

検定と信頼区間: 比率の比較

- 黒木玄
- 2022-06-14～2022-06-18

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

- [Julia言語のインストールの仕方の一例 \(https://nbviewer.org/github/genkuroki/msfd28/blob/master/install.ipynb\)](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 比率の比較に関するP値と信頼区間](#)
 - [1.1 比率の比較に関するP値と信頼区間を使って行いたいこと](#)
 - [1.2 2×2の分割表型データとその2つの二項分布モデル](#)
 - [1.3 比率の違いを表す2つの指標: オッズ比パラメータ OR とリスク比パラメータ RR](#)
- ▼ [2 Wald版のP値と信頼区間](#)
 - [2.1 Wald版のオッズ比に関するP値と信頼区間の定義](#)
 - [2.2 Wald版のオッズ比に関するP値と信頼区間の計算例](#)
 - [2.2.1 WolframAlphaによるWald版のオッズ比に関するP値と信頼区間の計算の仕方](#)
 - [2.2.2 Julia言語によるWald版のオッズ比に関するP値と信頼区間の計算の仕方\(1\)](#)
 - [2.2.3 Julia言語によるWald版のオッズ比に関するP値と信頼区間の計算の仕方\(2\)](#)
 - [2.2.4 Julia言語によるWald版のオッズ比に関するP値函数の視覚化](#)
 - [2.2.5 R言語でのWald版のオッズ比に関するP値と信頼区間の計算の仕方](#)
 - [2.3 確率変数の対数オッズ比の分散の近似値の導出](#)
 - [2.4 問題: 対数リスク比の場合の正規分布近似](#)
 - [2.5 問題: Wald版のリスク比に関するP値と信頼区間の定義](#)
 - [2.6 問題: Wald版のリスク比に関するP値と信頼区間の計算例](#)
 - [2.6.1 Julia言語によるWald版のリスク比に関するP値と信頼区間の計算の仕方](#)
 - [2.6.2 Julia言語によるWald版のリスク比に関するP値函数の視覚化](#)
 - [2.6.3 R言語によるWald版のリスク比に関する信頼区間の計算の仕方](#)
 - [2.6.4 Wald版のリスク比に関するP値と信頼区間の計算例の問題の解答例](#)
 - [2.7 問題: 比率の差の場合の正規分布近似](#)
 - [2.8 問題: Wald版の比率の差に関するP値と信頼区間の定義](#)
 - [2.9 問題: Wald版の比率の差に関するP値と信頼区間の計算例](#)
 - [2.9.1 Julia言語によるWald版の比率の差に関するP値と信頼区間の計算の仕方](#)
 - [2.9.2 Wald版の比率の差に関するP値と信頼区間の計算の解答例](#)
- ▼ [3 必修問題: Wald版のオッズ比とリスク比のP値と信頼区間](#)
 - [3.1 WolframAlphaによる必修問題の解答例](#)
 - [3.2 Julia言語による必修問題の解答例](#)
 - [3.3 必修問題解答例](#)
- ▼ [4 Pearsonの \$\chi^2\$ 検定版のP値と信頼区間](#)
 - [4.1 Pearsonの \$\chi^2\$ 検定版のオッズ比に関するP値と信頼区間の定義](#)
 - [4.2 Pearsonの \$\chi^2\$ 検定版のオッズ比に関するP値と信頼区間の計算例](#)
 - [4.2.1 WolframAlphaによるPearsonの \$\chi^2\$ 検定のP値の計算の仕方](#)
 - [4.2.2 Julia言語によるPearsonの \$\chi^2\$ 検定版のオッズ比に関するP値と信頼区間の計算の仕方\(1\)](#)
 - [4.2.3 Julia言語によるPearsonの \$\chi^2\$ 検定版のオッズ比に関するP値と信頼区間の計算の仕方\(2\)](#)
 - [4.2.4 Julia言語によるPearsonの \$\chi^2\$ 検定版のオッズ比に関するP値の視覚化](#)
 - [4.2.5 R言語によるPearsonの \$\chi^2\$ 検定のP値の計算の仕方](#)
 - [4.3 「オッズ比パラメータはOR=ωである」という仮説下での最尤法との関係](#)
 - [4.3.1 2つの二項分布モデル全体での最尤法](#)
 - [4.3.2 条件OR=ωで制限した2つの二項分布モデルでの最尤法](#)
 - [4.3.3 Wilks's theoremの適用](#)
 - [4.4 Wilks' theoremの数値的確認例](#)

| |
|--|
| 4.4.1 2つの二項分布モデルのパラメータを2つとも固定した場合(自由度2) |
| 4.4.2 2つの二項分布モデルの独立性に関する対数尤度比検定の場合(OR=1とした場合,自由度1) |
| 4.4.3 正規分布モデルのパラメータを2つとも固定した場合(自由度2) |
| 4.4.4 正規分布モデルでαのみを固定した場合 |
| 4.4.5 正規分布モデルでμのみを固定した場合(自由度1) |
| 4.4.6 ガンマ分布モデルのパラメータを2つとも固定した場合(自由度2) |
| 4.4.7 ガンマ分布モデルで形状パラメータαのみを固定した場合(自由度1) |
| 4.5 問題: 「リスク比パラメータは$RR=\rho$である」という仮説下での最尤法 |
| 4.6 問題: Pearsonのχ^2検定版のリスク比に関するP値と信頼区間の定義 |
| 4.7 問題: Pearsonのχ^2検定版のリスク比に関するP値と信頼区間の計算例 |
| 4.7.1 Julia言語によるPearsonのχ^2検定版のリスク比に関するP値と信頼区間の計算例 |
| 4.7.2 Julia言語によるPearsonのχ^2検定版のリスク比に関するP値函数の視覚化 |
| 4.7.3 Pearsonのχ^2検定版のリスク比に関するP値と信頼区間の計算の解答例 |
| 4.8 ニューサンスパラメータの問題に関する注意 |
| 5 Fisher検定版のオッズ比に関するP値と信頼区間 |
| 5.1 条件付き確率分布を利用するアイデア |
| 5.2 Sterne型のFisher検定のP値とそれに付随する信頼区間の定義 |
| 5.3 Clopper-Pearson型のFisher検定のP値とそれに付随する信頼区間 |
| 5.4 問題: Fisher検定版のオッズ比に関するP値と信頼区間の計算例 |
| 5.4.1 Julia言語によるFisher検定版のオッズ比に関するP値と信頼区間の計算例 |
| 5.4.2 Julia言語によるFisherχ^2検定のオッズ比に関するP値函数の視覚化 |
| 5.4.3 R言語によるFisher検定版のP値と信頼区間の計算の仕方 |
| 5.4.4 Fisher検定版のオッズ比に関するP値と信頼区間の計算の解答例 |
| 5.5 Fisher検定における第一種の過誤の確率 |
| 6 おまけ: Bayes信用区間 |
| 6.1 オッズ比パラメータとリスク比パラメータのBayes信用区間 |
| 6.2 Bayes信用区間の計算例 |
| 6.3 Bayes信用区間に付随するP値函数 |
| 6.4 Bayes版P値函数の視覚化 |
| 6.5 Bayes版P値函数の視覚化の別の例 |
| 7 関連講義動画 |

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 Printf
9 using QuadGK
10 using RCall
11 using Random
12 Random.seed!(4649373)
13 using Roots
14 using SpecialFunctions
15 using StaticArrays
16 using StatsBase
17 using StatsFuns
18 using StatsPlots
19 default(fmt = :png, size = (400, 250),
20         titlefontsize = 10, plot_titlefontsize = 12)
21 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}{c} " * 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 : isnan(x) ? typeof(x)(Inf) : x*y
2 safediv(x, y) = x == 0 ? x : isnan(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 oddsratiohat(a, b, c, d) = safediv(a*d, b*c)
2 stderr_logoddsratiohat(a, b, c, d) = sqrt(1/a + 1/b + 1/c + 1/d)
3
4 function pvalue_or_wald(a, b, c, d; ω=1)
5     logORhat = log(oddsratiohat(a, b, c, d))
6     SEhat_logORhat = stderr_logoddsratiohat(a, b, c, d)
7     2ccdf(Normal(0, 1), safediv(abs(logORhat - log(ω)), SEhat_logORhat))
8 end
9
10 function confint_or_wald(a, b, c, d; α=0.05)
11     z = quantile(Normal(), 1-α/2)
12     ORhat = oddsratiohat(a, b, c, d)
13     SEhat_logORhat = stderr_logoddsratiohat(a, b, c, d)
14     [safemul(exp(-z*SEhat_logORhat), ORhat), safemul(exp(z*SEhat_logORhat), ORhat)]
15 end
```

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

```
In [5]: 1 @show confint_or_wald(0, 1, 1, 1) confint_or_wald(1e-4, 1, 1, 1)
2 @show confint_or_wald(1, 0, 1, 1) confint_or_wald(1, 1e-4, 1, 1)
3 @show confint_or_wald(1, 1, 0, 1) confint_or_wald(1, 1, 1e-4, 1)
4 @show confint_or_wald(1, 1, 1, 0) confint_or_wald(1, 1, 1, 1e-4)
5 @show confint_or_wald(0, 0, 1, 1) confint_or_wald(1e-4, 1e-4, 1, 1)
6 @show confint_or_wald(1, 1, 0, 0) confint_or_wald(1, 1, 1e-4, 1e-4)
7 @show confint_or_wald(0, 1, 0, 1) confint_or_wald(1e-4, 1, 1e-4, 1)
8 @show confint_or_wald(1, 0, 1, 0) confint_or_wald(1, 1e-4, 1, 1e-4);

confint_or_wald(0, 1, 1, 1) = [0.0, Inf]
confint_or_wald(0.0001, 1, 1, 1) = [7.363403724036157e-90, 1.3580675968312418e81]
confint_or_wald(1, 0, 1, 1) = [0.0, Inf]
confint_or_wald(1, 0.0001, 1, 1) = [7.363403724036156e-82, 1.3580675968312418e89]
confint_or_wald(1, 1, 0, 1) = [0.0, Inf]
confint_or_wald(1, 1, 0.0001, 1) = [7.363403724036156e-82, 1.3580675968312418e89]
confint_or_wald(1, 1, 1, 0) = [0.0, Inf]
confint_or_wald(1, 1, 1, 0.0001) = [7.363403724036157e-90, 1.3580675968312418e81]
confint_or_wald(0, 0, 1, 1) = [0.0, Inf]
confint_or_wald(0.0001, 0.0001, 1, 1) = [4.1295675438516624e-121, 2.4215610699693177e120]
confint_or_wald(1, 1, 0, 0) = [0.0, Inf]
confint_or_wald(1, 1, 0.0001, 0.0001) = [4.1295675438516624e-121, 2.4215610699693177e120]
confint_or_wald(0, 1, 0, 1) = [0.0, Inf]
confint_or_wald(0.0001, 1, 0.0001, 1) = [4.1295675438516624e-121, 2.4215610699693177e120]
confint_or_wald(1, 0, 1, 0) = [0.0, Inf]
confint_or_wald(1, 0.0001, 1, 0.0001) = [4.1295675438516624e-121, 2.4215610699693177e120]
```

```
In [6]: 1 riskratiohat(a, b, c, d) = safediv(a*(c+d), (a+b)*c)
2 stderr_logriskratiohat(a, b, c, d) = sqrt(1/a - 1/(a+b) + 1/c - 1/(c+d))
3
4 function pvalue_rr_wald(a, b, c, d; ρ=1)
5     logRRhat = log(riskratiohat(a, b, c, d))
6     SEhat_logRRhat = stderr_logriskratiohat(a, b, c, d)
7     2ccdf(Normal(0, 1), safediv(abs(logRRhat - log(ρ)), SEhat_logRRhat))
8 end
9
10 function confint_rr_wald(a, b, c, d; α=0.05)
11     (a+b==0 || c+d==0) && return [0, Inf]
12     z = quantile(Normal(), 1-α/2)
13     RRhat = riskratiohat(a, b, c, d)
14     SEhat_logRRhat = stderr_logriskratiohat(a, b, c, d)
15     [safemul(exp(-z*SEhat_logRRhat), RRhat), safemul(exp(z*SEhat_logRRhat), RRhat)]
16 end
```

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

```
In [7]: 1 @show confint_rr_wald(0, 1, 1, 1) confint_rr_wald(1e-4, 1, 1, 1)
2 @show confint_rr_wald(1, 0, 1, 1) confint_rr_wald(1, 1e-4, 1, 1)
3 @show confint_rr_wald(1, 1, 0, 1) confint_rr_wald(1, 1, 1e-4, 1)
4 @show confint_rr_wald(1, 1, 1, 0) confint_rr_wald(1, 1, 1, 1e-4)
5 @show confint_rr_wald(0, 0, 1, 1) confint_rr_wald(1e-4, 1e-4, 1, 1)
6 @show confint_rr_wald(1, 1, 0, 0) confint_rr_wald(1, 1, 1e-4, 1e-4)
7 @show confint_rr_wald(0, 1, 0, 1) confint_rr_wald(1e-4, 1, 1e-4, 1)
8 @show confint_rr_wald(1, 0, 1, 0) confint_rr_wald(1, 1e-4, 1, 1e-4);

confint_rr_wald(0, 1, 1, 1) = [0.0, Inf]
confint_rr_wald(0.0001, 1, 1, 1) = [1.5239118697601845e-89, 2.62429881894243e81]
confint_rr_wald(1, 0, 1, 1) = [0.5001953065198115, 7.996876315834783]
confint_rr_wald(1, 0.0001, 1, 1) = [0.5000759918604454, 7.997184798065742]
confint_rr_wald(1, 1, 0, 1) = [0.0, Inf]
confint_rr_wald(1, 1, 0.0001, 1) = [3.810541668433138e-82, 6.562059262373016e88]
confint_rr_wald(1, 1, 1, 0) = [0.12504882662995287, 1.9992190789586957]
confint_rr_wald(1, 1, 1, 0.0001) = [0.12504400301489435, 1.999696078749301]
confint_rr_wald(0, 0, 1, 1) = [0.0, Inf]
confint_rr_wald(0.0001, 0.0001, 1, 1) = [6.426171133615772e-61, 1.5561365846124554e60]
confint_rr_wald(1, 1, 0, 0) = [0.0, Inf]
confint_rr_wald(1, 1, 0.0001, 0.0001) = [6.426171133615772e-61, 1.5561365846124554e60]
confint_rr_wald(0, 1, 0, 1) = [0.0, Inf]
confint_rr_wald(0.0001, 1, 0.0001, 1) = [4.245626444711269e-121, 2.3553650162644178e120]
confint_rr_wald(1, 0, 1, 0) = [1.0, 1.0]
confint_rr_wald(1, 0.0001, 1, 0.0001) = [0.9726638925089683, 1.0281043716144522]
```

```
In [8]: 1 riskdiffhat(a, b, c, d) = a/(a+b) - c/(c+d)
2
3 function stderr_riskdiffhat(a, b, c, d)
4     m, n = a+b, c+d
5     p̂, q̂ = a/m, c/n
6     √(p̂*(1-p̂)/m + q̂*(1-q̂)/n)
7 end
8
9 function pvalue_rd_wald(a, b, c, d; Δ=0)
10    RDhat = riskdiffhat(a, b, c, d)
11    SEhat_riskdiffhat = stderr_riskdiffhat(a, b, c, d)
12    2ccdf(Normal(0, 1), safediv(abs(RDhat - Δ), SEhat_riskdiffhat))
13 end
14
15 function confint_rd_wald(a, b, c, d; α=0.05)
16    (a+b==0 || c+d==0) && return [-Inf, Inf]
17    z = quantile(Normal(), 1-α/2)
18    RDhat = riskdiffhat(a, b, c, d)
19    SEhat_riskdiffhat = stderr_riskdiffhat(a, b, c, d)
20    [RDhat - z*SEhat_riskdiffhat, RDhat + z*SEhat_riskdiffhat]
21 end
```

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

```
In [9]: 1 @show confint_rd_wald(0, 1, 1, 1) confint_rd_wald(1e-4, 1, 1, 1)
2 @show confint_rd_wald(1, 0, 1, 1) confint_rd_wald(1, 1e-4, 1, 1)
3 @show confint_rd_wald(1, 1, 0, 1) confint_rd_wald(1, 1, 1e-4, 1)
4 @show confint_rd_wald(1, 1, 1, 0) confint_rd_wald(1, 1, 1, 1e-4)
5 @show confint_rd_wald(0, 0, 1, 1) confint_rd_wald(1e-4, 1e-6, 1, 1)
6 @show confint_rd_wald(1, 1, 0, 0) confint_rd_wald(1, 1, 1e-4, 1e-6)
7 @show confint_rd_wald(0, 1, 0, 1) confint_rd_wald(1e-4, 1, 1e-6, 1)
8 @show confint_rd_wald(1, 0, 1, 0) confint_rd_wald(1, 1e-4, 1, 1e-6);

confint_rd_wald(0, 1, 1, 1) = [-1.1929519121748402, 0.1929519121748402]
confint_rd_wald(0.0001, 1, 1, 1) = [-1.1931289644203495, 0.19332894442234927]
confint_rd_wald(1, 0, 1, 1) = [-0.1929519121748402, 1.1929519121748402]
confint_rd_wald(1, 0.0001, 1, 1) = [-0.19332894442234927, 1.1931289644203495]
confint_rd_wald(1, 1, 0, 1) = [-0.1929519121748402, 1.1929519121748402]
confint_rd_wald(1, 1, 0.0001, 1) = [-0.19332894442234927, 1.1931289644203495]
confint_rd_wald(1, 1, 1, 0) = [-1.1929519121748402, 0.1929519121748402]
confint_rd_wald(1, 1, 1, 0.0001) = [-1.1931289644203495, 0.19332894442234927]
confint_rd_wald(0, 0, 1, 1) = [-Inf, Inf]
confint_rd_wald(0.0001, 1.0e-6, 1, 1) = [-18.831608752513155, 19.811806772315137]
confint_rd_wald(1, 1, 0, 0) = [-Inf, Inf]
confint_rd_wald(1, 1, 0.0001, 1.0e-6) = [-19.811806772315137, 18.831608752513155]
confint_rd_wald(0, 1, 0, 1) = [0.0, 0.0]
confint_rd_wald(0.0001, 1, 1.0e-6, 1) = [-0.019595478984140494, 0.019793458988140295]
confint_rd_wald(1, 0, 1, 0) = [0.0, 0.0]
confint_rd_wald(1, 0.0001, 1, 1.0e-6) = [-0.019793458988136028, 0.019595478984136074]
```

In [10]:

```

1 function delta(a, b, c, d; ω=1)
2     A, B, C = 1-ω, a+d+ω*(b+c), a*d-ω*b*c
3     isinf(ω) ? typeof(ω)(-min(b, c)) : safediv(2C, B + √(B^2 - 4A*C))
4 end
5
6 # correction = 0.5 は連続性補正を与える。
7 function _chisqstat_or(a, b, c, d, δ; correction=0.0)
8     ā, ī, ē, ī = a-δ, b+δ, c+δ, d-δ
9     safemul(max(0, abs(δ)-correction)^2, 1/ā + 1/ī + 1/ē + 1/ī)
10 end
11
12 function chisqstat_or(a, b, c, d; ω=1, correction=0.0)
13     δ = delta(a, b, c, d; ω)
14     _chisqstat_or(a, b, c, d, δ; correction)
15 end
16
17 function pvalue_or_pearson_chisq(a, b, c, d; ω=1, correction=0.0)
18     χ² = chisqstat_or(a, b, c, d; ω, correction)
19     ccdf(Chisq(1), χ²)
20 end
21
22 function confint_or_pearson_chisq(a, b, c, d; α=0.05, correction=0.0)
23     (a+b==0 || c+d==0 || a+c==0 || b+d==0) && return [0, Inf]
24     f(logω) = logit(pvalue_or_pearson_chisq(a, b, c, d; ω=exp(logω), correction)) - logit(α)
25     ps = if a == 0 || d == 0
26         [0, exp(find_zero(f, 0.0))]
27     elseif b == 0 || c == 0
28         [exp(find_zero(f, 0.0)), Inf]
29     else
30         ORhat = oddsratiohat(a, b, c, d)
31         ω_L, ω_U = ORhat/2, 2ORhat
32         [exp(find_zero(f, log(ω_L))), exp(find_zero(f, log(ω_U)))]
33     end
34 end

```

Out[10]: confint_or_pearson (generic function with 1 method)

In [11]:

```

1 @show confint_or_pearson_chisq(0, 1, 1, 1) confint_or_pearson_chisq(1e-4, 1, 1, 1)
2 @show confint_or_pearson_chisq(1, 0, 1, 1) confint_or_pearson_chisq(1, 1e-4, 1, 1)
3 @show confint_or_pearson_chisq(1, 1, 0, 1) confint_or_pearson_chisq(1, 1, 1e-4, 1)
4 @show confint_or_pearson_chisq(1, 1, 1, 0) confint_or_pearson_chisq(1, 1, 1, 1e-4)
5 @show confint_or_pearson_chisq(0, 0, 1, 1) confint_or_pearson_chisq(1e-4, 1e-4, 1, 1)
6 @show confint_or_pearson_chisq(1, 1, 0, 0) confint_or_pearson_chisq(1, 1, 1e-4, 1e-4)
7 @show confint_or_pearson_chisq(0, 1, 0, 1) confint_or_pearson_chisq(1e-4, 1, 1e-4, 1)
8 @show confint_or_pearson_chisq(1, 0, 1, 0) confint_or_pearson_chisq(1, 1e-4, 1, 1e-4);

confint_or_pearson(0, 1, 1, 1) = [0.0, 11.137973937616904]
confint_or_pearson(0.0001, 1, 1, 1) = [2.602261452525905e-9, 11.140095739591606]
confint_or_pearson(1, 0, 1, 1) = [0.08978293589129746, Inf]
confint_or_pearson(1, 0.0001, 1, 1) = [0.08976583535507918, 3.842811409403394e8]
confint_or_pearson(1, 1, 0, 1) = [0.08978293589129746, Inf]
confint_or_pearson(1, 1, 0.0001, 1) = [0.08976583535507918, 3.842811409403394e8]
confint_or_pearson(1, 1, 1, 0) = [0.0, 11.137973937616904]
confint_or_pearson(1, 1, 1, 0.0001) = [2.602261452525905e-9, 11.140095739591601]
confint_or_pearson(0, 0, 1, 1) = [0.0, Inf]
confint_or_pearson(0.0001, 0.0001, 1, 1) = [1.3012947051505169e-5, 76846.54337277808]
confint_or_pearson(1, 1, 0, 0) = [0.0, Inf]
confint_or_pearson(1, 1, 0.0001, 0.0001) = [1.3012947051505169e-5, 76846.54337277808]
confint_or_pearson(0, 1, 0, 1) = [0.0, Inf]
confint_or_pearson(0.0001, 1, 0.0001, 1) = [1.3012947051505169e-5, 76846.54337277808]
confint_or_pearson(1, 0, 1, 0) = [0.0, Inf]
confint_or_pearson(1, 0.0001, 1, 0.0001) = [1.3012947051505169e-5, 76846.54337277808]

```

次のセルの計算結果を Cornfield (1956), p.139, (4.2) と比較せよ。

- Jerome Cornfield, A Statistical Problem Arising from Retrospective Studies, Berkeley Symposium on Mathematical Statistics and Probability, 1956: 135-148 (1956) [[link](https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Proceedings%20of%20the%20Third%20Berkeley%20Symposium%20on%20Mathematical%20Statistics%20and%20Probability) (<https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Proceedings%20of%20the%20Third%20Berkeley%20Symposium%20on%20Mathematical%20Statistics%20and%20Probability>)]

(4.2)

$$z'_2 = 0.6229$$

$$z'_1 = 0.0296.$$

In [12]:

```
1 # Cornfeild (1956), p.139, (4.2) と比較せよ.
2 a, b, c, d = 3, 11, 60, 32
3 @show confint_or_wald(a, b, c, d)
4 @show confint_or_pearson_chisq(a, b, c, d)
5 @show confint_or_pearson_chisq(a, b, c, d; correction=0.5);

confint_or_wald(a, b, c, d) = [0.037829803607039435, 0.5592686923030045]
confint_or_pearson(a, b, c, d) = [0.04084291410483445, 0.5253472140108638]
confint_or_pearson(a, b, c, d; correction = 0.5) = [0.02962955731952911, 0.6228119542455192]
```

上のセルで計算した連続補正版の信頼区間の値が Cornfeild (1956), p.139, (4.2) の値に一致している。

In [13]:

```
1 function Delta(a, b, c, d; ρ=1)
2     m, n = a+b, c+d
3     A, B, C = ρ-1, n-a+ρ*(m-c), a*n-ρ*m*c
4     isnan(ρ) ? typeof(ω)(-c) : safediv(2C, B + √(B^2 - 4A*C))
5 end
6
7 function _chisqstat_rr(a, b, c, d, Δ)
8     m, n = a+b, c+d
9     safemul(Δ^2, safediv(b, m*(a-Δ)) + safediv(d, n*(c+Δ)))
10 end
11
12 function chisqstat_rr(a, b, c, d; ρ=1)
13     Δ = Delta(a, b, c, d; ρ)
14     _chisqstat_rr(a, b, c, d, Δ)
15 end
16
17 function pvalue_rr_pearson_chisq(a, b, c, d; ρ=1)
18     χ² = chisqstat_rr(a, b, c, d; ρ)
19     ccdf(Chisq(1), χ²)
20 end
21
22 function confint_rr_pearson_chisq(a, b, c, d; α=0.05)
23     (a+b==0 || c+d==0 || a+c==0 || b+d==0) && return [0, Inf]
24     f(logρ) = logit(pvalue_rr_pearson_chisq(a, b, c, d; ρ=exp(logρ))) - logit(α)
25     RRhat = riskratiohat(a, b, c, d)
26     if a == 0
27         [0.0, exp(find_zero(f, 0.0))]
28     elseif c == 0
29         [exp(find_zero(f, 0.0)), Inf]
30     elseif b == 0
31         [0.0, exp(find_zero(f, log(2RRhat)))]
32     elseif d == 0
33         [exp(find_zero(f, log(RRhat/2))), Inf]
34     else
35         ρ_L, ρ_U = RRhat/2, 2RRhat
36         [exp(find_zero(f, log(ρ_L))), exp(find_zero(f, log(ρ_U)))]
37     end
38 end
```

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

In [14]:

```
1 @show confint_rr_pearson_chisq(0, 1, 1, 1) confint_rr_pearson_chisq(1e-4, 1, 1, 1)
2 @show confint_rr_pearson_chisq(1, 0, 1, 1) confint_rr_pearson_chisq(1, 1e-4, 1, 1)
3 @show confint_rr_pearson_chisq(1, 1, 0, 1) confint_rr_pearson_chisq(1, 1, 1e-4, 1)
4 @show confint_rr_pearson_chisq(1, 1, 1, 0) confint_rr_pearson_chisq(1, 1, 1, 1e-4)
5 @show confint_rr_pearson_chisq(0, 0, 1, 1) confint_rr_pearson_chisq(1e-4, 1e-4, 1, 1)
6 @show confint_rr_pearson_chisq(1, 1, 0, 0) confint_rr_pearson_chisq(1, 1, 1e-4, 1e-4)
7 @show confint_rr_pearson_chisq(0, 1, 0, 1) confint_rr_pearson_chisq(1e-4, 1, 1e-4, 1)
8 @show confint_rr_pearson_chisq(1, 0, 1, 0) confint_rr_pearson_chisq(1, 1e-4, 1, 1e-4);
```

```
confint_rr_pearson(0, 1, 1, 1) = [0.0, 4.10417860525958]
confint_rr_pearson(0.0001, 1, 1, 1) = [5.205303607358693e-9, 4.104616767401084]
confint_rr_pearson(1, 0, 1, 1) = [0.0, 10.578517350254106]
confint_rr_pearson(1, 0.0001, 1, 1) = [0.32665454735880284, 10.577873776111911]
confint_rr_pearson(1, 1, 0, 1) = [0.2436541135706135, Inf]
confint_rr_pearson(1, 1, 0.0001, 1) = [0.24362810383225364, 1.9211175282574818e8]
confint_rr_pearson(1, 1, 1, 0) = [0.09453120573423061, Inf]
confint_rr_pearson(1, 1, 1, 0.0001) = [0.0945369571584704, 3.0613380652042177]
confint_rr_pearson(0, 0, 1, 1) = [0.0, Inf]
confint_rr_pearson(0.0001, 0.0001, 1, 1) = [2.602945936963888e-5, 9.681923544184194]
confint_rr_pearson(1, 1, 0, 0) = [0.0, Inf]
confint_rr_pearson(1, 1, 0.0001, 0.0001) = [0.10328526097487344, 38418.00883361873]
confint_rr_pearson(0, 1, 0, 1) = [0.0, Inf]
confint_rr_pearson(0.0001, 1, 0.0001, 1) = [1.3016851156610122e-5, 76823.49501927945]
confint_rr_pearson(1, 0, 1, 0) = [0.0, Inf]
confint_rr_pearson(1, 0.0001, 1, 0.0001) = [0.20653685531726315, 4.84175087521499]
```

In [15]:

```
1 _pdf_le(x, (dist, y)) = pdf(dist, x) ≈ y
2
3 function _search_boundary(f, x0, Δx, param)
4     x = x0
5     if f(x, param)
6         while f(x - Δx, param) x -= Δx end
7     else
8         x += Δx
9         while !f(x, param) x += Δx end
10    end
11    x
12 end
13
14 function pvalue_sterne(dist::DiscreteUnivariateDistribution, x)
15     Px = pdf(dist, x)
16     Px == 0 && return Px
17     Px == 1 && return Px
18     m = mode(dist)
19     Px ≈ pdf(dist, m) && return one(Px)
20     if x < m
21         y = _search_boundary(_pdf_le, 2m - x, 1, (dist, Px))
22         cdf(dist, x) + ccdf(dist, y-1)
23     else # x > m
24         y = _search_boundary(_pdf_le, 2m - x, -1, (dist, Px))
25         cdf(dist, y) + ccdf(dist, x-1)
26     end
27 end
28
29 function pvalue_or_sterne(a, b, c, d; ω=1)
30     fnch = if ω == 1
31         Hypergeometric(a+b, c+d, a+c)
32     else
33         FisherNoncentralHypergeometric(a+b, c+d, a+c, ω)
34     end
35     pvalue_sterne(fnch, a)
36 end
37
38 function confint_or_sterne(a, b, c, d; α = 0.05)
39     (a+b==0 || c+d==0 || a+c==0 || b+d==0) && return [0, Inf]
40     f(logω) = logit(pvalue_or_sterne(a, b, c, d; ω=exp(logω))) - logit(α)
41     if a == 0 || d == 0
42         [0.0, exp(find_zero(f, 0.0))]
43     elseif b == 0 || c == 0
44         [exp(find_zero(f, 0.0)), Inf]
45     else
46         ω_L, ω_U = confint_or_wald(a, b, c, d; α = α/10)
47         ps = exp.(find_zeros(f, log(ω_L), log(ω_U)))
48         # 次の行は稀に区間にならない場合への対策
49         [first(ps), last(ps)]
50     end
51 end
```

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

```
In [16]: 1 @show confint_or_sterne(0, 10, 10, 10) confint_or_pearson_chisq(1e-4, 10, 10, 10)
2 @show confint_or_sterne(10, 0, 10, 10) confint_or_pearson_chisq(10, 1e-4, 10, 10)
3 @show confint_or_sterne(10, 10, 0, 10) confint_or_pearson_chisq(10, 10, 1e-4, 10)
4 @show confint_or_sterne(10, 10, 10, 0) confint_or_pearson_chisq(10, 10, 10, 1e-4)
5 @show confint_or_sterne(0, 0, 10, 10) confint_or_pearson_chisq(1e-4, 1e-4, 10, 10)
6 @show confint_or_sterne(10, 10, 0, 0) confint_or_pearson_chisq(10, 10, 1e-4, 1e-4)
7 @show confint_or_sterne(0, 10, 0, 10) confint_or_pearson_chisq(1e-4, 10, 1e-4, 10)
8 @show confint_or_sterne(10, 0, 10, 0) confint_or_pearson_chisq(10, 1e-4, 10, 1e-4);
```

```
confint_or_sterne(0, 10, 10, 10) = [0.0, 0.5431888559914949]
confint_or_pearson(0.0001, 10, 10, 10) = [2.6029641085026625e-10, 0.44168708894449726]
confint_or_sterne(10, 0, 10, 10) = [1.8409803311864292, Inf]
confint_or_pearson(10, 0.0001, 10, 10) = [2.264046255890583, 3.8417740633827114e9]
confint_or_sterne(10, 10, 0, 10) = [1.8409803311864295, Inf]
confint_or_pearson(10, 10, 0.0001, 10) = [2.264046255890583, 3.841774063392443e9]
confint_or_sterne(10, 10, 10, 0) = [0.0, 0.5431888559914949]
confint_or_pearson(10, 10, 10, 0.0001) = [2.6029641085026625e-10, 0.44168708894449726]
confint_or_sterne(0, 0, 10, 10) = [0.0, Inf]
confint_or_pearson(0.0001, 0.0001, 10, 10) = [1.301528947084912e-5, 76832.712959483]
confint_or_sterne(10, 10, 0, 0) = [0.0, Inf]
confint_or_pearson(10, 10, 0.0001, 0.0001) = [1.301528947084912e-5, 76832.7129594542]
confint_or_sterne(0, 10, 0, 10) = [0.0, Inf]
confint_or_pearson(0.0001, 10, 0.0001, 10) = [1.301528947084912e-5, 76832.7129594542]
confint_or_sterne(10, 0, 10, 0) = [0.0, Inf]
confint_or_pearson(10, 0.0001, 10, 0.0001) = [1.301528947084912e-5, 76832.712959483]
```

```
In [17]: 1 function pvalue_or_clopper_pearson_chisq(a, b, c, d; w=1)
2     fnch = if w == 1
3         Hypergeometric(a+b, c+d, a+c)
4     else
5         FisherNoncentralHypergeometric(a+b, c+d, a+c, w)
6     end
7     min(1, 2cdf(fnch, a), 2ccdf(fnch, a-1))
8 end
9
10 function confint_or_clopper_pearson_chisq(a, b, c, d; α = 0.05)
11     (a+b==0 || c+d==0 || a+c==0 || b+d==0) && return [0, Inf]
12     f(w) = logit(pvalue_or_clopper_pearson_chisq(a, b, c, d; w)) - logit(α)
13     if a == 0 || d == 0
14         [0.0, find_zero(f, 1.0)]
15     elseif b == 0 || c == 0
16         [find_zero(f, 1.0), Inf]
17     else
18         w_L, w_U = confint_or_wald(a, b, c, d; α = α/10)
19         find_zeros(f, w_L, w_U)
20     end
21 end
```

```
Out[17]: confint_or_clopper_pearson (generic function with 1 method)
```

```
In [18]: 1 @show confint_or_clopper_pearson_chisq(0, 10, 10, 10) confint_or_sterne(0, 10, 10, 10)
2 @show confint_or_clopper_pearson_chisq(10, 0, 10, 10) confint_or_sterne(10, 0, 10, 10)
3 @show confint_or_clopper_pearson_chisq(10, 10, 0, 10) confint_or_sterne(10, 10, 0, 10)
4 @show confint_or_clopper_pearson_chisq(10, 10, 10, 0) confint_or_sterne(10, 10, 10, 0)
5 @show confint_or_clopper_pearson_chisq(0, 0, 10, 10) confint_or_sterne(0, 0, 10, 10)
6 @show confint_or_clopper_pearson_chisq(10, 10, 0, 0) confint_or_sterne(10, 10, 0, 0)
7 @show confint_or_clopper_pearson_chisq(0, 10, 0, 10) confint_or_sterne(0, 10, 0, 10)
8 @show confint_or_clopper_pearson_chisq(10, 0, 10, 0) confint_or_sterne(10, 0, 10, 0);
```

```
confint_or_clopper_pearson(0, 10, 10, 10) = [0.0, 0.6246724957742202]
confint_or_sterne(0, 10, 10, 10) = [0.0, 0.5431888559914949]
confint_or_clopper_pearson(10, 0, 10, 10) = [1.6008388503812658, Inf]
confint_or_sterne(10, 0, 10, 10) = [1.8409803311864292, Inf]
confint_or_clopper_pearson(10, 10, 0, 10) = [1.600838850381266, Inf]
confint_or_sterne(10, 10, 0, 10) = [1.8409803311864295, Inf]
confint_or_clopper_pearson(10, 10, 10, 0) = [0.0, 0.6246724957742202]
confint_or_sterne(10, 10, 10, 0) = [0.0, 0.5431888559914949]
confint_or_clopper_pearson(0, 0, 10, 10) = [0.0, Inf]
confint_or_sterne(0, 0, 10, 10) = [0.0, Inf]
confint_or_clopper_pearson(10, 10, 0, 0) = [0.0, Inf]
confint_or_sterne(10, 10, 0, 0) = [0.0, Inf]
confint_or_clopper_pearson(0, 10, 0, 10) = [0.0, Inf]
confint_or_sterne(0, 10, 0, 10) = [0.0, Inf]
confint_or_clopper_pearson(10, 0, 10, 0) = [0.0, Inf]
confint_or_sterne(10, 0, 10, 0) = [0.0, Inf]
```

```
In [19]: 1 function sim_probabilities_of_type_I_error(m, n, p, q=p;
2         w=p*(1-q)/((1-p)*q), L=10^5)
3     bina, binc = Binomial(m, p), Binomial(n, q)
4     p_sterne = similar(zeros(), L)
5     p_clopper_pearson = similar(zeros(), L)
6     p_pearson = similar(zeros(), L)
7     p_wald = similar(zeros(), L)
8     @threads for i in 1:L
9         a, c = rand(bina), rand(binc)
10        b, d = m-a, n-c
11        p_sterne[i] = pvalue_or_sterne(a, b, c, d; w)
12        p_clopper_pearson[i] = pvalue_or_clopper_pearson_chisq(a, b, c, d; w)
13        p_pearson[i] = pvalue_or_pearson_chisq(a, b, c, d; w)
14        p_wald[i] = pvalue_or_wald(a, b, c, d; w)
15    end
16    (; p_sterne, p_clopper_pearson, p_pearson, p_wald)
17 end
18
19 function plot_probabilities_of_type_I_error(m, n, p, q=p;
20         w=p*(1-q)/((1-p)*q), L=10^6, kwargs...)
21     (; p_sterne, p_clopper_pearson, p_pearson, p_wald) =
22         sim_probabilities_of_type_I_error(m, n, p, q; w, L)
23     F_sterne = ecdf(p_sterne)
24     F_clopper_pearson = ecdf(p_clopper_pearson)
25     F_pearson = ecdf(p_pearson)
26     F_wald = ecdf(p_wald)
27
28     x = 0:0.001:1
29     tick = 0:0.1:1
30     P = plot(; legend=:topleft)
31     plot!(x, x → F_sterne(x); label="Fisher (Sterne)")
32     plot!(x, x → F_clopper_pearson_chisq(x); label="Fisher (CP)", ls=:dash)
33     plot!(x, x → F_pearson_chisq(x); label="Pearson  $\chi^2$ ", ls=:dashdot)
34     plot!(x, x → F_wald(x); label="Wald", ls=:dot, lw=2)
35     plot!(x, identity; label="", c=:black, ls=:dot)
36     plot!(; xtick=tick, ytick=tick, tickfontsize=6)
37     plot!(; xguide=" $\alpha$ ", yguide="probability of type I error",
38             guidefontsize=9)
39
40     x = 0:0.0001:0.1
41     tick = 0:0.01:1
42     Q = plot(; legend=:topleft)
43     plot!(x, x → F_sterne(x); label="Fisher (Sterne)")
44     plot!(x, x → F_clopper_pearson_chisq(x); label="Fisher (CP)", ls=:dash)
45     plot!(x, x → F_pearson_chisq(x); label="Pearson  $\chi^2$ ", ls=:dashdot)
46     plot!(x, x → F_wald(x); label="Wald", ls=:dot, lw=2)
47     plot!(x, identity; label="", c=:black, ls=:dot)
48     plot!(; xtick=tick, ytick=tick, tickfontsize=6)
49     plot!(; xguide=" $\alpha$ ", yguide="probability of type I error",
50             guidefontsize=9)
51
52     plot(P, Q; size=(700, 350), layout=(1, 2))
53     plot!(; plot_title="Binomial(m=$m, p=$p)×Binomial(n=$n, q=$q)",
54             plot_titlefontsize=10)
55     plot!(; kwargs...)
56 end
```

Out[19]: `plot_probabilities_of_type_I_error` (generic function with 2 methods)

1 比率の比較に関するP値と信頼区間

1.1 比率の比較に関するP値と信頼区間を使って行いたいこと

我々は以下のようなことを知りたいとする。

- (1) ウェブページのデザインAとBではどちらの側がどれだけ商品が売れ易いかを知りたい。
- (2) 薬Xを処方した側ではしなかった側よりも症状が悪化しなかった人の割合がどれだけ増えるかを知りたい。

例えば、(1)の場合に次のようなデータが得られたとする:

| | 商品を購入した | 購入しなかった | 合計 |
|-------|---------|---------|------|
| デザインA | 49 | 965 | 1014 |
| デザインB | 26 | 854 | 880 |
| 合計 | 75 | 1819 | 1894 |

このデータの数値は、デザインAの側にアクセスした1014人中の49人が商品を購入し、デザインBの側にアクセスした880人中の26人が商品を購入したことを表している。

このデータの数値からの印象では、デザインAの方がデザインBの方が商品の購入確率が高いように見える。実際、デザインAでの購入者割合は $49/1014 \approx 4.8\%$ 程度で、デザインBでの購入者割合の $26/880 \approx 3.0\%$ より大きい。

しかし、データの確率的揺らぎのせいで単なる偶然で、デザインAの側がよく売れるように見えるデータの数値が得られただけなのかもしれない。(実際には他にも様々な原因で偏ったデータが得られる場合がある。)

もしもそなならば、上のデータに基くウェブページのデザイン採用に関する意思決定は無駄に終わることになってしまう。だから、そのようなリスクの程度を見積もる必要がある。そのための道具がP値や信頼区間である。

このようなデータに適当な統計モデルを適用して、デザインAとデザインBでの購入確率の違いを信頼区間で表したい。

そのためには以下のようにすればよい:

1. 統計モデルの設定.
2. 購入確率の違いを表す指標を導入.
3. その指標の値が〇〇であるという仮説のP値を定義する.
4. そのP値を使って信頼区間を定義する.
5. 以上によって得られたP値函数や信頼区間を利用する.

1.2 2×2の分割表型データとその2つの二項分布モデル

このノートでは次の形の 2×2 の分割表型のデータを扱う:

| | $Y = 1$ | $Y = 0$ | |
|---------|-------------|-------------|---------------------|
| $X = 1$ | a | b | $m = a + b$ |
| $X = 0$ | c | d | $n = c + d$ |
| | $r = a + c$ | $s = b + d$ | $N = a + b + c + d$ |

ここで a, b, c, d は 0 以上の整数である。

このノートでは 2×2 の分割表型データに関する以下の設定の統計モデルを扱う。

(1) 横方向の合計 $m = a + b, n = c + d$ (ゆえに全体の合計 $N = m + n$) は固定されていると仮定する。

(2) a は成功確率パラメータ p の二項分布に従っており、 c は成功確率パラメータ q の二項分布に従っているとし、 a, c は独立であると仮定する。 p, q を比率と呼ぶこともある。

このとき、 a, b, c, d の同時確率質量函数は次のように表される:

$$P(a, b, c, d | m, n, p, q) = \binom{m}{a} p^a (1-p)^b \binom{n}{c} q^c (1-q)^d \\ (a, b, c, d \in \mathbb{Z}_{\geq 0}, a + b = m, c + d = n).$$

これを **2つの二項分布モデル** と呼ぶことにする。

(3) このモデルは上の分割表において、以下が成立しているという設定になっている:

- $X = 1$ のときには $Y = 1$ となる確率は p で、 $Y = 0$ となる確率は $1 - p$ である。
- $X = 0$ のときには $Y = 1$ となる確率は q で、 $Y = 0$ となる確率は $1 - q$ である。

そして、 $X = 1$ となる人達を m 人選び、 $X = 0$ となる人を n 人選んで Y がどうなるかを調べた。

具体的に X, Y として以下のような場合を想定している:

- $X = 1, 0$ はウェブページのデザインがA,Bのどちらであるかを意味し、 $Y = 1, 0$ は商品を購入したかしないかを意味する。
- $X = 1, 0$ は薬Xを処方したか否かを意味し、 $Y = 1, 0$ は症状が悪化したか否かを意味している。 $(Y = 1$ は何らかの基準で症状が悪化したことを意味する。)

注意: このモデルの採用が妥当であるかどうかはデータの数値とは別の情報を使って判断しなければいけない。例えば、比較したい母集団とは異なる偏った母集団からの無作為抽出になってしまっている疑いが強い場合には、モデルをデータの偏りを考慮したものに変更するか、データの取得の仕方を変更するか、どちらかのしくは両方の処置が必要になる。

注意: 分割表の縦方向と横方向の役割を逆転させた場合も数学的には同様である。ただし、データの分析結果の解釈では立場を交換したことによって違いが出るので注意する必要がある。

注意: 以上で説明した2つの二項分布モデル以外に、多項分布(四項分布)モデルや4つのPoisson分布モデルも考えられるが、以下では省略する。実は統計モデルをそのように変更しても以下で説明するP値や信頼区間の計算の仕方はどれも同じになることが知られている。 2×2 の分割表に関する統計モデルについては、

- 「条件付き確率分布、尤度、推定、記述統計」のノート

(<https://nbviewer.org/github/genkuroki/Statistics/blob/master/2022/06%20Conditional%20distribution%2C%20likelihood%2C%20>)

の「 2×2 の分割表の分布」の節を参照せよ。



1.3 比率の違いを表す2つの指標: オッズ比パラメータ OR とリスク比パラメータ RR

比率 $0 < p < 1$ に関する

$$u = \frac{p}{1-p}$$

を **オッズ**(odds)と呼び、この対数 $\log u = \log(p/(1-p))$ を **対数オッズ**(log odds)と呼ぶ。逆に p はオッズ u で

$$p = \frac{1}{1+u}$$

と表されるので、比率 $p = u/(1+u)$ の代わりにオッズ $u = p/(1-p)$ を扱っても同等のことをできる。(この事実は計算を見通り良くするときに役に立つ。)

2つの比率 $0 < p < 1, 0 < q < 1$ に関する

$$\text{OR} = \frac{p/(1-p)}{q/(1-q)} = \frac{p(1-q)}{(1-p)q}$$

を **オッズ比**(odds ratio, OR)と呼び、この対数 $\log \text{OR}$ を **対数オッズ比**(log odds ratio)と呼ぶ:

$$\log \text{OR} = \log \frac{p(1-q)}{(1-p)q} = \log \frac{p}{1-p} - \log \frac{q}{1-q}.$$

さらにこのノートでは、2つの比率 $0 < p < 1, 0 < q < 1$ に関する

$$\text{RR} = \frac{p}{q}$$

を **リスク比**(risk ratio, RR)と呼び、この対数 $\log \text{RR}$ を **対数リスク比**(log risk ratio)と呼ぶ:

$$\log \text{RR} = \log \frac{p}{q} = \log p - \log q.$$

もしも、 p, q の両方が 0 に近ければ、 $1-p \approx 1, 1-q \approx 1$ という近似を使えるので、リスク比はオッズ比で近似される:

$$1-p \approx 1, 1-q \approx 1 \implies \text{RR} \approx \text{OR}.$$

比率 p と q が等しいという条件はオッズ比やリスク比を使って以下のように言い換えられる:

$$p = q \iff \text{OR} = 1 \iff \text{RR} = 1.$$

だから、 p と q が等しいという仮説は $\text{OR} = 1$ や $\text{RR} = 1$ のように表現できる。(もちろん、 $\log \text{OR} = 0$ や $\log \text{RR} = 0$ とも書ける。)

だから、オッズ比 OR とリスク比 RR は2つの比率 p, q の違いを表すための指標として使える。(もちろん、それらの対数版である $\log \text{OR}$ や $\log \text{RR}$ が 0 を p, q の違いを表すための指標として採用してもよい。)

注意: 他にも **比率の差** $p - q$ も比率の違いを表す重要な指標だが、このノートでは問題と解答例の形式で軽く触れるだけにする予定である。

注意: オッズは賭け事ではよく使われる用語である。例えばある勝負で A が勝つことに賭けるときに

$$\text{Aが勝つ確率 : Aが負ける確率} = 1 : 4$$

だと思っているなら、あなたは「オッズは 1 対 4 だ」と思っていることになる。

注意: 例えは確率 p, q が「症状悪化の確率」や「死亡する確率」ならば「リスク」という呼び方は非常に適切だろう。しかし上ではそうでない場合も p/q を「リスク比」と呼ぶことにした。具体的な応用先で、この呼び方に違和感を感じる場合には適宜別の呼び方をするようにして欲しい。

注意: このノートでは比率の違いを表す指標として主にオッズ比を利用する。その理由は、応用上の理由ではなく、単に数学的に扱いやすいからである。そして、以下の注意で説明するように、オッズ比を考えることは、その対数を考えることによって、ロジスティック回帰の話とも関係付けることができる。

注意: p の対数オッズを $x = \log(p/(1-p))$ と書くことにする。このとき、以下の公式によつて、

$$0 < p < 1, \quad 0 < u < \infty, \quad -\infty < x < \infty$$

の3つが一一対応する：

$$\begin{aligned} p &= \frac{u}{1+u} = \frac{1}{1+e^{-x}} = \text{logistic}(x), \\ u &= \frac{p}{1-p} = \exp(x), \\ x &= \log(u) = \log \frac{p}{1-p} = \text{logit}(p). \end{aligned}$$

x を p に対応させる函数は **ロジスティック函数** と呼ばれ、その逆函数は **ロジット函数** と呼ばれる。

注意: 対数オッズ比を $\beta = \log \text{OR}$ と書き、さらに q の対数オッズを $\alpha = \log(q/(1-q))$ と書くと、 p, q の対数オッズはそれぞれ次のように表される：

$$\text{logit}(p) = \log \frac{p}{1-p} = \alpha + \beta, \quad \text{logit}(q) = \log \frac{q}{1-p} = \alpha.$$

これらは次と同値である：

$$p = \text{logistic}(\alpha + \beta), \quad q = \text{logistic}(\alpha).$$

モデルの2つの比率パラメータ p, q をこのように表して、データの数値から α, β の値を推定することを **ロジスティック回帰** (logistic regression)と呼ぶ。(実際にはもっと一般の場合もロジスティック回帰と呼ぶ。)

注意: 対数リスク比を $\beta = \log \text{RR} = \log p - \log q$ と書き、 q の対数を $\alpha = \log q$ と書くと、 p, q の対数はそれぞれ次のように表される：

$$\log p = \alpha + \beta, \quad \log q = \alpha.$$

これは次と同値である：

$$p = \exp(\alpha + \beta), \quad q = \exp(\alpha).$$

これと上の違いは logistic と exp の違いになっている。

2 Wald版のP値と信頼区間

2.1 Wald版のオッズ比に関するP値と信頼区間の定義

検定したい仮説: 検定したい仮説は

オッズ比パラメータは $\text{OR} = \omega$ である。

であるとする。ここで $\omega > 0$ は具体的な正の実数である。

正規分布近似: 仮に a, b, c, d が2つの二項分布モデルに従う確率変数ならば、それらの **オッズ比**

$$\widehat{\text{OR}} = \frac{a/b}{c/d} = \frac{ad}{bc}$$

の対数 $\log \widehat{\text{OR}}$ (確率変数としての対数オッズ比)は、平均がモデルのパラメータの対数オッズ比

$$\log \text{OR} = \log \frac{p(1-q)}{(1-p)q}$$

で、分散が

$$\widehat{\text{SE}}_{\log \text{OR}}^2 = \frac{1}{a} + \frac{1}{b} + \frac{1}{c} + \frac{1}{d}$$

であるような正規分布に近似的に従うことを見せる。ゆえに、

$$\frac{\log \widehat{OR} - \log OR}{\widehat{SE}_{\log \widehat{OR}}} \sim \text{Normal}(0, 1), \text{ approximately.}$$

(「大数の法則と中心極限定理」のノート

(<https://nbviewer.org/github/genkuroki/Statistics/blob/master/2022/05%20Central%20limit%20theorem.ipynb>)で解説したデルタ法を使えば示せる。この点については後の節で詳しく説明する。)

記号法に関する注意: 以上においてハットがついている \widehat{OR} や $\widehat{SE}_{\log \widehat{OR}}$ は確率変数であることに注意せよ。ハットがついていない OR はモデルのパラメータで表された値である。

P値の構成法: 上の近似を使えば、「オッズ比パラメータは $OR = \omega$ である」という仮説のP値を次のように定めることができる：

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

ただし、 $\log \widehat{OR}$ と $\widehat{SE}_{\log \widehat{OR}}$ はデータの数値 a, b, c, d から上で示した式で計算された値であるとする。

$pvalue_{Wald}(a, b, c, d | OR = \omega)$ の定義式は

- 標準正規分布に従ってランダムに生成される値の絶対値がデータから計算された値 $(\log \widehat{OR} - \log \omega) / \widehat{SE}_{\log \widehat{OR}}$ の絶対値以上になる確率

を意味している。

対応する信頼区間: このP値の定義に対応するオッズ比パラメータ OR に関する信頼度 $1 - \alpha$ の信頼区間は次のようになる：

$$\text{confint}_{Wald}^{\log OR}(a, b, c, d | \alpha) = [\log \widehat{OR} - z_{\alpha/2} \widehat{SE}_{\log \widehat{OR}}, \log \widehat{OR} + z_{\alpha/2} \widehat{SE}_{\log \widehat{OR}}]$$

対応するオッズ比パラメータ OR の信頼区間は次のようになる：

$$\text{confint}_{Wald}^{OR}(a, b, c, d | \alpha) = [\exp(-z_{\alpha/2} \widehat{SE}_{\log \widehat{OR}}) \widehat{OR}, \exp(z_{\alpha/2} \widehat{SE}_{\log \widehat{OR}}) \widehat{OR}]$$

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

2.2 Wald版のオッズ比に関するP値と信頼区間の計算例

データが次の場合の仮説「オッズ比パラメータは $OR = 1$ である」のWald版のP値とオッズ比パラメータ OR の95%信頼区間を計算してみよう：

| | 商品を購入した | 購入しなかった | 合計 |
|-------|---------|---------|------|
| デザインA | 49 | 965 | 1014 |
| デザインB | 26 | 854 | 880 |
| 合計 | 75 | 1819 | 1894 |

結果は次のようになる：

- (仮説「オッズ比パラメータは $OR = 1$ である」のWald版のP値) ≈ 0.03847
- (オッズ比パラメータ OR のWald版の95%信頼区間) $\approx [1.0275, 2.7072]$

2.2.1 WolframAlphaによるWald版のオッズ比に関するP値と信頼区間の計算の仕方

$\widehat{SE}_{\log \widehat{OR}}$ の計算：

`sqrt(1/a+1/b+1/c+1/d) where a=49.0, b=965, c=26, d=854` → 実行(<https://www.wolframalpha.com/input/?i=sqrt%281%2Fa%2B1%2Fb%2B1%2Fc%2B1%2Fd%29+where+a%3D49.0%2C+b%3D965%2C+c%3D26%2C+d%3D854>) → 0.247137

$(\log \widehat{OR} - \log 1) / \widehat{SE}_{\log \widehat{OR}}$ の計算：

`log(a*d/(b*c))/0.247137 where a=49.0, b=965, c=26, d=854` → 実行([https://www.wolframalpha.com/input/?i=log%28a*d%2Fb*c%29%29%29%2F0.247137+where+a%3D49.0%2C+b%3D965%2C+c%3D26%2C+d%3D854](https://www.wolframalpha.com/input/?i=log%28a*d%2Fb*c%29%29%2F0.247137+where+a%3D49.0%2C+b%3D965%2C+c%3D26%2C+d%3D854)) → 2.06981

P値の計算：

$2(1 - \text{cdf}(\text{NormalDistribution}(0,1), 2.06981)) \rightarrow$ 実行 (<https://www.wolframalpha.com/input/?i=2%281+-+cdf%28NormalDistribution%280%2C1%29%2C+2.06981%29%29>) $\rightarrow 0.0384701$ (P値)

$\alpha = 0.05$ の場合の $z_{\alpha/2}$ の計算:

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

95%信頼区間の計算:

```
{exp(-0.247137z)*a*d/(b*c), exp(0.247137z)*a*d/(b*c)} where a=49.0, b=965, c=26, d=854,  
z=1.95996 → 実行 (https://www.wolframalpha.com/input/?i=%7Bexp%28-0.247137z%29\*a\*d%2F28b\*c%29%2C+exp%280.247137z%29\*a\*d%2F28b\*c%29%7D+where+a%3D49.0%2C+b%3D965%2C)  
→ {1.02752, 2.70717}
```

2.2.2 Julia言語によるWald版のオッズ比に関するP値と信頼区間の計算の仕方(1)

素朴に定義通りにコードを入力すれば計算できる.

In [20]:

```
1 a, b, c, d = 49, 965, 26, 854  
2 @show a, b, c, d  
3 ω = 1.0  
4 @show ω  
5 @show ORhat = a*d/(b*c)  
6 @show SEhat = √(1/a + 1/b + 1/c + 1/d)  
7 @show pvalue = 2 * ccdf(Normal(), abs(log(ORhat) - log(ω)) / SEhat)  
8 α = 0.05  
9 @show α  
10 @show z = quantile(Normal(), 1-α/2)  
11 @show confint = [exp(-z * SEhat) * ORhat, exp(z * SEhat) * ORhat]  
12 ;
```

```
(a, b, c, d) = (49, 965, 26, 854)  
ω = 1.0  
ORhat = (a * d) / (b * c) = 1.6678357911518533  
SEhat = √(1 / a + 1 / b + 1 / c + 1 / d) = 0.247137474584997  
pvalue = 2 * ccdf(Normal(), abs(log(ORhat) - log(ω)) / SEhat) = 0.03847043006362489  
α = 0.05  
z = quantile(Normal(), 1 - α / 2) = 1.9599639845400576  
confint = [exp(-z * SEhat) * ORhat, exp(z * SEhat) * ORhat] = [1.0275181208761617, 2.707179727272727  
83877]
```

2.2.3 Julia言語によるWald版のオッズ比に関するP値と信頼区間の計算の仕方(2)

このノートブックの最初の方で定義した函数を使って計算.

In [21]:

```
1 a, b, c, d = 49, 965, 26, 854  
2 @show a, b, c, d  
3 @show pvalue_or_wald(a, b, c, d; ω=1)  
4 @show confint_or_wald(a, b, c, d; α=0.05);
```

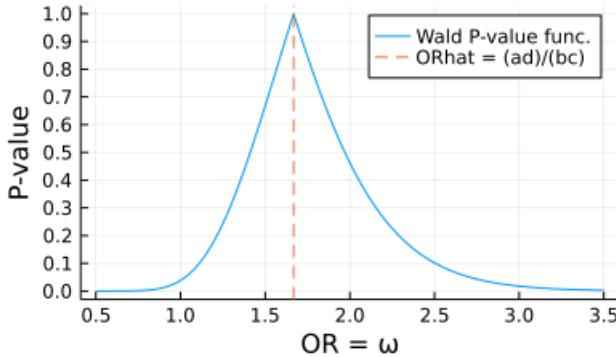
```
(a, b, c, d) = (49, 965, 26, 854)  
pvalue_or_wald(a, b, c, d; ω = 1) = 0.03847043006362489  
confint_or_wald(a, b, c, d; α = 0.05) = [1.0275181208761617, 2.7071797272783877]
```

2.2.4 Julia言語によるWald版のオッズ比に関するP値函数の視覚化

P値函数 $\omega \mapsto \text{pvalue}_{\text{Wald}}(a, b, c, d | \text{OR} = \omega)$ のグラフを描いてみよう.

```
In [22]: 1 a, b, c, d = 49, 965, 26, 854
2 plot(w → pvalue_or_wald(a,b,c,d; w), 0.5, 3.5; label="Wald P-value func.")
3 vline!([oddsratiohat(a,b,c,d)]; label="ORhat = (ad)/(bc)", ls=:dash)
4 plot!(; xguide="OR = ω", yguide="P-value")
5 plot!(; ytick=0:0.1:1)
6 title!("a, b, c, d = $a, $b, $c, $d")
```

Out[22]: a, b, c, d = 49, 965, 26, 854



このようにWald版P値函数は、 $\omega = \widehat{OR} = (ad)/(bc)$ で最大値の 1 になり、「とんがり帽子」型のグラフになる。

2.2.5 R言語でのWald版のオッズ比に関するP値と信頼区間の計算の仕方

定義通りにコードを入力すれば計算できる。

```
a = 49
b = 965
c = 26
d = 854
omega = 1
ORhat = a*d/(b*c)
SEhat = sqrt(1/a + 1/b + 1/c + 1/d)
p.value = 2*(1 - pnorm(abs(log(ORhat) - log(omega))/SEhat))
alpha = 0.05
z = qnorm(1 - alpha/2)
conf.int = c(exp(-z*SEhat)*ORhat, exp(z*SEhat)*ORhat)

cat("data: a b c d = ", a, b, c, d, "\n")
cat("null hypothesis: OR = ", omega, "\n")
cat("ORhat = ", ORhat, "\n")
cat("SEhat = ", SEhat, "\n")
cat("p.value = ", p.value, "\n")
cat("conf.level = ", 1 - alpha, "\n")
cat("conf.int = ", conf.int, "\n")
```

In [23]:

```

1 R"""
2 a = 49
3 b = 965
4 c = 26
5 d = 854
6 omega = 1
7 ORhat = a*d/(b*c)
8 SEhat = sqrt(1/a + 1/b + 1/c + 1/d)
9 p.value = 2*(1 - pnorm(abs(log(ORhat) - log(omega))/SEhat))
10 alpha = 0.05
11 z = qnorm(1 - alpha/2)
12 conf.int = c(exp(-z*SEhat)*ORhat, exp(z*SEhat)*ORhat)
13
14 cat("data: a b c d = ", a, b, c, d, "\n")
15 cat("null hypothesis: OR = ", omega, "\n")
16 cat("ORhat = ", ORhat, "\n")
17 cat("SEhat = ", SEhat, "\n")
18 cat("p.value = ", p.value, "\n")
19 cat("conf.level = ", 1 - alpha, "\n")
20 cat("conf.int = ", conf.int, "\n")
21 """

```

```

data: a b c d = 49 965 26 854
null hypothesis: OR = 1
ORhat = 1.667836
SEhat = 0.2471375
p.value = 0.03847043
conf.level = 0.95
conf.int = 1.027518 2.70718

```

\$95\%\$ 信頼区間はepiR パッケージを使えば

```

A = matrix(c(49, 965, 26, 854), 2, 2, byrow=T)
result = epiR::epi.2by2(A, digits=4, conf.level=0.95)

```

で計算できる。色々表示されるが、

| | |
|------------|-------------------------|
| Odds ratio | 1.6678 (1.0275, 2.7072) |
|------------|-------------------------|

の行の括弧の内側にWald版の信頼区間が表示されている。

In [24]:

```

1 R"""
2 A = matrix(c(49, 965, 26, 854), 2, 2, byrow=T)
3 result = epiR::epi.2by2(A, digits=4, conf.level=0.95)
4 """

```

Out[24]: RObject{VecSxp}

| | Outcome + | Outcome - | Total | Inc risk * | Odds |
|-----------|-----------|-----------|-------|------------|--------|
| Exposed + | 49 | 965 | 1014 | 4.83 | 0.0508 |
| Exposed - | 26 | 854 | 880 | 2.95 | 0.0304 |
| Total | 75 | 1819 | 1894 | 3.96 | 0.0412 |

Point estimates and 95% CIs:

```

-----
Inc risk ratio                      1.6356 (1.0254, 2.6088)
Odds ratio                          1.6678 (1.0275, 2.7072)
Attrib risk in the exposed *        1.8778 (0.1475, 3.6081)
Attrib fraction in the exposed (%) 38.8590 (2.4779, 61.6679)
Attrib risk in the population *    1.0053 (-0.4170, 2.4276)
Attrib fraction in the population (%) 25.3879 (-1.2434, 45.0140)
-----
```

Uncorrected chi2 test that OR = 1: chi2(1) = 4.368 Pr>chi2 = 0.037

Fisher exact test that OR = 1: Pr>chi2 = 0.044

Wald confidence limits

CI: confidence interval

* Outcomes per 100 population units

2.3 確率変数の対数オッズ比の分散の近似値の導出

この節では a, b, c, d は2つの二項分布モデルに従う確率変数であるとし、

$$\hat{p} = \frac{a}{a+b} = \frac{a}{m}, \quad \hat{q} = \frac{c}{c+d} = \frac{c}{n}$$

とおく。この対数オッズ比

$$\log \widehat{\text{OR}} = \log \frac{a/b}{c/d} = \log \frac{\hat{p}/(1-\hat{p})}{\hat{q}(1-\hat{q})} = \log \frac{\hat{p}}{1-\hat{p}} + \log \frac{\hat{q}}{1-\hat{q}}$$

の分散の近似値を求める。 $a \sim \text{Binomial}(m, p)$, $c \sim \text{Binomial}(n, q)$ のので、二項分布の中心極限定理より、

$$\hat{p} \sim \text{Normal}\left(p, \sqrt{\frac{p(1-p)}{m}}\right), \quad \hat{q} \sim \text{Normal}\left(q, \sqrt{\frac{q(1-q)}{n}}\right), \quad \text{approximately.}$$

a, c は独立なので \hat{p}, \hat{q} も独立になる。

$f(t) = \log(t/(1-t)) = \log(1/(1-t)-1)$ とおくと、 $f'(t) = ((1-t)/t)(1/(1-t)^2) = 1/(t(1-t))$ ので、

$$\log \frac{\hat{p}}{1-\hat{p}} = \frac{p}{1-p} + \frac{\hat{p}-p}{p(1-p)} + O((\hat{p}-p)^2).$$

これより、デルタ法を使って、次の近似が得られる：

$$\log \frac{\hat{p}}{1-\hat{p}} \sim \text{Normal}\left(\log \frac{p}{1-p}, \frac{1}{\sqrt{mp(1-p)}}\right), \quad \text{approximately.}$$

同様にして、

$$\log \frac{\hat{q}}{1-\hat{q}} \sim \text{Normal}\left(\log \frac{q}{1-q}, \frac{1}{\sqrt{nq(1-q)}}\right), \quad \text{approximately.}$$

デルタ法については

- 「[大数の法則と中心極限定理](#)」のノート
(<https://nbviewer.org/github/genkuroki/Statistics/blob/master/2022/05%20Central%20limit%20theorem.ipynb>)

に解説がある。しかし、上のように単に一次近似するだけの話に過ぎないので、自分でよく考えればそちらの解説を読む必要はない可能性が高い。

一般に X, Y が独立な確率変数で $X \sim \text{Normal}(\mu_X, \sigma_X)$, $Y \sim \text{Normal}(\mu_Y, \sigma_Y)$ のとき、

$X+Y \sim \text{Normal}\left(\mu_X+\mu_Y, \sqrt{\sigma_X^2+\sigma_Y^2}\right)$ となる。そして、

$$\begin{aligned} \left(\frac{1}{\sqrt{mp(1-p)}}\right)^2 + \left(\frac{1}{\sqrt{nq(1-q)}}\right)^2 &= \frac{1}{mp(1-p)} + \frac{1}{nq(1-q)} \\ &= \frac{1}{mp} + \frac{1}{m(1-p)} + \frac{1}{nq} + \frac{1}{n(1-q)} \end{aligned}$$

なので、

$$\text{SE}_{\log \widehat{\text{OR}}} = \sqrt{\frac{1}{mp} + \frac{1}{m(1-p)} + \frac{1}{nq} + \frac{1}{n(1-q)}}$$

において、 $\log \widehat{\text{OR}}$ の定義と

$$\log \text{OR} = \log \frac{p/(1-p)}{q/(1-q)} = \log \frac{p}{1-p} - \log \frac{q}{1-q}$$

を使うと、

$$\log \widehat{\text{OR}} \sim \text{Normal}\left(\log \text{OR}, \text{SE}_{\log \widehat{\text{OR}}}\right), \quad \text{approximately}$$

となることがわかる。

もしも $mp, m(1-p), nq, n(1-q)$ のそれぞれを a, b, c, d で近似できるならば、 SE の中の $mp, m(1-p), nq, n(1-q)$ のそれを a, b, c, d で置き換えて、

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

を作つて、

$$\log \widehat{\text{OR}} \sim \text{Normal}(\log \text{OR}, \widehat{\text{SE}}_{\log \text{OR}}), \quad \text{approximately}$$

という近似が得られる。この近似のもとで

$$\frac{\log \widehat{\text{OR}} - \log \text{OR}}{\widehat{\text{SE}}_{\log \text{OR}}} \sim \text{Normal}(0, 1), \quad \text{approximately.}$$

注意: 以上のような議論ではハット付きの $\widehat{\text{OR}}$ とハットが付いていない OR の区別が曖昧になると理解できなくなる。ハットが付いていない $\text{OR} = (p/(1-p))/(q/(1-q))$ はモデルのパラメータ p, q で書かれた量であり、ハットが付いている

$\widehat{\text{OR}} = (a/b)/(c/d)$ はモデルに従う確率変数 a, b, c, d で書かれた量になっていることに注意すれば区別がし易くなるだろう。実際にその方法で $\text{SE}_{\log \text{OR}}$ と $\widehat{\text{SE}}_{\log \text{OR}}$ の区別を容易にできることを確認してみよ。

注意: ハット付きの記号はデータの数値で表された同様の量にもそのまま流用されるので、文脈によって、確率変数になっているかと、データの数値から計算された数値になっているかを区別しなければいけない。この問題はモデル内の確率変数とデータの数値から計算された数値を記号的にも区別できるようにすれば解決するのだが、実際にやってみればわかるように、記号が煩雑過ぎるという点でつらくなる場合が多数出て来る。しかし、初学者にとって「文脈を利用して正しく解釈すること」は難しい場合があるので、実際に難しいと感じる人は、文脈に頼らずに記号を見ただけで区別できるように書かれたノートを **自分で** 作成してみるとよいだろう。(おそらくそれを本当に実行した人は結果的に文脈を利用して正しく解釈することができるようになるだろう。)

2.4 問題: 対数リスク比の場合の正規分布近似

a, b, c, d は2つの二項分布モデルに従う確率変数であるとし、 $\hat{p} = a/m, \hat{q} = c/n$ とおき、それらに関する確率変数としての対数リスク比を次のように定める：

$$\log \widehat{\text{RR}} = \log \frac{\hat{p}}{\hat{q}} = \log \hat{p} - \log \hat{q}.$$

このとき、

$$\log \text{RR} = \log \frac{p}{q}, \quad \widehat{\text{SE}}_{\log \text{RR}} = \sqrt{\frac{1}{a} - \frac{1}{a+b} + \frac{1}{c} - \frac{1}{c+d}}$$

とおくと、二項分布の中心極限定理を使って、次の近似が得られることを示せ：

$$\log \widehat{\text{RR}} \sim \text{Normal}(\log \text{RR}, \widehat{\text{SE}}_{\log \text{RR}}), \quad \text{approximately.}$$

解答例: デルタ法を利用する。 $f(t) = \log t$ とおくと、 $f'(t) = 1/t$ なので、

$$\log \hat{p} = \log p + \frac{\hat{p} - p}{p} + O((\hat{p} - p)^2).$$

これと二項分布の中心極限定理より近似的に $\hat{p} \sim \text{Normal}(p, \sqrt{p(1-p)/m})$ となるので次の近似を得る：

$$\log \hat{p} \sim \text{Normal}\left(\log p, \sqrt{\frac{1-p}{mp}}\right), \quad \text{approximately.}$$

同様にして、

$$\log \hat{q} \sim \text{Normal}\left(\log q, \sqrt{\frac{1-q}{nq}}\right), \quad \text{approximately.}$$

ゆえに、 $\text{SE}_{\log \text{RR}}$ を

$$\begin{aligned} \text{SE}_{\log \text{RR}} &= \sqrt{\frac{1-p}{mp} + \frac{1-q}{nq}} \\ &= \sqrt{\frac{1}{mp} - \frac{1}{m} + \frac{1}{nq} - \frac{1}{n}} \end{aligned}$$

と定めると、

$$\log \widehat{\text{RR}} \sim \text{Normal}(\log \text{RR}, \text{SE}_{\log \text{RR}}), \quad \text{approximately.}$$

ゆえにもしも、 mp, nq のそれぞれを a, c で近似できるならば、 $a + b = m, c + d = n$ も使って

$$\log \widehat{RR} \sim \text{Normal}\left(\log RR, \widehat{SE}_{\log \widehat{RR}}\right), \quad \text{approximately.}$$

が得られる。

解答終

2.5 問題: Wald版のリスク比に関するP値と信頼区間の定義

上の問題の結果を用いてWald版のリスク比に関するP値と信頼区間を適切に定義せよ。

注意: 数学的にややこしい事柄を理解したい場合には「定義だけは最初に覚えなければいけない」のように考えると失敗することが多い。適切な定義を自分で作れるような考え方をした方がよい。この問題を適切に解けた人は今度は自力でP値と信頼区間を適切に定義して自由自在に使えるようになるだろう。

解答例: 現実世界における調査によって 2×2 の分割表のデータの数値 a, b, c, d が得られているとする。そのデータの数値の対数リスク比を次のように定める：

$$\log \widehat{RR} = \frac{a/(a+b)}{c/(c+d)}.$$

仮説「リスク比パラメータは $RR = \rho$ である」の下での2つの二項分布モデル内でランダムに生成されたデータの対数リスク比が現実世界で得たデータの対数リスク比以上に $RR = \rho$ から離れる確率の近似値として、「リスク比パラメータは $RR = \rho$ である」という仮説のP値を定める：

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

ここで1つ前の問題の結果と記号

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

を使った。このP値に定義に対応する対数リスク比パラメータ $\log RR$ の信頼度 $1 - \alpha$ の信頼区間は

$$\text{confint}_{\text{Wald}}^{\log RR}(a, b, c, d | \alpha) = [\log \widehat{RR} - z_{\alpha/2} \widehat{SE}_{\log \widehat{RR}}, \log \widehat{RR} + z_{\alpha/2} \widehat{SE}_{\log \widehat{RR}}]$$

になり、リスク比パラメータ RR の信頼度 $1 - \alpha$ の信頼区間は

$$\text{confint}_{\text{Wald}}^{RR}(a, b, c, d | \alpha) = [\exp(-z_{\alpha/2} \widehat{SE}_{\log \widehat{RR}}) \widehat{RR}, \exp(z_{\alpha/2} \widehat{SE}_{\log \widehat{RR}}) \widehat{RR}]$$

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

2.6 問題: Wald版のリスク比に関するP値と信頼区間の計算例

データが次の場合のWald版の仮説「リスク比パラメータは $RR = 1$ である」のP値とリスク比パラメータ RR の 95% 信頼区間を計算してみよ：

| | 商品を購入した | 購入しなかった | 合計 |
|-------|---------|---------|------|
| デザインA | 49 | 965 | 1014 |
| デザインB | 26 | 854 | 880 |
| 合計 | 75 | 1819 | 1894 |

2.6.1 Julia言語によるWald版のリスク比に関するP値と信頼区間の計算の仕方

In [25]:

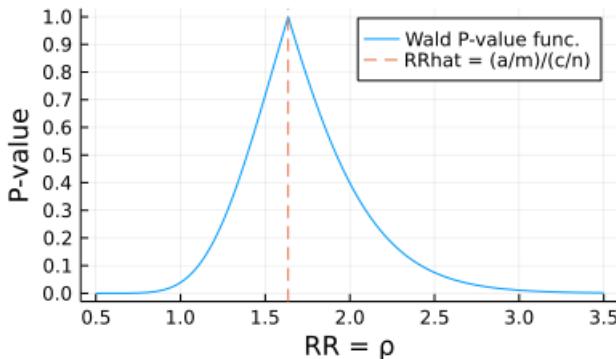
```
1 a, b, c, d = 49, 965, 26, 854
2 @show riskratiohat(a, b, c, d)
3 @show pvalue_rr_wald(a, b, c, d; ρ=1)
4 @show confint_rr_wald(a, b, c, d; α = 0.05);
```

```
riskratiohat(a, b, c, d) = 1.635563647397967
pvalue_rr_wald(a, b, c, d; ρ = 1) = 0.038895117565198624
confint_rr_wald(a, b, c, d; α = 0.05) = [1.025408984412048, 2.6087819449169145]
```

2.6.2 Julia言語によるWald版のリスク比に関するP値函数の視覚化

```
In [26]: 1 a, b, c, d = 49, 965, 26, 854
2 plot(p → pvalue_rr_wald(a,b,c,d; p), 0.5, 3.5; label="Wald P-value func.")
3 vline!([riskratiohat(a,b,c,d)]; label="RRhat = (a/m)/(c/n)", ls=:dash)
4 plot!(; xguide="RR = ρ", yguide="P-value")
5 plot!(; ytick=0:0.1:1)
6 title!("a, m, c, n = $a, $(a+b), $c, $(c+d)")
```

Out[26]: a, m, c, n = 49, 1014, 26, 880



2.6.3 R言語によるWald版のリスク比に関する信頼区間の計算の仕方

epiRパッケージをRにインストールしておけば、

```
A = matrix(c(49, 965, 26, 854), 2, 2, byrow=T)
result = epiR::epi.2by2(A, digits=4, conf.level=0.95)
```

でWald版のリスク比に関する信頼区間を計算できる。色々表示されているが、

| | |
|----------------|-------------------------|
| Inc risk ratio | 1.6356 (1.0254, 2.6088) |
|----------------|-------------------------|

の行の括弧の中に信頼区間が表示されている。

```
In [27]: 1 R"""
2 A = matrix(c(49, 965, 26, 854), 2, 2, byrow=T)
3 result = epiR::epi.2by2(A, digits=4, conf.level=0.95)
4 """
```

```
Out[27]: Robject{VecSxp}
      Outcome +   Outcome -   Total   Inc risk *   Odds
Exposed +        49         965     1014       4.83    0.0508
Exposed -        26         854     880        2.95    0.0304
Total          75        1819    1894       3.96    0.0412

Point estimates and 95% CIs:
-----
Inc risk ratio                  1.6356 (1.0254, 2.6088)
Odds ratio                      1.6678 (1.0275, 2.7072)
Attrib risk in the exposed *    1.8778 (0.1475, 3.6081)
Attrib fraction in the exposed (%) 38.8590 (2.4779, 61.6679)
Attrib risk in the population *  1.0053 (-0.4170, 2.4276)
Attrib fraction in the population (%) 25.3879 (-1.2434, 45.0140)

Uncorrected chi2 test that OR = 1: chi2(1) = 4.368 Pr>chi2 = 0.037
Fisher exact test that OR = 1: Pr>chi2 = 0.044
Wald confidence limits
CI: confidence interval
* Outcomes per 100 population units
```

2.6.4 Wald版のリスク比に関するP値と信頼区間の計算例の問題の解答例

- (仮説「リスク比パラメータは RR = 1 である」のWald版のP値) ≈ 0.03890
- (リスク比パラメータ RR のWald版の 95% 信頼区間) ≈ [1.0254, 2.6088]

注意: この場合にはリスク比がオッズ比で近似される場合になっている。

In [28]:

```

1 a, b, c, d = 49, 965, 26, 854
2 @show riskratiohat(a, b, c, d)
3 @show oddsratiohat(a, b, c, d)
4 @show pvalue_rr_wald(a, b, c, d; ρ=1)
5 @show pvalue_or_wald(a, b, c, d; ω=1)
6 @show confint_rr_wald(a, b, c, d; α = 0.05)
7 @show confint_or_wald(a, b, c, d; α = 0.05);

```

```

riskratiohat(a, b, c, d) = 1.635563647397967
oddsratiohat(a, b, c, d) = 1.6678357911518533
pvalue_rr_wald(a, b, c, d; ρ = 1) = 0.038895117565198624
pvalue_or_wald(a, b, c, d; ω = 1) = 0.03847043006362489
confint_rr_wald(a, b, c, d; α = 0.05) = [1.025408984412048, 2.6087819449169145]
confint_or_wald(a, b, c, d; α = 0.05) = [1.0275181208761617, 2.7071797272783877]

```

2.7 問題: 比率の差の場合の正規分布近似

a, b, c, d は2つの二項分布モデルに従う確率変数であるとし, $\hat{p} = a/m, \hat{q} = c/n$ とおき, $\widehat{\text{SE}}_{\hat{p}-\hat{q}}$ を

$$\widehat{\text{SE}}_{\hat{p}-\hat{q}} = \sqrt{\frac{\hat{p}(1-\hat{p})}{m} + \frac{\hat{q}(1-\hat{q})}{n}}$$

と定める. このとき, 二項分布の中心極限定理から, 次の近似が得られることを示せ:

$$\hat{p} - \hat{q} \sim \text{Normal}\left(p - q, \widehat{\text{SE}}_{\hat{p}-\hat{q}}\right) \quad \text{approximately.}$$

解答例: 二項分布の中心極限定理によって,

$$\hat{p} \sim \text{Normal}\left(p, \sqrt{\frac{p(1-p)}{m}}\right), \quad \hat{q} \sim \text{Normal}\left(q, \sqrt{\frac{q(1-q)}{n}}\right), \quad \text{approximately.}$$

\hat{p}, \hat{q} が独立であることと, 正規分布に従う独立な確率変数達の和が分散がそれらの分散の和であるような正規分布に従うことより,

$$\text{SE}_{\hat{p}-\hat{q}} = \sqrt{\frac{p(1-p)}{m} + \frac{q(1-q)}{n}}$$

とおくと,

$$\hat{p} - \hat{q} \sim \text{Normal}\left(q, \text{SE}_{\hat{p}-\hat{q}}\right), \quad \text{approximately.}$$

ゆえに, p, q が \hat{p}, \hat{q} で近似できているならば,

$$\hat{p} - \hat{q} \sim \text{Normal}\left(q, \widehat{\text{SE}}_{\hat{p}-\hat{q}}\right), \quad \text{approximately.}$$

解答終

2.8 問題: Wald版の比率の差に関するP値と信頼区間の定義

上の問題の結果を利用して, 仮説「 $p - q = \Delta$ 」のP値とパラメータ $p - q$ の信頼区間を適切に定義せよ.

解答例: 仮説「 $p - q = \Delta$ 」のP値を

$$\widehat{\text{SE}}_{\hat{p}-\hat{q}} = \sqrt{\frac{\hat{p}(1-\hat{p})}{m} + \frac{\hat{q}(1-\hat{q})}{n}}$$

を用いて,

$$\text{pvalue}_{\text{Wald}}(a, b, c, d | p - q = \Delta) = 2 \left(1 - \text{cdf}\left(\text{Normal}(0, 1), \frac{|\hat{p} - \hat{q} - \Delta|}{\widehat{\text{SE}}_{\hat{p}-\hat{q}}}\right) \right)$$

と定める. 対応する信頼度 $1 - \alpha$ の信頼区間は次のようになる.

$$\text{confint}_{\text{Wald}}^{p-q}(a, b, c, d | \alpha) = \left[\hat{p} - \hat{q} - z_{\alpha/2} \widehat{\text{SE}}_{\hat{p}-\hat{q}}, \hat{p} - \hat{q} + z_{\alpha/2} \widehat{\text{SE}}_{\hat{p}-\hat{q}} \right].$$

ただし $z_{\alpha/2} = \text{quantile}(\text{Normal}(0, 1), 1 - \alpha/2)$ (標準正規分布において $z_{\alpha/2}$ 以上になる確率は $\alpha/2$)である.

解答終

2.9 問題: Wald版の比率の差に関するP値と信頼区間の計算例

データが次の場合の仮説「 $p - q = 0$ 」のP値と $p - q$ のWald版の 95% 信頼区間を計算してみよ:

| | 商品を購入した | 購入しなかった | 合計 |
|-------|---------|---------|------|
| デザインA | 49 | 965 | 1014 |
| デザインB | 26 | 854 | 880 |
| 合計 | 75 | 1819 | 1894 |

2.9.1 Julia言語によるWald版の比率の差に関するP値と信頼区間の計算の仕方

以下のようにほぼ定義通りにコードを書けば計算できる.

```
riskdiffhat(a, b, c, d) = a/(a+b) - c/(c+d)

function stderr_riskdiffhat(a, b, c, d)
    m, n = a+b, c+d
    p̂, q̂ = a/m, c/n
    √(p̂*(1-p̂)/m + q̂*(1-q̂)/n)
end

function pvalue_rd_wald(a, b, c, d; Δ=0)
    RDhat = riskdiffhat(a, b, c, d)
    SEhat_riskdiffhat = stderr_riskdiffhat(a, b, c, d)
    2ccdf(Normal(0, 1), safediv(abs(RDhat - Δ), SEhat_riskdiffhat))
end

function confint_rd_wald(a, b, c, d; α=0.05)
    z = quantile(Normal(), 1-α/2)
    RDhat = riskdiffhat(a, b, c, d)
    SEhat_riskdiffhat = stderr_riskdiffhat(a, b, c, d)
    [RDhat - z*SEhat_riskdiffhat, RDhat + z*SEhat_riskdiffhat]
end
```

In [29]:

```
1 a, b, c, d = 49, 965, 26, 854
2 @show pvalue_rd_wald(a, b, c, d; Δ=0)
3 @show confint_rd_wald(a, b, c, d; α=0.05);
```

```
pvalue_rd_wald(a, b, c, d; Δ = 0) = 0.033414467036392496
confint_rd_wald(a, b, c, d; α = 0.05) = [0.0014752114457742202, 0.03608082226410564]
```

2.9.2 Wald版の比率の差に関するP値と信頼区間の計算の解答例

仮説「 $p - q = 0$ 」のP値: 0.03341 $p - q$ の 95% 信頼区間: [0.001475, 0.03608]

3 必修問題: Wald版のオッズ比とリスク比のP値と信頼区間

| | 症状悪化 | 悪化せず | 合計 |
|-------|------|------|------|
| 薬Iを処方 | 100 | 579 | 679 |
| プラセボ | 111 | 568 | 679 |
| 合計 | 211 | 1147 | 1358 |

この表は次の論文

- Gilmar Reis, M.D., Ph.D. et al., Effect of Early Treatment with Ivermectin among Patients with Covid-19, N Engl J Med 2022; 386:1721-1731. [[link](https://www.nejm.org/doi/full/10.1056/NEJMoa2115869) (<https://www.google.com/search?q=intention-to-treat>)]

の Table 2 から [Intention-to-treat population](https://www.google.com/search?q=intention-to-treat) (<https://www.google.com/search?q=intention-to-treat>)の部分だけを抜き出したものである。「症状悪化」の正確な意味などに関する詳しい説明は論文を参照せよ。

上のデータについて、Wald版の以下のP値と信頼区間を計算せよ:

- (1) 仮説「オッズ比パラメータは $OR = 1.0$ である」のP値
- (2) 仮説「オッズ比パラメータは $OR = 0.5$ である」のP値
- (3) オッズ比パラメータの 95% 信頼区間

可能ならば、前節の問題の解答例を見ながら、以下も計算してみよ:

- (4) 仮説「リスク比パラメータは $RR = 1.0$ である」のP値
- (5) 仮説「リスク比パラメータは $RR = 0.5$ である」のP値
- (6) リスク比パラメータの 95% 信頼区間

3.1 WolframAlphaによる必修問題の解答例

(1)

$\widehat{SE}_{\log OR}$ の計算:

```
sqrt(1/a+1/b+1/c+1/d) where a=100.0, b=579, c=111, d=568 → 実行(https://www.wolframalpha.com/input?i=sqrt%281%2Fa%2B1%2Fb%2B1%2Fc%2B1%2Fd%29+where+a%3D100.0%2C+b%3D579%2C+c%3D111%2C+d%3D568) → 0.149989
```

$(\log \widehat{OR} - \log 1)/\widehat{SE}_{\log OR}$ の計算:

```
log(a*d/(b*c))/0.149989 where a=100.0, b=579, c=111, d=568 → 実行(https://www.wolframalpha.com/input?i=log%28a\*d%2Fb\*c%29%29%2F0.149989+where+a%3D100.0%2C+b%3D579%2C+c%3D111%2C+d%3D568) → -0.823668
```

P値の計算:

```
2(1 - cdf(NormalDistribution(0,1), 0.823668)) → 実行(https://www.wolframalpha.com/input?i=2%281+-cdf%28NormalDistribution%280%2C1%29%2C+0.823668%29%29%29) → 0.410128 ((1)のP値)
```

(2)

$(\log \widehat{OR} - \log 0.5)/\widehat{SE}_{\log OR}$ の計算:

```
(log(a*d/(b*c)) - log(0.5))/0.149989 where a=100.0, b=579, c=111, d=568 → 実行(https://www.wolframalpha.com/input?i=log%28a\*d%2Fb\*c%29%29%2F0.149989+where+a%3D100.0%2C+b%3D579%2C+c%3D111%2C+d%3D568) → 3.79765
```

P値の計算:

```
2(1 - cdf(NormalDistribution(0,1), 3.79765)) → 実行(https://www.wolframalpha.com/input?i=2%281+-cdf%28NormalDistribution%280%2C1%29%2C+3.79765%29%29%29) → 0.000146074 ((2)のP値)
```

(3)

$\alpha = 0.05$ のときの $z_{\alpha/2}$ の計算:

```
quantile(NormalDistribution(0,1), 0.975) → 実行(https://www.wolframalpha.com/input?i=quantile%28NormalDistribution%280%2C1%29%2C+0.975%29) → 1.95996
```

オッズ比パラメータの信頼区間の計算:

```
{exp(-0.149989z)*a*d/(b*c), exp(0.149989z)*a*d/(b*c)} where a=100.0, b=579, c=111, d=568, z=1.95996 → 実行(https://www.wolframalpha.com/input?i=exp%28-0.149989z%29\*a\*d%2Fb\*c%29%2C+exp%280.149989z%29\*a\*d%2Fb\*c%29%7D+where+a%3D100.0%2C+b%3D579%2C+d%3D568) → {0.658683, 1.18582} ((3)の信頼区間)
```

(4)

$\widehat{SE}_{\log RR}$ の計算:

```
sqrt(1/a-1/(a+b)+1/c-1/(c+d)) where a=100.0, b=579, c=111, d=568 → 実行(https://www.wolframalpha.com/input?i=sqrt%281%2Fa-1%2Fb%2B1%2Fc-1%2Fc+d%29) → 0.126742
```

$(\log \widehat{RR} - \log 1) / \widehat{SE}_{\log \widehat{RR}}$ の計算:

```
log(a*(c+d)/((a+b)*c))/0.126742 where a=100.0, b=579, c=111, d=568 → 実行  
(https://www.wolframalpha.com/input?i=log%28a%28c%2Bd%29%2F%28a%2Bb%29\*c%29%29%2F0.126742+where+a%3D100.0%2C+b%3D579%2C+c%3D111%29)  
→ -0.823405
```

P値の計算:

```
2(1 - cdf(NormalDistribution(0,1), 0.823405)) → 実行 (https://www.wolframalpha.com/input?i=2%281+-cdf%28NormalDistribution%280%2C1%29%2C+0.823405%29%29) → 0.410278 ((4)のP値)
```

(5)

$(\log \widehat{RR} - \log 0.5) / \widehat{SE}_{\log \widehat{RR}}$ の計算:

```
(log(a*(c+d)/((a+b)*c)) - log(0.5))/0.126742 where a=100.0, b=579, c=111, d=568 → 実行  
(https://www.wolframalpha.com/input?i=log%28a%28c%2Bd%29%2F%28a%2Bb%29\*c%29%29%2F0.126742+where+a%3D100.0%2C+b%3D579%2C+c%3D111%2C+d%3D568) → 4.64556
```

P値の計算:

```
2(1 - cdf(NormalDistribution(0,1), 4.64556)) → 実行 (https://www.wolframalpha.com/input?i=2%281+-cdf%28NormalDistribution%280%2C1%29%2C+4.64556%29%29) → 3.39155×10^-6 = 3.39155e-6 ((5)のP値)
```

このP値は極めて小さい。2つの二項分布モデルに関するリスク比が 0.5 であるという仮説はデータとの整合性がほとんどないと考えられる。(この場合にリスク比が 0.5 であることは症状が悪化することが半分に減るという意味になるが、モデル内におけるその想定はP値が小さ過ぎるという理由で否定される。)

(6)

$\alpha = 0.05$ のときの $z_{\alpha/2}$ の計算:

```
quantile(NormalDistribution(0,1), 0.975) → 実行 (https://www.wolframalpha.com/input?i=quantile%28NormalDistribution%280%2C1%29%2C+0.975%29) → 1.95996
```

リスク比パラメータの信頼区間の計算:

```
{exp(-0.126742z)*a*(c+d)/((a+b)*c), exp(0.126742z)*a*(c+d)/((a+b)*c)} where a=100.0, b=579, c=111, d=568, z=1.95996 → 実行 (https://www.wolframalpha.com/input?i=%7Bexp%28-0.126742z%29\*a%28c%2Bd%29%2F%28a%2Bb%29\*c%29%2C+exp%280.126742z%29\*a%28c%2Bd%29%2F%28a%2Bb%29\*c%29%29%2C%7D) → {0.702739, 1.15494} ((6)の信頼区間)
```

3.2 Julia言語による必修問題の解答例

Julia言語のシンプルな使い方については

- [Julia言語を使った統計学の勉強の仕方 \(<https://nbviewer.org/github/genkuroki/Statistics/blob/master/2022/07-2%20How%20to%20use%20Julia%20language%20for%20learning%20statistics.ipynb>\)](https://nbviewer.org/github/genkuroki/Statistics/blob/master/2022/07-2%20How%20to%20use%20Julia%20language%20for%20learning%20statistics.ipynb)

を参照せよ。

左から \$0\$ をかけると常に \$0\$ になり、左から \$\text{lop}\{\text{Inf}\}\$ をかけると常に \$\text{lop}\{\text{Inf}\}\$ になる掛け算と分子が \$0\$ ならば何で割っても \$0\$ になり、分母が無限大なら分子が何であっても \$0\$ になる割り算の定義:

```
safemul(x, y) = x == 0 ? x : isnan(x) ? typeof(x)(Inf) : x*y  
safediv(x, y) = x == 0 ? x : isnan(y) ? zero(y) : x/y
```

これを使うと浮動小数点数に関する計算が少し安全になる場合がある。

$0/0$ は非数 NaN になる。計算の途中で NaN が生じることを防ぎたい。 $1/0$ は無限大 Inf になる。 Inf は悪さをせずに済むことがある。ただし、 $\text{Inf} - \text{Inf}$ や Inf/Inf も NaN になるので注意が必要である。

In [30]: 1 @show 0/0 1/0 Inf-Inf Inf/Inf;

```
0 / 0 = NaN  
1 / 0 = Inf  
Inf - Inf = NaN  
Inf / Inf = NaN
```

オッズ比に関するWald版のP値と信頼区間を計算する函数

ほぼ定義通りにコードが記述されている。

```
oddsratiohat(a, b, c, d) = safediv(a*d, b*c)
stderr_logoddsratiohat(a, b, c, d) = sqrt(1/a + 1/b + 1/c + 1/d)

function pvalue_or_wald(a, b, c, d; ω=1)
    logORhat = log(oddsratiohat(a, b, c, d))
    SEhat_logORhat = stderr_logoddsratiohat(a, b, c, d)
    2ccdf(Normal(0, 1), safediv(abs(logORhat - log(ω)), SEhat_logORhat))
end

function confint_or_wald(a, b, c, d; α=0.05)
    z = quantile(Normal(), 1-α/2)
    ORhat = oddsratiohat(a, b, c, d)
    SEhat_logORhat = stderr_logoddsratiohat(a, b, c, d)
    [safemul(exp(-z*SEhat_logORhat), ORhat), safemul(exp(z*SEhat_logORhat), ORhat)]
end
```

リスク比に関するWald版のP値と信頼区間を計算する函数

ほぼ定義通りにコードが記述されている。

```
riskratiohat(a, b, c, d) = safediv(a*(c+d), (a+b)*c)
stderr_logriskratiohat(a, b, c, d) = sqrt(1/a - 1/(a+b) + 1/c - 1/(c+d))

function pvalue_rr_wald(a, b, c, d; ρ=1)
    logRRhat = log(riskratiohat(a, b, c, d))
    SEhat_logRRhat = stderr_logriskratiohat(a, b, c, d)
    2ccdf(Normal(0, 1), safediv(abs(logRRhat - log(ρ)), SEhat_logRRhat))
end

function confint_rr_wald(a, b, c, d; α=0.05)
    (a+b==0 || c+d==0) && return [0, Inf]
    z = quantile(Normal(), 1-α/2)
    RRhat = riskratiohat(a, b, c, d)
    SEhat_logRRhat = stderr_logriskratiohat(a, b, c, d)
    [safemul(exp(-z*SEhat_logRRhat), RRhat), safemul(exp(z*SEhat_logRRhat), RRhat)]
end
```

In [31]: 1 a, b, c, d = 100, 579, 111, 568

Out[31]: (100, 579, 111, 568)

In [32]: 1 # (1)
2 @show pvalue_or_wald(a, b, c, d; ω=1.0);

pvalue_or_wald(a, b, c, d; ω = 1.0) = 0.4101283497257791

In [33]: 1 # (2)
2 @show pvalue_or_wald(a, b, c, d; ω=0.5);

pvalue_or_wald(a, b, c, d; ω = 0.5) = 0.0001460722899582654

In [34]: 1 # (3)
2 @show confint_or_wald(a, b, c, d; α=0.05);

confint_or_wald(a, b, c, d; α = 0.05) = [0.6586822480003675, 1.1858168777394855]

In [35]: 1 # (4)
2 @show pvalue_rr_wald(a, b, c, d; ρ=1.0);

pvalue_rr_wald(a, b, c, d; ρ = 1.0) = 0.4102771459204757

```
In [36]: 1 # (4)
2 @show pvalue_rr_wald(a, b, c, d; ρ=0.5);
pvalue_rr_wald(a, b, c, d; ρ = 0.5) = 3.391523737996353e-6

In [37]: 1 # (6)
2 @show confint_rr_wald(a, b, c, d; α=0.05);
confint_rr_wald(a, b, c, d; α = 0.05) = [0.7027391487578177, 1.1549412533494148]
```

3.3 必修問題解答例

- (1) 0.4101
- (2) 0.0001461
- (3) [0.6587, 1.1858]
- (4) 0.4103
- (5) 3.392e-6
- (6) [0.7027, 1.1549]

4 Pearsonの χ^2 検定版のP値と信頼区間

4.1 Pearsonの χ^2 検定版のオッズ比に関するP値と信頼区間の定義

天下り的になってしまい非常に申し訳ないのだが、オッズ比に関するP値のPearsonの χ^2 検定版を以下のように定義する。特にP値の構成法の(1)の段階がひどく天下り的である(後でWilks' theoremへの帰着の仕方を説明する)。しかし、それ以外の部分については計算の筋道を詳しく書いておくので、自分の手で計算して確認して欲しい。

そして、さらに、それだけだとどういうことなのか分かりにくいので、後の方で示す具体的な数値の計算例と以下で説明する各ステップの対応を繰り返し確認して欲しい。

検定したい仮説: 検定したい仮説は

オッズ比パラメータは $OR = \omega$ である。

であるとする。ここで $\omega > 0$ は具体的な正の実数である。

仮説が $OR = 1$ である場合に、以下で定義するP値はちょうど **独立性に関するPearsonの χ^2 検定** で使うP値になっている。

P値の構成法: オッズ比パラメータの値に関する仮説のP値のPearsonの χ^2 検定版の構成法は以下の通り。

(1) 分割表のデータの数値 a, b, c, d に対して、次を満たす $\delta = \delta(a, b, c, d|\omega)$ を求める:

$$\frac{(a - \delta)(d - \delta)}{(b + \delta)(c + \delta)} = \omega, \quad -\min(b, c) \leq \delta \leq \min(a, d).$$

この左側の等式は

$$A = 1 - \omega, \quad B = a + d + \omega(b + c), \quad C = ad - \omega bc$$

とおくと、次の二次方程式に書き直される:

$$A\delta^2 - B\delta + C = 0.$$

この解で $-\min(b, c) \leq \delta \leq \min(a, d)$ を満たす側が欲しい $\delta = \delta(a, b, c, d|\omega)$ である。それは次のように表される:

$$\delta = \delta(a, b, c, d|\omega) = \frac{2C}{B + \sqrt{B^2 - 4AC}}.$$

この表示で浮動小数点数の計算をした方が安全でかつ誤差も小さくなり易い。

特に $\omega = 1$ の場合(独立性検定の場合)には、 δ を決める方程式は $-(a + b + c + d)\delta + ad - bc = 0$ になるので、

$$\omega = 1 \implies \delta = \delta(a, b, c, d|1) = \frac{ad - bc}{a + b + c + d} = \frac{ad - bc}{N}.$$

(2) $\tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}$ を次のように定める:

$$\tilde{a} = a - \delta, \quad \tilde{b} = b + \delta, \quad \tilde{c} = c + \delta, \quad \tilde{d} = d - \delta.$$

これが実は仮説 $OR = 1$ 下での期待値の最尤推定値になっていることを後の方の節で説明する。

特に $\omega = 1$ の場合(独立性検定の場合)には次のようになっている:

$$\omega = 1 \implies \begin{cases} \tilde{a} = \frac{(a+b)(a+c)}{N} = \frac{ms}{N}, & \tilde{b} = \frac{(a+b)(b+d)}{N} = \frac{mr}{N}, \\ \tilde{c} = \frac{(c+d)(a+c)}{N} = \frac{ns}{N}, & \tilde{d} = \frac{(c+d)(b+d)}{N} = \frac{nr}{N}. \end{cases}$$

(3) データの数値に対応するPearsonの χ^2 統計量 $\chi^2 = \chi^2(a, b, c, d|\delta)$ の値を次のように定める:

$$\begin{aligned} \chi^2 &= \chi^2(a, b, c, d|\delta) = \delta^2 \left(\frac{1}{a-\delta} + \frac{1}{b+\delta} + \frac{1}{c+\delta} + \frac{1}{d-\delta} \right) \\ &= \frac{(a-\tilde{a})^2}{\tilde{a}} + \frac{(b-\tilde{b})^2}{\tilde{b}} + \frac{(c-\tilde{c})^2}{\tilde{c}} + \frac{(d-\tilde{d})^2}{\tilde{d}}. \end{aligned}$$

ここで, $\delta = \delta(a, b, c, d|\omega)$ は a, b, c, d, ω のみを使って計算できる数値なので, この $\chi^2 = \text{\textbackslash chi}^2(a,b,c,d|\text{\textbackslash delta}\text{\textbackslash a,b,c,d,\omega})$ のみを使って計算できる数値になっている。

上の χ^2 に関する後者の表示はPearsonの χ^2 統計量の一般的な表示

$$\chi^2 = \sum \frac{((\text{観測値}) - (\text{帰無仮説下のモデル内での期待値}))^2}{(\text{帰無仮説下のモデル内での期待値})}$$

の特別な場合についていることに注意せよ。ただし, 「帰無仮説下のモデル内での期待値」は最尤推定値として求めたものになっている。

特に $\omega = 1$ の場合(独立性検定の場合)には次のようになっている:

$$\chi^2 = \frac{N(ad - bc)^2}{(a+b)(c+d)(a+c)(b+d)} = \frac{N(ad - bc)^2}{mnrs}$$

これは独立性に関するPearsonの χ^2 統計量の公式として有名である。

(4) 仮説 $OR = \omega$ の下での統計モデル内で同様に定義されたPearsonの χ^2 統計量の値がデータから計算した χ^2 の値以上になる確率の近似値としてP値を定義したい。(一般にP値は、統計モデル内でデータの数値以上に極端な場合が生じる確率またはその近似値として定義されるのであった。)

仮に a, b, c, d が現実から得たデータの数値ではなく、仮説 $OR = \omega$ の下での統計モデル内でランダムに生成された仮想的なデータであるとき、(3)のように定義された χ^2 は a, b, c, d が十分に大きなとき(実はそう大きくなくてもよい), 近似的に自由度 1 の χ^2 分布に従うことを示せる。

このことを使って、仮説 $OR = \omega$ のP値を次のように定義する:

$$pvalue_{\text{Pearson}}(a, b, c, d | OR = \omega) = 1 - \text{cdf}(\text{Chisq}(1), \chi^2).$$

仮説 $OR = \omega$ に関するデータの数値 a, b, c, d に関するPearsonの χ^2 統計量の値 $\chi^2 = \chi^2(a, b, c, d | \delta(a, b, c, d | \omega))$ は a, b, c, d, ω だけで計算される数値なので、これでP値が定義されている。

対応する信頼区間: このP値の定義に対応するオッズ比パラメータ OR の信頼度 $1 - \alpha$ の信頼区間は次のように定義される:

$$\text{confint}_{\text{Pearson}}^{\text{OR}}(a, b, c, d | \alpha) = \{ \omega > 0 \mid pvalue_{\text{Pearson}}(a, b, c, d | OR = \omega) \geq \alpha \}.$$

この信頼区間を計算するために使えるシンプルな公式はないように思われる。

信頼区間の計算法1: ω に関する次の方程式の解を2つ求めれば、その2つがオッズ比パラメータ OR の信頼区間の両端の値になる:

$$pvalue_{\text{Pearson}}(a, b, c, d | OR = \omega) = \alpha.$$

信頼区間の計算法2: まず、 $\chi^2_{\nu, \alpha}$ を

$$\chi^2_{\nu, \alpha} = \text{quantile}(\text{Chisq}(\nu), 1 - \alpha)$$

と定める。そして、 $-\min(b, c) < \delta < \min(a, d)$ と次を満たす δ を2つ求める:

$$\chi^2(a, b, c, d | \delta) = \delta^2 \left(\frac{1}{a-\delta} + \frac{1}{b+\delta} + \frac{1}{c+\delta} + \frac{1}{d-\delta} \right) = \chi^2_{1, \alpha}$$

そして、それらの δ に対して、 ω を

$$\omega = \frac{(a - \delta)(d - \delta)}{(b + \delta)(c + \delta)}$$

によって対応させれば、信頼区間の両端の値が得られる。

文献: この節の構成は次の論文に書いてある方法の連続補正無し版になっている:

- Jerome Cornfield, A Statistical Problem Arising from Retrospective Studies, Berkeley Symposium on Mathematical Statistics and Probability, 1956: 135-148 (1956) [[link](https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Proceedings%20of%20the%20Third%20Berkeley%20Symposium%20on%20Mathematical%20Statistics%20and%20Probability) (<https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Proceedings%20of%20the%20Third%20Berkeley%20Symposium%20on%20Mathematical%20Statistics%20and%20Probability>)]

4.2 Pearsonの χ^2 検定版のオッズ比に関するP値と信頼区間の計算例

データが次の場合の仮説「オッズ比パラメータは $OR = 1$ である」のPearsonの χ^2 検定版のP値とオッズ比 OR の 95% 信頼区間を計算してみよう:

| | 商品を購入した | 購入しなかった | 合計 |
|-------|---------|---------|------|
| デザインA | 49 | 965 | 1014 |
| デザインB | 26 | 854 | 880 |
| 合計 | 75 | 1819 | 1894 |

結果は次のようになる:

- (仮説「オッズ比パラメータは $OR = 1$ である」のPearsonの χ^2 検定版のP値) ≈ 0.03661
- (オッズ比パラメータ OR のPearsonの χ^2 検定版の 95% 信頼区間) $\approx [1.0318, 2.6957]$

信頼区間についてはJulia言語版の計算例のみを示す。WolframAlphaでこの信頼区間を求めるることはかなり面倒である。

4.2.1 WolframAlphaによるPearsonの χ^2 検定のP値の計算の仕方

$\omega = 1$ の場合には、Pearsonの χ^2 統計量は、

$$\chi^2 = \frac{(a + b + c + d)(ad - bc)^2}{(a + b)(c + d)(a + c)(b + d)}$$

となるのであった。これを用いてP値を計算してみよう。

($a+b+c+d$) $(ad-bc)^2 / ((a+b)(c+d)(a+c)(b+d))$ where $a=49.0$, $b=965$, $c=26$, $d=854$ → 実行
([https://www.wolframalpha.com/input/?i=\(a+b+c+d\)\(ad-bc\)^2/\(\(a+b\)\(c+d\)\(a+c\)\(b+d\)\)+where+a=49.0+b=965+c=26+d=854](https://www.wolframalpha.com/input/?i=(a+b+c+d)(ad-bc)^2/((a+b)(c+d)(a+c)(b+d))+where+a=49.0+b=965+c=26+d=854))
→ 4.36824 (独立性に関するPearsonの χ^2 統計量)

1 - cdf(ChisqDistribution(1), 4.36824) → 実行 ([https://www.wolframalpha.com/input/?i=1-cdf\(ChisqDistribution\(1\),4.36824\)](https://www.wolframalpha.com/input/?i=1-cdf(ChisqDistribution(1),4.36824))) → 0.0366148 (P値)

4.2.2 Julia言語によるPearsonの χ^2 検定版のオッズ比に関するP値と信頼区間の計算の仕方(1)

$\omega = 1$ の場合のオッズ比に関するPearsonの χ^2 検定版のP値は、独立性に関するPearsonの χ^2 検定のP値そのものになる。それを上でやったのと同じ方法で計算する。

信頼区間の側は函数の零点を見つけてくれる函数を使って求めてみよう。

In [38]:

```

1 a, b, c, d = 49, 965, 26, 854
2 @show a, b, c, d
3 χ² = (a+b+c+d)*(a*d-b*c)^2/((a+b)*(c+d)*(a+c)*(b+d))
4 @show χ²
5 @show pvalue = ccdf(Chisq(1), χ²)
6 α = 0.05
7 @show α
8 confint = find_zeros(0.5, 3.0) do ω
9     A, B, C = 1-ω, a+d+ω*(b+c), a*d-ω*b*c
10    δ = 2C/(B + √(B^2 - 4A*C))
11    χ² = δ^2 * (1/(a-δ) + 1/(b+δ) + 1/(c+δ) + 1/(d-δ))
12    ccdf(Chisq(1), χ²) - α
13 end
14 @show confint;

```

```

(a, b, c, d) = (49, 965, 26, 854)
χ² = 4.368235947209686
pvalue = ccdf(Chisq(1), χ²) = 0.03661487827607113
α = 0.05
confint = [1.031778110800563, 2.6957054358850328]

```

`f(x, y) do z ... end` 構文については

- <https://docs.julialang.org/en/v1/manual/functions/#Do-Block-Syntax-for-Function-Arguments> (<https://docs.julialang.org/en/v1/manual/functions/#Do-Block-Syntax-for-Function-Arguments>)

を参照せよ. `find_zeros` フィルの使い方については,

- <https://juliamath.github.io/Roots.jl/stable/#Basic-usage> (<https://juliamath.github.io/Roots.jl/stable/#Basic-usage>).

を参照せよ.

4.2.3 Julia言語によるPearsonの χ^2 検定版のオッズ比に関するP値と信頼区間の計算の仕方(2)

このノートブックの最初の方で定義した函数を使って計算.

In [39]:

```

1 a, b, c, d = 49, 965, 26, 854
2 @show a, b, c, d
3 @show chisqstat_or(a, b, c, d; ω=1)
4 @show pvalue_or_pearson_chisq(a, b, c, d; ω=1)
5 @show confint_or_pearson_chisq(a, b, c, d; α=0.05);

```

```

(a, b, c, d) = (49, 965, 26, 854)
chisqstat_or(a, b, c, d; ω = 1) = 4.368235947209686
pvalue_or_pearson(a, b, c, d; ω = 1) = 0.03661487827607113
confint_or_pearson(a, b, c, d; α = 0.05) = [1.031778110800563, 2.6957054358850328]

```

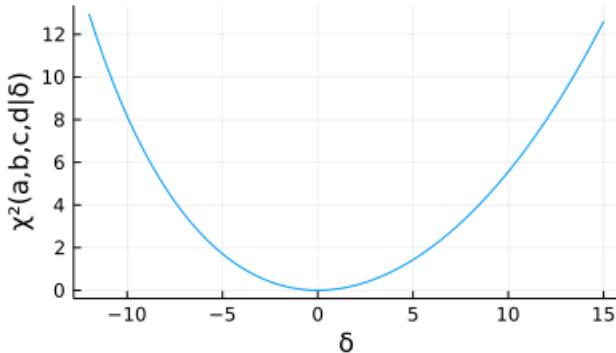
4.2.4 Julia言語によるPearsonの χ^2 検定版のオッズ比に関するP値の視覚化

まず, 次の函数のグラフを描いてみよう:

$$\delta \mapsto \chi^2(a, b, c, d | \delta) = \delta^2 \left(\frac{1}{a-\delta} + \frac{1}{b+\delta} + \frac{1}{c+\delta} + \frac{1}{d-\delta} \right).$$

```
In [40]: 1 a, b, c, d = 49, 965, 26, 854
2 plot(δ → _chisqstat_or(a,b,c,d,δ), -12, 15; label="")
3 plot!(; xguide="δ", yguide="χ²(a,b,c,d|δ)")
4 title!("a, b, c, d = $a, $b, $c, $d")
```

Out[40]: a, b, c, d = 49, 965, 26, 854

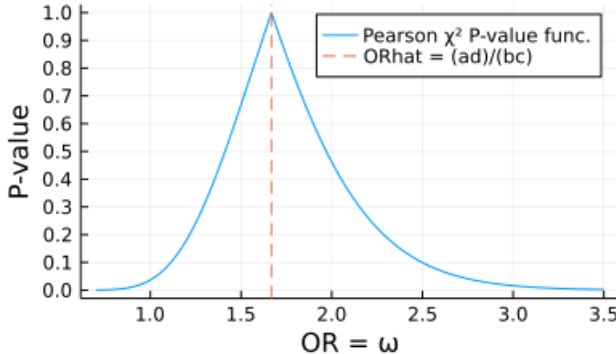


このように $\delta \mapsto \chi^2(a, b, c, d | \delta)$ は $\delta = 0$ で最小値 0 になるU型のグラフになる。ゆえに、 δ に関する方程式 $\chi^2(a, b, c, d | \delta) = \chi^2_{1,\alpha}$ の $-\min(b, c) < \delta < \min(a, d)$ を満たす解は負値と正値の2つになる。

次に、P値函数 $\omega \mapsto \text{pvalue}(a, b, c, d | \text{OR} = \omega)$ のグラフを描いてみよう。

```
In [41]: 1 a, b, c, d = 49, 965, 26, 854
2 plot(ω → pvalue_or_pearson_chisq(a,b,c,d;ω), 0.7, 3.5; label="Pearson χ² P-value func.")
3 vline!([oddsratiohat(a,b,c,d)]; label="ORhat = (ad)/(bc)", ls=:dash)
4 plot!(; xguide="OR = ω", yguide="P-value")
5 plot!(; ytick=0:0.1:1)
6 title!("a, b, c, d = $a, $b, $c, $d")
```

Out[41]: a, b, c, d = 49, 965, 26, 854

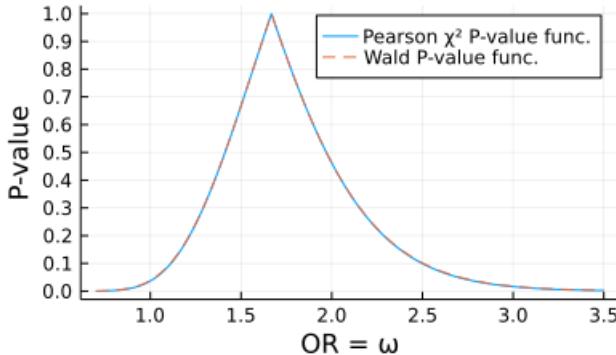


このようにP値函数 $\omega \mapsto \text{pvalue}(a, b, c, d | \text{OR} = \omega)$ は $\omega = \widehat{\text{OR}} = (ad)/(bc)$ で最大値 1 になり、「とんがり帽子」の形のグラフになる。

Wald版のP値函数のグラフと重ねてプロットすると次のようになる。

```
In [42]: 1 a, b, c, d = 49, 965, 26, 854
2 plot(w → pvalue_or_pearson_chisq(a,b,c,d; w), 0.7, 3.5; label="Pearson  $\chi^2$  P-value func.")
3 plot!(w → pvalue_or_wald(a,b,c,d; w), 0.7, 3.5; label="Wald P-value func.", ls=:dash)
4 plot!(; xguide="OR =  $\omega$ ", yguide="P-value")
5 plot!(; ytick=0:0.1:1)
6 title!("a, b, c, d = $a, $b, $c, $d")
```

Out[42]: a, b, c, d = 49, 965, 26, 854



この場合には a, b, c, d が十分に大きいので、ほぼぴったり重なっている!

このことから、 a, b, c, d が十分に大きな場合には、計算法が複雑なPearsonの χ^2 検定版のP値と信頼区間を使う必要がないことがわかる。そのような場合にはWald型のP値と信頼区間で実用的には十分である。

P値と 95% 信頼区間も以下のようにほぼ同じになっている:

- P値の比較:
 - (仮説「オッズ比パラメータは $OR = 1$ である」のPearsonの χ^2 検定版のP値) $\approx 3.7\%$
 - (仮説「オッズ比パラメータは $OR = 1$ である」のWald版のP値) $\approx 3.8\%$
- 信頼区間の比較:
 - (オッズ比パラメータ OR のPearsonの χ^2 検定版の 95% 信頼区間) $\approx [1.03, 2.70]$
 - (オッズ比パラメータ OR のWald版の 95% 信頼区間) $\approx [1.03, 2.71]$

数学的注意: 数学の世界では、定義が全く異なる2つの量がある条件のもとで近似的によく一致するというようなことが起こる。異なる動機や思想のもとで定義された2つの量が現実世界での応用では「同じ」とみなしてよいだけ近似的によく一致することがある。そのような場合に元の動機や思想にこだわって解釈しようとするることは誤りである。

注意: a, b, c, d が小さい場合には違いが大きくなる場合がある。

```
In [43]: 1 a, b, c, d = 4, 1, 1, 5
2 @show pvalue_or_pearson_chisq(a,b,c,d)
3 @show pvalue_or_wald(a,b,c,d)
4 println()
5 @show confint_or_pearson_chisq(a,b,c,d)
6 @show confint_or_wald(a,b,c,d);
```

```
pvalue_or_pearson(a, b, c, d) = 0.03568232934330229
pvalue_or_wald(a, b, c, d) = 0.05563253253702121
```

```
confint_or_pearson(a, b, c, d) = [1.1871236326245318, 337.7203835779611]
confint_or_wald(a, b, c, d) = [0.930440993688474, 429.9036722514896]
```

このように a, b, c, d が小さい場合に信頼区間の幅はものすごく大きくなる。

4.2.5 R言語によるPearsonの χ^2 検定のP値の計算の仕方

独立性に関するPearsonの χ^2 検定のP値は次のようにして計算できる。

```
A = matrix(c(49, 965, 26, 854), 2, 2, byrow=T)
result = chisq.test(A, correct=F)
```

```
In [44]: 1 R"""
2 A = matrix(c(49, 965, 26, 854), 2, 2, byrow=T)
3 result = chisq.test(A, correct=F)
4 """
```

Out[44]: RObject{VecSxp}

```
Pearson's Chi-squared test

data: A
X-squared = 4.3682, df = 1, p-value = 0.03661
```

4.3 「オッズ比パラメータはOR=ωである」という仮説下での最尤法との関係

データの数値 a, b, c, d が与えらていると仮定し, $a + b = m, c + d = n$ であるとする.

4.3.1 2つの二項分布モデル全体での最尤法

2つの二項分布モデルの対数尤度函数は

$$\begin{aligned}(p, q) \mapsto L &= \log P(a, b, c, d | m, n, p, q) \\&= \log \left(\binom{m}{a} p^a (1-p)^b \binom{n}{c} q^c (1-q)^d \right) \\&= a \log p + b \log(1-p) + c \log q + d \log(1-q) + (\text{constant w.r.t. } p, q).\end{aligned}$$

なので,

$$\frac{\partial L}{\partial p} = \frac{a}{p} - \frac{b}{1-p} = \frac{a - (a+b)p}{p(1-p)}.$$

これより, $\partial L / \partial p = 0$ と $p = \hat{p} := a/(a+b) = a/m$ は同値になる. 同様にして, $\partial L / \partial q = 0$ と $q = \hat{q} := c/(c+d) = c/n$ は同値になる. このことより, 対数尤度 L を最大化する p, q は $\hat{p} = a/m, \hat{q} = c/n$ になることがわかる.

$p = \hat{p} = a/m, q = \hat{q} = c/n$ のときの2つの二項分布モデルでの期待値は

$$m\hat{p} = a, \quad m(1-\hat{p}) = b, \quad n\hat{q} = c, \quad n(1-\hat{q}) = d$$

になる. これはパラメータの動く範囲に制限を付けない2つの二項分布モデルの場合には, パラメータに最尤推定値を代入して得られる確率分布(予測分布という)の期待値が, データの値 a, b, c, d そのものになることを意味している.

注意: この性質は, 4つのPoisson分布モデルでも四項分布モデルでも成立している.

4.3.2 条件OR=ωで制限した2つの二項分布モデルでの最尤法

$\omega > 0$ を任意に取って固定する.

条件 $OR = (p/(1-p))/(q/(1-q)) = \omega$ という条件の下で対数尤度を最大化する問題をLagrangeの未定乗数法を使って解こう. 関数 $(p, q, \delta) \mapsto M$ を次のように定める:

$$\begin{aligned}M &= L - \delta(\log OR - \log \omega) \\&= a \log p + b \log(1-p) + c \log q + d \log(1-q) \\&\quad - \delta(\log p - \log(1-p) - \log q + \log(1-q) - \log \omega) + \text{const.}\end{aligned}$$

このとき,

$$\begin{aligned}\frac{\partial M}{\partial p} &= \frac{a}{p} - \frac{b}{1-p} - \frac{\delta}{p} - \frac{\delta}{1-p} = \frac{a-\delta}{p} - \frac{b+\delta}{1-p}, \\ \frac{\partial M}{\partial q} &= \frac{c}{q} - \frac{d}{1-q} + \frac{\delta}{q} + \frac{\delta}{1-q} = \frac{c+\delta}{q} - \frac{d-\delta}{1-q}, \\ -\frac{\partial M}{\partial \delta} &= \log OR - \log \omega = \log \frac{p/(1-p)}{q/(1-q)} - \log \omega.\end{aligned}$$

これらがすべて 0 になることと次が成立することは同値である:

$$\frac{p}{1-p} = \frac{a-\delta}{b+\delta}, \quad \frac{q}{1-q} = \frac{c+\delta}{d-\delta}, \quad OR = \frac{(a-\delta)(d-\delta)}{(b+\delta)(c+\delta)} = \omega.$$

ゆえに

$$\frac{(a-\delta)(d-\delta)}{(b+\delta)(c+\delta)} = \omega, \quad -\min(b,c) < \delta < \min(a,d)$$

を満たすように δ を取り, \tilde{p}, \tilde{q} を

$$\frac{\tilde{p}}{1-\tilde{p}} = \frac{a-\delta}{b+\delta}, \quad \frac{\tilde{q}}{1-\tilde{q}} = \frac{c+\delta}{d-\delta}$$

を満たすように

$$\tilde{p} = \frac{\tilde{a}}{\tilde{a} + \tilde{b}} = \frac{\tilde{a}}{m}, \quad \tilde{q} = \frac{\tilde{c}}{\tilde{c} + \tilde{d}} = \frac{\tilde{c}}{n}$$

と定めれば, $p = \tilde{p}, q = \tilde{q}$ が条件 $OR = \omega$ の下での最尤推定値になる.

そのとき, モデルの期待値は

$$m\tilde{p} = a - \delta, \quad m(1-\tilde{p}) = b + \delta, \quad n\tilde{q} = c + \delta, \quad n(1-\tilde{q}) = d - \delta$$

になる. これでオッズ比に関するPearsonの χ^2 検定版のP値の構成法の(1)のステップが本質的に条件 $OR = \omega$ の下での最尤法そのものであったことがわかった.

4.3.3 Wilks's theoremの適用

[Wilks' theorem \(\[https://en.wikipedia.org/wiki/Wilks%27_theorem\]\(https://en.wikipedia.org/wiki/Wilks%27_theorem\)\)](https://en.wikipedia.org/wiki/Wilks%27_theorem) について説明しよう.

適当な緩い条件のもとで, パラメータ θ, η を持つ統計モデル $M(\theta, \eta) = D(\theta, \eta)^n$ (ただし $\theta = (\theta_1, \dots, \theta_\nu)$ は ν 個の独立なパラメータ達であるとする)について以下が成立している.

$X = (X_1, \dots, X_n)$ はパラメータの値を $\theta = \theta_0, \eta = \eta_0$ に設定して得られるモデルの分布 $M(\theta_0, \eta_0)$ 従う確率変数であると仮定する.

パラメータの動ける範囲に制限を付けないモデル $M(\theta, \eta)$ におけるデータ X に関する最尤推定量を $\hat{\theta}, \hat{\eta}$ と表す. ($\hat{\theta}, \hat{\eta}$ はモデル内確率変数になる.)

パラメータの動ける範囲に $\theta = \theta_0$ という制限によってパラメータ η しか動けない統計モデル $M(\theta_0, \eta)$ におけるデータ X に関する最尤推定量を $\tilde{\eta}$ と表す. ($\tilde{\eta}$ もモデル内確率変数になる.)

このとき, n が大きければ, 対数尤度比は自由度 ν の χ^2 分布に近似的に従う:

$$2 \log \frac{(X \text{ に関する } M(\hat{\theta}, \hat{\eta}) \text{ の尤度})}{(X \text{ に関する } M(\theta_0, \tilde{\eta}) \text{ の尤度})} \sim \text{Chisq}(\nu), \quad \text{approximately.} \quad (\text{W})$$

これが **Wilksの定理** である.

要するにパラメータの動ける範囲の次元を, $\theta = (\theta_1, \dots, \theta_\nu)$ を $\theta_0 = (\theta_{0,1}, \dots, \theta_{0,\nu})$ に固定することによって, ν 次元下げた場合の最尤法を考えると, 自然に自由度 ν の χ^2 分布に近似的に従う統計量(確率変数)が得られるという仕組みになっている.

注意: 以上では統計モデル $M(\theta, \eta)$ は標本分布の形をしていると仮定したが, その条件は緩められる. 例えばBernoulli分布の標本分布を二項分布で置き換えるても, 尤度函数はパラメータに依存しない定数倍を除いて変わらないので, 上と同様の結果が成立する. **注意終**

Wilksの定理(W)の左辺の対数尤度比は $n \rightarrow \infty$ で漸近的に同値な別の確率変数に置き換えられる.

例えば, 統計モデル $M(\theta, \eta)$ が2つの二項分布モデルで $\nu = 1, \theta = OR, \theta_0 = \omega, \eta = q$ のとき,

$$OR = \frac{p/(1-p)}{q/(1-q)}$$

という条件によって p を定めると, 統計モデルの確率質量函数は次のように書ける:

$$P(x|\theta, \eta) = P(a, b, c, d | OR, q) = \binom{a+b}{a} p^a (1-p)^b \binom{c+d}{c} q^c (1-q)^d.$$

この統計モデルに従う確率変数を $X = (a, b, c, d)$ と書き, $m = a + b, n = c + d, \hat{p} = a/m, \hat{q} = c/n$ とおくと,

$$\hat{\theta} = \widehat{OR} = \frac{ad}{bc} = \frac{\hat{p}/(1-\hat{p})}{\hat{q}(1-\hat{q})}, \quad \hat{\eta} = \hat{q}$$

となるのであった。さらに、

$$\frac{(a-\delta)(d-\delta)}{(b+\delta)(c+\delta)} = \omega, \quad -\min(b,c) < \delta < \min(a,d)$$

を満たす δ を取り、

$$\tilde{a} = a - \delta, \quad \tilde{b} = b + \delta, \quad \tilde{c} = c + \delta, \quad \tilde{d} = d - \delta, \quad \tilde{p} = \frac{\tilde{a}}{m}, \quad \tilde{q} = \frac{\tilde{c}}{n},$$

とおくと、 $\tilde{\eta} = \tilde{q}$ となることから、以下が成立することがわかる：

$$(X \text{ に関する } M(\hat{\theta}, \hat{\eta}) \text{ の尤度}) = \binom{a+b}{a} \left(\frac{a}{m} \right)^a \left(\frac{b}{m} \right)^b \binom{c+d}{c} \left(\frac{c}{n} \right)^c \left(\frac{d}{n} \right)^d,$$
$$(X \text{ に関する } M(\theta_0, \tilde{\eta}) \text{ の尤度}) = \binom{a+b}{a} \left(\frac{\tilde{a}}{m} \right)^a \left(\frac{\tilde{b}}{m} \right)^b \binom{c+d}{c} \left(\frac{\tilde{c}}{n} \right)^c \left(\frac{\tilde{d}}{n} \right)^d$$

となるので、

$$\frac{(X \text{ に関する } M(\hat{\theta}, \hat{\eta}) \text{ の尤度})}{(X \text{ に関する } M(\theta_0, \tilde{\eta}) \text{ の尤度})} = \left(\frac{a}{\tilde{a}} \right)^a \left(\frac{b}{\tilde{b}} \right)^b \left(\frac{c}{\tilde{c}} \right)^c \left(\frac{d}{\tilde{d}} \right)^d.$$

ゆえに近似的に自由度 1 の χ^2 分布に従う対数尤度比は次のようにシンプルな式で表される：

$$2 \log \frac{(X \text{ に関する } M(\hat{\theta}, \hat{\eta}) \text{ の尤度})}{(X \text{ に関する } M(\theta_0, \tilde{\eta}) \text{ の尤度})}$$
$$= 2a \log \frac{a}{\tilde{a}} + 2b \log \frac{b}{\tilde{b}} + 2c \log \frac{c}{\tilde{c}} + 2d \log \frac{d}{\tilde{d}}.$$

これに、 $f(x) = 2x \log(x/\tilde{x})$ のとき、 $f'(x) = 2 \log(x/\tilde{x}) + 2$, $f''(x) = 2/x$ より、

$$2x \log \frac{x}{\tilde{x}} = 2(x - \tilde{x}) + \frac{(x - \tilde{x})^2}{\tilde{x}} + O((x - \tilde{x})^2)$$

となることを適用すると、 $a + b + c + d = \tilde{a} + \tilde{b} + \tilde{c} + \tilde{d}$ より、

$$2a \log \frac{a}{\tilde{a}} + 2b \log \frac{b}{\tilde{b}} + 2c \log \frac{c}{\tilde{c}} + 2d \log \frac{d}{\tilde{d}}$$
$$\approx \frac{(a - \tilde{a})^2}{\tilde{a}} + \frac{(b - \tilde{b})^2}{\tilde{b}} + \frac{(c - \tilde{c})^2}{\tilde{c}} + \frac{(d - \tilde{d})^2}{\tilde{d}} = (\text{Pearsonの}\chi^2\text{統計量})$$

と Pearson の χ^2 統計量が得られ、

$$(\text{Pearsonの}\chi^2\text{統計量}) \sim \text{Chisq}(1), \quad \text{approximately}$$

が得られる。これが Pearson の χ^2 検定の基礎になる。

注意: 以上の議論は多項分布モデルや 2×2 よりもサイズが大きな分割表にも比較的容易に一般化される。

注意: 以上の議論をさらに一般化すると Pearson の χ^2 統計量の一般化である **スコア統計量** を使った χ^2 検定の構成法が得られる。

4.4 Wilks' theoremの数値的確認例

すでに、

- 「検定と信頼区間: 比率の検定と信頼区間」のノート
(<https://nbviewer.org/github/genkuroki/Statistics/blob/master/2022/09%20Hypothesis%20testing%20and%20confidence%20inter%20Proportion.ipynb>)

の「おまけ(より進んだ話題): 一般的 χ^2 検定で使う χ^2 分布の自由度」の節で、多項分布の場合で自由度が 1 以外の χ^2 分布を利用する場合の例を示している。

以下では例を少しだけ追加しておこう。

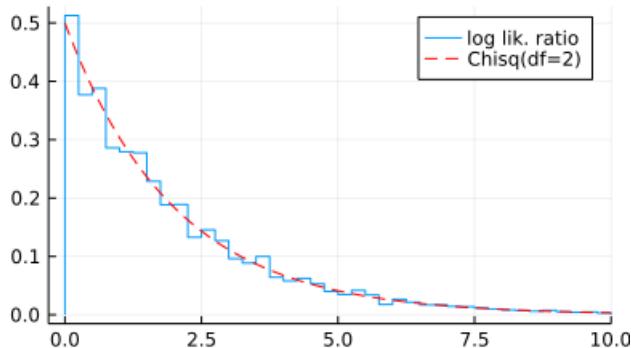
```
In [45]: 1 function plot_wilks_theorem(logmle1, logmle0, randsample0, tmpsample0, df;
2     L=10^5, bin=:auto, kwargs...)
3     loglikrat = similar=zeros(), L)
4     tmp = [tmpsample0() for _ in 1:nthreads()]
5     @threads for i in 1:L
6         X = randsample0(tmp[threadid()])
7         loglikrat[i] = 2(logmle1(X) - logmle0(X))
8     end
9     stephist(loglikrat; norm=true, bin, label="log lik. ratio")
10    plot!(Chisq(df); label="Chisq(df=$df)", ls=:dash, c=:red)
11    plot!(; kwargs...)
12 end
```

Out[45]: plot_wilks_theorem (generic function with 1 method)

4.4.1 2つの二項分布モデルのパラメータを2つとも固定した場合 (自由度2)

```
In [46]: 1 plot_wilks_theorem(
2     ((a,b,c,d),) → xlogx(a) + xlogx(b) + xlogx(c) + xlogx(d),
3     ((a,b,c,d),) → begin
4         m, n, p = 160, 320, 0.2
5         ā = m*p
6         ī = m*(1-p)
7         ē = n*p
8         ī = n*(1-p)
9         xlogy(a, ā) + xlogy(b, ī) + xlogy(c, ē) + xlogy(d, ī)
10    end,
11    tmp → begin
12        m, n, p = 160, 320, 0.2
13        a = rand(Binomial(m, p))
14        c = rand(Binomial(n, p))
15        a, m-a, c, n-c
16    end,
17    () → nothing,
18    2;
19    title="M(OR, q) vs M(1, 0.2) (m=160, n=320, p=0.2)",
20    xlim=(-0.3, 10),
21    bin=0:0.25:10,
22    L=10^6,
23 )
```

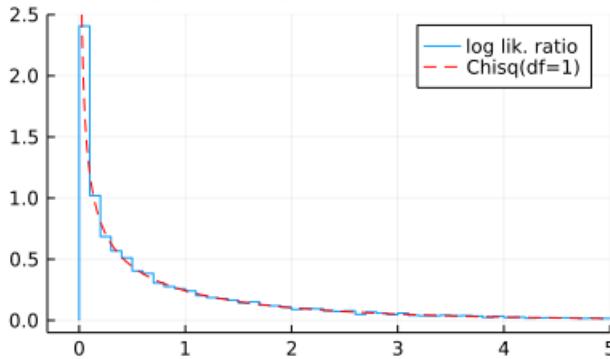
Out[46]: M(OR, q) vs M(1, 0.2) (m=160, n=320, p=0.2)



4.4.2 2つの二項分布モデルの独立性に関する対数尤度比検定の場合 (OR=1とした場合, 自由度1)

```
In [47]: 1 plot_wilks_theorem(
2     ((a,b,c,d),) → xlogx(a) + xlogx(b) + xlogx(c) + xlogx(d),
3     ((a,b,c,d),) → begin
4         N = a + b + c + d
5         ā = (a+b)*(a+c)/N
6         ī = (a+b)*(b+d)/N
7         ķ = (c+d)*(a+c)/N
8         ı = (c+d)*(b+d)/N
9         xlogy(a, ā) + xlogy(b, ī) + xlogy(c, ķ) + xlogy(d, ı)
10    end,
11    tmp → begin
12        m, n, p = 160, 320, 0.2
13        a = rand(Binomial(m, p))
14        c = rand(Binomial(n, p))
15        a, m-a, c, n-c
16    end,
17    () → nothing,
18    1;
19    title="M(OR, q) vs M(1, q) (m=160, n=320, p=0.2)",
20    xlim=(-0.3, 5), ylim=(-0.1, 2.5),
21    bin=0:0.1:10,
22    L=10^6,
23 )
```

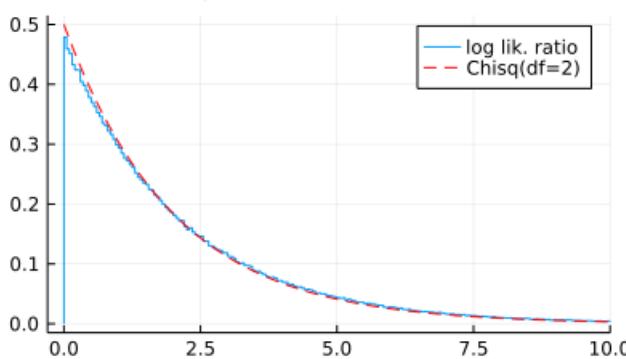
Out[47]: M(OR, q) vs M(1, q) (m=160, n=320, p=0.2)



4.4.3 正規分布モデルのパラメータを2つとも固定した場合 (自由度2)

```
In [48]: 1 plot_wilks_theorem(
2     X → (normal = fit_mle(Normal, X); loglikelihood(normal, X)),
3     X → loglikelihood(Normal(), X),
4     tmp → rand!(Normal(), tmp),
5     () → zeros(20),
6     2;
7     title="Normal(μ,σ)n vs. Normal(0,1)n (n=20)",
8     xlim=(-0.3, 10),
9     L=10^6
10 )
```

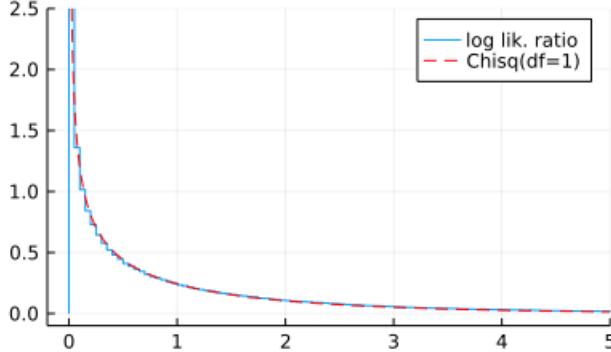
Out[48]: Normal(μ,σ)ⁿ vs. Normal(0,1)ⁿ (n=20)



4.4.4 正規分布モデルでσのみを固定した場合

```
In [49]: 1 plot_wilks_theorem(
2     X → (normal = fit_mle(Normal, X); loglikelihood(normal, X)),
3     X → loglikelihood(Normal(mean(X), 1), X),
4     tmp → rand!(Normal(), tmp),
5     () → zeros(20),
6     1;
7     title="Normal( $\mu, \sigma$ )n vs. Normal( $\mu, 1$ )n (n=20)",
8     xlim=(-0.2, 5), ylim=(-0.1, 2.5),
9     L=10^6
10 )
```

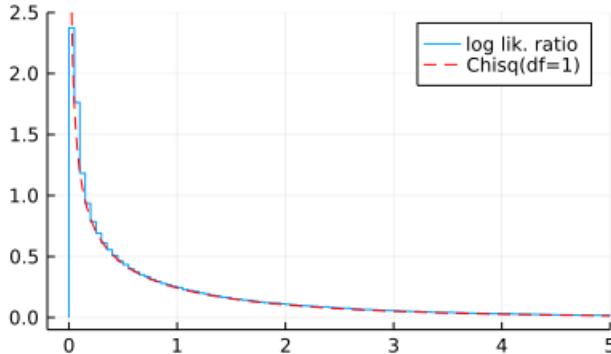
Out[49]: Normal(μ, σ)ⁿ vs. Normal($\mu, 1$)ⁿ (n=20)



4.4.5 正規分布モデルで μ のみを固定した場合 (自由度1)

```
In [50]: 1 plot_wilks_theorem(
2     X → (normal = fit_mle(Normal, X); loglikelihood(normal, X)),
3     X → loglikelihood(Normal(0, std(X; mean=0)), X),
4     tmp → rand!(Normal(), tmp),
5     () → zeros(20),
6     1;
7     title="Normal( $\mu, \sigma$ )n vs. Normal(0,  $\sigma$ )n (n=20)",
8     xlim=(-0.2, 5), ylim=(-0.1, 2.5),
9     L=10^6
10 )
```

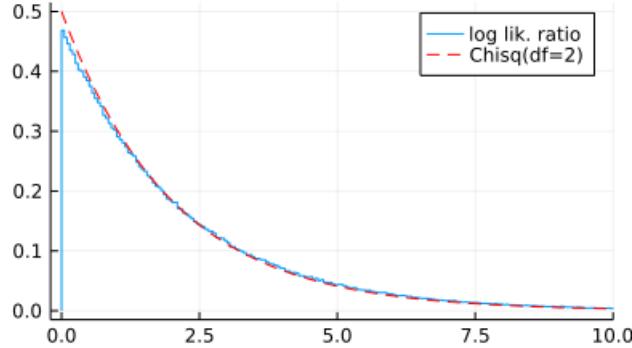
Out[50]: Normal(μ, σ)ⁿ vs. Normal(0, σ)ⁿ (n=20)



4.4.6 ガンマ分布モデルのパラメータを2つとも固定した場合 (自由度2)

```
In [51]: 1 plot_wilks_theorem(
2     X → (gamma = fit_mle(Gamma, X); loglikelihood(gamma, X)),
3     X → loglikelihood(Gamma(2, 3), X),
4     tmp → rand!(Gamma(2, 3), tmp),
5     () → zeros(20),
6     2;
7     title="Gamma(α, θ)n vs. Gamma(2, 3)n (n=20)",
8     xlim=(-0.2, 10),
9     L=10^6
10 )
```

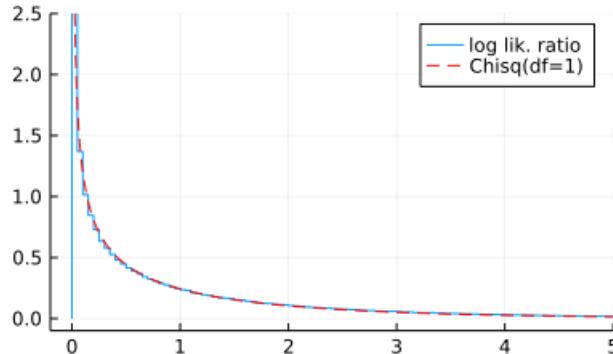
Out[51]: Gamma(α, θ)ⁿ vs. Gamma(2, 3)ⁿ (n=20)



4.4.7 ガンマ分布モデルで形状パラメータαのみを固定した場合 (自由度1)

```
In [52]: 1 plot_wilks_theorem(
2     X → (gamma = fit_mle(Gamma, X); loglikelihood(gamma, X)),
3     X → (μ = mean(X); loglikelihood(Gamma(2, μ/2), X)),
4     tmp → rand!(Gamma(2, 3), tmp),
5     () → zeros(20),
6     1;
7     title="Gamma(α, μ/α)n vs. Gamma(2, μ/2)n (n=20)",
8     xlim=(-0.2, 5), ylim=(-0.1, 2.5),
9     L=10^6
10 )
```

Out[52]: Gamma(α, μ/α)ⁿ vs. Gamma(2, μ/2)ⁿ (n=20)



4.5 問題: 「リスク比パラメータはRR=ρである」という仮説下での最尤法

以下示せ.

RR = ρ という条件の下での、データの数値 a, b, c, d に関する2つの二項分布モデルの最尤推定値 \tilde{p}, \tilde{q} が

$$\tilde{p} = \frac{a - \Delta}{a - \Delta + b}, \quad \tilde{q} = \frac{c + \Delta}{c + \Delta + d}$$

になることを示せ。ただし Δ は次の条件で定める:

$$\frac{(a - \Delta)(c + \Delta + d)}{(a - \Delta + b)(c + \Delta)} = \rho, \quad -c < \Delta < a.$$

注意: パラメータの値をその最尤推定値にしたとき、2つの二項分布モデルの期待値は次のようになる:

$$m\tilde{p} = \frac{m}{m - \Delta}(a - \Delta), \quad m(1 - \tilde{p}) = \frac{m}{m - \Delta}b,$$

$$n\tilde{q} = \frac{n}{n + \Delta}(c + \Delta), \quad m(1 - \tilde{q}) = \frac{n}{n + \Delta}d.$$

解答例: Lagrangeの未定乗数法をオッズ比を扱った場合と同様に用いる。

函数 $(p, q, \Delta) \mapsto M$ を次のように定める:

$$M = \log P(a, b, c, d|m, n, p, q) - \Delta(\log \text{RR} - \log \rho)$$

$$= a \log p + b \log(1 - p) + c \log q + d \log(1 - q)$$

$$- \Delta(\log p - \log q - \log \rho) + \text{const.}$$

このとき、

$$\frac{\partial M}{\partial p} = \frac{a}{p} - \frac{b}{1-p} - \frac{\Delta}{p} = \frac{a - \Delta}{p} - \frac{b}{1-p},$$

$$\frac{\partial M}{\partial q} = \frac{c}{q} - \frac{d}{1-q} + \frac{\Delta}{q} = \frac{c + \delta}{q} - \frac{d}{1-q},$$

$$-\frac{\partial M}{\partial \Delta} = \log \text{RR} - \log \omega = \log \frac{p}{q} - \log \omega.$$

これらがすべて 0 になることと次が成立することは同値である:

$$\frac{p}{1-p} = \frac{a - \Delta}{b}, \quad \frac{q}{1-q} = \frac{c + \Delta}{d}, \quad \text{RR} = \frac{p}{q} = \rho.$$

さらに

$$\frac{p}{1-p} = \frac{a - \Delta}{b}, \quad \frac{q}{1-q} = \frac{c + \Delta}{d}$$

と

$$p = \tilde{p} := \frac{a - \Delta}{a - \Delta + b}, \quad q = \tilde{q} := \frac{c + \Delta}{c + \Delta + d}$$

は同値であり、これが成立しているとき、 $\text{RR} = p/q = \rho$ は

$$\frac{(a - \Delta)(c + \Delta + d)}{(a - \Delta + b)(c + \Delta)} = \rho$$

と同値である。これと $-c < \Delta < a$ を満たす Δ を取って、すぐ上の式で \tilde{p}, \tilde{q} を定義すると、 $p = \tilde{p}, q = \tilde{q}$ が $\text{RR} = \rho$ という条件の下での最尤推定量になる。

解答終

4.6 問題: Pearsonの χ^2 検定版のリスク比に関するP値と信頼区間の定義

上の問題の結果を用いてリスク比に関するPearsonの χ^2 検定版のP値と信頼区間を適切に定義せよ。

注意: 適切な定義を自分で考えることは物事を楽に理解するために必要なことである。

解答例: Δ を

$$\frac{(a - \Delta)(n + \Delta)}{(m - \Delta)(c + \Delta)} = \rho, \quad -c < \Delta < a.$$

という条件で定める。このような Δ は

$$A = \rho - 1, \quad B = n - a + \rho(m - c), \quad C = an - \rho mc$$

と置き,

$$\Delta = \frac{2C}{B + \sqrt{B^2 - 4AC}}$$

とおけば求まる. さらに, $\tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}$ を次のように定める:

$$\begin{aligned}\tilde{a} &= \frac{m}{m - \Delta}(a - \Delta), & \tilde{b} &= \frac{m}{m - \Delta}b, \\ \tilde{c} &= \frac{n}{n + \Delta}(c + \Delta), & \tilde{d} &= \frac{n}{n + \Delta}d.\end{aligned}$$

このとき, $(\tilde{a}/m)/(\tilde{c}/n) = \rho$ となっていることに注意せよ.

Pearsonの χ^2 統計量 χ^2 を次のように定める:

$$\chi^2 = \frac{(a - \tilde{a})^2}{\tilde{a}} + \frac{(b - \tilde{b})^2}{\tilde{b}} + \frac{(c - \tilde{c})^2}{\tilde{c}} + \frac{(d - \tilde{d})^2}{\tilde{d}}.$$

上の定義のもとで,

$$\begin{aligned}a - \tilde{a} &= \frac{b\Delta}{m - \Delta}, & b - \tilde{b} &= \frac{-b\Delta}{m - \Delta}, \\ c - \tilde{c} &= \frac{-d\Delta}{n + \Delta}, & d - \tilde{d} &= \frac{d\Delta}{n + \Delta}\end{aligned}$$

でかつ

$$\frac{1}{\tilde{a}} + \frac{1}{\tilde{b}} = \frac{(m - \Delta)^2}{m(a - \Delta)b}, \quad \frac{1}{\tilde{c}} + \frac{1}{\tilde{d}} = \frac{(n + \Delta)^2}{n(c + \Delta)d}$$

なので,

$$\frac{(a - \tilde{a})^2}{\tilde{a}} + \frac{(b - \tilde{b})^2}{\tilde{b}} = \frac{b\Delta^2}{m(a - \Delta)}, \quad \frac{(c - \tilde{c})^2}{\tilde{c}} + \frac{(d - \tilde{d})^2}{\tilde{d}} = \frac{d\Delta^2}{n(c + \Delta)}$$

となることより,

$$\chi^2 = \Delta^2 \left(\frac{b}{m(a - \Delta)} + \frac{d}{n(c + \Delta)} \right).$$

仮説「リスク比パラメータは $RR = \rho$ である」のP値を次のように定めることができる:

$$\text{pvalue}_{\text{Pearson}}(a, b, c, d | RR = \rho) = 1 - \text{cdf}(\text{Chisq}(1), \chi^2).$$

これに対応するリスク比パラメータ RR の信頼度 $1 - \alpha$ の信頼区間は

$$\text{confint}_{\text{Pearson}}^{\text{RR}}(a, b, c, d | \alpha) = \{\rho > 0 \mid \text{pvalue}_{\text{Pearson}}(a, b, c, d | RR = \rho) \geq \alpha\}.$$

になる.

解答終

4.7 問題: Pearsonの χ^2 検定版のリスク比に関するP値と信頼区間の計算例

データが次の場合の仮説「リスク比パラメータは $RR = 1$ である」のPearsonの χ^2 検定版のP値とリスク比パラメータ RR の95% 信頼区間を計算してみよ:

| | 商品を購入した | 購入しなかった | 合計 |
|-------|---------|---------|------|
| デザインA | 49 | 965 | 1014 |
| デザインB | 26 | 854 | 880 |
| 合計 | 75 | 1819 | 1894 |

4.7.1 Julia言語によるPearsonの χ^2 検定版のリスク比に関するP値と信頼区間の計算例

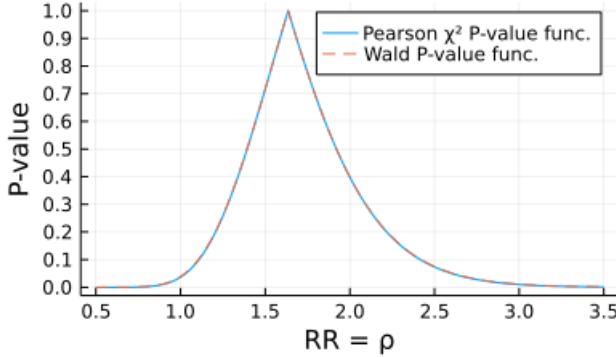
```
In [53]: 1 a, b, c, d = 49, 965, 26, 854
2 @show oddsratiohat(a, b, c, d)
3 @show riskratiohat(a, b, c, d)
4 @show pvalue_rr_wald(a, b, c, d)
5 @show pvalue_or_wald(a, b, c, d)
6 @show pvalue_rr_pearson_chisq(a, b, c, d)
7 @show pvalue_or_pearson_chisq(a, b, c, d)
8 @show confint_rr_wald(a, b, c, d)
9 @show confint_or_wald(a, b, c, d)
10 @show confint_rr_pearson_chisq(a, b, c, d)
11 @show confint_or_pearson_chisq(a, b, c, d);
```

```
oddsratiohat(a, b, c, d) = 1.6678357911518533
riskratiohat(a, b, c, d) = 1.635563647397967
pvalue_rr_wald(a, b, c, d) = 0.038895117565198624
pvalue_or_wald(a, b, c, d) = 0.03847043006362489
pvalue_rr_pearson(a, b, c, d) = 0.03661487827607109
pvalue_or_pearson(a, b, c, d) = 0.03661487827607113
confint_rr_wald(a, b, c, d) = [1.025408984412048, 2.6087819449169145]
confint_or_wald(a, b, c, d) = [1.0275181208761617, 2.7071797272783877]
confint_rr_pearson(a, b, c, d) = [1.03047149323473, 2.599845901327348]
confint_or_pearson(a, b, c, d) = [1.031778110800563, 2.6957054358850328]
```

4.7.2 Julia言語によるPearsonの χ^2 検定版のリスク比に関するP値函数の視覚化

```
In [54]: 1 a, b, c, d = 49, 965, 26, 854
2 plot(p → pvalue_rr_pearson_chisq(a,b,c,d; p), 0.5, 3.5; label="Pearson  $\chi^2$  P-value func.")
3 plot!(p → pvalue_rr_wald(a,b,c,d; p), 0.5, 3.5; label="Wald P-value func.", ls=:dash)
4 plot!(; xguide="RR =  $\rho$ ", yguide="P-value")
5 plot!(; ytick=0:0.1:1)
6 title!("a, b, c, d = $a, $b, $c, $d")
```

Out[54]: a, b, c, d = 49, 965, 26, 854



この場合には a, b, c, d が十分に大きいので、ほぼぴったり重なっている!

4.7.3 Pearsonの χ^2 検定版のリスク比に関するP値と信頼区間の計算の解答例

- (仮説「リスク比パラメータは $RR = 1$ である」のPearsonの χ^2 検定版P値) ≈ 0.03661
- (リスク比パラメータ RR のPearsonの χ^2 検定版 95% 信頼区間) $\approx [1.0305, 2.5998]$

4.8 ニューサンスパラメータの問題に関する注意

我々が想定している統計モデルには p と q という2つのパラメータが含まれている。そこに仮説 $OR = \omega$ によって制限を課しても、独立なパラメータの個数は1つしか減らず、たとえば q の側を自由に動けるパラメータとして採用し、仮説 $OR = \omega$ によって p の値は q の値から決まると考えることができる。

だから、P値を定義するために必要な「仮説 $OR = \omega$ の下での統計モデル内でデータの数値以上に極端な値が生じる確率」は、パラメータ q の値を決めるごとに別々に決まる値になってしまふ。このような状況のとき、 q はニューサンスパラメータ(nuisance parameter, 局外パラメータ、攪乱パラメータ、迷惑パラメータ)であるという。

しかし、「データの数値以上に極端な値」の意味を、Wald的な正規分布近似を使ったり、Pearsonの χ^2 統計量を使ったりして定義すれば、「仮説 $OR = \omega$ の下での統計モデル内でデータの数値以上に極端な値が生じる確率」が、 a, b, c, d が十分に大きなとき、近似的にパラメータ q の値によらなくなることを示せる。

以上で紹介したP値の定義にはこのような非常に巧妙な方法が使われている。

ニューサンスパラメータの問題への対処の仕方にはそれ以外にも、以下の節で説明する条件付き確率分布を利用する方法がある。その方法による検定は **Fisher検定** と呼ばれている。

5 Fisher検定版のオッズ比に関するP値と信頼区間

5.1 条件付き確率分布を利用するアイデア

2つの二項分布モデルの確率質量函数は

$$P(a, b, c, d|m, n, p, q) = \binom{m}{a} p^a (1-p)^{m-a} \binom{n}{c} q^c (1-q)^{n-c}$$
$$(a, b, c, d \in \mathbb{Z}_{\geq 0}, a + b = m, c + d = n).$$

と書けるのであった。この分布を条件 $a + c = r$ (そのとき $b + d = m + n - r$)という条件で制限して得られる条件付き確率分布を求めてみよう。パラメータのオッズ比 OR を

$$\text{OR} = \frac{p(1-q)}{(1-p)q}$$

と定めると、条件 $a + b = m, c + d = n, a + c = r$ の下で、

$$a = a, \quad b = m - a, \quad c = r - a, \quad d = n - r + a,$$
$$a \geq 0, \quad a \leq m, \quad a \leq r, \quad a \geq r - n$$

が成立しているので、上の確率質量函数は

$$P(a, b, c, d|m, n, p, q) = \binom{m}{a} \binom{n}{r-a} (1-p)^m q^r (1-q)^{n-r} (\text{OR})^a \propto \binom{m}{a} \binom{n}{r-a} (\text{OR})^a$$

ゆえに

$$Z(m, n, r, \text{OR}) = \sum_a \binom{m}{a} \binom{n}{r-a} (\text{OR})^a$$

とおいて、確率質量函数 $P(a|m, n, r, \text{OR})$ を

$$P(a|m, n, r, \text{OR}) = \frac{1}{Z(m, n, r, \text{OR})} \binom{m}{a} \binom{n}{r-a} (\text{OR})^a \quad (\max(0, r-n) \leq a \leq \min(m, r))$$

と定義すると、この確率質量函数で定義される確率分布が条件 $a + c = r$ で2つの二項分布モデルを制限して得られる条件付き確率分布になる。この確率分布を **Fisherの非心超幾何分布** (Fisher's noncentral hypergeometric distribution)と呼び、

$$\text{FisherNoncentralHypergeometric}(m, n, r, \text{OR})$$

と表すこととする。

このように、条件 $a + c = r$ によって条件付き確率分布に移れば、モデルのパラメータはオッズ比パラメータ OR だけになる。ニューサンスパラメータ(nuisance parameter, 局外パラメータ, 攪乱パラメータ, 迷惑パラメータ)がなくなるのである！

だから、 2×2 の分割表のデータの数値 a, b, c, d が得られたときに、 $a + b = m, c + d = n, a + c = r, b + d = m + n - r$ という縦と横方向のすべての合計(すべてのマージン)の値を固定することによって得られる条件付き確率分布(Fisherの非心超幾何分布)に移って、「データの数値以上に極端な値が生じる確率」を正確に求めることによってP値が定義できることがわかる。P値の定義の方法は二項分布モデルの場合にならねばよい。

この方法には利点と欠点がある。

- **利点:** Wald版やPearsonの χ^2 検定版と違って、確率の計算を近似計算にする必要がなくなる。それによって、第一種の過誤が起こる確率を確実に有意水準以下にできる。
- **欠点:** 有限離散分布では分布によって生じる値の個数が減ると、確率を正確に計算する方法で定義されたP値について、第一種の過誤の確率が有意水準よりも真に小さくなってしまうことが普通になる。しかもそれを条件付き確率分布に移すことによって行うと、第一種の過誤の確率が有意水準よりも大幅に小さくなってしまう場合が出て来る。

第一種の過誤の確率を確実に有意水準以下にしたい場合にはFisher検定は優れた方法になるが、検出力はその分だけ下がってしまうという欠点がある。利点と欠点を理解して、自分の目的に合わせて適切な道具を使えばよいだろう。

注意: Fisherの非心超幾何分布を $\text{OR} = 1$ に特殊化すると、超幾何分布が得られる：

$$P(a|m, n, r, 1) = \binom{m+n}{r}^{-1} \binom{m}{a} \binom{n}{r-a} \quad (\max(0, r-n) \leq a \leq \min(m, r)).$$

この確率は、赤い玉が m 個、白い玉が n 個入っている壺から、 r 個の玉を無作為に取り出すときに(非復元抽出、 r 個をまとめて一度に取り出す様子を想像すればよい)、取り出した r 個の玉の中に含まれる赤い玉の個数が a 個になる確率だと解釈される。

5.2 Sterne型のFisher検定のP値とそれに付随する信頼区間の定義

前節の記号の下で、データの数値 a, b, c, d に対して、 $m = a + b, n = c + d, r = a + c$ とおき、仮説「オッズ比パラメータは $\text{OR} = \omega$ である」のSterne型のP値を次のように定める:

$$\text{pvalue}_{\text{Sterne}}(a, b, c, d | \text{OR} = \omega) = \sum_{P(i|m, n, r, \omega) \leq P(a|m, n, r, \omega)} P(i|m, n, r, \omega).$$

これに対応する信頼度 $1 - \alpha$ の信頼区間を

$$\text{confint}_{\text{Sterne}}^{\text{OR}}(a, b, c, d | \alpha) = \{ \omega > 0 \mid \text{pvalue}_{\text{Sterne}}(a, b, c, d | \text{OR} = \omega) \geq \alpha \}$$

によって定める。もしくは、実用的には、この上限と下限に挟まれた区間として、 $\text{confint}_{\text{Sterne}}(a, b, c, d | \text{OR} = \omega)$ を定義し直したものを使用する。(このノートでの実装ではそうなっている。)

5.3 Clopper-Pearson型のFisher検定のP値とそれに付随する信頼区間

さらに続けて、Clopper-Pearson型のP値を次のように定める:

$$\text{pvalue}_{\text{CP}}(a, b, c, d | \text{OR} = \omega) = \min \left(\frac{1}{2 \operatorname{cdf}(\text{FisherNoncentralHypergeometric}(a+b, c+d, a+c, \omega), a)} \right) \left(\frac{1}{2(1 - \operatorname{cdf}(\text{FisherNoncentralHypergeometric}(a+b, c+d, a+c, \omega), a-1))} \right)$$

これに対応する信頼度 $1 - \alpha$ の信頼区間を

$$\text{confint}_{\text{CP}}^{\text{OR}}(a, b, c, d | \alpha) = \{ \omega > 0 \mid \text{pvalue}_{\text{CP}}(a, b, c, d | \text{OR} = \omega) \geq \alpha \}$$

によって定める。

5.4 問題: Fisher検定版のオッズ比に関するP値と信頼区間の計算例

以下のデータについて、Sterne型およびClopper-Pearsonの型の2種類のFisher検定のP値とそれに付随する信頼区間を計算せよ:

| | 商品を購入した | 購入しなかった | 合計 |
|-------|---------|---------|------|
| デザインA | 49 | 965 | 1014 |
| デザインB | 26 | 854 | 880 |
| 合計 | 75 | 1819 | 1894 |

ただし、検定する仮説は「オッズ比パラメータ $\text{OR} = 1$ である」であるとし(独立性に関するFisher検定の場合)、求める信頼区間は信頼度 95% のオッズ比パラメータの信頼区間であるとする。

5.4.1 Julia言語によるFisher検定版のオッズ比に関するP値と信頼区間の計算例

In [55]:

```

1 a, b, c, d = 49, 965, 26, 854
2 @show oddsratiohat(a, b, c, d)
3 @show pvalue_or_wald(a, b, c, d)
4 @show pvalue_or_pearson_chisq(a, b, c, d)
5 @show pvalue_or_sterne(a, b, c, d)
6 @show pvalue_or_clopper_pearson_chisq(a, b, c, d)
7 @show confint_or_wald(a, b, c, d)
8 @show confint_or_pearson_chisq(a, b, c, d)
9 @show confint_or_sterne(a, b, c, d)
10 @show confint_or_clopper_pearson_chisq(a, b, c, d);

```

```

oddsratiohat(a, b, c, d) = 1.6678357911518533
pvalue_or_wald(a, b, c, d) = 0.03847043006362489
pvalue_or_pearson(a, b, c, d) = 0.03661487827607113
pvalue_or_sterne(a, b, c, d) = 0.044079014645063516
pvalue_or_clopper_pearson(a, b, c, d) = 0.047080761570760604
confint_or_wald(a, b, c, d) = [1.0275181208761617, 2.7071797272783877]
confint_or_pearson(a, b, c, d) = [1.031778110800563, 2.6957054358850328]
confint_or_sterne(a, b, c, d) = [1.0202716568785046, 2.7656505208089848]
confint_or_clopper_pearson(a, b, c, d) = [1.0061031380773773, 2.8212565352058787]

```

5.4.2 Julia言語によるFisher²検定のオッズ比に関するP値函数の視覚化

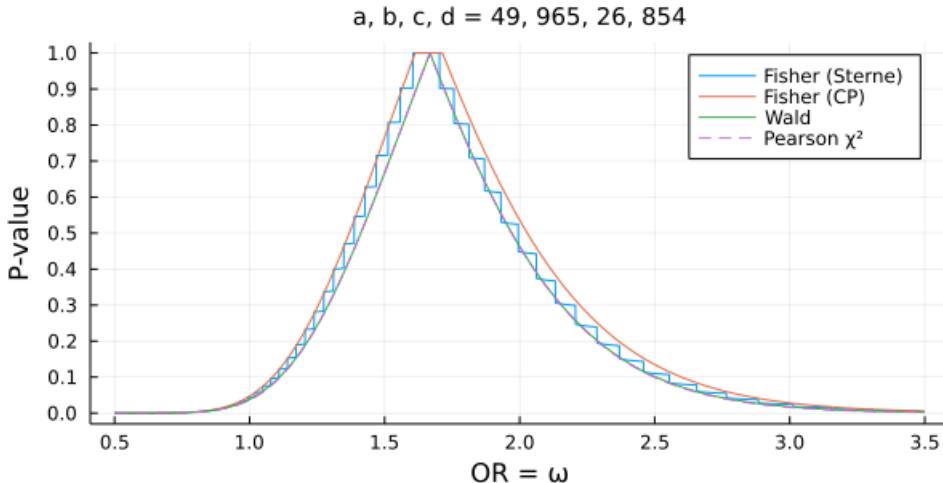
In [56]:

```

1 a, b, c, d = 49, 965, 26, 854
2 plot(w → pvalue_or_sterne(a,b,c,d; w), 0.5, 3.5; label="Fisher (Sterne)")
3 plot!(w → pvalue_or_clopper_pearson_chisq(a,b,c,d; w), 0.5, 3.5; label="Fisher (CP)")
4 plot!(w → pvalue_or_wald(a,b,c,d; w), 0.5, 3.5; label="Wald")
5 plot!(w → pvalue_or_pearson_chisq(a,b,c,d; w), 0.5, 3.5; label="Pearson  $\chi^2$ ", ls=:dash)
6 plot!(; xguide="OR = w", yguide="P-value")
7 plot!(; ytick=0:0.1:1)
8 title!("a, b, c, d = $a, $b, $c, $d")
9 plot!(size=(600, 300))

```

Out[56]:



Wald型とPearson型のP値函数はほぼぴったり一致している。

2種のFisher検定にP値函数はそれらと一致していないが、互いに相手を近似しあう関係になっている。

5.4.3 R言語によるFisher検定版のP値と信頼区間の計算の仕方

Sterne型の独立性検定の場合のP値とClopper-Pearson型のオッズ比の信頼区間をR言語の `fisher.test` フィルで求めることができます。

```

A = matrix(c(49, 965, 26, 854), 2, 2, byrow=T)
result = fisher.test(A)

```

```
In [57]: 1 R"""
2 A = matrix(c(49, 965, 26, 854), 2, 2, byrow=T)
3 result = fisher.test(A)
4 """
```

```
Out[57]: RObject{VecSxp}
```

```
Fisher's Exact Test for Count Data

data: A
p-value = 0.04408
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
1.006104 2.821251
sample estimates:
odds ratio
1.667393
```

R言語の `fisher.test` では表示されるP値と信頼区間のあいだに整合性が無くなる場合が出て来てしまう。

```
In [58]: 1 R"""
2 result = fisher.test(matrix(c(16, 4, 4, 6), 2, 2, byrow=T))
3 """
```

```
Out[58]: RObject{VecSxp}
```

```
Fisher's Exact Test for Count Data

data: matrix(c(16, 4, 4, 6), 2, 2, byrow = T)
p-value = 0.04486
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
0.8602909 44.0017191
sample estimates:
odds ratio
5.585888
```

仮説「オッズ比パラメータは 1 である」のP値が 5% を切っているのに、95% 信頼区間に 1 が含まれてしまっている。

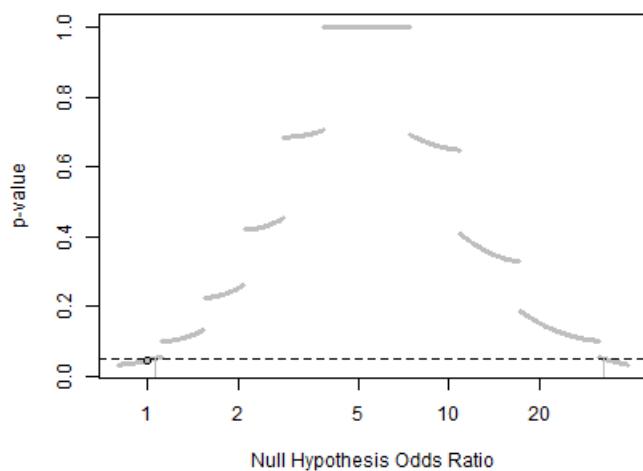
この問題は `exact2x2` パッケージを利用すれば解決する。

```
exact2x2 :: exact2x2(matrix(c(16, 4, 4, 6), 2, 2, byrow=T), plot=T)
exact2x2 :: exact2x2(matrix(c(16, 4, 4, 6), 2, 2, byrow=T), tsmethod="central", plot=T)
```

`exact2x2` フィルでは `tsmethod="minlik"` がSterne型を意味し、`tsmethod="central"` がClopper-Pearson型を意味している。

In [59]:

```
1 R"""
2 exact2x2::exact2x2(matrix(c(16, 4, 4, 6), 2, 2, byrow=T), plot=T)
3 """
```

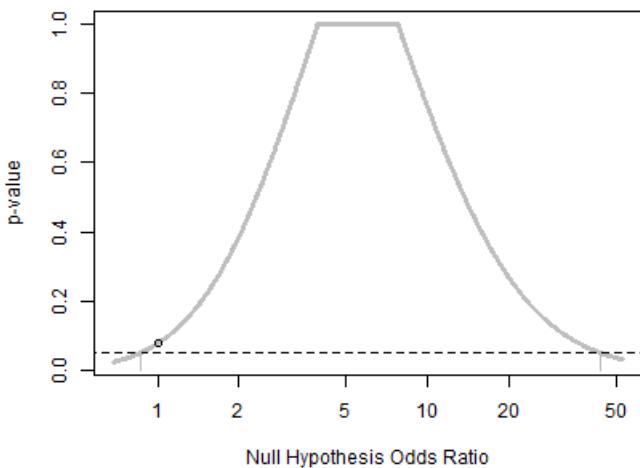


Out[59]: RObject{VecSxp}

```
Two-sided Fisher's Exact Test (usual method using minimum likelihood)

data: matrix(c(16, 4, 4, 6), 2, 2, byrow = T)
p-value = 0.04486
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.0549 32.6994
sample estimates:
odds ratio
 5.585888
```

```
In [60]: 1 R"""
2 exact2x2::exact2x2(matrix(c(16, 4, 4, 6), 2, 2, byrow=T), tsmethod="central", plot=T)
3 """
```



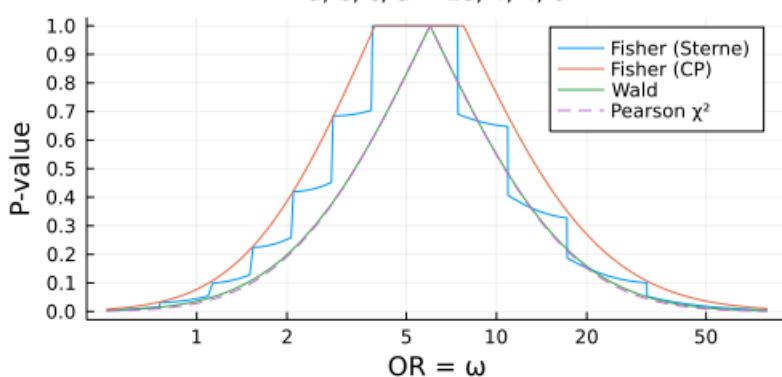
Out[60]: RObject{VecSxp}

```
Central Fisher's Exact Test

data: matrix(c(16, 4, 4, 6), 2, 2, byrow = T)
p-value = 0.07742
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
0.8602909 44.0017191
sample estimates:
odds ratio
5.585888
```

```
In [61]: 1 a, b, c, d = 16, 4, 4, 6
2 plot(w → pvalue_or_sterne(a,b,c,d; w), 0.5, 80; label="Fisher (Sterne)")
3 plot!(w → pvalue_or_clopper_pearson_chisq(a,b,c,d; w), 0.5, 80; label="Fisher (CP)")
4 plot!(w → pvalue_or_wald(a,b,c,d; w), 0.5, 80; label="Wald")
5 plot!(w → pvalue_or_pearson_chisq(a,b,c,d; w), 0.5, 80; label="Pearson χ²", ls=:dash)
6 plot!(; xguide="OR = w", yguide="P-value")
7 plot!(; ytick=0:0.1:1)
8 title!("a, b, c, d = $a, $b, $c, $d")
9 plot!(; size=(500, 250))
10 plot!(; xscale=:log, xtick=[1,2,5,10,20,50], string.([1,2,5,10,20,50])))
```

Out[61]:



5.4.4 Fisher検定版のオッズ比に関するP値と信頼区間の計算の解答例

P値

- ・(仮説「オッズ比パラメータは $OR = 1$ である」のSterne型のFisher検定版P値) $\approx 4.408\%$
- ・(仮説「オッズ比パラメータは $OR = 1$ である」のClopper-Pearson型のFisher検定版P値) $\approx 4.708\%$

95%信頼区間

- ・(オッズ比パラメータ OR の 95% Sterne型のFisher検定版の信頼区間) $\approx [1.0202, 2.7657]$
- ・(オッズ比パラメータ OR の 95% Clopper-Pearson型のFisher検定版の信頼区間) $\approx [1.006, 2.8213]$

5.5 Fisher検定における第一種の過誤の確率

「オッズ比パラメータは OR = 1 である」という仮説のP値を条件付き確率を正確に計算することによって計算することによって行われる検定を **Fisher検定** (Fisherの正確検定, Fisher's exact test) と呼ぶ。

上で説明したように, Fisher検定には少なくともSterne版とClopper-Pearson版の2種類があるが(実はさらに別の変種も考えることができるが), どれを使っても, 第一種の過誤の確率を有意水準 α 以下に抑えることができるという利点を持っている。

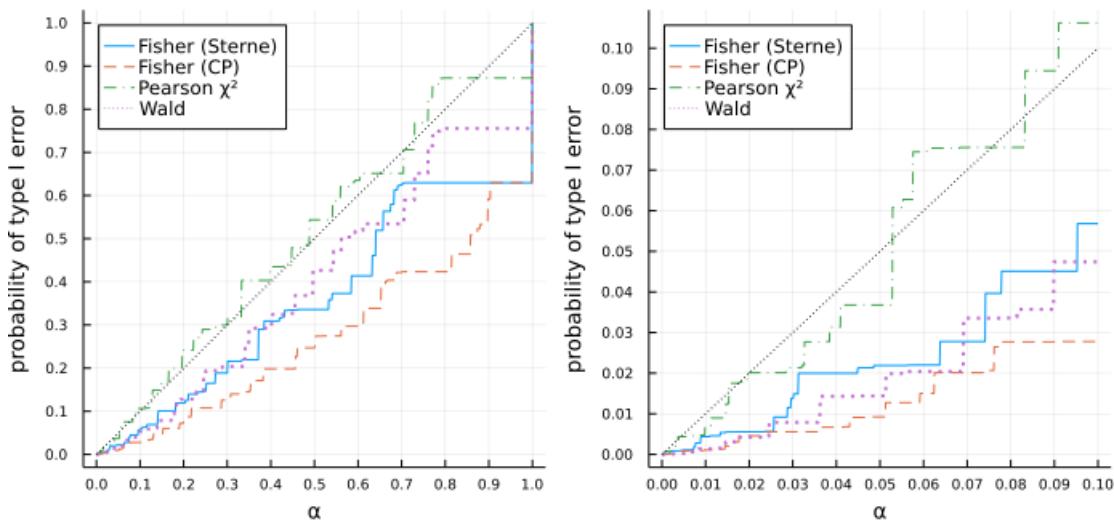
しかし, その利点は欠点の裏返しでもあり, Fisher検定での第一種の過誤の確率を有意水準 α よりかなり小さくなってしまいがちである。第一種の過誤の確率は有意水準 α に近い方がよい。

以下ではFisher検定における第一種の過誤の確率のグラフとPearsonの χ^2 検定およびWald検定での第一種の過誤の確率を重ねてプロットしてみよう。良し悪しの基準は次の2つである:

- ・第一種の過誤の確率は有意水準 α に近い方がよい。(グラフでは45度線に近い方がよい。)
- ・第一種の過誤の確率は有意水準 α 以下であって欲しい。(グラフでは45度線の上にはみ出していない方がよい。)

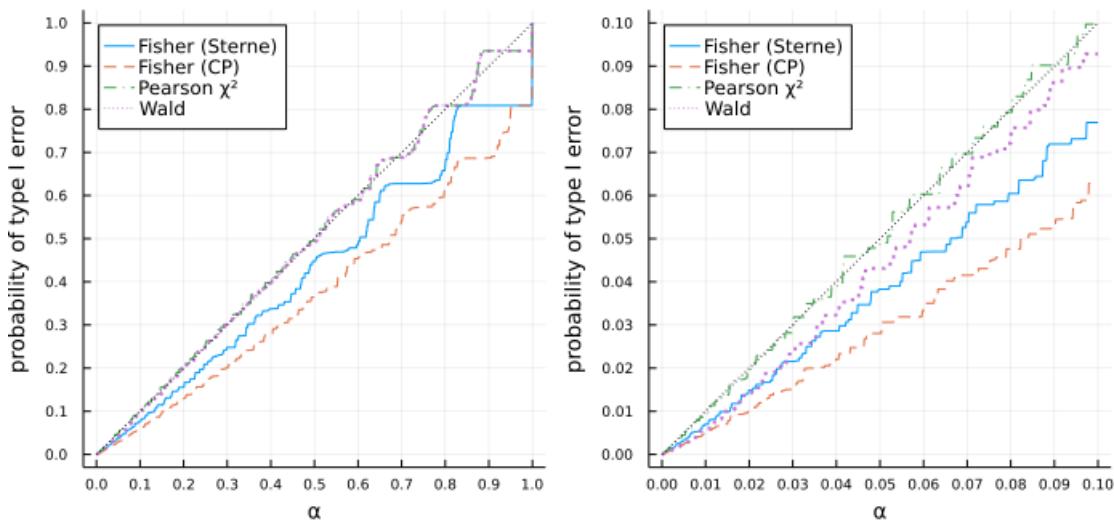
In [62]: 1 plot_probabilities_of_type_I_error(10, 20, 0.2)

Out[62]: Binomial(m=10, p=0.2)×Binomial(n=20, q=0.2)



In [63]: 1 plot_probabilities_of_type_I_error(40, 80, 0.2)

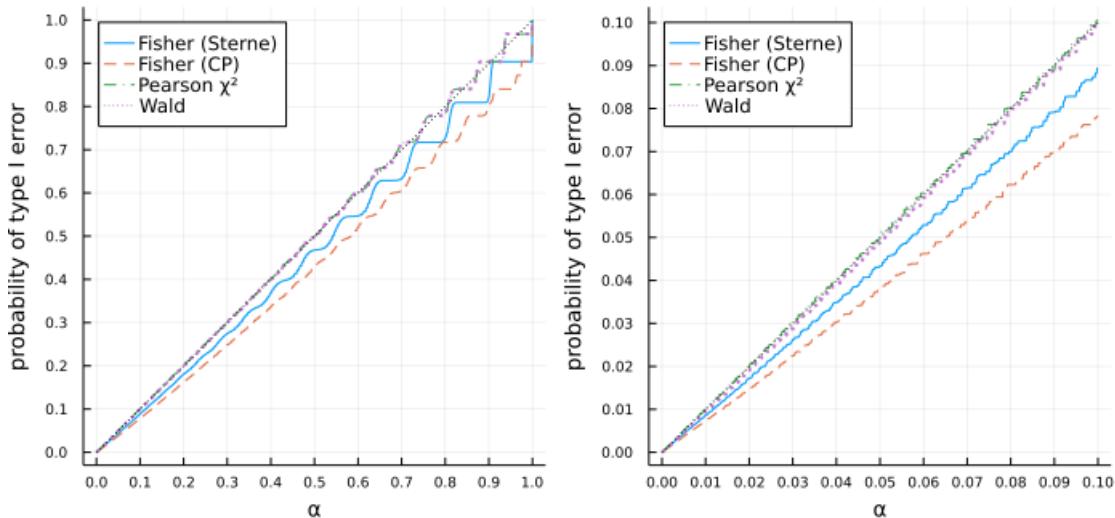
Out[63]: Binomial(m=40, p=0.2)×Binomial(n=80, q=0.2)



```
In [64]: 1 plot_probabilities_of_type_I_error(160, 320, 0.2)
```

Out[64]:

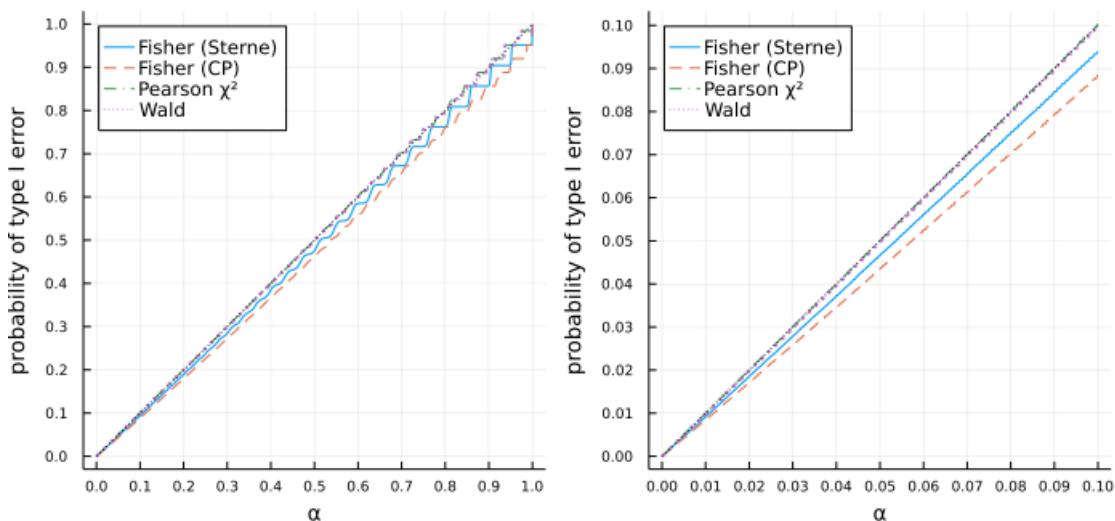
Binomial($m=160, p=0.2 \times \text{Binomial}(n=320, q=0.2)$)



```
In [65]: 1 plot_probabilities_of_type_I_error(640, 1280, 0.2)
```

Out[65]:

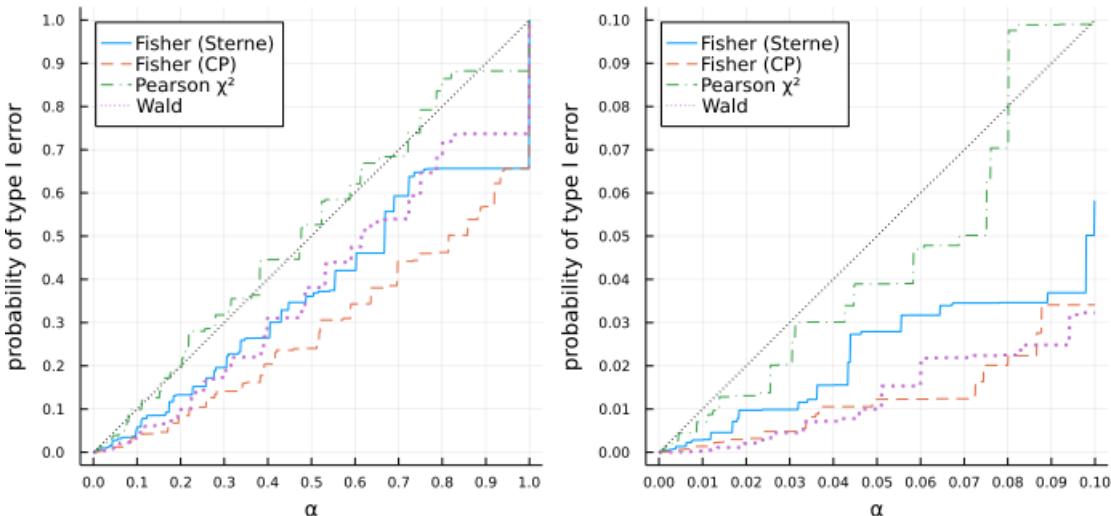
Binomial($m=640, p=0.2 \times \text{Binomial}(n=1280, q=0.2)$)



このように、Fisher検定はデータサイズを大きくしても、第一種の過誤の確率が有意水準 α になかなか近付かない。

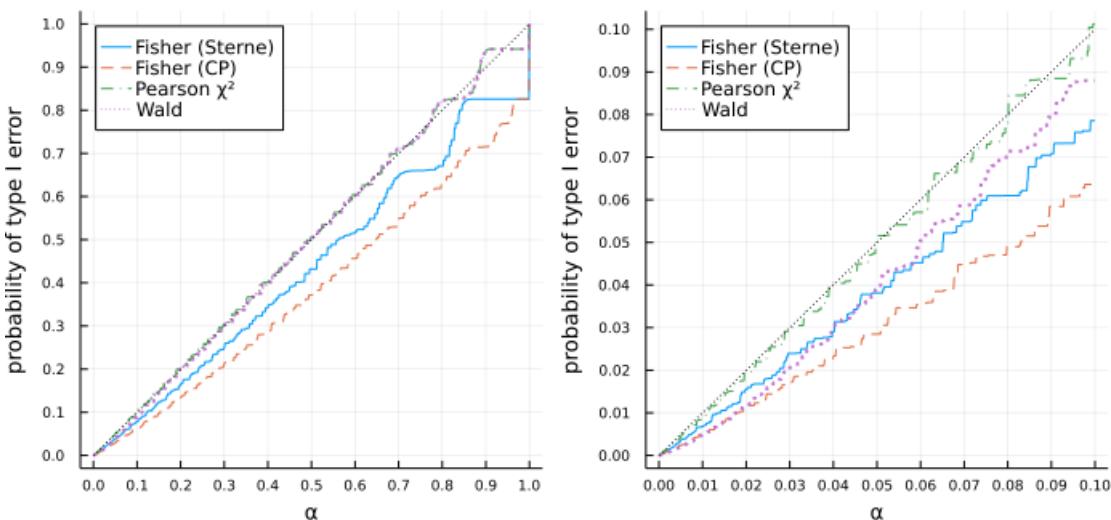
```
In [66]: 1 plot_probabilities_of_type_I_error(100, 200, 0.02)
```

Out[66]: Binomial(m=100, p=0.02)×Binomial(n=200, q=0.02)



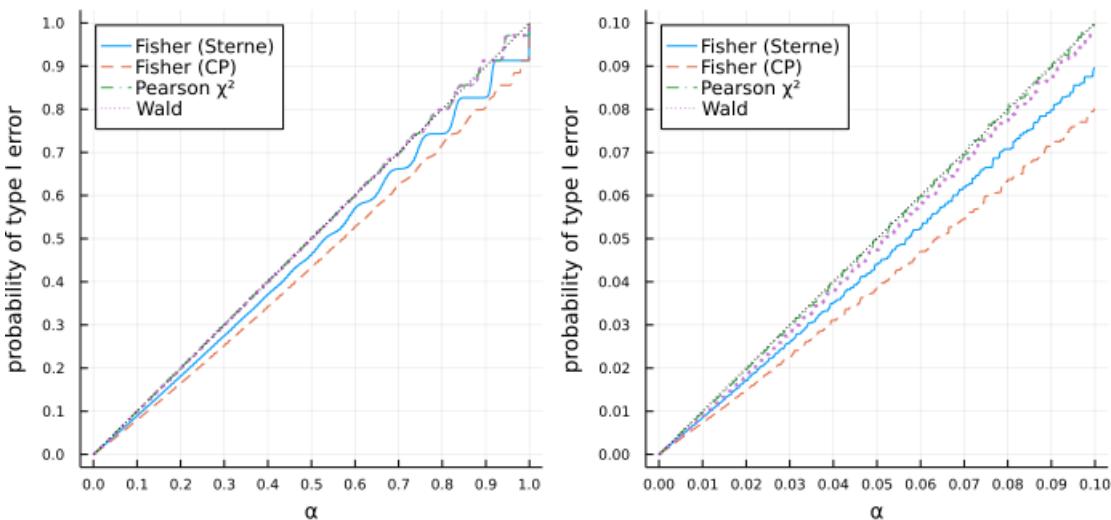
```
In [67]: 1 plot_probabilities_of_type_I_error(400, 800, 0.02)
```

Out[67]: Binomial(m=400, p=0.02)×Binomial(n=800, q=0.02)



```
In [68]: 1 plot_probabilities_of_type_I_error(1600, 3200, 0.02)
```

Out[68]: Binomial(m=1600, p=0.02)×Binomial(n=3200, q=0.02)



注意: 以上と同じことをR言語で行う場合には以下のことに注意する必要がある:

- R言語の `chisq.test` はデフォルトではYatesの連続性補正を行うので, `correct=F` として補正を無効にする必要がある.
- R言語の `fisher.test` が表示するP値はSterne版(もしくは "minlik" 版)である.
- Clopper-Pearson型のP値が必要ならば `exact2x2::exact2x2` を `method="central"` で使用すること.

6 おまけ: Bayes信用区間

6.1 オッズ比パラメータとリスク比パラメータのBayes信用区間

以下の手続きでオッズ比パラメータ OR やリスク比パラメータ RR の区間推定を行うことができる.

(1) パラメータ (p, q) の事前分布 $\text{Beta}(1, 1) \times \text{Beta}(1, 1)$ を用意する.

(2) データの数値 a, b, c, d から事後分布 $\text{Beta}(a + 1, b + 1) \times \text{Beta}(c + 1, d + 1)$ を作る.

(3) 事後分布の従う乱数を $M = 10^6$ 個生成する. その結果を $(p_i, q_i) (i = 1, 2, \dots, M)$ と表す.

(4) $i = 1, 2, \dots, M$ についてその乱数のオッズ比 $\omega_i = (p_i / (1 - p_i)) / (q_i / (1 - q_i))$ とリスク比 $\rho_i = p_i / q_i$ を計算する.

(5) $\omega_1, \omega_2, \dots, \omega_M$ の $\alpha/2$ 分位点 OR_L と $1 - \alpha/2$ 分位点 OR_U を計算する:

$$\text{OR}_L = \text{quantile}([\omega_1, \omega_2, \dots, \omega_M], \alpha/2), \quad \text{OR}_U = \text{quantile}([\omega_1, \omega_2, \dots, \omega_M], 1 - \alpha/2).$$

$\rho_1, \rho_2, \dots, \rho_M$ の $\alpha/2$ 分位点 RR_L と $1 - \alpha/2$ 分位点 RR_U を計算する:

$$\text{RR}_L = \text{quantile}([\rho_1, \rho_2, \dots, \rho_M], \alpha/2), \quad \text{RR}_U = \text{quantile}([\rho_1, \rho_2, \dots, \rho_M], 1 - \alpha/2).$$

(6) 区間 $[\text{OR}_L, \text{OR}_U]$ をオッズ比パラメータ OR の信用度 $1 - \alpha$ の 信用区間 (確信区間, credible interval) と呼び, 区間 $[\text{RR}_L, \text{RR}_U]$ をリスク比パラメータ RR の信用度 $1 - \alpha$ の 信用区間と呼ぶ:

$$\begin{aligned} \text{credint}_{\text{Bayes}}^{\text{OR}}(a, b, c, d | \alpha) &= [\text{OR}_L, \text{OR}_U], \\ \text{credint}_{\text{Bayes}}^{\text{RR}}(a, b, c, d | \alpha) &= [\text{RR}_L, \text{RR}_U]. \end{aligned}$$

これらの区間は乱数の出目によって確率的に少し揺らぐが $M = 10^6$ が大きなおかげで微小にしか揺らがない.

注意: 以上はMonte Carlo法によるP値と信頼区間の計算の仕方の説明である. 数値積分で実装することも可能である. リスク比の場合の数値積分による実装については次のノートを参照せよ:

- イベルメクチン論文の図の再現
(<https://github.com/genkuroki/public/blob/main/0029/Supplementary%20Appendix%20Figure%20S6.ipynb>).

```
In [69]: 1 function credint_or_bayes(a, b, c, d; α=0.05, M=10^6)
2     p = rand(Beta(a+1, b+2), M)
3     q = rand(Beta(c+1, d+2), M)
4     ω = @. (p*(1-q))/((1-p)*q)
5     [quantile(ω, α/2), quantile(ω, 1-α)]
6 end
7
8 function credint_rr_bayes(a, b, c, d; α=0.05, M=10^6)
9     p = rand(Beta(a+1, b+2), M)
10    q = rand(Beta(c+1, d+2), M)
11    ρ = @. p/q
12    [quantile(ρ, α/2), quantile(ρ, 1-α)]
13 end
```

Out[69]: credint_rr_bayes (generic function with 1 method)

6.2 Bayes信用区間の計算例

以下のデータについて今まで定義した信頼区間と信用区間を計算してみよう(信頼度は 95% とする).

| | 商品を購入した | 購入しなかった | 合計 |
|-------|---------|---------|------|
| デザインA | 49 | 965 | 1014 |
| デザインB | 26 | 854 | 880 |
| 合計 | 75 | 1819 | 1894 |

In [70]:

```

1 a, b, c, d = 49, 965, 26, 854
2 @show a, b, c, d
3 println()
4 @show confint_or_wald(a, b, c, d)
5 @show confint_or_pearson_chisq(a, b, c, d)
6 @show confint_or_sterne(a, b, c, d)
7 @show confint_or_clopper_pearson_chisq(a, b, c, d)
8 @show credint_or_bayes(a, b, c, d)
9 println()
10 @show confint_rr_wald(a, b, c, d)
11 @show confint_rr_pearson_chisq(a, b, c, d)
12 @show credint_rr_bayes(a, b, c, d);

```

(a, b, c, d) = (49, 965, 26, 854)

```

confint_or_wald(a, b, c, d) = [1.0275181208761617, 2.7071797272783877]
confint_or_pearson(a, b, c, d) = [1.031778110800563, 2.6957054358850328]
confint_or_sterne(a, b, c, d) = [1.0202716568785046, 2.7656505208089848]
confint_or_clopper_pearson(a, b, c, d) = [1.0061031380773773, 2.8212565352058787]
credint_or_bayes(a, b, c, d) = [1.0305933173977242, 2.486045662255798]

confint_rr_wald(a, b, c, d) = [1.025408984412048, 2.6087819449169145]
confint_rr_pearson(a, b, c, d) = [1.03047149323473, 2.599845901327348]
credint_rr_bayes(a, b, c, d) = [1.0289972191837617, 2.4014835283034417]

```

このようにどれも同じような区間になる。

6.3 Bayes信用区間に付随するP値函数

任意の区間推定から逆にP値函数を作れるのであった。

$\omega_1, \omega_2, \dots, \omega_M$ の中の ω 以下の値の割合を $\text{ecdf}([\omega_1, \omega_2, \dots, \omega_M], \omega)$ と書く:

$$\text{ecdf}([\omega_1, \omega_2, \dots, \omega_M], \omega) = (\omega_i \leq \omega \text{ となる } i \text{ の個数})/M.$$

これを $\omega_1, \omega_2, \dots, \omega_M$ の 経験累積分布函数 (empirical cumulative distribution function) と呼ぶ。

(7) 仮説「オッズ比パラメータは $OR = \omega$ である」の Bayes版P値 (Bayesian P-value) を

$$\text{pvalue}_{\text{Bayes}}(a, b, c, d | OR = \omega) = \min \left(\frac{1}{2 \text{ecdf}([\omega_1, \omega_2, \dots, \omega_M], \omega)}, \frac{1}{2(1 - \text{ecdf}([\omega_1, \omega_2, \dots, \omega_M], \omega))} \right)$$

と定める。同様に、仮説「リスク比パラメータは $RR = \rho$ である」の Bayes版P値を

$$\text{pvalue}_{\text{Bayes}}(a, b, c, d | RR = \rho) = \min \left(\frac{1}{2 \text{ecdf}([\rho_1, \rho_2, \dots, \rho_M], \rho)}, \frac{1}{2(1 - \text{ecdf}([\rho_1, \rho_2, \dots, \rho_M], \rho))} \right)$$

と定める。これらのP値達は乱数の出目によって確率的に少し揺らぐが $M = 10^6$ が大きなおかげで微小にしか揺らがない。

注意: これは乱数生成による Monte Carlo 法によるP値の計算法である。数値積分でP値を計算するようにもできる。以下のノートではリスク比のP値の場合に実際にそれを行っている:

- イベルメクチン論文の図の再現

(<https://github.com/genkuroki/public/blob/main/0029/Supplementary%20Appendix%20Figure%20S6.ipynb>).

In [71]:

```

1 """Bayes版P値函数達を作る函数"""
2 function make_pvalue_or_rr_bayes(a, b, c, d; M=10^6)
3     p = rand(Beta(a+1, b+2), M)
4     q = rand(Beta(c+1, d+2), M)
5     w = @. (p*(1-q))/((1-p)*q)
6     ρ = @. p/q
7     ecdf_OR = ecdf(w)
8     ecdf_RR = ecdf(ρ)
9     pvalue_or_bayes(w) = min(1, 2ecdf_OR(w), 2(1-ecdf_OR(w)))
10    pvalue_rr_bayes(ρ) = min(1, 2ecdf_RR(ρ), 2(1-ecdf_RR(ρ)))
11    pvalue_or_bayes, pvalue_rr_bayes
12 end
13
14 function plot_pvalue_functions(a, b, c, d;
15     xlim=Tuple(confint_or_pearson_chisq(a, b, c, d; α=1e-3)), kwargs...)
16     pvalue_or_bayes, pvalue_rr_bayes = make_pvalue_or_rr_bayes(a, b, c, d)
17
18     P = plot()
19     plot!(w → pvalue_or_bayes(w), xlim...; label="Bayesian")
20     plot!(w → pvalue_or_wald(a,b,c,d; w), xlim...; label="Wald", ls=:dashdot)
21     plot!(w → pvalue_or_pearson_chisq(a,b,c,d; w), xlim...; label="Pearson  $\chi^2$ ", ls=:dash)
22     #plot!(w → pvalue_or_sterne(a,b,c,d; w), xlim...; label="Fisher (Sterne)")
23     #plot!(w → pvalue_or_clopper_pearson_chisq(a,b,c,d; w), xlim...; label="Fisher (CP)")
24     plot!(; xguide="OR = w", yguide="P-value")
25     plot!(; ytick=0:0.1:1)
26     title!("a, b, c, d = $a, $b, $c, $d")
27
28     Q = plot()
29     plot!(ρ → pvalue_rr_bayes(ρ), xlim...; label="Bayesian")
30     plot!(ρ → pvalue_rr_wald(a,b,c,d; ρ), xlim...; label="Wald", ls=:dashdot)
31     plot!(ρ → pvalue_rr_pearson_chisq(a,b,c,d; ρ), xlim...; label="Pearson  $\chi^2$ ", ls=:dash)
32     plot!(; xguide="RR = ρ", yguide="P-value")
33     plot!(; ytick=0:0.1:1)
34     title!("a, b, c, d = $a, $b, $c, $d")
35
36     plot(P, Q; size=(800, 250),
37           leftmargin=4Plots.mm, bottommargin=4Plots.mm)
38     plot!(; kwargs...)
39 end

```

Out[71]: plot_pvalue_functions (generic function with 1 method)

6.4 Bayes版P値函数の視覚化

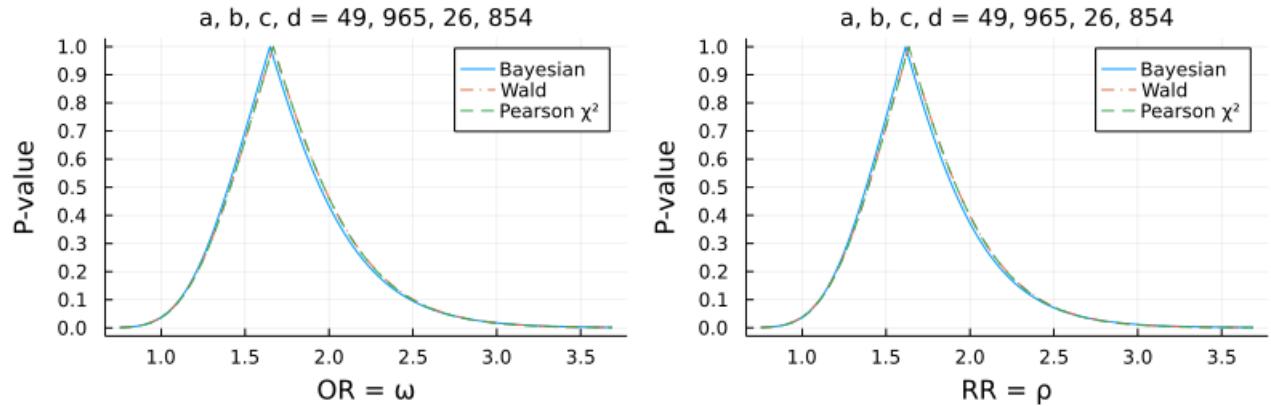
In [72]:

```

1 a, b, c, d = 49, 965, 26, 854
2 plot_pvalue_functions(a, b, c, d)

```

Out[72]:



このように、Bayes版P値函数は通常のWald版およびPearsonの χ^2 検定版のP値函数によく一致している。

6.5 Bayes版P値函数の視覚化の別の例

| | 症状悪化 | 悪化せず | 合計 |
|-------|------|------|------|
| 薬Iを処方 | 100 | 579 | 679 |
| プラセボ | 111 | 568 | 679 |
| 合計 | 211 | 1147 | 1358 |

この表は次の論文

- Gilmar Reis, M.D., Ph.D. et al., Effect of Early Treatment with Ivermectin among Patients with Covid-19, N Engl J Med 2022; 386:1721-1731. [[link](https://www.nejm.org/doi/full/10.1056/NEJMoa2115869) (<https://www.google.com/search?q=Intention-to-treat>)]

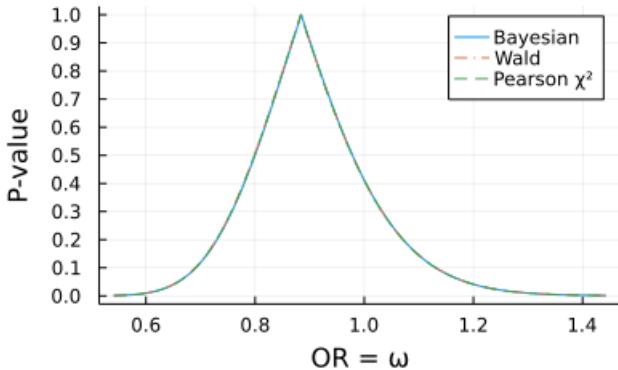
の Table 2 から [Intention-to-treat population](https://www.google.com/search?q=Intention-to-treat) (<https://www.google.com/search?q=Intention-to-treat>) の部分だけを抜き出したものである。「症状悪化」の正確な意味などに関する詳しい説明は論文を参照せよ。(このデータは必修問題において既出である。)

この論文ではリスク比の区間推定方法としてちょうど上で説明したリスク比のBayes信用区間を採用している。

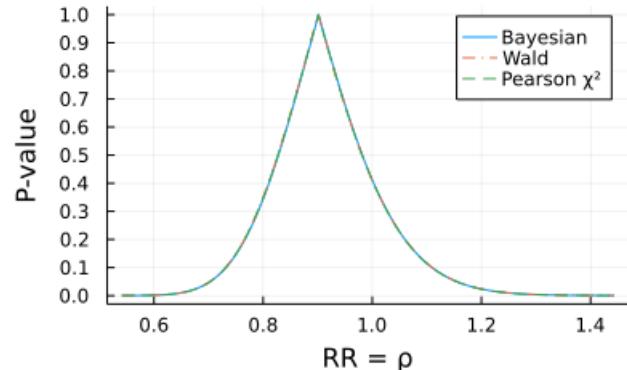
上の表のデータについてP値函数のグラフを描いてみよう。

```
In [73]: 1 a, b, c, d = 100, 579, 111, 568  
         2 plot_pvalue_functions(a, b, c, d)
```

Out[73]: a, b, c, d = 100, 579, 111, 568



a, b, c, d = 100, 579, 111, 568



論文ではBayes統計の方法を使って区間推定しているが、P値函数達に関するこのグラフを見れば分かるように、この場合には、Bayes統計版ではないWald版やPearsonの χ^2 検定版の信頼区間を使っても同じ結果が得られる。

この一致は偶然ではない。十分に a, b, c, d が大きいがゆえに正規分布近似が有効になっているので一致している。

巷に溢れる「ベイズ統計入門」では「ベイズ統計学は通常の統計学とは異なる主義・思想・哲学に基く本質的に異なる統計学である」というような感じの解説が目立つ。そのような解説に騙されないように注意することが必要である。

どんなに主義・思想・哲学が違っていても、実践的には無意味な小さな違いしかない量(実践的には同じ量とみなされる)を計算しているのであれば、主義・思想・哲学の違いは実践的には無視できるということを意味している。

7 関連講義動画

このノートの内容に関連したより先のより進んだ話題については

- 聴講コース 臨床研究者のための生物統計学 (<https://ocw.kyoto-u.ac.jp/course/328/>)

にある以下の3つの講義動画は視聴せよ:

- 佐藤俊哉, ランダム化ができないとき, 2018/10/25, 長さ 1:02:31) [[link](https://youtu.be/tUkyjZXU1vc) (<https://www.google.com/search?q=Intention-to-treat>)]
- 佐藤俊哉, 交絡とその調整, 2018/12/20, 長さ 1:00:52) [[link](https://youtu.be/ydkQFEdCPM) (<https://www.google.com/search?q=Intention-to-treat>)]
- 佐藤俊哉, 回帰モデルと傾向スコア, 2019/02/21, 長さ 1:04:44) [[link](https://youtu.be/cOHN444kBlo) (<https://www.google.com/search?q=Intention-to-treat>)]

このノートで解説した2×2の分割表の統計学の続きとして非常に楽しめるように思われる。

```
In [ ]: 1
```