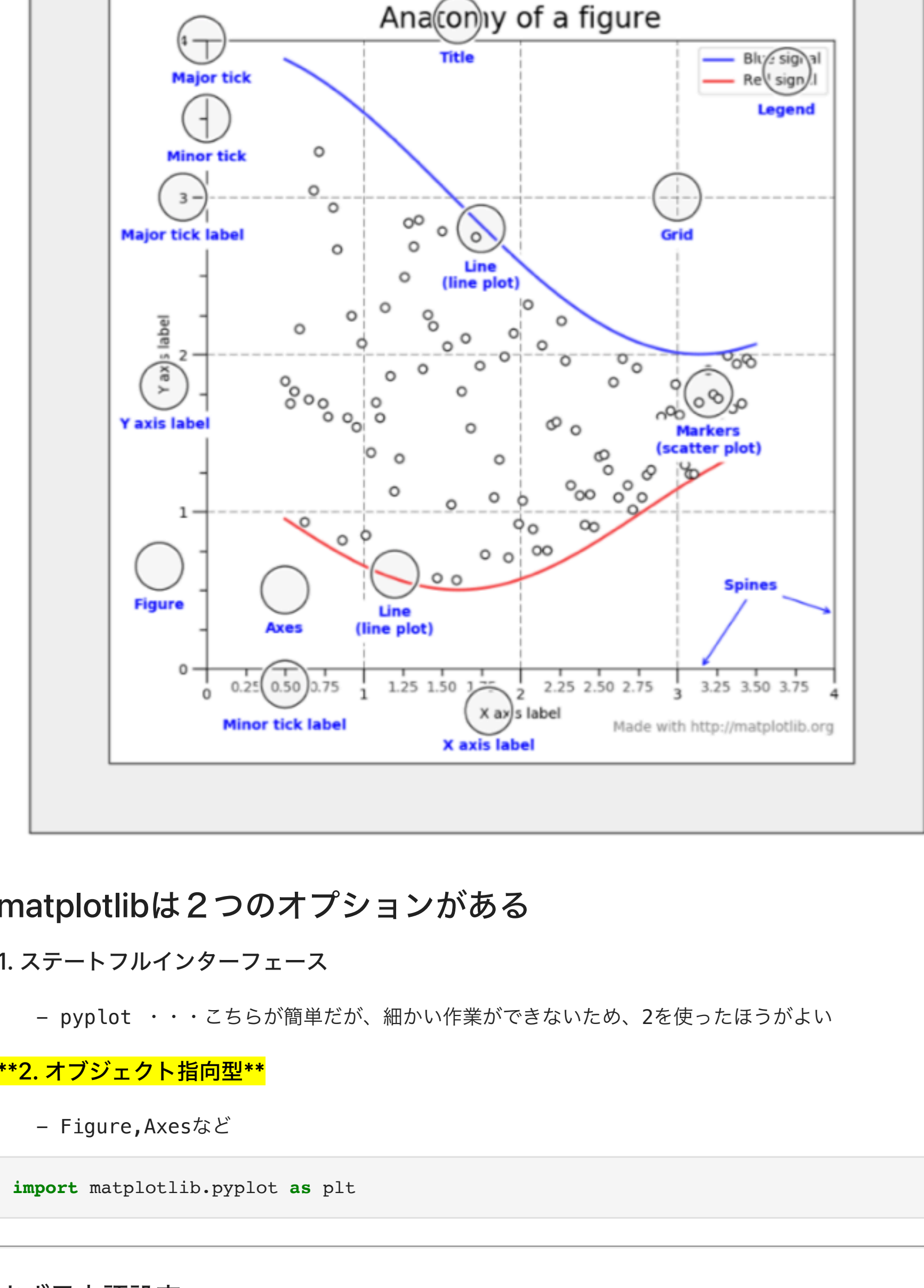


Chapter11 matplotlib,pandas,seabornによる可視化

- 可視化は探索的データ分析 (EDA) で重要
- EDAではデータをプロットすることで、外れ値や異常値の検出にも役立ち、新しい視点も見つかる、より深く理解もできる
- 基本的には「現在の自分の理解」を深めるために視覚化する。キレイなものをつくる必要はない
- ただレポートなど対外的な場合にはアプローチを変えなければならぬ
- 様々な可視化の手段からそのデータを一番よく表現できるものでなければならぬ
- 顧客を夢中にさせる映画同様、優れた可視化は興味をかき立てる情報を多数含んでいる

レシビ91 matplotlib入門 (P310)

パーツの名称



matplotlibは2つのオプションがある

1. ステートフルインターフェース

- plt.plot ... こちらが簡単だが、細かい作業ができないため、2を使ったほうがよい

2. オブジェクト指向型

- Figure, Axes など

```
In [2]: import matplotlib.pyplot as plt
```

まず日本語設定

```
In [3]: import matplotlib
```

```
In [4]: # 日本語フォントを設定
font_options = {'family': 'Hiragino Sans'}
plt.rc('font', **font_options)
```

以下はオブジェクト指向型の操作方法

fig, ax = plt.subplots()

- 戻り値は (Figureオブジェクト, Axesオブジェクト)
- グラフの描き方は数種類あるが、以下の記法が汎用性が高い

1. data.plot(kind=グラフ種類, ~)

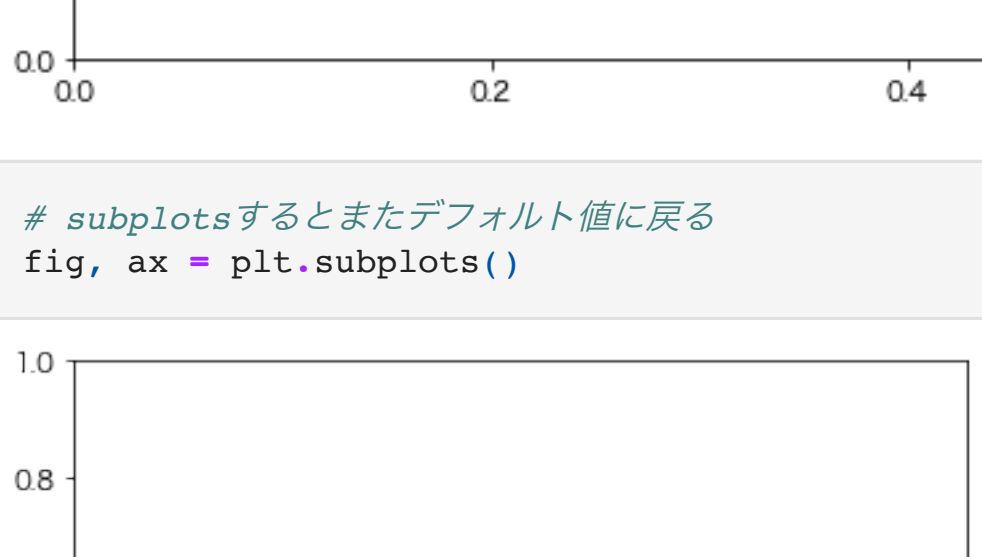
- (~) グラフにしたい data を plot する。グラフの種類は~

2. ax.plot(data, ~)

- (~) ax に plot する。(data を~)

まずfig, axを作る

```
In [5]: fig, ax = plt.subplots()
# その時のfigのサイズは画面デフォルトサイズに戻る
```



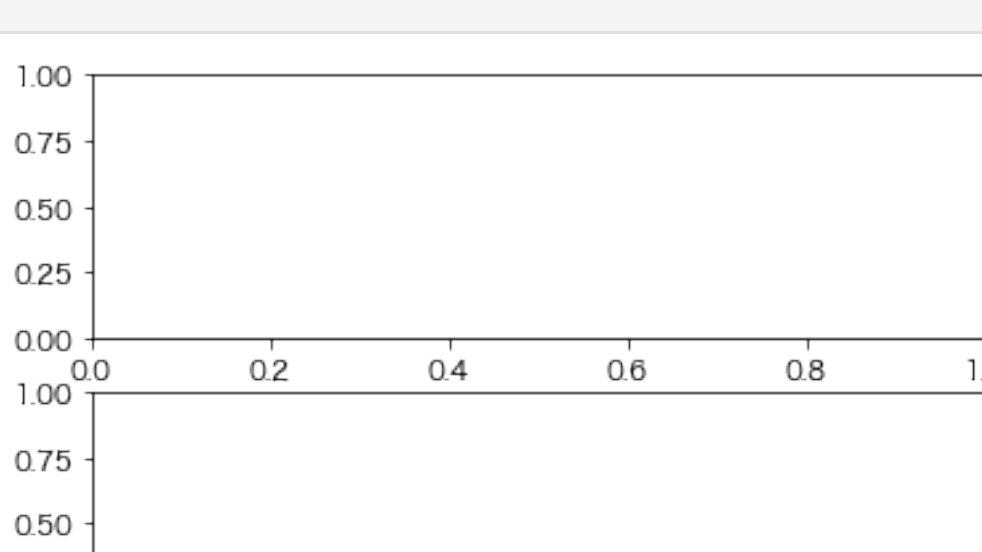
```
In [6]: # そのデフォルト値は以下
fig.get_size_inches()
```

```
Out[6]: array([6., 4.])
```

```
In [7]: # figのサイズを変更する
fig.set_size_inches(14, 4)
fig
```

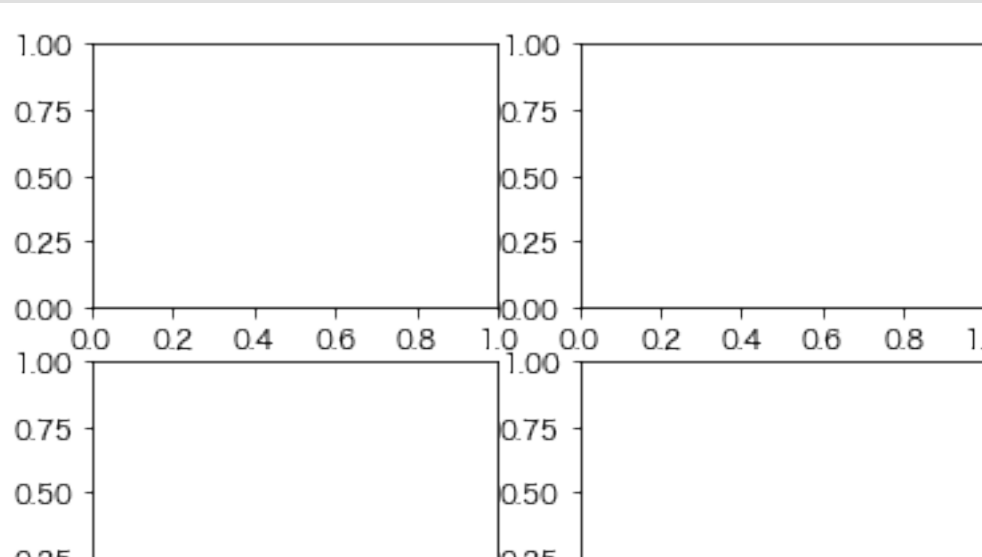


```
In [8]: # subplotsするとまたデフォルト値に戻る
fig, ax = plt.subplots()
```

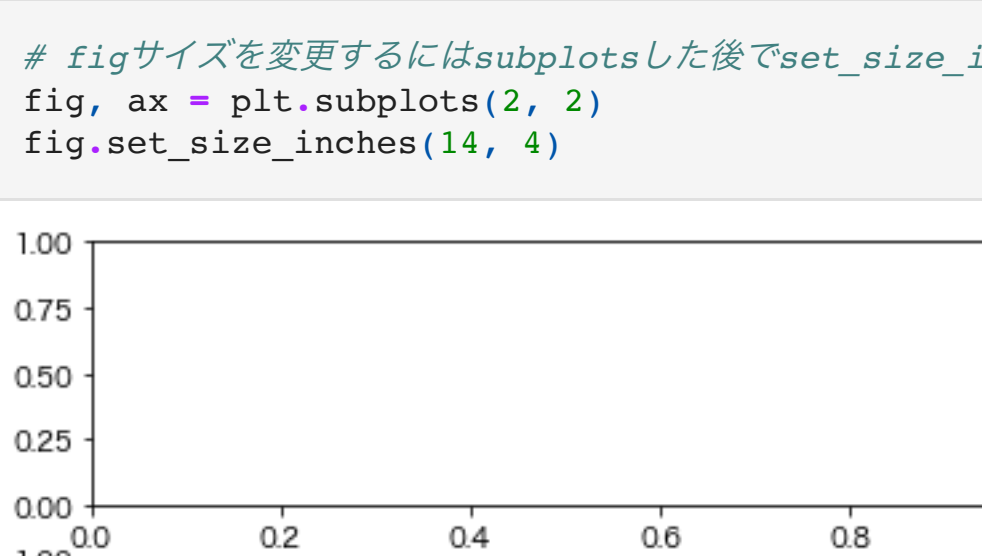


figを分割する

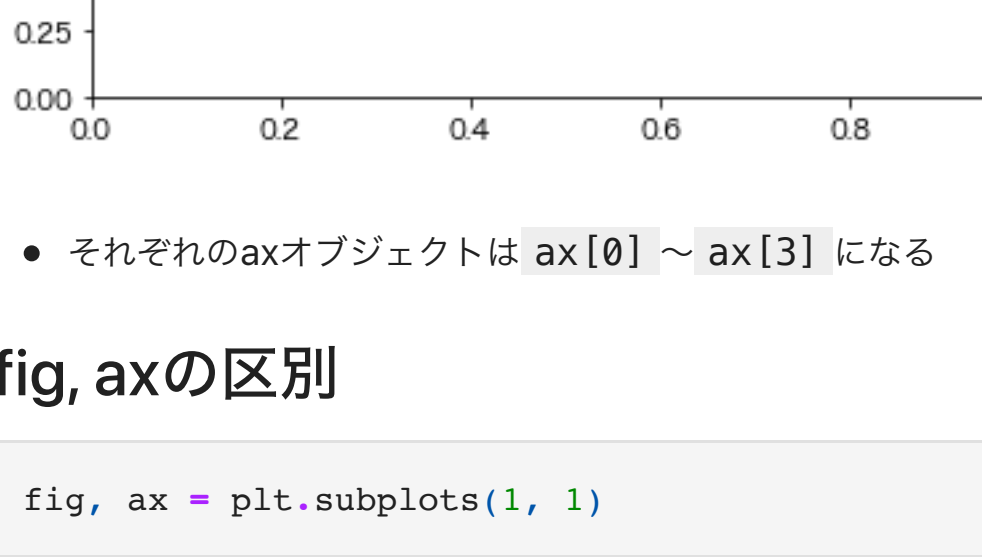
```
In [9]: fig, ax = plt.subplots(nrows=1, ncols=1)
```



```
In [10]: fig, ax = plt.subplots(2, 1)
```



```
In [11]: fig, ax = plt.subplots(2, 2)
```



```
In [12]: # figサイズを変更するにはsubplotsした後でset_size_inches()を使う
fig, ax = plt.subplots(2, 2)
fig.set_size_inches(14, 4)
```



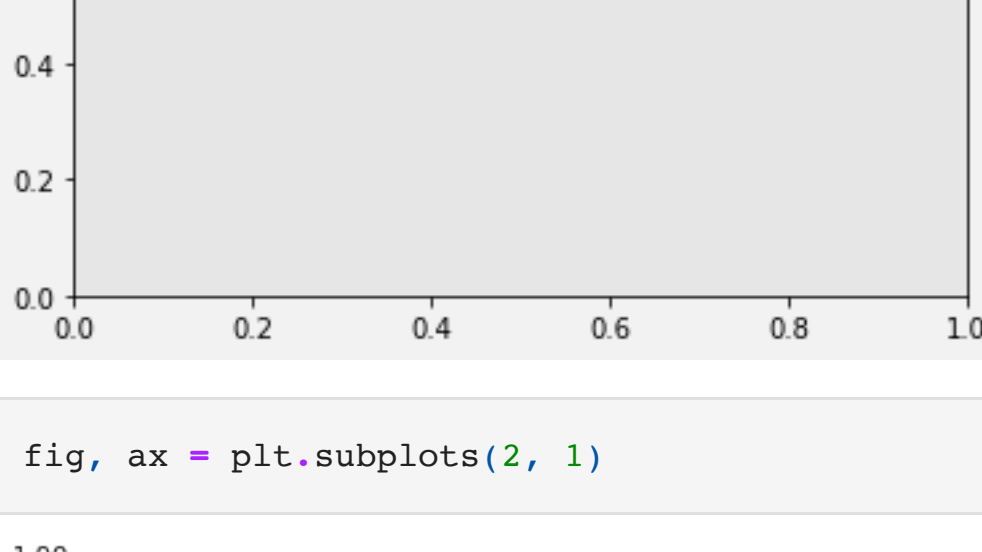
- それぞれのaxオブジェクトはax[0] ~ ax[3]になる

fig, axの区別

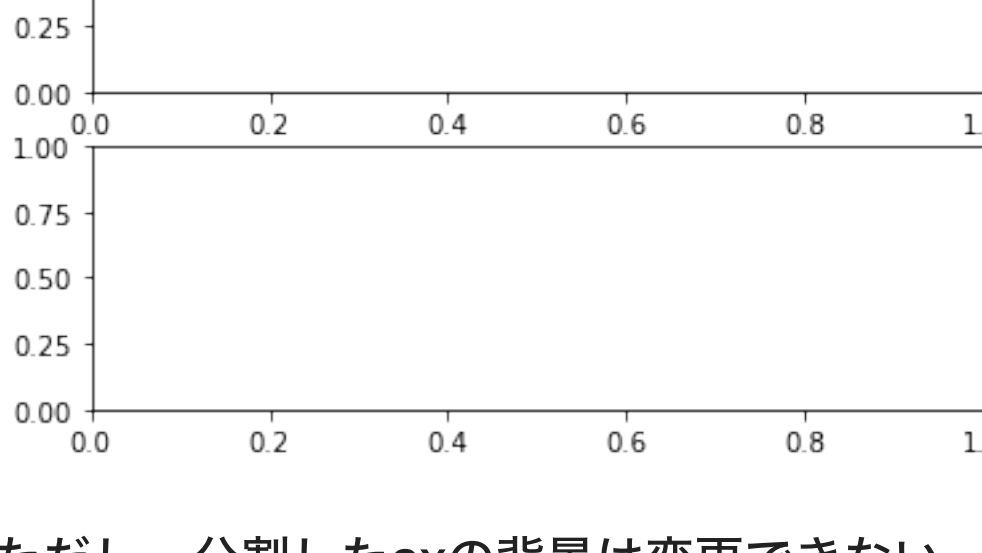
```
In [71]: fig, ax = plt.subplots(1, 1)
```



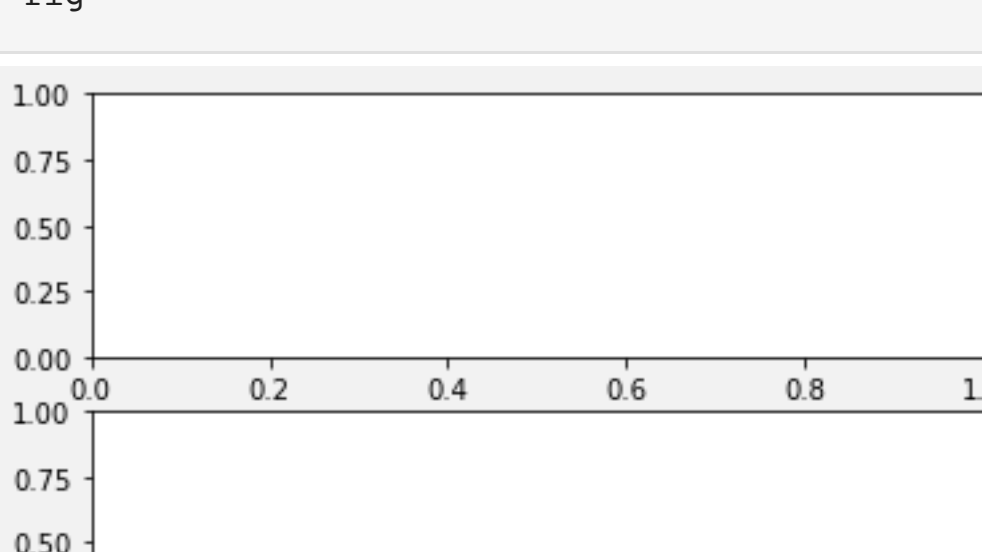
```
In [72]: fig.set_facecolor('.95')
fig
```



```
In [73]: ax.set_facecolor('.9')
fig
```

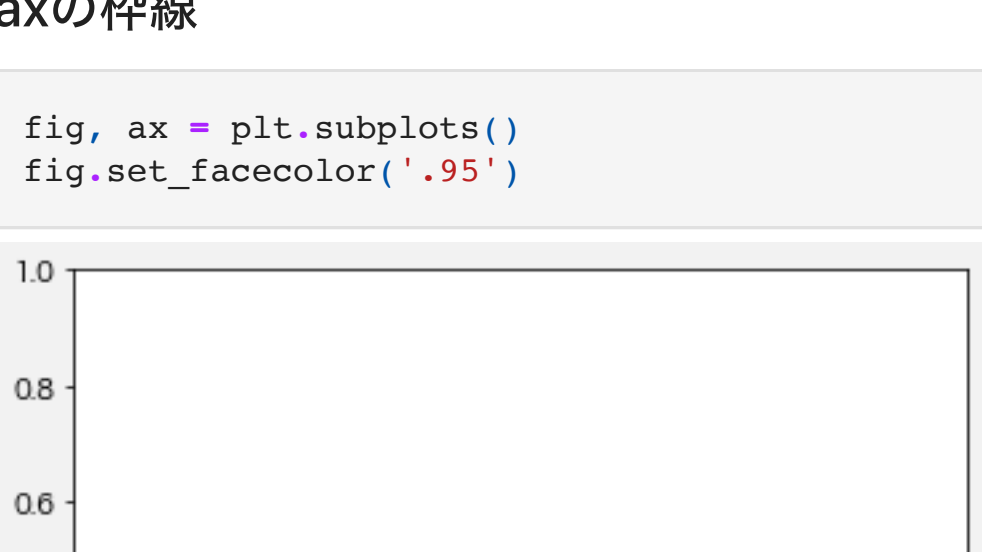


```
In [74]: fig, ax = plt.subplots(2, 1)
```



ただし、分割したaxの背景は変更できないっぽい (エラー)

```
In [75]: fig.set_facecolor('.95')
fig
```

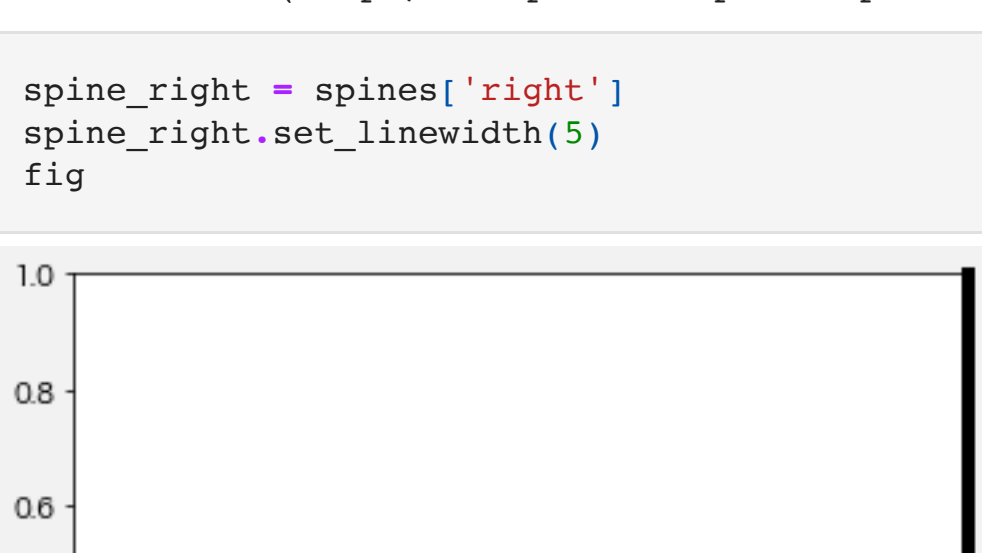


```
In [76]: ax.set_facecolor('.9')
fig
```

```
AttributeError                                Traceback (most recent call last)
<ipython-input-76-58fdeaf881f> in <module>
----> 1 ax.set_facecolor('.9')
      2 fig
AttributeError: 'numpy.ndarray' object has no attribute 'set_facecolor'
```

axの枠線

```
In [143]: fig, ax = plt.subplots()
fig.set_facecolor('.95')
```



```
In [144]: # axのspinesを取得すると4つある
spines = ax.spines
spines
```

```
Out[144]: OrderedDict([('left', <matplotlib.spines.Spine at 0x7fd046733dd0>),
('right', <matplotlib.spines.Spine at 0x7fd04685f3b0>),
('bottom', <matplotlib.spines.Spine at 0x7fd04685f4c0>),
('top', <matplotlib.spines.Spine at 0x7fd04685f560>)])
```

```
In [145]: spine_right = spines['right']
spine_right.set_linewidth(5)
fig
```



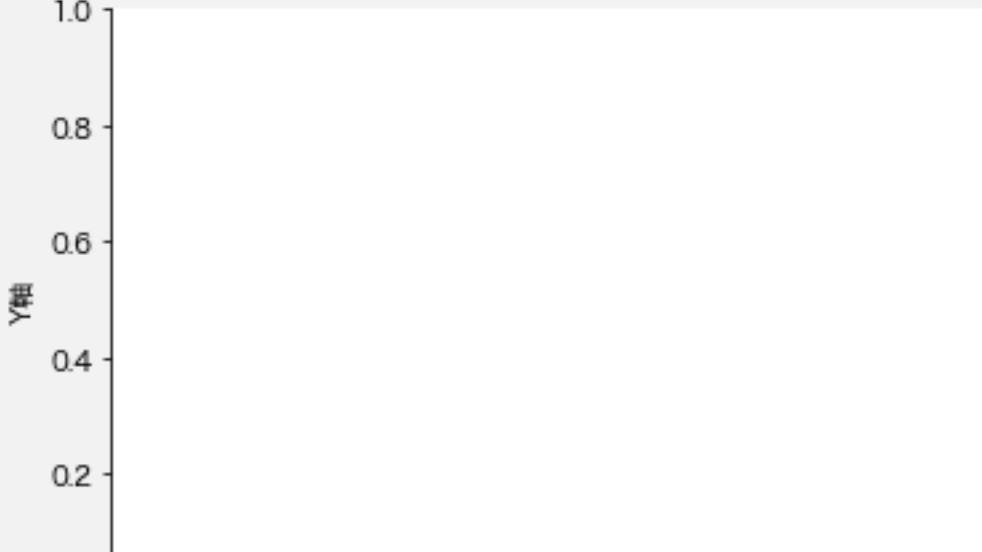
axの上・右の枠線を消す

```
In [146]: # top・rightの枠線が消したい場合
spines['top'].set_visible(False) # または'None'
spines['right'].set_visible(False)
fig
```

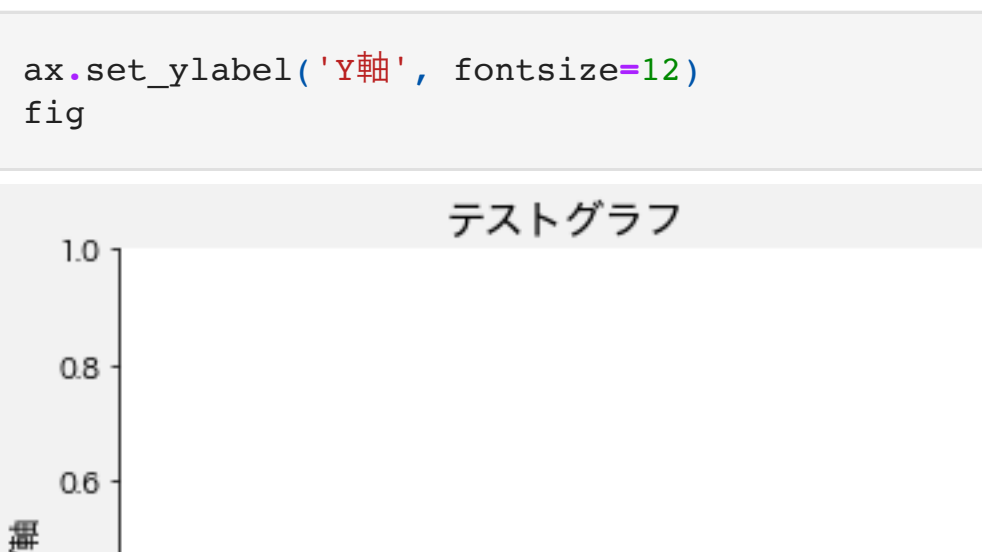


タイトルと軸のタイトル

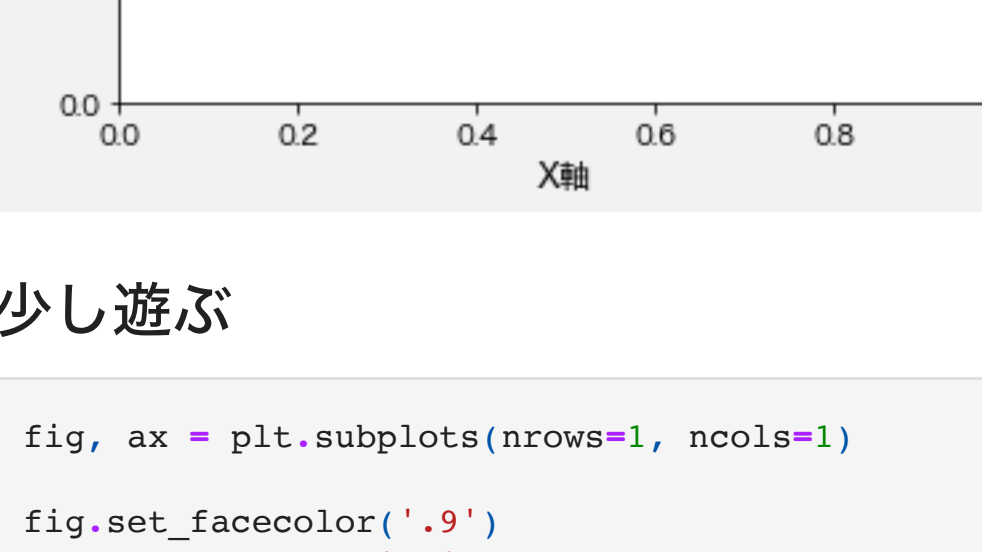
```
In [147]: ax.set_title('テストグラフ', fontsize=15)
fig
```



```
In [150]: ax.set_xlabel('X軸', fontsize=12)
fig
```



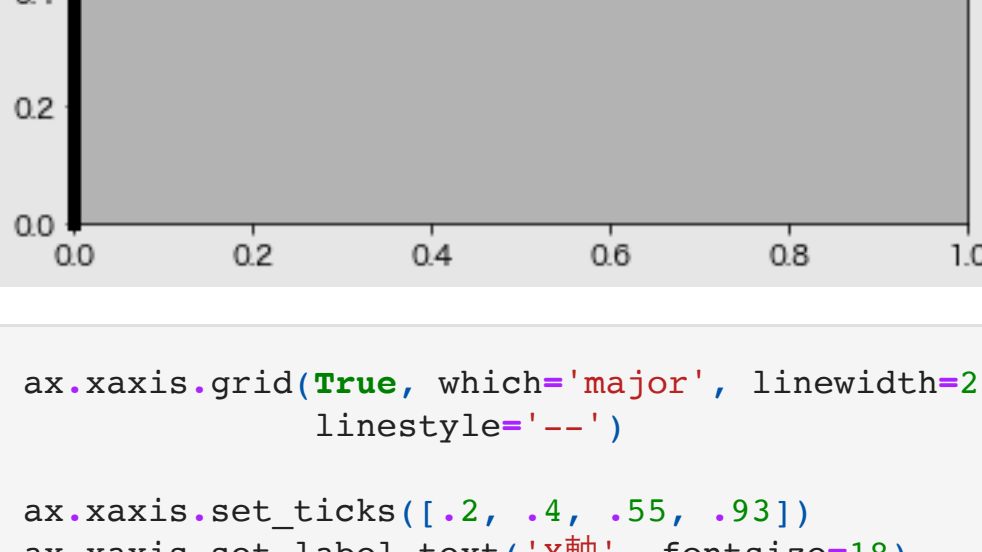
```
In [151]: ax.set_ylabel('Y軸', fontsize=12)
fig
```



少し遊ぶ

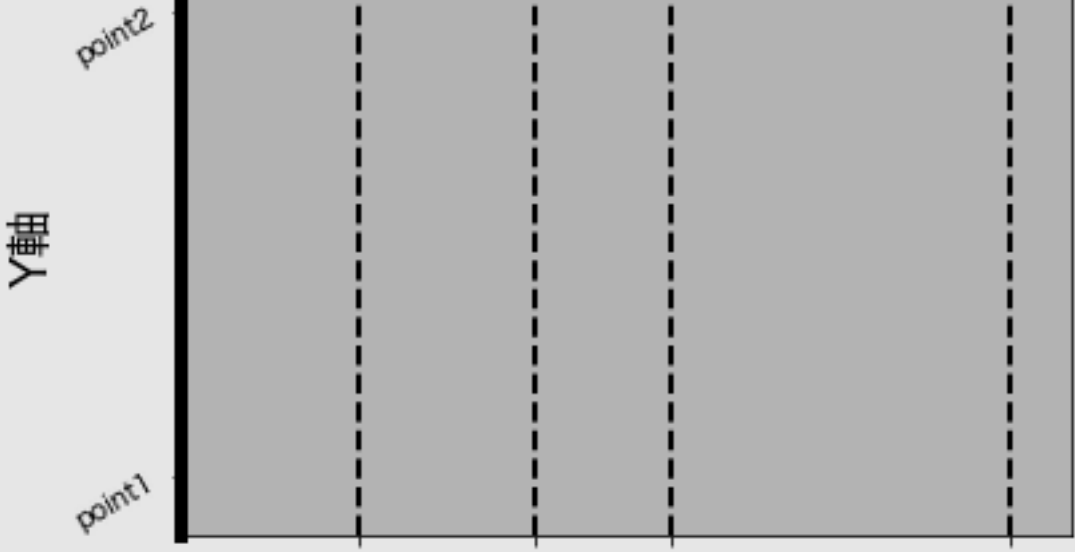
```
In [32]: fig, ax = plt.subplots(nrows=1, ncols=1)
```

```
fig.set_facecolor('.9')
ax.set_facecolor('.7')
```



```
In [33]: ax.xaxis.grid(True, which='major', linewidth=2, color='black',
                    linestyle='--')

ax.xaxis.set_ticks([.2, .4, .55, .93])
ax.xaxis.set_label_text('X軸', fontsize=18)
ax.set_ylabel('Y軸', fontsize=18)
ax.set_yticks([.1, .9])
ax.set_yticklabels(['point1', 'point2'], rotation=30)
fig
```



プロットオブジェクトの属性 (プロパティ) を確認する

- axisオブジェクトの特性を見ると、すべての属性と現在の設定値がわかる

```
In [34]: ax.xaxis.properties()
```

```
Out[34]: {'agg_filter': None,
'alpha': None,
'animated': False,
'children': (Text(0.5, 25.200000000000003, 'X軸'),
Text(1, 26.200000000000003, ''),
<matplotlib.axis.XTick at 0x7ff49362c700>,
<matplotlib.axis.XTick at 0x7ff49362cd30>,
<matplotlib.axis.XTick at 0x7ff49362cd30>,
<matplotlib.axis.XTick at 0x7ff4936b7a00>),
'clip_box': <matplotlib.transforms.TransformedsBox at 0x7ff493632d60>,
'clip_on': True,
'clip_path': None,
'contains': None,
'create_figure': True,
'data_interval': array([ inf, -inf]),
'figure': <Figure size 432x288 with 1 Axes>,
'gid': None,
'gridlines': <a list of 4 Line2D ticklines objects>,
'in_layout': True,
'inverted': False,
'label': Text(0.5, 25.200000000000003, 'X軸'),
'label_position': 'bottom',
'label_text': 'X軸',
'major_formatter': <matplotlib.ticker.ScalarFormatter at 0x7ff49362c880>,
'major_locator': <matplotlib.ticker.FixedLocator at 0x7ff4936c5d00>,
'major_ticks': (<matplotlib.axis.XTick at 0x7ff49362cd30>,
<matplotlib.axis.XTick at 0x7ff49362cd30>,
<matplotlib.axis.XTick at 0x7ff49362cd30>,
<matplotlib.axis.XTick at 0x7ff4936b7a00>),
'minor_locator': <matplotlib.ticker.NullLocator at 0x7ff49361ef00>,
'minor_ticks': [],
'minorticklabels': [],
'minorticklocs': <a list of 0 Line2D ticklines objects>,
'mpines': inf,
'offset_text': Text(1, 26.200000000000003, ''),
'path_effects': [],
'picker': None,
'pickradius': 15,
'pickrised': None,
'remove_overlapping_locs': True,
'scale': 'linear',
'setpath': None,
'smart_bounds': False,
'snap': None,
'tick_padding': 3.5,
'tick_space': 1,
'ticklabels': (Text(0.2, 0, '0.20'),
Text(0.4, 0, '0.40'),
Text(0.55, 0, '0.55'),
Text(0.93, 0, '0.93')),
'ticklines': <a list of 4 Line2D ticklines objects>,
'ticks': array([0.2, 0.4, 0.55, 0.93]),
'ticklocs': array([0.2, 0.4, 0.55, 0.93]),
'ticks_direction': array(['out', 'out', 'out', 'out'], dtype='<U3>'),
'ticks_position': 'bottom',
'transform': <matplotlib.transforms.IdentityTransform at 0x7ff49360b9a0>,
'transformed_clip_path_and_offine': (None, None),
'units': None,
'update': None,
'visible': True,
'zorder': 1.5}
```