A VISUAL REFERENCE · PGFplots / TikZ

PGFplots / TikZ, one change at a time

From a first TikZ path to data-driven axes, statistics, 3D surfaces and publication layouts. Complete source. A small change. Its actual result.

The Org file is the canonical source. Read the HTML guide beside your work, or download its PDF and source for offline reading.

> **Last changes: 2026-10-31.** First complete edition: native TeX figures, focused parameter explanations, small synthetic inputs and the plot families familiar from the gnuplot tour.

96
worked examples

16
topic families

1
canonical Org source

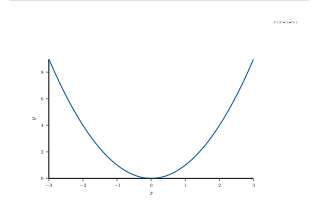
All figures at a glance

Choose a figure to open its PGFplots / TikZ source, explanation and downloads.

Compile your first plot

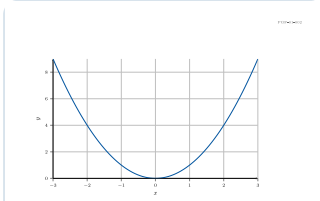
6 figures

PGFplots supplies numerical axes; TikZ supplies the picture. Each download is a standalone TeX document. Start with one curve, then change one visual choice.



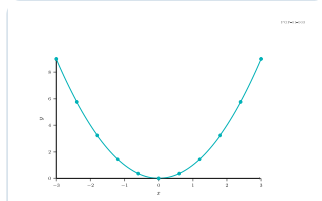
PGF-01-001

A quadratic on automatic axes



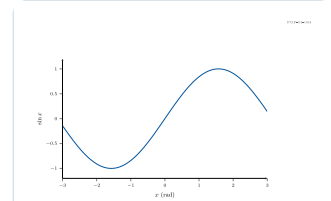
PGF-01-002

Add a major grid



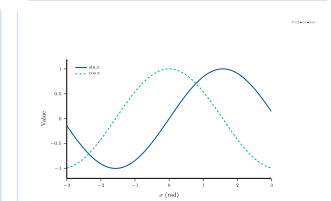
PGF-01-003

Change the line and markers



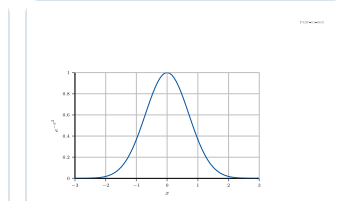
PGF-01-004

A sine curve with radian labels



PGF-01-005

Two functions and a legend



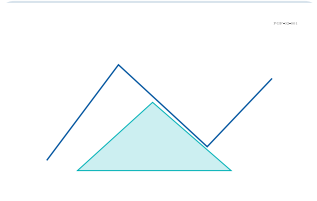
PGF-01-006

A measured-size figure

TikZ paths, shapes and fills

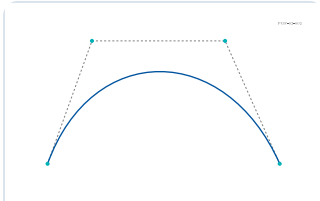
6 figures

TikZ works directly with paths and coordinates. These examples use one drawing unit per PostScript point, so distances are visible on a small fixed canvas.



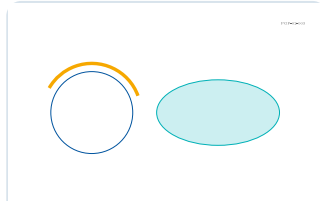
PGF-02-001

Lines and a closed polygon



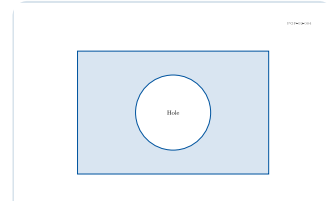
PGF-02-002

Bezier control points



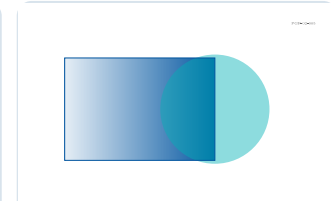
PGF-02-003

Circles, ellipses and arcs



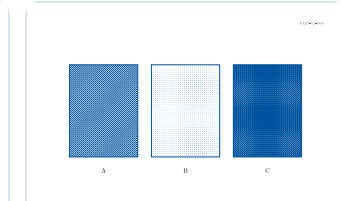
PGF-02-004

Fill rules and holes



PGF-02-005

Gradient and transparent overlap



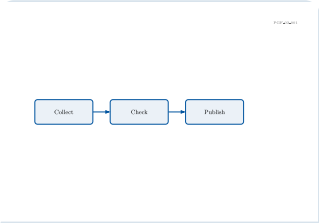
PGF-02-006

Patterned categorical regions

Nodes, arrows and diagrams

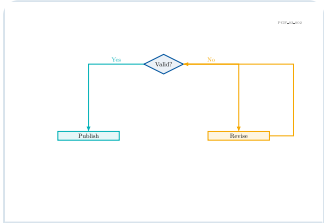
6 figures

Named nodes, anchors and relative placement are the building blocks of flowcharts and relationship diagrams. These drawings have no automatic graph-layout engine.



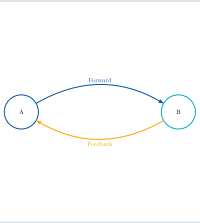
PGF-03-001

A three-step workflow



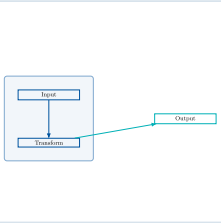
PGF-03-002

Branch and merge



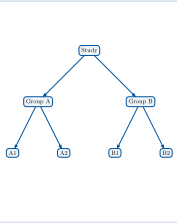
PGF-03-003

Curved feedback



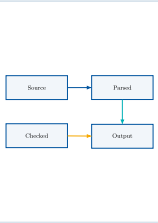
PGF-03-004

Fit a group boundary



PGF-03-005

A labelled tree



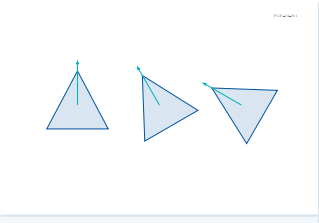
PGF-03-006

An aligned matrix of nodes

Coordinates, transformations and clipping

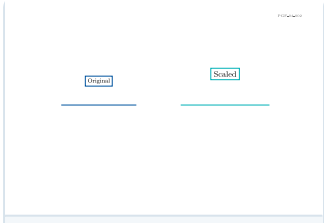
6 figures

Scopes apply local transformations. Coordinate calculations and clipping make constructions reusable without changing every point by hand.



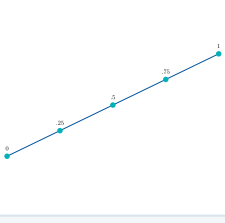
PGF-04-001

Translate and rotate a motif



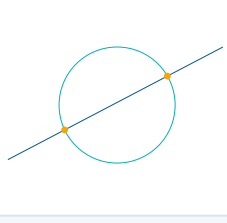
PGF-04-002

Scale coordinates and node shapes



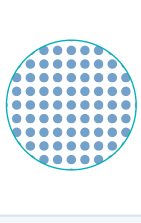
PGF-04-003

Interpolate between coordinates



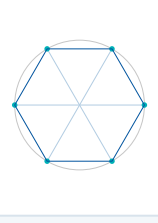
PGF-04-004

Named path intersections



PGF-04-005

Clip a repeated field



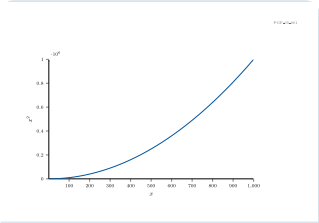
PGF-04-006

A polar coordinate construction

Axes, scales and ticks

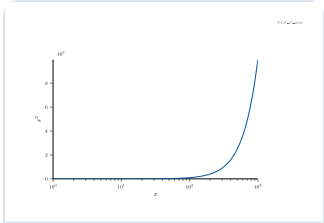
6 figures

Axes are part of the explanation. Keep units, ranges, transformations and tick labels explicit, especially when the visual distance is not linear in the original values.



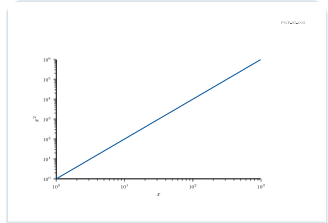
PGF-05-001

A linear reference frame



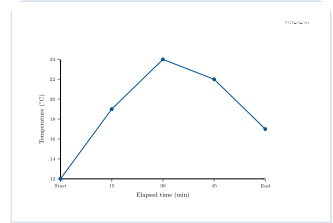
PGF-05-002

A logarithmic horizontal axis



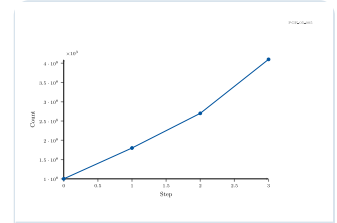
PGF-05-003

A log-log power law



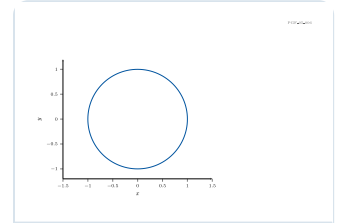
PGF-05-004

Custom ticks and units



PGF-05-005

Scientific notation and shared factors



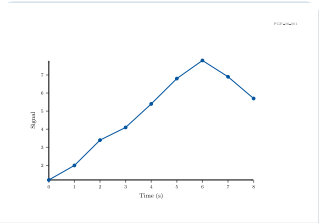
PGF-05-006

Equal coordinate units

Read and transform tabular data

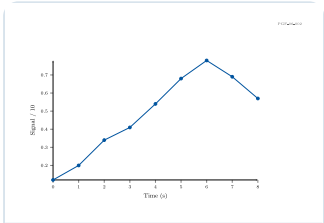
6 figures

The synthetic measurement table is source-owned and deliberately small. Columns have names, uncertainty values and categories. Every download declares and includes its input files.



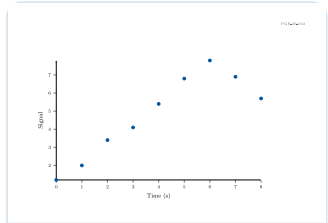
PGF-06-001

Plot named columns



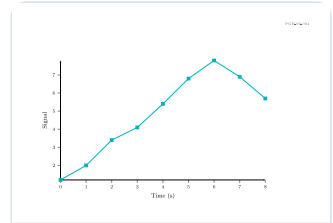
PGF-06-002

Transform a column expression



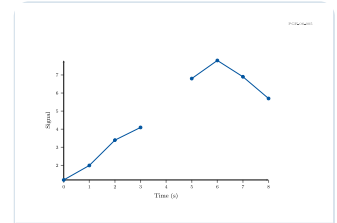
PGF-06-003

Skip a selected interval



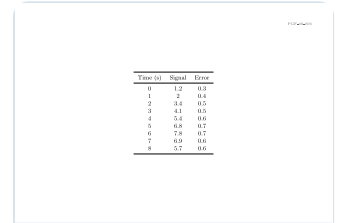
PGF-06-004

Read comma-separated input



PGF-06-005

Handle an explicit missing observation



Time (s)	Signal	Error
0	1.2	0.3
1	2	0.4
2	3.4	0.5
3	4.1	0.5
4	5.4	0.6
5	6.8	0.7
6	7.8	0.7
7	6.9	0.6
8	5.7	0.6

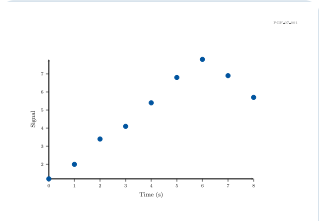
PGF-06-006

Typeset the input table

Scatter plots, uncertainty and annotations

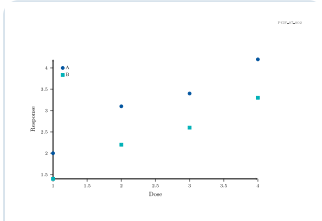
6 figures

Point appearance can encode a category or an additional variable. Error bars describe supplied uncertainty; PGFplots does not infer confidence intervals from these examples.



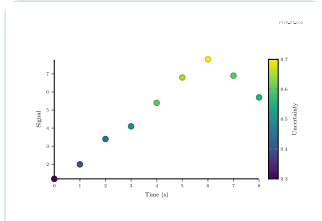
PGF-07-001

Markers without connecting lines



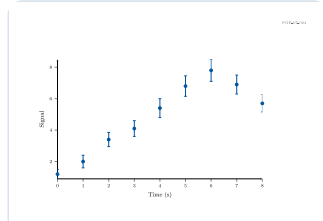
PGF-07-002

Encode categories with marker styles



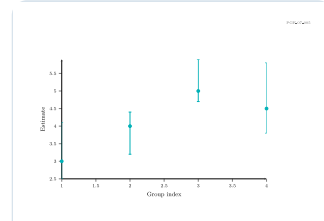
PGF-07-003

Colour points by a numeric value



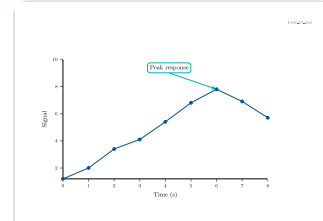
PGF-07-004

Vertical symmetric error bars



PGF-07-005

Asymmetric intervals



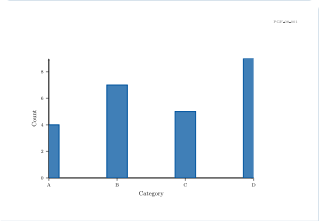
PGF-07-006

Annotate a selected observation

Bars and categorical comparisons

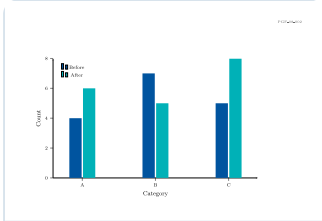
6 figures

Bar position, baseline and ordering carry meaning. These examples keep zero visible for absolute counts and state when values are stacked or normalized.



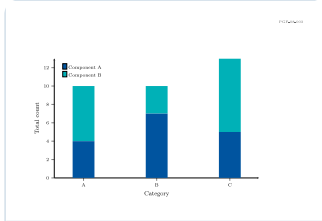
PGF-08-001

Vertical category bars



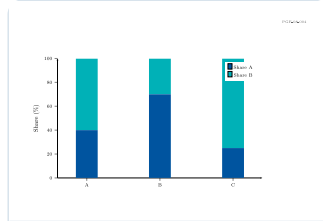
PGF-08-002

Grouped bars



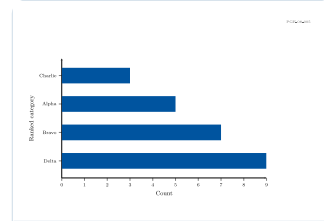
PGF-08-003

Stacked counts



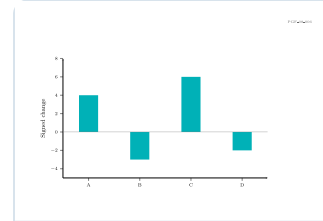
PGF-08-004

A normalized composition



PGF-08-005

Horizontal ranked bars



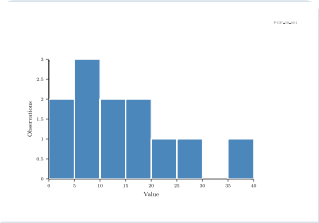
PGF-08-006

Diverging changes around zero

Distributions and statistical summaries

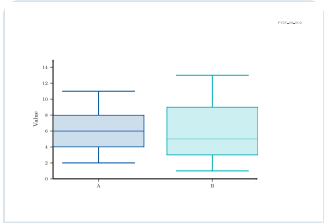
6 figures

State the estimator, bins and uncertainty definition. Small synthetic samples keep the drawing methods and statistical conventions visible.



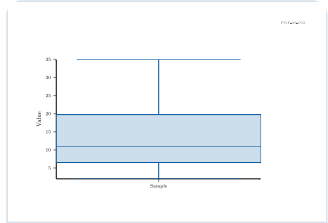
PGF-09-001

A histogram with fixed bin edges



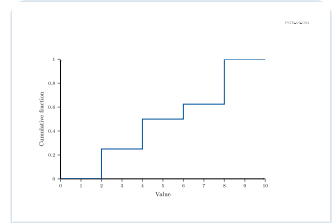
PGF-09-002

Prepared box-and-whisker summaries



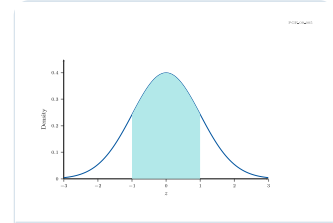
PGF-09-003

A boxplot calculated from samples



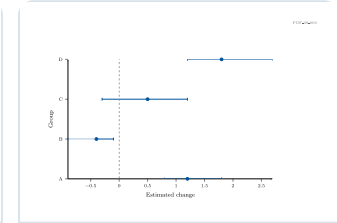
PGF-09-004

An explicit right-continuous ECDF



PGF-09-005

A density curve with a reference interval



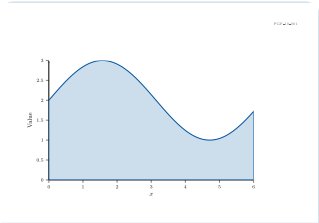
PGF-09-006

Point estimates with confidence intervals

Areas, bands and filled paths

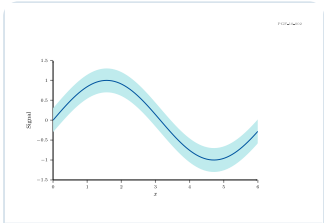
6 figures

Closed areas need a stated baseline. Bands between curves are distinct from stacked totals, and either can be drawn without rewriting the input values.



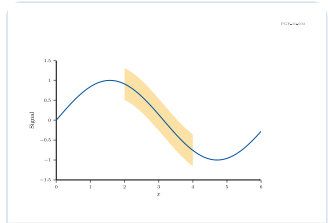
PGF-10-001

An area under a curve



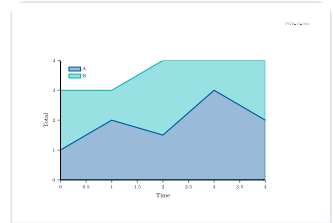
PGF-10-002

A band between named curves



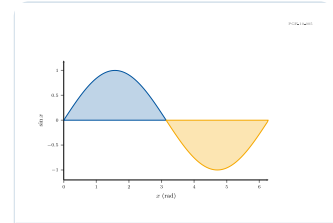
PGF-10-003

Restrict a band to an interval



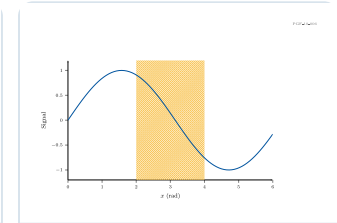
PGF-10-004

Stacked areas



PGF-10-005

Positive and negative filled regions



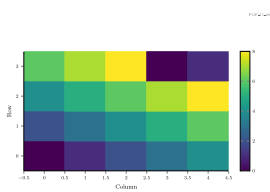
PGF-10-006

A hatched interval overlay

Heatmaps, colour scales and isolines

6 figures

Colour encodes a numeric variable only when its scale is stated. Smooth surfaces, cell-valued grids and analytic isolines describe different kinds of information.



PGF-11-001

An explicit grid of cell values



PGF-11-002

A smooth function viewed from above



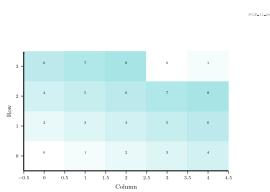
PGF-11-003

Choose discrete colour classes



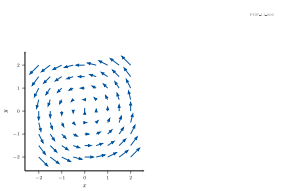
PGF-11-004

Overlay analytic isolines



PGF-11-005

Print numeric cell labels



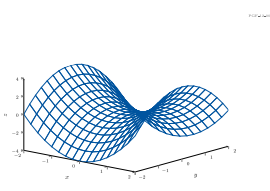
PGF-11-006

A directional vector field

Three-dimensional plots

6 figures

The camera and sampling grid affect how a surface is perceived. The native PDF is a fixed vector projection; it is not an interactive 3D object.



PGF-12-001

A wireframe saddle



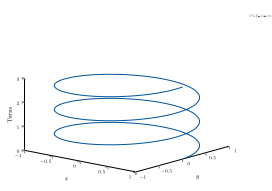
PGF-12-002

A smoothly shaded Gaussian surface



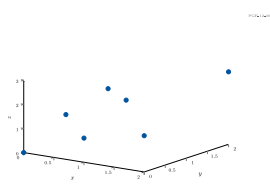
PGF-12-003

Change only the camera



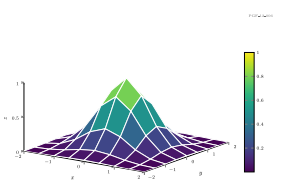
PGF-12-004

A parametric helix



PGF-12-005

Three-dimensional observations



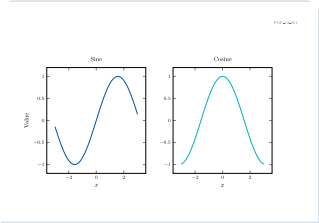
PGF-12-006

A faceted surface with visible patches

Multi-panel figures and shared guides

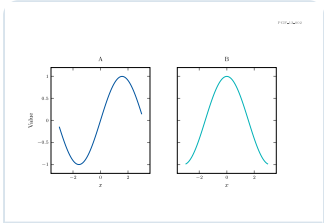
6 figures

Panels need comparable scales and deliberate spacing. These examples use groupplots or aligned axis rectangles, with a fixed overall canvas and explicit labels.



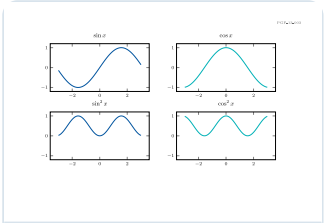
PGF-13-001

Two neighbouring panels



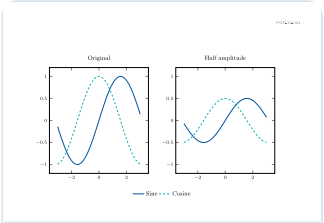
PGF-13-002

Keep labels on the outside



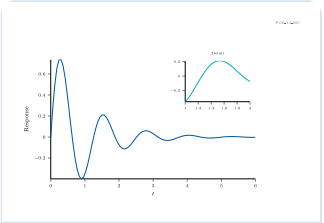
PGF-13-003

A two-by-two small-multiple figure



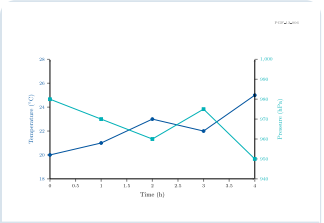
PGF-13-004

One shared legend



PGF-13-005

An inset with an explicit data window



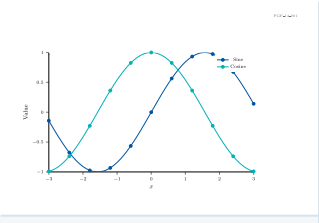
PGF-13-006

Two aligned unit systems

Styles, typography and export

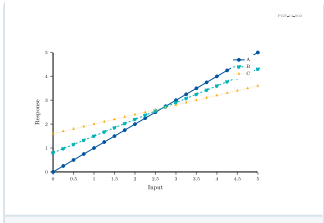
6 figures

Reusable styles make a collection consistent. Native figure PDFs are vector output with TeX typography; the publisher also provides Poppler-derived SVG and PNG previews for reuse.



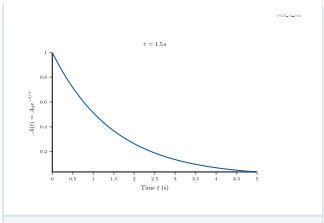
PGF-14-001

A reusable series style



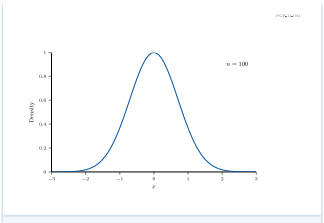
PGF-14-002

A coordinated cycle list



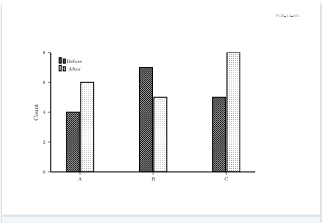
PGF-14-003

Mathematics in axis labels



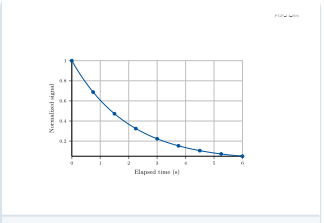
PGF-14-004

Place a note relative to the axis



PGF-14-005

A monochrome bar style



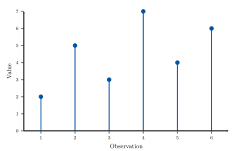
PGF-14-006

A compact figure for reuse

More plot types with small samples

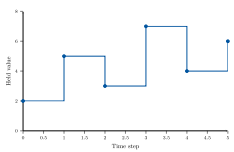
6 figures

Start with a handful of explicit values. Stem plots, steps, polar curves, bubbles, prepared hexagonal cells and ternary proportions show different encodings without a large dataset.



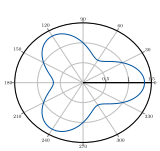
PGF-15-001

Discrete impulses with point markers



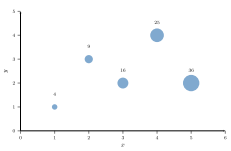
PGF-15-002

A step plot of held values



PGF-15-003

A polar radius function



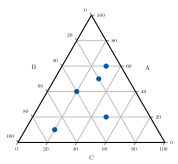
PGF-15-004

Bubble area from five values



PGF-15-005

Prepared hexagonal bin counts



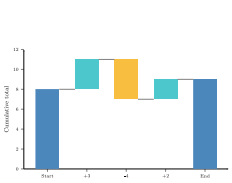
PGF-15-006

Ternary proportions

Specialized charts built from primitives

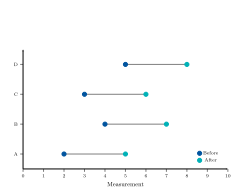
6 figures

A chart does not always need a dedicated plotting command. These small, explicit constructions show how TikZ paths and PGFplots axes can express specialized comparisons. All data in this chapter are synthetic.



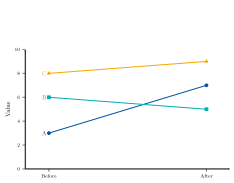
PGF-16-001

A waterfall of signed contributions



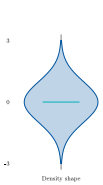
PGF-16-002

A dumbbell comparison



PGF-16-003

A slopegraph with direct labels



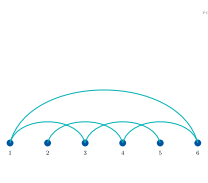
PGF-16-004

A mirrored density silhouette



PGF-16-005

A conserved two-way flow



PGF-16-006

An arc diagram of relationships

READING THE TOUR

Three ways to see a change

TeX source

Read the focused drawing commands and expand the complete standalone document.

Native figure

Compare actual PGFplots and TikZ output with the options and data that produced it.

Reproduce a figure

Compile from the command line or download a source-and-data bundle and its vector exports.

Read from the beginning or jump to a stable example ID. Each example includes its complete input and captured result. Copy link preserves the address of the guide you are reading, including its current domain and path.

From a TeX file to a figure

PGFplots and TikZ are TeX packages, not standalone plotting GUIs. TikZ draws paths and diagrams; PGFplots builds numerical axes and plots on top of TikZ. Compile a `.tex` file from a terminal, or let a LaTeX editor run the same compiler. A local TeX Live or MiKTeX installation needs the `standalone`, `pgfplots`, `pgfplotstable`, `amsmath` and `booktabs` packages, plus TikZ's standard libraries.

1. Download the first complete source bundle.
2. Unpack it and open a terminal in that directory.
3. Compile the document. A simple figure needs one pass; named legends and references need a second. The publisher always makes two passes.

```
pdflatex -no-shell-escape -interaction=nonstopmode -halt-on-error PGF-01-001.tex
pdflatex -no-shell-escape -interaction=nonstopmode -halt-on-error PGF-01-001.tex
```

Open `PGF-01-001.pdf`. The native PDF contains vector drawing instructions and TeX text. The `.aux` and `.log` files are normal local compiler products. They are not needed when sharing the finished PDF. The `-no-shell-escape` option keeps this compilation independent of external plotting commands.

The smallest working structure is a `standalone` document, a loaded `pgfplots` package, a `tikzpicture` and an `axis`. The full source of every gallery example includes its own preamble; it does not depend on an earlier example being run.

Read the coordinates and the options

The focused code isolates the drawing and axis choices. Expand **Complete standalone PGFplots / TikZ program** for the canvas, imports, colours and styles. `compat=1.18` selects a stable PGFplots behaviour level; it is not a request to install that historical package release. This edition records the installed PGFplots and PGF revisions in build provenance and prints them in native logs.

Direct TikZ examples use one coordinate unit per PostScript point (`bp`, 1/72 inch). PGFplots axes interpret coordinates as data values. `axis cs:` uses data coordinates, `rel axis cs:` uses fractions of the axis rectangle, and explicit dimensions such as `300bp` control physical layout. PGF math uses degrees for trigonometric functions: `sin(deg(x))` treats the plotted x as radians.

The examples use invented, bounded teaching data or explicit mathematical functions, beginning with one curve and a few coordinate pairs. The plot families follow the gnuplot tour, with TikZ drawing fundamentals added before more advanced chart construction. A named comparison links to an actual baseline. The drawing programs calculate no unstated statistical tests or causal conclusions.

Small inputs and reproducibility

The first chapters use functions and a few explicit coordinates. Later examples introduce nine invented measurements, four asymmetric intervals, twelve sample values or a small rectangular grid. Every value is visible in the source or in the input dictionary. These are teaching fixtures, not observations from a study. A function's `samples` option controls drawing resolution rather than the size of an observed dataset.

Run `python3 scripts/prepare-data.py --check` from the repository root to verify the fixture checksums, CSV/TSV equivalence and grid structure. This verification also runs before preparing native figures. Review both the values and the checksum manifest before changing an input.

Each **Source + data ZIP** contains the complete TeX document and every declared local input. Its figure PDF embeds the same source and inputs. The complete source and data kit is attached to the tour PDF for offline use. All teaching fixtures and original programs use CC0; the publishing framework retains its CC BY attribution.

Reuse a PDF, SVG or PNG

Use the native PDF when including a figure in another TeX document or sending vector artwork to print. A surrounding document can use `\includegraphics` with the `graphicx` package; avoid stretching width and height independently. The standalone document's border controls the padding around the figure.

The publisher provides SVG and PNG companions derived with Poppler. SVG text is outlined so labels travel with the figure; complex PDF shading can be represented differently by the converter. The native PDF is the reference for TeX typography and shading. PNG is a raster export, useful where vector formats are unsupported. For your own exports, run:

```
pdftocairo -svg PGF-01-001.pdf PGF-01-001.svg
pdftocairo -png -singlefile -r 180 PGF-01-001.pdf PGF-01-001
```

Compile the short standalone file while adjusting a plot, then reuse its PDF in a larger document. TikZ also has an externalization library for document workflows, but the figures in this tour compile without enabling shell escape.

Primary references and further work

The official PGFplots package page links the current manual for axes, plot styles, statistics, colour maps and the table library. The PGF/TikZ package page links its path, node, transformation and library reference. Installed copies are available with `texdoc pgfplots`, `texdoc pgfplotstable` and `texdoc pgf`.

For a separate comparison using common chart definitions across tools, open the data visualization comparison tour. For broader plotting and geometry examples, see the gnuplot, Asymptote, MetaPost and GMT tours.

Compile your first plot

PGFplots supplies numerical axes; TikZ supplies the picture. Each download is a standalone TeX document. Start with one curve, then change one visual choice.

PGF-01-001

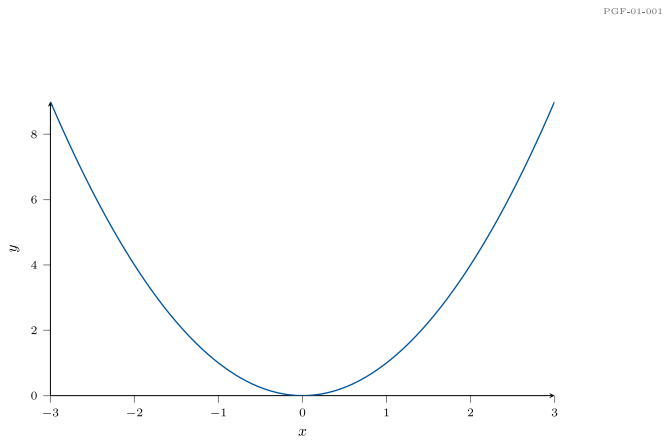
A quadratic on automatic axes

PGFplots evaluates a function on its domain and chooses ticks. The coordinates here are dimensionless.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={$x$},
  ylabel={$y$}
]
  \addplot[
    Blue, thick, no marks,
    domain=-3:3,
    samples=81
  ] {x^2};
\end{axis}
```

Actual output



PGF-01-001 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`domain=-3:3`	Evaluate x between minus three and three.
`x^2`	Square each sampled x coordinate.

PGF-01-002

Add a major grid

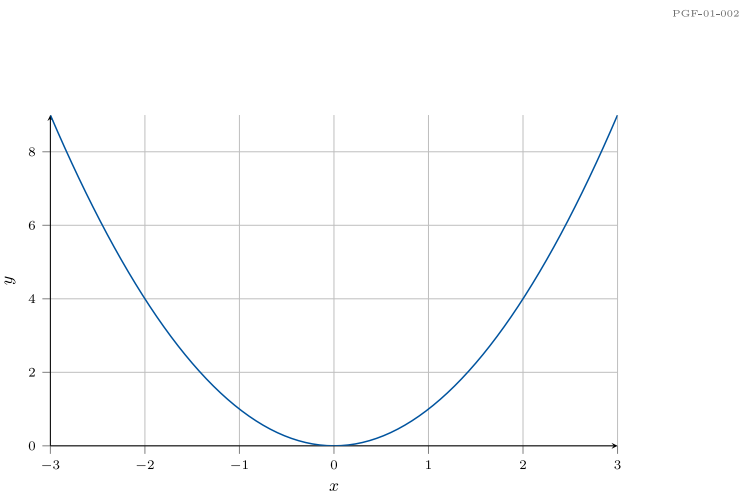
The function, samples and domain stay fixed. Major grid lines make values easier to read. Compare with PGF-01-001.

Compare with PGF-01-001 · A quadratic on automatic axes.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={$x$},
  ylabel={$y$},
  grid=major
]
  \addplot[
    Blue, thick, no marks,
    domain=-3:3,
    samples=81
  ] {x^2};
\end{axis}
```

Actual output



PGF-01-002 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`addplot`	Evaluate and draw the expression.
`samples=81`	Use a fixed, reproducible sampling grid.

PGF-01-003

Change the line and markers

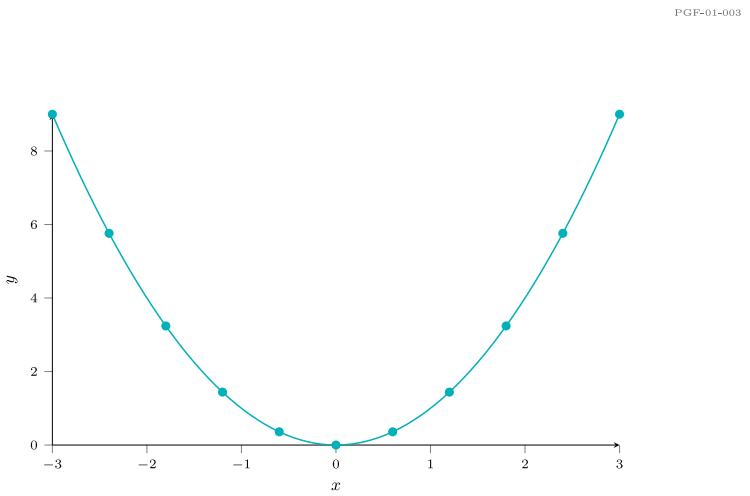
The same samples now have a contrasting line and visible marks. Marker spacing is controlled independently of the sample count. Compare with PGF-01-001.

Compare with PGF-01-001 · A quadratic on automatic axes.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={$x$},
  ylabel={$y$}
]
  \addplot[
    Teal, thick, mark=*, mark repeat=8,
    domain=-3:3,
    samples=81
  ] {x^2};
\end{axis}
```

Actual output



PGF-01-003 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`mark=*`	Place filled circular marks on the curve.
`mark repeat=8`	Show one mark for every eight samples.

PGF-01-004

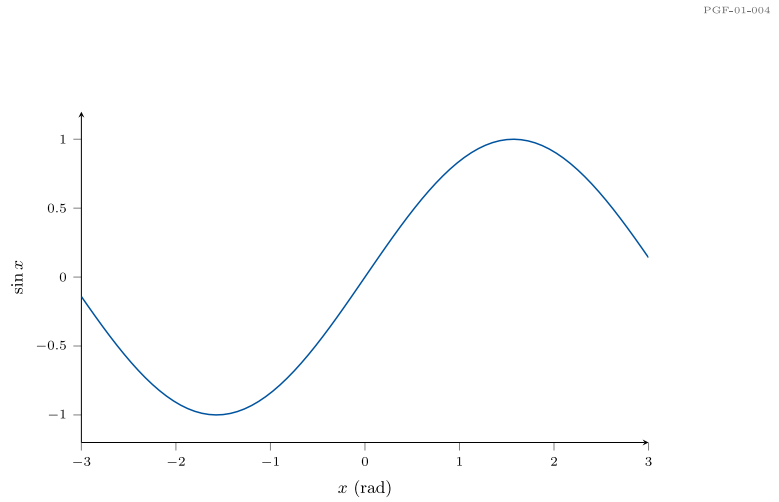
A sine curve with radian labels

PGF math trigonometric functions expect degrees. deg(x) converts the radian-valued x coordinate before evaluating sine.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={$x$ (rad)},
  ylabel={$\sin x$},
  ymin=-1.2,
  ymax=1.2
]
  \addplot[
    Blue, thick, no marks,
    domain=-3:3,
    samples=81
  ] {sin(deg(x))};
\end{axis}
```

Actual output



PGF-01-004 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`sin(deg(x))`	Convert radians to degrees for the PGF math evaluator.
`ymin / ymax`	Keep the vertical frame fixed at minus and plus 1.2.

PGF-01-005

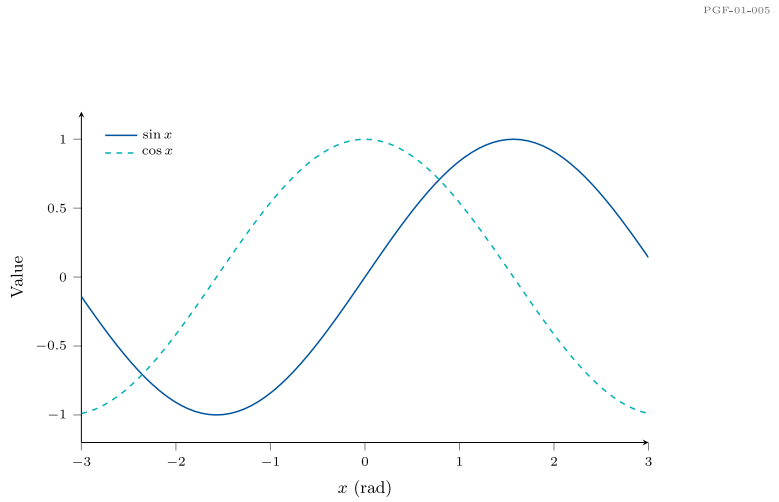
Two functions and a legend

Each addplot adds one series. The legend follows the series order; both curves share one set of axes.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={$x$ (rad)},
  ylabel={Value},
  ymin=-1.2,
  ymax=1.2,
  legend pos=north west
]
  \addplot[Blue, thick, domain=-3:3, samples=81] {sin(deg(x))};
  \addplot[Teal, thick, dashed, domain=-3:3, samples=81] {cos(deg(x))};
  \legend{$\sin x$, $\cos x$}
\end{axis}
```

Actual output



PGF-01-005 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
legend	Match legend entries to the plotted series in order.
dashed	Distinguish the second series by stroke as well as colour.

PGF-01-006

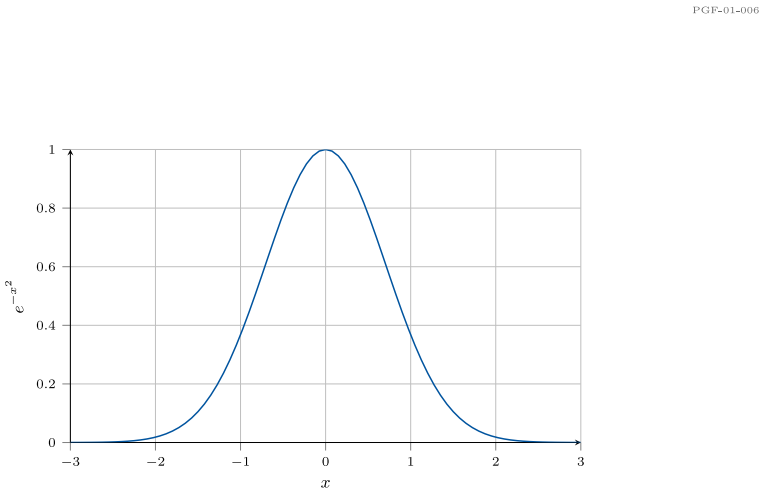
A measured-size figure

scale only axis gives the data rectangle an explicit size. Labels remain outside it; the standalone document supplies the surrounding figure bounds.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={$x$},
  ylabel={$e^{-x^2}$},
  width=270bp,
  height=155bp,
  scale only axis,
  grid=both
]
  \addplot[
    Blue, thick, no marks,
    domain=-3:3,
    samples=81
  ] {exp(-x*x)};
\end{axis}
```

Actual output



PGF-01-006 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`width / height`	Set the data rectangle in PostScript points.
`scale only axis`	Exclude labels from the specified axis dimensions.

TikZ paths, shapes and fills

TikZ works directly with paths and coordinates. These examples use one drawing unit per PostScript point, so distances are visible on a small fixed canvas.

PGF-02-001

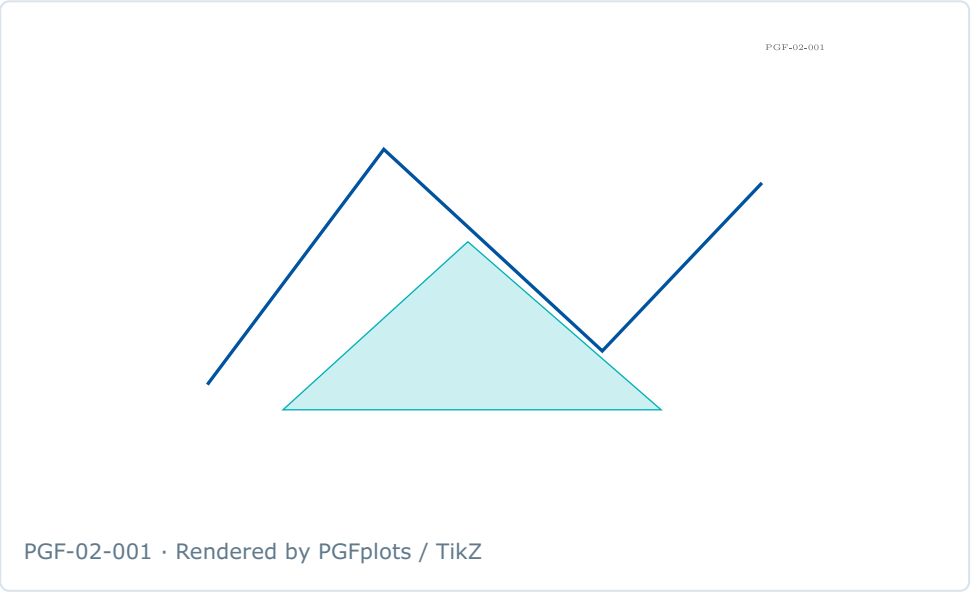
Lines and a closed polygon

Double hyphens join straight segments. cycle closes the last vertex back to the first.

PGFplots / TikZ drawing

```
\draw[Blue, line width=2pt] (45,70) -- (150,210) -- (280,90) -- (375,190);
\filldraw[fill=Teal!20, draw=Teal, thick]
  (90,55) -- (200,155) -- (315,55) -- cycle;
```

Actual output



PGF-02-001 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`--`	Connect successive points by straight segments.
`cycle`	Close the polygon before filling it.

PGF-02-002

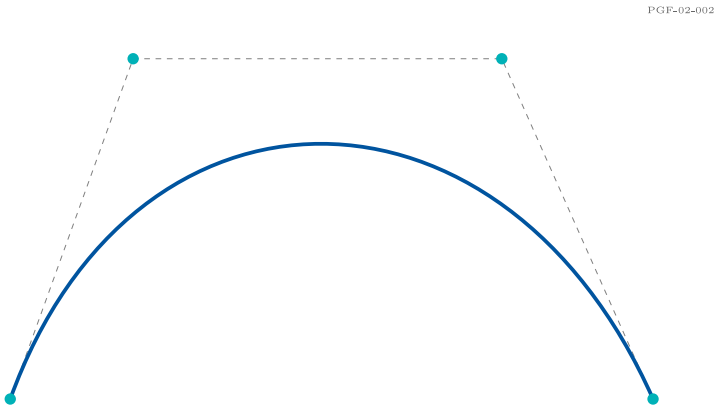
Bezier control points

Control points set a cubic curve tangent and bend. The dashed construction lines expose those controls.

PGFplots / TikZ drawing

```
\draw[gray, dashed] (40,65) -- (105,245) -- (300,245) -- (380,65);
\draw[Blue, line width=2pt] (40,65) .. controls (105,245) and (300,245) .. (380,65);
\foreach \p in {(40,65),(105,245),(300,245),(380,65)}
  \fill[Teal] \p circle[radius=3];
```

Actual output



PGF-02-002 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`controls ... and ...`	Specify the two cubic Bezier control points.
`dashed construction`	Show the control polygon without changing the curve.

PGF-02-003

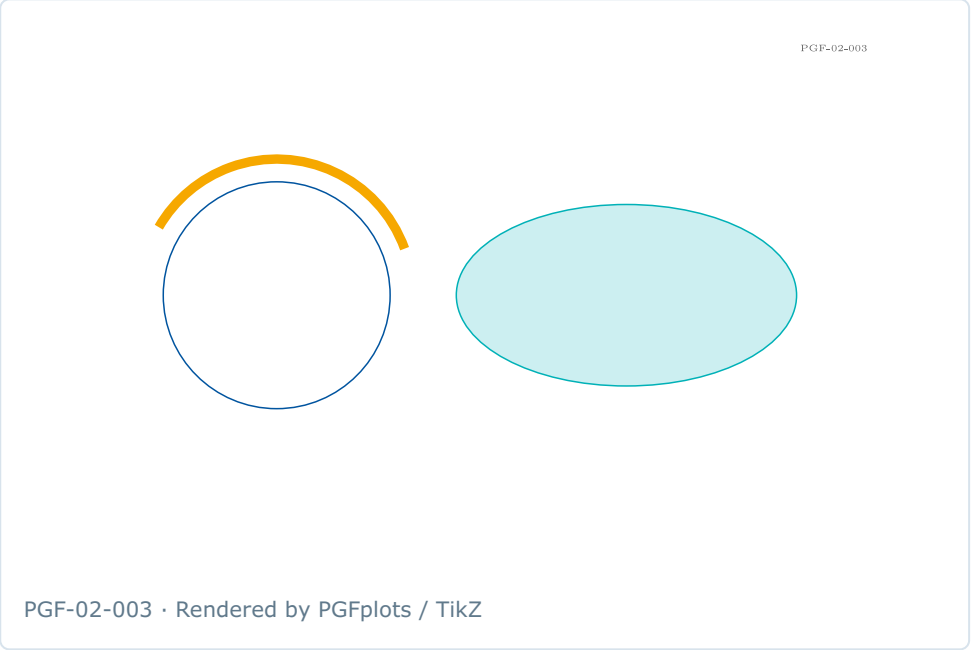
Circles, ellipses and arcs

TikZ can mix analytic shapes in one path-based drawing. Angles are in degrees.

PGFplots / TikZ drawing

```
\draw[Blue, thick] (100,140) circle[radius=60];
\filldraw[fill=Teal!20, draw=Teal, thick] (285,140) ellipse[x radius=90,y radius=48];
\draw[Orange, line width=5pt] (100,140) ++(20:72)
  arc[start angle=20,end angle=150,radius=72];
```

Actual output



PGF-02-003 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`circle / ellipse`	Choose radii in the picture coordinate system.
`arc`	Trace the specified angular portion of a circle.

PGF-02-004

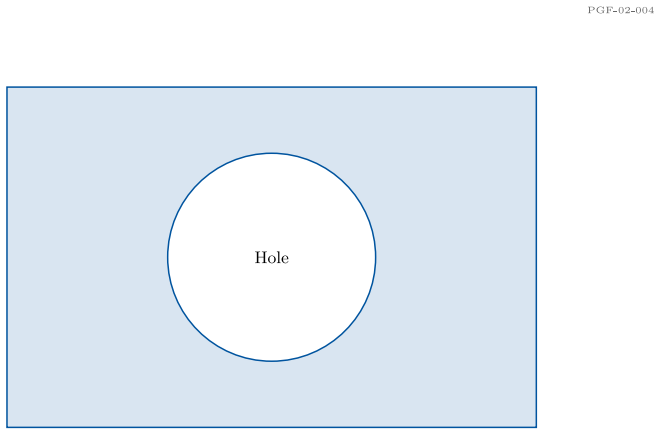
Fill rules and holes

Even-odd filling leaves a hole inside nested closed paths. The pattern illustrates the interior independently of the outline.

PGFplots / TikZ drawing

```
\path[fill=Blue!15, draw=Blue, thick, even odd rule]
  (70,50) rectangle (350,230)
  (210,140) circle[radius=55];
\node at (210,140) {Hole};
```

Actual output



PGF-02-004 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`even odd rule`	Alternate filled and unfilled regions across nested boundaries.
`nested circle`	Subtract the interior visually from the surrounding rectangle.

PGF-02-005

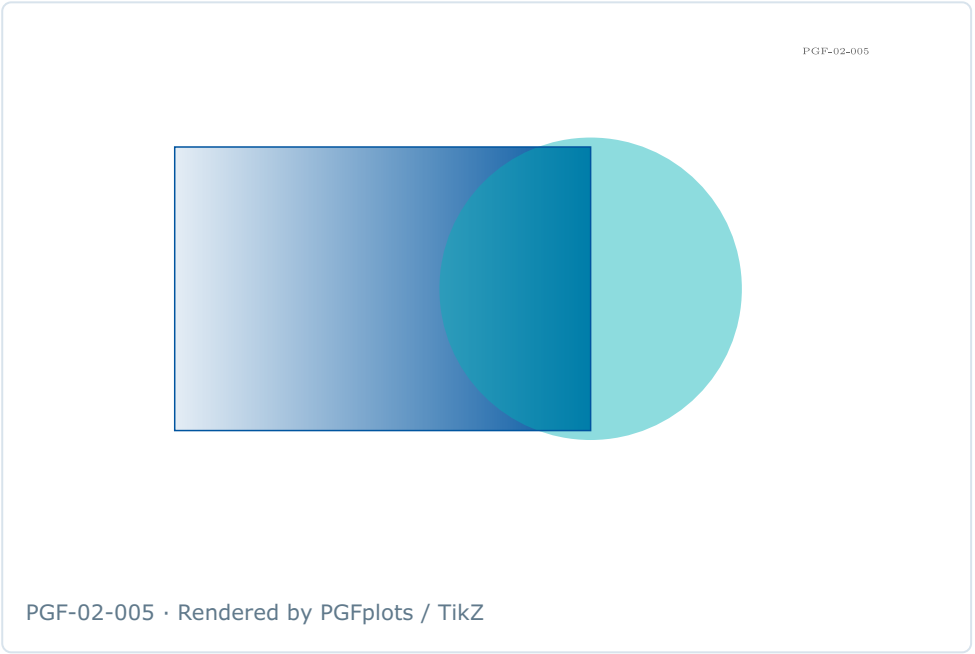
Gradient and transparent overlap

Shading controls colour inside a shape; opacity controls how overlapping shapes combine.

PGFplots / TikZ drawing

```
\shade[left color=Blue!10,right color=Blue] (45,70) rectangle (265,220);
\fill[Teal,opacity=.45] (265,145) circle[radius=80];
\draw[Blue,thick] (45,70) rectangle (265,220);
```

Actual output



PGF-02-005 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`left color / right color`	Interpolate a horizontal colour gradient.
`opacity=.45`	Mix the circle with the material behind it.

PGF-02-006

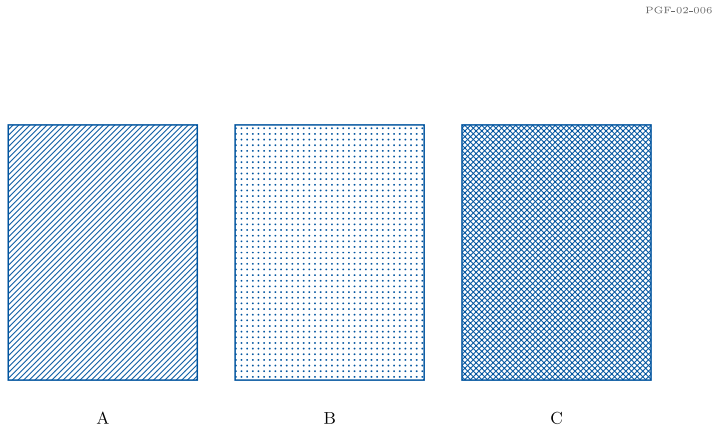
Patterned categorical regions

Patterns remain distinguishable when colours are similar or the figure is printed without colour.

PGFplots / TikZ drawing

```
\foreach \x/\p/\name in {40/north east lines/A,160/dots/B,280/crosshatch/C} {
  \path[pattern=\p,pattern color=Blue,draw=Blue,thick]
    (\x,75) rectangle ++(100,135);
  \node at (\x+50,55) {\name};
}
```

Actual output



PGF-02-006 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`pattern`	Select a repeatable hatch or dot fill.
`pattern color`	Set the pattern ink separately from the outline.

Nodes, arrows and diagrams

Named nodes, anchors and relative placement are the building blocks of flowcharts and relationship diagrams. These drawings have no automatic graph-layout engine.

PGF-03-001

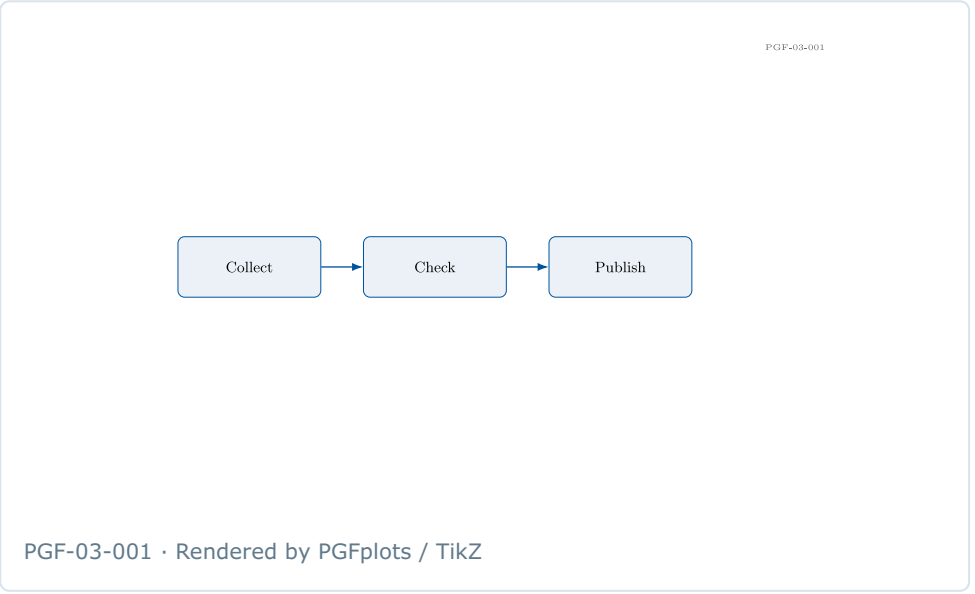
A three-step workflow

Relative positioning places each node beside its predecessor. Named anchors keep arrows attached when text sizes change.

PGFplots / TikZ drawing

```
\tikzset{tour step/.style={draw=Blue,fill=Blue!8,rounded corners,
  minimum width=85bp,minimum height=36bp,align=center}}
\node[tour step] (a) at (70,140) {Collect};
\node[tour step,right=25bp of a] (b) {Check};
\node[tour step,right=25bp of b] (c) {Publish};
\draw[-{Latex},thick,Blue] (a.east) -- (b.west);
\draw[-{Latex},thick,Blue] (b.east) -- (c.west);
```

Actual output



PGFplots / TikZ · exit 0

Command or parameter	Meaning
`right=... of`	Place a node relative to a named neighbour.
`east / west`	Attach arrows to the appropriate node boundaries.

PGF-03-002

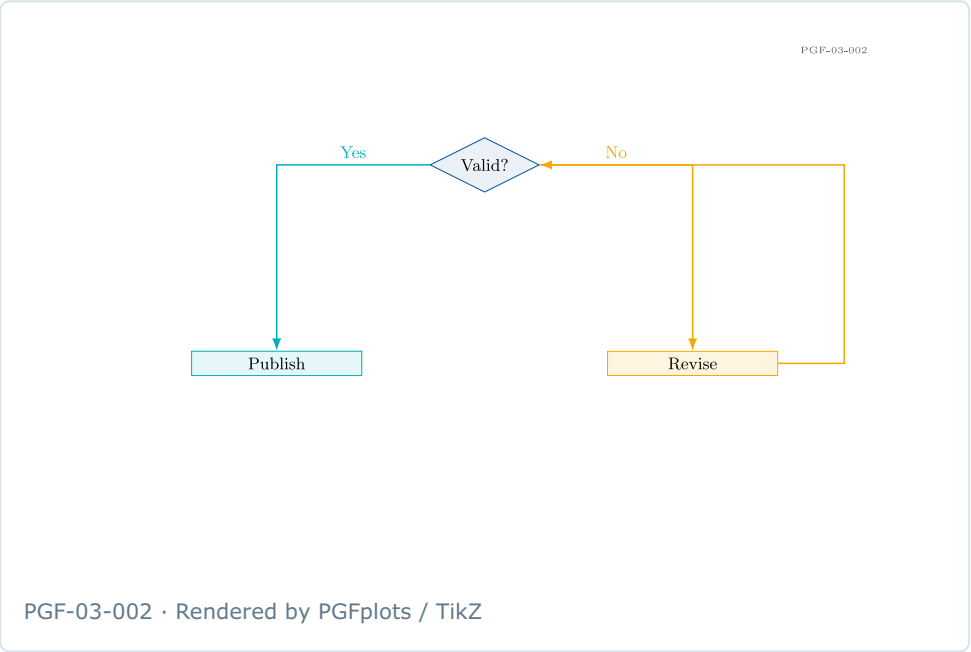
Branch and merge

A decision diamond creates two labelled routes. Orthogonal paths keep the connections away from the boxes.

PGFplots / TikZ drawing

```
\node[diamond,draw=Blue,fill=Blue!8,aspect=2] (q) at (210,210) {Valid?};
\node[draw=Teal,fill=Teal!10,minimum width=90bp] (a) at (100,105) {Publish};
\node[draw=Orange,fill=Orange!12,minimum width=90bp] (b) at (320,105) {Revise};
\draw[-{Latex},thick,Teal] (q) -| node[pos=.25,above] {Yes} (a);
\draw[-{Latex},thick,Orange] (q) -| node[pos=.25,above] {No} (b);
\draw[-{Latex},thick,Orange] (b.east) -- ++(35,0) |- (q.east);
```

Actual output



PGF-03-002 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`diamond`	Use a decision-shaped node.
`orthogonal paths`	Join horizontal and vertical segments at a right angle.

PGF-03-003

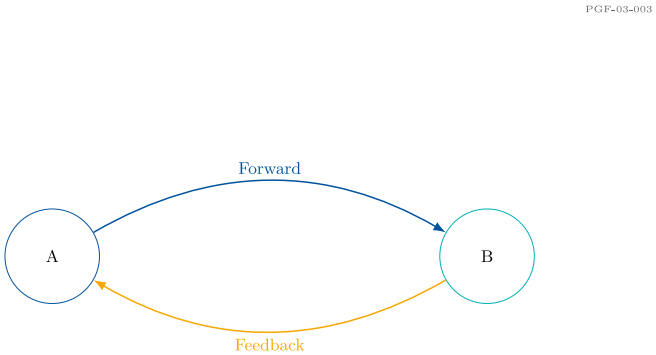
Curved feedback

Bending an edge distinguishes a feedback path from the forward direction. Edge labels can follow a sloped path.

PGFplots / TikZ drawing

```
\node[draw=Blue,circle,minimum size=50bp] (a) at (95,140) {A};
\node[draw=Teal,circle,minimum size=50bp] (b) at (325,140) {B};
\draw[-{Latex},thick,Blue] (a) to[bend left=30] node[above,sloped] {Forward} (b);
\draw[-{Latex},thick,Orange] (b) to[bend left=30] node[below,sloped] {Feedback} (a);
```

Actual output



PGF-03-003 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`bend left`	Bend the connection relative to its direction of travel.
`sloped`	Rotate the edge label to follow its segment.

PGF-03-004

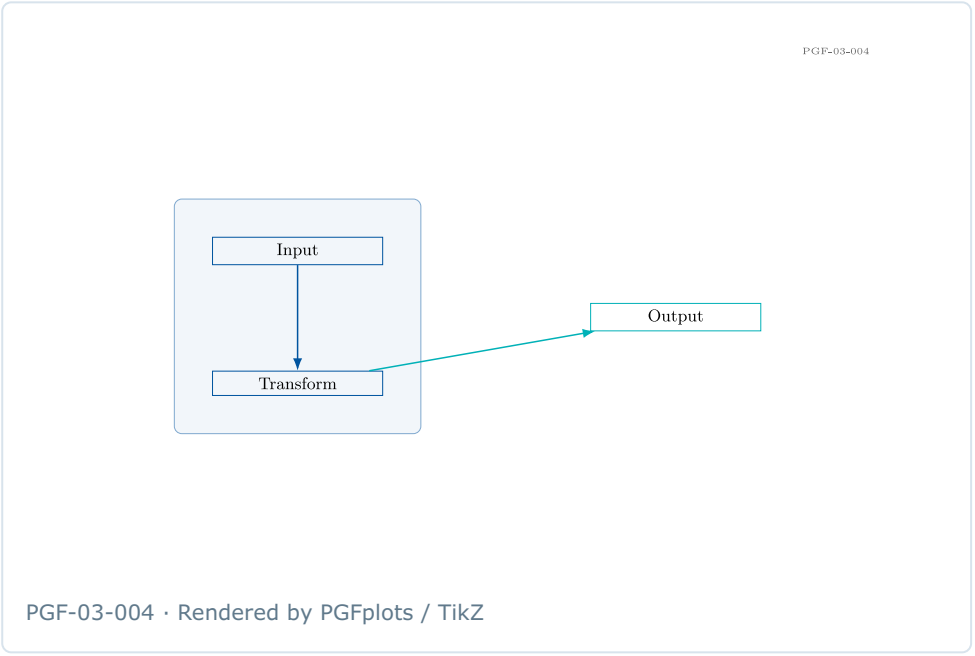
Fit a group boundary

The fit library computes a containing node around existing nodes. The background layer keeps that boundary behind its contents.

PGFplots / TikZ drawing

```
\node[draw=Blue,minimum width=90bp] (a) at (110,165) {Input};
\node[draw=Blue,minimum width=90bp] (b) at (110,95) {Transform};
\node[draw=Teal,minimum width=90bp] (c) at (310,130) {Output};
\begin{scope}[on background layer]
  \node[fit=(a)(b),draw=Blue!50,fill=Blue!5,rounded corners,inner sep=20bp] {};
\end{scope}
\draw[-{Latex},Blue,thick] (a) -- (b);
\draw[-{Latex},Teal,thick] (b) -- (c);
```

Actual output



PGF-03-004 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`fit=(a)(b)`	Calculate a bounding node around the named nodes.
`on background layer`	Draw the grouping box behind the foreground nodes.

PGF-03-005

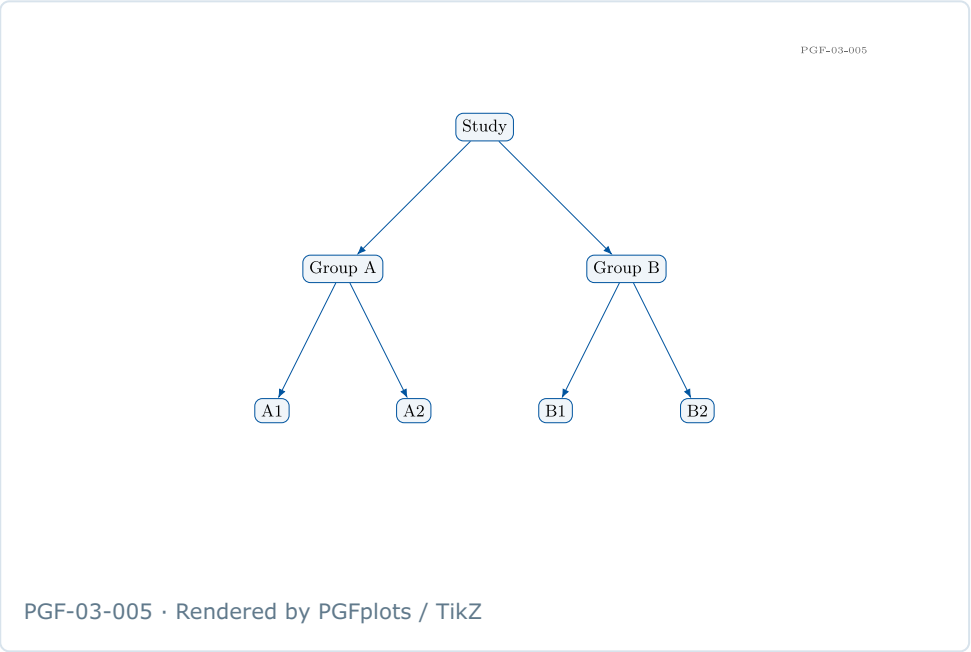
A labelled tree

Tree syntax declares children below a parent. Level distances and sibling distances control its geometry.

PGFplots / TikZ drawing

```
\begin{scope}[every node/.style={draw=Blue,rounded corners,fill=Blue!6},
  level distance=75bp,level 1/.style={sibling distance=150bp},
  level 2/.style={sibling distance=75bp},edge from parent/.style={draw=Blue,-{Latex}}]
\node at (210,230) {Study}
  child {node {Group A} child {node {A1}} child {node {A2}}}
  child {node {Group B} child {node {B1}} child {node {B2}}};
\end{scope}
```

Actual output



PGF-03-005 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`child`	Declare a child subtree.
`level / sibling distance`	Set the separation between generations and siblings.

PGF-03-006

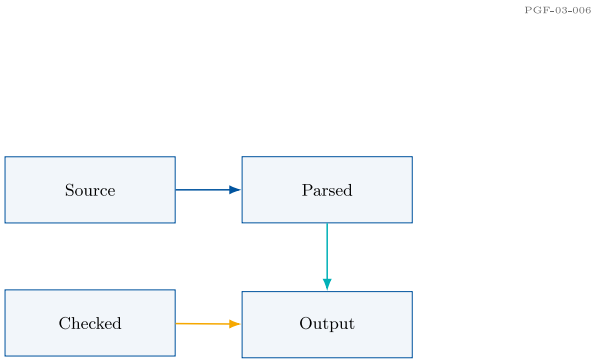
An aligned matrix of nodes

A matrix aligns heterogeneous labels in rows and columns. Named matrix cells can be connected directly.

PGFplots / TikZ drawing

```
\matrix (m) [matrix of nodes,nodes={draw=Blue,minimum width=90bp,
  minimum height=35bp,fill=Blue!5},column sep=35bp,row sep=35bp] at (210,140) {
  Source & Parsed \\
  Checked & Output \\
};
\draw[-{Latex},thick,Blue] (m-1-1) -- (m-1-2);
\draw[-{Latex},thick,Teal] (m-1-2) -- (m-2-2);
\draw[-{Latex},thick,Orange] (m-2-1) -- (m-2-2);
```

Actual output



PGF-03-006 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`matrix of nodes`	Create consistently aligned row and column cells.
`m-1-2`	Address a cell by matrix name, row and column.

Coordinates, transformations and clipping

Scopes apply local transformations. Coordinate calculations and clipping make constructions reusable without changing every point by hand.

PGF-04-001

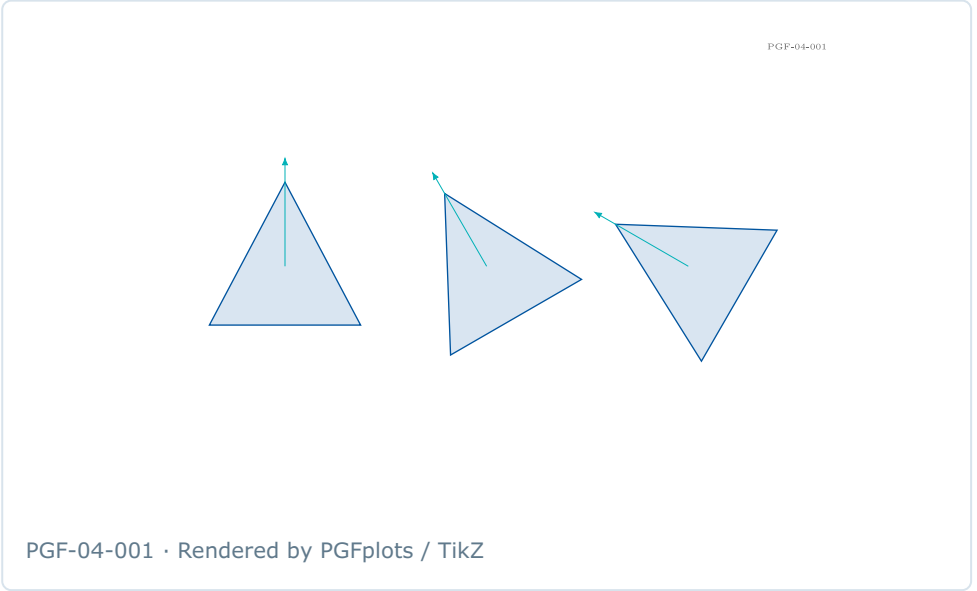
Translate and rotate a motif

The same triangle is translated and rotated within independent scopes. The original coordinate list stays unchanged.

PGFplots / TikZ drawing

```
\foreach \x/\angle in {90/0,210/30,330/60} {
  \begin{scope}[shift={(\x,140)},rotate=\angle]
    \filldraw[fill=Blue!15,draw=Blue,thick]
      (-45,-35) -- (45,-35) -- (0,50) -- cycle;
    \draw[Teal,-{Latex}] (0,0) -- (0,65);
  \end{scope}
}
```

Actual output



PGF-04-001 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`scope`	Limit transformations to one copy.
`shift / rotate`	Move the origin and rotate its coordinate frame.

PGF-04-002

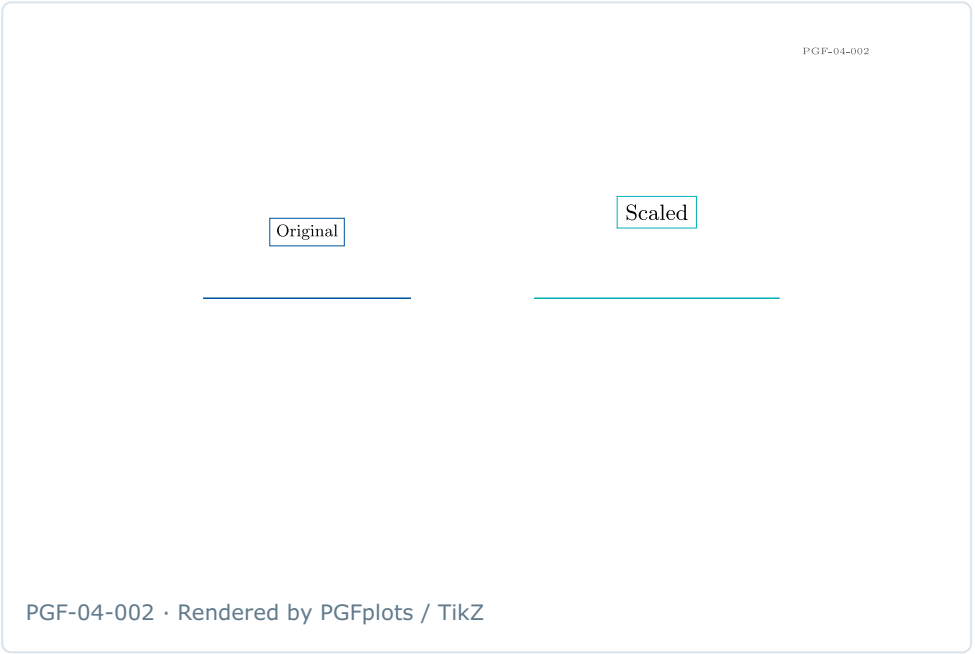
Scale coordinates and node shapes

Scaling coordinates changes distances; transform shape additionally scales text and node geometry. The two copies make the distinction visible.

PGFplots / TikZ drawing

```
\begin{scope}[shift={(60,140)}]
  \draw[Blue,thick] (0,0) -- (110,0);
  \node[draw=Blue] at (55,35) {Original};
\end{scope}
\begin{scope}[shift={(235,140)},scale=1.3,transform shape]
  \draw[Teal,thick] (0,0) -- (100,0);
  \node[draw=Teal] at (50,35) {Scaled};
\end{scope}
```

Actual output



PGFplots / TikZ · exit 0

Command or parameter	Meaning
`scale`	Scale the coordinate system.
`transform shape`	Apply that transformation to the node contents too.

PGF-04-003

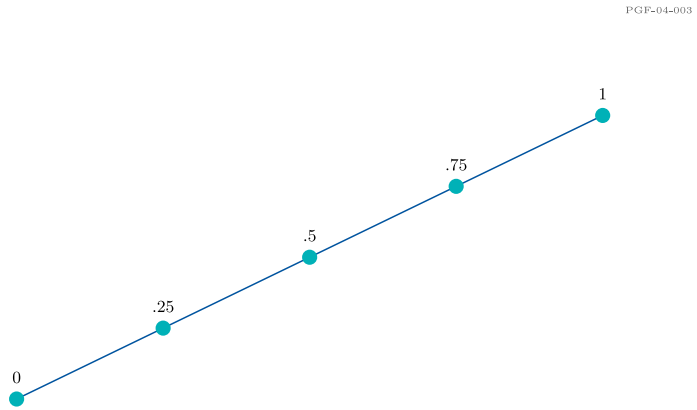
Interpolate between coordinates

The calc library computes points along a segment. Interpolated positions can host labels or perpendicular construction lines.

PGFplots / TikZ drawing

```
\coordinate (a) at (55,65);
\coordinate (b) at (365,215);
\draw[Blue,thick] (a) -- (b);
\foreach \t in {0,.25,.5,.75,1} {
  \fill[Teal] ($(a)!\t!(b)$) circle[radius=4];
  \node[above=5bp] at ($(a)!\t!(b)$) {\t};
}
```

Actual output



PGF-04-003 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`calc library`	Enable coordinate expressions inside dollar delimiters.
`interpolation fraction`	Choose a fraction of the distance from a to b.

PGF-04-004

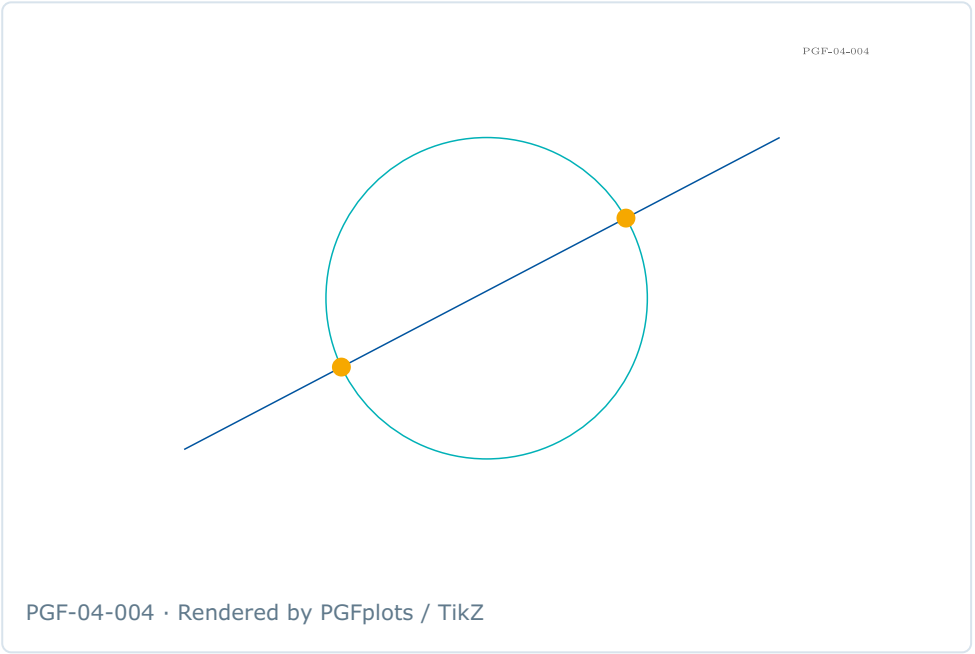
Named path intersections

Named paths can be queried for their intersection, avoiding a manually calculated crossing coordinate.

PGFplots / TikZ drawing

```
\path[name path=diagonal] (50,60) -- (365,225);
\path[name path=circle] (210,140) circle[radius=85];
\draw[Blue,thick] (50,60) -- (365,225);
\draw[Teal,thick] (210,140) circle[radius=85];
\path[name intersections={of=diagonal and circle,by={p,q}}];
\foreach \n in {p,q} \fill[Orange] (\n) circle[radius=5];
```

Actual output



PGF-04-004 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`name path`	Give a path a queryable name.
`name intersections`	Construct the crossing points of two named paths.

PGF-04-005

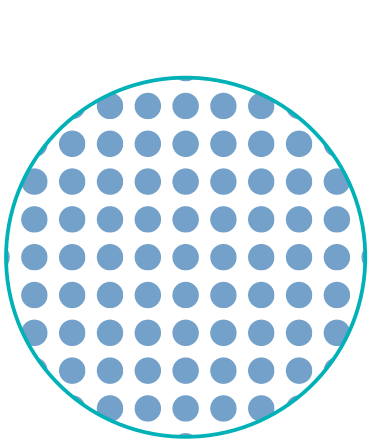
Clip a repeated field

Only material inside the clipping circle remains visible. The outline is drawn afterwards, outside the clipping scope.

PGFplots / TikZ drawing

```
\begin{scope}
  \clip (210,140) circle[radius=95];
  \foreach \x in {50,70,...,370}
    \foreach \y in {40,60,...,240}
      \fill[Blue!55] (\x,\y) circle[radius=7];
\end{scope}
\draw[Teal,line width=2pt] (210,140) circle[radius=95];
```

Actual output



PGF-04-005 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`clip`	Intersect subsequent drawing with a closed path.
`scope`	Restore unclipped drawing when the scope ends.

PGF-04-006

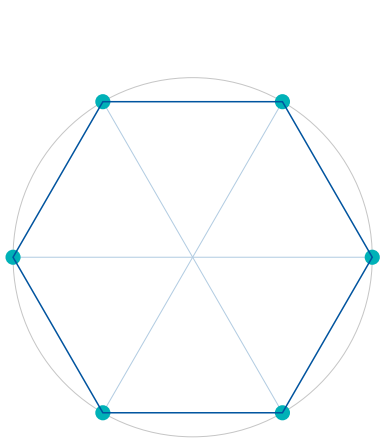
A polar coordinate construction

Angles are degrees and radii are picture units. A repeated polygon illustrates polar coordinates and a controlled origin.

PGFplots / TikZ drawing

```
\begin{scope}[shift={(210,140)}]
  \draw[gray!45] (0,0) circle[radius=95];
  \foreach \a in {0,60,...,300} {
    \draw[Blue!30] (0,0) -- (\a:95);
    \fill[Teal] (\a:95) circle[radius=4];
  }
  \draw[Blue,thick] (0:95) -- (60:95) -- (120:95) --
    (180:95) -- (240:95) -- (300:95) -- cycle;
\end{scope}
```

Actual output



PGF-04-006 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`angle:radius`	Specify a polar coordinate.
`shift`	Place the polar origin at the centre of the canvas.

Axes, scales and ticks

Axes are part of the explanation. Keep units, ranges, transformations and tick labels explicit, especially when the visual distance is not linear in the original values.

PGF-05-001

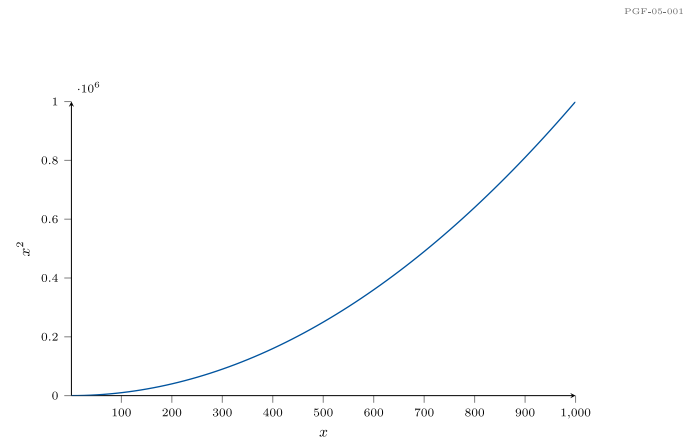
A linear reference frame

A linearly spaced axis is the reference for the logarithmic variations. Values are positive so both scale choices are valid.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xmin=1,
  xmax=1000,
  domain=1:1000,
  xlabel={$x$},
  ylabel={$x^2$}
]
  \addplot[
    Blue, thick, no marks,
    domain=1:1000,
    samples=81
  ] {x^2};
\end{axis}
```

Actual output



PGF-05-001 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`xmin / xmax`	Fix the horizontal range at one to one thousand.
`linear axis`	Equal horizontal distances represent equal differences.

PGF-05-002

A logarithmic horizontal axis

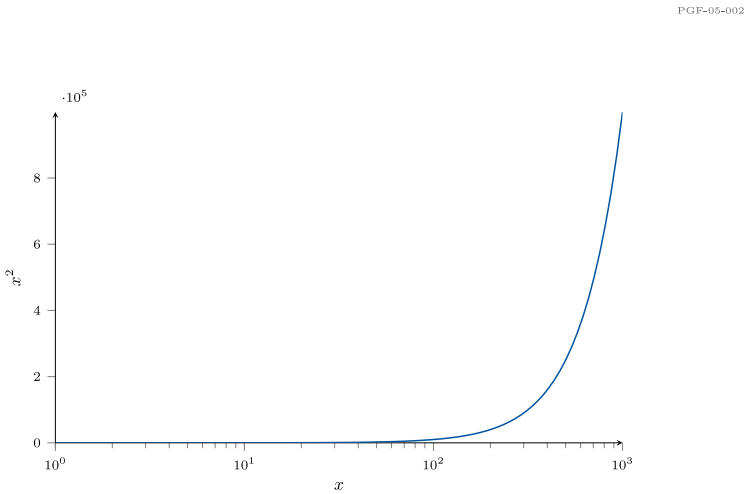
Logarithmic x spacing makes equal ratios occupy equal distances. Zero and negative x values are excluded. Compare with PGF-05-001.

Compare with PGF-05-001 · A linear reference frame.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xmode=log,
  log basis x=10,
  xmin=1,
  xmax=1000,
  xlabel={$x$},
  ylabel={$x^2$}
]
  \addplot[Blue,thick,domain=1:1000,samples=101] {x^2};
\end{axis}
```

Actual output



PGF-05-002 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`xmode=log`	Transform positive x coordinates logarithmically.
`log basis x=10`	Label the logarithmic axis in powers of ten.

PGF-05-003

A log-log power law

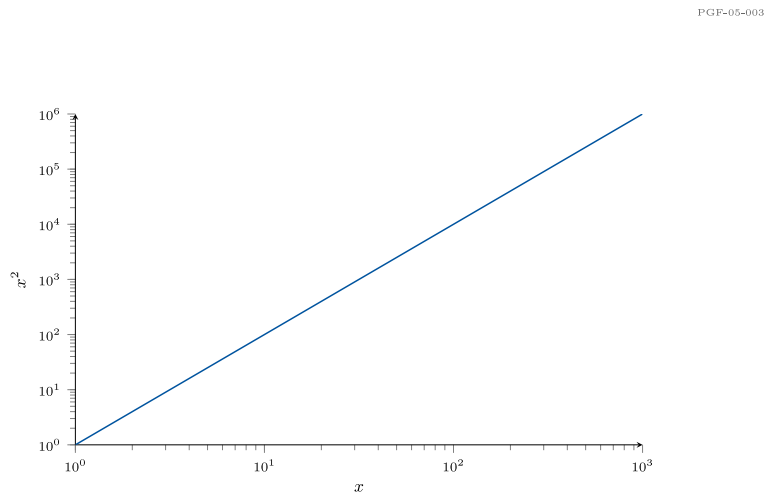
On logarithmic axes a power law becomes a straight line. The slope describes an exponent, not a change in original units. Compare with PGF-05-002.

Compare with PGF-05-002 · A logarithmic horizontal axis.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xmode=log,
  ymode=log,
  xmin=1,
  xmax=1000,
  xlabel={$x$},
  ylabel={$x^2$}
]
  \addplot[Blue,thick,domain=1:1000,samples=101] {x^2};
\end{axis}
```

Actual output



PGF-05-003 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`xmode / ymode`	Transform both axes to logarithmic coordinates.
`x^2`	A power law with exponent two.

PGF-05-004

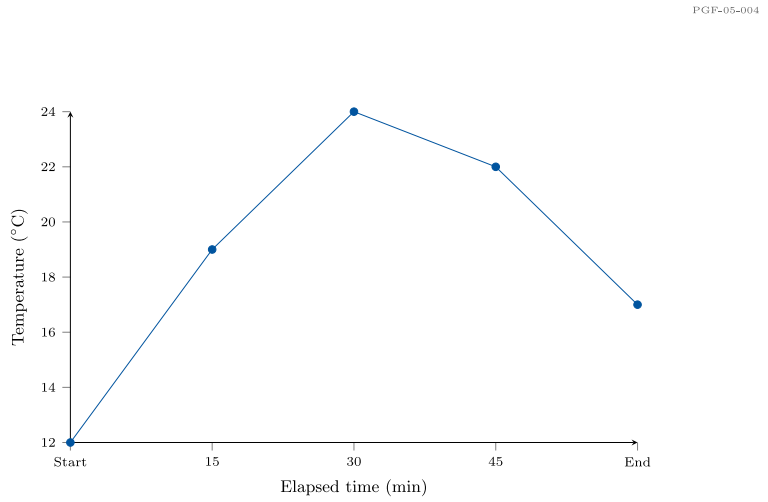
Custom ticks and units

Tick positions stay numeric while tick labels describe meaningful intervals. The axis label states the unit explicitly.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xtick={0,15,30,45,60},
  xticklabels={Start,15,30,45,End},
  xlabel={Elapsed time (min)},
  ylabel={Temperature ($^\circ$C)}
]
  \addplot[Blue,mark=*] coordinates {(0,12) (15,19) (30,24) (45,22) (60,17)};
\end{axis}
```

Actual output



PGF-05-004 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`xtick`	Choose the numeric tick coordinates.
`xticklabels`	Supply a corresponding display label for each tick.

PGF-05-005

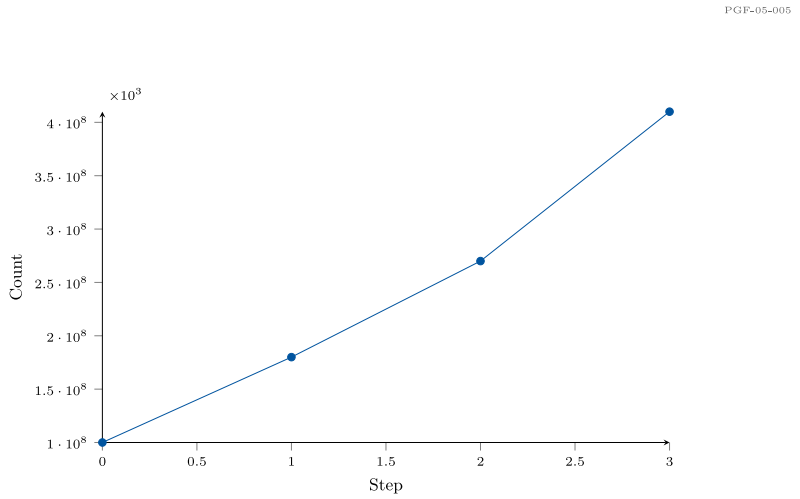
Scientific notation and shared factors

Large numeric values can share one displayed power-of-ten factor. The stored data remain in the original units.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={Step},
  ylabel={Count},
  scaled y ticks=base 10:3,
  ytick scale label code/.code={$\times 10^3$}
]
  \addplot[Blue,mark=*] coordinates {(0,100000) (1,180000) (2,270000) (3,410000)};
\end{axis}
```

Actual output



PGF-05-005 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`scaled y ticks`	Factor a common power of ten out of the tick labels.
`scale label`	Explain that the printed tick numbers are thousands.

PGF-05-006

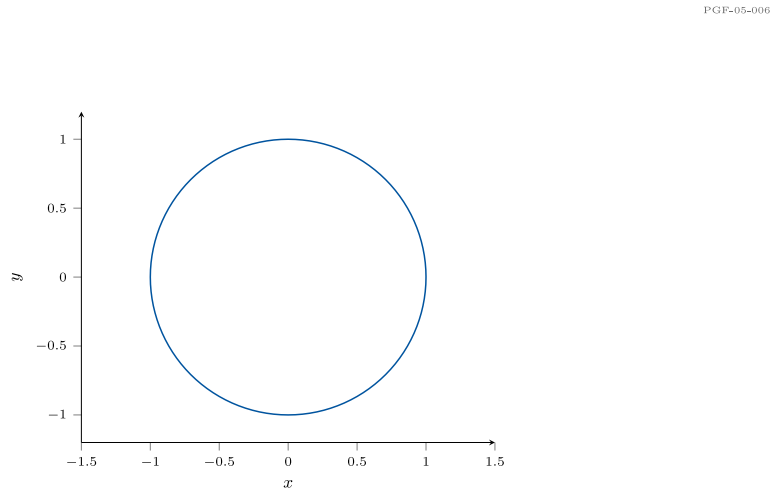
Equal coordinate units

axis equal image preserves the circle by using the same physical scale for x and y units.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  axis equal image,
  xmin=-1.5,
  xmax=1.5,
  ymin=-1.2,
  ymax=1.2,
  xlabel={$x$},
  ylabel={$y$}
]
  \addplot[Blue,thick,domain=0:360,samples=121] ({cos(x)},{sin(x)});
\end{axis}
```

Actual output



PGF-05-006 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`axis equal image`	Preserve geometric aspect without stretching the circle.
`parametric coordinates`	Compute x and y from one angular parameter.

Read and transform tabular data

The synthetic measurement table is source-owned and deliberately small. Columns have names, uncertainty values and categories. Every download declares and includes its input files.

PGF-06-001

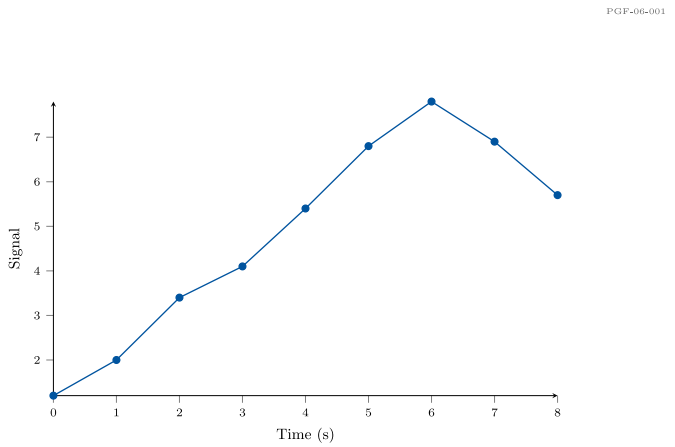
Plot named columns

A header names the x and y columns. Changing column order in the file does not change which variables are plotted.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={Time (s)},
  ylabel={Signal}
]
  \addplot[Blue,thick,mark=*] table[x=time,y=value]{data/measurements.tsv};
\end{axis}
```

Actual output



PGF-06-001 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`table[x=time,y=value]`	Select columns by their header names.
`TSV input`	Read whitespace-separated numeric values from the bundled table.

PGF-06-002

Transform a column expression

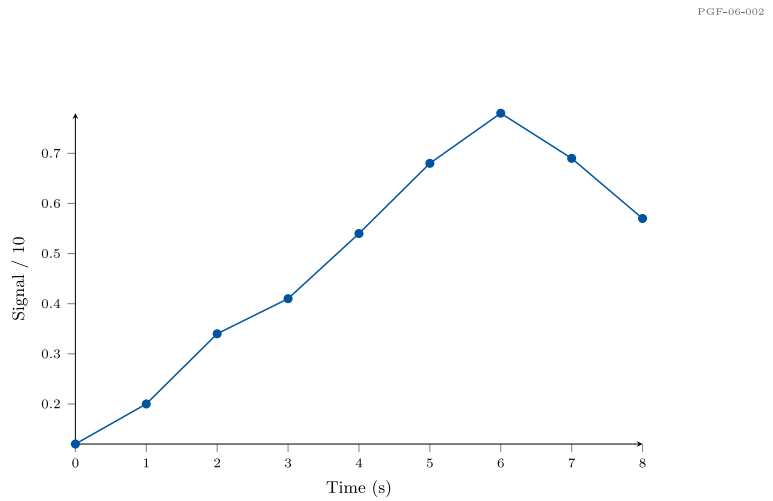
A table expression changes the plotted quantity without rewriting the input. Here the response is normalized to the reference value ten. Compare with PGF-06-001.

Compare with PGF-06-001 · Plot named columns.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={Time (s)},
  ylabel={Signal / 10}
]
  \addplot[Blue,thick,mark=*] table[x=time,y expr=\thisrow{value}/10]{data/
measurements.tsv};
\end{axis}
```

Actual output



PGF-06-002 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`y expr`	Evaluate an expression for every table row.
`thisrow`	Refer to a named column in the current row.

PGF-06-003

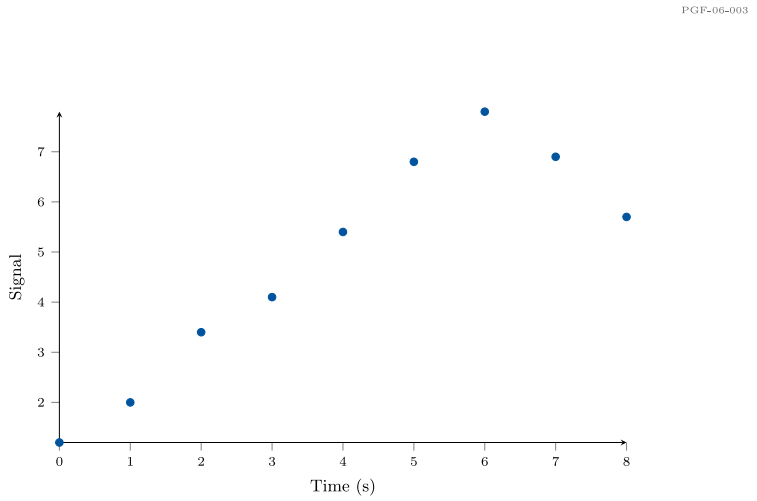
Skip a selected interval

A row predicate removes the middle observations from this view. The source table and the remaining coordinates are unchanged.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={Time (s)},
  ylabel={Signal}
]
  \addplot[Blue,only marks] table[
    x=time,y=value,
    row predicate/.code={\ifnum\pgfplotstablerow>2\relax
      \ifnum\pgfplotstablerow<6\relax\pgfplotstableuserowfalse\fi\fi}
  ]{data/measurements.tsv};
\end{axis}
```

Actual output



PGF-06-003 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`row predicate`	Choose whether each row enters the plotted series.
`pgfplotstablerow`	Use the zero-based input row index in the predicate.

PGF-06-004

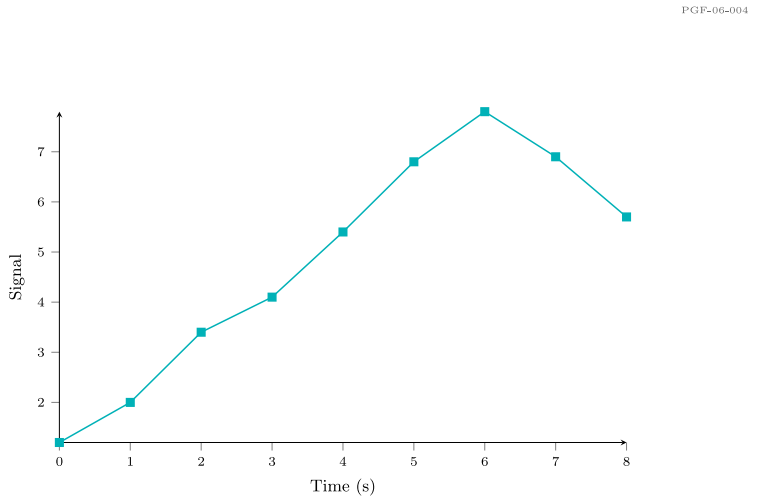
Read comma-separated input

col sep=comma changes the input parser; named-column selection works as in the TSV example.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={Time (s)},
  ylabel={Signal}
]
  \addplot[Teal,thick,mark=square*] table[col sep=comma,x=time,y=value]{data/
measurements.csv};
\end{axis}
```

Actual output



PGF-06-004 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`col sep=comma`	Read a CSV file with a header.
`x / y`	Choose the named input columns.

PGF-06-005

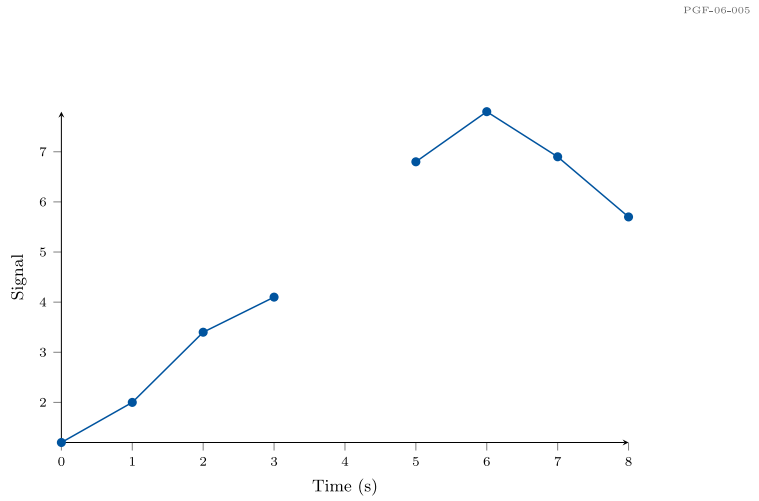
Handle an explicit missing observation

An unavailable value is written as nan. jump leaves a visible gap instead of connecting across the missing record.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  unbounded coords=jump,
  xlabel={Time (s)},
  ylabel={Signal}
]
  \addplot[Blue,thick,mark=*] table[x=time,y=value]{data/missing.tsv};
\end{axis}
```

Actual output



PGF-06-005 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
nan	Mark a missing numeric value explicitly.
unbounded coords=jump	Break the line at an unavailable coordinate.

PGF-06-006

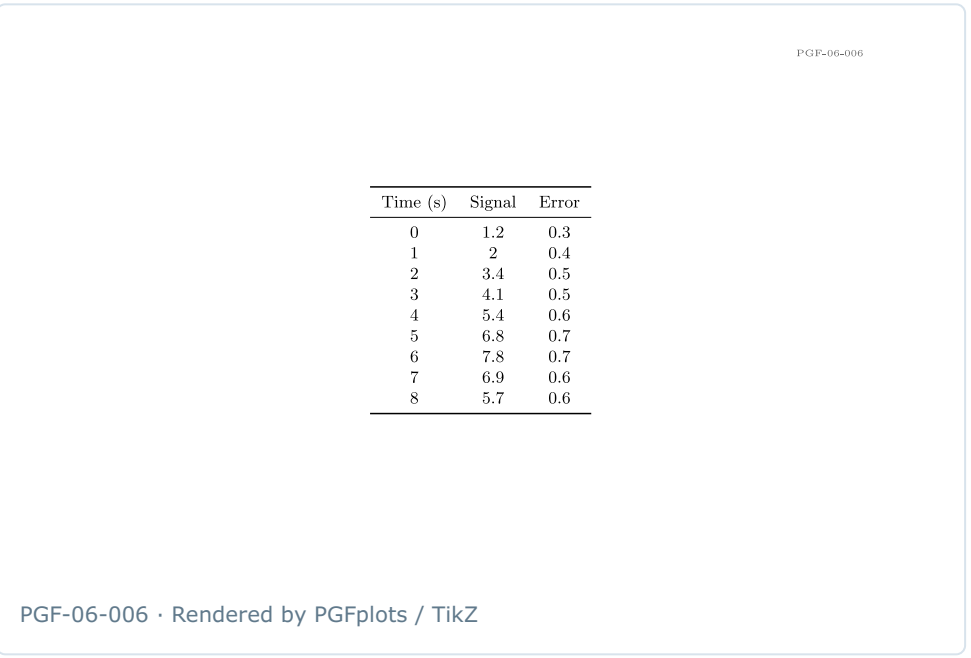
Typeset the input table

PGFplotsTable can present the same source table as a TeX table. Number formatting changes its display, not its stored values.

PGFplots / TikZ drawing

```
\node at (210,140) {\pgfplotstabletypeset[
  columns={time,value,error},
  columns/time/.style={column name={Time (s)},fixed,precision=0},
  columns/value/.style={column name={Signal},fixed,precision=1},
  columns/error/.style={column name={Error},fixed,precision=1},
  every head row/.style={before row=\toprule,after row=\midrule},
  every last row/.style={after row=\bottomrule}
]{data/measurements.tsv}};
```

Actual output



PGFplots / TikZ · exit 0

Command or parameter	Meaning
`pgfplotstabletypeset`	Create a TeX table directly from the input file.
`column style`	Choose display names, notation and precision independently.

Scatter plots, uncertainty and annotations

Point appearance can encode a category or an additional variable. Error bars describe supplied uncertainty; PGFplots does not infer confidence intervals from these examples.

PGF-07-001

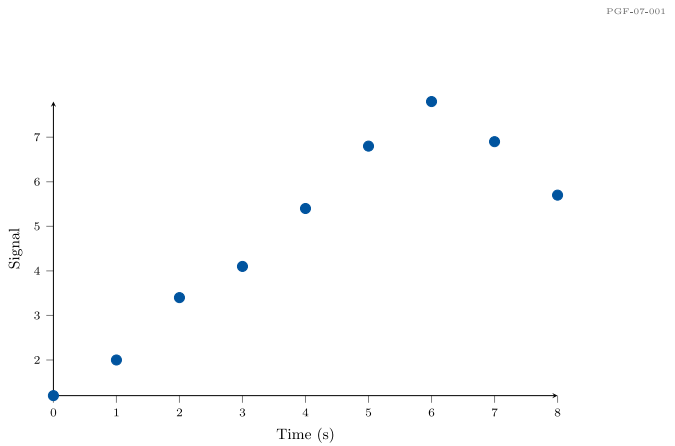
Markers without connecting lines

A scatter plot presents individual observations without implying interpolation between them.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={Time (s)},
  ylabel={Signal}
]
  \addplot[Blue,only marks,mark=*,mark size=3pt] table[x=time,y=value]{data/
measurements.tsv};
\end{axis}
```

Actual output



PGF-07-001 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`only marks`	Suppress line segments between records.
`mark size`	Set the visible symbol size in points.

PGF-07-002

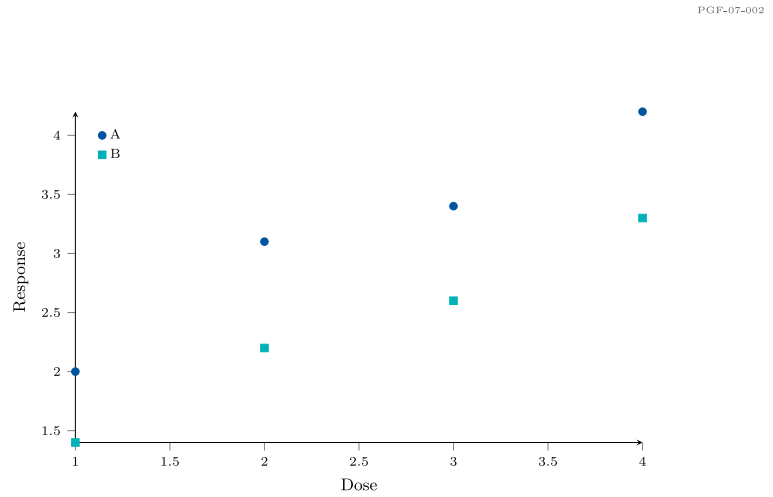
Encode categories with marker styles

Two explicit series use colour and shape together. The legend states their categories.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={Dose},
  ylabel={Response},
  legend pos=north west
]
  \addplot[Blue,only marks,mark=*] coordinates {(1,2) (2,3.1) (3,3.4) (4,4.2)};
  \addplot[Teal,only marks,mark=square*] coordinates {(1,1.4) (2,2.2) (3,2.6)
(4,3.3)};
  \legend{A,B}
\end{axis}
```

Actual output



PGF-07-002 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`mark / colour`	Encode the categories redundantly.
`legend`	Name the categories in plotting order.

PGF-07-003

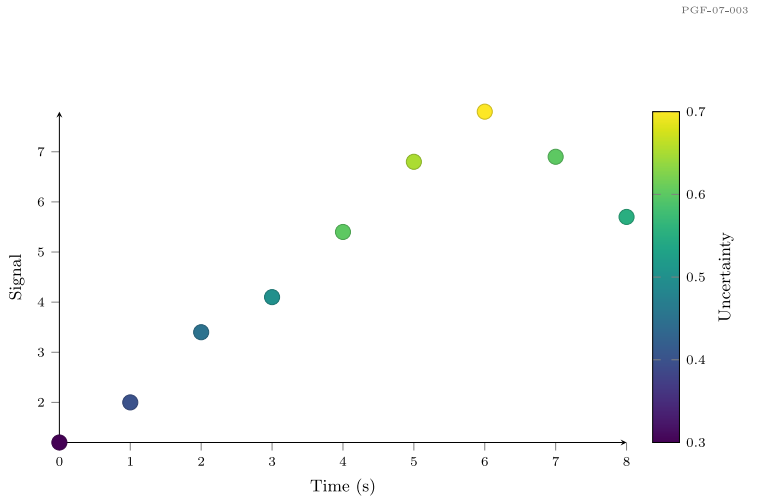
Colour points by a numeric value

Explicit point metadata drives the colour scale. The colourbar records its numeric interpretation.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={Time (s)},
  ylabel={Signal},
  colorbar,
  colormap/viridis,
  colorbar style={ylabel={Uncertainty}}
]
  \addplot[scatter,only marks,scatter src=explicit,mark size=4pt]
table[x=time,y=value,meta=error]{data/measurements.tsv};
\end{axis}
```

Actual output



PGF-07-003 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`scatter src=explicit`	Use the metadata column to assign colours.
`meta=error`	Map the uncertainty column to the colour scale.

PGF-07-004

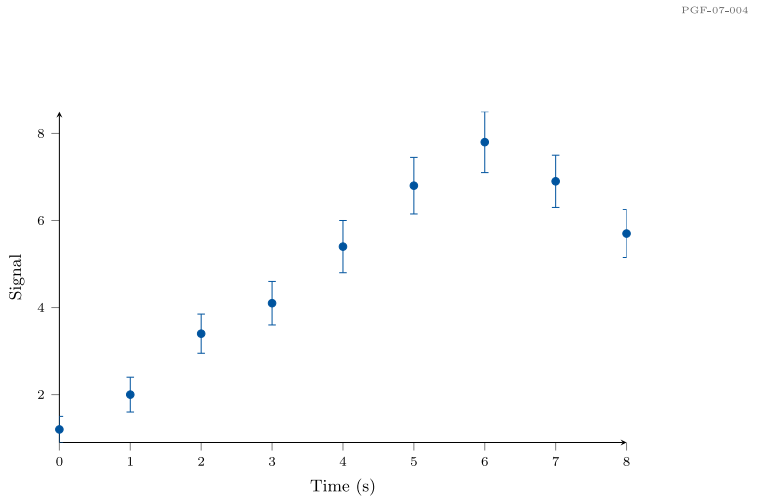
Vertical symmetric error bars

The error column supplies a plus/minus interval for each observation. Its meaning must come from the measurement protocol.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={Time (s)},
  ylabel={Signal}
]
  \addplot[Blue,only marks,mark=*,error bars/.cd,y dir=both,y explicit]
    table[x=time,y=value,y error=error]{data/measurements.tsv};
\end{axis}
```

Actual output



PGF-07-004 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`y explicit`	Read rather than calculate the error amounts.
`y dir=both`	Draw symmetric intervals above and below each point.

PGF-07-005

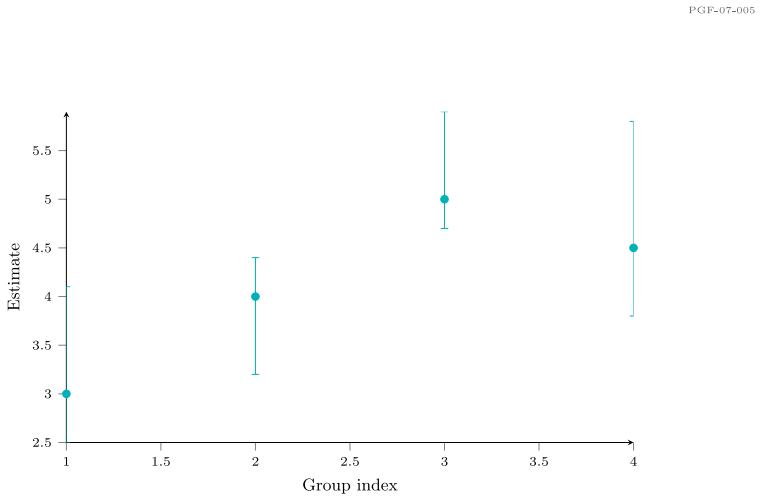
Asymmetric intervals

Separate columns describe lower and upper uncertainty. The point does not need to sit at the midpoint of the interval.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={Group index},
  ylabel={Estimate}
]
  \addplot[Teal,only marks,mark=*,error bars/.cd,y dir=both,y explicit]
    table[x=x,y=y,y error minus=lower,y error plus=upper]{data/intervals.tsv};
\end{axis}
```

Actual output



PGF-07-005 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`y error minus / plus`	Read independent lower and upper distances.
`estimate`	Keep the central estimate distinct from the interval midpoint.

PGF-07-006

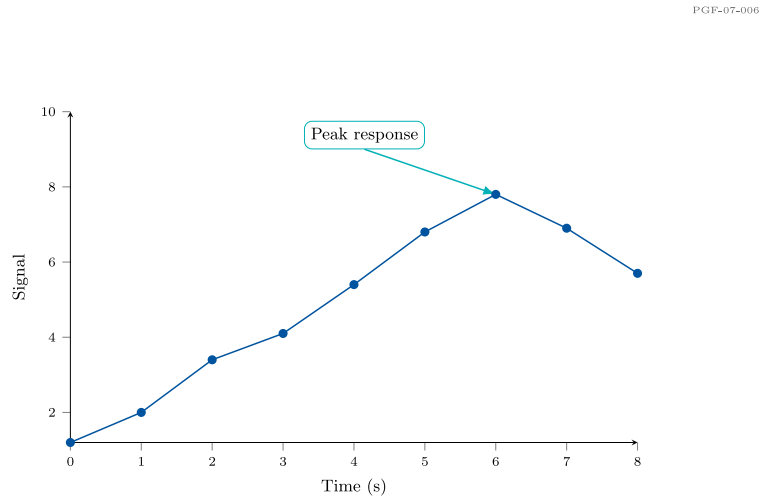
Annotate a selected observation

axis cs places the callout in data coordinates. The arrow continues to identify the same value when the axis range changes.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={Time (s)},
  ylabel={Signal},
  ymax=10
]
  \addplot[Blue,thick,mark=*] table[x=time,y=value]{data/measurements.tsv};
  \node[draw=Teal,fill=white,rounded corners,anchor=south east]
    (note) at (axis cs:5,9) {Peak response};
  \draw[-{Latex},Teal,thick] (note.south) -- (axis cs:6,7.8);
\end{axis}
```

Actual output



PGF-07-006 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`axis cs`	Interpret the coordinate in the axis data system.
`named note`	Connect a callout node to its observation.

Bars and categorical comparisons

Bar position, baseline and ordering carry meaning. These examples keep zero visible for absolute counts and state when values are stacked or normalized.

PGF-08-001

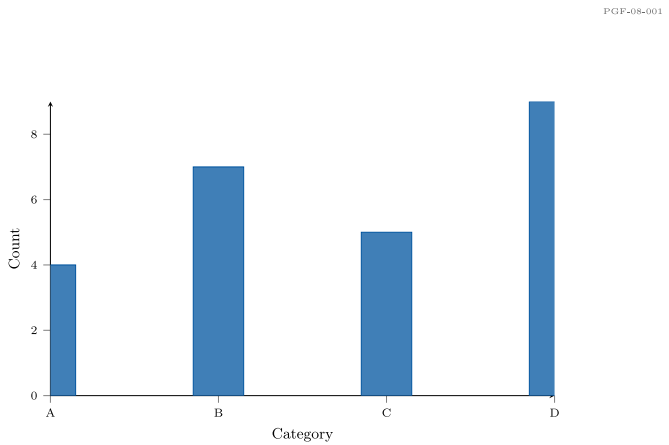
Vertical category bars

The bars share a zero baseline. Numeric x positions are given human-readable category labels.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  ymin=0,
  xtick={1,2,3,4},
  xticklabels={A,B,C,D},
  xlabel={Category},
  ylabel={Count}
]
  \addplot[ybar,Blue,fill=Blue!75,bar width=30bp] coordinates {(1,4) (2,7) (3,5)
(4,9)};
\end{axis}
```

Actual output



PGF-08-001 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`ybar`	Draw bars perpendicular to the x axis.
`bar width`	Give each bar a fixed physical width.

PGF-08-002

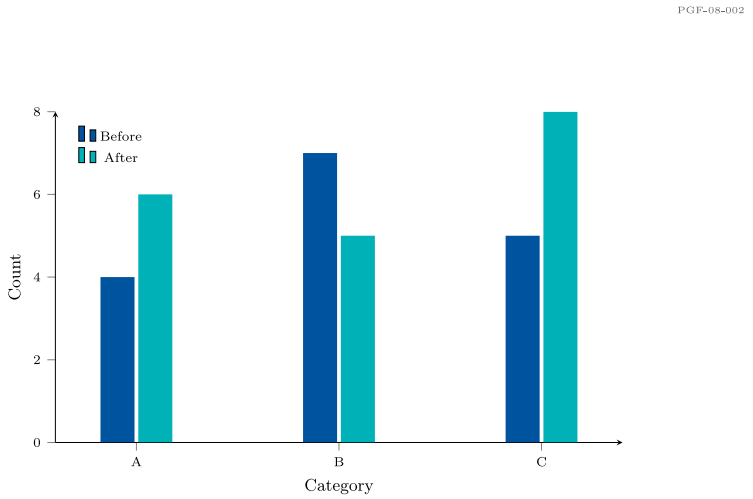
Grouped bars

A shared ybar axis shifts each series beside its neighbours. All groups use the same category positions and count scale.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  ybar,
  enlarge x limits=.2,
  bar width=18bp,
  ymin=0,
  xtick={1,2,3},
  xticklabels={A,B,C},
  xlabel={Category},
  ylabel={Count},
  legend pos=north west
]
  \addplot[fill=Blue,draw=none] coordinates {(1,4) (2,7) (3,5)};
  \addplot[fill=Teal,draw=none] coordinates {(1,6) (2,5) (3,8)};
  \legend{Before,After}
\end{axis}
```

Actual output



PGF-08-002 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`axis-level ybar`	Apply grouped bar shifts to the series.
`same x coordinates`	Align the series to one set of categories.

PGF-08-003

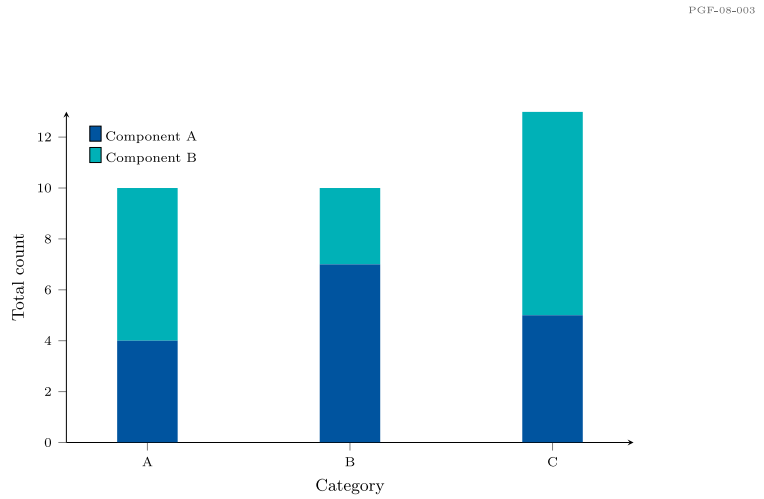
Stacked counts

Each bar height is the sum of its components. Stack ordering follows the order of the addplot commands.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  ybar stacked,
  enlarge x limits=.2,
  bar width=32bp,
  ymin=0,
  xtick={1,2,3},
  xticklabels={A,B,C},
  xlabel={Category},
  ylabel={Total count},
  legend pos=north west
]
  \addplot[fill=Blue,draw=none] coordinates {(1,4) (2,7) (3,5)};
  \addplot[fill=Teal,draw=none] coordinates {(1,6) (2,3) (3,8)};
  \legend{Component A,Component B}
\end{axis}
```

Actual output



PGF-08-003 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`ybar stacked`	Accumulate each series above the preceding series.
`series order`	Determine which component touches the zero baseline.

PGF-08-004

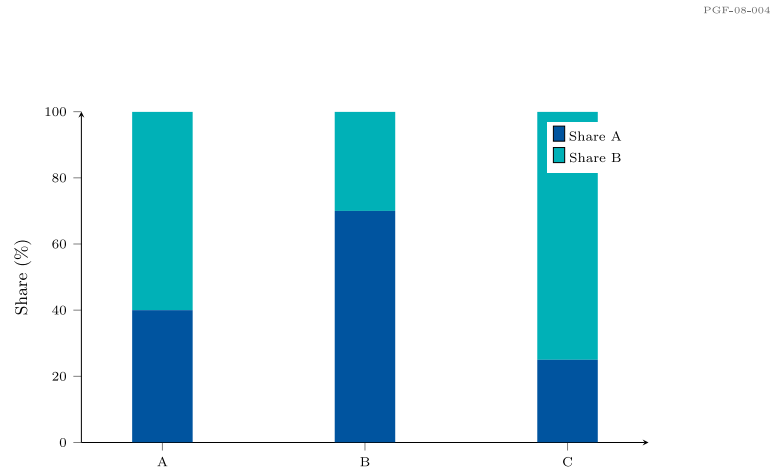
A normalized composition

Values are prepared percentages whose components sum to 100. PGFplots stacks these supplied values; it does not calculate the percentages here.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  ybar stacked,
  enlarge x limits=.2,
  bar width=32bp,
  ymin=0,
  ymax=100,
  xtick={1,2,3},
  xticklabels={A,B,C},
  ylabel={Share (\%)},
  legend pos=north east
]
  \addplot[fill=Blue,draw=none] coordinates {(1,40) (2,70) (3,25)};
  \addplot[fill=Teal,draw=none] coordinates {(1,60) (2,30) (3,75)};
  \legend{Share A,Share B}
\end{axis}
```

Actual output



PGF-08-004 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`prepared percentages`	Use a fixed total of one hundred for every category.
`ymax=100`	Make the compositional scale explicit.

PGF-08-005

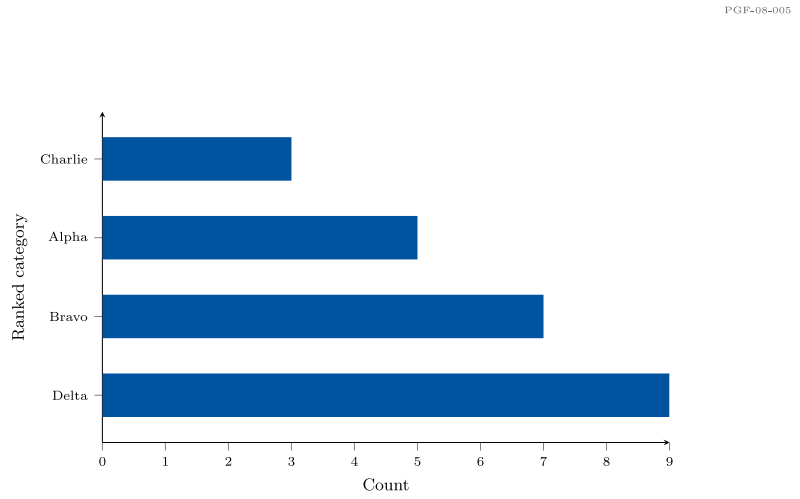
Horizontal ranked bars

Horizontal bars leave room for category names. Explicit y labels make the chosen ranking visible.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xmin=0,
  enlarge y limits=.2,
  ytick={1,2,3,4},
  yticklabels={Delta,Bravo,Alpha,Charlie},
  xlabel={Count},
  ylabel={Ranked category}
]
  \addplot[xbar,fill=Blue,draw=none,bar width=23bp] coordinates {(9,1) (7,2) (5,3)
(3,4)};
\end{axis}
```

Actual output



PGF-08-005 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`xbar`	Draw bars along the x direction.
`yticklabels`	Show category names beside their numeric y positions.

PGF-08-006

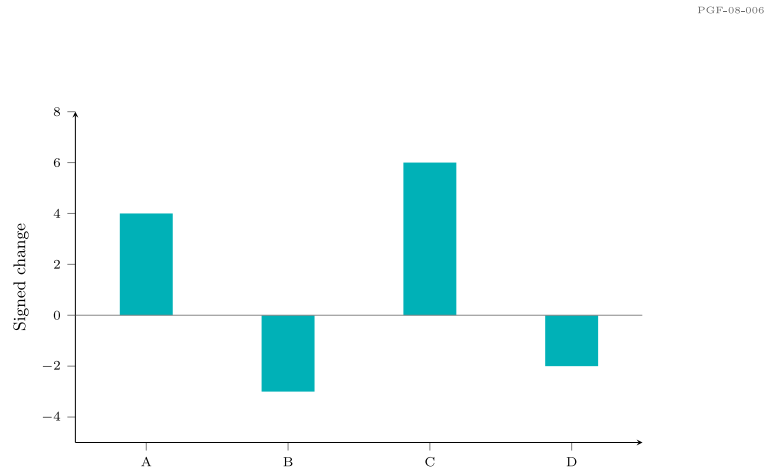
Diverging changes around zero

Positive and negative values retain their signs. The zero line separates gains from losses.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xmin=.5,
  xmax=4.5,
  ymin=-5,
  ymax=8,
  xtick={1,2,3,4},
  xticklabels={A,B,C,D},
  ylabel={Signed change}
]
  \addplot[ybar,fill=Teal,draw=none,bar width=28bp] coordinates {(1,4) (2,-3) (3,6)
(4,-2)};
  \draw[gray,thin] (axis cs:.5,0) -- (axis cs:4.5,0);
\end{axis}
```

Actual output



PGF-08-006 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`signed values`	Keep decreases below the zero baseline.
`zero reference`	Make the direction of the change explicit.

Distributions and statistical summaries

State the estimator, bins and uncertainty definition. Small synthetic samples keep the drawing methods and statistical conventions visible.

PGF-09-001

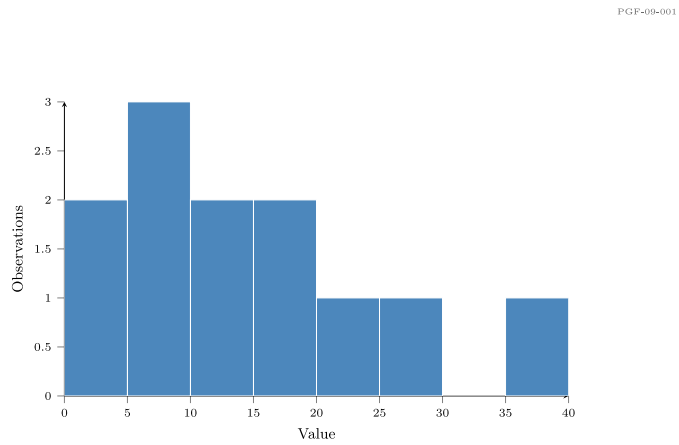
A histogram with fixed bin edges

The bin count and data range determine the bins. These are counts of the supplied observations, not a fitted density.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  ymin=0,
  xlabel={Value},
  ylabel={Observations}
]
  \addplot[ybar interval,fill=Blue!70,draw=white,
    hist={bins=8,data min=0,data max=40}]
    table[y index=0]{data/sample-values.tsv};
\end{axis}
```

Actual output



PGF-09-001 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`hist`	Count observations into explicit bins.
`data min / max`	Fix the range used for the histogram.

PGF-09-002

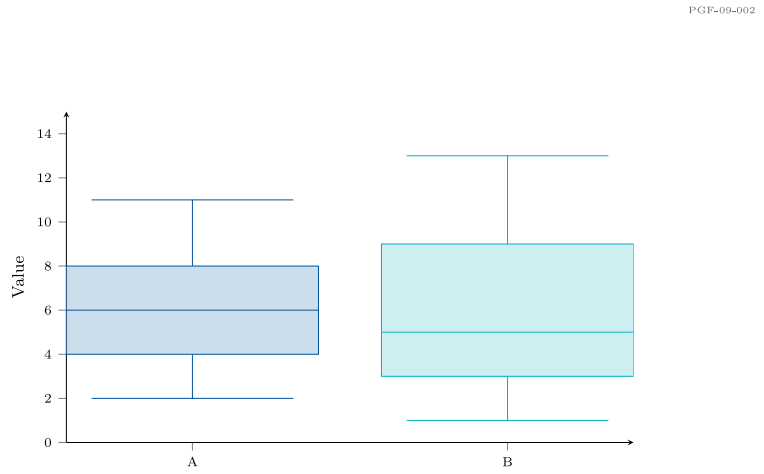
Prepared box-and-whisker summaries

The five supplied numbers determine the displayed box. Whiskers are stated endpoints; the plot does not recalculate a statistical convention.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  boxplot/draw direction=y,
  xtick={1,2},
  xticklabels={A,B},
  ylabel={Value},
  ymin=0,
  ymax=15
]
  \addplot[boxplot prepared={draw position=1,lower whisker=2,
    lower quartile=4,median=6,upper quartile=8,upper whisker=11},
    fill=Blue!20,draw=Blue] coordinates {};
  \addplot[boxplot prepared={draw position=2,lower whisker=1,
    lower quartile=3,median=5,upper quartile=9,upper whisker=13},
    fill=Teal!20,draw=Teal] coordinates {};
\end{axis}
```

Actual output



PGF-09-002 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`boxplot prepared`	Draw a declared five-number summary.
`draw direction=y`	Place the numeric distribution on the vertical axis.

PGF-09-003

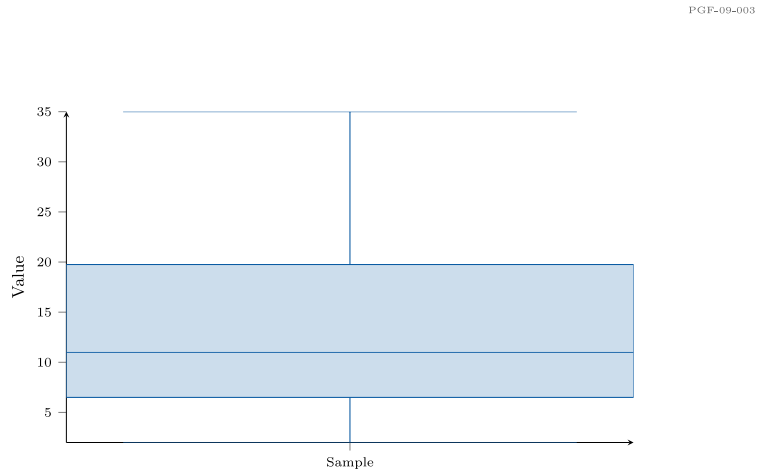
A boxplot calculated from samples

The statistics library derives a boxplot from the samples. Its native quartile and whisker choices can differ from an externally prepared summary.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  boxplot/draw direction=y,
  xtick={1},
  xticklabels={Sample},
  ylabel={Value}
]
  \addplot[boxplot={draw position=1},fill=Blue!20,draw=Blue]
    table[y index=0]{data/sample-values.tsv};
\end{axis}
```

Actual output



PGF-09-003 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`boxplot`	Calculate the summary using the PGFplots statistics library.
`native defaults`	Consult the manual before comparing quartile or whisker definitions.

PGF-09-004

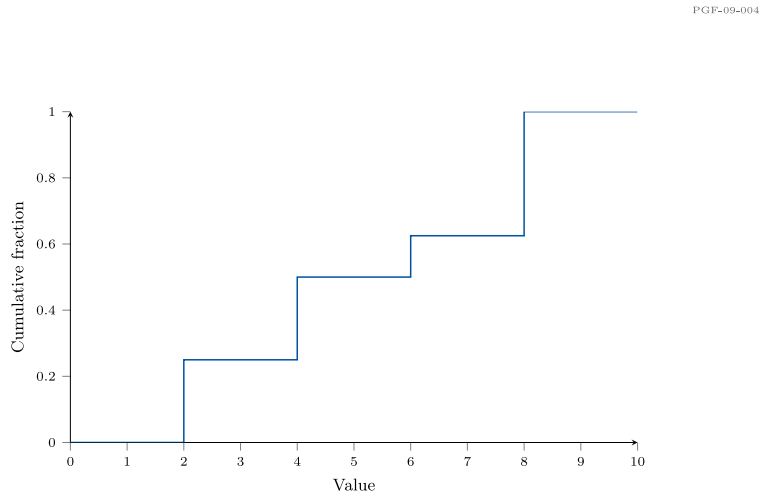
An explicit right-continuous ECDF

Horizontal and vertical vertices fix the step convention, including tied observations. The vertical axis is a cumulative fraction.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xmin=0,
  xmax=10,
  ymin=0,
  ymax=1,
  xlabel={Value},
  ylabel={Cumulative fraction}
]
  \addplot[Blue,thick] coordinates {
    (0,0) (2,0) (2,.25) (4,.25) (4,.5)
    (6,.5) (6,.625) (8,.625) (8,1) (10,1)
  };
\end{axis}
```

Actual output



PGF-09-004 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`explicit vertices`	Represent both sides of every step without an interpolation assumption.
`fraction scale`	Show values between zero and one.

PGF-09-005

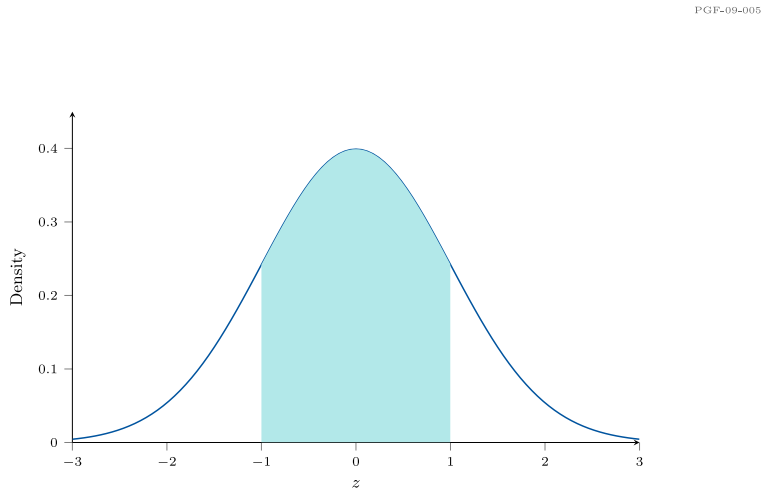
A density curve with a reference interval

This analytic normal density demonstrates area shading. It is not an estimate from the histogram sample.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={$z$},
  ylabel={Density},
  ymin=0,
  ymax=.45
]
  \addplot[Blue,thick,domain=-3:3,samples=101]
    {exp(-x*x/2)/sqrt(2*pi)};
  \addplot[draw=none,fill=Teal!30,domain=-1:1,samples=61]
    {exp(-x*x/2)/sqrt(2*pi)} \closedcycle;
\end{axis}
```

Actual output



PGF-09-005 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`normal density`	Evaluate the stated analytic probability density.
`closedcycle`	Close the highlighted curve against the baseline before filling.

PGF-09-006

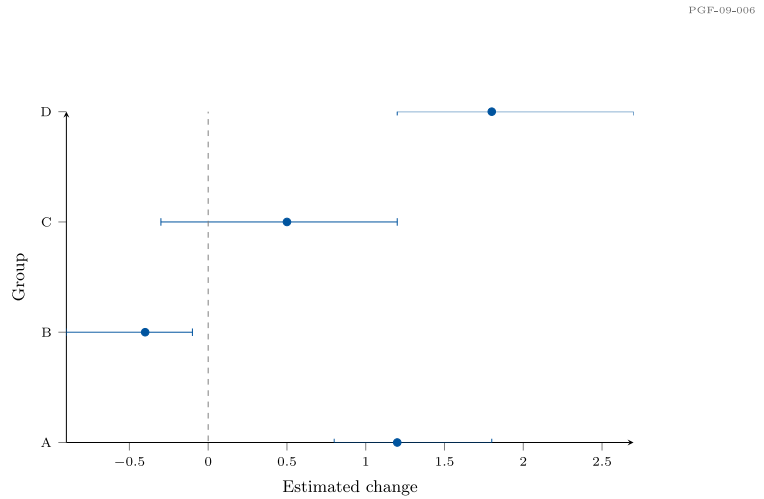
Point estimates with confidence intervals

The supplied asymmetric intervals are labelled as an illustrative confidence-interval example. They are not calculated by the plotting engine.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  ytick={1,2,3,4},
  yticklabels={A,B,C,D},
  xlabel={Estimated change},
  ylabel={Group}
]
  \addplot[Blue,only marks,mark=*,error bars/.cd,x dir=both,x explicit]
    table[x=estimate,y=group,x error minus=lower,x error plus=upper]
    {data/forest.tsv};
  \draw[gray,dashed] (axis cs:0,.5) -- (axis cs:0,4.5);
\end{axis}
```

Actual output



PGF-09-006 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`x explicit`	Use the supplied horizontal interval distances.
`zero line`	Distinguish intervals that cross the reference value.

Areas, bands and filled paths

Closed areas need a stated baseline. Bands between curves are distinct from stacked totals, and either can be drawn without rewriting the input values.

PGF-10-001

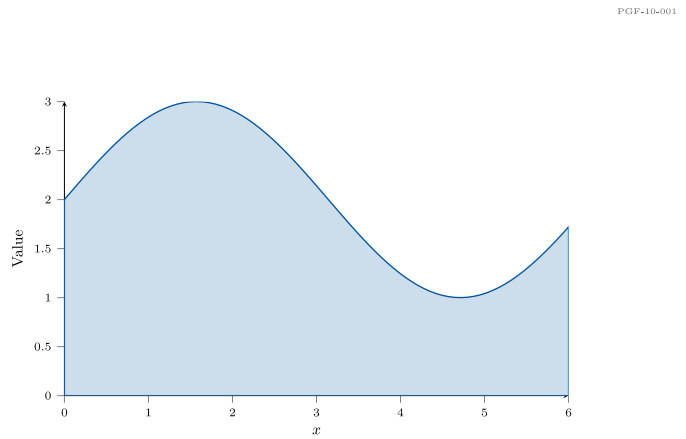
An area under a curve

The path closes against the baseline. This makes the area reference explicit rather than filling an arbitrary open path.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xmin=0,
  xmax=6,
  ymin=0,
  xlabel={$x$},
  ylabel={Value}
]
  \addplot[Blue,thick,fill=Blue!20,domain=0:6,samples=81] {2+sin(deg(x))}
\closedcycle;
\end{axis}
```

Actual output



PGF-10-001 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`fill`	Colour the area bounded by the closed path.
`closedcycle`	Return the curve to its baseline.

PGF-10-002

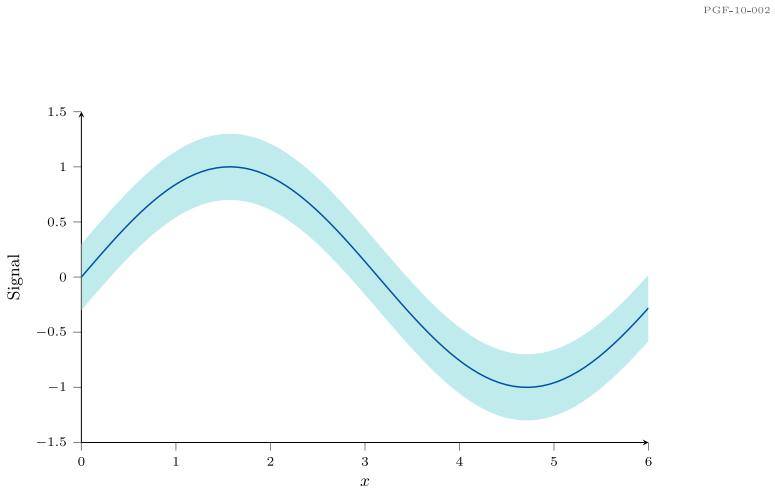
A band between named curves

The fillbetween library uses two named paths. Here the band is a supplied symmetric envelope, not a statistical estimate.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xmin=0,
  xmax=6,
  ymin=-1.5,
  ymax=1.5,
  xlabel={$x$},
  ylabel={Signal}
]
  \addplot[name path=upper,draw=none,domain=0:6,samples=81] {sin(deg(x))+.3};
  \addplot[name path=lower,draw=none,domain=0:6,samples=81] {sin(deg(x))-.3};
  \addplot[Teal!25] fill between[of=upper and lower];
  \addplot[Blue,thick,domain=0:6,samples=81] {sin(deg(x))};
\end{axis}
```

Actual output



PGF-10-002 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`name path`	Identify both envelope boundaries.
`fill between`	Fill the region between the two paths.

PGF-10-003

Restrict a band to an interval

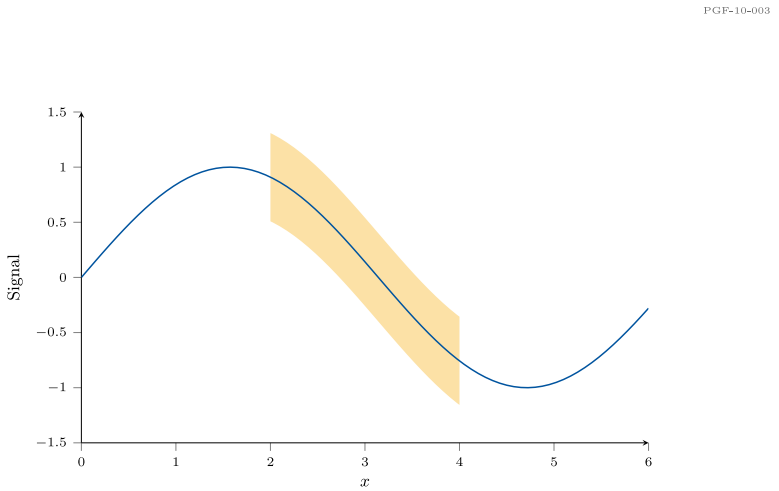
soft clip selects the interval where the band is shown. Both boundary paths still cover the full domain. Compare with PGF-10-002.

Compare with PGF-10-002 · A band between named curves.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xmin=0,
  xmax=6,
  ymin=-1.5,
  ymax=1.5,
  xlabel={$x$},
  ylabel={Signal}
]
  \addplot[name path=upper,draw=none,domain=0:6,samples=81] {sin(deg(x))+.4};
  \addplot[name path=lower,draw=none,domain=0:6,samples=81] {sin(deg(x))-.4};
  \addplot[Orange!35] fill between[of=upper and lower,soft clip={domain=2:4}];
  \addplot[Blue,thick,domain=0:6,samples=81] {sin(deg(x))};
\end{axis}
```

Actual output



PGF-10-003 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`soft clip`	Limit the filled region in data coordinates.
`domain=2:4`	Highlight only the chosen interval.

PGF-10-004

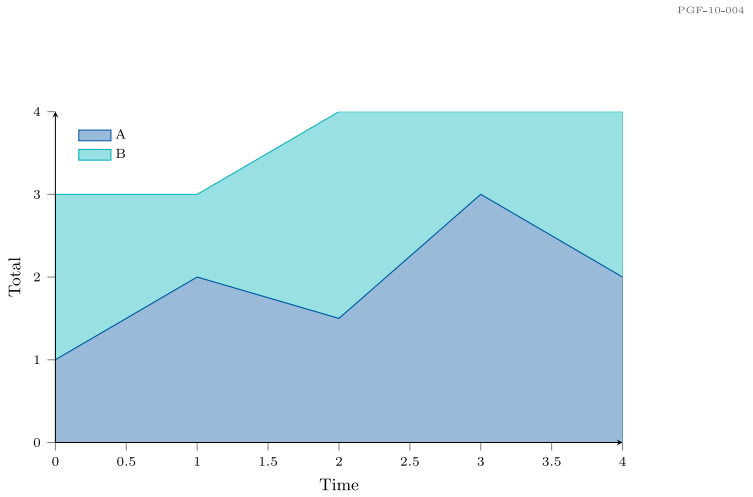
Stacked areas

stack plots=y accumulates the supplied components. The top boundary represents their sum.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  stack plots=y,
  area style,
  ymin=0,
  xlabel={Time},
  ylabel={Total},
  legend pos=north west
]
  \addplot[draw=Blue,fill=Blue!40] coordinates {(0,1) (1,2) (2,1.5) (3,3) (4,2)}
\closedcycle;
  \addplot[draw=Teal,fill=Teal!40] coordinates {(0,2) (1,1) (2,2.5) (3,1) (4,2)}
\closedcycle;
  \legend{A,B}
\end{axis}
```

Actual output



PGF-10-004 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`stack plots=y`	Accumulate each series on the preceding one.
`area style`	Set the stacking behaviour for filled areas.

PGF-10-005

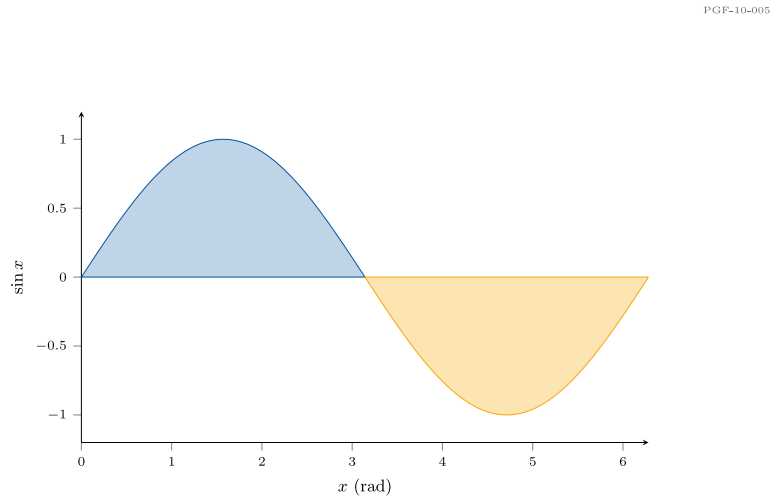
Positive and negative filled regions

Separate paths make the sign and zero baseline visible. The same curve is sampled on two intervals with different fill colours.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xmin=0,
  xmax=6.2832,
  ymin=-1.2,
  ymax=1.2,
  xlabel={$x$ (rad)},
  ylabel={$\sin x$}
]
  \addplot[Blue,fill=Blue!25,domain=0:pi,samples=61] {sin(deg(x))} \closedcycle;
  \addplot[Orange,fill=Orange!30,domain=pi:2*pi,samples=61] {sin(deg(x))}
\closedcycle;
\end{axis}
```

Actual output



PGF-10-005 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`separate domains`	Split at the known zero crossing.
`baseline`	Keep positive and negative signed areas distinct.

PGF-10-006

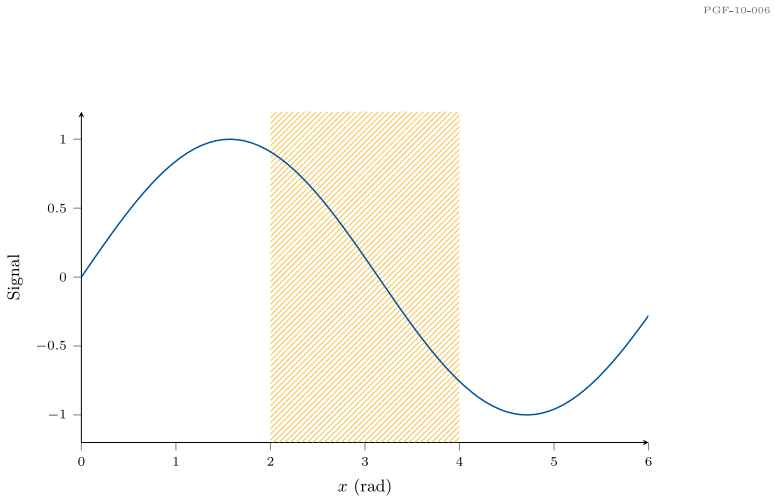
A hatched interval overlay

A transparent hatch marks an interval while preserving the data curve. The rectangle is placed in axis coordinates.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xmin=0,
  xmax=6,
  ymin=-1.2,
  ymax=1.2,
  xlabel={$x$ (rad)},
  ylabel={Signal}
]
  \path[pattern=north east lines,pattern color=Orange!70]
    (axis cs:2,-1.2) rectangle (axis cs:4,1.2);
  \addplot[Blue,thick,domain=0:6,samples=81] {sin(deg(x))};
\end{axis}
```

Actual output



PGF-10-006 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`axis cs`	Anchor the overlay to the data range.
`pattern`	Keep the highlighted region recognizable without solid colour.

Heatmaps, colour scales and isolines

Colour encodes a numeric variable only when its scale is stated. Smooth surfaces, cell-valued grids and analytic isolines describe different kinds of information.

PGF-11-001

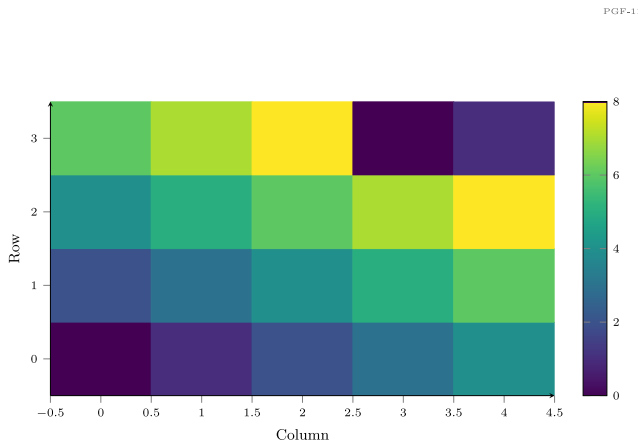
An explicit grid of cell values

matrix plot* draws the supplied rectangular grid. Point metadata selects the colour for each cell.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  view={0}{90},
  axis on top,
  xmin=-.5,
  xmax=4.5,
  ymin=-.5,
  ymax=3.5,
  xlabel={Column},
  ylabel={Row},
  colorbar,
  colormap/viridis,
  point meta min=0,
  point meta max=8
]
  \addplot[matrix plot*,point meta=explicit,mesh/cols=5]
    table[x=x,y=y,meta=value]{data/grid.tsv};
\end{axis}
```

Actual output



PGF-11-001 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`matrix plot*`	Draw a rectangular value grid.
`mesh/cols=5`	State the number of columns in each input row.

PGF-11-002

A smooth function viewed from above

A sampled surface with a top-down camera acts as a smooth colour field. The displayed function remains explicit.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  view={0}{90},
  xlabel={$x$},
  ylabel={$y$},
  colorbar,
  colormap/viridis,
  point meta min=0,
  point meta max=1
]
  \addplot3[surf,shader=interp,domain=-2:2,y domain=-2:2,
    samples=21,samples y=21] {exp(-x*x-y*y)};
\end{axis}
```

Actual output



PGF-11-002 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`view={0}{90}`	Look straight down on the sampled surface.
`shader=interp`	Interpolate colours within the surface patches.

PGF-11-003

Choose discrete colour classes

A piecewise constant colour map groups values into declared intervals. It represents categories of a numeric range. Compare with PGF-11-001.

Compare with PGF-11-001 · An explicit grid of cell values.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  view={0}{90},
  axis on top,
  xmin=-.5,
  xmax=4.5,
  ymin=-.5,
  ymax=3.5,
  xlabel={Column},
  ylabel={Row},
  colorbar,
  colormap={steps}{color(0cm)=(Blue!10);color(2cm)=(Blue!35);color(4cm)=(Teal!
65);color(6cm)=(Orange);color(8cm)=(Orange)},
  colormap access=piecewise constant,
  point meta min=0,
  point meta max=8
]
  \addplot[matrix plot*,point meta=explicit,mesh/cols=5]
    table[x=x,y=y,meta=value]{data/grid.tsv};
\end{axis}
```

Actual output



PGF-11-003 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`piecewise constant`	Assign each interval one fixed colour.
`declared stops`	Make the class boundaries part of the plotting specification.

PGF-11-004

Overlay analytic isolines

For f(x,y)=x squared plus y squared, constant-value contours are circles. These exact paths avoid a separate contouring process.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  view={0}{90},
  xlabel={$x$},
  ylabel={$y$},
  colorbar,
  colormap/viridis,
  point meta min=0,
  point meta max=8
]
  \addplot3[surf,shader=interp,domain=-2:2,y domain=-2:2,
    samples=21,samples y=21] {x*x+y*y};
  \pgfplotsinvokeforeach{.5,1,1.5}{
    \addplot3[white,thick,domain=0:360,samples=81,samples y=0]
      ({#1*cos(x)},{#1*sin(x)},{#1*#1+.01});
  }
\end{axis}
```

Actual output



PGF-11-004 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`analytic contours`	Draw paths whose function value is known exactly.
`small z offset`	Keep contour strokes visible above the surface patches.

PGF-11-005

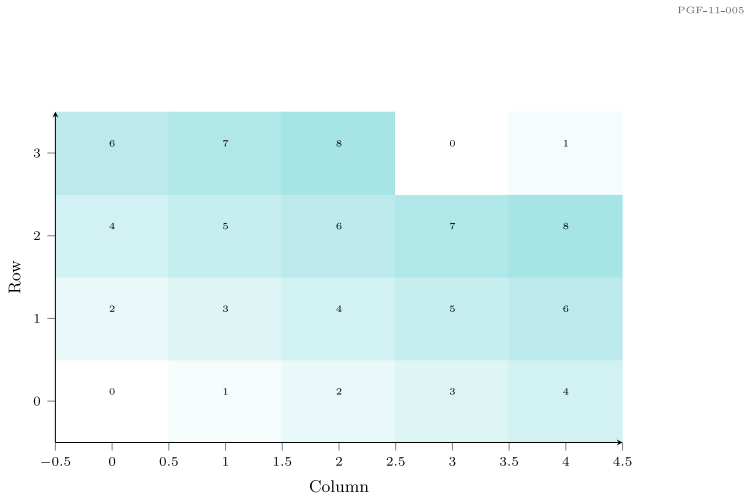
Print numeric cell labels

A colour field can be supplemented with values. Contrasting labels let readers recover the exact small grid without estimating from a colourbar.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  view={0}{90},
  axis on top,
  xmin=-.5,
  xmax=4.5,
  ymin=-.5,
  ymax=3.5,
  xlabel={Column},
  ylabel={Row},
  colormap={pale}{color=(white);color=(Teal!35)},
  point meta min=0,
  point meta max=8
]
  \addplot[matrix plot*,point meta=explicit,mesh/cols=5,
    nodes near coords, nodes near coords style={font=\tiny,text=black}]
    table[x=x,y=y,meta=value]{data/grid.tsv};
\end{axis}
```

Actual output



PGF-11-005 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`nodes near coords`	Print the supplied point metadata beside each cell.
`pale colour map`	Keep black numeric labels readable across the field.

PGF-11-006

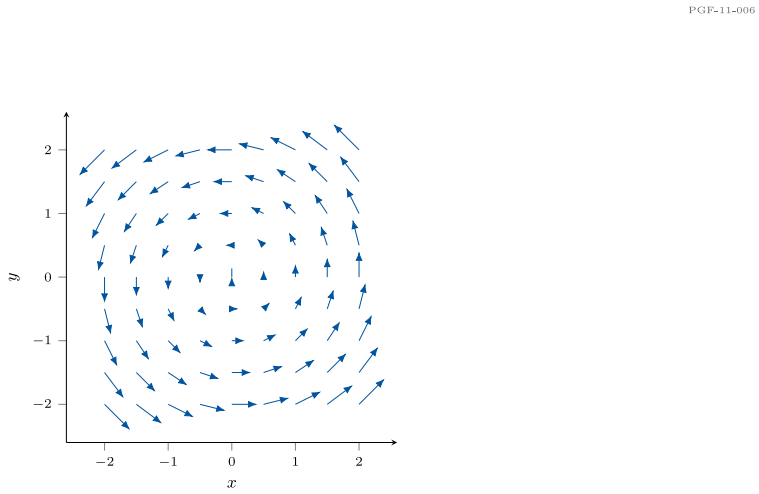
A directional vector field

Each arrow is a sampled vector (-y,x). Scaling changes arrow lengths while preserving their directions.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  view={0}{90},
  axis equal image,
  xmin=-2.6,
  xmax=2.6,
  ymin=-2.6,
  ymax=2.6,
  xlabel={$x$},
  ylabel={$y$}
]
  \addplot3[Blue,-{Latex},quiver={u=-y,v=x,w=0,scale arrows=.2},
    samples=9,samples y=9,domain=-2:2,y domain=-2:2] {0};
\end{axis}
```

Actual output



PGF-11-006 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`quiver`	Draw vectors anchored at the sampled coordinates.
`u / v`	Set the horizontal and vertical vector components.

Three-dimensional plots

The camera and sampling grid affect how a surface is perceived. The native PDF is a fixed vector projection; it is not an interactive 3D object.

PGF-12-001

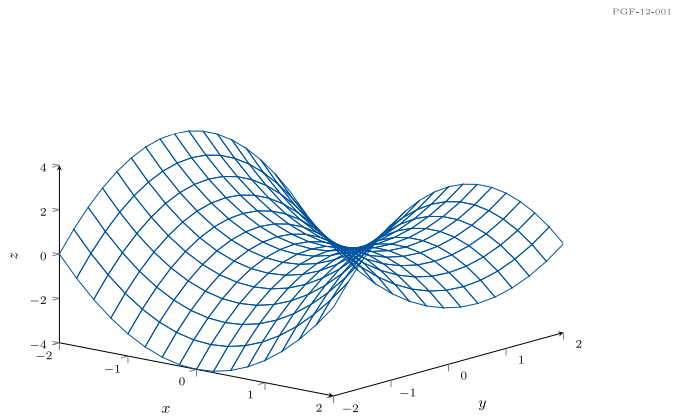
A wireframe saddle

A mesh emphasizes the sampling grid of z=x squared minus y squared. The view is fixed for reproducible comparison.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  view={40}{25},
  xlabel={$x$},
  ylabel={$y$},
  zlabel={$z$}
]
  \addplot3[mesh,Blue,domain=-2:2,y domain=-2:2,samples=17,samples y=17] {x*x-y*y};
\end{axis}
```

Actual output



PGF-12-001 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`mesh`	Draw the grid edges of a sampled surface.
`view`	Choose camera azimuth and elevation in degrees.

PGF-12-002

A smoothly shaded Gaussian surface

The scalar height supplies the colour metadata. Interpolated shading changes the appearance between fixed samples.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  view={35}{28},
  xlabel={$x$},
  ylabel={$y$},
  zlabel={$z$},
  colormap/viridis,
  colorbar
]
  \addplot3[surf,shader=interp,domain=-2:2,y domain=-2:2,samples=21,samples y=21]
{exp(-x*x-y*y)};
\end{axis}
```

Actual output



PGF-12-002 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`surf`	Draw filled surface patches.
`samples / samples y`	Choose a reproducible two-dimensional sampling grid.

PGF-12-003

Change only the camera

The surface, samples and colour map stay fixed. A higher camera makes the footprint easier to see. Compare with PGF-12-002.

Compare with PGF-12-002 · A smoothly shaded Gaussian surface.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  view={65}{55},
  xlabel={$x$},
  ylabel={$y$},
  zlabel={$z$},
  colormap/viridis,
  colorbar
]
  \addplot3[surf,shader=interp,domain=-2:2,y domain=-2:2,samples=21,samples y=21]
{exp(-x*x-y*y)};
\end{axis}
```

Actual output



PGF-12-003 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`view={65}{55}`	Rotate and elevate the camera.
`same sampled surface`	Separate a view change from a data or estimator change.

PGF-12-004

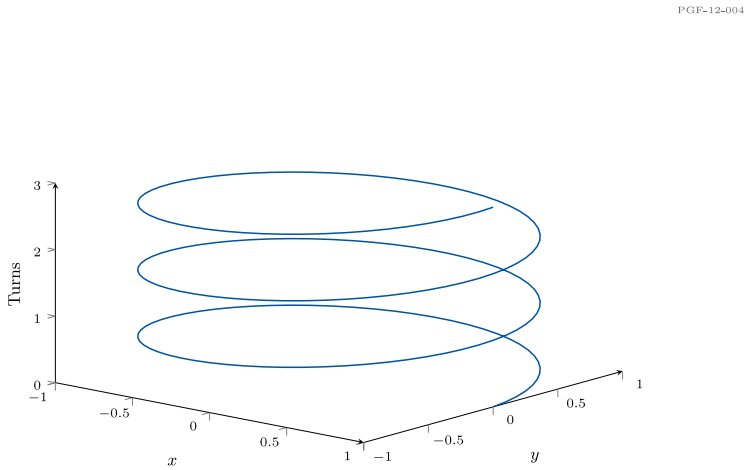
A parametric helix

Three expressions share one parameter. The vertical coordinate counts turns while the horizontal coordinates trace a circle.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  view={40}{25},
  xlabel={$x$},
  ylabel={$y$},
  zlabel={Turns}
]
  \addplot3[Blue,thick,domain=0:6*pi,samples=181,samples y=0]
    ({cos(deg(x))},{sin(deg(x))},{x/(2*pi)});
\end{axis}
```

Actual output



PGF-12-004 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`parametric addplot3`	Compute all three coordinates from one parameter.
`deg(x)`	Convert the parameter to degrees for trigonometric evaluation.

PGF-12-005

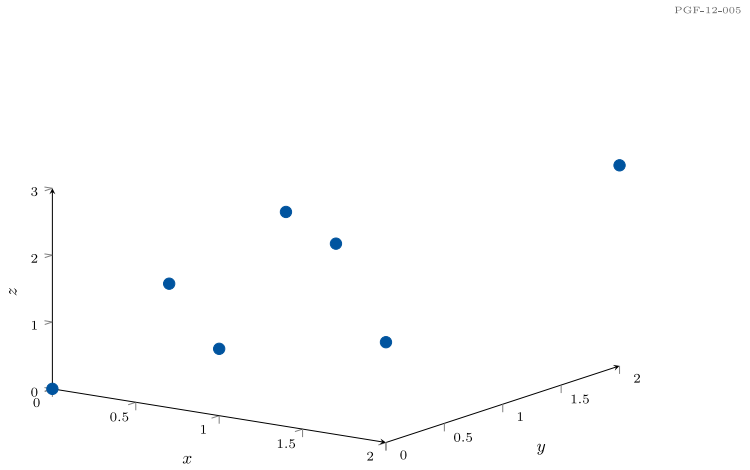
Three-dimensional observations

Markers show individual 3D coordinates. A fixed projection can conceal overlap, so the camera is part of the specification.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  view={35}{25},
  xlabel={$x$},
  ylabel={$y$},
  zlabel={$z$}
]
  \addplot3[only marks,Blue,mark=*,mark size=3pt] coordinates {
    (0,0,0) (1,0,1) (0,1,1) (1,1,2)
    (2,0,1.5) (0,2,1.5) (2,2,3)
  };
\end{axis}
```

Actual output



PGF-12-005 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`only marks`	Draw observations without a connecting surface.
`three coordinates`	Supply x, y and z for every record.

PGF-12-006

A faceted surface with visible patches

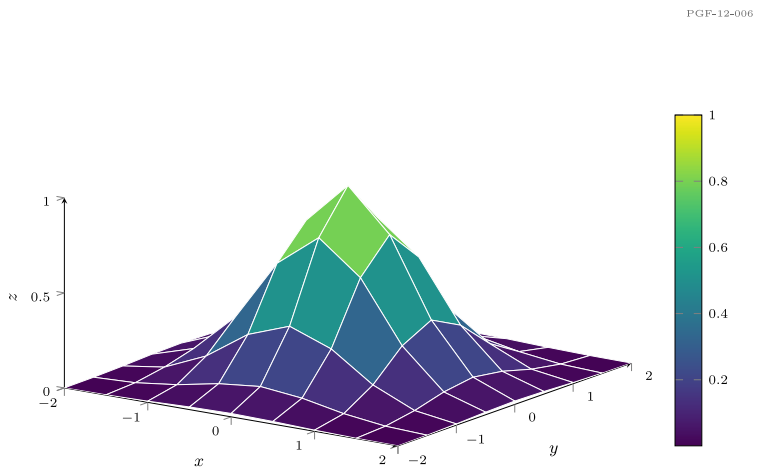
Flat shading makes the patch resolution visible. This is the same analytic Gaussian on a coarser grid. Compare with PGF-12-002.

Compare with PGF-12-002 · A smoothly shaded Gaussian surface.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  view={35}{28},
  xlabel={$x$},
  ylabel={$y$},
  zlabel={$z$},
  colormap/viridis,
  colorbar
]
  \addplot3[surf,shader=flat,draw=white,domain=-2:2,y domain=-2:2,samples=9,samples
y=9] {exp(-x*x-y*y)};
\end{axis}
```

Actual output



PGF-12-006 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`shader=flat`	Assign a uniform colour to each patch.
`coarser samples`	Expose the sampling grid rather than implying a smoother mesh.

Multi-panel figures and shared guides

Panels need comparable scales and deliberate spacing. These examples use groupplots or aligned axis rectangles, with a fixed overall canvas and explicit labels.

PGF-13-001

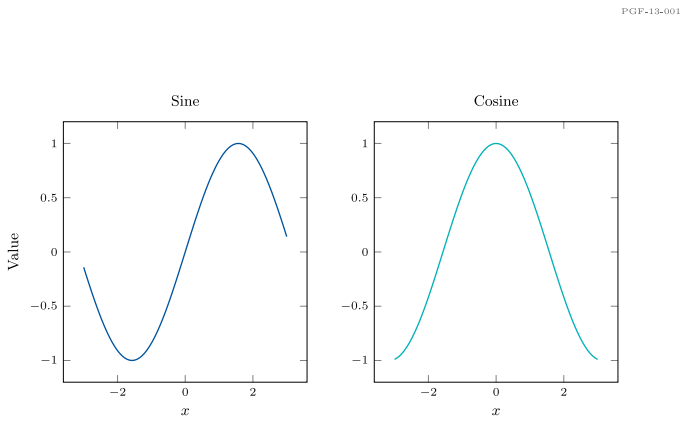
Two neighbouring panels

The groupplots library aligns a sine panel and a cosine panel. Both use the same coordinate ranges.

PGFplots / TikZ drawing

```
\begin{scope}[shift={(45,50)}]
\begin{groupplot}[
  group style={group size=2 by 1,horizontal sep=40bp,vertical sep=30bp},
  width=145bp,
  height=155bp,
  scale only axis,
  tick label style={font=\scriptsize}
]
  \nextgroupplot[title={Sine},xlabel={$x$},ylabel={Value},ymin=-1.2,ymax=1.2]
  \addplot[Blue,thick,domain=-3:3,samples=81] {sin(deg(x))};
  \nextgroupplot[title={Cosine},xlabel={$x$},ymin=-1.2,ymax=1.2]
  \addplot[Teal,thick,domain=-3:3,samples=81] {cos(deg(x))};
\end{groupplot}
\end{scope}
```

Actual output



PGF-13-001 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`group size=2 by 1`	Create two aligned axes in one row.
`fixed y range`	Keep equal vertical differences comparable between panels.

PGF-13-002

Keep labels on the outside

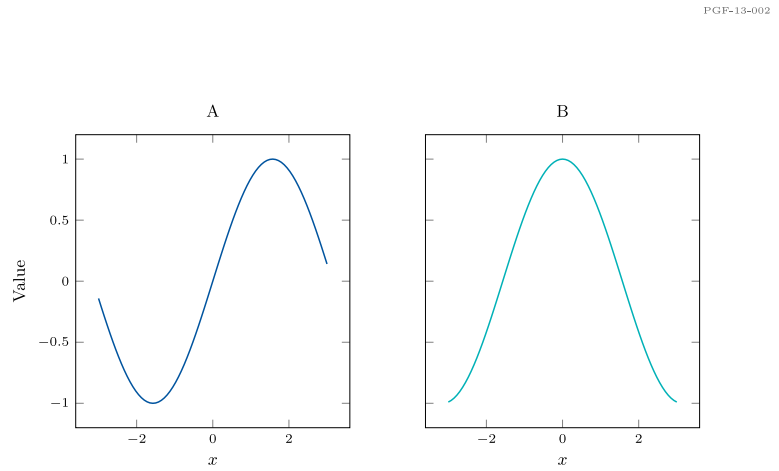
Outer-edge descriptions reduce repeated labels while retaining a common scale. Tick values remain available on both panels. Compare with PGF-13-001.

Compare with PGF-13-001 · Two neighbouring panels.

PGFplots / TikZ drawing

```
\begin{scope}[shift={(45,50)}]
\begin{groupplot}[
  group style={group size=2 by 1,horizontal sep=40bp,vertical sep=30bp},
  width=145bp,
  height=155bp,
  scale only axis,
  tick label style={font=\scriptsize},
  group style={group size=2 by 1,horizontal sep=40bp,vertical sep=30bp,y descriptions
at=edge left},
  xlabel={$x$},
  ylabel={Value},
  ymin=-1.2,
  ymax=1.2
]
  \nextgroupplot[title={A}]
  \addplot[Blue,thick,domain=-3:3,samples=81] {sin(deg(x))};
  \nextgroupplot[title={B}]
  \addplot[Teal,thick,domain=-3:3,samples=81] {cos(deg(x))};
\end{groupplot}
\end{scope}
```

Actual output



PGF-13-002 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`y descriptions at=edge left`	Show the shared y description only on the left edge.
`common limits`	Retain a common numerical scale in the group.

PGF-13-003

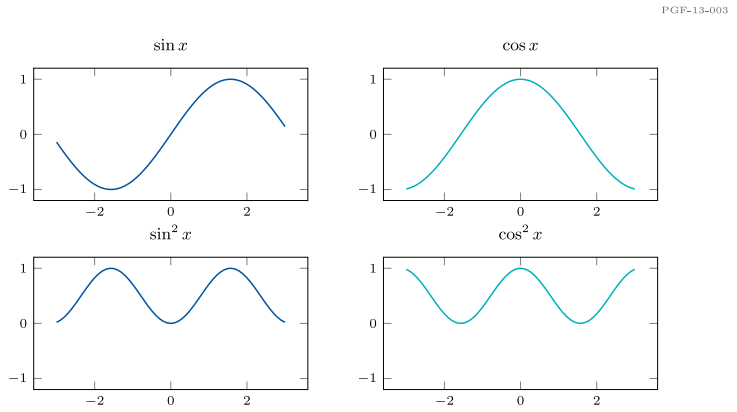
A two-by-two small-multiple figure

Four compact panels compare different analytic functions. Their individual titles identify the transformations.

PGFplots / TikZ drawing

```
\begin{scope}[shift={(45,170)}]
\begin{groupplot}[
  group style={group size=2 by 2,horizontal sep=40bp,vertical sep=30bp},
  width=145bp,
  height=70bp,
  scale only axis,
  tick label style={font=\scriptsize},
  ymin=-1.2,
  ymax=1.2
]
  \nextgroupplot[title={$\sin x$}]
  \addplot[Blue,thick,domain=-3:3,samples=61] {sin(deg(x))};
  \nextgroupplot[title={$\cos x$}]
  \addplot[Teal,thick,domain=-3:3,samples=61] {cos(deg(x))};
  \nextgroupplot[title={$\sin^2 x$}]
  \addplot[Blue,thick,domain=-3:3,samples=61] {sin(deg(x))^2};
  \nextgroupplot[title={$\cos^2 x$}]
  \addplot[Teal,thick,domain=-3:3,samples=61] {cos(deg(x))^2};
\end{groupplot}
\end{scope}
```

Actual output



PGF-13-003 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`group size=2 by 2`	Arrange four panels in a rectangular grid.
`vertical sep`	Reserve space for the row titles and tick labels.

PGF-13-004

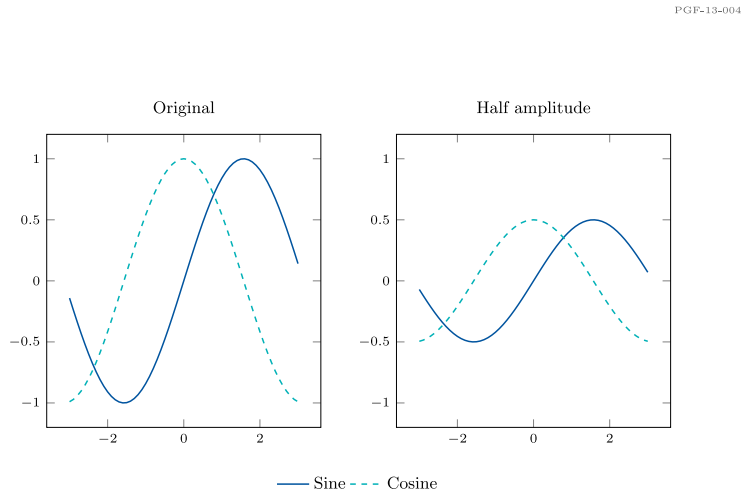
One shared legend

A named legend is placed below the complete group. Compile twice so the legend reference resolves in the standalone document.

PGFplots / TikZ drawing

```
\begin{scope}[shift={(45,50)}]
\begin{groupplot}[
  group style={group size=2 by 1,horizontal sep=40bp,vertical sep=30bp},
  width=145bp,
  height=155bp,
  scale only axis,
  tick label style={font=\scriptsize},
  ymin=-1.2,
  ymax=1.2
]
  \nextgroupplot[title={Original},legend to name=sharedlegend,
    legend columns=2,legend style={draw=none}]
  \addplot[Blue,thick,domain=-3:3,samples=61] {sin(deg(x))};
  \addplot[Teal,thick,dashed,domain=-3:3,samples=61] {cos(deg(x))};
  \legend{Sine,Cosine}
  \nextgroupplot[title={Half amplitude}]
  \addplot[Blue,thick,domain=-3:3,samples=61] {.5*sin(deg(x))};
  \addplot[Teal,thick,dashed,domain=-3:3,samples=61] {.5*cos(deg(x))};
\end{groupplot}
\end{scope}
\node at (210,20) {\pgfplotslegendfromname{sharedlegend}};
```

Actual output



PGF-13-004 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`legend to name`	Store a legend for use elsewhere in the document.
`second TeX pass`	Resolve the named legend in the auxiliary-file reference.

PGF-13-005

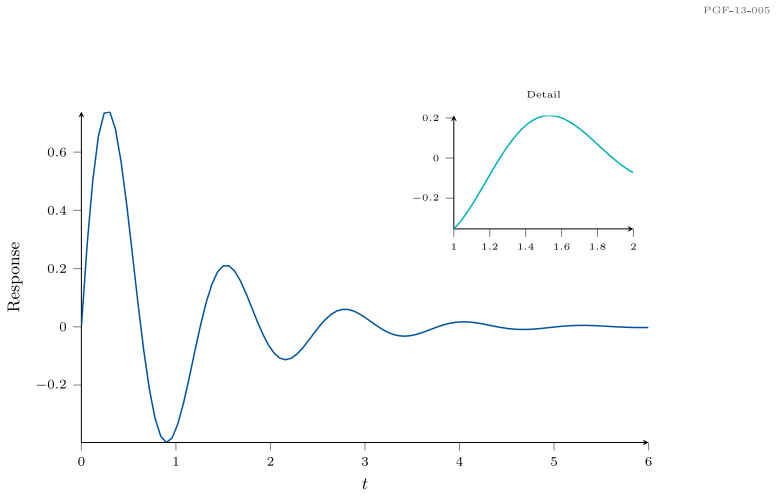
An inset with an explicit data window

A second, smaller axis zooms into a declared interval. Its white background keeps the inset legible against the main plot.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xmin=0,
  xmax=6,
  xlabel={$t$},
  ylabel={Response}
]
  \addplot[Blue,thick,domain=0:6,samples=101] {exp(-x)*sin(deg(5*x))};
\end{axis}
\begin{axis}[
  tour axis,
  at={(245bp,155bp)},
  width=95bp,
  height=60bp,
  xmin=1,
  xmax=2,
  title={Detail},
  axis background/.style={fill=white},
  tick label style={font=\tiny},
  title style={font=\tiny}
]
  \addplot[Teal,thick,domain=1:2,samples=61] {exp(-x)*sin(deg(5*x))};
\end{axis}
```

Actual output



PGF-13-005 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`second axis`	Give the inset its own ranges and dimensions.
`white background`	Make the zoomed panel independent of marks behind it.

PGF-13-006

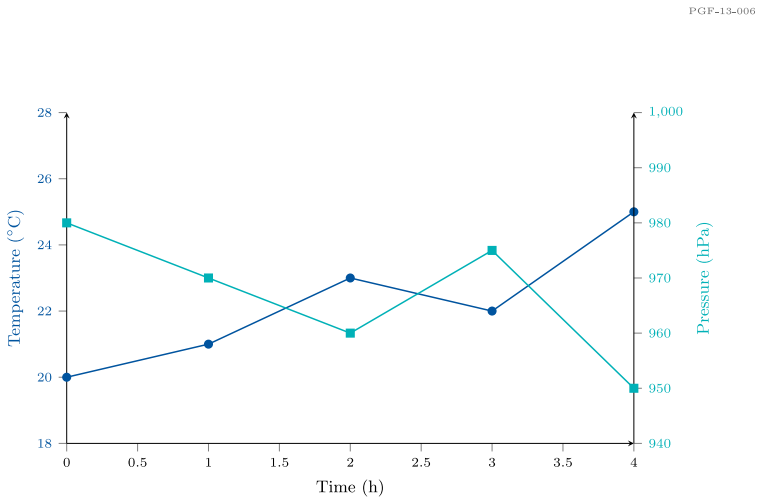
Two aligned unit systems

The two series share x coordinates but have different y units. Colours link each series to its own labelled axis; equal visual slopes are not equal physical changes.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xmin=0,
  xmax=4,
  ymin=18,
  ymax=28,
  axis y line*=left,
  axis x line*=bottom,
  xlabel={Time (h)},
  ylabel={Temperature ($^\circ$C)},
  ylabel style={text=Blue},
  yticklabel style={text=Blue}
]
  \addplot[Blue,thick,mark=*] coordinates {(0,20) (1,21) (2,23) (3,22) (4,25)};
\end{axis}
\begin{axis}[
  tour axis,
  xmin=0,
  xmax=4,
  ymin=940,
  ymax=1000,
  axis y line*=right,
  axis x line=none,
  ylabel={Pressure (hPa)},
  ylabel style={text=Teal},
  yticklabel style={text=Teal}
]
  \addplot[Teal,thick,mark=square*] coordinates {(0,980) (1,970) (2,960) (3,975) (4,950)};
\end{axis}
```

Actual output



PGF-13-006 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`same axis rectangle`	Align the x coordinate systems exactly.
`separate y labels`	Keep each series and unit system explicit.

Styles, typography and export

Reusable styles make a collection consistent. Native figure PDFs are vector output with TeX typography; the publisher also provides Poppler-derived SVG and PNG previews for reuse.

PGF-14-001

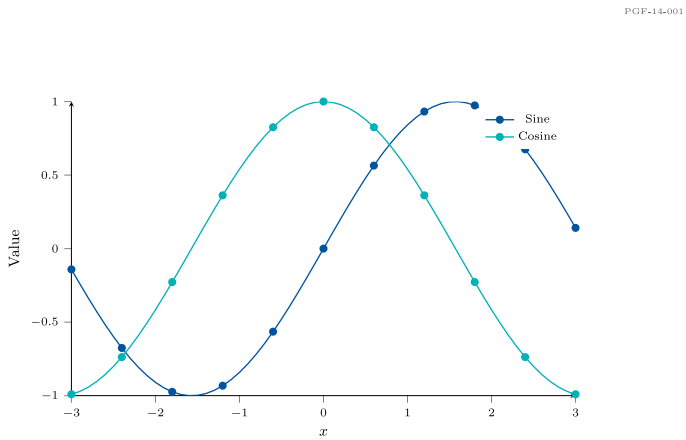
A reusable series style

One local style changes the stroke and marks of multiple series. The numerical data remain independent of presentation.

PGFplots / TikZ drawing

```
\pgfplotsset{response/.style={thick,mark=*,mark repeat=6,mark size=2pt}}
\begin{axis}[
  tour axis,
  xlabel={$x$},
  ylabel={Value}
]
  \addplot[response,Blue,domain=-3:3,samples=61] {sin(deg(x))};
  \addplot[response,Teal,domain=-3:3,samples=61] {cos(deg(x))};
  \legend{Sine,Cosine}
\end{axis}
```

Actual output



PGF-14-001 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`named style`	Reuse one set of drawing options.
`local pgfplotsset`	Keep this example independent of other documents.

PGF-14-002

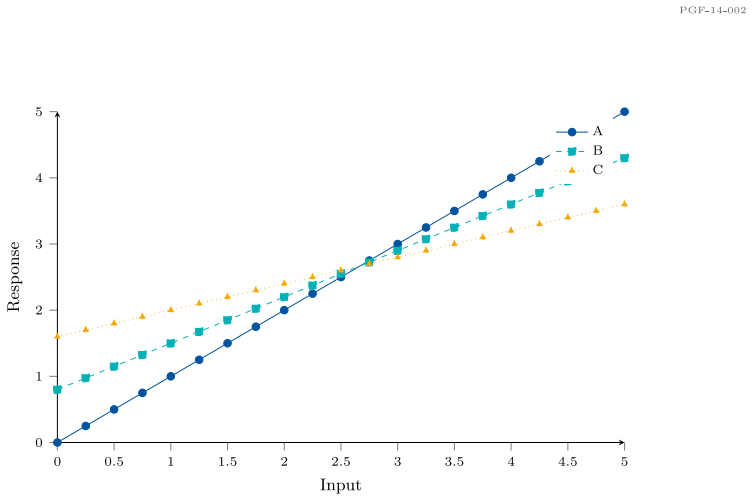
A coordinated cycle list

The cycle list supplies a colour, line style and marker for each new series. Redundant encodings support both colour and monochrome reading.

PGFplots / TikZ drawing

```
\pgfplotscreateplotcyclelist{responses}{
  Blue,solid,mark=*\\
  Teal,dashed,mark=square*\\
  Orange,dotted,mark=triangle*\\
}
\begin{axis}[
  tour axis,
  cycle list name=responses,
  xlabel={Input},
  ylabel={Response}
]
  \addplot+[domain=0:5,samples=21] {x};
  \addplot+[domain=0:5,samples=21] {.7*x+.8};
  \addplot+[domain=0:5,samples=21] {.4*x+1.6};
  \legend{A,B,C}
\end{axis}
```

Actual output



PGF-14-002 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`cycle list`	Assign a coordinated appearance to successive plots.
`addplot+`	Append explicit options to the selected cycle entry.

PGF-14-003

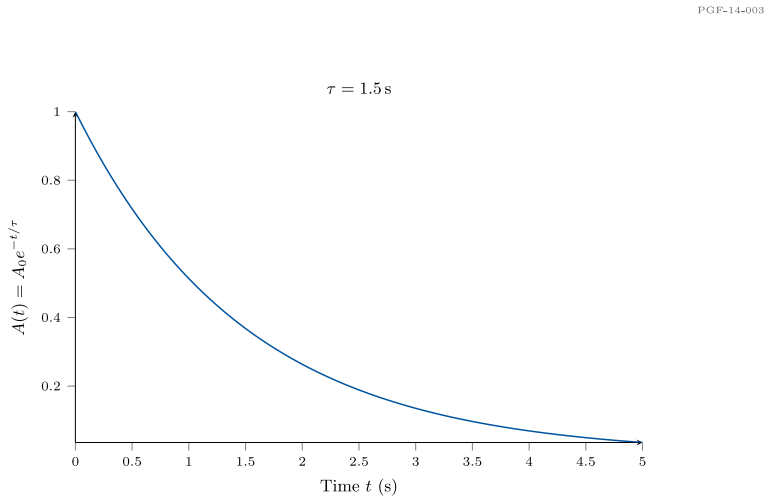
Mathematics in axis labels

TeX mathematics in labels stays consistent with the surrounding document. Explicit units prevent the symbolic expression from becoming ambiguous.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={Time $t$ (s)},
  ylabel={$A(t)=A_0 e^{-t/\tau}$},
  title={$\tau=1.5\,\mathrm{s}$}
]
  \addplot[Blue,thick,domain=0:5,samples=81] {exp(-x/1.5)};
\end{axis}
```

Actual output



PGF-14-003 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`math mode`	Use TeX mathematics directly in the labels.
`unit annotation`	State the unit separately from the variable name.

PGF-14-004

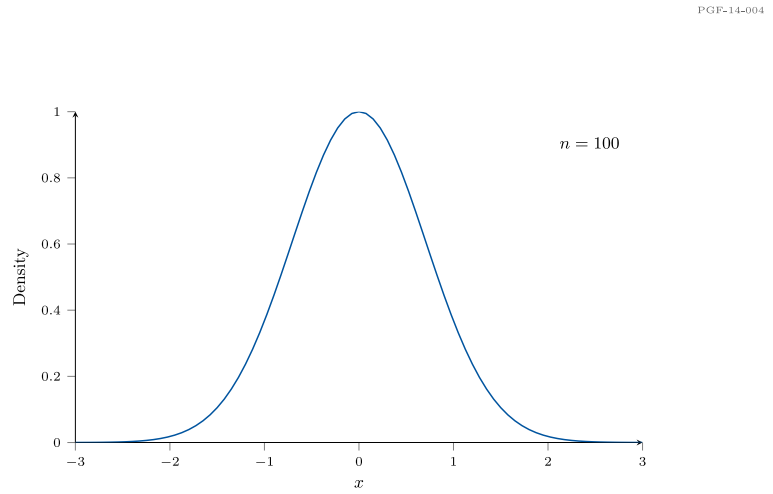
Place a note relative to the axis

Relative axis coordinates are independent of the data units. The note stays near the upper-right corner as the data domain changes.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xlabel={$x$},
  ylabel={Density}
]
  \addplot[Blue,thick,domain=-3:3,samples=81] {exp(-x*x)};
  \node[anchor=north east,fill=white,inner sep=3pt]
    at (rel axis cs:.97,.94) {$n=100$};
\end{axis}
```

Actual output



PGF-14-004 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`rel axis cs`	Use fractions of the axis rectangle rather than data values.
`anchor=north east`	Keep the note inside the selected corner.

PGF-14-005

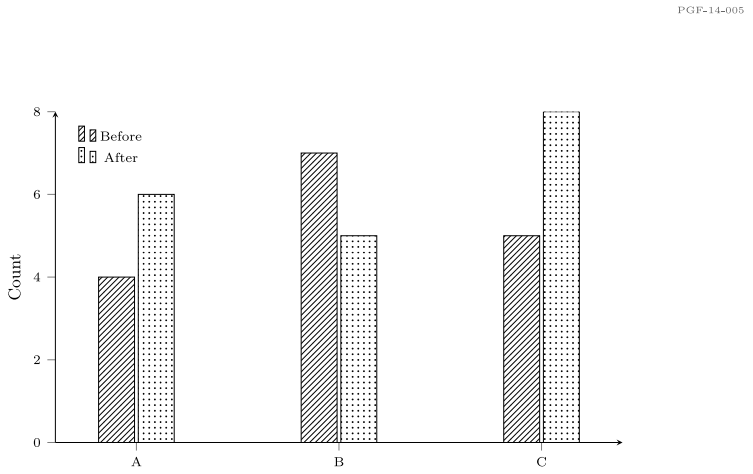
A monochrome bar style

Different hatch patterns can identify categories without relying on colour. Legend cells reproduce the corresponding fills.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  ybar,
  enlarge x limits=.2,
  bar width=19bp,
  ymin=0,
  xtick={1,2,3},
  xticklabels={A,B,C},
  ylabel={Count},
  legend pos=north west
]
  \addplot[pattern=north east lines,draw=black] coordinates {(1,4) (2,7) (3,5)};
  \addplot[pattern=dots,draw=black] coordinates {(1,6) (2,5) (3,8)};
  \legend{Before,After}
\end{axis}
```

Actual output



PGF-14-005 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`pattern`	Make category differences survive a monochrome print.
`legend sample`	Show the same fill convention beside each series label.

PGF-14-006

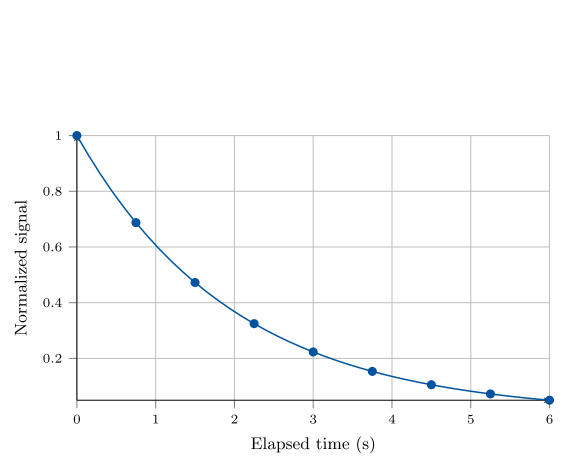
A compact figure for reuse

A smaller explicit data rectangle and restrained typography make this plot suitable for insertion into a larger document. Its downloaded PDF remains the native reference.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  at={(80bp,65bp)},
  width=250bp,
  height=140bp,
  xlabel={Elapsed time (s)},
  ylabel={Normalized signal},
  grid=major
]
  \addplot[Blue,thick,mark=*,mark repeat=5,domain=0:6,samples=41] {exp(-x/2)};
\end{axis}
```

Actual output



PGF-14-006 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`explicit dimensions`	Control the output before inserting it into another document.
`PDF / SVG / PNG`	Use the native PDF for vector print, SVG for web reuse and PNG for raster workflows.

More plot types with small samples

Start with a handful of explicit values. Stem plots, steps, polar curves, bubbles, prepared hexagonal cells and ternary proportions show different encodings without a large dataset.

PGF-15-001

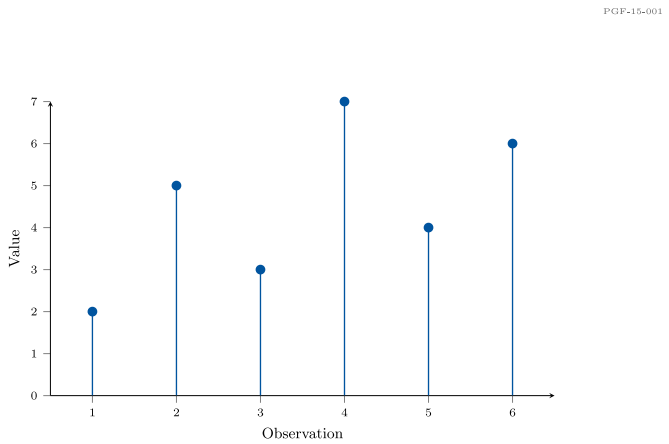
Discrete impulses with point markers

Six observations are drawn as stems from zero, with a marker at each value. The input is a short coordinate list.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  ymin=0,
  xmin=.5,
  xmax=6.5,
  xlabel={Observation},
  ylabel={Value}
]
  \addplot[ycomb,Blue,thick,mark=*,mark size=2.5pt] coordinates {
    (1,2) (2,5) (3,3) (4,7) (5,4) (6,6)
  };
\end{axis}
```

Actual output



PGF-15-001 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`ycomb`	Connect each value vertically to the zero baseline.
`mark=*`	Keep the discrete observations visible at the stem ends.

PGF-15-002

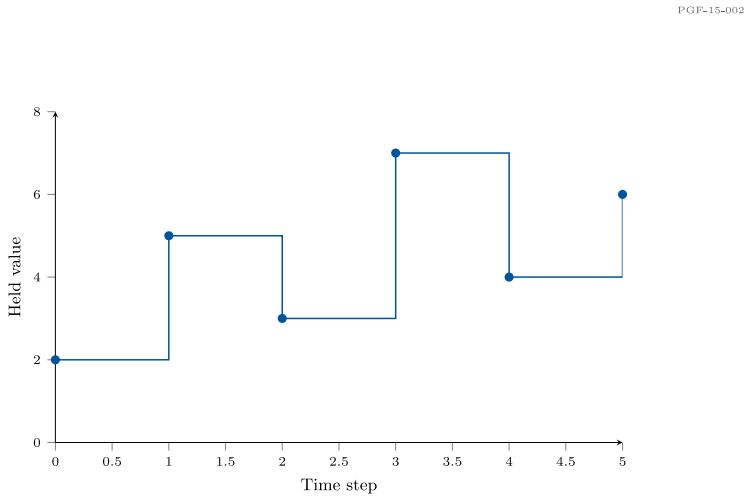
A step plot of held values

Each listed value is held over the interval to its right. The markers identify the original observations, rather than interpolated samples.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xmin=0,
  xmax=5,
  ymin=0,
  ymax=8,
  xlabel={Time step},
  ylabel={Held value}
]
  \addplot[const plot mark left,Blue,thick,mark=*] coordinates {
    (0,2) (1,5) (2,3) (3,7) (4,4) (5,6)
  };
\end{axis}
```

Actual output



PGF-15-002 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`const plot mark left`	Draw a constant segment from a sample towards the next x coordinate.
`markers`	Distinguish supplied observations from the horizontal holding intervals.

PGF-15-003

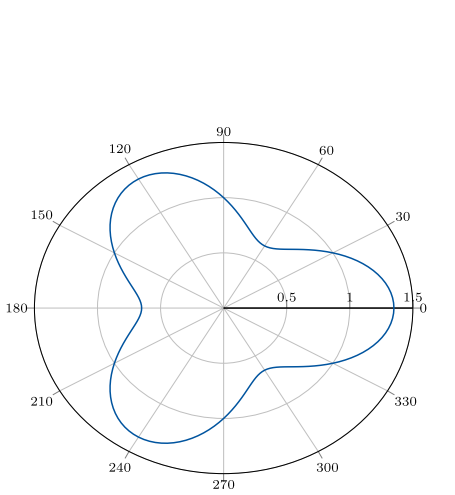
A polar radius function

The angle runs from zero to 360 degrees. A positive radius function makes a small three-lobed curve; the polar grid explains its coordinates.

PGFplots / TikZ drawing

```
\begin{polaraxis}[
  width=245bp,height=220bp,
  at={(75bp,25bp)},anchor=south west,
  grid=both,ymin=0,ymax=1.5,
  ytick={.5,1,1.5},tick label style={font=\scriptsize}
]
  \addplot[Blue,thick,domain=0:360,samples=181] {1+.35*cos(3*x)};
\end{polaraxis}
```

Actual output



PGF-15-003 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`polaraxis`	Interpret the horizontal coordinate as an angle and the vertical coordinate as radius.
`positive radius`	Keep the small example free of negative-radius folding.

PGF-15-004

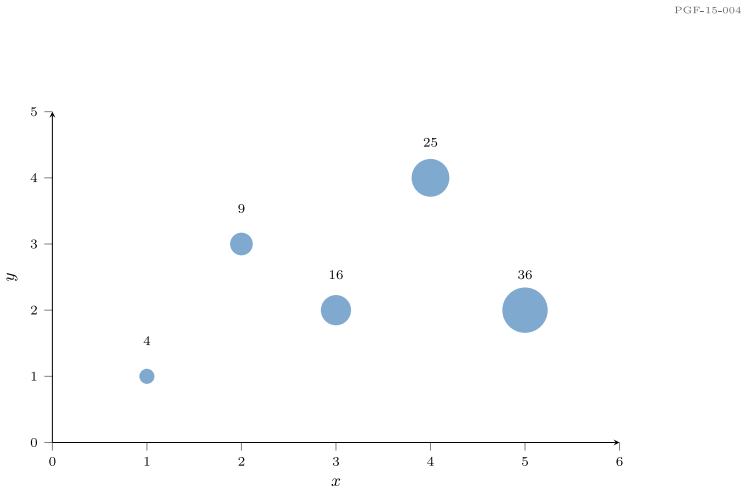
Bubble area from five values

Each circle area is proportional to its supplied n value: radius equals two times the square root of n. Labels expose those five numbers.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xmin=0,
  xmax=6,
  ymin=0,
  ymax=5,
  xlabel={$x$},
  ylabel={$y$}
]
  \pgfplotsextra{
  \foreach \x/\y/\n in {1/1/4,2/3/9,3/2/16,4/4/25,5/2/36} {
    \pgfmathsetmacro{\radius}{2*sqrt(\n)}
    \fill[Blue,opacity=.5] (axis cs:\x,\y) circle[radius=\radius bp];
    \node[above=13bp,font=\scriptsize] at (axis cs:\x,\y) {\n};
  }
  }
\end{axis}
```

Actual output



PGF-15-004 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`square-root radius`	Make circle area, rather than radius, proportional to n.
`explicit labels`	Show the small input values directly on the figure.

PGF-15-005

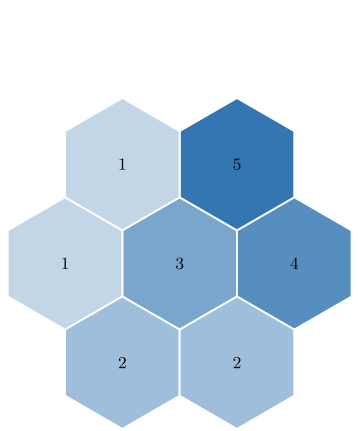
Prepared hexagonal bin counts

Seven regular hexagons share one lattice and one area. Their supplied counts determine the fill colour; this example draws prepared bins rather than binning raw observations.

PGFplots / TikZ drawing

```
\def\Radius{35}
\foreach \q/\r/\n in {0/0/3,1/0/4,-1/0/1,0/1/5,0/-1/2,1/-1/2,-1/1/1} {
  \pgfmathsetmacro{\cx}{210+sqrt(3)*\Radius*(\q+\r/2)}
  \pgfmathsetmacro{\cy}{140+1.5*\Radius*\r}
  \pgfmathsetmacro{\tone}{10+14*\n}
  \begin{scope}[shift={(\cx,\cy)}]
    \path[fill=Blue!\tone,draw=white,line width=1pt]
      (90:\Radius) -- (150:\Radius) -- (210:\Radius) --
      (270:\Radius) -- (330:\Radius) -- (30:\Radius) -- cycle;
    \node at (0,0) {\n};
  \end{scope}
}
```

Actual output



PGF-15-005 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`axial coordinates`	Place equal-sided hexagons on a consistent lattice.
`prepared counts`	Separate counting observations from drawing the cells.

PGF-15-006

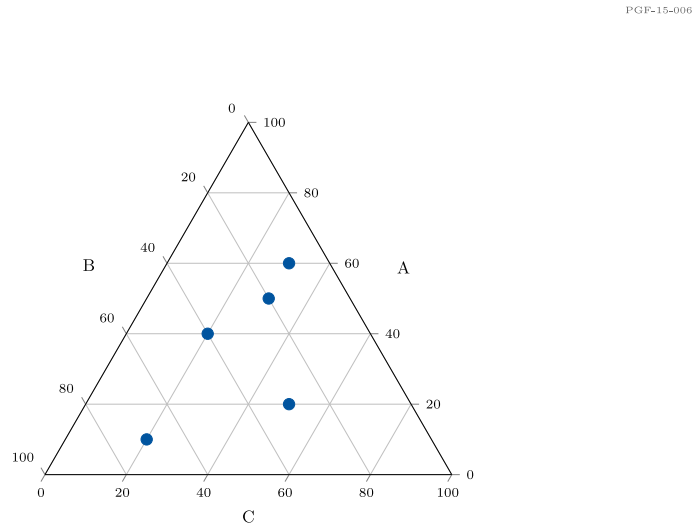
Ternary proportions

Every triple sums to one. The triangular coordinate system displays the relative shares of three components using only five observations.

PGFplots / TikZ drawing

```
\begin{ternaryaxis}[
  width=260bp,height=220bp,
  at={(70bp,25bp)},anchor=south west,
  xlabel={A},ylabel={B},zlabel={C},
  grid=major,tick label style={font=\scriptsize}
]
  \addplot3[only marks,Blue,mark=*,mark size=3pt] coordinates {
    (.2,.3,.5) (.5,.2,.3) (.1,.7,.2) (.4,.4,.2) (.6,.1,.3)
  };
\end{ternaryaxis}
```

Actual output



PGF-15-006 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`ternaryaxis`	Represent three nonnegative shares on a triangle.
`sum to one`	Make the compositional convention explicit in the input.

Specialized charts built from primitives

A chart does not always need a dedicated plotting command. These small, explicit constructions show how TikZ paths and PGFplots axes can express specialized comparisons. All data in this chapter are synthetic.

PGF-16-001

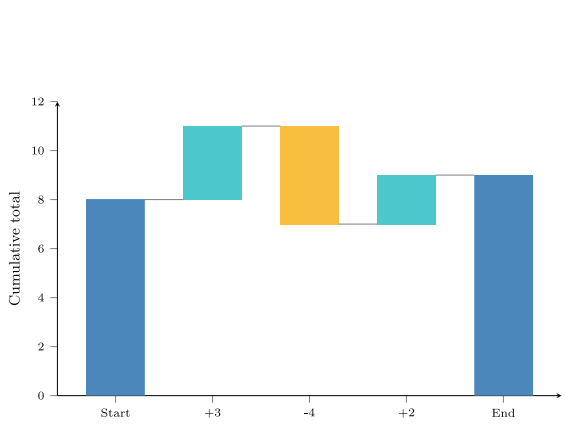
A waterfall of signed contributions

Each floating bar starts at the previous cumulative total. Grey connectors make the accounting sequence explicit.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xmin=.4,
  xmax=5.6,
  ymin=0,
  ymax=12,
  xtick={1,2,3,4,5},
  xticklabels={Start,+3,-4,+2,End},
  ylabel={Cumulative total}
]
  \path[fill=Blue!70] (axis cs:.7,0) rectangle (axis cs:1.3,8);
  \path[fill=Teal!70] (axis cs:1.7,8) rectangle (axis cs:2.3,11);
  \path[fill=Orange!75] (axis cs:2.7,7) rectangle (axis cs:3.3,11);
  \path[fill=Teal!70] (axis cs:3.7,7) rectangle (axis cs:4.3,9);
  \path[fill=Blue!70] (axis cs:4.7,0) rectangle (axis cs:5.3,9);
  \draw[gray] (axis cs:1.3,8) -- (axis cs:1.7,8);
  \draw[gray] (axis cs:2.3,11) -- (axis cs:2.7,11);
  \draw[gray] (axis cs:3.3,7) -- (axis cs:3.7,7);
  \draw[gray] (axis cs:4.3,9) -- (axis cs:4.7,9);
\end{axis}
```

Actual output



PGF-16-001 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`floating rectangles`	Place each contribution between its starting and ending cumulative levels.
`connectors`	Expose the sequence of accumulated totals.

PGF-16-002

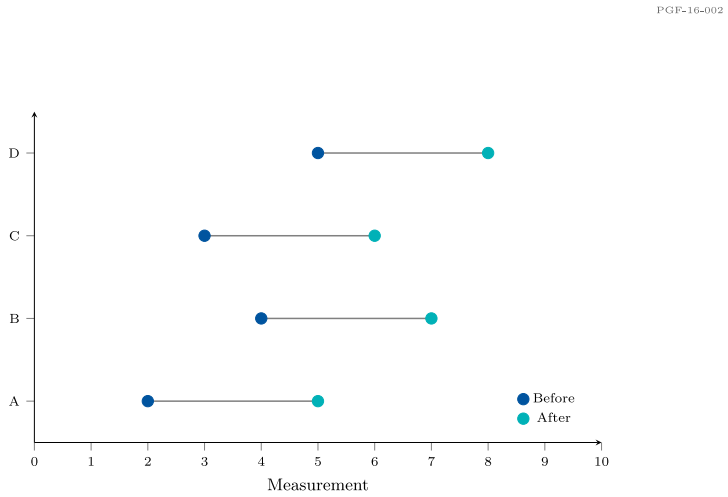
A dumbbell comparison

Each horizontal segment connects a pair of measurements for the same group. Colours and marker positions distinguish before and after.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xmin=0,
  xmax=10,
  ymin=.5,
  ymax=4.5,
  ytick={1,2,3,4},
  yticklabels={A,B,C,D},
  xlabel={Measurement},
  legend pos=south east
]
  \pgfplotsextra{
    \foreach \y/\a/\b in {1/2/5,2/4/7,3/3/6,4/5/8}
      \draw[gray,thick] (axis cs:\a,\y) -- (axis cs:\b,\y);
  }
  \addplot[only marks,Blue,mark=*,mark size=3pt]
    coordinates {(2,1) (4,2) (3,3) (5,4)};
  \addplot[only marks,Teal,mark=*,mark size=3pt]
    coordinates {(5,1) (7,2) (6,3) (8,4)};
  \legend{Before,After}
\end{axis}
```

Actual output



PGF-16-002 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`paired connectors`	Keep each comparison within its own category.
`two mark series`	Apply one consistent encoding to each endpoint type.

PGF-16-003

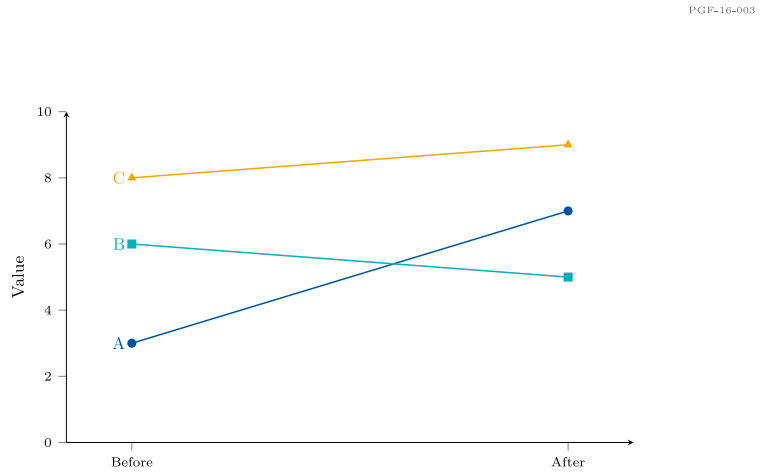
A slopegraph with direct labels

Two aligned columns show paired values. Direct labels connect the lines to their categories without a separate lookup legend.

PGFplots / TikZ drawing

```
\begin{axis}[
  tour axis,
  xmin=-.15,
  xmax=1.15,
  ymin=0,
  ymax=10,
  xtick={0,1},
  xticklabels={Before,After},
  ylabel={Value}
]
  \addplot[Blue,thick,mark=*] coordinates {(0,3) (1,7)};
  \addplot[Teal,thick,mark=square*] coordinates {(0,6) (1,5)};
  \addplot[Orange,thick,mark=triangle*] coordinates {(0,8) (1,9)};
  \node[anchor=east,text=Blue] at (axis cs:0,3) {A};
  \node[anchor=east,text=Teal] at (axis cs:0,6) {B};
  \node[anchor=east,text=Orange] at (axis cs:0,8) {C};
\end{axis}
```

Actual output



PGF-16-003 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`two x positions`	Align the start and end of each pair.
`direct labels`	Name each series beside its first point.

PGF-16-004

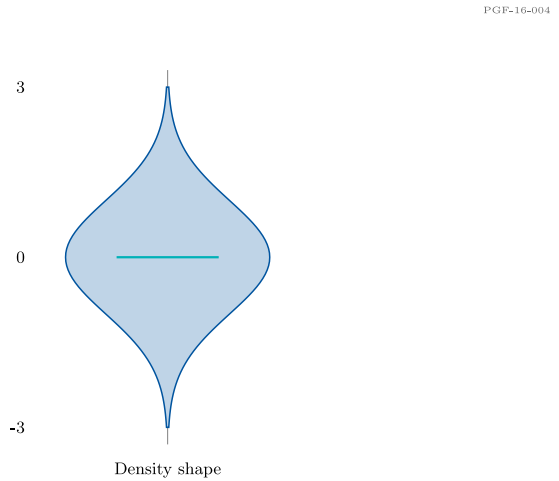
A mirrored density silhouette

Two reflected paths construct a violin-like silhouette from an analytic density shape. Width here is illustrative and is not inferred from observations.

PGFplots / TikZ drawing

```
\begin{scope}[shift={(210,140)},x=90bp,y=30bp]
  \draw[gray] (0,-3.3) -- (0,3.3);
  \filldraw[fill=Blue!25,draw=Blue,thick]
    plot[domain=-3:3,samples=81,variable=\t]
      ({.6*exp(-\t*\t/2)},{\t}) --
    plot[domain=3:-3,samples=81,variable=\t]
      ({-.6*exp(-\t*\t/2)},{\t}) -- cycle;
  \draw[Teal,very thick] (-.3,0) -- (.3,0);
  \foreach \v in {-3,0,3}
    \node[anchor=east] at (-.8,\v) {\v};
  \node[below] at (0,-3.5) {Density shape};
\end{scope}
```

Actual output



PGF-16-004 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`reflected paths`	Join equal positive and negative width functions.
`explicit density`	State the width function rather than implying a sample-based KDE.

PGF-16-005

A conserved two-way flow

A source total of 100 splits into 60 and 40. Ribbon thickness encodes these supplied quantities; the diagram does not optimize its layout.

PGFplots / TikZ drawing

```
\path[fill=Blue!35] (85,120) -- (85,180)
  .. controls (185,180) and (235,180) .. (320,180)
  -- (320,120) .. controls (235,120) and (185,120) .. cycle;
\path[fill=Teal!35] (85,80) -- (85,120)
  .. controls (185,120) and (235,95) .. (320,95)
  -- (320,55) .. controls (235,55) and (185,80) .. cycle;
\fill[Blue] (65,80) rectangle (85,180);
\fill[Blue] (320,120) rectangle (340,180);
\fill[Teal] (320,55) rectangle (340,95);
\node[anchor=east] at (60,130) {100};
\node[anchor=west] at (345,150) {60};
\node[anchor=west] at (345,75) {40};
```

Actual output



PGF-16-005 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`ribbon thickness`	Encode the supplied flow with a consistent scale.
`conservation`	Make the outgoing quantities sum to the source total.

PGF-16-006

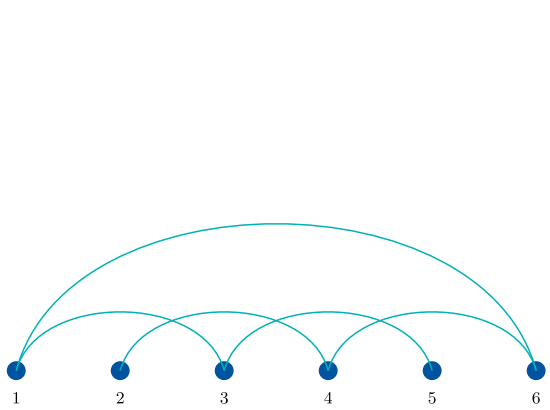
An arc diagram of relationships

Nodes share one baseline. Curved edges show selected relationships; crossing arcs are drawing choices, not hidden data values.

PGFplots / TikZ drawing

```
\foreach \i in {1,...,6} {
  \coordinate (n\i) at (40+55*\i,80);
  \fill[Blue] (n\i) circle[radius=5];
  \node[below=8bp] at (n\i) {\i};
}
\foreach \a/\b in {1/3,1/6,2/4,3/5,4/6}
  \draw[Teal,thick] (n\a) to[out=75,in=105] (n\b);
```

Actual output



PGF-16-006 · Rendered by PGFplots / TikZ

PGFplots / TikZ · exit 0

Command or parameter	Meaning
`shared baseline`	Make node order an explicit part of the layout.
`curved edge controls`	Choose how far relationships bend above the node row.

Example and command index

- PGF-01-001 · A quadratic on automatic axes
- PGF-01-002 · Add a major grid
- PGF-01-003 · Change the line and markers
- PGF-01-004 · A sine curve with radian labels
- PGF-01-005 · Two functions and a legend
- PGF-01-006 · A measured-size figure
- PGF-02-001 · Lines and a closed polygon
- PGF-02-002 · Bezier control points
- PGF-02-003 · Circles, ellipses and arcs
- PGF-02-004 · Fill rules and holes
- PGF-02-005 · Gradient and transparent overlap
- PGF-02-006 · Patterned categorical regions
- PGF-03-001 · A three-step workflow
- PGF-03-002 · Branch and merge
- PGF-03-003 · Curved feedback
- PGF-03-004 · Fit a group boundary
- PGF-03-005 · A labelled tree
- PGF-03-006 · An aligned matrix of nodes
- PGF-04-001 · Translate and rotate a motif
- PGF-04-002 · Scale coordinates and node shapes
- PGF-04-003 · Interpolate between coordinates
- PGF-04-004 · Named path intersections
- PGF-04-005 · Clip a repeated field
- PGF-04-006 · A polar coordinate construction
- PGF-05-001 · A linear reference frame
- PGF-05-002 · A logarithmic horizontal axis
- PGF-05-003 · A log-log power law
- PGF-05-004 · Custom ticks and units
- PGF-05-005 · Scientific notation and shared factors
- PGF-05-006 · Equal coordinate units
- PGF-06-001 · Plot named columns
- PGF-06-002 · Transform a column expression
- PGF-06-003 · Skip a selected interval
- PGF-06-004 · Read comma-separated input
- PGF-06-005 · Handle an explicit missing observation
- PGF-06-006 · Typeset the input table
- PGF-07-001 · Markers without connecting lines
- PGF-07-002 · Encode categories with marker styles
- PGF-07-003 · Colour points by a numeric value
- PGF-07-004 · Vertical symmetric error bars
- PGF-07-005 · Asymmetric intervals
- PGF-07-006 · Annotate a selected observation
- PGF-08-001 · Vertical category bars
- PGF-08-002 · Grouped bars
- PGF-08-003 · Stacked counts
- PGF-08-004 · A normalized composition
- PGF-08-005 · Horizontal ranked bars
- PGF-08-006 · Diverging changes around zero
- PGF-09-001 · A histogram with fixed bin edges
- PGF-09-002 · Prepared box-and-whisker summaries
- PGF-09-003 · A boxplot calculated from samples
- PGF-09-004 · An explicit right-continuous ECDF
- PGF-09-005 · A density curve with a reference interval
- PGF-09-006 · Point estimates with confidence intervals

- PGF-10-001 · An area under a curve
- PGF-10-002 · A band between named curves
- PGF-10-003 · Restrict a band to an interval
- PGF-10-004 · Stacked areas
- PGF-10-005 · Positive and negative filled regions
- PGF-10-006 · A hatched interval overlay
- PGF-11-001 · An explicit grid of cell values
- PGF-11-002 · A smooth function viewed from above
- PGF-11-003 · Choose discrete colour classes
- PGF-11-004 · Overlay analytic isolines
- PGF-11-005 · Print numeric cell labels
- PGF-11-006 · A directional vector field
- PGF-12-001 · A wireframe saddle
- PGF-12-002 · A smoothly shaded Gaussian surface
- PGF-12-003 · Change only the camera
- PGF-12-004 · A parametric helix
- PGF-12-005 · Three-dimensional observations
- PGF-12-006 · A faceted surface with visible patches
- PGF-13-001 · Two neighbouring panels
- PGF-13-002 · Keep labels on the outside
- PGF-13-003 · A two-by-two small-multiple figure
- PGF-13-004 · One shared legend
- PGF-13-005 · An inset with an explicit data window
- PGF-13-006 · Two aligned unit systems
- PGF-14-001 · A reusable series style
- PGF-14-002 · A coordinated cycle list
- PGF-14-003 · Mathematics in axis labels
- PGF-14-004 · Place a note relative to the axis
- PGF-14-005 · A monochrome bar style
- PGF-14-006 · A compact figure for reuse
- PGF-15-001 · Discrete impulses with point markers
- PGF-15-002 · A step plot of held values
- PGF-15-003 · A polar radius function
- PGF-15-004 · Bubble area from five values
- PGF-15-005 · Prepared hexagonal bin counts
- PGF-15-006 · Ternary proportions
- PGF-16-001 · A waterfall of signed contributions
- PGF-16-002 · A dumbbell comparison
- PGF-16-003 · A slopegraph with direct labels
- PGF-16-004 · A mirrored density silhouette
- PGF-16-005 · A conserved two-way flow
- PGF-16-006 · An arc diagram of relationships

Content: CC0 1.0. Structure and publishing framework: CC BY 4.0 by Lukas C. Bossert. Repository source · Publishing framework source · License scope and third-party notices.