

比率のP値函数達

- 黒木玄
- 2022-09-08

このノートでは A の確率を $P(A)$ と表し, 確率変数 X の函数 $f(X)$ の期待値を $E[f(X)]$ と表す.

予備知識: 二項分布, 正規分布, ベータ分布.

```
In [1]: 1 using Distributions
2 using StatsPlots
3 default(fmt=:png, size=(500, 300),
4         titlefontsize=10, tickfontsize=6, guidefontsize=9)
5
6 # Sterneの信頼区間の定義で使用する
7 using Roots
8 using StatsFuns: logit
```

1 データと統計モデルの設定

当たりまたは外れの目が出るルーレットを n 回まわしたとき当たりが k 回出たと仮定する.

このとき「 n 回中当たりが k 回」をこのルーレットに関するデータと呼ぶ.

そのデータの確率的揺らぎのモデルとして, 二項分布を考える.

すなわち, 次の確率質量函数で定義される k に関する確率分布を考える:

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

ここで, $0 \leq p \leq 1$ であり, $\binom{n}{k}$ は次で定義される二項係数である:

$$\binom{n}{k} = \frac{n(n-1) \cdots (n-k+1)}{k!} = \frac{n!}{k!(n-k)!}.$$

2 P値函数と信頼区間の一般論

以下では具体的な数値 p_0 に関する仮説「当たりが出る確率は p である」と「 n 回中当たりが k 回」というデータの **両立性** (compatibility) もしくは **整合性** (consistency) の指標の1つである(両側検定の) **P値** (P-value) を複数定義して比較してみる.

以下ではそのP値を $\text{pvalue}(k|n, p = p_0)$ のように書くことにする.

データ「 n 回中当たりが k 回」が与えられたときに, パラメータ値 p に $\text{pvalue}(k|n, p = p_0)$ を対応させる函数

$$p \mapsto \text{pvalue}(k|n, p)$$

をデータ「 n 回中当たりが k 回」に関する **P値函数** (P-value function) と呼ぶ.

P値函数のグラフは, 与えられたデータとパラメータ値 p の両立性の様子の全体の視覚化になっている.

$0 \leq \alpha \leq 1$ と仮定する. α は **有意水準** (significance level) と呼ばれる.

データ「 n 回中当たりが k 回」のP値函数が与えられたとき, パラメータ p に関する **信頼度** (confidence level) $1 - \alpha$ の **信頼区間** (confidence interval) が次によって定義される:

$$\text{confint}(k|n, \alpha) = \{ p \mid \text{pvalue}(k|n, p) \geq \alpha \}.$$

すなわち, データ「 n 回中当たりが k 回」に関するパラメータ p の信頼度 $1 - \alpha$ の信頼区間は, データの数値とパラメータ値 p の両立性の指標であるP値が最初に決めておいた閾値 α 以上になるような値 $p = p_0$ 全体の集合である.

この集合が区間にならない場合には, この集合を含む最小の区間を信頼区間と定義することもある.

3 累積分布函数 cdf と分位点函数 quantile

確率分布 D に従う確率変数 X に対して, $x \in \mathbb{R}$ の函数

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

を分布 D の **累積分布関数** (cumulative distribution function, **cdf**)と呼ぶ.

例えば, 試行回数 n , 成功確率 p の二項分布 $\text{Binomial}(n, p)$ の累積分布関数は次のように表される:

$$\text{cdf}(\text{Binomial}(n, p), x) = \sum_{k \leq x} \binom{n}{k} p^k (1-p)^{n-k}.$$

例えば, 平均 μ , 標準偏差 σ の正規分布 $\text{Normal}(\mu, \sigma)$ の累積分布関数は次のように表される:

$$\text{cdf}(\text{Normal}(\mu, \sigma), x) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^x \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right) dt.$$

もしも, 累積分布関数 $p = \text{cdf}(D, x)$ の逆関数が存在するならば, それを

$$x = \text{quantile}(D, p)$$

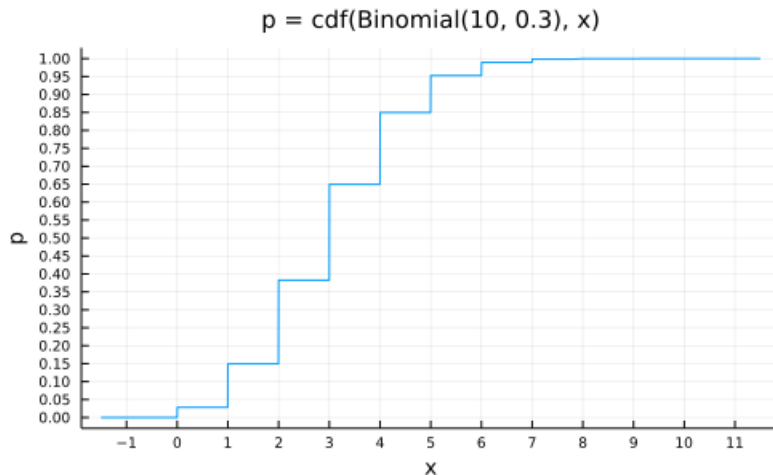
と表し, 分布 D の **分位点関数** (quantile function)と呼ぶ.

例えば, 正規分布の累積分布関数は狭義単調増加関数なので, 分位点関数 $x = \text{quantile}(\text{Normal}(\mu, \sigma), p)$ が定義される.

累積分布関数の逆関数が存在しない場合にも, その代わりとなる便利な関数が定義可能な場合には, それも分位点関数と呼び, 同じ記号で表す.

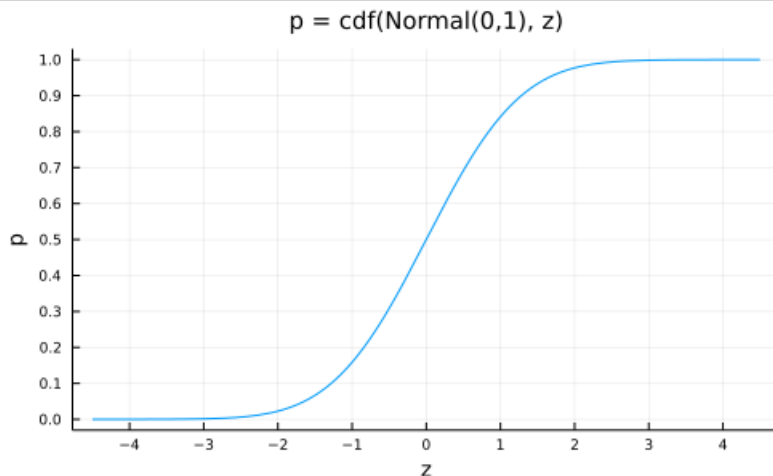
```
In [2]: 1 plot(x → cdf(Binomial(10, 0.3), x), -1.5, 11.5; label="")
        2 plot!(xtick=-1:11, ytick=0:0.05:1)
        3 plot!(xguide="x", yguide="p")
        4 title!("p = cdf(Binomial(10, 0.3), x)")
```

Out[2]:



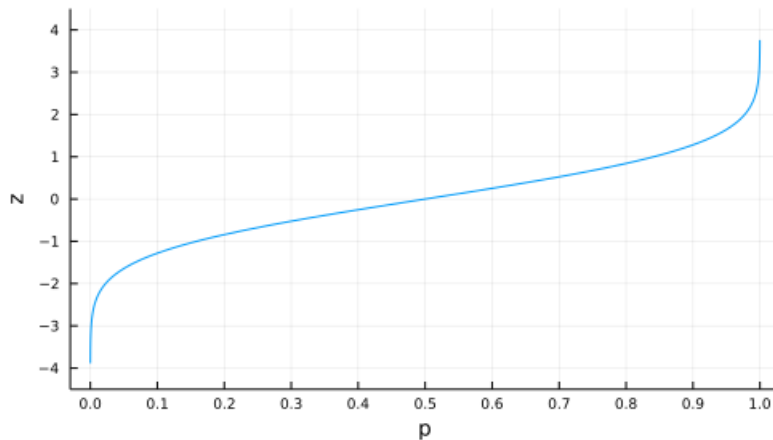
```
In [3]: 1 plot(z → cdf(Normal(0,1), z), -4.5, 4.5; label="")
        2 plot!(xtick=-10:10, ytick=0:0.1:1)
        3 plot!(xguide="z", yguide="p")
        4 title!("p = cdf(Normal(0,1), z)")
```

Out[3]:



```
In [4]: 1 plot(p → quantile(Normal(0,1), p), 0, 1; label="")
2 plot!(ylim=(-4.5, 4.5))
3 plot!(ytick=-10:10, xtick=0:0.1:1)
4 plot!(yguide="z", xguide="p")
5 title!("z = quantile(Normal(0,1), p)")
```

Out[4]: z = quantile(Normal(0,1), p)



3.1 補累積分布関数 ccdf と補分位点関数 cquantile

$\text{ccdf}(D, x) = 1 - \text{cdf}(D, x) = P(X > x)$ を **補累積分布関数** (complementary cumulative distribution function, **ccdf**) と呼ぶ。

補累積分布関数は単調減少関数になる。

補累積分布関数 $p = \text{ccdf}(D, x) = 1 - \text{cdf}(D, x)$ の逆関数を

$$x = \text{cquantile}(D, p) = \text{quantile}(D, 1 - p)$$

と表し, **補分位点関数** (complementary quantile function) と呼ぶ。

補分位点関数も単調減少関数になる。

例えば,

$$\text{cquantile}(\text{Normal}(0, 1), 0.05/2) = 1.95996398454 \dots \approx 1.96$$

は非常に有名である。実践的にはこの値が 1.96 にぴったり等しいとして扱って問題ない。

```
In [5]: 1 cquantile(Normal(0,1), 0.05/2)
```

Out[5]: 1.9599639845400592

4 WilsonのP値と信頼区間

WilsonのP値と信頼区間は二項分布の正規分布近似によって定義される。

4.1 WilsonのP値

k を二項分布 $\text{Binomial}(n, p)$ に従う確率変数とすると,

$$\hat{p} = \frac{k}{n}$$

とおくと, np と $n(1 - p)$ が十分に大きいならば, 中心極限定理より次の近似が成立している:

$$\frac{k - np}{\sqrt{np(1 - p)}} = \frac{\hat{p} - p}{\sqrt{p(1 - p)/n}} \sim \text{Normal}(0, 1), \text{ approximately.}$$

これを用いて, データ「 n 回中当たりは k 回」に関する「当たりが出る確率は p である」という仮説の **WilsonのP値** を次のように定義する:

$$\text{pvalue}_{\text{Wilson}}(k|n, p) = 2\text{ccdf}\left(\text{Normal}(0, 1), \left|\frac{\hat{p} - p}{\sqrt{p(1-p)/n}}\right|\right).$$

4.2 Wilsonの信頼区間

WilsonのP値に対応する信頼度 $1 - \alpha$ の **Wilsonの信頼区間**

$$\text{confint}_{\text{Wilson}}(k|n, \alpha) = \{ p \mid \text{pvalue}_{\text{Wilson}}(k|n, p) \geq \alpha \}$$

がどうなるかを計算してみよう. これはP値函数から信頼区間がどのように得られるかを理解するために手頃な複雑さを持つ良い練習問題になっている.

$z_{\alpha/2}$ を次のように定義する:

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

例えば, $\alpha = 5\%$ のとき, $z = z_{5\%/2} \approx 1.96$ になることは非常に有名である. このとき

$$\begin{aligned} p \in \text{confint}_{\text{Wilson}}(k|n, \alpha) &\iff \text{pvalue}_{\text{Wilson}}(k|n, p) \geq \alpha \\ &\iff 2\text{ccdf}\left(\text{Normal}(0, 1), \left|\frac{\hat{p} - p}{\sqrt{p(1-p)/n}}\right|\right) \geq \alpha \\ &\iff \text{ccdf}\left(\text{Normal}(0, 1), \left|\frac{\hat{p} - p}{\sqrt{p(1-p)/n}}\right|\right) \geq \frac{\alpha}{2} \\ &\iff \left|\frac{\hat{p} - p}{\sqrt{p(1-p)/n}}\right| \leq \text{cquantile}\left(\text{Normal}(0, 1), \frac{\alpha}{2}\right) = z_{\alpha/2} \\ &\iff \frac{(p - \hat{p})^2}{p(1-p)/n} \leq z_{\alpha/2}^2 \\ &\iff (p - \hat{p})^2 \leq \frac{z_{\alpha/2}^2}{n} p(1-p) \\ &\iff \left(1 + \frac{z_{\alpha/2}^2}{n}\right)p^2 - 2\left(\hat{p} + \frac{z_{\alpha/2}^2}{2n}\right)p + \hat{p}^2 \leq 0. \end{aligned}$$

ゆえに,

$$a = 1 + \frac{z_{\alpha/2}^2}{n}, \quad b = \hat{p} + \frac{z_{\alpha/2}^2}{2n}, \quad c = \hat{p}^2$$

とおき,

$$p_{\pm} = \frac{b \pm \sqrt{b^2 - ac}}{a}$$

とおくと, Wilsonの信頼区間は次のように表される:

$$\text{confint}_{\text{Wilson}}(k|n, \alpha) = [p_-, p_+].$$

注意: $f(p)$ を

$$f(p) = \left(1 + \frac{z_{\alpha/2}^2}{n}\right)p^2 - 2\left(\hat{p} + \frac{z_{\alpha/2}^2}{2n}\right)p + \hat{p}^2 = ap^2 - 2bp + c$$

とおくと,

$$f(0) = \hat{p}^2 \geq 0, \quad f(1) = (1 - \hat{p})^2 \geq 0$$

となるので, Wilsonの信頼区間は区間 $[0, 1]$ に自然に含まれることがわかる.

4.3 WilsonのP値と信頼区間の実装例

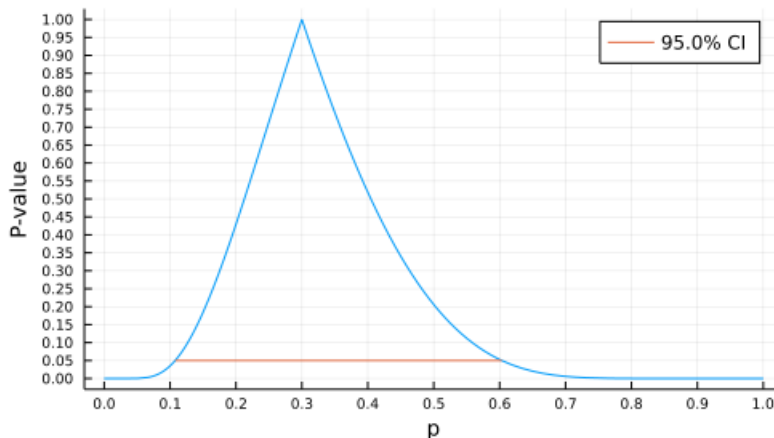
```
In [6]: 1 function pvalue_wilson(k, n, p)
2       p-hat = k/n
3       se = sqrt(p*(1 - p)/n)
4       2*ccdf(Normal(0,1), abs(p-hat - p)/se)
5   end
6
7   function confint_wilson(k, n, alpha)
8       z = quantile(Normal(0,1), alpha/2)
9       p-hat = k/n
10      a, b, c = 1 + z^2/n, p-hat + z^2/(2n), p-hat^2
11      sqrtD = sqrt(b^2 - a*c)
12      (b - sqrtD)/a, (b + sqrtD)/a
13  end
```

Out[6]: confint_wilson (generic function with 1 method)

```
In [7]: 1 @show n, k, alpha = 10, 3, 0.05
2 @show ci = confint_wilson(k, n, alpha) > collect
3
4 P_Wilson = plot(p -> pvalue_wilson(k, n, p), 0, 1; label="")
5 plot!(xguide="p", yguide="P-value")
6 plot!(xtick=0:0.1:1, ytick=0:0.05:1)
7 plot!(ci, fill(alpha, 2); label="$ (100(1-alpha))% CI")
8 title!("Wilson's P-value function for n=$n, k=$k")
```

```
(n, k, alpha) = (10, 3, 0.05) = (10, 3, 0.05)
ci = confint_wilson(k, n, alpha) > collect = [0.10779126740630075, 0.6032218525388553]
```

Out[7]: Wilson's P-value function for n=10, k=3



このように信頼度 $1 - \alpha$ の信頼区間はP値函数のグラフを高さ α で切断して得られる区間になる。

上のグラフは次のように解釈される。

- データ「 $n = 10$ 回中当たりが $k = 3$ 回」と最も **相性がよい** パラメータ値は $p = 0.3$ である(点推定)。
- 0 近くの p や 0.8 以上の p は「 $n = 10$ 回中当たりが $k = 3$ 回」というデータと **両立しない**。
- 閾値を $\alpha = 5\%$ に定めておいて、P値が α 未満になったときに「パラメータ値とデータの数値が両立しなさすぎる」と判断することにしたとき、そのように判断されずにすむ p の範囲はおおよそ 10.8% から 60.3% になる(信頼区間)。

これが 95% 信頼区間の穏健な(自信過剰にならずにすむ)解釈になる。

5 WaldのP値と信頼区間

WaldのP値と信頼区間は、WilsonのP値と信頼区間の構成における二項分布における $\hat{p} = k/n$ の分散 $p(1 - p)/n$ をその推定値 $\hat{p}(1 - \hat{p})/n$ で置き換えることによって定義される。

これによって、WaldのP値と信頼区間の誤差はWilsonのP値と信頼区間よりも大きくなってしまふ。

その代わりに、Waldの信頼区間はWilsonの信頼区間と違って二次方程式を解くことなく容易に計算可能になる。

5.1 WaldのP値

データ「 n 回中当たりは k 回」に関する「当たりが出る確率は p である」という仮説の **WaldのP値** を次のように定義する：

$$\text{pvalue}_{\text{wald}}(k|n, p) = 2\text{ccdf}\left(\text{Normal}(0, 1), \left| \frac{\hat{p} - p}{\sqrt{\hat{p}(1 - \hat{p})/n}} \right| \right).$$

5.2 Waldの信頼区間

WaldのP値に対応する信頼度 $1 - \alpha$ の **Waldの信頼区間**

$$\text{confint}_{\text{wald}}(k|n, \alpha) = \{ p \mid \text{pvalue}_{\text{wald}}(k|n, p) \geq \alpha \}$$

がどうなるかを計算してみよう. 以下のようにWilsonの信頼区間の場合よりもずっと簡単になる.

$$\begin{aligned} p \in \text{confint}_{\text{wald}}(k|n, \alpha) &\iff \text{pvalue}_{\text{wald}}(k|n, p) \geq \alpha \\ &\iff 2\text{ccdf}\left(\text{Normal}(0, 1), \left| \frac{\hat{p} - p}{\sqrt{\hat{p}(1 - \hat{p})/n}} \right| \right) \geq \alpha \\ &\iff \text{ccdf}\left(\text{Normal}(0, 1), \left| \frac{\hat{p} - p}{\sqrt{\hat{p}(1 - \hat{p})/n}} \right| \right) \geq \frac{\alpha}{2} \\ &\iff \left| \frac{\hat{p} - p}{\sqrt{\hat{p}(1 - \hat{p})/n}} \right| \leq \text{cquantile}\left(\text{Normal}(0, 1), \frac{\alpha}{2}\right) = z_{\alpha/2} \\ &\iff \hat{p} - z_{\alpha/2} \sqrt{\frac{p(1 - p)}{n}} \leq p \leq \hat{p} + z_{\alpha/2} \sqrt{\frac{p(1 - p)}{n}}. \end{aligned}$$

すなわち,

$$\text{confint}_{\text{wald}}(k|n, \alpha) = \left[\hat{p} - z_{\alpha/2} \sqrt{\frac{p(1 - p)}{n}}, \hat{p} + z_{\alpha/2} \sqrt{\frac{p(1 - p)}{n}} \right].$$

Waldの信頼区間は区間 $[0, 1]$ をはみ出すこともある.

5.3 WaldのP値と信頼区間の実装例

```
In [8]: 1 function pvalue_wald(k, n, p)
2         p_hat = k/n
3         sehat = sqrt(p_hat*(1 - p_hat)/n)
4         2ccdf(Normal(0,1), abs(p_hat - p)/sehat)
5     end
6
7     function confint_wald(k, n, alpha)
8         z = cquantile(Normal(0,1), alpha/2)
9         p_hat = k/n
10        sehat = sqrt(p_hat*(1 - p_hat)/n)
11        p_hat - z*sehat, p_hat + z*sehat
12    end
```

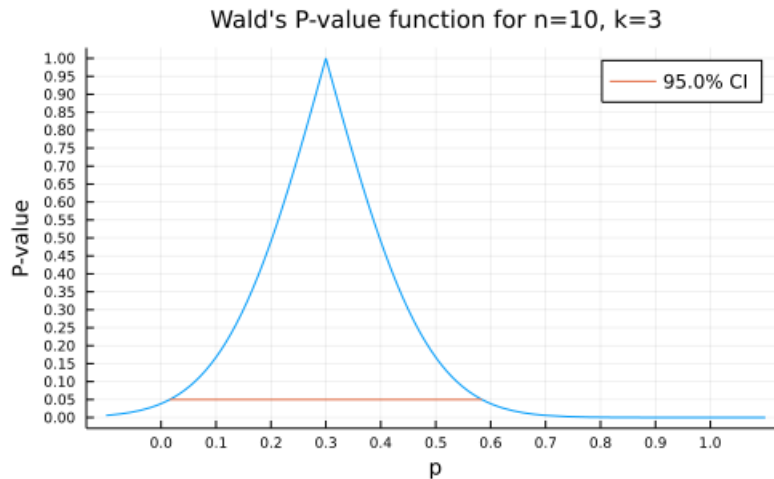
Out[8]: confint_wald (generic function with 1 method)

In [9]:

```
1 @show n, k, α = 10, 3, 0.05
2 @show ci = confint_wald(k, n, α) ▷ collect
3
4 P_Wald = plot(p → pvalue_wald(k, n, p), -0.1, 1.1; label="")
5 plot!(xguide="p", yguide="P-value")
6 plot!(xtick=0:0.1:1, ytick=0:0.05:1)
7 plot!(ci, fill(α,2); label="$ (100(1-α))% CI")
8 title!("Wald's P-value function for n=$n, k=$k")
```

```
(n, k, α) = (10, 3, 0.05) = (10, 3, 0.05)
ci = confint_wald(k, n, α) ▷ collect = [0.0159742349106739, 0.5840257650893261]
```

Out[9]:



WaldのP値関数は左右対称になり、一般に台(0 でない値を取る範囲)が区間 $[0, 1]$ をはみ出す。

WaldのP値関数とそれに付随する信頼区間はこうのように誤差が大きくなってしまふものだが、信頼区間の計算が簡単になるという利点がある。

np と $n(1-p)$ が非常に大きければ、Waldの信頼区間はWilsonの信頼区間とほぼぴったり一致するようになり、二次方程式を解いてWilsonの信頼区間を求めるメリットはなくなる。

コンピュータで自動的に計算するならば、Wilsonの信頼区間の方を計算するべきだが、暗算でどんぶり勘定したい場合にはWaldの信頼区間の方が便利である。

このように統計学的道具にはそれぞれ異なる利点と欠点があり、適材適所で使い分けるのがよい。

6 Clopper-PearsonのP値と信頼区間

二項分布における確率を正規分布で近似せず、正確に求めることによって定義されるP値と信頼区間の一種がClopper-PearsonのP値と信頼区間である。

6.1 Clopper-PearsonのP値

データ「 n 回中当たりは k 回」に関する「当たりが出る確率は p である」という仮説の **Clopper-PearsonのP値** を次のように定義する:

$$pvalue_{CP}(k|n, p) = \min \left(\begin{array}{c} 1 \\ 2cdf(\text{Binomial}(n, p), k) \\ 2ccdf(\text{Binomial}(n, p), k-1) \end{array} \right).$$

ここで、 $cdf(\text{Binomial}(n, p), k)$ と $ccdf(\text{Binomial}(n, p), k-1)$ はそれぞれ、二項分布において k 以下になる確率と k 以下になる確率なので、この定義はおおよそ「片側検定のP値の2倍」になっている。その定義のままだと1より大きくなる場合があるので、1以下になるようにしてある。

6.2 二項分布とベータ分布の関係

Clopper-PearsonのP値に対応する信頼度 $1-\alpha$ のClopper-Pearsonの信頼区間を計算するためには、二項分布とベータ分布の素晴らしい関係が必要になる。次が成立している:

$$\begin{aligned} cdf(\text{Binomial}(n, p), k) &= ccdf(\text{Beta}(k+1, n-k), p) \quad (0 \leq k < n), \\ ccdf(\text{Binomial}(n, p), k-1) &= cdf(\text{Beta}(k, n-k+1), p) \quad (0 < k \leq n). \end{aligned}$$

前者は後者で k を $k + 1$ に置き換えれば得られるので、後者のみを示そう。

後者は具体的には次が成立するということを意味している:

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

両辺の $p = 0$ での値は 0 になる。

ゆえに両辺の p に関する導関数が等しいことを示せば、両辺が等しいことがわかる。

左辺を p で微分すると以下のようになる:

$$\begin{aligned} \frac{\partial}{\partial p} \sum_{i \geq k} \binom{n}{i} p^i (1-p)^{n-i} &= \sum_{i \geq k} \binom{n}{i} i p^{i-1} (1-p)^{n-i} - \sum_{i \geq k} \binom{n}{i} (n-i) p^{i-1} (1-p)^{n-i-1} \\ &= \sum_{i \geq k} \frac{n!}{(i-1)!(n-i)!} p^{i-1} (1-p)^{n-i} - \sum_{i \geq k} \frac{n!}{i!(n-i-1)!} p^{i-1} (1-p)^{n-i-1} \\ &= \sum_{i \geq k} \frac{n!}{(i-1)!(n-i)!} p^{i-1} (1-p)^{n-i} - \sum_{i \geq k+1} \frac{n!}{(i-1)!(n-i)!} p^{i-1} (1-p)^{n-i-1} \\ &= \frac{n!}{(k-1)!(n-k)!} p^{k-1} (1-p)^{n-k} \end{aligned}$$

右辺を p で微分すると

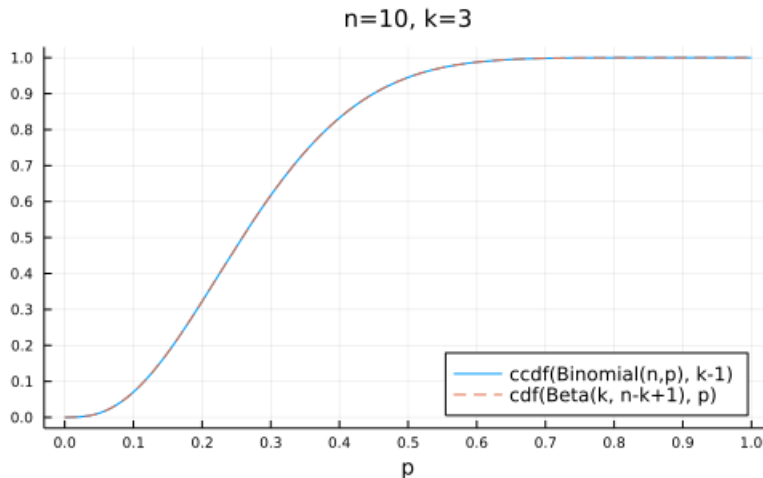
$$\begin{aligned} \frac{1}{B(k, n-k+1)} \frac{\partial}{\partial p} \int_0^p t^{k-1} (1-t)^{n-k} dt &= \frac{1}{B(k, n-k+1)} p^{k-1} (1-p)^{n-k} \\ &= \frac{\Gamma(k + (n-k+1))}{\Gamma(k)\Gamma(n-k+1)} p^{k-1} (1-p)^{n-k} \\ &= \frac{n!}{(k-1)!(n-k)!} p^{k-1} (1-p)^{n-k}. \end{aligned}$$

これで左辺と右辺の導関数が等しいことが証明された。

注意: 二項分布の累積分布関数と補累積分布関数は素朴に和で実装するよりも、上の公式を用いて、ベータ分布の補累積分布関数と累積分布関数で計算するようにした方が効率が良い。ベータ分布の累積分布関数は regularized incomplete Beta function としてコンピュータの基本特殊関数ライブラリで効率的に実装されている。

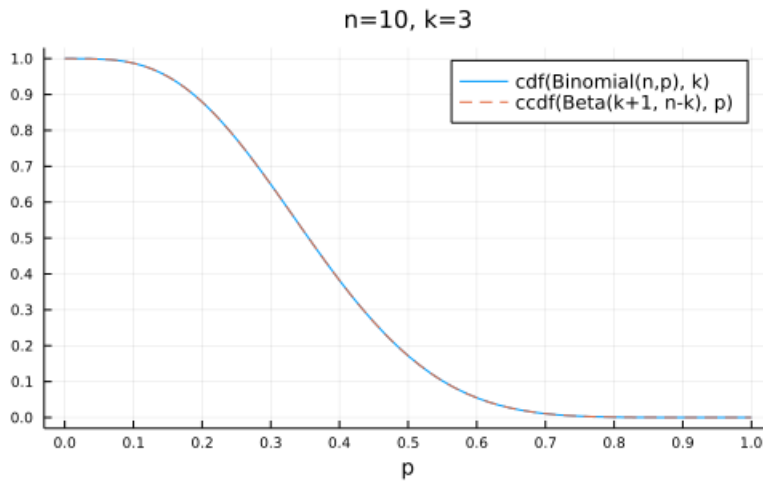
```
In [10]: 1 n, k = 10, 3
2 plot(p -> ccdf(Binomial(n,p), k-1);
3       label="ccdf(Binomial(n,p), k-1)")
4 plot!(p -> cdf(Beta(k, n-k+1), p), 0, 1;
5       label="cdf(Beta(k, n-k+1), p)", ls=:dash)
6 plot!(legend=:bottomright)
7 plot!(xguide="p")
8 plot!(xtick=0:0.1:1, ytick=0:0.1:1)
9 title!("n=$n, k=$k")
```

Out[10]:




```
In [11]: 1 n, k = 10, 3
2 plot(p → cdf(Binomial(n,p), k);
3       label="cdf(Binomial(n,p), k)")
4 plot!(p → ccdf(Beta(k+1, n-k), p), 0, 1;
5       label="ccdf(Beta(k+1, n-k), p)", ls=:dash)
6 plot!(legend=:topright)
7 plot!(xguide="p")
8 plot!(xtick=0:0.1:1, ytick=0:0.1:1)
9 title!("n=$n, k=$k")
```

Out[11]:



6.3 Clopper-Pearsonの信頼区間

Clopper-PearsonのP値に対応する信頼度 $1 - \alpha$ の Clopper-Pearsonの信頼区間

$$\text{confint}_{\text{CP}}(k|n, \alpha) = \{ p \mid \text{pvalue}_{\text{CP}}(k|n, p) \geq \alpha \}$$

がどうなるかを計算してみよう。

$$\begin{aligned} p &\in \text{confint}_{\text{CP}}(k|n, \alpha) \\ &\Leftrightarrow \text{pvalue}_{\text{CP}}(k|n, p) \geq \alpha \\ &\Leftrightarrow \text{cdf}(\text{Binomial}(n, p), k) \geq \alpha/2 \text{ and } \text{ccdf}(\text{Binomial}(n, p), k-1) \geq \alpha/2 \\ &\Leftrightarrow \text{ccdf}(\text{Beta}(k+1, n-k), p) \geq \alpha/2 \text{ and } \text{cdf}(\text{Beta}(k, n-k+1), p) \geq \alpha/2 \\ &\Leftrightarrow p \leq \text{cquantile}(\text{Beta}(k+1, n-k), \alpha/2) \text{ and } p \geq \text{quantile}(\text{Beta}(k, n-k+1), \alpha/2), \end{aligned}$$

ゆえに

$$p_L = \text{quantile}(\text{Beta}(k, n-k+1), \alpha/2), \quad p_U = \text{cquantile}(\text{Beta}(k+1, n-k), \alpha/2)$$

とおくと,

$$\text{confint}_{\text{CP}}(k|n, \alpha) = [p_L, p_U].$$

6.4 Clopper-PearsonのP値と信頼区間の実装例

```
In [12]: 1 function pvalue_cp(k, n, p)
2     bin = Binomial(n, p)
3     min(1, 2cdf(bin, k), 2ccdf(bin, k-1))
4 end
5
6 function confint_cp(k, n, α)
7     quantile(Beta(k, n-k+1), α/2), cquantile(Beta(k+1, n-k), α/2)
8 end
```

Out[12]: confint_cp (generic function with 1 method)

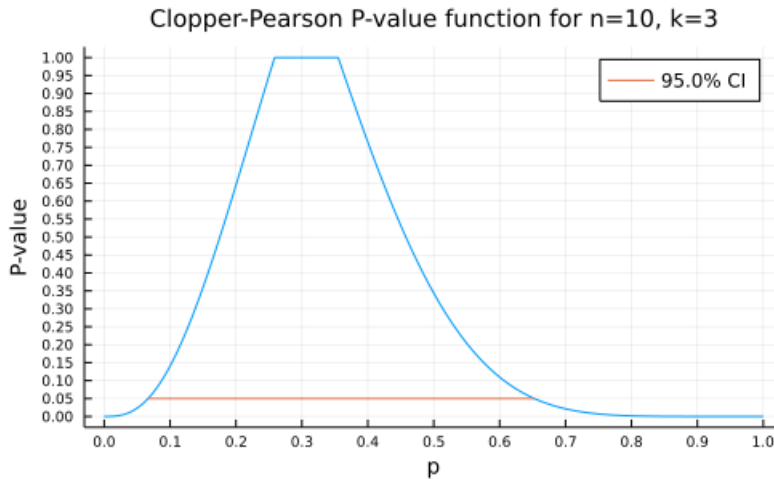
In [13]:

```
1 @show n, k, α = 10, 3, 0.05
2 @show ci = confint_cp(k, n, α) ▷ collect
3
4 P_CP = plot(p → pvalue_cp(k, n, p), 0, 1; label="")
5 plot!(xguide="p", yguide="P-value")
6 plot!(xtick=0:0.1:1, ytick=0:0.05:1)
7 plot!(ci, fill(α,2); label="$ (100(1-α))% CI")
8 title!("Clopper-Pearson P-value function for n=$n, k=$k")
```

(n, k, α) = (10, 3, 0.05) = (10, 3, 0.05)

ci = confint_cp(k, n, α) ▷ collect = [0.06673951117773438, 0.6524528500599971]

Out[13]:



Clopper-Pearsonの信頼区間は小さめの n については過剰に保守的になることが多い。

しかし、上のコードを見れば分かるように、実装は極めてシンプルである。

そして、さらにデータが二項分布 $\text{Binomial}(n, p)$ に従ってランダムに生成されているとき、P値が α 以下になる確率が確実に α 以下になってくれるという利点もある。

7 SterneのP値と信頼区間

SterneのP値と信頼区間も、二項分布における確率を正規分布で近似せずに正確に求めることによって定義される。Sterneの信頼区間の方がClopper-Pearsonの信頼区間よりも概ね狭くなってくれるという利点があるが、Clopper-Pearsonの信頼区間の場合と違ってシンプルな公式がないので、実装はかなり面倒になる。

7.1 SterneのP値と信頼区間

データ「 n 回中当たりは k 回」に関する「当たりが出る確率は p である」という仮説の **SterneのP値** を次のように定義する：

$$\text{pvalue}_{\text{Sterne}}(k|n, p) = \sum_{P(i|n, p) \leq P(k|n, p)} P(i|n, p).$$

ここで、

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

SterneのP値は二項分布 $\text{Binomial}(n, p)$ 内でデータの数値よりも生じる確率が低い値の確率の総和になる。

これに対応するSterneの信頼区間は

$$\text{confint}_{\text{Sterne}}(k|n, \alpha) = \{ p \mid \text{pvalue}_{\text{Sterne}}(k|n, p) \geq \alpha \}$$

であるが、この定義のままだと区間にならない場合があるので、実装時にはこの集合を含む最小の区間として実装することにする。

7.2 SterneのP値と信頼区間の実装例

効率的でない実装でよいならば、P値関数の定義は、

```
In [14]: 1 x ≲ y = x < y || x ≈ y
2
3 function pvalue_sterne_naive(k, n, p)
4     bin = Binomial(n, p)
5     sum(pdf(bin, i) for i in support(bin) if pdf(bin, i) ≲ pdf(bin, k))
6 end
```

Out[14]: pvalue_sterne_naive (generic function with 1 method)

で十分だが、次のセルのように計算は非常に遅くなる。

```
In [15]: 1 @time pvalue_sterne_naive(3*10^6, 10*10^6, 0.301)
2 @time pvalue_sterne_naive(3*10^6, 10*10^6, 0.301)
3 @time pvalue_sterne_naive(3*10^6, 10*10^6, 0.301)

2.910797 seconds
2.933301 seconds
2.907158 seconds
```

Out[15]: 5.354898851685295e-12

```
In [16]: 1 x ≲ y = x < y || x ≈ y
2
3 _pdf_le(x, (dist, y)) = pdf(dist, x) ≲ y
4
5 function _search_boundary(f, x0, Δx, param)
6     x = x0
7     if f(x, param)
8         while f(x - Δx, param) x -= Δx end
9     else
10        x += Δx
11        while !f(x, param) x += Δx end
12    end
13    x
14 end
15
16 function pvalue_sterne(dist::DiscreteUnivariateDistribution, x)
17     Px = pdf(dist, x)
18     Px == 0 && return Px
19     Px == 1 && return Px
20     m = mode(dist)
21     Px ≈ pdf(dist, m) && return one(Px)
22     if x < m
23         y = _search_boundary(_pdf_le, 2m - x, 1, (dist, Px))
24         cdf(dist, x) + ccdf(dist, y-1)
25     else # x > m
26         y = _search_boundary(_pdf_le, 2m - x, -1, (dist, Px))
27         cdf(dist, y) + ccdf(dist, x-1)
28     end
29 end
30
31 function pvalue_sterne(k, n, p)
32     pvalue_sterne(Binomial(n, p), k)
33 end
34
35 function confint_sterne(k, n, α)
36     a, b = confint_cp(k, n, α/10)
37     ps = find_zeros(a-√eps(), b+√eps()) do p
38         logit(0 < p ≤ 1 ? pvalue_sterne(k, n, p) : zero(p)) - logit(α)
39     end
40     first(ps), last(ps)
41 end
```

Out[16]: confint_sterne (generic function with 1 method)

```
In [17]: 1 @time pvalue_sterne(3*10^6, 10*10^6, 0.301)
2 @time pvalue_sterne(3*10^6, 10*10^6, 0.301)
3 @time pvalue_sterne(3*10^6, 10*10^6, 0.301)

0.000042 seconds (8 allocations: 1.875 KiB)
0.000010 seconds (8 allocations: 1.875 KiB)
0.000009 seconds (8 allocations: 1.875 KiB)
```

Out[17]: 5.354898860502031e-12

```
In [18]: 1 @time confint_sterne(3*10^6, 10*10^6, 0.001)
2 @time confint_sterne(3*10^6, 10*10^6, 0.001)
3 @time confint_sterne(3*10^6, 10*10^6, 0.001)
```

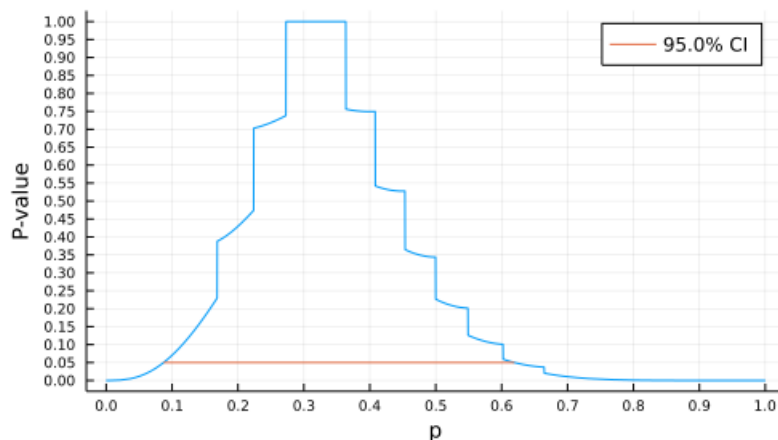
```
0.134561 seconds (51.68 k allocations: 3.764 MiB, 97.91% compilation time)
0.002748 seconds (11.48 k allocations: 2.402 MiB)
0.002652 seconds (11.48 k allocations: 2.402 MiB)
```

```
Out[18]: (0.2995232976841354, 0.30047704792862706)
```

```
In [19]: 1 @show n, k,  $\alpha$  = 10, 3, 0.05
2 @show ci = confint_sterne(k, n,  $\alpha$ ) ▷ collect
3
4 P_Sterne = plot(p → pvalue_sterne(k, n, p), 0, 1; label="")
5 plot!(xguide="p", yguide="P-value")
6 plot!(xtick=0:0.1:1, ytick=0:0.05:1)
7 plot!(ci, fill( $\alpha$ ,2); label="$ (100(1- $\alpha$ ))% CI")
8 title!("Sterne's P-value function for n=$n, k=$k")
```

```
(n, k,  $\alpha$ ) = (10, 3, 0.05) = (10, 3, 0.05)
ci = confint_sterne(k, n,  $\alpha$ ) ▷ collect = [0.0872644339141502, 0.6194106589132863]
```

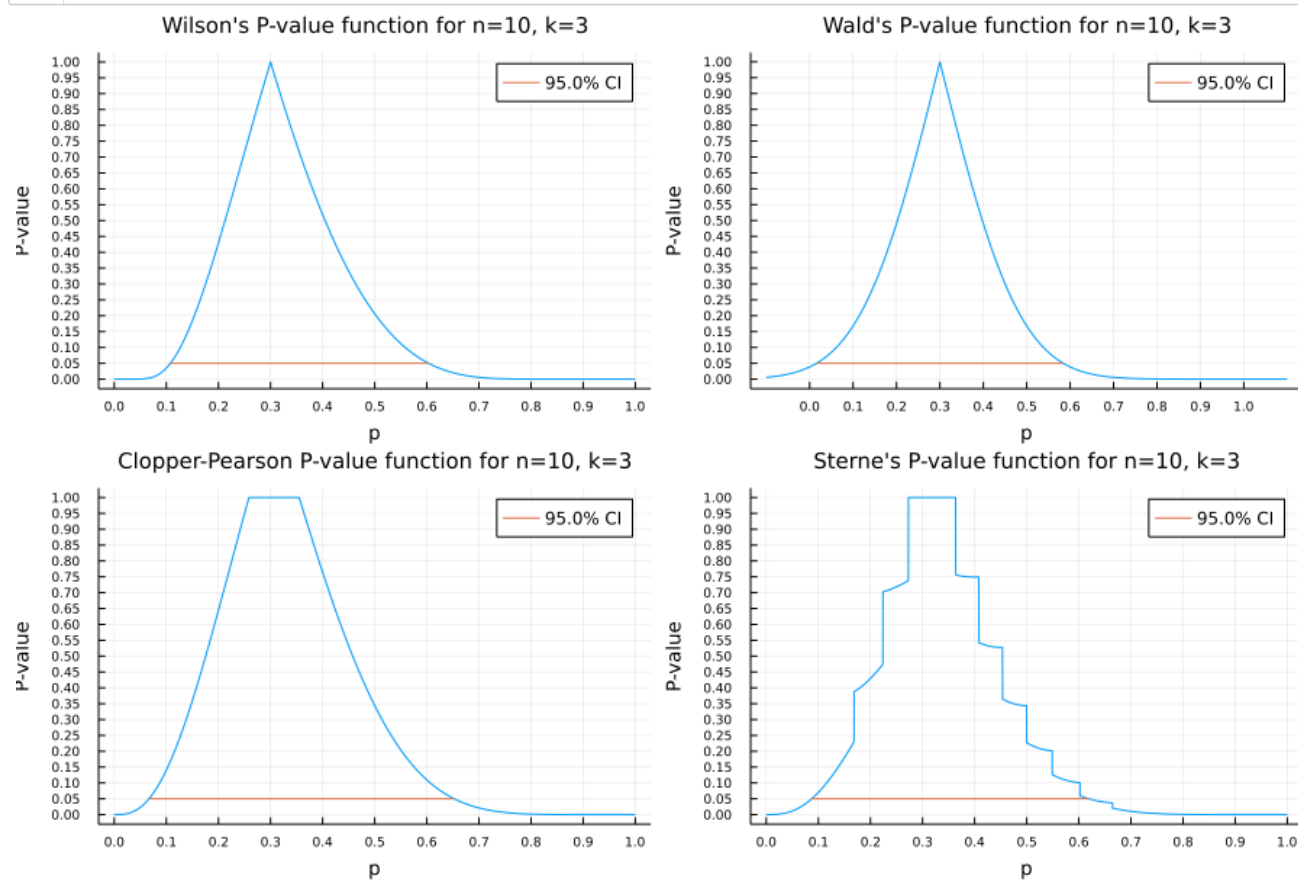
```
Out[19]: Sterne's P-value function for n=10, k=3
```



8 4種のP値関数と信頼区間を同時にプロット

```
In [20]: 1 plot(P_Wilson, P_Wald, P_CP, P_Sterne;  
2          layout=(2, 2), size=(900, 600))
```

Out[20]:



In []:

1