# ATLAS MPL Style

**Beojan Stanislaus**

**Apr 19, 2024**

# CONTENTS:

CHAPTER

# ONE

# ATLAS MATPLOTLIB STYLE

**Despite the last commit date, this package is still maintained. If you have any comments or improvements, open an issue or PR.**

Provides a Matplotlib style replicating that used by the ATLAS collaboration.

**Please open an issue if you find this style deviates from the guidelines.**

Install from PyPI using pip: `pip install atlas-mpl-style`

Documentation: https://atlas-mpl.readthedocs.io/en/latest/index.html

In addition, this package also provides:

- A function to draw the ATLAS label
- **A `plot` module containing functions to plot pre-binned histograms and** limits. This includes functionality for plotting stacked backgrounds along with data and ratios in the usual ATLAS style.
- A matplotlib style based on the background / foreground from the VIM Paper color scheme, along with a print version with a white background. - The default color cycle in all three styles is generated with HCL Wizard
- **Additional Matplotlib color definitions based on the Paper theme, and the** Oceanic Next theme

## 1.1 UHI and the PlottableHistogram protocol

With the development of the UHI interface, this package now has support for histogram objects that follow the `PlottableHistogram` protocol. `plot.Background` objects can be constructed using `PlottableHistograms` and a list of such `Backgrounds` can be passed to `plot.plot_backgrounds` omitting the `bins` argument. The other histogram plotting functions could not be modified to accept `PlottableHistogram` in a backward compatible manner since they take `bins` before the histogram argument. Alternate versions of these functions are therefore provided in the `uhi` module.

As a result of this support, the histogram objects returned by Uproot 4 can be plotted directly, as can Boost-Histogram histograms and Hist objects (once the relevent PRs are merged into those repositories).

## 1.2 `usetex=False` Mode

`usetex=False` is now the default, removing the LaTeX dependency. However, there are a few points to bear in mind.

1. The call to `draw_atlas_label` should be the last thing done before calling `savefig`.
2. The figure `dpi` is set to 72 to match that of the PDF backend. This *must not* be changed if the plot will be exported in PDF or (E)PS format since doing so would cause the spacing in the ATLAS label to be incorrect.
3. Due to the above, the `dpi` parameter should not be passed when exporting to a raster format.
4. When converting a plotting script that uses `usetex=True` mode, ensure labels are updated to remove LaTeX macros that are not supported by Matplotlib's mathtext.

## 1.3 TeXLive and Fonts Needed

If you have a full LaTeX installation available, you can use LaTeX to typeset the text by passing `usetex=True` to `use_atlas_style`. This will give you much greater options in terms of what can be included in labels.

A working TeXLive installation providing the following is required:

- pdflatex
- amsmath
- TeX Gyre Heros
- mathastext
- physics (the package)
- siunitx

If no LaTeX installation is available, the style will warn and fall back to the `usetex=False` behaviour. To check if all necessary packages are installed, try building `atlas_mpl_style/testing/ampl-test.tex`.

On Arch (and related distributions), the `texlive-most` group is sufficient.

On Debian (Jessie or above) or Ubuntu (18.04+), the following set of packages should be sufficient. It is however highly recommended that you install *texlive-full* to obtain a complete installation of texlive.

- texlive
- texlive-latex-extra
- texlive-fonts-recommended
- texlive-lang-greek
- tex-gyre
- dvipng
- ghostscript

On CentOS 7, the supplied TeXLive (2012) is extremely old. TeXLive should be installed from upstream.

**TeXLive is not required for the "slides" or "print" style.** Fira Sans and Iosevka should be installed for these styles to appear as intended. However, neither is *necessary*.

CHAPTER

# TWO

# EXAMPLE

This example will show you how to make a plot with data, a number of stacked MC components, a representative signal, and a ratio plot.

First, we just load some packages.

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats.distributions as dist
import boost_histogram as bh
```

## 2.1 Make Histograms

We need some histograms to plot, so let's generate some. We're making Boost histograms here but you can also read a TH1 from a ROOT file with Uproot and convert it. We can deal with anything that follows the UHI PlottableHistogram protocol.

```
rng = np.random.default_rng()
bkg1_dist = dist.expon(0, 3)
bkg2_dist = dist.expon(1, 3)
part1_dist = dist.norm(2, 0.2)
part2_dist = dist.norm(3, 0.4)

bkg1_data = 100*bkg1_dist.rvs(4000, rng)
bkg2_data = 100*bkg2_dist.rvs(1000, rng)
part1_data = 100*part1_dist.rvs(300, rng)
part2_data = 100*part2_dist.rvs(100, rng)
noise_data = 100*rng.uniform(0, 10, 200)

x_axis = bh.axis.Regular(30, 0, 1000)
bkg1_h = bh.Histogram(x_axis).fill(bkg1_data)
bkg2_h = bh.Histogram(x_axis).fill(bkg2_data)
part1_h = bh.Histogram(x_axis).fill(part1_data)
part2_h = bh.Histogram(x_axis).fill(part2_data)
error_h = bh.Histogram(x_axis).fill(part2_data)
data_h = bh.Histogram(x_axis).fill(bkg1_data).fill(bkg2_data).fill(part1_data).
↪fill(part2_data).fill(noise_data)
```

## 2.2 Make Plot

We start by loading ATLAS MPL Style, and activating the configuration.

```
import atlas_mpl_style as ampl
ampl.use_atlas_style()
```

First we setup the axes. `ratio_axes()` splits the figure into a large main area, and a smaller area below for the ratio plot. The two will have no space between them vertically, and share the x-axis.

```
fig, ax, rax = ampl.ratio_axes()
ax.set_xlim(0, 1000)
ax.set_ylim(0, 800);
```

First we plot the MC histograms. There's a slight misnomer here, and all the stacked histograms are called "Backgrounds", but they need not necessarily be backgrounds.

The return value is used to make the ratio plot. Note that unlike the other plotting functions this one is only in `ampl.plot`. Nevertheless it can still take UHI histograms.

```
bkg = ampl.plot.plot_backgrounds([
    ampl.plot.Background(label="Background 1", hist=bkg1_h, color="paper:blue"),
    ampl.plot.Background(label="Background 2", hist=bkg2_h, color="paper:green"),
    ampl.plot.Background(label="Particle 1", hist=part1_h, color="on:yellow"),
    ampl.plot.Background(label="Particle 2", hist=part2_h, color="on:red"),
], ax=ax)
```

Next we plot the data, and a "signal". This `plot_signal` function is used to plot a representative signal that is layered on top of the other histograms (rather than being stacked), and is drawn unfilled. You might want to boost the strength of this signal to ensure it is visible.

If you are plotting a signal component whose strength relative to the other MC components is accurate (e.g. the signal component of a fit) you should include that in the stack of "Background"s.

```
ampl.uhi.plot_data(hist=data_h, ax=ax)
ampl.uhi.plot_signal(label="Signal", hist=part1_h, color="paper:red")
ampl.uhi.plot_ratio(data_h, bkg, ratio_ax=rax, plottype='diff')
```

Now we set the x and y labels. Note that the `set_xlabel` function can be given the main axes, and the label will still be drawn below the ratio axes.

```
ampl.set_xlabel("Mass [GeV]", ax=ax)
ampl.set_ylabel("Count", ax=ax)

# This one uses the axis set_ylabel because we want it centre aligned
rax.set_ylabel(r"$\frac{{Data} - {Bkg}}{{Bkg}}$");
```

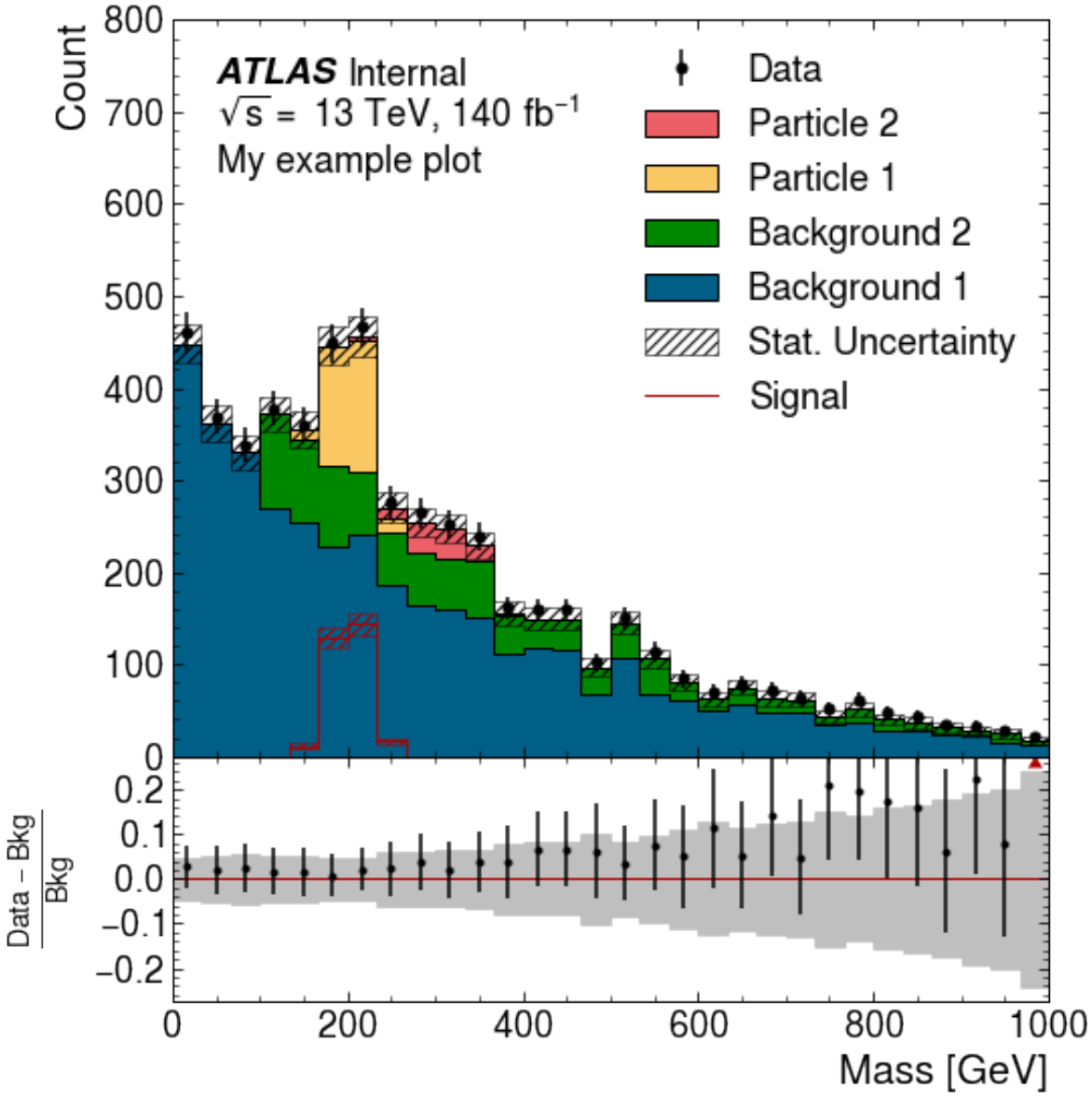
Finally we draw the ATLAS label and the legend. So long as the components of the plot have been drawn using the ATLAS MPL style functions the order of items in the legend will be determined automatically if you use the `ampl.draw_legend` function.

```
ampl.draw_atlas_label(0.05, 0.95, ax=ax, status='int', simulation=False, energy='13 TeV',
↪ lumi=140, desc="My example plot")
ampl.draw_legend(ax=ax); # Using one column here. If you have space, you can use ncols=2␣
↪for two columns
```

And save the figure, ensuring everything is visible.

```
fig.tight_layout()
plt.savefig('test.png')
```

## 2.3 Output



CHAPTER

THREE

# ATLAS STYLE

The main purpose of this package is to provide a Matplotlib style closely resembling that used by the ATLAS experiment for it's plots.

This style can be activated by calling

```
import atlas_mpl_style as ampl
ampl.use_atlas_style()
```

When the ATLAS style is active, text is typeset using LaTeX, and the standard ATLAS label can be drawn using the `ampl.draw_atlas_label` function. The use of LaTeX can be disabled by instead calling `ampl.use_atlas_style(usetex=False)`.

The axis labels should be set using the `ampl.set_xlabel` and `ampl.set_ylabel` functions, to ensure they are correctly right / top aligned.

CHAPTER

# FOUR

# OTHER STYLES

Additionally, two other styles based on the Paper VIM color scheme are provided. **Slides** has an off-white central background, and works well on slides. **Print** has a white background, for use in print. These styles do not use LaTeX for text typesetting, and can be activated using

```
# import matplotlib in the usual way
import matplotlib.pyplot as plt
import atlas_mpl_style as ampl
plt.style.use('slides')
# or
plt.style.use('print')
```

CHAPTER

# FIVE

# ADDITIONAL COLORS

ATLAS MPL Style adds a number of additional color definitions to Matplotlib.

## 5.1 Default Color Cycle

These are the colors of the default color cycle.

| | | |
|---|---|---|
| series:cyan | #54c9d1 |  |
| series:orange | #eca89a |  |
| series:blue | #95bced |  |
| series:olive | #ceb776 |  |
| series:purple | #d3a9ea |  |
| series:green | #9bc57f |  |
| series:pink | #f0a1ca |  |
| series:turquoise | #5fcbaa |  |

## 5.2 Paper Colors

These colors are from the Paper color scheme for VIM.

| | |
|---|---|
| paper:bg | #eeeeee |
| paper:fg | #444444 |
| paper:bgAlt | #e4e4e4 |
| paper:red | #af0000 |
| paper:green | #008700 |
| paper:blue | #005f87 |
| paper:yellow | #afaf00 |
| paper:orange | #d75f00 |
| paper:pink | #d70087 |
| paper:purple | #8700af |
| paper:lightBlue | #0087af |
| paper:olive | #5f7800 |

## 5.3 Oceanic Next Colors

These colors are from the Oceanic Next color scheme.

| | |
|---|---|
| on:bg | #1b2b34 |
| on:fg | #cdd3de |
| on:bgAlt | #343d46 |
| on:fgAlt | #d8dee9 |
| on:red | #ec5f67 |
| on:orange | #f99157 |
| on:yellow | #fac863 |
| on:green | #99c794 |
| on:cyan | #5fb3b3 |
| on:blue | #6699cc |
| on:pink | #c594c5 |
| on:brown | #ab7967 |

## 5.4 ATLAS Color Cycle

These are the colors from the `ATLAS` color cycle. Green and yellow also have the additional aliases `atlas:onesigma` and `atlas:twosigma` respectively for use in limit plots.

| | |
|---|---|
| `atlas:onesigma` | `#00ff26` |
| `atlas:twosigma` | `#fbff1f` |
| `series2:green` | `#00ff26` |
| `series2:yellow` | `#fbff1f` |
| `series2:blue` | `#00a1e0` |
| `series2:red` | `#a30013` |
| `series2:purple` | `#5100c2` |

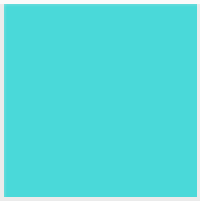
## 5.5 HDBS Colors

These are the ATLAS HDBS physics groups colors. Mint cream is not included in the color cycle.

| | |
|---|---|
| hdbs:starcommandblue | #047cbc |
| hdbs:spacecadet | #283044 |
| hdbs:maroonX11 | #b8336a |
| hdbs:outrageousorange | #fa7e61 |
| hdbs:pictorialcarmine | #ca1551 |
| hdbs:mintcream | #ebf5ee |

## 5.6 HH Colors

These are the ATLAS di-Higgs colors. Light turquoise and off-white are not included in the automatic color cycle.

| | | |
|---|---|---|
| hh:darkblue | #343844 |  |
| hh:darkpink | #f2385a |  |
| hh:darkyellow | #fdc536 |  |
| hh:medturquoise | #36b1bf |  |
| hh:lightturquoise | #4ad9d9 |  |
| hh:offwhite | #e9f1df |  |

## 5.7 Transparent

A fully transparent color is also provided for convenience.

| | |
|---|---|
| transparent | #ffffff00 |

CHAPTER

# SIX

# ATLAS_MPL_STYLE MODULE

atlas_mpl_style.**ratio_axes**(*extra_axes=None*)

Splits axes for ratio plots.

**Parameters**

- **extra_axes** (*int, optional*) – Number of additional axes. If not given, defaults to one.
- **square** (*bool, optional*) – Whether the plot should be square or tall. Defaults to True (square)

**Returns**

- **fig** (*figure*)
- **main_ax** (*axes*)
- **ratio_ax** (*axes or list of axes*) – Returns list if extra_axes is passed

atlas_mpl_style.**set_color_cycle**(*pal=None*, *n=4*)

Sets a different color cycle.

The ATLAS palette includes the standard green and yellow.

**Parameters**

- **pal** (*{'ATLAS', 'Paper', 'Oceanic', 'MPL', "HDBS", "HH", None}*) – The palette to use. None resets to default palette. The ATLAS palette is suitable for histograms, not lines.

  'MPL' (alias 'Tab') provides the default matplotlib palette.

- **n** (*int, optional*) – Number of lines or histograms.

atlas_mpl_style.**use_atlas_style**(*atlasLabel='ATLAS'*, *fancyLegend=False*, *usetex=False*)

Setup ATLAS style.

**Parameters**

- **atlasLabel** (*str, optional*) – Replace ATLAS with a custom label
- **fancyLegend** (*bool, optional*) – Use matplotlib's fancy legend frame (defaults to False)
- **usetex** (*bool, optional*) – Use LaTeX installation to set text (defaults to False) If no LaTeX installation is found, this package will fallback to usetex=False. This is on a best-effort basis, since the detected LaTeX installation may be incomplete.

CHAPTER

# SEVEN

## ATLAS_MPL_STYLE.PLOT MODULE

**class** atlas_mpl_style.plot.**Background**(*label*, *hist*, *stat_errs=None*, *syst_errs=None*, *color=None*)

Histogram and errors corresponding to a single background

**exception** atlas_mpl_style.plot.**BinningMismatchError**(*msg*)

Error due to histogram binning mismatch

**exception** atlas_mpl_style.plot.**DimensionError**(*msg*)

Error due to incorrect / unsupported histogram dimension

**exception** atlas_mpl_style.plot.**ViolatesPlottableHistogramError**(*msg*)

Error due to histogram object violating the PlottableHistogram protocol

atlas_mpl_style.plot.**draw_atlas_label**(*x*, *y*, *ax=None*, *status='int'*, *simulation=False*, *energy=None*, *lumi=None*, *desc=None*, *lumi_lt=False*, **args*, ***kwargs*)

Draw ATLAS label.

Additional parameters are passed through to `ax.text`.

**Parameters**

- **x** (*float*) – x position (top left)
- **y** (*float*) – y position (top left)
- **ax** (*mpl.axes.Axes, optional*) – Axes to draw label in
- **status** ([ *'int'* | 'wip' | 'prelim' | 'final' | 'opendata' ], optional) – Approval status
- **simulation** (bool (optional, default `False`)) – Does the plot show only MC simulation results
- **energy** (*str, optional*) – Centre of mass energy, including units
- **lumi** (*float or str, optional*) – Integrated luminosity in /fb. If str, the units should be included.
- **lumi_lt** (*bool, optional*) – True if only a subset of data was processed
- **desc** (*str, optional*) – Additional description

atlas_mpl_style.plot.**draw_legend**(**args*, *ax=None*, ***kwargs*)

Add legend to axes with data first, and uncertainties last.

**Parameters**

- **ax** (*mpl.axes.Axes, optional*) – Axes to draw legend on (defaults to current axes)
- ***args** – Passed to `ax.legend`
- ****kwargs** – Passed to `ax.legend`

atlas_mpl_style.plot.**draw_tag**(*text*, *ax=None*)

Draw tag just outside plot region

**Parameters**

- **text** (*str*)
- **ax** (*mpl.axes.Axes, optional*) – Axes to draw on (defaults to current axes)

atlas_mpl_style.plot.**plot_1d**(*label*, *bins*, *hist*, *stat_errs=None*, *color=None*, *attach_bands=False*, *ax=None*, ***kwargs*)

Plot single 1D histogram

NB: *atlas_mpl_style.uhi.plot_1d()* provides a version of this function that accepts a `PlottableHistogram`.

**Parameters**

- **label** (*str*) – Label for legend
- **bins** (*array_like*) – Bin edges
- **hist** (*array_like*) – Bin contents
- **stat_errs** (*array_like, optional*) – Statistical errors
- **color** (*color, optional*) – Line color
- **attach_bands** (*boolean, optional*) – Attach bands to line in legend. Defaults to False.
- **ax** (*mpl.axes.Axes, optional*) – Axes to draw on (defaults to current axes)
- ****kwargs** – Extra parameters passed to `plt.hist`

atlas_mpl_style.plot.**plot_2d**(*xbins*, *ybins*, *hist*, *ax=None*, *pad=0.05*, ***kwargs*)

Plot 2D histogram

NB: *atlas_mpl_style.uhi.plot_2d()* provides a version of this function that accepts a `PlottableHistogram`.

**Parameters**

- **xbins** (*array_like*) – x bin edges
- **bins** (*array_like*) – y bin edges
- **hist** (*array_like*) – Bin contents
- **ax** (*mpl.axes.Axes, optional*) – Axes to draw on (defaults to current axes)
- **pad** (*float, optional*) – Padding for colorbar in inches (defaults to 0.05)
- ****kwargs** – Extra parameters passed to `pcolormesh`

**Returns**

- **mesh** (*QuadMesh*)
- **cbar** (*mpl.colorbar.Colorbar*)

atlas_mpl_style.plot.**plot_backgrounds**(*backgrounds*, *bins=None*, ***, *total_err=None*, *empty_stat_legend=False*, *ax=None*)

Plot stacked backgrounds

**Parameters**

- **backgrounds** (*[Background]*) – List of backgrounds to be plotted, in order (bottom to top)

- **bins** (*array_like, optional*) – Bin edges. To preserve backward compatibility, `backgrounds` and `bins` may be exchanged.
- **total_err** (*array_like, optional*) – Total uncertainty. If given, overrides per-background systematics. This is useful for showing post-fit uncertainties.
- **empty_stat_legend** (*boolean, optional*) – Add stat error band to legend even if empty. Defaults to False.
- **ax** (*mpl.axes.Axes, optional*) – Axes to draw on (defaults to current axes)

**Returns**

- **total_hist** (*array_like*) – Total background histogram
- **total_err** (*array_like*) – Total error on background histogram

atlas_mpl_style.plot.**plot_band**(*bins*, *low*, *high*, *label=None*, *ax=None*, ***kwargs*)

Draw a shaded band between high and low

Use this for drawing error bands

**Parameters**

- **bins** (*array_like*) – Bin edges
- **low** (*array_like*) – Bin contents defining lower bound
- **high** (*array_like*) – Bin contents defining upper bound
- **label** (*str, optional*) – Label for legend. If label matches a line, the band will be attached to that line if `draw_legend` is used.
- **ax** (*mpl.axes.Axes, optional*) – Axes to draw band on (defaults to current axes)
- ****kwargs** – Keyword arguments passed to *fill_between*

atlas_mpl_style.plot.**plot_cutflow**(*labels*, *hist*, *ax=None*, *text=True*, *textcolor='w'*, *horizontal=True*, ***kwargs*)

Plot cutflow from PlottableHistogram

**Parameters**

- **labels** (*[str]*) – Cutflow labels
- **hist** (*PlottableHistogram*) – Cutflow histogram
- **ax** (*mpl.axes.Axes, optional*) – Axes to draw on (defaults to current axes)
- **text** (*bool, optional*) – Whether to label bars (default: True)
- **textcolor** (*str, optional*) – Text color
- **horizontal** (*bool, optional*) – Whether to draw horizontal bars (default: True)
- ****kwargs** – Extra parameters passed to `bar` or `barh`

atlas_mpl_style.plot.**plot_data**(*bins*, *hist*, *stat_errs=None*, *color='k'*, *label='Data'*, *ax=None*)

Plot data

NB: *atlas_mpl_style.uhi.plot_data()* provides a version of this function that accepts a `PlottableHistogram`.

**Parameters**

- **label** (*str, optional*) – Label for legend (default: “Data”)
- **bins** (*array_like*) – Bin edges

- **hist** (*array_like*) – Bin contents
- **stat_errs** (*array_like, optional*) – Statistical errors
- **color** (*color, optional*) – Point color, defaults to black
- **ax** (*mpl.axes.Axes, optional*) – Axes to draw on (defaults to current axes)

**Returns**

- **hist** (*array_like*) – Data histogram
- **stat_errs** (*array_like*) – Statistical errors

atlas_mpl_style.plot.**plot_limit**(*expected_label*, *x*, *expected*, *minus_one_sigma=None*, *plus_one_sigma=None*, *minus_two_sigma=None*, *plus_two_sigma=None*, *observed_label=None*, *observed=None*, *color=None*, *ax=None*)

Plot a limit

**Parameters**

- **expected_label** (*str*) – Label for expected limit (for legend)
- **x** (*array_like*) – x values
- **expected** (*array_like*) – Expected limit
- **minus_one_sigma** (*array_like, optional*) – Lower edge of one sigma band
- **plus_one_sigma** (*array_like, optional*) – Upper edge of one sigma band
- **minus_two_sigma** (*array_like, optional*) – Lower edge of two sigma band
- **plus_two_sigma** (*array_like, optional*) – Upper edge of two sigma band
- **observed_label** (*str, optional*) – Label for observed limit
- **observed** (*array_like, optional*) – Observed limit
- **color** (*color, optional*) – Line color (if multiple limits are being drawn)
- **ax** (*mpl.axes.Axes, optional*) – Axes to draw on (defaults to current axes)

atlas_mpl_style.plot.**plot_ratio**(*bins*, *data*, *data_errs*, *bkg*, *bkg_errs*, *ratio_ax*, *max_ratio=None*, *plottype='diff'*, *offscale_errs=False*)

Plot ratio plot

NB: *atlas_mpl_style.uhi.plot_ratio()* provides a version of this function that accepts ``PlottableHistogram``s.

**Parameters**

- **bins** (*array_like*) – Bin edges
- **data** (*array_like*) – Data histogram bin contents
- **data_errs** (*array_like*) – Statistical errors on data
- **bkg** (*array_like*) – Total background histogram bin contents
- **bkg_errs** (*array_like*) – Total errors on total background
- **ratio_ax** (*mpl.axes.Axes*) – Ratio axes (produced using *atlas_mpl_style.ratio_axes()*)
- **max_ratio** (*float, optional*) – Maximum ratio (defaults to 0.25 for "diff", 1.25 for "raw", 3.5 for "significances").

- **plottype** (*{"diff", "raw", "significances"}*) –

  Type of ratio to plot.
  "diff" : (data - bkg) / bkg
  "raw" : data / bkg
  "significances" : Significances (using *atlas_mpl_style.utils.significance()*)

- **offscale_err** (*boolean*) – Draw error bars on off-scale points

atlas_mpl_style.plot.**plot_signal**(*label*, *bins*, *hist*, *stat_errs=None*, *syst_errs=None*, *color=None*, *attach_bands=False*, *ax=None*)

Plot signal histogram

NB: *atlas_mpl_style.uhi.plot_signal()* provides a version of this function that accepts a `PlottableHistogram`.

**Parameters**

- **label** (*str*) – Label for legend
- **bins** (*array_like*) – Bin edges
- **hist** (*array_like*) – Bin contents
- **stat_errs** (*array_like*) – Statistical errors
- **syst_errs** (*array_like*) – Systematic errors
- **color** (*color*) – Line color
- **attach_bands** (*boolean, optional*) – Attach bands to line in legend. Defaults to False.
- **ax** (*mpl.axes.Axes, optional*) – Axes to draw on (defaults to current axes)

atlas_mpl_style.plot.**register_band**(*label*, *artist*, *ax=None*)

Register a manually draw (e.g. with `fill_between`) error band.

**Parameters**

- **label** (*str*) – Label of line to attach band to.
- **artist** (*mpl.artist.Artist*) – Band artist, e.g. `PolyCollection` returned by `fill_between`.
- **ax** (*mpl.axes.Axes, optional*) – Axes to register band in (defaults to current axes).

atlas_mpl_style.plot.**set_xlabel**(*label*, *ax=None*, **args*, ***kwargs*)

Set x label in ATLAS style (right aligned). If `ratio_axes` was used, the label will be set on the lowest ratio axes.

Additional parameters are passed through to *ax.set_xlabel*.

**Parameters**

- **label** (*str*) – Label (LaTeX permitted)
- **ax** (*mpl.axes.Axes, optional*) – Axes to set x label on

atlas_mpl_style.plot.**set_ylabel**(*label*, *ax=None*, **args*, ***kwargs*)

Set y label in ATLAS style (top aligned).

Additional parameters are passed through to `ax.set_ylabel`.

**Parameters**

- **label** (*str*) – Label (LaTeX permitted)

- **ax** (*mpl.axes.Axes, optional*) – Axes to set y label on

atlas_mpl_style.plot.**set_zlabel**(*label*, *cbar=None*, *ax=None*, ***kwargs*)

Set z label in ATLAS style (top aligned)

The colorbar to add the label to is *required* unless `plot_2d` was used.

**Parameters**

- **label** (*str*) – Label (LaTeX permitted)
- **cbar** (*mpl.colorbar.Colorbar, optional*) – Colorbar to set label on. Not required if `plot_2d` was used.
- **ax** (*mpl.axes.Axes, optional*) – If `plot_2d` was used, the axes can optionally be provided here.

CHAPTER

# EIGHT

# ATLAS_MPL_STYLE.STATS MODULE

**exception** atlas_mpl_style.stats.**IncorrectAxesError**(*msg*)

Error due to passing incorrect axes to draw_pull_impact_legend

atlas_mpl_style.stats.**draw_pull_impact_legend**(**args*, *ax=None*, ***kwargs*)

Add legend to a pull / impact plot.

**Parameters**

- **ax** (*mpl.axes.Axes, optional*) – Pull axes
- ****kwargs** – Passed to `ax.legend`

atlas_mpl_style.stats.**make_impact_figure**(*num_parameters*)

Create a new figure for a pull / impact plot.

**Parameters**
**num_parameters** (*Integer*) – Number of parameters on this plot

**Returns**
**fig** – Created figure

**Return type**
Figure

atlas_mpl_style.stats.**plot_impacts**(*data*, *draw_prefit=False*, *up_color='paper:blue'*, *down_color='paper:red'*, *ax=None*)

Plot impacts from a Pandas dataframe (`data`).

The dataframe must have the following columns. The prefit columns are not required if `draw_prefix == False`.

| name | Parameter name |
|---|---|
| impact_prefit_up | Impact on POI from fit with parameter fixed to +1 (using prefit ) |
| impact_prefit_down | Impact on POI from fit with parameter fixed to -1 (using prefit ) |
| impact_postfit_up | Impact on POI from fit with parameter fixed to +1 (using postfit ) |
| impact_postfit_down | Impact on POI from fit with parameter fixed to -1 (using postfit ) |

**Parameters**

- **data** (*pd.DataFrame*) – Pandas dataframe containing impacts
- **draw_prefit** (*Boolean, optional*) – Whether to draw the prefit bands
- **up_color** (*Color specification*) – Color to use for upper band (postfit band will be at 50% opacity)

- **up_color** – Color to use for lower band (postfit band will be at 50% opacity)
- **ax** (*mpl.axes.Axes, optional*) – Axes to pulls were drawn on (defaults to current axes). Impact axes will be a twin of these.

**Returns**
**ax** – The axes impacts were drawn on

**Return type**
mpl.axes.Axes

atlas_mpl_style.stats.**plot_pulls**(*data*, *ax=None*, ***kwargs*)

Plot pulls from a Pandas dataframe (`data`).

The dataframe must have at least the following columns:

| | |
|---|---|
| name | Parameter name |
| value | Post nominal fit central value of parameter |
| err_high | Post nominal fit error (high side) on parameter |
| err_low | Post nominal fit error (low side) on parameter. |

**Parameters**

- **data** (*pd.DataFrame*) – Pandas dataframe containing pulls
- **ax** (*mpl.axes.Axes, optional*) – Axes to draw pulls on (defaults to current axes)
- ****kwargs** – Keyword arguments passed to `errorbar`

**Returns**

- **ax** (*mpl.axes.Axes*) – The axes pulls were drawn on
- **pull_plot** (*mpl.container.ErrorbarContainer*) – The return value of `errorbar`

atlas_mpl_style.stats.**sort_impacts**(*data*)

Sort an impact (and pull) dataframe in-place by descending postfit impacts.

The frame must have at least the following columns:

| | |
|---|---|
| name | Parameter name |
| impact_postfit_up | Impact on POI from fit with parameter fixed to +1 (using postfit ) |
| impact_postfit_down | Impact on POI from fit with parameter fixed to -1 (using postfit ) |

**Parameters**
**data** (*pd.DataFrame*) – Pandas dataframe containing impacts

CHAPTER

# NINE

# ATLAS_MPL_STYLE.UHI MODULE

This module contains versions of the histogram plotting functions that take PlottableHistograms

These are in a separate module to preserve backward compatibility since the array versions of these functions take the array of bins before the histogram.

*atlas_mpl_style.plot.Background* can be constructed using a `PlottableHistogram` and therefore there is no `atlas_mpl_style.uhi.plot_backgrounds` function.

**exception** atlas_mpl_style.uhi.**LabeledBinsError**(*msg*)

Labeled bins when edges expected (or vice versa)

atlas_mpl_style.uhi.**plot_1d**(*hist*, *label*, *ignore_variances=False*, *stat_err=True*, *color=None*, *attach_bands=False*, *ax=None*, ***kwargs*)

Plot single 1D histogram from PlottableHistogram

**Parameters**

- **hist** (*PlottableHistogram*) – Histogram
- **label** (*str*) – Label for legend
- **ignore_variances** (*bool*) – Ignore variances and substitute `hist`. Defaults to False.
- **stat_err** (*bool*) – Draw statistical errors. Defaults to True.
- **color** (*color, optional*) – Line color
- **attach_bands** (*boolean, optional*) – Attach bands to line in legend. Defaults to False.
- **ax** (*mpl.axes.Axes, optional*) – Axes to draw on (defaults to current axes)
- ****kwargs** – Extra parameters passed to `plt.hist`

atlas_mpl_style.uhi.**plot_2d**(*hist*, *ax=None*, *pad=0.05*, ***kwargs*)

Plot 2D histogram from PlottableHistogram

**Parameters**

- **hist** (*PlottableHistogram*) – Histogram
- **ax** (*mpl.axes.Axes, optional*) – Axes to draw on (defaults to current axes)
- **pad** (*float, optional*) – Padding for colorbar in inches (defaults to 0.05)
- ****kwargs** – Extra parameters passed to `pcolormesh`

**Returns**

- **mesh** (*QuadMesh*)
- **cbar** (*mpl.colorbar.Colorbar*)

atlas_mpl_style.uhi.**plot_cutflow**(*hist*, *ax=None*, *text=True*, *textcolor='w'*, *horizontal=True*, ***kwargs*)

Plot cutflow from PlottableHistogram

**Parameters**

- **hist** (*PlottableHistogram*) – Cutflow histogram
- **ax** (*mpl.axes.Axes, optional*) – Axes to draw on (defaults to current axes)
- **text** (*bool, optional*) – Whether to label bars (default: True)
- **textcolor** (*str, optional*) – Text color
- **horizontal** (*bool, optional*) – Whether to draw horizontal bars (default: True)
- ****kwargs** – Extra parameters passed to `bar` or `barh`

atlas_mpl_style.uhi.**plot_data**(*hist*, *ignore_variances=False*, *color='k'*, *label='Data'*, *ax=None*)

Plot data from PlottableHistogram

**Parameters**

- **label** (*str, optional*) – Label for legend (default: "Data")
- **hist** (*PlottableHistogram*) – Histogram
- **ignore_variances** (*bool*) – Ignore variances and substitute `hist`. Defaults to False.
- **color** (*color, optional*) – Point color, defaults to black
- **ax** (*mpl.axes.Axes, optional*) – Axes to draw on (defaults to current axes)

**Returns**

- **hist** (*array_like*) – Data histogram
- **stat_errs** (*array_like*) – Statistical errors

atlas_mpl_style.uhi.**plot_ratio**(*data*, *total_bkg*, *ratio_ax*, *max_ratio=None*, *plottype='diff'*)

Plot ratio plot from PlottableHistogram

**Parameters**

- **data** (*PlottableHistogram*) – Data histogram
- **total_bkg** (*(array_like, array_like)*) – Tuple returned from *atlas_mpl_style.plot.plot_backgrounds()*
- **ratio_ax** (*mpl.axes.Axes*) – Ratio axes (produced using *atlas_mpl_style.ratio_axes()*)
- **max_ratio** (*float, optional*) – Maximum ratio (defaults to 0.2 for "diff", 1.2 for "raw", 3 for "significances")
- **plottype** (*{"diff", "raw", "significances"}*) –

  Type of ratio to plot.
  "diff" : (data - bkg) / bkg
  "raw" : data / bkg
  "significances" : Significances (from *ampl.utils.significance()*)

atlas_mpl_style.uhi.**plot_signal**(*hist*, *label*, *ignore_variances=False*, *syst_errs=None*, *color=None*, *attach_bands=False*, *ax=None*)

Plot signal histogram from PlottableHistogram

**Parameters**

- **hist** (*PlottableHistogram*) – Histogram
- **label** (*str*) – Label for legend
- **ignore_variances** (*bool*) – Ignore variances and substitute `hist`. Defaults to False.
- **syst_errs** (*array_like or PlottableHistogram, optional*) – Systematic errors
- **color** (*color*) – Line color
- **attach_bands** (*boolean, optional*) – Attach bands to line in legend. Defaults to False.
- **ax** (*mpl.axes.Axes, optional*) – Axes to draw on (defaults to current axes)

CHAPTER

# TEN

## ATLAS_MPL_STYLE.UTILS MODULE

atlas_mpl_style.utils.**significance**(*data*, *data_errs*, *bkg*, *bkg_errs*)

Calculates significance in each bin

Uses the significance definition in https://cds.cern.ch/record/2643488

**Parameters**

- **data** (*array_like*)
- **data_errs** (*array_like*) – Errors / uncertainties on *data*
- **bkg** (*array_like*) – Total background prediction
- **bkg_errs** (*array_like*) – Errors / uncertainties on *bkg*

CHAPTER

ELEVEN

# INDICES AND TABLES

- genindex
- modindex
- search

# PYTHON MODULE INDEX

## a

# INDEX