

比率のP値函数達

- ・ 黒木玄
- ・ 2022-09-08～2022-09-12
- ・ 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>)

このノートでは事象 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、ベータ分布などの確率分布、P値と信頼区間。

終わりの方で必要になる追加の予備知識: 二項分布モデルのベイズ統計。

想定読者:

- ・ すでに統計学入門を学習済みだが、P値や信頼区間について明瞭なイメージをまだ持てていない人達。
- ・ 比率に関するP値と信頼区間の複数の定義について正確な知識を得たい人達。
- ・ P値と信頼区間のあいだの表裏一体性について理解したい人達。
- ・ P値と信頼区間にに関する **データの数値と統計モデルのcompatibility** という解釈について知りたい人達。
- ・ P値と使う統計学とベイズ統計が水と油ではないことを具体的計算で納得したい人達。

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](#)
 - [5.4 P値函数のcompatibilityの指標としての解釈の仕方](#)
- ▼ [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](#)
 - [6.4 WaldのP値と信頼区間の構成で使った正規分布近似の粗さの視覚化](#)
- ▼ [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 二項分布モデルのETI版の信用区間とそれに対応するP値](#)
 - [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 ETI版信用区間とClopper-Pearsonの信頼区間の類似と比較](#)
 - [10.9 主義の押し付けをやめよう](#)
- [11 被覆確率の比較](#)
- [12 P値の累積分布函数の比較](#)

```
In [1]: 1 using Printf
2
3 using Distributions
4 using StatsPlots
5 default(fmt=:png, size=(500, 300),
6         titlefontsize=10, tickfontsize=6, guidefontsize=9,
7         plot_titlefontsize=10)
8
9 # Sterneの信頼区間の定義で使用
10 using Roots
11 using StatsFuns: logit
12
13 # highest density interval (HDI) の構成で使用
14 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 $pvalue(k|n, p = p_0)$ のように書くこととする。

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：

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

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

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

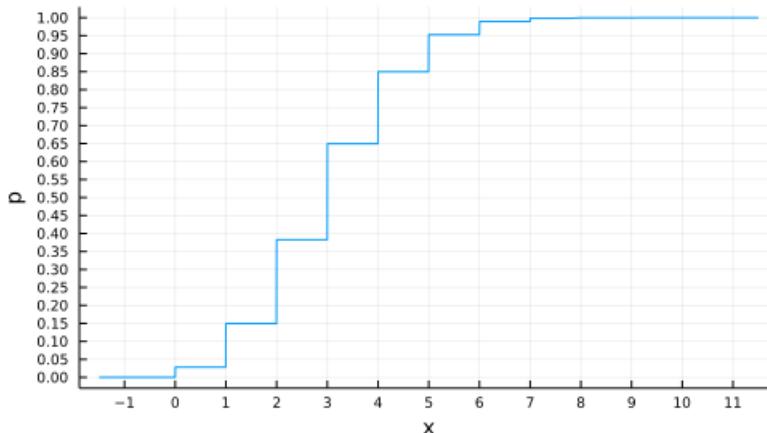
と表し、分布 D の **分位点函数** (quantile function)と呼ぶ。

例えば、正規分布の累積分布函数は狭義単調增加函数なので、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。

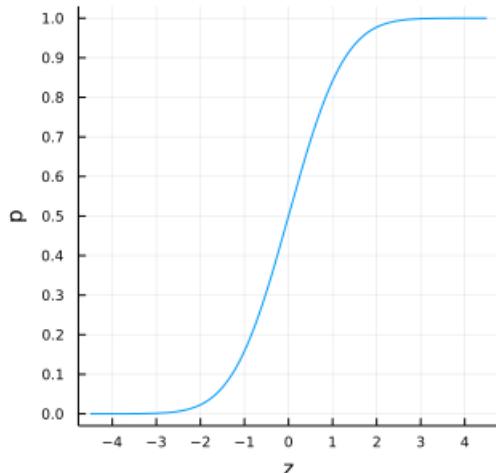
```
In [2]: 1 plot(x → cdf(Binomial(10, 0.3), x), -1.5, 11.5; label="")
2 plot!(xtick=-1:11, ytick=0:0.05:1)
3 plot!(xguide="x", yguide="p")
4 title!("p = cdf(Binomial(10, 0.3), x)")
```

Out[2]: $p = \text{cdf}(\text{Binomial}(10, 0.3), x)$

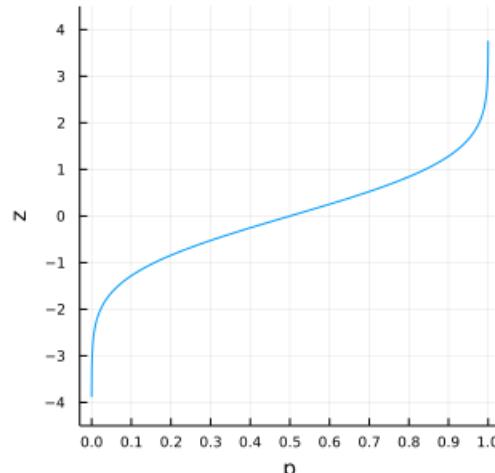


```
In [3]: 1 P1 = plot(z → cdf(Normal(0,1), z), -4.5, 4.5; label="")
2 plot!(xtick=-10:10, ytick=0:0.1:1)
3 plot!(xguide="z", yguide="p")
4 title!("p = cdf(Normal(0,1), z)")
5
6 P2 = plot(p → quantile(Normal(0,1), p), 0, 1; label="")
7 plot!(ylim=(-4.5, 4.5))
8 plot!(ytick=-10:10, xtick=0:0.1:1)
9 plot!(yguide="z", xguide="p")
10 title!("z = quantile(Normal(0,1), p)")
11
12 plot(P1, P2; size=(640, 320))
```

Out[3]: $p = \text{cdf}(\text{Normal}(0,1), z)$



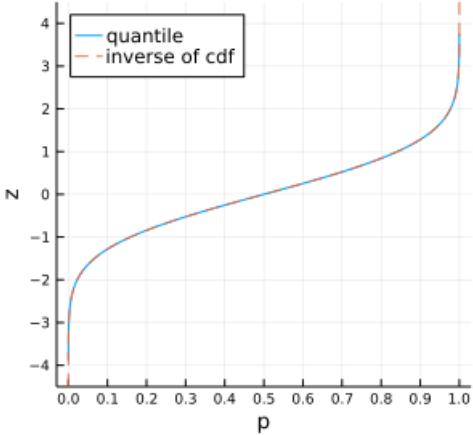
$z = \text{quantile}(\text{Normal}(0,1), p)$



```
In [4]: 1 x = range(-4.5, 4.5, 1000)
2 plot(p → quantile(Normal(0,1), p), 0, 1; label="quantile")
3 plot!(cdf.(Normal(0,1), x), x; label="inverse of cdf", ls=:dash)
4 plot!(ylim=(-4.5, 4.5))
5 plot!(y tick=-10:10, x tick=0:0.1:1)
6 plot!(yguide="z", xguide="p")
7 title!("Normal(0,1)")
8 plot!(size=(320, 320), legend=:topleft)
```

Out[4]:

Normal(0,1)

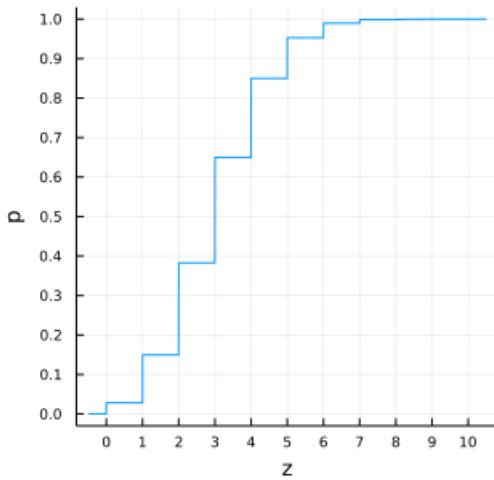


離散分布 D の場合には累積分布函数 $p = \text{cdf}(D, x)$ のグラフに平坦な部分ができてしまうので、その逆函数は存在しなくなってしまうが、適当にほぼ逆函数とみなせるものが $x = \text{quantile}(D, p)$ として定義されている。

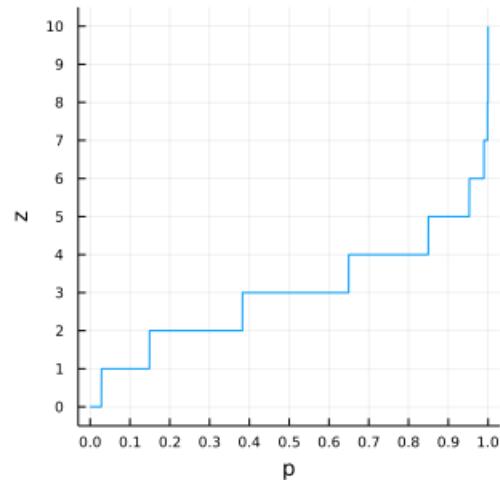
```
In [5]: 1 P1 = plot(z → cdf(Binomial(10, 0.3), z), -0.5, 10.5; label="")
2 plot!(x tick=-1:100, y tick=0:0.1:1)
3 plot!(xguide="z", yguide="p")
4 title!("p = cdf(Binomial(10, 0.3), z)")
5
6 P2 = plot(p → quantile(Binomial(10, 0.3), p), 0, 1; label="")
7 plot!(y limit=(-0.5, 10.5))
8 plot!(y tick=-1:100, x tick=0:0.1:1)
9 plot!(yguide="z", xguide="p")
10 title!("z = quantile(Binomial(10, 0.3), p)")
11
12 plot(P1, P2; size=(640, 320))
```

Out[5]:

$p = \text{cdf}(\text{Binomial}(10, 0.3), z)$



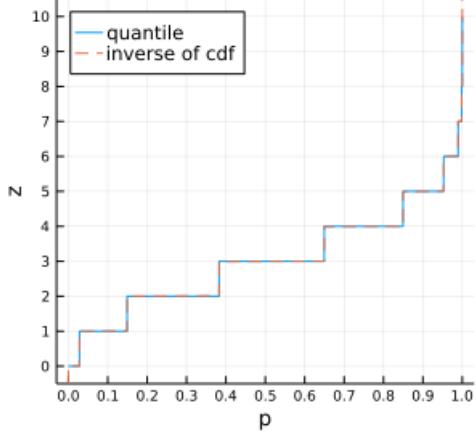
$z = \text{quantile}(\text{Binomial}(10, 0.3), p)$



```
In [6]: 1 x = range(-0.5, 10.5, 1000)
2 plot(p → quantile(Binomial(10, 0.3), p), 0, 1; label="quantile")
3 plot!(cdf.(Binomial(10, 0.3), x), x; label="inverse of cdf", ls=:dash)
4 plot!(ylim=(-0.5, 10.5))
5 plot!(y tick=-1:100, x tick=0:0.1:1)
6 plot!(y guide="z", x guide="p")
7 title!("Binomial(10, 0.3)")
8 plot!(size=(320, 320), legend=:topleft)
```

Out[6]:

Binomial(10, 0.3)



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。補累積分布函数は単調減少函数になる。

補累積分布函数 $p = \text{ccdf}(D, x) = 1 - \text{cdf}(D, x)$ の逆函数を

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

と表し、**補分位点函数** (complementary quantile function)と呼ぶ。

補分位点函数も単調減少函数になる。

例えば、

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

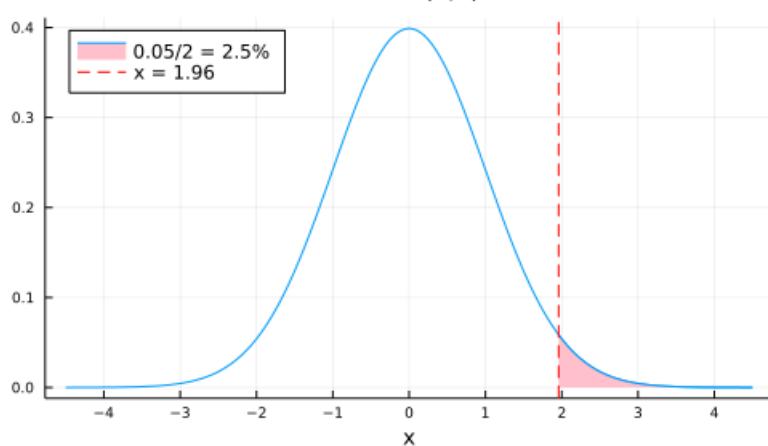
は非常に有名である。実践的にはこの値が 1.96 にぴったり等しいとみなして扱って問題ない。

標準正規分布における $\text{cquantile}(\text{Normal}(0, 1), 0.05/2) \approx 1.96$ よりも右側の確率(面積)が 2.5% になる。

```
In [7]: 1 @show c = cquantile(Normal(0,1), 0.05/2)
2 plot(Normal(0,1), -4.5, 4.5; label="", xtick=-5:5, xguide="x")
3 plot!(Normal(0,1), c, 4.5;
4     fillrange=0, c=1, fc=:pink, label="0.05/2 = 2.5%")
5 vline!([c]; label="x = $(round(c; digits=2))", c=:red, ls=:dash)
6 title!("Normal(0,1)"; legend=:topleft)
```

c = cquantile(Normal(0, 1), 0.05 / 2) = 1.9599639845400592

Out[7]:

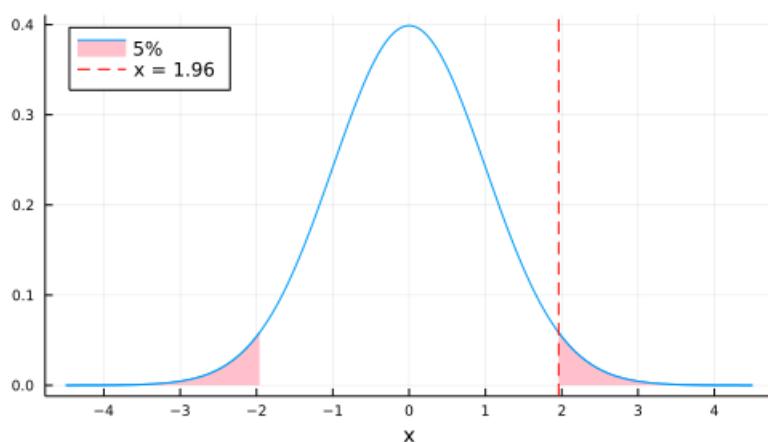


標準正規分布における $c = \text{cquantile}(\text{Normal}(0, 1), 0.05/2) \approx 1.96$ よりも右側の確率(面積)と反対側の確率(面積)の和が 5% になる。

```
In [8]: 1 @show c = cquantile(Normal(0,1), 0.05/2)
2 plot(Normal(0,1), -4.5, 4.5; label="", xtick=-5:5, xguide="x")
3 plot!(Normal(0,1), c, 4.5;
4     fillrange=0, c=1, fc=:pink, label="5%")
5 plot!(Normal(0,1), -4.5, -c;
6     fillrange=0, c=1, fc=:pink, label="")
7 vline!([c]; label="x = $(round(c; digits=2))", c=:red, ls=:dash)
8 title!("Normal(0,1)"; legend=:topleft)
```

c = cquantile(Normal(0, 1), 0.05 / 2) = 1.9599639845400592

Out[8]:



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。

注意: 二項分布の正規分布近似を使った検定には少なくとも, Wilson検定(スコア検定)とWald検定がある. 標本サイズが小さい場合のそれらの性質の良さは大きく違い, 以下で説明するWilson検定の性質は非常に良いことが知られている. 例えは, 次の論文を参照せよ:

- Alan Agresti and Brent A. Coull. Approximate is Better than "Exact" for Interval Estimation of Binomial Proportions. *The American Statistician*, Vol. 52, No. 2 (May, 1998), pp. 119-126. [[Google Scholar](https://scholar.google.co.jp/scholar?cluster=5129299358902170657) (<https://scholar.google.co.jp/scholar?cluster=5129299358902170657>)]

この論文を読めば分かるように, 二項分布モデルにおけるWilsonのP値と信頼区間は, 二項分布における確率を正確に計算する方法と比較しても, 広く使用が勧められる優れた方法になっている.

注意: その論文で良い性質を持つと結論されているのは, Yatesの連続性補正をしていないナマのWilsonのP値と信頼区間である. だから, この解説ノートでは連続性補正版を扱わない.

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き,

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

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,

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

とおく,

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

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

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

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

以下はほぼ以上で説明した公式通りの実装になっている.

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

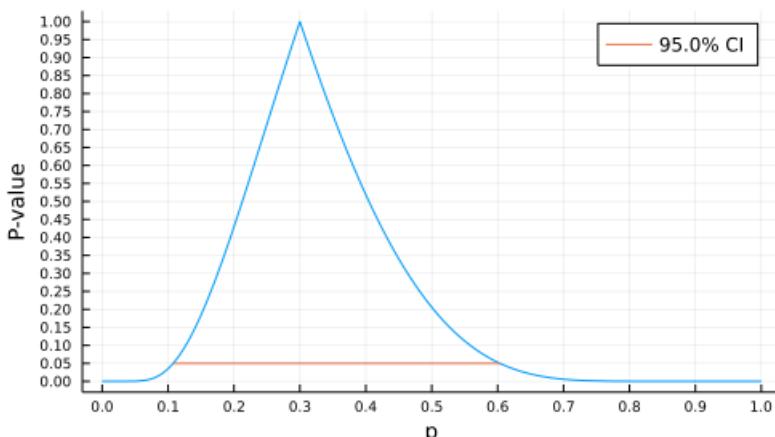
Out[9]: confint_wilson (generic function with 1 method)

5.4 P値函数のcompatibilityの指標としての解釈の仕方

```
In [10]: 1 @show n, k, α = 10, 3, 0.05
2 @show ci = confint_wilson(k, n, α) ▷ 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(α,2); label="$(100(1-α))% CI")
8 title!("Wilson's P-value function for n=$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[10]: Wilson's P-value function for n=10, k=3



上のグラフは「 $n = 10$ 回中当たりが $k = 3$ 回」というデータに関する二項分布モデルのパラメータ p のP値函数と95%信頼区間のプロットである。

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 の範囲はおよそ 10.8% から 60.3% になる。

これが 95% 信頼区間の稳健な(自信過剰にならずにすむ)解釈になる。

このように信頼区間も「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」というように言うことがある。

「棄却」という強い響きを持つ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。(ゆえに信頼区間の解釈においても同様である。)

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。

例えば、上で述べた「棄却」という強い響きを持つ言い方に実質的な力を持たせるためには、P値の計算以外に沢山の議論が必要になる。

なお、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)の指標とみなすこの解説ノートの立場は

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

米国統計学会による声明なので、現代的なP値の理解はこれが標準的であるとも考えられる。

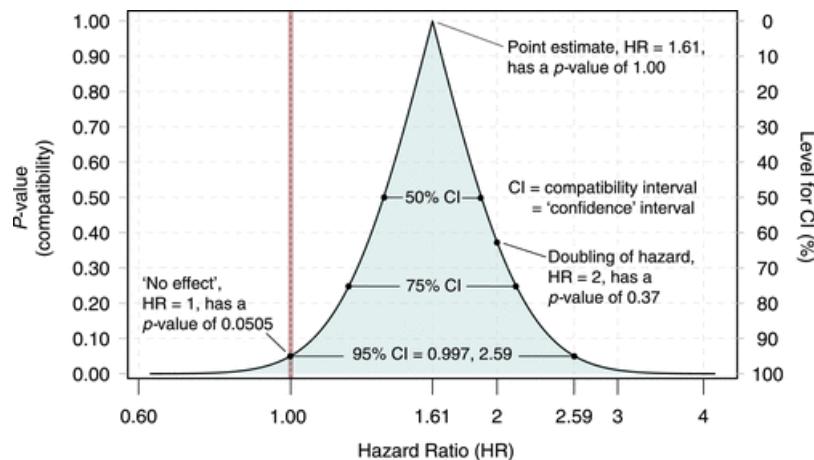
次の有名な記事でもP値の解釈の説明について **compatibility** がキーワードになっている:

- Valentin Amrhein , Sander Greenland, and Blake McShane. Scientists rise up against statistical significance. Valentin Amrhein, Sander Greenland, Blake McShane and more than 800 signatories call for an end to hyped claims and the dismissal of possibly crucial effects. [[DOI \(https://doi.org/10.1038/d41586-019-00857-9\)](https://doi.org/10.1038/d41586-019-00857-9)]

P値函数や信頼区間の解釈については次の論文が非常にわかりやすい:

- Discuss practical importance of results based on interval estimates and p-value functions, not only on point estimates and null p-values. Journal of Information Technology, First Published June 3, 2022. Nature 567, 305-307 (2019) [[DOI \(https://doi.org/10.1177%2F02683962221105904\)](https://doi.org/10.1177%2F02683962221105904)]

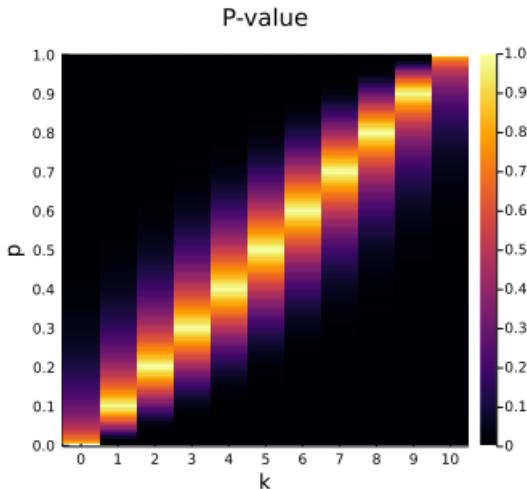
この論文には次の図がある:



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.

```
In [11]: 1 n = 10
2 k = 0:n
3 p = range(0, 1, 200)
4 heatmap(k, p, (k,p)→pvalue_wilson(k, n, p); clim=(0,1))
5 plot!(xtick=0:n, ytick=0:0.1:1)
6 plot!(xguide="k", yguide="p")
7 plot!(size=(350, 320))
8 title!("P-value")
```

Out[11]:



上のグラフは, $n = 10$ とし, 横軸を $k = 0, 1, \dots, 10$ とし, 縦軸を区間 $p \in [0, 1]$ としたときの, P値函数 $pvalue_{Wilson}(k|n, p)$ の値のヒートマップである. ヒートマップでは値が大きな所ほど明るくプロットされる.

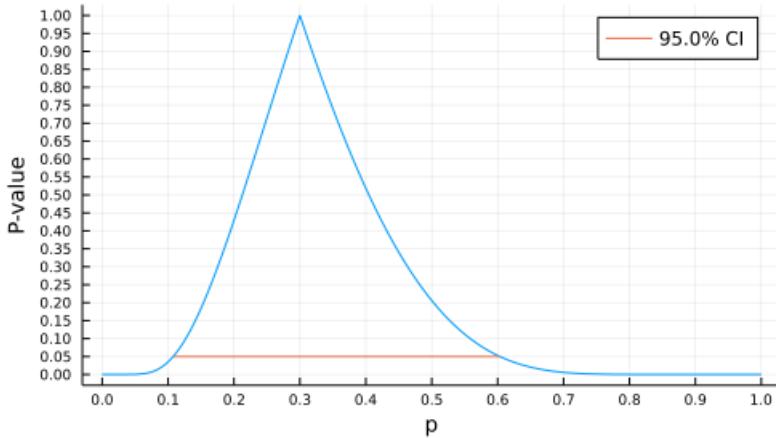
P値はデータの数値 k とモデルのパラメータ値 p の相性の良さ(compatibility, 両立性)の指標の1つであった.

ゆえに, 上のグラフにおける明るい部分はデータの数値 k とパラメータ値 p の相性が良い部分であり, 暗い部分は相性が悪い部分である.

$k = 3$ での「切断面」のグラフがすでに示した以下のグラフになる.

In [12]: 1 P_Wilson

Out[12]: Wilson's P-value function for n=10, k=3



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う。

その代わりに、WaldのP値に対応する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：

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

この定義はWilsonのP値の定義における分母の平方根内の $p(1 - p)/n$ を $\hat{p}(1 - \hat{p})/n$ で置き換えることによって得られる。

その置き換えは中心極限定理による二項分布の正規分布近似を悪化させる。

その代わりに、次の節で示すようにこのP値の定義に対応する信頼区間の導出はWilsonのP値の場合よりもずっと易しくなり、対応する信頼区間の計算も易しくなる。(二次方程式を解く必要がなくなる。)

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。

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

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 $[0, 1]$ をはみ出すことがある.

6.3 WaldのP値と信頼区間の実装例

In [13]:

```

1 function pvalue_wald(k, n, p)
2   p̂ = k/n
3   sehat = √(p̂*(1 - p̂)/n)
4   2ccdf(Normal(0,1), abs(p̂ - p)/sehat)
5 end
6
7 function confint_wald(k, n, α)
8   z = cquantile(Normal(0,1), α/2)
9   p̂ = k/n
10  sehat = √(p̂*(1 - p̂)/n)
11  p̂ - z*sehat, p̂ + z*sehat
12 end

```

Out[13]: confint_wald (generic function with 1 method)

In [14]:

```

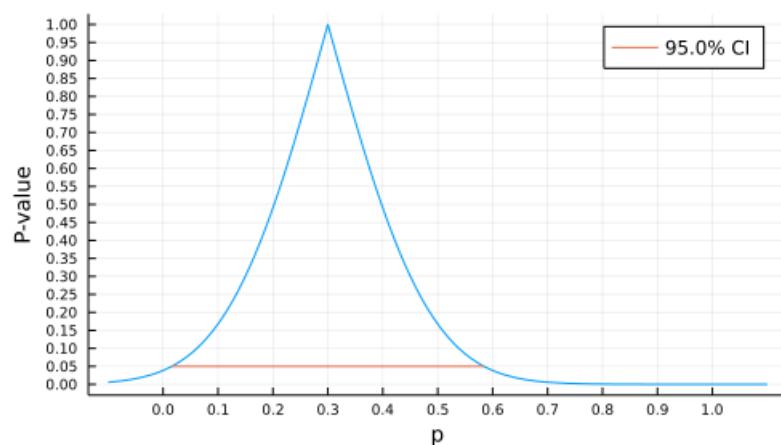
1 @show n, k, α = 10, 3, 0.05
2 @show ci = confint_wald(k, n, α) ▷ collect
3
4 P_Wald = plot(p → pvalue_wald(k, n, p), -0.1, 1.1; label="")
5 plot!(xguide="p", yguide="P-value")
6 plot!(xtick=0:0.1:1, ytick=0:0.05:1)
7 plot!(ci, fill(α,2); label="$(100(1-α))% CI")
8 title!("Wald's P-value function for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

```

$(n, k, \alpha) = (10, 3, 0.05) = (10, 3, 0.05)$
 $ci = \text{confint_wald}(k, n, \alpha) \triangleright \text{collect} = [0.0159742349106739, 0.5840257650893261]$

Out[14]:

Wald's P-value function for $n=10, k=3$



WaldのP値函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値函数とそれに付随する信頼区間はこのように誤差が大きくなってしまうものだが、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。

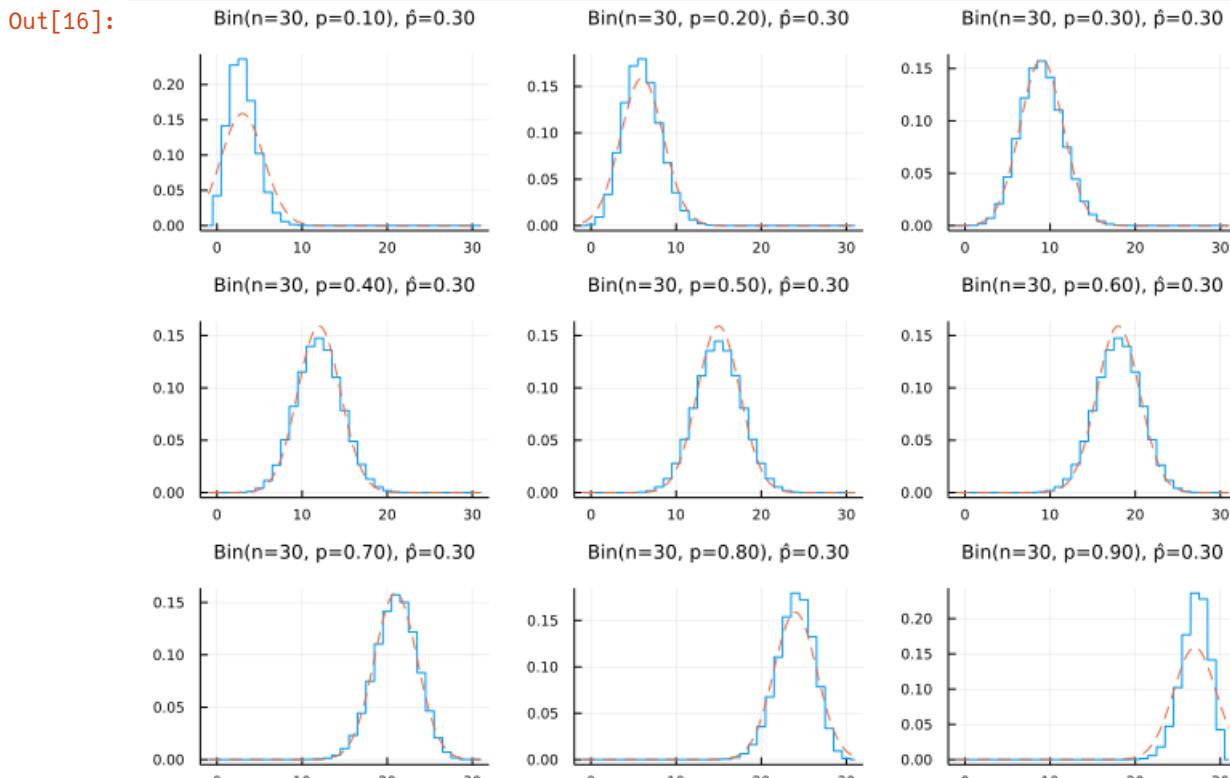
このように統計学的道具にはそれぞれ異なる利点と欠点があり、適材適所で使い分けるのがよい。

6.4 WaldのP値と信頼区間の構成で使った正規分布近似の粗さの視覚化

```
In [15]: 1 mypdf(dist::DiscreteUnivariateDistribution, x) = pdf(dist, round(Int, x))
2
3 function plot_bin_normal(n, p, p̂;
4     μ = n*p, σ = √(n*p*(1-p)), δ̂ = √(n*p̂*(1-̂p)),
5     xlim = (max(-1, μ - 4.5σ), min(n+1, μ + 4.5σ)),
6     titlehead = "", kwargs...)
7     bin = Binomial(n, p)
8     normal = Normal(μ, δ̂)
9     title = titlehead *
10        "Bin(n=$n, p=$(@sprintf("%.2f", p))), p̂=$(@sprintf("%.2f", p̂))"
11    plot(; legend=:outertop, legendfontsize=10, title)
12    plot!(x → mypdf(bin, x), xlim..., label="")
13    plot!(normal, xlim..., label="", ls=:dash)
14    plot!(; kwargs...)
15 end
```

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

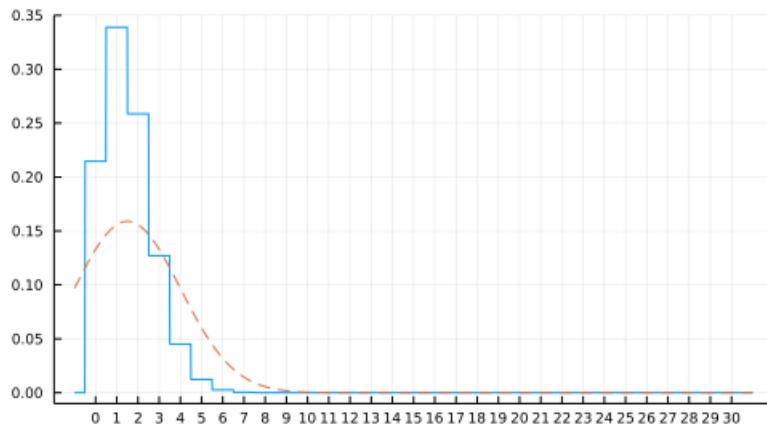
```
In [16]: 1 n, p̂ = 30, 0.3
2 PP = [plot_bin_normal(n, p, p̂; xlim=(-1, n+1), titlefontsize=8)
3         for p in 0.1:0.1:0.9]
4 plot(PP..., layout=(3, 3), size=(700, 500))
```



```
In [17]: 1 n, p̂ = 30, 0.3
2 ps = 0.05:0.01:0.95
3 ps = [fill(ps[begin], 10); ps; fill(ps[end], 10); reverse(ps)]
4 anim = @animate for p in ps
5     plot_bin_normal(n, p, p̂;
6         xlim=(-1, n+1), ylim=(-0.01, 0.35),
7         xtick=0:n, ytick=0:0.05:1,
8         titlehead="Wald: ")
9 end
10 gif(anim, "Wald_normal_approx.gif")
```

[Info: Saved animation to
fn = D:\OneDrive\public\0036\Wald_normal_approx.gif
@ Plots D:\.julia\packages\Plots\gl4j3\src\animation.jl:137

Out[17]: Wald: Bin(n=30, p=0.05), p̂=0.30



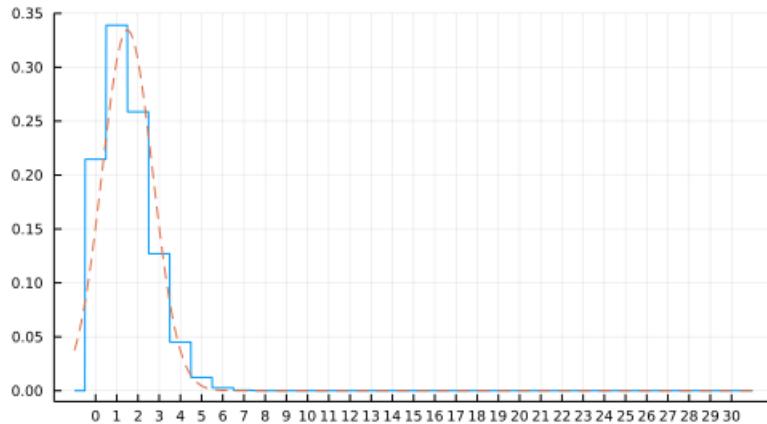
上の動画は以下の場所で見ることができる:

- https://github.com/genkuroki/public/blob/main/0036/Wald_normal_approx.gif
(https://github.com/genkuroki/public/blob/main/0036/Wald_normal_approx.gif)

```
In [18]: 1 n = 30
2 ps = 0.05:0.01:0.95
3 ps = [fill(ps[begin], 10); ps; fill(ps[end], 10); reverse(ps)]
4 anim = @animate for p in ps
5     plot_bin_normal(n, p, p;
6         xlim=(-1, n+1), ylim=(-0.01, 0.35),
7         xtick=0:n, ytick=0:0.05:1,
8         titlehead="Wilson: ")
9 end
10 gif(anim, "Wilson_normal_approx.gif")
```

[Info: Saved animation to
fn = D:\OneDrive\public\0036\Wilson_normal_approx.gif
@ Plots D:\.julia\packages\Plots\gl4j3\src\animation.jl:137

Out[18]: Wilson: Bin(n=30, p=0.05), $\hat{p}=0.05$



上の動画は以下の場所で見ることができる:

- https://github.com/genkuroki/public/blob/main/0036/Wilson_normal_approx.gif
[\(https://github.com/genkuroki/public/blob/main/0036/Wilson_normal_approx.gif\)](https://github.com/genkuroki/public/blob/main/0036/Wilson_normal_approx.gif)

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。

注意: 二項分布モデルでのClopper-Pearsonの信頼区間には、シンプルな良い公式が存在するので(後述)、平易な実装で効率的に計算できるという利点がある。しかし、一般に離散分布モデルにおいて「片側確率の2倍」で定義されたClopper-Pearson型のP値は他の流儀のP値の定義よりも過剰に大きくなる傾向があり、Clopper-Pearson型の信頼区間には過剰に広くなる傾向がある。

注意: ところが非常に困ったことに、トレードオフ(ある利点を得るために別の欠点を受け入れること)の考え方ではなく、独自の「主義」によって、「両側検定のP値は片側確率の2倍で定義しなければいけない」と不合理に信じ、そう信じることを他人に勧めている人達がいる。そのような人達の不合理なこだわりに影響を受けないように注意するべきである。

7.1 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:

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

ここで、 $\text{cdf}(\text{Binomial}(n, p), k)$ と $\text{ccdf}(\text{Binomial}(n, p), k - 1)$ 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」 = 「片側確率の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:

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

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 に関する導函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

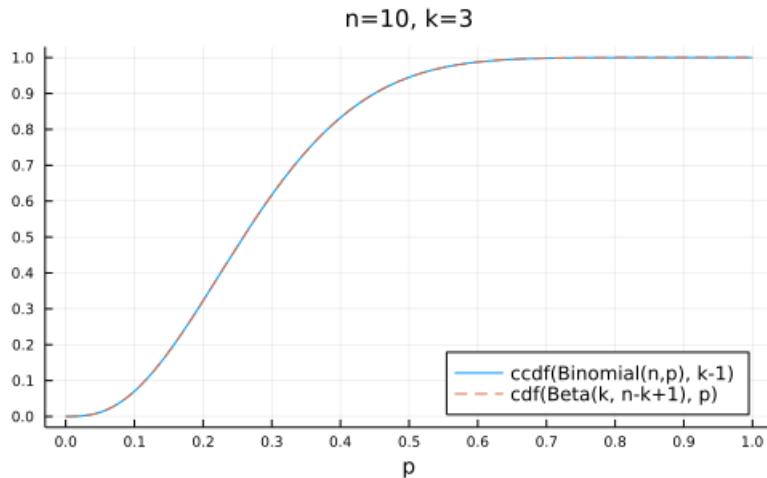
$$\begin{aligned} \frac{1}{B(k, n-k+1)} \frac{\partial}{\partial p} \int_0^p t^{k-1} (1-t)^{n-k} dt &= \frac{1}{B(k, n-k+1)} p^{k-1} (1-p)^{n-k} \\ &= \frac{\Gamma(k+(n-k+1))}{\Gamma(k)\Gamma(n-k+1)} p^{k-1} (1-p)^{n-k} \\ &= \frac{n!}{(k-1)!(n-k)!} p^{k-1} (1-p)^{n-k}. \end{aligned}$$

これで左辺と右辺の導函数が等しいことが証明された。

注意：二項分布の累積分布函数と補累積分布函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函数と累積分布函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函数は regularized incomplete Beta function としてコンピュータの基本特殊函数ライブラリで効率的に実装されている。

```
In [19]: 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[19]:



ぴったり重なっている。

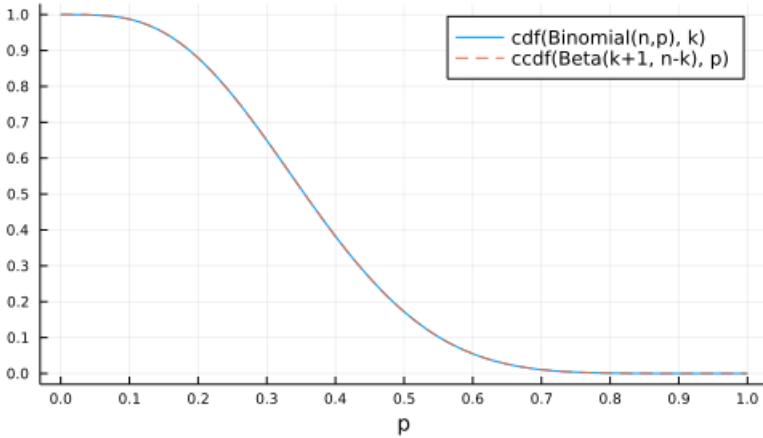
In [20]:

```

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[20]:

 $n=10, k=3$ 

ぴったり重なっている。

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

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

が具体的にどのように表されるかを計算しよう。Clopper-PearsonのP値の定義は

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

であった。これは二項分布とベータ分布の関係

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

を使えば、

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

に書き直される。ゆえに、

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

のことより、

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

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

以下のようにClopper-PearsonのP値と信頼区間の実装は非常にシンプルになる。

In [21]:

```
1 function pvalue_cp(k, n, p)
2     bin = Binomial(n, p)
3     min(1, 2ccdf(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[21]: confint_cp (generic function with 1 method)

In [22]:

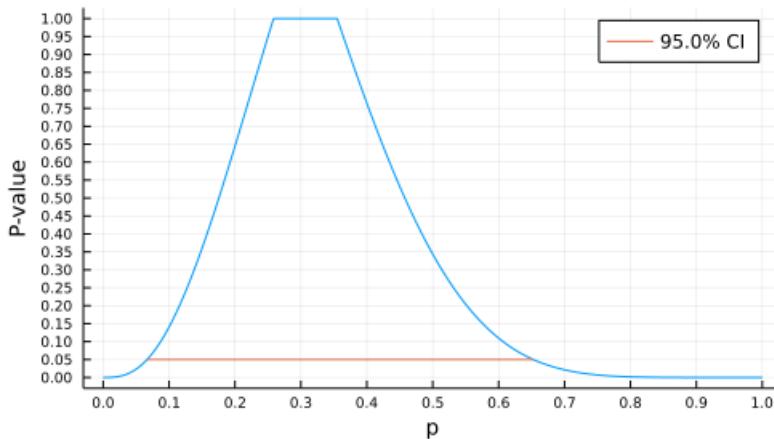
```
1 @show n, k, α = 10, 3, 0.05
2 @show ci = confint_cp(k, n, α) ▷ collect
3
4 P_CP = plot(p → pvalue_cp(k, n, p), 0, 1; label="")
5 plot!(xguide="p", yguide="P-value")
6 plot!(xtick=0:0.1:1, ytick=0:0.05:1)
7 plot!(ci, fill(α, 2); label="$(100(1-α))% CI")
8 title!("Clopper-Pearson P-value function for n=$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[22]:

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。

再注意: 独自の「主義」に基く不合理なこだわりによって、「両側検定のP値は片側確率の2倍で定義しなければいけない」という誤解を積極的に宣伝している人達がいる。そのような人達は、「片側確率の2倍ではないSterneのP値とそれに対応する信頼区間を使うことは、より小さなP値とより狭い信頼区間を不当な方法で求める行為に近い」のように言って来るかもしれない。そのような不合理な考え方を一切受け入れないように注意した方がよい。Sterneの方法とClopper-Pearsonの方法には互いに異なる利点と欠点がある。どちらを使うかはトレードオフの問題になる。

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 [23]: 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[23]: `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 [24]: 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)
```

2.917044 seconds
2.846243 seconds
2.857276 seconds

Out[24]: `5.354898851685295e-12`

注意: 上で、 $x \leq y$ ではなく、 $x \approx y$ 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 [25]: 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[25]: confint_sterne (generic function with 1 method)

上のセルでのSterneのP値函数の実装結果は大幅に高速化されている!

```

In [26]: 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)

0.000065 seconds (8 allocations: 1.875 KiB)
0.000014 seconds (8 allocations: 1.875 KiB)
0.000009 seconds (8 allocations: 1.875 KiB)

```

Out[26]: 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 [27]: 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)

0.116101 seconds (50.27 k allocations: 3.701 MiB, 97.75% compilation time)
0.002703 seconds (11.48 k allocations: 2.402 MiB)
0.002874 seconds (11.48 k allocations: 2.402 MiB)

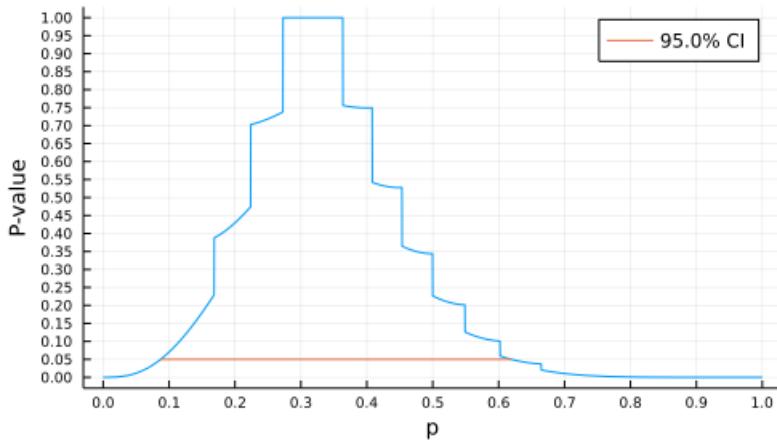
```

Out[27]: (0.2995232976841354, 0.30047704792862706)

```
In [28]: 1 @show n, k, α = 10, 3, 0.05
2 @show ci = confint_sterne(k, n, α) ▷ 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(α,2); label="$(100(1-α))% CI")
8 title!("Sterne's P-value function for n=$n, k=$k")
```

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

Out[28]: Sterne's P-value function for n=10, k=3

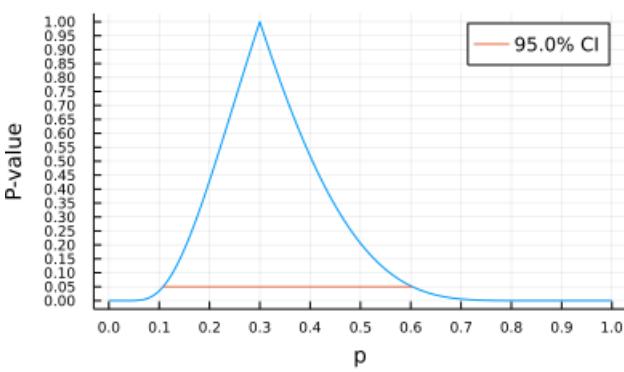


SterneのP値函数はこのように不連続函数になる。

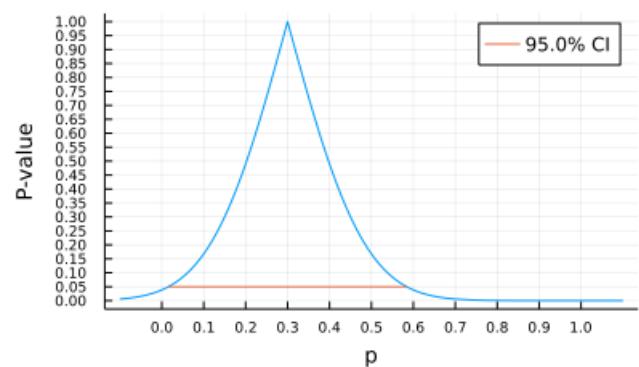
9 4種のP値函数と信頼区間を同時にプロット

```
In [29]: 1 plot(P_Wilson, P_Wald, P_CP, P_Sterne; layout=(2, 2), size=(800, 500))
```

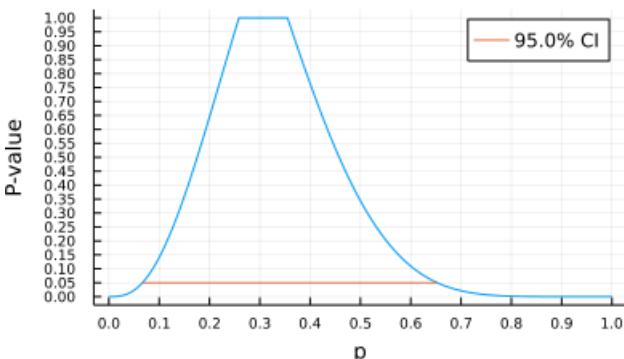
Out[29]: Wilson's P-value function for n=10, k=3



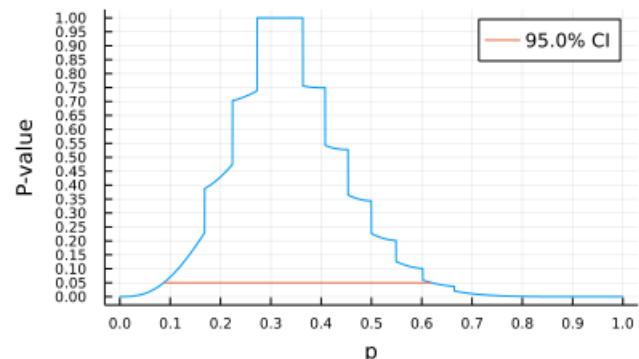
Wald's P-value function for n=10, k=3



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



Sterne's P-value function for n=10, k=3



In [30]:

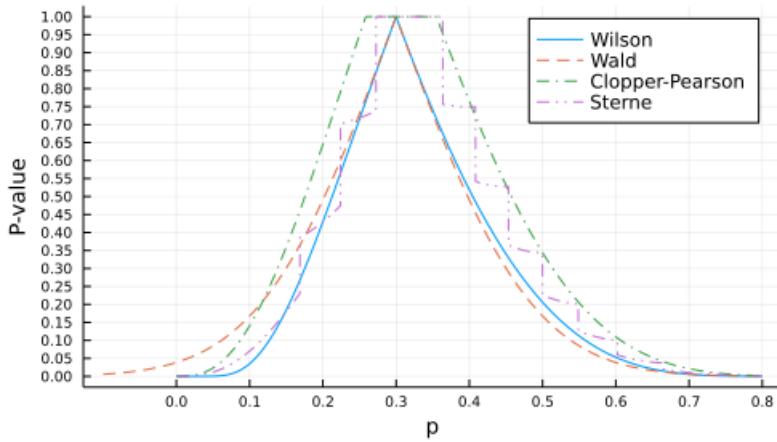
```

1 @show n, k, α = 10, 3, 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), -0.1, 0.8; 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=0:0.1:1, ytick=0:0.05:1)
14 title!("P-value functions for n=$n, k=$k")

```

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

Out[30]:

P-value functions for $n=10, k=3$ 

In [31]:

```

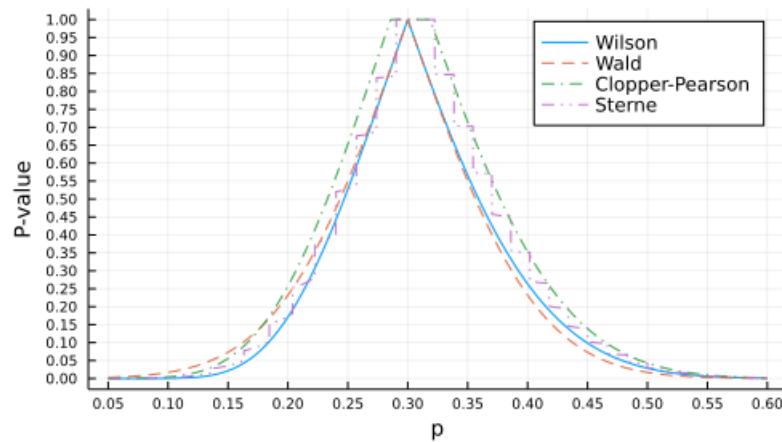
1 @show n, k, α = 30, 9, 0.05
2 @show confint_wilson(k, n, α)
3 @show confint_wald(k, n, α)
4 @show confint_cp(k, n, α)
5 @show confint_sterne(k, n, α)
6
7 plot()
8 plot!(p → pvalue_wilson(k, n, p), 0.05, 0.6; label="Wilson")
9 plot!(p → pvalue_wald(k, n, p); label="Wald", ls=:dash)
10 plot!(p → pvalue_cp(k, 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")

```

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

Out[31]:

P-value functions for n=30, k=9



In [32]:

```

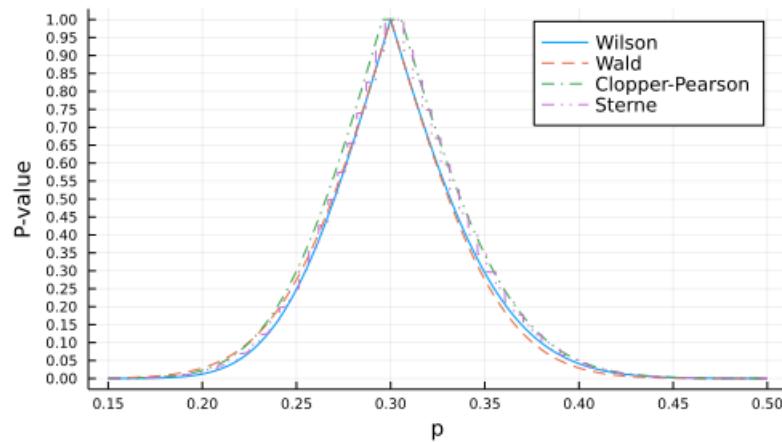
1 @show n, k, α = 100, 30, 0.05
2 @show confint_wilson(k, n, α)
3 @show confint_wald(k, n, α)
4 @show confint_cp(k, n, α)
5 @show confint_sterne(k, n, α)
6
7 plot()
8 plot!(p → pvalue_wilson(k, n, p), 0.15, 0.5; label="Wilson")
9 plot!(p → pvalue_wald(k, n, p); label="Wald", ls=:dash)
10 plot!(p → pvalue_cp(k, 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")

```

$(n, k, \alpha) = (100, 30, 0.05) = (100, 30, 0.05)$
 $\text{confint_wilson}(k, n, \alpha) = (0.21894885294932742, 0.39584854633346683)$
 $\text{confint_wald}(k, n, \alpha) = (0.21018316681457921, 0.38981683318542076)$
 $\text{confint_cp}(k, n, \alpha) = (0.21240642048953662, 0.39981467617980404)$
 $\text{confint_sterne}(k, n, \alpha) = (0.21419730401454767, 0.3996107372232876)$

Out[32]:

P-value functions for n=100, k=30

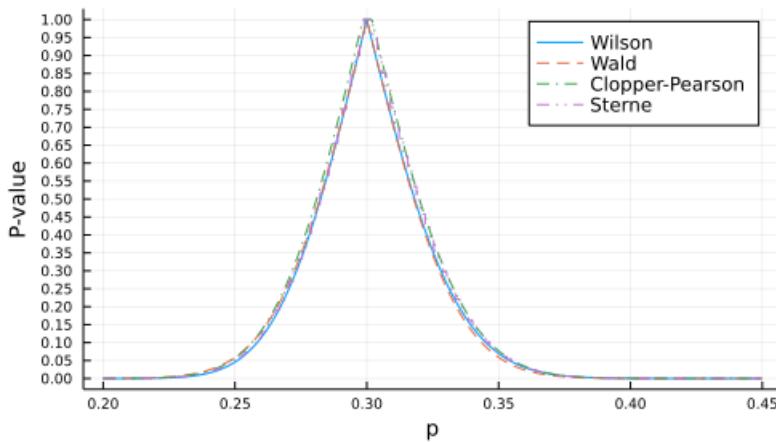


n を大きくすると、4種のP値函数はほぼ一致するようになる。

```
In [33]: 1 @show n, k, α = 300, 90, 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), 0.2, 0.45; 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")
```

$(n, k, \alpha) = (300, 90, 0.05) = (300, 90, 0.05)$
 $\text{confint_wilson}(k, n, \alpha) = (0.2509396637847664, 0.3541175247049418)$
 $\text{confint_wald}(k, n, \alpha) = (0.2481442271826376, 0.3518557728173624)$
 $\text{confint_cp}(k, n, \alpha) = (0.2486815765240469, 0.3553195044630606)$
 $\text{confint_sterne}(k, n, \alpha) = (0.24972109503883086, 0.3548448258761688)$

Out[33]: P-value functions for n=300, k=90



10 ベイズ統計での信用区間に対応するP値函数

P値函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 \(<https://twilog.org/genkuroki/search?word=P%E5%80%A4%20%E4%BF%A1%E9%A0%BC%E5%8C%BA%E9%96%93%20%E8%A1%A8%E8%A3%8F%E4%B8%>\)](https://twilog.org/genkuroki/search?word=P%E5%80%A4%20%E4%BF%A1%E9%A0%BC%E5%8C%BA%E9%96%93%20%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値函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値函数を構成することができる。

ベイズ版のP値函数と通常のP値函数を比較したい。

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おう。

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函数を

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

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値函数

HDI版信用区間に対応するP値函数 $\text{pvalue}_{\text{HDI}}(k|n, p, a, b)$ 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値函数** (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値函数の実装例

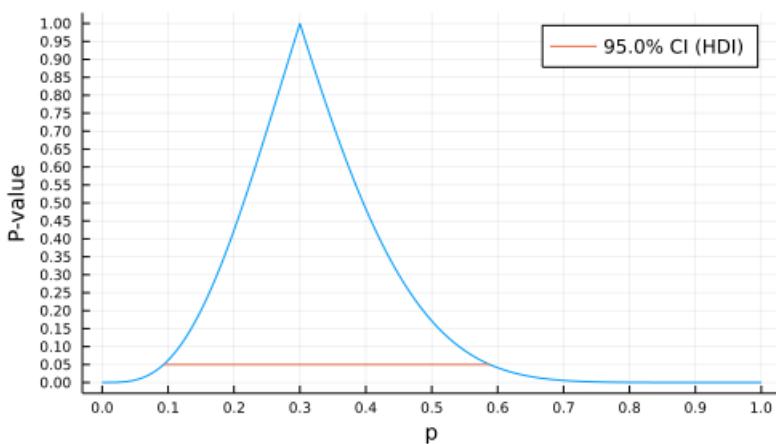
```
In [34]: 1 function hdi(dist::ContinuousUnivariateDistribution, α = 0.05; alg = Brent())
2     f(p) = quantile(dist, p + (1 - α)) - quantile(dist, p)
3     o = optimize(f, 0, α, alg)
4     p = o.minimizer
5     quantile.(dist, (p, p + (1 - α)))
6 end
7
8 function pvalue_hdi(dist::ContinuousUnivariateDistribution, 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     if k+a ≤ 1
26         return ccdf(posterior, p)
27     elseif n-k+b ≤ 1
28         return cdf(posterior, p)
29     end
30     pvalue_hdi(posterior, p)
31 end
32
33 function credint_hdi(k, n, α; a=1, b=1)
34     posterior = Beta(k+a, n-k+b)
35     hdi(posterior, α)
36 end
```

Out[34]: credint_hdi (generic function with 1 method)

```
In [35]: 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[35]: Bayesian P-value function for n=10, k=3, a=1, b=1



このグラフも通常のP値函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)

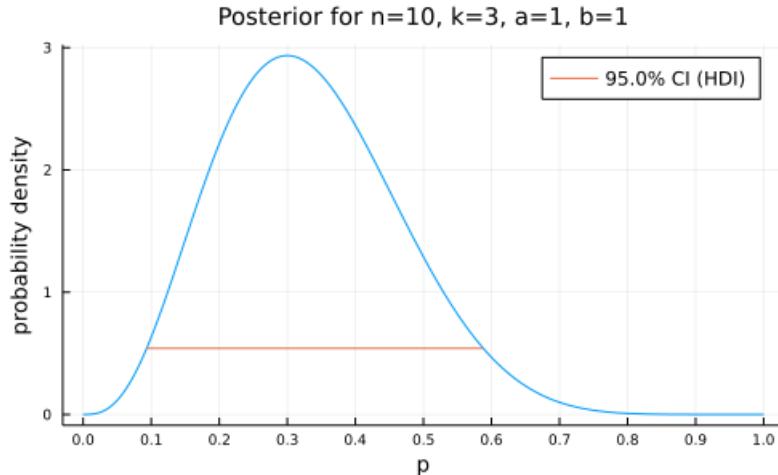
を表すグラフだと解釈可能である. この解釈は次の事後分布のグラフにもそのまま一般化される.

In [36]:

```
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[36]:



10.4 WilsonのP値函数とベイズ版P値函数のHDI版の比較

この節の目的は通常のP値函数とベイズ版P値函数を比較することであった。

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

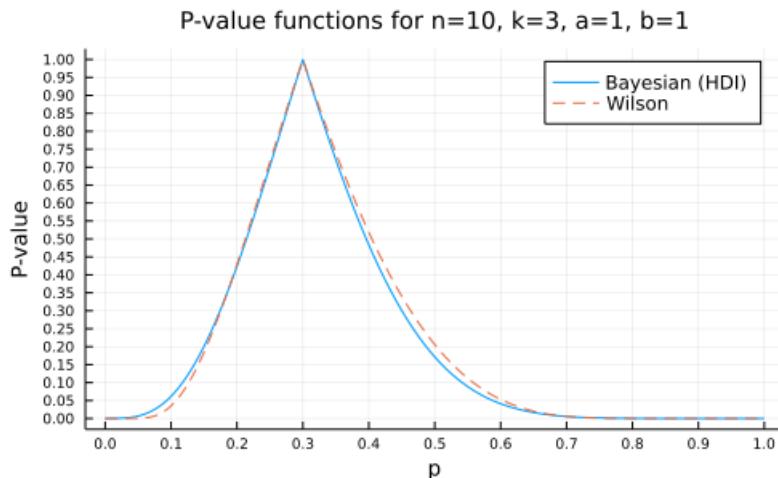
上のベイズ版P値函数をWilsonのP値函数と重ねてプロットしてみよう。

In [37]:

```
1 @show n, k, α = 10, 3, 0.05
2 @show a, b = 1, 1
3
4 plot(p → pvalue_hdi(k, n, p; a, b), 0, 1; 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=0:0.1:1, ytick=0:0.05:1)
8 title!("P-value functions 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)

Out[37]:



ベイズ版P値函数が通常のWilsonのP値函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.

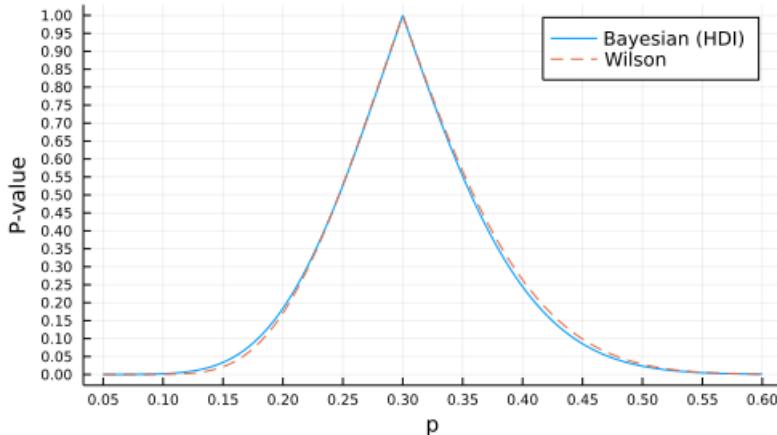
In [38]:

```
1 @show n, k, α = 30, 9, 0.05
2 @show a, b = 1, 1
3
4 plot(p → pvalue_hdi(k, n, p; a, b), 0.05, 0.6; 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) = (30, 9, 0.05) = (30, 9, 0.05)
(a, b) = (1, 1) = (1, 1)

Out[38]:

P-value functions for $n=30, k=9, a=1, b=1$



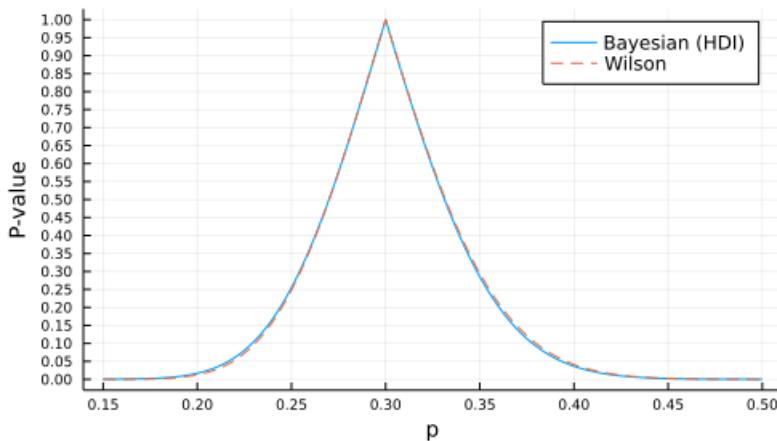
In [39]:

```
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[39]:

P-value functions for $n=100, k=30, a=1, b=1$



$n = 100, k = 30$ ではほぼぴったり重なっている!

10.5 二項分布モデルのETI版の信用区間とそれに対応するP値

ETI版の信用区間に対応するP値は以下のように「片側確率の2倍」で定義されるので, Clopper-PearsonのP値のベイズ版であると考えられる.

posterior は事後分布を表すこととする:

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

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値函数は次になる:

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

10.6 ETI版の信用区間とそれに対応するベイズ版P値函数の実装例

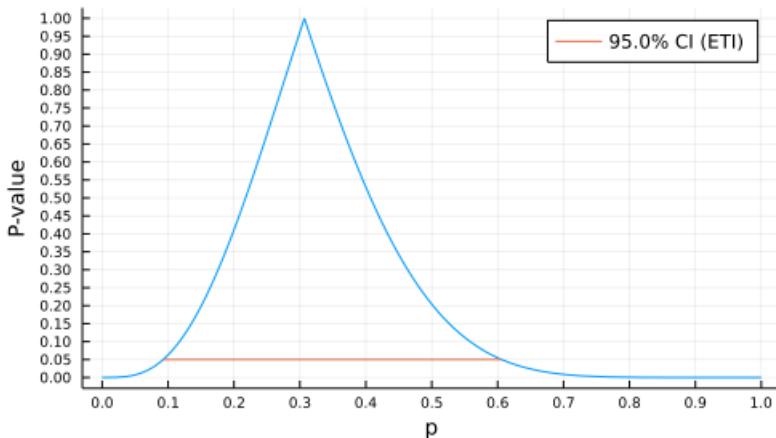
```
In [40]: 1 function pvalue_eti(k, n, p; a=1/2, b=1/2)
2     posterior = Beta(k+a, n-k+a)
3     min(1, 2cdf(posterior, p), 2ccdf(posterior, p))
4 end
5
6 function credint_eti(k, n, α; a=1/2, b=1/2)
7     posterior = Beta(k+a, n-k+a)
8     quantile(posterior, α/2), cquantile(posterior, α/2)
9 end
```

Out[40]: credint_eti (generic function with 1 method)

```
In [41]: 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")
```

$(n, k, \alpha) = (10, 3, 0.05) = (10, 3, 0.05)$
 $(a, b) = (1/2, 1/2) = (0.5, 0.5)$
 $\text{ci} = \text{credint}_\text{eti}(k, n, \alpha; a, b) \triangleright \text{collect} = [0.09269459393815314, 0.6058183181486713]$

Out[41]: Bayesian P-value function for n=10, k=3, a=0.5, b=0.5



10.7 WilsonのP値函数とベイズ版P値函数のETI版の比較

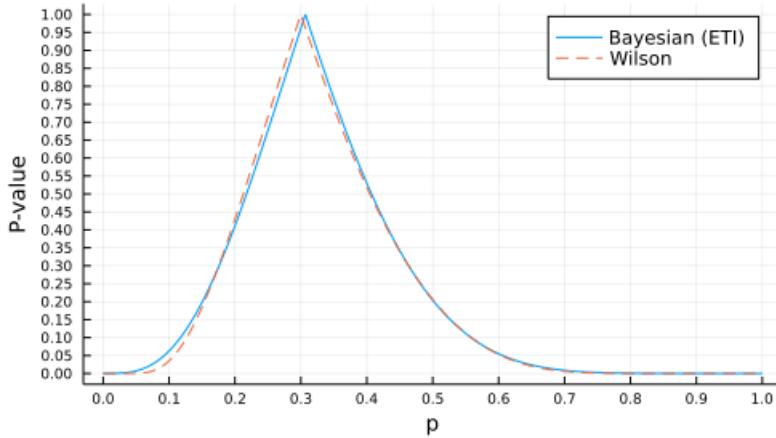
ベイズ版P値函数のETI版は, Jeffreys事前分布 $\text{Beta}(1/2, 1/2)$ の場合にWilsonのP値とよく一致することを確認しよう.

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 [42]: 1 @show n, k, α = 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")
```

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

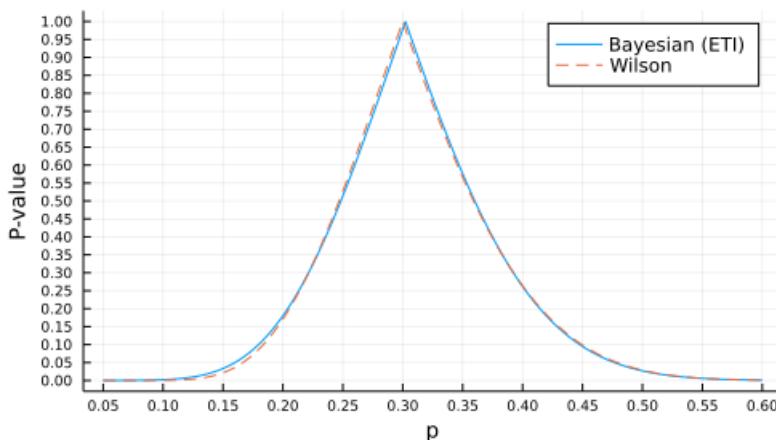
Out[42]: P-value functions for n=10, k=3, a=0.5, b=0.5



```
In [43]: 1 @show n, k, α = 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")
```

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

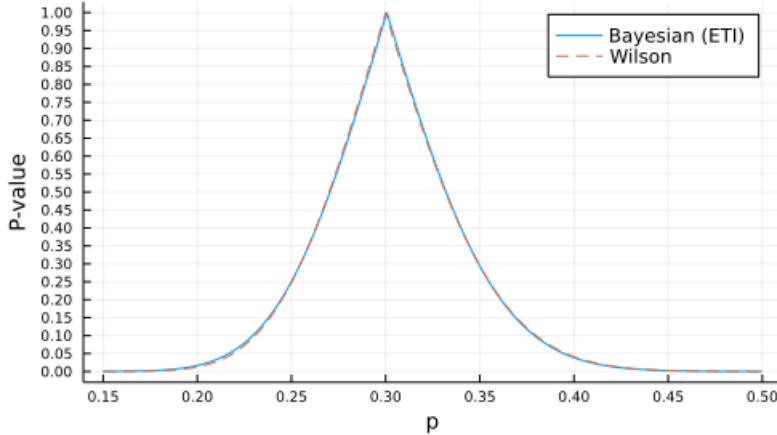
Out[43]: P-value functions for n=30, k=9, a=0.5, b=0.5



```
In [44]: 1 @show n, k, α = 100, 30, 0.05
2 @show a, b = 1/2, 1/2
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")
```

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

Out[44]: P-value functions for n=100, k=30, a=0.5, b=0.5



$n = 100, k = 30$ ではほぼぴったり重なっている!

ベイズ版P値函数の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 Beta(1/2, 1/2) ではなく、事前分布 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。

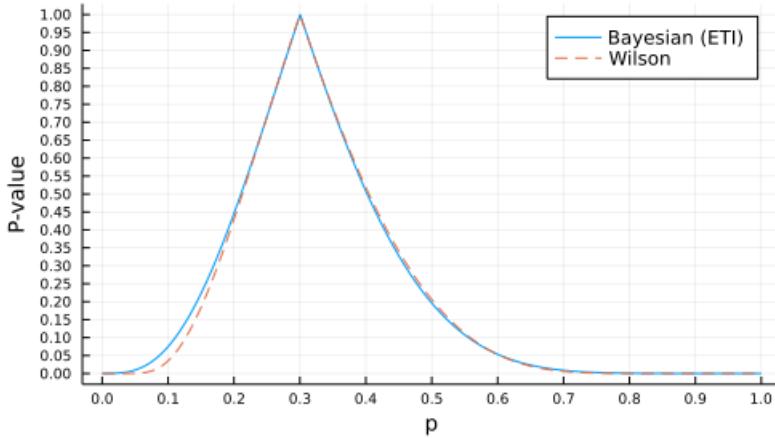
ベータ分布についてはウィキペディアを参照せよ:

- 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>)

```
In [45]: 1 @show n, k, α = 10, 3, 0.05
2 @show a, b = 1//3, 1//3
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")
```

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

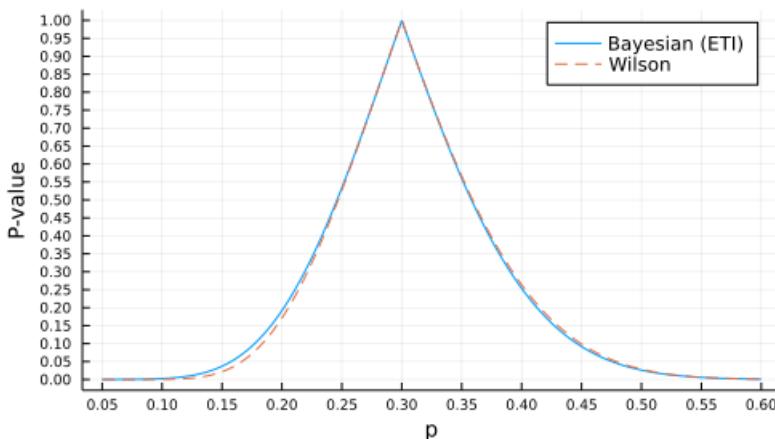
Out[45]: P-value functions for $n=10, k=3, a=1/3, b=1/3$



```
In [46]: 1 @show n, k, α = 30, 9, 0.05
2 @show a, b = 1//3, 1//3
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")
```

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

Out[46]: P-value functions for $n=30, k=9, a=1/3, b=1/3$



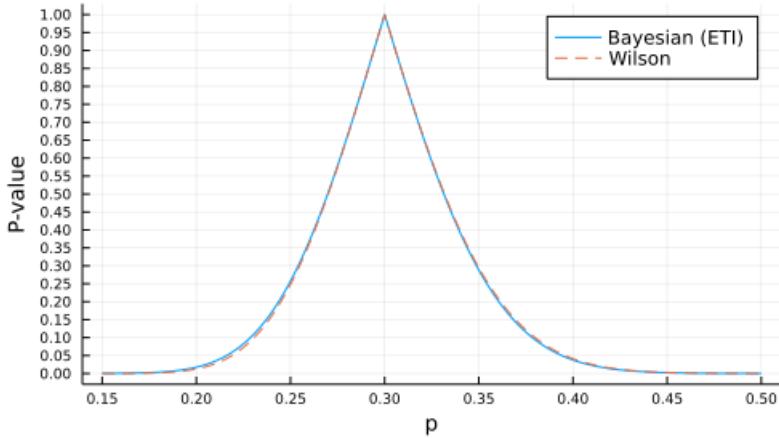
In [47]:

```

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")

```

$(n, k, \alpha) = (100, 30, 0.05) = (100, 30, 0.05)$
 $(a, b) = (1 // 3, 1 // 3) = (1/3, 1/3)$

Out[47]: P-value functions for $n=100, k=30, a=1/3, b=1/3$ 

$n = 100, k = 30$ ではほぼぴったり重なっている!

10.8 ETI版信用区間とClopper-Pearsonの信頼区間の類似と比較

ETI版の信用区間に付随するP値函数の定義は

$$pvalue_{\text{ETI}}(k|n, p, a, b) = \min \left(\frac{1}{2\text{cdf}(\text{Beta}(k + a, n - k + b), p)}, \frac{1}{2\text{ccdf}(\text{Beta}(k + a, n - k + b), p)} \right)$$

であり, Clopper-PearsonのP値函数の定義は

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

と書き直せるのであった. これらは非常によく似ている.

ETI版の信用区間の定義は

$$\begin{aligned} \text{credint}_{\text{ETI}}(k|n, p, a, b) \\ = [\text{quantile}(\text{Beta}(k + a, n - k + b), \alpha/2), \text{cquantile}(\text{Beta}(k + a, n - k + b), \alpha/2)] \end{aligned}$$

と書け, Clopper-Pearsonの信頼区間の定義は

$$\begin{aligned}\text{confint}_{\text{CP}}(k|n, p) \\ = [\text{quantile}(\text{Beta}(k, n - k + 1), \alpha/2), \text{cquantile}(\text{Beta}(k + 1, n - k), \alpha/2)]\end{aligned}$$

と書ける。当然のことながら、これらも非常によく似ている。

各 a, b ごとに $k, n - k$ が十分に大きければ次の分布の近似関係が成立している:

$$\begin{aligned}\text{Beta}(k + a, n - k + b) &\approx \text{Beta}(k, n - k + 1), \\ \text{Beta}(k + a, n - k + b) &\approx \text{Beta}(k + 1, n - k).\end{aligned}$$

ゆえにこの近似が成立するとき、

$$\begin{aligned}\text{pvalue}_{\text{ETI}}(k|n, p, a, b) &\approx \text{pvalue}_{\text{CP}}(k|n, p), \\ \text{credint}_{\text{ETI}}(k|n, p, a, b) &\approx \text{confint}_{\text{CP}}(k|n, p).\end{aligned}$$

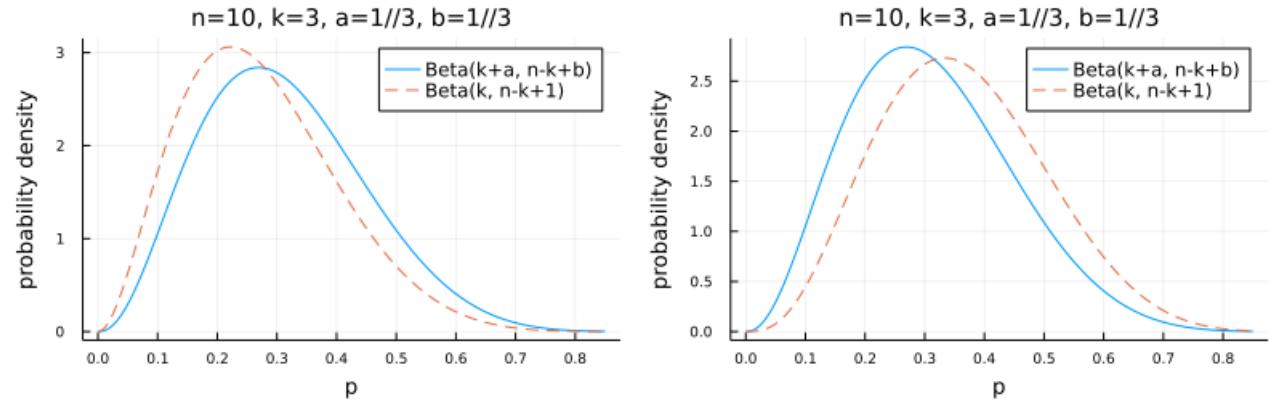
という近似も成立することになる。

In [48]:

```
1 @show n, k = 10, 3
2 @show a, b = 1//3, 1//3
3
4 P1 = plot(Beta(k+a, n-k+b), 0, 0.85; label="Beta(k+a, n-k+b)")
5 plot!(Beta(k, n-k+1), 0, 0.85; label="Beta(k, n-k+1)", ls=:dash)
6 plot!(xguide="p", yguide="probability density")
7 plot!(xtick=0:0.1:1)
8 title!("n=$n, k=$k, a=$a, b=$b")
9
10 P2 = plot(Beta(k+a, n-k+b), 0., 0.85; label="Beta(k+a, n-k+b)")
11 plot!(Beta(k+1, n-k), 0, 0.85; label="Beta(k, n-k+1)", ls=:dash)
12 plot!(xguide="p", yguide="probability density")
13 plot!(xtick=0:0.1:1)
14 title!("n=$n, k=$k, a=$a, b=$b")
15
16 plot(P1, P2; size=(800, 250), leftmargin=4Plots.mm, bottommargin=4Plots.mm)
```

$$\begin{aligned}(n, k) &= (10, 3) = (10, 3) \\ (a, b) &= (1/3, 1/3) = (1/3, 1/3)\end{aligned}$$

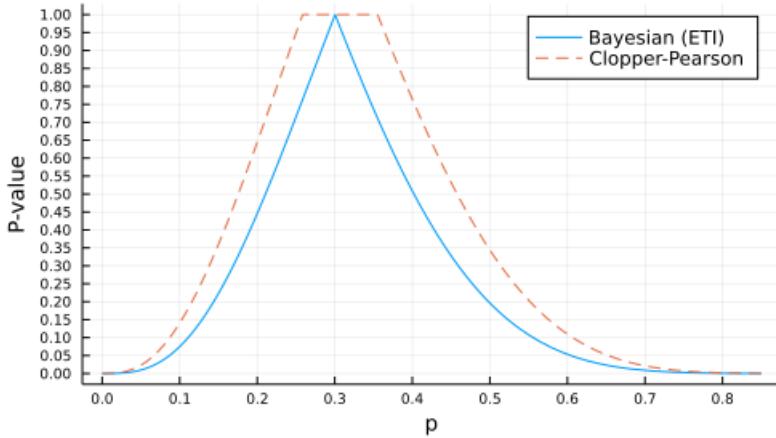
Out[48]:



```
In [49]: 1 @show n, k, α = 10, 3, 0.05
2 @show a, b = 1//3, 1//3
3
4 plot(p → pvalue_eti(k, n, p; a, b), 0, 0.85; label="Bayesian (ETI)")
5 plot!(p → pvalue_cp(k, n, p); label="Clopper-Pearson", 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[49]: P-value functions for $n=10, k=3, a=1/3, b=1/3$



このように、ベイズ版P値函数のETI版のグラフの幅は、Clopper-PearsonのP値函数のグラフの幅よりも狭くなる。

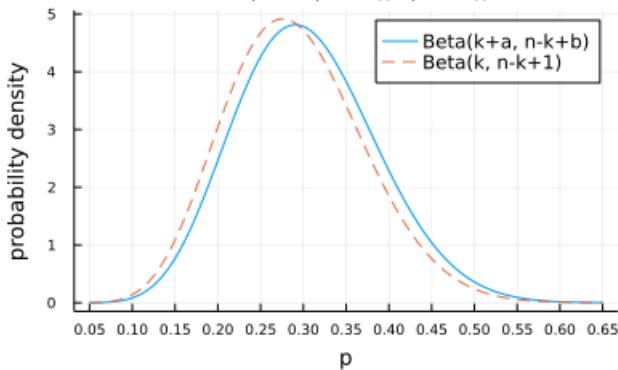
つまり、ベイズ版P値函数のETI版は、「幅を縮める変更」をClopper-PearsonのP値函数に施して得られたP値函数であると考えることができる。

n を大きくするとそれらはほぼ一致する。

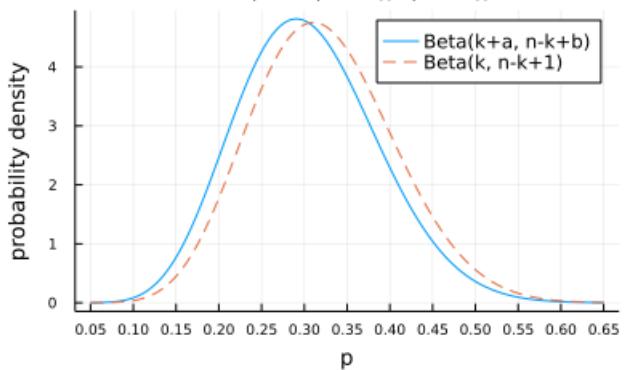
```
In [50]: 1 @show n, k = 30, 9
2 @show a, b = 1//3, 1//3
3
4 P1 = plot(Beta(k+a, n-k+b), 0.05, 0.65; label="Beta(k+a, n-k+b)")
5 plot!(Beta(k, n-k+1), 0.05, 0.65; label="Beta(k, n-k+1)", ls=:dash)
6 plot!(xguide="p", yguide="probability density")
7 plot!(xtick=0:0.05:1)
8 title!("n=$n, k=$k, a=$a, b=$b")
9
10 P2 = plot(Beta(k+a, n-k+b), 0.05, 0.65; label="Beta(k+a, n-k+b)")
11 plot!(Beta(k+1, n-k), 0.05, 0.65; label="Beta(k, n-k+1)", ls=:dash)
12 plot!(xguide="p", yguide="probability density")
13 plot!(xtick=0:0.05:1)
14 title!("n=$n, k=$k, a=$a, b=$b")
15
16 plot(P1, P2; size=(800, 250), leftmargin=4Plots.mm, bottommargin=4Plots.mm)
```

$(n, k) = (30, 9) = (30, 9)$
 $(a, b) = (1 // 3, 1 // 3) = (1/3, 1/3)$

Out[50]: n=30, k=9, a=1/3, b=1/3



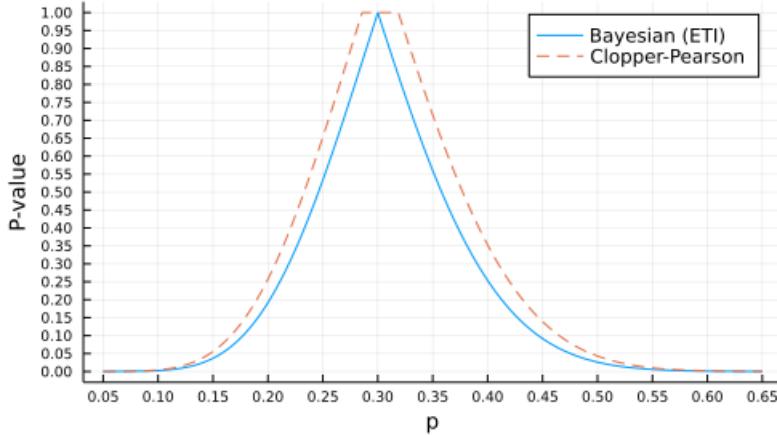
n=30, k=9, a=1/3, b=1/3



```
In [51]: 1 @show n, k, α = 30, 9, 0.05
2 @show a, b = 1//3, 1//3
3
4 plot(p → pvalue_eti(k, n, p; a, b), 0.05, 0.65; label="Bayesian (ETI)")
5 plot!(p → pvalue_cp(k, n, p); label="Clopper-Pearson", 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")
```

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

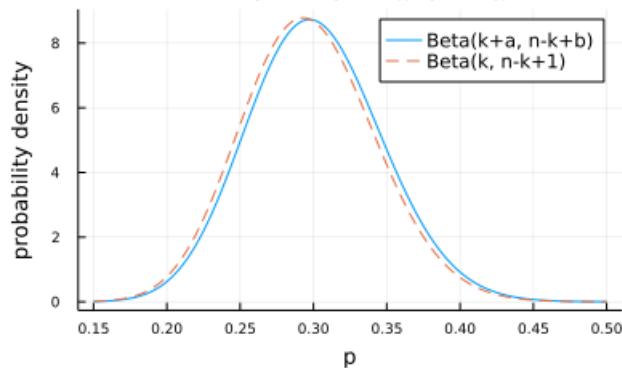
Out[51]: P-value functions for n=30, k=9, a=1/3, b=1/3



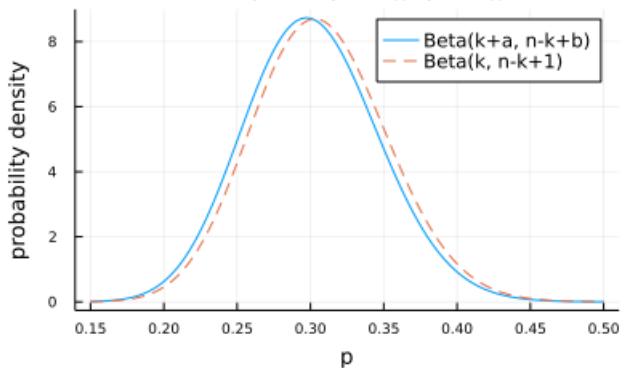
```
In [52]: 1 @show n, k = 100, 30
2 @show a, b = 1//3, 1//3
3
4 P1 = plot(Beta(k+a, n-k+b), 0.15, 0.5; label="Beta(k+a, n-k+b)")
5 plot!(Beta(k, n-k+1), 0.15, 0.5; label="Beta(k, n-k+1)", ls=:dash)
6 plot!(xguide="p", yguide="probability density")
7 plot!(xtick=0:0.05:1)
8 title!("n=$n, k=$k, a=$a, b=$b")
9
10 P2 = plot(Beta(k+a, n-k+b), 0.15, 0.5; label="Beta(k+a, n-k+b)")
11 plot!(Beta(k+1, n-k), 0.15, 0.5; label="Beta(k, n-k+1)", ls=:dash)
12 plot!(xguide="p", yguide="probability density")
13 plot!(xtick=0:0.05:1)
14 title!("n=$n, k=$k, a=$a, b=$b")
15
16 plot(P1, P2; size=(800, 250), leftmargin=4Plots.mm, bottommargin=4Plots.mm)
```

$(n, k) = (100, 30) = (100, 30)$
 $(a, b) = (1 // 3, 1 // 3) = (1/3, 1/3)$

Out[52]: n=100, k=30, a=1/3, b=1/3



n=100, k=30, a=1/3, b=1/3

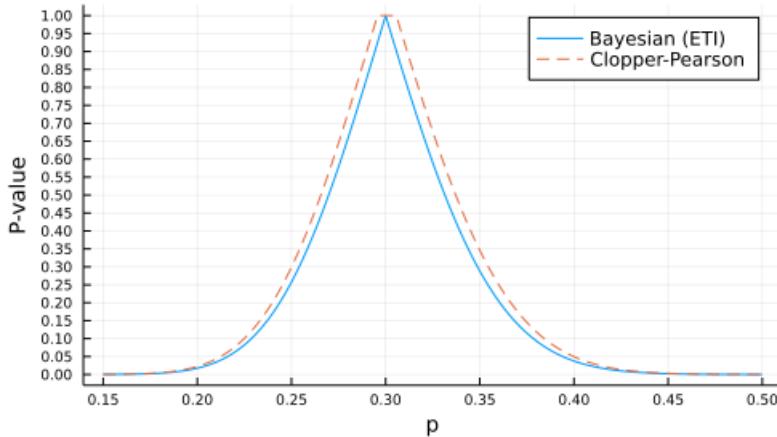


```
In [53]: 1 @show n, k, α = 100, 30, 0.05
2 @show a, b = 1//3, 1//3
3
4 plot(p → pvalue_eti(k, n, p; a, b), 0.15, 0.5; label="Bayesian (ETI)")
5 plot!(p → pvalue_cp(k, n, p); label="Clopper-Pearson", ls=:dash)
6 plot!(xguide="p", yguide="P-value")
7 plot!(xtick=0:0.05:1, ytick=0:0.05:1)
8 title!("P-value functions for n=$n, k=$k, a=$a, b=$b")
```

$$(n, k, \alpha) = (100, 30, 0.05) = (100, 30, 0.05)$$

$$(a, b) = (1 // 3, 1 // 3) = (1/3, 1/3)$$

Out[53]: P-value functions for n=100, k=30, a=1/3, b=1/3

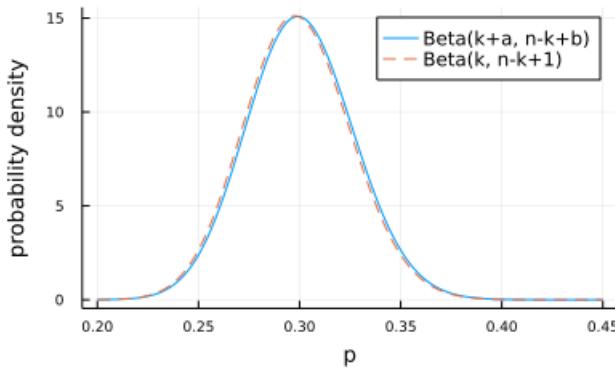


```
In [54]: 1 @show n, k = 300, 90
2 @show a, b = 1//3, 1//3
3
4 P1 = plot(Beta(k+a, n-k+b), 0.2, 0.45; label="Beta(k+a, n-k+b)")
5 plot!(Beta(k, n-k+1), 0.2, 0.45; label="Beta(k, n-k+1)", ls=:dash)
6 plot!(xguide="p", yguide="probability density")
7 plot!(xtick=0:0.05:1)
8 title!("n=$n, k=$k, a=$a, b=$b")
9
10 P2 = plot(Beta(k+a, n-k+b), 0.15, 0.45; label="Beta(k+a, n-k+b)")
11 plot!(Beta(k+1, n-k), 0.15, 0.5; label="Beta(k, n-k+1)", ls=:dash)
12 plot!(xguide="p", yguide="probability density")
13 plot!(xtick=0:0.05:1)
14 title!("n=$n, k=$k, a=$a, b=$b")
15
16 plot(P1, P2; size=(800, 250), leftmargin=4Plots.mm, bottommargin=4Plots.mm)
```

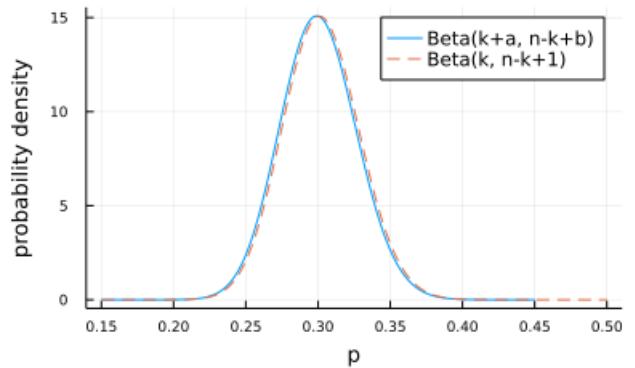
$$(n, k) = (300, 90) = (300, 90)$$

$$(a, b) = (1 // 3, 1 // 3) = (1/3, 1/3)$$

Out[54]: n=300, k=90, a=1/3, b=1/3



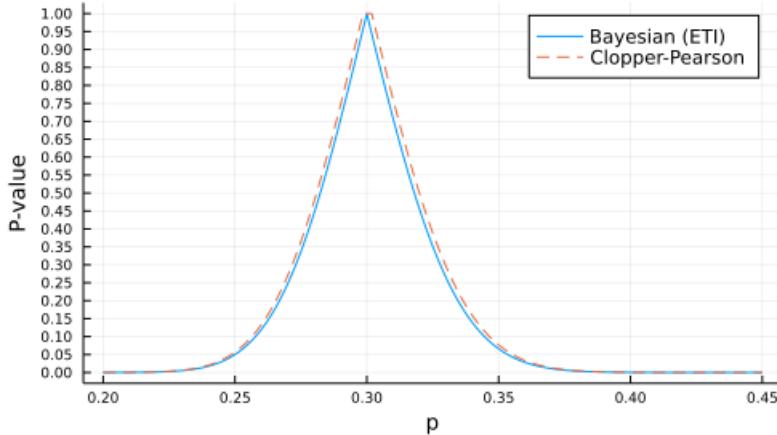
n=300, k=90, a=1/3, b=1/3



```
In [55]: 1 @show n, k, α = 300, 90, 0.05
2 @show a, b = 1//3, 1//3
3
4 plot(p → pvalue_eti(k, n, p; a, b), 0.2, 0.45; label="Bayesian (ETI)")
5 plot!(p → pvalue_cp(k, n, p); label="Clopper-Pearson", 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")
```

$(n, k, \alpha) = (300, 90, 0.05) = (300, 90, 0.05)$
 $(a, b) = (1 // 3, 1 // 3) = (1/3, 1/3)$

Out[55]: P-value functions for n=300, k=90, a=1/3, b=1/3



10.9 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とは、「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、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。

11 被覆確率の比較

二項分布 $\text{Binomial}(n, p)$ に従って生成されたデータから計算された信頼度 $1 - \alpha$ の信頼区間に p が含まれる確率を信頼区間の被覆確率 (coverage probability) と呼ぶ。

信頼区間の定義

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

より、被覆確率は二項分布 $\text{Binomial}(n, p)$ に従って生成されたデータから計算された「当たりが出る確率は p である」という仮説のP値が α 以上になる確率に一致する。

その確率は複数のP値函数について計算して比較してみよう。

おおまかな傾向は以下の通り:

- Clopper-Pearsonの信頼区間の被覆確率は $1 - \alpha$ より過剰に大きくなる傾向がある。
- Sterneの信頼区間ではその点が改善される。
- Waldの信頼区間の被覆確率は過剰に小さくなりやすい。
- Wilsonの信頼区間の被覆確率は p の値を動かすと $1 - \alpha$ を中心に上限に振動する。
- 無情報事前分布の場合のベイズ版信用区間の被覆確率の様子はWilsonの信頼区間に近い。

In [56]:

```
1 x ≈ y = x > y || x ≈ y
2
3 function coverprob(pvaluefunc, n, p, α)
4     bin = Binomial(n, p)
5     sum(support(bin)) do k
6         (pvaluefunc(k, n, p) ≈ α) * pdf(bin, k)
7     end
8 end
```

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

In [57]:

```

1 function plot_coverprob(n, α;
2     prior_hdi = (1, 1), prior_eti = (1//3, 1//3),
3     ylim = (1-2α, 1), lw = 1, kwargs...)
4     @show n, α
5     p = range(0, 1, 2001)
6
7     CP_Wilson = plot(; ylim, xtick=0:0.1:1, ytick=0:0.01:1)
8     plot!(p, p → coverprob(pvalue_wilson, n, p, α); label="", lw)
9     hline!([1-α]; label="", ls=:dot)
10    plot!(xguide="p")
11    title!("Wilson: n=$n, α=$α")
12
13    CP_Wald = plot(; ylim, xtick=0:0.1:1, ytick=0:0.01:1)
14    plot!(p, p → coverprob(pvalue_wald, n, p, α); label="", lw)
15    hline!([1-α]; label="", ls=:dot)
16    plot!(xguide="p")
17    title!("Wald: n=$n, α=$α")
18
19    CP_CP = plot(; ylim, xtick=0:0.1:1, ytick=0:0.01:1)
20    plot!(p, p → coverprob(pvalue_cp, n, p, α); label="", lw)
21    hline!([1-α]; label="", ls=:dot)
22    plot!(xguide="p")
23    title!("Clopper-Pearson: n=$n, α=$α")
24
25    CP_Sterne = plot(; ylim, xtick=0:0.1:1, ytick=0:0.01:1)
26    plot!(p, p → coverprob(pvalue_sterne, n, p, α); label="", lw)
27    hline!([1-α]; label="", ls=:dot)
28    plot!(xguide="p")
29    title!("Sterne: n=$n, α=$α")
30
31    a, b = prior_hdi
32    CP_HDI = plot(; ylim, xtick=0:0.1:1, ytick=0:0.01:1)
33    plot!(p, p → coverprob((k,n,p)→pvalue_hdi(k,n,p; a,b), n, p, α); label="", lw)
34    hline!([1-α]; label="", ls=:dot)
35    plot!(xguide="p")
36    title!("Bayesian HDI: prior=(a, b), n=$n, α=$α")
37
38    a, b = prior_eti
39    CP_ETI = plot(; ylim, xtick=0:0.1:1, ytick=0:0.01:1)
40    plot!(p, p → coverprob((k,n,p)→pvalue_eti(k,n,p; a,b), n, p, α); label="", lw)
41    hline!([1-α]; label="", ls=:dot)
42    plot!(xguide="p")
43    title!("Bayesian ETI: prior=(a, b), n=$n, α=$α")
44
45    plot(CP_CP, CP_Sterne, CP_Wilson, CP_Wald, CP_HDI, CP_ETI;
46          size=(800, 1000), layout=(3, 2))
47    plot!(plot_title="coverage probabilities")
48    plot!(; kwargs...)
49 end

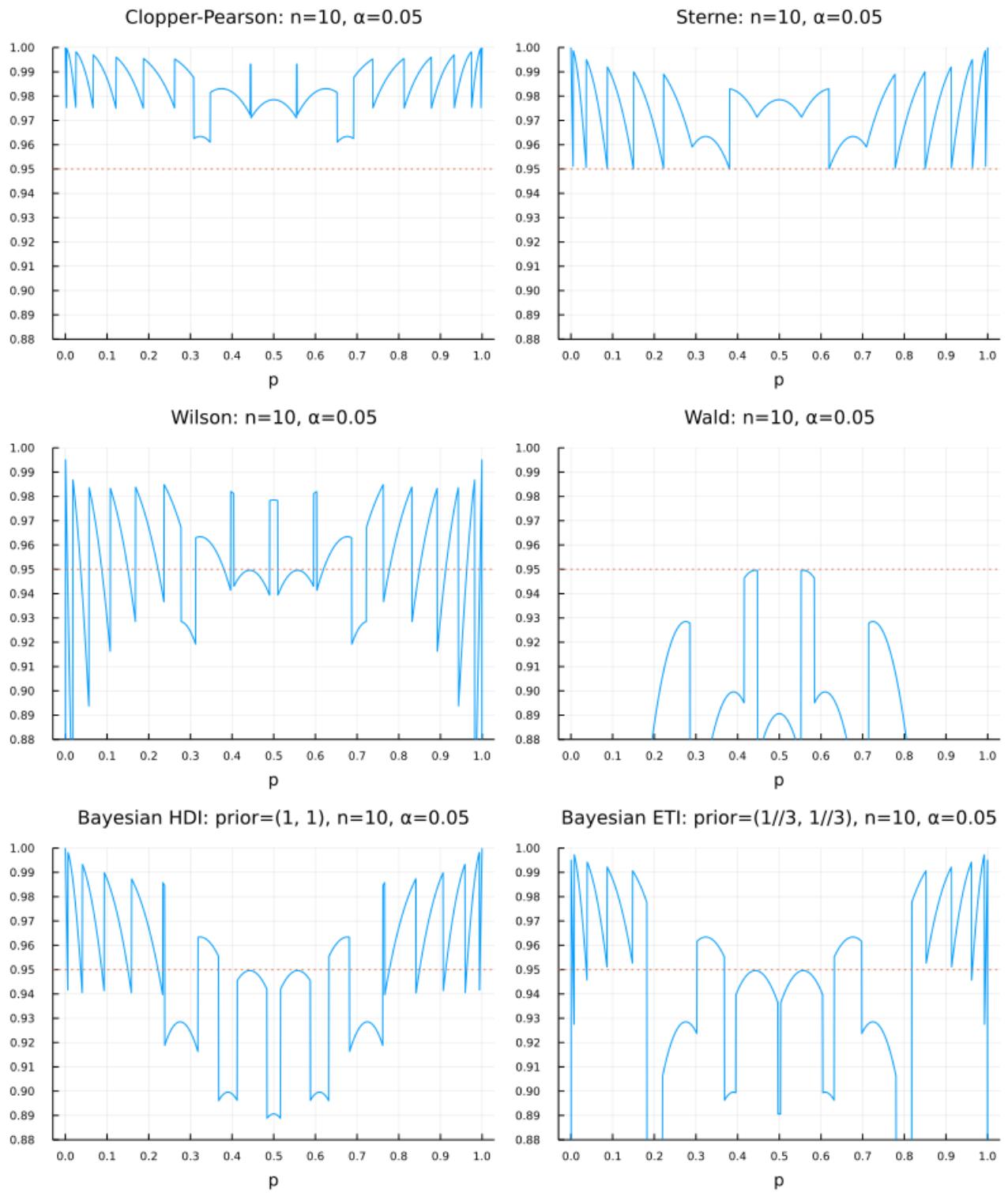
```

Out[57]: plot_coverprob (generic function with 1 method)

```
In [58]: 1 plot_coverprob(10, 0.05; ylim=(0.88, 1))
          (n, α) = (10, 0.05)
```

Out[58]:

coverage probabilities



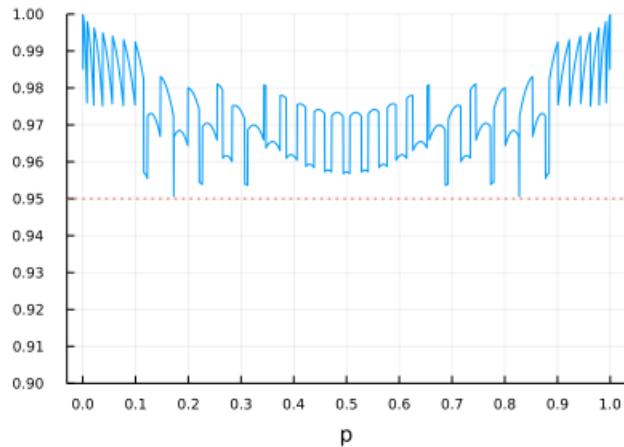
```
In [59]: 1 plot_coverprob(30, 0.05)
```

```
(n, α) = (30, 0.05)
```

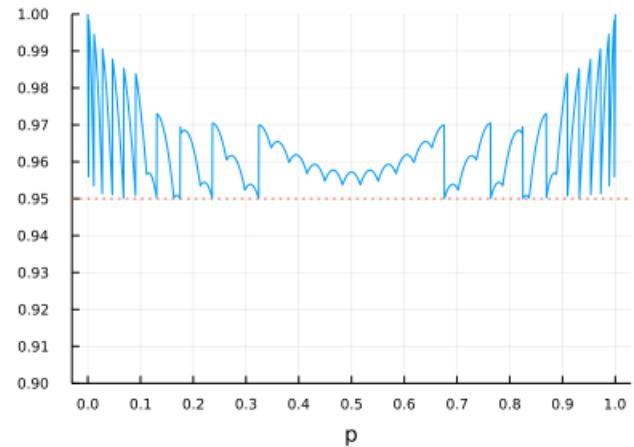
Out[59]:

coverage probabilities

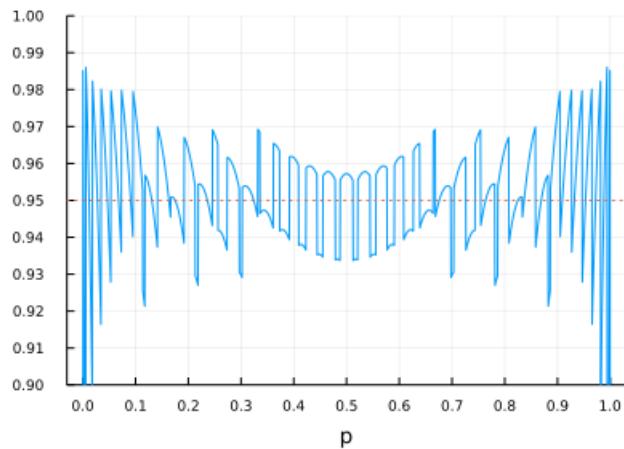
Clopper-Pearson: n=30, $\alpha=0.05$



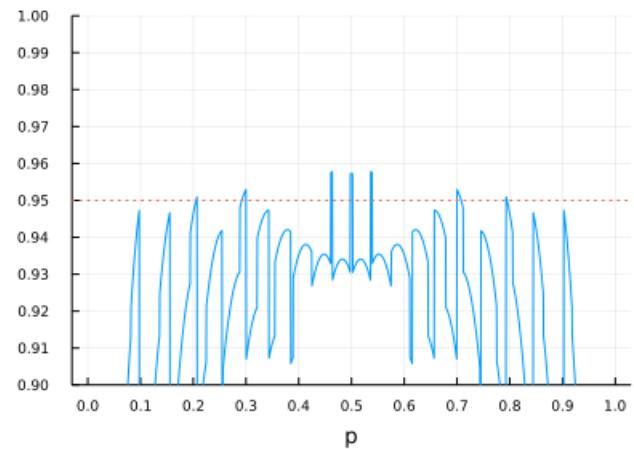
Sterne: n=30, $\alpha=0.05$



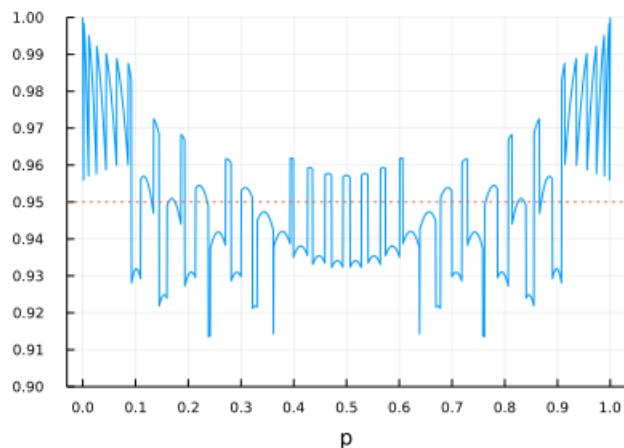
Wilson: n=30, $\alpha=0.05$



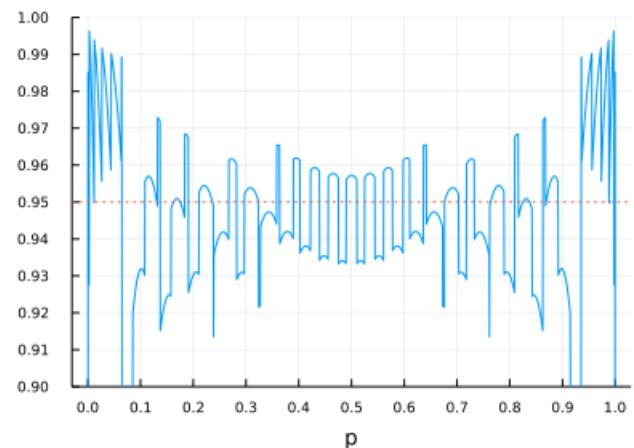
Wald: n=30, $\alpha=0.05$



Bayesian HDI: prior=(1, 1), n=30, $\alpha=0.05$



Bayesian ETI: prior=(1/3, 1/3), n=30, $\alpha=0.05$

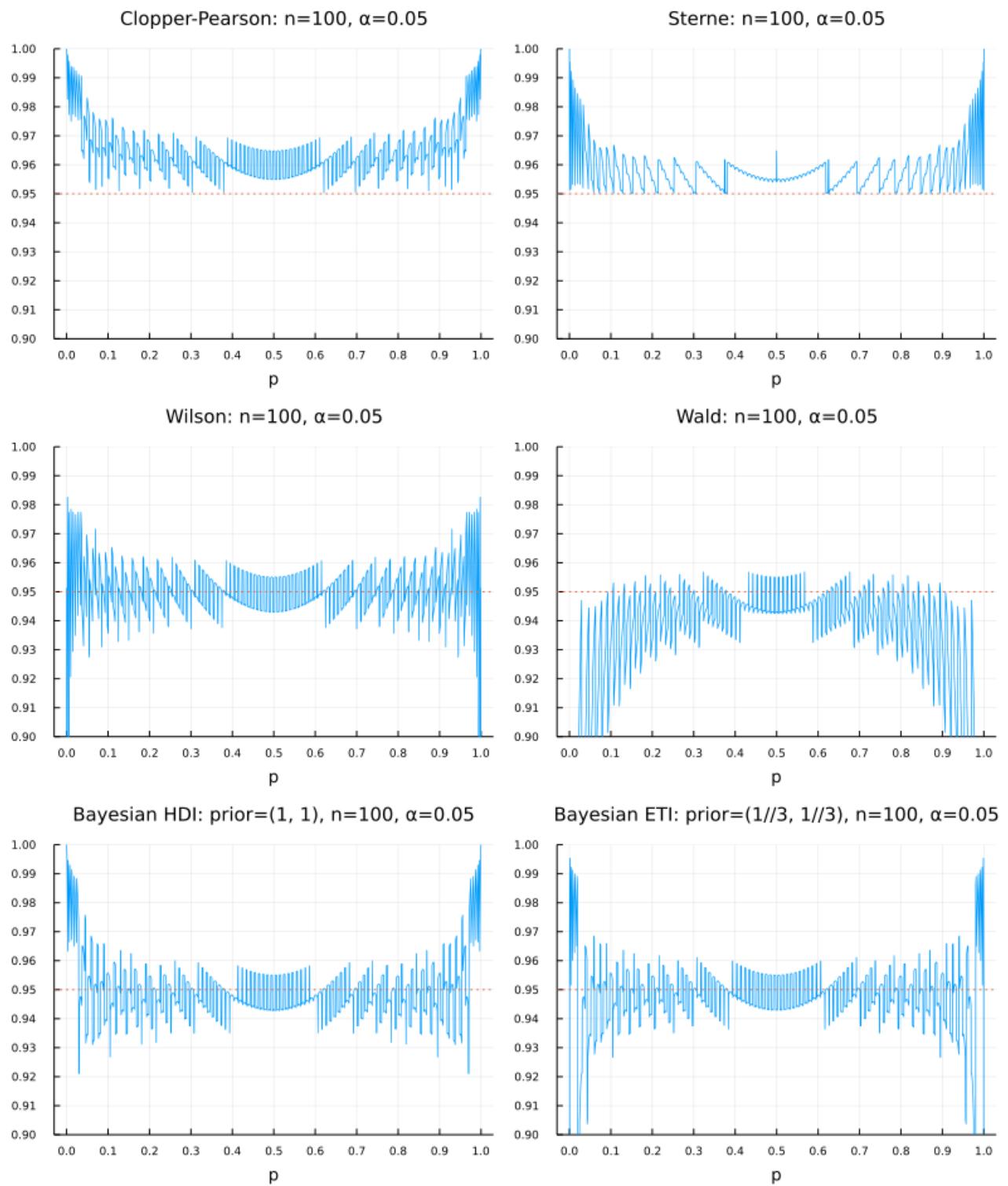


```
In [60]: 1 plot_coverprob(100, 0.05; lw=0.7)
```

```
(n, α) = (100, 0.05)
```

Out[60]:

coverage probabilities

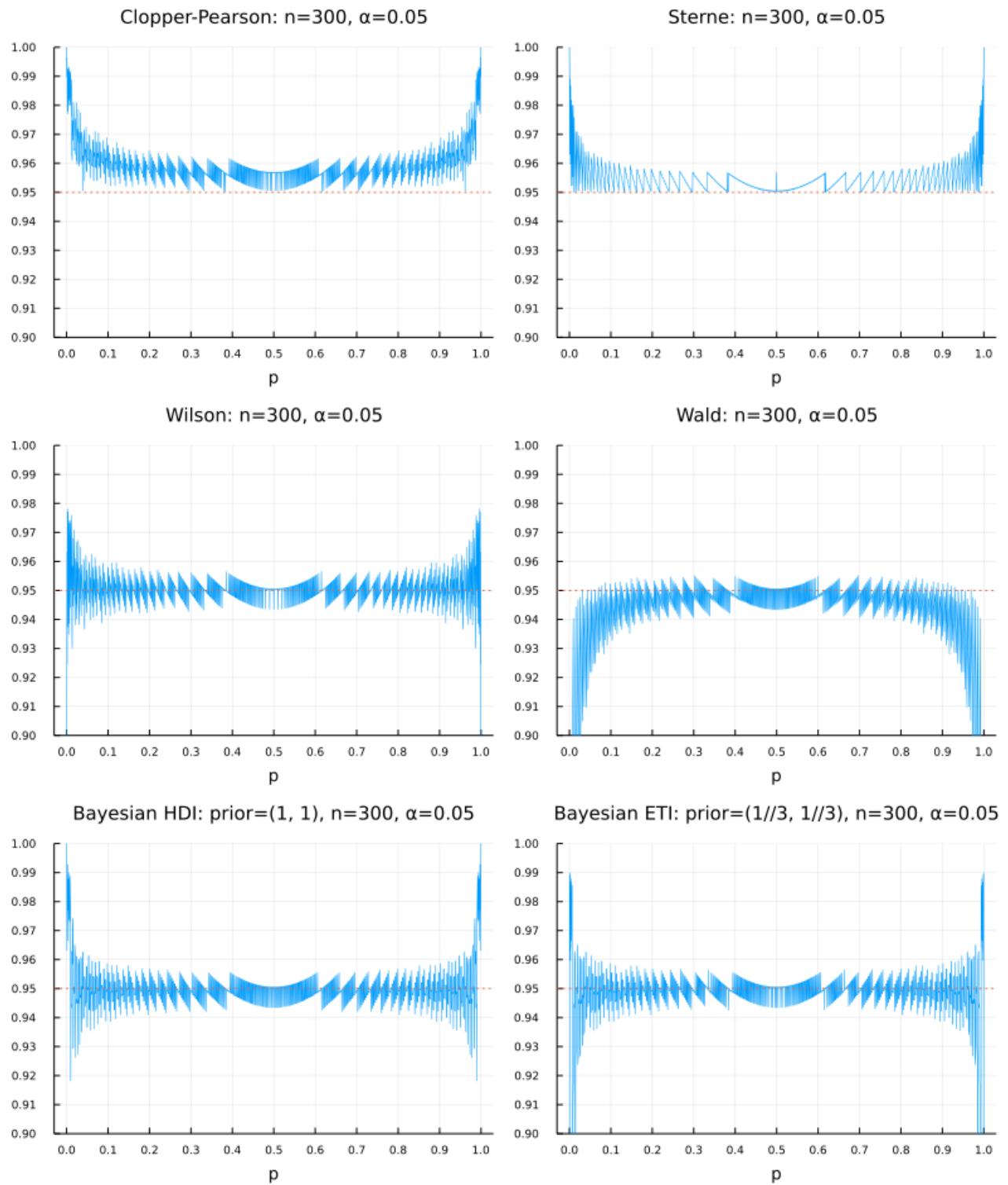


```
In [61]: 1 plot_coverprob(300, 0.05; lw=0.5)
```

```
(n, α) = (300, 0.05)
```

Out[61]:

coverage probabilities



複数のP値函数について、データ k が二項分布 $\text{Binomial}(n, p)$ に従ってランダムに生成されるという仮定の下で、そのデータから計算された P 値の確率分布の累積分布函数すなわち α にP値が α 以下になる確率を対応させる函数を比較してみよう。

P 値の累積分布函数の値は名目有意水準 α で第一種の過誤が起こる確率であると考えられる。

第一種の過誤が起こる確率は名目有意水準 α に近いほどよく、第一種の過誤が起こる確率は名目有意水準 α をできるだけ上回らない方がよい。

大まかな傾向として以下のようになることを確認できる:

- 小さな n で Wald の P 値での第一種の過誤が起こる確率は大きくなり過ぎる場合がある。
- Wilson の P 値の累積分布函数のグラフは斜め45度線の上下を振動し、斜め45度線に近くなる(好ましい性質)。
- Clopper-Pearson の P 値と Sterne の P 値には、第一種の過誤が起こる確率が確実に有意水準以下になるという利点がある。
- Clopper-Pearson の P 値では第一種の過誤が起こる確率が過剰に小さくなり過ぎる。
- Sterne の P 値ではその問題が緩和される。

```
In [62]: 1 x ≈ y = x < y || x ≈ y
2
3 function cdfpval(pvaluefunc, n, p, α)
4     bin = Binomial(n, p)
5     sum(support(bin)) do k
6         (pvaluefunc(k, n, p) ≈ α) * pdf(bin, k)
7     end
8 end
```

Out[62]: cdfpval (generic function with 1 method)

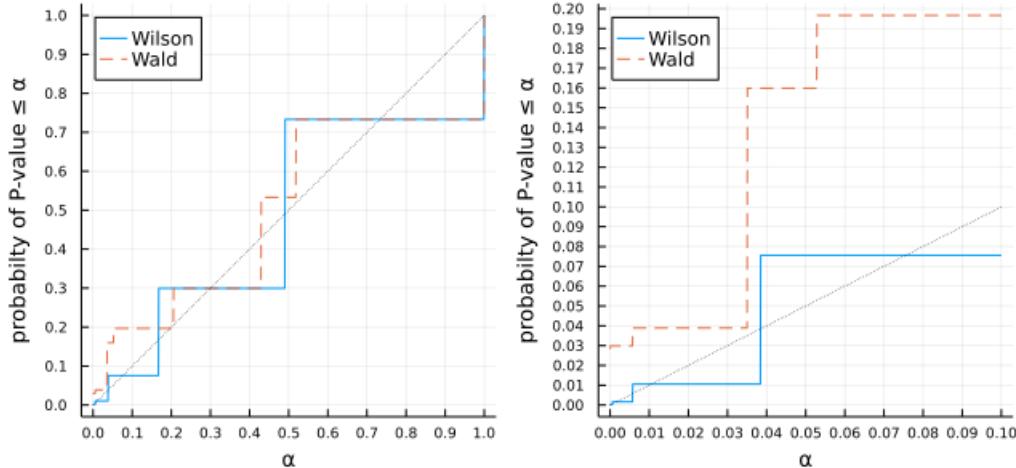
```
In [63]: 1 function plot_cdfpval(n, p; f=Bool[1,1,1,1])
2     α = range(0, 1, 1001)
3     P1 = plot(legend=:topleft)
4     f[1] && plot!(α, α→cdfpval(pvalue_wilson, n, p, α); label="Wilson", c=1)
5     f[2] && plot!(α, α→cdfpval(pvalue_wald, n, p, α); label="Wald", ls=:dash, c=2)
6     f[3] && plot!(α, α→cdfpval(pvalue_cp, n, p, α); label="Clopper-Pearson", ls=:dot, lw=1.3,
7     f[4] && plot!(α, α→cdfpval(pvalue_sterne, n, p, α); label="Sterne", ls=:dashdot, lw=1.3,
8     plot!([0,1], [0,1]; label="", c=:black, ls=:dot, lw=0.5)
9     plot!(xtick=0:0.1:1, ytick=0:0.1:1)
10    plot!(xguide="α", yguide="probabilty of P-value ≤ α")
11
12    α = range(0, 0.1, 1001)
13    P2 = plot(legend=:topleft)
14    f[1] && plot!(α, α→cdfpval(pvalue_wilson, n, p, α); label="Wilson", c=1)
15    f[2] && plot!(α, α→cdfpval(pvalue_wald, n, p, α); label="Wald", ls=:dash, c=2)
16    f[3] && plot!(α, α→cdfpval(pvalue_cp, n, p, α); label="Clopper-Pearson", ls=:dot, lw=1.3,
17    f[4] && plot!(α, α→cdfpval(pvalue_sterne, n, p, α); label="Sterne", ls=:dashdot, lw=1.3,
18    plot!([0,0.1], [0,0.1]; label="", c=:black, ls=:dot, lw=0.5)
19    plot!(xtick=0:0.01:1, ytick=0:0.01:1)
20    plot!(xguide="α", yguide="probabilty of P-value ≤ α")
21
22    plot!(P1, P2; size=(640, 320))
23    plot!(plot_title="Assumption: data is generated by Binomial($n, $p).")
24 end
```

Out[63]: plot_cdfpval (generic function with 1 method)

```
In [64]: 1 n, p = 10, 0.3
2 plot_cdfpval(n, p; f=Bool[1,1,0,0])
```

Out[64]:

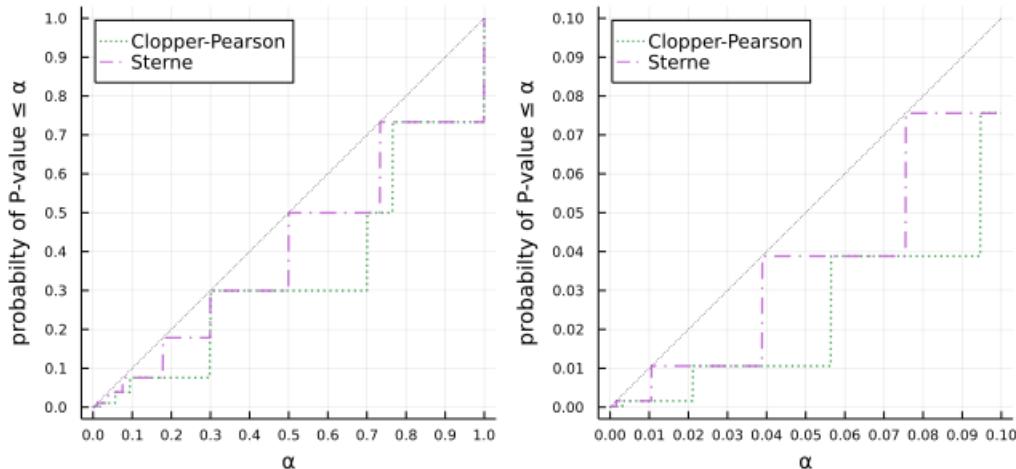
Assumption: data is generated by Binomial(10, 0.3).



```
In [65]: 1 plot_cdfpval(n, p; f=Bool[0,0,1,1])
```

Out[65]:

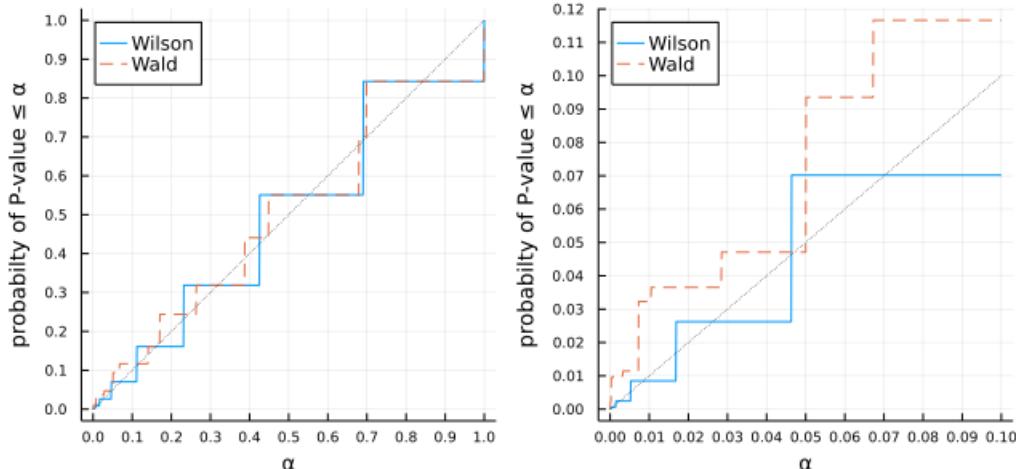
Assumption: data is generated by Binomial(10, 0.3).



```
In [66]: 1 n, p = 30, 0.3
2 plot_cdfpval(n, p; f=Bool[1,1,0,0])
```

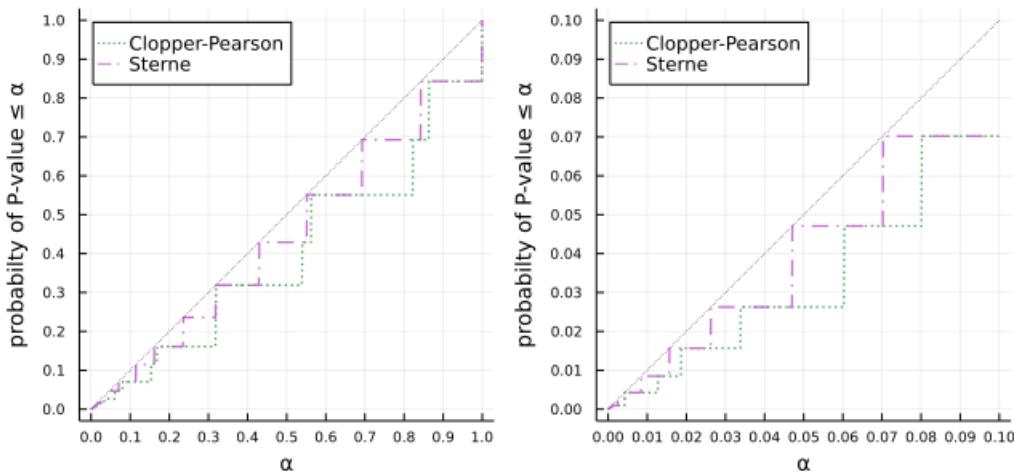
Out[66]:

Assumption: data is generated by Binomial(30, 0.3).



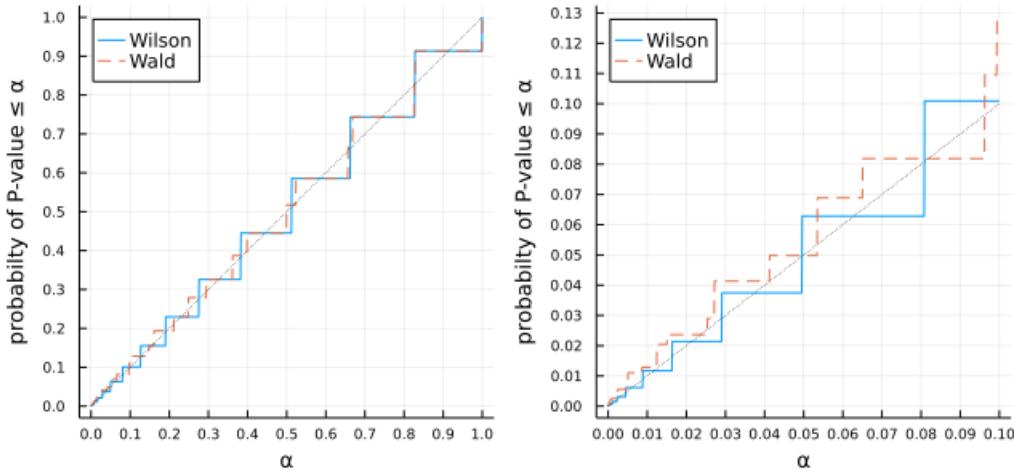
```
In [67]: 1 plot_cdfpval(n, p; f=Bool[0,0,1,1])
```

Out[67]: Assumption: data is generated by Binomial(30, 0.3).



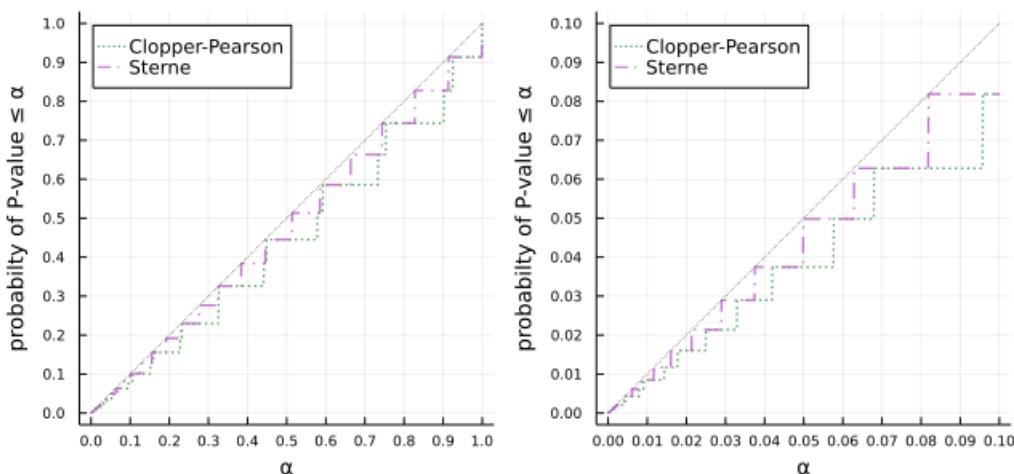
```
In [68]: 1 n, p = 100, 0.3  
2 plot_cdfpval(n, p; f=Bool[1,1,0,0])
```

Out[68]: Assumption: data is generated by Binomial(100, 0.3).



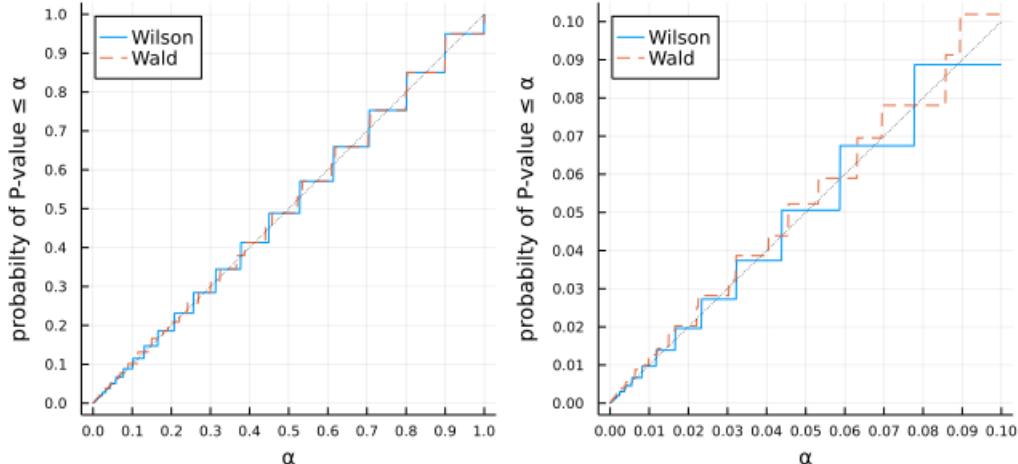
```
In [69]: 1 plot_cdfpval(n, p; f=Bool[0,0,1,1])
```

Out[69]: Assumption: data is generated by Binomial(100, 0.3).



```
In [70]: 1 n, p = 300, 0.3  
2 plot_cdfpval(n, p; f=Bool[1,1,0,0])
```

Out[70]: Assumption: data is generated by Binomial(300, 0.3).



```
In [71]: 1 plot_cdfpval(n, p; f=Bool[0,0,1,1])
```

Out[71]: Assumption: data is generated by Binomial(300, 0.3).

