dependent on `\a`.

```

472   \pgfmathsetmacro{\a}{5}
473   \pgfmathsetlengthmacro{\r}{1mm}
474   \pgfmathsetlengthmacro{\t}{\r*(cos(\a)-cos(90-\a))/(1-cos(90-\a))}

```

The drawing process is: move to the start of the tick, draw the tick, draw the four arcs, then draw a small part of the tick again. The last step is not mathematically necessary for a smooth join, but ensures that the join *appears* smooth.

```

475   \scope[/timechart/major-tick,line-join=miter]
476   \pgfpathmoveto{\pgfpointorigin}
477   \pgfpathlineto{\pgfpoint{0}{\__timechart_major_tick_length_pgf}}
478   \pgfpatharc{0}{90-\a}{\t-and-\r}
479   \pgfpatharc{270+\a}{360-\a}{\r}
480   \pgfpatharc{180+\a}{270-\a}{\r}
481   \pgfpatharc{90+\a}{180-\a}{\t-and-\r}
482   \pgfpathlineto{\pgfpoint{0}{\__timechart_major_tick_length_pgf-1pt}}
483   \pgfusepath{draw}
484   \endscope

```

There is no year 0, so label the zero mark with the 1 before and 1 after the epoch.

```

485   \__timechart_axis_draw_year_label:nnnnnn
486   { 0 }
487   { -1 }
488   { \c_true_bool }
489   { base-west }
490   { -.5mm }
491   { 5.5mm }
492   \__timechart_axis_draw_year_label:nnnnnn
493   { 0 }
494   { 1 }
495   { \c_true_bool }
496   { north-west }
497   { .5mm }
498   { 5.5mm }

```

```

499     \group_end:
500   }

```

(End of definition for `\__timechart_axis_draw_zero_tick:N`.)

```

\__timechart_axis_draw_line Draw the axis line itself.
501 \cs_new:Npn\__timechart_draw_axis_line
502   {
503     \scope[/timechart/axis~line]
504     \pgfpathmoveto{ \pgfpoint{\l__timechart_start_x}{0} }
505     \pgfpathlineto{ \pgfpoint{\l__timechart_finish_x}{0} }
506     \pgfusepath{draw}
507     \endscope
508   }

```

(End of definition for `\__timechart_axis_draw_line`.)

```

\__timechart_axis_draw_year_label:nnnnnn Draw a year label.
#1 : x coordinate.
#2 : Year for label.
#3 : Boolean literal indicating whether the era should be shown.
#4 : Anchor for node.
#5 : x offset (dimension).
#6 : y offset (dimension).
509 \cs_new:Npn\__timechart_axis_draw_year_label:nnnnnn #1#2#3#4#5#6
510   {
511     \group_begin:
512     \pgftransformshift{ \pgfpoint{#1+#5}{#6} }
513     \pgfmathtruncatemacro{\absyear}{ abs(#2) }
514     \scope[/timechart/major~tick~label]
515     \bool_if:NTF #3
516     {
517       \__timechart_pgfmashsetbool:nn{\l__timechart_tmpa_bool}{#2<0}
518       \bool_if:NTF\l__timechart_tmpa_bool
519       { \cs_set_eq:NN\__timechart_make_year:n\timechartmakebeforeyear }
520       { \cs_set_eq:NN\__timechart_make_year:n\timechartmakeafteryear }
521       \__timechart_hsmash_pgfnode:nnnnn
522       {rectangle}
523       {#4}
524       {\__timechart_make_year:n{\absyear}}
525       {}
526       {}
527     }
528     { \pgfnode{rectangle}{#4}{\absyear}{}{} }
529     \endscope
530     \group_end:
531   }

```

(End of definition for `\__timechart_axis_draw_year_label:nnnnnn`.)

```

\timechartmakebeforeyear User-redefineable macros to format a year as before or after the epoch.
\timechartmakeafteryear
532 \cs_new:Npn\timechartmakebeforeyear #1
533   {
534     #1\nobreakspace\textsc{bce}

```

```

535 }
536 \cs_new:Npn\timechartmakeafteryear #1
537 {
538   #1\nobreakspace\textsc{ce}
539 }

```

(End of definition for `\timechartmakebeforeyear` and `\timechartmakeafteryear`. These functions are documented on page 13.)

## 14.11 Bounding box

`\_timechart_nogrid_bounding_box_set:` Set the bounding box when no grid is being drawn.

```

540 \cs_new:Npn\_timechart_nogrid_bounding_box_set:
541 {
542   \pgfpathmoveto
543     { \pgfpoint{\l__timechart_start_x}{\l__timechart_content_bottom_y} }
544   \pgfpathmoveto
545     { \pgfpoint{\l__timechart_finish_x}{\l__timechart_content_top_y} }
546   \pgfusepath{discard}
547 }

```

(End of definition for `\_timechart_nogrid_bounding_box_set:`.)

## 14.12 Positioning

`\_timechart_set_y_user:m` Set current  $y$  coordinate to #1. This macro will be made available as `\timechartsety` in the `timechart` environment.

```

548 \cs_new:Npn\_timechart_set_y_user:m #1
549 {
550   \pgfmathsetmacro{\l__timechart_current_y}{#1}
551 }

```

(End of definition for `\_timechart_set_y_user:m`.)

`\_timechart_save_y_user:` Save the current  $y$  coordinate. This macro will be made available as `\timechartsavey` in the `timechart` environment.

```

552 \cs_new:Npn\_timechart_save_y_user:
553 {
554   \pgfmathsetmacro{\l__timechart_saved_y}{\l__timechart_current_y}
555 }

```

(End of definition for `\_timechart_save_y_user:`.)

`\_timechart_reset_y_user:` Set the current  $y$  coordinate to the last saved coordinate. This macro will be made available as `\timechartsresety` in the `timechart` environment.

```

556 \cs_new:Npn\_timechart_reset_y_user:
557 {
558   \pgfmathsetmacro{\l__timechart_current_y}{\l__timechart_saved_y}
559 }

```

(End of definition for `\_timechart_reset_y_user:`.)

`\_timechart_set_y_minimum_auto_reset_user:m` Set a  $y$  coordinate below which `\_timechart_step_y_user:` will automatically reset the current  $y$  coordinate to the last saved  $y$  coordinate. This macro will be made available as `\timechartssetyminimumautoreset` in the `timechart` environment.

```
560 \cs_new:Npn\_timechart_set_y_minimum_auto_reset_user:m #1
561 {
562   \pgfmathsetmacro{\_timechart_auto_reset_minimum_y}{#1}
563 }
```

*(End of definition for `\_timechart_set_y_minimum_auto_reset_user:m`.)*

`\_timechart_set_y_maximum_auto_reset_user:m` Set a  $y$  coordinate above which `\_timechart_step_y_user:` will automatically reset the current  $y$  coordinate to the last saved  $y$  coordinate. This macro will be made available as `\timechartssetymaximumautoreset` in the `timechart` environment.

```
564 \cs_new:Npn\_timechart_set_y_maximum_auto_reset_user:m #1
565 {
566   \pgfmathsetmacro{\_timechart_auto_reset_maximum_y}{#1}
567 }
```

*(End of definition for `\_timechart_set_y_maximum_auto_reset_user:m`.)*

`\_timechart_step_y_user:0` Increment the current  $y$  coordinate by `#1` times the length specified in `/timechart/ystep`. This macro will be made available as `\timechartstepy` in the `timechart` environment.

```
568 \NewDocumentCommand{\_timechart_step_y_user:0}{ 0{1} }
569 {
570   \pgfmathsetmacro{\_timechart_current_y}
571     {\_timechart_current_y+#1*\_timechart_ystep_pgf}
572   \_timechart_pgfmathsetbool:nn{\_timechart_tmpa_bool}
573   {
574     or(
575       \_timechart_current_y<\_timechart_auto_reset_minimum_y,
576       \_timechart_current_y>\_timechart_auto_reset_maximum_y
577     )
578   }
579   \bool_if:nT{\_timechart_tmpa_bool}
580     { \pgfmathsetmacro{\_timechart_current_y}{\_timechart_saved_y} }
581 }
```

*(End of definition for `\_timechart_step_y_user:0`.)*

### 14.13 Bounds checking

`\_timechart_if_x_in_bounds:nT` Check if  $x$  coordinate `#1` is (strictly) within the bounds of the chart; if so, execute `#2`.

```
582 \cs_new:Npn\_timechart_if_x_in_bounds:nT #1#2
583 {
584   \_timechart_pgfmathsetbool:nn{\_timechart_tmpa_bool}{
585     and(
586       #1>=\_timechart_start_x,
587       #1<=\_timechart_finish_x
588     )
589   }
590   \bool_if:NT\_timechart_tmpa_bool
591     {#2}
592 }
```

(End of definition for `\__timechart_if_x_in_bounds:nT`.)

`\__timechart_if_x_in_tolerance_bounds_x:nT` Check if  $x$  coordinate #1 is within the specified tolerance of the bounds of the chart; if so, execute #2.

```
594 \cs_new:Npn\__timechart_if_x_in_tolerance_bounds:nT #1#2
595 {
596   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}{
597     and(
598       #1>=\l__timechart_start_tolerance_x,
599       #1<=\l__timechart_finish_tolerance_x
600     )
601   }
602   \bool_if:NT\l__timechart_tmpa_bool{#2}
603 }
```

(End of definition for `\__timechart_if_x_in_tolerance_bounds_x:nT`.)

`\if_x_range_intersect_tolerance_bounds_x:nnT` Check if the range between  $x$  coordinates #1 and #2 intersects the range of the bounds of the chart plus the specified tolerance; if so, execute #3.

```
604 \cs_new:Npn\__timechart_if_x_range_intersect_tolerance_bounds:nnT #1#2#3
605 {
606   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}{
607     or(
608       or(
609         and(
610           #2>=\l__timechart_start_tolerance_x,
611           #2<=\l__timechart_finish_tolerance_x
612         ),
613         and(
614           #1>=\l__timechart_start_tolerance_x,
615           #1<=\l__timechart_finish_tolerance_x
616         )
617       ),
618       and(
619         #1<\l__timechart_start_tolerance_x,
620         #2>\l__timechart_finish_tolerance_x
621       )
622     )
623   }
624   \bool_if:NT\l__timechart_tmpa_bool
625     {#3}
626 }
```

(End of definition for `\__timechart_if_x_range_intersect_tolerance_bounds_x:nnT`.)

## 14.14 Date and date range parsing

`\__timechart_parse_date_or_daterange:NNNNNn` Parse the text in #6, which should represent a date or date range, into parameters #1–#5.

- #1 : range indicator boolean variable.
- #2 : minimum circa indicator boolean variable.
- #3 : minimum variable.
- #4 : maximum circa indicator boolean variable.
- #5 : maximum variable.
- #6 : text to parse.

```

627 \cs_new:Npn\__timechart_parse_date_or_daterange:NNNNNn #1#2#3#4#5#6
628 {
629   \bool_set:Nn #1 {\__timechart_is_nondaterange_p:w #6/\q_stop}
630   \bool_set_inverse:N #1
631   \bool_if:NTF #1
632     { \__timechart_parse_range:w #2#3#4#5\q_mark #6\q_stop }
633     {
634       \__timechart_parse_date:NNn #2#3{#6}
635       \bool_set_eq:NN #4#2
636       \pgfmathsetmacro{#5}{#3}
637     }
638 }

```

(End of definition for \\_\_timechart\_parse\_date\_or\_daterange:NNNNNn.)

`\__timechart_is_nondaterange_p:w` To be called in the form `\__timechart_is_nondaterange_p:w<text>/\q_stop`. Return boolean true if and only if `<text>` (known to be either a date or date range) contains a range marker.

```

639 \cs_new:Npn\__timechart_is_nondaterange_p:w #1/#2\q_stop
640 {
641   \tl_if_empty_p:n{#2}
642 }

```

(End of definition for \\_\_timechart\_is\_nondaterange\_p:w.)

`\__timechart_parse_range:w` To be called in the form `\__timechart_parse_range:w<min><min><max><max>\q_mark<text>\q_stop`. Parse `<text>` (known to represent a date range) into minimum circa indicator boolean variable `<min>`, minimum variable `<min>`, maximum circa indicator boolean variable `<max>`, maximum variable `<max>`.

```

643 \cs_new:Npn\__timechart_parse_range:w #1#2#3#4\q_mark #5/#6\q_stop
644 {
645   \__timechart_parse_date:NNn #1#2{#5}
646   \__timechart_parse_date:NNn #3#4{#6}
647 }

```

(End of definition for \\_\_timechart\_parse\_range:w.)

`\__timechart_parse_date:NNn` Parse text (known to represent a date) into the supplied variables. Parameters `#1` and `#2` are the variables for (respectively) circa indicator boolean and date, and `#3` is the text to be parsed:

**#1** : circa indicator boolean variable.  
**#2** : date variable.  
**#3** : text to parse.

```

648 \cs_new:Npn\__timechart_parse_date:NNn #1#2#3
649 {
650   \bool_set:Nn #1 { \__timechart_is_circa_p:w #3c\q_stop }
651   \bool_if:NTF #1
652     { \__timechart_parse_circa_date:w #2\q_mark #3\q_stop }
653     { \__timechart_parse_noncirca_date:Nn #2{#3} }
654 }

```

(End of definition for \\_\_timechart\_parse\_date:NNn.)

`\__timechart_is_circa_p:w` To be called in the form `\__timechart_is_circa_p:w<text>c\q_stop`. Return boolean true if and only if `<text>` (known to be either a date or a date with a circa indicator) has a circa indicator.

```

655 \cs_new:Npn\__timechart_is_circa_p:w #1c#2\q_stop
656 {
657   \tl_if_empty_p:n{#1}
658 }

```

*(End of definition for \\_\_timechart\_is\_circa\_p:w.)*

`\__timechart_parse_circa_date:w` To be called in the form `\__timechart_parse_circa_date:w<var>\q_mark<text>\q_stop`. Parse `<text>` (known to represent a circa date) into the supplied variable. #1 is the variable for the date and #2 is the text to be parsed.

```

659 \cs_new:Npn\__timechart_parse_circa_date:w #1\q_mark c#2\q_stop
660 {
661   \__timechart_parse_noncirca_date:Nn #1{#2}
662 }

```

*(End of definition for \\_\_timechart\_parse\_circa\_date:w.)*

`\__timechart_parse_noncirca_date:w` To be called in the form `\__timechart_parse_noncirca_date:w<var>\q_mark<text>\q_stop`. Parse `<text>` (known to represent a non-circa date) into the supplied variable. #1 is the variable for the date and #2 is the text to be parsed.

```

663 \cs_new:Npn\__timechart_parse_noncirca_date:Nn #1#2
664 {
665   \bool_if:nTF { \__timechart_is_before_p:w #2-\q_stop }
666     { \__timechart_parse_before_date:w #1\q_mark #2\q_stop }
667     { \__timechart_parse_signed_date:w #1\q_mark #2-0-0\q_stop }
668 }

```

*(End of definition for \\_\_timechart\_parse\_noncirca\_date:w.)*

`\__timechart_is_before_p:w` To be called in the form `\__timechart_is_before_p:w<text>-\q_stop`. Return boolean true if and only if `<text>` (known to be a date without a circa indicator) begins with a `-`.

```

669 \cs_new:Npn\__timechart_is_before_p:w #1-#2\q_stop
670 {
671   \tl_if_empty_p:n{#1}
672 }

```

*(End of definition for \\_\_timechart\_is\_before\_p:w.)*

`\__timechart_parse_before_date:w` To be called in the form `\__timechart_parse_before_date:w<var>\q_mark<text>\q_stop`. Parse `<text>` (known to represent a date with a leading `-`) into the supplied variable. #1 is the variable for the date and #2 is the text to be parsed.

```

673 \cs_new:Npn\__timechart_parse_before_date:w #1\q_mark-#2\q_stop
674 {
675   \__timechart_parse_signed_date:w #1-\q_mark #2-0-0\q_stop
676 }

```

(End of definition for `\_timechart_parse_before_date:w`.)

Now comes that actual parsing of an ISO-format date YYYY-MM-DD. The following macros serve as lookup tables for the number of days in the  $n$ -th month and the number of days in the year up to the start of the  $n$ -th month.

```

677 \cs_new:cpn{c__timechart_year_days_pgf}{365}
678 \cs_new:cpn{c__timechart_month_days_1_pgf}{31}
679 \cs_new:cpn{c__timechart_month_days_2_pgf}{28}
680 \cs_new:cpn{c__timechart_month_days_3_pgf}{31}
681 \cs_new:cpn{c__timechart_month_days_4_pgf}{30}
682 \cs_new:cpn{c__timechart_month_days_5_pgf}{31}
683 \cs_new:cpn{c__timechart_month_days_6_pgf}{30}
684 \cs_new:cpn{c__timechart_month_days_7_pgf}{31}
685 \cs_new:cpn{c__timechart_month_days_8_pgf}{31}
686 \cs_new:cpn{c__timechart_month_days_9_pgf}{30}
687 \cs_new:cpn{c__timechart_month_days_10_pgf}{31}
688 \cs_new:cpn{c__timechart_month_days_11_pgf}{30}
689 \cs_new:cpn{c__timechart_month_days_12_pgf}{31}
690 \cs_new:cpn{c__timechart_cumulative_days_1_pgf}{0}
691 \cs_new:cpn{c__timechart_cumulative_days_2_pgf}{31}
692 \cs_new:cpn{c__timechart_cumulative_days_3_pgf}{59}
693 \cs_new:cpn{c__timechart_cumulative_days_4_pgf}{90}
694 \cs_new:cpn{c__timechart_cumulative_days_5_pgf}{120}
695 \cs_new:cpn{c__timechart_cumulative_days_6_pgf}{151}
696 \cs_new:cpn{c__timechart_cumulative_days_7_pgf}{181}
697 \cs_new:cpn{c__timechart_cumulative_days_8_pgf}{212}
698 \cs_new:cpn{c__timechart_cumulative_days_9_pgf}{243}
699 \cs_new:cpn{c__timechart_cumulative_days_10_pgf}{273}
700 \cs_new:cpn{c__timechart_cumulative_days_11_pgf}{304}
701 \cs_new:cpn{c__timechart_cumulative_days_12_pgf}{334}

```

`\_timechart_parse_signed_date:w` To be called in the form `\_timechart_parse_positive_date:w⟨var⟩⟨sign⟩\q_mark⟨text⟩-0-0\q_stop`. Parse `⟨text⟩` (known to represent a non-circa date) into the supplied variable. #1 is the variable for the date and #2 is possibly `-`.

There is a trick in the parsing:

1. If `⟨text⟩` has the form `⟨year⟩-⟨month⟩-⟨day⟩`, then parameters #3, #4, and #5 will be, respectively, `⟨year⟩`, `⟨month⟩`, and `⟨day⟩-0-0`. Thus #5 will be evaluated by `pgfmath` to `⟨day⟩`.
2. If `⟨text⟩` has the form `⟨year⟩-⟨month⟩`, then parameters #3, #4, and #5 will be, respectively, `⟨year⟩`, `⟨month⟩`, and `0-0`. Thus #5 will be evaluated by `pgfmath` to `0`.
3. If `⟨text⟩` is simply `⟨year⟩`, then parameters #3, #4, and #5 will be, respectively, `⟨year⟩`, `0`, and `0`.

```

702 \cs_new:Npn\_timechart_parse_signed_date:w #1#2\q_mark #3-#4-#5\q_stop
703 {
704   \pgfmathtruncatemacro{\_timechart_parsed_year_pgf}{#2#3}
705   \pgfmathtruncatemacro{\_timechart_parsed_month_pgf}{#4}
706   \pgfmathtruncatemacro{\_timechart_parsed_day_pgf}{#5}
707   \_timechart_pgfmthsetbool:nn{\_tmpa_bool}{
708     or(
709       \_timechart_parsed_month_pgf < 1,
710       \_timechart_parsed_month_pgf > 12,

```



```

711     )
712   }
713   \bool_if:NTF\l_tmpa_bool
714   {

```

*Case: no valid month is given.* Use only the year.

```

715     \pgfmathsetmacro{#1}{#2#3}
716   }
717   {

```

*Case: a valid month is given.* Get the number of days in the year, in the month, and in the year up to the month. Then check if the year is a leap year and, if so, make the appropriate adjustments.

```

718     \cs_set_eq:NN\l__timechart_year_days_pgf\c__timechart_year_days_pgf
719     \cs_set_eq:Nc\l__timechart_month_days_pgf
720     { c__timechart_month_days_\__timechart_parsed_month_pgf _pgf }
721     \cs_set_eq:Nc\l__timechart_cumulative_days_pgf
722     { c__timechart_cumulative_days_\__timechart_parsed_month_pgf _pgf }
723     \__timechart_pgfmathsetbool:nn{\l_tmpa_bool}{
724       or(
725         Mod(\__timechart_parsed_year_pgf,400) == 0,
726         and(
727           Mod(\__timechart_parsed_year_pgf,4) == 0,
728           Mod(\__timechart_parsed_year_pgf,100) != 0
729         )
730       )
731     }
732     \bool_if:NT\l_tmpa_bool
733     {
734       \pgfmathsetmacro{\l__timechart_year_days_pgf}
735       { \l__timechart_year_days_pgf+1 }
736       \__timechart_pgfmathsetbool:nn{\l_tmpb_bool}
737       { \__timechart_parsed_month == 1 }
738       \bool_if:NF\l_tmpb_bool
739       {
740         \__timechart_pgfmathsetbool:nn{\l_tmpb_bool}
741         { \__timechart_parsed_month == 2 }
742         \bool_if:NF\l_tmpb_bool
743         {
744           \pgfmathsetmacro{\l__timechart_month_days_pgf}
745           { \l__timechart_month_days_pgf + 1 }
746         }
747         {
748           \pgfmathsetmacro{\l__timechart_cumulative_days_pgf}
749           { \l__timechart_cumulative_days_pgf + 1 }
750         }
751       }
752     }
753     \__timechart_pgfmathsetbool:nn{\l_tmpa_bool}{
754       or(
755         \__timechart_parsed_day_pgf < 1,
756         \__timechart_parsed_day_pgf > \l__timechart_month_days_pgf,
757       )
758     }
759     \bool_if:NTF\l_tmpa_bool

```

```

760     {
Sub-case: no valid day is given. Use only the year and month.
761     \pgfmathsetmacro{#1}
762     {
763         #2#3
764         + \l__timechart_cumulative_days_pgf/\l__timechart_year_days_pgf
765     }
766 }
767 {
Sub-case: a valid day is given. Use the year, month, and day.
768     \pgfmathsetmacro{#1}
769     {
770         #2#3
771         + (
772             \l__timechart_cumulative_days_pgf
773             + \__timechart_parsed_day_pgf
774             )/\l__timechart_year_days_pgf
775     }
776 }
777 }
778 }

```

(End of definition for \\_\_timechart\_parse\_signed\_date:w.)

## 14.15 Interval drawing

### 14.15.1 Preliminaries

These boolean variables will be used to hold parsed data for the start and finish of an interval: whether it is a range, whether the beginning of that range is qualified by ‘circa’, and whether the end of that range is qualified by ‘circa’.

```

\l__timechart_start_is_range_bool
\l__timechart_startmin_circa_bool
\l__timechart_startmax_circa_bool
\l__timechart_finish_is_range_bool
\l__timechart_finishmin_circa_bool
\l__timechart_finishmax_circa_bool

```

```

779 \bool_new:N\l__timechart_start_is_range_bool
780 \bool_new:N\l__timechart_startmin_circa_bool
781 \bool_new:N\l__timechart_startmax_circa_bool
782 \bool_new:N\l__timechart_finish_is_range_bool
783 \bool_new:N\l__timechart_finishmin_circa_bool
784 \bool_new:N\l__timechart_finishmax_circa_bool

```

(End of definition for \l\_\_timechart\_start\_is\_range\_bool and others.)

### 14.15.2 Error message definition

```

785 \msg_new:nnn{timechart}{interval_dates_invalid}
786 { Invalid~interval~dates:~#1~to~#2 }

```

### 14.15.3 Main macros

```

\__timechart_interval_user:0mmm

```

Draw an interval. This macro will be made available as `\timechartinterval` inside the `timechart` environment.

**#1** : PGF keys under `/timechart/` to apply.  
**#2** : Start year.  
**#3** : Finish year.  
**#4** : Label.

```

787 \NewDocumentCommand{\__timechart_interval_user:Ommm}{ O{} m m m }
788 {
789   \group_begin:

```

Parse the start and finish dates or date ranges.

```

790   \__timechart_parse_date_or_daterange:NNNNNn
791   \l__timechart_start_is_range_bool
792   \l__timechart_startmin_circa_bool\l__timechart_startmin_year
793   \l__timechart_startmax_circa_bool\l__timechart_startmax_year
794   {#2}
795   \__timechart_parse_date_or_daterange:NNNNNn
796   \l__timechart_finish_is_range_bool
797   \l__timechart_finishmin_circa_bool\l__timechart_finishmin_year
798   \l__timechart_finishmax_circa_bool\l__timechart_finishmax_year
799   {#3}

```

Check the results of parsing and only proceed if they are valid.

```

800   \__timechart_pgfmathsetbool:nn{\l_tmpa_bool}{
801     and(
802       \l__timechart_startmin_year <= \l__timechart_startmax_year,
803       and(
804         \l__timechart_startmax_year <= \l__timechart_finishmin_year,
805         \l__timechart_finishmin_year <= \l__timechart_finishmax_year
806       )
807     )
808   }
809   \bool_if:NTF\l_tmpa_bool
810   {
811     \__timechart_interval_checked:nn{#1}{#4}
812     \group_end:
813     \bool_if:NT\l__timechart_autostep_bool
814     { \__timechart_step_y_user:0 }
815   }
816   {
817     \msg_error:nnnn{timechart}{interval_dates_invalid}{#2}{#3}
818     \group_end:
819   }
820 }

```

*(End of definition for \\_\_timechart\_interval\_user:Ommm.)*

`\__timechart_interval_checked:nn` Draw an interval using the parsed and checked years or year ranges.

**#1** : PGF keys under `/timechart/` to apply.  
**#2** : Label.

```

821 \cs_new:Npn \__timechart_interval_checked:nn #1#2
822 {

```

Process keys supplied locally and retrieve the only value needed at this stage.

```

823   \pgfqkeys{/timechart}{
824     #1,
825     circa~uncertainty/.get=\l__timechart_circa_uncertainty_year
826   }

```

Do the minimum amount of calculation necessary to check whether any part of interval is visible.

```

827 \bool_if:NTF\l__timechart_startmin_circa_bool
828 {
829   \pgfmathsetmacro{\l__timechart_start_extreme_x}
830   { yeartox(\l__timechart_startmin_year
831     - \l__timechart_circa_uncertainty_year) }
832 }
833 {
834   \pgfmathsetmacro{\l__timechart_start_extreme_x}
835   { yeartox(\l__timechart_startmin_year) }
836 }
837 \bool_if:NTF\l__timechart_finishmax_circa_bool
838 {
839   \pgfmathsetmacro{\l__timechart_finish_extreme_x}
840   { yeartox(\l__timechart_finishmax_year
841     + \l__timechart_circa_uncertainty_year) }
842 }
843 {
844   \pgfmathsetmacro{\l__timechart_finish_extreme_x}
845   { yeartox(\l__timechart_finishmax_year) }
846 }

```

Draw the interval if some part of it is visible.

```

847 \__timechart_if_x_range_intersect_tolerance_bounds:nnT
848   {\l__timechart_start_extreme_x}{\l__timechart_finish_extreme_x}
849   { \__timechart_draw_visible_interval:nn{#1}{#2} }
850 }

```

*(End of definition for \\_\_timechart\_interval\_checked:nn.)*

`\__timechart_draw_visible_interval:nn` Draw an interval of which some part is known to be visible.

**#1** : PGF keys

**#2** : Label

```

851 \cs_new:Npn\__timechart_draw_visible_interval:nn #1#2
852 {

```

Retrieve PGF key values.

```

853 \pgfqkeys{/timechart}{
854   ref/.get=\l__timechart_ref_text,
855   mark/.get=\l__timechart_mark_text,
856   interval~minimum~width/.get=\l__timechart_minimum_width_pgf,
857   interval~bar~color/.get=\l__timechart_bar_color,
858   interval~bar~thickness/.get=\l__timechart_bar_thickness_pgf,
859   interval~mark~color/.get=\l__timechart_mark_color,
860   interval~label~baseline/.get=\l__timechart_text_baseline_pgf,
861   interval~label~node~name/.get=\l__timechart_interval_label_node_name,
862 }

```

Do the remaining calculations.

```

863 \pgfmathsetlengthmacro{\l__timechart_bar_half_thickness_pgf}
864   {.5*\l__timechart_bar_thickness_pgf}
865 \cs_set:Npn\l__timechart_label_text{#2}
866 \bool_if:NTF\l__timechart_startmax_circa_bool
867 {

```

```

868     \pgfmathsetmacro{\l__timechart_start_definite_x}
869     { yeartox(\l__timechart_startmax_year
870       + \l__timechart_circa_uncertainty_year) }
871   }
872   {
873     \pgfmathsetmacro{\l__timechart_start_definite_x}
874     { yeartox(\l__timechart_startmax_year) }
875   }
876   \bool_if:NTF\l__timechart_finishmin_circa_bool
877   {
878     \pgfmathsetmacro{\l__timechart_finish_definite_x}
879     { yeartox(\l__timechart_finishmin_year
880       - \l__timechart_circa_uncertainty_year) }
881   }
882   {
883     \pgfmathsetmacro{\l__timechart_finish_definite_x}
884     { yeartox(\l__timechart_finishmin_year) }
885   }

```

It is possible that circa indicators have made `\l__timechart_start_definite_x` greater than `\l__timechart_finish_definite_x`. Check for this; if so, set them both to their average.

```

886   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
887   {
888     \l__timechart_start_definite_x > \l__timechart_finish_definite_x
889   }
890   \bool_if:NT\l__timechart_tmpa_bool
891   {
892     \pgfmathsetmacro{\l__timechart_start_definite_x}
893     {
894       .5*(
895         \l__timechart_start_definite_x
896         + \l__timechart_finish_definite_x
897       )
898     }
899     \pgfmathsetmacro{\l__timechart_finish_definite_x}
900     { \l__timechart_start_definite_x }
901   }

```

Calculate whether it is necessary to make an adjustment to ensure that the minimum width requirement is satisfied, and store in `\l__timechart_tmpa_bool`.

```

902   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
903   {
904     (\l__timechart_finish_extreme_x-\l__timechart_start_extreme_x)
905     < \l__timechart_minimum_width_pgf
906   }

```

Calculate whether it is necessary to make an adjustment to prevent a rendering glitch when there are two fadings and a zero-width (or very small) definite part, and store in `\l__timechart_tmpb_bool`. Note the quick test for both ranges being fadings.

```

907   \int_if_zero:nTF
908   {
909     \l__timechart_start_range_type_int
910     + \l__timechart_finish_range_type_int
911   }

```

```

912     {
913       \__timechart_pgfmathsetbool:nn{\l__timechart_tmpb_bool}
914       {
915         (
916           \l__timechart_finish_definite_x
917           - \l__timechart_start_definite_x
918         )
919         < \l__timechart_minimum_width_pgf
920       }
921     }
922     { \bool_set_false:N\l__timechart_tmpb_bool }

```

Make the adjustment if necessary.

```

923     \bool_if:nT{
924       \l__timechart_tmpa_bool || \l__timechart_tmpb_bool
925     }{
926       \pgfmathsetmacro{\l__timechart_width_adjust}
927       { .5*\l__timechart_minimum_width_pgf }
928       \pgfmathsetmacro{\l__timechart_start_definite_x}
929       { \l__timechart_start_definite_x-\l__timechart_width_adjust }
930       \pgfmathsetmacro{\l__timechart_finish_definite_x}
931       { \l__timechart_finish_definite_x+\l__timechart_width_adjust }
932       \pgfmathsetmacro{\l__timechart_start_extreme_x}
933       {
934         min(
935           \l__timechart_start_extreme_x,
936           \l__timechart_start_definite_x
937         )
938       }
939       \pgfmathsetmacro{\l__timechart_finish_extreme_x}
940       {
941         max(
942           \l__timechart_finish_extreme_x,
943           \l__timechart_finish_definite_x
944         )
945       }
946     }
947     \pgfmathsetmacro{\l__timechart_start_solid_x}
948     { \l__timechart_start_definite_x }
949     \pgfmathsetmacro{\l__timechart_finish_solid_x}
950     { \l__timechart_finish_definite_x }

```

All the data needed to draw the interval are now ready. Shift to the correct vertical coordinate and open a scope for drawing.

```

951     \pgftransformshift{ \pgfpoint{0}{\l__timechart_current_y} }
952     \pgfscope

```

First, do the necessary clipping if the interval extends beyond the specified tolerance from the chart, then process the ranges, then draw the solid part of the bar, then do the labelling, then define the node for the bar. Note that ranges with the ‘slant’ style function by adding a new clipping path and extending the solid part, so that there is only *one* solid rectangle drawn.

```

953     \__timechart_interval_beyond_clip:
954     \int_case:nn {\l__timechart_start_range_type_int}
955     {

```

```

956         {0}{ \__timechart_interval_start_range_fade: }
957         {1}{ \__timechart_interval_start_range_slant: }
958     }
959 \int_case:nn {\l__timechart_finish_range_type_int}
960 {
961     {0}{ \__timechart_interval_finish_range_fade: }
962     {1}{ \__timechart_interval_finish_range_slant: }
963 }
964 \__timechart_interval_draw_solid:
965 \__timechart_interval_mark:
966 \__timechart_interval_define_bar_node:
967 \endpgfscope
968 \__timechart_interval_label:
969 }

```

(End of definition for `\__timechart_draw_visible_interval:nn`.)

#### 14.15.4 Clipping

`\__timechart_interval_beyond_clip:` If the interval bar extends beyond the specified tolerance from the chart, clip it appropriately. To avoid repeated computation, use `\l__timechart_tmpa_bool` and `\l__timechart_tmpb_bool` to store whether it is necessary to clip on the left and right (respectively). If both are false, there is no need for any clipping.

```

970 \cs_new:Npn\__timechart_interval_beyond_clip:
971 {
972     \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
973     {\l__timechart_start_extreme_x<=\l__timechart_start_tolerance_x}
974     \__timechart_pgfmathsetbool:nn{\l__timechart_tmpb_bool}
975     {\l__timechart_finish_extreme_x>=\l__timechart_finish_tolerance_x}
976     \bool_if:nT{ \l__timechart_tmpa_bool || \l__timechart_tmpb_bool }
977     { \__timechart_interval_beyond_clip_aux: }
978 }

```

(End of definition for `\__timechart_interval_beyond_clip:.`)

`\__timechart_interval_beyond_clip_aux:` This macro does the actual clipping. The clipping path starts at the north-east and proceeds clockwise.

```

979 \cs_new:Npn\__timechart_interval_beyond_clip_aux:
980 {
981     \pgfinterruptboundingbox
982     \bool_if:NTF\l__timechart_tmpb_bool
983     {
984         \pgfpathmoveto{
985             \pgfpoint
986             {\l__timechart_finish_beyond_x}
987             {\l__timechart_bar_thickness_pgf}
988         }
989         \pgfpathlineto{
990             \pgfpoint
991             {\l__timechart_finish_beyond_x}
992             {\l__timechart_bar_half_thickness_pgf}
993         }
994         \pgfpatharc
995         {90}

```

```

996         {270}
997         {\l__timechart_beyond_x_radius_pgf
998          ~and~\l__timechart_bar_half_thickness_pgf}
999         \pgfpathlineto{
1000         \pgfpoint
1001         {\l__timechart_finish_beyond_x}
1002         {-\l__timechart_bar_thickness_pgf}
1003         }
1004     }
1005     {
1006         \pgfpathmoveto{
1007         \pgfpoint
1008         {\l__timechart_right_nonclip_x}
1009         {\l__timechart_bar_thickness_pgf}
1010         }
1011         \pgfpathlineto{
1012         \pgfpoint
1013         {\l__timechart_right_nonclip_x}
1014         {-\l__timechart_bar_thickness_pgf}
1015         }
1016     }
1017     \bool_if:NTF\l__timechart_tmpa_bool
1018     {
1019         \pgfpathlineto{
1020         \pgfpoint
1021         {\l__timechart_start_beyond_x}
1022         {-\l__timechart_bar_thickness_pgf}
1023         }
1024         \pgfpathlineto{
1025         \pgfpoint
1026         {\l__timechart_start_beyond_x}
1027         {-\l__timechart_bar_half_thickness_pgf}
1028         }
1029         \pgfpatharc
1030         {-90}
1031         {90}
1032         {\l__timechart_beyond_x_radius_pgf
1033          ~and~\l__timechart_bar_half_thickness_pgf}
1034         \pgfpathlineto{
1035         \pgfpoint
1036         {\l__timechart_start_beyond_x}
1037         {\l__timechart_bar_thickness_pgf}
1038         }
1039     }
1040     {
1041         \pgfpathlineto{
1042         \pgfpoint
1043         {\l__timechart_left_nonclip_x}
1044         {-\l__timechart_bar_thickness_pgf}
1045         }
1046         \pgfpathlineto{
1047         \pgfpoint
1048         {\l__timechart_left_nonclip_x}
1049         {\l__timechart_bar_thickness_pgf}

```



```

1050     }
1051   }
1052   \pgfpathclose
1053   \pgfusepath{clip}
1054   \endpgfinterruptboundingbox
1055 }

```

(End of definition for `\__timechart_interval_beyond_clip_aux:.`)

### 14.15.5 Fading in/out

Set up the PGF ‘in’ fading.

```

1056 \pgfdeclarehorizontalshading{\__timechart_fade_in_shading}{4bp}{
1057   color(0bp)=(transparent!100);
1058   color(1bp)=(transparent!100);
1059   color(3bp)=(transparent!0);
1060   color(4bp)=(transparent!0)
1061 }
1062 \pgfdeclarefading
1063   {\__timechart_fade_in}
1064   {\pgfuseshading{\__timechart_fade_in_shading}}

```

`\__timechart_interval_start_range_fade:` Draw fading to indicate a start range. Because some PDF renderers may otherwise leave a gap between the fade and solid part, compute `\l__timechart_fade_extra_pgf` and draw this amount of overlap with the solid part. Because of the definition of the ‘in’ fading, `\l__timechart_fade_extra_pgf` must not exceed half of the fade width. And the overlap should not exceed the actual length of the solid part.

```

1065 \cs_new:Npn\__timechart_interval_start_range_fade:
1066   {
1067     \pgfmathsetlengthmacro{\l__timechart_fade_width_pgf}
1068       { \l__timechart_start_solid_x-\l__timechart_start_extreme_x }
1069     \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
1070       { \l__timechart_fade_width_pgf
1071         > \pgfkeysvalueof{/timechart/fade~minimum~width} }
1072     \bool_if:NT \l__timechart_tmpa_bool
1073       {
1074         \__timechart_if_x_range_intersect_tolerance_bounds:nnT
1075           {\l__timechart_start_extreme_x}{\l__timechart_start_solid_x}
1076           {
1077             \pgfmathsetmacro{\l__timechart_fade_extra_pgf}
1078               {
1079                 min(
1080                   .49*\l__timechart_fade_width_pgf,
1081                   \l__timechart_finish_solid_x-\l__timechart_start_solid_x
1082                 )
1083               }
1084             \pgfscope
1085             \pgfpathrectanglecorners{
1086               \pgfpoint
1087                 {\l__timechart_start_extreme_x}
1088                 {-\l__timechart_bar_half_thickness_pgf}
1089             }{
1090               \pgfpoint
1091                 {\l__timechart_start_solid_x+\l__timechart_fade_extra_pgf}

```

```

1092         {\l__timechart_bar_half_thickness_pgf}
1093     }
1094     \pgfgettransform{\__timechart_transform_current}
1095     \pgfsetfading{\__timechart_fade_in}{
1096         \pgfsettransform{\__timechart_transform_current}
1097         \pgftransformshift{
1098             \pgfpoint
1099             {
1100                 .5*\l__timechart_start_extreme_x
1101                 +.5*\l__timechart_start_solid_x
1102             }
1103             {0}
1104         }
1105         \pgftransformxscale{
1106             (\l__timechart_start_solid_x-\l__timechart_start_extreme_x)
1107             /2bp
1108         }
1109         \pgftransformyscale{\l__timechart_bar_thickness_pgf/4bp}
1110     }
1111     \pgfsetfillcolor{\l__timechart_bar_color}
1112     \pgfusepath{fill}
1113     \endpgfscope
1114 }
1115 }
1116 }

```

(End of definition for `\__timechart_interval_start_range_fade:`)

Set up the PGF 'out' fading.

```

1117 \pgfdeclarehorizontalshading{\__timechart_fade_out_shading}{4bp}{
1118     color(0bp)=(transparent!0);
1119     color(1bp)=(transparent!0);
1120     color(3bp)=(transparent!100);
1121     color(4bp)=(transparent!100)
1122 }
1123 \pgfdeclarefading
1124     {\__timechart_fade_out}
1125     {\pgfuses shading{\__timechart_fade_out_shading}}

```

`\__timechart_interval_finish_range_fade:` Draw fading to indicate a finish range. As in `\__timechart_interval_start_range_fade:`, a small overlap is computed.

```

1126 \cs_new:Npn\__timechart_interval_finish_range_fade:
1127     {
1128     \pgfmathsetlengthmacro{\l__timechart_fade_width_pgf}
1129     { \l__timechart_finish_extreme_x-\l__timechart_finish_solid_x }
1130     \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
1131     { \l__timechart_fade_width_pgf
1132     > \pgfkeysvalueof{/timechart/fade~minimum~width} }
1133     \bool_if:NT \l__timechart_tmpa_bool
1134     {
1135     \__timechart_if_x_range_intersect_tolerance_bounds:nnT
1136     {\l__timechart_finish_solid_x}{\l__timechart_finish_extreme_x}
1137     {
1138     \pgfmathsetmacro{\l__timechart_fade_extra_pgf}
1139     {

```

```

1140         min(
1141             .49*\l__timechart_fade_width_pgf,
1142             \l__timechart_finish_solid_x-\l__timechart_start_solid_x
1143         )
1144     }
1145     \pgfscope
1146     \pgfpathrectanglecorners{
1147         \pgfpoint
1148             {\l__timechart_finish_solid_x-\l__timechart_fade_extra_pgf}
1149             {-\l__timechart_bar_half_thickness_pgf}
1150     }{
1151         \pgfpoint
1152             {\l__timechart_finish_extreme_x}
1153             {\l__timechart_bar_half_thickness_pgf}
1154     }
1155     \pgfgettransform{\__timechart_transform_current}
1156     \pgfsetfading{\__timechart_fade_out}{
1157         \pgfsettransform{\__timechart_transform_current}
1158         \pgftransformshift{
1159             \pgfpoint{
1160                 .5*\l__timechart_finish_solid_x
1161                 +.5*\l__timechart_finish_extreme_x
1162             }
1163             {0}
1164         }
1165         \pgftransformxscale{
1166             (\l__timechart_finish_extreme_x-\l__timechart_finish_solid_x)
1167             /2bp
1168         }
1169         \pgftransformyscale{\l__timechart_bar_thickness_pgf/4bp}
1170     }
1171     \pgfsetfillcolor{\l__timechart_bar_color}
1172     \pgfusepath{fill}
1173     \endpgfscope
1174 }
1175 }
1176 }

```

(End of definition for \\_\_timechart\_interval\_finish\_range\_fade:.)

#### 14.15.6 Slant in/out

\\_\_timechart\_interval\_start\_range\_slant: Clip and modify \l\_\_timechart\_start\_solid\_x to draw a slanting shape to indicate the start range.

```

1177 \cs_new:Npn\__timechart_interval_start_range_slant:
1178 {
1179     \__timechart_if_equal:nnF
1180         {\l__timechart_start_extreme_x}{\l__timechart_start_solid_x}
1181         {
1182             \__timechart_if_x_range_intersect_tolerance_bounds:nnT
1183                 {\l__timechart_start_extreme_x}{\l__timechart_start_solid_x}
1184                 {
1185                     \pgfinterruptboundingbox
1186                     \pgfpathmoveto{

```

```

1187         \pgfpoint
1188             {\l__timechart_start_extreme_x}
1189             {-\l__timechart_bar_half_thickness_pgf}
1190     }
1191     \pgfpathlineto{
1192         \pgfpoint
1193             {\l__timechart_start_solid_x}
1194             {\l__timechart_bar_half_thickness_pgf}
1195     }
1196     \pgfpathlineto{
1197         \pgfpoint
1198             {\l__timechart_start_solid_x}
1199             {\l__timechart_bar_thickness_pgf}
1200     }
1201     \pgfpathlineto{
1202         \pgfpoint
1203             {\l__timechart_right_nonclip_x}
1204             {\l__timechart_bar_thickness_pgf}
1205     }
1206     \pgfpathlineto{
1207         \pgfpoint
1208             {\l__timechart_right_nonclip_x}
1209             {-\l__timechart_bar_thickness_pgf}
1210     }
1211     \pgfpathlineto{
1212         \pgfpoint
1213             {\l__timechart_start_extreme_x}
1214             {-\l__timechart_bar_thickness_pgf}
1215     }
1216     \pgfpathclose
1217     \pgfusepath{clip}
1218     \endpgfinterruptboundingbox
1219     \pgfmathsetmacro{\l__timechart_start_solid_x}
1220         {\l__timechart_start_extreme_x}
1221     }
1222 }
1223 }

```

(End of definition for `\__timechart_interval_start_range_slant:`)

`\__timechart_interval_finish_range_slant:` Clip and modify `\l__timechart_finish_solid_x` to draw a slanting shape to indicate the finish range.

```

1224 \cs_new:Npn\__timechart_interval_finish_range_slant:
1225 {
1226     \__timechart_if_equal:nnF
1227         {\l__timechart_finish_solid_x}{\l__timechart_finish_extreme_x}
1228     {
1229         \__timechart_if_x_range_intersect_tolerance_bounds:nnT
1230             {\l__timechart_finish_solid_x}{\l__timechart_finish_extreme_x}
1231         {
1232             \pgfinterruptboundingbox
1233             \pgfpathmoveto{
1234                 \pgfpoint
1235                 {\l__timechart_finish_solid_x}

```

```

1236         {-\l__timechart_bar_half_thickness_pgf}
1237     }
1238     \pgfpathlineto{
1239         \pgfpoint
1240         {\l__timechart_finish_extreme_x}
1241         {\l__timechart_bar_half_thickness_pgf}
1242     }
1243     \pgfpathlineto{
1244         \pgfpoint
1245         {\l__timechart_finish_extreme_x}
1246         {\l__timechart_bar_thickness_pgf}
1247     }
1248     \pgfpathlineto{
1249         \pgfpoint
1250         {\l__timechart_left_nonclip_x}
1251         {\l__timechart_bar_thickness_pgf}
1252     }
1253     \pgfpathlineto{
1254         \pgfpoint
1255         {\l__timechart_left_nonclip_x}
1256         {-\l__timechart_bar_thickness_pgf}
1257     }
1258     \pgfpathlineto{
1259         \pgfpoint
1260         {\l__timechart_finish_solid_x}
1261         {-\l__timechart_bar_thickness_pgf}
1262     }
1263     \pgfpathclose
1264     \pgfusepath{clip}
1265     \endpgfinterruptboundingbox
1266     \pgfmathsetmacro{\l__timechart_finish_solid_x}
1267         {\l__timechart_finish_extreme_x}
1268     }
1269 }
1270 }

```

*(End of definition for \\_\_timechart\_interval\_finish\_range\_slant:.)*

#### 14.15.7 Solid bar

`\__timechart_interval_draw_solid:` Draw the solid part of an interval.

```

1271 \cs_new:Npn\__timechart_interval_draw_solid:
1272 {
1273     \__timechart_if_x_range_intersect_tolerance_bounds:nnT
1274     {\l__timechart_start_solid_x}
1275     {\l__timechart_finish_solid_x}
1276     {
1277         \pgfpathrectanglecorners{
1278             \pgfpoint
1279             {\l__timechart_start_solid_x}
1280             {-\l__timechart_bar_half_thickness_pgf}
1281         }{
1282             \pgfpoint
1283             {\l__timechart_finish_solid_x}

```

```

1284         {\l__timechart_bar_half_thickness_pgf}
1285     }
1286     \pgfsetfillcolor{\l__timechart_bar_color}
1287     \pgfusepath{fill}
1288 }
1289 }

```

(End of definition for `\__timechart_interval_draw_solid:`)

#### 14.15.8 Marks

```

1290 \msg_new:nnn{timechart}{interval_mark_outside}
1291 { Attempt~to~mark~outside~interval~at~date~#1 }

```

`\__timechart_interval_mark:` Draw marks on an interval.

```

1292 \cs_new:Npn\__timechart_interval_mark:
1293 {
1294     \pgfscope
1295     \pgfsetstrokecolor{\l__timechart_mark_color}
1296     \foreach \year in \l__timechart_mark_text
1297     {
1298         \pgfmathsetmacro{\l__timechart_mark_x}{yeartox(\year)}
1299         \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
1300         {
1301             or (
1302                 \l__timechart_mark_x < \l__timechart_start_extreme_x,
1303                 \l__timechart_mark_x > \l__timechart_finish_extreme_x
1304             )
1305         }
1306         \bool_if:NT\l__timechart_tmpa_bool
1307         { \msg_error:nne{timechart}{interval_mark_outside}{\year} }
1308         \pgfpathmoveto{
1309             \pgfpoint
1310             {\l__timechart_mark_x}
1311             {-\l__timechart_bar_half_thickness_pgf}
1312         }
1313         \pgfpathlineto{
1314             \pgfpoint
1315             {\l__timechart_mark_x}
1316             {\l__timechart_bar_half_thickness_pgf}
1317         }
1318         \pgfusepath{draw}
1319     }
1320 \endpgfscope
1321 }

```

(End of definition for `\__timechart_interval_mark:`)

#### 14.15.9 Bar node

`\__timechart_interval_define_bar_node:` Define a node with into which the bar fits exactly.

```

1322 \cs_new:Npn\__timechart_interval_define_bar_node:
1323 {
1324     \__timechart_make_rectangle_node:nnnn
1325     {

```

```

1326     \pgfpoint
1327     {\l__timechart_start_extreme_x}
1328     {-\l__timechart_bar_half_thickness_pgf}
1329   }{
1330     \pgfpoint
1331     {\l__timechart_finish_extreme_x}
1332     {\l__timechart_bar_half_thickness_pgf}
1333   }
1334   {\pgfkeysvalueof{/timechart/interval-bar-node-name}}
1335   {\c_false_bool}
1336 }

```

(End of definition for `\__timechart_interval_define_bar_node:.`)

#### 14.15.10 Label

`\__timechart_interval_label:` Place the label for the item.

```

1337 \cs_new:Npn\__timechart_interval_label:
1338 {
1339   \str_if_empty:NF \l__timechart_label_text
1340   {
1341     \pgfinterruptboundingbox
1342     \int_case:nn {\l__timechart_label_pos_int}
1343     {
1344       {0}{ \__timechart_interval_label_left: }
1345       {1}{ \__timechart_interval_label_center: }
1346       {2}{ \__timechart_interval_label_right: }
1347     }
1348     \endpgfinterruptboundingbox
1349   }
1350 }

```

(End of definition for `\__timechart_interval_label:.`)

`\__timechart_interval_label_left:` Place the label on the left of the interval.

```

1351 \cs_new:Npn\__timechart_interval_label_left:
1352 {
1353   \__timechart_if_x_in_bounds:nT{\l__timechart_start_extreme_x}
1354   {
1355     \group_begin:
1356     \pgftransformshift{
1357       \pgfpoint
1358       {\l__timechart_start_extreme_x}
1359       {\l__timechart_text_baseline_pgf}
1360     }
1361     \node[/timechart/interval-label,anchor=base-east]
1362     (\l__timechart_interval_label_node_name)
1363     at (0,0)
1364     {
1365       \__timechart_make_ref:NN
1366       \l__timechart_ref_text
1367       \l__timechart_label_text
1368     };
1369     \group_end:

```

```

1370     }
1371 }

```

(End of definition for `\_timechart_interval_label_left:`)

`\_timechart_interval_label_right:` Place the label on the right of the interval.

```

1372 \cs_new:Npn\_timechart_interval_label_right:
1373 {
1374   \_timechart_if_x_in_bounds:nT{\l_timechart_finish_extreme_x}
1375   {
1376     \group_begin:
1377     \pgftransformshift{
1378       \pgfpoint
1379       {\l_timechart_finish_extreme_x}
1380       {\l_timechart_text_baseline_pgf}
1381     }
1382     \node[/timechart/interval-label,anchor=base-west]
1383     (\l_timechart_interval_label_node_name)
1384     at (0,0)
1385     {
1386       \_timechart_make_ref:NN
1387       \l_timechart_ref_text
1388       \l_timechart_label_text
1389     };
1390   \group_end:
1391 }
1392 }

```

(End of definition for `\_timechart_interval_label_right:`)

`\_timechart_interval_label_center:` Place the label at the center of the interval.

```

1393 \cs_new:Npn\_timechart_interval_label_center:
1394 {
1395   \pgfmathsetlengthmacro{\l_timechart_label_anchor_x}
1396   {
1397     .5*max(\l_timechart_start_definite_x,\l_timechart_start_x)
1398     + .5*min(\l_timechart_finish_definite_x,\l_timechart_finish_x)
1399   }

```

First draw the ‘background label’ on the layer below the bar.

```

1400   \group_begin:
1401   \pgftransformshift{
1402     \pgfpoint
1403     {\l_timechart_label_anchor_x}
1404     {\l_timechart_text_baseline_pgf}
1405   }
1406   \pgfonlayer{labelbg}
1407   \node[
1408     /timechart/interval~label~centered~background,
1409     anchor=base
1410   ]
1411   at (0,0)
1412   { \l_timechart_label_text };
1413   \endpgfonlayer
1414   \group_end:

```



Then draw the label on top of the bar, clipping it to the bar outline.

```

1415 \pgfscope
1416 \pgfpathrectanglecorners{
1417 \pgfpoint
1418 {\l__timechart_start_extreme_x}
1419 {-\l__timechart_bar_half_thickness_pgf}
1420 }{
1421 \pgfpoint
1422 {\l__timechart_finish_extreme_x}
1423 {\l__timechart_bar_half_thickness_pgf}
1424 }
1425 \pgfusepath{clip}
1426 \group_begin:
1427 \pgftransformshift{
1428 \pgfpoint
1429 {\l__timechart_label_anchor_x}
1430 {\l__timechart_text_baseline_pgf}
1431 }
1432 \node[
1433 /timechart/interval-label~centered,
1434 anchor=base
1435 ]
1436 (\l__timechart_interval_label_node_name)
1437 at (0,0)
1438 {
1439 \__timechart_make_ref:NN
1440 \l__timechart_ref_text
1441 \l__timechart_label_text
1442 };
1443 \group_end:
1444 \endpgfscope
1445 }

```

(End of definition for `\__timechart_interval_label_center:`.)

## 14.16 Text

`\__timechart_text_user:0mm` Place text. This macro will be made available as `\timecharttext` inside the `timechart` environment.

**#1** : PGF keys under `/timechart/` to apply.

**#2** : Year at which to place text.

**#3** : Text.

```

1446 \NewDocumentCommand{\__timechart_text_user:0mm}{ O{} m m }
1447 {
1448 \str_if_empty:nF{#3}{
1449 \group_begin:
1450 \pgfmathsetmacro{\l__timechart_text_x}{yeartox(#2)}
1451 \__timechart_if_x_in_tolerance_bounds:nT{\l__timechart_text_x}
1452 {

```

Process keys supplied locally and retrieve needed keys.

```

1453 \pgfqkeys{/timechart}{
1454 #1,

```

```

1455         ref/.get=\l__timechart_ref_text,
1456         text~node~name/.get=\l__timechart_node_name_text,
1457         text~baseline/.get=\l__timechart_text_baseline_pgf,
1458     }

```

Shift to the correct vertical coordinate and place the text.

```

1459     \pgftransformshift{
1460         \pgfpoint{0}{\l__timechart_current_y}
1461     }
1462     \pgfinterruptboundingbox
1463     \group_begin:
1464     \pgftransformshift{
1465         \pgfpoint{\l__timechart_text_x}{\l__timechart_text_baseline_pgf}
1466     }
1467     \cs_set:Npn\l__timechart_text{#3}
1468     \int_case:nn {\l__timechart_text_pos_int}
1469     {
1470         {0}{ \cs_set:Npn\l__timechart_node_anchor_text{base~east} }
1471         {1}{ \cs_set:Npn\l__timechart_node_anchor_text{base} }
1472         {2}{ \cs_set:Npn\l__timechart_node_anchor_text{base~west} }
1473     }
1474     \node[/timechart/text,anchor=\l__timechart_node_anchor_text]
1475         (\l__timechart_node_name_text)
1476         at (0,0)
1477         {
1478             \__timechart_make_ref:NN
1479             \l__timechart_ref_text
1480             \l__timechart_text
1481         };
1482     \group_end:
1483     \endpgfinterruptboundingbox
1484 }
1485 \group_end:
1486 }

```

Since the text itself does not affect the bounding box, create a space (which will handle the automatic step).

```

1487     \__timechart_space_user:0[#1]
1488 }

```

*(End of definition for \\_\_timechart\_text\_user:0mm.)*

## 14.17 Space

`\__timechart_space_user:0` Create a vertical space as if there were an interval at the current coordinate. This macro will be made available as `\timechartspace` inside the `timechart` environment.

```

1489 \NewDocumentCommand{\__timechart_space_user:0}{ 0{} }
1490 {
1491     \group_begin:

```

Process keys supplied locally and retrieve the one needed value.

```

1492     \pgfqkeys{/timechart}{
1493         #1,
1494         interval~bar~thickness/.get=\l__timechart_bar_thickness_pgf,
1495     }

```

```

1496 \pgfmathsetlengthmacro{\l__timechart_bar_half_thickness_pgf}
1497 { .5*\l__timechart_bar_thickness_pgf }

```

Shift to the correct vertical coordinate and create the space.

```

1498 \pgftransformshift{
1499 \pgfpoint{0}{\l__timechart_current_y}
1500 }
1501 \pgfpathmoveto{
1502 \pgfpoint{0}{-\l__timechart_bar_half_thickness_pgf}
1503 }
1504 \pgfpathmoveto{
1505 \pgfpoint{0}{\l__timechart_bar_half_thickness_pgf}
1506 }
1507 \pgfusepath{discard}
1508 \group_end:
1509 \bool_if:NT\l__timechart_autostep_bool{
1510 \__timechart_step_y_user:0
1511 }
1512 }

```

(End of definition for \\_\_timechart\_space\_user:0.)

## 14.18 Legends

`\timechartlegenditem` Draw a bar suitable for use in a legend, applying style in #1.

```

1513 \NewDocumentCommand{\timechartlegenditem}{ 0{} }
1514 {
1515 \__timechart_legend_aux:nn{#1}{
1516 \pgfmathsetlengthmacro{\l__timechart_start_solid_x}{0}
1517 \pgfmathsetlengthmacro{\l__timechart_finish_solid_x}
1518 {\l__timechart_legenditem_width_pgf}
1519 \__timechart_interval_draw_solid:
1520 }
1521 }

```

(End of definition for \timechartlegenditem. This function is documented on page 14.)

`\timechartlegendstartrange` Draw a bar with start/finish range, suitable for use in a legend, applying style in #1.  
`\timechartlegendfinishrange`

```

1522 \NewDocumentCommand{\timechartlegendstartrange}{ 0{} }
1523 {
1524 \__timechart_legend_aux:nn{#1}{
1525 \pgfmathsetlengthmacro{\l__timechart_start_extreme_x}{0}
1526 \pgfmathsetlengthmacro{\l__timechart_start_solid_x}
1527 {\pgfkeysvalueof{/timechart/legend-item-range-width}}
1528 \pgfmathsetlengthmacro{\l__timechart_finish_solid_x}
1529 {\l__timechart_legenditem_width_pgf}
1530 \int_case:nn {\l__timechart_start_range_type_int}
1531 {
1532 {0}{ \__timechart_interval_start_range_fade: }
1533 {1}{ \__timechart_interval_start_range_slant: }
1534 }
1535 \__timechart_interval_draw_solid:
1536 }
1537 }

```

```

1538 \NewDocumentCommand{\timechartlegendfinishrange}{ 0{ } }
1539 {
1540   \__timechart_legend_aux:nn{#1}{
1541     \pgfmathsetlengthmacro{\l__timechart_start_solid_x}{0}
1542     \pgfmathsetlengthmacro{\l__timechart_finish_solid_x}
1543     {
1544       \l__timechart_legenditem_width_pgf
1545       -\pgfkeysvalueof{/timechart/legend~item~range~width}
1546     }
1547     \pgfmathsetlengthmacro{\l__timechart_finish_extreme_x}
1548     {\l__timechart_legenditem_width_pgf}
1549     \int_case:nn {\l__timechart_finish_range_type_int}
1550     {
1551       {0}{ \__timechart_interval_finish_range_fade: }
1552       {1}{ \__timechart_interval_finish_range_slant: }
1553     }
1554     \__timechart_interval_draw_solid:
1555   }
1556 }

```

(End of definition for `\timechartlegendstartrange` and `\timechartlegendfinishrange`. These functions are documented on page 14.)

`\__timechart_legend_aux:nn` Auxiliary command for legend items. Draw a TikZ picture, applying PGF keys #1 under `/timechart/` and using TikZ code #2.

```

1557 \cs_new:Npn\__timechart_legend_aux:nn #1#2
1558 {
1559   \tikzpicture
1560     \pgfkeys{
1561       /timechart/.cd,
1562       #1,
1563       interval~bar~thickness/.get=\l__timechart_bar_thickness_pgf,
1564       interval~bar~color/.get=\l__timechart_bar_color,
1565       interval~minimum~width/.get=\l__timechart_minimum_width_pgf,
1566       beyond~length/.get=\l__timechart_beyond_length_pgf,
1567       legend~item~width/.get=\l__timechart_legenditem_width_pgf,
1568     }
1569     \pgfmathsetlengthmacro{\l__timechart_bar_half_thickness_pgf}
1570     {.5*\l__timechart_bar_thickness_pgf}
1571     \pgfmathsetmacro{\l__timechart_start_x}{0}
1572     \pgfmathsetmacro{\l__timechart_finish_x}
1573     {\l__timechart_legenditem_width_pgf}
1574     \pgfmathsetlengthmacro{\l__timechart_start_beyond_x}
1575     {\l__timechart_start_x-\l__timechart_beyond_length_pgf}
1576     \pgfmathsetlengthmacro{\l__timechart_finish_beyond_x}
1577     {\l__timechart_finish_x+\l__timechart_beyond_length_pgf}
1578     \pgfmathsetmacro{\l__timechart_current_y}{0}

```

Ensure that the legend is ‘visible’ from the perspective of the drawing macros.

```

1579     \pgfmathsetmacro{\l__timechart_start_tolerance_x}
1580     {\l__timechart_start_x-10mm}
1581     \pgfmathsetmacro{\l__timechart_finish_tolerance_x}
1582     {\l__timechart_finish_x+10mm}
1583     #2

```

Make sure that the legend has the required bounding box.

```
1584     \pgfresetboundingbox
1585     \pgfmoveto{
1586         \pgfpoint{0}{-\l__timechart_bar_half_thickness_pgf}
1587     }
1588     \pgfmoveto{
1589         \pgfpoint
1590             {\l__timechart_legenditem_width_pgf}
1591             {\l__timechart_bar_half_thickness_pgf}
1592     }
1593     \endtikzpicture%
1594 }

(End of definition for \__timechart_legend_aux:nn.)

1595 </package>
```