

---

# **Plot Toolkit Documentation**

***Release 0.1***

**Karl Debiec**

April 14, 2014



## CONTENTS

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Functions</b>	<b>3</b>
2.1	Formatting . . . . .	3
2.1.1	Axes . . . . .	3
2.1.2	Text . . . . .	4
2.1.3	Legend . . . . .	6
2.2	Auxiliary . . . . .	6
2.2.1	General . . . . .	6
2.2.2	Matplotlib . . . . .	7
<b>3</b>	<b>Decorators</b>	<b>9</b>
3.1	Figure_Output . . . . .	9
	<b>Index</b>	<b>11</b>



## **INTRODUCTION**

Plot Toolkit is a set of python functions, classes, and decorators intended to simplify usage of the matplotlib package. Matplotlib provides an excellent framework for rapidly generating plots using python. However, applying strict formatting specifications to these plots requires writing a large amount of code that is difficult to read and difficult to remember. Plot Toolkit's purpose is to allow users to generate plots with strict formatting specifications using matplotlib, without writing large amounts of complex code.



## FUNCTIONS

## 2.1 Formatting

### 2.1.1 Axes

`plot_toolkit.format.axes.set_xaxis(...)`

Formats an X axis

**Arguments:**

*subplot* <matplotlib.axes.AxesSubplot> on which to act

*ticks* X Axis ticks

*tick\_kwargs* Keyword arguments to be passed to `set_xticks`

*ticklabels* X Axis tick labels

*tick\_fp* X Axis tick font in form of '##L'

*ticklabel\_kwargs* Keyword arguments to be passed to `set_xticklabels`

*label* X Axis label

*label\_fp* X Axis label font in form of '##L'

*label\_kwargs* Keyword arguments to be passed to `set_xlabel`

*outline* Add dark outline to axis

`plot_toolkit.format.axes.set_yaxis(...)`

Formats a Y axis

**Arguments:**

*subplot* <matplotlib.axes.AxesSubplot> on which to act

*ticks* Y Axis ticks

*tick\_kwargs* Keyword arguments to be passed to `set_yticks`

*ticklabels* Y Axis tick labels

*tick\_fp* Y Axis tick font in form of '##L'

*ticklabel\_kwargs* Keyword arguments to be passed to `set_yticklabels`

*label* Y Axis label

*label\_fp* Y Axis label font in form of '##L'

*label\_kwargs* Keyword arguments to be passed to `set_ylabel`

*outline* Add dark outline to axis

`plot_toolkit.format.axes.set_colorbar(...)`  
Formats a colorbar

**Arguments:**

*cbar* <matplotlib.colorbar.ColorBar> to be acted on

*ticks* Color bar ticks

*ticklabels* Color bar tick labels

*tick\_fp* Color bar tick label font in form of '##L'

*label* Color bar label

*label\_fp* Color bar label font in form of '##L'

## 2.1.2 Text

`plot_toolkit.format.text.set_text(...)`  
Prints text on a figure or subplot

**Arguments:**

*figure\_or\_subplot* <matplotlib.figure.Figure> or <matplotlib.axes.AxesSubplot> on which to act

*text* Text ('s' also supported)

*fp* Text font in form of '##L' (default: '11b')

*ha* Text horizontal alignment (default: 'center')

*va* Text vertical alignment (default: 'center')

**Returns:**

*text* <matplotlib.text.Text>

`plot_toolkit.format.text.set_inset(...)`  
Prints text as an inset to a subplot

**Arguments:**

*subplot* matplotlib.axes.AxesSubplot on which to act

*text* Inset text ('s' and 'inset' also supported)

*fp* Inset text font in form of '##L'

*xpos* Horizontal position of title in subplot reference frame; proportion (0.0-1.0)

*ypos* Vertical position of title in subplot reference frame; proportion (0.0-1.0)

*x* Horizontal position of inset in subplot reference frame; proportion (0.0-1.0); (overrides *xpos*)

*y* Vertical position of inset in subplot reference frame; proportion (0.0-1.0); (overrides *ypos*)

**Returns:**

*text* <matplotlib.text.Text>

`plot_toolkit.format.text.set_title(...)`  
Prints a title for a figure

**Arguments:**



*figure* <matplotlib.figure.Figure> on which to act  
*text* Figure title text ('s' and 'title' also supported)  
*edge\_distance* Distance between top of subplots and vertical center of title; proportion (0.0-1.0)  
*fp* Figure title font in form of '##L'  
*x* Horizontal position of title in figure reference frame (0.0-1.0); (override)  
*y* Vertical position of title in figure reference frame (0.0-1.0); (override)

```
plot_toolkit.format.text.set_subtitle(...)
```

Prints a title above a subplot

#### Arguments:

*subplot* <matplotlib.axes.AxesSubplot> on which to act  
*label* Subplot label  
*fp* Subplot label font in form of '##L'

```
plot_toolkit.format.text.set_bigxlabel(...)
```

Prints a large X axis label shared by multiple subplots

#### Arguments:

*figure* <matplotlib.figure.Figure> on which to act  
*text* Figure X axis label text ('s' and 'label' also supported)  
*edge\_distance* Distance between bottom of subplots and vertical center of title; proportion (0.0-1.0)  
*fp* Figure X axis label font in form of '##L'  
*x* Horizontal position of title in figure reference frame (0.0-1.0); (override)  
*y* Vertical position of title in figure reference frame (0.0-1.0); (override)

#### Returns:

*text* <matplotlib.text.Text>

```
plot_toolkit.format.text.set_bigylabel(...)
```

Prints a large Y axis label shared by multiple subplots

#### Arguments:

*figure* <matplotlib.figure.Figure> on which to act  
*text* Figure Y axis label text ('s' and 'label' also supported)  
*fp* Figure Y axis label font in form of '##L'  
*edge\_distance* Distance between furthest left of subplots and horizontal center of title; proportion (0.0-1.0)  
*x* Horizontal position of label in figure reference frame; proportion (0.0-1.0); (override)  
*y* Vertical position of label in figure reference frame; proportion (0.0-1.0); (override)

#### Returns:

*text* <matplotlib.text.Text>

## 2.1.3 Legend

`plot_toolkit.format.legend.set_legend(...)`

Draws and formats a legend on *subplot*

By default includes all series, or accepts manual lists of *handles* and *labels* for plotted series

**Arguments:**

*subplot* <matplotlib.axes.AxesSubplot> on which to act

*handles* List of handles for plotted series (e.g. <matplotlib.lines.Line2D>)

*labels* List of labels for plotted series

*fp* Legend font in form of '##L'

*loc* Legend location

*kwargs* Keyword arguments to be passed to subplot.legend

**Returns:**

*legend* matplotlib.legend.Legend

## 2.2 Auxiliary

### 2.2.1 General

`plot_toolkit.auxiliary.abs_listdir(...)`

**Arguments:**

*directory* directory whose contents to list

**Yields:**

*file* Full path to each file

`plot_toolkit.auxiliary.hsl_to_rgb(...)`

**Arguments:**

*h* hue (0.0 - 1.0)

*s* saturation (0.0 - 1.0)

*l* luminescence (0.0 - 1.0)

**Returns:**

*rgb* Numpy array of equivalent red, green, blue

`plot_toolkit.auxiliary.rgb_to_hsl(...)`

**Arguments:**

*r* red (0.0 - 1.0)

*g* green (0.0 - 1.0)

*b* blue (0.0 - 1.0)

**Returns:**

*hsl* Numpy array of equivalent hue, saturation, luminescence

```
plot_toolkit.auxiliary.pad_zero(...)
```

**Arguments:**

*ticks* List or numpy array of ticks

**Returns:**

*tick\_labels* Tick labels, each with the same number of trailing zeros

## 2.2.2 Matplotlib

```
plot_toolkit.auxiliary.get_edges(...)
```

**Arguments:**

*figure* <matplotlib.figure.Figure> on which to act

**Returns:**

*edges* Dictionary; keys are 'x' and 'y', values are numpy arrays with dimensions (axis, min...max)

```
plot_toolkit.auxiliary.gen_font(...)
```

**Arguments:**

*fp* String of form '##L' in which '##' is the font size and 'L' is 'r' for regular or 'b' for bold

**Returns:**

*fp* <matplotlib.font\_manager.FontProperties> object to given specifications

```
plot_toolkit.auxiliary.gen_contour_levels(...)
```

**Arguments:**

*I* Intensity

*cutoff* Proportion of data below minimum level (0.0-1.0)

*include\_negative* Return levels for negative intensity as well as positive

**Returns:**

*levels* Numpy array of levels

```
plot_toolkit.auxiliary.gen_cmap(...)
```

Returns colormap that is *color* over all values

Not useful for heatmaps; useful for countours

**Arguments:**

*color* Tuple, list, or numpy array of red, green, and blue (0.0-1.0); Alternatively, string of named matplotlib color

**Returns:**

*cmap* <matplotlib.colors.LinearSegmentedColormap>

```
plot_toolkit.auxiliary.gen_figure_subplots(...)
```

Generates figure and subplots according to selected format

**Arguments:**

*format* String of form 'L#' in which 'L' is 'l' for landscape or 'p' for portrait and '#' is the number of subplots

*fig\_w* Figure width  
*fig\_h* Figure height  
*sub\_w* Subplot width  
*sub\_h* Subplot height  
*mar\_t* Top margin  
*mar\_r* Right margin  
*mar\_w* Horizontal margin between subplots  
*mar\_h* Vertical margin between subplots

**Returns:**

*figure* matplotlib.figure.Figure  
*subplots* OrderedDict of <matplotlib.axes.AxesSubplot> (1-indexed)

## DECORATORS

### 3.1 Figure\_Output

**class** `plot_toolkit.Figure_Output.Figure_Output(...)`

Decorator class to help functions that generate plots save figures more easily

**Arguments:**

*outfile* Output file name or `<matplotlib.backends.backend_pdf.PdfPages>`

**Behavior:**

Calls decorated function, which should return a `<matplotlib.figure.Figure>`

If *outfile* is a string ending in `'png'`, saves figure as a png file.

If *outfile* is a string ending in `'pdf'`, saves figure as a pdf file using PdfPages

If *outfile* is a PdfPages object, appends figure to that object as a page



**A**

abs\_listdir() (in module plot\_toolkit.auxiliary), 6

**F**

Figure\_Output (class in plot\_toolkit.Figure\_Output), 9

**G**

gen\_cmap() (in module plot\_toolkit.auxiliary), 7

gen\_contour\_levels() (in module plot\_toolkit.auxiliary), 7

gen\_figure\_subplots() (in module plot\_toolkit.auxiliary),  
7

gen\_font() (in module plot\_toolkit.auxiliary), 7

get\_edges() (in module plot\_toolkit.auxiliary), 7

**H**

hsl\_to\_rgb() (in module plot\_toolkit.auxiliary), 6

**P**

pad\_zero() (in module plot\_toolkit.auxiliary), 6

**R**

rgb\_to\_hsl() (in module plot\_toolkit.auxiliary), 6

**S**

set\_bigxlabel() (in module plot\_toolkit.format.text), 5

set\_bigylabel() (in module plot\_toolkit.format.text), 5

set\_colorbar() (in module plot\_toolkit.format.axes), 4

set\_inset() (in module plot\_toolkit.format.text), 4

set\_legend() (in module plot\_toolkit.format.legend), 6

set\_subtitle() (in module plot\_toolkit.format.text), 5

set\_text() (in module plot\_toolkit.format.text), 4

set\_title() (in module plot\_toolkit.format.text), 4

set\_xaxis() (in module plot\_toolkit.format.axes), 3

set\_yaxis() (in module plot\_toolkit.format.axes), 3