

## rayon

[Documentation](#)
[Downloads](#)

### Table Of Contents

[rytimeseries](#)
[SYNOPSIS](#)
[DESCRIPTION](#)
[TERMINOLOGY](#)
[CONFIGURATION](#)
[READING DATA](#)
[Filter Expressions](#)
[VISUALIZATION STYLES](#)
[ARGUMENTS AND](#)
[OPTIONS](#)
[Required Arguments](#)
[Input](#)
[Sorting](#)
[Scaling](#)
[Binning](#)
[STATISTICS](#)
[DISPLAY](#)
[Displaying Outliers](#)
[Width and Height](#)
[Visualization Style](#)
[Options](#)
[dots](#)
[lines](#)
[filled\\_lines](#)
[bars](#)
[Tick Mark Display](#)
[Options](#)
[Annotation Display](#)
[Options](#)
[Border Options](#)
[Backgrounds](#)
[Gridlines](#)
[Padding](#)
[EXAMPLES](#)
[Default Values](#)
[Two Columns, Top and](#)
[Bottom](#)
[One Column, Top and](#)

## rytimeseries

### SYNOPSIS

```
rytimeseries [options]
```

### DESCRIPTION

`rytimeseries` is a command-line tool for generating visualizations from time-series data.

Some features of `rytimeseries` include:

- Specification of different visualization styles.
- Support for many ways of extracting time series from input data.
- Visualization of single or multiple time series.
- Automatic (re-)binning of input data, if desired.
- Visualization of data's central tendency and variability.
- Annotations of highest points in data.

### TERMINOLOGY

This section describes the visual layout of an `rytimeseries` visualization, and defines terms that will be used later in this document to refer to parts of the visualization.

One can think of an `rytimeseries` visualization as a list of smaller visualizations, laid out vertically. We call these visualizations *subcharts*. Each subchart can plot up to 2 data series, or *sides*. On the *top side*, the horizontal axis represents time and runs from left to right; the vertical axis represents value, with larger values above smaller ones. On a *1-sided* subchart, the top side takes up the full vertical space. A *2-sided* subchart has both a top and *bottom side*. On the bottom side, the horizontal axis again represents time and runs left to right, and the vertical axis represents value, but the axis is inverted, such that larger values are below smaller ones. Each side takes 50% of the vertical space, with the lowest point of their value scale occupying the center.

When data only exists for one subchart, that chart takes up all available horizontal and vertical space. When the data suggests multiple subcharts, an equal amount of space to the left of each subchart is set aside for a *chart label* to identify each subchart.

### CONFIGURATION

Every option available at the command line may be specified in a

[Bottom](#)  
[Grouping Into Multiple](#)  
[Subcharts](#)  
[SEE ALSO](#)

#### Previous topic

[ryscatterplot](#)

#### Next topic

[rytools](#)

#### Quick search

 

configuration file. For more information on the format of the configuration file, see [ryrc\(5\)](#).

## READING DATA

Time series data can be embedded in tabular data in many different ways. `rytimeseries` supports the extraction of this data in several ways using the `--top-column`, `--bottom-column`, `--top-filter` and `--bottom-filter` options. (See [Input](#).) The `--top-column` and `--bottom-column` select the columns in the input from which the data for the top and bottom sides of the subcharts will come. The `--top-filter` and `--bottom-filter` options express which rows in those columns will be displayed on the top and bottom sides as tests. The tests can be done on any item of data in the row, not just the data that will be displayed in the top or bottom side.

## Filter Expressions

The `--top-filter` and `--bottom-filter` options take filter expressions as their values. A filter expression has the following form:

---

```
colspec oper value
```

---

Where *colspec*, *oper* and *value* are defined as:

### *colspec*

A column name or index, enclosed in parentheses. For a three-column dataset with column names *foo*, *bar* and *baz*, valid colspecs would be `[0]`, `[1]`, `[2]`, `[foo]`, `[bar]`, and `[baz]`.

### *oper*

Either `=`, indicating that the value in *colspec* should be equal to *value*, or `!=`, indicating that the value in *colspec* should NOT be equal to *value*.

### *value*

A value to compare to the value at each row in *colspec*.

The following are valid filter expressions for a 4-column dataset with column names *foo*, *bar*, *baz*, and *quux*:

```
[0]==1
```

Selects all rows for which the value in the first column is 1.

```
[foo]==1
```

Identical to the preceding example.

```
[3]!=gargle
```

Selects all rows for which the value in the fourth column is NOT the string "gargle."

```
[quux]!=1
```

Identical to the preceding example.

Filter expressions will be coerced into the data type of the column. So a value of “1.2.3.4” will match the string “1.2.3.4” or the IP address 1.2.3.4, depending on the type of the column.

## VISUALIZATION STYLES

`rytimeseries` can visualize data in one of four styles: `dots`, `lines`, `filled_lines`, and `bars`.

Style	Result
<code>dots</code>	A scatterplot. Individual observations are plotted as marks (usually dots).
<code>lines</code>	A line plot. Lines are drawn from one observation to the next.
<code>filled_lines</code>	Similar to <code>lines</code> , but the area underneath the line is shaded (“filled”).
<code>bars</code>	A bar plot. Each bar represents a time range. Bar height signifies the sum of all observations in that range.

The `--style` argument controls which visual style to use. If the user specifies no style, `rytimeseries` will use the `dots` style.

## ARGUMENTS AND OPTIONS

The following arguments and options control `rytimeseries` behavior.

### Required Arguments

`--input-path <file>`

Required. A file containing the input data. If `--input-path` is a hyphen (`-`), input will be read from standard input. The data should be in the format described in [rydataformat](#).

`--output-path <file>`

Required. A file containing the output visualization. If the file does not exist, it will be created; if it does exist it will be overwritten. The extension of the file determines the output file format. Understood extensions are `.png` (PNG), `.svg` (SVG), `.ps` (PostScript) and `.pdf` (PDF).

### Input

`--first-line-colnames` If this option is specified, `rytimeseries` will extract column names from the first line of input, ignoring it as data. Use this when reading input from tools such as SiLK, where this form is the standard. You don’t need to specify this flag if the column name information is in the format described in [rydataformat](#), but doing so is harmless.

`--group-by <colname>`  
A column name or index. Each unique value in the column will form a subchart in the visualization.

`--show`  
A list of unique values in the `--group-by` column. Only subcharts for these values will be drawn.

`--labels <list>`  
A list of values. If `--show` is supplied, this list should have the same number of items. Otherwise, it should have the same number of unique values as there are in the `--group-by` column. (If `--group-by` is not supplied, this option is ignored.) These names will be substituted, in order, for values in `--group-by` for the purposes of labeling subcharts.

`--time-column <colname>`  
A column name or index of a column containing datetime values. The data in this column will be used for time values for all sides and subcharts.

`--top-column <colname>`  
A column name or index of a column containing numeric values. The data in this column will be used to populate the top side of the subcharts.

`--top-filter <filter-expr>`  
A filter expression (see [Filter Expressions](#)) that will be used to select data from `top-column` to display on the top side.

`--bottom-column <colname>`  
A column name or index of a column containing numeric values. The data in this column will be used to populate the bottom side of the subcharts.

`--bottom-filter <filter-expr>`  
A filter expression (see [Filter Expressions](#)) that will be used to select data from `top-column` to display on the bottom side.

`--start-time <time>`  
A datetime, in ISO-8601 format. Input data before to this time will be ignored.

`--end-time <time>`  
A datetime, in ISO-8601 format. Input data after this time will be ignored.

`--value-min <num>`  
A numeric value. Data points less than this value (after binning, if performed) will be ignored.

`--value-max <num>`

A numeric value. Data points greater than this value (after binning, if performed) will be ignored (or plotted as outliers; see [Displaying Outliers](#))

`--value-min-pct <int>`

An integer between 0 and 100, representing a percentile. For the data to be plotted, data points in percentiles less than this will be ignored (or plotted as outliers; see [Displaying Outliers](#))

`--value-max-pct <int>`

An integer between 0 and 100, representing a percentile. For the data to be plotted, data points in percentiles greater than this will be ignored (or plotted as outliers; see [Displaying Outliers](#))

`--fix-scale-min <num>`

A numeric value signifying the fixed lower bound of the value scale. (If not supplied, the lower bound will be the lowest-valued data point if there is negative data, or 0 if all the data has positive value.) If multiple plots are generated, they will ALL use this minimum scale. Points below this threshold are not plotted, regardless of the value of `--value-min`. (`--value-min` may still be used to make `rytimeseries` ignore values between `--value-min` and `--fix-scale-min`.)

`--fix-scale-max <num>`

A numeric value signifying the fixed upper bound of the value scale. (If not supplied, the upper bound will be the highest-valued data point.) If multiple plots are generated, they will ALL use this maximum scale. Points above this threshold are not plotted, regardless of the value of `--value-max`. (`--value-max` may still be used to make `rytimeseries` ignore values between `--value-max` and `--fix-scale-max`.)

## Sorting

`--presorted-input`

Flag. By default, `rytimeseries` will sort incoming data on the value column. Passing this switch indicates that the input is already sorted by time. This option can speed up processing, but will cause errors if the data is not actually sorted.

## Scaling

`--value-scale <scaletype>`

Type of scale to use for values. Valid values for this option are `linear` (linear scale), `log` (log scale), or `clog` (a “counting log” scale that distorts values between 0 and 1,

but accepts zero as a value). By default, a linear scale is used.

## Binning

Some visualizations (such as barplots) make more sense when the input data is binned. `rytimeseries` supports the binning (or re-binning) of input data. (Note, however, that it may be more efficient to do this prior to calling `rytimeseries`. For instance, when dealing with [SiLK](#) data, it will be faster to run your output through `rwcount` rather than using the binning in `rytimeseries`.)

`--bin-size`

A `timedelta`, in ISO-8601 format. If the data is to be binned, this specifies the size of the bins.

`--prebinned-input`

Indicates that the input is already in *bin-size* bins. This option can speed up processing, but will cause errors if the data is not actually correctly binned.

## STATISTICS

---

In addition to displaying the raw data, `rytimeseries` can visualize basic statistics on the central tendency and variation of the data.

`--trend-line <trendtype>`

Add a trend line showing central tendency over time; the argument determines the algorithm used to compute the central tendency. Valid values are `kernel` (kernel smoothing using a [Nadaraya-Watson](#) estimator), `ols` (linear regression using the [ordinary least squares](#) method), and `moving_avg` (calculation of averages over a sliding time window).

`--trend-line-color <color>`

A specification of the color of the trend line. (See [ryspecs\(5\)](#).)

`--trend-line-width <num>`

The width of the trend line, in pixels or points.

`--variation-field <fieldtype>`

Add a color field showing the variation of the data over time; the argument determines the algorithm used to compute variation. Currently, the only valid value is `stdev` (moving standard deviation, expressed relative to the moving average).

`--variation-field-color <color>`

A specification of the color of the variation field. (See [ryspecs\(5\)](#).)

`--variation-line-color <color>`

A specification of the color of the line surrounding the variation field. (See [ryspecs\(5\)](#).)

`--variation-line-width <num>`

The width of the line surrounding the variation field, in pixels or points.

`--variation-line-style <style>`

The style of line to draw around the variation field. (See [ryspecs\(5\)](#).)

## DISPLAY

---

The following options control how the data is displayed and what *decorations* (titles, captions, tick marks) are applied to the visualization.

`--style`

Required. Select the visualization style. Valid values are `dots`, `lines`, `filled_lines` and `bars`. Default is `dots`. (See [VISUALIZATION STYLES](#).)

`--value-units <string>`

Units in which the value is measured.

`--top-label <string>`

Label to be printed on the top half of visualizations, presumably representing the way the data on top is different from the data on the bottom.

`--bottom-label <string>`

Label to be printed on the bottom half of visualizations, presumably representing the way the data on bottom is different from the data on the top.

`--title <string>`

Title of the visualization, printed on the top in a large typeface.

`--caption <string>`

Caption of the visualization, printed on the bottom in a smaller typeface.

`--draw-as-multiple`

Flag. Draw a single timeseries in the style used for drawing multiple timeseries. This is useful when “stitching” a multiple-series visualization together using individual images. (As an HTML page, for example.)

`--no-timeline`

Flag. Do not draw labeled time axis on bottom of visualization. This may be used with `--draw-as-multiple` to generate composite visualizations using individual images.

```
--group-label-size <sizespec>
```

A textual size specification of the size of the tick mark label. (See [ryspecs\(5\)](#).) For example, to specify a tick label size of 12 pixels, use a value of 12px.

## Displaying Outliers

Instead of ignoring the data trimmed with `--value-max`, the user may instead wish to plot it at the edge of the data area as outliers. The following options control this behavior.

```
--plot-high-outliers
```

Flag. Display values above those of `--value-max` as outliers. Outliers will be plotted at the appropriate position on the time axis, and at the very top of the value axis.

```
--outlier-marker-color <color>
```

A specification of the color of the outlier marker. (See [ryspecs\(5\)](#).)

```
--outlier-marker-size <num>
```

Size of the outlier marker, as a number of points or pixels.

```
--outlier-marker-shape <shape>
```

Shape of the outlier marker.

## Width and Height

The following options control the dimensions of the output image `rytimeseries` generates.

By default, the number of subcharts in a visualization will determine the height of the image; if desired, the user can specify a static height regardless of the number of subcharts.

```
--width <num>
```

The width of the image, as a number of points or pixels. By default, `rytimeseries` will use a width of 800 pixels.

```
--height <num>
```

The height of the image, as a number of points or pixels. By default, `rytimeseries` will determine height dynamically. See `--height-per-subchart`.)

```
--height-per-subchart <num>
```

The height of the image, as a number of points or pixels per subchart. (e.g., `--height-per-subchart 100` would yield a 100-pixel height for one subchart, 200 pixels for two, 1000 pixels for ten, etc.) By default, `rytimeseries` will allocate 450 pixels per subchart.

## Visualization Style Options

The following options are used for specific values of the `--style`. If the



--style value does not apply to these options, they will be ignored.

### **dots**

--marker-color <color>

A specification of the color of the observation marker. (See [ryspecs\(5\)](#).)

--marker-size <num>

Size of the observation marker, as a number of points or pixels.

--marker-shape <shape>

Shape of the observation marker.

### **lines**

--line-color <color>

A specification of the color of the line. (See [ryspecs\(5\)](#).)

--line-width <num>

Width of the line, as a number of points or pixels.

--line-style <style>

The style of the line. (See [ryspecs\(5\)](#) for valid line style values.)

### **filled\_lines**

--line-color <color>

A specification of the color of the line. (See [ryspecs\(5\)](#).)

--line-width <num>

Width of the line, as a number of points or pixels.

--line-style <style>

The style of the line. (See [ryspecs\(5\)](#).)

--top-field-color <color>

A specification of the color of the filled area below the line. (See [ryspecs\(5\)](#).)

--bottom-field-color <color>

A specification of the color of the filled area above the line. (See [ryspecs\(5\)](#).)

### **bars**

--bar-width <proportion>

A number between 0 and 1, indicating the width of the bars, as a proportion of the available width. For example, a value of 1 indicates that the bar should occupy all available width

(so there is no empty space between bars). A value of .5 will occupy half the available width.

`--bar-border-width <num>`

Width of the border of the bar, as a number of points or pixels.

`--bar-fill-color <color>`

A specification of the color of the filled area inside the bar. (See [ryspecs\(5\)](#).)

`--bar-border-color <color>`

A specification of the color of the border of the bar. (See [ryspecs\(5\)](#).)

### Tick Mark Display Options

These options control how tick marks look, and where `rytimeseries` places them on the value (Y) axis.

`--value-ticks <tickspec>`

A specification of where to place tickmarks on the value (vertical) axis. (See [ryspecs\(5\)](#))

`--value-tick-size <num>`

Length of the tick mark, as a number of points or pixels.

`--value-tick-label-format <format>`

A format string or labeling style specifying how `rytimeseries` should derive the label from its position on the value scale.

The valid label styles are `autofloat`, `binary` or `metric`.

Style	Result
<code>autofloat</code>	Display the value with a number of decimal places chosen to compromise readability and accuracy.
<code>binary</code>	Format the value using SI binary prefix notation. (E.g., 1024 bytes == 1 kibibyte, or 1KiB.) This style is appropriate for quantities that are meaningful as powers of two.
<code>metric</code>	Format the value using the SI metric prefix notation. (E.g.,

1000 bytes == 1 kilobyte, or 1KB). This style is appropriate for quantities that are meaningful as powers of ten.

---

(Note that bytes are commonly counted using both the binary and metric styles.)

If the value is not one of these literals, it is presumed to be a format string, following the rules of [Python 2.0 string formatting](#). A dictionary is used as the input to the format string, containing these keys:

---

```
.. list-table::
    * - Key
      - Description

    * - ``value``
      - The value the tick mark represents

    * - ``autofloat_value``
      - The value the tick mark represents,

    * - ``units``
      - The value of the ``value-units`` opt
```

---

`--value-tick-label-size <sizespec>`

A textual size specification of the size of the tick mark label. (See [ryspecs\(5\)](#).) For example, to specify a tick label size of 12 pixels, use a value of 12px.

`--value-tick-label-spacing <num>`

Space between a tick mark and its label, as a number of points or pixels.

`--value-tick-label-halign <halign>`

One of the values “left”, “center” or “right,” indicating whether the tick label should be aligned to the left, center or right of the text, respectively.

`--value-tick-label-valign <valign>`

One of the values “top”, “center” or “bottom,” indicating whether the tick label should be aligned to the top, center or bottom of the text, respectively.

`--value-tick-label-angle <angle>`

A number from 0 to 360, indicating the angle that the tick label should be rotated. A value of 0 (or 360) indicates that the label should be drawn perfectly horizontally, drawn left to right. Increasing values from 0 will rotate the text counterclockwise—a value of 90 results in text drawn

vertically, bottom to top, 180 is text drawn horizontally but upside-down, right to left.

`--time-major-ticks (auto|none)`

A specification of where to place major tick marks on the time (horizontal) axis. At this time, the only two options are `auto` (place tickmarks automatically) and `none` (don't place any tick marks).

`--time-major-tick-size <num>`

Length of the major tick mark, as a number of points or pixels.

`--time-major-tick-label-size <sizespec>`

A textual size specification of the size of the major tick mark label. (See [ryspecs\(5\)](#).) For example, to specify a tick label size of 12 pixels, use a time of 12px.

`--time-major-tick-label-spacing <num>`

Space between a major tick mark and its label, as a number of points or pixels.

`--time-minor-ticks (auto|none)`

A specification of where to place minor tick marks on the time (horizontal) axis. At this time, the only two options are `auto` (place tickmarks automatically) and `none` (don't place any tick marks).

`--time-minor-tick-size <num>`

Length of the minor tick mark, as a number of points or pixels.

`--time-minor-tick-label-size <sizespec>`

A textual size specification of the size of the minor tick mark label. (See [ryspecs\(5\)](#).) For example, to specify a tick label size of 12 pixels, use a time of 12px.

`--time-minor-tick-label-spacing <num>`

Space between a minor tick mark and its label, as a number of points or pixels.

`--time-tick-label-halign <halign>`

One of the values "left", "center" or "right," indicating whether the tick label should be aligned to the left, center or right of the text, respectively.

`--time-tick-label-valign <valign>`

One of the values "top", "center" or "bottom," indicating whether the tick label should be aligned to the top, center or bottom of the text, respectively.

`--time-tick-label-angle <angle>`

A number from 0 to 360, indicating the angle that the tick label should be rotated. A value of 0 (or 360) indicates that

the label should be drawn perfectly horizontally, drawn left to right. Increasing values from 0 will rotate the text counterclockwise—a value of 90 results in text drawn vertically, bottom to top, 180 is text drawn horizontally but upside-down, right to left.

### Annotation Display Options

`--annotate-max`

Flag. Place an annotation at the first instance of the maximum value plotted in the visualization. Annotated points are called out with a circular marker and labeled with the value of the annotated observation.

`--annotation-marker-color <color>`

A specification of the color of the marker calling out the annotated point. (See [ryspecs\(5\)](#).)

`--annotation-marker-size <num>`

Number of points/pixels describing the radius of the annotation callout marker.

`--annotation-label-size <sizespec>`

A textual size specification of the size of the tick mark label. (See [ryspecs\(5\)](#).) For example, to specify an annotation label size of 12 pixels, use a time of 12px.

`--annotation-label-color <color>`

A specification of the color of the annotation label. (See [ryspecs\(5\)](#).)

`--annotation-label-background-color <color>`

A specification of the color of the background of the annotation label. (See [ryspecs\(5\)](#).)

`--annotation-label-spacing <num>`

A number indicating the space between the annotation marker and its label, as points or pixels.

### Border Options

`--vertical-border-line-style <stylespec>`

The style of line to draw around on the vertical border of the visualization. (See [ryspecs\(5\)](#).) For visualizations with a top and bottom component, this option will affect the vertical border for both components.

`--horizontal-border-line-style <stylespec>`

The style of line to draw around on the horizontal border of the visualization. (See [ryspecs\(5\)](#).)

### Backgrounds

`--chart-background-color <color>`

A specification of the color of the background of the entire subchart. (See [ryspecs\(5\)](#).)

`--plot-background-color <color>`

A specification of the color of the background of just the plots. (See [ryspecs\(5\)](#).)

## Gridlines

`--vgrid`

Flag. Draw vertical grid lines.

`--vgrid-color <color>`

A specification of the color of the vertical grid lines. (See [ryspecs\(5\)](#).)

`--vgrid-style <stylespec>`

The style of line to draw for the vertical grid lines.

`--vgrid-width <num>`

A number indicating the width of the vertical grid lines, in points or pixels.

`--vgrid-lines-at <tickspec>`

A specification of where to place the vertical grid lines, relative to the horizontal axis. (See [ryspecs\(5\)](#).)

`--hgrid`

Flag. Draw horizontal grid lines.

`--hgrid-color <color>`

A specification of the color of the horizontal grid lines. (See [ryspecs\(5\)](#).)

`--hgrid-style <stylespec>`

The style of line to draw for the horizontal grid lines.

`--hgrid-width <num>`

A number indicating the width of the horizontal grid lines, in points or pixels.

`--hgrid-lines-at <tickspec>`

A specification of where to place the horizontal grid lines, relative to the vertical axis. (See [ryspecs\(5\)](#).)

## Padding

`--padding <num>`

A number of pixels/points representing the amount of padding to be applied to the left, right, top and bottom edges of the visualization. (For example, `--padding 4` will add 4 pixels/points of padding each to every edge, so total vertical padding is 8 pixels, and total horizontal padding is also 8 pixels.) This option may not be used with any of the

other padding options.

`--pad-left <num>`

A number of pixels/points representing the amount of padding to be applied to the left edge of the visualization. This option may not be used with `--padding`.

`--pad-right <num>`

A number of pixels/points representing the amount of padding to be applied to the right edge of the visualization. This option may not be used with `--padding`.

`--pad-top <num>`

A number of pixels/points representing the amount of padding to be applied to the top edge of the visualization. This option may not be used with `--padding`.

`--pad-bottom <num>`

A number of pixels/points representing the amount of padding to be applied to the bottom edge of the visualization. This option may not be used with `--padding`.

## EXAMPLES

The following examples all assume that the first column is the time column. The time column may be set using the `--time-column` option.

### Default Values

Visualize a single, one-sided series of data in the file `in.txt` to the file `out.png`, using the defaults. (First column is time, second column is data):

---

```
rytimeseries --input-path=in.txt --output-path=out.png
```

---

This is equivalent to:

---

```
rytimeseries --input-path=in.txt --output-path=out.png
              --time-column=0 --top-column=1
```

---

### Two Columns, Top and Bottom

Consider the following input as `in.txt`:

---

```
2000-04-01 00:00:00+00:00|982.74|516.37
2000-04-01 01:00:00+00:00|1033.26|541.63
2000-04-01 02:00:00+00:00|1049.99|550.00
2000-04-01 03:00:00+00:00|1031.77|540.88
2000-04-01 04:00:00+00:00|979.87|514.93
2000-04-01 05:00:00+00:00|897.91|473.95
# ...
```

---

Visualize the input with data from column 1 (the first column is column 0) on the top side, and data from column 2 on the bottom side:

---

```
rytimeseries --input-path in.txt --output-path out.png
```

---

---

```
--top-column=1 --bottom-column=2
```

---

### One Column, Top and Bottom

Consider the following input as `in.txt`:

---

```
2000-04-01 00:00:00+00:00|a|982.74
2000-04-01 01:00:00+00:00|b|1033.26
2000-04-01 02:00:00+00:00|a|1049.99
# ...
```

---

Visualize the input with data on the top if column 1 is `a` and on the bottom if column 1 is `b`:

---

```
rytimeseries --input-path in.txt --output-path out.png
              --top-column=2 --top-filter="[1]==a" \
              --bottom-column=2 --bottom-filter="[1]==b"
```

---

Rows where column 1 is neither `a` nor `b` will be ignored.

### Grouping Into Multiple Subcharts

Consider the following input as `in.txt`:

---

```
2000-04-01 00:00:00+00:00|moe  |a|982.74
2000-04-01 00:00:00+00:00|moe  |b|483.79
2000-04-01 01:00:00+00:00|larry|a|1033.26
2000-04-01 01:00:00+00:00|larry|b|243.31
2000-04-01 02:00:00+00:00|curly|a|1049.99
2000-04-01 02:00:00+00:00|curly|b|492.01
# ...
```

---

Visualize all values from column 3 as a subchart for each of the values of column 1 (`moe`, `larry` and `curly`):

---

```
rytimeseries --input-path in.txt --output-path out.png
              --group-by=1 \
              --top-column=3
```

---

Put rows where column 2 is `a` on top and rows where column 2 is `b` on the bottom

```
rytimeseries --input-path in.txt --output-path out.png
              --group-by=1 --top-column=3 --top-filter="[2]==a" --bottom-
              column=3 --bottom-filter="[2]==b"
```

### SEE ALSO

[\*rytools\(5\)\*](#)