
fast-plotter Documentation

Release 0.9.0

['Ben Krikler']

May 01, 2019

Contents

1	Features	3
1.1	Command-line Usage	3
1.1.1	fast_plotter	3
1.2	The Plotting Config	4
1.3	Example repositories	6
1.3.1	Related Presentations	6
1.4	fast_plotter package	6
1.4.1	Submodules	6
2	Indices and tables	9
	Python Module Index	11

To install:

```
pip install fast-plotter
```

Or for development:

```
pip install -e git+https://gitlab.cern.ch/fast-hep/public/fast-plotter.git#egg=fast-  
↪plotter
```

- Free software: Apache Software License 2.0

- Functions to turn binned dataframes in the form produced from [fast-carpenter](#)
- A simple command-line tool to produce plots with decent defaults

1.1 Command-line Usage

The command-line tools are the primary way to use [fast-carpenter](#) and friends at this point. All of the FAST commands provide built-in help by providing the `--help` option.

1.1.1 `fast_plotter`

Take a list of [fast-carpenter](#) output binned dataframe tables and turns these into plots.

To configure how these plots are made, use either the command-line options, or provide these in a [YAML](#) configuration. If an option is provided to both, then the command-line value will take precedence.

```
$ fast_plotter --help
usage: fast_plotter [-h] [-c CONFIG] [-o OUTDIR] [-e EXTENSION] [-w WEIGHTS]
                  [-d DATA] [-s SIGNAL] [--dataset-col DATASET_COL]
                  [-l LUMI] [-y {log,linear}] [--halt-errors]
                  tables [tables ...]

Turn them tables into plots

positional arguments:
  tables                Table files to process

optional arguments:
  -h, --help            show this help message and exit
  -c CONFIG, --config CONFIG
                        YAML config to control common plotting options
```

(continues on next page)

(continued from previous page)

```
-o OUTDIR, --outdir OUTDIR
    Output directory to save plots to
-e EXTENSION, --extension EXTENSION
    File extension for images
-w WEIGHTS, --weights WEIGHTS
    comma-separated list of weight schemes to plot things
    for
-d DATA, --data DATA
    Regular expression to identify real data datasets from
    their name
-s SIGNAL, --signal SIGNAL
    Regular expression to identify signal MC datasets from
    their name
--dataset-col DATASET_COL
    Name of column to be used to define multiple-lines for
    1D plots
-l LUMI, --lumi LUMI
    Scale the MC yields by this lumi
-y {log,linear}, --yscale {log,linear}
    Use this scale for the y-axis
--halt-errors
    Stop at the first time an error occurs
```

1.2 The Plotting Config

Use the config file to fine-tune your plots.

Valid options:

Option	Default	Description
<code>data</code>	<code>"data"</code>	Regular expression for values in <code>dataset_col</code> to plot as scatter markers
<code>weights</code>	None	Which weight columns to use for plotting.
<code>yscale</code>	<code>"linear"</code>	Should y-axis be on a linear or a log scale.
<code>lumi</code>	1	Multiply all simulated datasets by this value.
<code>ylabel</code>		Give the Y-axis a title.
<code>legend</code>		Control the legend placement options. A dictionary of kwarg pairs passed directly to <code>matplotlib.pyplot.legend()</code>
<code>limits</code>		Set the axis ranges. Takes a dictionary with <code>x</code> and / or <code>y</code> as the keys, and a list with upper and lower bounds.
<code>annotations</code>		Add text to the plot. Should be a list of text labels, each described by a dictionary containing the text and the position. All other paramers are passed as keyword-argument pairs to <code>matplotlib.pyplot.annotate()</code>

Todo: Describe the `bin_variable_replacements` and `value_replacements` options for the config.

See also:

An example of a plotting config cms_public_tutorial demo repository: https://gitlab.cern.ch/fast-hep/public/fast cms_public_tutorial/blob/master/plot_config.yml

1.3 Example repositories

- A full demo based on the public CMS tutorial using 2012 data: https://gitlab.cern.ch/fast-hep/public/fast_cms_public_tutorial

1.3.1 Related Presentations

1. [IRIS-HEP 4th March 2019](<https://indico.cern.ch/event/802182/contributions/3334624/>)

1.4 fast_plotter package

1.4.1 Submodules

fast_plotter.interval_from_str module

fast_plotter.interval_from_str.**convert_intervals** (df, to='mid', level=[], column=[])

fast_plotter.interval_from_str.**convert_intervals_column** (df, to='mid', select=[])

fast_plotter.interval_from_str.**convert_intervals_level** (df, to='mid', select=[])

fast_plotter.interval_from_str.**interval_from_string** (series)

fast_plotter.plotting module

class fast_plotter.plotting.**BarColl** (n_colors=10, ax=None, fill=True, line=True)

Bases: fast_plotter.plotting.FillColl

class fast_plotter.plotting.**FillColl** (n_colors=10, ax=None, fill=True, line=True)

Bases: object

pre_call (col)

fast_plotter.plotting.**actually_plot** (df, x_axis, y, yerr, kind, label, ax, dataset_col='dataset')

fast_plotter.plotting.**add_annotations** (annotations, ax)

fast_plotter.plotting.**pad_zero** (x, *y_values)

fast_plotter.plotting.**plot_1d** (df, kind='line', yscale='lin')

fast_plotter.plotting.**plot_1d_many** (df, prefix="", data='data', signal=None, dataset_col='dataset', plot_sims='stack', plot_data='sum', plot_signal=None, kind_data='scatter', kind_sims='fill-error-last', kind_signal='line', scale_sims=None, summary='ratio')

fast_plotter.plotting.**plot_all** (df, project_1d=True, project_2d=True, data='data', signal=None, dataset_col='dataset', yscale='log', lumi=None, annotations=[], dataset_order='sum-ascending', continue_errors=True, bin_variable_replacements={}, **kwargs)

fast_plotter.plotting.**plot_ratio** (data, sims, x, y, yvar, ax)

fast_plotter.utils module

fast_plotter.utils.**interval_from_string**(*series*)

fast_plotter.utils.**convert_intervals**(*df*, *to='mid'*, *level=[]*, *column=[]*)

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

f

`fast_plotter`, 6
`fast_plotter.interval_from_str`, 6
`fast_plotter.plotting`, 6
`fast_plotter.utils`, 7

A

`actually_plot()` (in module `fast_plotter.plotting`), 6
`add_annotations()` (in module `fast_plotter.plotting`), 6

B

`BarColl` (class in `fast_plotter.plotting`), 6

C

`convert_intervals()` (in module `fast_plotter.interval_from_str`), 6
`convert_intervals()` (in module `fast_plotter.utils`), 7
`convert_intervals_column()` (in module `fast_plotter.interval_from_str`), 6
`convert_intervals_level()` (in module `fast_plotter.interval_from_str`), 6

F

`fast_plotter` (module), 6
`fast_plotter.interval_from_str` (module), 6
`fast_plotter.plotting` (module), 6
`fast_plotter.utils` (module), 7
`FillColl` (class in `fast_plotter.plotting`), 6

I

`interval_from_string()` (in module `fast_plotter.interval_from_str`), 6
`interval_from_string()` (in module `fast_plotter.utils`), 7

P

`pad_zero()` (in module `fast_plotter.plotting`), 6
`plot_1d()` (in module `fast_plotter.plotting`), 6
`plot_1d_many()` (in module `fast_plotter.plotting`), 6
`plot_all()` (in module `fast_plotter.plotting`), 6
`plot_ratio()` (in module `fast_plotter.plotting`), 6
`pre_call()` (`fast_plotter.plotting.FillColl` method), 6