

Numpy編 ～インポート (import numpy as np) を忘れずに～

配列の生成

<code>np.array([a1, a2, a3, ...])</code>	リストから, Numpy 配列を作成します.
<code>np.array([[a1, a2, a3], [b1, b2, b3]])</code>	2次元以上の配列も作ることができます.
<code>In: np.arange(5, 30, 5)</code> <code>Out: np.array([5, 10, 15, 20, 25])</code>	5から始まり, 30未満で間隔が5の等差数列を生成します. 30は含まれない ことに注意.
<code>In: np.arange(6)</code> <code>Out: np.array([0, 1, 2, 3, 4, 5])</code>	引数を省略すると, 間隔1で0始まりの等差数列を生成します.
<code>In: np.linspace(0, 10, 5)</code> <code>Out: np.array([0, 2, 4, 6, 8, 10])</code>	0以上10以下の数直線を5等分します. (10も含む ので要素数は6個になります.)
<code>np.append(array, [a1, a2, a3, ...])</code>	配列に要素を追加します.
<code>np.random.rand(N)</code>	N個の乱数を生成できます.

形状変換

<code>np.flatten()</code>	多次元配列を一次元に直します.
<code>np.transpose()</code>	配列を転置 (行, 列の入れ替え) します.
<code>In: a = np.array([[1, 2, 3], [4, 5, 6]])</code> <code>a.shape</code> <code>Out: (2, 3)</code>	配列の形状を確認できます. ()をつける必要はありません.
<code>In: np.array([1, 2, 3, 4, 5, 6]).reshape(2, 3)</code> <code>Out: np.array([[1, 2, 3], [4, 5, 6]])</code>	配列の形状を変換します.

演算 (1) ～四則演算～

<code>array1 + array2, array + 5, array - 2.2</code>	配列同士の加減算は簡単です. 数字を足す (引く) と配列全ての要素に対して計算されます.
<code>array1 * array2, array * 5</code>	掛け算 (<code>*</code>) や
<code>array1 / array2, array / 5, 1 / array</code>	割り算 (<code>/</code>) についても同様にできます.

演算 (2) ～応用編～

<code>array.max(), array.min(), array.sum()</code>	最大値, 最小値, 合計値を計算できます.
<code>array.mean(), array.median()</code>	平均値や中央値も計算できます.
<code>np.dot(A, B)</code>	行列積やベクトルの内積を計算します.

判定

np.all(array > 20)	全ての要素が () 内の条件を満たすと True
np.any(array > 20)	1つでも () 内の条件を満たすと True
np.where(array > 20)	条件を満たす要素の index を配列で返す

Matplotlib編 ～インポート (import matplotlib.pyplot as plt) を忘れずに～

1. グラフの種類を選択

折れ線グラフ	plt.plot(x, y, color=色, linewidth=線の太さ, linestyle=線のスタイル)
棒グラフ	plt.bar(x軸上の数値, height=データ, width=太さ, bottom=余白の高さ)
散布図	plt.scatter(x, y, c=色, s=点のサイズ, alpha=透明度)
円グラフ	plt.pie(数値, labels=ラベル, colors=色)

2-1. グラフの属性に関する指定

plt.title('グラフのタイトル')	タイトルを指定します。
plt.figure(figsize=(縦の長さ, 横の長さ))	グラフのサイズ (inch) を指定します。
plt.legend(['凡例名1', '凡例名2', ...], loc='upper right')	リスト型で凡例を指定します。 locを用いて、位置も指定できます。
plt.xlabel('x軸のラベル名') plt.ylabel('y軸のラベル名', fontsize=20)	グラフの軸ラベルを指定します。 フォントの大きさも指定できます。
plt.text(x, y, 表示する数 or 文字の配列, ha='center', va='bottom')	データラベルを挿入できます。x, yで入れる位置を指定し、表示したい数や文字を配列で与えます。haで水平位置, vaで垂直位置を細かく設定できます。

2-2. グラフの目盛に関する指定

plt.xlim([min, max]), plt.ylim([min, max])	x軸, y軸の表示範囲を指定します。
plt.grid(True)	グリッドを挿入します。
plt.xscale('log')	対数目盛で表示します。
plt.xticks(rotation=Ω)	軸目盛をΩ度回転させます。

3. 描画

グラフの種類を選択し、設定をいじった後、最後に plt.show() で描画できます！