

比率のP値函数達

- 黒木玄
- 2022-09-08
- Copyright 2022 Gen Kuroki
- License: <https://opensource.org/licenses/MIT> (<https://opensource.org/licenses/MIT>)
- nbviewer版 (<https://nbviewer.org/github/genkuroki/public/blob/main/0036/P-value%20functions%20of%20proportions.ipynb>)
- repository版 (<https://github.com/genkuroki/public/blob/main/0036/P-value%20functions%20of%20proportions.ipynb>)
- ダウンロードPDF (<https://github.com/genkuroki/public/raw/main/0036/P-value%20functions%20of%20proportions.pdf>)
- PDF版(repository) (<https://github.com/genkuroki/public/blob/main/0036/P-value%20functions%20of%20proportions.pdf>)

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. 終わりの方ではベイズ統計.

目次

- [1 データと統計モデルの設定](#)
- [2 P値函数と信頼区間の一般論](#)
- [3 累積分布函数 cdf と分位点函数 quantile](#)
- [4 補累積分布函数 ccdf と補分位点函数 cquantile](#)
- ▼ [5 WilsonのP値と信頼区間](#)
 - [5.1 WilsonのP値](#)
 - [5.2 Wilsonの信頼区間](#)
 - [5.3 WilsonのP値と信頼区間の実装例とP値函数の解釈の仕方](#)
- ▼ [6 WaldのP値と信頼区間](#)
 - [6.1 WaldのP値](#)
 - [6.2 Waldの信頼区間](#)
 - [6.3 WaldのP値と信頼区間の実装例](#)
- ▼ [7 Clopper-PearsonのP値と信頼区間](#)
 - [7.1 Clopper-PearsonのP値](#)
 - [7.2 二項分布とベータ分布の関係](#)
 - [7.3 Clopper-Pearsonの信頼区間](#)
 - [7.4 Clopper-PearsonのP値と信頼区間の実装例](#)
- ▼ [8 SterneのP値と信頼区間](#)
 - [8.1 SterneのP値と信頼区間](#)
 - [8.2 SterneのP値と信頼区間の実装例](#)
- [9 4種のP値函数と信頼区間を同時にプロット](#)
- ▼ [10 ベイズ統計での信用区間に対応するP値函数](#)
 - [10.1 二項分布モデルのHDI版信用区間](#)
 - [10.2 HDI版信用区間に対応するP値函数](#)
 - [10.3 HDI版の信用区間とそれに対応するベイズ版P値函数の実装例](#)
 - [10.4 WilsonのP値函数とベイズ版P値函数のHDI版の比較](#)
 - [10.5 ETI版の信用区間とそれに対応するP値函数](#)
 - [10.6 ETI版の信用区間とそれに対応するベイズ版P値函数の実装例](#)
 - [10.7 WilsonのP値函数とベイズ版P値函数のETI版の比較](#)
 - [10.8 「主義の押し付け」をやめよう](#)

```
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
        7 using Roots
        8 using StatsFuns: logit
        9
       10 # highest density interval (HDI) の構成で使用
       11 using Optim
```

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、次の確率質量関数で定義される 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 $pvalue(k|n, p = p_0)$ 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 $pvalue(k|n, p = p_0)$ を対応させる函数

$$p \mapsto pvalue(k|n, p)$$

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：

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

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 **累積分布函数** (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 $\text{Binomial}(n, p)$ の累積分布函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 $\text{Normal}(\mu, \sigma)$ の累積分布函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、累積分布函数 $p = \text{cdf}(D, x)$ の逆函数が存在するならば、それを

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

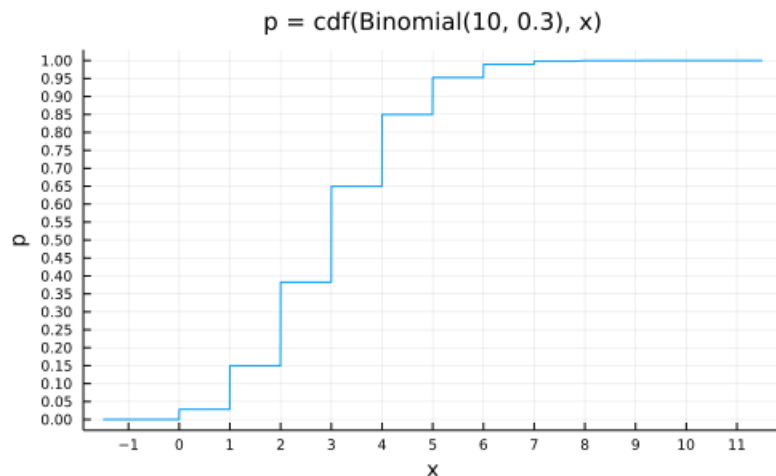
と表し、分布 D の **分位点函数** (quantile function) と呼ぶ。

例えば、正規分布の累積分布函数は狭義単調増加函数なので、分位点函数 $x = \text{quantile}(\text{Normal}(\mu, \sigma), p)$ が定義される。

累積分布函数の逆函数が存在しない場合にも、その代わりとなる便利な函数が定義可能な場合には、それも分位点函数と呼び、同じ記号で表す。

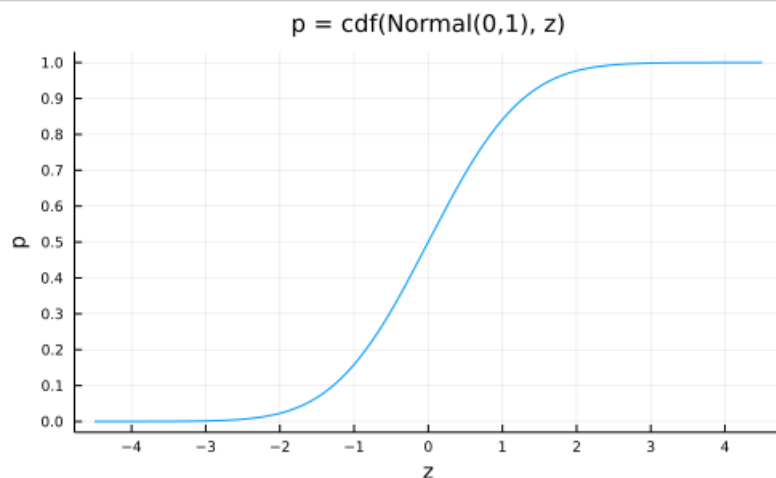
```
In [2]: 1 plot(x → cdf(Binomial(10, 0.3), x), -1.5, 11.5; label="")
2 plot!(xtick=-1:11, ytick=0:0.05:1)
3 plot!(xguide="x", yguide="p")
4 title!("p = cdf(Binomial(10, 0.3), x)")
```

Out[2]:



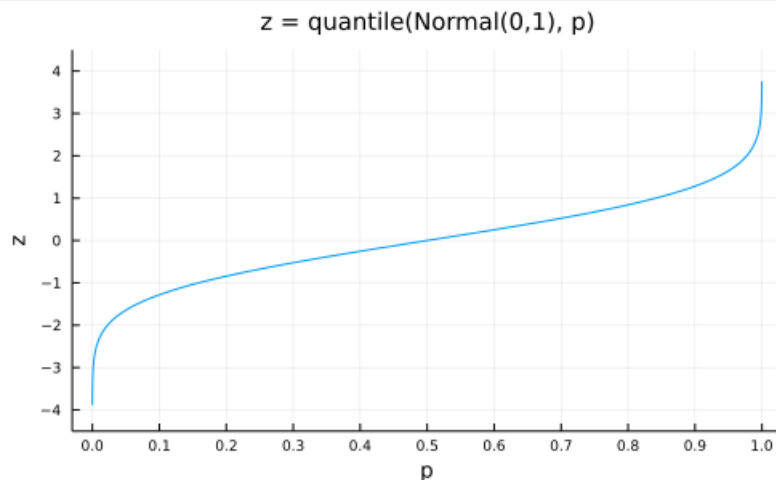
```
In [3]: 1 plot(z → cdf(Normal(0,1), z), -4.5, 4.5; label="")
2 plot!(xtick=-10:10, ytick=0:0.1:1)
3 plot!(xguide="z", yguide="p")
4 title!("p = cdf(Normal(0,1), z)")
```

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]:



4 補累積分布函数 ccdf と補分位点函数 cquantile

確率分布 D の 補累積分布函数 (complementary cumulative distribution function, **ccdf**) を

$$\text{ccdf}(D, x) = 1 - \text{cdf}(D, x) = P(X > x)$$

と定める. 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みなして扱って問題ない.

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

Out[5]: 1.9599639845400592

5 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.

5.1 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:

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

5.2 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値関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.

5.3 WilsonのP値と信頼区間の実装例とP値函数の解釈の仕方

```

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

```

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

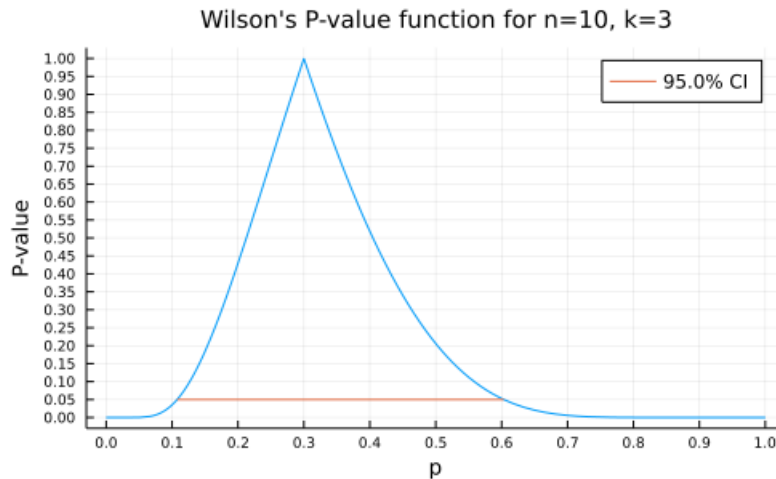
In [7]:

```
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, α) = (10, 3, 0.05) = (10, 3, 0.05)

ci = confint_wilson(k, n, α) > collect = [0.10779126740630075, 0.6032218525388553]

Out[7]:



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。

(1) 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。

一般にP値函数が最大になるパラメータ値をパラメータの **点推定値** とみなせる。

そのとき、点推定値はデータの数値と最も相性が良いモデルのパラメータ値であると解釈される。

(2) 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**。

(3) 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。

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。

信頼区間も「データの数値とモデル+パラメータ値の相性の良さ、整合性、両立性」で解釈すればよい。

閾値 α で「パラメータ値 θ とデータの数値 x が両立しなさすぎる」と判断することを、「データの数値 x によってパラメータ値 θ は **棄却** された」のように言うことがある。「棄却」という強い言葉を使っているが、実際にはデータの数値とモデルのパラメータ値の相性を見ているに過ぎない。

データの数値とモデルのパラメータ値の相性の良し悪しは、P値を計算することによって、データの数値が現実においてどのように生成されたかと無関係に判定される。

もしもP値を使って得たデータの数値とモデルのパラメータ値の相性の良し悪しに関する結果を、データの数値を生成した現実の法則にも適用したければ、各目的ごとに統計モデルが現実においてどのようにどれだけ妥当であるかについて別の議論が必要になる。

P値をデータの数値と統計モデルの矛盾の無さ(整合性(consistency), 両立性(compatibility), 相性の良さ)の指標とみなす立場は

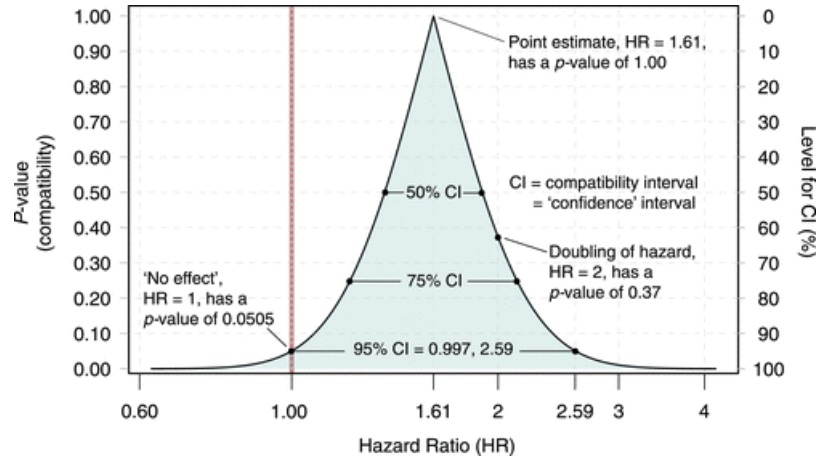
- [P値に関する米国統計学会による声明\(ASA声明\)](https://www.biometrics.gr.jp/news/all/ASA.pdf) (<https://www.biometrics.gr.jp/news/all/ASA.pdf>)

でも採用されており、P値を使った検定の結果だけを見て自信過剰にならないように注意を促している。

P値函数や信頼区間の解釈については次の最近の論文がわかりやすい:

- Amrhein-Greenland June 3, 2022 (<https://doi.org/10.1177%2F02683962221105904>)

この論文には次の図がある:



この図ではP値は **compatibility** (両立性)の指標とみなされており, confidence interval (信頼区間)は compatibility interval (両立性区間)に言い換えられている。

6 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。

6.1 WaldのP値

データ「 n 回中あたりは 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:

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

6.2 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。

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

すなわち,

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

Waldの信頼区間は区間 $[0, 1]$ をはみ出すことがある。

6.3 WaldのP値と信頼区間の実装例

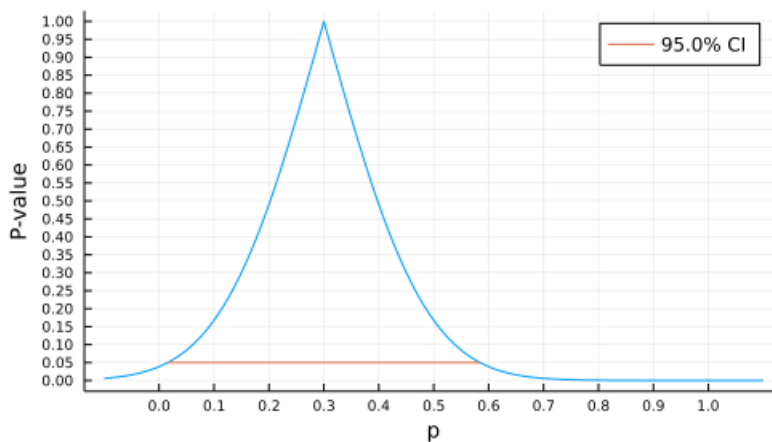
```
In [8]: 1 function pvalue_wald(k, n, p)
2        $\hat{p} = k/n$ 
3       sehat =  $\sqrt{(\hat{p}*(1 - \hat{p})/n)}$ 
4       2ccdf(Normal(0,1), abs( $\hat{p} - p$ )/sehat)
5   end
6
7   function confint_wald(k, n,  $\alpha$ )
8       z = cquantile(Normal(0,1),  $\alpha/2$ )
9        $\hat{p} =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

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

```
In [9]: 1 @show n, k,  $\alpha = 10, 3, 0.05$ 
2   @show ci = confint_wald(k, n,  $\alpha$ ) > 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( $\alpha, 2$ ); label="$ (100(1- $\alpha$ ))% CI")
8   title!("Wald's P-value function for n=$n, k=$k")
```

```
(n, k,  $\alpha$ ) = (10, 3, 0.05) = (10, 3, 0.05)
ci = confint_wald(k, n,  $\alpha$ ) > collect = [0.0159742349106739, 0.5840257650893261]
```

Out[9]: Wald's P-value function for n=10, k=3



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。

7 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に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。

7.1 Clopper-PearsonのP値

データ「 n 回中あたりは k 回」に関する「あたりが出る確率は 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：

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

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.

7.2 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.

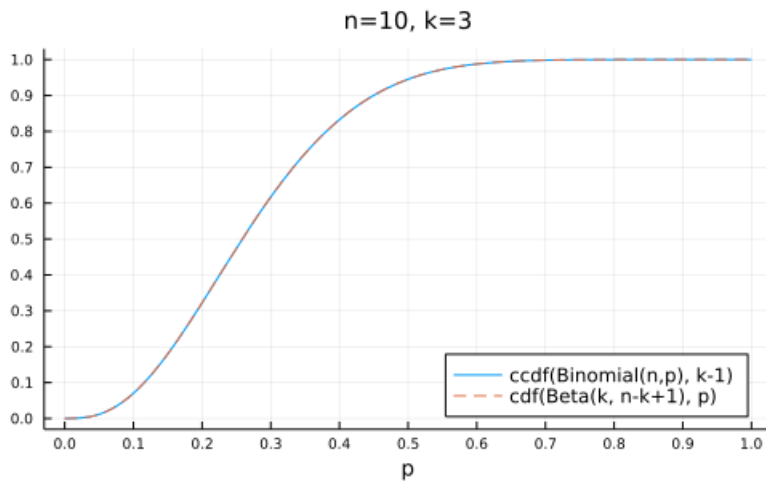
注意: 二項分布の累積分布関数と補累積分布関数は素朴に和で実装するよりも, 上の公式を用いて, ベータ分布の補累積分布関数と累積分布関数で計算するようにした方が効率がよい. ベータ分布の累積分布関数は regularized incomplete Beta function としてコンピュータの基本特殊関数ライブラリで効率的に実装されている.

```

In [10]: 1 n, k = 10, 3
2 plot(p → cdf(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]:

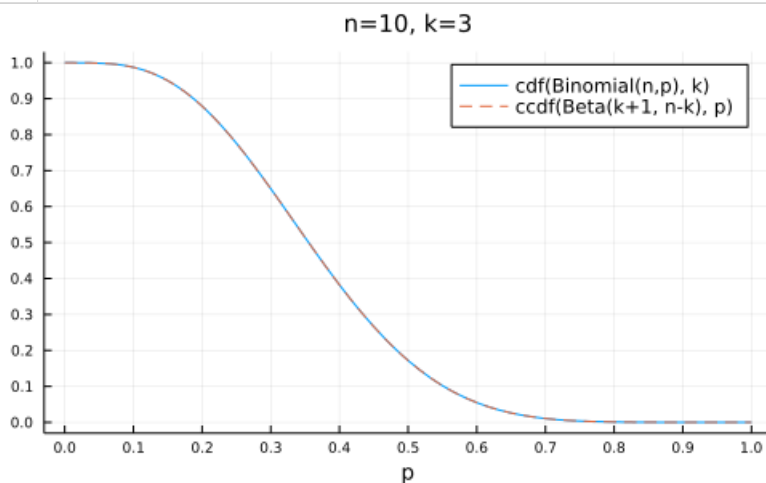


```

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]:



7.3 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。

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

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,

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

7.4 Clopper-PearsonのP値と信頼区間の実装例

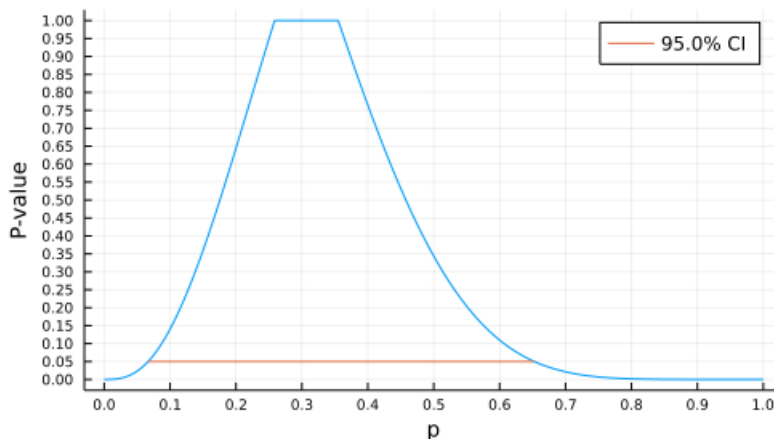
```
In [12]: 1 function pvalue_cp(k, n, p)
2         bin = Binomial(n, p)
3         min(1, 2cdf(bin, k), 2ccdf(bin, k-1))
4     end
5
6     function confint_cp(k, n, α)
7         quantile(Beta(k, n-k+1), α/2), cquantile(Beta(k+1, n-k), α/2)
8     end
```

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 P-value function for n=10, k=3



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。

8 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。

8.1 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。

二項分布モデルでのSterneの信頼区間にはシンプルな公式はないので直接的な数値計算で計算する必要がある。

8.2 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値函数の実装の仕方は

```
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)

          2.943461 seconds
          2.932301 seconds
          2.926920 seconds
```

Out[15]: 5.354898851685295e-12

注意: 上で、 $x \leq y$ ではなく、 $x \lesssim y$ を使っている理由は、数学的に $\text{pdf}(\text{bin}, j) = \text{pdf}(\text{bin}, k)$ が成立していても、浮動小数点数の計算では $\text{pdf}(\text{bin}, j) \neq \text{pdf}(\text{bin}, k)$ となってしまうかもしれないからである。そのような場合に \leq を使うとバグが発生することになる。

以下はより最適化されたSterneのP値函数の実装である。かなり複雑になっている。

```

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)

上のセルでのSterneのP値関数の実装結果は大幅に高速化されている!

```

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)

```

```

0.000046 seconds (8 allocations: 1.875 KiB)
0.000008 seconds (8 allocations: 1.875 KiB)
0.000008 seconds (8 allocations: 1.875 KiB)

```

Out[17]: 5.354898860502031e-12

この高速化によって, Sterneの信頼区間の計算も十分に高速化されることになる.

```

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)

```

```

0.157010 seconds (51.68 k allocations: 3.764 MiB, 98.07% compilation time)
0.002740 seconds (11.48 k allocations: 2.402 MiB)
0.002796 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")

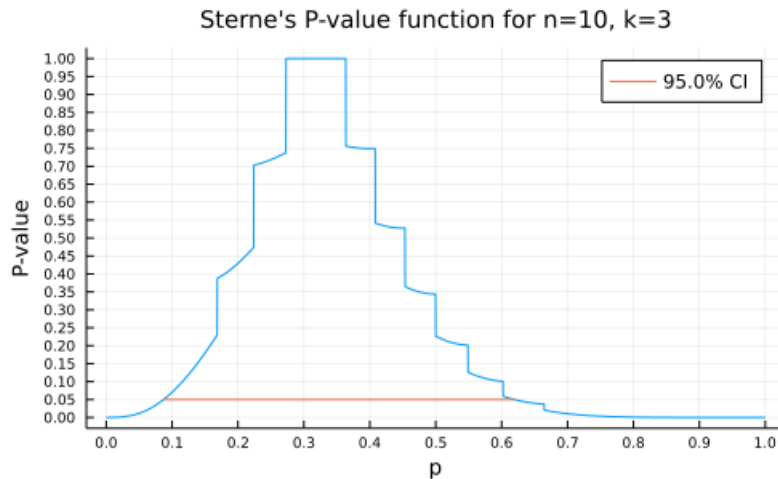
```

```

(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のP値関数はこのように不連続関数になる。

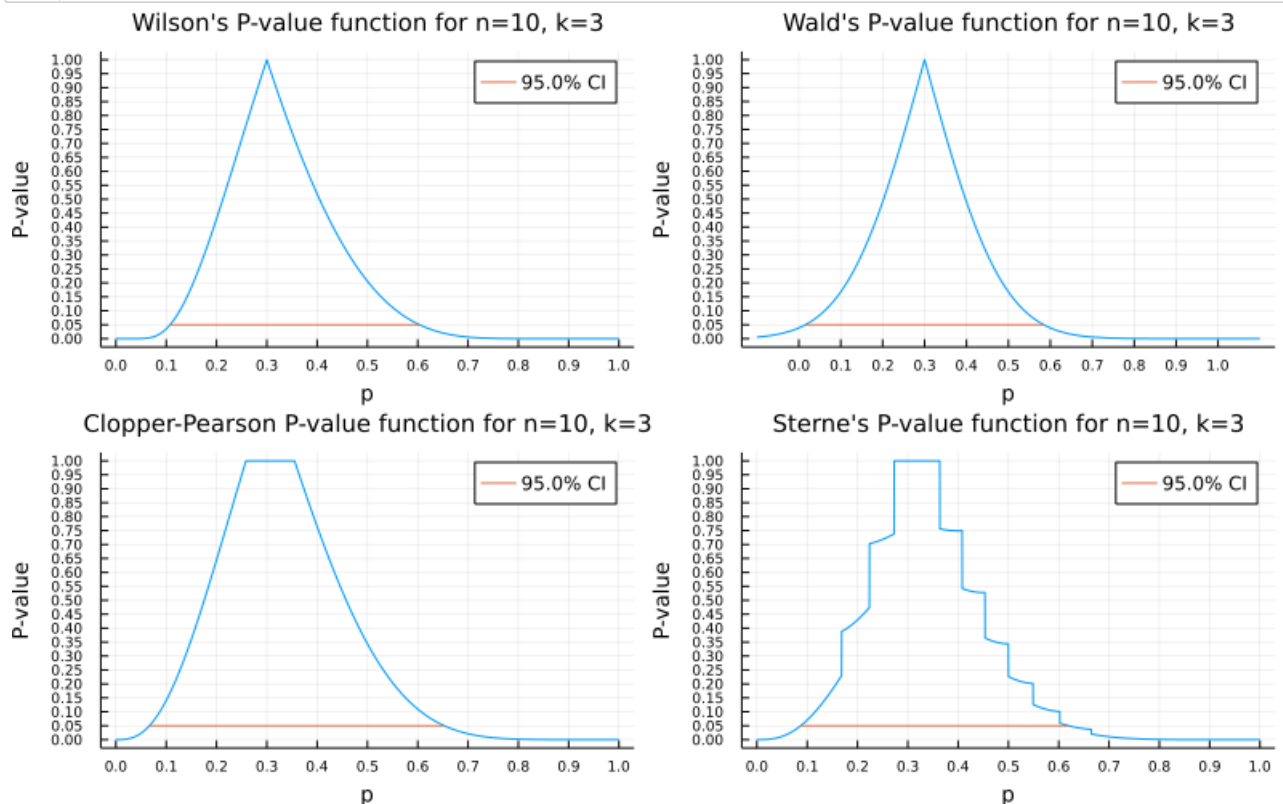
9 4種のP値関数と信頼区間を同時にプロット

```

In [20]: 1 plot(P_Wilson, P_Wald, P_CP, P_Sterne; layout=(2, 2), size=(800, 500))

```

Out[20]:



```

In [21]: 1 @show n, k,  $\alpha$  = 10, 3, 0.05
2 @show confint_wilson(k, n,  $\alpha$ )
3 @show confint_wald(k, n,  $\alpha$ )
4 @show confint_cp(k, n,  $\alpha$ )
5 @show confint_sterne(k, n,  $\alpha$ )
6
7 plot()
8 plot!(p  $\rightarrow$  pvalue_wilson(k, n, p), -0.1, 0.8; label="Wilson")
9 plot!(p  $\rightarrow$  pvalue_wald(k, n, p); label="Wald", ls=:dash)
10 plot!(p  $\rightarrow$  pvalue_cp(k, n, p); label="Clopper-Pearson", ls=:dashdot)
11 plot!(p  $\rightarrow$  pvalue_sterne(k, n, p); label="Sterne", ls=:dashdotdot)
12 plot!(xguide="p", yguide="P-value")
13 plot!(xtick=0:0.1:1, ytick=0:0.05:1)
14 title!("P-value functions for n=$n, k=$k")

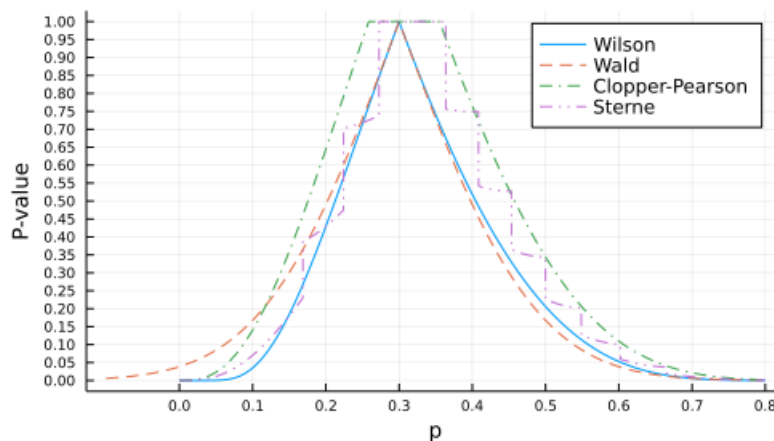
```

```

(n, k,  $\alpha$ ) = (10, 3, 0.05) = (10, 3, 0.05)
confint_wilson(k, n,  $\alpha$ ) = (0.10779126740630075, 0.6032218525388553)
confint_wald(k, n,  $\alpha$ ) = (0.0159742349106739, 0.5840257650893261)
confint_cp(k, n,  $\alpha$ ) = (0.06673951117773438, 0.6524528500599971)
confint_sterne(k, n,  $\alpha$ ) = (0.0872644339141502, 0.6194106589132863)

```

Out[21]: P-value functions for n=10, k=3

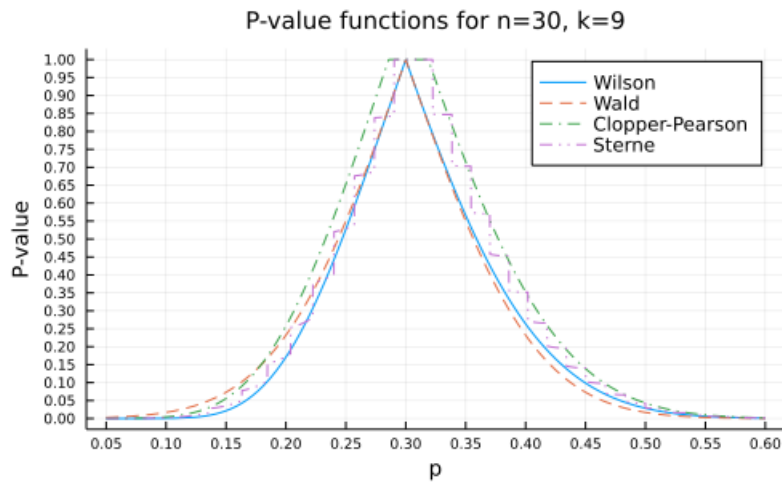


In [22]:

```
1 @show n, k,  $\alpha$  = 30, 9, 0.05
2 @show confint_wilson(k, n,  $\alpha$ )
3 @show confint_wald(k, n,  $\alpha$ )
4 @show confint_cp(k, n,  $\alpha$ )
5 @show confint_sterne(k, n,  $\alpha$ )
6
7 plot()
8 plot!(p  $\rightarrow$  pvalue_wilson(k, n, p), 0.05, 0.6; label="Wilson")
9 plot!(p  $\rightarrow$  pvalue_wald(k, n, p); label="Wald", ls=:dash)
10 plot!(p  $\rightarrow$  pvalue_cp(k, n, p); label="Clopper-Pearson", ls=:dashdot)
11 plot!(p  $\rightarrow$  pvalue_sterne(k, n, p); label="Sterne", ls=:dashdotdot)
12 plot!(xguide="p", yguide="P-value")
13 plot!(xtick=0:0.05:1, ytick=0:0.05:1)
14 title!("P-value functions for n=$n, k=$k")
```

```
(n, k,  $\alpha$ ) = (30, 9, 0.05) = (30, 9, 0.05)
confint_wilson(k, n,  $\alpha$ ) = (0.1666474826824377, 0.4787578745871499)
confint_wald(k, n,  $\alpha$ ) = (0.13601764806888816, 0.4639823519311118)
confint_cp(k, n,  $\alpha$ ) = (0.14734518475470101, 0.49395904146255365)
confint_sterne(k, n,  $\alpha$ ) = (0.16323067162201024, 0.4831441777228542)
```

Out[22]:

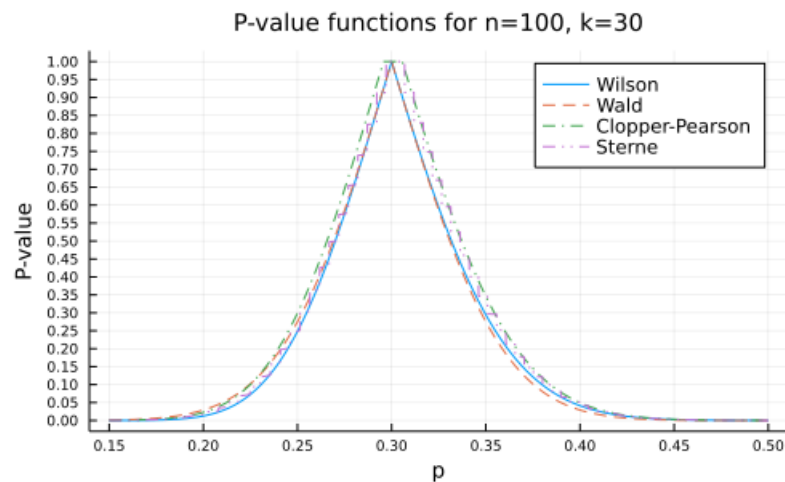


In [23]:

```
1 @show n, k, α = 100, 30, 0.05
2 @show confint_wilson(k, n, α)
3 @show confint_wald(k, n, α)
4 @show confint_cp(k, n, α)
5 @show confint_sterne(k, n, α)
6
7 plot()
8 plot!(p → pvalue_wilson(k, n, p), 0.15, 0.5; label="Wilson")
9 plot!(p → pvalue_wald(k, n, p); label="Wald", ls=:dash)
10 plot!(p → pvalue_cp(k, n, p); label="Clopper-Pearson", ls=:dashdot)
11 plot!(p → pvalue_sterne(k, n, p); label="Sterne", ls=:dashdotdot)
12 plot!(xguide="p", yguide="P-value")
13 plot!(xtick=0:0.05:1, ytick=0:0.05:1)
14 title!("P-value functions for n=$n, k=$k")
```

```
(n, k, α) = (100, 30, 0.05) = (100, 30, 0.05)
confint_wilson(k, n, α) = (0.21894885294932742, 0.39584854633346683)
confint_wald(k, n, α) = (0.21018316681457921, 0.38981683318542076)
confint_cp(k, n, α) = (0.21240642048953662, 0.39981467617980404)
confint_sterne(k, n, α) = (0.21419730401454767, 0.3996107372232876)
```

Out[23]:



n を大きくすると、4種のP値関数はほぼ一致するようになる。

10 ベイズ統計での信用区間に対応するP値関数

P値関数と区間推定の表裏一体性については、私のツイログの検索を参照せよ:

- [P値 信頼区間 表裏一体 \(https://twilog.org/genkuroki/search?word=P%E5%80%A4%E2%04%BF%A1%E9%A0%BC%E5%8C%BA%E9%96%93%E8%A1%A8%E8%A3%8F%E4%B8%\)](https://twilog.org/genkuroki/search?word=P%E5%80%A4%E2%04%BF%A1%E9%A0%BC%E5%8C%BA%E9%96%93%E8%A1%A8%E8%A3%8F%E4%B8%)

区間推定の方法には、通常の信頼区間以外にも、ベイズ統計の **信用区間** (credible interval, 確信区間)の方法もある。

ゆえに、P値関数と区間推定の表裏一体性より、ベイズ統計の方法を使っても、P値関数を構成することができる。

ベイズ版のP値関数と通常のP値関数を比較したい。

10.1 二項分布モデルのHDI版信用区間

簡単のため共役事前分布の場合のみを扱う。

二項分布モデルのベイズ統計では以下の手続きでパラメータ p の事後分布が得られる。

- (1) パラメータ p の事前分布 $\text{prior} = \text{Beta}(a, b)$ ($a, b > 0$)を任意に用意する。
- (2) 「 n 回中当たりが k 回」の型のデータを取得する。
- (3) パラメータ p の事後分布 $\text{posterior} = \text{Beta}(k + a, n - k + b)$ が得られる。

この手続きの中身については任意のベイズ統計に関する教科書を参照せよ。

このとき、パラメータ p に関する $100(1 - \alpha)\%$ 信用区間が次のように定義される。

- (4) 事後分布を

$[0, 1]$ に含まれる区間 $[p_L, p_U]$ で事後分布で測ったその区間に含まれる確率が

$$\text{cdf}(\text{posterior}, p_U) - \text{cdf}(\text{posterior}, p_L) = 1 - \alpha$$

となるものの中で長さ $p_U - p_L$ が最小のものを highest density interval (HDI)として定義された信用度 $1 - \alpha$ の **信用区間** と呼び、次のように表す:

$$\text{credint}_{\text{HDI}}(k|n, p, a, b) = [p_L, p_U].$$

ベータ分布の確率密度関数を

$$\text{pdf}(\text{Beta}(a, b), t) = \frac{1}{B(a, b)} t^{a-1} (1-t)^{b-1}$$

と書くと, HDI版信用区間 $[p_L, p_U]$ について,

$$\text{pdf}(\text{Beta}(k+a, n-k+b), p_L) = \text{pdf}(\text{Beta}(k+a, n-k+b), p_U)$$

が成立しており, この条件と上の確率に関する条件でHDI版信用区間は一意に特徴付けられる.

注意: 実際の実装では,

$$p_L = \text{quantile}(\text{posterior}, t), \quad p_U = \text{quantile}(\text{posterior}, t + (1 - \alpha))$$

のあいだの長さ,

$$f(t) = p_U - p_L = \text{quantile}(\text{posterior}, t + (1 - \alpha)) - \text{quantile}(\text{posterior}, t)$$

を最小化する t を求め, それに対応する区間 $[p_L, p_U]$ としてHDI版信用区間を計算する.

10.2 HDI版信用区間に対応するP値関数

HDI版信用区間に対応するP値関数 $\text{pvalue}_{\text{HDI}}(k|n, p, a, b)$ は以下のように定義される.

与えられた値 p に対して, t に関する方程式

$$\text{pdf}(\text{Beta}(k+a, n-k+b), t) = \text{pdf}(\text{Beta}(k+a, n-k+b), p)$$

の2つの解を $p_L \leq p_U$ と書き,

$$\text{pvalue}_{\text{HDI}}(k|n, p, a, b) = \text{cdf}(\text{Beta}(k+a, n-k+b), p_L) + \text{ccdf}(\text{Beta}(k+a, n-k+b), p_U).$$

これを, **ベイズ版P値関数** (Bayesian P-value function)のHDI版と呼ぶ.

10.3 HDI版の信用区間とそれに対応するベイズ版P値関数の実装例

```

In [24]: 1 function hdi(dist::ContinuousUnivariateDistribution, α = 0.05; alg = Brent())
2         f(p) = quantile(dist, p + (1 - α)) - quantile(dist, p)
3         o = optimize(f, 0, α, alg)
4         p = o.minimizer
5         quantile.(dist, (p, p + (1 - α)))
6     end
7
8     function pvalue_hdi(dist::ContinuousUnivariateDistribution, x₀; xlim = extrema(dist))
9         p₀ = pdf(dist, x₀)
10        m = mode(dist)
11        f(x) = pdf(dist, x) - p₀
12        if x₀ == m
13            1.0
14        elseif x₀ > m
15            x₁ = find_zero(f, (xlim[begin], m))
16            cdf(dist, x₁) + ccdf(dist, x₀)
17        else
18            x₁ = find_zero(f, (m, xlim[end]))
19            cdf(dist, x₀) + ccdf(dist, x₁)
20        end
21    end
22
23    function pvalue_hdi(k, n, p; a=1, b=1)
24        posterior = Beta(k+a, n-k+b)
25        pvalue_hdi(posterior, p)
26    end
27
28    function credint_hdi(k, n, α; a=1, b=1)
29        posterior = Beta(k+a, n-k+b)
30        hdi(posterior, α)
31    end

```

Out[24]: credint_hdi (generic function with 1 method)

```

In [25]: 1 @show n, k, α = 10, 3, 0.05
2 @show a, b = 1, 1
3 @show ci = credint_hdi(k, n, α; a, b) ▷ collect
4
5 P_HDI = plot(p → pvalue_hdi(k, n, p; a, b), 0, 1; label="")
6 plot!(xguide="p", yguide="P-value")
7 plot!(xtick=0:0.1:1, ytick=0:0.05:1)
8 plot!(ci, fill(α,2); label="$ (100(1-α))% CI (HDI)")
9 title!("Bayesian P-value function for n=$n, k=$k, a=$a, b=$b")

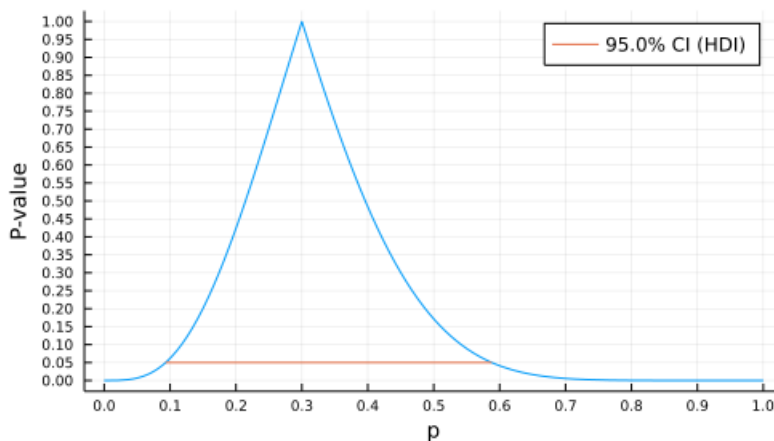
```

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

(a, b) = (1, 1) = (1, 1)

ci = credint_hdi(k, n, α; a, b) ▷ collect = [0.09337233320249291, 0.5879525593275807]

Out[25]: Bayesian P-value function for n=10, k=3, a=1, b=1



このグラフも通常のP値関数と信頼区間と同様に、

- データの数値とパラメータ値の相性の良さ(両立性, compatibility)

を表すグラフだと解釈可能である. この解釈は次の事後分布のグラフにもそのまま一般化される.

In [26]:

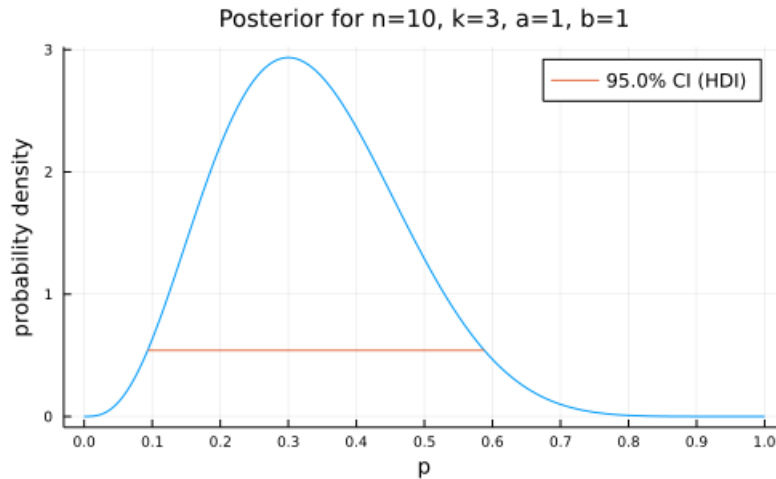
```
1 @show n, k, α = 10, 3, 0.05
2 @show a, b = 1, 1
3 @show ci = credint_hdi(k, n, α; a, b) ▷ collect
4
5 f(p) = pdf(Beta(k+a, n-k+b), p)
6 Q_HDI = plot(f, 0, 1; label="")
7 plot!(xguide="p", yguide="probability density")
8 plot!(xtick=0:0.1:1)
9 plot!(ci, fill(f(ci[end]), 2); label="$ (100(1-α))% CI (HDI)")
10 title!("Posterior for n=$n, k=$k, a=$a, b=$b")
```

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

(a, b) = (1, 1) = (1, 1)

ci = credint_hdi(k, n, α; a, b) ▷ collect = [0.09337233320249291, 0.5879525593275807]

Out[26]:



10.4 WilsonのP値関数とベイズ版P値関数のHDI版の比較

この節の目的は通常のP値関数とベイズ版P値関数を比較することであった。

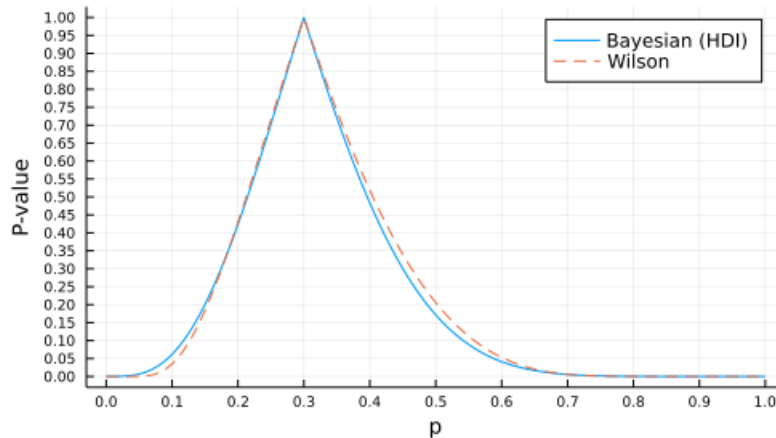
以下では、通常のP値関数の代表例としてWilsonのP値関数を選び、ベイズ版P値関数の代表例として一様事前分布 $\text{Beta}(1, 1)$ から得られるベイズ版P値関数を選んで比較することにする。

上のベイズ版P値関数をWilsonのP値関数と重ねてプロットしてみよう。

```
In [27]: 1 @show n, k,  $\alpha$  = 10, 3, 0.05
2 @show a, b = 1, 1
3
4 plot(p  $\rightarrow$  pvalue_hdi(k, n, p; a, b), 0, 1; label="Bayesian (HDI)")
5 plot!(p  $\rightarrow$  pvalue_wilson(k, n, p); label="Wilson", ls=:dash)
6 plot!(xguide="p", yguide="P-value")
7 plot!(xtick=0:0.1:1, ytick=0:0.05:1)
8 title!("P-value functions for n=$n, k=$k, a=$a, b=$b")
```

```
(n, k,  $\alpha$ ) = (10, 3, 0.05) = (10, 3, 0.05)
(a, b) = (1, 1) = (1, 1)
```

Out[27]: P-value functions for n=10, k=3, a=1, b=1



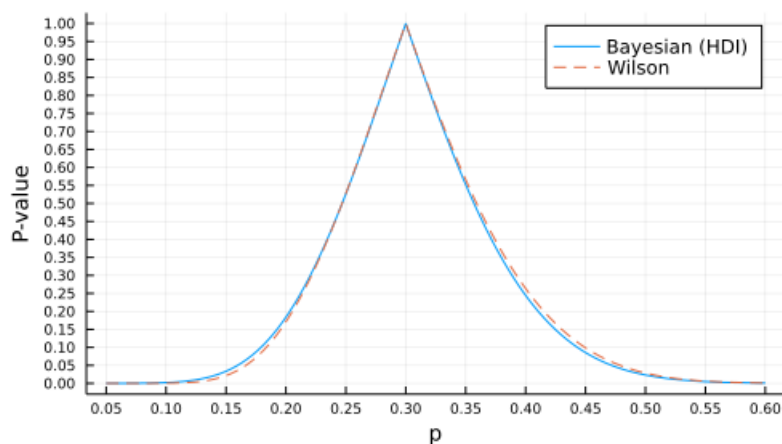
ベイズ版P値関数が通常のWilsonのP値関数によく一致していることが確認された。

n, k を大きくすると違いはさらに小さくなる。

```
In [28]: 1 @show n, k,  $\alpha$  = 30, 9, 0.05
2 @show a, b = 1, 1
3
4 plot(p  $\rightarrow$  pvalue_hdi(k, n, p; a, b), 0.05, 0.6; label="Bayesian (HDI)")
5 plot!(p  $\rightarrow$  pvalue_wilson(k, n, p); label="Wilson", ls=:dash)
6 plot!(xguide="p", yguide="P-value")
7 plot!(xtick=0:0.05:1, ytick=0:0.05:1)
8 title!("P-value functions for n=$n, k=$k, a=$a, b=$b")
```

```
(n, k,  $\alpha$ ) = (30, 9, 0.05) = (30, 9, 0.05)
(a, b) = (1, 1) = (1, 1)
```

Out[28]: P-value functions for n=30, k=9, a=1, b=1

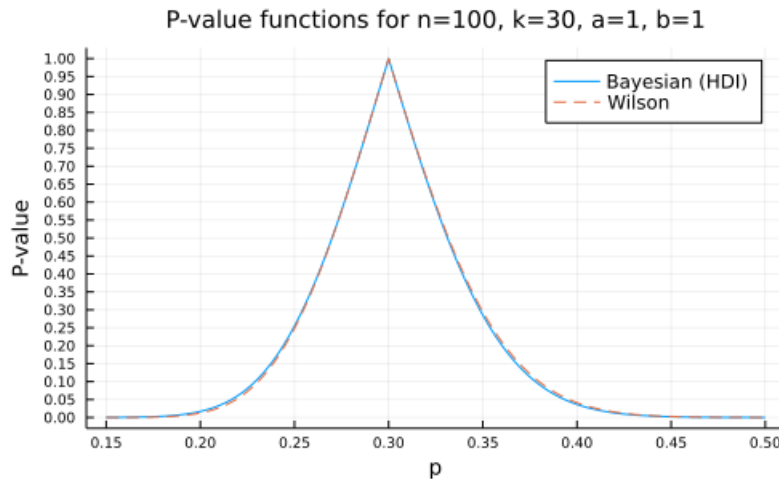


In [29]:

```
1 @show n, k, α = 100, 30, 0.05
2 @show a, b = 1, 1
3
4 plot(p → pvalue_hdi(k, n, p; a, b), 0.15, 0.5; label="Bayesian (HDI)")
5 plot!(p → pvalue_wilson(k, n, p); label="Wilson", ls=:dash)
6 plot!(xguide="p", yguide="P-value")
7 plot!(xtick=0:0.05:1, ytick=0:0.05:1)
8 title!("P-value functions for n=$n, k=$k, a=$a, b=$b")
```

```
(n, k, α) = (100, 30, 0.05) = (100, 30, 0.05)
(a, b) = (1, 1) = (1, 1)
```

Out[29]:



$n = 100, k = 30$ ではほぼびったり重なっている!

10.5 ETI版の信用区間とそれに対応するP値関数

posterior は事後分布を表すことにする:

$$\text{posterior} = \text{Beta}(k + a, n - k + b).$$

HDI版ではなく, equal tailed interval (ETI)版の信用区間も定義しておこう:

$$\text{credint}_{\text{ETI}}(k|n, p, a, b) = [\text{quantile}(\text{posterior}, \alpha/2), \text{cquantile}(\text{posterior}, \alpha/2),].$$

これに対応するP値関数は次になる:

$$\text{pvalue}_{\text{ETI}}(k|n, p, a, b) = \min \begin{pmatrix} 1 \\ 2\text{cdf}(\text{posterior}, p) \\ 2\text{ccdf}(\text{posterior}, p) \end{pmatrix}.$$

10.6 ETI版の信用区間とそれに対応するベイズ版P値関数の実装例

```
In [30]: 1 function pvalue_eti(k, n, p; a=1/2, b=1/2)
2         posterior = Beta(k+a, n-k+a)
3         min(1, 2cdf(posterior, p), 2ccdf(posterior, p))
4     end
5
6     function credint_eti(k, n, α; a=1/2, b=1/2)
7         posterior = Beta(k+a, n-k+a)
8         quantile(posterior, α/2), cquantile(posterior, α/2)
9     end
```

Out[30]: credint_eti (generic function with 1 method)

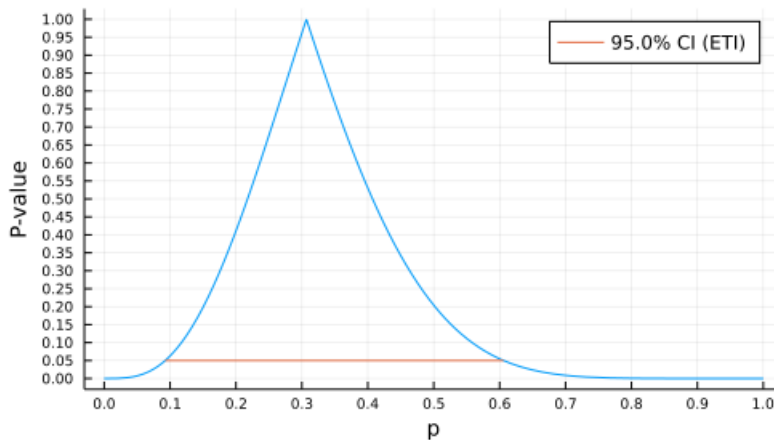
```
In [31]: 1 @show n, k, α = 10, 3, 0.05
2 @show a, b = 1/2, 1/2
3 @show ci = credint_eti(k, n, α; a, b) ▷ collect
4
5 P_ETI = plot(p → pvalue_eti(k, n, p; a, b), 0, 1; label="")
6 plot!(xguide="p", yguide="P-value")
7 plot!(xtick=0:0.1:1, ytick=0:0.05:1)
8 plot!(ci, fill(α,2); label="$ (100(1-α))% CI (ETI)")
9 title!("Bayesian P-value function for n=$n, k=$k, a=$a, b=$b")
```

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

(a, b) = (1 / 2, 1 / 2) = (0.5, 0.5)

ci = credint_eti(k, n, α; a, b) ▷ collect = [0.09269459393815314, 0.6058183181486713]

Out[31]: Bayesian P-value function for n=10, k=3, a=0.5, b=0.5



10.7 WilsonのP値関数とベイズ版P値関数のETI版の比較

ベイズ版P値関数のETI版は、Jeffreys事前分布 $\text{Beta}(1/2, 1/2)$ の場合にWilsonのP値とよく一致することを確認しよう。

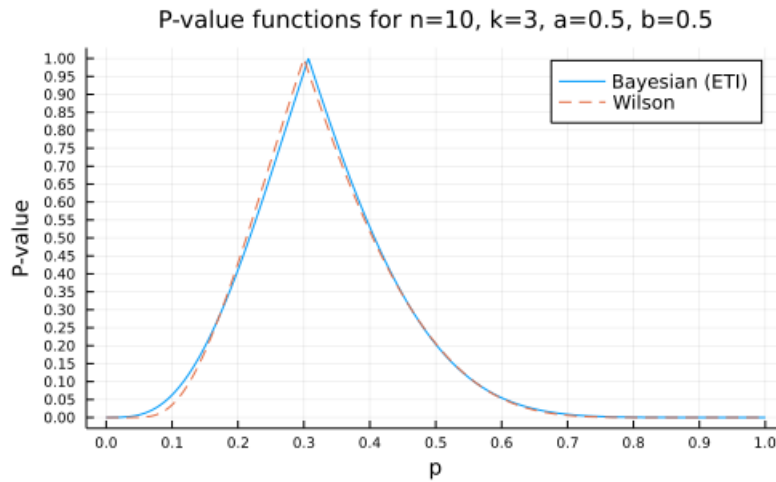
Jeffreys事前分布は一様事前分布と共に **無情報事前分布** の代表例である。

In [32]:

```
1 @show n, k,  $\alpha$  = 10, 3, 0.05
2 @show a, b = 1/2, 1/2
3
4 plot(p → pvalue_eti(k, n, p; a, b), 0, 1; label="Bayesian (ETI)")
5 plot!(p → pvalue_wilson(k, n, p); label="Wilson", ls=:dash)
6 plot!(xguide="p", yguide="P-value")
7 plot!(xtick=0:0.1:1, ytick=0:0.05:1)
8 title!("P-value functions for n=$n, k=$k, a=$a, b=$b")
```

$(n, k, \alpha) = (10, 3, 0.05) = (10, 3, 0.05)$
 $(a, b) = (1/2, 1/2) = (0.5, 0.5)$

Out[32]:

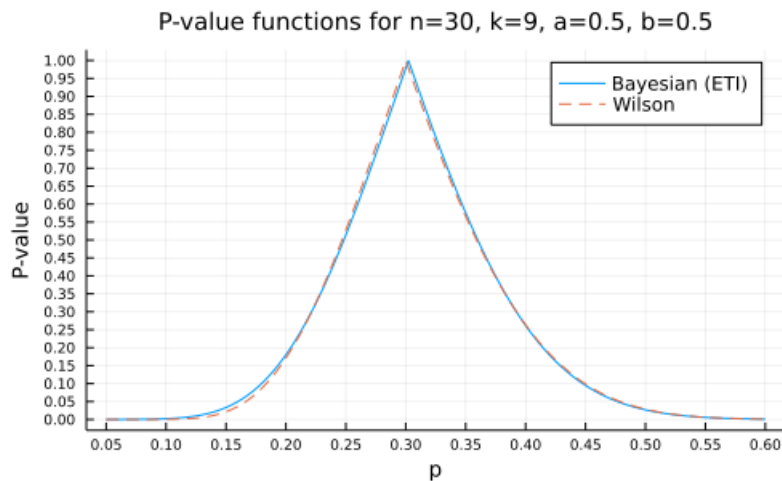


In [33]:

```
1 @show n, k,  $\alpha$  = 30, 9, 0.05
2 @show a, b = 1/2, 1/2
3
4 plot(p → pvalue_eti(k, n, p; a, b), 0.05, 0.6; label="Bayesian (ETI)")
5 plot!(p → pvalue_wilson(k, n, p); label="Wilson", ls=:dash)
6 plot!(xguide="p", yguide="P-value")
7 plot!(xtick=0:0.05:1, ytick=0:0.05:1)
8 title!("P-value functions for n=$n, k=$k, a=$a, b=$b")
```

$(n, k, \alpha) = (30, 9, 0.05) = (30, 9, 0.05)$
 $(a, b) = (1/2, 1/2) = (0.5, 0.5)$

Out[33]:

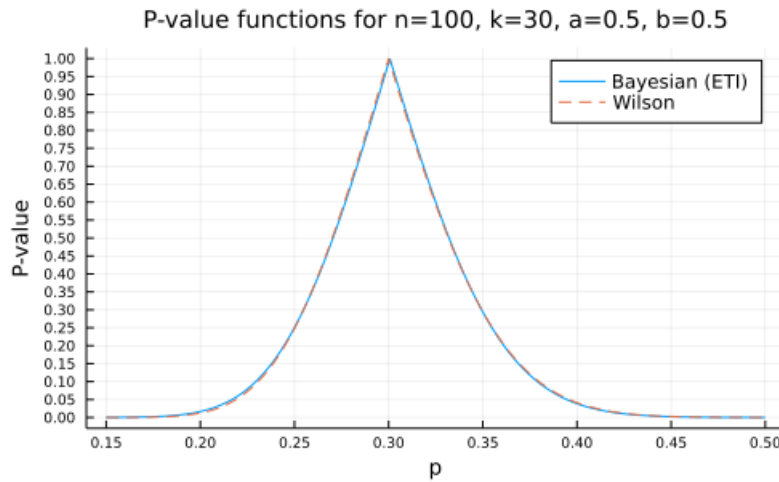


In [34]:

```
1 @show n, k,  $\alpha$  = 100, 30, 0.05
2 @show a, b = 1/2, 1/2
3
4 plot(p  $\rightarrow$  pvalue_eti(k, n, p; a, b), 0.15, 0.5; label="Bayesian (ETI)")
5 plot!(p  $\rightarrow$  pvalue_wilson(k, n, p); label="Wilson", ls=:dash)
6 plot!(xguide="p", yguide="P-value")
7 plot!(xtick=0:0.05:1, ytick=0:0.05:1)
8 title!("P-value functions for n=$n, k=$k, a=$a, b=$b")
```

```
(n, k,  $\alpha$ ) = (100, 30, 0.05) = (100, 30, 0.05)
(a, b) = (1 / 2, 1 / 2) = (0.5, 0.5)
```

Out[34]:



$n = 100, k = 30$ ではほぼびったり重なっている!

ベイズ版P値関数のETI版が与える点推定の結果を最尤推定値によりよく一致させるためには、事前分布として、Jeffreys事前分布 $\text{Beta}(1/2, 1/2)$ ではなく、事前分布 $\text{Beta}(1/3, 1/3)$ を採用すればよい。

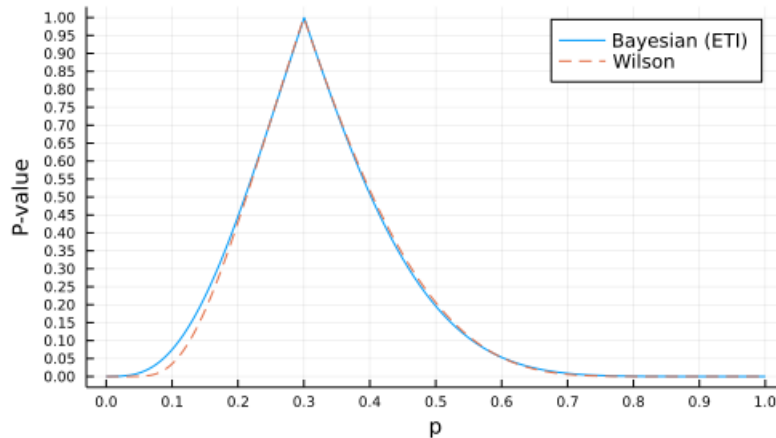
その理由はベータ分布 $\text{Beta}(a, b)$ の中央値が $(a - 1/3)/(a + b - 2/3)$ で近似されるからである。ベータ分布についてはウィキペディアを参照せよ:

- [ベータ分布 - Wikipedia \(https://ja.wikipedia.org/wiki/%E3%83%99%E3%83%BC%E3%82%BF%E5%88%86%E5%B8%83\)](https://ja.wikipedia.org/wiki/%E3%83%99%E3%83%BC%E3%82%BF%E5%88%86%E5%B8%83)

```
In [35]: 1 @show n, k,  $\alpha$  = 10, 3, 0.05
2 @show a, b = 1//3, 1//3
3
4 plot(p  $\rightarrow$  pvalue_eti(k, n, p; a, b), 0, 1; label="Bayesian (ETI)")
5 plot!(p  $\rightarrow$  pvalue_wilson(k, n, p); label="Wilson", ls=:dash)
6 plot!(xguide="p", yguide="P-value")
7 plot!(xtick=0:0.1:1, ytick=0:0.05:1)
8 title!("P-value functions for n=$n, k=$k, a=$a, b=$b")
```

```
(n, k,  $\alpha$ ) = (10, 3, 0.05) = (10, 3, 0.05)
(a, b) = (1 // 3, 1 // 3) = (1//3, 1//3)
```

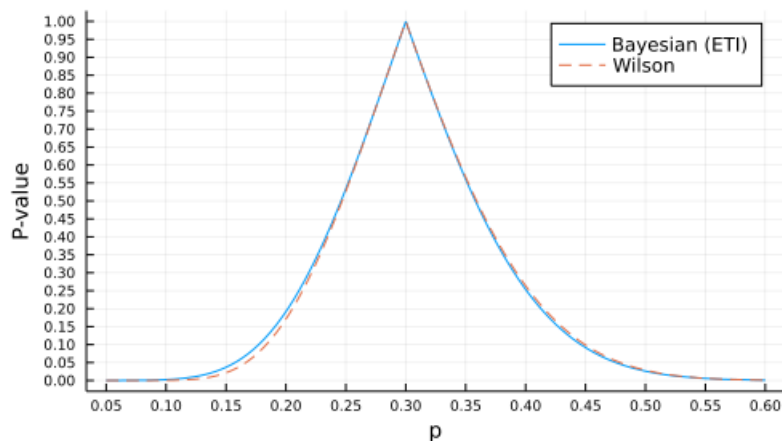
Out[35]: P-value functions for n=10, k=3, a=1//3, b=1//3



```
In [36]: 1 @show n, k,  $\alpha$  = 30, 9, 0.05
2 @show a, b = 1//3, 1//3
3
4 plot(p  $\rightarrow$  pvalue_eti(k, n, p; a, b), 0.05, 0.6; label="Bayesian (ETI)")
5 plot!(p  $\rightarrow$  pvalue_wilson(k, n, p); label="Wilson", ls=:dash)
6 plot!(xguide="p", yguide="P-value")
7 plot!(xtick=0:0.05:1, ytick=0:0.05:1)
8 title!("P-value functions for n=$n, k=$k, a=$a, b=$b")
```

```
(n, k,  $\alpha$ ) = (30, 9, 0.05) = (30, 9, 0.05)
(a, b) = (1 // 3, 1 // 3) = (1//3, 1//3)
```

Out[36]: P-value functions for n=30, k=9, a=1//3, b=1//3

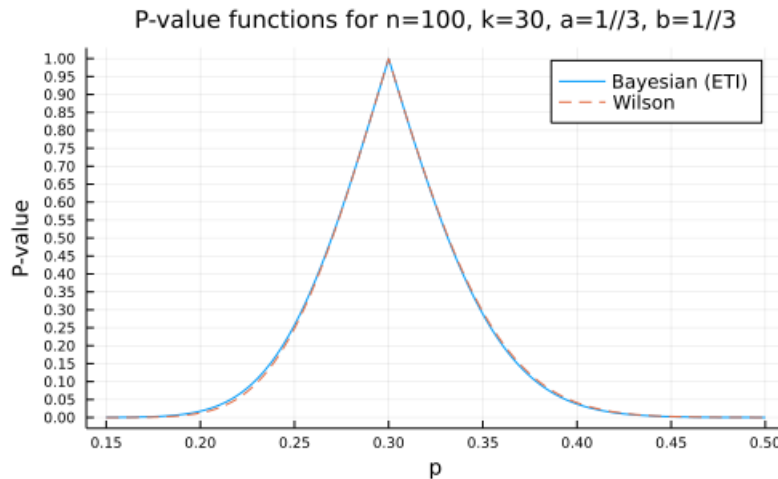


In [37]:

```
1 @show n, k, α = 100, 30, 0.05
2 @show a, b = 1//3, 1//3
3
4 plot(p → pvalue_eti(k, n, p; a, b), 0.15, 0.5; label="Bayesian (ETI)")
5 plot!(p → pvalue_wilson(k, n, p); label="Wilson", ls=:dash)
6 plot!(xguide="p", yguide="P-value")
7 plot!(xtick=0:0.05:1, ytick=0:0.05:1)
8 title!("P-value functions for n=$n, k=$k, a=$a, b=$b")
```

```
(n, k, α) = (100, 30, 0.05) = (100, 30, 0.05)
(a, b) = (1 // 3, 1 // 3) = (1//3, 1//3)
```

Out[37]:



$n = 100, k = 30$ ではほぼぴったり重なっている!

10.8 「主義の押し付け」をやめよう

所謂「ベイズ主義」にこだわったベイズ統計の解説では、通常の信頼区間とベイズ版の信用区間は解釈が異なり、まったく違うものであるかのように説明されている。

しかし、上で確認したように、二項分布モデルのような統計学入門の教科書でよく説明されているシンプルなモデルの場合には、ベイズ版の信用区間は通常の信頼区間と実用的には区別する必要がないほどよく一致する。

そのことは、対応するP値函数がよく一致することと数学的には同値である。

むしろ、通常の4種のP値函数のあいだの違いの方が、ベイズ版P値函数とWilsonのP値函数の違いよりも大きいと言える。

この事実は、ベイズ版信用区間が通常の信頼区間の近似計算法だとみなせ、逆に、通常の信頼区間は無情報事前分布のベイズ版信用区間の近似計算法だとみなせることを意味している。

実践的には「同じもの」とみなしても害がないのに、主義の違いによって異なる解釈を強制しようとするのは、「かけ算の意味によって、 3×4 は正解だが、 4×3 は誤りになる」と教えている困った人達と同じ立場に立つことになる。

そういうことはもうやめた方がよいだろう。

「シンプルなモデルにおいては無情報事前分布のベイズ統計と通常のP値を使う方法は実践的には同等だとみなされる」という事実を理解して、使える道具の幅を気軽に広げた方が生産的だと思われる。

それを妨げているのは「主義の押し付け」であり、極めて有害だと思われる。

注意: 以上では統計モデルが単独の二項分布の場合に通常のP値函数とベイズ版P値函数の近似的な一致を確認した。同様のことは、2つの二項分布モデルにおける「オッズ比」「リスク比」「比率の差」に関するP値函数についても成立していることを比較的容易に確認できる。

注意: 1標本のt検定の信用区間と予測区間は, 正規分布モデルで事前分布を $d\mu d(\sigma^2)/\sigma^2$ のimproper事前分布とした場合のベイズ統計によって, exactに(誤差無しに)再現される. ゆえに, その事前分布を近似する事前分布について, 正規分布モデルのベイズ統計は1標本のt検定を近似することになる.

注意: 通常の線形回帰の信用区間と予測区間も, 事前分布を回帰係数 β と分散の対数 $\log \sigma^2$ について平坦なimproper事前分布とした場合のベイズ版線形回帰によってexactに(誤算無しに)再現される. ゆえに, その事前分布を近似する事前分布について, ベイズ版線形回帰は通常の線形回帰を近似することになる.

注意: このように, ベイズ統計では統計モデルの構築において事前分布の分だけ自由度が増えており, シンプルなモデルでは事前分布を特別な場合に取りれば, 通常のP値を使う統計分析法をベイズ統計の方法によって再現できるので, シンプルなモデルにおいてベイズ統計は通常のP値を使う方法の上位互換だと考えられる.

In []:

1	
---	--