

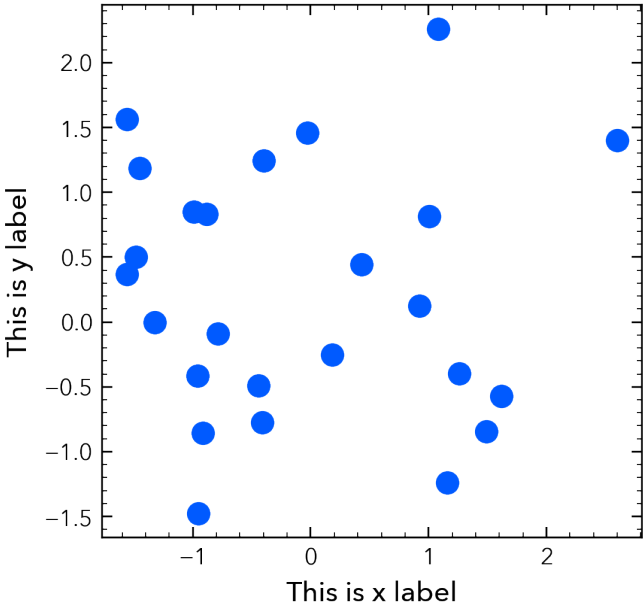
matplotlib で作成する図の調整方法

sugayu

2025 年 1 月 1 日

目次

1	基準の図	1
2	目盛り	1
2.1	目盛りの数字	1
3	矢印	1
3.1	arrow	1



4	大量の線	
---	------	--

5 グリッド分け

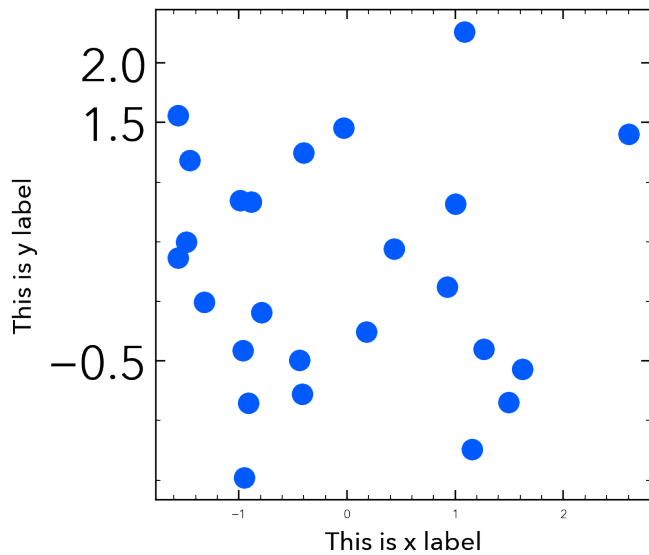
1 基準の図

```
1 from numpy.random import default_rng
2 from sugayutils.figure import makefig
3
4 rng = default_rng(222)
5 data = rng.standard_normal(50).reshape(2, 25)
6
7
8 def plot_fiducial():
9     fig = makefig(figsize=['small', 1.0])
10    ax = fig.add_subplot(1, 1, 1)
11    ax.scatter(data[0], data[1], c='blue')
12    ax.set_xlabel('This is x label', 'This is
    ↳ y label')
13    return ax
14
15 _ = plot_fiducial()
```

2 目盛り

2	2.1 目盛りの数字	
	目盛りの軸ラベルのサイズを変更し、縦軸の目盛りを指定する。	

```
2 1 ax = plot_fiducial()
2 2 ax.tick_params(labelsize='xx-small') # both
    ↳ axes
3 3 ax.tick_params(axis='y', labelsz=20)
4 4 _ = ax.set_yticks([-0.5, 1.5, 2.0])
```



3 矢印

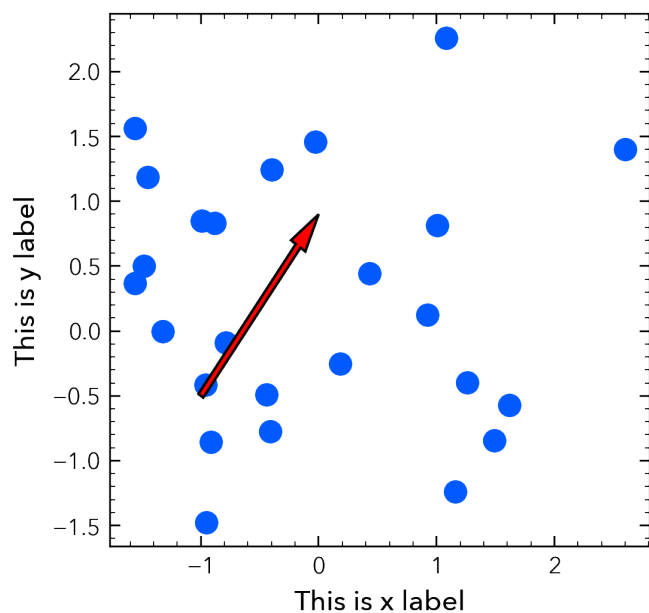
3.1 arrow

データ座標を使って矢印を描く。

```

1 ax = plot_fiducial()
2 _ = ax.arrow(
3     x=-1.0,
4     y=-0.5,
5     dx=1.0,
6     dy=1.4,
7     width=0.05,
8     head_length=0.3,
9     length_includes_head=True,
10    fc='red',
11 )

```



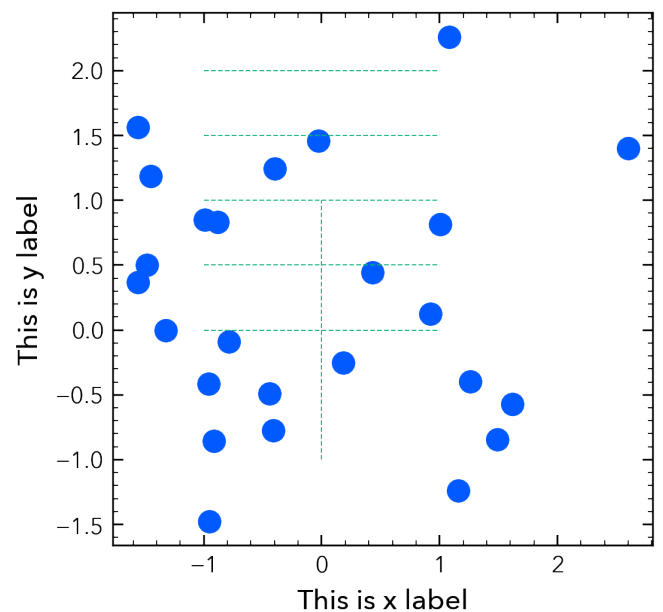
4 大量の線

一斉に同じ種類の線をプロットするには `mcoll.LineCollection` を使って、返り値を `ax.add_collection()` で加えると良い。

```

1 import matplotlib.collections as mcoll
2 from sugayutils import colors
3
4 ax = plot_fiducial()
5 segments = (
6     ((-1.0, 0.0), (1.0, 0.0)),
7     ((-1.0, 0.5), (1.0, 0.5)),
8     ((-1.0, 1.0), (1.0, 1.0)),
9     ((-1.0, 1.5), (1.0, 1.5)),
10    ((-1.0, 2.0), (1.0, 2.0)),
11    ((0.0, -1.0), (0.0, 1.0)),
12 )
13 linecollection = mcoll.LineCollection(segments,
14    ↪ colors=colors.green, lw=0.5, ls='--')
15 _ = ax.add_collection(linecollection)

```



5 グリッド分け

`fig.subplots()` と `fig.subplots_adjust()` の組み合わせでもグリッドを切れるが、`matplotlib.gridspec.GridSpec` を使うと引数 `width_ratios` などを使ってより柔軟なグリッドを作ることができる。`GridSpec` は `fig.subplots()` に引数として与えることもできるが、`fig.add_subplot()` で個別にパネルを作ることによって `projection` などを柔軟に対応させられる。

```

1 from matplotlib.gridspec import GridSpec
2 gs = GridSpec(1, 2, width_ratios=(1.5, 1))
3 fig = plt.figure(figsize=[7.2, 3.5])
4 ax0 = fig.add_subplot(gs[0])
5 ax1 = fig.add_subplot(gs[1], projection='3d')
6 ax0.scatter(data[0], data[1], c='blue')
7 ax0.set_xlabel('This is x label')
8 ax0.set_ylabel('This is y label')
9 _ = ax1.plot(np.arange(0., 10.0), np.arange(0.,
  ↳ 10.0), np.arange(0., 10.0))

```

