科学計算研究室 Python ゼミ

～ 可視化ツール ～

2021-02-24 福田　浩

1　matplotlib

1.1　各種グラフ

matplotlib をインポートして plot("x データ","y データ") でデータを指定し，show() で表示する．plot(x,y) だと折れ線グラフになるが，scatter(x,y) だと散布図，bar(x,y) だと棒グラフ，pie(x,y) だと円グラフになる．

```
import numpy as np                  # デモデータの作成に便利
import matplotlib.pyplot as plt

x = np.linspace(0,10,11)            # 0 から 1 まで，11 個の刻み
y = x ** 2
#plt.plot(x,y)                      # 折れ線グラフ
#plt.scatter(x,y)                   # 散布図
#plt.bar(x,y)                       # 棒グラフ
#plt.pie(y)                         # 円グラフ
plt.show()
```

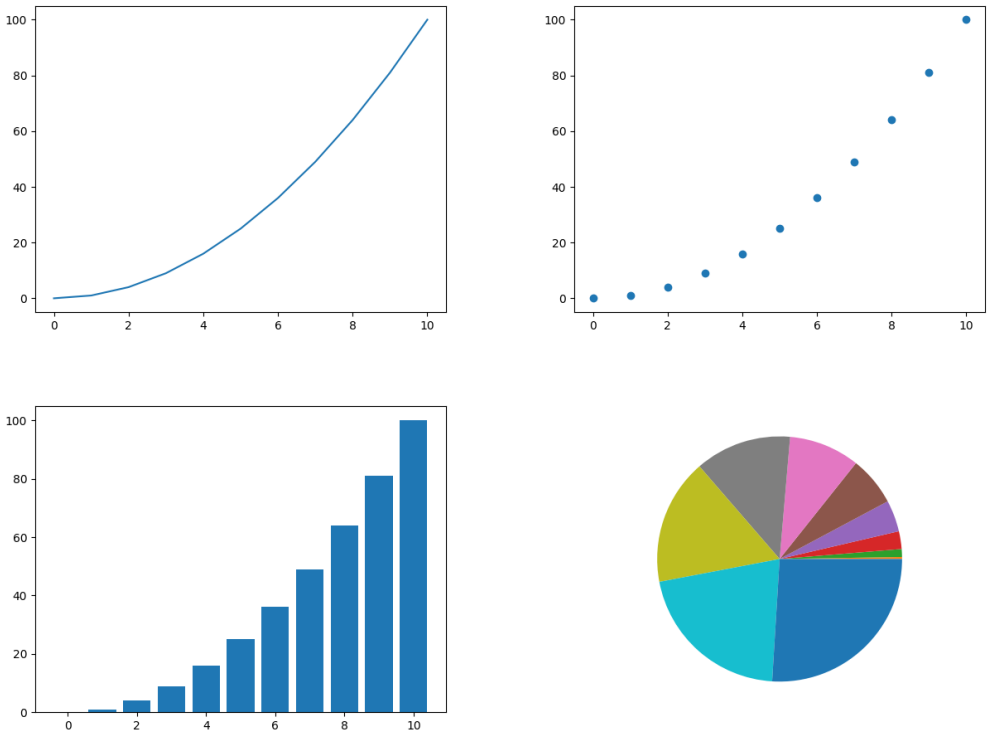

Figure 1: 折れ線グラフ，散布図，棒グラフ，円グラフの例

1.2　散布図のカスタマイズ

散布図の例で，マーカー形状，マーカーサイズ，色，線の太さなどを変更してみる．

```
import numpy as np # デモデータの作成に便利
import matplotlib.pyplot as plt

x = np.linspace(0,10,11) # 0から1まで，11個の刻み
y = x ** 2
#plt.scatter(x, y)                          # 散布図
#plt.scatter(x, y, marker='D')              # マーカー(Diamond)変更
#plt.scatter(x, y, s=10)                    # サイズ変更
#plt.scatter(x, y, c='red')                 # 色変更
#plt.scatter(x, y, linewidth=10)            # 線の太さ変更
#plt.scatter(x, y, edgecolors='yellow')     # 線の色変更
                                            # 全部のせ
plt.scatter(x, y, marker='D', s=30, c='red', linewidth=2, edgecolors='yellow')
plt.show()
```

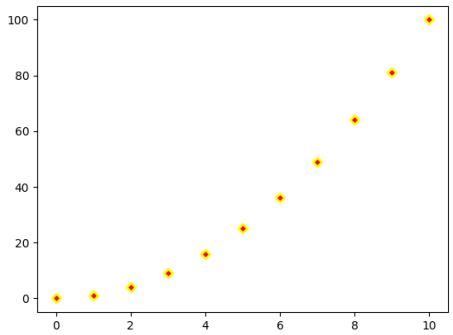

Figure 2: 散布図カスタマイズの例

1.3　軸などの調整

グラフの表示範囲を設定するには `xlim("最小値","最大値")` と `ylim("最小値","最大値")` を使う．罫線を表示したいときは `grid()` を使う．対数軸を指定するには `xscale('log')` と `yscale('log')` を使う．

```
import numpy as np # デモデータの作成に便利
import matplotlib.pyplot as plt

x = np.linspace(0,100,101) # 0 から 1 まで，11 個の刻み
y = x ** 2
plt.scatter(x, y)  # 散布図
plt.xlim(1, 100)   # x 軸の範囲設定
plt.ylim(1, 10000) # y 軸の範囲設定
plt.grid()         # 罫線
plt.xscale('log')  # x 軸対数
plt.yscale('log')  # y 軸対数
plt.show()
```

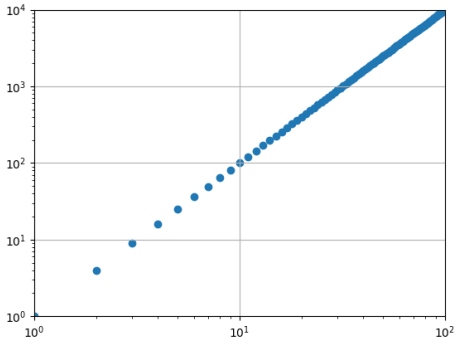

Figure 3: 軸調整の例

1.4　複数プロット

複数のプロットを重ね書きすることも可能.

```
import numpy as np # デモデータの作成に便利
import matplotlib.pyplot as plt

x = np.linspace(0,100,101) # 0 から 1 まで, 11 個の刻み
y = x ** 2
z = x ** 3
w = x ** 3 - 10 * x ** 2
plt.scatter(x, y)  # 1 つめのグラフ
plt.scatter(x, z)  # 2 つめのグラフ
plt.scatter(x, w)  # 3 つめのグラフ
plt.xlim(1,100)
plt.ylim(1,10000)
plt.xscale('log')
plt.yscale('log')
plt.show()
```

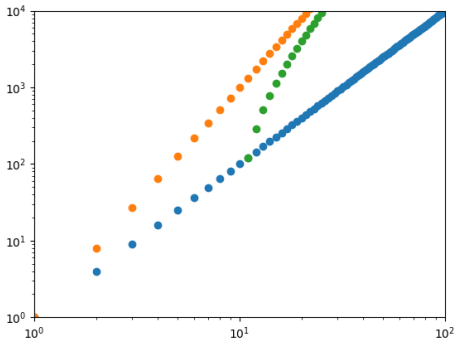

Figure 4: 複数プロットの例

1.5　複数グラフ

複数のグラフを整列して表示することも可能.

```
import numpy as np # デモデータの作成に便利
import matplotlib.pyplot as plt

x = np.linspace(0,100,101) # 0 から 1 まで, 11 個の刻み
y = x ** 2
z = x ** 3
w = x ** 3 - 10 * x ** 2

plt.subplot(2,2,1)
plt.scatter(x, y)  # 1 つめのグラフ
plt.xlim(1,100)
plt.ylim(1,10000)
plt.xscale('log')
plt.yscale('log')

plt.subplot(2,2,2)
plt.scatter(x, z)  # 2 つめのグラフ
plt.xlim(1,100)
plt.ylim(1,10000)
plt.xscale('log')
plt.yscale('log')

plt.subplot(2,2,3)
plt.scatter(x, w)  # 3 つめのグラフ
plt.xlim(1,100)
plt.ylim(1,10000)
plt.xscale('log')
plt.yscale('log')

plt.show()
```

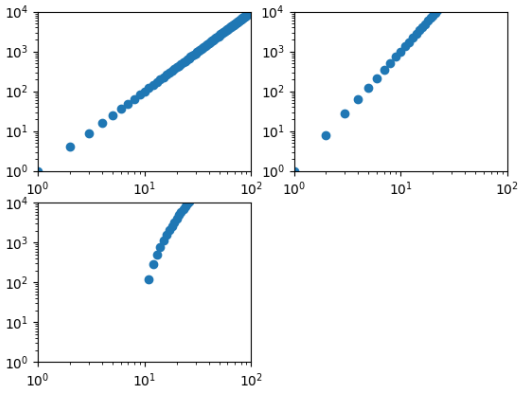

Figure 5: 複数グラフの例

2　田口がやりたかったこと (?)

前回のゼミの中で田口がやりたかったことを想像しながらやってみた.

```
import numpy as np
import math
import matplotlib.pyplot as plt

x = np.linspace(-1,1,101)
y = np.linspace(-1,1,101)
ans = 0.16890435790992342
for i in range(101):
    y[i] = x[i]**2 - 5*x[i] + 2 - math.exp(x[i])
plt.subplot(2,1,1)
plt.plot(x,y)
plt.scatter(ans,0, marker='*', s=300, color='yellow', edgecolor='red')
plt.grid()
plt.subplot(2,1,2)
plt.plot(x,y)
plt.scatter(ans,0, marker='*', s=300, color='yellow', edgecolor='red')
plt.xlim(0.167,0.17)
plt.ylim(-0.01,0.01)
plt.grid()
plt.show()
```

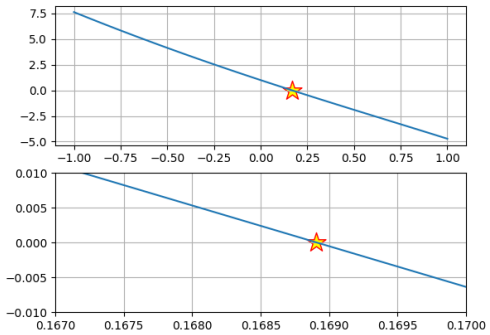

Figure 6: 田口がやりたかったこと???

3　その他の可視化ツール

3.1　gnuplot

Linux 環境でよく使われる可視化ツール．使い勝手は subplot など matplotlib と似ているが，グラフに特化した環境なので，カスタマイズのバリエーションが豊富で，表示速度も高速．卒論執筆などで大量のグラフを生成する場合は，gnuplot を使うことを推奨する．

3.2　matlab(行列計算が得意なアプリケーション)

科技大はライセンス契約しているので，誰でも利用可能 (のはず)．大規模行列計算が出来るアプリケーションだが，グラフ描画も得意．matplotlib の mat は matlab に由来しているらしい．