Python超入門
講座付録

プログラミングで学ぶ大学数学

1. 線形代数編
2. 解析学編
3. 統計学編

Numpy

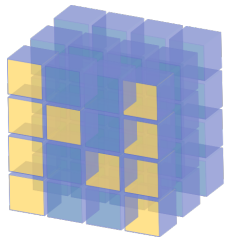
インポート方法

```
import numpy as np
```

数値計算を効率的に行うための拡張モジュール

- 内部はC言語(及びFortran) で実装されており、高速
- BLAS APIを実装した行列演算ライブラリ

機能はこの辺で調べると良い。
https://note.nkmk.me/numpy/
https://deepage.net/features/numpy/
https://docs.scipy.org/doc/numpy/index.html (マニュアル)



NumPy

Scipy

プログラミング数学、科学、工学のための数値解析ソフトウェア

■　Numpyと組み合わせて使う

■　統計、最適化、積分、線形代数、フーリエ変換、
信号・イメージ処理、遺伝的アルゴリズム、ODE ソルバ、特殊関数、
その他のモジュールを提供

マニュアル
https://docs.scipy.org/doc/scipy/reference/index.html

Matplotlib

科学計算用ライブラリNumpyのためのグラフ描画ライブラリ (matplotlib)

- グラフを描画する時に使う
- 折れ線グラフ, ヒストグラム, 散布図, 3Dグラフ等が描ける

※　機能がめっちゃたくさんあるので、使う時は頑張ってググりましょう。

Matplotlib Plot Examples

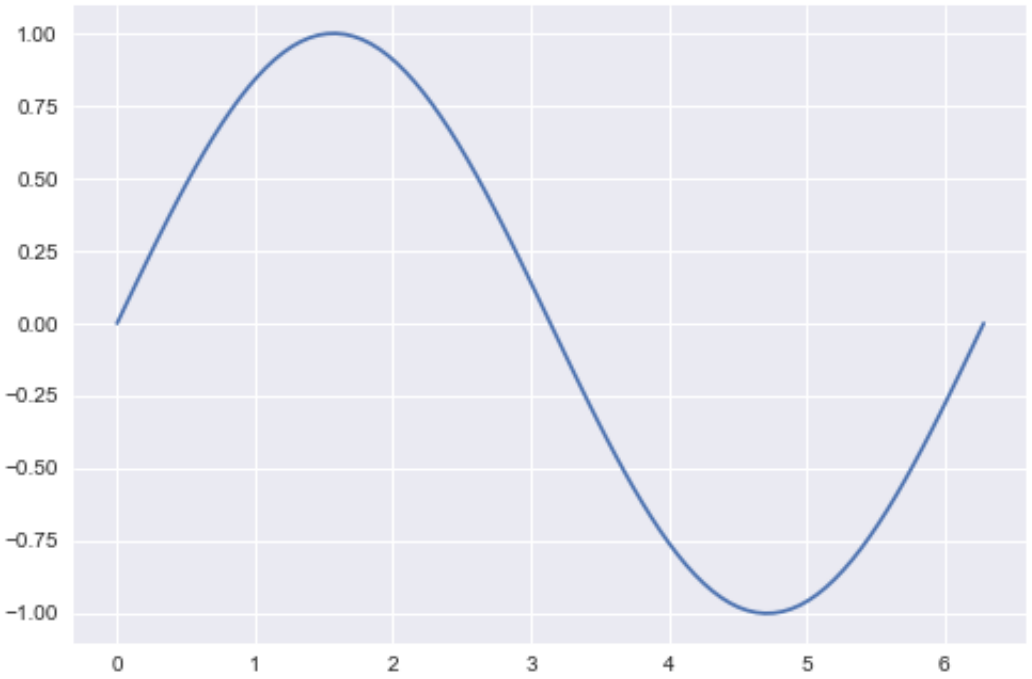
line chart

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

x = np.linspace(0.0,2*np.pi,100)
y = np.sin(x)
plt.plot(x,y)

# ファイルに図を保存
plt.savefig("sin.png")

# 画面表示なら
# plt.show()
```



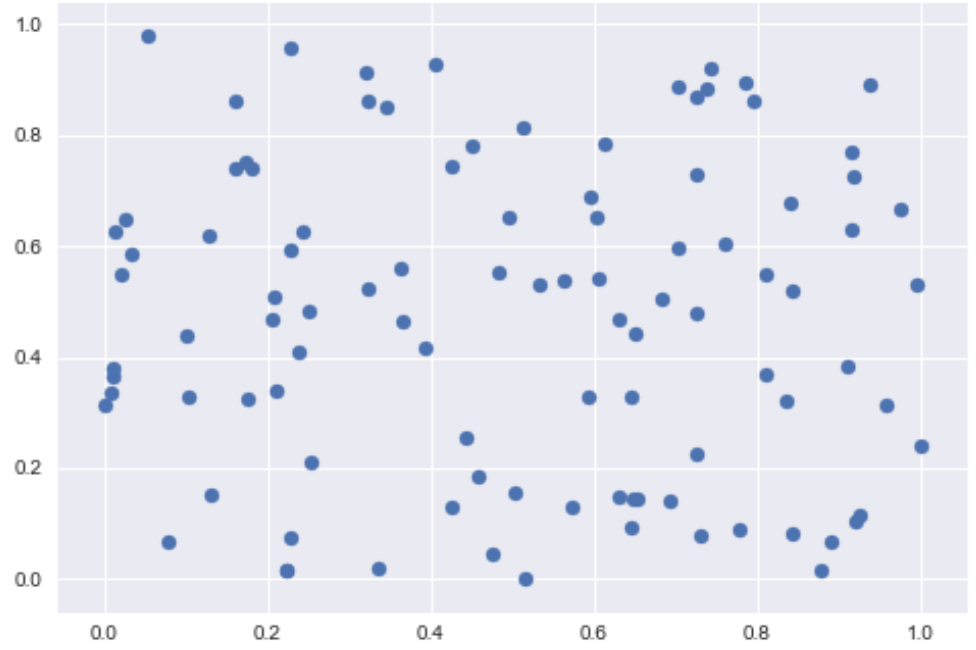
Matplotlib Plot Examples

scatter plot

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

# [0.0, 1.0]の一様乱数を100個生成
a = np.random.rand(100)
b = np.random.rand(100)

plt.scatter(a,b)
plt.savefig("scatter.png")
```



liner algebara

線形代数編

行列の作り方

- 基本 : numpy.array(<List>)

  Listが1次元ならベクトル, 2次元なら行列となる.

  (3次元以上を指定してもテンソルが作れる.)

$$\begin{pmatrix} 1 & 2 \\ 4 & 5 \end{pmatrix}$$

```
np.array( [ [ 1, 2 ], [ 4, 5] ] )
```

- 単位行列 : numpy.eye(<N>)

  単位正方行列. 整数値Nでサイズを指定.

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

```
np.eye( 2 )
```

- 零行列 : numpy.zeros((<N>,<M>, ...))

  零行列を作成. タプルでサイズを指定.

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

```
np.zeros( ( 2,3 ) )
```

※ リストの内包記法を使って作ると便利.

※ numpy.matrix もあるけど使ったことがない.

行列のサイズ, 基本演算

- 行列サイズ : <Array>.shape
  行列のサイズがタプルで取得できる.

- 基本計算

算術演算子	結果
+	足し算
-	引き算
*	要素ごとの掛け算 (アダマール積)
@	行列積 (numpy.dotでもok)
/	要素ごとの割り算
**	行列の累乗 (負の指数は無理)
//	要素ごとの切り捨て割り算

行列式, 転置行列, 逆行列, 行列ランク

```
import numpy as np
import scipy.linalg as LA
```

とインポートする。

■ 行列式, 転置行列, 逆行列, 行列ランク

	記述方法
行列式 $\|A\|$	LA.det (A)
転置行列 A^T	np.transpose(A)
逆行列 A^{-1}	LA.inv(A)
行列ランク rank A	LA.matrix_rank(A)

固有値

- 固有値, 固有ベクトル : scipy.linalg.eig(<Array>)

  固有値と対応する固有ベクトルを求める.

```
A = np.array( [ [ 8,1], [4,5] ] )
value, vector = LA.eig( A )

print("Eigenvalue")
print(value)

print("Eigenvector")
print(vector)
```

実行結果

```
Eigenvalue
[ 9.+0.j  4.+0.j]
Eigenvector
[[ 0.70710678 -0.24253563]
 [ 0.70710678  0.9701425 ]]
```

正規直交基底(Gram-schmidt 直交化)

- 正規直交基底 (Gram-schmidt 直交化) : scipy.linalg.qr(<Array>)
  Gram-schmidt 直交化法を用いて, QR分解を行なっている.

```
A = np.array( [ [ 8,1], [ 4,5] ] )

# QR分解のQが求まった正規直交行列
Q, R = LA.qr( A )

print(Q)
```

実行結果

```
[[-0.89442719 -0.4472136 ]
 [-0.4472136   0.89442719]]
```

Analysis

解析学編

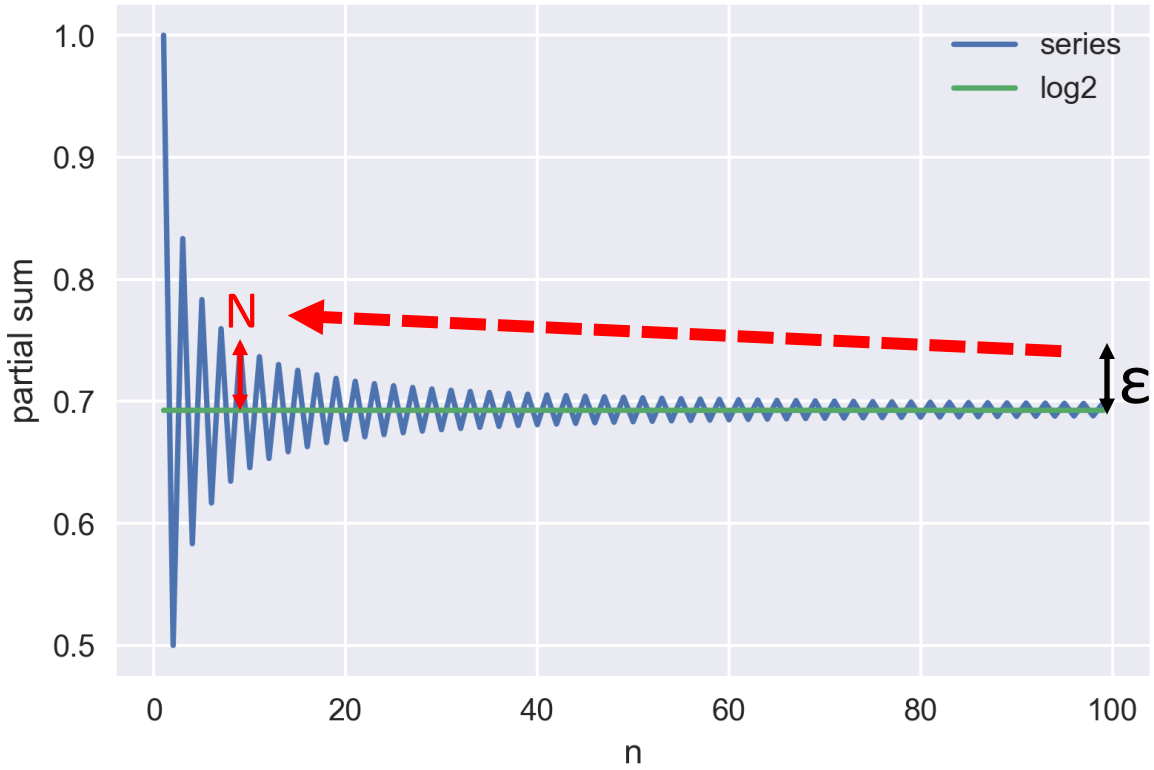
数列, 級数, ε-N論法

- 交代級数 (メルカトル級数)

$$1-\frac{1}{2}+\frac{1}{3}-\frac{1}{4}+\cdots=\sum_{k=1}^{\infty}\frac{(-1)^{k-1}}{k}$$

$$x\text{ 軸}:n,\ y\text{ 軸}:s_n=\sum_{k=1}^{n}\frac{(-1)^{(k-1)}}{k}$$

ちゃんとlog2に収束していく



実験で用いた
プログラム

```
import matplotlib.pyplot as plt
import numpy as np
numerator = 1
sumValue = 0
x = list(range(1,100))
y = []
for n in x:
    sumValue += (numerator / n)
    numerator += (-1)
    y.append(sumValue)

y_true = np.array( [ np.log(2.0)] + len(x))
plt.plot(x,y,label="series")
plt.plot(x,y_true,label="log2")
plt.savefig("series.png",dpi=300)
```

カントールの区間縮小法

~ 二分法の数学的定式化 ~

閉区間 $I_n = [a_n,\ b_n]\ (n = 1,\ 2,\ \cdots)$ において,

1. $I_1 \supset I_2 \supset \cdots$
2. $\lim_{n\to\infty}(b_n - a_n) = 0$

を満たす時, ある実数 c が存在して, $\cap_{n=1}^{\infty}[a_n,\ b_n] = \{c\}$ が成立.

言い換えると, $\lim_{n\to\infty} a_n = \lim_{n\to\infty} b_n = c$ が成立.

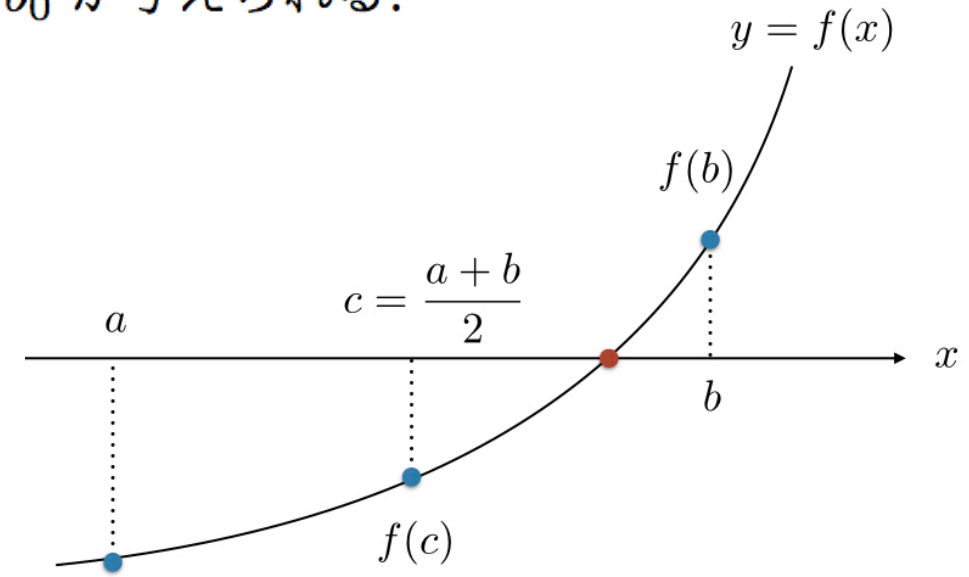
実数の完備性と同値な命題.

二分法： 与えられた関数 f に対して，$f(x)=0$ を満たす x を求めるアルゴリズムの 1 つ．

[条件] f は連続，$f(a_0)f(b_0)<0$ を満たす a_0，b_0 が与えられる．

[アルゴリズム]

1. $m=\frac{a_i+b_i}{2}$ を求める．
2. $\begin{cases} f(m)f(b_i)>0 \text{ のとき：} a_{i+1}=a_i,\ b_{i+1}=m \\ f(m)f(b_i)\leq 0 \text{ のとき：} a_{i+1}=m,\ b_{i+1}=b_i \end{cases}$
3. 1 へ戻る



[証明]

この更新では，$f(a_n)f(b_n)\leq 0 \quad (\forall n\in\mathbb{N})$ が成り立っている．

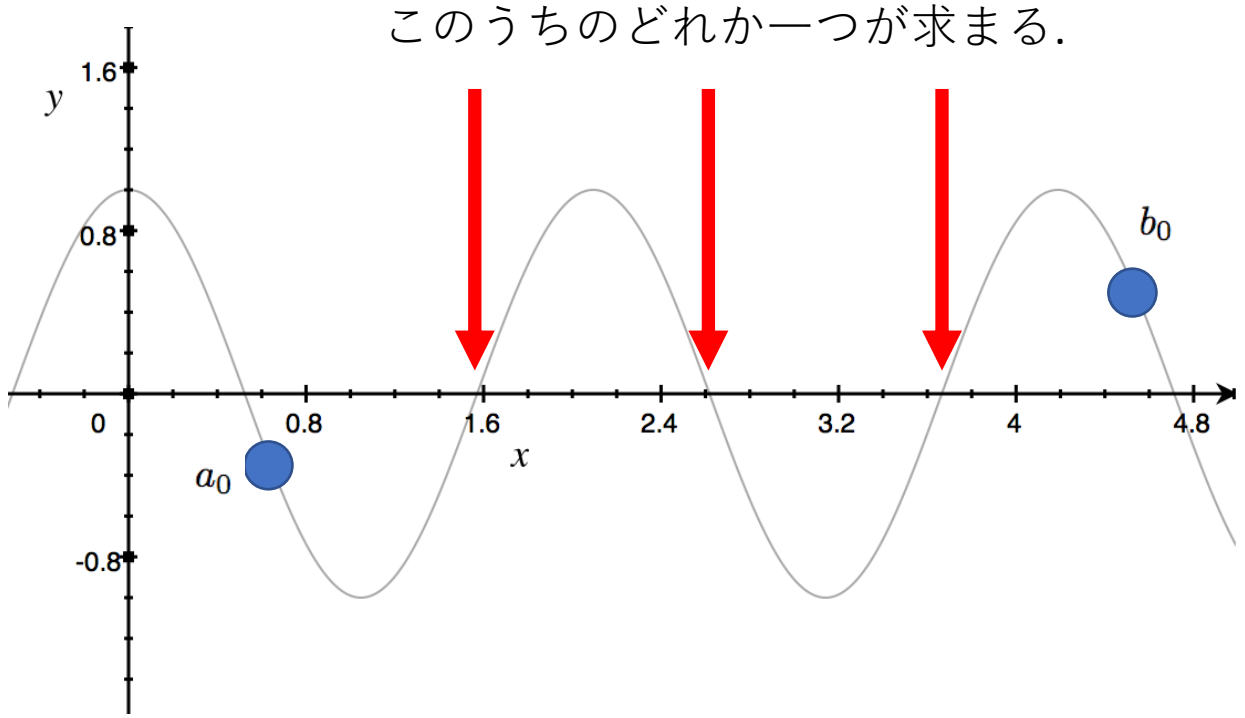
$I_n=[a_n,\ b_n]$ とすれば，$I_i\supset I_{i+1}$，$b_n-a_n=\frac{(b_0-a_0)}{2^n}\to 0 \quad (n\to\infty)$ を満たす．

カントールの区間縮小法により，$\lim_{n\to\infty}a_n=\lim_{n\to\infty}b_n=c$ となる c が存在して，

$$f(c)f(c)\leq 0\Rightarrow f(c)=0$$

補足

初期区間 [a_0, b_0] 内にfの値を0にする値が複数存在する場合



解は一意に定まらないが, 使える

二分法の拡張

■　二分法の問題を以下のように言い換えることができる.

$$\forall \epsilon > 0,\ f(c-\epsilon) \in Y \land f(c+\epsilon) \in Y^c \text{ を満たす } c \text{ を 1 つ求める.}$$
$$\text{ただし, } Y = \{y \in \mathbb{R} \mid y < 0\} \text{ または } \{y \in \mathbb{R} \mid y > 0\}.$$

集合Yを一般化して

$$Y = \{\omega \mid P(w) \text{ が真 }\} \quad (P \text{ は命題関数})\}$$

とすれば, 実数値連続関数に限らず適用することが可能.

(※) 二分探索は関数の定義域を自然数とした場合と考えられるが,
最初のカントールの区間縮小定理を用いた議論は行えない.
この拡張は同様に行うことができる. (競技プログラミングでも頻出)

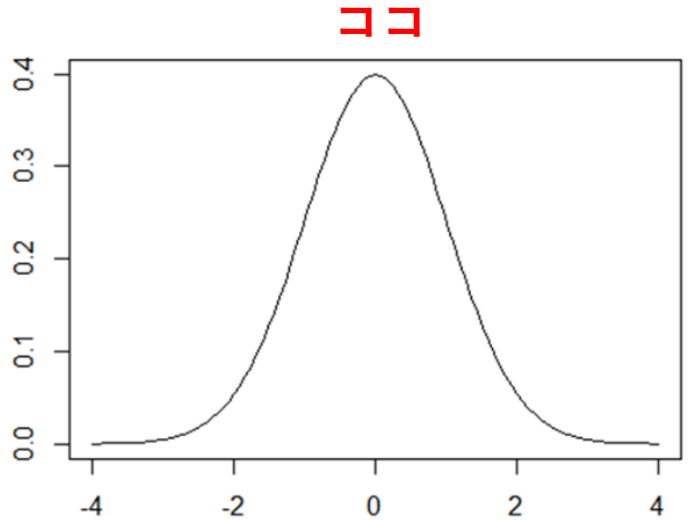
Statistics

統計学編

確率分布

```
import numpy as np
import scipy.stats as ST
```

をインポート



- 扱える分布

https://docs.scipy.org/doc/scipy/reference/stats.html#continuous-distributions

正規分布を例に進めていく

```
dist = ST.norm
```

- 確率密度関数 : dist.pdf (<x>, <paramater> ...)

$$pdf(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp -\frac{(x-\mu)^2}{2\sigma^2}$$

```
# 標準正規分布の山の値
# x = 0, μ=0, σ=1
dist.pdf ( x=0, loc=0, scale=1 )
```

■ 累積分布関数 : dist.cdf(<x> , <paramater> ...)

$$cdf(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi\sigma^2}} \exp -\frac{(t-\mu)^2}{2\sigma^2} dt$$

```
# x = 0, μ=0, σ=1
dist.cdf ( x=0, loc=0, scale=1)
```

■ パーセント点 : dist.ppf (<y>, <paramater> ...)

$$ppf(q) = cdf^{-1}(q)$$

```
# 左側確率が0.5のパーセント点
dist.ppf ( q=0.5, loc=0, scale=1 )
```

■ 確率変数 : dist.rvs(<paramater> ... , size=<n>)

$$X \sim N(\mu,\ \sigma^2)$$

```
# μ=0, σ=1
dist.rvs( loc=0, scale=1)
```

複数の確率変数を同時に作れる

$$X_1,\ X_2,\ \cdots,\ X_n \sim N(\mu,\ \sigma^2)$$

```
# μ=0, σ=1, n = 10
dist.rvs( loc=0, scale=1, size=10)
```

中心極限定理(実験)

■ 中心極限定理

期待値 μ, 分散 σ^2 をもつ独立同分布に従う確率変数列 $X_1,\ X_2,\ \cdots,\ X_n$ に対し, $S_n = \sum_{i=1}^{n} X_i$ とおくと,

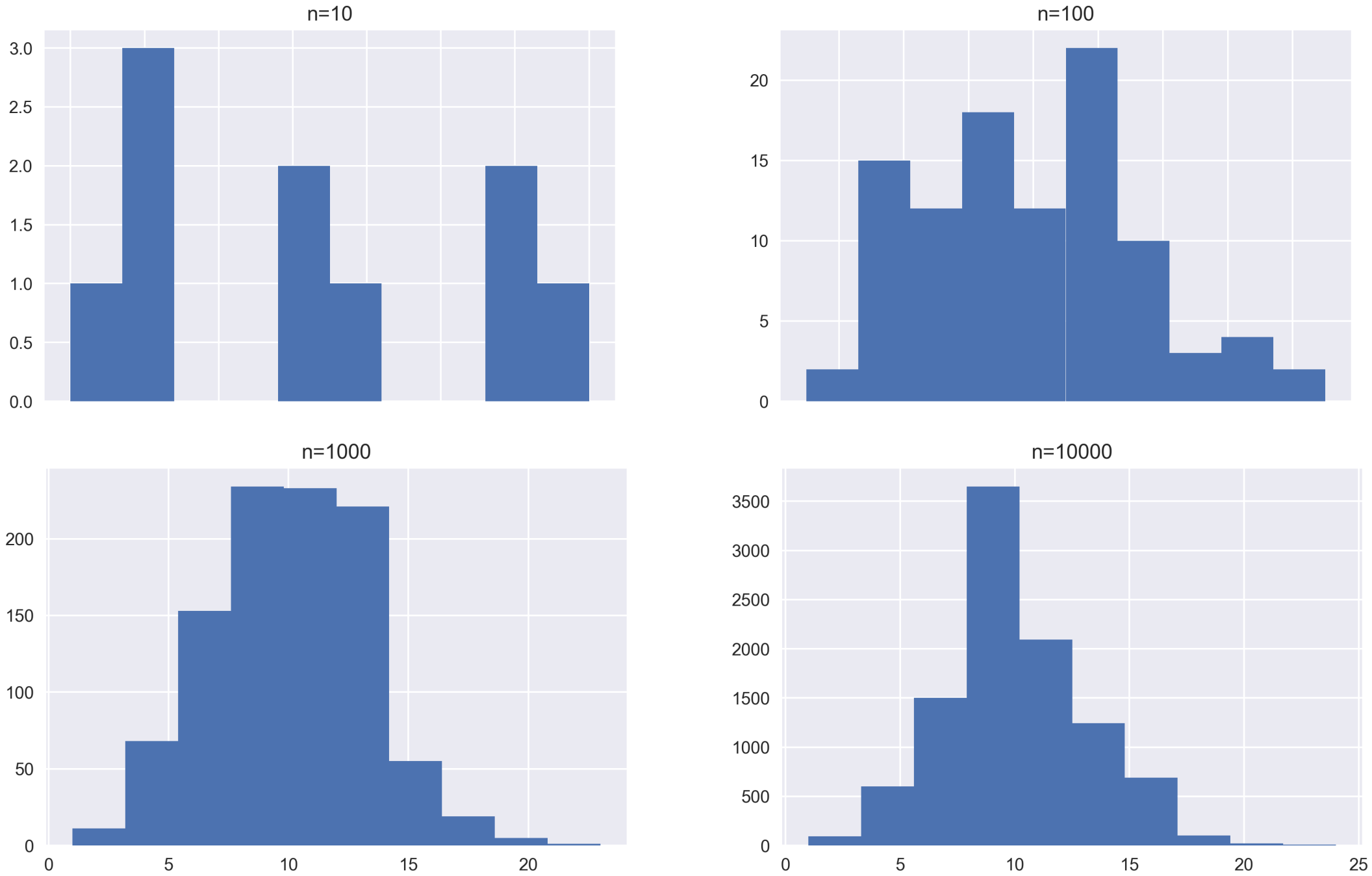
$$P(\frac{S_n - n\mu}{\sqrt{n\sigma^2}} \leq \alpha) \to \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\alpha} \exp(-\frac{x^2}{2}) dx$$

■ 実験

ポアソン分布に従うn個の確率変数をつくる.

$$X_1,\ X_2,\ \cdots,\ X_n \sim Po(\lambda),\ (\mu = \lambda,\ \sigma^2 = \lambda)$$

平均 λ = 10 とする. n の値を少しずつ大きくして, 分布の形の変化を見る.


n=10
3.0
2.5
2.0
1.5
1.0
0.5
0.0
n=100
20
15
10
5
0
n=1000
200
150
100
50
0
0
5
10
15
20
n=10000
3500
3000
2500
2000
1500
1000
500
0
0
5
10
15
20
25

仮説検定

- t検定 (1群) : ST.ttest_1samp ([<X_1>, <X_2, ... ,<X_n>], <mu_0>)

$X_1,\ X_2,\ \cdots,\ X_n \sim N(\mu,\ \sigma^2)$ に従うと仮定. ただし$\mu,\ \sigma$は未知.

帰無仮説: $\mu = \mu_0$, 対立仮説: $\mu \neq \mu_0$, 統計量 $t = \dfrac{\bar{X} - \mu_0}{\sqrt{\frac{s^2}{n}}}$, s^2は不偏分散

```
# N( 0,1 ) から n=100でsampling
# mu_0 = 10 で検定を行う.
dist = ST.norm
rvs = dist.rvs( loc=0, scale=1, size=50)
print( ST.ttest_1samp( rvs, 10) )
```

実行結果

```
Ttest_1sampResult(
    statistic=-67.000797324427779,
    pvalue=7.4690313161094049e-50
)
```

統計量 t の値とp値が返ってくる. 上の実験では有意水準1%のとき, pvalue < 0.01 より棄却

■ t検定 (対応のない2群) :

ST.ttest_ind ([X_1, ... ,X_n], [Y_1, ..., Y_m], *equal_var=(True/False)*)

$$X_1,\ X_2,\ \cdots,\ X_n \sim N(\mu_X,\ \sigma_X^2),\ Y_1,\ Y_2,\ \cdots,\ Y_m \sim N(\mu_Y,\ \sigma_Y^2)$$

$$\text{帰無仮説: } \mu_X = \mu_Y,\ \text{対立仮説: } \mu_X \neq \mu_Y$$

- 等分散性が仮定される場合: equal_var = True

$$\text{統計量 } t = \frac{\bar{X_n} - \bar{X_m}}{s^2\sqrt{\frac{1}{n} + \frac{1}{m}}},\ s = \frac{(n-1)s_X^2 + (m-1)s_Y^2}{n+m-2},\ s_X,\ s_Y \text{は不偏分散}$$

- 等分散性が仮定できない場合: equal_var = False

$$\text{統計量 } t = \frac{\bar{X_n} - \bar{X_m}}{\sqrt{\frac{s_X}{n} + \frac{s_Y}{m}}},\ s_X,\ s_Y \text{は不偏分散}$$

```
# n = 100,  X_n ~ N( -2,10 )
# m = 200, Y_m ~ N(5, 10) .
dist = ST.norm
X = dist.rvs( loc=-2, scale=1, size=100)
Y = dist.rvs( loc=5, scale=10, size=200)
print( ST.ttest_ind( X,Y, equal_var=True) )
```

実行結果

```
Ttest_indResult(
    statistic=-7.4502221494680771,
    pvalue=1.014329824048142e-12
)
```

pvalue < 0.01 より帰無仮説を棄却