



CERT NetSA Security Suite

Monitoring for Large-Scale Networks

rayon

Documentation

rayon

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

Table Of Contents

[ryscatterplot](#)[SYNOPSIS](#)[DESCRIPTION](#)[CONFIGURATION](#)[REQUIRED](#)[ARGUMENTS](#)[MARKERS](#)[INPUT ASSOCIATION](#)[SCALING DATA](#)[FILTERING DATA](#)[STATISTICS](#)[DISPLAY](#)[Borders and Border](#)[Labels](#)[Layout, Decoration and](#)[Annotation](#)[Displaying Tickmarks](#)[GRIDDED](#)[SCATTERPLOTS](#)[DEPRECATED OPTIONS](#)[EXAMPLES](#)[SEE ALSO](#)

Previous topic

[rypiechart](#)

Next topic

[rytimeseries](#)

Quick search

ryscatterplot

SYNOPSIS

`ryscatterplot [options]`

DESCRIPTION

`ryscatterplot` reads data from a file or standard input and writes a scatterplot of the data to a PNG, SVG, PDF or PostScript file. The user may specify which columns of data to plot, parts of the plotted data columns to ignore, how to style the visualization, and whether to plot a trend line in addition to the data.

CONFIGURATION

Every option available at the command line may be specified in a configuration file. For more information on the format of the configuration file, see [ryrc\(5\)](#).

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\(5\)](#).

`--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).

MARKERS

The following options control the presentation of markers on the scatterplot. A marker's spatial position, size and color can all be determined from the input data. In addition, marker size and color can be set to a static value if the user chooses not to map data onto this axis.

`--marker <marker>`

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

`--marker-color <color>`

If the marker color will not vary with data, a specification of

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

`--marker-color-input <colspec>`

If the marker color will vary with data, the name or index of the column in the input data to use for marker color.

`--marker-color-scale-min <color>`

A specification of the color to be used for the lowest value in `--marker-color-input`. (See [ryspecs\(5\)](#).)

`--marker-color-scale-max`

A specification of the color to be used for the highest value in `--marker-color-input`. (See [ryspecs\(5\)](#).)

`--marker-size <num>`

If the marker size will not vary with data, a number specifying the size of the marker, in points or pixels. (This size value will be interpreted differently for different marker sizes, with the intent of producing a mark that fits in a square no greater than `--marker-size-input` pixels/points on a side.)

`--marker-size-input <colspec>`

If the marker size will vary with data, the name or index of the column in the input data to use for marker size. (See [ryspecs\(5\)](#).)

`--marker-size-scale-min <num>`

A specification of the size to be used for the lowest value in `--marker-size-input`.

`--marker-size-scale-max <num>`

A specification of the size to be used for the highest value in `--marker-size-input`.

INPUT ASSOCIATION

The following options associate data with dimensions in the visualization.

`--x-input <colspec>`

A column specification (see [ryspecs\(5\)](#)) of the data column to be plotted on the X axis.

`--y-input`

A column specification (see [ryspecs\(5\)](#)) of the data column to be plotted on the Y axis.

SCALING DATA

The following options control how data is scaled to the dimensions of the visualization.

`--x-scale <scale>`

The type of scale to use on the X axis. (See [ryspecs\(5\)](#).)

By default, a linear scale is used.

`--y-scale <scale>`

The type of scale to use on the y axis. (See [ryspecs\(5\)](#).)

By default, a linear scale is used.

By default, `rscatterplot` will start both the X and Y axes at zero if the input data for that axis is all positive, and with end axes at the highest points of their respective input data. The following options allow the user to manually set the high and low points of the X and Y scales.

`--x-scale-dont-zero`

Flag. Start the X axis scale at the lowest point in the X data, instead of zero.

`--y-scale-dont-zero`

Flag. Start the Y axis scale at the lowest point in the X data, instead of zero.

`--x-scale-min <num>`

A numeric value representing the lowest end of the X axis scale.

`--y-scale-min <num>`

A numeric value representing the lowest end of the Y axis scale.

`--x-scale-max <num>`

A numeric value representing the highest end of the X axis scale.

`--y-scale-max <num>`

A numeric value representing the highest end of the Y axis scale.

FILTERING DATA

It may sometimes be desirable to ignore some of the input data. The following options specify which input data should be ignored.

`--x-floor <num>`

A numeric value. Data below this point in the input will not be visualized.

`--y-floor <num>`

A numeric value. Data below this point in the input will not be visualized.

`--x-floor-pct <pct>`

A percentile value between 1 and 100. Values below this percentile in the data will not be visualized.

`--y-floor-pct <pct>`

A percentile value between 1 and 100. Values below this

percentile in the data will not be visualized.

`--x-ceiling <num>`

A numeric value. Data above this point in the input will not be visualized.

`--y-ceiling <num>`

A numeric value. Data above this point in the input will not be visualized.

`--x-ceiling-pct <pct>`

A percentile value between 1 and 100. Values above this percentile in the data will not be visualized.

`--y-ceiling-pct <pct>`

A percentile value between 1 and 100. Values above this percentile in the data will not be visualized.

STATISTICS

`ryscatterplot` can fit a trend line to the data. The following options control the trending method used and the line properties.

`--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.

`--trend-untrimmed-data`

Flag. Normally, the data used to compute the trend is the input data after it has been filtered through `--x-floor`, `--x-ceiling`, `--y-floor`, `--y-ceiling`, `--x-floor-pct`, `--y-floor-pct`, `--x-ceiling-pct`, or `--y-ceiling-pct`. If this flag is supplied, the raw data will be used before it was trimmed by these options.

DISPLAY

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

`--width <num>`

The width of the image, as a number of points or pixels.

`--height <num>`

The width of the image, as a number of points or pixels.

`--padding <num>`

A number of pixels or points of padding to applied uniformly to each side of the image. A `--padding` value of 10, for instance, will apply ten pixels/points of padding to each of the top, bottom, left and right edges, reducing the drawable width and height by 20 pixels/points.

`--pad-top <num>`

A number of pixels or points of padding to add to the top edge of the image.

`--pad-bottom <num>`

A number of pixels or points of padding to add to the top edge of the image.

`--pad-left <num>`

A number of pixels or points of padding to add to the top edge of the image.

`--pad-right <num>`

A number of pixels or points of padding to add to the top edge of the image.

`--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.

Borders and Border Labels

`--bottom-border-line-style <style>`

The style of the line to draw (if any) along the bottom axis of the frame to plot. (See [ryspecs\(5\)](#) for valid line style values.)

`--left-border-line-style <style>`

The style of the line to draw (if any) along the left axis of the frame to plot. (See [ryspecs\(5\)](#) for valid line style values.)

`--bottom-label <string>`

A string label which will be printed below the scatterplot.

`--left-label <string>`

A string label which will be printed to the left of the scatterplot.

`--x-label-angle <angle>`

A number from 0 to 360, indicating the angle that the left

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.

`--y-label-angle <angle>`

A number from 0 to 360, indicating the angle that the bottom 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.

`--x-label-halign <halign>`

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

`--x-label-valign <valign>`

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

`--y-label-halign <halign>`

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

`--y-label-valign <valign>`

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

`--x-label-spacing <num>`

Space between the bottom border and its label, as a number of points or pixels.

`--y-label-spacing <num>`

Space between the left border and its label, as a number of points or pixels.

Layout, Decoration and Annotation

`--chart-bgcolor <color>`

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

`--plot-bgcolor <color>`

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

Displaying Tickmarks

`--x-ticks <tickspec>`

A specification of where to place tick marks on the X axis.
(See [ryspecs\(5\)](#).)

`--y-ticks <tickspec>`

A specification of where to place tick marks on the Y axis.
(See [ryspecs\(5\)](#).)

GRIDDED SCATTERPLOTS

`rscatterplot` can generate grids of scatterplot data as well as single plots. This can be useful for rapidly comparing sets of similar data to each other.

In order to be plotted in a grid scatterplot, a dataset must contain an additional column called a *key column*. This column effectively partitions the dataset into smaller datasets; each of the smaller datasets shares the same label in the key column.

The following switches control gridded display of scatterplots.

`--grid-plot`

Flag. Enable gridded scatterplot display.

`--grid-key-input <colname>`

A column name or index. Each unique value in the column will form a cell in the scatterplot grid.

`--grid-label <label>`

Specifies a label to be given to each subplot. The label may contain the wildcard pattern `%s`, which will be replaced with the label in the grid key column for that subplot.

DEPRECATED OPTIONS

The following options will work, but have been deprecated. They will be removed in Rayon 2.x. Where applicable, alternatives are provided.

Deprecated option	Alternative
<code>--xborder</code>	<code>--bottom-border-line-style</code>
<code>--yborder</code>	<code>--left-border-line-style</code>
<code>--xticks</code>	<code>--x-ticks</code>
<code>--yticks</code>	<code>--y-ticks</code>
<code>--xscale</code>	<code>--x-scale</code>
<code>--xscale-max</code>	<code>--x-scale-max</code>
<code>--xscale-min</code>	<code>--x-scale-min</code>
<code>--yscale</code>	<code>--y-scale</code>
<code>--yscale-max</code>	<code>--y-scale-max</code>

--yscale-min	--y-scale-min
--xfloor	--x-floor
--xfloor-pct	--x-floor-pct
--xceiling	--x-ceiling
--xceiling-pct	--x-ceiling-pct
--yfloor	--y-floor
--yfloor-pct	--y-floor-pct
--yceiling	--y-ceiling
--yceiling-pct	--y-ceiling-pct
--xlabel	--bottom-label
--ylabel	--left-label
--background-color	--chart-bgcolor
--grid	--grid-plot

EXAMPLES

Create a visualization in PDF format using the default options:

```
ryscatterplot --input-path foo.txt --output-path bar.pdf
```

Using data from the SiLK `rwuniq(1)` tool, plot the number of bytes in for all netflows grouped by source IP address against the number of records. Output will go to the PNG file `bar.png`:

```
rwuniq --fields=sip --bytes --flows --no-titles | \
ryscatterplot --x-input=1 --y-input=2 --output-path=bar
```

Visualize a grid of scatterplots from data in `foo.txt`:

```
ryscatterplot --input-path=foo.txt \
--output-path=bar.png \
--grid \
--grid-key-input=2 \
--grid-label="Server %s"
```

SEE ALSO

[rytools\(5\)](#)