



PGFPlots in $\text{\LaTeX}$

Prof. Pranay Kumar Saha

April 9, 2025





What is PGFPlots?

High-Quality Data Visualization

- `pgfplots` is a powerful $\text{\LaTeX}$ package for creating scientific plots and charts directly within your document.
- It builds upon the **TikZ** drawing package (`pgf/tikz`).
- It aims to provide a user-friendly interface for common plotting tasks while leveraging the typesetting quality of $\text{\TeX}$.



What is PGFPlots?

High-Quality Data Visualization

Why Use PGFPlots?

- **Consistency:** Plots use the same fonts and styles as your main document text and math.
- **Quality:** Produces vector graphics (PDF/EPS) suitable for publication. Excellent math typesetting in labels/ticks.
- **Integration:** Code is directly within your `.tex` file. Easy label/caption integration.
- **Data Handling:** Can plot mathematical functions, inline data, or data from external files.
- **Customization:** Highly customizable appearance (styles, colors, markers, axes, legends, etc.).



Loading the Package

Basic Setup

Using `pgfplots` requires loading the package in your preamble.

Preamble Code

```
\usepackage{pgfplots}

% Recommended: Set compatibility mode for stable behavior
% configure the behaviour of pgfplots in the document preamble
\pgfplotsset{width=10cm,compat=1.9}
```

- `\usepackage{pgfplots}` automatically loads the underlying TikZ package.
- `\pgfplotsset{compat=...}` ensures your plots use features and defaults from a specific version, preventing unexpected changes if you update the package later. Use a version you have installed.



The Core Environments

`tikzpicture` and `axis`

Plots are created within a `tikzpicture` environment (from TikZ), which contains an `axis` environment (from PGFPlots).

Basic Structure

```
\begin{tikzpicture}
  \begin{axis}[
    % Axis options go here (e.g., labels, title)
  ]
    % Plotting commands (\addplot) go here
  \end{axis}
\end{tikzpicture}
```



The Core Environments

`tikzpicture` and `axis`

`tikzpicture`:

- The drawing canvas provided by TikZ.
- Can contain TikZ drawing commands alongside the plot.

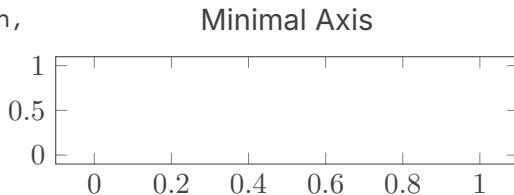
`axis`:

- The main PGFPlots environment.
- Automatically draws axes, ticks, and grids.
- Takes options (in `[ ]`) to control appearance.
- Contains one or more `\addplot` commands.



The Core Environments

```
\begin{tikzpicture}
  \begin{axis}[width=0.6\textwidth,
              height=3cm,
              title=Minimal Axis]
    % No plots yet
  \end{axis}
\end{tikzpicture}
```



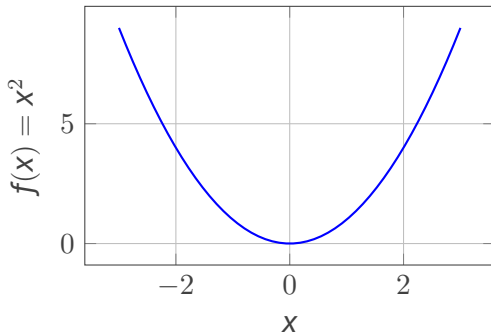


Plotting Mathematical Expressions

Using `\addplot {expression};`

The easiest way to plot a function is to provide the mathematical expression directly.

```
1 \begin{tikzpicture}
2 \begin{axis}[ width=0.9\textwidth,% Adjust size
3   height=5cm,
4   xlabel={\mathit{x}}, % Math in labels!
5   ylabel={\mathit{f(x) = x^2}},
6   grid=major, % Add grid lines ]
7 \addplot[
8   domain=-3:3, % Range for x
9   samples=50, % Number of points
10  blue, thick % Line color & thickness]
11  {\mathit{x^2}}; % The function
12 \end{axis}
13 \end{tikzpicture}
```





Plotting Mathematical Expressions

Using `\addplot {expression};`

Key Points:

- Expression uses x as the variable.
- domain, samples control calculation range/smoothness.
- Options for `\addplot` control line style (color, thickness, markers).

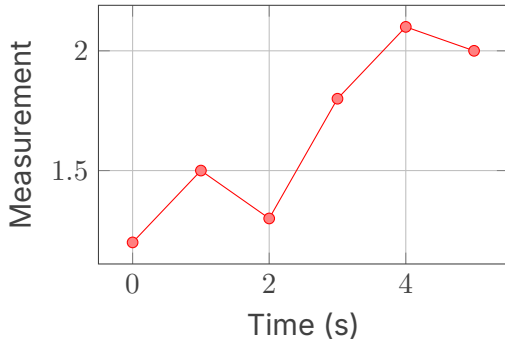


Plotting Inline Coordinates

Using `\addplot coordinates {...};`

You can provide explicit data points directly in your code.

```
1 \begin{tikzpicture}
2 \begin{axis}[ width=0.9\textwidth, height=5cm,
3   xlabel={Time (s)}, ylabel={Measurement},
4   grid=major,]
5 \addplot[
6   red,           % Color
7   mark=*,       % Marker type
8   mark options={fill=red!50!white}]
9   coordinates { % List of (x,y) pairs
10    (0, 1.2) (1, 1.5) (2, 1.3)
11    (3, 1.8) (4, 2.1) (5, 2.0)
12 }; % Semicolon is important!
13 \end{axis}
```





Plotting Inline Coordinates

Using `\addplot coordinates {...};`

Key Points:

- Points are (x,y) pairs separated by spaces.
- Good for small datasets.
- Use plot options like `mark=*` to show data points.
- *: A small point, x: A cross, +: A plus sign, o: A circle, asterisk: An asterisk, star: A star



Plotting Data from Files

Using `\addplot table {...};`

For larger datasets, read data from external text files.

Sample Data File (mydata.dat)

```
# Simple data file (comments ignored)
```

X_Val	Y_Val	Error
1	2.5	0.2
2	3.8	0.3
3	4.1	0.25
4	5.5	0.4
5	6.2	0.5

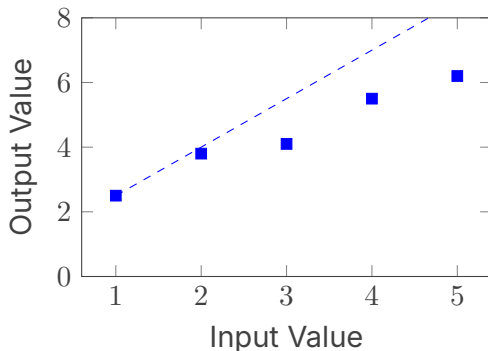


Plotting Data from Files

Using `\addplot table {...};`

For larger datasets, read data from external text files.

```
1 \begin{tikzpicture}
2 \begin{axis}[
3   width=0.9\textwidth, height=5cm,
4   xlabel={Input Value},
5   ylabel={Output Value},]
6   % Plot data using column names
7   \addplot[blue, mark=square*]
8     table[x=X_Val, y=Y_Val]
9     {mydata.dat}; % The data file
10   \addplot[blue, mark=none, domain=1:5,
11     samples=2, dashed] {1.5*x+1};
12 \end{axis}
13 \end{tikzpicture}
```





Plotting Data from Files

Using `\addplot table {...};`

Note: You need to create the `mydata.dat` file in the same directory as your `.tex` file. **Key Points:**

- `table {filename}` reads the data.
- Use `x=ColName`, `y=ColName` to specify columns (or indices `x index=0`, `y index=1`).
- Handles comments (lines starting with `#` or `%`).
- Best practice for managing experimental data.

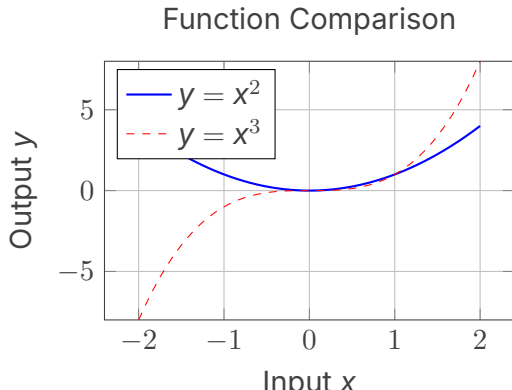


Labels, Titles, and Legends

Annotating Your Plot

Add descriptive elements using options to the `axis` environment.

```
1 \begin{tikzpicture}
2 \begin{axis}[
3   width=0.9\textwidth, height=5cm,
4   title={Function Comparison}, % Plot title
5   xlabel={Input  $x$ },          % X-axis label
6   ylabel={Output  $y$ },          % Y-axis label
7   grid=major,
8   legend pos=north west, % Legend position]
9   \addplot[blue, thick, domain=-2:2] % First Plot
10    {x^2};
11   \addlegendentry{$y = x^2$} % Legend entry
12   \addplot[red, dashed, domain=-2:2] % Second Plot
13    {x^3};
14   \addlegendentry{$y = x^3$}
```





Labels, Titles, and Legends

Annotating Your Plot

Key Axis Options:

- `title={...}`
- `xlabel={...}`, `ylabel={...}`, `zlabel={...}` (for 3D)
- `legend pos=...` (e.g., 'north west', 'south east', 'outer north east')
- Use `\addlegendentry{...}` after each `\addplot` to label it in the legend.
Math mode works here too!



Data Plots: Histograms (via PGFPlots)

Using the 'pgfplots' Package

While TikZ can draw anything, for data plots like histograms, the `pgfplots` package (built on TikZ) is much more convenient. It handles data binning and axis scaling automatically.

Required Packages (Preamble)

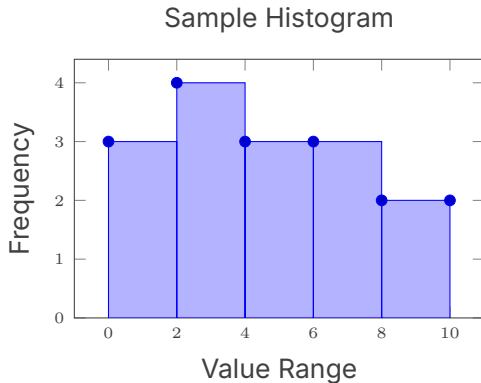
```
\usepackage{pgfplots}
\pgfplotsset{compat=1.18} % Use appropriate version
\usepgfplotslibrary{statistics} % Needed for 'hist' option
```



Data Plots: Histograms (via PGFPlots)

Using the 'pgfplots' Package

```
1 \begin{tikzpicture}
2 \begin{axis}[
3   width=1.0\textwidth, height=5cm, % Adjust size
4   ymin=0, % Start y-axis at 0
5   xlabel={Value Range},
6   ylabel={Frequency},
7   title={Sample Histogram},
8 ]
9 \addplot+ [ % '+' helps with default styles
10  hist={
11    bins=5, % Number of bins
12    data min=0, data max=10 % Range of data for binning
13  },
14  fill=blue!30 % Fill color for bars]
15  table [y index=0] { % Data is in the first column
16    Data % Column header (ignored by y index=0)
17    1.5
18    2.1
19    8.5
20    3.5
21    4.1
22    5.5 }; % Semicolon needed!
23 \end{axis}
24 \end{tikzpicture}
```





Data Plots: Histograms (via PGFPlots)

Using the 'pgfplots' Package

Note: Refer to the pgfplots manual (texdoc pgfplots) for many more options! **Key Points:**

- Uses axis environment from pgfplots.
- Requires `\usepgfplotslibrary{statistics}`.
- `\addplot+ [hist={options}] ...` draws the histogram.
- Key hist options: bins (number), data min, data max.
- Data provided via table [`y index=0`] for raw data points (or from a file).



A Glimpse of PGFPlots' Power

PGFPlots can do much more:

- **Plot Types:** Scatter plots, contour plots, surface plots (3D), quiver plots, box plots, ...
- **Custom Styles:** Define reusable styles for plots and axes ("").
- **Markers and Lines:** Extensive control over marker shapes, sizes, colors, line styles, line thickness.
- **Error Bars:** Add error bars to data plots.
- **Axis Control:** Logarithmic axes ('xmode=log'), date axes, broken axes, custom tick labels, axis positioning.
- **Color Maps:** Use color maps for 3D plots or scatter plots with a third value.
- **Filling:** Fill area under/between curves.
- **Multiple Axes:** Plot data with different scales using multiple y-axes or x-axes.
- **TikZ Integration:** Combine plots with arbitrary TikZ drawings (arrows, annotations, shapes).



PGFPlots in $\text{\LaTeX}$

Thank You for Listening!