

数学的な関数をpythonで扱うには、math モジュールを利用します。最初に、次のコードを実行して下さい。

```
1 # 最初に、このコードを実行して下さい
2 import math
```

数学的な関数の値は、math モジュールを import すれば、計算に利用できます。

円周率 π

```
math.pi
```

```
1 math.pi

3. 141592653589793
```

ネイピア数 e

```
math.e
```

```
1 math.e

2. 718281828459045
```

+ コード

+ テキスト

三角関数も math モジュールを使います。

```
math.sin(math.pi/6)
```

```
1 math.sin(math.pi/6)

0. 49999999999999994
```

基本的に python では累乗を `**` と書きます。例えば、 2^3 は `2**3` と入力します。

```
1 2**3

8
```

正規分布の確率密度関数などでは、ネイピア数を底とする指数関数が登場します。 e^3 は、python では `math.e**3` と入力します。

```
1 math.e**3
```

```
20.085536923187664
```

指数関数 e^x は、`exp`関数を用いることもできます。

```
1 math.exp(3)
```

```
20.085536923187668
```

$\lambda > 0$ をパラメタとする。確率密度関数が $f(x) = \lambda e^{-\lambda x}$ ($x \in (0, \infty)$) 与えられる確率分布を、指数分布という。 X がパラメタ λ の指数分布に従うことを、 $X \sim \text{Exp}(\lambda)$ と表すことにする。指数分布は、行列の待ち時間などに利用される。

$X \sim \text{Exp}(1)$ とする。途中まで書かれたコードを利用して、区分解法により、 $P(0 < X < 1)$ を計算せよ。

```
1 # このまま実行してもエラーで停止します
2 expDst = lambda x: math.exp(-x) # 指数分布の定義
3 bottom, top = 0.0, 1.0 # 積分区間の下端と上端
4 divnum = 10**5 # 区間の分割数
5
6 interval = top - bottom
7 integral = 0.0
8
9 for i in range(divnum):
10     integral += (expDst(bottom + i * interval / divnum) + 4 * expDst(bottom + (i + .5) * interval / divnum) + expDst(top)) / 6
11 print(integral)
```

```
0.6321205588285571
```

標準正規分布 $\mathcal{N}(0,1)$ に従う確率変数 Z に対して、確率 $\begin{matrix} P(0 \leq Z \leq u) \end{matrix}$ を、 u のいろいろな値に対して計算して表にまとめたものを**正規分布表**という（教科書p.155）。正規分布表は、次回以降の授業で正規分布に従う確率変数についての計算を行う際に、頻繁に利用される。

教科書の巻末には、 u の値を 0.01 刻みにして計算した正規分布表がある。以下の手順によって、正規分布表を出力せよ。

1. 初期値の設定

```
u = 0.0
```

```
integral = 0.0
```

2. 刻み幅を設定（今回は0.01）

```
diff = 0.01
```

3. whileループで、uの値が4.0を超えるまで、0.01刻みで確率の値（何らかの近似を行った積分値）をintegralの値に加え続ける。uの値とループ内で加えたintegralの値をprint関数で出力する。ただし、whileループは以下の構造をもつ:

```
while(u <= 4.0):
    # ループさせたい操作（今回はdiff分の刻みでの確率を加える）
    # uの値とintegralの値を出力する
    # uの値にdiffを加えることを忘れない
```

4. 出力された結果を、教科書巻末の正規分布表と比較せよ。

```
1 u = 0.0
2 integral = 0.0
3 diff = 0.01
4
5 nord = lambda x: (2 * math.pi) ** (-0.5) * math.exp(- x**2 / 2)
6
7 print("自作: 標準正規分布表")
8 while (u <= 4.0):
9     print("{:.2f}".format(u), ":", "{:.6f}".format(integral))
10    integral += (nord(u) + 4 * nord(u + 0.5 * diff) + nord(u + diff)) / 6 * diff # 0.01刻みなので、幅0.01の短冊を加
11    u += diff
```

自作: 標準正規分布表

```
0.00 : 0.000000
0.01 : 0.003989
0.02 : 0.007978
0.03 : 0.011966
0.04 : 0.015953
0.05 : 0.019939
0.06 : 0.023922
0.07 : 0.027903
0.08 : 0.031881
0.09 : 0.035856
0.10 : 0.039828
0.11 : 0.043795
0.12 : 0.047758
0.13 : 0.051717
0.14 : 0.055670
0.15 : 0.059618
0.16 : 0.063559
0.17 : 0.067495
0.18 : 0.071424
0.19 : 0.075345
0.20 : 0.079260
0.21 : 0.083166
0.22 : 0.087064
0.23 : 0.090954
0.24 : 0.094835
0.25 : 0.098706
0.26 : 0.102568
```

```
0.27 : 0.106420
0.28 : 0.110261
0.29 : 0.114092
0.30 : 0.117911
0.31 : 0.121720
0.32 : 0.125516
0.33 : 0.129300
0.34 : 0.133072
0.35 : 0.136831
0.36 : 0.140576
0.37 : 0.144309
0.38 : 0.148027
0.39 : 0.151732
0.40 : 0.155422
0.41 : 0.159097
0.42 : 0.162757
0.43 : 0.166402
0.44 : 0.170031
0.45 : 0.173645
0.46 : 0.177242
0.47 : 0.180822
0.48 : 0.184386
0.49 : 0.187933
0.50 : 0.191462
0.51 : 0.194974
0.52 : 0.198468
0.53 : 0.201944
0.54 : 0.205401
0.55 : 0.208840
0.56 : 0.212260
```

<比較結果の考察> 下のテキストボックスを**ダブルクリック**して、上書きして下さい。
[Markdown記法](#)や、[LaTeX記法](#)も利用できます。

(ここに比較結果の考察を記入する)

<本授業の学び> 本授業で学んだことを、下のテキストボックスに記入して下さい。

(ここに本授業の学びを記入する)

