# 1 Introduction

tikz-helper is a set of common lisp functions and macros to make plots. This is done by generating LaTeX code using pgf and Ti*k*Z.

To generate a plot, one of the macros with-tikz-to-file, with-tikz-to-string or with-tikz-to-stream is called. with-tikz-to-file and with-tikz-to-string are just wrappers for with-tikz-to-stream. The macros set up the latex environment needed by the figures, collects information needed to perform transformations between the data frame and a default frame, and draws axis for the plot. The transformations are linear and works so that (plot-x-min,plot-y-min) is at (0cm,0cm) in the default frame, and (plot-x-max,plot-x-min) is at (width cm, height cm)

The axis-style should be one of :rectangle :cross :left-bottom :popped-out or :none. Examples of all the different axis styles are below. Axis ticks are added to the axis. The position of the ticks is so that they are placed with a spacing of 1,2 or 5 times 10 to a power such that you get between 4 and 10 ticks on the axis. If custom ticks are needed, or ticks with names, not numbers, use :none, and call the corresponding draw-axis-* function.

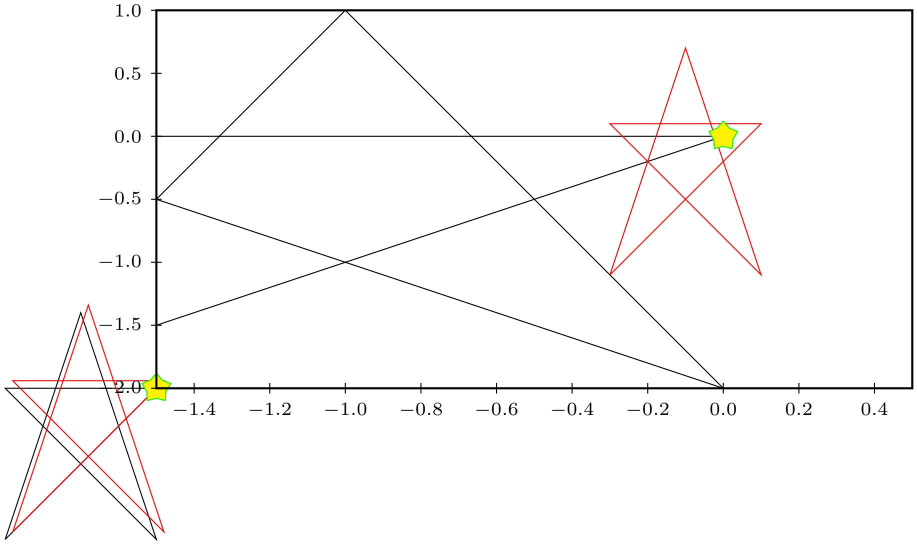

Figure 1: By default paths and nodes are drawn in a frame where origin is the lower left corner of the plot, and the units in x and y is 1cm. The transform macro generates tikz transformations, so that all points within the scope are drawn in the plot frame, defined by plot- x-min x-max and y-min y-max. The clip-and-transform macro also clips the plotting area. Values with units like cm or pt are scaled, but all poits are translated.

# 2 Simple plots

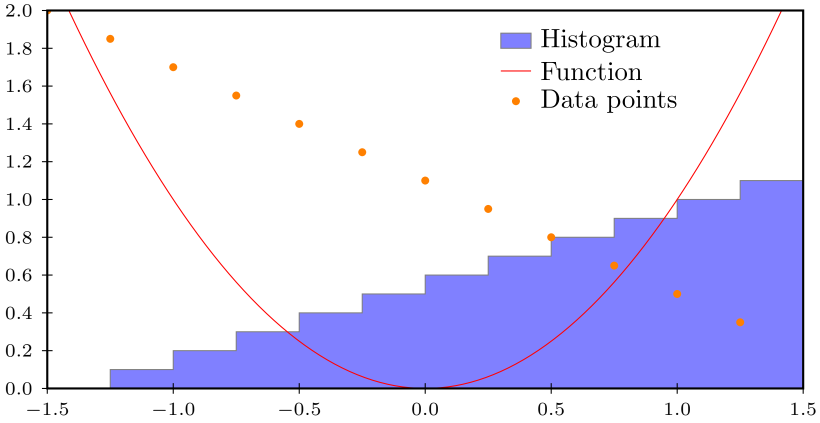

Figure 2: A histogram, a function and some datapoints. Most functions dealing with sets of data points call the clip-and-transform macro themselves, so calling it from top level is not necessary.

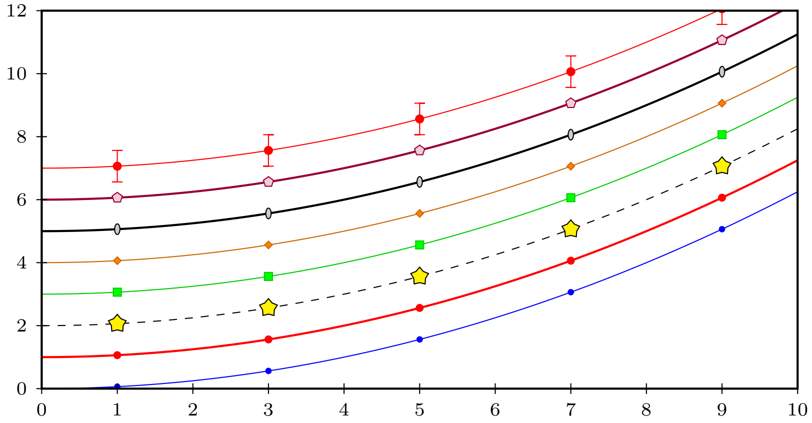

Figure 3: Different styles of lines and nodes. The styles are just regular tikz options.

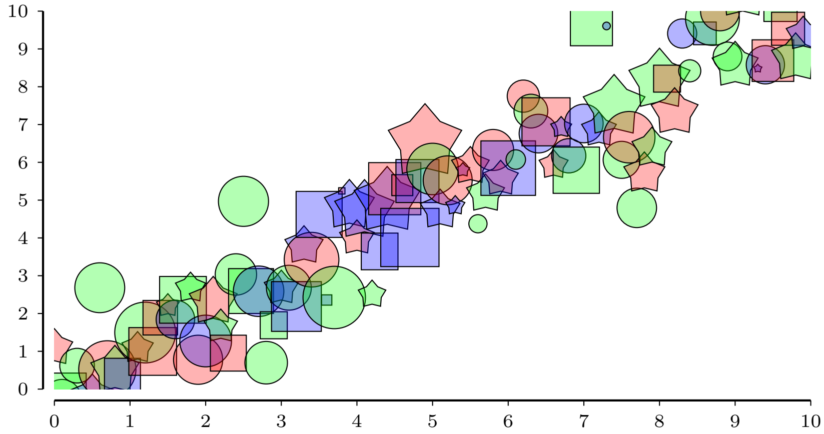

Figure 4: Data points of varying sizes, shapes and colors. Draw node does not automatically transform, since it can be useful in the default frame.

It's possible to draw nodes in captions.

```
%The preamble needs:
\usepackage[singlelinecheck=off]{caption}
%Inside the figure environment
\captionsetup{singlelinecheck=off}
\caption[foo bar]{\node at (0,0) ...}
```

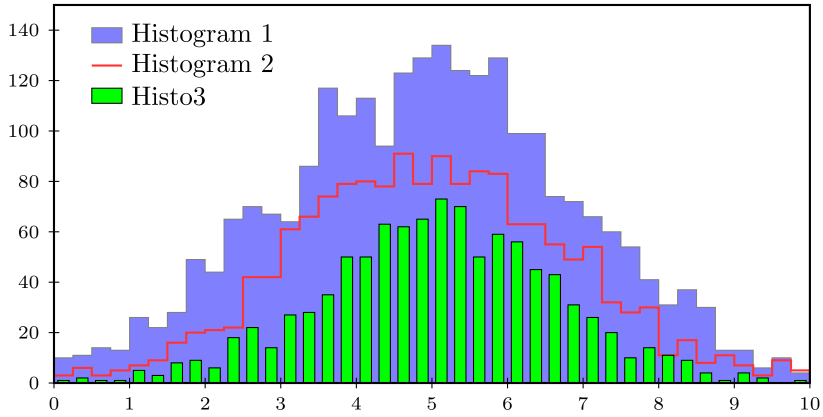

Figure 5: Some Gaussian histograms with different styles and with legend entries. The legend entries are placed in the default cm frame, unless draw-histogram is called within (transform (tikz) ...). With some trickery it is also possible to get legends in captions: 🟦 Histogram 1, — Hisogram 2, 🟩 Hisogram 3.

Simple sparkline:

More complex sparkline: 1.4, ▨ indicates the $1\sigma$ band

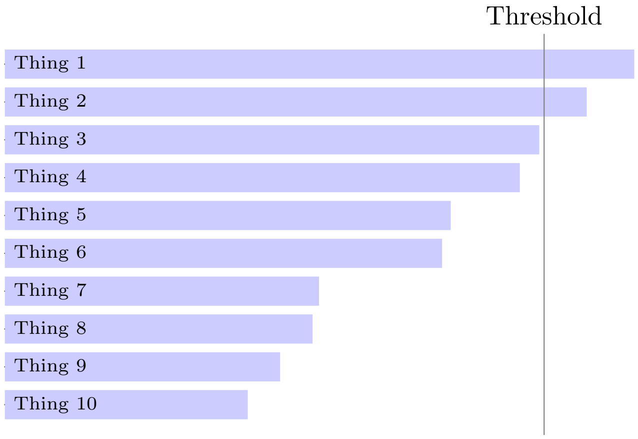

Figure 6: Histogram with bins extending in the horizontal direction. This is just a tikzpicture with the transformation rotate=-90. The text is not rotated, so acis ticks require extra care, and text in legend entries will not be properly aligned. The bins are named with the draw-axis-ticks function.

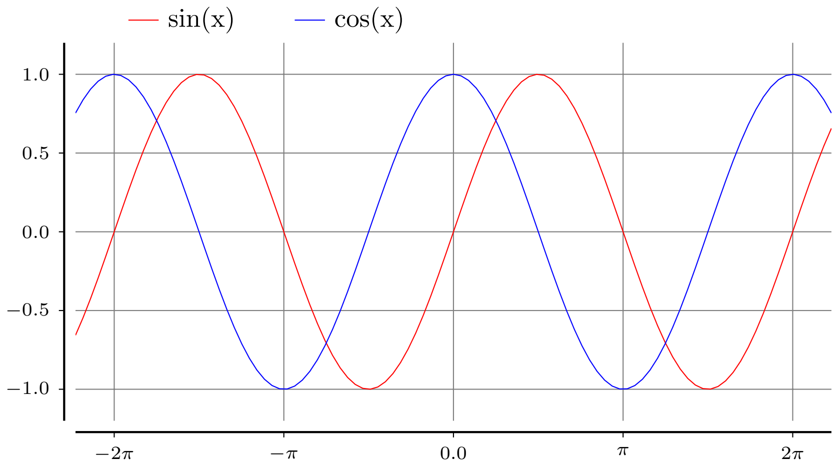

Figure 7: Plotting sin(x) and cos(x), with grid lines and tick names on the x-axis.

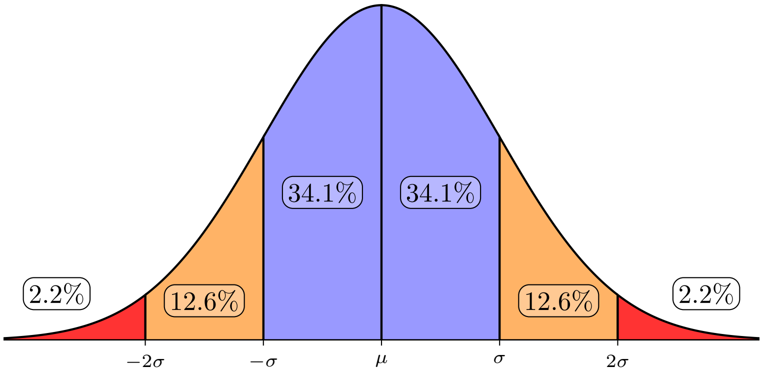

Figure 8: Gaussian function, made by closing, drawing and filling function segments.

# 3 Fitting with levenberg marquart

The Levenberg-Marquart algorithm minimizes the squared distance in the y-direction between a function and a set of datapoints by manipulating function parameters. If errors are supplied the $\chi^2$, or the squared normalized differences, is minimized.

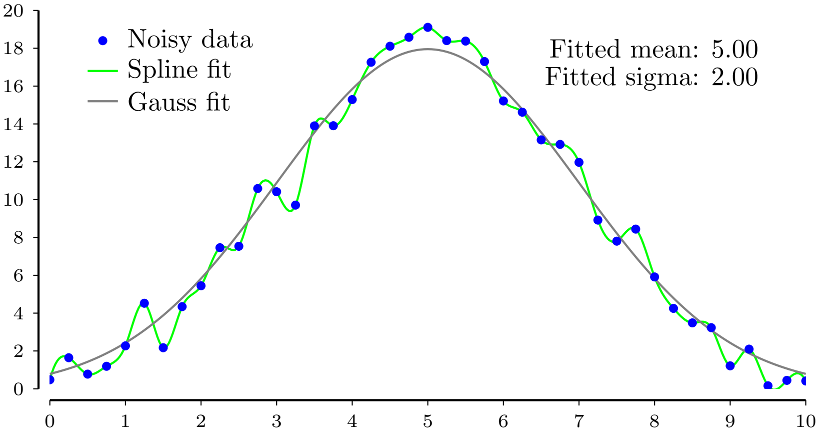

Figure 9: Some Gauss smeared data points, fitted with the Gaussian function. Fit parameters are printed in the plot. A spline fit is also plotted.

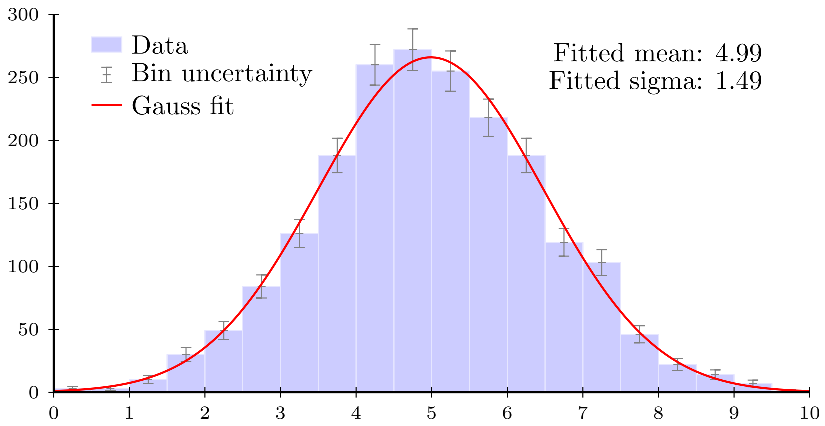

Figure 10: Same as above, except the data points have errors. The error bars are calculated from bin content. Empty bins are discarded in the fit.

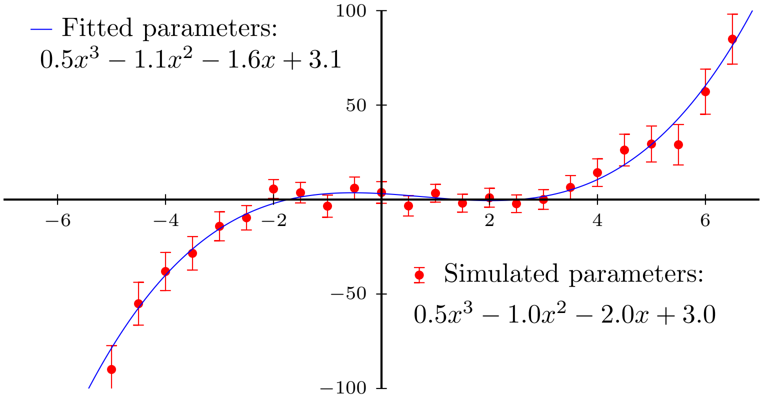

Figure 11: Noisy data points, with known errors, fitted with a polynomial of the third degree. The ”Simulated parameter” legend is placed in the default frame, the ”Fitted parameters” in the data frame.

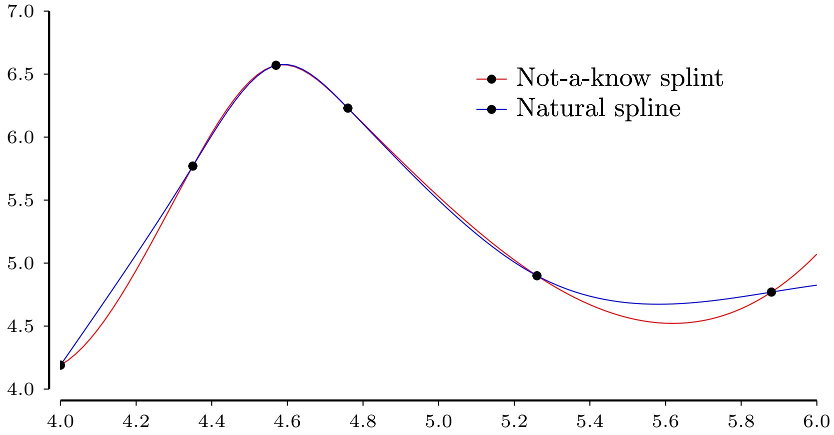

Figure 12: Cubic splines, with different end point conditions.

# 4 Sub figures

The with-sub-figure macro makes it possible to draw with a new set of transformations within the same tikz figure. In addition to the with-tikz-to-stream arguments, the sub-figure also expects the offset from (0,0) in the default cm frame.

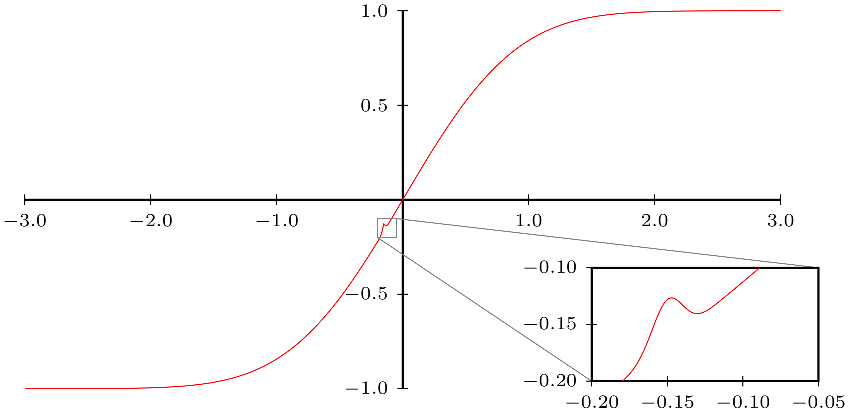

Figure 13: More than one plot can be plotted in the same figure by using sub figures. Sub figures are basically a new set of transformations, and do not affect the default cm frame at all. Here is a function with a zoomed view of a region of interest.

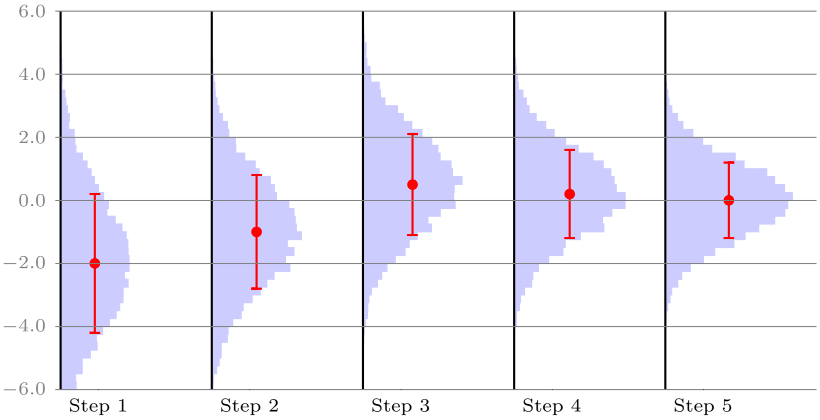

Figure 14: Horizontal histograms, in sub figures side by side. The mean and $\sigma$ are indicated in red.

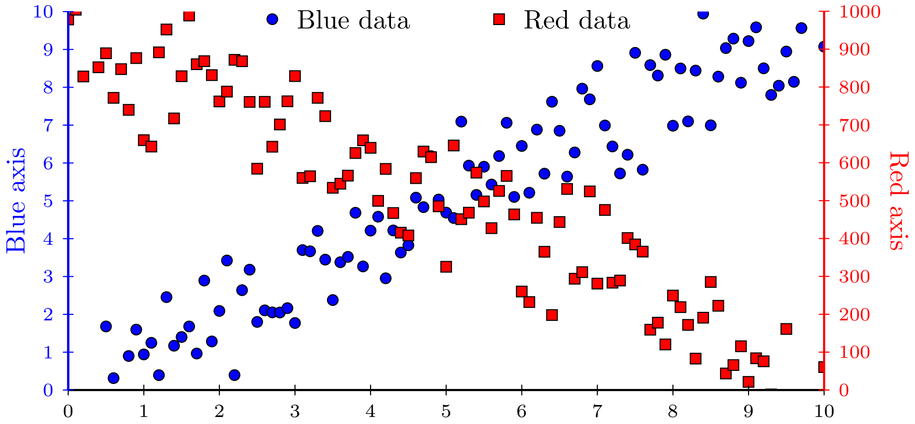

Figure 15: Two data sets with different transformations are plotted on top of ech other.

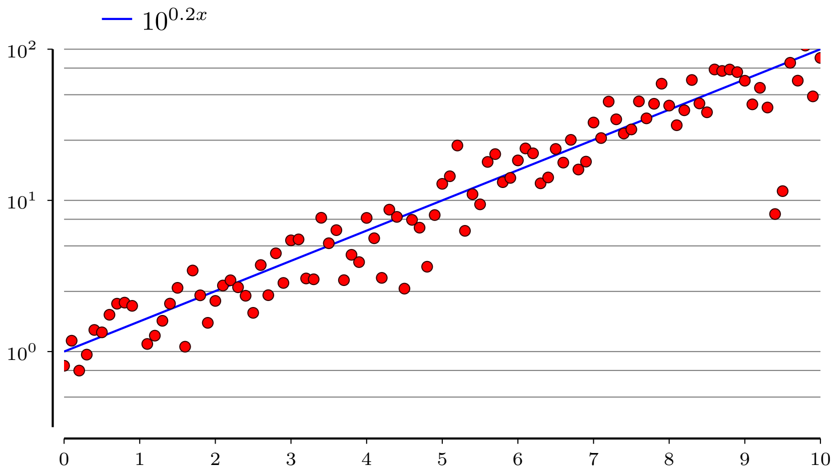

Figure 16: Plot with log scale in the y direction. Explicit transformation.

## 5 2D histograms

2D representations of 2D histograms.

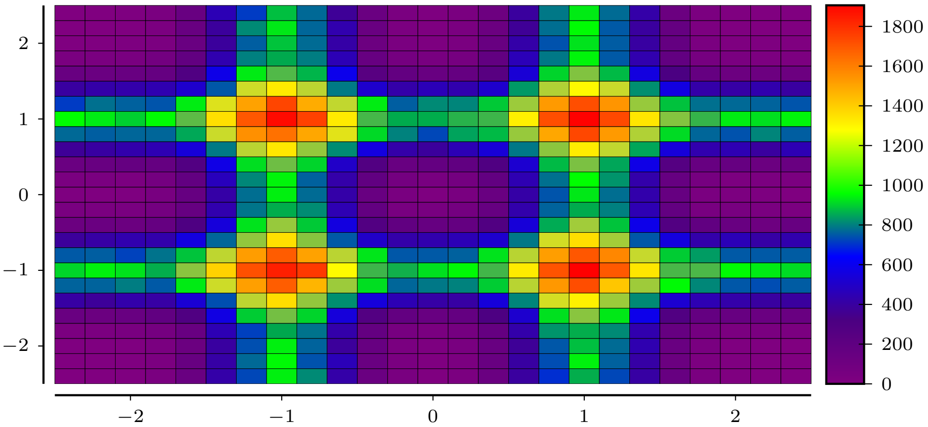

Figure 17: 2D histogram drawn as filled rectangles. Takes a while to compile with pdflatex, especially if the binning is fine.

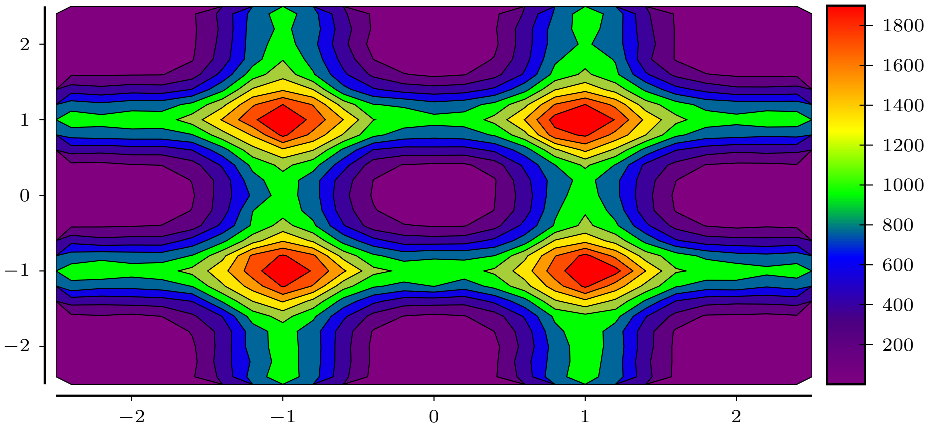

Figure 18: 2D histogram drawn as filled contour regions. The points making up the contour lines are just linear interpolation between neighbors on either side of the contour height.

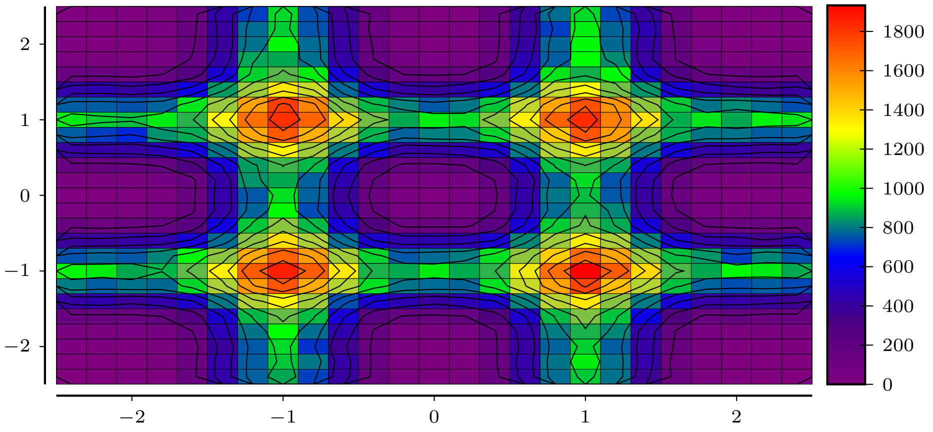

Figure 19: 2D histogram drawn as filled rectangles with contour lines.

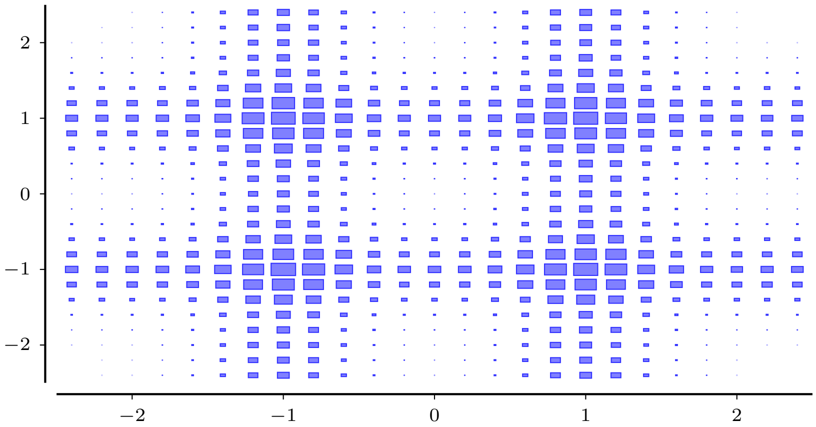

Figure 20: 2D histograms drawn as nodes of varying sizes.

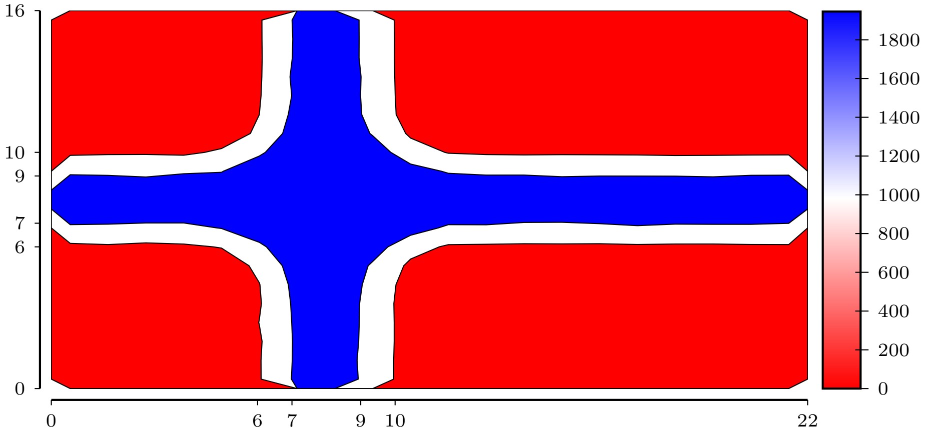

Figure 21: 2D histogram drawn as filled contour regions, not using rainbow colors, and using non uniformly distributed tick marks.