

# まとめ

- 黒木玄
- 2022-07-20~2022-07-22

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

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

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

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

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

## 目次

- ▼ [1 二項分布でのClopper-Pearsonの信頼区間](#)
  - [1.1 Clopper-Pearsonの信頼区間を例に信頼区間の解釈の仕方について説明](#)
  - [1.2 二項分布モデルのClopper-Pearsonの信頼区間の効率的な計算の仕方](#)
  - [1.3 二項分布モデルのBayes統計](#)
  - [1.4 P値とBayes統計の関係](#)
- ▼ [1.5 復習: 二項分布とベータ分布の関係の証明](#)
  - [1.5.1 両辺を  \$p\$  で微分して確認する方法](#)
  - [1.5.2 ベータ分布の一様分布の順序統計量の分布として解釈を使う方法](#)
- ▼ [2 P値関数について](#)
  - [2.1 P値はモデルのパラメータ値とデータの数値の整合性の指標の1つとして定義される](#)
  - [2.2 P値は信頼区間を考えたいすべてのパラメータ値について定義されている](#)
  - [2.3 分割表の場合のP値関数](#)
  - [2.4 P値関数から信頼区間が定義される](#)
  - [2.5 2×2の分割表のP値と信頼区間とP値関数の例](#)
  - [2.6 2×2の分割表での検定ではどれを使うべきか](#)
  - [2.7 P値関数と最尤法の関係](#)
- ▼ [3 Welchの t 検定について](#)
  - [3.1 Welchの t 検定のP値と信頼区間の定義](#)

```
In [1]: 1 ENV["LINES"], ENV["COLUMNS"] = 100, 100
2 using Base.Threads
3 using BenchmarkTools
4 using DataFrames
5 using Distributions
6 using LinearAlgebra
7 using Memoization
8 using Optim
9 using Printf
10 using QuadGK
11 using RCall
12 @rimport stats as R
13 using Random
14 Random.seed!(4649373)
15 using Roots
16 using SpecialFunctions
17 using StaticArrays
18 using StatsBase
19 using StatsFuns
20 using StatsPlots
21 default(fmt = :png, size = (400, 250),
22         titlefontsize = 10, guidefontsize=9, plot_titlefontsize = 12)
23 using SymPy
```

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

```
In [3]: 1 safemul(x, y) = x == 0 ? x : isinf(x) ? typeof(x)(Inf) : x*y
2 safediv(x, y) = x == 0 ? x : isinf(y) ? zero(y) : x/y
3
4 x ≲ y = x < y || x ≈ y
5
6 mypdf(dist, x) = pdf(dist, x)
7 mypdf(dist::DiscreteUnivariateDistribution, x) = pdf(dist, round(Int, x))
8
9 distname(dist::Distribution) = replace(string(dist), r"{.*}" => "")
10 myskewness(dist) = skewness(dist)
11 mykurtosis(dist) = kurtosis(dist)
12 function standardized_moment(dist::ContinuousUnivariateDistribution, m)
13     μ, σ = mean(dist), std(dist)
14     quadgk(x → (x - μ)^m * pdf(dist, x), extrema(dist)...)[1] / σ^m
15 end
16 myskewness(dist::MixtureModel{Univariate, Continuous}) =
17     standardized_moment(dist, 3)
18 mykurtosis(dist::MixtureModel{Univariate, Continuous}) =
19     standardized_moment(dist, 4) - 3
```

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

```
In [4]: 1 function logtick(; xlim=(0.03, 30))
2     xmin, xmax = xlim
3     a = floor(Int, log10(xmin))
4     b = ceil(Int, log10(xmax))
5     nums = [1, 2, 3, 4, 5, 6, 7, 8, 9]
6     mask = Bool[1, 1, 0, 0, 1, 0, 0, 0, 0]
7
8     logtick = foldl(vcat, ([10.0^k*x for x in nums if xmin ≤ 10.0^k*x ≤ xmax] for k in a:b))
9     logticklabel_a = foldl(vcat,
10         ([mask[i] ? string(round(10.0^k*x; digits=-k)) : ""
11             for (i, x) in enumerate(nums) if xmin ≤ 10.0^k*x ≤ xmax]
12             for k in a:-1))
13     logticklabel_b = foldl(vcat,
14         ([mask[i] ? string(10^k*x) : ""
15             for (i, x) in enumerate(nums) if xmin ≤ 10.0^k*x ≤ xmax]
16             for k in 0:b))
17     logticklabel = vcat(logticklabel_a, logticklabel_b)
18     (logtick, logticklabel)
19 end
20
21 # logtick()
```

Out[4]: logtick (generic function with 1 method)

## 1 二項分布でのClopper-Pearsonの信頼区間

### 1.1 Clopper-Pearsonの信頼区間を例に信頼区間の解釈の仕方について説明

「 $n$  回中  $k$  回当たりが出た」「 $n$  人中  $k$  人が重症化」「 $n$  人中  $k$  人が商品を購入」などの型のデータを扱うための統計モデルとして、成功確率パラメータ  $p$  を持つ二項分布

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

を使う状況を考える。このとき、仮説  $p = p_0$  のP値(データの数値とモデル+パラメータ値の整合性の指標の1つ)を次のように定義できるのであった:

$$\text{pvalue}_{\text{CP}}(k|n, p = p_0) = \min \left( \frac{1}{2 \text{cdf}(\text{Binomial}(n, p_0), k)}, \frac{1}{2(1 - \text{cdf}(\text{Binomial}(n, p_0), k - 1))} \right)$$

これをClopper-PearsonのP値と呼ぶことにしていた。このP値は以下の3つのうちの最小値として定義されている:

- 1
- 仮説  $p = p_0$  の下での二項分布で  $n$  回中の当たりの回数が  $k$  回以下の確率の2倍
- 仮説  $p = p_0$  の下での二項分布で  $n$  回中の当たりの回数が  $k$  回以上の確率の2倍

本当はモデル内で仮想的に生成されたデータの値  $i$  ( $n$  回中の当たりの回数)が期待値  $np_0$  からデータの数値  $k$  以上離れる確率をP値としたいのだが(P値の定義は仮説下のモデル内でデータの数値以上に極端な値が生成される確率またはその近似値であった)、反対側の確率を自然に決めることができないので、2倍して反対側の確率も足し上げたことにしている。(これは1つの処方箋に過ぎず、別の方法もあるのであった。)

これに対応する信頼度  $1 - \alpha$  の信頼区間の定義は次の通り:

$$\text{confint}_{\text{CP}}(k|n, \alpha) = \{p_0 \in [0, 1] \mid \text{pvalue}_{\text{CP}}(k|n, p = p_0) \geq \alpha\}.$$

仮説  $p = p_0$  のP値  $\text{pvalue}_{\text{CP}}(k|n, p = p_0)$  はデータの数値「 $n$  回中  $k$  回」とモデルのパラメータに関する仮説  $p = p_0$  の整合性の指標として使われるのであった。

信頼区間を定義している条件  $\text{pvalue}_{\text{CP}}(k|n, p = p_0) \geq \alpha$  はその整合性の指標の値がある閾値  $\alpha$  (この閾値は有意水準と呼ばれる)以上になっているという条件になっている。

通常、有意水準  $\alpha$  は目的に合わせて小さな値を採用し、検定の手続きではP値が  $\alpha$  未満になるパラメータ値  $p = p_0$  を棄却する。

信頼度  $1 - \alpha$  信頼区間は有意水準  $\alpha$  で棄却されないパラメータ値  $p = p_0$  全体の集合になっている。

「棄却されないこと」は「判断を保留すること」を意味する。

信頼区間に含まれるパラメータ値  $p = p_0$  については判断を保留することになるが、もしも信頼区間が十分に狭くなっていけば、その外側にある大部分のパラメータ値は検定の手続きによって棄却されており、可能性として考慮すればよいのは、十分に狭くなっている信頼区間に含まれるパラメータ値だけになっているので、科学的に十分に意味のある結論を出すことができる場合が出て来る。

ただし、信頼区間を使って科学的に信頼できる結論を出すためには、データの取得法が信頼できるか、採用した統計モデルは妥当であるか、その他見落としている重要な問題はないか、などの多くの問題をクリアする必要がある。

機械的に信頼区間を計算して科学的な御墨付きが得られたような態度を取ることは誤りなので注意して欲しい。

## 1.2 二項分布モデルのClopper-Pearsonの信頼区間の効率的な計算の仕方

Clopper-Pearsonの信頼区間は定義をうまく変形すると以下のように書き直される:

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

ここで、 $p_L, p_U$  は次の条件で特徴付けられる値である:

$$\begin{aligned} 1 - \text{cdf}(\text{Binomial}(n, p_L), k - 1) &= \alpha/2, \\ \text{cdf}(\text{Binomial}(n, p_U), k) &= \alpha/2. \end{aligned}$$

すなわち、小さい方の  $p_L$  は  $p = p_L$  の二項分布内で  $n$  回中の当たりの回数が  $k$  以上になる確率が  $\alpha/2$  になるという条件で特徴付けられ、大きい方の  $p_U$  は  $p = p_U$  の二項分布内で  $n$  回中当たりの回数が  $k$  回以下になる確率が  $\alpha/2$  になるという条件で特徴付けられる。

問題はこの  $p_L, p_U$  をどのように計算するかである。

これには二項分布とベータ分布の間の関係を使う驚くべき方法がある。

二項分布の累積分布函数(cumulative distribution function, cdf)の定義は

$$\text{cdf}(\text{Binomial}(n, p), k) = \sum_{i=0}^k \binom{n}{i} p^i (1-p)^{n-i}$$

である。場合によっては万単位の個数の値を足し上げる計算になる。

しかし、そういう和を取る計算をコンピュータで素朴に行うのはコンピュータ資源の無駄遣いになってしまう。

なぜならば、すでに説明したように非常にうまい方法があるからである。

### 二項分布とベータ分布の関係:

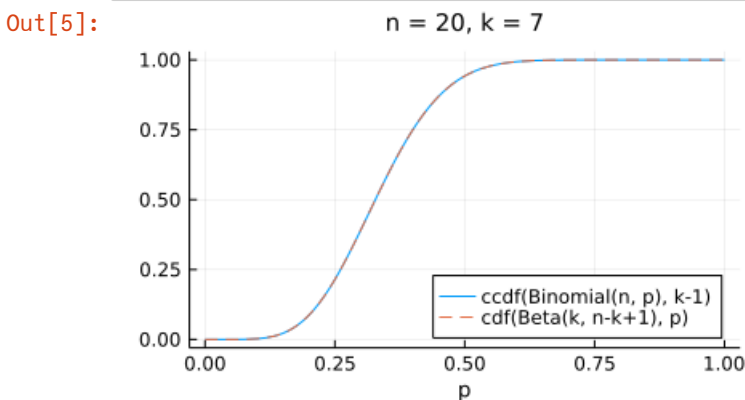
$$1 - \text{cdf}(\text{Binomial}(n, p), k-1) = \text{cdf}(\text{Beta}(k, n-k+1), p),$$

$$\text{cdf}(\text{Binomial}(n, p), k) = 1 - \text{cdf}(\text{Beta}(k+1, n-k), p).$$

後者は前者から導かれる。

以下でこれの前者を  $n = 20, k = 7$  の場合に数値的に確認してみよう。

```
In [5]: 1 n, k = 20, 7
2 plot(p → ccdf(Binomial(n, p), k-1), 0, 1; label="ccdf(Binomial(n, p), k-1)")
3 plot!(p → cdf(Beta(k, n-k+1), p); label="cdf(Beta(k, n-k+1), p)", ls=:dash)
4 title!("n = $n, k = $k", xguide="p", legend=:bottomright)
```



2つのグラフがぴったり重なり合っている。(ccdf は  $1 - \text{cdf}$  の意味である。)

ベータ分布の累積分布関数  $\text{cdf}(\text{Beta}(k+1, n-k), p)$  は正規化された不完全ベータ関数という名前がついている基本特殊関数になっており、その  $p$  に関する逆関数(分位点関数, quantile function)もコンピュータの基本特殊関数ライブラリの中で効率的に実装されている。だから、

$$1 - \text{cdf}(\text{Binomial}(n, p_L), k-1) = \text{cdf}(\text{Beta}(k, n-k+1), p_L) = \alpha/2$$

を満たす  $p_L$  は、ベータ分布の分位点関数(quantile function)を用いて、

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

を使えばコンピュータで効率的に計算できる。同様に

$$\text{cdf}(\text{Binomial}(n, p_U), k) = 1 - \text{cdf}(\text{Beta}(k+1, n-k), p_U) = \alpha/2$$

を満たす  $p_U$  は

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

を使えばコンピュータで効率的に計算できる。

**まとめ:** 二項分布モデルでのClopper-Pearsonの信頼区間  $[p_L, p_U]$  は

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

と計算できる。

**二項分布とベータ分布の間の関係が、コンピュータ上に実装された基本特殊関数ライブラリを通して、二項分布モデルの場合の信頼区間の効率的な計算に役に立っている!**

- 異なる確率分布に間の関係に関する数学
- 不完全ベータ関数などの基本特殊関数に関する数学
- それをコンピュータで実装するプログラミングの技術

などの多くの技術を組み合わせることによって, Clopper-Pearsonの信頼区間のシンプルで効率的な計算法が実現されている!

このようなことから, このClopper-Pearsonの信頼区間についてこのように理解することは, 高等教育を受けた人達にとっての重要な教養になり得る と思われる.

例えば,  $n = 20, k = 7, \alpha = 0.05$  の場合の信頼度  $1 - \alpha$  のClopper-Pearsonの信頼区間は次のように計算される.

In [6]:

```
1 n, k, α = 20, 7, 0.05
2 [quantile(Beta(k, n-k+1), α/2), quantile(Beta(k+1, n-k), 1-α/2)] ▶ println
[0.1539092047845412, 0.5921885345328282]
```

R言語を使って計算した結果も同じになっている:

In [7]:

```
1 rcopy(R""binom.test(7, 20, p=0.05)""[:conf_int] ▶ println
[0.15390920478454115, 0.5921885345328282]
```

WolframAlphaでも以下のようにすれば容易に計算できる.

$p_L \rightarrow \text{quantile}(\text{BetaDistribution}(7, 20-7+1), 0.025) \rightarrow$  [実行 (<https://www.wolframalpha.com/input?i=quantile%28BetaDistribution%287%2C+20-7%2B1%2C+0.025%29>)]  $\rightarrow 0.153909$

$p_U \rightarrow \text{quantile}(\text{BetaDistribution}(7+1, 20-7), 0.975) \rightarrow$  [実行 (<https://www.wolframalpha.com/input?i=quantile%28BetaDistribution%287%2B1%2C+20-7%2C+0.975%29>)]  $\rightarrow 0.592189$

### 1.3 二項分布モデルのBayes統計

二項分布とベータ分布の関係は通常の検定とBayes統計の関係を理解するためにも役に立つ.

二項分布のベイズ統計では先に用意した二項分布の確率質量関数  $P(k|n, p)$  だけではなく, パラメータ  $p$  に関する確率密度関数  $\varphi(p)$  を任意に与え(目的に合わせて適切に事前分布を決めることになるが, 理論的には任意でよい),  $(k, p)$  に関する同時分布 (joint distribution)

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

を考える. そして, 「 $n$  回中  $k$  回」の形のデータの数値が与えられたとき, このモデルの同時確率分布を, データと同じ数値が生成されたという条件で制限して得られるパラメータ  $p$  に関する条件付き確率分布  $\varphi(p|n, k)$  を考える:

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

このとき,  $\varphi(p)$  を **事前分布** (prior)と呼び,  $\varphi(p|n, k)$  を **事後分布** (posterior)と呼ぶ.

条件付き確率分布については

- ・「[条件付き確率分布, 尤度, 推定, 記述統計](https://nbviewer.org/github/genkuroki/Statistics/blob/master/2022/06%20Conditional%20distribution%2C%20likelihood%2C%20)」のノート (<https://nbviewer.org/github/genkuroki/Statistics/blob/master/2022/06%20Conditional%20distribution%2C%20likelihood%2C%20>)

に解説がある. 数学的には

- ・ 確率 = 全体に対する部分の割合
- ・ 条件付き確率 = 部分に対する部分の割合

のように考えると直観的に理解し易いと思う. 上の事後分布の公式を **ベイズの定理** と呼ぶこともあるが, ベイズの定理を知らなくても, 条件付き確率分布について理解していれば困らない.

事後分布  $\varphi(p|n, k)$  の  $p = p_0$  での値  $\varphi(p = p_0|n, k)$  はデータの数値  $p = p_0$  とデータの数値  $n, k$  の統計モデル  $P(k, p|n) = P(k|n, p)\varphi(p)$  の下での整合性の指標として利用できる. (P値とはまた別のモデルとデータの整合性の指標が得られたことになる.)

さて, 例によって, 問題はどのように事後分布と呼ばれる条件付き確率分布を計算するかである.

事後分布の公式を見れば分母が積分

$$Z(k|n) := \int_0^1 P(k|n, p)\varphi(p) dp$$

になっていて、その積分の部分の計算が大変そうなのがわかる。(事後分布の分母を **周辺尤度** (marginal likelihood)と呼ぶことがある。他にも様々な呼び方があるが、呼び方自体は重要ではない。)

二項分布の場合には、事前分布としてベータ分布を採用すると、事後分布もベータ分布になり、事後分布の計算が著しく単純化される。この事実をベータ分布は二項分布の **共役事前分布** (conjugate prior)であるという。

以下では  $\varphi(p)$  が  $\text{Beta}(a, b)$  ( $a, b > 0$ )の確率密度関数であると仮定する:

$$\varphi(p) = \frac{p^{a-1}(1-p)^{b-1}}{B(a, b)} \quad (0 < p < 1).$$

このとき、事後分布の分母は次のように計算される:

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

ゆえに、事後分布は次のようになる:

$$\begin{aligned} \varphi(p|n, k) &= \frac{P(k|n, p)\varphi(p)}{Z(k|n)} \\ &= \binom{n}{k} \frac{1}{B(a, b)} p^{a+k-1} (1-p)^{b+n-k} \times \left( \binom{n}{k} \frac{B(a+k, b+n-k)}{B(a, b)} \right)^{-1} \\ &= \frac{p^{a+k-1} (1-p)^{b+n-k}}{B(a+k, b+n-k)}. \end{aligned}$$

以上によって、事前分布が共役事前分布  $\text{Beta}(a, b)$  のとき、データ「 $n$  回中  $k$  回」から得られる事後分布は  $\text{Beta}(a+k, b+n-k)$  になることがわかった。

$a, b > 0$  と仮定していたが、 $a = 0$  または  $b = 0$  の場合にも、 $a+k$  と  $b+n-k$  が共に正になるならば、事後分布は  $\text{Beta}(a+k, b+n-k)$  はうまく定義されている。このような場合のうまく定義されていない事前分布は **improper事前分布** (improper prior)と呼ばれている。

この節の事柄については

- ・ [「例：ベータ関数と二項分布の関係とその応用」のノート](https://nbviewer.org/github/genkuroki/Statistics/blob/master/2022/07-1%20Relationship%20between%20beta%20and%20binomial%20distributions.ipynb)  
(<https://nbviewer.org/github/genkuroki/Statistics/blob/master/2022/07-1%20Relationship%20between%20beta%20and%20binomial%20distributions.ipynb>)

の最後の方にも解説がある。

## 1.4 P値とBayes統計の関係

二項分布とベータ分布の累積分布関数の間には以下の関係があるのであった:

$$\begin{aligned} 1 - \text{cdf}(\text{Binomial}(n, p_0), k-1) &= \text{cdf}(\text{Beta}(k, n-k+1), p_0), \\ \text{cdf}(\text{Binomial}(n, p_0), k) &= 1 - \text{cdf}(\text{Beta}(k+1, n-k), p_0). \end{aligned}$$

一般に分布  $D$  について累積分布関数  $\text{cdf}(D, x)$  は分布  $D$  に従う確率変数  $X$  が  $x$  以下になる確率を意味することに注意せよ。

ゆえに上の公式の前者の両辺は以下のような解釈を持つ:

- ・  $1 - \text{cdf}(\text{Binomial}(n, p_0), k-1)$  (分布  $\text{Binomial}(n, p_0)$  で  $k$  以上になる確率)は仮説  $p \leq p_0$  に関する片側検定のP値である。
- ・  $\text{cdf}(\text{Beta}(k, n-k+1), p_0)$  はimproper事前分布  $\text{Beta}(0, 1)$  の下での事後分布  $\text{Beta}(k, n-k+1)$  において仮説  $p \leq p_0$  が成立する確率である。

このように、仮説  $p \leq p_0$  の片側検定のP値はBayes統計での事後分布において仮説  $p \leq p_0$  が成立する確率に一致する。同様に、

- ・  $\text{cdf}(\text{Binomial}(n, p_0), k)$  (分布  $\text{Binomial}(n, p_0)$  で  $k$  以下になる確率)は仮説  $p \geq p_0$  に関する片側検定のP値である。
- ・  $1 - \text{cdf}(\text{Beta}(k+1, n-k), p_0)$  はimproper事前分布  $\text{Beta}(1, 0)$  の下での事後分布  $\text{Beta}(k+1, n-k+1)$  において仮説  $p \geq p_0$  が成立する確率である。

ゆえに、仮説  $p \geq p_0$  の片側検定のP値はBayes統計での事後分布において仮説  $p \geq p_0$  が成立する確率に一致する。

このように、数学的にはP値とBayes統計のあいだにexactな関係が付けられる場合もある。

### P値を使う統計学とBayes統計は決して水と油ではない。

以上によって、片側検定のP値については、Bayes統計でのexactな解釈が得られたことになる。

しかし、通常使われるP値は両側検定のP値(Clopper-Pearson型のP値では片側検定のP値の2倍を両側検定のP値として採用する)とBayes統計の関係はどうなっているのだろうか？

Clopper-Pearson型のP値のBayes統計での類似物を構成するには、上の片側検定の場合に関する結果を利用すればよい。

そのときに問題になることは、片側検定の向きを変えると、使用する(improper)事前分布が  $\text{Beta}(0, 1)$  と  $\text{Beta}(1, 0)$  のあいだで変化してしまうことである。

そこで、両側検定のP値のBayes統計によるexactな解釈を作ることはあきらめて、近似的な関係を得ることを目標にしよう。

そのためには事前分布として  $\text{Beta}(0, 1)$  と  $\text{Beta}(1, 0)$  の中間に位置する  $\text{Beta}(1/2, 1/2)$  を採用すると良さそうであると予想され、実際にそうであることが計算によって確認可能である。

$\text{Beta}(1/2, 1/2)$  は二項分布の **Jeffreys事前分布** と呼ばれる。

他の選択肢として、事前分布として平坦事前分布  $\text{Beta}(1, 1)$  を採用することも考えられる。 $\text{Beta}(1, 1)$  の確率密度関数は  $\varphi(p) = 1$  ( $0 < p < 1$ ) となるので、**平坦事前分布** または **一様事前分布** と呼んだりする。

事前分布として平坦事前分布を採用すると、両側検定のP値との対応において、Jeffreys事前分布を採用した場合よりも誤差が増えるが、 $k$  と  $n - k$  が大きければ、 $\text{Beta}(k + 1/2, n - k + 1/2)$  と  $\text{Beta}(k + 1, n - k + 1)$  の違いは小さくなるので、実践的な応用の場面では問題でなくなる。

以下では、事前分布が  $\text{Beta}(a, a)$  である場合を考える。

このとき、事後分布は  $\text{Beta}(k + a, n - k + a)$  になる。

この事後分布における仮説  $p = p_0$  の両側検定に関するClopper-Pearson型のP値関数の類似物は次のように定義される：

$$\text{pvalue}_{\text{Bayes}}(k|n, p = p_0) = \min \left( \frac{1}{2 \text{cdf}(\text{Beta}(k + a, n - k + a), p_0)}, \frac{1}{2(1 - \text{cdf}(\text{Beta}(k + a, n - k + a), p_0))} \right).$$

これは、二項分布に関するClopper-Pearson型のP値をそのままベータ分布に一般化しただけの定義になっている。

さらに、通常のP値として、Clopper-Pearson型のP値だけではなく、正規分布近似(中心極限定理)を使ったWilson型のP値関数も考えることにしよう：

$$\text{pvalue}_{\text{Wilson}}(k|n, p = p_0) = 2 \left( 1 - \text{cdf} \left( \text{Normal}(0, 1), \frac{|k - np_0|}{\sqrt{np_0(1 - p_0)}} \right) \right).$$

このとき、次が成立している。

**結論：**  $k$  と  $n - k$  が大きければ、以上の3つのP値  $\text{pvalue}_{\text{CP}}(k|n, p = p_0)$ ,  $\text{pvalue}_{\text{Bayes}}(k|n, p = p_0)$ ,  $\text{pvalue}_{\text{Wilson}}(k|n, p = p_0)$  は近似的に一致する。

要するに、両側検定の通常使われるP値についても、Bayes統計での近似的な解釈が存在する。

これを  $n = 100$ ,  $k = 30$  の場合に数値的に確認してみよう。

In [8]:

```

1 # 上に書いてある通りのP値の定義
2
3 function pvalue_cp(k, n, p)
4     bin = Binomial(n, p)
5     min(1, 2cdf(bin, k), 2ccdf(bin, k-1))
6 end
7
8 function pvalue_wilson(k, n, p)
9     z = (k - n*p)/sqrt(n*p*(1-p))
10    2ccdf(Normal(0, 1), abs(z))
11 end
12
13 function pvalue_bayes(k, n, p; a=1/2)
14     beta = Beta(k+a, n-k+a)
15     min(1, 2cdf(beta, p), 2ccdf(beta, p))
16 end
17
18 # 後で使うための別のP値の定義
19
20 function pvalue_sterne(k, n, p)
21     bin = Binomial(n, p)
22     # 次はnaiveな定義で計算効率は非常に悪い
23     sum(pdf(bin, i) for i in support(bin) if pdf(bin, i) <= pdf(bin, k))
24 end
25
26 function pvalue_wald(k, n, p)
27     phat = k/n
28     z = (k - n*p)/sqrt(n*phat*(1-phat))
29     2ccdf(Normal(0, 1), abs(z))
30 end

```

Out[8]: pvalue\_wald (generic function with 1 method)

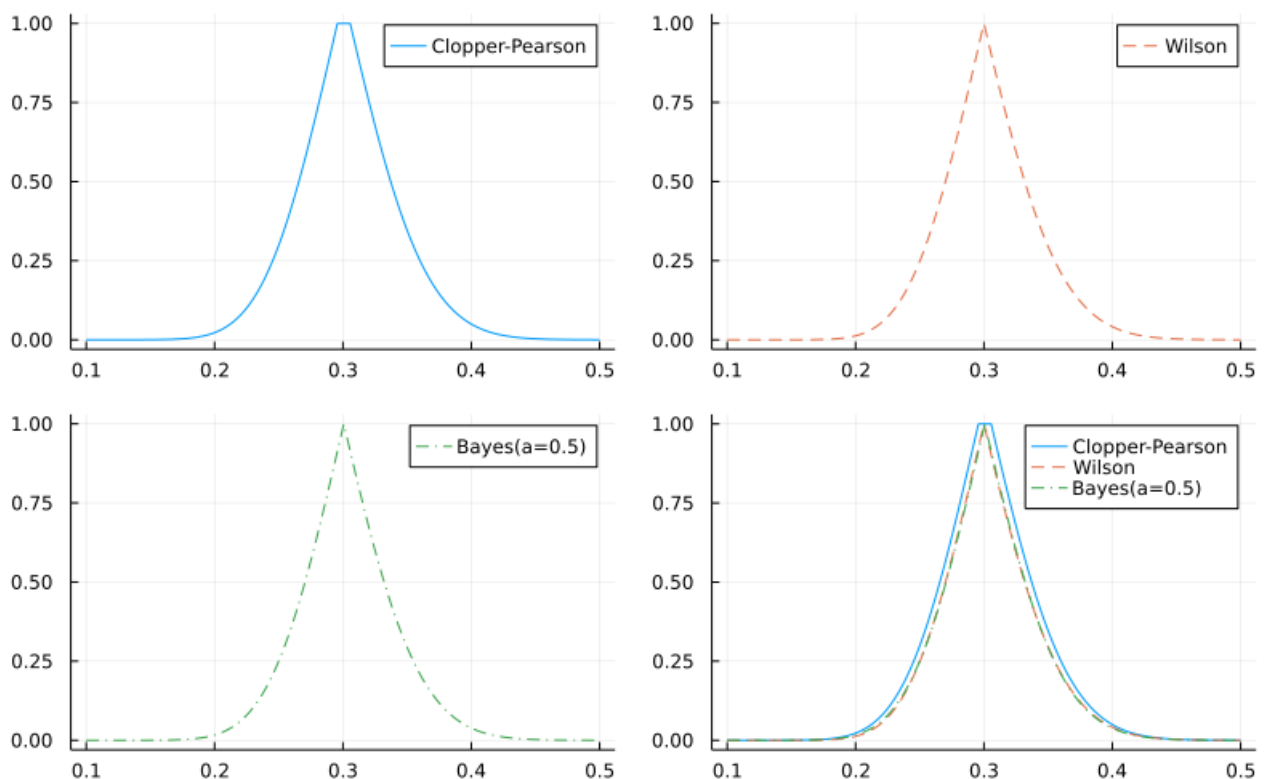
In [9]:

```

1 n, k, a = 100, 30, 1/2
2 xlim = (0.1, 0.5)
3 P1 = plot(p -> pvalue_cp(k, n, p), xlim...; label="Clopper-Pearson")
4 P2 = plot(p -> pvalue_wilson(k, n, p), xlim...; label="Wilson", c=2, ls=:dash)
5 P3 = plot(p -> pvalue_bayes(k, n, p; a), xlim...; label="Bayes(a=$a)", c=3, ls=:dashdot)
6 P4 = plot(p -> pvalue_cp(k, n, p), xlim...; label="Clopper-Pearson")
7 plot!(p -> pvalue_wilson(k, n, p), xlim...; label="Wilson", c=2, ls=:dash)
8 plot!(p -> pvalue_bayes(k, n, p; a), xlim...; label="Bayes(a=$a)", c=3, ls=:dashdot)
9
10 plot(P1, P2, P3, P4; size=(800, 500))

```

Out[9]:

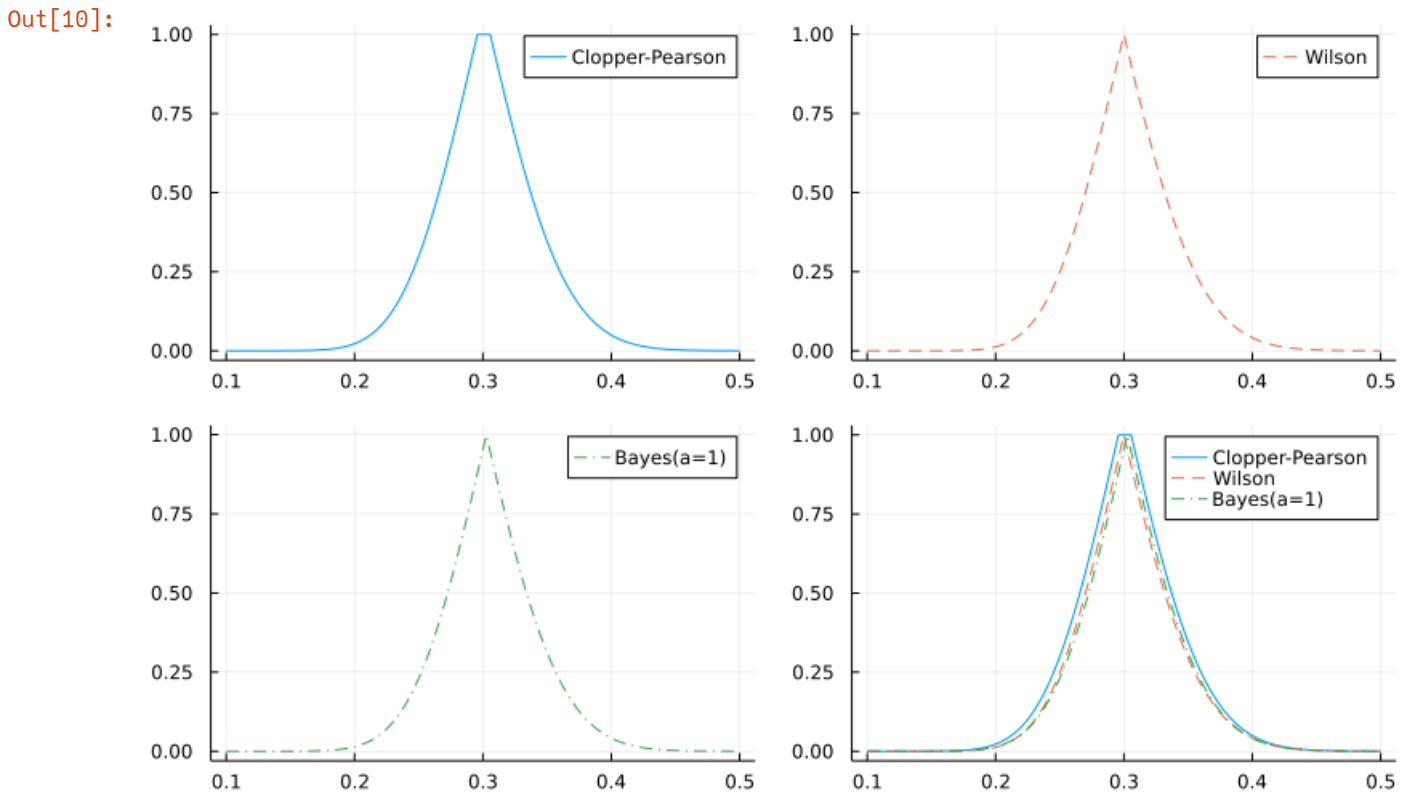


3つのP値関数が近似的に一致しており、特にJeffreys事前分布でのBayes版のP値関数は正規分布近似を使って定義されたWilson版



のP値関数と非常によく一致している。

```
In [10]: 1 n, k, a = 100, 30, 1
2 xlim = (0.1, 0.5)
3 P1 = plot(p → pvalue_cp(k, n, p), xlim...; label="Clopper-Pearson")
4 P2 = plot(p → pvalue_wilson(k, n, p), xlim...; label="Wilson", c=2, ls=:dash)
5 P3 = plot(p → pvalue_bayes(k, n, p; a), xlim...; label="Bayes(a=$a)", c=3, ls=:dashdot)
6 P4 = plot(p → pvalue_cp(k, n, p), xlim...; label="Clopper-Pearson")
7 plot!(p → pvalue_wilson(k, n, p), xlim...; label="Wilson", c=2, ls=:dash)
8 plot!(p → pvalue_bayes(k, n, p; a), xlim...; label="Bayes(a=$a)", c=3, ls=:dashdot)
9
10 plot(P1, P2, P3, P4; size=(800, 500))
```



平坦事前分布の場合にも、Bayes版のP値関数はWilson型のP値関数とよく一致している。

以上のグラフを見ても、P値を使う統計学とBayes統計が水と油だと考えることはバカげているように思われる。

## 1.5 復習: 二項分布とベータ分布の関係の証明

以上を理解できた人は二項分布とベータ分布の累積分布関数の関係を完璧に理解しておくことの価値は大きいと感じると思われる。

以下では、再度、二項分布とベータ分布の累積分布関数の関係を証明しておく。

**大事なことは何度でも証明しておいた方がよい。**

### 1.5.1 両辺を $p$ で微分して確認する方法

証明したい公式

$$1 - \text{cdf}(\text{Binomial}(n, p), k - 1) = \text{cdf}(\text{Beta}(k, n - k + 1), p) \quad (k = 1, 2, \dots, n)$$

の両辺を具体的に書き下すと以下ようになる:

$$\sum_{i=k}^n \binom{n}{i} p^i (1-p)^{n-i} = \frac{\int_0^p t^{k-1} (1-t)^{n-k} dt}{B(k, n-k+1)} \quad (k = 1, 2, \dots, n). \quad (*)$$

$p = 0$  のとき、 $k \geq 1$  より(\*)の左辺は0になり、(\*)の右辺も0になる。

(\*)の左辺を  $p$  で微分すると、

$$\begin{aligned}
\frac{\partial}{\partial p}((*)\text{の左辺}) &= \sum_{i \geq k} \frac{n!}{i!(n-i)!} i p^{i-1} (1-p)^{n-i} - \sum_{i \geq k} \frac{n!}{i!(n-i)!} (n-i) p^i (1-p)^{n-i-1} \\
&= \sum_{i \geq k} \frac{n!}{(i-1)!(n-i)!} p^{i-1} (1-p)^{n-i} - \sum_{i \geq k} \frac{n!}{i!(n-i-1)!} p^i (1-p)^{n-i-1} \\
&= \sum_{i \geq k} \frac{n!}{(i-1)!(n-i)!} p^{i-1} (1-p)^{n-i} - \sum_{i \geq k+1} \frac{n!}{(i-1)!(n-i)!} p^{i-1} (1-p)^{n-i} \\
&\quad - \frac{n!}{(k-1)!(n-k)!} p^{k-1} (1-p)^{n-k}.
\end{aligned}$$

3つめの等号は2つめの和の項のインデックスの  $i$  を  $i-1$  で置き換えることによって得られる。

一方,  $\Gamma(i+1) = i!$  ( $i = 0, 1, 2, \dots$ )より,

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

なので,  $(*)$ の右辺を  $p$  で微分すると,

$$\frac{\partial}{\partial p}((*)\text{の右辺}) = \frac{n!}{(k-1)!(n-k)!} p^{k-1} (1-p)^{n-k}.$$

$(*)$ の両辺は  $p = 0$  のとき互いに等しく, 導関数も互いに等しい. ゆえに $(*)$ の左辺と右辺は等しい.

### 1.5.2 ベータ分布の一樣分布の順序統計量の分布として解釈を使う方法

前節での $(*)$ の証明法では公式 $(*)$ が成立する理由はよくわからないので, この節では一樣分布の順序統計量としてベータ分布が解釈できることを使った証明を紹介しよう.

$T_1, \dots, T_n$  は 0 から 1 のあいだの  $n$  個の独立な一樣乱数であるとし, その中で  $k$  番目に小さなものを  $T_{(k)}$  と書く ( $T_{(k)}$  を一樣分布の順序統計量と呼ぶ).

このとき,  $T_{(k)} \leq p$  となることと,  $T_1, \dots, T_n$  の中に  $p$  以下のものが  $k$  個以上あることは同値である. 各々の  $T_i$  が  $p$  以下になる確率は  $p$  なので,  $T_1, \dots, T_n$  の中に含まれる  $p$  以下のものの個数は二項分布  $\text{Binomial}(n, p)$  に従う. したがって,

$$(T_{(k)} \leq p \text{ となる確率}) = (\text{二項分布 } \text{Binomial}(n, p) \text{ において } k \text{ 以上になる確率}) = \sum_{i \geq k} \binom{n}{i} p^i (1-p)^{n-i}.$$

一方,  $t \leq T_{(k)} \leq t + dt$  となる確率は

$$(t \leq T_{(k)} \leq t + dt \text{ となる確率}) \approx \frac{n!}{(k-1)!1!(n-k)!} t^{k-1} (1-t)^{n-k} dt$$

と近似される.  $n!/((k-1)!1!(n-k)!)$  は  $n$  個の  $T_1, \dots, T_n$  を  $p$  より小さな  $i-1$  個と  $t$  と  $t+dt$  の間に含まれる 1 個と残りの  $n-k$  個にグループ分けする方法の個数である. そして,

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

なので,  $T_{(k)}$  が従う分布の確率密度関数はベータ分布  $\text{Beta}(k, n-k+1)$  の密度関数

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

に一致することがわかる. したがって,

$$(T_{(k)} \leq p \text{ となる確率}) = \frac{\int_0^p t^{k-1} (1-t)^{n-k} dt}{B(k, n-k+1)}.$$

これを上で示した結果と比較することによって次が得られる:

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

## 2 P値関数について

P値関数に関する考え方の簡潔な説明については次の論文を参照せよ:

- Valentin Amrhein and Sander Greenland, 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. [[doi \(https://doi.org/10.1177%2F02683962221105904\)](https://doi.org/10.1177%2F02683962221105904)]

## 2.1 P値はモデルのパラメータ値とデータの数値の整合性の指標の1つとして定義される

我々は、データとその生成のされ方を確率分布でモデル化し、そのとき使われたパラメータ付きの確率分布を統計モデルと呼ぶのであった。

例えば、「 $n$  回中  $k$  回当たりが出た」の形のデータの生成のされ方の統計モデルの代表例は二項分布モデル  $\text{Binomial}(n, p)$  であり、当たりが出る確率は成功確率パラメータ  $p$  でモデル化される。

以下では、データの数値  $x$  の生成のされ方をパラメータ  $\theta$  を持つ統計モデル  $M(\theta)$  でモデル化している状況を考える。

そしてさらに、データの数値  $x$  に関する仮説  $\theta = \theta_0$  のP値  $\text{pvalue}(x|\theta = \theta_0)$  が適切に定義されている状況を考える。

P値は、データの数値  $x$  と仮説  $\theta = \theta_0$  の下での統計モデル  $M(\theta = \theta_0)$  の整合性の指標の1つである。

ここでの整合性は英語では compatibility や consistency という意味である。

例えば、上の二項分布モデルの場合には、「100 回中 40 回当たりが出た」というデータが得られたとき、仮説  $p = 1/2$  のP値 (Clopper-Pearsonの信頼区間を与えるP値)は約 5.7% になる。

**その 5.7% というP値は「100 回中 40 回当たりが出た」というデータと「当たりが出る確率は  $p = 1/2$  である」という仮説の整合性の指標として使われることになる。**

そのとき、背景にある統計モデルの妥当性に注意を払う必要がある。

P値の定義の仕方によって、その値は微妙に異なった値になる。そういう細かな違いに結果が影響されないような頑健な議論を目指すべきである。

In [11]:

```
1 @show k, n, p = 40, 100, 1/2
2 @show pvalue_cp(k, n, p)
3 @show pvalue_sterne(k, n, p)
4 @show pvalue_wilson(k, n, p)
5 @show pvalue_wald(k, n, p);
```

```
(k, n, p) = (40, 100, 1 / 2) = (40, 100, 0.5)
pvalue_cp(k, n, p) = 0.05688793364098078
pvalue_sterne(k, n, p) = 0.05688793364098067
pvalue_wilson(k, n, p) = 0.04550026389635841
pvalue_wald(k, n, p) = 0.04122683337163676
```

## 2.2 P値は信頼区間を考えたいすべてのパラメータ値について定義されている

二項分布モデルの場合には仮説  $p = 1/2$  に限らず、0 から 1 のあいだの任意の数値  $p_0$  について、仮説  $p = p_0$  についてP値が定義されている。

それによって、「100 回中 40 回当たりが出た」というデータが得られたとき、仮説  $p = p_0$  における  $p_0$  を動かしながらP値を計算することによって、 $p = 1/2$  の場合以外についても仮説  $p = p_0$  と「100 回中 40 回当たりが出た」というデータの整合性達がどうなっているかを知ることができる。

一般に、値  $p_0$  に対して、データ  $x$  に関する仮説  $p = p_0$  のP値を対応させる関数を **P値関数** (P-value function)と呼ぶ。

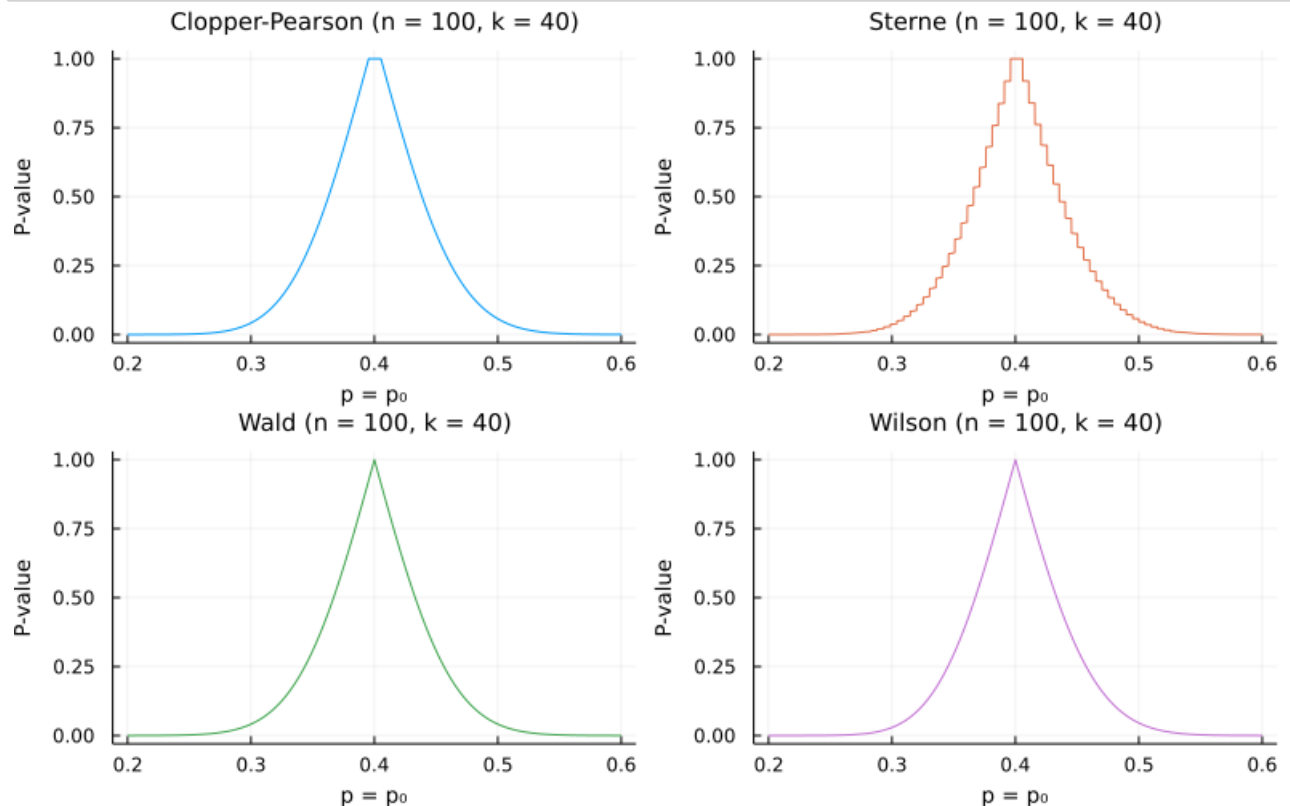
P値関数は、統計モデルについて、データの数値とパラメータ値のあいだの整合性(相性)の様子全体の情報を持っている重要な関数である。

以下で4種のP値関数のグラフをプロットしてみよう。

In [12]:

```
1 k, n = 40, 100
2 xlim = (0.2, 0.6)
3 P1 = plot(p → pvalue_cp(k, n, p), xlim...;
4           label="", title="Clopper-Pearson (n = $n, k = $k)")
5 P2 = plot(p → pvalue_sterne(k, n, p), xlim...;
6           label="", title="Sterne (n = $n, k = $k)", c=2)
7 P3 = plot(p → pvalue_wald(k, n, p), xlim...;
8           label="", title="Wald (n = $n, k = $k)", c=3)
9 P4 = plot(p → pvalue_wilson(k, n, p), xlim...;
10          label="", title="Wilson (n = $n, k = $k)", c=4)
11 plot(P1, P2, P3, P4; xguide="p = p₀", yguide="P-value", size=(800, 500))
```

Out[12]:



近似的にはどれもほぼ同じ形をしている。

## 2.3 分割表の場合のP値関数

「Aでは  $m = a + b$  回中  $a$  回当たりが出て、Bでは  $n = c + d$  回中  $c$  回当たりが出た」の形式のデータを

	当たり	はずれ	合計
A	$a$	$b$	$m$
B	$c$	$d$	$n$

のように  $2 \times 2$  の分割表で表すのであった。

このようなデータの生成のされ方の統計モデルとして、2つの二項分布モデル  $\text{Binomial}(m, p) \times \text{Binomial}(n, q)$  を採用しよう。

このモデル内で、 $a$  の値は分布  $\text{Binomial}(m, p)$  に従ってランダムに決まり、 $c$  の値はそれとは独立に分布  $\text{Binomial}(n, q)$  に従ってランダムに決まっていると考える。 $a + b = m, c + d = n$  は固定されているので、 $a, c$  から  $b, d$  が自動的に決まる。

応用時に重要になるのは、 $p$  と  $q$  の違いである。

入門的な多くの教科書では非常に残念なことに、仮説  $p = q$  のP値の定義しか書かれておらず、 $p$  と  $q$  の違いの大きさを表す指標の信頼区間の定義が書かれていない。

これが前節で、P値は信頼区間を考えたいすべてのパラメータ値について定義されている と強調したくなった理由である。

しかし、我々はすでに

- 「検定と信頼区間: 比率の比較」のノート  
(<https://nbviewer.org/github/genkuroki/Statistics/blob/master/2022/11%20Hypothesis%20testing%20and%20confidence%20interval%20Two%20proportions.ipynb>).

で  $p$  と  $q$  の違いを表す3つの指標についてP値と信頼区間を定義している:

- オッズ比パラメータ  $OR = \frac{p/(1-p)}{q/(1-q)} = \frac{p(1-q)}{(1-p)q}$ ,
- リスク比パラメータ  $RR = \frac{p}{q}$ ,
- リスク差パラメータ  $RD = p - q$ .

この点は入門的な教科書の多くとは一線を画する点である。この点に関しては、Rothmanさん達の有名な疫学の教科書

- Rothman, Lash, and Greenland, Modern Epidemiology [[Googleで検索 \(https://www.google.com/search?q=Rothman+Lash+Greenland+Modern+Epidemiology\)](https://www.google.com/search?q=Rothman+Lash+Greenland+Modern+Epidemiology)]

が詳しい。講義動画

- 佐藤俊哉, 臨床研究者のための生物統計学「回帰モデルと傾向スコア」2019年2月21日 [[YouTube \(https://youtu.be/cOHN444kBlo\)](https://youtu.be/cOHN444kBlo)]

も参照せよ。8:30以降の「変換しない場合」がリスク差パラメータを  $p$  と  $q$  の違いの指標として採用した場合に対応しており、10:30以降の「対数変換の場合」がリスク比パラメータを  $p$  と  $q$  の違いの指標として採用した場合に対応しており、12:00以降の「ロジット変換の場合」がオッズ比パラメータ(正確にはその対数)を  $p$  と  $q$  の違いの指標として採用した場合に対応している。「ロジット変換の場合」はロジスティック回帰の話になっている。ただし、複数の分割表のデータを扱うより複雑な場合を扱っている。

## 2.4 P値函数から信頼区間が定義される

P値が信頼区間を考えたいすべてのパラメータ値について定義されていることの利点は、言うまでもないことだが、信頼区間を定義できることである。

一般に、データ  $x$  の生成のされ方に関する統計モデル  $M(\theta)$  に関するP値函数  $\text{pvalue}(x|\theta = \theta_0)$  が与えられているとき、パラメータ  $\theta$  に関する信頼度  $1 - \alpha$  の信頼区間 ( $100(1 - \alpha)\%$  信頼区間)は次のように定義される:

$$\text{confint}^\theta(x|\alpha) = \{ \theta_0 \mid \text{pvalue}(x|\theta = \theta_0) \geq \alpha \}.$$

すなわち、データの数値  $x$  について仮説  $\theta = \theta_0$  のP値が  $\alpha$  以上になるような  $\theta_0$  全体の集合が信頼区間になる。

ここで使われている閾値  $\alpha$  は有意水準と呼ばれることがある。

コンピュータで計算するときには、 $\text{pvalue}(x|\theta = \theta_0) = \alpha$  を満たす  $\theta_0$  の値を2分法やNewton法などで計算すれば、一般的な場合にも信頼区間を概ね計算できる。

しかし、それだとバグも発生し易くなるし、計算効率も悪くなりがちなので、信頼区間を直接できるシンプルな公式がある場合にはそれを使うことになる。

## 2.5 2×2の分割表のP値と信頼区間とP値函数の例

以下では2×2の分割表のデータの数値

	当たり	はずれ
A	$a = 30$	$b = 70$
B	$c = 20$	$d = 80$

に関するP値と信頼区間の例を示そう。

簡単のためオッズ比パラメータ  $OR$  の場合(実際には対数オッズ比パラメータ  $\log OR$  を扱う場合)のWald型のP値函数のみを扱う。

$a + b = m, c + d = n$  が十分に大きければ, 仮説  $OR = \omega$  の統計モデル内で, モデル内での仮想的データの数値の対数オッズ比が次のように近似的に正規分布に従うことを示せる:

$$\log \widehat{OR} = \log \frac{ad}{bc} \sim \text{Normal}(\log \omega, \widehat{SE}_{\log \widehat{OR}}), \text{ approximately.}$$

ここで,

$$\widehat{SE}_{\log \widehat{OR}} = \sqrt{\frac{1}{a} + \frac{1}{b} + \frac{1}{c} + \frac{1}{d}}$$

この近似については

- [「検定と信頼区間: 比率の比較」のノート](https://nbviewer.org/github/genkuroki/Statistics/blob/master/2022/11%20Hypothesis%20testing%20and%20confidence%20interval%20Two%20proportions.ipynb)

(<https://nbviewer.org/github/genkuroki/Statistics/blob/master/2022/11%20Hypothesis%20testing%20and%20confidence%20interval%20Two%20proportions.ipynb>).

の「Wald版のオッズ比に関するP値と信頼区間の定義」に関する節を参照せよ.

これによって, 以下のように仮説  $OR = \omega$  のP値と  $OR$  の信頼区間が定義される:

$$\begin{aligned} \text{pvalue}_{\text{Wald}}(a, b, c, d | OR = \omega) &= 2 \left( 1 - \text{cdf} \left( \text{Normal}(0, 1), \frac{|\log \widehat{OR} - \log \omega|}{\widehat{SE}_{\log \widehat{OR}}} \right) \right), \\ \text{confint}_{\text{Wald}}^{\text{OR}}(a, b, c, d | \alpha) &= \left[ \exp \left( -z_{\alpha/2} \widehat{SE}_{\log \widehat{OR}} \right) \widehat{OR}, \exp \left( z_{\alpha/2} \widehat{SE}_{\log \widehat{OR}} \right) \widehat{OR} \right]. \end{aligned}$$

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

In [13]:

```
1 # 上の定義通りのP値関数と信頼区間関数の定義
2
3 oddsratiohat(a, b, c, d) = safediv(a*d, b*c)
4 stderrhat_logoddsratiohat(a, b, c, d) = sqrt(1/a + 1/b + 1/c + 1/d)
5
6 function pvalue_or_wald(a, b, c, d; w=1)
7     logORhat = log(oddsratiohat(a, b, c, d))
8     SEhat_logORhat = stderrhat_logoddsratiohat(a, b, c, d)
9     2ccdf(Normal(0, 1), safediv(abs(logORhat - log(w)), SEhat_logORhat))
10 end
11
12 function confint_or_wald(a, b, c, d; alpha=0.05)
13     z = quantile(Normal(), 1-alpha/2)
14     ORhat = oddsratiohat(a, b, c, d)
15     SEhat_logORhat = stderrhat_logoddsratiohat(a, b, c, d)
16     [safemul(exp(-z*SEhat_logORhat), ORhat), safemul(exp(z*SEhat_logORhat), ORhat)]
17 end
```

Out[13]: confint\_or\_wald (generic function with 1 method)

In [14]:

```

1 # グラフのプロット用の関数
2 # この手の関数は計算用の関数よりも複雑になりがち
3
4 function plot_pvalue_function_of_or_wald(;
5     a=30, b=70, c=20, d=80, ω = 1.0, α = 0.05,
6     xlim = confint_or_wald(a, b, c, d; α=1e-3)
7 )
8     @show α
9     @show ω
10    @show z = quantile(Normal(0, 1), 1 - α/2)
11    println()
12    @show ORhat = oddsratiohat(a, b, c, d)
13    @show exp(z * stderrhat_logoddsratiohat(a, b, c, d))
14    println()
15    @show P_value = pvalue_or_wald(a, b, c, d; ω)
16    @show CI = confint_or_wald(a, b, c, d; α)
17
18    plot(ω → pvalue_or_wald(a, b, c, d; ω), xlim...;
19        label="P-value function (Wald)")
20    vline!([ORhat]; label="ORhat=$(round(ORhat, digits=5))", ls=:dash)
21    plot!(CI, fill(α, 2); label="$ (100(1-α))% CI", lw=2)
22    vline!([ω]; label="", c=:black, lw=0.5)
23    scatter!([ω], [P_value]; label="P-value of OR = $ω", c=:red)
24    plot!(xguide="OR = ω (log scale)", yguide="P-value")
25    plot!(xscale=:log, xtick=logtick(; xlim), ytick=0:0.05:1)
26    title!("a, b, c, d = $a, $b, $c, $d")
27    plot!(size=(600, 400))
28 end

```

Out[14]: plot\_pvalue\_function\_of\_or\_wald (generic function with 1 method)

In [15]:

```

1 a, b, c, d = 30, 70, 20, 80
2 graph_of_pvalue_function =
3     plot_pvalue_function_of_or_wald(; a, b, c, d, ω = 1.0, α = 0.05)

```

α = 0.05

ω = 1.0

z = quantile(Normal(0, 1), 1 - α / 2) = 1.9599639845400576

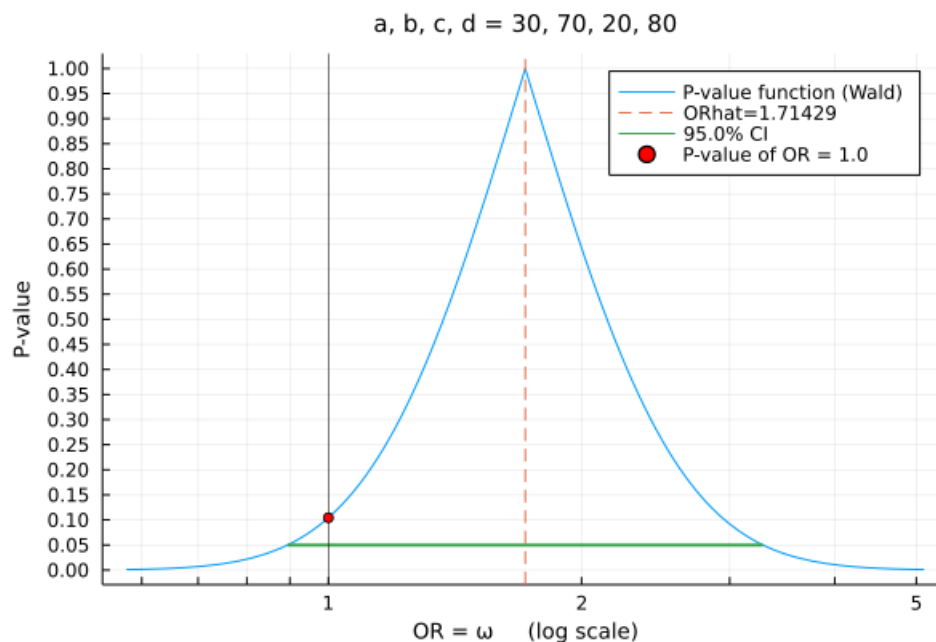
ORhat = oddsratiohat(a, b, c, d) = 1.7142857142857142

exp(z \* stderrhat\_logoddsratiohat(a, b, c, d)) = 1.9163037024703367

P\_value = pvalue\_or\_wald(a, b, c, d; ω) = 0.10432099005636014

CI = confint\_or\_wald(a, b, c, d; α) = [0.8945793467266185, 3.28509206137772]

Out[15]:



仮説 OR = 1 すなわち「AとBで当たりが出る確率は同じ」という仮説のP値は 10.4% 程度になっている。

これが意味するところは以下の通り. データの数値

	当たり	はずれ
A	$a = 30$	$b = 70$
B	$c = 20$	$d = 80$

を見ると、Aの方がBよりも当たりが出る確率が高そうに見えるが、統計モデル内ではAとBで当たりが出る確率が同じであったとしても、モデル内でこのデータの数値以上に偏った値が生じる確率は約 10.4% もある。

さらに、上のグラフを見ると、仮説  $OR = 4$  のP値は 1% 程度に見えるが、実際に計算してもその程度の値になる。

```
In [16]: 1 pval4 = pvalue_or_wald(a, b, c, d; ω=4)
```

```
Out[16]: 0.010670206411228012
```

WolframAlphaでも仮説  $OR = 4$  のP値を計算してみよう。

$\widehat{SE}_{\log \widehat{OR}} \rightarrow \text{sqrt}(1/a + 1/b + 1/c + 1/d)$  where  $a=30.0, b=70, c=20, d=80 \rightarrow$  [実行 (<https://www.wolframalpha.com/input?i=sqrt%281%2Fa+%2B1%2Fb+%2B1%2Fc+%2B1%2Fd%29+where+a%3D30.0%2C+b%3D70%2C+c%3D20%2C+d%3D80%29>)]  $\rightarrow 0.331842$

$\log \widehat{OR} - \log \omega \rightarrow \log((a d)/(b c)) - \log(4)$  where  $a=30.0, b=70, c=20, d=80 \rightarrow$  [実行 (<https://www.wolframalpha.com/input?i=log%28a+d%29%2F%28b+c%29%29+-log%284%29+where+a%3D30.0%2C+b%3D70%2C+c%3D20%2C+d%3D80%29>)]  $\rightarrow -0.847298$

$(\log \widehat{OR} - \log \omega) / \widehat{SE}_{\log \widehat{OR}} \rightarrow 0.847298 / 0.331842 \rightarrow$  [実行 (<https://www.wolframalpha.com/input?i=0.847298%2F0.331842%29>)]  $\rightarrow 2.55332$

P値  $\rightarrow 2(1 - \text{cdf}(\text{NormalDistribution}(0,1), 2.55332)) \rightarrow$  [実行 (<https://www.wolframalpha.com/input?i=2%281+-cdf%28NormalDistribution%280%2C1%29%2C+2.55332%29%29>)]  $\rightarrow 0.0106701$

```
In [17]: 1 # WolframAlphaで求めたP値のJuliaで求めたP値に対する相対誤差が小さいことの確認
          2 0.0106701/pval4 - 1
```

```
Out[17]: -9.972743160724384e-6
```

WolframAlphaでも  $OR$  の 95% 信頼区間を求めてみよう。

$z_{\alpha/2} \rightarrow \text{quantile}(\text{Normal}(0,1), 0.975) \rightarrow$  [実行 (<https://www.wolframalpha.com/input?i=quantile%28Normal%280%2C1%29%2C+0.975%29>)]  $\rightarrow 1.95996$

信頼区間  $\rightarrow \{\exp(-1.95996 \cdot 0.331842) (a d)/(b c), \exp(1.95996 \cdot 0.331842) (a d)/(b c)\}$  where  $a=30.0, b=70, c=20, d=80 \rightarrow$  [実行 ([https://www.wolframalpha.com/input?i=%7Bexp%28-1.95996\\*0.331842%29+%28a+d%29%2F%28b+c%29%2C++exp%281.95996\\*0.331842%29+%28a+d%29%2F%28b+c%29%29%7D+w%28a%3D30.0%2C+b%3D70%2C+c%3D20%2C+d%3D80%29%29](https://www.wolframalpha.com/input?i=%7Bexp%28-1.95996*0.331842%29+%28a+d%29%2F%28b+c%29%2C++exp%281.95996*0.331842%29+%28a+d%29%2F%28b+c%29%29%7D+w%28a%3D30.0%2C+b%3D70%2C+c%3D20%2C+d%3D80%29%29))]  $\rightarrow \{0.89458, 3.28509\}$

```
In [18]: 1 # WolframAlphaで求めた信頼区間のJuliaで求めた信頼区間に対する相対誤差が小さいことの確認
          2 [0.89458, 3.28509] ./ confint_or_wald(a, b, c, d; α=0.05) .- 1
```

```
Out[18]: 2-element Vector{Float64}:
 7.302576165990615e-7
-6.274946581230623e-7
```

## 2.6 2×2の分割表での検定ではどれを使うべきか

以上においては計算の仕方がシンプルだという理由でWald型のP値関数と信頼区間を扱ったが、標本サイズが小さいときに誤差が大きくなるという欠点がある。他にも以下の選択肢がある：

(1) Pearsonの  $\chi^2$  検定 (ただし扱う仮説を  $OR = \omega$  と  $RR = \rho$  の場合に一般化しておくこと、連続性補正はかけない)

(2) Fisher検定 (ただし扱う仮説を  $OR = \omega$  の場合に一般化しておくこと)

(3)  $p$  と  $q$  の違いの指標としてそれらの差  $p - q$  を扱う場合には Zou-Donner 2004 の方法が使える。 ([Julia言語による実装例 \(https://github.com/genkuroki/public/blob/main/0033/probability%20of%20alpha%20error%20of%20Zou-Donner.ipynb\)](https://github.com/genkuroki/public/blob/main/0033/probability%20of%20alpha%20error%20of%20Zou-Donner.ipynb))

それぞれ長所と短所があるので目的に合わせて適当に使い分けられよい。標本サイズが十分に大きい場合にはWald型のP値関数と信頼区間も十分な精度を持ち、問題なく使える。



**注意:** よく使われているのは, Pearsonの  $\chi^2$  検定とFisher検定である. 標本サイズが小さい場合にはFisher検定を使えと解説されている場合もあるようだが, 実際に色々計算するとそれは誤りだと分かる. 実際には標本サイズが小さなき, Fisher検定には過剰にP値が大きくなり過ぎるという欠点がある. しかし, その欠点は「第一種の過誤の確率が常に有意水準以下である」というFisher検定の優れた性質の裏返しなので, どの条件を重視するかによって, (Yatesの連続性補正をかけない)Pearsonの  $\chi^2$  検定と使い分けるとよいと思われる. 次の解説も参考になる:

- 朝倉こう子, 濱崎俊光, 連載 第3回 医学データの統計解析の基本 2つの割合の比較 [\[PDF \(https://www.jstage.jst.go.jp/article/dds/30/3/30\\_235/\\_pdf\)\]](https://www.jstage.jst.go.jp/article/dds/30/3/30_235/_pdf)

## 2.7 P値函数と最尤法の関係

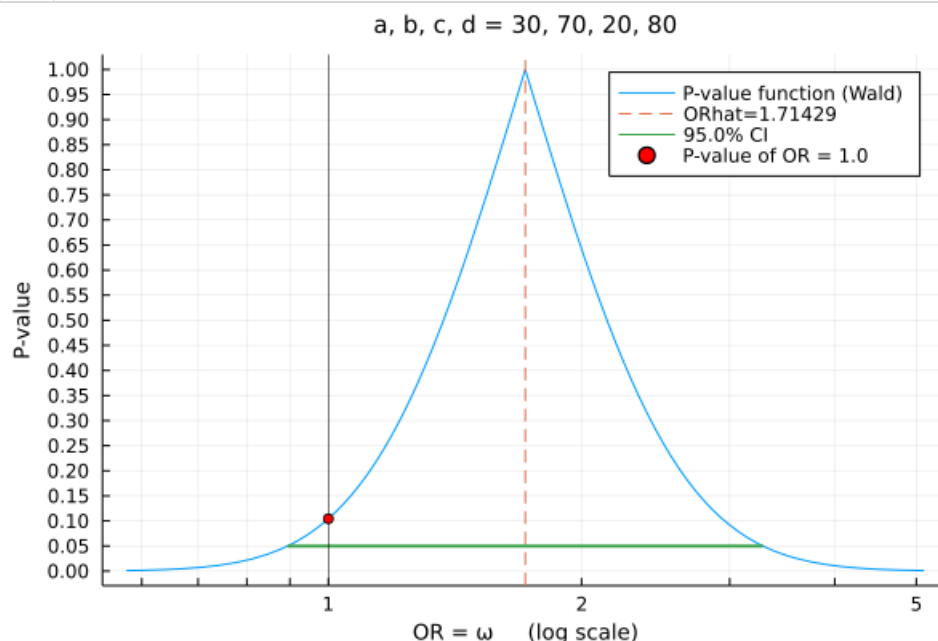
2×2の分割表のデータの数値

	当たり	はずれ
A	$a = 30$	$b = 70$
B	$c = 20$	$d = 80$

に関するP値函数などは1つのグラフの中に以下のようにプロットされるのであった.

In [19]: 1 graph\_of\_pvalue\_function

Out[19]:



このグラフにおけるデータの数値のオッズ比

$$\widehat{OR} = \frac{ad}{bc} = \frac{30 \cdot 80}{70 \cdot 20} = \frac{12}{7} \approx 1.7142857142857142$$

の値でP値函数が最大値の1になっていることに注意せよ. P値函数の定義

$$pvalue_{Wald}(a, b, c, d | OR = \omega) = 2 \left( 1 - cdf \left( \text{Normal}(0, 1), \frac{|\log \widehat{OR} - \log \omega|}{\widehat{SE}_{\log \widehat{OR}}} \right) \right)$$

を見れば,  $\omega = \widehat{OR}$  のとき最大値1になることはすぐにわかる.

実はこのデータの数値のオッズ比  $OR = (ad)/(bc)$  は2つの二項分布モデルにおける最尤法(さいゆうほう)の解

$$\hat{p} = \frac{a}{a+b}, \quad \hat{q} = \frac{c}{c+d} \quad \left( \text{このとき } 1 - \hat{p} = \frac{b}{a+b}, \quad 1 - \hat{q} = \frac{d}{c+d} \right)$$

のオッズ比になっている:

$$\frac{\hat{p}(1-\hat{q})}{(1-\hat{p})\hat{q}} = \frac{ad}{bc} = \widehat{OR}.$$

この場合に限らず, **多くの場合にP値函数が最大になるパラメータ値は最尤法の解に対応する値になっている.**

少なくとも, 近似的にはP値関数を最大化するパラメータ値は最尤法の解に対応する値だと考えてよい.

データの数値  $x$  に関する仮説  $\theta = \theta_0$  のP値は, 統計モデルを前提にしたときの仮説  $\theta = \theta_0$  とデータの数値  $x$  の整合性の指標の1つであった.

ゆえに, 以上で述べたことは, **最尤法の解に対応するパラメータ値はデータの数値との整合性が最も高いパラメータ値である** とみなせる.

最尤法による点推定にはこのような解釈がある.

**注意・警告:** 統計学を実践的に使う場合には, 最尤法による点推定の結果のみを報告せずに, 信頼区間などの区間推定の結果も報告する習慣になっている. 点推定の値は単に統計モデルの前提の下でデータの数値と最も整合性が高いパラメータ値を計算したにすぎず, 確率的な揺らぎが考慮されていない. 確率的揺らぎの影響に配慮した区間推定の結果も報告しないと, 点推定の値がどこまで正確な推定値だと考えられるかが全くわからなくなってしまう.

**注意:** 尤度(ゆうど, likelihood)については

- [「条件付き確率分布, 尤度, 推定, 記述統計」に関するノート](https://nbviewer.org/github/genkuroki/Statistics/blob/master/2022/06%20Conditional%20distribution%2C%20likelihood%2C%20)  
(<https://nbviewer.org/github/genkuroki/Statistics/blob/master/2022/06%20Conditional%20distribution%2C%20likelihood%2C%20>)

における「尤度 (ゆうど) と推定」の節を参照せよ. 尤度の定義は **統計モデル内でデータと同じ数値が生成される確率もしくは確率の密度** であり, モデルのデータの数値への適合度の指標としては使えるが, 「もっともらしさ」の指標としては不適切であることに注意せよ.

### 3 Welchの t 検定について

2つの群の平均の差に関するWelchの  $t$  検定に付随するP値と信頼区間については

- [「検定と信頼区間: 平均の比較」に関するノート](https://nbviewer.org/github/genkuroki/Statistics/blob/master/2022/12%20Hypothesis%20testing%20and%20confidence%20interval%20Two%20means.ipynb)  
(<https://nbviewer.org/github/genkuroki/Statistics/blob/master/2022/12%20Hypothesis%20testing%20and%20confidence%20interval%20Two%20means.ipynb>)

で非常に詳しく解説したが, 手続きが複雑なので以下で再度まとめておくことにする.

#### 3.1 Welchの t 検定のP値と信頼区間の定義

In [ ]:

1