

中心極限定理などについて

- ・ 黒木玄
- ・ 2022-04-11～2022-05-08

このノートではJulia言語 (<https://julialang.org/>)を使用している:

- ・ [Julia言語のインストールの仕方の一例 \(https://nbviewer.org/github/genkuroki/msfd28/blob/master/install.ipynb\)](https://nbviewer.org/github/genkuroki/msfd28/blob/master/install.ipynb)

自明な誤りを見つけたら、自分で訂正して読んで欲しい。大文字と小文字の混同や書き直しが不完全な場合や符号のミスは非常によくある。

このノートに書いてある式を文字通りにそのまま読んで正しいと思ってしまうとひどい目に会う可能性が高い。しかし、数が使われている文献には大抵の場合に文字通りに読むと間違っている式や主張が書いてあるので、内容を理解した上で訂正しながら読んで利用しなければいけない。実践的に数学を使う状況では他人が書いた式をそのまま信じていけない。

このノートの内容よりもさらに詳しいノートを自分で作ると勉強になるだろう。膨大な時間を取られることになるが、このノートの内容に関係することで飯を食っていく可能性がある人にはそのためにかけた時間は無駄にならないと思われる。

目次

- ▼ [1 大数の法則](#)
 - [1.1 Markovの不等式](#)
 - [1.2 Chebyshevの不等式](#)
 - [1.3 大数の弱法則](#)
 - [1.4 不偏分散の \$n \rightarrow \infty\$ での挙動](#)
 - ▼ [1.5 大数の強法則](#)
 - [1.5.1 Borel-Cantelliの補題](#)
 - [1.5.2 4次以下のモーメントの存在を仮定した場合の大数の強法則](#)
 - [1.6 大数の強法則に関する文献](#)
 - [1.7 問題: 大数の法則に関する誤解](#)
 - ▼ [1.8 大数の法則のデモンストレーション](#)
 - [1.8.1 Bernoulli試行](#)
 - [1.8.2 正規分布のサンプルの場合](#)
 - [1.8.3 ガンマ分布の場合](#)
 - [1.8.4 大数の法則が成立しない場合 \(Cauchy分布\)](#)
- ▼ [2 二項分布の中心極限定理](#)
 - [2.1 二項分布の中心極限定理の内容](#)
 - [2.2 二項分布の中心極限定理のプロット](#)
 - ▼ [2.3 必修重要問題: 二項分布の正規分布近似を使った確率の近似計算1](#)
 - [2.3.1 WolframAlphaによる計算1](#)
 - [2.3.2 Julia言語による計算1](#)
 - [2.3.3 Julia言語による視覚化1](#)
 - ▼ [2.4 必修重要問題: 二項分布の正規分布近似を使った確率の近似計算2](#)
 - [2.4.1 WolframAlphaによる計算](#)
 - [2.4.2 Julia言語による計算2](#)
 - [2.4.3 Julia言語による視覚化2](#)
 - [2.5 問題: 二項分布の中心極限定理を使った極限の計算問題](#)
 - [2.6 二項分布の中心極限定理の証明の方針](#)
 - [2.7 \(1\) Stirlingの公式を使った二項分布の確率密度函数の近似](#)
 - [2.8 注意: Kullback-Leibler情報量とSanovの定理との関係](#)
 - [2.9 \(2\) 二項分布の確率質量函数から正規分布の密度函数が出て来ること](#)
- ▼ [3 中心極限定理](#)
 - [3.1 中心極限定理のラフな説明](#)
 - [3.2 中心極限定理の特性函数を使った証明](#)
 - [3.3 中心極限定理の収束の速さと歪度](#)
 - [3.4 中心極限定理のキュムラント母函数を使った証明](#)
 - [3.5 中心極限定理の収束の速さと歪度と尖度](#)
 - ▼ [3.6 中心極限定理のTaylorの定理のみを使う証明](#)
 - [3.6.1 多重積分の書き方](#)
 - [3.6.2 積分剩余項型のTaylorの定理](#)
 - [3.6.3 積分剩余項の別の表示](#)
 - [3.6.4 微分剩余項型のTaylorの定理](#)
 - [3.6.5 Taylorの定理を使った中心極限定理の証明](#)

3.6.6 中心極限定理の収束の速さと歪度と尖度(再)

3.7 問題: 中心極限定理の収束の様子のグラフ

▼ 4 デルタ法

4.1 1次のデルタ法と2次のデルタ法

4.2 おまけ: 2次のデルタ法使用時の近似分散と歪度と尖度に関する不等式

4.3 問題: 2次のデルタ法の不偏分散の平方根の期待値の近似への応用

4.4 問題: 1次のデルタ法の正規分布近似への応用

4.5 問題: 1次のデルタ法による分布の逆数の近似

4.6 例: 自由度が大きな χ^2 分布の逆数の近似

4.7 問題: デルタ法による分布の対数の近似

4.8 例: 自由度が大きな χ^2 分布の対数の近似

4.9 問題: ベータ分布の正規分布近似

4.10 問題: ベータ分布の正規分布近似の直接的証明

▼ 5 確率分布の正規分布近似

5.1 二項分布の正規分布近似

5.2 負の二項分布の正規分布近似

5.3 Poisson分布の正規分布近似

5.4 ガンマ分布の正規分布近似

5.5 χ^2 分布の正規分布近似

5.6 逆ガンマ分布の正規分布近似

5.7 ベータ分布の正規分布近似

5.8 t 分布の正規分布近似

5.9 ベータプライム分布の正規分布近似

5.10 F 分布の正規分布近似

5.11 超幾何分布の正規分布近似

5.12 ベータ二項分布の正規分布近似

5.13 ベータ負の二項分布の正規分布近似

```
In [1]: 1 ENV["LINES"], ENV["COLUMNS"] = 100, 100
2 using BenchmarkTools
3 using Distributions
4 using Printf
5 using QuadGK
6 using Random
7 Random.seed!(4649373)
8 using Roots
9 using SpecialFunctions
10 using StaticArrays
11 using StatsBase
12 using StatsFuns
13 using StatsPlots
14 default(fmt = :png, titlefontsize = 10, size = (400, 250))
15 using SymPy
```

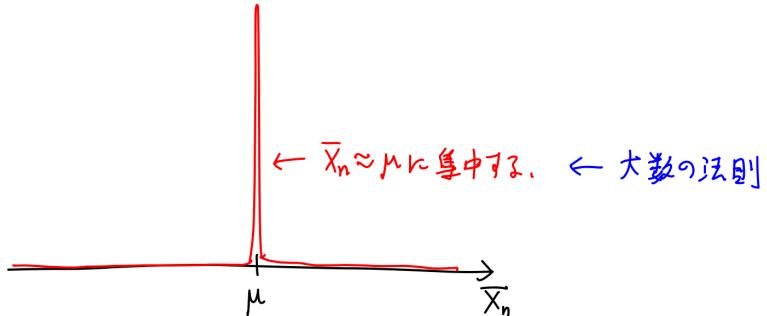
```
In [2]: 1 # Override the Base.show definition of SymPy.jl:
2 # https://github.com/JuliaPy/SymPy.jl/blob/29c5bfd1d10ac53014fa7fef468bc8deccadc2fc/src/types.jl
3
4 @eval SymPy function Base.show(io::IO, ::MIME"text/latex", x::SymbolicObject)
5     print(io, as_markdown("\displaystyle " * sympy.latex(x, mode="plain", fold_short_frac=false))
6 end
7 @eval SymPy function Base.show(io::IO, ::MIME"text/latex", x::AbstractArray{Sym})
8     function toeqnarray(x::Vector{Sym})
9         a = join(["\displaystyle " * sympy.latex(x[i]) for i in 1:length(x)], "\\\\"")
10        """\left[ \begin{array}{r} a \end{array} \right]"""
11    end
12    function toeqnarray(x::AbstractArray{Sym,2})
13        sz = size(x)
14        a = join([join("\displaystyle " .* map(sympy.latex, x[i,:]), "&") for i in 1:sz[1]], "\\\\"")
15        """\left[ \begin{array}{c} \left[ \begin{array}{r} a \end{array} \right] \end{array} \right]"""
16    end
17    print(io, as_markdown(toeqnarray(x)))
18 end
```

大数の法則と中心極限定理のイメージ

$\mu = E[X_i]$, $\sigma = \sqrt{E[(X_i - \mu)^2]}$ とおく.

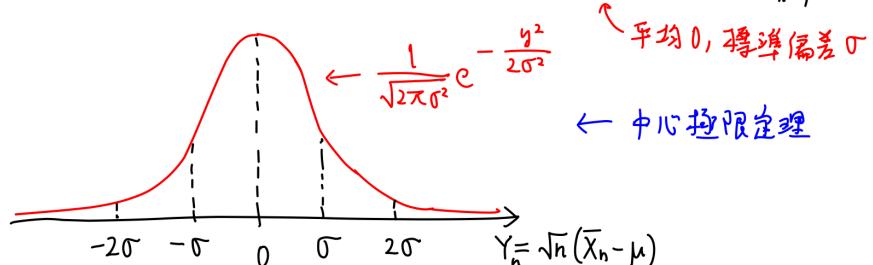
X_1, X_2, X_3, \dots は独立同分布確率変数列であるとし, $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$ とおく.

サイズ n の標本の平均 \bar{X}_n の分布の $n \rightarrow \infty$ の様子: 平均 μ , 標準偏差 $\frac{\sigma}{\sqrt{n}} \rightarrow 0$



$\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$ の分布は $n \rightarrow \infty$ で $\mu = E[X_i]$ に集中する.

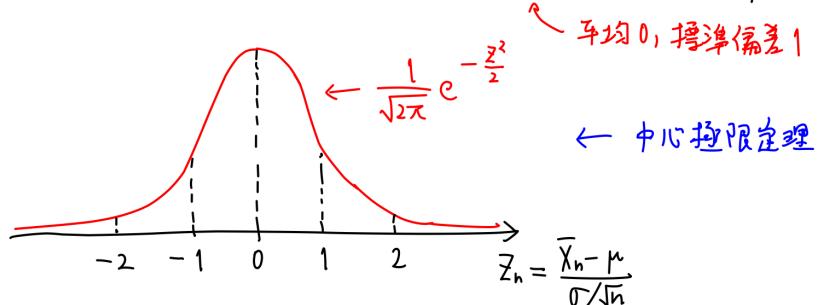
\bar{X}_n と μ の差を \sqrt{n} 倍して拡大して見るとこうなる: $Y_n = \sqrt{n}(\bar{X}_n - \mu) = \frac{1}{\sqrt{n}} \sum_{i=1}^n (X_i - \mu)$.



$\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$ と μ の差の大きさはだいたい $n \frac{\sigma}{\sqrt{n}}$ の大きさであり,

差を \sqrt{n} 倍拡大して分布を見ると標準偏差 σ の正規分布にこうている.

さらに σ 分の1倍するところ: $Z_n = \frac{\sqrt{n}}{\sigma} (\bar{X}_n - \mu) = \frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}} = \frac{1}{\sqrt{n}} \sum_{i=1}^n \frac{X_i - \mu}{\sigma}$



以上のように、大数の法則によって1点に集中する標本平均の分布を \sqrt{n} 倍 (もしくは σ/\sqrt{n} 分の1倍)することにより精密に見ると中心極限定理が得られる.

1 大数の法則

大数の法則 (law of large numbers, LLN)とは、大雑把に言えば、独立同分布確率変数列 X_1, X_2, \dots の最初の n 個の標本平均を $\bar{X}_n = (1/n) \sum_{i=1}^n X_i$ と書くと、 \bar{X}_n が X_i 達の共通の期待値に収束するという結果のことである。(より正確に言うと、その「収束」の仕方に強弱があるって、それについて「大数の強法則」と呼んだり、「大数の弱法則」と呼んだりする。しかし、応用上、多くの場合にその違いに神経質になる必要はない。)

以下では大数の法則について解説する.

1.1 Markovの不等式

確率変数 X と任意の $a > 0$ について

$$P(|X| \geq a) \leq \frac{1}{a} E[|X|].$$

証明: フィル $1_{|x| \geq a}(x)$ を

$$1_{|x| \geq a}(x) = \begin{cases} 1 & (|x| \geq a) \\ 0 & (|x| < a) \end{cases}$$

とおくと, $|x| \geq a$ と $1 \leq |x|/a$ は同値なので常に $1_{|x| \geq a}(x) \leq |x|/a$ となる. ゆえに期待値を取る操作の単調性と線形性より,

$$P(|X| \geq a) = E[1_{|x| \geq a}(x)] \leq E\left[\frac{|X|}{a}\right] = \frac{1}{a} E[|X|].$$

証明終

注意: Markovの不等式, Chebyshevの不等式, Jensenの不等式, Gibbsの情報不等式などについては「Bernoulli試行と関連確率分布」のノートすでに説明していたのであった.

1.2 Chebyshevの不等式

Markovの不等式で X を $(X - \mu)^2$ ($\mu = E[X]$) で置き換えた場合をChebyshevの不等式と呼ぶ.

確率変数 X について $E[|X|] < \infty$ が成立しているならば, X の期待値 $\mu = E[X]$ がうまく定義されている. そのとき, X の分散を $\sigma^2 = E[(X - \mu)^2]$ と書くと, 任意の $\varepsilon > 0$ について

$$P(|X - \mu| \geq \varepsilon) \leq \frac{\sigma^2}{\varepsilon^2}.$$

証明: 前節のMarkovの不等式を X, a がそれぞれ $(X - \mu)^2, \varepsilon^2$ の場合に適用すると,

$$P(|X - \mu| \geq \varepsilon) = P((X - \mu)^2 \geq \varepsilon^2) \leq \frac{1}{\varepsilon^2} E[(X - \mu)^2] = \frac{\sigma^2}{\varepsilon^2}.$$

証明終

1.3 大数の弱法則

X_1, X_2, X_3, \dots は独立同分布確率変数の列であるとし, 共通の期待値 $\mu = E[X_i]$ が定義されており, 共通の分散 $\sigma^2 = E[(X_i - \mu)^2]$ は有限の値であると仮定する. このとき, 最初の n 個の標本平均を

$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$$

と書くと, 任意の $\varepsilon > 0$ について,

$$\lim_{n \rightarrow \infty} P(|\bar{X}_n - \mu| \geq \varepsilon) = 0.$$

注意: この結論は

$$\lim_{n \rightarrow \infty} P(|\bar{X}_n - \mu| < \varepsilon) = 1$$

と同値なので標本平均 \bar{X}_n の分布が $n \rightarrow \infty$ で μ の近く(任意に小さな $\varepsilon > 0$ に関する μ から距離 ε 未満の範囲)に集中して行くことを意味している. このとき \bar{X}_n は μ に 確率収束 するというが, 標本平均 \bar{X}_n の分布が $n \rightarrow \infty$ で μ の近くに集中して行くことを理解していれば, 確率変数の収束に関する細かいことは応用上あまり神経質になる必要はない. だから, このノートでは「大数の弱法則」と「大数の強法則」の区別に神経質にならずに単に「大数の法則」ということが多い.

証明: 標本平均 \bar{X}_n の期待値と分散は以下のようになるのであった:

$$\begin{aligned}
E[\bar{X}_n] &= \frac{1}{n} \sum_{i=1}^n E[X_i] = \frac{1}{n} \sum_{i=1}^n \mu = \mu, \\
(\bar{X}_n - \mu)^2 &= \left(\frac{1}{n} \sum_{i=1}^n (X_i - \mu) \right)^2 = \frac{1}{n^2} \sum_{i,j=1}^n (X_i - \mu)(X_j - \mu), \\
\text{var}(\bar{X}_n) &= E[(\bar{X}_n - \mu)^2] = \frac{1}{n^2} \sum_{i,j=1}^n E[(X_i - \mu)(X_j - \mu)] = \frac{1}{n^2} \sum_{i,j=1}^n \delta_{ij} \sigma^2 = \frac{\sigma^2}{n}.
\end{aligned}$$

最後から2つめの等号で X_i 達が独立(特に無相関)であることを使った。ゆえに, Chebyshevの不等式を X, μ, σ^2 がそれぞれ $\bar{X}_n, \mu, \text{var}(\bar{X}_n) = \sigma^2/n$ の場合に適用すると, 任意の $\epsilon > 0$ について,

$$P(|\bar{X}_n - \mu| \geq \epsilon) \leq \frac{\sigma^2/n}{\epsilon^2} \rightarrow 0 \quad \text{as } n \rightarrow \infty.$$

証明終

1.4 不偏分散の $n \rightarrow \infty$ での挙動

X_1, X_2, X_3, \dots は独立同分布確率変数の列であるとし, その最初の n 個の標本平均と不偏分散をそれぞれ次のように書くこととする:

$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i, \quad S_n^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_n)^2.$$

以下では $\mu = E[X_i]$ がうまく定義されており, $\sigma^2 = E[(X_i - \mu)^2] < \infty$, $E[|X_i - \mu|^3] < \infty$, $E[(X_i - \mu)^4] < \infty$ となつていると仮定する。

「標本分布について」のノートで以下を示していたのであった。 X_i 達が共通に従う分布の歪度(わいど, skewness)と尖度(せんど, kurtosis)をそれぞれ

$$\bar{\kappa}_3 = E \left[\left(\frac{X_i - \mu}{\sigma} \right)^3 \right], \quad \bar{\kappa}_4 = E \left[\left(\frac{X_i - \mu}{\sigma} \right)^4 \right] - 3$$

と書くと,

$$\begin{aligned}
E[\bar{X}_n] &= \mu, \quad E[S_n^2] = \sigma^2, \\
\text{var}(\bar{X}_n) &= \frac{\sigma^2}{n}, \quad \text{cov}(\bar{X}_n, S_n^2) = \sigma^3 \frac{\bar{\kappa}_3}{n}, \quad \text{var}(S_n^2) = \sigma^4 \left(\frac{\bar{\kappa}_4}{n} + \frac{2}{n-1} \right).
\end{aligned}$$

ゆえに, Chebyshevの不等式を X, μ, σ^2 がそれぞれ $S_n^2, E[S_n^2] = \sigma^2, \text{var}(S_n^2) = \sigma^4(\bar{\kappa}_4/n + 2/(n-1))$ の場合に適用すると, 任意の $\epsilon > 0$ について,

$$P(|S_n^2 - \sigma^2| \geq \epsilon) \leq \frac{\sigma^4}{\epsilon^2} \left(\frac{\bar{\kappa}_4}{n} + \frac{2}{n-1} \right) \rightarrow 0 \quad \text{as } n \rightarrow \infty.$$

このように, 不偏分散 S_n^2 についても大数の弱法則が成立しており, S_n^2 の分布は $n \rightarrow \infty$ で σ^2 の近くに集中して行く。

1.5 大数の強法則

この節の内容は無理して読まなくてもよい。定義していない用語や記号法が登場する。大数の強法則の取り扱いには測度論的確率論が必要になるが, この一連のノート群では測度論を避ける方針なので, 大数の強法則について詳しく説明することはできない。以下で紹介する議論は「よく見る議論」のコピー&ペーストに過ぎない。必要な用語の定義もされていないし, 内容的にも雑である。しかし, 大数の強法則の証明の雰囲気を伝えたいと思ったので, このような説明をしてみることにした。

確率空間とは σ -algebra (= σ -field) \mathcal{F} と確率測度 $\mathcal{F} \ni A \mapsto P(A) \in \mathbb{R}_{\geq 0}$ が与えられた集合 Ω のことである。以下では確率空間 Ω 上の点を ω と表す。 Ω 上の実数値可測函数を確率変数と呼ぶ。

1.5.1 Borel-Cantelliの補題

確率空間 Ω において次が成立する:

$$\sum_{k=1}^{\infty} P(A_k) < \infty \implies P \left(\bigcap_{n=1}^{\infty} \bigcup_{k \geq n} A_k \right) = 0.$$

証明: $\sum_{k=1}^{\infty} P(A_k) < \infty$ と仮定し, $B_n = \bigcup_{k \geq n} A_k$ とおく. $P(\bigcap_{n=1}^{\infty} B_n) = 0$ を示せばよい. $\sum_{k=1}^{\infty} P(A_k) < \infty$ より,

$$P(B_b) \leq \sum_{k \geq n} P(A_k) = \sum_{k=1}^{\infty} P(A_k) - \sum_{k=1}^{n-1} P(A_k) \rightarrow 0 \text{ as } n \rightarrow \infty.$$

$B_1 \supset B_2 \supset B_3 \supset \dots$ なので

$$P\left(\bigcap_{n=1}^{\infty} B_n\right) = \lim_{n \rightarrow \infty} P(B_n) = 0.$$

証明終

1.5.2 4次以下のモーメントの存在を仮定した場合の大数の強法則

X_1, X_2, X_3, \dots は独立同分布確率変数の列であるとし, その最初の n 個の標本平均を $\bar{X}_n = (1/n) \sum_{i=1}^n X_i$ と書くことにする. さらに, $\mu = E[X_i]$ がうまく定義されており, $\sigma^2 = E[(X_i - \mu)^2] < \infty$, $\mu_4 = E[(X_i - \mu)^4] < \infty$ となっていると仮定する. このとき,

$$P\left(\lim_{n \rightarrow \infty} \bar{X}_n = \mu\right) = 1.$$

注意: これは標本平均が μ に確率 1 で収束することを意味している.

証明: 大数の弱法則の場合と同様に補事象の側の同値な主張を示す. すなわち, 確率変数(可測函数)としての標本平均 \bar{X}_n が $n \rightarrow \infty$ で μ に収束しない確率が 0 であることを示そう.

確率変数(可測函数)としての最初の n 個の標本平均 \bar{X}_n の ω における値 $\bar{X}_n(\omega)$ が $n \rightarrow \infty$ で μ に収束しないという条件は以下のように書き換えられる. すなわち,

$$A_k(\varepsilon) = \{\omega \mid |\bar{X}_k(\omega) - \mu| \geq \varepsilon\}$$

とおくと, 以下が成立している:

$\bar{X}_n(\omega)$ は $n \rightarrow \infty$ で μ に収束しない.

\Leftrightarrow ある $\varepsilon > 0$ が存在して, 任意の n について, ある $k \geq n$ で $|\bar{X}_k(\omega) - \mu| \geq \varepsilon$ を満たすものが存在する.

\Leftrightarrow ある $\varepsilon > 0$ が存在して, $\omega \in \bigcap_{n=1}^{\infty} \bigcup_{k \geq n} A_k(\varepsilon)$.

$\Leftrightarrow \omega \in \bigcup_{L=1}^{\infty} \bigcap_{n=1}^{\infty} \bigcup_{k \geq n} A_k(1/L)$.

この条件の書き換えによって, 任意の $\varepsilon > 0$ について,

$$\sum_{k=1}^{\infty} P(A_k(\varepsilon)) = \sum_{k=1}^{\infty} P(|\bar{X}_k - \mu| \geq \varepsilon) < \infty$$

を示せば, Borel-Cantelliの補題より $P(\bigcap_{n=1}^{\infty} \bigcup_{k \geq n} A_k(\varepsilon)) = 0$ が得られ, それより

$$P\left(\bigcup_{L=1}^{\infty} \bigcap_{n=1}^{\infty} \bigcup_{k \geq n} A_k(1/L)\right) \leq \sum_{L=1}^{\infty} P\left(\bigcap_{n=1}^{\infty} \bigcup_{k \geq n} A_k(1/L)\right) = 0$$

が導かれ, \bar{X}_n が $n \rightarrow \infty$ で μ に収束しない確率が 0 であることが示されることがわかる.

必要ならば X_i を $X_i - \mu$ で置き換えることによって, $\mu = 0$ と仮定しても一般性が失われないので, $\mu = 0$ と仮定する.

$\varepsilon > 0$ を任意に取る. このとき,

$$P(|\bar{X}_n| \geq \varepsilon) = P\left(\frac{\bar{X}_n^4}{\varepsilon^4} \geq 1\right) = E\left[1_{\bar{X}_n^4/\varepsilon^4 \geq 1}(\bar{X}_n)\right] \leq E\left[\frac{\bar{X}_n^4}{\varepsilon^4}\right] = \frac{1}{\varepsilon^4} E\left[\bar{X}_n^4\right],$$

でかつ

$$\begin{aligned} E\left[\bar{X}_n^4\right] &= \frac{1}{n^4} \sum_{i,j,k,l=1}^n E[X_i X_j X_k X_l] = \frac{1}{n^4} \left(nE[X_1^4] + \underbrace{\binom{4}{2} \binom{n}{2} E[X_1^2]^2}_{=3n(n-1)} \right) \\ &= \frac{\mu_4}{n^3} + \frac{3(n-1)\sigma^4}{n^3} \leq \frac{\mu_4 + 3\sigma^4}{n^2} \end{aligned}$$

なので,

$$\sum_{k=1}^{\infty} P(A_k) = \sum_{n=1}^{\infty} P(|\bar{X}_n| \geq \varepsilon) \leq \frac{\mu_4 + 3\sigma^4}{\varepsilon^4} \sum_{n=1}^{\infty} \frac{1}{n^2} < \infty.$$

これで示すべきことが示された。

証明終

注意: 要するに, Borel-Cantelliの補題から, 任意の $\varepsilon > 0$ について,

$$\sum_{n=1}^{\infty} P(|\bar{X}_n - \mu| \geq \varepsilon) < \infty$$

を示せれば, 大数の強法則が証明される。大数の弱法則の証明で示したことは

$$P(|\bar{X}_n - \mu| \geq \varepsilon) \leq \frac{\sigma^2}{\varepsilon^2} \frac{1}{n}$$

に過ぎず, $\sum_{n=1}^{\infty} 1/n = \infty$ と調和級数の和は無限大になるので, 大数の弱法則の証明の議論だけで大数の強法則を示すことはできない。

注意: $\zeta(s) = \sum_{n=1}^{\infty} 1/n^s$ (はRiemannのゼータ函数と呼ばれ, $s = 1$ では調和級数になって無限大に発散するが, $s > 1$ ならば有限の値に収束する。 $\zeta(2) = \sum_{n=1}^{\infty} 1/n^2$ を求めよという問題は **Basel問題** (バーゼル問題)と呼ばれ, 18世紀の超大数学者のEulerが解いた。答えは $\pi^2/6$ になる。その逆数の $6/\pi^2 = 0.6079271\dots$ は2つの整数の互いに素になる(最大公約数が 1 になる)「確率」に等しい。

In [3]: 1 6/ π^2

Out[3]: 0.6079271018540267

```

In [4]: 1 """
2 エラトステネスの篩(sieve of Eratosthenes)によって,
3 以下の条件を満たす長さ `N` のビットベクトル `IsPrime` を作成する:
4
5 >`IsPrime = make_isprime(N)` のとき `IsPrime[p]` は `p` が素数なら真になり, 素数でないなら偽に
6 """
7 function make_isprime(N)
8     IsPrime = trues(N)
9     # 1は素数ではない
10    IsPrime[1] = false
11    @inbounds for p in 2:isqrt(N)
12        if IsPrime[p]
13            for i in 2p:p:N
14                # 素数の非自明な倍数は素数ではない
15                IsPrime[i] = false
16            end
17        end
18    end
19    IsPrime
20 end
21 """
22 """
23 N以下の2つの正の整数が互いに素になる確率を篩法で求める.
24 """
25 function prob_mutually_prime(N)
26     IsPrime = make_isprime(N)
27     C = trues(N, N)
28     @inbounds for p in 2:N
29         if IsPrime[p]
30             for i in p:p:N
31                 for j in i:p:N
32                     # 共通の素因数 p を持つ i, j を除外する
33                     C[i, j] = false
34                     C[j, i] = false
35                 end
36             end
37         end
38     end
39     c = count(C)
40     c/N^2
41 end

```

Out[4]: prob_mutually_prime

In [5]: 1 @doc make_isprime

Out[5]: エラトステネスの篩(sieve of Eratosthenes)によって, 以下の条件を満たす長さ N のビットベクトル IsPrime を作成する:

IsPrime = make_isprime(N) のとき IsPrime[p] は p が素数なら真になり, 素数でないなら偽になる.

In [6]: 1 IsPrime = make_isprime(100)
2 primes = findall(IsPrime)
3 @show primes;

```
primes = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]
```

In [7]: 1 # 1億以下の素数のリストの作成も一瞬で終わる.
2 @time make_isprime(10^8);

```
0.536479 seconds (3 allocations: 11.921 MiB)
```

In [8]: 1 @time prob_mutually_prime(10000)

```
0.079873 seconds (5 allocations: 11.922 MiB)
```

Out[8]: 0.60794971

```
In [9]: 1 @time prob_mutually_prime(100000)
```

```
17.176402 seconds (5 allocations: 1.164 GiB, 0.05% gc time)
```

```
Out[9]: 0.6079301507
```

10万以下の2つの正の整数が互いに素な確率を求めてみたが、確かに理論値の $1/\zeta(2) = 0.6079271\cdots$ に近い値になっている。

1.6 大数の強法則に関する文献

大数の強法則は前節で採用した条件よりもずっと弱い条件のもとで成立している。そのようなことに興味がある人達は以下の文献を参照せよ。

- N. Etemadi, An elementary proof of the strong law of large numbers, Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete volume 55, pages 119–122 (1981) [[Springer \(<https://link.springer.com/article/10.1007/BF01013465#citeas>\)](https://link.springer.com/article/10.1007/BF01013465#citeas)] [[pdf \(<https://people.math.umass.edu/~yaoli/ptrl.pdf>\)](https://people.math.umass.edu/~yaoli/ptrl.pdf)]
- N. Lizia, A simple proof of the strong law of large numbers with rates, Bulletin of the Australian Mathematical Society, Volume 97, Issue 3, June 2018, pp. 513 - 517. [[doi \(<https://doi.org/10.1017/S0004972718000059>\)](https://doi.org/10.1017/S0004972718000059)] [[pdf \(<http://www.im.ufrj.br/nuno/SLLN.pdf>\)](http://www.im.ufrj.br/nuno/SLLN.pdf)]

X_1, X_2, X_3, \dots が同分布な確率変数の列であるとし、共通の期待値 $\mu = E[X_i]$ のwell-definedness ($E[|X_i|] < \infty$)を仮定する。

前者の文献では、 X_1, X_2, X_3, \dots に関する(独立性よりもずっと弱い)対ごとの独立性の仮定のもとで大数の強法則を証明している(分散の有限性も仮定していない)。

後者の文献では、 X_1, X_2, X_3, \dots の独立性と分散の有限性($\sigma^2 = \text{var}(X_i) < \infty$)のもとで、収束の速さが分かるような大数の強法則の証明を与えている。

1.7 問題: 大数の法則に関する誤解

当たりとはずれが毎回独立にどちらも 50% の確率で出るルーレットを回し続ける状況を考える。当たりが出たら、胴元から1万円もらい、はずれが出たら1万円支払うというルールになっているとする。この賭け事に参加しているある人がこう言った:

はずれの方が沢山出続けてたくさん負けてしまった! でも、確率論の大数の法則によって、これからは当たりの方が出易くなっている、負けは帳消しになるだろう。

この発言が間違っていることについて説明せよ。

解答例: 大数の法則は、はずれが沢山出た後に当たりの方が出易くなったり、当たりが沢山出たあとにはずれが出易くなることによって成立するのではない。

例えば、いきなり 10 回続けてはずれが出てしまったとする。

その後の90回分では、当たりとはずれが両方とも45回出たとする。そのとき、トータルでの当たりが出た割合は $55/100 = 55\%$ で 50% に近付く。

はずれ10回後の990回分でも、当たりとはずれがどちらも同じ回数の495回出たとする。そのとき、トータルでの当たりが出た割合は $505/1000 = 50.5\%$ でさらに 50% に近付く。

そのとき、最初に10回外れを引いて10万円負けていたその人は、大数の法則が予想する結果に近付いているにもかかわらず、1000回の勝負で最初に負けた10万円を取り戻せていない。

大数の法則は単に公平なギャンブルを続けると勝ちの回数の割合と負けの回数の割合が 50% に近付くことを主張しているだけで、沢山負けた後に勝ち易くなったり、その逆が起こることを何も主張していない。

解答終

他にも沢山の正しい解答例を作れると思います。

1.8 大数の法則のデモンストレーション

```
In [10]: 1 function plot_law_of_large_numbers(dist, N; μ = mean(dist), kwargs...)
2     X = rand(dist, N)
3     Ī = cumsum(X) ./ (1:N)
4     plot(1:N, Ī; label="")
5     hline!([μ]; label="")
6     plot!(; kwargs...)
7 end
8
9 function plot_law_of_large_numbers_6x3(dist, N;
10     μ = mean(dist), seed=4649363, title="", kwargs...)
11     Random.seed!(seed)
12     PP = []
13     for _ in 1:18
14         P = plot_law_of_large_numbers(dist, N; μ, kwargs...)
15         push!(PP, P)
16     end
17     P0 = plot(; title,
18                 framesstyle=nothing, showaxis=false, ticks=false, margin=0Plots.mm)
19     layout = @layout [a{0.005h}; grid(6, 3)]
20     plot(P0, PP...; size=(800, 800), layout,
21           titlefontsize=12, guidefontsize=8, tickfontsize=6)
22 end
```

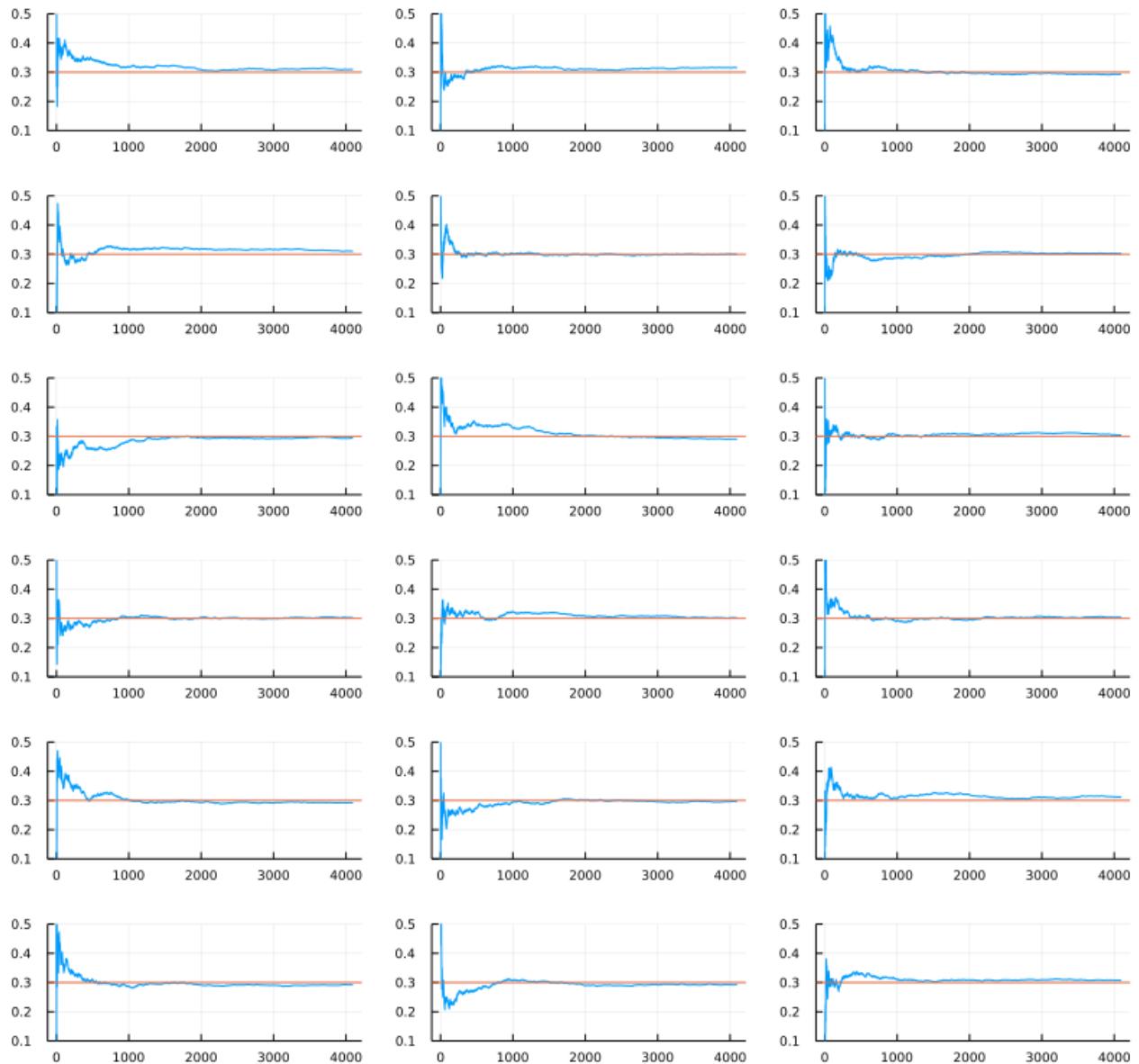
Out[10]: plot_law_of_large_numbers_6x3 (generic function with 1 method)

1.8.1 Bernoulli試行

```
In [11]: 1 plot_law_of_large_numbers_6x3(Bernoulli(0.3), 2^12;  
2 title="Law of large numbers for Bernoulli(0.3)", ylim=(0.1, 0.5))
```

Out[11]:

Law of large numbers for Bernoulli(0.3)

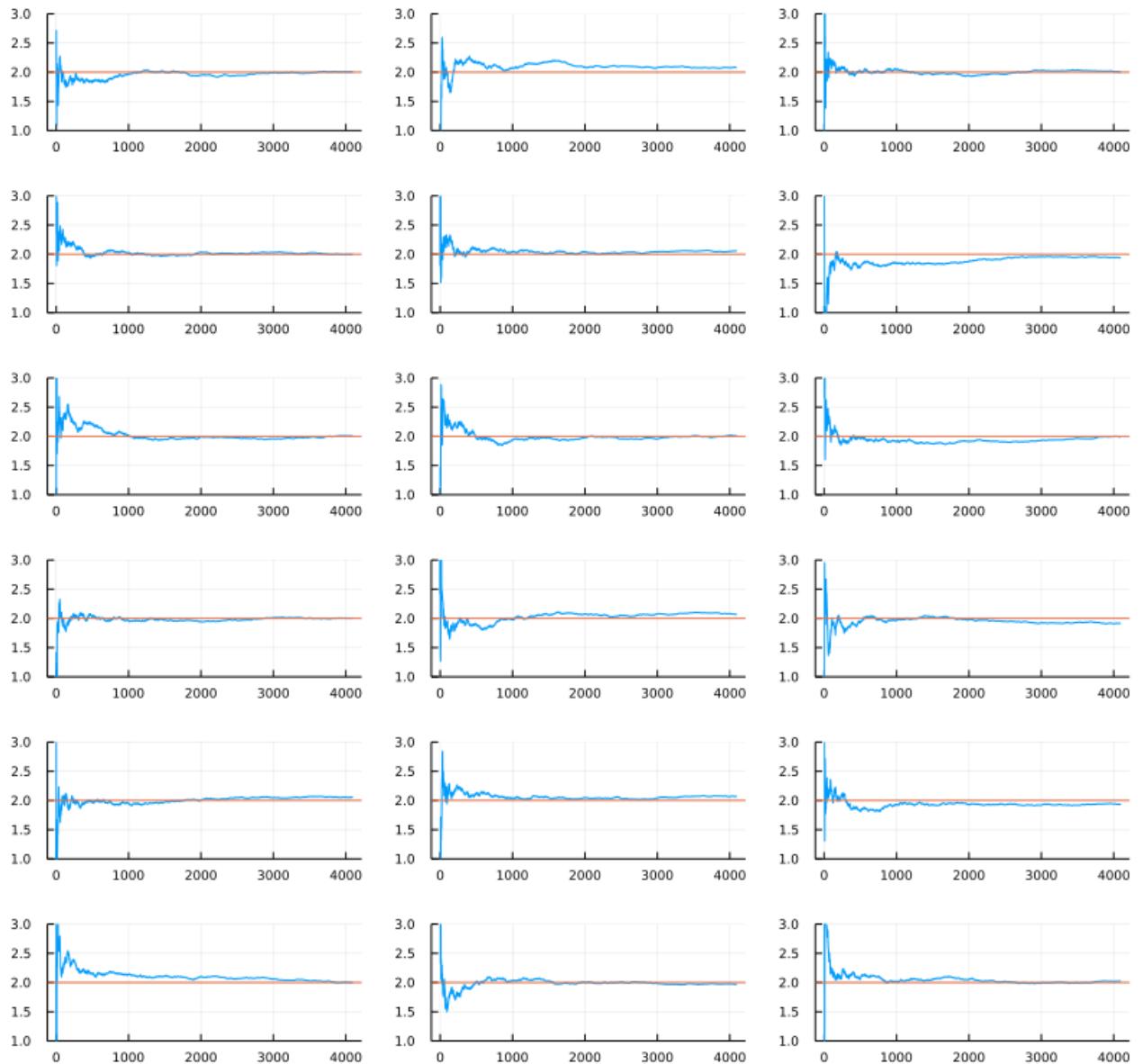


1.8.2 正規分布のサンプルの場合

```
In [12]: 1 plot_law_of_large_numbers_6x3(Normal(2, 3), 2^12;  
2 title="Law of large numbers for Normal(2, 3)", ylim=(1, 3))
```

Out[12]:

Law of large numbers for Normal(2, 3)

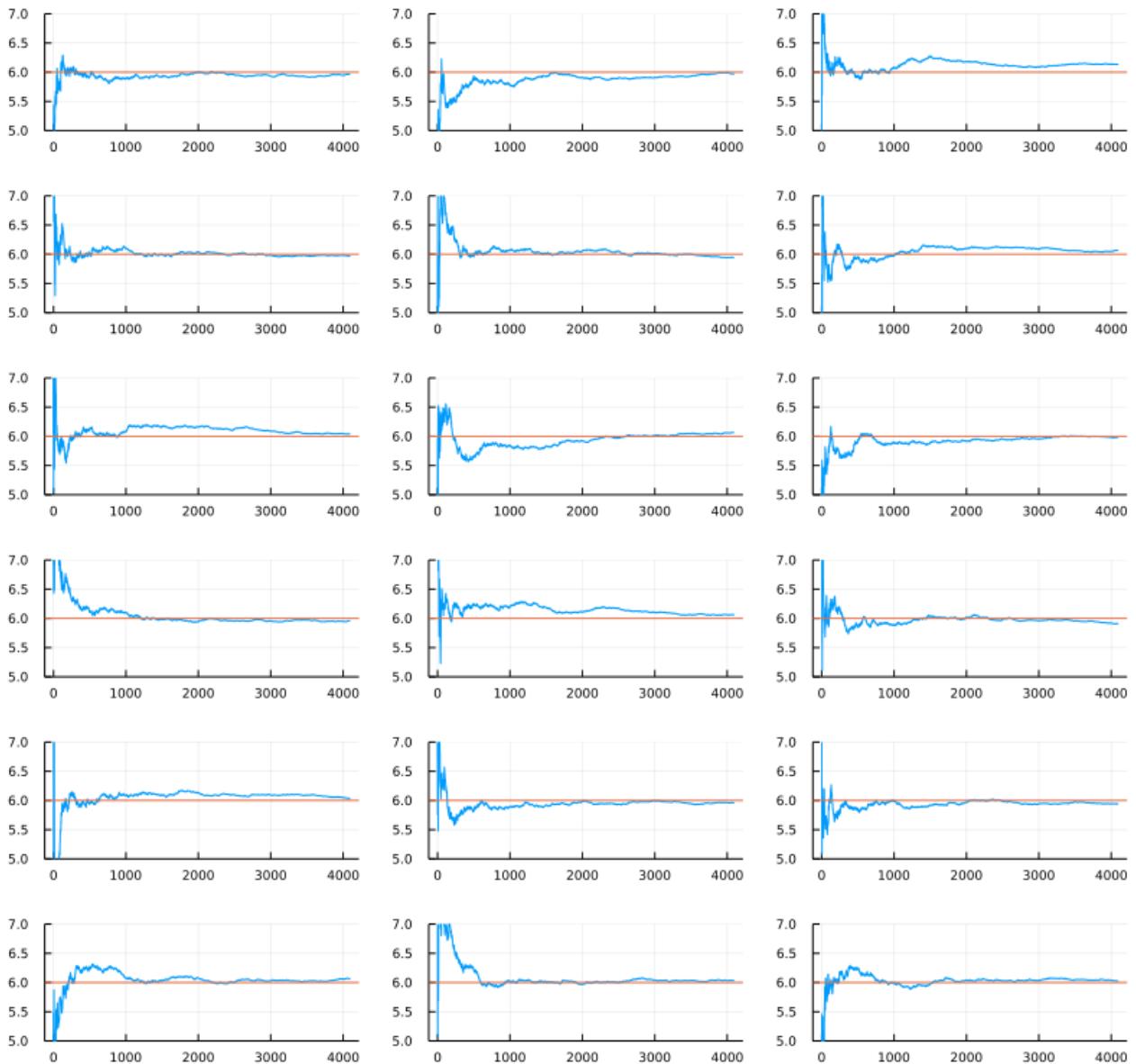


1.8.3 ガンマ分布の場合

```
In [13]: 1 plot_law_of_large_numbers_6x3(Gamma(2, 3), 2^12;
2         title="Law of large numbers for Gamma(2, 3)", ylim=(5, 7))
```

Out[13]:

Law of large numbers for Gamma(2, 3)



1.8.4 大数の法則が成立しない場合 (Cauchy分布)

確率密度函数が

$$p(x) = \frac{1}{\pi} \frac{1}{1+x^2}$$

のCauchy分布 Cauchy() に従う確率変数 X については

$$E[|X|] = \frac{2}{\pi} \int_0^\infty \frac{x}{1+x^2} dx = \infty$$

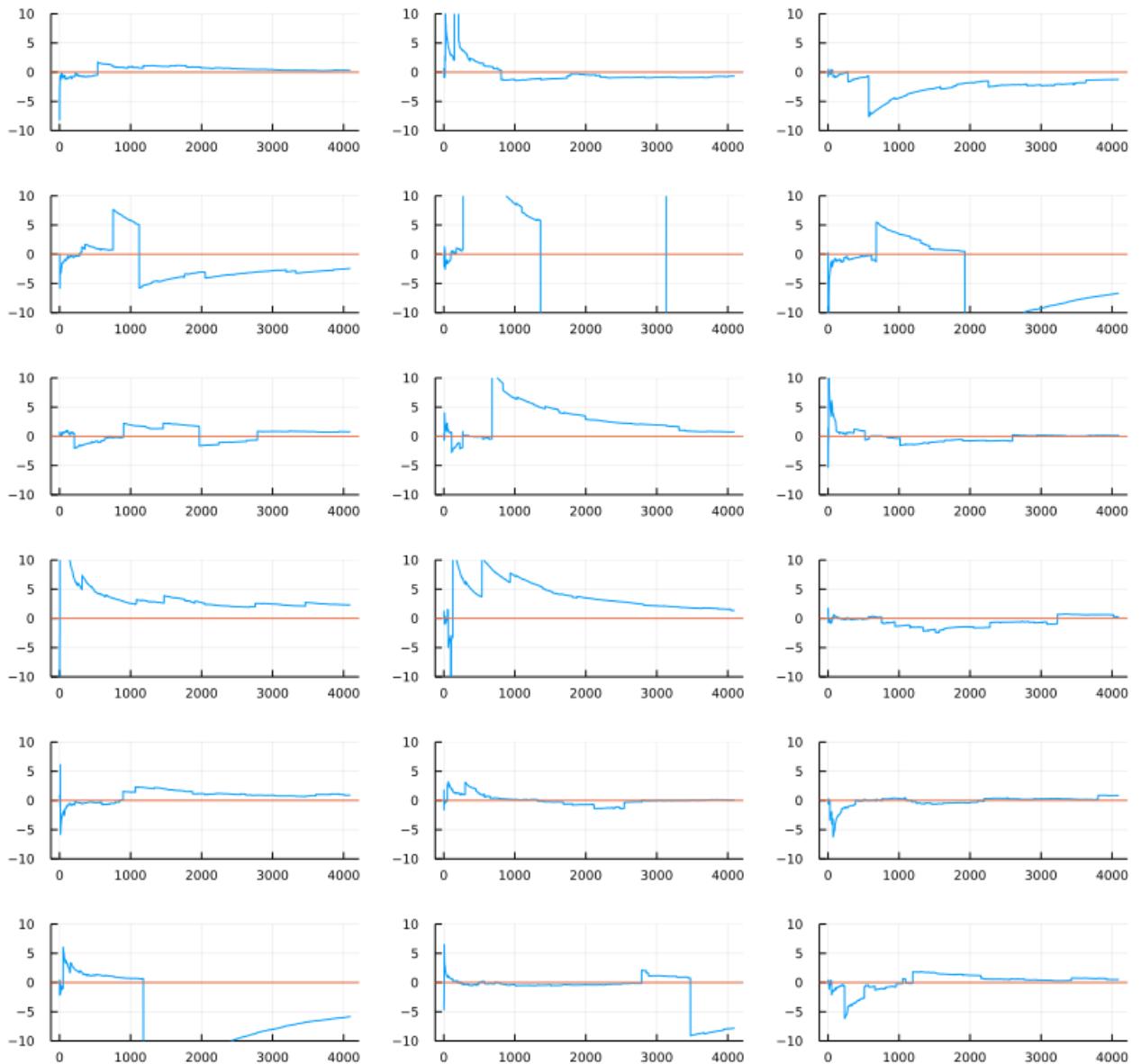
なので、期待値 $E[X]$ が定義されない。ゆえに大数の法則も成立しない。

In [14]:

```
1 plot_law_of_large_numbers_6x3(Cauchy(), 2^12; μ = 0.0,
2     title="Law of large numbers for Cauchy() does not hold", ylim=(-10, 10))
```

Out[14]:

Law of large numbers for Cauchy() does not hold



注意: X_1, X_2, \dots, X_n がCauchy分布のサイズ n の標本であるとき, その標本平均 \bar{X} もCauchy分布に従うことを示せる. (証明したい人はCauchy分布の特性函数が

$$\varphi(t) = E[e^{itX}] = e^{-|t|} \quad (X \sim \text{Cauchy}())$$

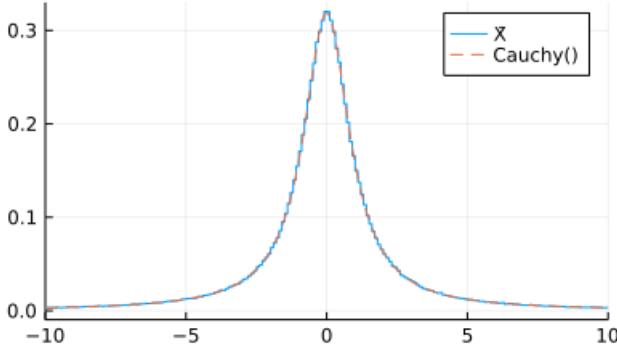
となることを示せ.)

```
In [15]: 1 function plot_cauchy(n; dist=Cauchy(), L=10^6, xlim=(-10, 10), kwargs...)
2     X = Vector{Float64}(undef, L)
3     tmp = [Vector{Float64}(undef, n) for _ in 1:Threads.nthreads()]
4     Threads.@threads for i in 1:L
5         X = rand!(dist, tmp[Threads.threadid()])
6         X[i] = mean(X)
7     end
8     a, b = quantile.(Ref(X), (0.001, 0.999))
9     stephist(X; bin=a:0.1:b, norm=true, label="X")
10    plot!(Cauchy(), xlim...; label="Cauchy()", ls=:dash)
11    title!("Sample means of n = $n size samples of Cauchy()")
12    plot!(; xlim, kwargs...)
13 end
```

Out[15]: `plot_cauchy` (generic function with 1 method)

In [16]: 1 `plot_cauchy(10)`

Out[16]: Sample means of n = 10 size samples of Cauchy()



2 二項分布の中心極限定理

2.1 二項分布の中心極限定理の内容

二項分布 $\text{Binomial}(n, p)$ が正規分布 $\text{Normal}(np, \sqrt{np(1-p)})$ で近似されることを **二項分布の中心極限定理** と呼ぶことにしよう。(ただしその近似の精度をよくするためには np と $n(1-p)$ をある程度以上に大きくする必要がある。「大きくする」と言っても粗い近似でよければ3から5程度でも十分な場合がある。)

二項分布の中心極限定理には以下のような様々な言い換えがある。 $0 < p < 1$ と仮定し, n は十分に大きいと仮定する。そのとき,

(1) $K \sim \text{Binomial}(n, p)$ ならば K の分布は平均 np , 分散 $np(1-p)$ の正規分布で近似される。

(2) $K \sim \text{Binomial}(n, p)$ ならば $\frac{K}{n}$ の分布は平均 p , 分散 $p(1-p)/n$ の正規分布で近似される。

(3) $K \sim \text{Binomial}(n, p)$ ならば $\frac{K - np}{\sqrt{n}}$ の分布は平均 0, 分散 $p(1-p)$ の正規分布で近似される。

(4) $K \sim \text{Binomial}(n, p)$ ならば $\frac{K - np}{\sqrt{np(1-p)}}$ の分布は標準正規分布で近似される。

例えば, (1)の結果は, $K \sim \text{Binomial}(n, p)$, $X \sim \text{Normal}(np, \sqrt{np(1-p)})$ のとき,

$$P(K \leq a) \approx P(X \leq a)$$

という近似を使えることを意味する。(2),(3),(4)についても同様のことが言える。これを使うと、二項分布における確率を対応する正規分布の確率で置き換えて近似計算できる。二項分布に限らず、正規分布で近似できる確率分布であれば、どのような分布であっても確率の計算を正規分布という一種類の分布における確率で近似的に計算できる。

例えば, (4)の結果は, $K \sim \text{Binomial}(n, p)$ のとき,

$$E \left[f \left(\frac{K - np}{\sqrt{np(1-p)}} \right) \right] = \sum_{k=0}^n f \left(\frac{k - np}{\sqrt{np(1-p)}} \right) \binom{n}{k} p^k (1-p)^{n-k}$$

が, $n \rightarrow \infty$ で, $Z \sim \text{Normal}(0, 1)$ に関する

$$E[f(Z)] = \int_{-\infty}^{\infty} f(z) \frac{e^{-z^2/2}}{\sqrt{2\pi}} dz.$$

に収束することと同じことだと考えてよい。(3)についても同様のことが言える。

2.2 二項分布の中心極限定理のプロット

以下は、試行回数 n 、成功確率 p の二項分布(期待値と分散はそれぞれ np , $np(1-p)$ になる)の確率質量函数と平均 np , 分散 $np(1-p)$ を持つ正規分布の確率密度函数の同時プロットである。それらを見れば、証明しなくても、二項分布の中心極限定理が成立していることは明らかだろう。(証明する前にコンピュータでグラフを確認するべき!)

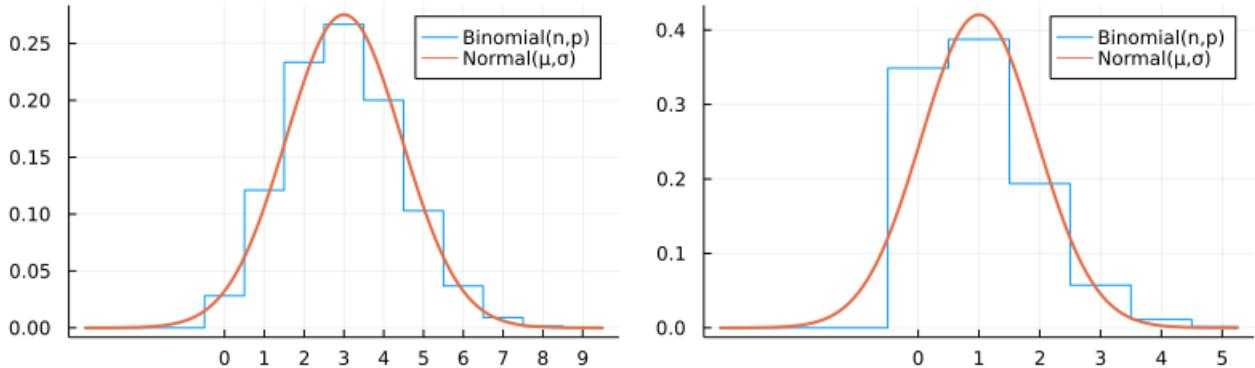
注意: 以下のグラフを見れば、二項分布の正規分布近似の精度を高めるためには、 p が小さいほど n を大きくする必要があることがある。 $(p$ が 1 に近い場合にも n を大きくする必要がある。)

```
In [17]: 1 function plot_binomial_clt(n, p, s = 1; c = 4.5)
2     μ, σ = n*p, sqrt(n*p*(1-p))
3     xlim = (μ-c*σ, μ+c*σ)
4     plot(x → pdf(Binomial(n, p), round(x)), xlim..., label="Binomial(n,p)")
5     plot!(Normal(μ, σ), xlim..., label="Normal(μ,σ)", lw=2)
6     # plot!(x → pdf(Poisson(μ), round(x)), xlim..., label="Poisson(μ)", ls=:dash)
7     title!("n = $n, p = $p, μ=np, σ²=np(1-p)")
8     plot!(; xtick=0:s:n)
9 end
```

Out[17]: `plot_binomial_clt` (generic function with 2 methods)

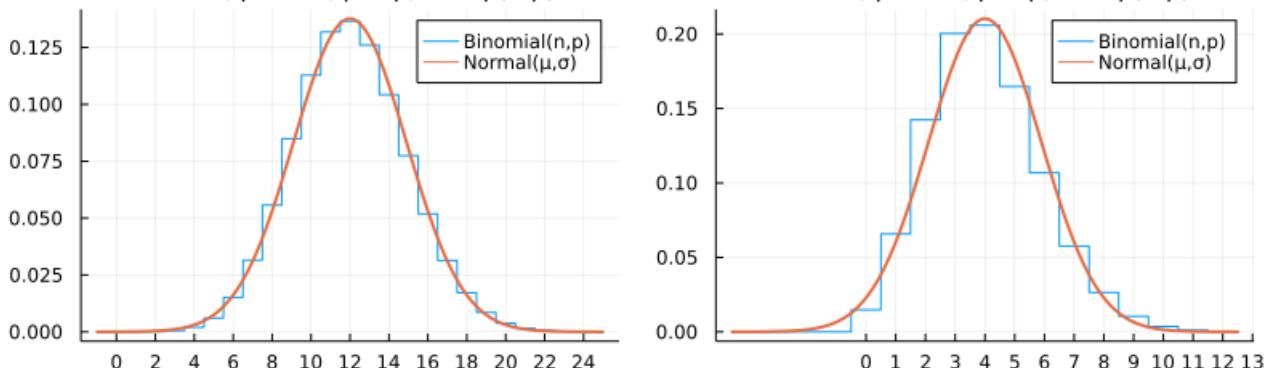
```
In [18]: 1 plot(plot_binomial_clt(10, 0.3), plot_binomial_clt(10, 0.1); size=(800, 250))
```

Out[18]: $n = 10, p = 0.3, \mu=np, \sigma^2=np(1-p)$ $n = 10, p = 0.1, \mu=np, \sigma^2=np(1-p)$



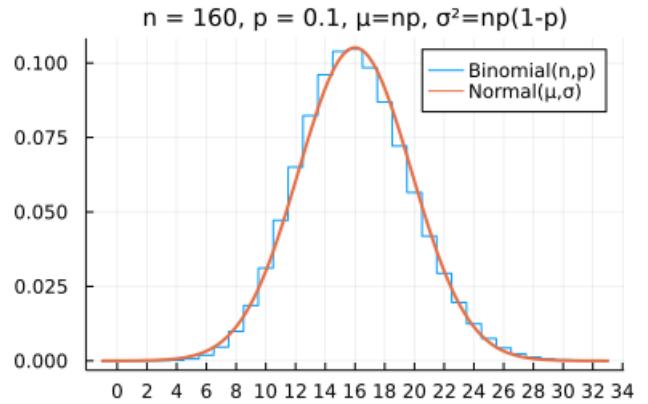
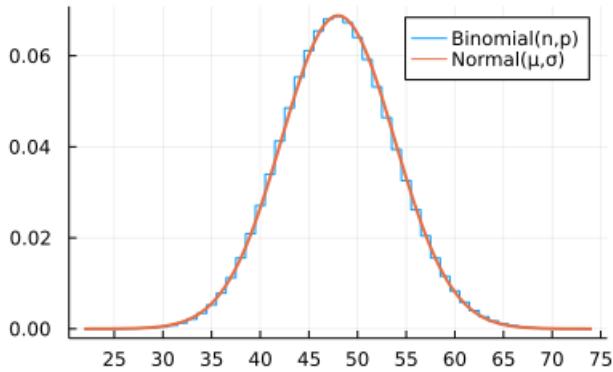
```
In [19]: 1 plot(plot_binomial_clt(40, 0.3, 2), plot_binomial_clt(40, 0.1); size=(800, 250))
```

Out[19]: $n = 40, p = 0.3, \mu=np, \sigma^2=np(1-p)$ $n = 40, p = 0.1, \mu=np, \sigma^2=np(1-p)$



```
In [20]: 1 plot(plot_binomial_clt(160, 0.3, 5), plot_binomial_clt(160, 0.1, 2); size=(800, 250))
```

```
Out[20]: n = 160, p = 0.3, mu=np, sigma^2=np(1-p)
```



2.3 必修重要問題: 二項分布の正規分布近似を使った確率の近似計算1

確率変数 K は二項分布 $\text{Binomial}(100, 1/2)$ に従っていると仮定する。

(1) 確率 $P(K \leq 40)$ を小数点以下第5桁まで求めよ。

(2) 二項分布 $\text{Binomial}(100, 1/2)$ と同じ期待値と分散を持つ正規分布に従う確率変数 X について確率 $P(X \leq 40)$ を小数点以下第5桁まで求めよ。

解答例:

(1) $K \sim \text{Binomial}(100, 1/2)$ のとき,

$$P(K \leq 40) \approx 0.02844$$

(2) 一般に $\text{Binomial}(n, p)$ の平均と分散はそれぞれ $\mu = np$, $\sigma^2 = np(1 - p)$ なので, $n = 100$, $p = 1/2$ の場合は $\mu = 50$, $\sigma^2 = 25$, $\sigma = 5$ なので, $X \sim \text{Normal}(50, 5)$ で

$$P(X \leq 40) \approx 0.02275$$

解答終

2.3.1 WolframAlphaによる計算1

(1) [cdf\(BinomialDistribution\(n..p\), 40\) where n=100..p=0.5](https://www.wolframalpha.com/input/?i=cdf%28BinomialDistribution%28n..p%29,40%29+where+n%3D100%2Cp%3D0.5%29) (<https://www.wolframalpha.com/input/?i=cdf%28BinomialDistribution%28n..p%29,40%29+where+n%3D100%2Cp%3D0.5%29>)

(2) [cdf\(NormalDistribution\(50..5\), 40\)](https://www.wolframalpha.com/input/?i=cdf%28NormalDistribution%2850..5,40%29%29) (<https://www.wolframalpha.com/input/?i=cdf%28NormalDistribution%2850..5,40%29%29>)

2.3.2 Julia言語による計算1

```
In [21]: 1 n, p = 100, 1/2
```

```
Out[21]: (100, 0.5)
```

```
In [22]: 1 # (1)
2 p_B = cdf(Binomial(n, p), 40)
```

```
Out[22]: 0.028443966820490434
```

```
In [23]: 1 # (2)
          2 q_N = cdf(Normal(n*p, sqrt(n*p*(1-p))), 40)
```

```
Out[23]: 0.022750131948179205
```

2.3.3 Julia言語による視覚化1

In [24]:

```

1 # (1)
2 plot(x → mypdf(Binomial(n, p), x), 30, 70; label="", c=1)
3 vline!([n*p]; label="np", c=1)
4 plot!(x → mypdf(Binomial(n, p), x), 30, 40.5;
5     label="", fillrange=0, c=1, alpha=0.5)
6 annotate!(35, 0.01, text(@sprintf("%0.5f", p_B), 10, :blue))
7 vline!([40]; label="", c=:gray)
8 title!("Binomial(n = $n, p = $p)")

```

UndefVarError: mypdf not defined

Stacktrace:

```

[1] (::var"#10#11")(x::Float64)
@ Main .\In[24]:2
[2] (::ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#10#11"})(x::Float64; kw::Base.
Pairs{Symbol, Union{}, Tuple{}, NamedTuple{(), Tuple{}}})
@ Base .\operators.jl:1085
[3] (::ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#10#11"})(x::Float64)
@ Base .\operators.jl:1085
[4] (::ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#10#11"}, Reci
pesPipeline.var"#9#10"\{Symbol}})(x::Float64; kw::Base.Pairs{Symbol, Union{}, Tuple{}, NamedTuple
{(), Tuple{}}})
@ Base .\operators.jl:1085
[5] ComposedFunction
@ .\operators.jl:1085 [inlined]
[6] (::PlotUtils.var"#27#29"\{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"\{Symbo
l}, var"#10#11"}, RecipesPipeline.var"#9#10"\{Symbol}}})(x::Float64)
@ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:49
[7] _broadcast_getindex_evalf
@ .\broadcast.jl:670 [inlined]
[8] _broadcast_getindex
@ .\broadcast.jl:643 [inlined]
[9] getindex
@ .\broadcast.jl:597 [inlined]
[10] copy
@ .\broadcast.jl:899 [inlined]
[11] materialize(bc::Base.Broadcast.Broadcasted{Base.Broadcast.DefaultDisplayStyle{1}, Nothing, P
lotUtils.var"#27#29"\{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#1
0#11"}, RecipesPipeline.var"#9#10"\{Symbol}}}, Tuple{Vector{Float64}}})
@ Base.Broadcast .\broadcast.jl:860
[12] adapted_grid(f::Any, minmax::Tuple{Float64, Float64}; max_recursions::Int64, max_curvature
e::Float64, n_points::Int64)
@ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:60
[13] adapted_grid(f::Any, minmax::Tuple{Float64, Float64})
@ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:21
[14] _scaled_adapted_grid(f::Function, xscale::Symbol,yscale::Symbol, xmin::Int64, xmax::Int6
4)
@ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:353
[15] macro expansion
@ D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:293 [inlined]
[16] apply_recipe(plotattributes::AbstractDict{Symbol, Any}, f::Function, xmin::Number, xmax::N
umber)
@ RecipesPipeline D:\.julia\packages\RecipesBase\qpxEX\src\RecipesBase.jl:289
[17] _process_userrecipes!(plt::Any, plotattributes::Any, args::Any)
@ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:36
[18] recipe_pipeline!(plt::Any, plotattributes::Any, args::Any)
@ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\RecipesPipeline.jl:70
[19] _plot!(plt::Plots.Plot, plotattributes::Any, args::Any)
@ Plots D:\.julia\packages\Plots\nzdhU\src\plot.jl:208
[20] #plot#139
@ D:\.julia\packages\Plots\nzdhU\src\plot.jl:91 [inlined]
[21] top-level scope
@ In[24]:2
[22] eval
@ .\boot.jl:373 [inlined]
[23] include_string(mapexpr::typeof(REPL.softscope), mod::Module, code::String, filename::Stri
ng)
@ Base .\loading.jl:1196

```

WARNING: importing deprecated binding Colors.RGB1 into PlotUtils.

WARNING: importing deprecated binding Colors.RGB4 into PlotUtils.

In [25]:

```

1 # (2)
2 plot(x → mypdf(Normal(n*p, √(n*p*(1-p)))), x), 30, 70; label="", c=:blue, ls=:dash)
3 vline!([n*p]; label="np", ls=:dashdot, c=:blue)
4 plot!(x → mypdf(Normal(n*p, √(n*p*(1-p)))), x), 30, 40;
5     label="", fillrange=0, c=:blue, alpha=0.5)
6 annotate!(35, 0.01, text(@sprintf("%0.5f", q_N), 10, :blue))
7 vline!([40]; label="", c=:gray)
8 title!("Normal( $\mu = 100 \times p$ ,  $\sigma = \sqrt{n \cdot p \cdot (1-p)}$ )")

```

UndefVarError: mypdf not defined

Stacktrace:

```

[1] (::var"#12#13")(x::Float64)
@ Main .\In[25]:2
[2] (::ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#12#13"})(x::Float64; kw::Base.Pairs{Symbol, Union{}, Tuple{}, NamedTuple{(), Tuple{}}})
@ Base .\operators.jl:1085
[3] (::ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#12#13"})(x::Float64)
@ Base .\operators.jl:1085
[4] (::ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#12#13"}, RecipesPipeline.var"#9#10"{Symbol}})(x::Float64; kw::Base.Pairs{Symbol, Union{}, Tuple{}, NamedTuple{(), Tuple{}}})
@ Base .\operators.jl:1085
[5] ComposedFunction
@ .\operators.jl:1085 [inlined]
[6] (::PlotUtils.var"#27#29"{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#12#13"}, RecipesPipeline.var"#9#10"{Symbol}}})(x::Float64)
@ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:49
[7] _broadcast_getindex_evalf
@ .\broadcast.jl:670 [inlined]
[8] _broadcast_getindex
@ .\broadcast.jl:643 [inlined]
[9] getindex
@ .\broadcast.jl:597 [inlined]
[10] copy
@ .\broadcast.jl:899 [inlined]
[11] materialize(bc::Base.Broadcast.Broadcasted{Base.Broadcast.DefaultArrayStyle{1}, Nothing, PlotUtils.var"#27#29"{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#12#13"}, RecipesPipeline.var"#9#10"{Symbol}}}}, Tuple{Vector{Float64}})
@ Base.Broadcast .\broadcast.jl:860
[12] adapted_grid(f::Any, minmax::Tuple{Float64, Float64}; max_recursions::Int64, max_curvature::Float64, n_points::Int64)
@ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:60
[13] adapted_grid(f::Any, minmax::Tuple{Float64, Float64})
@ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:21
[14] _scaled_adapted_grid(f::Function, xscale::Symbol, yscale::Symbol, xmin::Int64, xmax::Int64)
@ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:353
[15] macro expansion
@ D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:293 [inlined]
[16] apply_recipe(plotattributes::AbstractDict{Symbol, Any}, f::Function, xmin::Number, xmax::Number)
@ RecipesPipeline D:\.julia\packages\RecipesBase\qpxEX\src\RecipesBase.jl:289
[17] _process_userrecipes!(plt::Any, plotattributes::Any, args::Any)
@ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:36
[18] recipe_pipeline!(plt::Any, plotattributes::Any, args::Any)
@ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\RecipesPipeline.jl:70
[19] _plot!(plt::Plots.Plot, plotattributes::Any, args::Any)
@ Plots D:\.julia\packages\Plots\nzdhU\src\plot.jl:208
[20] #plot#139
@ D:\.julia\packages\Plots\nzdhU\src\plot.jl:91 [inlined]
[21] top-level scope
@ In[25]:2
[22] eval
@ .\boot.jl:373 [inlined]
[23] include_string(mapexpr::typeof(REPL.softscope), mod::Module, code::String, filename::String)
@ Base .\loading.jl:1196

```

In [26]:

```

1 # (1) and (2)
2 plot(x → mypdf(Binomial(n, p), x), 30, 70; label="", c=1)
3 vline!([n*p]; label="np", c=1)
4
5 plot!(x → mypdf(Normal(n*p, √(n*p*(1-p))), x), 30, 70;
6   label="", c=:blue, ls=:dash)
7 #vline!([n*p]; label="np", ls=:dashdot, c=:blue)
8
9 vline!([40]; label="", c=:gray)
10 title!("Binomial(n = $n, p = $p) and normal approx")

```

UndefVarError: mypdf not defined

Stacktrace:

```

[1] (::var"#14#15")(x::Float64)
@ Main .\In[26]:2
[2] (::ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#14#15"})(x::Float64; kw::Base.Pairs{Symbol, Union{}, Tuple{}}, NamedTuple{(), Tuple{}})
@ Base .\operators.jl:1085
[3] (::ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#14#15"})(x::Float64)
@ Base .\operators.jl:1085
[4] (::ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#14#15"}, RecipesPipeline.var"#9#10"\{Symbol}})(x::Float64; kw::Base.Pairs{Symbol, Union{}, Tuple{}}, NamedTuple{(), Tuple{}})
@ Base .\operators.jl:1085
[5] ComposedFunction
@ .\operators.jl:1085 [inlined]
[6] (::PlotUtils.var"#27#29"\{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#14#15"}, RecipesPipeline.var"#9#10"\{Symbol}}})(x::Float64)
@ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:49
[7] _broadcast_getindex_evalf
@ .\broadcast.jl:670 [inlined]
[8] _broadcast_getindex
@ .\broadcast.jl:643 [inlined]
[9] getindex
@ .\broadcast.jl:597 [inlined]
[10] copy
@ .\broadcast.jl:899 [inlined]
[11] materialize(bc::Base.Broadcast.Broadcasted{Base.Broadcast.DefaultArrayStyle{1}, Nothing, PlotUtils.var"#27#29"\{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#14#15"}, RecipesPipeline.var"#9#10"\{Symbol}}}, Tuple{Vector{Float64}}})
@ Base.Broadcast .\broadcast.jl:860
[12] adapted_grid(f::Any, minmax::Tuple{Float64, Float64}; max_recursions::Int64, max_curvature::Float64, n_points::Int64)
@ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:60
[13] adapted_grid(f::Any, minmax::Tuple{Float64, Float64})
@ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:21
[14] _scaled_adapted_grid(f::Function, xscale::Symbol,yscale::Symbol, xmin::Int64, xmax::Int64)
@ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:353
[15] macro expansion
@ D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:293 [inlined]
[16] apply_recipe(plotattributes::AbstractDict{Symbol, Any}, f::Function, xmin::Number, xmax::Number)
@ RecipesPipeline D:\.julia\packages\RecipesBase\qpxEX\src\RecipesBase.jl:289
[17] _process_userrecipes!(plt::Any, plotattributes::Any, args::Any)
@ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:36
[18] recipe_pipeline!(plt::Any, plotattributes::Any, args::Any)
@ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\RecipesPipeline.jl:70
[19] _plot!(plt::Plots.Plot, plotattributes::Any, args::Any)
@ Plots D:\.julia\packages\Plots\nzdhU\src\plot.jl:208
[20] #plot#139
@ D:\.julia\packages\Plots\nzdhU\src\plot.jl:91 [inlined]
[21] top-level scope
@ In[26]:2
[22] eval
@ .\boot.jl:373 [inlined]
[23] include_string(mapexpr::typeof(REPL.softscope), mod::Module, code::String, filename::String)
@ Base .\loading.jl:1196

```

2.4 必修重要問題: 二項分布の正規分布近似を使った確率の近似計算2

$0 < p < 1$ であると仮定し, K は二項分布 $\text{Binomial}(100, p)$ に従う確率変数であるとする.

(1) $P(K \leq 40) = 0.025$ を満たす p の値 p_U を小数点以下第4桁まで求めよ.

(2) $P(K \geq 40) = 0.025$ を満たす p の値 p_L を小数点以下第4桁まで求めよ.

$\hat{p} = 40/100 = 0.4$ とおく. X は K と同じ平均 np と推定された分散 $n\hat{p}(1 - \hat{p})$ を持つ正規分布に従う確率変数であるとする.

(3) $P(X \leq 40) = 0.025$ を満たす p の値 q_U を小数点以下第4桁まで求めよ.

(4) $P(X \geq 40) = 0.025$ を満たす p の値 q_L を小数点以下第4桁まで求めよ.

注意: これらによって計算される区間 $[p_L, p_U]$ と $[q_L, q_U]$ はデータ「100回中40回当たり」というデータから得られる p の95%信頼区間と呼ばれる. 前者は Clopper-Pearsonの信頼区間 と, 後者は Waldの信頼区間 と呼ばれる.

解答例: $n = 100, k = 40$ とおく.

累積分布函数を cdf と書き, その逆函数の分位点函数を quantile と書くことにする: X が分布 D に従う確率変数であるとき,

$$\begin{aligned} \text{cdf}(D, x) &= P(X \leq x), \\ \text{quantile}(D, a) &= (\text{the value } x \text{ satisfying } P(X \leq x) = a). \end{aligned}$$

以上の記号法のもとで,

$$\begin{aligned} F_K(p) &= P(K \leq k) = \text{cdf}(\text{Binomial}(n, p), k) \\ &= 1 - \text{cdf}(\text{Beta}(k+1, n-k), p), \\ G_K(p) &= P(K \geq k) = 1 - \text{cdf}(\text{Binomial}(n, p), k-1) \\ &= \text{cdf}(\text{Beta}(k, n-k+1), p) \end{aligned}$$

について,

$$\begin{aligned} F_K^{-1}(\alpha/2) &= \text{quantile}(\text{Beta}(k+1, n-k), 1-\alpha/2), \\ G_K^{-1}(\alpha/2) &= \text{quantile}(\text{Beta}(k, n-k+1), \alpha/2). \end{aligned}$$

これらを $n = 100, k = 40, \alpha = 0.05$ に適用すると,

(1) $p_U \approx 0.5028$

(2) $p_L \approx 0.3033$

$X \sim \text{Normal}(np, \sigma)$ のとき,

$$\begin{aligned} F_X(p) &= P(X \leq k) = \text{cdf}(\text{Normal}(np, \sigma), k) \\ &= \text{cdf}(\text{Normal}(0, 1), (k-np)/\sigma), \\ G_X(p) &= P(X \geq k) = 1 - \text{cdf}(\text{Normal}(np, \sigma), k) \\ &= 1 - \text{cdf}(\text{Normal}(0, 1), (k-np)/\sigma) \end{aligned}$$

について

$$z_{\alpha/2} = \text{quantile}(\text{Normal}(0, 1), 1-\alpha/2)$$

とおくと, $\text{quantile}(\text{Normal}(0, 1), \alpha/2) = -z_{\alpha/2}$ なので,

$$\begin{aligned} F_X^{-1}(\alpha/2) &= \frac{k - \sigma \text{quantile}(\text{Normal}(0, 1), \alpha/2)}{n} = \frac{k + z_{\alpha/2}\sigma}{n} \\ G_X^{-1}(1-\alpha/2) &= \frac{k - \sigma \text{quantile}(\text{Normal}(0, 1), 1-\alpha/2)}{n} = \frac{k - z_{\alpha/2}\sigma}{n} \end{aligned}$$

$n = 100, \hat{p} = 0.4$ のとき,

$$\hat{\sigma}^2 = np(1 - \hat{p}) = 24.$$

であり, 上の公式を $n = 100, k = 40, \sigma = \hat{\sigma}$ の場合に使うと,

(3) $q_U \approx 0.4960$

(4) $q_L \approx 0.3040$

解答終

2.4.1 WolframAlphaによる計算

(1) [\(https://www.wolframalpha.com/input?
i=quantile%28BetaDistribution%2841%2C+60%29%2C+0.975%29\)](https://www.wolframalpha.com/input/?i=quantile(BetaDistribution(41, 60), 0.975))

(1) 検算 [cdf\(BinomialDistribution\(100, 0.502791\), 40\)](https://www.wolframalpha.com/input/?i=cdf%28BinomialDistribution%28100%2C+0.502791%29%2C+40%29) (<https://www.wolframalpha.com/input/?i=cdf%28BinomialDistribution%28100%2C+0.502791%29%2C+40%29>)

(2) [quantile\(BetaDistribution\(40, 61\), 0.025\)](https://www.wolframalpha.com/input/?i=quantile%28BetaDistribution%2840, 61%29, 0.025%29) (<https://www.wolframalpha.com/input/?i=quantile%28BetaDistribution%2840, 61%29, 0.025%29>)

(2) 検算 [1 - cdf\(BinomialDistribution\(100, 0.303295\), 39\)](https://www.wolframalpha.com/input/?i=1-cdf%28BinomialDistribution%28100, 0.303295%29, 39%29) (<https://www.wolframalpha.com/input/?i=1-cdf%28BinomialDistribution%28100, 0.303295%29, 39%29>)

[quantile\(NormalDistribution\(0, 1\), 0.975\)](https://www.wolframalpha.com/input/?i=quantile%28NormalDistribution%280, 1%29, 0.975%29) (<https://www.wolframalpha.com/input/?i=quantile%28NormalDistribution%280, 1%29, 0.975%29>) → 1.95996

(3) [\(40 + 1.95996 * sqrt\(24\)\)/100](https://www.wolframalpha.com/input/?i=(40+1.95996*sqrt%2824%29)/100) ([https://www.wolframalpha.com/input/?i=\(40+1.95996*sqrt%2824%29\)/100](https://www.wolframalpha.com/input/?i=(40+1.95996*sqrt%2824%29)/100))

(3) 検算 [cdf\(NormalDistribution\(49.6018, sqrt\(24\)\), 40\)](https://www.wolframalpha.com/input/?i=cdf%28NormalDistribution%2849.6018, sqrt%2824%29, 40%29%29) (<https://www.wolframalpha.com/input/?i=cdf%28NormalDistribution%2849.6018, sqrt%2824%29, 40%29%29>)

(4) [\(40 - 1.95996 * sqrt\(24\)\)/100](https://www.wolframalpha.com/input/?i=(40-1.95996*sqrt%2824%29)/100) ([https://www.wolframalpha.com/input/?i=\(40-1.95996*sqrt%2824%29\)/100](https://www.wolframalpha.com/input/?i=(40-1.95996*sqrt%2824%29)/100))

(4) 検算 [1 - cdf\(NormalDistribution\(30.3982, sqrt\(24\)\), 40\)](https://www.wolframalpha.com/input/?i=1-cdf%28NormalDistribution%2830.3982, sqrt%2824%29, 40%29%29) (<https://www.wolframalpha.com/input/?i=1-cdf%28NormalDistribution%2830.3982, sqrt%2824%29, 40%29%29>)

2.4.2 Julia言語による計算2

`ccdf(d, x) = 1 - cdf(d, x)` で `cdf(d, x)` 分布 `d` の累積分布函数.

`cdf(Binomial(n, p), k) == ccdf(Beta(k+1, n-k), p)` の確認:

```
In [27]: 1 n, p = 8, 0.3
          2 [
          3   [cdf(Binomial(n, p), k) for k in 0:n-1]'
          4   [ccdf(Beta(k+1, n-k), p) for k in 0:n-1]'
          5 ]
```

```
Out[27]: 2x8 Matrix{Float64}:
 0.057648  0.255298  0.551774  0.805896  0.942032  0.988708  0.99871  0.999934
 0.057648  0.255298  0.551774  0.805896  0.942032  0.988708  0.99871  0.999934
```

`ccdf(Binomial(n, p), k-1) == cdf(Beta(k, n-k+1), p)` の確認:

```
In [28]: 1 n, p = 8, 0.3
          2 [
          3   [ccdf(Binomial(n, p), k-1) for k in 1:n]'
          4   [cdf(Beta(k, n-k+1), p) for k in 1:n]'
          5 ]
```

```
Out[28]: 2x8 Matrix{Float64}:
 0.942352  0.744702  0.448226  0.194104  0.0579676  0.0112922  0.00129033  6.561e-5
 0.942352  0.744702  0.448226  0.194104  0.0579676  0.0112922  0.00129033  6.561e-5
```

```
In [29]: 1 n, k = 100, 40
          2 α = 0.05
```

```
Out[29]: 0.05
```

```
In [30]: 1 # (1)
          2 p_U = quantile(Beta(k+1, n-k), 1 - α/2)
```

```
Out[30]: 0.5027908495776652
```

```
In [31]: 1 # (1) 検算
          2 cdf(Binomial(n, p_U), k)
```

```
Out[31]: 0.02499999999999988
```

```
In [32]: 1 # (2)
          2 p_L = quantile(Beta(k, n-k+1), α/2)
```

```
Out[32]: 0.3032947687028774
```

```
In [33]: 1 # (2)検算  
2 ccdf(Binomial(n, p_L), k-1)
```

Out[33]: 0.02499999999999998

```
In [34]: 1  $\hat{p} = k/n$ 
```

Out[34]: 0.4

```
In [35]: 1  $\hat{\sigma}^2 = n * \hat{p} * (1 - \hat{p})$ 
```

Out[35]: 24.0

```
In [36]: 1  $\hat{\sigma} = \sqrt{\hat{\sigma}^2}$ 
```

Out[36]: 4.898979485566356

```
In [37]: 1 # z_{\alpha/2}  
2 z = quantile(Normal(0, 1), 1 - alpha/2)
```

Out[37]: 1.9599639845400576

```
In [38]: 1 # (3)  
2 q_U = (k + z * hat{sigma}) / n
```

Out[38]: 0.4960182335271064

```
In [39]: 1 # (3)検算  
2 cdf(Normal(n*q_U, hat{sigma}), k)
```

Out[39]: 0.02499999999999776

```
In [40]: 1 # (4)  
2 q_L = (k - z * hat{sigma}) / n
```

Out[40]: 0.3039817664728936

```
In [41]: 1 # (4)検算  
2 1 - cdf(Normal(n*q_L, hat{sigma}), k)
```

Out[41]: 0.02499999999999998

2.4.3 Julia言語による視覚化2

In [42]:

```

1 # (1)
2 plot(x → mypdf(Binomial(n, p_U), x), 10, 70; label="", c=1)
3 vline!([n*p_U]; label="n p_U", c=1)
4 plot!(x → mypdf(Binomial(n, p_U), x), 10, 40.5;
5     label="", fillrange=0, c=1, alpha=0.5)
6 annotate!(35, 0.01, text("2.5%", 10, :blue))
7 vline!([40]; label="", c=:gray)
8 title!("Binomial(n = $n, p_U = $($@sprintf "%.4f" p_U))")
9 plot!(; size=(600, 250))

```

UndefVarError: mypdf not defined

Stacktrace:

```

[1] (::var"#24#25")(x::Float64)
  @ Main .\In[42]:2
[2] (::ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#24#25"})(x::Float64; kw::Base.
Pairs{Symbol, Union{}, Tuple{}, NamedTuple{(), Tuple{}}})
  @ Base .\operators.jl:1085
[3] (::ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#24#25"})(x::Float64)
  @ Base .\operators.jl:1085
[4] (::ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#24#25"}, Reci
pesPipeline.var"#9#10"\{Symbol}})(x::Float64; kw::Base.Pairs{Symbol, Union{}, Tuple{}, NamedTuple
{(), Tuple{}}})
  @ Base .\operators.jl:1085
[5] ComposedFunction
  @ .\operators.jl:1085 [inlined]
[6] (::PlotUtils.var"#27#29"\{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"\{Symbo
l}, var"#24#25"}, RecipesPipeline.var"#9#10"\{Symbol}}})(x::Float64)
  @ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:49
[7] _broadcast_getindex_evalf
  @ .\broadcast.jl:670 [inlined]
[8] _broadcast_getindex
  @ .\broadcast.jl:643 [inlined]
[9] getindex
  @ .\broadcast.jl:597 [inlined]
[10] copy
  @ .\broadcast.jl:899 [inlined]
[11] materialize(bc::Base.Broadcast.Broadcasted{Base.Broadcast.DefaultArrayStyle{1}, Nothing, P
lotUtils.var"#27#29"\{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#2
4#25"}, RecipesPipeline.var"#9#10"\{Symbol}}}, Tuple{Vector{Float64}}})
  @ Base.Broadcast .\broadcast.jl:860
[12] adapted_grid(f::Any, minmax::Tuple{Float64, Float64}; max_recursions::Int64, max_curvatur
e::Float64, n_points::Int64)
  @ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:60
[13] adapted_grid(f::Any, minmax::Tuple{Float64, Float64})
  @ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:21
[14] _scaled_adapted_grid(f::Function, xscale::Symbol, yscale::Symbol, xmin::Int64, xmax::Int6
4)
  @ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:353
[15] macro expansion
  @ D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:293 [inlined]
[16] apply_recipe(plotattributes::AbstractDict{Symbol, Any}, f::Function, xmin::Number, xmax::N
umber)
  @ RecipesPipeline D:\.julia\packages\RecipesBase\qpxEX\src\RecipesBase.jl:289
[17] _process_userrecipes!(plt::Any, plotattributes::Any, args::Any)
  @ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:36
[18] recipe_pipeline!(plt::Any, plotattributes::Any, args::Any)
  @ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\RecipesPipeline.jl:70
[19] _plot!(plt::Plots.Plot, plotattributes::Any, args::Any)
  @ Plots D:\.julia\packages\Plots\nzdhU\src\plot.jl:208
[20] #plot#139
  @ D:\.julia\packages\Plots\nzdhU\src\plot.jl:91 [inlined]
[21] top-level scope
  @ In[42]:2
[22] eval
  @ .\boot.jl:373 [inlined]
[23] include_string(mapexpr::typeof(REPL.softscope), mod::Module, code::String, filename::Strin
g)
  @ Base .\loading.jl:1196

```

In [43]:

```
1 # (2)
2 plot(x → mypdf(Binomial(n, p_L), x), 10, 70; label="", c=2)
3 vline!([n*p_L]; label="n p_L", c=2)
4 plot!(x → mypdf(Binomial(n, p_L), x), 39.5, 70;
5     label="", fillrange=0, c=2, alpha=0.5)
6 annotate!(45, 0.01, text("2.5%", 10, :red))
7 vline!([40]; label="", c=:gray)
8 title!("Binomial(n = $n, p_L = $(@sprintf "%.4f" p_L))")
9 plot!(; size=(600, 250))
```

UndefVarError: mypdf not defined

Stacktrace:

```
[1] (::var"#26#27")(x::Float64)
@ Main .\In[43]:2
[2] (::ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#26#27"})(x::Float64; kw::Base.Pairs{Symbol, Union{}, Tuple{}, NamedTuple{(), Tuple{}}})
@ Base .\operators.jl:1085
[3] (::ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#26#27"})(x::Float64)
@ Base .\operators.jl:1085
[4] (::ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#26#27"}, RecipesPipeline.var"#9#10"\{Symbol}})(x::Float64; kw::Base.Pairs{Symbol, Union{}, Tuple{}, NamedTuple{(), Tuple{}}})
@ Base .\operators.jl:1085
[5] ComposedFunction
@ .\operators.jl:1085 [inlined]
[6] (::PlotUtils.var"#27#29"\{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#26#27"}, RecipesPipeline.var"#9#10"\{Symbol}}})(x::Float64)
@ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:49
[7] _broadcast_getindex_evalf
@ .\broadcast.jl:670 [inlined]
[8] _broadcast_getindex
@ .\broadcast.jl:643 [inlined]
[9] getindex
@ .\broadcast.jl:597 [inlined]
[10] copy
@ .\broadcast.jl:899 [inlined]
[11] materialize(bc::Base.Broadcast.Broadcasted{Base.Broadcast.DefaultArrayStyle{1}, Nothing, PlotUtils.var"#27#29"\{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#26#27"}, RecipesPipeline.var"#9#10"\{Symbol}}}, Tuple{Vector{Float64}}})
@ Base.Broadcast .\broadcast.jl:860
[12] adapted_grid(f::Any, minmax::Tuple{Float64, Float64}; max_recursions::Int64, max_curvature::Float64, n_points::Int64)
@ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:60
[13] adapted_grid(f::Any, minmax::Tuple{Float64, Float64})
@ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:21
[14] _scaled_adapted_grid(f::Function, xscale::Symbol,yscale::Symbol, xmin::Int64, xmax::Int64)
@ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:353
[15] macro expansion
@ D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:293 [inlined]
[16] apply_recipe(plotattributes::AbstractDict{Symbol, Any}, f::Function, xmin::Number, xmax::Number)
@ RecipesPipeline D:\.julia\packages\RecipesBase\qpxEX\src\RecipesBase.jl:289
[17] _process_userrecipes!(plt::Any, plotattributes::Any, args::Any)
@ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:36
[18] recipe_pipeline!(plt::Any, plotattributes::Any, args::Any)
@ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\RecipesPipeline.jl:70
[19] _plot!(plt::Plots.Plot, plotattributes::Any, args::Any)
@ Plots D:\.julia\packages\Plots\nzdhU\src\plot.jl:208
[20] #plot#139
@ D:\.julia\packages\Plots\nzdhU\src\plot.jl:91 [inlined]
[21] top-level scope
@ In[43]:2
[22] eval
@ .\boot.jl:373 [inlined]
[23] include_string(mapexpr::typeof(REPL.softscope), mod::Module, code::String, filename::String)
@ Base .\loading.jl:1196
```

In [44]:

```

1 # (1) and (2)
2 plot(x → mypdf(Binomial(n, p_U), x), 10, 70; label="", c=1)
3 vline!([n*p_U]; label="n p_U", c=1)
4 plot!(x → mypdf(Binomial(n, p_U), x), 10, 40.5;
5      label="", fillrange=0, c=1, alpha=0.5)
6 annotate!(35, 0.01, text("2.5%", 10, :blue))
7
8 plot!(x → mypdf(Binomial(n, p_L), x), 10, 70; label="", c=2)
9 vline!([n*p_L]; label="n p_L", c=2)
10 plot!(x → mypdf(Binomial(n, p_L), x), 39.5, 70;
11      label="", fillrange=0, c=2, alpha=0.5)
12 annotate!(45, 0.01, text("2.5%", 10, :red))
13
14 vline!([40]; label="", c=:gray)
15 title!("[p_L, p_U] = [$(@sprintf "%.4f" p_L), $(@sprintf "%.4f" p_U)]")
16 plot!(; size=(600, 250))

```

UndefVarError: mypdf not defined

Stacktrace:

```

[1] (::var"#28#29")(x::Float64)
  @ Main .\In[44]:2
[2] (::ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#28#29"})(x::Float64; kw::Base.Pairs{Symbol, Union{}, Tuple{}, NamedTuple{(), Tuple{}}})
  @ Base .\operators.jl:1085
[3] (::ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#28#29"})(x::Float64)
  @ Base .\operators.jl:1085
[4] (::ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#28#29"}, RecipesPipeline.var"#9#10"{Symbol}})(x::Float64; kw::Base.Pairs{Symbol, Union{}, Tuple{}, NamedTuple{(), Tuple{}}})
  @ Base .\operators.jl:1085
[5] ComposedFunction
  @ .\operators.jl:1085 [inlined]
[6] (::PlotUtils.var"#27#29"{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#28#29"}, RecipesPipeline.var"#9#10"{Symbol}}})(x::Float64)
  @ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:49
[7] _broadcast_getindex_evalf
  @ .\broadcast.jl:670 [inlined]
[8] _broadcast_getindex
  @ .\broadcast.jl:643 [inlined]
[9] getindex
  @ .\broadcast.jl:597 [inlined]
[10] copy
  @ .\broadcast.jl:899 [inlined]
[11] materialize(bc::Base.Broadcast.Broadcasted{Base.Broadcast.DefaultArrayStyle{1}, Nothing, PlotUtils.var"#27#29"{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#28#29"}, RecipesPipeline.var"#9#10"{Symbol}}}}, Tuple{Vector{Float64}})
  @ Base.Broadcast .\broadcast.jl:860
[12] adapted_grid(f::Any, minmax::Tuple{Float64, Float64}; max_recursions::Int64, max_curvature::Float64, n_points::Int64)
  @ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:60
[13] adapted_grid(f::Any, minmax::Tuple{Float64, Float64})
  @ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:21
[14] _scaled_adapted_grid(f::Function, xscale::Symbol, yscale::Symbol, xmin::Int64, xmax::Int64)
  @ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:353
[15] macro expansion
  @ D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:293 [inlined]
[16] apply_recipe(plotattributes::AbstractDict{Symbol, Any}, f::Function, xmin::Number, xmax::Number)
  @ RecipesPipeline D:\.julia\packages\RecipesBase\qpxEX\src\RecipesBase.jl:289
[17] _process_userrecipes!(plt::Any, plotattributes::Any, args::Any)
  @ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:36
[18] recipe_pipeline!(plt::Any, plotattributes::Any, args::Any)
  @ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\RecipesPipeline.jl:70
[19] _plot!(plt::Plots.Plot, plotattributes::Any, args::Any)
  @ Plots D:\.julia\packages\Plots\nzdhU\src\plot.jl:208
[20] #plot#139
  @ D:\.julia\packages\Plots\nzdhU\src\plot.jl:91 [inlined]
[21] top-level scope
  @ In[44]:2
[22] eval
  @ .\boot.jl:373 [inlined]
[23] include_string(mapexpr::typeof(REPL.softscope), mod::Module, code::String, filename::String)

```

@ Base .\loading.jl:1196

In [45]:

```
1 # (3)
2 plot(x → mypdf(Normal(n*q_U, ̂), x), 10, 70; label="", c=:blue, ls=:dash)
3 vline!([n*q_U]; label="n q_U", ls=:dashdot, c=:blue)
4 plot!(x → mypdf(Normal(n*q_U, ̂), x), 10, 40;
5     label="", fillrange=0, c=:blue, alpha=0.5)
6 annotate!(35, 0.01, text("2.5%", 10, :blue))
7 vline!([40]; label="", c=:gray)
8 title!("Normal(μ = 100×$(@sprintf "%4f" q_U)), σ = √($̂²))")
9 plot!(; size=(600, 250))
```

UndefVarError: mypdf not defined

Stacktrace:

```
[1] (::var"#30#31")(x::Float64)
    @ Main .\In[45]:2
[2] (::ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#30#31"})(x::Float64; kw::Base.Pairs{Symbol, Union{}, Tuple{}, NamedTuple{(), Tuple{}}})
    @ Base .\operators.jl:1085
[3] (::ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#30#31"})(x::Float64)
    @ Base .\operators.jl:1085
[4] (::ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#30#31"}, RecipesPipeline.var"#9#10"\{Symbol}})(x::Float64; kw::Base.Pairs{Symbol, Union{}, Tuple{}, NamedTuple{(), Tuple{}}})
    @ Base .\operators.jl:1085
[5] ComposedFunction
    @ .\operators.jl:1085 [inlined]
[6] (::PlotUtils.var"#27#29"\{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#30#31"}, RecipesPipeline.var"#9#10"\{Symbol}}})(x::Float64)
    @ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:49
[7] _broadcast_getindex_evaluate
    @ .\broadcast.jl:670 [inlined]
[8] _broadcast_getindex
    @ .\broadcast.jl:643 [inlined]
[9] getindex
    @ .\broadcast.jl:597 [inlined]
[10] copy
    @ .\broadcast.jl:899 [inlined]
[11] materialize(bc::Base.Broadcast.Broadcasted{Base.Broadcast.DefaultArrayStyle{1}, Nothing, PlotUtils.var"#27#29"\{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#30#31"}, RecipesPipeline.var"#9#10"\{Symbol}}}, Tuple{Vector{Float64}}})
    @ Base.Broadcast .\broadcast.jl:860
[12] adapted_grid(f::Any, minmax::Tuple{Float64, Float64}; max_recursions::Int64, max_curvature::Float64, n_points::Int64)
    @ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:60
[13] adapted_grid(f::Any, minmax::Tuple{Float64, Float64})
    @ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:21
[14] _scaled_adapted_grid(f::Function, xscale::Symbol,yscale::Symbol, xmin::Int64, xmax::Int64)
    @ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:353
[15] macro expansion
    @ D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:293 [inlined]
[16] apply_recipe(plotattributes::AbstractDict{Symbol, Any}, f::Function, xmin::Number, xmax::Number)
    @ RecipesPipeline D:\.julia\packages\RecipesBase\qpxEX\src\RecipesBase.jl:289
[17] _process_userrecipes!(plt::Any, plotattributes::Any, args::Any)
    @ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:36
[18] recipe_pipeline!(plt::Any, plotattributes::Any, args::Any)
    @ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\RecipesPipeline.jl:70
[19] _plot!(plt::Plots.Plot, plotattributes::Any, args::Any)
    @ Plots D:\.julia\packages\Plots\nzdhU\src\plot.jl:208
[20] #plot#139
    @ D:\.julia\packages\Plots\nzdhU\src\plot.jl:91 [inlined]
[21] top-level scope
    @ In[45]:2
[22] eval
    @ .\boot.jl:373 [inlined]
[23] include_string(mapexpr::typeof(REPL.softscope), mod::Module, code::String, filename::String)
    @ Base .\loading.jl:1196
```

In [46]:

```

1 # (4)
2 plot(x → mypdf(Normal(n*q_L, ̂), x), 10, 70; label="", c=:red, ls=:dash)
3 vline!([n*q_L]; label="n q_L", ls=:dashdot, c=:red)
4 plot!(x → mypdf(Normal(n*q_L, ̂), x), 40, 70;
5         label="", fillrange=0, c=:red, alpha=0.5)
6 annotate!(45, 0.01, text("2.5%", 10, :red))
7 vline!([40]; label="", c=:gray)
8 title!("Normal(μ = 100×$(@sprintf "%.4f" q_L)), σ = √($̂²))")
9 plot!(; size=(600, 250))

```

UndefVarError: mypdf not defined

Stacktrace:

```

[1] (::var"#32#33")(x::Float64)
@ Main .\In[46]:2
[2] (::ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#32#33"})(x::Float64; kw::Base.Pairs{Symbol, Union{}, Tuple{}, NamedTuple{(), Tuple{}}})
@ Base .\operators.jl:1085
[3] (::ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#32#33"})(x::Float64)
@ Base .\operators.jl:1085
[4] (::ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#32#33"}, RecipesPipeline.var"#9#10"{Symbol}})(x::Float64; kw::Base.Pairs{Symbol, Union{}, Tuple{}, NamedTuple{(), Tuple{}}})
@ Base .\operators.jl:1085
[5] ComposedFunction
@ .\operators.jl:1085 [inlined]
[6] (::PlotUtils.var"#27#29"{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#32#33"}, RecipesPipeline.var"#9#10"{Symbol}}})(x::Float64)
@ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:49
[7] _broadcast_getindex_evalf
@ .\broadcast.jl:670 [inlined]
[8] _broadcast_getindex
@ .\broadcast.jl:643 [inlined]
[9] getindex
@ .\broadcast.jl:597 [inlined]
[10] copy
@ .\broadcast.jl:899 [inlined]
[11] materialize(bc::Base.Broadcast.Broadcasted{Base.Broadcast.DefaultArrayStyle{1}, Nothing, PlotUtils.var"#27#29"{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#32#33"}, RecipesPipeline.var"#9#10"{Symbol}}}, Tuple{Vector{Float64}}})
@ Base.Broadcast .\broadcast.jl:860
[12] adapted_grid(f::Any, minmax::Tuple{Float64, Float64}; max_recursions::Int64, max_curvature::Float64, n_points::Int64)
@ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:60
[13] adapted_grid(f::Any, minmax::Tuple{Float64, Float64})
@ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:21
[14] _scaled_adapted_grid(f::Function, xscale::Symbol,yscale::Symbol, xmin::Int64, xmax::Int64)
@ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:353
[15] macro expansion
@ D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:293 [inlined]
[16] apply_recipe(plotattributes::AbstractDict{Symbol, Any}, f::Function, xmin::Number, xmax::Number)
@ RecipesPipeline D:\.julia\packages\RecipesBase\qpxEX\src\RecipesBase.jl:289
[17] _process_userrecipes!(plt::Any, plotattributes::Any, args::Any)
@ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:36
[18] recipe_pipeline!(plt::Any, plotattributes::Any, args::Any)
@ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\RecipesPipeline.jl:70
[19] _plot!(plt::Plots.Plot, plotattributes::Any, args::Any)
@ Plots D:\.julia\packages\Plots\nzdhU\src\plot.jl:208
[20] #plot#139
@ D:\.julia\packages\Plots\nzdhU\src\plot.jl:91 [inlined]
[21] top-level scope
@ In[46]:2
[22] eval
@ .\boot.jl:373 [inlined]
[23] include_string(mapexpr::typeof(REPL.softscope), mod::Module, code::String, filename::String)
@ Base .\loading.jl:1196

```

In [47]:

```
1 # (3) and (4)
2 plot(x → mypdf(Normal(n*q_U, ̂), x), 10, 70; label="", c=:blue, ls=:dash)
3 vline!([n*q_U]; label="n q_U", ls=:dashdot, c=:blue)
4 plot!(x → mypdf(Normal(n*q_U, ̂), x), 10, 40;
5         label="", fillrange=0, c=:blue, alpha=0.5)
6 annotate!(35, 0.01, text("2.5%", 10, :blue))
7
8 plot!(x → mypdf(Normal(n*q_L, ̂), x), 10, 70; label="", c=:red, ls=:dash)
9 vline!([n*q_L]; label="n q_L", ls=:dashdot, c=:red)
10 plot!(x → mypdf(Normal(n*q_L, ̂), x), 40, 70;
11         label="", fillrange=0, c=:red, alpha=0.5)
12 annotate!(45, 0.01, text("2.5%", 10, :red))
13
14 vline!([40]; label="")
15 title!("[q_L, q_U] = [$(@sprintf "%.4f" q_L), $(@sprintf "%.4f" q_U)]")
16 plot!(; size=(600, 250))
```

UndefVarError: mypdf not defined

Stacktrace:

```
[1] (::var"#34#35")(x::Float64)
    @ Main .\In[47]:2
[2] (::ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#34#35"})(x::Float64; kw::Base.Pairs{Symbol, Union{}, Tuple{}, NamedTuple{(), Tuple{}}})
    @ Base .\operators.jl:1085
[3] (::ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#34#35"})(x::Float64)
    @ Base .\operators.jl:1085
[4] (::ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#34#35"}, RecipesPipeline.var"#9#10"\{Symbol}})(x::Float64; kw::Base.Pairs{Symbol, Union{}, Tuple{}, NamedTuple{(), Tuple{}}})
    @ Base .\operators.jl:1085
[5] ComposedFunction
    @ .\operators.jl:1085 [inlined]
[6] (::PlotUtils.var"#27#29"\{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#34#35"}, RecipesPipeline.var"#9#10"\{Symbol}}})(x::Float64)
    @ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:49
[7] _broadcast_getindex_evalf
    @ .\broadcast.jl:670 [inlined]
[8] _broadcast_getindex
    @ .\broadcast.jl:643 [inlined]
[9] getindex
    @ .\broadcast.jl:597 [inlined]
[10] copy
    @ .\broadcast.jl:899 [inlined]
[11] materialize(bc::Base.Broadcast.Broadcasted{Base.Broadcast.DefaultDisplayStyle{1}, Nothing, PlotUtils.var"#27#29"\{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"\{Symbol}, var"#34#35"}, RecipesPipeline.var"#9#10"\{Symbol}}}}, Tuple{Vector{Float64}})
    @ Base.Broadcast .\broadcast.jl:860
[12] adapted_grid(f::Any, minmax::Tuple{Float64, Float64}; max_recursions::Int64, max_curvature::Float64, n_points::Int64)
    @ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:60
[13] adapted_grid(f::Any, minmax::Tuple{Float64, Float64})
    @ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:21
[14] _scaled_adapted_grid(f::Function, xscale::Symbol,yscale::Symbol, xmin::Int64, xmax::Int64)
    @ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:353
[15] macro expansion
    @ D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:293 [inlined]
[16] apply_recipe(plotattributes::AbstractDict{Symbol, Any}, f::Function, xmin::Number, xmax::Number)
    @ RecipesPipeline D:\.julia\packages\RecipesBase\qpxEX\src\RecipesBase.jl:289
[17] _process_userrecipes!(plt::Any, plotattributes::Any, args::Any)
    @ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:36
[18] recipe_pipeline!(plt::Any, plotattributes::Any, args::Any)
    @ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\RecipesPipeline.jl:70
[19] _plot!(plt::Plots.Plot, plotattributes::Any, args::Any)
    @ Plots D:\.julia\packages\Plots\nzdhU\src\plot.jl:208
[20] #plot#139
    @ D:\.julia\packages\Plots\nzdhU\src\plot.jl:91 [inlined]
[21] top-level scope
    @ In[47]:2
[22] eval
    @ .\boot.jl:373 [inlined]
[23] include_string(mapexpr::typeof(REPL.softscope), mod::Module, code::String, filename::String)
    @ Base .\loading.jl:1196
```


In [48]:

```

1 # (1)(2)(3)(4)
2 plot(x → mypdf(Binomial(n, p_U), x), 10, 70; label="", c=1)
3 vline!([n*p_U]; label="n p_U", c=1)
4
5 plot!(x → mypdf(Binomial(n, p_L), x), 10, 70; label="", c=2)
6 vline!([n*p_L]; label="n p_L", c=2)
7
8 plot!(x → mypdf(Normal(n*q_U, ̂), x), 10, 70; label="", c=:blue, ls=:dash)
9 vline!([n*q_U]; label="n q_U", ls=:dashdot, c=:blue)
10
11 plot!(x → mypdf(Normal(n*q_L, ̂), x), 10, 70; label="", c=:red, ls=:dash)
12 vline!([n*q_L]; label="n q_L", ls=:dashdot, c=:red)
13
14 vline!([40]; label="", c=:gray)
15 title!"[p_L, p_U] = [$(@sprintf "%.4f" p_L), $(@sprintf "%.4f" p_U)]*"
16     ", [q_L, q_U] = [$(@sprintf "%.4f" q_L), $(@sprintf "%.4f" q_U)]")
17 plot!(; size=(600, 250))

```

UndefVarError: mypdf not defined

Stacktrace:

```

[1] (::var"#36#37")(x::Float64)
    @ Main .\In[48]:2
[2] (::ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#36#37"})(x::Float64; kw::Base.Pairs{Symbol, Union{}, Tuple{}, NamedTuple{(), Tuple{}}})
    @ Base .\operators.jl:1085
[3] (::ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#36#37"})(x::Float64)
    @ Base .\operators.jl:1085
[4] (::ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#36#37"}, RecipesPipeline.var"#9#10"{Symbol}})(x::Float64; kw::Base.Pairs{Symbol, Union{}, Tuple{}, NamedTuple{(), Tuple{}}})
    @ Base .\operators.jl:1085
[5] ComposedFunction
    @ .\operators.jl:1085 [inlined]
[6] (::PlotUtils.var"#27#29"{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#36#37"}, RecipesPipeline.var"#9#10"{Symbol}}})(x::Float64)
    @ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:49
[7] _broadcast_getindex_evalf
    @ .\broadcast.jl:670 [inlined]
[8] _broadcast_getindex
    @ .\broadcast.jl:643 [inlined]
[9] getindex
    @ .\broadcast.jl:597 [inlined]
[10] copy
    @ .\broadcast.jl:899 [inlined]
[11] materialize(bc::Base.Broadcast.Broadcasted{Base.Broadcast.DefaultArrayStyle{1}, Nothing, PlotUtils.var"#27#29"{ComposedFunction{ComposedFunction{RecipesPipeline.var"#7#8"{Symbol}, var"#36#37"}, RecipesPipeline.var"#9#10"{Symbol}}}}, Tuple{Vector{Float64}})
    @ Base.Broadcast .\broadcast.jl:860
[12] adapted_grid(f::Any, minmax::Tuple{Float64, Float64}; max_recursions::Int64, max_curvature::Float64, n_points::Int64)
    @ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:60
[13] adapted_grid(f::Any, minmax::Tuple{Float64, Float64})
    @ PlotUtils D:\.julia\packages\PlotUtils\jNTYY\src\adapted_grid.jl:21
[14] _scaled_adapted_grid(f::Function, xscale::Symbol,yscale::Symbol, xmin::Int64, xmax::Int64)
    @ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:353
[15] macro expansion
    @ D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:293 [inlined]
[16] apply_recipe(plotattributes::AbstractDict{Symbol, Any}, f::Function, xmin::Number, xmax::Number)
    @ RecipesPipeline D:\.julia\packages\RecipesBase\qpxEX\src\RecipesBase.jl:289
[17] _process_userrecipes!(plt::Any, plotattributes::Any, args::Any)
    @ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\user_recipe.jl:36
[18] recipe_pipeline!(plt::Any, plotattributes::Any, args::Any)
    @ RecipesPipeline D:\.julia\packages\RecipesPipeline\F2mWY\src\RecipesPipeline.jl:70
[19] _plot!(plt::Plots.Plot, plotattributes::Any, args::Any)
    @ Plots D:\.julia\packages\Plots\nzdhU\src\plot.jl:208
[20] #plot#139
    @ D:\.julia\packages\Plots\nzdhU\src\plot.jl:91 [inlined]
[21] top-level scope
    @ In[48]:2
[22] eval
    @ .\boot.jl:373 [inlined]
[23] include_string(mapexpr::typeof(REPL.softscope), mod::Module, code::String, filename::String)

```

2.5 問題: 二項分布の中心極限定理を使った極限の計算問題

$0 < p < 1$ と $m = 0, 1, 2, \dots$ について以下の極限を求めよ:

$$\lim_{n \rightarrow \infty} \sum_{k=0}^n \left(\frac{k - np}{\sqrt{np(1-p)}} \right)^{2m} \binom{n}{k} p^k (1-p)^{n-k}.$$

解答例: 二項分布の中心極限定理より,

$$\begin{aligned} & \lim_{n \rightarrow \infty} \sum_{k=0}^n \left(\frac{k - np}{\sqrt{np(1-p)}} \right)^{2m} \binom{n}{k} p^k (1-p)^{n-k} \\ &= \int_{-\infty}^{\infty} \left(\frac{x}{\sqrt{p(1-p)}} \right)^{2m} \frac{1}{\sqrt{2\pi p(1-p)}} \exp\left(-\frac{x^2}{2p(1-p)}\right) dx \\ &= \int_{-\infty}^{\infty} z^{2m} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz = \frac{2}{\sqrt{2\pi}} \int_0^{\infty} e^{-z^2/2} z^{2m} dz \\ &= \frac{1}{\sqrt{2\pi}} \int_0^{\infty} e^{-y/2} y^{m+1/2-1} dy = \frac{1}{\sqrt{2\pi}} 2^{m+1/2} \Gamma(m + 1/2) \\ &= \frac{1}{\sqrt{2\pi}} 2^{m+1/2} \frac{2m-1}{2} \cdots \frac{3}{2} \frac{1}{2} \Gamma(1/2) = 1 \cdot 3 \cdots (2m-1). \end{aligned}$$

最初の等号で二項分布の中心極限定理を使い、2番目の等号で $x = \sqrt{p(1-p)} z$ とおき、4番目の等号で $z = \sqrt{y}$ とおいた。5番目の等号では $\int_0^{\infty} e^{-y/\theta} y^{\alpha-1} dy = \theta^{\alpha} \Gamma(\alpha)$ を使った。

解答終

上の問題の結果を数値的に確認してみよう。

```
In [49]: 1 g(n, p, m, k) = ((k - n*p)/\sqrt(n*p*(1-p)))^(2m)
2 p = 0.4
3 @show p
4 for m in 0:5
5     println()
6     @show m
7     @show n = 10^(m+1)
8     @show prod(1:2:2m-1)
9     @show sum(g(n, p, m, k)*pdf(Binomial(n, p), k) for k in 0:n)
10 end;
p = 0.4

m = 0
n = 10^(m + 1) = 10
prod(1:2:2m - 1) = 1
sum((g(n, p, m, k) * pdf(Binomial(n, p), k) for k = 0:n)) = 1.0000000000000001

m = 1
n = 10^(m + 1) = 100
prod(1:2:2m - 1) = 1
sum((g(n, p, m, k) * pdf(Binomial(n, p), k) for k = 0:n)) = 1.0000000000000095

m = 2
n = 10^(m + 1) = 1000
prod(1:2:2m - 1) = 3
sum((g(n, p, m, k) * pdf(Binomial(n, p), k) for k = 0:n)) = 2.99816666666703

m = 3
n = 10^(m + 1) = 10000
prod(1:2:2m - 1) = 15
sum((g(n, p, m, k) * pdf(Binomial(n, p), k) for k = 0:n)) = 14.997416790272869

m = 4
n = 10^(m + 1) = 100000
prod(1:2:2m - 1) = 105
sum((g(n, p, m, k) * pdf(Binomial(n, p), k) for k = 0:n)) = 104.99661670588064

m = 5
n = 10^(m + 1) = 1000000
prod(1:2:2m - 1) = 945
sum((g(n, p, m, k) * pdf(Binomial(n, p), k) for k = 0:n)) = 944.9952750033706
```

2.6 二項分布の中心極限定理の証明の方針

二項分布の中心極限定理を以下の2段階で示す。

記号法に関する準備: $0 < p < 1$ は固定するが, n, k は動かす. k は n ごとに決まっていると仮定する. 混乱を完全に防ぐには k を k_n のように書いた方がよいが, 記号が煩雑になって書くのが面倒になるので, 単に k と書くことにする. $n \rightarrow \infty$ で 0 に収束する量を $o(1)$ と書く. より一般に a_n で割った結果が $n \rightarrow \infty$ で 0 に収束する量を $o(a_n)$ と書く. 例えば $o(1/\sqrt{n})$ は \sqrt{n} をかけた後であっても $n \rightarrow \infty$ で 0 に収束する量になる. この記号法のもとで, Stirlingの公式は次のように書ける:

$$n! = n^n e^{-n} \sqrt{2\pi n} (1 + o(1)).$$

(1) n ごとに決まっている k は $n \rightarrow \infty$ で

$$\frac{k}{n} = p + o(1)$$

という条件を満たしていると仮定する. このとき, 二項分布の確率質量函数

$$P(k|n, p) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k} \quad (k = 0, 1, \dots, n)$$

における二項係数の中の $n!, k!, (n-k)!$ に階乗に関するStirlingの公式による近似を適用すると次の近似式が得られる:

$$P(k|n, p) = \frac{1}{\sqrt{2\pi np(1-p)}} \left(\left(\frac{k/n}{p} \right) \left(\frac{1-k/n}{1-p} \right) \right)^{-n} (1 + o(1)).$$

途中の段階としてこの形に整理しておくと計算の見通しが良くなる. (しかもこの形には重要な意味がある. その点については後述する.)

(2) 固定された実数 x について, 上の(1)での近似式を示すために使った仮定 $k/n = p + o(1)$ よりも強い条件

$$\frac{k}{n} = p + \frac{x}{\sqrt{n}} + o(n^{-1/2})$$

を仮定すると、次が得られる：

$$P(k|n, p) = \underbrace{\frac{1}{\sqrt{2\pi p(1-p)}} \exp\left(-\frac{x^2}{2p(1-p)}\right)}_{\text{pdf of normal distribution}} \underbrace{\frac{1}{\sqrt{n}}}_{dx/dk} (1 + o(1)).$$

これより、

$$K_n \sim \text{Binomial}(n, p), \quad X_n = \frac{K_n - np}{\sqrt{n}}, \quad X_\infty \sim \text{Normal}(0, p(1-p))$$

ならば

$$\lim_{n \rightarrow \infty} E[f(X_n)] \rightarrow E[f(X_\infty)]$$

となることがわかる。

この結果は、 n が大きなとき、 $K_n \sim \text{Binomial}(n, p)$ に対する $X_n = (K_n - np)/\sqrt{n}$ が従う分布が $\text{Normal}(0, p(1-p))$ で近似されることを意味している($p(1-p)$ は分布 $\text{Bernoulli}(p)$ の分散に等しい)。この結果を二項分布の中心極限定理と呼ぶ。

2.7 (1) Stirlingの公式を使った二項分布の確率密度函数の近似

n ごとに決まっている k は $n \rightarrow \infty$ で

$$\frac{k}{n} = p + o(1) \quad \left(\Leftrightarrow 1 - \frac{k}{n} = 1 - p + o(1) \right)$$

という条件を満たしていると仮定する。 $0 < p < 1$ と仮定していたので、 $n \rightarrow \infty$ のとき $k \rightarrow \infty$ だけではなく、 $n - k \rightarrow \infty$ も成立している。ゆえに $n, k, n - k$ について次のStirlingの公式が成立している：

$$\begin{aligned} n! &= n^n e^{-n} \sqrt{2\pi n} (1 + o(1)), \\ k! &= k^k e^{-k} \sqrt{2\pi k} (1 + o(1)), \\ (n - k)! &= (n - k)^{n-k} e^{-(n-k)} \sqrt{2\pi(n - k)} (1 + o(1)). \end{aligned}$$

これを代入すると

$$\begin{aligned} \frac{n!}{k!(n - k)!} &= \frac{n^n e^{-n} \sqrt{2\pi n}}{k^k e^{-k} \sqrt{2\pi k} (n - k)^{n-k} e^{-(n-k)} \sqrt{2\pi(n - k)}} (1 + o(1)) \\ &= \frac{1}{(k/n)^k (1 - k/n)^{n-k} \sqrt{2\pi n (k/n)(1 - k/n)}} (1 + o(1)) \\ &= \frac{1}{(k/n)^k (1 - k/n)^{n-k} \sqrt{2\pi np(1-p)}} (1 + o(1)) \end{aligned}$$

2つめの等号で分子の e^{-n} と分母の $e^{-k} e^{-(n-k)}$ がキャンセルさせ、分子分母を n^k と n^{n-k} と $\sqrt{2\pi n}$ で割った。3つめの等号で $k/n = p + o(1)$, $1 - k/n = 1 - p + o(1)$ を使った。これを二項分布の確率質量函数

$$P(k|n, p) = \frac{n!}{k!(n - k)!} p^k (1 - p)^{n-k} \quad (k = 0, 1, \dots, n)$$

に適用すると、

$$\begin{aligned} P(k|n, p) &= \frac{1}{\sqrt{2\pi np(1-p)}} \left(\frac{p}{k/n} \right)^k \left(\frac{1-p}{1-k/n} \right)^{n-k} (1 + o(1)) \\ &= \frac{1}{\sqrt{2\pi np(1-p)}} \left(\left(\frac{k/n}{p} \right)^{k/n} \left(\frac{1-k/n}{1-p} \right)^{1-k/n} \right)^{-n} (1 + o(1)). \end{aligned}$$

2.8 注意: Kullback-Leibler情報量とSanovの定理との関係

前節でわざわざ最後の形に変形した理由は、対数を取ると

$$\log \left(\left(\left(\frac{k/n}{p} \right)^{k/n} \left(\frac{1-k/n}{1-p} \right)^{1-k/n} \right)^{-n} \right) = -n \left(\frac{k}{n} \log \frac{k/n}{p} + \left(1 - \frac{k}{n} \log \frac{1-k/n}{1-p} \right) \right)$$

となることを分かり易くするためである。これを $-n$ で割った結果は

$$D(Q||P) = \sum_{i=1}^r Q(i) \log \frac{Q(i)}{P(i)} \quad \left(P(i), Q(i) > 0, \sum_{i=1}^r P(i) = \sum_{i=1}^r Q(i) = 1 \right)$$

の特別な場合になっている($r = 2, P(1) = p, P(2) = 1 - p, Q(1) = k/n, Q(2) = 1 - k/n$)。この $D(Q||P)$ は **Kullback-Leibler 情報量 (KL情報量)** と呼ばれ、そして、上の計算結果は $r = 2$ の場合のKL情報量のSanovの証明にもなっている。Sanovの定理の易しい解説が以下の場所にある：

- [Kullback-Leibler情報量とSanovの定理 \(<https://genkuroki.github.io/documents/20160616KullbackLeibler.pdf>\)](https://genkuroki.github.io/documents/20160616KullbackLeibler.pdf)

Sanovの定理の内容を要約すると、Kullback-Leibler情報量 $D(Q||P)$ は確率分布 P に従う乱数の繰り返し生成において確率分布 Q がどれだけ出て来難いかを表しているという主張になる。上の二項分布の場合には、 Q に対応する k/n が生じる確率が $\exp(-nD(Q||P) + O(\log n))$ の形をしていることが示されていることになる（分母の $\sqrt{2\pi np(1-p)}$ は $\exp(O(\log n))$ になる）、KL情報量 $D(Q||P)$ が大きいほど k/n が生じる確率は急激に小さくなる。

Sanovの定理は統計学でのモデル選択における情報量規準の考え方の基礎になっている。KL情報量は確率論では **大偏差原理** に出て来る。KL情報量は物理的には統計力学的な相対エントロピーの -1 倍になる。このような分野を超えた広がりについては、**赤池情報量規準 (AIC)** で有名な赤池弘次氏による1980年の2つの論説も参照せよ：

- 赤池弘次、エントロピーとモデルの尤度(<講座>物理学周辺の確率統計), 日本物理学会誌, 1980年第35巻7号, pp. 608-614. [link](https://www.jstage.jst.go.jp/article/butsuri1946/35/7/35_7_608/_article/-char/ja/) (https://www.jstage.jst.go.jp/article/butsuri1946/35/7/35_7_608/_article/-char/ja/)
- 赤池弘次、統計的推論のパラダイムの変遷について、統計数理研究所彙報, 1980年第27巻第1号, pp. 5-12. [link](https://ismrepo.ism.ac.jp/index.php?active_action=repository_view_main_item_detail&page_id=13&block_id=21&item_id=32568&item_no=1) (https://ismrepo.ism.ac.jp/index.php?active_action=repository_view_main_item_detail&page_id=13&block_id=21&item_id=32568&item_no=1)

赤池弘次さんは前者のp.612に「筆者によって導入されたこの統計量はAIC(an information criterionの略記*)と呼ばれ」と書いているが、それに対して会誌編集委員会が「一般には Akaike's information criterionの略と解されている」と脚注を付けている。このことを確認するだけでもダウンロードする価値があるだろう。

上で紹介した計算は二項分布の確率質量函数にStirlingの公式を適用する単純な計算に過ぎないのだが、その先には分野を超えた非常に面白い世界が広がっている。

2.9 (2) 二項分布の確率質量函数から正規分布の密度函数が出て来ること

n ごとに決まっている k は $n \rightarrow \infty$ で

$$\frac{k}{n} = p + o(1)$$

という条件を満たしているならば

$$P(k|n, p) = \frac{1}{\sqrt{2\pi np(1-p)}} \left(\left(\frac{k/n}{p} \right)^{k/n} \left(\frac{1-k/n}{1-p} \right)^{1-k/n} \right)^{-n} (1 + o(1))$$

となることを上で示した。以下では $k/n = p + o(1)$ よりも強い条件

$$\frac{k - np}{\sqrt{n}} = x + \varepsilon_n, \quad \lim_{n \rightarrow \infty} \varepsilon_n = 0$$

を仮定する。このとき、

$$\frac{k}{n} = p + \frac{x + \varepsilon_n}{\sqrt{n}}, \quad 1 - \frac{k}{n} = 1 - p - \frac{x + \varepsilon_n}{\sqrt{n}}.$$

以上の仮定のもとで、 $\log(1+t) = t - t^2/2 + t^3/3 - \dots$ を使うと、

$$\begin{aligned}
\log \left(\frac{k/n}{p} \right)^{k/n} &= \frac{k}{n} \log \frac{k/n}{p} = \left(p + \frac{x + \varepsilon_n}{\sqrt{n}} \right) \log \left(1 + \frac{x + \varepsilon_n}{\sqrt{n} p} \right) \\
&= \left(p + \frac{x + \varepsilon_n}{\sqrt{n}} \right) \left(\frac{x + \varepsilon_n}{\sqrt{n} p} - \frac{x^2}{2np^2} + o(1/n) \right) \\
&= \frac{x + \varepsilon_n}{\sqrt{n}} - \frac{x^2}{2np} + \frac{x^2}{np^2} + o(1/n) \\
&= \frac{x + \varepsilon_n}{\sqrt{n}} + \frac{x^2}{2np} + o(1/n),
\end{aligned}$$

$$\begin{aligned}
\log \left(1 - \frac{k/n}{1-p} \right)^{1-k/n} &= \left(1 - \frac{k}{n} \right) \log \frac{k/n - 1}{p} = \left(1 - p - \frac{x + \varepsilon_n}{\sqrt{n}} \right) \log \left(1 - \frac{x + \varepsilon_n}{\sqrt{n}(1-p)} \right) \\
&= \left(1 - p - \frac{x + \varepsilon_n}{\sqrt{n}} \right) \left(-\frac{x + \varepsilon_n}{\sqrt{n}(1-p)} - \frac{x^2}{2n(1-p)^2} + o(1/n) \right) \\
&= -\frac{x + \varepsilon_n}{\sqrt{n}} - \frac{x^2}{2n(1-p)} + \frac{x^2}{n(1-p)^2} + o(1/n) \\
&= -\frac{x + \varepsilon_n}{\sqrt{n}} + \frac{x^2}{2n(1-p)} + o(1/n).
\end{aligned}$$

ゆえに,

$$\begin{aligned}
&\log \left(\left(\frac{k/n}{p} \right)^{k/n} \left(\frac{1-k/n}{1-p} \right)^{1-k/n} \right)^{-n} \\
&= -n \left(\log \left(\frac{k/n}{p} \right)^{k/n} + \log \left(1 - \frac{k/n}{1-p} \right)^{1-k/n} \right) \\
&= -\frac{x^2}{2p} - \frac{x^2}{2(1-p)} + o(1) = -\frac{x^2}{2p(1-p)} + o(1).
\end{aligned}$$

したがって,

$$P(k|n, p) = \frac{1}{\sqrt{2\pi p(1-p)}} \exp \left(-\frac{x^2}{2p(1-p)} \right) \frac{1}{\sqrt{n}} (1 + o(1)).$$

この公式の $1/\sqrt{n}$ より前の部分は平均 0, 分散 $p(1-p)$ の正規分布の確率密度函数であり, $1/\sqrt{n}$ の因子は $k = np + \sqrt{n}(x + \varepsilon_n)$ から形式的に得られる $dk = \sqrt{n} dx$ の \sqrt{n} の因子とキャンセルすると自然に見える.

これより, $K_n \sim \text{Binomial}(n, p)$, $X \sim \text{Normal}(0, p(1-p))$ のとき,

$$X_n = \frac{K_n - np}{\sqrt{n}}, \quad x_n(k) = \frac{k - np}{\sqrt{n}}, \quad \Delta x_n = \frac{1}{\sqrt{n}}$$

とおくと, 有界な連続函数 $f(x)$ について, $n \rightarrow \infty$ のとき

$$\begin{aligned}
E[f(X_n)] &= \sum_{k=0}^n f(x_n(k)) P(k|n, p) \\
&= \sum_{k=0}^n f(x_n(k)) \frac{1}{\sqrt{2\pi p(1-p)}} \exp \left(-\frac{x_n(k)^2}{2p(1-p)} \right) \Delta x_n (1 + o(1)) \\
&= \int_{-\infty}^{\infty} f(x) \frac{1}{\sqrt{2\pi p(1-p)}} \exp \left(-\frac{x^2}{2p(1-p)} \right) dx = E[f(X_\infty)].
\end{aligned}$$

以上の議論は厳密には少しギャップがあるのだが, 二項分布に関して中心極限定理が成立する理由の本質は十分に分かる内容になっている.

3 中心極限定理

中心極限定理 (central limit theorem)は「中心に収束する極限定理」というような意味ではなく、「確率論における中心的な極限定理」という意味である.

注意: 特性函数, モーメント母函数, キュムラント母函数, 歪度(わいど), 尖度(せんど)などについては「標本分布について」のノートに詳しい解説がある.

3.1 中心極限定理のラフな説明

X_1, \dots, X_n は各々が期待値 μ , 分散 σ^2 を持つ分布に従う n 個の独立同分布確率変数達であるとし, n は十分に大きいと仮定する.

中心極限定理は以下のように同値な言い方が色々ある:

(1) それらの和 $X_1 + \dots + X_n$ が従う分布は期待値 $n\mu$, 分散 $n\sigma^2$ の正規分布で近似される:

$$\sum_{i=1}^n X_i \sim \text{Normal}(n\mu, \sqrt{n}\sigma) \quad \text{approximately.}$$

(2) それらの加法平均 $\bar{X}_n = (X_1 + \dots + X_n)/n$ が従う分布は期待値 μ , 分散 σ^2/n の正規分布で近似される:

$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i \sim \text{Normal}(\mu, \sigma/\sqrt{n}) \quad \text{approximately.}$$

(3) $\sqrt{n}(\bar{X}_n - \mu)$ が従う分布は期待値 0, 分散 σ^2 の正規分布で近似される:

$$\sqrt{n}(\bar{X}_n - \mu) = \frac{1}{\sqrt{n}} \sum_{i=1}^n (X_i - \mu) \sim \text{Normal}(0, \sigma) \quad \text{approximately.}$$

(4) 次の Z_n が従う分布は標準正規分布で近似される:

$$Z_n = \frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma} = \frac{1}{\sqrt{n}} \sum_{i=1}^n \frac{X_i - \mu}{\sigma} \sim \text{Normal}(0, 1) \quad \text{approximately.}$$

これらはどれも便利であり, 今後, 自由に使われることになるだろう.

注意・警告: 統計学に中心極限定理を応用する場合には, $n \rightarrow \infty$ とできる数学的に理想的な状況における中心極限定理ではなく, 有限の固定された n における近似として中心極限定理は使われる. だから, 場合ごとにその近似の誤差が問題になる. この問題は目的ごとに異なる許容される誤差との兼ね合いの問題になるので, きれいな一般論で解決できる話ではなく, 中心極限定理のユーザー自身が自分自身の目的に合わせて誤差が大きくなるリスクをどこまで許容するかを決定する必要がある. そのためには, 中心極限定理による近似の精度がどのような場合に悪くなりそうかについて前もって理解しておく必要がある.

補足: 分布 D_n が分布 D_∞ で近似されるとは, 分布 D_n に従う確率変数 X_n と分布 D_∞ に従う確率変数 X_∞ と適切なクラスに含まれる任意の函数 $f(x)$ について

$$\lim_{n \rightarrow \infty} E[f(X_n)] = E[f(X_\infty)]$$

が成立することだと, 大雑把に定義しておく. (函数 $f(x)$ としてどのような函数を許すかについて詳細な説明をするためにはさらなる数学的道具立てが必要でかる, 難に扱っても害はほとんどない場合にもなっているので, このノートで説明しない. 所謂「分布収束」「法収束」の話になる.) 統計学で必要な確率論の計算は期待値を取る操作経由で可能なのでこのような定義にしておいても困ることはない.

補足の補足: 数学的詳細を理解できそうもない人であっても, 中心極限定理の良いユーザーになることは可能である. そのためには, 正規分布で近似したい分布のグラフとそれを近似すると期待される正規分布のグラフを重ねて描く作業を十分に沢山行えばよい. 数学的詳細を理解可能の人であっても, そういう視覚化による理解をサボってしまうと, 有限の n でどのように正規分布による近似の精度が悪くなるかについての感覚が身に付かなくなってしまう. 数学的詳細を理解していないなくても, 中心極限定理の利用でどのようなリスクが発生するかを具体例を沢山見ることによって把握していれば, その人は十分に良い中心極限定理ユーザーになれるだろう.

3.2 中心極限定理の特性函数を使った証明

中心極限定理: X_1, X_2, X_3, \dots が独立同分布な確率変数の列であるとき, $\mu = E[X_k]$ が定義されていて, $\sigma^2 = \text{var}(X_k) = E[(X_k - \mu)^2] < \infty$ でかつ, $E[|X_k - \mu|^3] < \infty$ となっていると仮定する. このとき,

$$\bar{X}_n = \frac{1}{n} \sum_{k=1}^n X_k, \quad Z_n = \frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma}$$

とおくと, $n \rightarrow \infty$ で Z_n の分布は標準正規分布に近付く.

証明: X_k の標準化を $Y_k = (X_k - \mu)/\sigma$ と書くことにする. Y_1, Y_2, \dots も独立同分布になり, $E[Y_k] = 0$, $E[Y_k^2] = 1$ が成立している. ゆえに Y_k の特性函数 $\varphi(t)$ は k によらず,

$$\varphi(t) = E[e^{itY_k}] = 1 + iE[Y_k]t - E[Y_k^2]\frac{t^2}{2} + O(t^3) = 1 - \frac{t^2}{2} + O(t^3)$$

という形になる。そして、

$$\frac{1}{\sqrt{n}} \sum_{k=1}^n Y_k = \frac{\sqrt{n}}{\sigma} \frac{1}{n} \sum_{k=1}^n (X_k - \mu) = \frac{\sqrt{n}}{\sigma} (\bar{X}_n - \mu) = Z_n$$

なので、 Z_n の特性函数は、 $n \rightarrow \infty$ で

$$\begin{aligned} \varphi_{Z_n}(t) &= E \left[\exp \left(it \frac{1}{\sqrt{n}} \sum_{k=1}^n Y_k \right) \right] = E \left[\prod_{k=1}^n \exp \left(i \frac{t}{\sqrt{n}} Y_k \right) \right] \\ &= \prod_{k=1}^n E \left[\exp \left(i \frac{t}{\sqrt{n}} Y_k \right) \right] = \varphi \left(\frac{t}{\sqrt{n}} \right)^n = \left(1 - \frac{t^2}{2n} + O(n^{-3/2}) \right)^n \rightarrow e^{-t^2/2}. \end{aligned}$$

と標準正規分布の特性函数 $e^{-t^2/2}$ に収束する。(3つめの等号で Y_1, \dots, Y_n の独立性を使った。)

ゆえに Z_n の分布は $n \rightarrow \infty$ で標準正規分布に近付く。

証明終

注意: $f(x)$ のFourier変換を

$$\hat{f}(t) = \int_{-\infty}^{\infty} f(x) e^{-itx} dx$$

と定義するとき、逆Fourier変換によって、

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(t) e^{itx} dt$$

が成立しているので、確率変数 X について $f(X)$ の期待値は、 X の特性函数 $\varphi_X(t) = E[e^{itX}]$ を使って、

$$E[f(X)] = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(x) \varphi_X(t) dt$$

と書ける(積分と期待値を取る操作を交換できる)。このことから、確率変数 X の特性函数と確率変数 Y の特性函数が近ければ、 $f(X)$ と $f(Y)$ の期待値も近くなる。このことから、分布の近似は特性函数の近似で扱えることがわかる。

3.3 中心極限定理の収束の速さと歪度

前節の証明より、独立同分布確率変数列 X_1, X_2, \dots に対して、

$$\mu = E[X_k], \quad \sigma = \sqrt{E[(X - \mu)^2]}, \quad Y_k = \frac{X_k - \mu}{\sigma}, \quad \varphi(t) = E[e^{itY_k}]$$

とおいたときの、 $n \rightarrow \infty$ での $\varphi(t/\sqrt{n})^n \rightarrow e^{-t^2/2}$ という収束の速さを調べれば、中心極限定理による正規分布への収束の速さがわかる。

$$\bar{\mu}_3 = E[Y_k^3] = E \left[\left(\frac{X_k - \mu}{\sigma} \right)^3 \right]$$

とおくと、

$$\varphi \left(\frac{t}{\sqrt{n}} \right) = E[e^{itY_k}] = 1 - \frac{t^2}{2n} - i\bar{\mu}_3 \frac{t^3}{6n\sqrt{n}} + O(n^{-2})$$

なので、

$$\log \varphi \left(\frac{t}{\sqrt{n}} \right)^n = n \log \left(1 - \frac{t^2}{2n} - i\bar{\mu}_3 \frac{t^3}{6n\sqrt{n}} + O(n^{-2}) \right) = -\frac{t^2}{2} - i\bar{\mu}_3 \frac{t^3}{6\sqrt{n}} + O(n^{-1}).$$

これは $n \rightarrow \infty$ での $\log \varphi(t/\sqrt{n})^n \rightarrow -t^2/2$ の収束の速さは、 $Y_k = (X_k - \mu)/\sigma$ の3次のモーメント $\bar{\mu}_3$ の絶対値の大きさで大体決まっていることがわかる。 $\bar{\mu}_3$ の絶対値が小さいほど収束が速く、大きいほど収束が遅い。

$\bar{\mu}_3$ は X_k の分布の期待値 μ を中心とする非対称性の σ によって適切に正規化した尺度になっている。 $\bar{\mu}_3$ は $Y_k = (X_k - \mu)/\sigma$ の3次のキュムラントにも一致している：

$$K_{Y_k}(t) = \log E[e^{tY_k}] = \log \left(1 + \frac{t^2}{2} + \bar{\mu}_3 \frac{t^3}{3!} + O(t^4) \right) = \frac{t^2}{2} + \bar{\mu}_3 \frac{t^3}{3!} + O(t^4).$$

ここでの $t^3/3!$ の係数 $\bar{\kappa}_3 = \bar{\mu}_3$ は X_k の歪度 (skewness) と呼ばれるのであった.

3.4 中心極限定理のキュムラント母函数を使った証明

中心極限定理: X_1, X_2, X_3, \dots が独立同分布な確率変数の列であるとき, $\mu = E[X_k]$ が定義されていて, $\sigma^2 = \text{var}(X_k) = E[(X_k - \mu)^2] < \infty$ でかつ, $E[|X_k - \mu|^3] < \infty$ となっており, さらに各 X_k のキュムラント母函数がうまく定義されているとする. このとき,

$$\bar{X}_n = \frac{1}{n} \sum_{k=1}^n X_k, \quad Z_n = \frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma}$$

とおくと, $n \rightarrow \infty$ で Z_n の分布は標準正規分布に近付く.

証明: X_k の標準化を $Y_k = (X_k - \mu)/\sigma$ と書くことにする. Y_1, Y_2, \dots も独立同分布になり, $E[Y_k] = 0$, $E[Y_k^2] = 1$ が成立している. ゆえに Y_k のキュムラント母函数 $K(t)$ は k によらず,

$$K(t) = K_{Y_k}(t) = \frac{t^2}{2} + O(t^3)$$

という形になる. そして,

$$\frac{1}{\sqrt{n}} \sum_{k=1}^n Y_k = \frac{\sqrt{n}}{\sigma} \frac{1}{n} \sum_{k=1}^n (X_k - \mu) = \frac{\sqrt{n}}{\sigma} (\bar{X}_n - \mu) = Z_n$$

なので, Z_n のキュムラント母函数は, $n \rightarrow \infty$ で

$$\begin{aligned} K_{Z_n}(t) &= K_{Y_1/\sqrt{n} + \dots + Y_n/\sqrt{n}}(t) = K_{Y_1}\left(\frac{t}{\sqrt{n}}\right) + \dots + K_{Y_1}\left(\frac{t}{\sqrt{n}}\right) \\ &= nK\left(\frac{t}{\sqrt{n}}\right) = n\left(\frac{t^2}{2n} + O(n^{-3/2})\right) = \frac{t^2}{2} + O(n^{-1/2}) \rightarrow \frac{t^2}{2} \end{aligned}$$

と標準正規分布のキュムラント母函数 $t^2/2$ に収束する.

ゆえに Z_n の分布は $n \rightarrow \infty$ で標準正規分布に近付く.

証明終

注意: このようにキュムラント母函数を使うと証明が非常にシンプルになる.

3.5 中心極限定理の収束の速さと歪度と尖度

前節の証明より, 独立同分布確率変数列 X_1, X_2, \dots に対して,

$$\mu = E[X_k], \quad \sigma = \sqrt{E[(X_k - \mu)^2]}, \quad K(t) = K_{(X_k - \mu)/\sigma}(t) = \log E\left[\left(\frac{X_k - \mu}{\sigma}\right)\right]$$

とおいたときの, $n \rightarrow \infty$ での $nK(t/\sqrt{n}) \rightarrow t^2/2$ という収束の速さを調べれば, 中心極限定理による正規分布への収束の速さがわかる. X_k の歪度(わいど) $\bar{\kappa}_3$ と尖度(せんど) $\bar{\kappa}_4$ は

$$\bar{\kappa}_3 = E\left[\left(\frac{X_k - \mu}{\sigma}\right)^3\right], \quad \bar{\kappa}_4 = E\left[\left(\frac{X_k - \mu}{\sigma}\right)^4\right] - 3$$

と表され, X_k の標準化のキュムラント母函数 $K(t)$ の展開の係数になっているのであった:

$$K(t) = \frac{t^2}{2} + \bar{\kappa}_3 \frac{t^3}{3!} + \bar{\kappa}_4 \frac{t^4}{4!} + O(t^5).$$

このとき,

$$nK\left(\frac{t}{\sqrt{n}}\right) = n\left(\frac{t^2}{2n} + \bar{\kappa}_3 \frac{t^3}{3! n^{3/2}} + \bar{\kappa}_4 \frac{t^4}{4! n^2} + O(n^{-5/2})\right) = \frac{t^2}{2} + \bar{\kappa}_3 \frac{t^3}{3! \sqrt{n}} + \bar{\kappa}_4 \frac{t^4}{4! n} + O(n^{-3/2})$$

これが $t^2/2$ に収束する速さは $\bar{\kappa}_3 \neq 0$ ならば $O(n^{-1/2})$ のオーダーになり, 歪度 $\bar{\kappa}_3$ の絶対値が大きいほど遅くなる. そして, $\bar{\kappa}_3 = 0$ ならば $O(n^{-1})$ のオーダーでの収束になり, 尖度 $\bar{\kappa}_4$ の絶対値が大きいほど収束は遅くなる.

3.6 中心極限定理のTaylorの定理のみを使う証明

以上では中心極限定理を特性函数やキュムラント母函数を使って証明する方法を(論理的には大雑把に)紹介した.

その方法は本質的にFourier解析に依存している.

実はFourier解析に依存せずに、本質的にTaylorの定理(Taylor展開の剩余項版)しか使わない初等的な証明も存在する.

しかし、読者の中にはTaylorの定理も証明をよく知らない人もいるかもしれない。以下では丁寧にTaylorの定理の証明の概略から説明して行くことにする。

3.6.1 多重積分の書き方

以下では多重積分での括弧を略して書くために、次の式の右辺を左辺のように書くこととする:

$$\int_a^b dx f(x) = \int_a^b f(x) dx.$$

このように書くと、次の右辺のように括弧が沢山必要になる多重積分の式を左辺のようにシンプルに書くことができる:

$$\int_a^x dx_1 \int_a^{x_1} dx_2 \int_a^{x_2} dx_3 g(x_3) = \int_a^x \left(\int_a^{x_1} \left(\int_a^{x_2} g(x_3) dx_3 \right) dx_2 \right) dx_1$$

しかも、この式の右辺の書き方だと、積分変数の x_i 達の指定が式の最後の方でされているので、 $\int_a^{x_i}$ の部分の解釈を確定させるためには式を最後の方まで全部見る必要が生じてしまう。左辺の書き方だとそのような問題が生じない。

3.6.2 積分剩余項型のTaylorの定理

以下では次の結果を示そう:

$$R_4(x, a) = \int_a^x dx_1 \int_a^{x_1} dx_2 \int_a^{x_2} dx_3 \int_a^{x_3} dx_4 f^{(4)}(x_4)$$

とおくと、

$$f(x) = f(a) + f'(a)(x - a) + f''(a) \frac{(x - a)^2}{2} + f'''(a) \frac{(x - a)^3}{3!} + R_4(x, a).$$

これを示すためには4階の導函数 $f^{(4)}(x)$ を4回不定積分すればよい。すなわち、「微分して不定積分するときに適切に積分定数を決めてやればもとの函数に戻る」という結果

$$g(x) = g(a) + \int_a^x dx' g'(x')$$

を4回使うと、

$$\begin{aligned} f'''(x_3) &= f'''(a) + \int_a^{x_3} dx_4 f^{(4)}(x_4), \\ f''(x_2) &= f''(a) + \int_a^{x_2} dx_3 f'''(x_3) \\ &= f''(a) + f'''(a)(x_2 - a) + \int_a^{x_2} dx_3 \int_a^{x_3} dx_4 f^{(4)}(x_4) \\ f'(x_1) &= f'(a) + \int_a^{x_1} dx_2 f''(x_2) \\ &= f'(a) + f''(a)(x_1 - a) + f'''(a) \frac{(x_1 - a)^2}{2} \\ &\quad + \int_a^{x_1} dx_2 \int_a^{x_2} dx_3 \int_a^{x_3} dx_4 f^{(4)}(x_4) \\ f(x) &= f(a) + \int_a^x dx_1 f'(x_1) \\ &= f(a) + f'(a)(x - a) + f''(a) \frac{(x - a)^2}{2} + f'''(a) \frac{(x - a)^3}{3!} \\ &\quad + \underbrace{\int_a^x dx_1 \int_a^{x_1} dx_2 \int_a^{x_2} dx_3 \int_a^{x_3} dx_4 f^{(4)}(x_4)}_{=R_4(x, a)}. \end{aligned}$$

不定積分を繰り返す過程で、

$$\int_a^x dx' \frac{(x' - a)^k}{k!} = \frac{(x - a)^{k+1}}{(k + 1)!}$$

となることから、次の形の式が得られた:

$$f(a) + f'(a)(x - a) + f''(a)\frac{(x - a)^2}{2} + f'''(a)\frac{(x - a)^3}{3!}$$

以上とまったく同様にして次を示すことができる:

$$R_n(x, a) = \int_a^x dx_1 \int_a^{x_1} dx_2 \cdots \int_a^{x_{n-1}} dx_n f^{(n)}(x_n)$$

とおくと、

$$f(x) = \sum_{k=0}^{n-1} f^{(k)}(a) \frac{(x - a)^k}{k!} + R_n(x, a).$$

この結果を **積分剩余項型のTaylorの定理** と呼ぶこととする。

もしかしたら、大学1年生のときの講義で剩余項の形が別のTaylorの定理について複雑な証明を聴いた人が多いかも知れないが、Taylorの定理の本質は「沢山微分して沢山積分すればもとの函数に戻る」ということに過ぎず、直観的にはほとんど明らかな結果に過ぎない。明らかな結果について「わけがわからないがとにかく成立する定理」だと認識してしまうことは数学を理解するときに避けるべきことである。

3.6.3 積分剩余項の別の表示

剩余項 $R_n(x, t)$ は次の式で定義されると考えてよい:

$$R_n(x, t) = f(x) - \sum_{k=0}^{n-1} f^{(k)}(t) \frac{(x - t)^k}{k!}.$$

このとき、 $R_n(x, x) = 0$ だから、両辺を t で偏微分すると、

$$\begin{aligned} \frac{\partial}{\partial t} R_n(x, t) &= \sum_{k=1}^{n-1} f^{(k)}(t) \frac{(x - t)^{k-1}}{(k - 1)!} - \sum_{k=0}^{n-1} f^{(k+1)}(t) \frac{(x - t)^k}{k!} \\ &= \sum_{k=1}^{n-1} f^{(k)}(t) \frac{(x - t)^{k-1}}{(k - 1)!} - \sum_{k=1}^n f^{(k)}(t) \frac{(x - t)^{k-1}}{(k - 1)!} \\ &= -f^{(n)}(t) \frac{(x - t)^{n-1}}{(n - 1)!}. \end{aligned}$$

この偏導函数を t について x から a まで積分することによって(a から x まで積分して -1 倍することと同じ)，次が得られる:

$$R_n(x, a) = \int_a^x f^{(n)}(t) \frac{(x - t)^{n-1}}{(n - 1)!} dt.$$

以上によっても積分剩余項型のTaylorの定理がシンプルに証明されたことになる。

Taylorの定理の証明は易しい!

注意: $h = x - a$, $t = a + sh$ とおくと、 $R_n(x, a)$ は次のようにも表される:

$$R_n(a + h, a) = h^n \int_0^1 f^{(n)}(a + sh) \frac{(1 - s)^{n-1}}{(n - 1)!} ds.$$

すなわち、

$$f(a + h) = \sum_{k=0}^{n-1} f^{(k)}(a) \frac{h^k}{k!} + h^n \int_0^1 f^{(n)}(a + sh) \frac{(1 - s)^{n-1}}{(n - 1)!} ds.$$

後でTaylorの定理をこの形で使うことになる。

注意: 上で求めた積分剩余項の公式と前節で求めた公式の関係。前節では

$$R_n(x, a) = \int_a^x dx_1 \int_a^{x_1} dx_2 \cdots \int_a^{x_{n-1}} dx_n f^{(n)}(x_n)$$

を示した。簡単のため $a \leq x$ と仮定する($a \geq x$ の場合も同様である)。そのとき、すぐ上の式の右辺の積分の範囲は

$$a \leq x_n \leq x_{n-1} \leq x_{n-2} \leq \cdots \leq x_2 \leq x_1 \leq x$$

なので, x_n による積分を一番外側に出すと次のように書き直される:

$$R_n(x, a) = \int_a^x f^{(n)}(x_n) \left(\int_{x_n}^x dx_1 \int_{x_n}^{x_1} dx_2 \cdots \int_{x_n}^{x_{n-2}} dx_{n-1} \right) dx_n.$$

$\int_{x_n}^x dx_1 \int_{x_n}^{x_1} dx_2 \cdots \int_{x_n}^{x_{n-2}} dx_{n-1}$ の部分は 1 の $n - 1$ 回の不定積分なので $(x - x_n)^{n-1}/(n - 1)!$ に等しい:

$$\int_{x_n}^x dx_1 \int_{x_n}^{x_1} dx_2 \cdots \int_{x_n}^{x_{n-2}} dx_{n-1} = \frac{(x - x_n)^{n-1}}{(n - 1)!}.$$

これより, 前節で求めた積分剩余項の公式から上で求めた公式が得られることがわかった.

3.6.4 微分剩余項型のTaylorの定理

$f(t)$ は C^n 級であると仮定する.

簡単のため $a \leq x$ と仮定する($a \geq x$ の場合も同様である). a 以上 x 以下の t についての $f^{(n)}(t)$ の最小値と最大値をそれぞれ $f^{(n)}(\alpha), f^{(n)}(\beta)$ ($a \leq \alpha, \beta \leq x$) と書くと,

$$f^{(n)}(\alpha) \frac{(x - a)^n}{n!} \leq R_n(x, a) = \int_a^x f^{(n)}(t) \frac{(x - t)^{n-1}}{(n - 1)!} dt \leq f^{(n)}(\beta) \frac{(x - a)^n}{n!}.$$

ゆえに, 中間値の定理より, α と β のあいだのある実数 ξ が存在して,

$$f^{(n)}(\xi) \frac{(x - a)^n}{n!} = R_n(x, a).$$

これで積分を使わずに微分だけを使った剩余項の表示が得られた.

3.6.5 Taylorの定理を使った中心極限定理の証明

X_1, X_2, X_3, \dots は同分布独立確率変数列であるとし, それらの共通の期待値 $\mu = E[X_i]$ は 0 であるとし, 共通の分散 $\sigma^2 = E[(X_i - \mu)^2]$ は 1 であるとし, $E[|X_i|^m] < \infty$ ($m = 3, 4, 5$) であると仮定し, X_i の歪度と尖度を次のように書くことにする:

$$\bar{\kappa}_3 = E[X_i^3], \quad \bar{\kappa}_4 = E[X_i^4] - 3.$$

このとき, Z を標準正規分布に従う確率変数であるとし,

$$Z_n = \frac{1}{\sqrt{n}} \sum_{i=1}^n X_i.$$

とおくと, 有限区間の外で 0 になる C^5 級函数 $f(z)$ に対して, $m = 3, 4, 5$ に関する $|f^{(m)}(z)|$ の最大値を M_m と書くと, $n \rightarrow \infty$ において次が成立している:

$$\begin{aligned} & |E[f(Z_n)] - E[f(Z)]| \\ & \leq \frac{M_3}{3! \sqrt{n}} |\bar{\kappa}_3| + \frac{M_4}{4! n} |\bar{\kappa}_4| + \frac{M_5}{5! n \sqrt{n}} (E[|X_i|^5] + E[|Z|^5]). \end{aligned}$$

特に次が成立している:

$$\lim_{n \rightarrow \infty} E[f(Z_n)] = E[f(Z)].$$

注意: 必要ならば X_i を $(X_i - \mu)/\sigma$ で置き換えることによって, 上のように仮定しても一般性が失われないことに注意せよ.

証明:

Step 1. Y_1, Y_2, Y_3, \dots は X_1, X_2, X_3, \dots とは別の独立同分布確率変数列であるとし, X_i, Y_i の全体は独立でかつ $E[Y_i] = 0$, $E[Y_i^2] = 1$, $E[|Y_i|^3] < \infty$ を満たしていると仮定する.

$W_n, Z_n^{(k)}, A_n^{(k)}$ を次のように定める:

$$W_n = \frac{1}{\sqrt{n}} \sum_{i=1}^n Y_i,$$

$$Z_n^{(k)} = \frac{1}{\sqrt{n}} (X_1 + \cdots + X_k + Y_{k+1} + \cdots + Y_n),$$

$$A_n^{(k)} = \frac{1}{\sqrt{n}} (X_1 + \cdots + X_{k-1} + Y_{k+1} + \cdots + Y_n).$$

このとき,

$$Z_n = Z_n^{(n)}, \quad W_n = Z_n^{(0)}, \quad Z_n^{(k)} = A_n^{(k)} + \frac{X_k}{\sqrt{n}}, \quad Z_n^{(k-1)} = A_n^{(k)} + \frac{Y_k}{\sqrt{n}}.$$

以下では、大雑把に言うと、 $Z_n^{(k)}$ の分布と $Z_n^{(k-1)}$ の分布の違いは小さいことを示す。

まず、 $G(a, h)$ を次のように定める:

$$G(a, h) = \int_0^1 f^{(5)}(a + sh) \frac{(1-s)^4}{4!} ds.$$

$m = 3, 4, 5$ に対する $|f^{(m)}(z)|$ の最大値を M_m と書くと、

$$|G(a, h)| \leq M_5 \int_0^1 \frac{(1-s)^4}{4!} ds = \frac{M_5}{5!}.$$

$f(z)$ に Taylor の定理を適用すると、

$$f(a+h) = f(a) + hf'(a) + \frac{h^2}{2} f''(a) + \frac{h^3}{3!} f'''(a) + \frac{h^4}{4!} f^{(4)}(a) + h^3 G(a, h).$$

これを $a = A_n^{(k)}$, $h = X_k/\sqrt{n}, Y_k/\sqrt{n}$ に適用すると、

$$\begin{aligned} f(Z_n^{(k)}) &= f(A_n^{(k)}) + \frac{X_k}{\sqrt{n}} f'(A_n^{(k)}) + \frac{X_k^2}{2n} f''(A_n^{(k)}) \\ &\quad + \frac{X_k^3}{3! n \sqrt{n}} f'''(A_n^{(k)}) + \frac{X_k^4}{4! n^2} f^{(4)}(A_n^{(k)}) + \frac{X_k^5}{n^2 \sqrt{n}} G\left(A_n^{(k)}, \frac{X_k}{\sqrt{n}}\right), \\ f(Z_n^{(k-1)}) &= f(A_n^{(k)}) + \frac{Y_k}{\sqrt{n}} f'(A_n^{(k)}) + \frac{Y_k^2}{2n} f''(A_n^{(k)}) \\ &\quad + \frac{Y_k^3}{3! n \sqrt{n}} f'''(A_n^{(k)}) + \frac{Y_k^4}{4! n^2} f^{(4)}(A_n^{(k)}) + \frac{Y_k^5}{n^2 \sqrt{n}} G\left(A_n^{(k)}, \frac{Y_k}{\sqrt{n}}\right). \end{aligned}$$

$A_n^{(k)}, X_k, Y_k$ は独立で、 $E[X_k] = E[Y_k] = 0$, $E[X_k^2] = E[Y_k^2] = 1$ であることより、

$$\begin{aligned} &E[f(Z_n^{(k)})] - E[f(Z_n^{(k-1)})] \\ &= \frac{E[f'''(A_n^{(k)})]}{3! n \sqrt{n}} (E[X_k^3] - E[Y_k^3]) + \frac{E[f^{(4)}(A_n^{(k)})]}{4! n^2} (E[X_k^4] - E[Y_k^4]) + \frac{1}{n^2 \sqrt{n}} \left(E \left[X_k^5 G\left(A_n^{(k)}, \frac{X_k}{\sqrt{n}}\right) \right] + E \left[Y_k^5 G\left(A_n^{(k)}, \frac{Y_k}{\sqrt{n}}\right) \right] \right) \end{aligned}$$

$m = 3, 4, 5$ に対する $|f^{(m)}(z)|$ の最大値を M_m と書くことにしてあつたので、

$$\begin{aligned} &\left| E[f(Z_n^{(k)})] - E[f(Z_n^{(k-1)})] \right| \\ &\leq \frac{M_3}{3! n \sqrt{n}} |E[X_k^3] - E[Y_k^3]| + \frac{M_4}{4! n^2} |E[X_k^4] - E[Y_k^4]| \\ &\quad + \frac{M_5}{5! n^2 \sqrt{n}} (E[|X_k|^5] + E[|Y_k|^5]). \end{aligned}$$

Step 2. X_k, Y_k の分布が k によらないことに注意しながら、これを $k = 1, 2, \dots, n$ について足し上げることによって次が得られる:

$$\begin{aligned}
& |E[f(Z_n)] - E[f(W_n)]| \\
& \leq \frac{M_3}{3! \sqrt{n}} |E[X_1^3] - E[Y_1^3]| + \frac{M_4}{4! n} |E[X_1^4] - E[Y_1^4]| \\
& + \frac{M_5}{5! n \sqrt{n}} (E[|X_1|^5] + E[|Y_1|^5]).
\end{aligned}$$

Step 3. Y_1, Y_2, Y_3, \dots の各々が標準正規分布に従っている場合. そのとき, $\sum_{i=1}^n Y_i$ は平均 0, 分散 n の正規分布に従うので, $W_n = (1/\sqrt{n}) \sum_{i=1}^n Y_i$ は標準正規分布に従う. ゆえに $E[f(W_n)] = E[f(Z)]$. さらにこのとき, $E[Y_1^3] = 0$, $E[Y_1^4] = 3$ となつていることにも注意せよ. ゆえに, この場合に Step 2 の結果を適用すると次が得られる:

$$\begin{aligned}
& |E[f(Z_n)] - E[f(Z)]| \\
& \leq \frac{M_3}{3! \sqrt{n}} |\bar{\kappa}_3| + \frac{M_4}{4! n} |\bar{\kappa}_4| + \frac{M_5}{5! n \sqrt{n}} (E[|X_i|^5] + E[|Z|^5]).
\end{aligned}$$

これが示したい不等式であった.

証明終

注意: Z を標準正規分布に従う確率変数とするとき,

$$E[|Z|^m] = \frac{\Gamma((m+1)/2)}{\sqrt{2\pi}} = \begin{cases} 1 \cdot 3 \cdots (2k-1) & (m = 2k \in 2\mathbb{Z}_{\geq 0} + 1) \\ 2^{k+1} k! / \sqrt{2\pi} & (m = 2k+1 \in 2\mathbb{Z}_{\geq 0}) \end{cases}$$

```
In [50]: 1 [(2quadgk(z → z^(2k)*pdf(Normal(), z), 0, Inf)[1],
 2 2^(k+1/2)*gamma(k+1/2)/sqrt(2π), prod(1:2:2k-1)) for k in 0:4]
```

```
Out[50]: 5-element Vector{Tuple{Float64, Float64, Int64}}:
(0.9999999999999998, 1.0000000000000002, 1)
(1.0000000000000009, 1.0000000000000002, 1)
(3.0000000000000005, 3.0000000000000004, 3)
(15.000000000000029, 15.000000000000004, 15)
(105.0000000000988, 105.0000000000001, 105)
```

```
In [51]: 1 [(2sqrt(2π)*quadgk(z → z^(2k+1)*pdf(Normal(), z), 0, Inf)[1],
 2 2^(k+1)*gamma(k+1), 2^(k+1)*factorial(k)) for k in 0:4]
```

```
Out[51]: 5-element Vector{Tuple{Float64, Float64, Int64}}:
(1.999999999999993, 2.0, 2)
(4.0, 4.0, 4)
(16.00000000000001, 16.0, 16)
(96.00000000000463, 96.0, 96)
(767.999999999999, 768.0, 768)
```

3.6.6 中心極限定理の収束の速さと歪度と尖度(再)

前節で示した不等式

$$\begin{aligned}
& |E[f(Z_n)] - E[f(Z)]| \\
& \leq \frac{M_3}{3! \sqrt{n}} |\bar{\kappa}_3| + \frac{M_4}{4! n} |\bar{\kappa}_4| + \frac{M_5}{5! n \sqrt{n}} (E[|X_i|^5] + E[|Z|^5]).
\end{aligned}$$

より, 再び, X_i 達に共通の歪度 $\bar{\kappa}_3$ と尖度 $\bar{\kappa}_4$ の大きさが中心極限定理の収束の速さにどのように関係しているかがわかる.

$\bar{\kappa}_3 = 0$ ならば中心極限定理の収束の速さのオーダーは $O(1/\sqrt{n})$ から $O(1/n)$ に速くなる. $\bar{\kappa}_3 = 0$ の場合には $\bar{\kappa}_4$ が 0 に近いと中心極限定理の収束はさらに速くなる.

3.7 問題: 中心極限定理の収束の様子のグラフ

中心極限定理による正規分布への収束の様子をコンピュータでグラフを描いて確認せよ. 収束が速い場合($n = 10$ すでに正規分布に十分に近い場合)と遅い場合の両方の例を作れ.

注意: 特に収束が遅い場合の様子を確認することが重要である. 中心極限定理は統計学において空気のごとく使用されるが, 数学的には収束が遅い場合があるので, 現実の分析で中心極限定理を使うことには注意を要する. しかし, 単に注意を要することを知っているだけでは役に立たない. 中心極限定理による収束が遅い場合について具体的な例を知っていれば, 危険を察知し易くなるだろう.

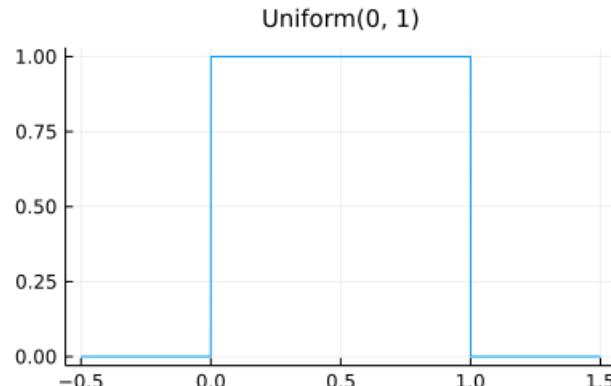
解答例は以下の通り.

In [52]:

```
1 # 中心極限定理による収束が速い場合1
2 # distを左右対称な分布でかつ「おとなしめなもの」とする
3 dist = Uniform(0, 1)
4 @show skewness(dist) kurtosis(dist)
5 plot(dist, -0.5, 1.5; label="")
6 title!("Uniform(0, 1)")
```

```
skewness(dist) = 0.0
kurtosis(dist) = -1.2
```

Out[52]:



In [53]:

```
1 # 分布distの独立同分布確率変数達(n個)の実現値(要するに乱数)を大量に(L個)生成
2 n = 10
3 L = 10^6
4 Xs = rand(dist, n, L) # Xs の s は X = (X_1, ..., X_n) 達意味(複数という意味)
```

Out[53]: 10×1000000 Matrix{Float64}:

0.164167	0.114852	0.986542	0.888	...	0.536473	0.947754	0.138456	0.0076674
0.872171	0.74528	0.541005	0.514866		0.0430303	0.736294	0.106297	0.153195
0.00141177	0.591825	0.710928	0.539015		0.487957	0.459399	0.59537	0.0062952
0.218302	0.92295	0.801769	0.389409		0.859562	0.552549	0.291198	0.298221
0.671796	0.180609	0.268402	0.0441984		0.580346	0.18101	0.313622	0.592354
0.868913	0.981163	0.335596	0.292188	...	0.582831	0.942536	0.414669	0.955135
0.874996	0.836155	0.52118	0.944482		0.60142	0.813661	0.813661	0.105761
0.882613	0.0352336	0.316746	0.78903		0.513295	0.907255	0.137298	0.487413
0.327937	0.214683	0.375124	0.633256		0.962393	0.88394	0.78909	0.93782
0.323124	0.883388	0.241298	0.250172		0.0678124	0.806403	0.073501	0.634382

In [54]:

```
1 # Z_n = √n(̄X_n - μ)/σ を大量に計算
2 μ = mean(dist)
3 σ = std(dist)
4 Zns = [√n*(mean(X) - μ)/σ for X in eachcol(Xs)] # Zns はZ_n達という意味
5 first(Zns, 5)
```

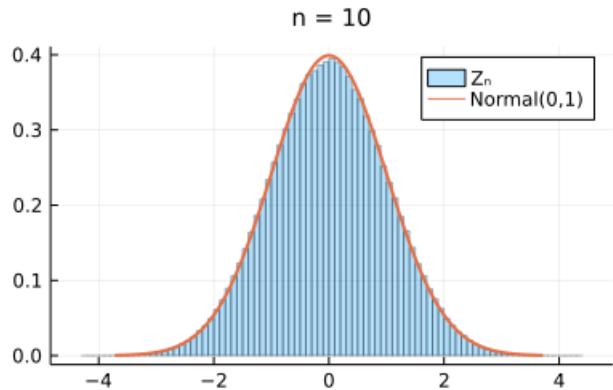
Out[54]: 5-element Vector{Float64}:

0.22503723771796913
0.5544486638722309
0.107999638970695
0.311780992831801
1.0651219742532825

In [55]:

```
1 # Z_n達のヒストグラムと標準正規分布の密度函数を比較
2 histogram(Zns; norm=true, alpha=0.3, bin=100, label="Zn")
3 plot!(Normal(0, 1); label="Normal(0,1)", lw=2)
4 title!("n = $n")
```

Out[55]:



In [56]:

```

1 # 以上の手続きを函数化
2 # さらに
3 #   * メモリアロケーションの節約と並列化による最適化
4 #   * 離散分布の場合にも対応
5 #   * 歪度と尖度を混合モデルの場合にも表示
6
7 distname(dist::Distribution) = replace(string(dist), r"{.*}" ⇒ "")
8 myskewness(dist) = skewness(dist)
9 mykurtosis(dist) = kurtosis(dist)
10 function standardized_moment(dist::ContinuousUnivariateDistribution, m)
11     μ, σ = mean(dist), std(dist)
12     quadgk(x → (x - μ)^m * pdf(dist, x), extrema(dist)...)[1] / σ^m
13 end
14 myskewness(dist::MixtureModel{Univariate, Continuous}) = standardized_moment(dist, 3)
15 mykurtosis(dist::MixtureModel{Univariate, Continuous}) = standardized_moment(dist, 4) - 3
16
17 function plot_central_limit_theorem(dist, n;
18     L=10^6,
19     μ = mean(dist),
20     σ = std(dist),
21     a = max(minimum(dist), μ - 5σ),
22     b = min(maximum(dist), μ + 5σ),
23     disttitle = distname(dist),
24     bin = 100,
25     kwargs...
26 )
27     println("skewness(dist) = ", myskewness(dist))
28     println("kurtosis(dist) = ", mykurtosis(dist))
29
30     # 分布distをプロット
31     if dist isa DiscreteUnivariateDistribution
32         P1 = bar(round(Int, a):round(Int, b), x → pdf(dist, x), ; alpha=0.3, label="")
33     else
34         P1 = plot(x → pdf(dist, x), a, b; label="")
35     end
36     title!(disttitle)
37
38     # 分布distの独立同分布確率変数達(n個)の実現値(要するに乱数)を大量に(L個)生成し,
39     #  $Z_n = \sqrt{n}(\bar{X}_n - \mu)/\sigma$  を大量に計算することを並列化で実行
40     # このような計算には並列化が非常に有効である
41     Zns = Vector{Float64}(undef, L)
42     tmp = [Vector{Float64}(undef, n) for _ in 1:Threads.nthreads()]
43     Threads.@threads for i in 1:L # Threads.@threads マクロで並列化
44         X = rand!(dist, tmp[Threads.threadid()]) # rand!を使ってアロケーションを節約
45         Zns[i] = √(n)*(mean(X) - μ)/σ
46     end
47
48     # Z達のヒストグラムと標準正規分布の密度函数を比較
49     if dist isa DiscreteUnivariateDistribution
50         summin = max(n*minimum(dist), round(n*μ - 6/n*σ))
51         summax = min(n*maximum(dist), round(n*μ + 6/n*σ))
52         sumran = summin-0.5:summax+0.5
53         bin = @. √(n)*(sumran/n - μ)/σ
54         P2 = histogram(Zns; norm=true, alpha=0.3, bin, label="Zn")
55     else
56         P2 = histogram(Zns; norm=true, alpha=0.3, bin, label="Zn")
57     end
58     plot!(Normal(0,1); label="Normal(0,1)", lw=2)
59     title!("n = $n")
60
61     plot(P1, P2; size=(800, 250), kwargs...)
62 end

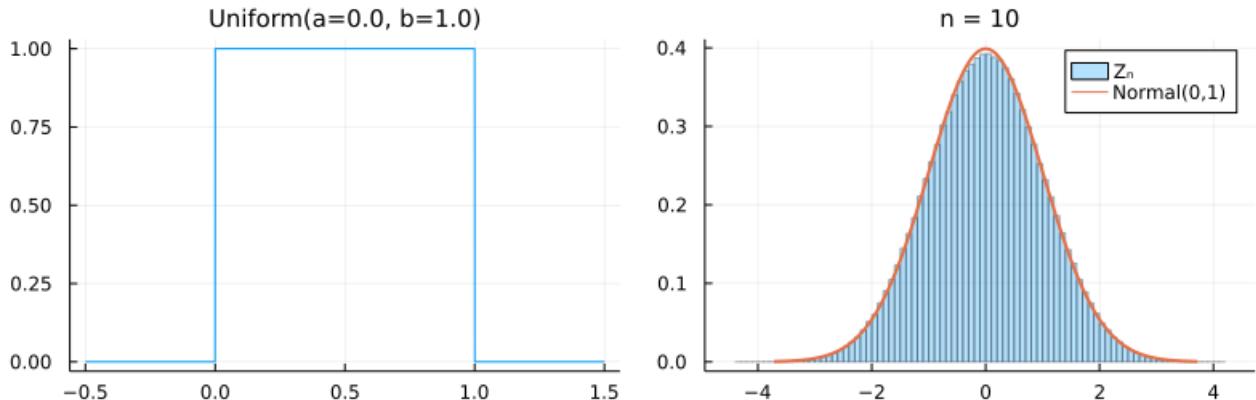
```

Out[56]: plot_central_limit_theorem (generic function with 1 method)

```
In [57]: 1 # 中心極限定理による収束が速い場合1の再現
2 plot_central_limit_theorem(Uniform(0, 1), 10; a=0.5, b=1.5)
```

skewness(dist) = 0.0
kurtosis(dist) = -1.2

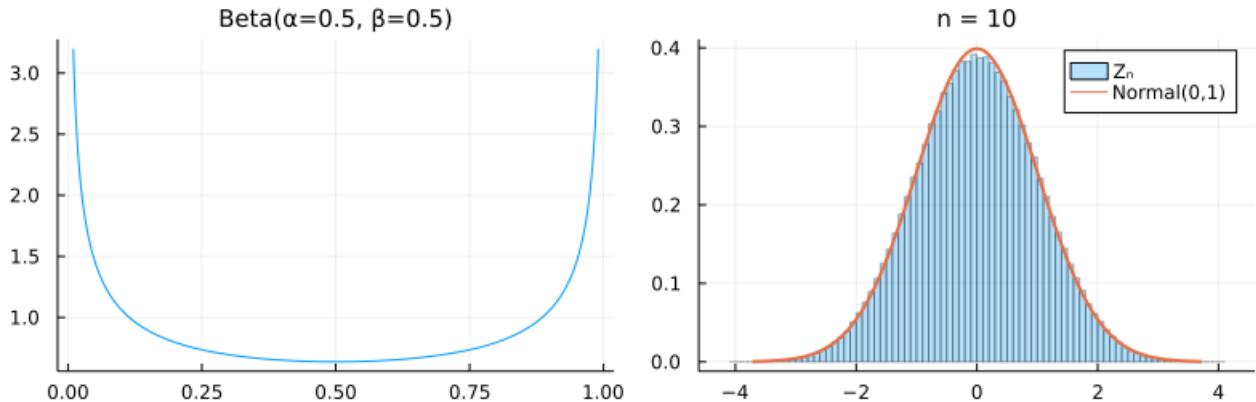
Out[57]:



```
In [58]: 1 # 中心極限定理による収束が速い場合2
2 plot_central_limit_theorem(Beta(0.5, 0.5), 10; a=0.01, b=0.99)
```

skewness(dist) = 0.0
kurtosis(dist) = -1.5

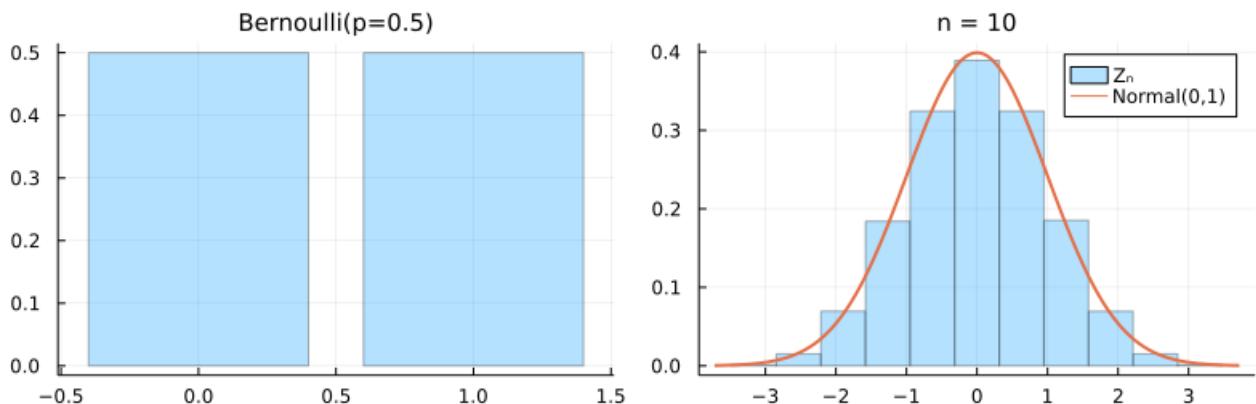
Out[58]:



```
In [59]: 1 # 中心極限定理による収束が速い場合3
2 plot_central_limit_theorem(Bernoulli(0.5), 10)
```

skewness(dist) = 0.0
kurtosis(dist) = -2.0

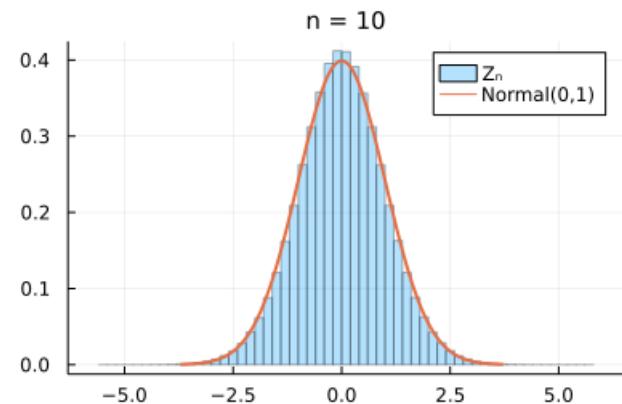
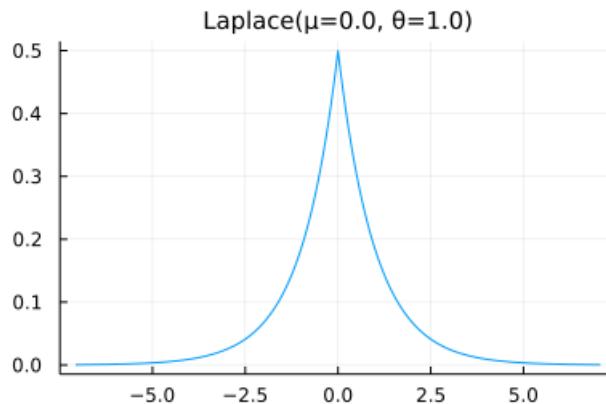
Out[59]:



```
In [60]: 1 # 中心極限定理による収束が速い場合4  
2 plot_central_limit_theorem(Laplace(), 10)
```

```
skewness(dist) = 0.0  
kurtosis(dist) = 3.0
```

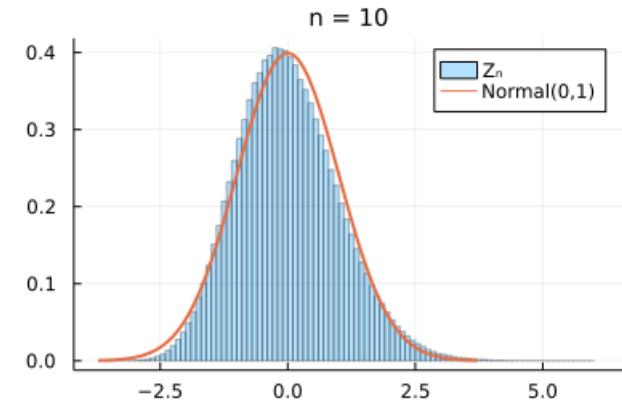
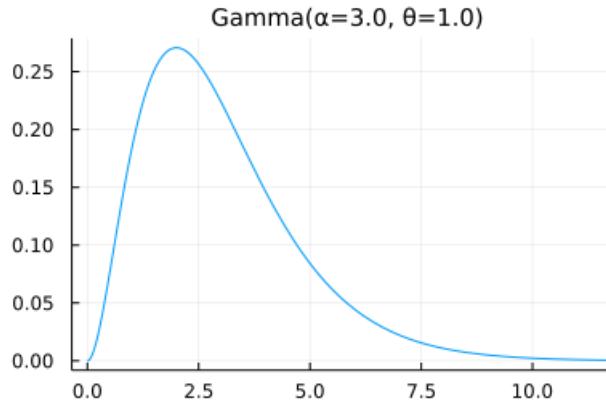
Out[60]:



```
In [61]: 1 # 左右非対称な分布の場合1  
2 plot_central_limit_theorem(Gamma(3, 1), 10)
```

```
skewness(dist) = 1.1547005383792517  
kurtosis(dist) = 2.0
```

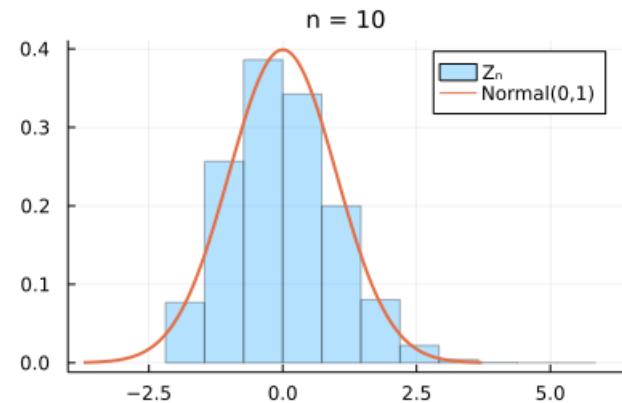
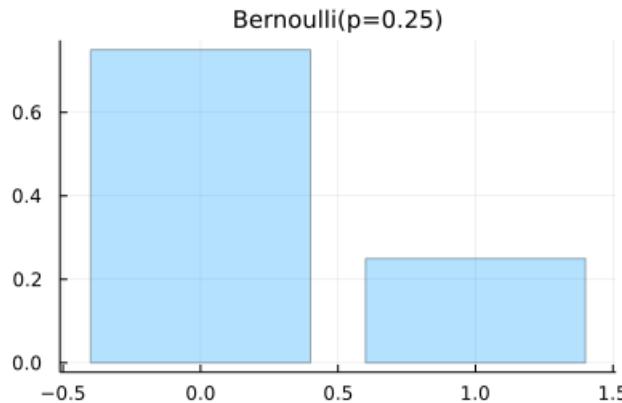
Out[61]:



```
In [62]: 1 # 左右非対称な分布の場合2  
2 plot_central_limit_theorem(Bernoulli(0.25), 10)
```

```
skewness(dist) = 1.1547005383792517  
kurtosis(dist) = -0.6666666666666667
```

Out[62]:

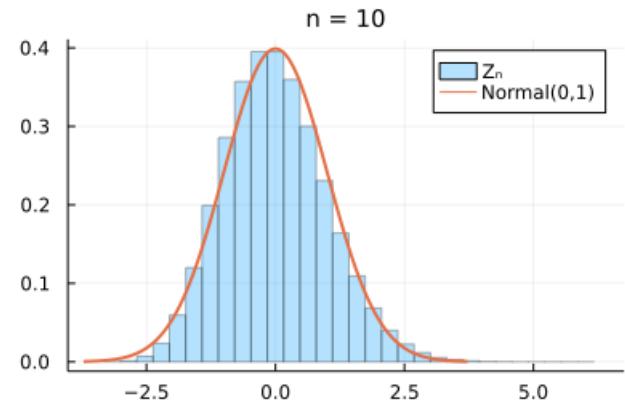
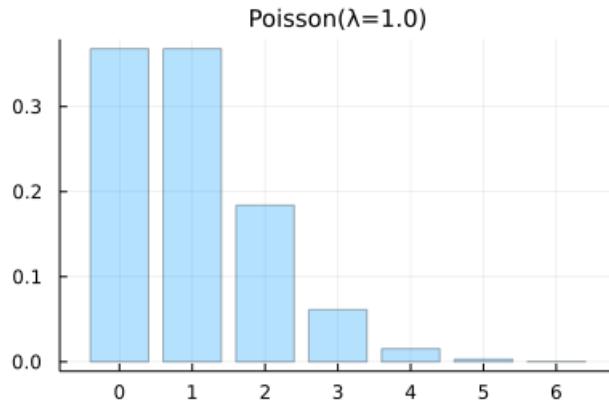


In [63]:

```
1 # 左右非対称な分布の場合3  
2 plot_central_limit_theorem(Poisson(1), 10)
```

```
skewness(dist) = 1.0  
kurtosis(dist) = 1.0
```

Out[63]:

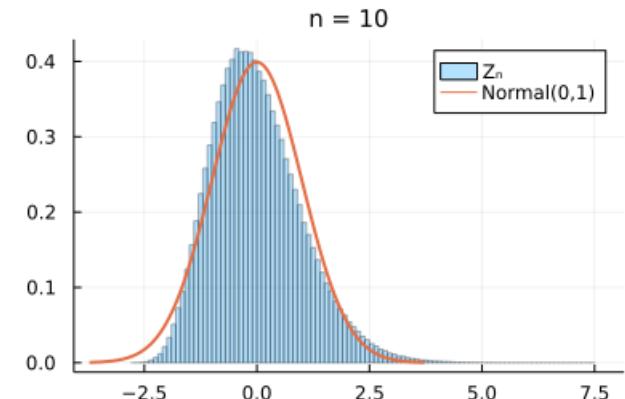
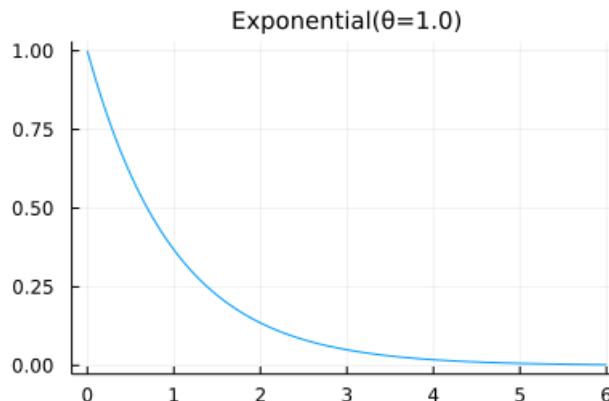


In [64]:

```
1 # 中心極限定理による収束が遅い場合1  
2 # 左右の非対称性が大きな分布を試してみる  
3 # 指数分布は左右の非対称性が大きな分布になっている  
4 plot_central_limit_theorem(Exponential(), 10)
```

```
skewness(dist) = 2.0  
kurtosis(dist) = 6.0
```

Out[64]:

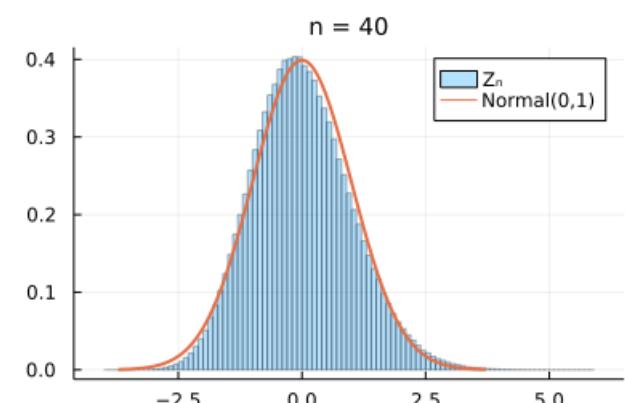
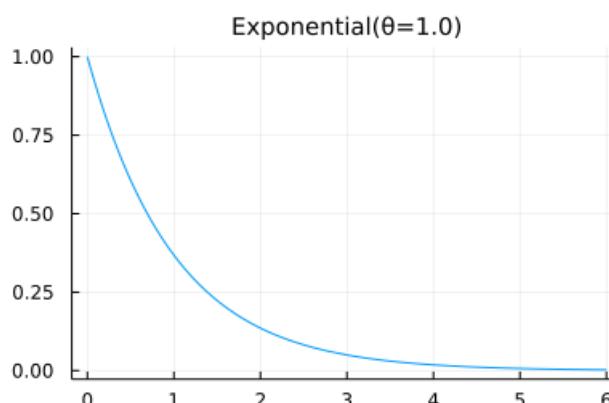


In [65]:

```
1 plot_central_limit_theorem(Exponential(), 40)
```

```
skewness(dist) = 2.0  
kurtosis(dist) = 6.0
```

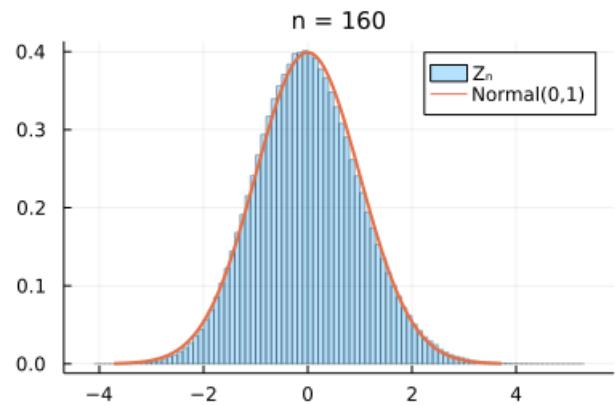
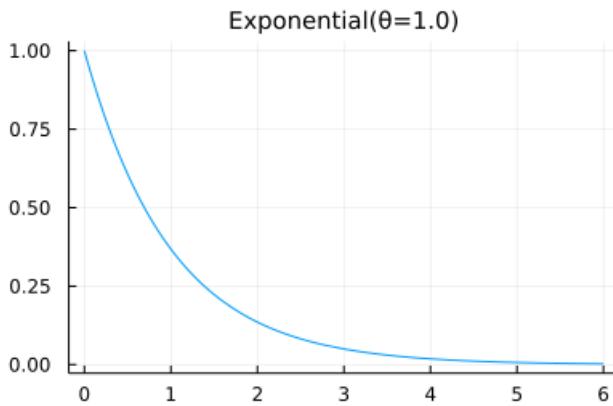
Out[65]:



```
In [66]: 1 plot_central_limit_theorem(Exponential(), 160)
```

```
skewness(dist) = 2.0  
kurtosis(dist) = 6.0
```

Out[66]:

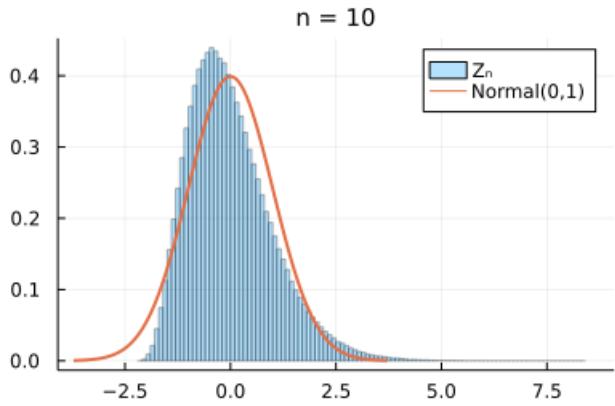
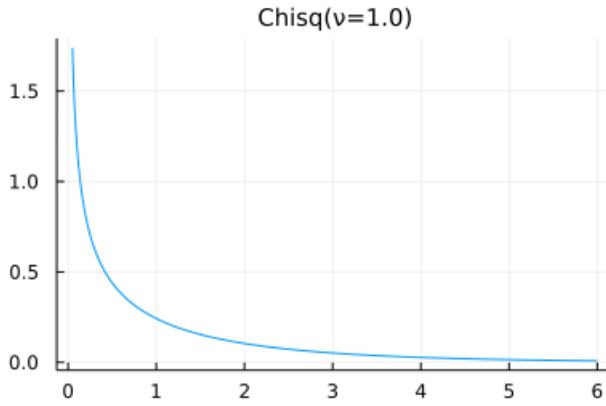


In [67]:

```
1 # 中心極限定理による収束が遅い場合2  
2 # 自由度1の $\chi^2$ 分は左右の非対称性が指数分布よりも大きな分布になっている  
3 plot_central_limit_theorem(Chisq(1), 10; a=0.05, b=6)
```

```
skewness(dist) = 2.8284271247461903  
kurtosis(dist) = 12.0
```

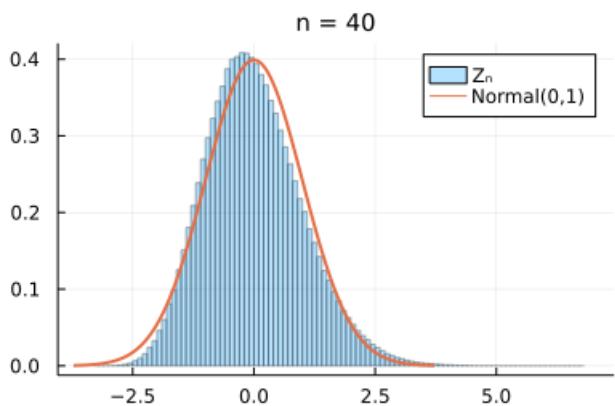
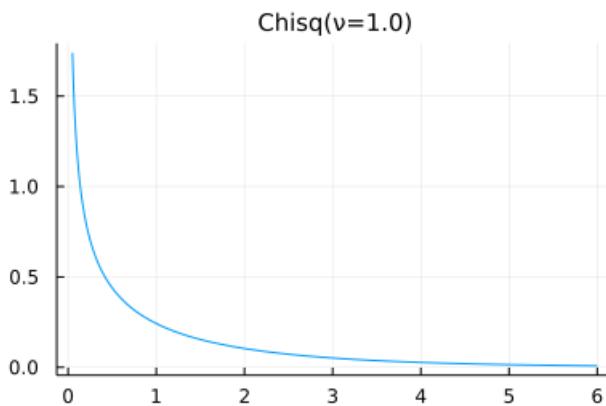
Out[67]:



```
In [68]: 1 plot_central_limit_theorem(Chisq(1), 40; a=0.05, b=6)
```

```
skewness(dist) = 2.8284271247461903  
kurtosis(dist) = 12.0
```

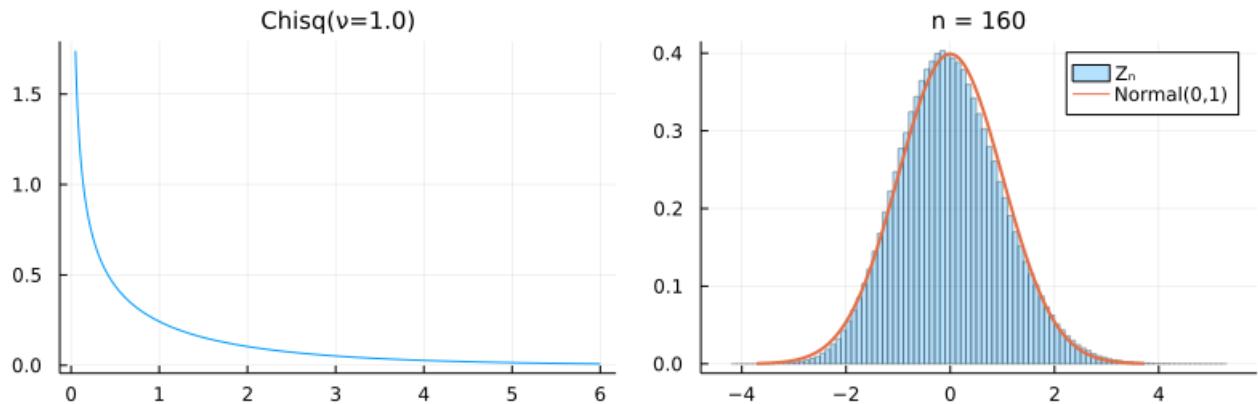
Out[68]:



```
In [69]: 1 plot_central_limit_theorem(Chisq(1), 160; a=0.05, b=6)
```

```
skewness(dist) = 2.8284271247461903
kurtosis(dist) = 12.0
```

Out[69]:



以下で使う、分布 `MixtureModel([Normal(0, 1), Normal(20, 1)], [0.95, 0.05])` の確率密度函数は次の形になる：

$$p(x) = 0.95 \frac{e^{-x^2/2}}{\sqrt{2\pi}} + 0.05 \frac{e^{-(x-20)^2/2}}{\sqrt{2\pi}}.$$

これは標準正規分布に割合が 5% の極端な外れ値を付け加えた分布になっている。何らかの原因で分布に極端な外れ値が混ざっている場合には中心極限定理を使うときに注意を要する。

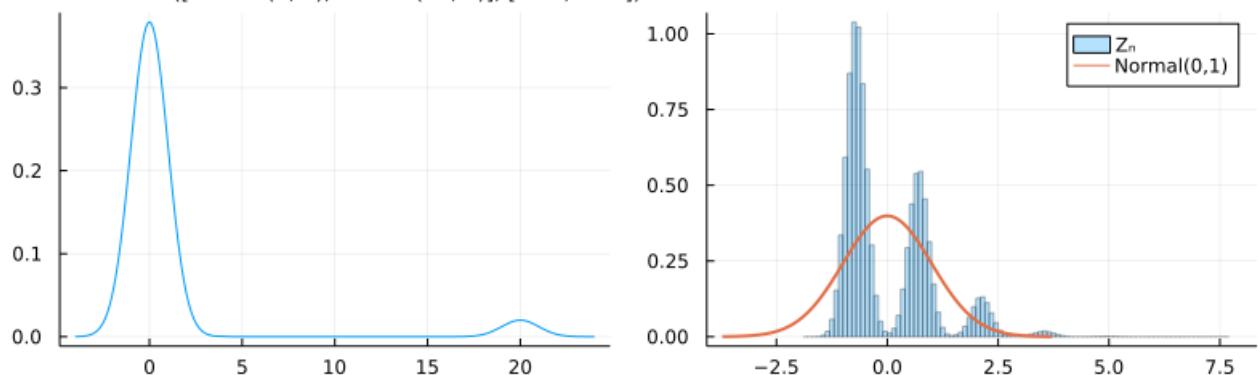
このように確率密度函数が複数の正規分布の確率密度函数の一次結合になっている分布は **混合正規分布** (mixture normal distribution)と呼ばれている。

```
In [70]: 1 # 中心極限定理による収束が遅い場合3
```

```
2 # 以下のような分布distも左右の非対称性が大きな分布に分類される
3 dist = MixtureModel([Normal(0, 1), Normal(20, 1)], [0.95, 0.05])
4 disttitle = "MixtureModel([Normal(0, 1), Normal(20, 1)], [0.95, 0.05])"
5 titlefontsize = 9
6 a, b = -4, 24
7 plot_central_limit_theorem(dist, 10; a, b, disttitle, titlefontsize)
```

```
skewness(dist) = 3.8236762415246486
kurtosis(dist) = 13.584999999990963
```

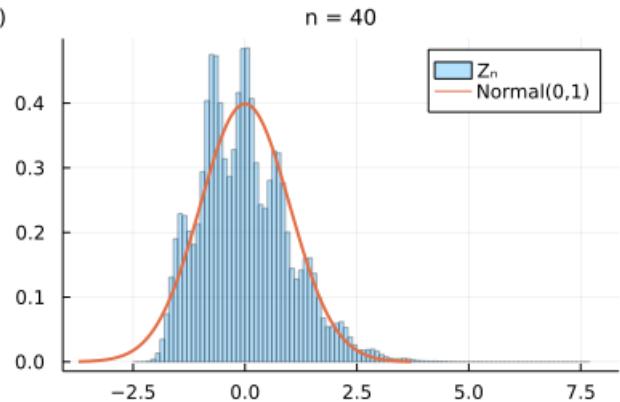
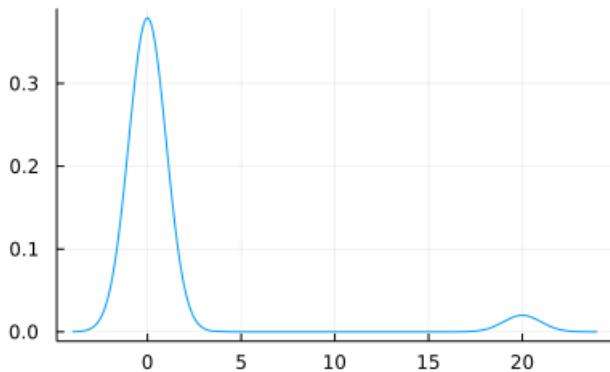
Out[70]: `MixtureModel([Normal(0, 1), Normal(20, 1)], [0.95, 0.05])`



```
In [71]: 1 plot_central_limit_theorem(dist, 40; a, b, disttitle, titlefontsize)
```

```
skewness(dist) = 3.8236762415246486  
kurtosis(dist) = 13.584999999990963
```

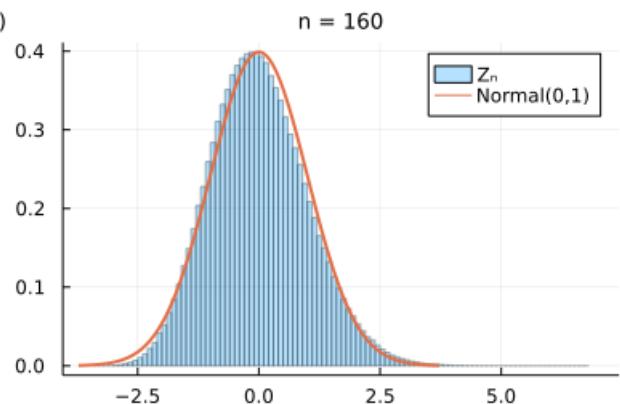
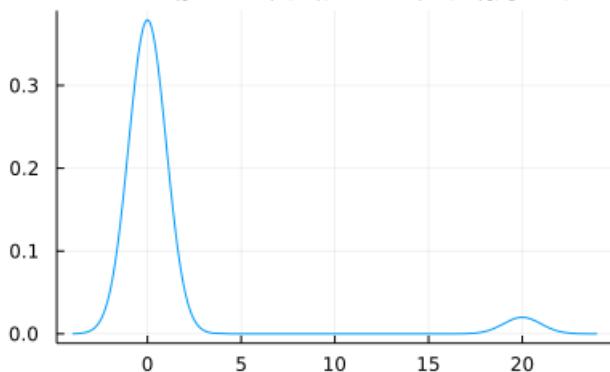
```
Out[71]: MixtureModel([Normal(0, 1), Normal(20, 1)], [0.95, 0.05])
```



```
In [72]: 1 plot_central_limit_theorem(dist, 160; a, b, disttitle, titlefontsize)
```

```
skewness(dist) = 3.8236762415246486  
kurtosis(dist) = 13.584999999990963
```

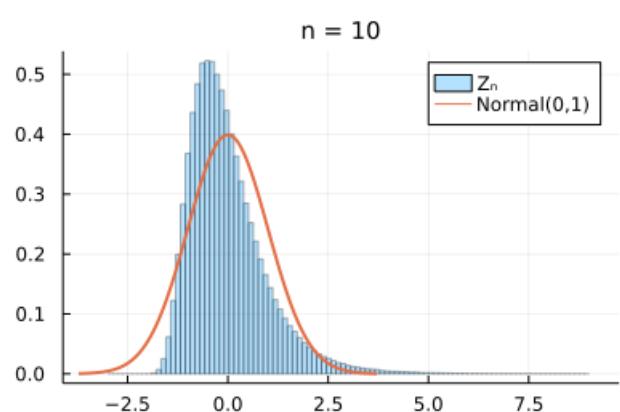
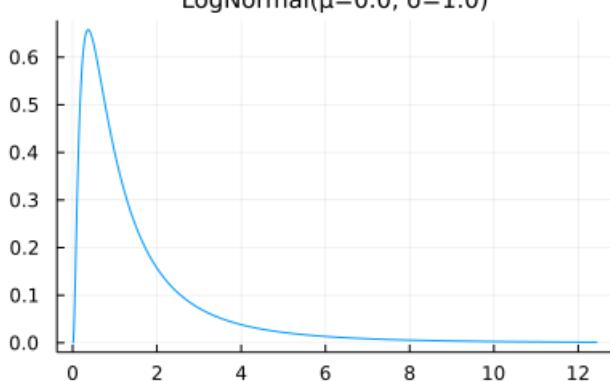
```
Out[72]: MixtureModel([Normal(0, 1), Normal(20, 1)], [0.95, 0.05])
```



```
In [73]: 1 # 中心極限定理による収束が遅い場合4  
2 # 対数正規分布は左右の非対称性が非常に大きな分布である  
3 # 右側の裾が太く、外れ値が出やすい  
4 plot_central_limit_theorem(LogNormal(), 10; bin=range(-3, 9, 100))
```

```
skewness(dist) = 6.184877138632554  
kurtosis(dist) = 110.9363921763115
```

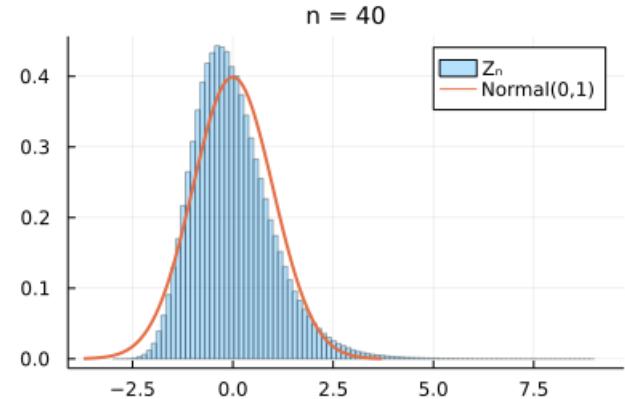
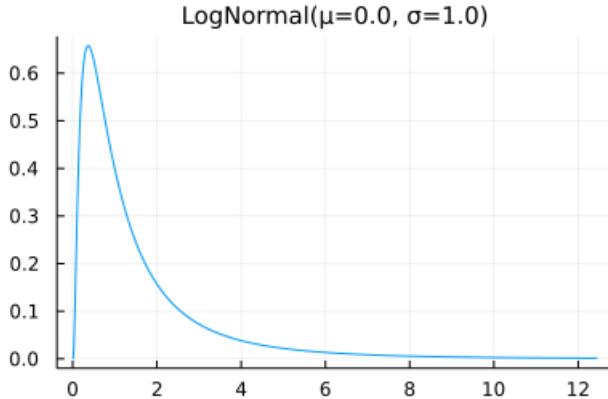
```
Out[73]: LogNormal( $\mu=0.0, \sigma=1.0$ )
```



```
In [74]: 1 plot_central_limit_theorem(LogNormal(), 40; bin=range(-3, 9, 100))
```

skewness(dist) = 6.184877138632554
kurtosis(dist) = 110.9363921763115

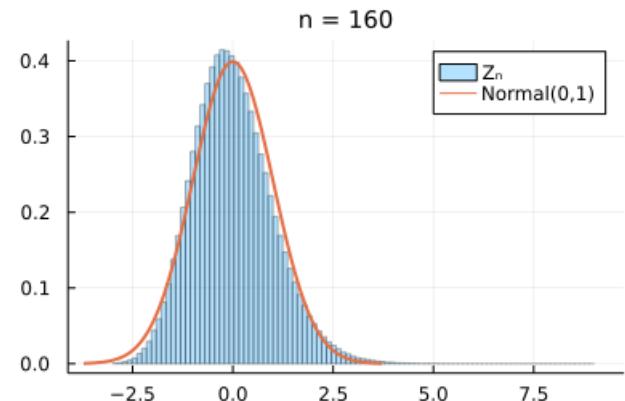
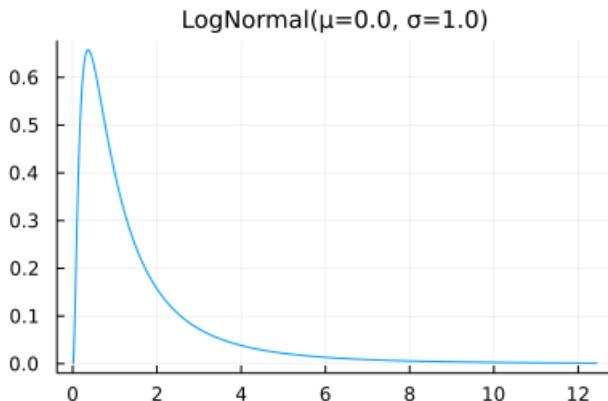
Out[74]:



```
In [75]: 1 plot_central_limit_theorem(LogNormal(), 160; bin=range(-3, 9, 100))
```

skewness(dist) = 6.184877138632554
kurtosis(dist) = 110.9363921763115

Out[75]:



4 デルタ法

4.1 1次のデルタ法と2次のデルタ法

X は平均 μ , 分散 σ^2 を持つ確率変数であるとし, 滑らかな函数 $f(x)$ にそれを代入して得られる確率変数 $Y = f(X)$ について考える.

x が μ に十分近ければ, $y = f(x)$ はその $x = \mu$ での接線

$$y = f(\mu) + f'(\mu)(x - \mu)$$

で近似される. σ が十分小さければ X の分布の大部分は μ の近くに集まり, 確率変数 $Y = f(X)$ は確率変数

$$Y_1 = f(\mu) + f'(\mu)(X - \mu)$$

で近似される. ゆえに

$$E[Y] \approx E[Y_1] = f(\mu) + f'(\mu)E[X - \mu] = f(\mu), \\ \text{var}(Y) \approx \text{var}(Y_1) = f'(\mu)^2\sigma^2$$

という近似が使える. これを **1次のデルタ法** と呼ぶことにする.

接線よりも精密に2次までの項を加えた

$$y = f(\mu) + f'(\mu)(x - \mu) + \frac{f''(\mu)}{2}(x - \mu)^2$$

は $y = f(x)$ のさらに良い近似になっている。ゆえに、確率変数 $Y = f(X)$ は、上の Y_1 よりも、

$$Y_2 = f(\mu) + f'(\mu)(X - \mu) + \frac{f''(\mu)}{2}(X - \mu)^2$$

でより良く近似される。この近似を使えば、 $E[Y] \approx E[Y_1] = f(\mu)$ よりも精度が高い

$$E[Y] \approx E[Y_2] = f(\mu) + \frac{f''(\mu)}{2}E[(X - \mu)^2] = f(\mu) + \frac{f''(\mu)\sigma^2}{2}$$

という近似が得られる。これを **2次のデルタ法** と呼ぶことにする。

4.2 おまけ: 2次のデルタ法使用時の近似分散と歪度と尖度に関する不等式

確率変数 X の期待値を μ 、分散を σ^2 、歪度を $\bar{\kappa}_3 = E[((X - \mu)/\sigma)^3]$ 、尖度を $\bar{\kappa}_4 = E[((X - \mu)/\sigma)^4] - 3$ と表し、任意の実数 a, b, c に対して、

$$Y_2 = a + b(X - \mu) + c(X - \mu)^2$$

とおく。このとき、 $E[Y_2] = a + c\sigma^2$ となることは易しい。 Y_2 の分散は以下のようにして求められる：

$$\begin{aligned} \text{var}(Y_2) &= E[(Y_2 - (a + c\sigma^2))^2] = E[(-c\sigma^2 + b(X - \mu) + c(X - \mu)^2)^2] \\ &= E[c^2\sigma^4 - 2bc\sigma^2(X - \mu) + (b^2 - 2c^2\sigma^2)(X - \mu)^2 + 2bc(X - \mu)^3 + c^2(X - \mu)^4] \\ &= c^2\sigma^4 + (b^2 - 2c^2\sigma^2)\sigma^2 + 2bc\sigma^3\bar{\kappa}_3 + c^2\sigma^4(\bar{\kappa}_4 + 3) \\ &= b^2\sigma^2 + 2bc\sigma^3\bar{\kappa}_3 + c^2\sigma^4(\bar{\kappa}_4 + 2) \\ &= [b\sigma \quad c\sigma^2] \begin{bmatrix} 1 & \bar{\kappa}_3 \\ \bar{\kappa}_3 & \bar{\kappa}_4 + 2 \end{bmatrix} \begin{bmatrix} b\sigma \\ c\sigma^2 \end{bmatrix}. \end{aligned}$$

これが任意の実数 b, c について 0 以上になることから、この結果に現れた 2×2 行列の行列式は 0 以上になることがわかるので、次の不等式が得られる：

$$\bar{\kappa}_4 + 2 \geq \bar{\kappa}_3^2.$$

「標本分布について」のノートで標本平均と不偏分散の分散共分散行列の非負性を用いて示した「歪度と尖度に関する不等式」が再び証明された。

注意: 以上の証明は本質的に $X - \mu$ と $(X - \mu)^2$ の分散共分散行列が半正定値になることを使う証明と本質的に同じである。

4.3 問題: 2次のデルタ法の不偏分散の平方根の期待値の近似への応用

X_1, X_2, \dots, X_n を独立同分布な確率変数達であるとし、それらの標本平均と不偏分散を

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i, \quad S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$$

と書き、 X_i 達共通の期待値を μ 、共通の分散を σ^2 と書き、共通の尖度を $\bar{\kappa}_4 = E[((X_i - \mu)/\sigma)^4] - 3$ と書く。このとき、 S^2 の期待値と分散は

$$E[S^2] = \sigma^2, \quad \text{var}(S^2) = \sigma^4 \left(\frac{\bar{\kappa}_4}{n} + \frac{2}{n-1} \right)$$

となるのであった。これに2次のデルタ法を適用すると、 $S = \sqrt{S^2}$ の期待値の σ からのずれが近似的に

$$E[S] \approx \sigma - \frac{\text{var}(S^2)}{8\sigma^3} = \sigma \left(1 - \frac{1}{8} \left(\frac{\bar{\kappa}_4}{n} + \frac{2}{n-1} \right) \right)$$

のようく表されることを確認せよ。

解答例: $f(x) = \sqrt{x}$ とおくと、

$$f'(x) = \frac{1}{2}x^{-1/2}, \quad f''(x) = -\frac{1}{4}x^{-3/2}, \quad f(\sigma^2) = \sigma, \quad f''(\sigma^2) = -\frac{1}{4\sigma^3}$$

なので、 $X = S^2$ に2次のデルタ法を適用すると、

$$\begin{aligned} E[S] &\approx f(\sigma^2) + \frac{f''(\sigma^2) \text{var}(S^2)}{2} \\ &= \sigma - \frac{\text{var}(S^2)}{8\sigma^3} = \sigma \left(1 - \frac{1}{8} \left(\frac{\bar{\kappa}_4}{n} + \frac{2}{n-1} \right) \right). \end{aligned}$$

解答終

注意: 不偏分散 S^2 は分散 σ^2 の不偏推定量だが、その平方根 $S = \sqrt{S^2}$ は標準偏差 σ の不偏推定量にならないことが上の問題からわかる。 $\text{var}(S^2) \geq 0$ なので $E[S]$ の値は σ よりも少し小さくなる。

注意: これを理由に $S = \sqrt{S^2}$ は標準偏差の推定量として不適切であると考えてはいけない。推定量が不偏であること自体に特別に優先度が高い価値があるわけではない。推定法に不偏性の条件を課すと誤差が増えてしまう場合がある。

In [76]:

```
1 function check_deltamethod_for_S(dist, n; L=10^7)
2     σ = std(dist)
3     ku = mykurtosis(dist)
4     @show dist
5     println("kurtosis = ", ku)
6     @show n
7
8     S = Vector{Float64}(undef, L)
9     tmp = [Vector{Float64}(undef, n) for _ in 1:Threads.nthreads()]
10    Threads.@threads for i in 1:L
11        X = rand!(dist, tmp[Threads.threadid()])
12        S[i] = std(X)
13    end
14    println("approximation = ", σ*(1 - (1/8)*(ku/n + 2/(n-1))))
15    println("mean(S) = ", mean(S))
16    println("σ = ", σ)
17 end
```

Out[76]: check_deltamethod_for_S (generic function with 1 method)

In [77]:

```
1 check_deltamethod_for_S(Normal(2, 3), 10)

dist = Normal{Float64}(\mu=2.0, σ=3.0)
kurtosis = 0.0
n = 10
approximation = 2.9166666666666665
mean(S) = 2.9181545827642346
σ = 3.0
```

In [78]:

```
1 check_deltamethod_for_S(Uniform(), 10)

dist = Uniform{Float64}(a=0.0, b=1.0)
kurtosis = -1.2
n = 10
approximation = 0.28498650787499025
mean(S) = 0.2847306864542023
σ = 0.28867513459481287
```

In [79]:

```
1 check_deltamethod_for_S(Gamma(4, 3), 10)

dist = Gamma{Float64}(α=4.0, θ=3.0)
kurtosis = 1.5
n = 10
approximation = 5.720833333333333
mean(S) = 5.757149279188369
σ = 6.0
```

In [80]:

```
1 check_deltamethod_for_S(Bernoulli(0.4), 10)

dist = Bernoulli{Float64}(p=0.4)
kurtosis = -1.833333333333333
n = 10
approximation = 0.4875165001955964
mean(S) = 0.4859474505692021
σ = 0.4898979485566356
```

In [81]:

```
1 check_deltamethod_for_S(LogNormal(), 10; L=10^6)

dist = LogNormal{Float64}(\mu=0.0, σ=1.0)
kurtosis = 110.9363921763115
n = 10
approximation = -0.895778896909659
mean(S) = 1.7197020645384062
σ = 2.1611974158950877
```

In [82]: 1 check_deltamethod_for_S(LogNormal(), 100; L=10^6)

```
dist = LogNormal{Float64}(\mu=0.0, σ=1.0)
kurtosis = 110.9363921763115
n = 100
approximation = 1.856045541537819
mean(S) = 2.0440841038908797
σ = 2.1611974158950877
```

In [83]: 1 check_deltamethod_for_S(LogNormal(), 1000; L=10^6)

```
dist = LogNormal{Float64}(\mu=0.0, σ=1.0)
kurtosis = 110.9363921763115
n = 1000
approximation = 2.1306871451883986
mean(S) = 2.1406390439132257
σ = 2.1611974158950877
```

4.4 問題: 1次のデルタ法の正規分布近似への応用

確率変数 X が近似的に正規分布 $\text{Normal}(\mu, \sigma)$ に従っているとき, 滑らかな函数 $f(x)$ に対して, σ が十分に小さければ, 確率変数 $Y = f(X)$ は近似的に正規分布 $\text{Normal}(f(\mu), |f'(\mu)|\sigma)$ に従うことを示す.

解答例: x が μ に十分近ければ $y = f(x)$ は $x = \mu$ における接線 $y = f(\mu) + f'(\mu)(x - \mu)$ で近似される. σ が十分に小さければ確率変数 X の分布の大部分は μ の十分近くに留まる. 以上の状況のもとで, 確率変数 $Y = f(X)$ は確率変数 $Y_1 = f(\mu) + f'(\mu)(X - \mu)$ で近似される. X が正規分布 $\text{Normal}(\mu, \sigma)$ に近似的に従っていると仮定しているので, そのアフィン変換である Y_1 も近似的に正規分布に従い, その期待値と分散は

$$E[Y_1] = f(\mu) + f'(\mu)E[X - \mu] \approx f(\mu), \quad \text{var}(Y_1) = f'(\mu)^2 \text{var}(X) \approx f'(\mu)^2 \sigma^2$$

となる. 以上をまとめると, Y が近似的に正規分布 $\text{Normal}(f(\mu), |f'(\mu)|\sigma)$ に従うことがわかる.

解答終

注意: デルタ法を使うと, 正規分布で近似される確率変数から正規分布に近似的に従う別の確率変数を作れる. 中心極限定理による正規分布近似はデルタ法を使えば大幅に拡張される. 以上では一変数函数 $f(x)$ に正規分布に近似的に従う確率変数 X を代入する場合のみを扱ったが, 多変数函数に複数の正規分布に近似的に従う確率変数達を代入する場合も同じように一次近似で考えればよい.

4.5 問題: 1次のデルタ法による分布の逆数の近似

正値の確率変数 X は正規分布 $\text{Normal}(\mu, \sigma)$ に近似的に従っていると仮定し, σ/μ は十分に小さいと仮定する. このとき, $Y = 1/X$ が正規分布 $\text{Normal}(1/\mu, \sigma^2/\mu^2)$ に近似的に従うことを示せ.

解答例: $f(x) = 1/x$ とおくと,

$$f(x) = \frac{1/\mu}{1 + (x/\mu - 1)} = \frac{1}{\mu} \left(1 - \left(\frac{x}{\mu} - 1 \right) + \left(\frac{x}{\mu} - 1 \right)^2 - \dots \right)$$

なので, $y = f(x)$ は x/μ が 1 に近ければ, その $x = \mu$ での接線

$$y = \frac{1}{\mu} \left(1 - \left(\frac{x}{\mu} - 1 \right) \right) = \frac{1}{\mu} - \frac{x - \mu}{\mu^2}$$

で近似される. σ/μ が十分に小さければ X/μ の分布の大部分は 1 の近くに留まる. ゆえに $Y = 1/X$ の分布は $Y_1 = 1/\mu - (X - \mu)/\mu^2$ の分布で近似される. このことから, $Y = 1/X$ が平均 $1/\mu$, 分散 σ^2/μ^4 の正規分布に近似的に従うことがわかる.

解答終

4.6 例: 自由度が大きな χ^2 分布の逆数の近似

例えば, $X \sim \text{Chisq}(v)$ のとき, $E[X] = v$, $\text{var}(X) = 2v$ であり, v が大きなとき X は近似的に平均 $\mu = v$, 分散 $\sigma^2 = 2v$ の正規分布に近似的に従い, v が大きなとき $\sigma/\mu = \sqrt{2/v}$ の値は小さくなる. ゆえに上の問題の結果より, $1/X$ は平均 $1/\mu = 1/v$, 分散 $\sigma^2/\mu^4 = 2/v^3$ の正規分布に近似的に従う.

注意: $X \sim \text{Chisq}(v) = \text{Gamma}(v/2, 2)$ のとき, $1/X \sim \text{InverseGamma}(v/2, 1/2)$ なので,

$$E[1/X] = \frac{1/2}{v/2 - 1} = \frac{1}{v-2} \approx \frac{1}{v},$$

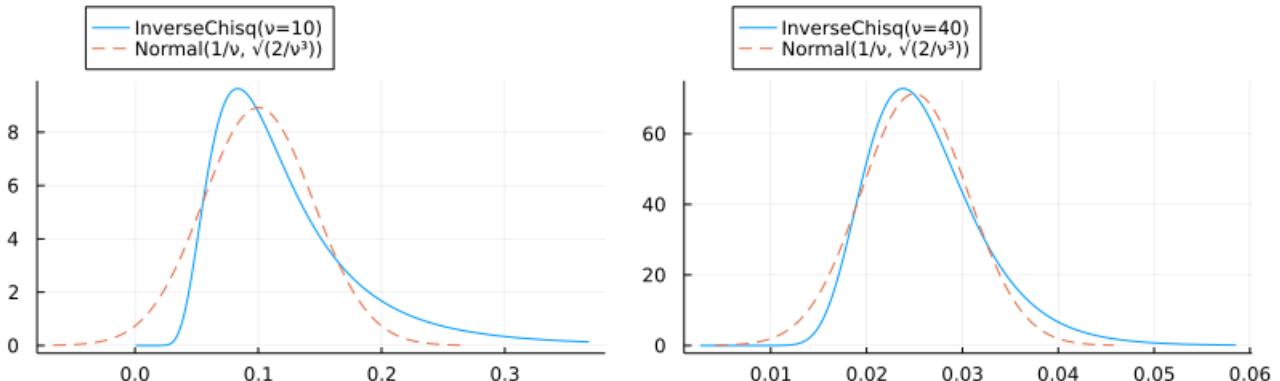
$$\text{var}(1/X) = \frac{(1/2)^2}{(v/2 - 1)^2(v/2 - 2)} = \frac{2}{(v-2)^2(v-4)} \approx \frac{2}{v^3}.$$

```
In [84]: 1 InverseChisq(v) = InverseGamma(v/2, 1/2)
2
3 function plot_invchisq(v)
4     xlim = (max(0, 1/v - 4\sqrt(2/v^3)), 1/v + 6\sqrt(2/v^3))
5     plot(InverseChisq(v), xlim..., label="InverseChisq(v=$v)")
6     plot!(Normal(1/v, \sqrt(2/v^3)); label="Normal(1/v, \sqrt(2/v^3))", ls=:dash)
7     plot!(; legend=:outertop)
8 end
```

Out[84]: plot_invchisq (generic function with 1 method)

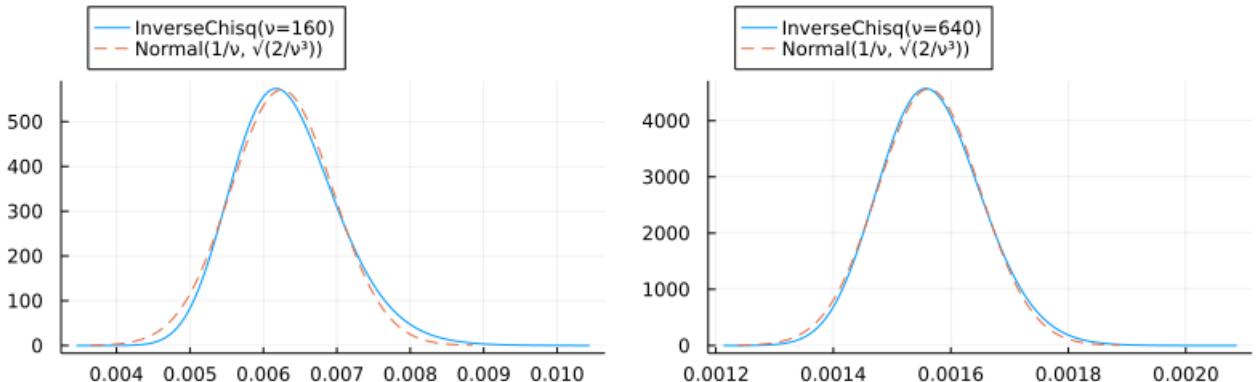
```
In [85]: 1 plot(plot_invchisq.((10, 40))...; size=(800, 250), layout=(1, 2))
```

Out[85]:



```
In [86]: 1 plot(plot_invchisq.((160, 640))...; size=(800, 250), layout=(1, 2))
```

Out[86]:



4.7 問題: デルタ法による分布の対数の近似

正値の確率変数 X は正規分布 $\text{Normal}(\mu, \sigma)$ に近似的に従っていると仮定し, σ/μ は十分に小さいと仮定する. このとき, $Y = \log X$ が正規分布 $\text{Normal}(\log \mu, \sigma^2/\mu^2)$ に近似的に従うことを示せ.

解答例: $f(x) = \log x$ とおくと,

$$f(x) = \log \nu \left(1 + \left(\frac{x}{\mu} - 1 \right) \right) = \log \mu + \left(\frac{x}{\mu} - 1 \right) - \frac{1}{2} \left(\frac{x}{\mu} - 1 \right)^2 + \dots$$

なので, $y = f(x)$ は x/μ が 1 に近ければ, その $x = \mu$ での接線

$$y = \log \mu + \left(\frac{x}{\mu} - 1 \right) = \log \mu + \frac{x - \mu}{\mu}$$

で近似される. σ/μ が十分に小さければ X/μ の分布の大部分は 1 の近くに留まる. ゆえに $Y = \log X$ の分布は $Y_1 = \log \mu + (X - \mu)/\mu$ の分布で近似される. このことから, $Y = \log X$ が平均 $\log \mu$, 分散 σ^2/μ^2 の正規分布に近似的に従うことがわかる.

解答終

注意: 正値の確率変数の分布は対数を取ることによって左右の非対称性が緩和されて、正規分布近似がより正確になる場合がある。次の例を見よ。

4.8 例: 自由度が大きな χ^2 分布の対数の近似

例えば、 $X \sim \text{Chisq}(v)$ のとき、 $E[X] = v$, $\text{var}(X) = 2v$ であり、 v が大きなとき X は近似的に平均 $\mu = v$, 分散 $\sigma^2 = 2v$ の正規分布に近似的に従い、 v が大きなとき $\sigma/\mu = \sqrt{2/v}$ の値は小さくなる。ゆえに上の問題の結果より、 $\log X$ は平均 $\log \mu = \log v$, 分散 $\sigma^2/\mu^2 = 2/v$ の正規分布に近似的に従う。

この場合には余り大きくない自由度 v における χ^2 分布の左右の非対称性が対数を取ることによって緩和され、正規分布による近似の精度が高まる。

In [87]:

```

1 pdf_logchisq(v, y) = pdf(Chisq(v), exp(y))*exp(y)
2
3 function plot_logchisq(v)
4   xlim = (max(0, v - 4\sqrt(2v)), v + 4\sqrt(2v))
5   P0 = plot(Chisq(v), xlim..., label="Chisq(v=$v)")
6   plot!(Normal(v, \sqrt(2v)); label="Normal(v, \sqrt(2v))", ls=:dash)
7   plot!(; legend=:outertop)
8
9   xlim = (log(v) - 5\sqrt(2/v), log(v) + 4\sqrt(2/v))
10  P1 = plot(y → pdf(Chisq(v), exp(y))*exp(y), xlim..., label="LogChisq(v=$v)")
11  plot!(Normal(log(v), \sqrt(2/v)); label="Normal(log v, \sqrt(2/v))", ls=:dash)
12  plot!(; legend=:outertop)
13
14  plot(P0, P1; size=(800, 250))
15 end

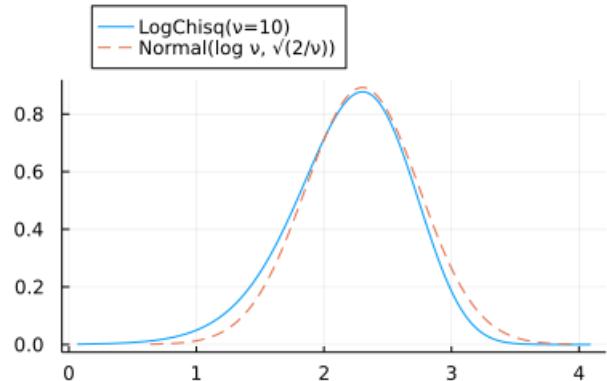
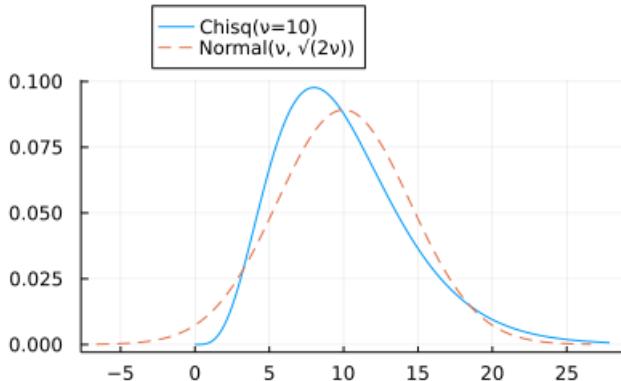
```

Out[87]: plot_logchisq (generic function with 1 method)

In [88]:

```
1 plot_logchisq(10)
```

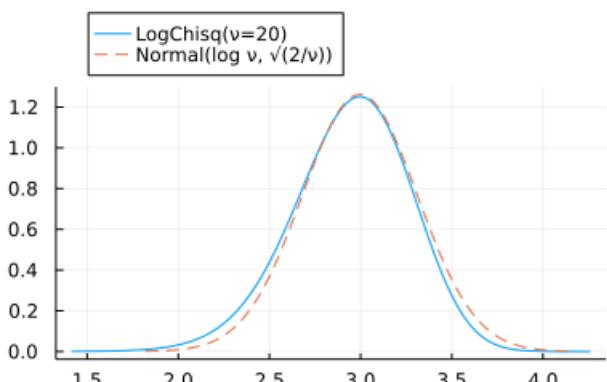
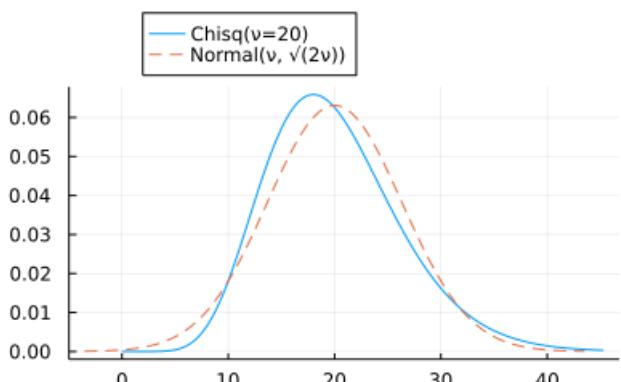
Out[88]:



In [89]:

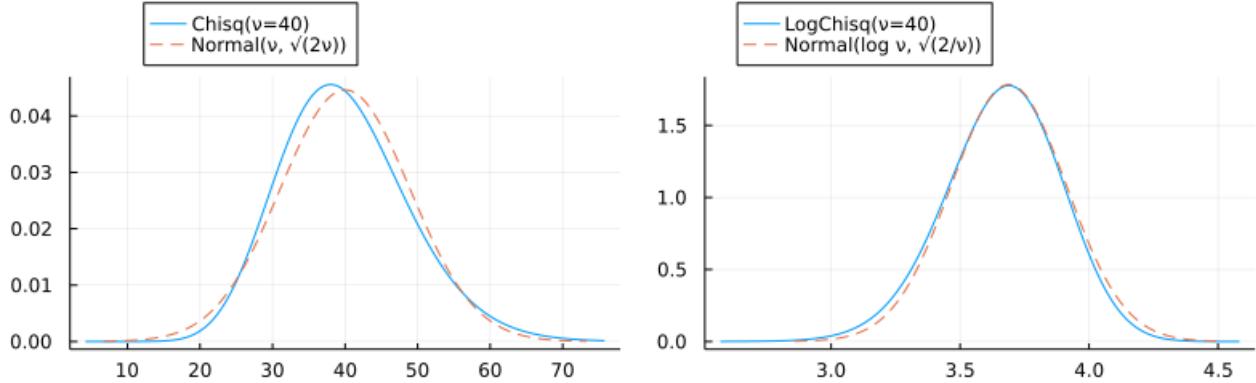
```
1 plot_logchisq(20)
```

Out[89]:



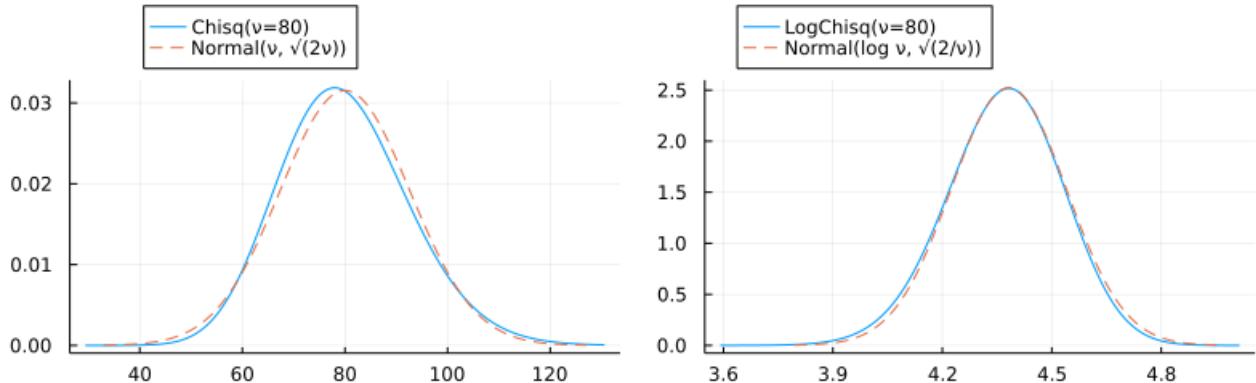
In [90]: 1 plot_logchisq(40)

Out[90]:



In [91]: 1 plot_logchisq(80)

Out[91]:



4.9 問題: ベータ分布の正規分布近似

確率変数達 X, Y は独立であり, $X \sim \text{Gamma}(\alpha, 1), Y \sim \text{Gamma}(\beta, 1)$ を満たしていると仮定する. このとき $T = X/(X + Y)$ とおくと, $T \sim \text{Beta}(\alpha, \beta)$ となるのであった. α, β が大きなとき, X, Y の分布はそれぞれ $\text{Normal}(\alpha, \sqrt{\alpha}), \text{Normal}(\beta, \sqrt{\beta})$ に近似的に従う. 2変数函数 $f(x, y) = x/(x + y)$ に1次のデルタ法を適用して, α, β が大きなとき, $T = X/(X + Y) \sim \text{Beta}(\alpha, \beta)$ が

$$\text{Normal}\left(\frac{\alpha}{\alpha + \beta}, \sqrt{\frac{\alpha\beta}{(\alpha + \beta)^3}}\right)$$

に近似的に従うことを示せ.

注意: $T \sim \text{Beta}(\alpha, \beta)$ の正確な期待値と分散は次のようにになる:

$$E[T] = \frac{\alpha}{\alpha + \beta}, \quad \text{var}(T) = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}.$$

解答例: $f(x, y) = x/(x + y) = 1 - y/(x + y)$ を偏微分すると,

$$\begin{aligned} f_x(x, y) &= \frac{y}{(x + y)^2}, & f_y(x, y) &= \frac{-x}{(x + y)^2}, \\ f_{xx}(x, y) &= \frac{-2y}{(x + y)^3}, & f_{xy}(x, y) &= \frac{x - y}{(x + y)^3}, & f_{yy}(x, y) &= \frac{2x}{(x + y)^3} \end{aligned}$$

なので,

$$\begin{aligned}
T &= \frac{X}{X+Y} = f(X, Y) \\
&= f(\alpha, \beta) + f_x(\alpha, \beta)(X - \alpha) + f_y(\alpha, \beta)(Y - \beta) \\
&\quad + \frac{1}{2}(f_{xx}(\alpha, \beta)(X - \alpha)^2 + 2f_{xy}(\alpha, \beta)(X - \alpha)(Y - \beta) + f_{yy}(\alpha, \beta)(Y - \beta)^2) + \dots \\
&= \frac{\alpha}{\alpha + \beta} + \frac{\beta}{(\alpha + \beta)^2}(X - \alpha) + \frac{-\alpha}{(\alpha + \beta)^2}(Y - \beta) \\
&\quad + \frac{-\beta}{(\alpha + \beta)^3}(X - \alpha)^2 + \frac{\alpha - \beta}{(\alpha + \beta)^3}(X - \alpha)(Y - \beta) + \frac{\alpha}{(\alpha + \beta)^3}(Y - \beta)^2 + \dots.
\end{aligned}$$

ゆえに, α, β の両方が大きいならば,

$$\frac{\text{std}(X)}{\alpha + \beta} = \frac{\sqrt{\alpha}}{\alpha + \beta}, \quad \frac{\text{std}(Y)}{\alpha + \beta} = \frac{\sqrt{\beta}}{\alpha + \beta}$$

がどちらも小さくなり, 1次近似

$$T \approx \frac{\alpha}{\alpha + \beta} + \frac{\beta}{(\alpha + \beta)^2}(X - \alpha) + \frac{-\alpha}{(\alpha + \beta)^2}(Y - \beta)$$

が有効になり,

$$\begin{aligned}
E[T] &\approx \frac{\alpha}{\alpha + \beta}, \\
\text{var}(T) &\approx \frac{\beta^4 \text{var}(X)}{(\alpha + \beta)^4} + \frac{\alpha^2 \text{var}(Y)}{(\alpha + \beta)^4} = \frac{\beta^2 \alpha + \alpha^2 \beta}{(\alpha + \beta)^4} = \frac{\alpha \beta}{(\alpha + \beta)^3}.
\end{aligned}$$

さらにそのとき, X, Y の分布は正規分布で近似されるので, T の分布も正規分布で近似される.

解答終

```
In [92]: 1 @vars x y
2 expr = x/(x+y)
3 [
4     expr
5     diff(expr, x).simplify()
6     diff(expr, y).simplify()
7     ;;
8     diff(expr, x, x).simplify()
9     diff(expr, x, y).simplify()
10    diff(expr, y, y).simplify()
11 ]
```

```
Out[92]: 
$$\left[ \begin{array}{cc} \frac{x}{x+y} & -\frac{2y}{(x+y)^3} \\ \frac{y}{(x+y)^2} & \frac{x-y}{(x+y)^3} \\ -\frac{x}{(x+y)^2} & \frac{2x}{(x+y)^3} \end{array} \right]$$

```

注意: X_1, X_2, \dots, X_n が独立同分布な確率変数達で $X_i \sim \text{Gamma}(\alpha/n, \theta)$ のとき, それらの和もガンマ分布に $X_1 + X_2 + \dots + X_n \sim \text{Gamma}(\alpha, \theta)$ と従う. この事実と中心極限定理から, α が大きなときのガンマ分布 $\text{Gamma}(\alpha, \theta)$ が正規分布 $\text{Normal}(\alpha, \sqrt{\alpha\theta^2})$ で近似されることが導かれる.

ゆえに, 上の問題の結果は, パラメータの値が大きなときのベータ分布が正規分布で近似されることが, ガンマ分布に関する中心極限定理の帰結になっていることを示していることになる.

ベータ分布の確率密度函数の形から直接的に正規分布近似を導くことも易しい.

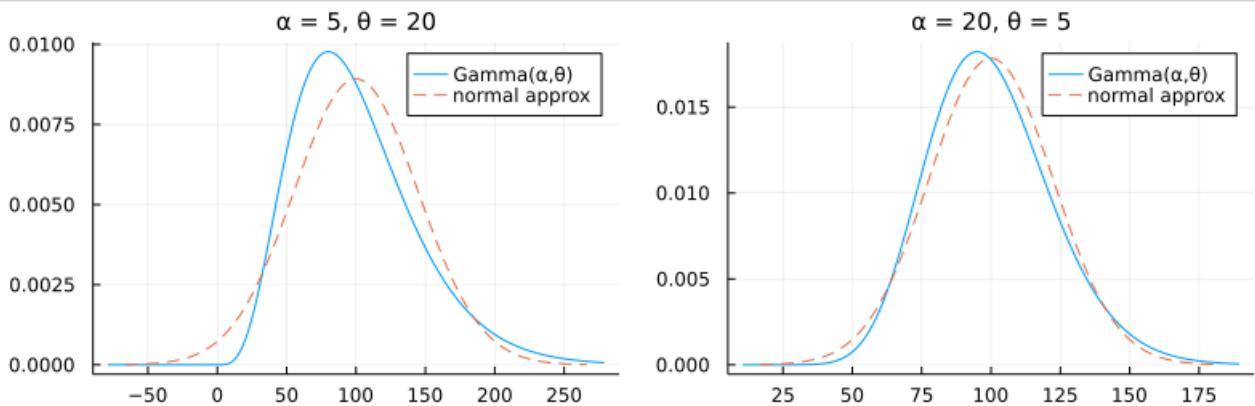
注意終

```
In [93]: 1 function plot_gammanormal(α, θ)
2   m, s = α*θ, √(α*θ^2)
3   xlim = (m - 4s, m + 4s)
4   plot(Gamma(α, θ), xlim...; label="Gamma(α,θ)")
5   plot!(Normal(m, s); label="normal approx", ls=:dash)
6   #plot!(; legend=:outertop)
7   title!("α = $α, θ = $θ")
8 end
```

Out[93]: plot_gammanormal (generic function with 1 method)

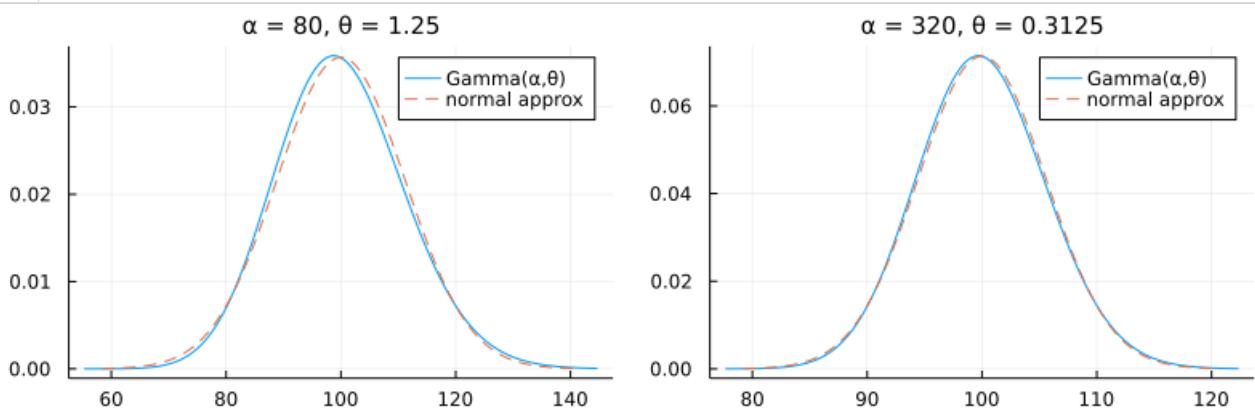
```
In [94]: 1 plot(plot_gammanormal(5, 20), plot_gammanormal(20, 5); size=(800, 250))
```

Out[94]:



```
In [95]: 1 plot(plot_gammanormal(80, 1.25), plot_gammanormal(320, 0.3125); size=(800, 250))
```

Out[95]:



```
In [96]: 1 function plot_betanormal(α, β)
2   m, s = α/(α+β), √(α*β/(α+β)^3)
3   xlim = (m - 4s, m + 4s)
4   plot(Beta(α, β), xlim...; label="Beta(α,β)")
5   plot!(Normal(m, s); label="normal approx", ls=:dash)
6   #plot!(; legend=:outertop)
7   title!("α = $α, β = $β")
8 end
```

Out[96]: plot_betanormal (generic function with 1 method)

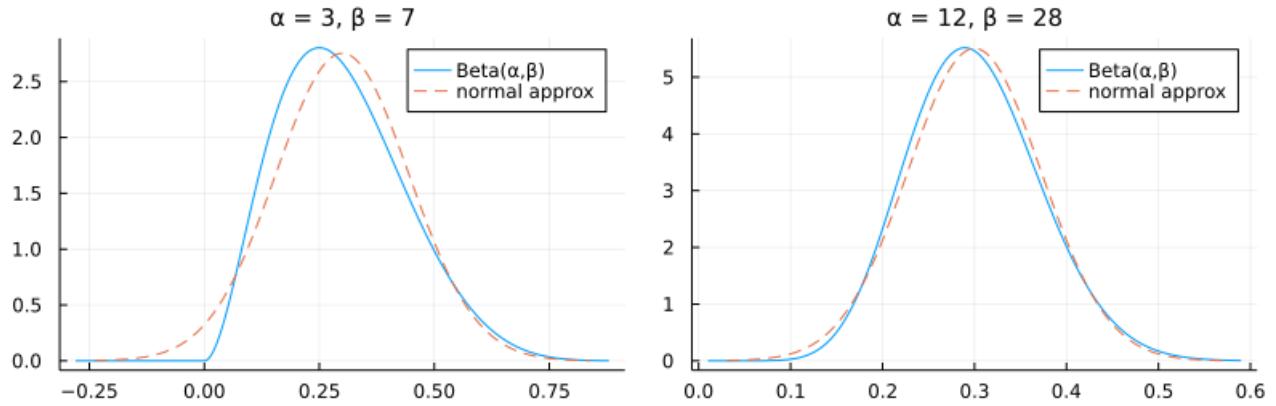
```
In [97]: 1 [3, 7] .* (4 .^ (0:3))'
```

Out[97]: 2×4 Matrix{Int64}:

3	12	48	192
7	28	112	448

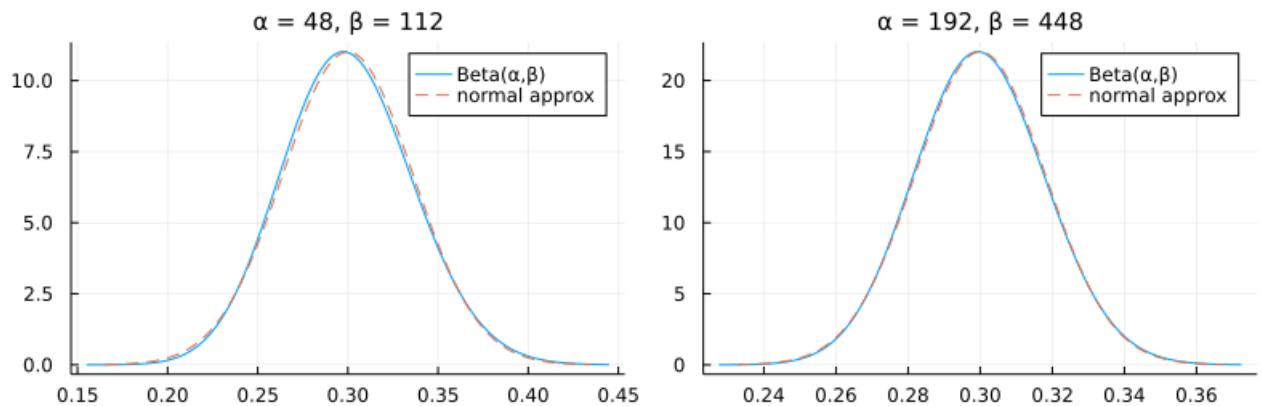
```
In [98]: 1 plot(plot_betanormal(3, 7), plot_betanormal(12, 28); size=(800, 250))
```

Out[98]:



```
In [99]: 1 plot(plot_betanormal(48, 112), plot_betanormal(192, 448); size=(800, 250))
```

Out[99]:



4.10 問題: ベータ分布の正規分布近似の直接的証明

α, β が大きなときにベータ分布 $\text{Beta}(\alpha, \beta)$ が正規分布で近似されることを、その確率密度函数の具体形を使って直接証明せよ。

解答例: $0 < \mu < 1$ と仮定し、

$$\alpha = N\mu, \quad \beta = N(1 - \mu), \quad t = \mu + \frac{x}{\sqrt{N}}$$

において、 N を大きくすると、Stirlingの公式より、

$$\begin{aligned} \frac{1}{B(\alpha, \beta)} &= \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} = \frac{\Gamma(N)}{\Gamma(N\mu)\Gamma(N(1 - \mu))} \\ &\sim \frac{N\mu N(1 - \mu)}{N} \frac{N^N e^{-N}}{(N\mu)^N e^{-N\mu} \sqrt{2\pi N\mu} (N(1 - \mu))^{N(1-\mu)} e^{-N(1-\mu)} \sqrt{2\pi N(1 - \mu)}} \\ &= \frac{N\mu(1 - \mu)}{\mu^{N\mu} \sqrt{2\pi N\mu} (1 - \mu)^{N(1-\mu)} \sqrt{1 - \mu}} \\ &= \frac{1}{\sqrt{2\pi\mu(1 - \mu)}} \frac{\sqrt{N}}{\mu^{N\mu-1} (1 - \mu)^{N(1-\mu)-1}}. \end{aligned}$$

そして

$$\begin{aligned}
& N\mu \log \left(1 + \frac{x}{\sqrt{N}\mu} \right) \\
&= N\mu \left(\frac{x}{\sqrt{N}\mu} - \frac{x^2}{2N\mu^2} + O(N^{-3/2}) \right) \\
&= \sqrt{N}x - \frac{x^2}{2\mu} + O(N^{-1/2}), \\
& N(1-\mu) \log \left(1 - \frac{x}{\sqrt{N}(1-\mu)} \right) \\
&= N(1-\mu) \left(-\frac{x}{\sqrt{N}(1-\mu)} - \frac{x^2}{2N(1-\mu)^2} + O(N^{-3/2}) \right) \\
&= -\sqrt{N}x - \frac{x^2}{2(1-\mu)} + O(N^{-1/2}), \\
& \frac{x^2}{2\mu} + \frac{x^2}{2(1-\mu)} = \frac{x^2}{2\mu(1-\mu)}
\end{aligned}$$

より、以下が成立することがわかる：

$$\begin{aligned}
& t^{\alpha-1}(1-t)^{\beta-1} dt \\
&= \left(\mu + \frac{x}{\sqrt{N}} \right)^{N\mu-1} \left(1 - \mu - \frac{x}{\sqrt{N}} \right)^{N(1-\mu)-1} \frac{dx}{\sqrt{N}} \\
&= \mu^{N\mu-1} (1-\mu)^{N(1-\mu)-1} \left(1 + \frac{x}{\sqrt{N}\mu} \right)^{N\mu-1} \left(1 - \frac{x}{\sqrt{N}(1-\mu)} \right)^{N(1-\mu)-1} \frac{dx}{\sqrt{N}} \\
&= \frac{\mu^{N\mu-1} (1-\mu)^{N(1-\mu)-1}}{\sqrt{N}} \exp \left(-\frac{x^2}{2\mu(1-\mu)} \right) (1 + O(n^{-1/2})) dx.
\end{aligned}$$

以上の結果より、

$$\frac{t^{\alpha-1}(1-t)^{\beta-1} dt}{B(\alpha, \beta)} = \frac{1}{\sqrt{2\pi\mu(1-\mu)}} \exp \left(-\frac{x^2}{2\mu(1-\mu)} \right) (1 + O(n^{-1/2})) dx.$$

この公式はベータ分布が上で導入した x 座標では平均 0 , 分散 $\mu(1-\mu)$ の正規分布で近似されることを意味している。

解答終

注意: $\alpha = N\mu$, $\beta = N(1-\mu)$, $T \sim \text{Beta}(\alpha, \beta)$ のとき、

$$E[T] = \frac{\alpha}{\alpha + \beta} = \mu, \quad \text{var}(T) = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)} = \frac{\mu(1-\mu)}{N+1}$$

なので、 $X = \sqrt{N}(T - \mu)$ とおくと、 $N \rightarrow \infty$ のとき、

$$E[X] = 0, \quad \text{var}(X) = N \text{ var}(T) = \frac{N}{N+1} \mu(1-\mu) \rightarrow \mu(1-\mu).$$

この $\mu(1-\mu)$ が上の解答例の中で登場した正規分布の分散である。

5 確率分布の正規分布近似

以下に挙げるもの以外にも正規分布で近似される確率分布はたくさんある。各自、研究してノートにまとめておくとよいだろう。具体的な数値例をコンピュータで作っておくと、どのようなときにどれだけ近似がうまく行くかについての「土地勘」が得られる。

```
In [100]: 1 mypdf(dist, x) = pdf(dist, x)
2 mypdf(dist::DiscreteUnivariateDistribution, x) = pdf(dist, round(x))
3
4 function plot_normalapprox(dist; title=distname(dist), kwargs...)
5     μ, σ = mean(dist), std(dist)
6     normal = Normal(μ, σ)
7     xlim = (μ - 4.5σ, μ + 4.5σ)
8     plot(x → mypdf(dist, x), xlim...; label="")
9     plot!(normal, xlim...; label="normal approx", ls=:dash)
10    plot!(; title, kwargs...)
11 end
```

Out[100]: plot_normalapprox (generic function with 1 method)

```
In [101]: 1 # 2022-05-13: Distributions.jl で BetaNegativeBinomial(k, α, β)
2 # がまだ実装されていないので自前で必要最小限の実装を行う。
3
4 module O
5
6 using Distributions
7 using SpecialFunctions
8
9 struct BetaNegativeBinomial{T} <: DiscreteUnivariateDistribution
10    k::T
11    α::T
12    β::T
13 end
14
15 function logpdf_betanegbin(x, k, α, β)
16    m = round(x)
17    m < 0 && return -Inf
18    (loggamma(k+m) - loggamma(m+1) - loggamma(k) + logbeta(α+k, β+m) - logbeta(α, β))
19 end
20
21 function Distributions.logpdf(dist::BetaNegativeBinomial, x)
22    (; k, α, β) = dist
23    logpdf_betanegbin(x, k, α, β)
24 end
25
26 function Distributions.mean(dist::BetaNegativeBinomial)
27    (; k, α, β) = dist
28    k*β/(α - 1)
29 end
30
31 function Distributions.var(dist::BetaNegativeBinomial)
32    (; k, α, β) = dist
33    k*(α+k-1)*β*(α+β-1)/((α-1)^2*(α-2))
34 end
35
36 end
```

Out[101]: Main.O

5.1 二項分布の正規分布近似

$0 < p < 1$ ありかつ,

$$(X_1, \dots, X_n) \sim \text{Bernoulli}(p)^n = \text{Binomial}(1, p)^n$$

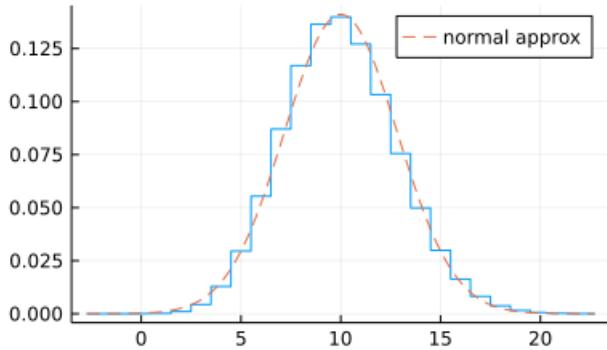
であるとき,

$$K_n = X_1 + \dots + X_n \sim \text{Binomial}(n, p)$$

であり, 中心極限定理によって独立同分布な確率変数達の和の分布が正規分布で近似されることより, n が大きなとき, K_n の分布は正規分布で近似される。

```
In [102]: 1 plot_normalapprox(Binomial(50, 0.2))
```

Out[102]: Binomial(n=50, p=0.2)



5.2 負の二項分布の正規分布近似

$0 < p < 1$ ありかつ,

$$(Y_1, \dots, Y_k) \sim \text{Geometric}(p) = \text{NegativeBinomial}(1, p)^k$$

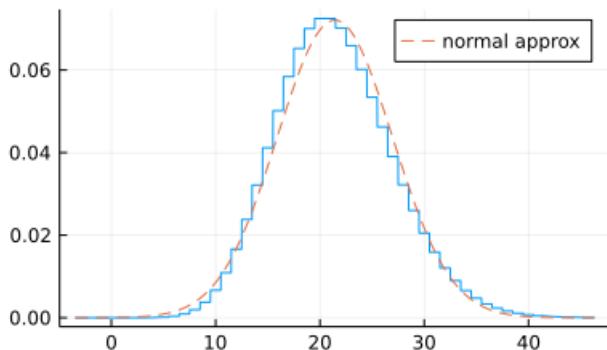
であるとき,

$$N_k = Y_1 + \dots + Y_k \sim \text{NegativeBinomial}(k, p)$$

であり、中心極限定理によって独立同分布な確率変数達の和の分布が正規分布で近似されることより、 k が大きなとき、 N_k の分布は正規分布で近似される。

```
In [103]: 1 plot_normalapprox(NegativeBinomial(50, 0.7))
```

Out[103]: NegativeBinomial(r=50.0, p=0.7)



5.3 Poisson分布の正規分布近似

$\lambda > 0$ ありかつ,

$$(X_1, \dots, X_n) \sim \text{Poisson}(\lambda/n)^n$$

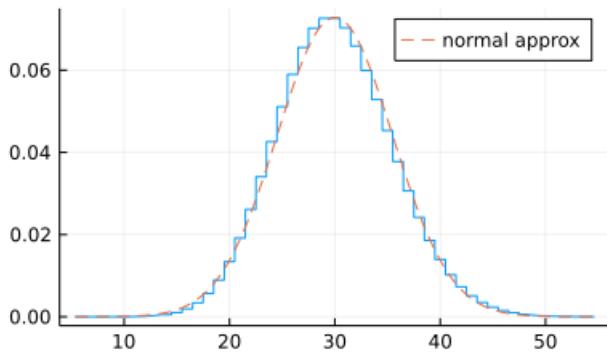
であるとき,

$$K = X_1 + \dots + X_n \sim \text{Poisson}(\lambda)$$

であり、中心極限定理によって独立同分布な確率変数達の和の分布が正規分布で近似されることより、 λ が大きなとき、 K の分布は正規分布で近似される。

```
In [104]: 1 plot_normalapprox(Poisson(30))
```

Out[104]: Poisson($\lambda=30.0$)



5.4 ガンマ分布の正規分布近似

$\alpha > 0, \theta > 0$ ありかつ,

$$(Y_1, \dots, Y_n) \sim \text{Gamma}(\alpha/n, \theta)$$

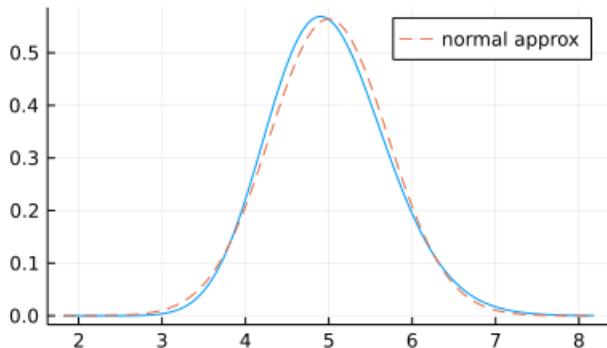
であるとき,

$$T = Y_1 + \dots + Y_n \sim \text{Gamma}(\alpha, \theta)$$

であり, 中心極限定理によって独立同分布な確率変数達の和の分布が正規分布で近似されることより, α が大きなとき, T の分布は正規分布で近似される.

```
In [105]: 1 plot_normalapprox(Gamma(50, 0.1))
```

Out[105]: Gamma($\alpha=50.0, \theta=0.1$)



5.5 χ^2 分布の正規分布近似

$v > 0$ ありかつ,

$$(Y_1, \dots, Y_n) \sim \text{Chisq}(v/n) = \text{Gamma}(v/(2n), 2)$$

であるとき,

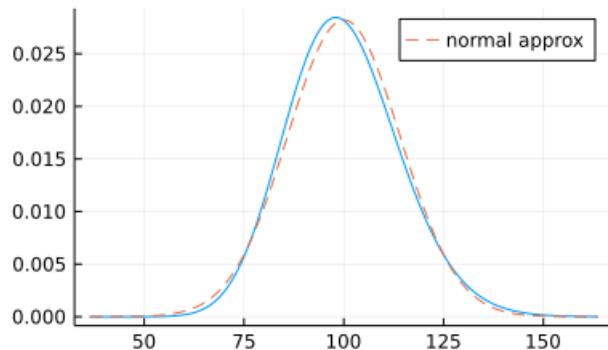
$$Y = Y_1 + \dots + Y_n \sim \text{Chisq}(v) = \text{Gamma}(v/2, 2)$$

であり, 中心極限定理によって独立同分布な確率変数達の和の分布が正規分布で近似されることより, v が大きなとき, Y の分布は正規分布で近似される.

```
In [106]: 1 plot_normalapprox(Chisq(100))
```

Out[106]:

Chisq($\nu=100.0$)



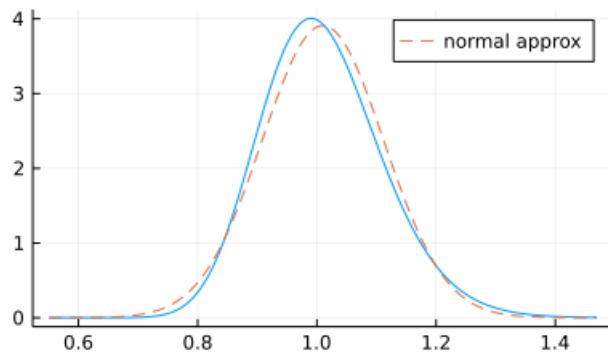
5.6 逆ガンマ分布の正規分布近似

$\alpha > 0$ が大きなとき, 逆ガンマ分布 $\text{InverseGamma}(\alpha, \theta)$ は正規分布で近似される.

```
In [107]: 1 plot_normalapprox(InverseGamma(100, 100); title="InverseGamma(100, 100)")
```

Out[107]:

InverseGamma(100, 100)



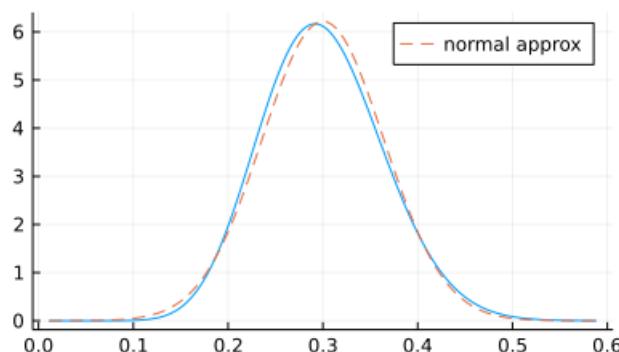
5.7 ベータ分布の正規分布近似

すでに上で示したように, $\alpha, \beta > 0$ が大きなとき, ベータ分布 $\text{Beta}(\alpha, \beta)$ は正規分布で近似される.

```
In [108]: 1 plot_normalapprox(Beta(15, 35))
```

Out[108]:

Beta($\alpha=15.0, \beta=35.0$)

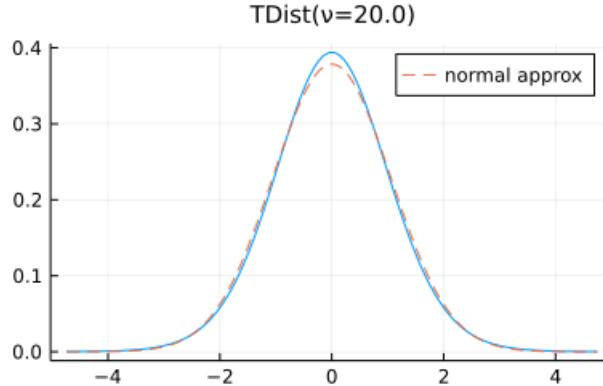


5.8 t 分布の正規分布近似

「Gauss積分, ガンマ函数, ベータ函数」のノートすでに示したように, $\nu > 0$ が大きなとき, t 分布 $T\text{Dist}(\nu)$ は標準正規分布で近似される.

In [109]: 1 plot_normalapprox(TDist(20))

Out[109]:

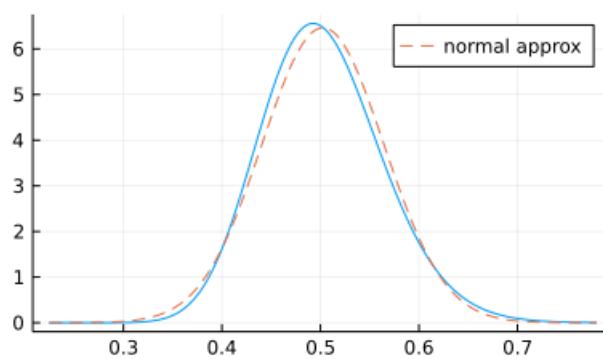


5.9 ベータプライム分布の正規分布近似

$\alpha, \beta > 0$ が大きなとき, ベータプライム分布 $\text{BetaPrime}(\alpha, \beta)$ は正規分布で近似される. (証明に興味がある人は自分で証明してみよ.)

In [110]: 1 plot_normalapprox(BetaPrime(100, 200))

Out[110]:

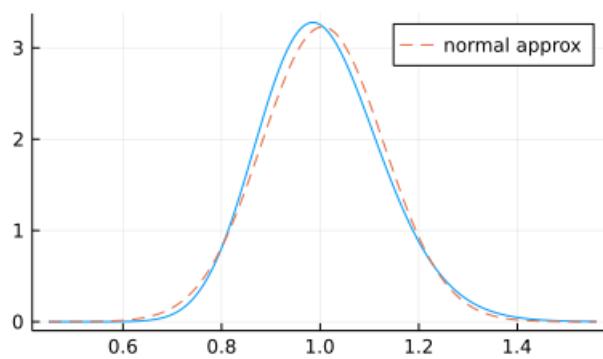


5.10 F 分布の正規分布近似

$\nu_1, \nu_2 > 0$ が大きなとき, F 分布 $F\text{Dist}(\nu_1, \nu_2) = (\nu_2/\nu_1) \text{BetaPrime}(\nu_1/2, \nu_2/2)$ は正規分布で近似される.

In [111]: 1 plot_normalapprox(FDist(200, 400))

Out[111]:

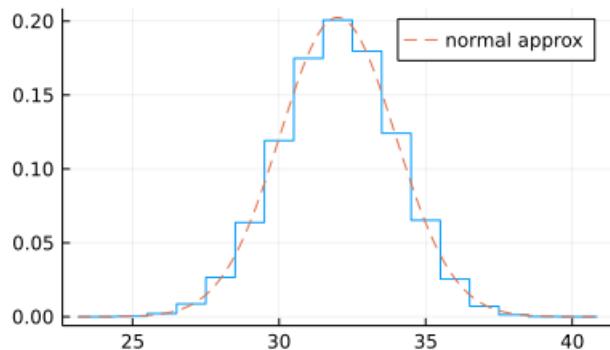


5.11 超幾何分布の正規分布近似

s, f, n が大きなとき, 超幾何分布 $\text{Hypergeometric}(s, f, n)$ は正規分布で近似される.

```
In [112]: 1 plot_normalapprox(Hypergeometric(40, 60, 80))
```

Out[112]: Hypergeometric(ns=40, nf=60, n=80)

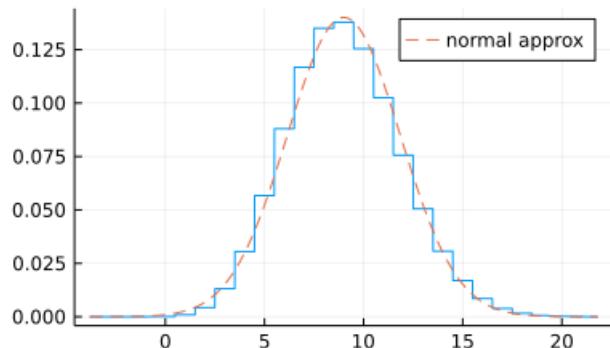


5.12 ベータ二項分布の正規分布近似

n, α, β が大きなとき, ベータ二項分布 BetaBinomial(n, α, β) は正規分布で近似される.

```
In [113]: 1 plot_normalapprox(BetaBinomial(30, 30, 70))
```

Out[113]: BetaBinomial(n=30, α=30.0, β=70.0)

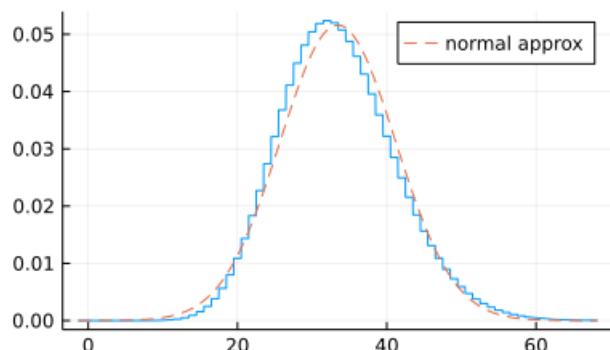


5.13 ベータ負の二項分布の正規分布近似

k, α, β が大きなとき, ベータ二項分布 BetaNegativeBinomial(k, α, β) は正規分布で近似される.

```
In [114]: 1 plot_normalapprox(0.BetaNegativeBinomial(100, 300, 100);  
2     title="BetaNegativeBinomial(100, 200, 400)")
```

Out[114]: BetaNegativeBinomial(100, 200, 400)



```
In [ ]: 1
```

