

中心極限定理のサイコロによる例示

著者：関勝寿 公開日：2024年8月30日 ソース：<https://sekika.github.io/2024/08/30/dice/>

中心極限定理によると、独立同分布の確率変数の n 個の和（あるいは平均）は、 n が大きくなると正規分布に近づく。そのことを、サイコロの目を例に説明する。なお、この記事は正規分布の必然性に関する考察の中で中心極限定理を説明することを目的として作成されたものである。

1つのサイコロを振ると、出る目は1から6までの整数であり、それぞれの目が出る確率は等しい。現実のサイコロでは重心の偏りによる出目のばらつきは生じるが、ここでは同様に確からしいという仮定をする。このような分布を一様分布と呼ぶ。

次に、サイコロを複数回振ったときを考える。サイコロを n 回振ったときに、それぞれの確率を $X_1, X_2, \dots, X_n$ と書く。すなわち、各 X_i はサイコロの目（1から6の一様分布）であり、期待値（平均値） μ は3.5、標準偏差 σ は約1.71である。このときに、これらの変数の和

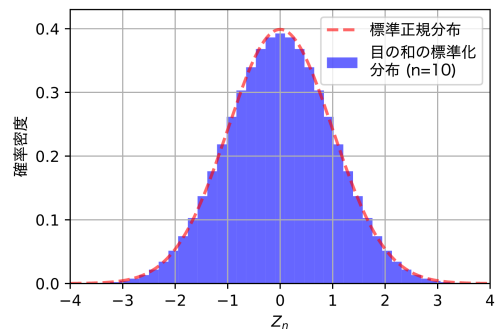
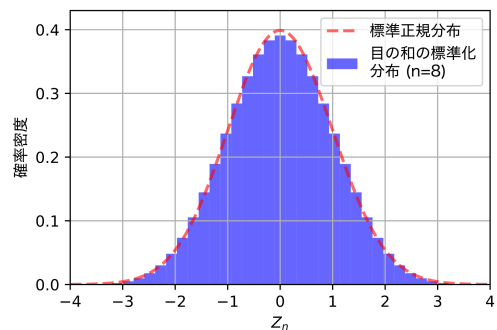
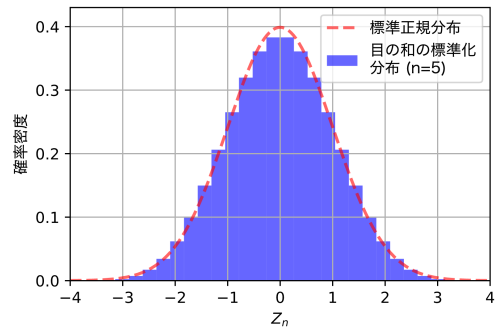
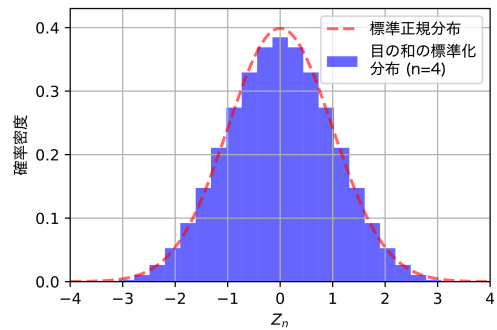
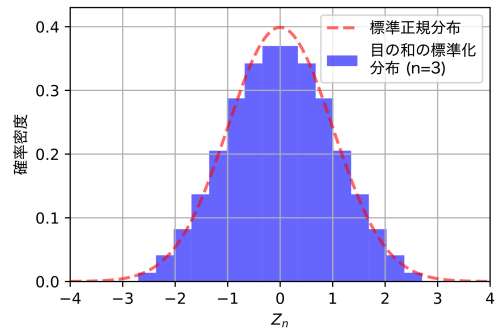
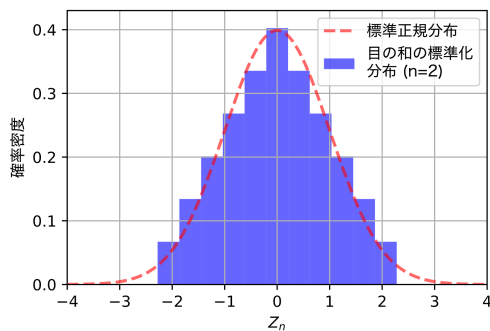
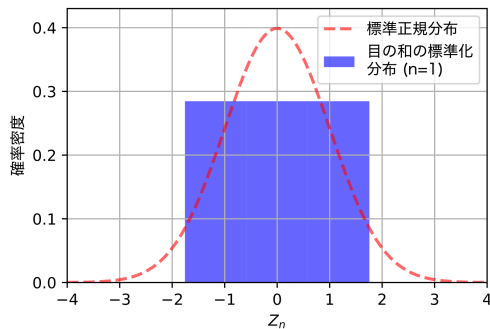
$$S_n = X_1 + X_2 + \dots + X_n$$

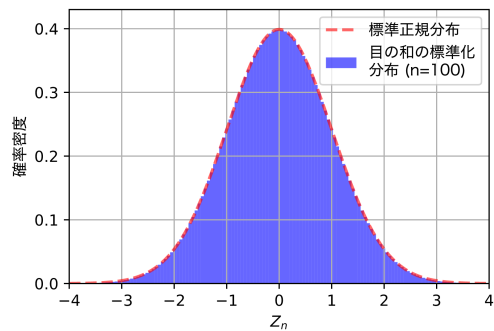
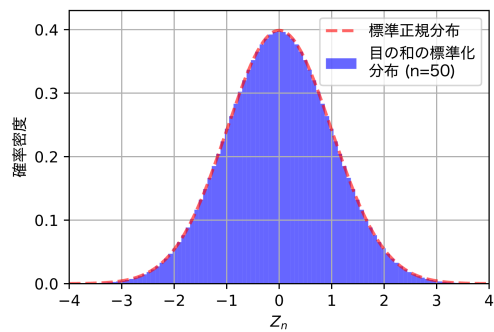
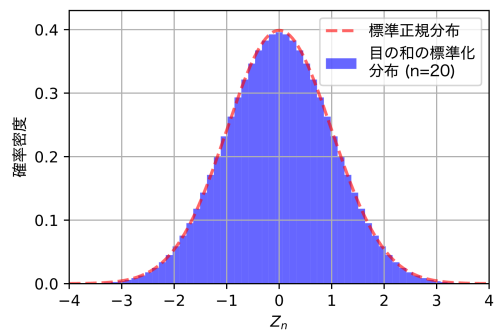
は平均 $n\mu$ 、標準偏差 $\sigma\sqrt{n}$ となる。このときに、 S_n を標準化した Z_n

$$Z_n = \frac{S_n - n\mu}{\sigma\sqrt{n}}$$

は、中心極限定理により $n \rightarrow \infty$ において標準正規分布 $N(0, 1)$ に収束する。

以下に、 $n = 1, 2, 3, 4, 5, 8, 10, 20, 50, 100$ の場合についてこの計算をして、標準正規分布に近づく様子を示す。なお、この図において縦軸は確率密度であり、確率そのものではない。サイコロの目の和の標準化分布において、バーの面積、すなわちバーの高さにバーの幅をかけたものが、確率をあらわす。





このように、 n が大きくなるにつれて標準正規分布に近づくことがわかる。

なお、この図は乱数でサイコロの目を発生させて分布を描いたものではない。乱数を発生させるシミュレーションでは、同じ n で比較するとこの図ほどきれいな正規分布とはならないが（たとえば $n = 2$ の場合には2つの目が見えるだけで正規分布にはまったく近づくかない）、それでも n を大きくして同様の計算をすれば同様に正規分布に近づく。中心極限定理は一樣分布でなくても成立するため、出目に偏りがあるサイコロでも正規分布に近づく（ただし、平均と標準偏差は出目の偏りによって変わる）。

1. プログラム

このページの図を書くために作成した Python のプログラムを掲載する。日本語のフォントを使うために、フォントファイルのパスを FONT_PATH に設定する。詳しくは [Matplotlib で日本語のグラフを作成する](#) を参照。

使い方

usage: dice.py [-h] [-e EXT] [-b]

中心極限定理のサイコロによる例示

options:

-h, --help show this help message and exit
-e EXT, --ext EXT 図のファイルの拡張子（デフォルト svg）
-b, --black 白黒の図を生成

プログラム

```
#!/usr/bin/env python3
#
# 中心極限定理のサイコロによる例示
#
# サイコロを n 回振ったときの目の和の確率の標準化分布が標準正規分布に漸近することを示す。
#
# Author: Katsutoshi Seki
# License: MIT License
#
# 日本語フォントのパスを設定
# 詳しくは https://sekika.github.io/2023/03/11/pyplot-japanese/
FONT_PATH = '/path/to/fontfile/HiraKakuProN-W4-AlphaNum-02.otf'
```

```
def main():
    import argparse
    parser = argparse.ArgumentParser(description=' 中心極限定理のサイコロによる例示 ')
    parser.add_argument('-e', '--ext', default='svg',
                        help=' 図のファイルの拡張子（デフォルト svg） ')
    parser.add_argument('-b', '--black', action='store_true',
                        help=' 白黒の図を生成 ')
    args = parser.parse_args()
    for n in (1, 2, 3, 4, 5, 6, 8, 10, 20, 30, 40, 50, 100):
        dice_roll(n, args)
```

```
def dice_roll(n, args):
    import os
    import numpy as np
    import matplotlib.pyplot as plt
    from scipy.stats import norm
    from matplotlib.font_manager import FontProperties
    # サイコロの目の平均と標準偏差を計算
    mu = 3.5
    sigma = np.sqrt(np.mean((np.arange(1, 7) - mu)**2))

    # サイコロの目の和の理論的な確率分布を計算
    x = np.arange(n, 6 * n + 1)
    prob = np.zeros(len(x))
    memo = {}
```

```
def dice_sum_prob(n, total):
    if (n, total) in memo:
        return memo[(n, total)]
    if n == 0:
        return 1 if total == 0 else 0
    if total < 0 or total > 6 * n:
        return 0
    result = sum(dice_sum_prob(n - 1, total - i) for i in range(1, 7))
    memo[(n, total)] = result
    return result
for i in range(len(x)):
    prob[i] = dice_sum_prob(n, x[i])
```

```
# 標準化
mu_sum = n * mu
sigma_sum = np.sqrt(n) * sigma
z = (x - mu_sum) / sigma_sum
```

```
# 標準正規分布の理論曲線
x_norm = np.linspace(-4, 4, 1000)
y_norm = norm.pdf(x_norm, 0, 1)
```

```
# プロット開始
if args.black:
    color = ['grey', 'black']
else:
    color = ['blue', 'red']
plt.figure(figsize=(4.5, 3))
plt.subplots_adjust(left=0.13, right=0.98, top=0.98, bottom=0.15)
fp = FontProperties(fname=os.path.expanduser(FONT_PATH))
plt.plot(x_norm, y_norm,
        color=color[1], linestyle="dashed",
        alpha=0.6, linewidth=2, label=" 標準正規分布 ")
```

```

# 確率をバーの幅で割った確率密度をバーの高さとする。すなわちバーの面積が
確率となる。
width = z[1] - z[0]
plt.bar(z, prob / (6 ** n) / width, width=width,
        color=color[0], alpha=0.6, label=f"目の和の標準化 \n 分布
(n={n})")
plt.axis([-4, 4, 0, 0.43])

plt.xlabel("$Z_n$")
plt.ylabel("確率密度", fontproperties=fp)
plt.legend(prop=fp)
plt.grid(True)
plt.savefig(f'dice{n}.args.ext')

if __name__ == "__main__":
    main()

```