

# 検定と信頼区間: 比率の比較

- 黒木玄
- 2022-06-14~2022-06-14

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

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

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

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

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

## 目次

- ▼ [1 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 OR とリスク比 RR](#)
  - [1.4 Wald版のオッズ比に関するP値と信頼区間の定義](#)
- ▼ [1.5 Wald版のオッズ比に関するP値と信頼区間の計算例](#)
  - [1.5.1 WolframAlphaによるWald版のP値と信頼区間の計算の仕方](#)
  - [1.5.2 Julia言語によるWald版のP値と信頼区間の計算の仕方\(1\)](#)
  - [1.5.3 Julia言語によるWald版のP値と信頼区間の計算の仕方\(2\)](#)
  - [1.5.4 R言語での計算の仕方](#)
- [1.6 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](#)
- ▼ [1.7 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](#)
  - [1.7.1 WolframAlphaによるPearsonの \$\chi^2\$ 検定のP値の計算の仕方](#)
  - [1.7.2 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\(1\)](#)
  - [1.7.3 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\(2\)](#)
  - [1.7.4 R言語によるPearsonの \$\chi^2\$ 検定のP値の計算の仕方](#)
- [1.8 ニューサンスパラメータの問題に関する注意](#)

```
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}{r} * repeat("r",sz[2]) * "}" * a * "\end{array}\right]"
19     end
20     print(io, as_markdown(toeqnarray(x)))
21 end

```

```

In [3]: 1 safemul(x, y) = x == 0 ? x : x*y
2 safediv(x, y) = x == 0 ? x : 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 confint_or_wald(a, b, c, d;  $\alpha=0.05$ )
5     z = quantile(Normal(), 1- $\alpha/2$ )
6     ORhat = oddsratiohat(a, b, c, d)
7     SElogORhat = stderr_logoddsratiohat(a, b, c, d)
8     [exp(-z*SElogORhat)*ORhat, exp(z*SElogORhat)*ORhat]
9 end
10
11 function pvalue_or_wald(a, b, c, d,  $\omega=1$ )
12     logORhat = log(oddsratiohat(a, b, c, d))
13     SElogORhat = stderr_logoddsratiohat(a, b, c, d)
14     2ccdf(Normal(0, 1), abs(log( $\omega$ ) - logORhat)/SElogORhat)
15 end
16
17 function delta(a, b, c, d,  $\omega=1$ )
18     A, B, C = 1- $\omega$ , a+d* $\omega$ *(b+c), a*d- $\omega$ *b*c
19     isinf( $\omega$ ) ? typeof( $\omega$ )(-min(b, c)) : safediv(2C, B +  $\sqrt{B^2 - 4A*C}$ )
20 end
21
22 # correction = 0.5 は連続性補正を与える.
23 function chisqstat(a, b, c, d,  $\omega=1$ ; correction=0.0)
24      $\delta$  = delta(a, b, c, d,  $\omega$ )
25      $\hat{a}$ ,  $\hat{b}$ ,  $\hat{c}$ ,  $\hat{d}$  = a- $\delta$ , b+ $\delta$ , c+ $\delta$ , d- $\delta$ 
26     safemul(max(0, abs( $\delta$ )-correction)^2, 1/ $\hat{a}+1/\hat{b}+1/\hat{c}+1/\hat{d}$ )
27 end
28
29 function pvalue_or_pearson(a, b, c, d,  $\omega=1$ ; correction=0.0)
30      $\chi^2$  = chisqstat(a, b, c, d,  $\omega$ ; correction)
31     ccdf(Chisq(1),  $\chi^2$ )
32 end
33
34 function confint_or_pearson(a, b, c, d;  $\alpha=0.05$ , correction=0.0)
35      $\omega_L$ ,  $\omega_U$  = confint_or_wald(a, b, c, d;  $\alpha$ )
36     f( $\omega$ ) = logit(pvalue_or_pearson(a, b, c, d,  $\omega$ ; correction)) - logit( $\alpha$ )
37     [find_zero(f,  $\omega_L$ ), find_zero(f,  $\omega_U$ )]
38 end

```

Out[4]: confint\_or\_pearson (generic function with 1 method)

```

In [5]: 1 a, b, c, d = 49, 965, 26, 854

```

Out[5]: (49, 965, 26, 854)

```

In [6]: 1 oddsratiohat(a, b, c, d)

```

Out[6]: 1.6678357911518533

```

In [7]: 1 [
2     pvalue_or_wald(a, b, c, d, 1.1)
3     pvalue_or_pearson(a, b, c, d, 1.1)
4     pvalue_or_pearson(a, b, c, d, 1.1; correction=0.5)
5 ]

```

Out[7]: 3-element Vector{Float64}:  
0.09215260921137496  
0.09006489933310749  
0.1149227968821219

```

In [8]: 1 [
2     confint_or_wald(a, b, c, d),
3     confint_or_pearson(a, b, c, d),
4     confint_or_pearson(a, b, c, d; correction=0.5)
5 ]

```

Out[8]: 3-element Vector{Vector{Float64}}:  
[1.0275181208761617, 2.7071797272783877]  
[1.0317781108000563, 2.6957054358850328]  
[1.002870215221586, 2.7855253257917676]

```
In [9]: 1 # cf. Cornfeild (1956), p.139, (4.2)
2 let (a, b, c, d) = (3, 11, 60, 32)
3 [
4     confint_or_wald(a, b, c, d),
5     confint_or_pearson(a, b, c, d),
6     confint_or_pearson(a, b, c, d; correction=0.5)
7 ]
8 end
```

```
Out[9]: 3-element Vector{Vector{Float64}}:
 [0.037829803607039435, 0.5592686923030045]
 [0.04084291410483445, 0.5253472140108638]
 [0.02962955731952912, 0.6228119542455193]
```

```
In [10]: 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[10]: 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

```
In [11]: 1 @rget result
```

```
Out[11]: OrderedCollections.OrderedDict{Symbol, Any} with 10 entries:
 :method      => "cohort.count"
 :n_strata    => 1
 :digits      => 4.0
 :conf_level  => 0.95
 :interp      => false
 :units       => ["Outcomes per 100 population units", "per 100 population units"]
 :tab         => 3x5 DataFrame...
 :massoc_summary => 6x4 DataFrame...
 :massoc_interp => 7x2 DataFrame...
 :massoc_detail => OrderedCollections.OrderedDict{Symbol, Any}(:RR_strata_wald=>1x3 DataFrame
```

```
In [12]: 1 md = result[:massoc_detail]
```

```
Out[12]: OrderedCollections.OrderedDict{Symbol, Any} with 19 entries:
:RR_strata_wald      => 1x3 DataFrame...
:RR_strata_taylor    => 1x3 DataFrame...
:RR_strata_score     => 1x3 DataFrame...
:OR_strata_wald      => 1x3 DataFrame...
:OR_strata_cfield    => 1x3 DataFrame...
:OR_strata_score     => 1x3 DataFrame...
:OR_strata_mle       => 1x3 DataFrame...
:ARisk_strata_wald   => 1x3 DataFrame...
:ARisk_strata_score  => 1x3 DataFrame...
:NNT_strata_wald     => 1x3 DataFrame...
:NNT_strata_score    => 1x3 DataFrame...
:AFRisk_strata_wald  => 1x3 DataFrame...
:PARisk_strata_wald  => 1x3 DataFrame...
:PARisk_strata_piri  => 1x3 DataFrame...
:PAFRisk_strata_wald => 1x3 DataFrame...
:chi2_strata_uncor   => 1x5 DataFrame...
:chi2_strata_yates   => 1x4 DataFrame...
:chi2_strata_fisher  => 1x4 DataFrame...
:chi2_correction     => false
```

```
In [13]: 1 ms = result[:massoc_summary]
```

```
Out[13]: 6 rows × 4 columns
```

	var	est	lower	upper
	String	Float64	Float64	Float64
1	Inc risk ratio	1.63556	1.02541	2.60878
2	Odds ratio	1.66784	1.02752	2.70718
3	Attrib risk *	1.8778	0.147521	3.60808
4	Attrib fraction in exposed (%)	38.859	2.47794	61.6679
5	Attrib risk in population *	1.00533	-0.41699	2.42765
6	Attrib fraction in population (%)	25.3879	-1.2434	45.014

## 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) 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快復者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偶然に、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。)

もしもそうならば、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切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値関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  $2 \times 2$  の分割表型のデータを扱う:

	$Y = 1$	$Y = 0$	
$X = 1$	$a$	$b$	$m = a + b$
$X = 0$	$c$	$d$	$n = c + d$
	$s = a + c$	$t = b + d$	$N = a + b + c + d$

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ような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,  $a, c$  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,  $a, b, c, d$  の同時確率質量関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.

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

$X, Y$  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,  $Y = 1, 0$  は快復したか否かを意味する.

**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こと.

**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%20estimation%20of%20the%202x2%20contingency%20table%20distribution) (<https://nbviewer.org/github/genkuroki/Statistics/blob/master/2022/06%20Conditional%20distribution%2C%20likelihood%2C%20estimation%20of%20the%202x2%20contingency%20table%20distribution>) 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 OR とリスク比 RR

比率  $0 < p < 1$  に関する

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

を **オッズ(odds)**と呼び, 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、これの対数  $\beta$  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、これの対数  $\gamma$  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、それらの対数版である  $\log \text{OR} = 0$  や  $\log \text{RR} = 0$  を採用してもよい.)

そして、オッズ比 OR とリスク比 RR が 1 からどれだけどのように離れているかは、2つの比率  $p, q$  の違いを表す指標として使える。(もちろん、それらの対数版である  $\log \text{OR}$  や  $\log \text{RR}$  が 0 からどれだけどのように離れているかを指標として採用してもよい.)

**注意:** 他にも **比率の差**  $p - q$  も重要な指標だが、このノートでは面倒だという理由で扱わないことにする。

**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

$$\text{勝つ確率} : \text{負ける確率} = 1 : 4$$

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:** 例えば確率  $p, q$  が「死亡確率」ならば「リスク」という呼び方は適切になる。ここではそうでない場合も  $p, q$  を「リスク」と呼んでしまい、 $p/q$  を「リスク比」と呼ぶことにする。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:** このノートでは主にオッズ比を比率の違いを表す指標として利用する。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することができる。

**注意:** オッズ  $u = p/(1-p)$  の対数を  $x$  と書き、**対数オッズ**(log odds)と呼ぶ。このとき、以下の公式によって  $0 < p < 1$ ,  $0 < u < \infty, -\infty < x < \infty$  が一対一に対応する:

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

$x$  を  $p$  に対応させる関数は **ロジスティック関数** と呼ばれ、その逆関数は **ロジット関数** と呼ばれる。

**注意:** 対数オッズ比を  $\beta = \log \text{OR}$  と書き、さらに  $q$  の対数を  $\alpha = \log q$  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:

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

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  $\alpha, \beta$  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.

## 1.4 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.

**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**

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

の対数  $\log \widehat{OR}$  (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

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

で, 分散が

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

であるような正規分布に近似的に従うことを示せる. ゆえに,

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

(「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.)

**P値の構成法:** 上の近似を使えば, 「オッズ比は  $OR = \omega$  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:

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

ただし,  $\log \widehat{OR}$  と  $\widehat{SE}$  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.

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

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

対応するオッズ比  $OR$  の信頼区間は次のようになる:

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

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

## 1.5 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Wald版の仮説「オッズ比は  $OR = 1$  である」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:

- (Wald版の仮説「オッズ比は  $OR = 1$  である」のP値)  $\approx 3.847\%$
- (Wald版のオッズ比  $OR$  の 95% 信頼区間)  $\approx [1.0275, 2.7072]$



### 1.5.1 WolframAlphaによるWald版のP値と信頼区間の計算の仕方

$\sqrt{1/a+1/b+1/c+1/d}$  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%29>) → 0.247137

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

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

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

$\{\exp(-0.247137z)*a*d/(b*c), \exp(0.247137z)*a*d/(b*c)\}$  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%2F%28b\\*c%29%2C+exp%280.247137z%29\\*a\\*d%2F%28b\\*c%29%29%7D+where+a%3D49.0%2C+b%3D965%2C+d%3D854%2C+z%3D1.95996%29](https://www.wolframalpha.com/input?i=%7Bexp%28-0.247137z%29*a*d%2F%28b*c%29%2C+exp%280.247137z%29*a*d%2F%28b*c%29%29%7D+where+a%3D49.0%2C+b%3D965%2C+d%3D854%2C+z%3D1.95996%29)) → {1.02752, 2.70717} (95%信頼区間)

### 1.5.2 Julia言語によるWald版の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 [14]:

```
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 = 2ccdf(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, 2.7071797272783877]
```

### 1.5.3 Julia言語によるWald版の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関数を使って計算。

In [15]:

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

### 1.5.4 R言語での計算の仕方

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 [16]:

```

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 [17]:

```
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[17]:

```
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

## 1.6 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

天下り的になってしまい非常に申し訳ないのだが、オッズ比の値に関する仮説のP値のPearsonの $\chi^2$ 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。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、それだけだとどうと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 $\chi^2$ 検定** で使うP値になっている。

**P値の構成法:** オッズ比の値に関する仮説のP値のPearsonの $\chi^2$ 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  $\delta$  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、次のように書き直される:

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

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

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

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、 $\delta$  を決める方程式は  $-(a + b + c + d)\delta + ad - bc = 0$  になるので、

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

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

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

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:

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

(3) データの数値に対応するPearsonの $\chi^2$ 統計量  $\chi^2$  の値を次のように定める:

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

この  $\chi^2$  は  $a, b, c, d, \omega$  のみを使って計算できる数値になっている。

上の  $\chi^2$  に関する前者の式はPearsonの $\chi^2$ 統計量の一般的な表式

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

の特別な場合になっていることに注意せよ。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 $\chi^2$ 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 $\chi^2$ 統計量の値がデータから計算した  $\chi^2$  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  $\chi^2$  は  $a, b, c, d$  が十分に大きくなるとき(実はそう小さくなくてもよい)、近似的に自由度 1 の $\chi^2$  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:

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

仮説  $OR = \omega$  に関するデータの数値  $a, b, c, d$  に関するPearsonの $\chi^2$ 統計量の値  $\chi^2$  は  $a, b, c, d, \omega$  だけで計算される値なので、これでP値がうまく定義されている。

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

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

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。このノートでは、 $\text{pvalue}_{\text{Pearson}}(a, b, c, d | OR = \omega) = \alpha$  を満たす  $\omega$  を数値的に求めることによって信頼区間を計算している。

**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)

## 1.7 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

データが次の場合のPearsonの $\chi^2$ 検定版仮説「オッズ比は  $OR = 1$  である」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:

- (Pearsonの $\chi^2$ 検定版の仮説「オッズ比は  $OR = 1$  である」のP値)  $\approx 3.661\%$
- (Pearsonの $\chi^2$ 検定版のオッズ比  $OR$  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ことはかなり面倒である.

### 1.7.1 WolframAlphaによるPearsonの $\chi^2$ 検定のP値の計算の仕方

$\omega = 1$  の場合には, Pearsonの $\chi^2$ 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  $a=49.0, b=965, c=26, d=854 \rightarrow$  [実行 \(https://www.wolframalpha.com/input?i=%28a%2Bb%2Bc%2Bd%29%28ad-bc%29%5E2%2F%28%28a%2Bb%29%28c%2Bd%29%28a%2Bc%29%28b%2Bd%29%29+where+a%3D49.0%2C+b%3D965%2C+c%3D26%2C+d%3D854%29\)](https://www.wolframalpha.com/input?i=%28a%2Bb%2Bc%2Bd%29%28ad-bc%29%5E2%2F%28%28a%2Bb%29%28c%2Bd%29%28a%2Bc%29%28b%2Bd%29%29+where+a%3D49.0%2C+b%3D965%2C+c%3D26%2C+d%3D854%29)  
 $\rightarrow 4.36824$  (独立性に関するPearsonの $\chi^2$ 統計量)

$1 - \text{cdf}(\text{ChisqDistribution}(1), 4.36824) \rightarrow$  [実行 \(https://www.wolframalpha.com/input?i=1-cdf%28ChisqDistribution%281%29%2C+4.36824%29\)](https://www.wolframalpha.com/input?i=1-cdf%28ChisqDistribution%281%29%2C+4.36824%29)  $\rightarrow 0.0366148$  (P値)

### 1.7.2 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)

$\omega = 1$  の場合のPearsonの $\chi^2$ 検定版のオッズ比に関するP値は, 独立性に関するPearsonの $\chi^2$ 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.

信頼区間の側は関数の零点を見つけてくれる関数を使って求めてみよう.

```
In [23]: 1 a, b, c, d = 49, 965, 26, 854
2 @show a, b, c, d
3  $\chi^2 = (a+b+c+d)*(a*d-b*c)^2 / ((a+b)*(c+d)*(a+c)*(b+d))$ 
4 @show  $\chi^2$ 
5 @show pvalue = ccdf(Chisq(1),  $\chi^2$ )
6  $\alpha = 0.05$ 
7 @show  $\alpha$ 
8 confint = find_zeros(0.5, 3.0) do  $\omega$ 
9     A, B, C = 1- $\omega$ , a+d+ $\omega*(b+c)$ , a*d- $\omega*b*c$ 
10      $\delta = 2C / (B + \sqrt{B^2 - 4A*C})$ 
11      $\chi^2 = \delta^2 * (1/(a-\delta) + 1/(b+\delta) + 1/(c+\delta) + 1/(d-\delta))$ 
12     ccdf(Chisq(1),  $\chi^2$ ) -  $\alpha$ 
13 end
14 @show confint;
```

```
(a, b, c, d) = (49, 965, 26, 854)
 $\chi^2 = 4.368235947209686$ 
pvalue = ccdf(Chisq(1),  $\chi^2$ ) = 0.03661487827607113
 $\alpha = 0.05$ 
confint = [1.031778110800563, 2.6957054358850328]
```

$f(x, y)$  do  $z \dots$  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 関数の使い方については,

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

を参照せよ.

### 1.7.3 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)

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 [24]: 1 a, b, c, d = 49, 965, 26, 854
2 @show a, b, c, d
3 @show chisqstat(a, b, c, d, 1)
4 @show pvalue_or_pearson(a, b, c, d, 1)
5 @show confint_or_pearson(a, b, c, d;  $\alpha=0.05$ );

(a, b, c, d) = (49, 965, 26, 854)
chisqstat(a, b, c, d, 1) = 4.368235947209686
pvalue_or_pearson(a, b, c, d, 1) = 0.03661487827607113
confint_or_pearson(a, b, c, d;  $\alpha = 0.05$ ) = [1.031778110800563, 2.6957054358850328]
```

### 1.7.4 R言語によるPearsonの $\chi^2$ 検定のP値の計算の仕方

独立性に関するPearsonの $\chi^2$ 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 [29]: 1 R""
2 A = matrix(c(49, 965, 26, 854), 2, 2, byrow=T)
3 result = chisq.test(A, correct=F)
4 ""
```

```
Out[29]: RObject{VecSxp}

      Pearson's Chi-squared test

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

## 1.8 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う. このような状況のとき,  $q$  はニューサンスパラメータ(nuisance parameters, 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 $\chi^2$ 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ら以外にも, 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.

```
In [ ]: 1
```