

Python の動作確認

目 次

1. 表紙	
2. 目次	
3. Python の動作確認	
3.1 コマンドライン環境上の Python 起動	3
3.2 グラフィカル環境 IDLE 上の Python 起動	5
4. NumPy の動作確認	
5. matplotlib の動作確認	
5.1 グラフ描画の動作確認	7
5.2 少し複雑なグラフ描画	7
5.3 ファイル読み込みしたデータをグラフ化する	9
6. Scipy の動作確認	
7. Sympy の動作確認	
7.1 Sympy の動作確認	12
7.2 Sympy での数式処理	12
8. scikit-learn の動作確認	
9. 参考資料	

3 Python の動作確認

Python が正しくインストールされていることを確認する。OS のバージョンなどにより、多少、操作法が画面出力が違う場合があるので注意。

3.1 コマンドライン環境上の Python 起動

まず、コマンドライン環境上の動作確認を行う。

PC 画面左下の Windows アイコンのボタンを押して、プログラム検索の場所に「python」と打ち込むと、プログラム「python」が見つかる。

注：見つからなかった場合は、探す。おそらく、Python をインストールしたフォルダ

```
C:\Anaconda3
```

に入っている。どうしても見つからない場合は、ファイル検索で「python」を検索する。

見つかったプログラム「python」を実行すると、下の「黒い画面」が表示される。これは、コマンドライン環境上で python が実行された状態である。

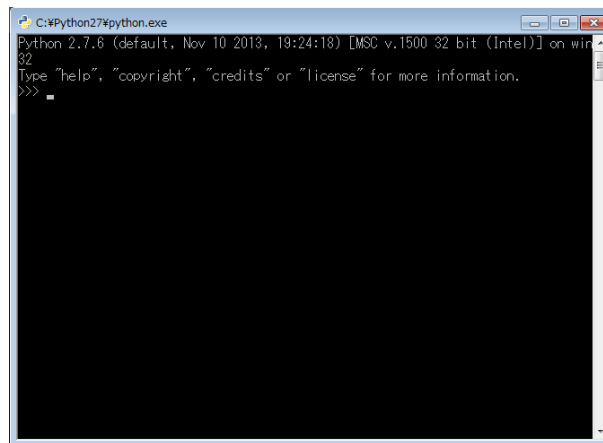


図 1: Python コマンドライン起動画面

その中に以下の内容を打ち込み、「Enter」キーを押す。

```
>>> print("Hello")
```

次の行に Hello と出力されたら Python が正常にインストールされ動かせることの確認ができた。

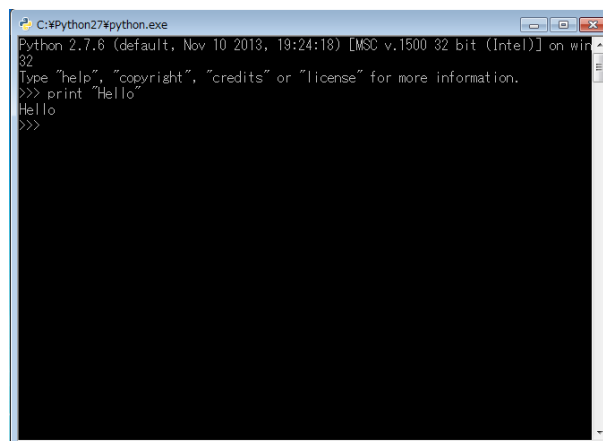


図 2: 内容を打ち込み、「Enter」キーを押した後の画面

上記のように表示されれば、動作確認は完了である．終了するために

```
>>> exit()
```

を入力した後「Enter」を押す．

3.2 グラフィカル環境 IDLE 上の Python 起動

次に、IDLE (アイドル) というグラフィカル環境での動作確認をする。前節と同様に、PC 画面左下の Windows アイコンのボタンを押して、プログラム検索の場所に「idle」と打ち込むと、「idle」というプログラムが見つかる。

注：見つからなかった場合は、探す。おそらく、Python をインストールしたフォルダの下の

```
C:\Anaconda3\Script
```

に入っている。どうしても見つからない場合は、ファイル検索で「python」を検索する。

見つかったプログラム「idle」を実行すると、下の「白い画面」が表示される。これは、グラフィカル環境 IDLE 上で python が実行された状態である。

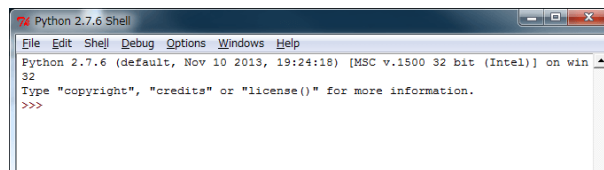


図 3: IDLE 起動画面

左上にある「File」から「New File」を選択すると、新しいウインドウが表示される。このウインドウ内に以下の内容を打ち込み、Python が保存されているフォルダにファイル名を「sample_1.py」とし、保存する。

```
print("Hello")
```

保存するには「File」から「Save」もしくは「Save as ...」を選択する。

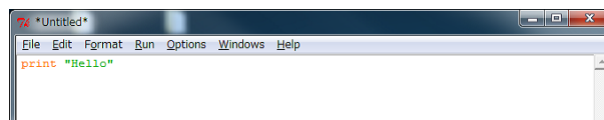


図 4: 新しいウインドウに、内容を打ち込んだ後の画面

次に「Run」から「Run Module」を選択すると、IDLE 画面に「Hello」と表示されれば、動作確認は完了である。

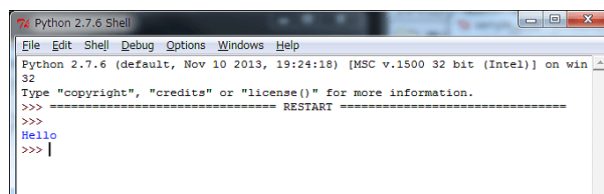


図 5: 実行後の画面

4 NumPy の動作確認

数値計算モジュール NumPy³ が正しくインストールされていることを確認する。

まず、グラフィカル環境 IDLE 上で Python を起動する (Sec. 3.2 を参照)。次に、以下のサンプルコードを一行ずつ上から順に打ち込む。これは、円周率 π を出力するプログラムである。

```
>>> import numpy
>>> numpy.pi
```

下のように円周率が出力されれば動作確認は完了である。

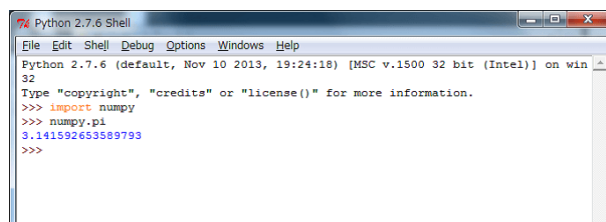


図 6: 動作確認画面

³ NumPy(公式):<http://www.numpy.org/>

5 matplotlibの動作確認

5.1 グラフ描画の動作確認

グラフ描画モジュール matplotlib⁴ が正しくインストールされていることを確認する。

まず、グラフィカル環境 IDLE 上で Python を起動する (Sec. 3.2 を参照)。次に、左上にある「File」から「New File」を選択して新しいウインドウを表示する。このウインドウ内に以下のサンプルコードを打ち込み、Python が保存されているフォルダにファイル名を「sample_test.py」とし、保存する。

```
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(-3,3,0.1)
y= np.sin(x)
plt.plot(x,y)
plt.show()
```

保存後、「Run」から「Run Module」を選択すると、プログラムが実行される。下のように、三角関数が表示されれば、動作確認は完了である。

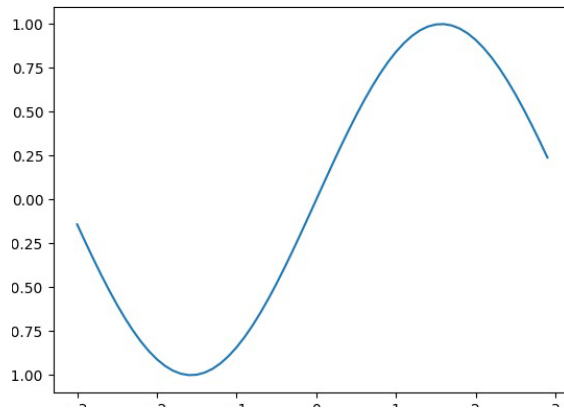


図 7: サンプルコード (三角関数のプロット) 実行画面

5.2 少し複雑なグラフ描画

次に、以下のもう少し長いプログラムを同様に実行する。以下のサンプルコードを打ち込み、Python が保存されているフォルダにファイル名を「sample.2.py」とし、保存する。

```
from pylab import *
subplot(111, axisbg='darkslategray')
#subplot(111, axisbg='#ababab')
t = arange(0.0, 2.0, 0.01)
s = sin(2*pi*t)
plot(t, s, 'r')
xlabel('time (s)', color='r')
ylabel('voltage (mV)', color='0.5') # grayscale color
title('About as silly as it gets, folks', color='#afeeee')
show()
```

⁴ matplotlib(公式) <http://matplotlib.org/>

注：サポートページ [1] には sample_2.py の実物があるので、それをダウンロードしても良い．保存後、sample_2.py ファイルの「Run」から「Run Module」を選択すると、プログラムが実行される．下のような画面が表示されれば、動作確認は完了である．

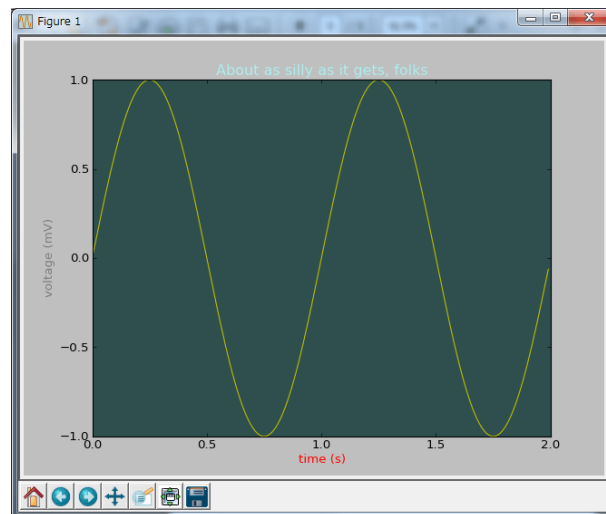


図 8: サンプルコード実行画面

注：ギャラリーページ⁵ には、多彩なグラフ描画例がある．

⁵ <http://matplotlib.org/gallery.html>

5.3 ファイル読み込みしたデータをグラフ化する

卒業論文などでは、自分で作ったデータ (実験・シミュレーション) をグラフにすることが多い。そのような場合の使い方を紹介する。

入力するデータファイルとして、

```
1 1.1
2 1.2
3 1.6
```

を考える。1 列めを x 軸に、2 列めを y 軸にしてグラフ化することにする。このデータファイルは、サポートページ [1] には plot_graph_data.txt というファイル名で用意されている。

下記のプログラムを実行すれば良い。ただし、同じフォルダに、上記データファイルが存在している必要がある。

```
import matplotlib.pyplot as plt
x=[]
y=[]
fp = open('plot_graph_data.txt','r')
for i, line in enumerate(fp):
    data=line.split()
    x.append(data[0])
    y.append(data[1])
plt.plot(x,y, 'ro') # red circle
plt.title('My graph')
plt.xlabel('time (s)')
plt.ylabel('hight (m)')
plt.text(1.2, 1.2, 'Hello Hello', rotation=45)
plt.xlim(0.0, 4.0)
plt.ylim(0.0, 2.0)
plt.show()
```

このプログラムファイルは、サポートページ [1] には plot_graph.py というファイル名で用意されている。実行すると、以下の画像ができる；

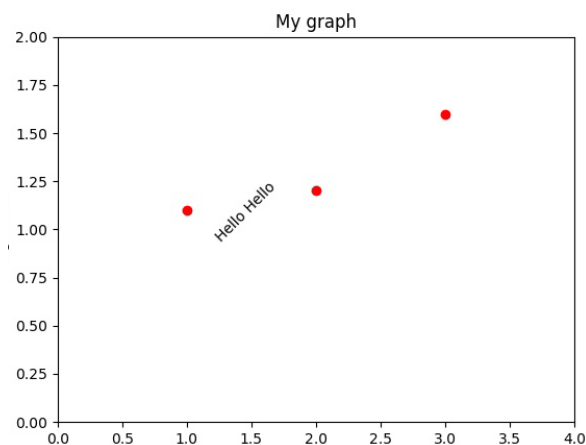


図 9: ファイル読み込みしたデータをグラフ化する

注：プログラムには、「plt.title('My graph')」など、装飾用のコマンドも入っている．意味ついては、各自で調べてみること．

6 Scipy の動作確認

科学計算モジュール Scipy ⁶ が正しくインストールされていることを確認する。

まず、グラフィカル環境 IDLE 上で Python を起動する (Sec. 3.2 を参照)。以下のサンプルコードを一行ずつ上から順に打ち込む。これは、円周率 π を出力するプログラムである。

```
>>> import scipy
>>> scipy.pi
```

下のように、円周率が出力されれば動作確認は完了である。

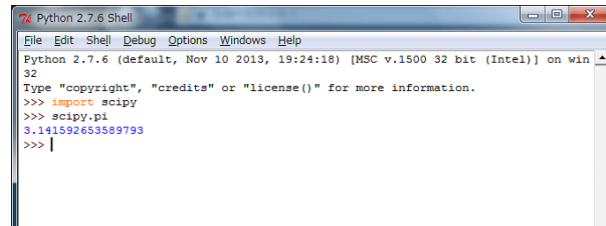


図 10: 動作確認画面

⁶ Scipy.org(公式): <http://www.scipy.org/>

7 Sympy の動作確認

7.1 Sympy の動作確認

数式処理モジュール Sympy ⁷ が正しくインストールされていることを確認する .

まず、グラフィカル環境 IDLE 上で Python を起動する (Sec. 3.2 を参照) . 以下のサンプルコードを一行ずつ上から順に打ち込む . これは、円周率 π を出力するプログラムである .

```
>>> from sympy import *
>>> pi.evalf()
```

下のように、円周率が出力されれば動作確認は完了である .

```
3.14159265358979
```

7.2 Sympy での数式処理

sympy は Maxima のような数式処理ができる . 解説ページとして , Scipy Lecture Note Sec. 3.2 「Sympy : Python での代数計算」⁸ などを参照せよ .

例 1(分数) : $(1/2) + (1/3) = (5/6)$

```
>>> a = Rational(1,2)
>>> b = Rational(1,3)
>>> a+b
5/6
```

例 2(式の展開) : $(x + y)^2 = x^2 + 2xy + y^2$

```
>>> x = Symbol('x')
>>> y = Symbol('y')
>>> expand((x+y)**2)
x**2 + 2*x*y + y**2
```

例 3(微分)

```
>>> x = Symbol('x')
>>> diff(x**2,x)
2*x
```

例 4(定積分)

```
>>> x = Symbol('x')
>>> integrate(x**2, (x,0,1))
1/3
```

例 5(方程式) : $x^2 - 1 = 0$

⁷ Sympy(公式): <http://www.sympy.org/>

⁸ <http://www.turbare.net/transl/scipy-lecture-notes/packages/sympy.html>

```
>>> x = Symbol('x')
>>> solve(x**2-1)
[-1, 1]
```

8 scikit-learn の動作確認

人工知能 (機械学習) 型データ解析モジュール `scikit-learn`⁹ が正しくインストールされていることを確認する。

まず、グラフィカル環境 IDLE 上で Python を起動する (Sec. 3.2 を参照)。

```
>>> import sklearn
```

を実行して、エラーがでなければ、`scikit-learn` がインストールされていることを意味する。

サポートページ [1] にあるプログラム「sample_4.py」(下記) を実行する。

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
# create samples
sample_size = 30
err_sigma = 0.1
x = np.random.rand(sample_size, 1)
err = err_sigma * np.random.randn(sample_size, 1)
y = 3 * x + 1 + err
# train a linear regression model
regr = LinearRegression ()
regr.fit(x, y)
# make predictions
xt = np.linspace (0.0, 1.0, num = 1000).reshape((1000, 1))
yt = regr.predict (xt)
# plot samples and regression result
plt.plot(x, y, 'o')
plt.plot(xt, yt)
plt.show()
```

下のように出力されれば、`scikit-learn` の動作確認は完了である。

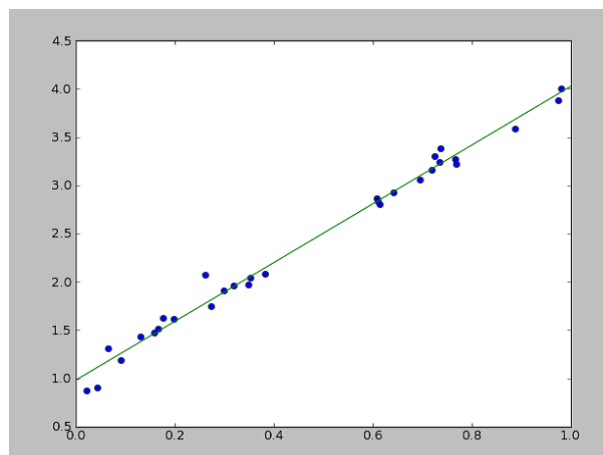


図 11: サンプルコード実行画面

⁹ scikit learn(公式): <http://scikit-learn.org>

9. 参考資料

- [1] Python チュートリアル (学生むけ), Tentative Page by T. HOSHI,
http://www.damp.tottori-u.ac.jp/~hoshi/info/python_tutorial_20171121r1.pdf