



CERT NetSA Security Suite

Monitoring for Large-Scale Networks

rayon

Documentation

rayon

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

Table Of Contents

[rycategories](#)[SYNOPSIS](#)[DESCRIPTION](#)[Ordering](#)[CONFIGURATION](#)[REQUIRED](#)[ARGUMENTS](#)[INPUT ASSOCIATION](#)[SCALING DATA](#)[DISPLAY](#)[Horizontal Bar Plots](#)[Category Layout](#)[Bars](#)[Displaying Tickmarks](#)[Gridlines](#)[Borders and Border](#)[Labels](#)[Layout, Decoration and](#)[Annotation](#)[EXAMPLES](#)[SEE ALSO](#)

Previous topic

[ryhilbert](#)

Next topic

[rypiechart](#)

Quick search

rycategories

SYNOPSIS

`rycategories [options]`

DESCRIPTION

`rycategories` reads data from a file or standard input and writes a plot of the data to a PNG, SVG, PDF or PostScript file. `rycategories` renders categorical data—data that is aggregated by a set of discrete, unordered categories.

Categorical data can be presented in a number of ways. Currently, `rycategories` only supports presentation as a bar plot. Other presentation styles are planned for the future.

Ordering

Categories in categorical data should not, in theory, have any order relationship to other categories. However, in practice when visualizing categorical data it is sometimes useful to order categories. For example, it may be easier for a viewer to locate categories of interest if the categories are ordered alphabetically, or in such a way that “families” of categories can be grouped together.

In recognition of this, `rycategories` will treat the order of data in the input file as a form of sort order, and will use this order to lay out categories if no other order is specified (e.g., with `--reverse-cats`). If rows with different keys are interleaved, the order used will be the relative order of the first rows to contain the keys.

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).

INPUT ASSOCIATION

The following options associate data with dimensions in the visualization.

`--cat-input <colspec>`

A column specification (see [ryspecs\(5\)](#)) of the data column to be used for categorical input. This will determine the number and placement of bars on the plot.

`--val-input <colspec>`

A column specification (see [ryspecs\(5\)](#)) of the data column to be used for categorical input. This will determine the bars' height. (Or width, if `--horizontal` is supplied.)

SCALING DATA

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

`--val-scale <scale>`

The type of scale to use on the value axis, which determines bar height (width if `--horizontal` is supplied). See [ryspecs\(5\)](#) for options. By default, a linear scale is used.

`--val-scale-min <num>`

A numeric value representing the lowest end of the value axis scale. This is not the same thing as the origin of the bar; set that using `--bar-origin`.

`--bar-origin <num>`

A numeric value representing the point on the axis where the bars originate. When the value data is all positive, this is always the bottom of the bar; when the value data is all negative, this is always the top of the bar.

This value defaults to 1 if `--val-scale` is `log`, otherwise 0.

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.

Horizontal Bar Plots

`--horizontal`

Flag. Render the data using horizontal bars arranged from top to bottom. (Default behavior is vertical bars arranged left to right.)

Category Layout

`--reverse-cats`

Flag. Reverse the order of the bars from their positioni in the dataset.

Bars

`--bar-width <num>`

A number between 0 and 1, representing the proportion of available space each bar will use. For example, `--bar-width .4` directs `rycategories` to use 40% of the allocated space for each bar. `--bar-width 1` means

use all allocated space, leaving no space between bars. Empty space will always be allocated evenly to either side of the bar.

If the `--horizontal` option is supplied, `--bar-width` refers to the vertical dimension of the bar, rather than the horizontal dimension.

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

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

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

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

`--bar-border-line-style <style>`

The style of line to draw as the bar border. (See [ryspecs\(5\)](#).)

`--bar-border-width <size>`

The width of the bar border, in pixels or points.

Displaying Tickmarks

`--val-ticks <tickspec>`

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

`--no-cat-ticks`

Flag. Do not place tick marks on each category.

`--cat-tick-size <size>`

The length of the category tickmarks, in pixels or points.

`--val-tick-size <size>`

The length of the value tickmarks, in pixels or points.

`--cat-label-angle <angle>`

A number from 0 to 360, indicating the angle that the category tickmark labels 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.

`--val-label-angle <angle>`

A number from 0 to 360, indicating the angle that the value tickmark labels 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.

`--cat-label-halign <halign>`

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

`--cat-label-valign <valign>`

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

`--val-label-halign <halign>`

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

`--val-label-valign <valign>`

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

`--cat-label-spacing <num>`

Space between the category tickmark and its label, as a number of points or pixels.

`--val-label-spacing <num>`

Space between the value tickmark and its label, as a number of points or pixels.

Gridlines

`--vgrid`

Flag. Draw vertical grid lines.

`--vgrid-color <color>`

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

`--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 [ryspects\(5\)](#).)

`--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\(5\)](#).)

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.)

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\(5\)](#).)

EXAMPLES

Create PDF visualization using the default options:

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

Using the output from the SiLK [rwuniq\(1\)](#) tool, plot the byte volume of traffic in the file `input.rw`, grouped by source IP:

```
rwuniq --fields=sip --bytes \
  --no-titles input.rw | \
rycategories --input-path - \
  --output-path bar.png
```

SEE ALSO

[rytools\(5\)](#)