例：ベータ函数と二項分布の関係とその応用

- 黒木玄
- 2022-05-27～2022-05-27

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)

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

In [1]:

```
ENV["LINES"], ENV["COLUMNS"] = 100, 100
using BenchmarkTools
using Distributions
using LinearAlgebra
using Printf
using QuadGK
using Random
Random.seed!(4649373)
using Roots
using SpecialFunctions
using StaticArrays
using StatsBase
using StatsFuns
using StatsPlots
default(fmt = :png, titlefontsize = 10, size = (400, 250))
using SymPy
```

In [2]:
```
# Override the Base.show definition of SymPy.jl:
# https://github.com/JuliaPy/SymPy.jl/blob/29c5bfd1d10ac53014fa7fef468bc8deccadc2fc/src/types.

@eval SymPy function Base.show(io::IO, ::MIME"text/latex", x::SymbolicObject)
    print(io, as_markdown("\\displaystyle " * sympy.latex(x, mode="plain", fold_short_frac=fal
end
@eval SymPy function Base.show(io::IO, ::MIME"text/latex", x::AbstractArray{Sym})
    function toeqnarray(x::Vector{Sym})
        a = join(["\\displaystyle " * sympy.latex(x[i]) for i in 1:length(x)], "\\\\")
        """\\left[ \\begin{array}{r}$a\\end{array} \\right]"""
    end
    function toeqnarray(x::AbstractArray{Sym,2})
        sz = size(x)
        a = join([join("\\displaystyle " .* map(sympy.latex, x[i,:]), "&") for i in 1:sz[1]],
        "\\left[ \\begin{array}{" * repeat("r",sz[2]) * "}" * a * "\\end{array}\\right]"
    end
    print(io, as_markdown(toeqnarray(x)))
end
```

In [3]:
```
safemul(x, y) = x == 0 ? x : x*y
safediv(x, y) = x == 0 ? x : x/y

x ≲ y = x < y || x ≈ y

mypdf(dist, x) = pdf(dist, x)
mypdf(dist::DiscreteUnivariateDistribution, x) = pdf(dist, round(Int, x))

distname(dist::Distribution) = replace(string(dist), r"{.*}" => "")
myskewness(dist) = skewness(dist)
mykurtosis(dist) = kurtosis(dist)
function standardized_moment(dist::ContinuousUnivariateDistribution, m)
    μ, σ = mean(dist), std(dist)
    quadgk(x -> (x - μ)^m * pdf(dist, x), extrema(dist)...)[1] / σ^m
end
myskewness(dist::MixtureModel{Univariate, Continuous}) = standardized_moment(dist, 3)
mykurtosis(dist::MixtureModel{Univariate, Continuous}) = standardized_moment(dist, 4) - 3
```

Out[3]:
```
mykurtosis (generic function with 2 methods)
```

1 ベータ分布と二項分布の累積分布函数の関係

このノートではすでに紹介済み(「確率分布達の解釈」のノートの「ベータ分布の一様乱数生成の繰り返しによる解釈」の節を参照)のベータ分布と二項分布の累積分布函数の関係について再度説明することにする. 再度説明する理由は複数ある:

- 二項分布モデルでの信頼区間の計算で使われるから.
- 異なる確率分布のあいだの関係の典型的な例になっているから.
- コンピュータによる確率分布の効率的な計算とも関係しているから.
- その他にも色々なことと関係しているから.

以下で証明したいこと: $k = 1, 2, \ldots, n$ のとき,

$$\begin{aligned}&(\text{二項分布 } \mathrm{Binomial}(n, p) \text{ で成功回数が } k \text{ 以上になる確率})\\ &= (\text{ベータ分布 } \mathrm{Beta}(k, n-k+1) \text{ において } p \text{ 以下になる確率})\end{aligned}$$

数式で書くと,

$$\sum_{i \geq k} \binom{n}{i} p^i (1-p)^{n-i} = \frac{\int_0^p t^{k-1}(1-t)^{n-i}\,dt}{B(k, n-k+1)}.$$

これは, 補事象を取り, 「k 未満になる」という条件を「k 以下になる」に置き換えるために, k を $k+1$ で置き換えると次と同値である: $k = 0, 1, \ldots, n-1$ のとき,

$$\begin{aligned}&(\text{二項分布 } \mathrm{Binomial}(n, p) \text{ で成功回数が } k \text{ 以下になる確率})\\ &= (\text{ベータ分布 } \mathrm{Beta}(k+1, n-k) \text{ において } p \text{ 以上になる確率})\end{aligned}$$

数式で書くと,

$$\sum_{i \leq k} \binom{n}{i} p^i (1-p)^{n-i} = \frac{\int_p^1 t^{k}(1-t)^{n-i-1}\,dt}{B(k+1, n-k)}.$$

1.1 ベータ分布と二項分布の累積分布函数の関係の証明

$k = 1, 2, \ldots, n$ と仮定し, 次が成立することを示そう:

$$\begin{aligned}&(\text{二項分布 } \mathrm{Binomial}(n, p) \text{ で成功回数が } k \text{ 以上になる確率})\\ &= (\text{ベータ分布 } \mathrm{Beta}(k, n-k+1) \text{ において } p \text{ 以下になる確率})\end{aligned}$$

証明: $p = t$ のときの左辺(二項分布 $\mathrm{Binomial}(n, t)$ において k 以上になる確率)を $F(t)$ と書くと,

$$F(t) = \sum_{i \geq k} \binom{n}{i} t^i (1-t)^{n-i} = \sum_{i \geq k} \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}.$$

このとき, $k > 0$ より, $F(0) = 0$ であり,

$$\begin{aligned}F'(t) &= \sum_{i \geq k} \frac{n!}{i!(n-i)!} i t^{i-1} (1-t)^{n-i} + \sum_{i \geq k} \frac{n!}{i!(n-i)!} (n-i) t^i (1-t)^{n-i-1} \\ &= \sum_{i \geq k} \frac{n!}{(i-1)!(n-i)!} t^{i-1} (1-t)^{n-i} + \sum_{i \geq k} \frac{n!}{i!(n-i-1)!} t^i (1-t)^{n-i-1} \\ &= \sum_{i \geq k} \frac{n!}{(i-1)!(n-i)!} t^{i-1} (1-t)^{n-i} + \sum_{i \geq k+1} \frac{n!}{(i-1)!(n-i)!} t^{i-1} (1-t)^{n-i} \\ &= \frac{n!}{(k-1)!(n-k)!} t^{k-1} (1-t)^{n-k}.\end{aligned}$$

3つめの等号で2つめの和の i を $i-1$ に置き換えた. さらに, $\Gamma(a+1) = a!$ と $\Gamma(a)\Gamma(b) = B(a,b)\Gamma(a+b)$ を使うと,

$$\frac{n!}{(k-1)!(n-k)!} = \frac{\Gamma(n+1)}{\Gamma(k)\Gamma(n-k+1)} = \frac{1}{B(k, n-k+1)}$$

なので,

$$F'(t) = \frac{t^{k-1}(1-t)^{n-k}}{B(k, n-k+1)}.$$

これと $F(0) = 0$ を合わせると,

$$F(p) = \frac{\int_0^p t^{k-1}(1-t)^{n-k}\, dt}{B(k, n-k+1)}.$$

これは, ベータ分布 $\mathrm{Beta}(k, n-k+1)$ において p 以下になる確率である.

証明終

1.2 基本特殊函数ライブラリを使うと効率的に計算できる

以上によって, 次の公式が示された:

$$\sum_{i \geq k} \binom{n}{i} p^i (1-p)^{n-i} = \frac{\int_0^p t^{k-1}(1-t)^{n-k}\, dt}{B(k, n-k+1)}.$$

次は誤解である:

- 左辺の和よりも右辺の積分の方が計算が大変である.

実際には, 右辺の方が効率的に計算できる!

- 左辺の和の計算は $n \geq 10^5$ のような場合にはそれなりに大変である.
- 左辺の計算は一度だけ計算すればよいのであれば大変ではない.
- しかし, 繰り返し計算する場合には時間がかかってしまう場合がある.
- 左辺は正則化不完全ベータ函数 $I_p(k, n-k+1)$ であり, 必要な計算量は小さい.
- 実際, 正則化不完全ベータ函数は基本特殊函数ライブラリに含まれている.

In [4]:
```
n, k, p = 10^6, 123456, 0.123
```

Out[4]:
```
(1000000, 123456, 0.123)
```

この場合には, 左辺の和は単純に計算しようとするとそもそも計算できない.

```
sum(binomial(n, i) * p^i * (1-p)^(n-i) for i in k:n)
```

```
OverflowError: binomial(1000000, 123456) overflows
```

次のように工夫すれば計算できる.

In [5]:
```
F_logsumexp(n, k, p) = exp(logsumexp(
        logabsbinomial(n, i)[1] + i*log(p) + (n-i)*log(1-p) for i in k:n))
F_logsumexp(n, k, p)
```

Out[5]:
```
0.0827943265753984
```

正則化不完全ベータ函数という名の特殊函数による計算は以下の通り.

In [6]:
```
F_beta_inc(n, k, p) = beta_inc(k, n-k+1, p)[1]
F_beta_inc(n, k, p)
```

Out[6]:
```
0.08279432657965562
```

Julia言語のDistributions.jlによる計算結果は上とぴったり同じになる.

In [7]:
```
F_distributionsjl(n, k, p) = ccdf(Binomial(n, p), k-1)
F_distributionsjl(n, k, p)
```

Out[7]:
```
0.08279432657965562
```

計算にかかる時間の計測:

In [8]:
```
@btime F_logsumexp(n, k, p)
@btime F_beta_inc(n, k, p);
```

```
  80.615 ms (1 allocation: 16 bytes)
  409.548 ns (5 allocations: 976 bytes)
```

この場合には, 正則化不完全ベータ函数を使った方が計算が約200倍以上速くなる!

1.3 ベータ分布と二項分布の累積分布函数の関係の別証明

0 から 1 のあいだの値になる一様乱数(コンピュータでの `rand()` と同じだと思ってよい)を n 個独立に生成した結果を $T_1, T_2, \ldots, T_n$ と書き, その中で k 番目に小さな値を $T_{(k)}$ と書く. T_i や $T_{(i)}$ は確率変数とみなされる.

$$\begin{aligned}(T_{(k)} \leq p \text{ となる確率}) &= (T_1, T_2, \ldots, T_n \text{ の中に } p \text{ 以下のものが } k \text{ 個以上ある確率}) \\ &= (\text{二項分布 } \mathrm{Binomial}(n, p) \text{ で成功回数が } k \text{ 以上になる確率})\end{aligned}$$

が成立している. 一方, $T_{(k)}$ が $[t, t+dt]$ に含まれる確率は dt に関する高次の微小量を無視する近似で

$$\frac{n!}{(k-1)!1!(n-k)!} t^{k-1}\, dt\, (1-t)^{n-k} = \frac{t^{k-1}(1-t)^{(n-k+1)-1}\, dt}{B(k, n-k+1)}$$

になる. なぜならば, $T_{(k)}$ が $[t, t+dt]$ に含まれることは, 大雑把には, $T_1, T_2, \ldots, T_n$ の中に t 未満のものが $k-1$ 個存在し, 1 個が $[t, t+dt]$ に含まれ, それ以外の $n-k$ 個が t より大きくなることと同値である. このことから, 左辺が得られる. $n!/((k-1)!1!(n-k)!)$ は $T_1, T_2, \ldots, T_n$ の $k-1$ 個, 1 個, $n-k$ 個へのグループ分けの仕方の個数になっている.

これで, $T_{(k)}$ は確率密度函数

$$p(t) = \frac{t^{k-1}(1-t)^{(n-k+1)-1}}{B(k, n-k+1)}$$

を持つことがわかった. すなわち, $T_{(k)} \sim \mathrm{Beta}(k, n-k+1)$ である. ゆえに,

$$(T_{(k)} \leq p \text{ となる確率}) = (\text{ベータ分布 } \mathrm{Beta}(k, n-k+1) \text{ において } p \text{ 以下になる確率})$$

以上を合わせると,

$$\begin{aligned}&(\text{二項分布 } \mathrm{Binomial}(n, p) \text{ で成功回数が } k \text{ 以上になる確率}) \\ &= (\text{ベータ分布 } \mathrm{Beta}(k, n-k+1) \text{ において } p \text{ 以下になる確率})\end{aligned}$$

証明終

注意: 以上の証明法の利点は以下の通り:

- 二項分布の累積分布函数をパラメータ p で微分するという面倒な計算を避けられる.
- ベータ函数の逆数 $1/B(a,b) = \Gamma(a+b)/\Gamma(a)\Gamma(b)$ が $\Gamma(a) = (a-1)!$ であり, 階乗とぴったり一致しないせいで, 二項係数に一致しない理由もわかる. ベータ函数の逆数は二項係数ではなく, n 個の $k-1$ 個, 1 個, $n-k$ 個へのグループ分けの仕方の個数(多項係数の特別な場合)になっている.
- 上と同様の方法ならば, 複数の $T_{(k)}$ 達の同時確率分布が本質的にDirichlet分布になっていることも容易に確認できる.

1.4 累積分布函数と分位点函数について

分布 D に従う確率変数 X について x の函数

$$\mathrm{cdf}(D,x) = F_D(x) = P(X \leq x)$$

を確率変数 X が従う分布 D の **累積分布函数** (cumulative distribution function, cdf)と呼び, その逆函数(もしくは逆函数の適切な類似物)

$$\mathrm{quantile}(D,p) = Q_D(p) = F_X^{-1}(p)$$

を確率変数 X が従う分布 D の **分位点函数** (quantile function)と呼ぶのであった.

これらの函数はコンピュータにおいて確率分布を扱うライブラリの中で定義されていることが多い.

例えば, Julia言語のDistributions.jlパッケージでは

```
ccdf(Binomial(10^6, 0.123), 123456-1)
cdf(Beta(123456, 10^6-123456+1), 0.123)
quantile(Beta(123456, 10^6-123456+1), 0.0828)
```

のように書く. ただし, $\mathrm{ccdf}(D,x) = 1 - \mathrm{cdf}(D,x)$ であり, `ccdf(Binomial(10^6, 0.123), 123456-1)` は二項分布 $\mathrm{Binomial}(10^6, 0.123)$ で 123456 以上になる確率に一致する.

In [9]:
```
ccdf(Binomial(10^6, 0.123), 123456-1) # 下と一致
```

Out[9]:
```
0.08279432657965562
```

In [10]:
```
cdf(Beta(123456, 10^6-123456+1), 0.123) # 上と一致
```

Out[10]:
```
0.08279432657965562
```

In [11]:
```
quantile(Beta(123456, 10^6-123456+1), 0.0828)
```

Out[11]:
```
0.12300001220665846
```

WolframAlphaでは

```
1 - CDF[BinomialDistribution[10^6, 0.123], 123455]
CDF[BetaDistribution[123456, 10^6 -  123456 + 1], 0.123]
Quantile[BetaDistribution[123456, 10^6 -  123456 + 1], 0.0828]
```

のように入力すれば計算してくれる. WolframAlphaでは表記の揺れ(大文字小文字, 括弧の違い)があっても適切に解釈して計算してくれる:

- 1 - cdf(BinomialDistribution(10^6, 0.123), 123455) (https://www.wolframalpha.com/input?i=1+-+cdf%28BinomialDistribution%2810%5E6%2C+0.123%29%2C+123455%29&lang=ja) → 0.0827943 (下と一致)
- cdf(BetaDistribution(123456, 10^6 - 123456 + 1), 0.123) (https://www.wolframalpha.com/input?i=cdf%28BetaDistribution%28123456%2C+10%5E6+-+123456+%2B+1%29%2C+0.123%29) → 0.0827943 (上と一致)
- quantile(BetaDistribution(123456, 10^6 - 123456 + 1), 0.0828) (https://www.wolframalpha.com/input?i=quantile%28BetaDistribution%28123456%2C+10%5E6+-+123456+%2B+1%29%2C+0.0828%29) → 0.123

これらの函数ではすでに特殊函数を使った効率的な実装がされており, 使用可能ならばこれを使うとよい.

1.5 パラメータに関する区間推定での利用の仕方

コンピュータにおける確率分布を扱うライブラリ(パッケージ)を利用すれば, 確率分布の累積分布函数やその逆函数である分位点函数を効率的に計算できる.

しかし, 統計学においては, パラメータ θ を持つ分布 $D(\theta)$ とその分布で扱えるデータの数値 x から定まるパラメータ θ の函数

$$\theta \mapsto (\text{分布 } D(\theta) \text{ における数値 } x \text{ から決まる確率})$$

の逆函数が必要になることがある.

例えば, データ「n 回中 k 回成功」というデータが得られたとき, それを二項分布モデル $\mathrm{Binomial}(n, p)$ で扱うときに, 成功確率 p の函数

$$p \mapsto (\text{分布 } \mathrm{Binomial}(n, p) \text{ で成功回数が } k \text{ 以上になる確率})$$

の逆函数が成功確率 p の区間推定(Clopper-Pearsonの信頼区間)のために必要になる.

データの数値からモデルのパラメータを推定することは統計学では最も普通に行われることなので, このような場合は非常に多い.

一般には上のようにして定義されたパラメータの逆函数の計算は面倒だが, 二項分布モデルの場合にはベータ分布との関係を用いて極めて効率的にそれを計算できる!

なぜならば, すでに示したように

$$(\text{分布 } \mathrm{Binomial}(n, p) \text{ で成功回数が } k \text{ 以上になる確率}) = \mathrm{cdf}(\mathrm{Beta}(k, n-k+1), p)$$

が成立しているからである. 左辺でパラメータであった p が右辺では累積分布函数の引数になっている. ゆえに,

$$p \mapsto a = (\text{分布 } \mathrm{Binomial}(n, p) \text{ で成功回数が } k \text{ 以上になる確率})$$

の逆函数は

$$a \mapsto p = \mathrm{quantile}(\mathrm{Beta}(k, n-k+1), a)$$

になる. これは確率分布を扱うライブラリで効率的に計算可能である!

1.6 まとめ

次が成立している:

$$\sum_{i \geq k} \binom{n}{i} p^i (1-p)^{n-i} = \frac{\int_0^p t^{k-1}(1-t)^{n-i}\,dt}{B(k, n-k+1)},$$
$$\sum_{i \leq k} \binom{n}{i} p^i (1-p)^{n-i} = \frac{\int_p^1 t^{k}(1-t)^{n-i-1}\,dt}{B(k+1, n-k)}.$$

すなわち, 以下が成立している:

$$(\text{分布 } \mathrm{Binomial}(n, p) \text{ で成功回数が } k \text{ 以上になる確率}) = \mathrm{cdf}(\mathrm{Beta}(k, n-k+1), p),$$
$$(\text{分布 } \mathrm{Binomial}(n, p) \text{ で成功回数が } k \text{ 以下になる確率}) = \mathrm{ccdf}(\mathrm{Beta}(k+1, n-k), p).$$

ここで $a = \mathrm{ccdf}(D, x) = 1 - \mathrm{cdf}(D, x)$ であり, これの逆函数は $x = \mathrm{quantile}(D, 1-a)$ になる.

したがって,

$$p \mapsto a = (\text{分布 } \mathrm{Binomial}(n, p) \text{ で成功回数が } k \text{ 以上になる確率})$$

の逆函数は

$$a \mapsto p = \mathrm{quantile}(\mathrm{Beta}(k, n-k+1), a)$$

になり,

$$p \mapsto a = (\text{分布 } \mathrm{Binomial}(n, p) \text{ で成功回数が } k \text{ 以下になる確率})$$

の逆函数は

$$a \mapsto p = \mathrm{quantile}(\mathrm{Beta}(k+1, n-k), 1-a)$$

になる. これらの逆函数はコンピュータにおける確率分布を扱うライブラリ(パッケージ)を使えば効率的にかつ楽に計算できる.

次節で使う図の準備

一般に図の準備は相当に面倒である.

In [12]:
```
n, k = 100, 30
kmin, kmax = 0, 65

p = 0.27
var"p = 0.27 は n = 100, k = 30 にそこそこ適合している" = plot(; size=(500, 300))
plot!(k -> mypdf(Binomial(n, p), k), kmin-0.5, kmax+0.5;
    label="", c=1)
plot!(k -> mypdf(Binomial(n, p), k), k-0.5, kmax+0.5;
    fillrange=0, c=1, fc=:blue, fa=0.5, label="")
annotate!(k+6, 0.03, text("probability of K ≥ $k", :blue, :left, 10))
title!("Binomial(n=$n, p=$p)")
vline!([k]; label="k=$k", c=:black, ls=:dot)
vline!([n*p]; label="np=$(n*p)", c=:red, ls=:dot)
plot!(; xguide="K", ylim=(-0.007, 0.105));
```

In [13]:
```
n, k = 100, 30
kmin, kmax = 0, 65

p = 0.19
var"p = 0.19 は n = 100, k = 30 にあまりにも適合しない" = plot(; size=(500, 300))
plot!(k -> mypdf(Binomial(n, p), k), kmin-0.5, kmax+0.5;
    label="", c=1)
plot!(k -> mypdf(Binomial(n, p), k), k-0.5, kmax+0.5;
    fillrange=0, c=1, fc=:blue, fa=0.5, label="")
annotate!(k+1, 0.01, text("probability of K ≥ $k", :blue, :left, 10))
title!("Binomial(n=$n, p=$p)")
vline!([k]; label="k=$k", c=:black, ls=:dot)
vline!([n*p]; label="np=$(n*p)", c=:red, ls=:dot)
plot!(; xguide="K", ylim=(-0.007, 0.105));
```

In [14]:
```
p = 0.42
var"p = 0.42 は n = 100, k = 30 にあまりにも適合しない" = plot(; size=(500, 300))
plot!(k -> mypdf(Binomial(n, p), k), kmin-0.5, kmax+0.5;
    label="", c=2)
plot!(k -> mypdf(Binomial(n, p), k), kmin-0.5, k+0.5;
    fillrange=0, c=2, fc=:red, fa=0.5, label="")
annotate!(k-1, 0.01, text("probability of K ≤ $k", :red, :right, 10))
title!("Binomial(n=$n, p=$p)")
vline!([k]; label="k=$k", c=:black, ls=:dot)
vline!([n*p]; label="np=$(n*p)", c=:red, ls=:dot)
plot!(; xguide="K", ylim=(-0.007, 0.105));
```

In [15]:
```
α = 0.05
p_L = quantile(Beta(k, n-k+1), α/2)
var"下側では p = p_L がぎりぎり" = plot(; size=(500, 300))
plot!(k -> mypdf(Binomial(n, p_L), k), kmin-0.5, kmax+0.5;
    label="", c=1)
plot!(k -> mypdf(Binomial(n, p_L), k), k-0.5, kmax+0.5;
    fillrange=0, c=1, fc=:blue, fa=0.5, label="")
annotate!(k+2, 0.01, text("$(100α/2)%", :blue, :left, 10))
title!("Binomial(n=$n, p=p_L),  p_L=$(round(p_L; digits=4))")
vline!([k]; label="k=$k", c=:black, ls=:dot)
vline!([n*p_L]; label="n p_L=$(round(n*p_L; digits=2))", c=:red, ls=:dot)
plot!(; xguide="K", ylim=(-0.007, 0.105));
```

In [16]:
```
p_U = quantile(Beta(k+1, n-k), 1 - 0.025)
var"上側では p = p_U がぎりぎり" = plot(; size=(500, 300))
plot!(k -> mypdf(Binomial(n, p_U), k), kmin-0.5, kmax+0.5;
    label="", c=2)
plot!(k -> mypdf(Binomial(n, p_U), k), kmin-0.5, k+0.5;
    fillrange=0, c=2, fc=:red, fa=0.5, label="")
annotate!(k-2, 0.01, text("$(100α/2)%", :red, :right, 10))
title!("Binomial(n=$n, p=p_U),  p_U=$(round(p_U; digits=4))")
vline!([k]; label="k=$k", c=:black, ls=:dot)
vline!([n*p_U]; label="n p_L=$(round(n*p_U; digits=2))", c=:red, ls=:dot)
plot!(; xguide="K", ylim=(-0.007, 0.105));
```

In [17]:
```
var"n × 95%信頼区間" = plot(; size=(500, 300))

plot!(k -> mypdf(Binomial(n, p_L), k), kmin-0.5, kmax+0.5;
    label="", c=1)
plot!(k -> mypdf(Binomial(n, p_L), k), k-0.5, kmax+0.5;
    fillrange=0, c=1, fc=:blue, fa=0.5, label="")
annotate!(k+2, 0.01, text("$(100α/2)%", :blue, :left, 10))

plot!(k -> mypdf(Binomial(n, p_U), k), kmin-0.5, kmax+0.5;
    label="", c=2)
plot!(k -> mypdf(Binomial(n, p_U), k), kmin-0.5, k+0.5;
    fillrange=0, c=2, fc=:red, fa=0.5, label="")
annotate!(k-2, 0.01, text("$(100α/2)%", :red, :right, 10))

vline!([k]; label="k=$k", c=:black, ls=:dot)
vline!([n*p_L]; label="n p_L", c=:red, ls=:dot)
vline!([n*p_U]; label="n p_U", c=:red, ls=:dot)
plot!([n*p_L, n*p_U], fill(-0.003, 2); label="", c=:red, lw=5)
title!("[n p_L, n p_U] = [$(round(n*p_L; digits=2)), $(round(n*p_U; digits=2))]")
plot!(; xguide="K", ylim=(-0.007, 0.105));
```

2 二項分布モデルでのClopper-Pearsonの信頼区間

信頼区間については後でもっと詳しく扱う.

以下では二項分布モデルにおけるClopper-Pearsonの信頼区間について必要最小限の事柄を説明する.

2.1 Clopper-Pearsonの信頼区間の定義

「n 回中 k 回成功」の型のデータが得られたとする.

このデータの数値と二項分布モデルを比較して, データの数値にあまりにも適合しない成功確率パラメータ値 p を排除して得られる区間を成功確率パラメータ p の **信頼区間** と呼ぶ(もしくは **比率** p の信頼区間と呼んだりする).

より正確には, **信頼度** (もしくは信頼係数) $1-\alpha$ $(0<\alpha<1$ を **有意水準** と呼ぶ)という閾値を決めて, その閾値を超えてデータに適合しないようなパラメータ値を除いてできる区間を信頼度 $1-\alpha$ の信頼区間と呼ぶ. (データへのモデルの適合度の定義の仕方は別に決める. その決め方は複数あり, 以下ではClopper-Pearsonの信頼区間の場合について説明する.)

$\alpha=5\%$ のときには, 単に 95% 信頼区間と呼ぶことが多い.

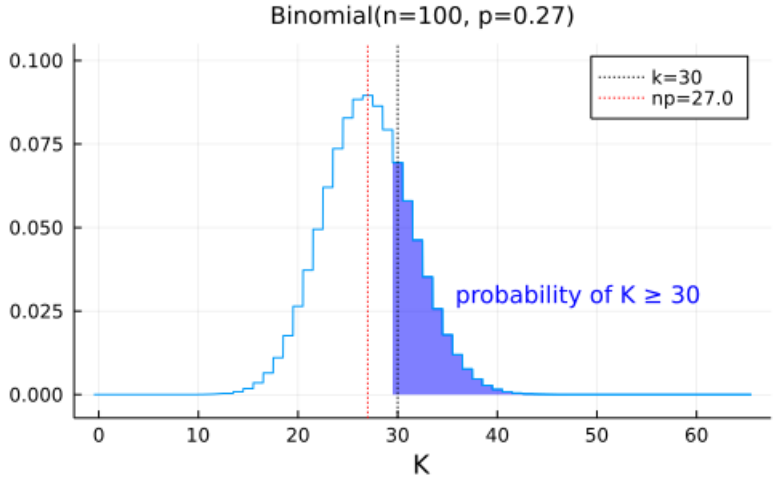
二項分布モデルで伝統的によく使われているClopper-Pearsonの信頼区間における「あまりにも適合しない」の定義は以下の通り:

- 二項分布モデル $\mathrm{Binomial}(n,p)$ において成功回数が k 回以上になる確率は $\alpha/2$ 未満である.
- 二項分布モデル $\mathrm{Binomial}(n,p)$ において成功回数が k 回以下になる確率は $\alpha/2$ 未満である.

このどちらかの条件を満たす p 達を除いてできる区間が信頼係数 $1-\alpha$ の **Clopper-Pearsonの信頼区間** と呼ばれる.

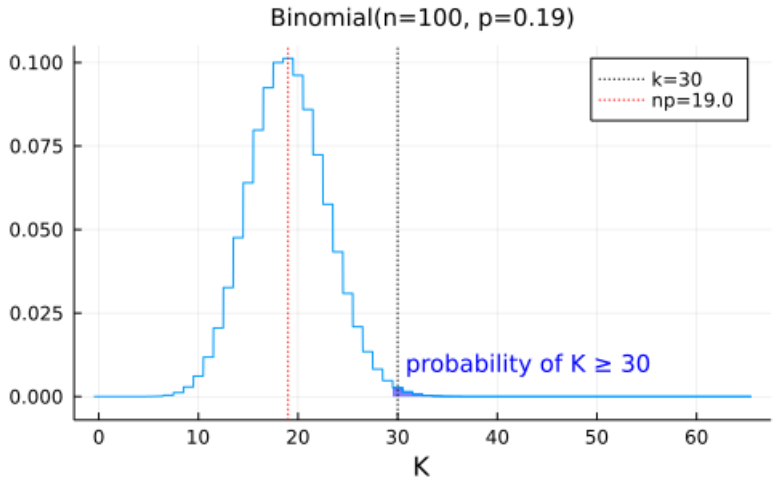
In [18]:
```
var"p = 0.27 は n = 100, k = 30 にそこそこ適合している"
```

Out[18]:



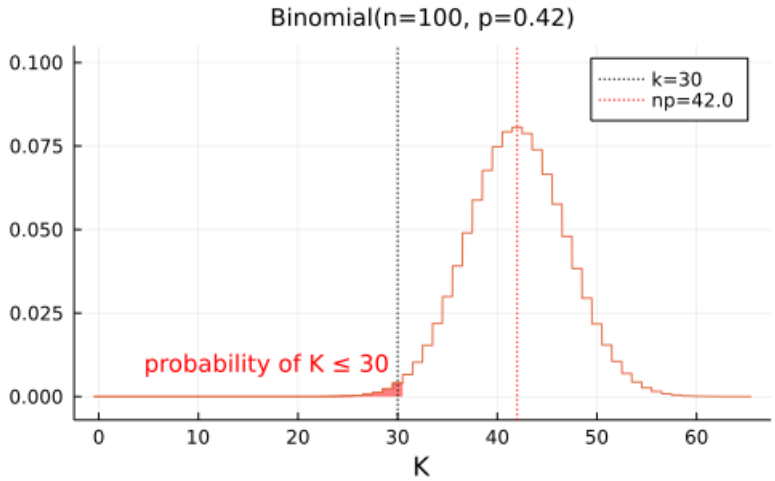
In [19]:
```
1 var"p = 0.19 は n = 100, k = 30 にあまりにも適合しない"
```

Out[19]:



In [20]:
```
1 var"p = 0.42 は n = 100, k = 30 にあまりにも適合しない"
```

Out[20]:



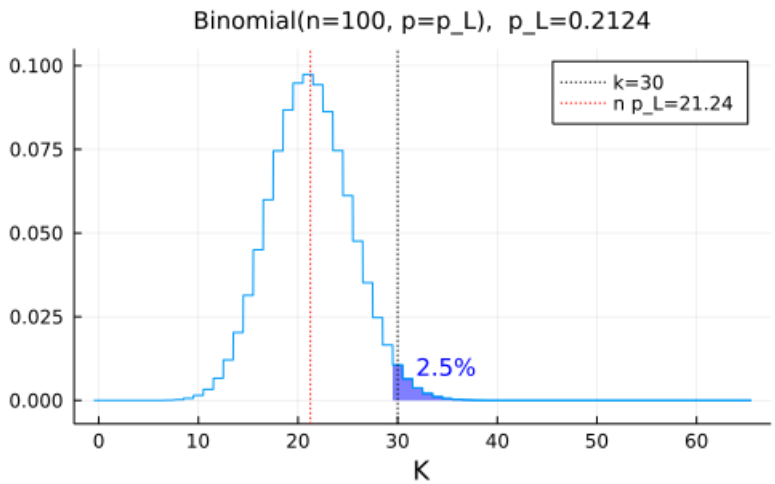
Clopper-Pearsonの信頼区間は具体的には以下のように計算される.

- 二項分布モデル $\mathrm{Binomial}(n, p)$ において成功回数が k 回以上になる確率が $\alpha/2$ に等しくなる p を求め p_L と書く.
- 二項分布モデル $\mathrm{Binomial}(n, p)$ において成功回数が k 回以下になる確率が $\alpha/2$ に等しくなる p を求め p_U と書く.
- Clopper-Pearsonの信頼区間は $[p_L, p_U]$ になる.

以下は $\alpha = 5\%$ の場合の $p_L = p\_L$ と $p_U = p\_U$ の図である.

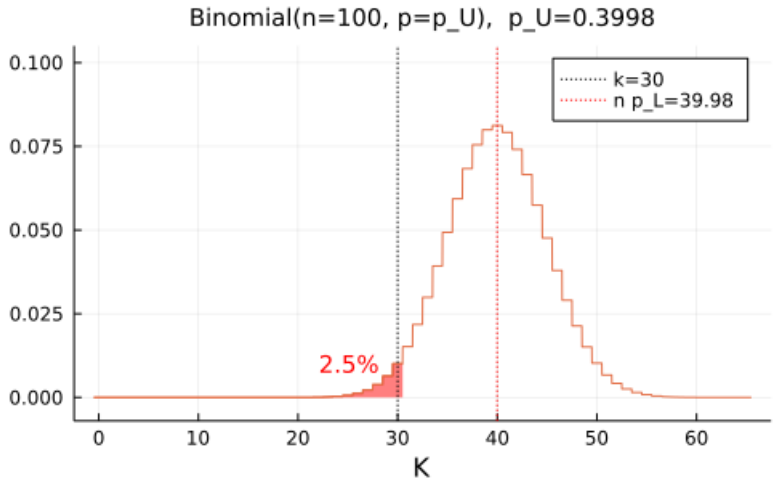
In [21]:
```
1 var"下側では p = p_L がぎりぎり"
```

Out[21]:



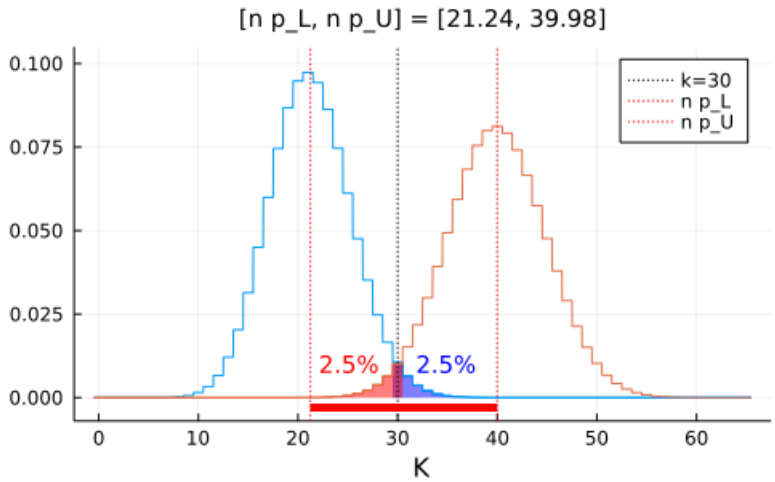
In [22]:
```
var"上側では p = p_U がぎりぎり"
```

Out[22]:



In [23]:
```
var"n × 95%信頼区間"
```

Out[23]:



2.2 問題: n=100, k=30 の95%Clopper-Pearson信頼区間

「$n = 100$ 回中 $k = 30$ 回成功」というデータから決まる 95% Clopper-Pearson信頼区間を求めよ. (値は小数点以下第4桁まで求めよ.)

解答例: データ $n = 100, k = 30$ から決まる 95% Clopper-Pearson信頼区間は

$$[0.2124, 0.3998]$$

である. (計算の仕方については以下を見よ.)

解答終

In [24]:
```
# n回中k回成功というデータが得られたとする
n, k = 100, 30
```

Out[24]: (100, 30)

In [25]:
```
# 有意水準を5%に設定 (α は \alpha TAB で入力できる)
α = 0.05
```

Out[25]: 0.05

In [26]:
```
# 二項分布Binomial(n,p)でk以上になる確率が2.5%になるp
p_L = quantile(Beta(k, n-k+1), α/2)
```

Out[26]: 0.21240642048953662

In [27]:
```
# 検算
ccdf(Binomial(n, p_L), k-1)
```

Out[27]: 0.025000000000000026

In [28]:
```
# 二項分布Binomial(n,p)でk以下になる確率が2.5%になるp
p_U = quantile(Beta(k+1, n-k), 1 - α/2)
```

Out[28]: 0.39981467617980404

In [29]:
```
# 検算
cdf(Binomial(n, p_U), k)
```

Out[29]: 0.02500000000000001

In [30]:
```
# n回中k回成功というデータから決まるClopper-Pearsonの信頼区間
@show [p_L, p_U];
```

```
[p_L, p_U] = [0.21240642048953662, 0.39981467617980404]
```

In [31]:
```
@show round.([p_L, p_U]; digits=4);
```

```
round.([p_L, p_U]; digits = 4) = [0.2124, 0.3998]
```

以上で求めたClopper-Pearson信頼区間は R (https://cran.r-project.org/) の `binom.test` の結果に一致する.

```
> binom.test(30, 100)

    Exact binomial test

data:  30 and 100
number of successes = 30, number of trials = 100, p-value = 7.85e-05
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.2124064 0.3998147
sample estimates:
probability of success
                   0.3
```

WolframAlphaで求めるには以下のようにすればよい.

- quantile(BetaDistribution(30, 71), 0.025) (https://www.wolframalpha.com/input?i=quantile%28BetaDistribution%2830%2C+71%29%2C+0.025%29) → 0.212406
- quantile(BetaDistribution(31, 70), 0.975) (https://www.wolframalpha.com/input?i=quantile%28BetaDistribution%2831%2C+70%29%2C+0.975%29) → 0.399815

2.3 問題: n=400, k=120 の95% Clopper-Pearson信頼区間

前節の例は「$n = 100$ 回中 $k = 30$ 回成功」というデータから決まる 95% Clopper-Pearson信頼区間であった.

n を 4 倍した「$n = 400$ 回中 $k = 120$ 回成功」というデータから決まる 95% Clopper-Pearson信頼区間を求め, n を4倍すると信頼区間の幅は約半分になることを確認せよ. (値は小数点以下第4桁まで求めよ.)

注意: 大雑把には, 信頼区間の幅は $\sqrt{n}$ に反比例する. 例えば, 精度を1桁上げるためには, n を 100 倍する必要がある.

解答例: データ $n = 100$, $k = 30$ から決まる 95% Clopper-Pearson信頼区間は

$$[0.2124, 0.3998]$$

であった. データ $n = 400$, $k = 120$ から決まる 95% Clopper-Pearson信頼区間は

$$[0.2555, 0.3475]$$

になる. 確かに幅が約半分になっているように見える.

解答終

In [32]:
```
# n回中k回成功というデータが得られたとする
n, k = 400, 120
```

Out[32]: (400, 120)

In [33]:
```
# 有意水準を5%に設定 (α は \alpha TAB で入力できる)
α = 0.05
```

Out[33]: 0.05

In [34]:
```
# 二項分布Binomial(n,p)でk以上になる確率が2.5%になるp
p_L = quantile(Beta(k, n-k+1), α/2)
```

Out[34]: 0.25546672809530985

In [35]:
```
# 検算
ccdf(Binomial(n, p_L), k-1)
```

Out[35]: 0.025000000000000112

In [36]:
```
# 二項分布Binomial(n,p)でk以下になる確率が2.5%になるp
p_U = quantile(Beta(k+1, n-k), 1 - α/2)
```

Out[36]: 0.3475218790736496

In [37]:
```
# 検算
cdf(Binomial(n, p_U), k)
```

Out[37]: 0.025000000000000085

In [38]:
```
# n回中k回成功というデータから決まるClopper-Pearsonの信頼区間
@show [p_L, p_U];
```

```
[p_L, p_U] = [0.25546672809530985, 0.3475218790736496]
```

In [39]:
```
@show round.([p_L, p_U]; digits=4);
```

```
round.([p_L, p_U]; digits = 4) = [0.2555, 0.3475]
```

以上で求めたClopper-Pearson信頼区間は R (https://cran.r-project.org/) の `binom.test` の結果に一致する.

```
> binom.test(120, 400)

    Exact binomial test

data:  120 and 400
number of successes = 120, number of trials = 400, p-value = 7.666e-16
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.2554667 0.3475219
sample estimates:
probability of success
                   0.3
```

WolframAlphaで求めるには以下のようにすればよい.

- quantile(BetaDistribution(120, 281), 0.025) (https://www.wolframalpha.com/input?i=quantile%28BetaDistribution%28120%2C+281%29%2C+0.025%29) → 0.255467
- quantile(BetaDistribution(121, 280), 0.975) (https://www.wolframalpha.com/input?i=quantile%28BetaDistribution%28121%2C+280%29%2C+0.975%29) → 0.347522

3 関連の問題

3.1 問題: Waldの信頼区間との比較

以下では信頼度を $1-\alpha$ と表す($0<\alpha<1$ は有意水準と呼ばれる). さらに,

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

とおく. これは標準正規分布において c_α 以上になる確率が $\alpha/2$ になることを意味している. 例えば, $\alpha = 5\%$ のとき,

$$c_{0.05} = 1.95996\cdots \approx 1.96.$$

この 1.96 は非常に有名な数値である. 大雑把な分析で十分な場合には 2 で代替してよい.

Waldの信頼区間の定義: データ「n 回中 k 回成功」が得られたとする. このとき,

$$\hat{p} = \frac{k}{n}, \quad \widehat{\text{SE}} = \sqrt{\frac{p(1-p)}{n}}$$

とおき,

$$\hat{p} - c_\alpha \widehat{\text{SE}} \le p \le \hat{p} + c_\alpha \widehat{\text{SE}}$$

を成功確率パラメータ p の信頼度 $1-\alpha$ の **Waldの信頼区間** と呼ぶ. 95% Wald信頼区間は近似的に

$$\hat{p} - 1.96\widehat{\text{SE}} \le p \le \hat{p} + 1.96\widehat{\text{SE}}$$

と書ける.

(1) データ「$n=100$ 回中 $k=30$ 回成功」の 95% Wald信頼区間を求めよ.

(2) データ「$n=400$ 回中 $k=120$ 回成功」の 95% Wald信頼区間を求めよ.

(3) これらの場合にWald信頼区間とClopper-Pearson信頼区間は大きくは違わないことを確認せよ.

数値は小数点以下第4桁まで求めよ.

注意: Waldの信頼区間の計算はClopper-Pearsonの信頼区間と比較すると圧倒的に易しい. k と $n-k$ が十分に大きければ, 実用的には計算が容易なWaldの信頼区間で用が足りることが多い.

解答例: (1),(3) データ「$n=100$ 回中 $k=30$ 回成功」の 95% Wald信頼区間は

$$[0.2102, 0.3898].$$

同じデータから決まる95% Clopper-Pearson信頼区間は

$$[0.2124, 0.3998]$$

確かに大きくは違わない.

(2),(3) データ「$n=400$ 回中 $k=120$ 回成功」の 95% Wald信頼区間は

$$[0.2551, 0.3449].$$

同じデータから決まる95% Clopper-Pearson信頼区間は

$$[0.2555, 0.3475]$$

この2つは近似的によく一致しているように見える.

解答終

In [40]:
```
α = 0.05
c = quantile(Normal(), 1 - α/2)
```

Out[40]:
```
1.9599639845400576
```

In [41]:
```
# (1)
n, k = 100, 30
p̂ = k/n
SE = √(p̂*(1 - p̂)/n)
p_L, p_U = p̂ - c*SE, p̂ + c*SE
@show [p_L, p_U]
@show round.([p_L, p_U]; digits=4);
```

```
[p_L, p_U] = [0.21018316681457927, 0.3898168331854207]
round.([p_L, p_U]; digits = 4) = [0.2102, 0.3898]
```

In [42]:
```
# (2)
n, k = 400, 120
p̂ = k/n
SE = √(p̂*(1 - p̂)/n)
p_L, p_U = p̂ - c*SE, p̂ + c*SE
@show [p_L, p_U]
@show round.([p_L, p_U]; digits=4);
```

```
[p_L, p_U] = [0.25509158340728966, 0.3449084165927103]
round.([p_L, p_U]; digits = 4) = [0.2551, 0.3449]
```

WolframAlphaで

- quantile(NormalDistribution(0,1), 0.975) (https://www.wolframalpha.com/input?i=quantile%28NormalDistribution%280%2C1%29%2C+0.975%29)

によって, $c = 1.95996$ を得ることができれば, 後は平方根が計算できる電卓ですべてを計算できる.

3.2 問題: Poisson分布とガンマ分布の関係

$\lambda > 0$ のとき, 公式

$$\sum_{i \geq k} \binom{n}{i} p^i (1-p)^{n-i} = \frac{\int_0^p t^{k-1}(1-t)^{n-i}\, dt}{B(k, n-k+1)}$$

において, $p = \lambda/n$ とおき, $t = x/n$ とおき, $n \to \infty$ の極限を取ることによって次の公式が得られることを示せ:

$$e^{-\lambda} \sum_{i \geq k} \frac{\lambda^i}{i!} = \frac{\int_0^\lambda e^{-x} x^{k-1}\, dx}{\Gamma(k)}.$$

これは次が成立することを意味している:

$$\begin{aligned}&(\text{Poisson分布 } \mathrm{Poisson}(\lambda) \text{ において } k \text{ 以上になる確率})\\ &= (\text{ガンマ分布 } \mathrm{Gamma}(k, 1) \text{ において } \lambda \text{ 以下になる確率}).\end{aligned}$$

注意: この結果は

- 単位時間のあいだに平均して 1 回起こるイベントが時間 λ のあいだに起こる回数が k 以上になる確率

と次が等しいことを意味している:

- 単位時間のあいだに平均して 1 回起こるイベントが k 回起こるまでにかかる時間が λ 以下の確率.

注意: 意味的には, ガンマ分布に収束するのは負の二項分布である. ベータ分布は負の二項分布とも関係がある. それを使えば負の二項分布を使っても同様の結果を得ることができる.

解答例: $p = \lambda/n$ とおき, $n \to \infty$ とすると,

$$\begin{aligned}\binom{n}{i} p^i (1-p)^{n-i} &= \frac{n(n-1)\cdots(n-i+1)}{i!} \frac{\lambda^i}{n^i} \left(1 - \frac{\lambda}{n}\right)^{n-i} \\ &= \underbrace{\frac{n(n-1)\cdots(n-i+1)}{n^i}}_{\to 1} \underbrace{\left(1 - \frac{\lambda}{n}\right)^{n-i}}_{\to \exp(-\lambda)} \frac{\lambda^i}{i!} \to e^{-\lambda} \frac{\lambda^i}{i!}\end{aligned}$$

なので,

$$\sum_{i \geq k} \binom{n}{i} p^i (1-p)^{n-i} \to e^{-\lambda} \sum_{i \geq k} \frac{\lambda^i}{i!}.$$

$t = x/n$ とおき, $n \to \infty$ とすると,

$$\begin{aligned}&n^k B(k, n-k+1) \to \Gamma(k), \\ &n^k \int_0^p t^{k-1}(1-t)^{n-i}\, dt = \int_0^\lambda x^{k-1} \left(1 - \frac{x}{n}\right)^{n-i} dt \to \int_0^\lambda e^{-x} x^{k-1}\, dx\end{aligned}$$

となることより,

$$\frac{\int_0^p t^{k-1}(1-t)^{n-i}\,dt}{B(k,n-k+1)} = \frac{n^k\int_0^p t^{k-1}(1-t)^{n-i}\,dt}{n^k B(k,n-k+1)} \to \frac{\int_0^\lambda e^{-x}x^{k-1}\,dx}{\Gamma(k)}.$$

解答終

4 おまけ: 二項分布モデルのBayes統計との関係

このノート群ではBayes統計に深く触れるつもりはない.

しかし, 興味がある人は多いと思われるので, 二項分布モデルの場合のBayes統計について簡単に説明しておくことにする.

4.1 二項分布モデルのBayes統計

二項分布モデル $\mathrm{Binomial}(n,p)$ とパラメータ p に関する確率密度函数 $\varphi(p)$ $(0\leq p\leq 1)$の組をモデルとして採用していると仮定する. すなわち, このモデルは離散変数 k と連続変数 p に関する確率分布を記述する次の函数で与えられていると考えることができる:

$$P(k,p|n) = \binom{n}{k}p^k(1-p)^{n-k}\varphi(p) \quad (k=0,1,2,\dots,\ 0\leq p\leq 1).$$

この $P(k,p|n)$ の値は 0 以上であり,

$$\sum_{k=0}^{n} P(k,p|n) = \varphi(p), \qquad \int_0^1 \left(\sum_{k=0}^{n} P(k,p|n)\right)dp = \int_0^1 \varphi(p)\,dp = 1$$

を満たしている.

「n 回中 k 回成功」というデータが得られたとき, データと同じ数値がモデル内で生成されたという条件が定める条件付き確率分布を考えることが **Bayes法** の基本である(Bayesの定理やBayesルールのような用語を持ち出す必要はない).
この場合には p に関する条件付き確率分布を考える. その確率密度函数を $\varphi(p|n,k)$ と書くと,

$$P(k|n) = \int_0^1 P(k,p|n)\,dp = \int_0^1 \binom{n}{k}p^k(1-p)^{n-k}\varphi(p)\,dp,$$
$$\varphi(p|n,k) = \frac{P(k,p|n)}{P(k|n)} = \frac{\binom{n}{k}p^k(1-p)^{n-k}\varphi(p)}{\int_0^1 \binom{n}{k}p^k(1-p)^{n-k}\varphi(p)\,dp} = \frac{p^k(1-p)^{n-k}\varphi(p)}{\int_0^1 p^k(1-p)^{n-k}\varphi(p)\,dp}.$$

$\varphi(p)$ は **事前分布** (prioir)と呼ばれ, $\varphi(p|n,k)$ はデータ「n 回中 k 回成功」に関する **事後分布** (posterior)と呼ばれる.

4.2 事前分布が共役事前分布(ベータ分布)の場合

この場合には, 事前分布としてベータ分布 $\mathrm{Beta}(a,b)$ を採用すると計算が著しく易しくなる. 以下では事前分布は $\mathrm{Beta}(a,b)$ であると仮定する:

$$\varphi(p) = \frac{p^{a-1}(1-p)^{b-1}}{B(a,b)}.$$

このとき,

$$p^k(1-p)^{n-k}\varphi(p) = p^{a+k-1}(1-p)^{b+n-k-1},$$
$$\int_0^1 p^k(1-p)^{n-k}\varphi(p)\,dp = B(a+k,b+n-k)$$

なのでデータ「n 回中 k 回成功」に関する事後分布は次の形になる:

$$\varphi(p|n,k) = \frac{p^{a+k-1}(1-p)^{b+n-k-1}}{B(a+k,b+n-k)}.$$

つまり, 事後分布は $\mathrm{Beta}(a+k,b+n-k)$ になる.

ベータ分布 $\mathrm{Beta}(\alpha,\beta)$ は α,β が大きなとき, $p=\alpha/(\alpha+\beta)$ の周囲に集中した分布になる.

ゆえに, 事後分布は $\mathrm{Beta}(a+k,b+n-k)$ は $a+k$ と $b+n-k$ が大きいならば $p=(a+k)/(a+b+n)$ の周囲に集中した分布になり, $k,n-k$ が a,b よりも十分に大きいならば $p=k/n$ に集中した分布になり, 最尤法と同じ結果が得られる.

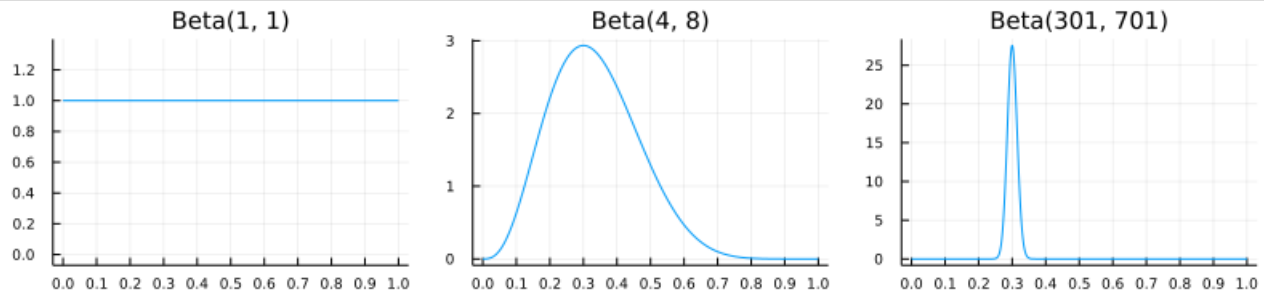
In [43]:
```
function plot_beta(a, b; label="", title="Beta($a, $b)", kwargs...)
    if (a == 0 && b > 0) || (a > 0 && b == 0)
        plot(x -> x^(a-1)*(1-x)^(b-1), 0, 1; ylim=(-0.5, 10.0), label, kwargs...)
    else
        beta = Beta(a, b)
        if a == 1 && b == 1
            plot(beta, 0, 1; ylim=(-0.07, 1.4), label, kwargs...)
        elseif a ≥ 1 && b ≥ 1
            plot(beta, 0, 1; label, kwargs...)
        else
            plot(beta, 0, 1; ylim=(-0.5, 10.0), label, kwargs...)
        end
    end
    plot!(; xtick=0:0.1:1)
    title!(title)
end

function plot_betas(a, b, n, k)
    P0 = plot_beta(a, b)
    P1 = plot_beta(a+k, b+n-k)
    P2 = plot_beta(a+100k, b+100(n-k))
    plot(P0, P1, P2; size=(800, 180), layout=(1, 3))
    plot!(; tickfontsize=6)
end
```

Out[43]: plot_betas (generic function with 1 method)

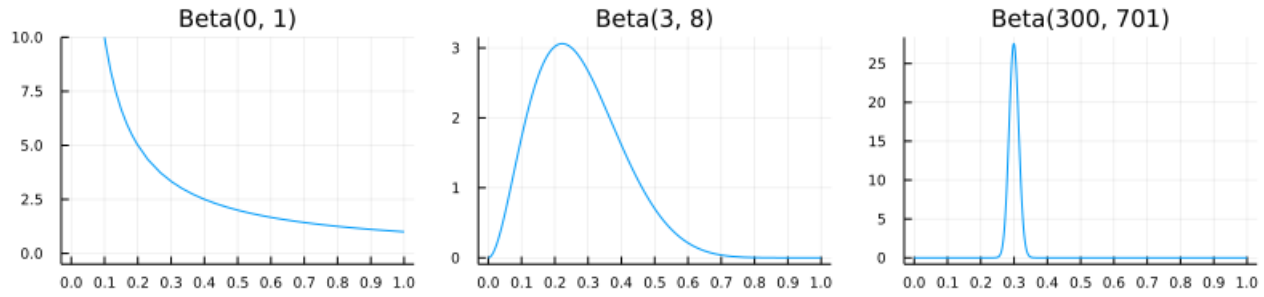
In [44]:
```
# 一様事前分布の場合
plot_betas(1, 1, 10, 3)
```

Out[44]:



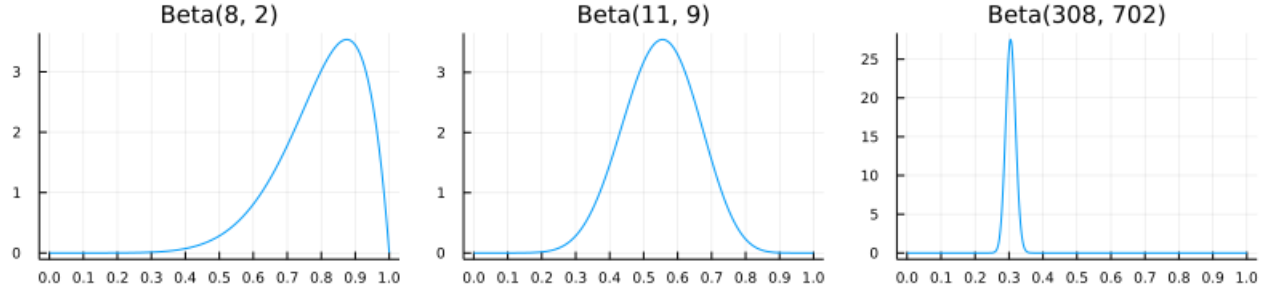
In [45]:
```
# 非対称なimproper事前分布の場合
plot_betas(0, 1, 10, 3)
```

Out[45]:



In [46]:
```
# 偏った事前分布の場合
plot_betas(8, 2, 10, 3)
```

Out[46]:



4.3 二項分布モデルでの片側P値のBayes統計での解釈

事前分布無しの試行回数 n の二項分布モデルにおける仮説「成功確率 p は p_0 以下である」のデータ「n 回中 k 回成功」に関する片側検定のP値を二項分布 $\mathrm{Bionmial}(n, p_0)$ において成功回数が k 以上になる確率と定義し, $\mathrm{pvalue}(\leq p_0|n, k)$ と書くことにする. それは, ベータ分布 $\mathrm{Beta}(k, n-k+1)$ において p_0 以下になる確率に等しいのであった:

$$\mathrm{pvalue}(\leq p_0|n,k) = \sum_{i\geq k} \binom{n}{i} p_0^i (1-p_0)^{n-i} = \frac{\int_0^{p_0} p^{k-1}(1-p)^{n-k}}{B(k, n-k+1)}.$$

これは形式的には, 成功確率 p に関する事前分布が $\mathrm{Beta}(0, 1)$ のときの, 事後分布 $\mathrm{Beta}(k, n-k+1)$ において仮説「成功確率 p は p_0 以下である」が成立する確率に一致する. すなわち, この場合には

- 仮説「成功確率 p は p_0 以下である」のP値

と

- Bayes統計における事後分布で仮説「成功確率 p は p_0 以下である」が成立する確率

がぴったり一致していることになる.

ただし, ベータ分布 $\mathrm{Beta}(\alpha, \beta)$ は α も β も正の場合にのみ定義されているので, 事前分布として $\mathrm{Beta}(0, 1)$ を採用することは形式的である. このような場合に事前分布 $\mathrm{Beta}(0, 1)$ はimproperであるという.

しかし, 事前分布が $\mathrm{Beta}(a, b)$ のとき, $k, n-k$ が a, b よりも十分に大きければ, 事後分布 $\mathrm{Beta}(a+k, b+n-k)$ はほとんど a, b に依存しなくなる. ゆえに, improper事前分布 $\mathrm{Beta}(0, 1)$ を例えば一様事前分布 $\mathrm{Beta}(1, 1)$ やJeffreys事前分布 $\mathrm{Beta}(1/2, 1/2)$ に取り換えても, 上の場合のP値とBayes統計での事後分布における確率の一致は近似的に成立することになる.

この意味で, $k, n-k$ が十分大きな場合にはBayes統計の結果と通常のP値を使った統計分析の結果は実践的な違いを生まない.

注意: だから, 二項分布モデル(やBernoulli試行モデル)の場合を例に使って, P値よりもBayes統計を使うべきだと安易に主張している文献を読むときにはその主張を疑いながら読む必要がある.

注意: 上のようなデータサイズが大きい場合の(近似的)一致は, 正則モデルと呼ばれる「単純な」モデルで広く成立している.

注意: 事前分布が大きく偏っている場合には, その分だけデータサイズをさらに大きくしないと, 上のような近似的な一致は得られない. しかし, 偏りの小さい「おとなしめ」の事前分布が採用されている場合には, それほど大きなデータサイズでなくても, 近似的な一致が成立している可能性が高い.

4.4 Pólyaの壺との関係

事前分布 $\varphi(p)$ と二項分布 $\mathrm{Binomial}(n, p)$ の組で構成されたBayes統計のモデル内における成功回数 k の分布の確率質量函数は

$$P(k|n) = \int_0^1 P(k, p|n)\, dp = \int_0^1 \binom{n}{k} p^k (1-p)^{n-k} \varphi(p)\, dp$$

になるのであった. 事前分布が $\mathrm{Beta}(a, b)$ のとき, これは以下のようになる:

$$P(k|n) = \int_0^1 \binom{n}{k} p^k (1-p)^{n-k} \frac{p^{a-1}(1-p)^{b-1}}{B(a,b)}\, dp = \binom{n}{k} \frac{B(a+k, b+n-k)}{B(a,b)}.$$

さらにこれは, $\Gamma(\alpha)\Gamma(\beta) = \Gamma(\alpha+\beta)B(\alpha, \beta)$ や $\Gamma(x+k) = x(x+1)\cdots(x+k-1)\Gamma(x)$ $(k = 0, 1, 2, \ldots)$を使うと,

$$P(k|n) = \binom{n}{k} \frac{a(a+1)\cdots(a+k-1)\cdot b(b+1)\cdots(b+n-k-1)}{(a+b)(a+b+1)\cdots(a+b+n-1)}$$

と書き直される. これは, 最初に壺の中に赤い玉が a 個と白い玉が b 個入っているときに, そこから無作為に玉を取り出して, その色を記録し, 取り出された玉と同じ色の玉を壺に2個返すことを n 回繰り返したときに, 赤い玉が全部で k 回取り出される確率に一致する($n = 5$ の場合に色々な例を考えてみよ, 仕組みはすぐにわかる). ただし, この解釈は a, b がともに整数の場合にのみ成立するが, 上の式は a, b が整数でなくても確率質量函数を与える. この解釈を **Pólyaの壺** (Pólya's urn) による解釈と呼ぶ.

k の分布の

$$P(k|n) = \int_0^1 \binom{n}{k} p^k (1-p)^{n-k} \varphi(p)\, dp$$

という表示は, k が次のようにランダムに決まっているという解釈を与える:

(B) 最初に事前分布 $\varphi(p)$ によって一定の成功確率 p をランダムに決定し, その後はその成功確率 p のBernoulli試行を n 回行って, 成功回数を k とする.

この解釈は「最初に "才能" p がランダムに決まっている」と要約することができるだろう.

それに対して, Pólyaの壺の解釈は k が次のようにランダムに決まっているという解釈を与える:

(P) 最初に壺の中には赤い玉が a 個と白い玉 b 個が入っている. その後, 赤い玉が壺から取り出されれば壺の中の赤い玉の個数は1個増え, 白い玉についても同様である. そのような試行を n 回繰り返したときに赤い玉が出た回数 k とする.

赤い玉を「成功」とみなすと, この解釈は「最初の才能 a, b は誰もが一定だが, 成功すればするほど成功し易くなり, 失敗すればするほど失敗し易くなる」と要約することができるだろう. Pólyaの壺の試行は「富める者はさらに富み、貧しい者はさらに貧しくなる」と要約されることがある.

以上の(B)と(P)では話が全然違っている. (B)と(P)が与える確率分布が等しいという数学的結果は, 生まれ付きの才能で k がランダムに決まっているのか, それとも「富める者はさらに富み、貧しい者はさらに貧しくなる」という仕組みが働いて最終的に差がついたのかが, 成功回数 k を観察しただけではわからないことを意味している.

In [ ]:
```
1
```