
Seabred Documentation

Release 0.0.1

Julia Ebert

Nov 30, 2018

Contents:

1	Functions	1
	Python Module Index	3

`seabred.plotting.lineplot(x, y, ctr_type='mean', err_type='std', show_trials=False, **kwargs)`

Plot the values with mean/median and std/95% CI/quartile as shading. This uses the automatic/default/preset color cycling for lines. Provided kwargs (including overriding color) will be passed to the line plotting

The style is meant to match that produced by `seaborn.lineplot`, but on numpy arrays and with way less overhead (i.e., not putting it into a pandas dataframe and using the seaborn plotting). It doesn't support a lot of the fancy extras of its seaborn cousin.

Note: This uses numpy's nan-functions (e.g., `nanmean` and `nanstd`) so your data can include nan values, and they will not contribute to summary statistic plots.

Parameters

- **x** (`np.ndarray`) – (n,) shape array of x-values to plot
- **y** (`np.ndarray`) – (m, n) shape array of y-values to plot, where m is the number of trials
- **ctr_type** (`str`, *optional*) – Which central statistic to plot is the primary summary metric. Options are 'mean' and 'median'. (the default is 'mean', which uses `numpy.nanmean`)
- **err_type** (`str`, *optional*) – Which error type to show for shading. Options are:
 - **std**: Standard deviation
 - **95ci**: 95% confidence interval
 - **quartile**: [25%, 75%] confidence interval
 - **None**: No error plotting
 (the default is 'std', which is standard deviation)
- **show_trials** (`bool`, *optional*) – Whether or not to show plots of the individual trials (each row of y data). (the default is False, which means only summary data shown)

`seabred.lineplot(x, y, ctr_type='mean', err_type='std', show_trials=False, **kwargs)`

Plot the values with mean/median and std/95% CI/quartile as shading. This uses the automatic/default/preset color cycling for lines. Provided kwargs (including overriding color) will be passed to the line plotting

The style is meant to match that produced by `seaborn.lineplot`, but on numpy arrays and with way less overhead (i.e., not putting it into a pandas dataframe and using the seaborn plotting). It doesn't support a lot of the fancy extras of its seaborn cousin.

Note: This uses numpy's nan-functions (e.g., `nanmean` and `nanstd`) so your data can include nan values, and they will not contribute to summary statistic plots.

Parameters

- **x** (`np.ndarray`) – (n,) shape array of x-values to plot
- **y** (`np.ndarray`) – (m, n) shape array of y-values to plot, where m is the number of trials
- **ctr_type** (`str`, *optional*) – Which central statistic to plot is the primary summary metric. Options are 'mean' and 'median'. (the default is 'mean', which uses `numpy.nanmean`)
- **err_type** (`str`, *optional*) – Which error type to show for shading. Options are:
 - **std**: Standard deviation
 - **95ci**: 95% confidence interval
 - **quartile**: [25%, 75%] confidence interval
 - **None**: No error plotting(the default is 'std', which is standard deviation)
- **show_trials** (`bool`, *optional*) – Whether or not to show plots of the individual trials (each row of y data). (the default is False, which means only summary data shown)

S

seabred, [1](#)
seabred.plotting, [1](#)

L

[lineplot\(\)](#) (in module `seabred`), 1
[lineplot\(\)](#) (in module `seabred.plotting`), 1

S

[seabred](#) (module), 1
[seabred.plotting](#) (module), 1