

PrettyPlots in L^AT_EX

EDA group

July 16, 2025

Contents

1	Introduction	2
2	Quick Reference	4
3	Plots	6
3.1	Line	6
3.2	Box	7
3.3	Bar	8
3.4	Scatter	10
3.5	Error Line	11
3.6	Group Plots	13
3.7	Examples	14
4	Colors and Style	18
4.1	Color Palettes	18
4.2	Custom Colors	19
4.3	Line Style	20
4.4	Scatter Classes	21
4.5	Grids	23
4.6	Labels	24
4.7	Highlighting	26
4.8	Axis Style	28
5	Advanced Usage	29

1 Introduction

This package provides tools for creating easy-to-read plots in Latex. To use it, include

```
\usepackage{prettyplots}
```

in your preamble. Then, you can use it together with the `pgfplots` package,

```
\usepackage{tikz}
\usepackage{pgfplots}
\pgfplotsset{compat=1.18}
```

where it is recommended to specify a version using `compat` to avoid backward compatibility issues with `pgfplots` features. It will be useful to have some basic familiarity with `pgfplots`¹.

Basic Usage

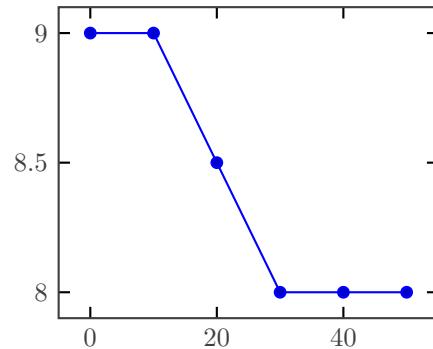
Let's say you want to create a line plot from a dataset of the following shape,

```
/data/line.tsv
```

```
%Metric      Method
x          y1
0           9
10          9
20          8.5
...
...
```

Using `pgfplots`, you can generate a line plot as follows.

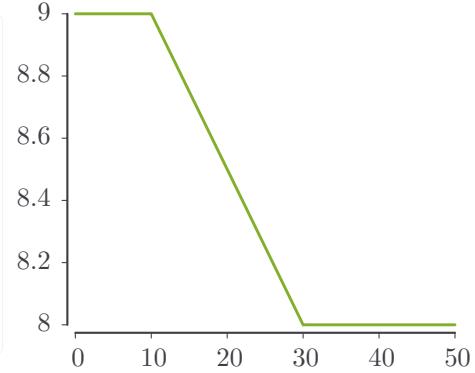
```
\begin{tikzpicture}
  \begin{axis}[]
    \addplot table [x=x, y=y1] {data/line.tsv};
  \end{axis}
\end{tikzpicture}
```



¹<https://pgfplots.sourceforge.net/pgfplots.pdf>

To turn this into a pretty plot, add the `pretty line` command to the pgf axis keys,

```
%preamble
\usepackage{prettyplots}
%
\begin{tikzpicture}
\begin{axis}[pretty line]
\addplot table [x=x, y=y1] {data/line.tsv};
\end{axis}
\end{tikzpicture}
```



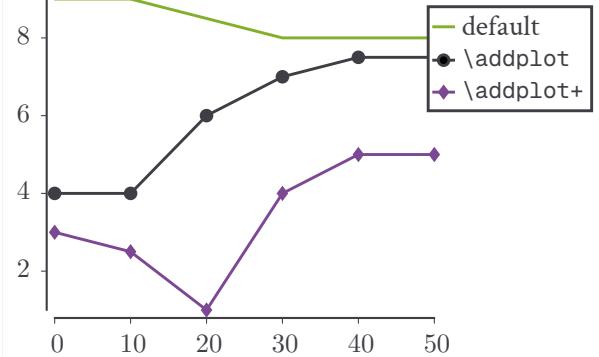
Basic Customization

You can keep using the common pgfplots keys to change the appearance of plots. The two main ways to do this are passing keys to `axis` to change the global appearance of the axis environment, or passing keys to `addplot` to change the appearance of specific lines or plots.

- `axis` keys: There are many axis keys you can find in pgfplots. For example, you can set colors and line style to iterate through using a `cycle list`, such as `cycle list=pr-line` which is set by default.
- `addplot` keys: arguments you pass to `addplot` change the appearance of each plot that you add to the axis. This overwrites any style defined in the global axis environment, so use `addplot+` if you want to preserve the default style defined in the axis environment.

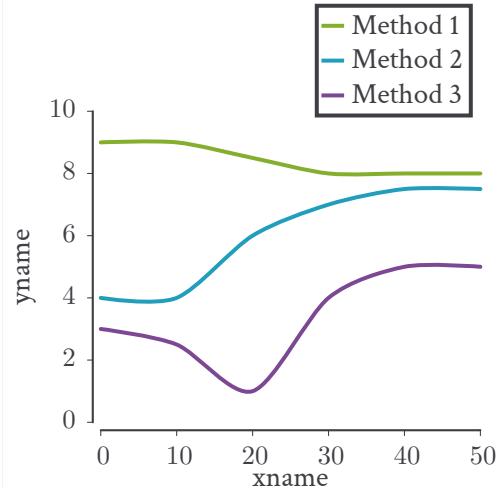
For example, we can add a line using default style, another with markers, and another that keeps colors unchanged from the default cycle list `pr-colors`.

```
\begin{tikzpicture}
\begin{axis}[
    pretty line,
    cycle list name = pr-colors,
    legend style = {anchor=north west}
]
\addplot table [x=x, y=y1] {data/line.tsv};
\addlegendentry{default};
\addplot[mark=*, mark size=2pt]
table [x=x, y=y2] {data/line.tsv};
\addlegendentry{\texttt{\tbs addplot}};
\addplot+[mark=diamond*, mark size=2pt]
table [x=x, y=y3] {data/line.tsv};
\addlegendentry{\texttt{\tbs addplot+}};
\end{axis}
\end{tikzpicture}
```



Here is a larger example using some common pgfplots axis keys to customize sizes, positioning, and labels.

```
\begin{tikzpicture}
\begin{axis}[
    pretty line,
    width = \textwidth,
    xlabel = {xname},
    ylabel = {yname},
    ytick = {0,2,4,6,8,10},
    ymin = 0, ymax = 10,
    legend style = {anchor=south east},
    legend entries =
        {Method 1, Method 2, Method 3},
    smooth,
    line width = 1.5pt,
]
\foreach \i in {1,2,3} {
    \addplot table [x=x, y=y\i] {data/line.tsv};
}
\end{axis}
\end{tikzpicture}
```



To create a different type of plot, modify the command depending on the type using `pretty <plotname>`. The main axis environments are shown in Section 3, color schemes in Section 4, and macros in Section 5.

2 Quick Reference

Prettyplots

Table 1: Plot styles in `prettyplots`

Style Name	Use Case
<code>pretty line</code>	Line plots
<code>pretty boxplot</code>	Box plots for distributions
<code>pretty xbar</code>	Horizontal bar plots
<code>pretty ybar</code>	Vertical bar plots
<code>pretty ybar stacked</code>	Stacked bar plots
<code>pretty scatter</code>	Scatter plots, style for discrete labels
<code>pretty scatter colormap</code>	Scatter plots, style for continuous labels
<code>pretty axis</code>	Base style (sizing, placement, ticks, etc.)

Pgfplots

Table 2: Useful pgfplots axis keys

Category	Key	Description
Size & Layout	width=\textwidth	Set plot width
	height=6cm	Optional: set plot height
	scale only axis=true	Prevent label size from scaling axis
	enlargelimits=0.1	Add margin around data
Labels	clip=false	Allow overflow (e.g., annotations)
	xlabel={X label}	Label for x-axis
	ylabel={Y label}	Label for y-axis
	title={Plot Title}	Optional: add plot title
Ticks	label style={font=\bfseries}	Bold axis labels
	xtick={0, 1, ..., 10}	Set specific x-tick positions
	yticklabels={Low, Mid, High}	Custom y-axis tick labels
	tick align=outside	Align ticks outside the axis
	tick label style={font=\small}	Style tick label appearance
	xticklabel style={rotate=45}	Rotate x-axis tick labels
Axis Range	minor tick num=1	Add minor ticks between major ticks
	xmin=0, xmax=10	Set visible x-axis range
	ymin=0, ymax=10	Set visible y-axis range
	legend entries={A, B, C}	Set legend items manually
Legend	legend pos=north east	Quick legend placement
	legend style={anchor=south east}	Custom placement/formatting
	legend cell align=left	Align legend text to the left
	smooth	Smooth lines between data points
Line & Style	line width=1.5pt	Control line thickness
	mark options={fill=white, size=2pt}	Customize markers
	cycle list name=pr-colors	Use a predefined style/color cycle
	cycle list={red, blue, green}	Manual color/style cycle
	every axis plot/.append style=...	Apply style to all plots
	grid=major	Enable major gridlines
Grid & Axis	minor grid style={dashed, gray!30}	Style minor gridlines
	axis lines=left	Axes only on bottom/left (math-style)
	axis on top=true	Draw axes over data/background
	xmode=log, ymode=log	Logarithmic axis scaling
Log Axes	log basis x=10	Base of logarithm (x-axis)
	scaled x ticks=base 10:3	Scientific notation (e.g., 10^3)

3 Plots

This section shows examples for each main plot type.

3.1 Line

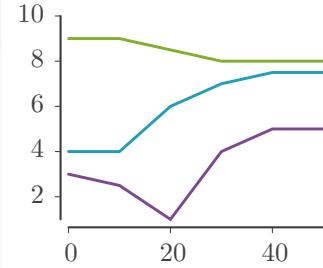
For basic line plots, you can pass `pretty line` to the axis keys.

```
/data/line.tsv
```

```
%Metric      Method1 ... MethodN
x      y1      y2      y3
0      9       4       3
...
...
```

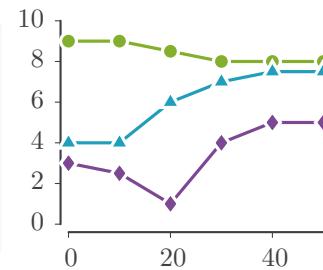
```
/examples/line.tex
```

```
\begin{axis}[ pretty line ]
\pgfplotsinvokeforeach{1, ..., 3}{
    \addplot table[x=x, y=y#1] {data/line.tsv};
}
\end{axis}
```



The default color scheme `pr-colors`, you can find others in Section 4. For example,

```
\begin{axis}[pretty line, cycle list name=pr-colors]
\pgfplotsinvokeforeach{1, ..., 3}{
    \addplot+ table[x=x, y=y#1] {data/line.tsv};
}
\end{axis}
```



3.2 Box

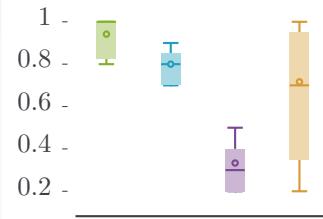
For box plots, you can use pretty `boxplot`.

```
/data/box.tsv
```

%method1	...	methodM	
y1	y2	y3	y4
0.10	0.15	0.12	0.18
...			
0.95	0.98	0.93	0.97

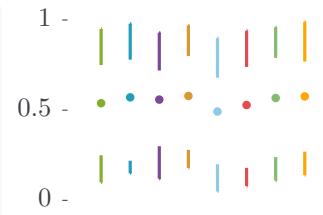
```
/examples/box.tex
```

```
\begin{axis}[ pretty boxplot ]
\foreach \col in {y1, y2, y3, y4} {
    \addplot table[y=\col] {data/box.tsv};
}
\end{axis}
```



Here is a different look [Tufte(2001)].

```
\begin{axis}[ pretty boxplot simple ]
\foreach \col in {y1,y2,y3,y4,y5,y6,y7,y8} {
    \addplot table[y=\col] {data/boxes.tsv};
}
\end{axis}
```



3.3 Bar

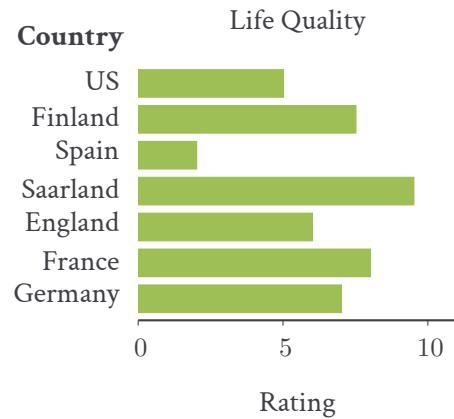
You can create bar plots using `pretty xbar` for horizontal and `pretty ybar` for vertical bars.

```
/data/countries.tsv
```

%id	country	lifeq	foodq
id	country	life	food
1	Germany	8	2
...			
5	US	1	7

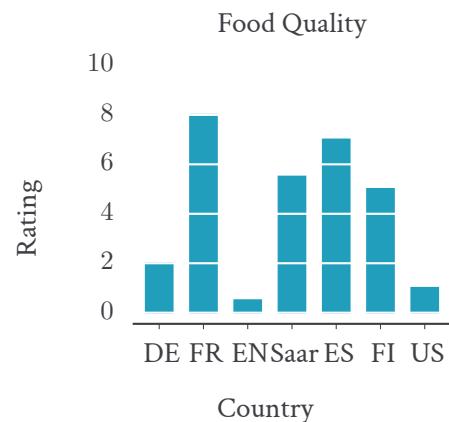
```
/examples/bar.tex
```

```
\begin{axis}[
    pretty xbar,
    yticklabels = {Germany,France,England,
        Saarland,Spain,Finland,US},
    xmax = 10,
    xlabel = {Rating},
    ylabel = {\textbf{Country}},
    title = {Life Quality},
]
\addplot table[x=life,y=id] {data/countries.tsv};
\end{axis}
```



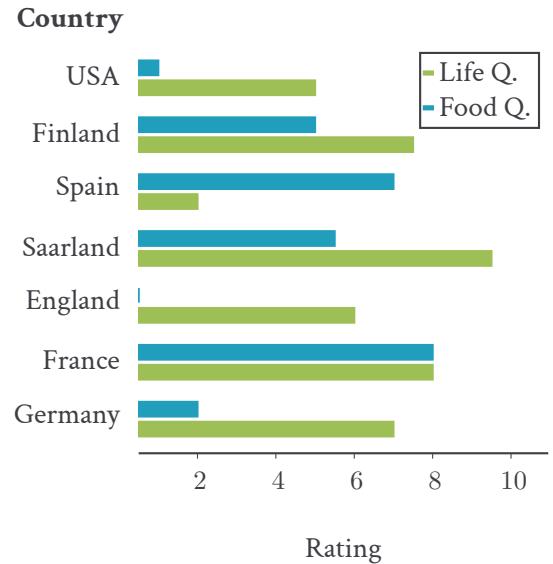
To obtain a vertical bar plot, swap `xbar` with `ybar`. You can also add a white grid [Tufte(2001)],

```
\begin{axis}[
    pretty ybar,
    pretty grid ybar,
    xticklabels = {DE,FR,EN,Saar,ES,FI,US},
    xlabel = {Rating},
    ylabel = {Country},
    title = {Food Quality},
]
\addplot[pr-colorib, fill=pr-colorib]
table[x=life,y=id] {data/countries.tsv};
\end{axis}
```

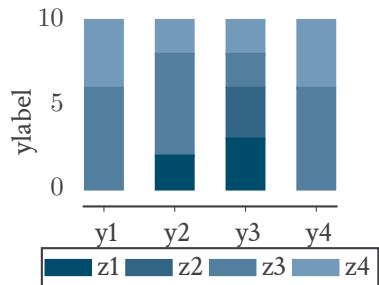


Here are some variations.

```
\begin{axis}[
    pretty xbar,
    yticklabels = {Germany,France,England,
        Saarland,Spain,Finland,US},
    xmax = 10,
    bar width = 0.5em,
    xlabel = {Rating},
    ylabel = {\textbf{Country}},
    legend entries = {Life Q., Food Q.},
]
\addplot table[x=life,y=id] {data/countries.tsv};
\addplot table[x=food,y=id] {data/countries.tsv};
\end{axis}
```



```
\begin{axis}[
    pretty ybar stacked,
    cycle list name = pr-box2,
    bar width      = .5cm,
    ylabel         = {ylabel},
    symbolic x coords= {y1,y2,y3,y4}
]
\addplot plot coordinates
    {(y1,0) (y2,2) (y3,3) (y4,0)};
\addplot plot coordinates
    {(y1,0) (y2,0) (y3,3) (y4,0)};
\addplot plot coordinates
    {(y1,6) (y2,6) (y3,2) (y4,6)};
\addplot plot coordinates
    {(y1,4) (y2,2) (y3,2) (y4,4)};
\legend{z1, z2, z3, z4}
\end{axis}
```



3.4 Scatter

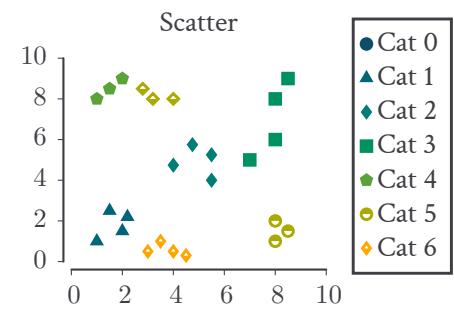
Use this plot type if you want color and style of markers to depend on a given class label as follows.

```
/data/classes.tsv
```

x	y	scatter class
x	y	label
1	1	0
2	3	1

```
/examples/scatter.tex
```

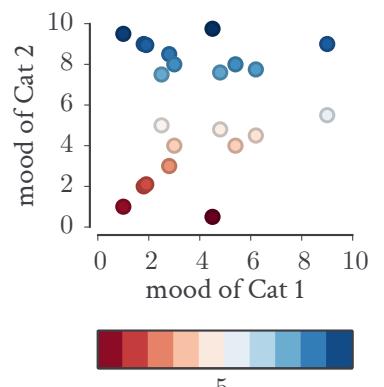
```
\begin{axis}[
    pretty scatter,
    title = {Scatter},
    legend entries = {Class 0,Class 1,Class 2,
                      Class 3,Class 4,Class 5,Class 6},
    legend style={at={(1.1,1.2)}, anchor=north west},
]
\addplot+[pr-scatter] table[x=x, y=y, meta=label]
{data/classes.tsv};
\end{axis}
```



Have a look at Section 4 to change the scatter class colors.
If you have continuous color labels, use a colormap,

```
/examples/colormap.tex
```

```
\begin{axis}[
    pretty scatter colormap horizontal,
    colormap/RdBu-11, % default colormap
    colorbar style = { xlabel =
        {harmony between Cat 1 \& 2}, },
    xlabel           = {Cat 1's mood},
    ylabel           = {Cat 2's mood},
    mark size       = 2.5 pt
]
\addplot table[x=x, y expr=10-0.5*\thisrow{y}]
{data/marks.txt};
\addplot table[x=x, y=y] {data/marks.txt};
\end{axis}
```



harmony between Cat 1 & 2

3.5 Error Line

If you want to plot confidence regions around your line, include them in your dataset as follows,

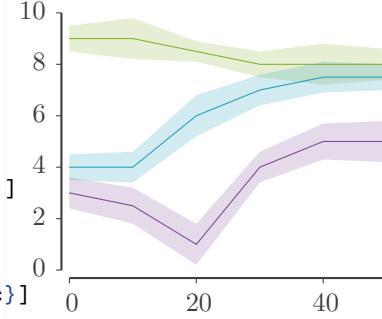
```
/data/line_conf.tsv

%Metric    Method1    Method1_conf ...    MethodN_conf
x      y1      y1_c      y1      y1_c      y3      y3_c
0       9       0.5      4       0.5      3       0.6
...
...
```

Then, add both the lines and the confidence bands as follows,

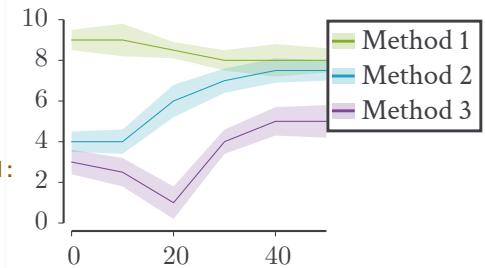
```
/examples/line_conf.tex

\begin{axis}[
    pretty line,
    % cycle list to skip the confidence regions:
    cycle list name = pr-colors-conf,
    \foreach \r in {1,2,3} {
        \addplot table[x=x, y=y\r] {data/line_conf.tsv};
        \addplot[forget plot, draw=none, name path=low]
            table [x=x, y expr=\thisrow{y\r} - \thisrow{y\r_c}]
            {data/line_conf.tsv};
        \addplot[forget plot, draw=none, name path=up]
            table [x=x, y expr= \thisrow{y\r} + \thisrow{y\r_c}]
            {data/line_conf.tsv};
        \addplot fill between[of=low and up];
    }
]
\end{axis}
```



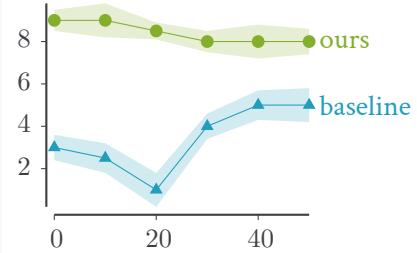
Note that the upper and lower confidence lines are each added using `addplot`, where `forget plot` makes sure that these lines are disregarded in the legend and cycle list. The last `fill-between` line is added without `forget plot` as we define its color in the cycle list. If you use a legend, skip these plots as follows.

```
\begin{axis}[
    pretty line,
    pretty fill legend,
    cycle list name = pr-colors-conf,
    % double ,, skips the confidence regions in the legend:
    legend entries = {Method 1,,Method 2,,Method 3},
]
%
```



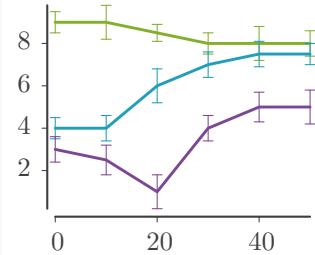
For lines with error regions, the cycle lists contain the colors of the confidence bands. You can look at the -conf cycle lists in `prettyplots.sty` for different styles. Here is an example with `pr-marks-conf`.

```
\begin{axis}[
    pretty line,
    cycle list name = pr-marks-conf
]
% our plot
\addplot table[x=x, y=y1] {data/line_conf.tsv}
node[pos=1, right] (n) {ours};
\addplot[forget plot, draw=none, name path=low]
table [x=x, y expr=\thisrow{y1} - \thisrow{y1_c}]
{data/line_conf.tsv};
\addplot[forget plot, draw=none, name path=up]
table [x=x, y expr= \thisrow{y1} + \thisrow{y1_c}]
{data/line_conf.tsv};
\addplot fill between[of=low and up];
% baseline plot
\addplot table[x=x, y=y1] {data/line_conf.tsv}
node[pos=1, right] (n) {baseline};
\addplot[forget plot, draw=none, name path=low]
table [x=x, y expr=\thisrow{y3} - \thisrow{y3_c}]
{data/line_conf.tsv};
\addplot[forget plot, draw=none, name path=up]
table [x=x, y expr= \thisrow{y3} + \thisrow{y3_c}]
{data/line_conf.tsv};
\addplot fill between[of=low and up];
\end{axis}
```



To show error bars instead, use

```
\begin{axis}[pretty line]
\foreach \r in {1, ..., 3}{%
    \addplot+[error bars/.cd, y explicit, y dir=both]
    table[ x=x, y=y\r,
    y error plus expr=\thisrow{y\r_c},
    y error minus expr=\thisrow{y\r_c},
    ] {data/line_conf.tsv};
}
\end{axis}
```

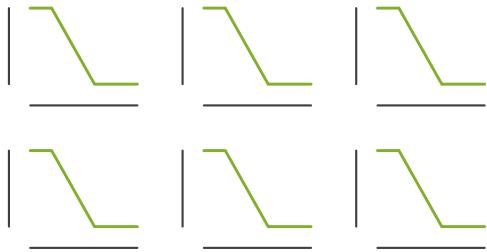


3.6 Group Plots

To place multiple axes side by side with easy alignment, replace `axis` by `groupplot` and create subfigures using `\nextgroupplot[<axis keys>]`. That is, pass your customizations to `\nextgroupplot` and then use `addplot` as usual.

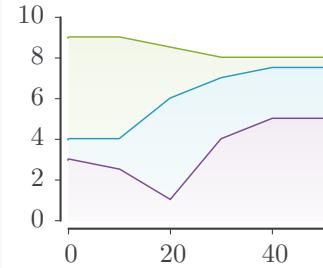
```
/examples/groupplots.tex
```

```
%preamble
\usetikzlibrary{pgfplots.groupplots}
%
\begin{tikzpicture}
\begin{groupplot}[
    group style={
        group size=3 by 2,
        horizontal sep=25pt,
        vertical sep=25pt},
    xtick align=center, ytick align=center,
    width=3cm
]
\nextgroupplot[pretty line] \addplot table {data/lines.tsv};
\end{groupplot}
\end{tikzpicture}
```



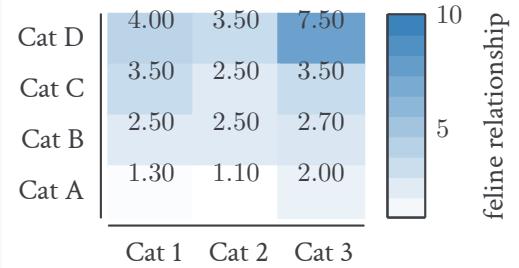
3.7 Examples

```
\begin{axis}[
    pretty line,
    cycle list name= pr-shade,
]
\foreach \r in {1,2,3} {
    \addplot table[x=x, y=y\r] {data/line.tsv};
    \addplot table[x=x, y=y\r] {data/line.tsv}
    \closedcycle;
}
\end{axis}
```

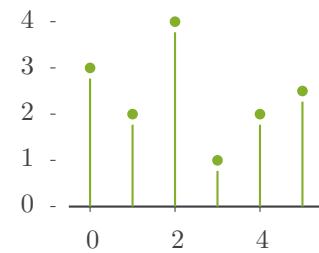


/examples/colorbar.tex

```
\begin{axis}[
    pretty heatmatrix,
    xticklabels = {Cat 1, Cat 2, Cat 3},
    yticklabels = {Cat D, Cat C, Cat B, Cat A},
    colorbar style = { ylabel = {feline relationship}, },
    point meta max = 10, % z range
    pretty labelshift
]
\addplot[pr-matrix, mesh/cols=3, % y
] table [meta=z] {data/colorbar.txt};
\end{axis}
```

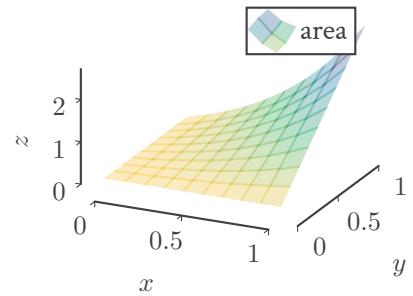


```
\begin{axis}[ pretty comb ]
\addplot+[ycomb, mark=white*] coordinates
{(0,3) (1,2) (2,4) (3,1) (4,2) (5,2.5)};
\end{axis}
```

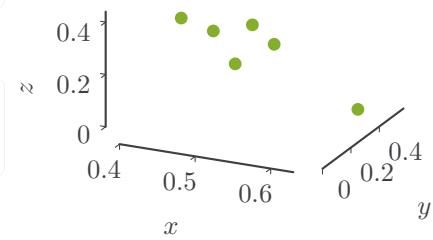


```
/examples/threedim.tex
```

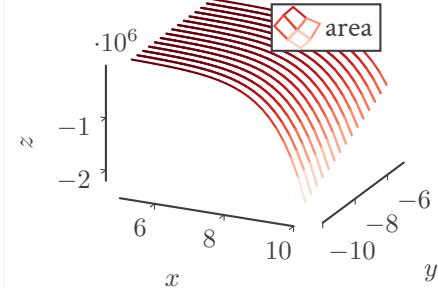
```
\begin{axis}[
    pretty 3daxis,
    mesh legend,
    samples = 10,
    domain = 0:1,
    xlabel = {$x$},
    ylabel = {$y$},
    zlabel = {$z$},
]
\addplot3 [pr-surf] {x*y*exp(x)};
\legend{area};
\end{axis}
```



```
\addplot3+ [pr-scatter] table {data/ternary.tsv};
```

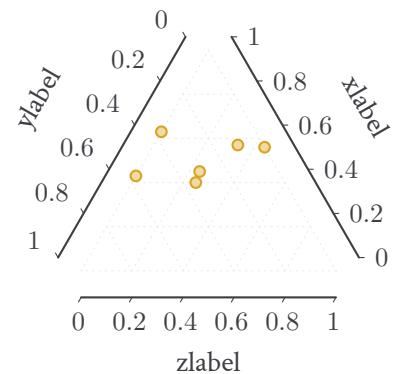


```
\begin{axis}[
    pretty 3daxis,
    scaled ticks = true,
    domain = 5:10,
    domain y = -10:-5,
    samples y = 15,
    xlabel = {$x$},
    ylabel = {$y$},
    zlabel = {$z$},
    mesh legend
]
\addplot3 [pr-contour] {x*y*exp(x)};
\legend{area};
\end{axis}
```



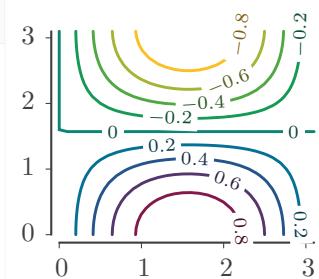
```
/examples/ternary.tex

% preamble
\pgfplotsset{compat=1.18}
\usepgfplotslibrary{ternary}
%
\begin{ternaryaxis}[
    pretty ternary axis,
    xlabel= { xlabel },
    ylabel= { ylabel },
    zlabel= { zlabel },
]
\addplot3+[pr-scatter] table {data/ternary.tsv} ;
\end{ternaryaxis}
```



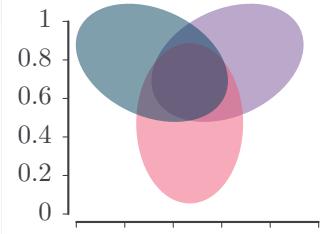
```
/examples/contour.tex

\begin{axis}[ pretty contour ]
\addplot[
    contour prepared,
    contour prepared format=matlab,
] table {contours.txt};
\end{axis}
```

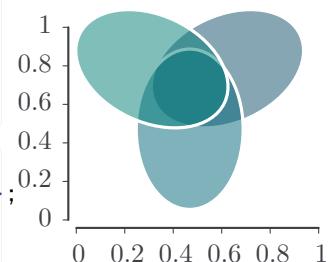


```
/examples/venn.tex
```

```
\begin{tikzpicture}
\begin{axis}[pretty line] \end{axis}
\coordinate (c1) at (2,2);
\coordinate (c2) at (1.5,1.2);
\coordinate (c3) at (1,2);
\node[prellipse, fill=prcl-palette3c, rotate=-65] at (c1) {};
\node[prellipse, fill=prcl-palette3e, rotate=0] at (c2) {};
\node[prellipse, fill=prcl-palette3a, rotate=65] at (c3) {};
\end{tikzpicture}
```

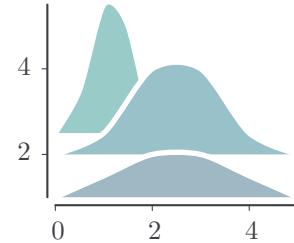


```
\node[prellipse white, fill=pr-color2a, rotate=-65] at (c1) {};
\node[prellipse white, fill=pr-color2b, rotate=0] at (c2) {};
\node[prellipse white, fill=pr-color2c, rotate=65] at (c3) {};
```

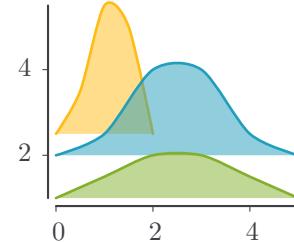


```
/examples/ridgeline.tex
```

```
\begin{axis}[
    pretty ridge line,
    cycle list name = pr-ridgeline
]
\addplot coordinates {((0,1) (1,1.5) (2,2) (3,2) (4,1.5) (5,1)} ;
\addplot coordinates {((0,1) (1,1) (2,2) (3,2) (4,1) (5,1)} ;
\addplot coordinates {((0,0.5) (0.5,1) (1,1.5) (1.5,1) (2,0)} ;
\end{axis}
```



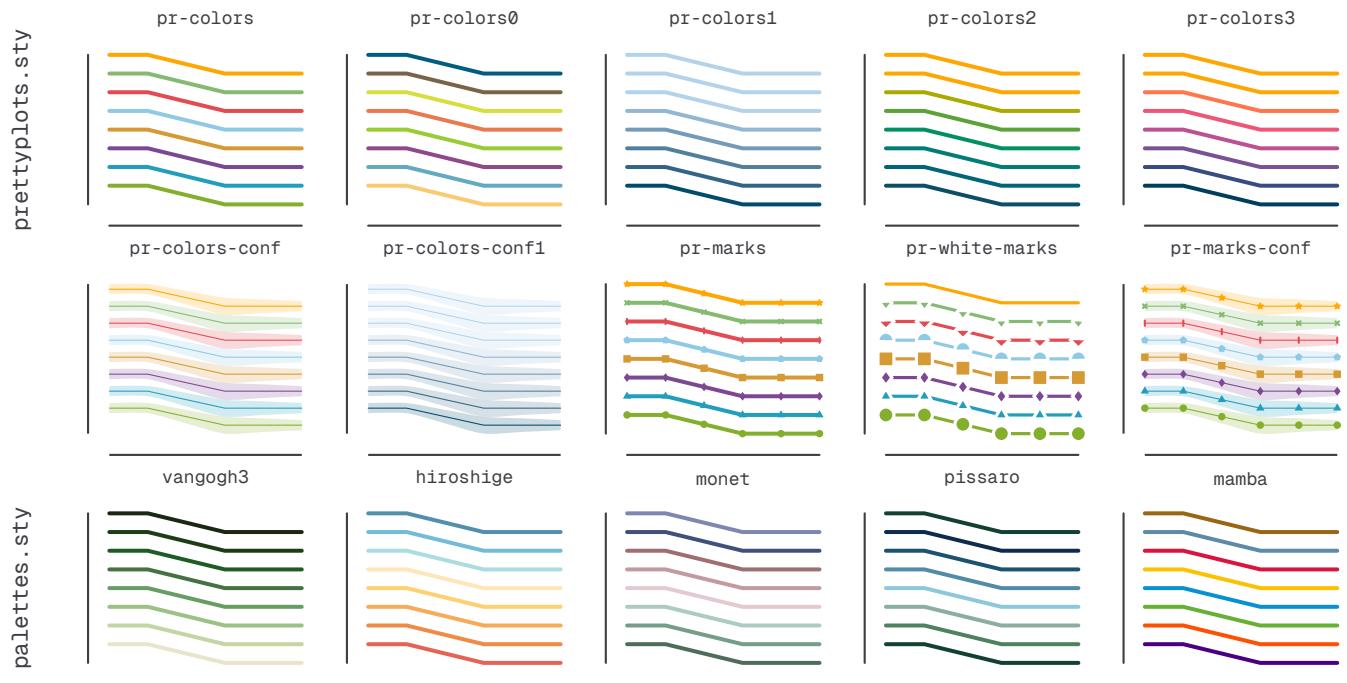
```
\begin{axis}[
    pretty ridge line,
    cycle list name=pr-ridgeline-transparent
]
% ...
\end{axis}
```



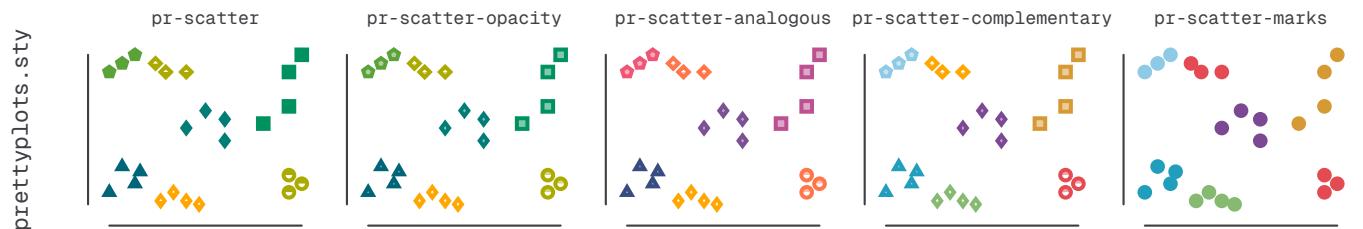
4 Colors and Style

4.1 Color Palettes

Here are some color schemes as a starting point.



Example 1: Cycle lists for line plots, use with `\begin{axis}[cycle list name=<name>]`.



Example 2: Scatter classes, use with `\addplot+[<name>]` in an `\begin{axis}[pretty scatter]` environment.

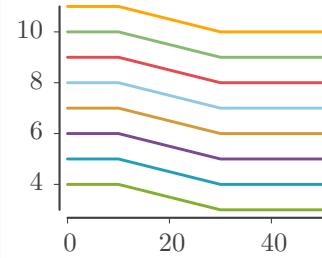
4.2 Custom Colors

To set your own colors manually, pass your color of choice to `\addplot[black]`. You can also pre-define a list of colors that each call of `\addplot` will use through in order. This is known as a cycle list, which you can pass to the axis keys as `cycle list=<name>`.

```
/examples/line_colors.tex
```

```
%% Create a cycle list defining the color of each line
\pgfplotscreateplotcyclelist{pr-colors}{
    {pr-color1a},
    {pr-color1b},
    {pr-color1c},
    {pr-color1d},
    {pr-color1e},
    {pr-color1f},
    {pr-color1g},
    {pr-color1h}
}

% Usage in an axis environment
\begin{axis}[pretty line, cycle list name=pr-colors]
\pgfplotsinvokeforeach{1, ..., 8}{%
    \addplot table[x=x, y=y#1] {data/lines.tsv}; }
\end{axis}
```

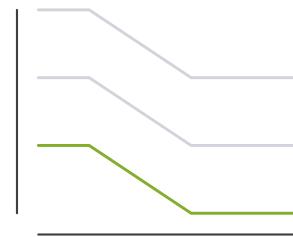


You can change the above definition of `pr-colors` directly in `prettyplots.sty` to use consistent colors in all of your work. For experiments, you can also define method-specific colors as follows

```
%% Pick your colors
\colorlet{color-ourmethod}{pr-color1a}
\definecolor{color-baselineA}{HTML}{f3f4f6}
\definecolor{color-baselineB}{HTML}{f3f4f6}

% Create a cycle list
\pgfplotscreateplotcyclelist{colors-methods}{
    {color-ourmethod},
    {color-baselineA},
    {color-baselineB}
}

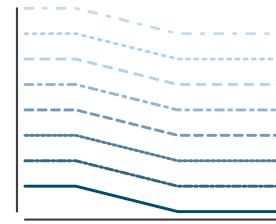
% Usage in an axis environment
\begin{axis}[pretty line, cycle list name=colors-methods]
\pgfplotsinvokeforeach{1, ..., 3}{%
    \addplot table[x=x, y=y#1] {data/lines.tsv}; }
\end{axis}
```



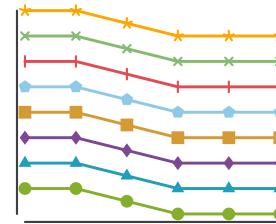
4.3 Line Style

Similar to color, you can set up the line style for your plots as follows.

```
%% Create a cycle list that changes linestyle
\pgfplotscreateplotcyclelist{pr-linestyle}{
    {pr-color0a, solid},
    {pr-color0b, densely dashdotted},
    {pr-color0c, densely dotted},
    {pr-color0d, densely dashed},
    {pr-color0e, dashdotted},
    {pr-color0f, dashed},
    {pr-color0g, dotted},
    {pr-color0g!80, loosely dashdotted}
}
\lines{data/lines.tsv}[cycle list name=pr-linestyle]
```



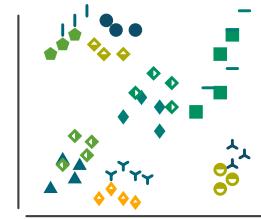
```
%% Create a cycle list that changes marker style
\pgfplotscreateplotcyclelist{pr-marks} {
    {pr-color1a,mark=*, mark options={pr-color1a,fill=pr-color1a}},
    {pr-color1b,mark=triangle*, mark options={pr-color1b,fill=pr-color1b}},
    {pr-color1c,mark=square*, mark options={pr-color1c,fill=pr-color1b}},
    % ...
}
\lines{data/lines.tsv}[cycle list name=pr-marks]
```



4.4 Scatter Classes

For scatter plots, `\pr-scatter` defines how each of the classes should appear and is passed to `\addplot`,

```
\begin{axis}[pretty scatter,pretty nolabels]
  \addplot [pr-scatter] table[x=x1,y=x2,meta=y,
    col sep=comma] {data/classes.csv};
\end{axis}
```

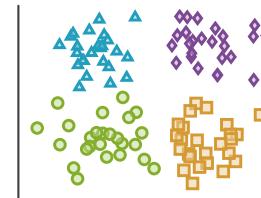


To replace this with your own,

```
/examples/scatter_colors.tex
```

```
%% Define custom scatter colors
\pgfplotsset{
  my-scatter/.style={
    only marks, mark options={fill opacity=0.50},
    scatter src=explicit symbolic,
    scatter/classes={
      0={mark=*, pr-color0a, fill=pr-color0a!50},
      1={mark=triangle*, pr-color1b, fill=pr-color1b!50},
      2={mark=diamond*, pr-color1c, fill=pr-color1c!50},
      3={mark=square*, pr-color1d, fill=pr-color1d!50}
      %no comma in the last line, otherwise latex complains
    }
  }
}

\begin{axis}[pretty scatter,pretty nolabels]
  \addplot [my-scatter]
  table[x=x1,y=x2,meta=y, col sep=comma] {data/blobs.csv};
\end{axis}
```



Here us an example for cases where your label is continuous instead of discrete.

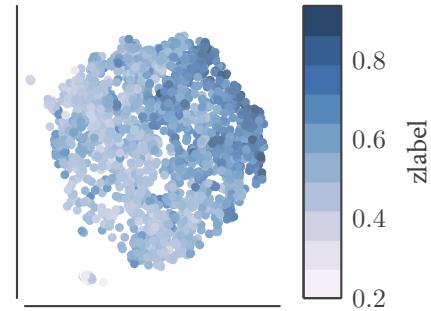
```
/data/blob.tsv
```

x	y	color value
x	y	col
-82.87	16.81	0.52
-75.32	44.21	0.41

Then, you can pick a few transition colors and set up a color map as follows.

```
/examples/scatter_colormap_custom.tex
```

```
\begin{axis}[
    pretty scatter colormap,
    colormap={mymap}{rgb255=(253,247,251)
        rgb255=(234,230,241) rgb255=(207,208,228)
        rgb255=(171,188,217) rgb255=(133,168,204)
        rgb255=(92,143,189) rgb255=(64,111,172)
        rgb255=(52,90,138) rgb255=(34,59,88)},
]
\addplot+[pr-scatter-meta]
    table[meta=col] {data/blob.tsv};
\end{axis}
```



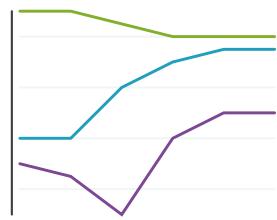
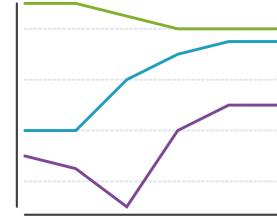
4.5 Grids

The plot types come with some grids that you can use, usually called `\prgridcolor` `\gridname`.

```
/examples/grids.tex

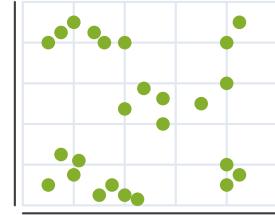
% preamble
\renewcommand{\prgridcolor}{tcss-slate1}
%
\begin{axis}[pretty line, pretty grid line] % ...
\end{axis}

%% defining custom grids
\pgfplotsset{
    pretty grid line/.style={
        ymajorgrids,
        major grid style = {tcss-slate1, solid, thick},
    },
}
\begin{axis}[ pretty line, pretty grid line] %...
\end{axis}
```



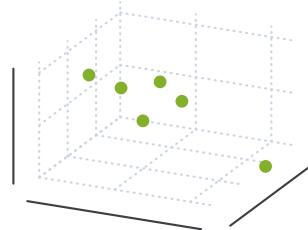
Similarly for the other plot types.

```
\pgfplotsset{
    pretty grid scatter/.style={
        grid = major,
        major grid style = {tcss-slate1, thick},
        width=\textwidth
    },
}
\begin{axis}[ pretty scatter, pretty grid scatter] %...
\end{axis}
```



The default grid color for all grids is defined via `\prgridcolor`.

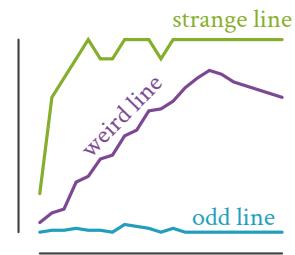
```
\renewcommand{\prgridcolor}{tcss-slate3}
\begin{axis}[pretty 3daxis, pretty grid 3d]
    \addplot3+ [pr-scatter] table {data/ternary.tsv} ;
\end{axis}
```



4.6 Labels

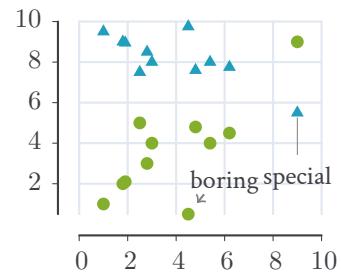
/examples/line_labels.tex

```
\begin{axis}[pretty line,pretty nolabels,very thick,smooth]
\addplot table[x = x, y = y2] {data/line_odd.tsv}
node [sloped,above,font=\small,inner sep=2pt,pos=0.8]
    {strange line};
\addplot table[x = x, y = y3] {data/line_odd.tsv}
node [sloped,above,font=\small,inner sep=2pt,pos=0.8]
    {odd line};
\addplot table[x = x, y = y1] {data/line_odd.tsv}
node [sloped,above,font=\small,inner sep=4pt,
      pos=0.45,rotate=20] {weird line};
\end{axis}
```



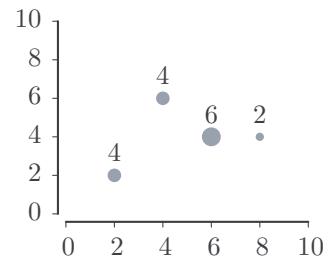
/examples/marks_labels.tex

```
\begin{axis}[
    pretty line marks,
    pretty grid scatter
]
\addplot table[x=x, y=y] {data/marks.txt};
\addplot table[x=x, y expr=10-0.5*\thisrow{y}]
    {data/marks.txt};
\node at (axis cs:6, 2.1) {\small boring};
\node at (axis cs:5.5, 2) (node1) {};
\node at (axis cs:4.5, 0.5) (node2) {};
\draw [->, gray, thick] (node1) --(node2);
\node[pin=-90:\small special] at (axis cs:9,5.8) {};
\end{axis}
```



```
/examples/scatter_labels.tex
```

```
\begin{axis}[
    pretty scatter,
    scatter/use mapped color =
        {draw=gray, fill=gray} ]
% \addplot[pr-markweights] table {data/marks_weighted.tsv};
\addplot[only marks,
    nodes near coords*={
        \prnodefontsize {\pgfmathprintnumber {\weight}}
    },
    every node near coord/.append style = {yshift = 1 pt},
    visualization depends on={\thisrow{\weight} \as \weight},
    scatter/@pre marker code/.append style=
        {/tikz/mark size=0.5*\weight}
] table {data/marks_weighted.tsv};
\end{axis}
```



```
/examples/bar_labels.tex
```

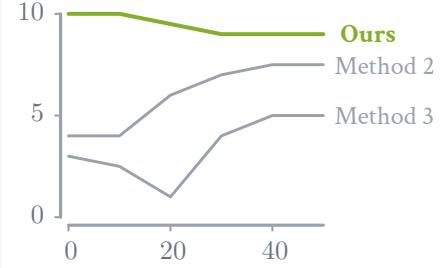
```
\begin{axis}[
    pretty ybar,
    %% label each bar with y value
    visualization depends on=y \as \y,
    nodes near coords = {
        \pgfmathprintnumber[precision=2]{\y}},
    %% opt. label layout
    %every node near coord/.append style={
    %    font=\prnodefontsize, color=pr-gray4},
    bar width = 1em,
    ylabel = {Rating},
    symbolic x coords = {Oslo,Berlin,Tokyo,Rio},
]
\addplot table[x=x, y=y3] {data/cities.tsv};
\end{axis}
```



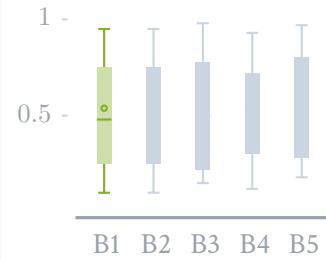
4.7 Highlighting

The following examples are inspired from [Knaflc(2015)].

```
\begin{axis}[
    pretty line,
    ytick = {0, 5, 10},
    x tick label style = {pr-color-gray5},
    y tick label style = {pr-color-gray5},
    xtick style = {pr-color-gray4, thick},
    ytick style = {pr-color-gray4, thick},
    x axis line style = {pr-color-gray4, very thick},
    y axis line style = {pr-color-gray4, very thick}
]
\addplot+[line width=1.5pt]
table[x=x, y expr=\thisrow{y1}+1]
    {data/line.tsv}
    node[pos=1, right] (n) {\small \textbf{Ours}};
\pgfplotsinvokeforeach{2, ..., 3}{
\addplot+[pr-color-gray4] table[x=x, y=y#1]
    {data/line.tsv}
    node[pos=1, right] (n) {\small Method #1};
}
\end{axis}
```



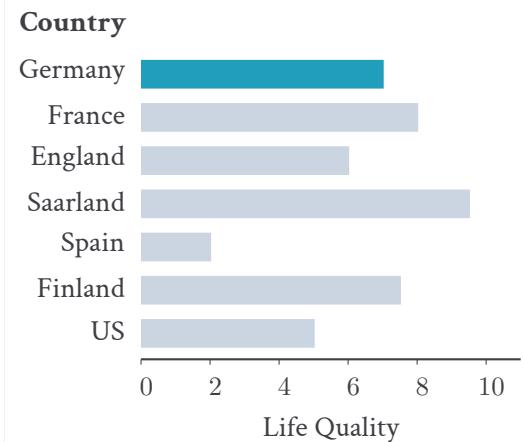
```
\begin{axis}[
    pretty boxplot,
    xticklabels = {B1,B2,B3,B4},
    xtick = {1,2,3,4},
    y tick label style = {pr-color-gray4,
        font=\prtickfontsize},
    x tick label style = {pr-color-gray4,
        font=\prtickfontsize},
    ytick style = {pr-color-gray3, thick},
    x axis line style = {pr-color-gray4, thick},
    ymin=0
]
\addplot table[y=y1] {data/boxes.tsv};
\foreach \col in {y1,y2,y3,y4,y5} {
    \addplot+[pr-color-gray3] table[y=\col]
        {data/boxes.tsv};
}
\end{axis}
```



```

\begin{tikzpicture}
\pgfplotstablegetrowsof{data/countries.tsv}
\edef\numberofrows{\pgfplotsretval}
\begin{axis}[
    pretty xbar,
    bar shift = 0pt,
    y dir = reverse,
    ytick = {0, ..., \numberofrows},
    yticklabels from table =
        {data/countries.tsv}{[index]0},
    xlabel = {Life Quality},
    ylabel = {\textbf{Country}}
]
\addplot+[pr-color-bg, fill=pr-color-bg,
show except={country}{Germany}] table [
    y expr=\coordindex, x index=1
    {data/countries.tsv};
\addplot+[pr-color1b, fill=pr-color1b,
show only={country}{Germany}] table [
    y expr=\coordindex,
    x index=1
    {data/countries.tsv};
\end{axis}
\end{tikzpicture}

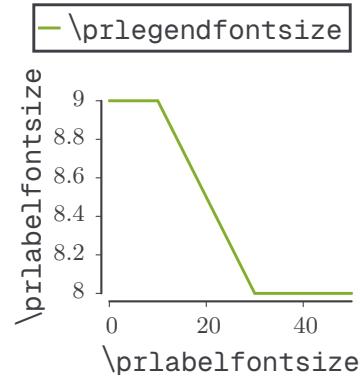
```



4.8 Axis Style

The following macros can be redefined to adjust label sizes,

```
\renewcommand{\prlabelsize}{\large}
\renewcommand{\prticksize}{\small}
\renewcommand{\prlegendsize}{\large}
```



Or apply changes to the axis environment directly,

```
\begin{axis}[
    pretty line,
    xlabel style = {pr-color-gray4, font=\prlabelsize},
    ylabel style = {pr-color-gray4, font=\prlabelsize},
    x tick label style = {pr-color-gray5,
        font=\prticksize},
    y tick label style = {pr-color-gray5,
        font=\prticksize},
    xtick style = {pr-color-gray4, thick},
    ytick style = {pr-color-gray4, thick},
    x axis line style = {pr-color-gray4, very thick},
    y axis line style = {pr-color-gray4, very thick}
]
\addplot table[x=x, y expr=\thisrow{y1}+1]
    {data/line.tsv}
    node[pos=1, right] (n) {\small Ours};
\pgfplotsinvokeforeach{2, ..., 3}{
    \addplot+[pr-color-gray4] table[x=x, y=y#1]
    {data/line.tsv}
    node[pos=1, right] (n) {\small Method #1};
}
\end{axis}
```



5 Advanced Usage

Commands for Line Plots

```
\addlines    <data.tsv> [<addplot options>] [<table options>]  
            adds all line plots from a given file, from columns (x, y1), ..., (x, yn).  
  
\addconflines <data.tsv> [<addplot options>] [<table options>]  
            adds line plots and confidence regions from a given file, from columns (x,  
            y1, y1_c), ..., (x, yn, yn_c).  
  
\grouplines   <data.tsv> [<addplot options>] [<table options>]  
            adds line plots to a group plot.  
  
\groupconflines <data.tsv> [<addplot options>] [<table options>]  
            adds line plots to a group plot.
```

Arguments

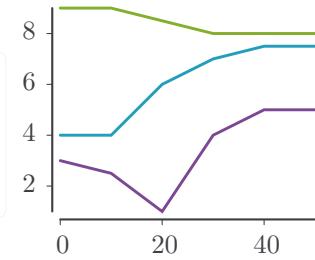
- <data.tsv> — (*Required*) data file.
- [<addplot options>] — (*Optional*) style for adding plots, such as color, line-, and markers style. (*Examples*: black, smooth, line width=1.2pt, mark=*, mark options={fill opacity=0.5,mark size=1pt})
- [<table options>] — (*Optional*) keys to table. (*Examples*: col sep=comma)

Examples

```
/data/line.tsv
```

```
%Metric    Method1 ... MethodN  
x      y1      y2      y3  
0       9       4       3  
%...
```

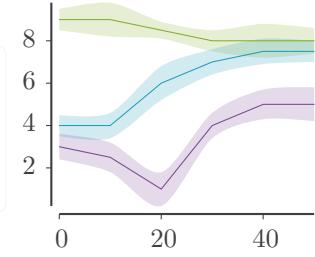
```
\begin{axis}[ pretty line ]  
\addlines{data/line.tsv}  
\end{axis}
```



```
/data/line_conf.tsv
```

```
%Metric    Method1    Method1_conf ...    MethodN_conf  
x      y1      y1_c      y1      y1_c      y3      y3_c  
0       9       0.5       4       0.5       3       0.6  
%....
```

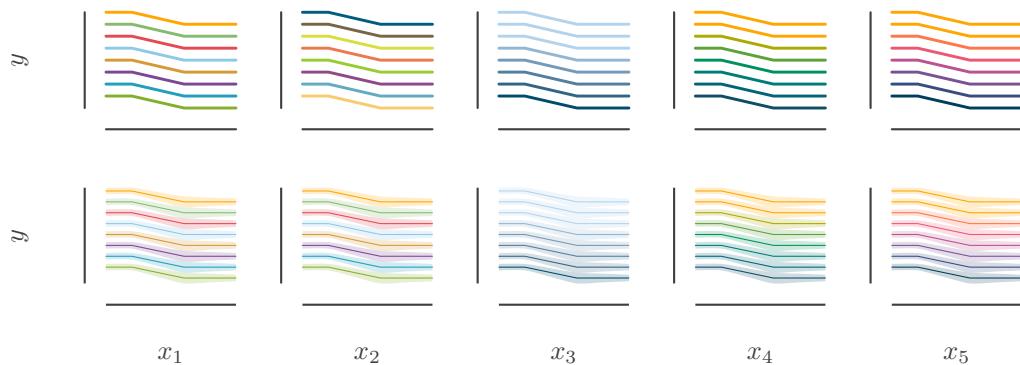
```
\begin{axis}[pretty confline]  
\addconflines{data/line_conf.tsv}  
\end{axis}
```



```

\begin{tikzpicture}
\def\file{data/lines_conf.tsv}
\begin{groupplot}[
    group style={
        group size=5 by 2,
        horizontal sep=25pt,
        vertical sep=30pt},
    xtick align=center, ytick align=center,
    width=0.25\textwidth
]
\grouplines{data/lines.tsv}[ylabel=$y$]
\grouplines{data/lines.tsv}[cycle list name=pr-colors0]
\grouplines{data/lines.tsv}[cycle list name=pr-colors1]
\grouplines{data/lines.tsv}[cycle list name=pr-colors2]
\grouplines{data/lines.tsv}[cycle list name=pr-colors3]
\\
\groupconfines{data/lines_conf.tsv}[ylabel=$y$, xlabel=$x_1$]
\groupconfines{data/lines_conf.tsv}[cycle list name=pr-colors-conf0]
\groupconfines{data/lines_conf.tsv}[cycle list name=pr-colors-conf1]
\groupconfines{data/lines_conf.tsv}[cycle list name=pr-colors-conf2]
\groupconfines{data/lines_conf.tsv}[cycle list name=pr-colors-conf3]
]
\end{groupplot}
\end{tikzpicture}

```



Index

\addconflines (*command*), 29
\addlines (*command*), 29
\groupconflines (*command*), 29
\grouplines (*command*), 29

Colors and Style

defining axis colors, 26, 28
defining continuous colors (colormap), 22
defining line colors (cycle lists), 18
defining line style, 20
defining marker colors (scatter classes), 18
defining marker style, 20

Grids, 23

Labels, 24

label names from columns, 26
near bars, 25
near lines, 24, 26
near marks, 25
showing points, 24

Plot Types

bar, 8
 simplified, 26
 stacked, 9
xbar, 8
ybar, 8
box, 7
 basic, 7
 simplified, 26
comb, 14

References

[Knafllic(2015)] Knafllic, C. *Storytelling with Data: A Data Visualization Guide for Business Professionals*. Wiley, 2015.

[Tufte(2001)] Tufte, E. R. *The Visual Display of Quantitative Information*. Graphics Press, 2001.