

---

# **Plot Toolkit Documentation**

***Release 0.1***

**Karl Debiec**

May 14, 2014



<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 . . . . .	6
<b>3</b>	<b>Decorators</b>	<b>9</b>
3.1	Figure_Output . . . . .	9
	<b>Python Module Index</b>	<b>11</b>
	<b>Index</b>	<b>13</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, however, applying strict formatting specifications to these plots typically requires writing a lot amount of code. Plot Toolkit's purpose is to reduce this.



## FUNCTIONS

## 2.1 Formatting

### 2.1.1 Axes

Functions for formatting axes

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

Formats an x-axis

**Arguments:**

*subplot* <Axes> on which to act

*ticks* Ticks

*tick\_kw* Keyword arguments to be passed to `set_xticks(...)`

*ticklabels* Tick labels

*tick\_fp* Tick label font; passed to `gen_font(...)`

*ticklabel\_kw* Keyword arguments to be passed to `set_xticklabels(...)`

*tick\_params* Keyword arguments to be passed to `set_tick_params(...)`

*label* Label text

*label\_fp* Label font; passed to `gen_font(...)`

*label\_kw* Keyword arguments to be passed to `set_xlabel(...)`

*lw* Width of x-axis lines

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

Formats a y-axis

**Arguments:**

*subplot* <Axes> on which to act

*ticks* Ticks

*tick\_kw* Keyword arguments to be passed to `set_yticks(...)`

*ticklabels* Tick labels

*tick\_fp* Tick label font; passed to `gen_font(...)`

*ticklabel\_kw* Keyword arguments to be passed to `set_yticklabels(...)`

*tick\_params* Keyword arguments to be passed to `set_tick_params(...)`

*label* Label text

*label\_fp* Label font; passed to `gen_font(...)`

*label\_kw* Keyword arguments to be passed to `set_ylabel(...)`

*lw* Width of y-axis lines

`plot_toolkit.axes.set_colorbar(...)`

Formats a colorbar

**Arguments:**

*cbar* <ColorBar> to act on

*ticks* Ticks

*ticklabels* Tick labels

*tick\_fp* Tick label font; passed to `gen_font(...)`

*label* Label text

*label\_fp* Label font; passed to `gen_font(...)`

## 2.1.2 Text

Functions for adding text labels and annotations

`plot_toolkit.text.set_title(...)`

Prints a title for a figure or subplot

**Arguments:**

*figure\_or\_subplot* <Figure> or <Axes> on which to act

*text* Title text; *s*, *t*, *tiile*, and *label* also supported

*fp* Title font; *fontproperties* also supported; passed to `gen_font(...)`

*top* Distance between top of figure and title (inches); applies to Figure title only

**Returns:**

*text* New <Text>

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

Prints a large x-axis label shared by multiple subplots

**Arguments:**

*figure* <Figure> on which to act

*text* Label text; *s*, *label*, and *xlabel* also supported

*fp* Label font; *fontproperties* also supported; passed to `gen_font(...)`

*bottom* Distance between bottom of figure and label (inches)

*top* Distance between top of figure and label (inches); overrides *bottom*

*x* Horizontal position of title in figure reference frame (proportion 0.0-1.0); overrides *bottom/top*

*y* Vertical position of title in figure reference frame (proportion 0.0-1.0); overrides *bottom/top*

**Returns:**

*text* New <Text>



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

Prints a large x-axis label shared by multiple subplots

**Arguments:**

*figure* <Figure> on which to act

*text* Label text; *s*, *label*, and *ylabel* also supported

*fp* Label font; *fontproperties* also supported; passed to *gen\_font(...)*

*left* Distance between left side of figure and label (inches)

*right* Distance between right side of figure and label (inches); overrides *left*

*x* Horizontal position of title in figure reference frame (proportion 0.0-1.0); overrides *left/right*

*y* Vertical position of title in figure reference frame (proportion 0.0-1.0); overrides *left/right*

*rotation* Label rotation; default = 'vertical'

**Returns:**

*text* New <Text>

```
plot_toolkit.text.set_inset(...)
```

Prints an inset to a subplot

**Arguments:**

*subplot* <Axes> on which to act

*text* Inset text; *s* and *inset* also supported

*fp* Inset font; *fontproperties* also supported; passed to *gen\_font(...)*

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

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

*x* Horizontal position of inset in subplot reference frame; overrides *xpos*

*y* Vertical position of inset in subplot reference frame; overrides *ypos*

*ha* Text horizontal alignment; default = 'left'

*va* Text vertical alignment; default = 'top'

**Returns:**

*text* New <Text>

```
plot_toolkit.text.set_text(...)
```

Prints text on a figure or subplot

**Arguments:**

*figure\_or\_subplot* <Figure> or <Axes> on which to act

*text* Text; *s* also supported

*fp* Font; *fontproperties* also supported; passed to *gen\_font(...)*

*ha* Text horizontal alignment; default = 'center'

*va* Text vertical alignment; default = 'center'

**Returns:**

*text* New <Text>

### 2.1.3 Legend

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

Draws and formats a legend on *subplot*

By default includes all series; may alternatively accept manual lists of *handles* and *labels* for plotted series

**Arguments:**

*subplot* <Axes> on which to act

*handles* List of handles for plotted series

*labels* List of labels for plotted series

*fp* Legend font; *fontproperties* and *prop* also supported; passed to `gen_font(...)`

**Returns:**

*legend* <Legend>

## 2.2 Auxiliary

### 2.2.1 General

General functions

`plot_toolkit.multi_kw(...)`

Function to allow arguments to be set by one of several potential keyword arguments. For example, the keyword argument *s* representing a string might be set using *s*, *text*, *label*, or if none of these are present, a default value. Note that *kwargs* should not be passed to this function using the `**` syntax.

**Arguments:**

*keywords* List of acceptable keyword arguments; first match is used and other are deleted

*default* Default value to use if none of *keywords* are present in *kwargs*

*kwargs* Dictionary of keyword arguments to be tested

**Returns:**

*value* Value from *kwargs* of first matching keyword in *keywords*, or *default* if none are present

`plot_toolkit.pad_zero(...)`

Returns a list of tick labels, each with the same number of digits after the decimal

**Arguments:**

*ticks* List or numpy array of ticks

*digits* Number of digits to include after the decimal

**Returns:**

*tick\_labels* Tick labels, each with the same number of digits after the decimal

### 2.2.2 Matplotlib

`plot_toolkit.get_edges(...)`

**Arguments:**

*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.gen_font(...)`

**Arguments:**

*fp* Font settings

**Behavior:**

If *fp* is <FontProperties>, acts as a pass-through, return *fp* argument

If *fp* is a String of form '##L', makes new <FontProperties>

'##' = size; 'L' = {'r': regular, 'b' bold}

If *fp* is a Dict, makes new <FontProperties> using given keyword arguments

**Returns:**

*fp* <FontProperties> object to given specifications

`plot_toolkit.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.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); or string of named matplotlib color

**Returns:**

*cmap* <LinearSegmentedColormap>

`plot_toolkit.gen_figure_subplots(...)`

Generates a figure and subplots to specifications

**Differs from matplotlib's built-in functions in that it:**

- Accepts input in inches rather than relative figure coordinates
- Optionally calculates figure dimensions from provided subplot dimensions, rather than the reverse
- Returns subplots in an OrderedDict
- Smoothly adds additional subplots to a previously-generated figure (i.e. can be called multiple times)

**Arguments:**

*nrows* Number of rows of subplots

*ncols* Number of columns of subplots

*sub\_width* Width of subplot(s)

*sub\_height* Height of subplot(s)

*top* Distance between top of figure and highest subplot(s)

*bottom* Distance between bottom of figure and lowest subplot(s)

*right* Distance between right side of figure and rightmost subplot(s)

*left* Distance between left side of figure and leftmost subplots(s)

*hspace* Vertical distance between adjacent subplots

*wspace* Horizontal distance between adjacent subplots

*fig\_width* Width of figure; by default calculated from above arguments

*fig\_height* Height of figure, by default calculated from above arguments

**Returns:**

*figure* <Figure>

*subplots* OrderedDict of subplots (1-indexed)

`plot_toolkit.identify(...)`

Identifies index of each subplot with inset text

**Arguments:**

*subplots* OrderedDict of subplots

## DECORATORS

### 3.1 Figure\_Output

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

Decorator class to allow plotting functions to 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



**p**

plot\_toolkit, 6  
plot\_toolkit.axes, 3  
plot\_toolkit.legend, 6  
plot\_toolkit.text, 4





## F

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

## G

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

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

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

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

get\_edges() (in module plot\_toolkit), 6

## I

identify() (in module plot\_toolkit), 8

## M

multi\_kw() (in module plot\_toolkit), 6

## P

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

plot\_toolkit (module), 6

plot\_toolkit.axes (module), 3

plot\_toolkit.legend (module), 6

plot\_toolkit.text (module), 4

## S

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

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

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

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

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

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

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

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

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