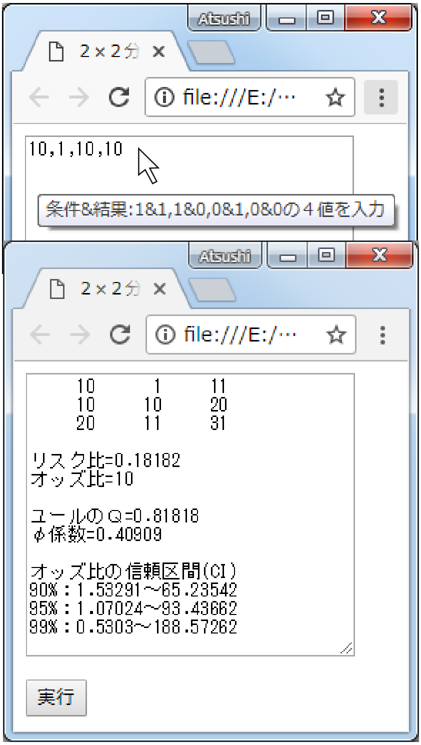
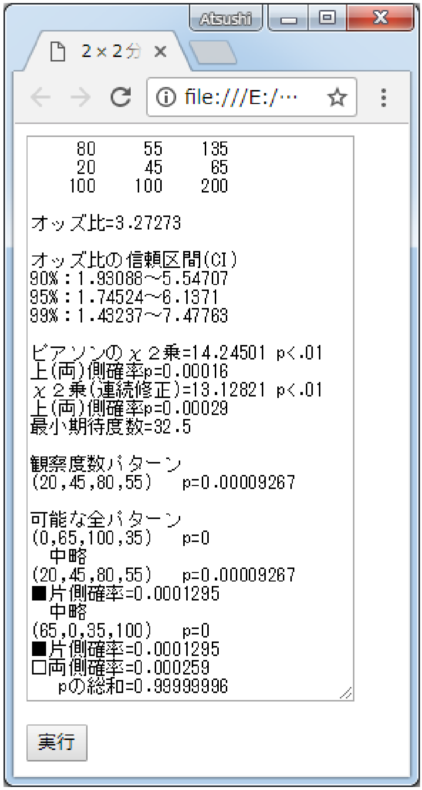
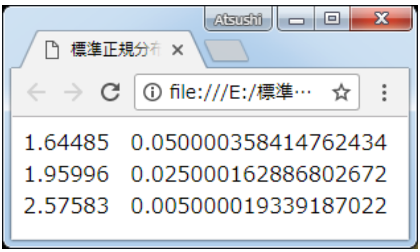
２×２分割表再訪

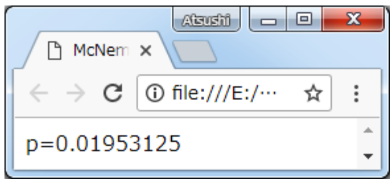
2018 年９月　加藤 厚

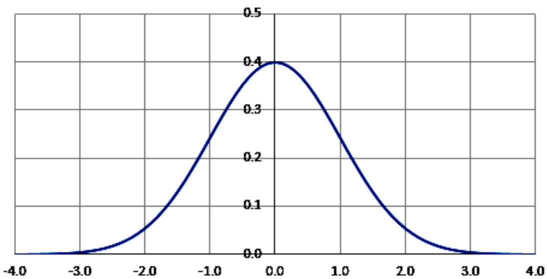
― オッズ比と直接確率を中心的話題として「組合せ」にも戻りつつ ―











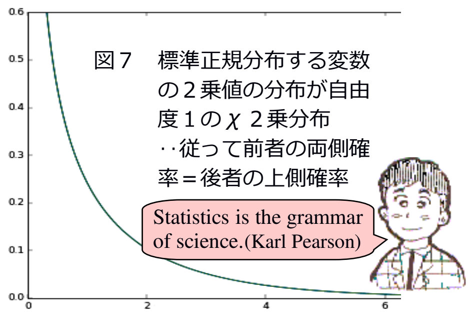


図７　標準正規分布する変数の２乗値の分布が自由度１のχ２乗分布 ‥従って前者の両側確率＝後者の上側確率

目次

Ａ５判見開きでの閲読を想定して作成した本冊子の内容は、筆者が Windows 7 を OS とする PC 上で確認した結果をまとめたものです。お気づきの点などありましたら atsushi.kato.1958@gmail.com までお知らせ頂ければ嬉しく思います。

【用語・人名】



効果量（effect size）：
差や連関などの大きさを標準化した値。実質的有意性の側面から確率的有意性（帰無仮説の下でその結果以上に偏った結果が得られる確率）を補完する。例：オッズ比やリスク比＠比率、φ係数やＶ・ｗ＠連関、η２乗＠平均など。

コーンフィールド（Jerome Cornfield 1912-1979）：【左の丸顔のオジサン】
疫学研究におけるオッズ比の有用性を示した。喫煙と肺ガンに関するフィッシャーとの議論は科学史に刻まれている：「統計的分布の全体理論を数学的にしっかりと基礎づけた癇癪持ちの天才」フィッシャーと「重要な統計量をつくるのに忙しくて修士号や博士号をとることができなかった」コーンフィールドとの間の「知的な宝物は今もわれわれとともにある。」※p.245

順列（permutation ｎＰｒ）と**組合せ**（combination ｎＣｒ）：
ABCDE から２文字の順列を選ぶ場合、１文字目は５通り、２文字目は４通りあるので 5×4 で 20 通り。20=5!/(5-2)!の一般式はｎＰｒ=n!/(n-r)!。
組合せなら AB=BA なので 20/2=10。割る数はｒ文字で可能な並べ替えの総数なのでｒ!（例：r=3 なら ABC,ACB,BAC,BCA,CAB,CBA で計６　r!=3×２×1=6）。
従って、10=5Ｃ2=5Ｐ2/2!=5!/(5-2)!/2!で、その一般式はn!/(n-r)!/r!。

信頼区間（Confidence Interval）：
標本から推定された“母数のある範囲”。オッズ比の 95%信頼区間が１を含まないなら母オッズ比が１である危険率は 5%未満なので $p<.05$ と言える。

ピアソン（Karl Pearson 1857-1936）：【中央のシブいオジサン】
観測値と理論値のズレの問題に取り組み「最初の『適合度検定』をつくりだした」。これにより「現実を説明するための二つかそれ以上の競合する数学モデルを提示し、それらのモデルのうち一つを棄却するためにデータを使うこと」つまり“仮説検定という手法”が提示された。※p.120

標準誤差（推定の）（standard error of estimate）：
標本統計量（例：標本の平均やオッズ比）の分布の標準偏差。

フィッシャー(Ronald Fisher 1890-1962)：【右のヒゲのオジサン】
「一般に利用されている有意性検定手法の大多数を開発し」た一方、「再現する方法がわからない有意な結果がぽつんとあっても、これはあらためて解明されるまで未決定のまま」であり、$.01<p<.20$ の場合には効果について言明せず「効果がよく得られるために次の実験計画をどのようにすべきか議論する」のが適切という態度をとった。※pp.122-125

※「」内はD.サルツブルグ 2006「統計学を拓いた異才たち」の各頁からの引用。

01　いきなりχ（カイ）２乗に跳ぶ前に

ボクたちが学生だった1970年代、周り(教育心理学周辺)では、対応のないカテゴリ変数間の関連の検討はχ２乗検定で“決り”でした※。

※直接確率法＝フィッシャーの正確検定は「知って」はいましたが、卒論などで実際に使った人はいたのでしょうか？

しかし、今振り返ると「期待度数と観察度数の差」に注目するχ２乗※というのは随分と工夫を凝らした考え方の指標です。

※考案者はカール・ピアソン（Karl Pearson 1857-1936）。標準偏差や相関係数の考案者でもあります。名言：「数学は科学の言語、統計学は科学の文法。」

もしボクが、心理統計の授業を受けずに新たな指導法などを試みて従来の方法を上回る成果を得たら、まずは成果（例：理解や納得）が認められた事例とそうでない事例の割合に注目することでしょう。

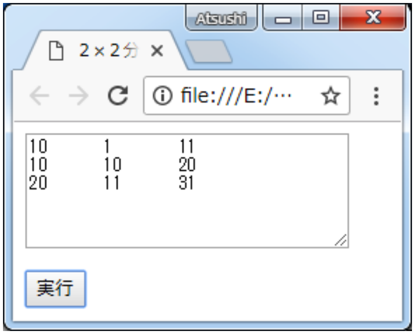
例えば、従来の方法では20人に実施して10対10（五分五分）だった「成果の有無」が、新しい方法を11人に試してみたら10対1（９割以上に成果あり）だったなら、従来の方法と比べて新しい方法は「かなり有望そう」です。

本冊子では、上の例のような「２×２分割表」という“素朴な集計結果”から意味を引き出すいくつかの方法と指標について、その特徴（求めやすさとか分かりやすさとか汎用性とか発展性とか‥）を具体的な「求め方」と共に検討していきます。

さて、まずは２組の成果の有無の度数ａ：ｂとｃ：ｄ（上の例ならa=10：b=1とc=10：d=10）から図１のような小計・総計つきの２×２分割表を作成するscriptが必要でしょう※。

※各群における成果の有無の件数は「集計済」とします。

図１　カンマ区切り数値から２×２分割表を作成する script の実行結果



下に示した<form>から</script>までの11行をコピーしてメモ帳などに貼り付け「２×２分割表01.htm」※といったファイル名で保存します。Wクリックで起動したら[実行]をクリックしてみましょう（//以降はコメント）。起動時の例：10,1,10,10（··有10=a：無1=b@新しい方法 vs. 有10=c：無10=d@従来の方法）が集計され、図１の結果が表示されるハズです。
※.txt ではなく.htm として保存します（文字コードは UTF-8 がオススメ）。

↓この一行（実行許可@IE）は Internet Explorer 以外のブラウザでは不要。

```
<!-- saved from url=(0008)about:internet -->
<form><textarea rows="6" cols="30">10,1,10,10</textarea></form>
<input value="実行" type="button" onClick="処理()">
<script>
function 処理(){
  v=document.forms[0].elements[0].value.split(",")   //文字列を,で分割
  a=v[0]*1;b=v[1]*1;c=v[2]*1;d=v[3]*1   //v[0]～[3]を数値として a～d に代入
  e=a+b;f=c+d;g=a+c;h=b+d;n=g+h   //行小計 e,f、列小計 g,h、総計 n を計算
  出力=a+"¥t"+b+"¥t"+e+"¥n"+c+"¥t"+d+"¥t"+f+"¥n"+g+"¥t"+h+"¥t"+n
  document.forms[0].elements[0].value=出力  //値をタブと改行で整形して出力
}  //↑出力指示行
</script>
```

<form>～</form>と<input ··>で6行30列の入出力欄及びクリックで function 処理(){}を実行するボタンを定義し、function 処理()の{}内で文字列の分割→数値化（*1＝１倍）と変数 a～d への代入→計算→簡易整形（+"¥t"と+"¥n"）及び変数：出力の定義→入出力欄への代入を行っています。例の 10,1,10,10 を別の値に書き換えて[実行]をクリックし、結果を確認してみましょう。

補足１：JavaScript でとまどいがちな点・便利な特長など

①<textarea>は入出力両用です（共に = の右から左への代入で実行）。
②ある対象への処理は後ろに続けて書け、文字列を記号で分割すると自動的に配列になります。例：上の function 処理()の第１行（v=··）

02　成果あり：なしの度数に注目すると

01の例では、新しい方法での成果の有無（a：b）は10：1、従来の方法のそれ（c：d）は10：10でした。これらの値から求められる指標の１つにリスク比（Risk Ratio:RR＝相対危険）※があります。

※成果ではなくてリスク（危険）に注目する理由は、この指標が疾病などの発生条件の解明に取り組む「疫学」の分野で用いられてきたからです。

そこで「指導を受けたのに成果なし」をリスクと見なすと、新しい方法での発生率は1/11、従来の方法でのそれは10/20で、リスク比（新しい方法のリスク÷従来の方法のリスク）は(1/11)/(10/20)≒0.18、つまり新しい方法は従来の方法と比べて「成果なし」に終わるリスク（危険）が５分の１未満であることが示されます。

では「リスク比を指標にすればＯＫか」というと（例のように最初から件数の定まった２群に異なる条件＝指導を与えた場合＝介入研究ならＯＫなのですが）「リスク群がまずあり、その後に集めたリスクなし群との間で条件の有無を比べる」という資料の集め方＝症例対照研究（case-control study）の場合にはＮＧです。なぜなら、リスク比は「リスクなし群の件数」によって変動してしまうからです。

Script で計算して確認してみましょう。例ではa=10：b=1→e=a+b=11、c=10：d=10→f=20なのでリスク比(1/11)/(10/20)は(b/e)/(d/f)、そこで function 処理()の出力指示行の前＝上に内容を追加する次の１行を書き込んで「２×２分割表02.htm」などとして保存します：

```
出力=出力+"¥n¥n"+"リスク比="+(b/e)/(d/f)
```

10,1,10,10で実行するとリスク比は0.18‥です。次に比率は同じままリスクなし群（aとc）の件数を半分の5と5に書き換えて5,1,5,10で実行するとリスク比は0.25=(1/6)/(10/15)に変わってしまいます。

横断調査（断面研究）や縦断・追跡調査に加え問題（例：食中毒）の発生後に条件（例：汚染食材）を遡及的に探すことも多い疫学では、事前・事後いずれの群設定※であっても共に使える指標が望まれます。
※設定基準：条件（例:指導法）の差@事前 vs. 問題（例:食中毒）の有無@事後

そんな都合のいい指標がオッズ比（Odds Ratio:OR）です。リスクの計算で分母に行小計（例：20@従来の方法）を用いるのとは異なり、オッズの計算では分母にリスクなしの件数（例：10@従来の方法）を用います（つまり「あり÷なし」）。従って、新しい方法のオッズは 10/1=10、従来の方法のそれは 10/10=1 でオッズ比は 10/1=10、つまり新しい方法は従来の方法と比べて掛け率（＝見込み）で「10 倍有利」なのです。

では、リスクなし群の件数の変化のオッズ比への影響を script で確認してみましょう。リスク比を求める行にオッズ比関連の指示を次行のように追記して同名（２×２分割表 02.htm など）で保存します：

```
出力=出力+"¥n¥n"+"リスク比="+(b/e)/(d/f)+"¥n"+"オッズ比="+(a/b)/(c/d)
```

まず 10,1,10,10 で実行するとリスク比=0.18･･、オッズ比=10。次に 5,1,5,10 にして実行するとリスク比=0.25、オッズ比=10（図２参照）。フィッシャー（Ronald Fisher 1890-1962）が交差積比（cross-product ratio）と呼んだように（a/b）/（c/d）＝a*d/（b*c）＝（d/b）/（c/a）であるオッズ比は、a と c を（0 倍以外の）何倍にしても“分子分母打ち消し”の結果「同値」なのです。

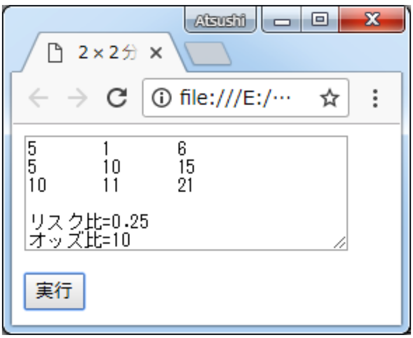


図２　値を 5,1,5,10 に変更した場合の実行結果（オッズ比は不変）

03　値を常に定め分かりやすい範囲に収める

２群の設定方法※に影響されないオッズ比の欠点は２つ：

※事前（prospective）・事後（retrospective）・同時（cross-sectional）··

1. (a/b)/(c/d)の式が示すように、b,c,dのどれかが0だと値が∞＝不定になる点。この欠点はa～dのいずれか※が0ならa～dの全てに0.5を加えて近似値を求めることで補います(ウールフの修正)。

※a～dのいずれかが0だとオッズ比の標準誤差（05参照）が∞になるため。
因みに十分多いnでd=0やb=0ならその“方法”は「百発百中」←それは「まずありえない」ことです（··d=0なら改善不能、b=0なら検定不要）。

2. 値が、２つのオッズが等しい時の1を中心として0以上∞未満に上側に広く裾を引いて分布するため「値の意味が分かりにくい」点。この欠点は、オッズ比の分布を-1～+1に変換したユールの連関係数Q※や元々-1～+1に分布する相関係数の一種のφ（ファイ＝四分点相関）係数＃との併用で補います。

※(オッズ比-1)/(オッズ比+1)=Q=(a*d-b*c)/(a*d+b*c)

＃ φ=(a*d-b*c)/√((a+b)*(c+d)*(a+c)*(b+d))

ウールフの修正とQ・φの計算を加えたfunction 処理()の一例：

```
function 処理(){
  v=document.forms[0].elements[0].value.split(",")
  a=v[0]*1;b=v[1]*1;c=v[2]*1;d=v[3]*1
  e=a+b;f=c+d;g=a+c;h=b+d;n=g+h
  出力=a+"¥t"+b+"¥t"+e+"¥n"+c+"¥t"+d+"¥t"+f+"¥n"+g+"¥t"+h+"¥t"+n
  OR=(a/b)/(c/d);l=""   //←初期値""で条件成立なら入替↓(l=label)
  if(0==a*b*c*d){OR=((a+.5)/(b+.5))/((c+.5)/(d+.5));l="(+.5修正)"}
  出力=出力+"¥n¥n"+"リスク比="+(b/e)/(d/f)+"¥n"+"オッズ比="+OR+l
  出力=出力+"¥n¥n ユールのQ="+(a*d-b*c)/(a*d+b*c)
  出力=出力+"¥nφ係数="+(a*d-b*c)/Math.sqrt(e*f*g*h)
  document.forms[0].elements[0].value=出力
}
```

例では「a～dのいずれかが0」を4値の積で判断した結果（=l（エル））及びQとφの説明文字列・値を出力に追加しています。

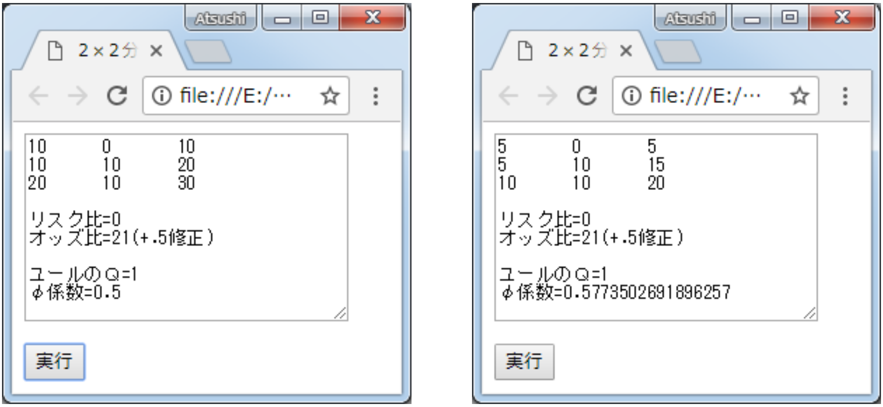


図3-1 3-2　0 を含む場合及び「リスクなし群」を半分にした場合の実行結果

function の内容を入れ替え、出力行数の増加に対応して<form>の rows="6"を rows="10"にしたら２×２分割表 03.htm などとして保存します。まず 10,0,10,10 にして実行すると、a～d に.5 を加えた値に基づくオッズ比とその旨の注(+.5 修正)が出力されます（図 3-1）。

次に、リスクあり群の件数を半分にして 5,0,5,10 で実行しても（オッズ比が同じなので）Ｑの値は変わりません。他方、φの値はリスクあり群の件数及び周辺度数（e,f,g,h）に影響されて変わりますが、こちらには「効果量」としての価値※があります（図 3-2）。
※クラメールのＶやピアソンの積率相関係数と値が一致します。

リスクあり群を倍にした 20,1,20,10、リスクなし群を倍にした 10,2,10,20、また弱・中・強の連関に対応する 10,5,5,10、10,3,3,10、10,1,1,10 などで実行してみましょう（一般に｜φ｜＜｜Q｜）※。
※「処理の繰り返し」には「再読み込み」（Windows なら[F5]キー）が便利です。

補足２：数値と文字の使い分け、等号・不等号、改行記号とタブ記号

①数字は数的処理（例：*1、-0）をすると数値に、文字を結合（例：+""）すると文字になります（123+456*1→579 vs. 123+"456"→"123456"）。
②「等しい」は（=ではなく）==です（「等しくない」は!=）。
③¥n は改行、¥t はタブ（半角８文字跳び）です（01 から既出）。

04　桁揃えや四捨五入で体裁を整える

これまでは「結果」優先で、出力書式などには無頓着でした。しかし、表は数値右詰が読み易く、また無闇に多い小数点以下の桁数は無意味です。そこで 03 の script に表の数値右詰表示と小数第５位への丸め（第６位四捨五入）の機能（function=関数）を追加します。併せて、長くなってきた処理()を作表()と計算()に function 化します。

```
<form><textarea rows="10" cols="30">10,1,10,10</textarea></form>
<input value="実行" type="button" onClick="処理()">
<script>
白=new Array(""," ","  ","   ","    ","     ","      ")
function s(x){return 白[7-(x+"").length]+x}   //spacing
function r(x){return Math.round(100000*x)/100000}   //rounding
function 作表(){
  v=document.forms[0].elements[0].value.split(",")
  a=v[0]*1;b=v[1]*1;c=v[2]*1;d=v[3]*1;e=a+b;f=c+d;g=a+c;h=b+d;n=g+h
  出力=s(a)+s(b)+s(e)+"¥n"+s(c)+s(d)+s(f)+"¥n"+s(g)+s(h)+s(n)+"¥n¥n"
}
function 計算(){
  OR=(a/b)/(c/d);l=""
  if(0==a*b*c*d){OR=((a+.5)/(b+.5))/((c+.5)/(d+.5));l="(+.5 修正)"}
  出力=出力+"リスク比="+r((b/e)/(d/f))+"¥n"+"オッズ比="+r(OR)+l
  出力=出力+"¥n¥n ユールの Q="+r((a*d-b*c)/(a*d+b*c))
  出力=出力+"¥nφ係数="+r((a*d-b*c)/Math.sqrt(e*f*g*h))
}
function 処理(){作表();計算();document.forms[0].elements[0].value=出力}
</script>
```

右詰７桁表示化には、まず半角ブランク０～６個を要素とする配列：白を用意し、その（7-添え字 x の文字としての長さ）番目の要素を x の前に付けて返す関数：function s(x)を追加します。この関数は s(x)で呼び出され、例えば a が 10 なら s(a)は文字 10 の長さ 2 を 7 から引いた 5 に対応する配列：白の５番目の要素”   ”+10 を返します。

小数第５位までの丸めは、添え字 x を 100000 倍して四捨五入（Math

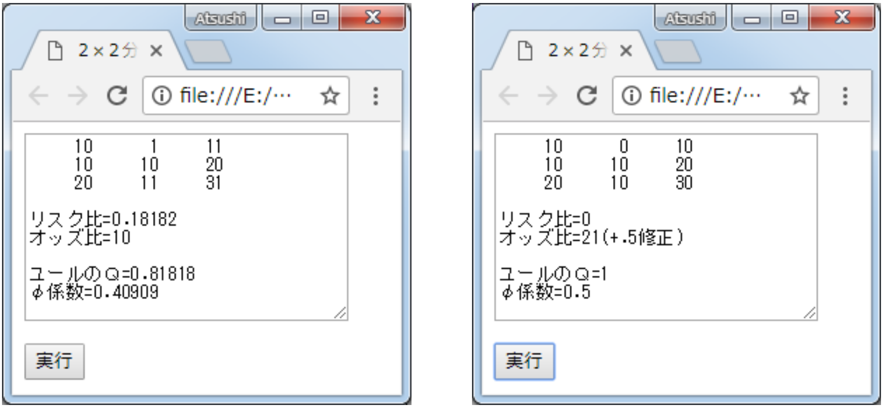


図 4-1 4-2　10,1,10,10 のまま及び 10,0,10,10 にした場合の実行結果

.round）した値を 100000 で割って返す関数：function r()を追加して r(x)で呼び出せばＯＫです。例えば x が 0.123456‥なら 100000 倍した 12345.6‥を四捨五入して 100000 で割った 0.12346 を返します。

03 の処理()中の作表部分を作表()、計算部分を計算()に function 化し、処理()の中にはそれぞれの名称：作表()と計算()及び出力指示行のみを書き並べます。<form>から</script>までの 20 行をコピーして２×２分割表 04.htm といったファイル名で保存し、10,1,10,10 のまま実行すると上の左の結果（図 4-1）が、10,0,10,10 にして実行すると右の結果（同-2）が示されます。分割表の値は右詰、指標は小数第６位四捨五入で示され、読み易い出力が得られています。

補足３：基本的な表記・指示と本資料での script の書き方など

①;で区切れば短い指示を複数列挙できます。例：OR=(a/b)/(c/d);l=""
②あるいは＝or＝論理和は||、かつ＝and＝論理積は&&。
③本資料では、「正式」な書き方に拘らず、無くても大丈夫な表現は省略しています（例：行末の;は複数の指示を列挙する場合以外は省略）。

なお、内容の積み上げ的理解には本資料とは逆の展開の方が（実は）適切：まず「組合せの比率」に基づくマクネマー検定@p.30（及び p.14 の“吹き出し”）→次に「組合せの積」を使う直接確率@pp.19-20→その上で「標準正規分布と df=1 のχ２乗分布」に基づく“近似”としてのχ２乗検定@p.14→最後に「標本オッズ比とその標準誤差」に基づく区間推定@pp.9-10

05　標本の結果から母数のある範囲を推定する

これまで、２×２分割表の集計結果から意味を引き出す以下の数値指標の特徴などを検討して来ました：

①リスク比（Risk Ratio:RR）は「危険率がｎ倍」と表現できて分かりやすいが、事後に対照（control）群を設定する症例対照研究の場合「リスクなし群の件数」に影響されて値が変わってしまう。

②オッズ比（Odds Ratio:OR）は事後の群設定でも事前のそれと同一の値が得られるが、0〜∞に分布するため値の意味が分かりにくい。

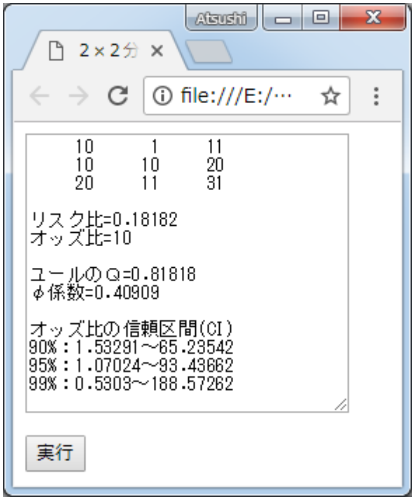
③オッズ比の意味は「値の範囲」を-1〜+1 に変換したユールの連関係数Ｑで補足できる。また相関係数の一種のφ係数＝四分点相関係数はその「筋の良さ」により効果量として使える（03 参照）。

しかし、いずれを使うにせよ、その値の意味（例：「新しい方法」の有望さ）は今回の「標本」での結果にすぎません。“一般的な意味”を実証的に示すには追試（replication :同様の実験や調査の反復実施）が確実※ですが、それには大きな手間がかかります。
※例：追試のオッズ比が順に 8,11,9,12,10,11‥なら「真の値」は約 10。

１つの標本の結果から“一般的な意味”をある確からしさで合理的に求める方法があったらとても便利なことでしょう。そんな（魔法のような？）方法が区間推定（Interval Estimation）です。

区間推定では、１つの標本から得た統計量（例：標本平均）とその分布の推測された標準偏差（例：√不偏分散）から「真の値（例：母平均）を一定の確率（例：95％）で含む区間」（信頼区間=Confidence Interval=CI）を推定します。そして上記の指標の中ではオッズ比の区間推定が最も広く利用されているようです（‥というのは疫学や社会調査などの分野の話で、教育心理学ではまず見ませんが）。

図５　オッズ比の信頼区間の計算を
追加した script の実行結果



03 でも触れたように、標本オッズ比は「1 を中心として 0 以上∞未満に上側に広く裾を引いて分布」しますが、その値を自然対数に変換した対数標本オッズ比の分布は「分散（v）が 1/a+1/b+1/c+1/d の正規分布」で近似できます。従って、対数標本オッズ比(Log OR)±1.96×√v※の値を指数変換して真数に戻せば、その２値（95%信頼限界）を両端とする範囲がオッズ比の 95%信頼区間になります。

※母平均μを区間推定する場合の「標本平均±1.96×不偏標準偏差」と同様です。

以上の処理※を行う以下の function 推定()を追加し、function 処理(){･･}内の計算();の後ろに推定();を追記し、<form>の textarea の rows=″10″を rows=″15″にしたら２×２分割表 05.htm などとして保存します。起動→実行クリックで図５の結果が表示されます。

※95%信頼区間に加え 90%と 99%のそれらも求めています。

```
function 推定(){
  LOR=Math.log(OR);v=1/a+1/b+1/c+1/d
  if(0==a*b*c*d)v=1/(a+.5)+1/(b+.5)+1/(c+.5)+1/(d+.5)
  p=new Array(90,95,99);z=new Array(1.64485,1.95996,2.57583)
  出力=出力+″¥n¥nオッズ比の信頼区間(CI)″+|
  for(i=0;i<3;i++){  //p（%）とz（値）のi番要素を代入して３回実行
    幅=z[i]*Math.sqrt(v);出力=出力+″¥n″+p[i]+″%：″
    出力=出力+r(Math.exp(LOR-幅))+″～″+r(Math.exp(LOR+幅))}
}
```

オッズ比の真の値（母オッズ比）は指定した確率※で信頼区間内にあります。従って、信頼区間に 1(同オッズ＝優劣なし)を含まない「95%：1.07･･～93.43･･」という結果は「新しい方法は従来の方法よりも有効」という“一般的な意味”を 95%以上の確率で示すものです。

※例：95%なら 100 回の試行中の 95 回。

06　「実例」で再確認

これまでの内容を実例で再確認してみましょう。ビタミンが発見されていない明治時代、脚気は原因不明の病気でした。そんな中、海軍医官高木兼寛は食事に注目し、白米に大麦を混ぜるなどの変更を加えて 1884 年に 287 日間の遠洋航海を行い、前年に同様の航海を行った場合と発病状況を比較しました。その内訳は右の分割表の通りです（杉 2013 pp. 96-97）※。

	非発病	発病	小計
改善食	319	14	333
従来食	207	169	376
小　計	526	183	709

※杉 晴夫 2013 「栄養学を拓いた巨人たち」 講談社 ブルーバックス B-1811

これまでのパターンに従って上の行が望ましい条件（改善食）、左側が望ましい結果（非発病）とすると、入力順は改善食&非発病, 改善食&発病, 従来食&非発病, 従来食&発病で各数値は 319, 14, 207, 169 です。例の 10, 1, 10, 10 をこれらの値に置き換えて実行すると図 6 の結果が得られます。

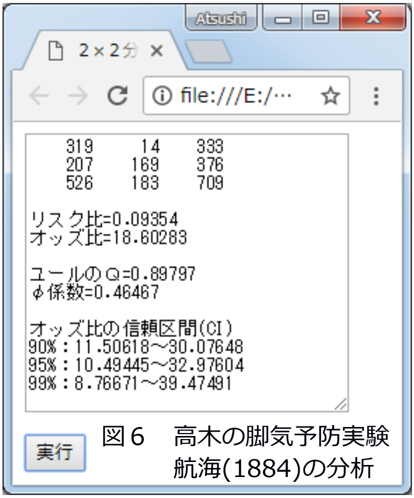


図 6　高木の脚気予防実験航海(1884)の分析

リスク比（14/333÷169/376）の 0.09 は改善食により脚気発病の危険度が 1/10 以下に減少することを、オッズ比（319/14÷207/169＝(319*169)÷(14*207)）の 18.6 は従来食の約 19 倍の「非発病」効果が改善食に見込めることを示すものです。ユールの連関係数Qは. 90 と高く、効果量としてのφ係数も. 46 で「中程度の連関」が認められます。オッズ比の信頼区間は全て 1.0（優劣なし）を大きく上回り、オッズ比の真の値（母オッズ比）が 10.5～33.0 の区間内にあることが 95%の確率で推定できる結果が得られました※。

※a～d の値が大きいと標準誤差は小さくなり、より狭い（＝特定的な）信頼区間が得られます。

入出力欄にポインタ（矢印）を重ねると簡単な入力ガイドを表示する機能（title="表示内容"）を追加した script を以下に提示します：

```
<form><textarea rows="15" cols="30" title="条件&結果:1&1,1&0,0&1,0&0 の４値を入力">
10,1,10,10</textarea></form>
<input value="実行" type="button" onClick="処理()">
<script>
白=new Array(""," ","  ","   ","    ","     ","      ")
function s(x){return 白[7-(x+"").length]+x}   //spacing
function r(x){return Math.round(100000*x)/100000}   //rounding
function 作表(){
  v=document.forms[0].elements[0].value.split(",")
  a=v[0]*1;b=v[1]*1;c=v[2]*1;d=v[3]*1;e=a+b;f=c+d;g=a+c;h=b+d;n=g+h
  出力=s(a)+s(b)+s(e)+"¥n"+s(c)+s(d)+s(f)+"¥n"+s(g)+s(h)+s(n)+"¥n¥n"
}
function 計算(){
  OR=(a/b)/(c/d);l=""
  if(0==a*b*c*d){OR=((a+.5)/(b+.5))/((c+.5)/(d+.5));l="(+.5 修正)"}
  出力=出力+"リスク比="+r((b/e)/(d/f))+"¥n"+"オッズ比="+r(OR)+l
  出力=出力+"¥n¥n ユールの Q="+r((a*d-b*c)/(a*d+b*c))
  出力=出力+"¥n φ係数="+r((a*d-b*c)/Math.sqrt(e*f*g*h))
}
function 推定(){
  LOR=Math.log(OR);v=1/a+1/b+1/c+1/d
  if(0==a*b*c*d)v=1/(a+.5)+1/(b+.5)+1/(c+.5)+1/(d+.5)
  p=new Array(90,95,99);z=new Array(1.64485,1.95996,2.57583)
  出力=出力+"¥n¥n オッズ比の信頼区間(CI)"+l
  for(i=0;i<3;i++){  //p（%）と z（値）の i 番要素を代入して３回実行
    幅=z[i]*Math.sqrt(v);出力=出力+"¥n"+p[i]+"%："
    出力=出力+r(Math.exp(LOR-幅))+"～"+r(Math.exp(LOR+幅))}
}
function 処理(){作表();計算();推定();document.forms[0].elements[0].value=出力}
</script>
```

この script を２×２分割表 06.htm などとして保存し、入出力欄にポインタ（矢印）を重ねると「入力ガイド」が右のようにポップアップします。

07　さて、やっとχ２乗ですが‥

統計量としてのχ２乗は（２×２以上の）分割表一般でそれを求められる点がオッズ比より優れています（χ２乗から求める連関の指標のクラメールのＶは２×２分割表の場合にはφ係数と同値）。他方、ｐ値しか与えないχ２乗に対し、（ｐ値も判断できる※）信頼区間が示せるオッズ比の方が「引き出せる意味は豊か」です。
※95%信頼区間が１を含まない（例：1.2～4.8、0.6～0.8）ならp<.05で有意。
　なぜなら「母オッズ比＝1（２条件に優劣なし）の確率が５％未満」だから。

加えて、χ２乗検定には「期待度数が１未満のセルがあってはいけない」とか「期待度数が５未満のセルが全体の20%以上あってはいけない」とか「イェーツの補正の使用が望ましい」とか逆に「使用は不要」とか、正確検定の近似に関する様々な条件や（異なる）意見？が付随しており「スッキリしない感じ」がつきまといます。が、（順序として）まずその来歴と求め方などを紹介しましょう。

1897年にゴールトン（ダーウィンの従兄弟で「指紋の独自性」の発見者としても有名）からロンドンの生物測定研究所を引き継いだカール・ピアソンの功績の１つが「適合度検定」（当てはまりのよさの検定）の考案です。人間を含む生物について様々な測定を行っていた彼らには、収集できた資料（標本）とその全体（母集団）の実態との関連を示す必要がありました。そこでまず「理論などから期待される分布と実際の標本における分布とのズレ」を評価する指標としてのχ２乗を発案したワケです。

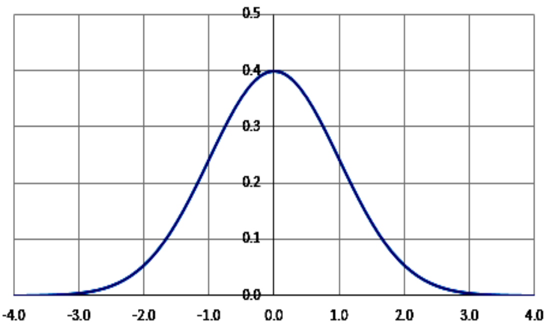
簡単な例として「コイン投げ」を取り上げましょう。10回あるいは10個投げて、表と裏の割合の偏りがどの程度までなら（表裏半々の意味で）「正しいコイン」と言えるでしょう‥５：５なら期待通り、４：６や６：４もアリガチ、では３：７や２：８ならどうでしょうか？

10回中ｒ回表が出る組合せの数は10Ｃr※なのでr=0から順に1, 10, 45, 120, 210, 252, 210, 120･･､組合せ総数1024(2の10乗)で割った確率は0から順に約.001, .010, .044, .117, .205, .246, .205, .117･･。従ってそれ以上の偏りが偶然起こる確率は２：８や８：２で約.055(=.044+.010+.001)、１：９や９：１で約.011です。※nＣr=n!/(n-r)!/r!

カテゴリ数が２のこの問題は組合せで容易に解けますが、３以上の場合にも簡単に近似値を得るための方法※が、理論から期待される度数（期待度数）と標本における実際の度数（観察度数）の差の２乗を期待度数で割った値の総和（＝χ２乗）を指標とする適合度検定です。
※様々な場合（自由度）のχ２乗分布の確率数表が整備されています。

今回のχ２乗は５：５なら0、４：６なら(-1*-1)/5+(1*1)/5=0.4、３：7なら(-2*-2)/5+(2*2)/5=1.6、２：８なら(-3*-3)/5+(3*3)/5=3.6、１：９なら(-4*-4)/5+(4*4)/5=6.4･･です。他方、カテゴリ数が２＝自由度１のχ２乗分布の５％値は標準正規分布の上･下側確率2.5%値約±1.96（05の「信頼限界」参照）の２乗の約3.84です。従って、偏りが２：８では3.6<3.84@両側検定のため危険率５％では「正しくない」とは言えません。

ここで要注意なのは-∞～+∞に分布する標準正規分布（下左図）を２乗した値は0～+∞に分布（下右図）するため、その上側５％には標準正規分布の上・下側が2.5%ずつ含まれる点です。従って、前段落の判断は８：２側の偏りも含んでおり、２：８側の偏りだけに注目するなら上側10%値の約2.71（約1.645の２乗）が５％値に当たるため、2.71<3.6で「危険率５％で正しくない」と言えます（片側検定）。他方、この判断は組合せに基づく結果（p=.055）とは異なります（これが「近似の悪さ」の問題）。



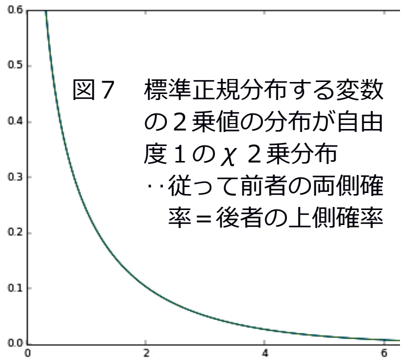


図７　標準正規分布する変数の２乗値の分布が自由度１のχ２乗分布
･･従って前者の両側確率＝後者の上側確率

08　独立性のχ２乗検定と連続性の修正

07 の適合度検定を「２つのカテゴリ変数は独立」と仮定した場合に拡張し、仮定が真の場合の期待値と実際の度数の差からχ２乗を求めるのが独立性のχ２乗検定です。実際の度数 a,b,c,d（a+b=e,c+d=f,a+c=g,b+d=h,a+b+c+d=n）の期待値は、仮定が真なら総数 n の均等配分値なので順に e*g/n[←n*(e/n)*(g/n)]、e*h/n、f*g/n、f*h/n です。そこで、以下の function chi2()を追加し、function 処理()の推定();の次に chi2();を追記の上、rows="17"にして例の 10,1,10,10 を処理すると「ピアソンのχ２乗=5.18802」と表示されます。

```
function chi2(){ea=e*g/n;eb=e*h/n;ec=f*g/n;ed=f*h/n
CHI2=(a-ea)*(a-ea)/ea+(b-eb)*(b-eb)/eb+(c-ec)*(c-ec)/ec+(d-ed)*(d-ed)/ed
出力=出力+"¥n¥n ピアソンのχ２乗="+r(CHI2)
}
```

例で検討したいのは新しい方法の有効性なので逆（１：10）側の偏りは無視すると、上側確率 10％の約 2.71 と２％の約 5.41（約 2.326 の２乗）が片側確率５％と１％の値となり、約 5.19 は 2.71<5.19<5.41 のため危険率５％で有意です。他方、a=10,b=1,c=10,d=10 の期待値は順に、7.1,3.9,12.9,7.1 で b のそれが５未満です。従って正確検定の近似としての条件※中の「期待値５未満のセルが全体の 20％以上あってはいけない」は満たされていません。

※これらが課される理由は、離散変数のχ２乗を連続分布で近似する証明にスターリングの公式（階乗の指数関数近似の公式）を使っているためだそうです。

このような場合に「イェーツの補正」を勧める意見があります。逆に「補正は常に不要」という意見もあります。連続性の修正でフィッシャーの正確検定に「近づける」なら、正確検定で直接確率を求めるのが“本道”とも言えます（10 参照）。ここではその議論はさておいて処理を進めます。上の function chi2()の計算式は「考え方をそのまま式にした」定義式でした。理解しやすい点が長所ですが、式中の全

値が揃っていないと複数段階の処理が必要、除算が多いと丸め誤差が大きくなるなどの短所があります。そこで、簡潔さと精度を重視して「最少の段階と除算で定義式と同じ答を得られるように工夫した式」が計算式です。χ２乗のそれは下の function chi2()の１行目、「差が 0.5 より大きければ 0.5 を引いてから２乗する」（差が 0.5 以下なら０とする）が定義のイェーツの補正を加えたχ２乗（YCC2）のそれは２行目です。最小期待度数（me）を求めて出力に追記する script を含む以下の function chi2()を以前のそれと置き換え、textarea を rows="19"にして２×２分割表 08.htm などとして保存します。

```
function chi2(){tmp=Math.abs(a*d-b*c);CHI2=n*tmp*tmp/(a+b)/(c+d)/(a+c)/(b+d)
  YCC2=n*(tmp-n/2)*(tmp-n/2)/(a+b)/(c+d)/(a+c)/(b+d);if(tmp<=n/4)YCC2=0
  me=e*g/n;if(e*h/n<me)me=e*h/n;if(f*g/n<me)me=f*g/n;if(f*h/n<me)me=f*h/n
  出力=出力+"¥n¥n ピアソンのχ２乗="+r(CHI2)+"¥nχ２乗(連続修正)="+r(YCC2)
  出力=出力+"¥n 最小期待度数="+r(me)
}
```

起動して実行をクリックすると図８の結果が表示されます。連続性を修正したχ２乗の値は約 3.55※で 2.71<3.55<5.41 なのでやはり５％水準で有意です。
※連続性の修正はχ２乗の値を減らします。

オッズ比の信頼区間では p<.05（05 参照）、修正なしのχ２乗では p<.05、修正χ２乗でも p<.05 で、今回の例については連続性の修正は「大まかな判断」を変えませんでした。

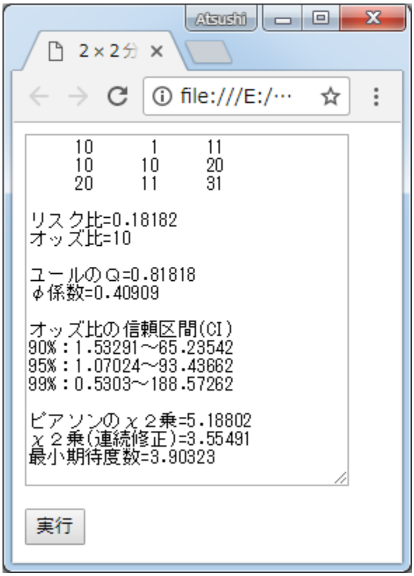


図８　連続修正なし・ありのχ２乗値及び最小期待度数の表示を追加した script の実行結果

09　確率的有意性と近似確率の自動表示

自由度が１のχ２乗分布の場合、上側確率 10,5,2,1%に対応する値は順に約 2.71,3.84,5.41,6.64 です。また、偏りの一方を無視する片側検定の場合、具体的には対立仮説の内容が≠ではなくて＞や＜（例：新しい方法は従来の方法より有効）の場合、各値は半分の確率（順に 5,2.5,1,0.5%）に対応します。

毎回これらの値を参照するのは面倒なので、上側確率 p を n.s.,p<.10,p<.05,p<.01 などと自動表示させたいなら function chi2()に以下の３行を追加し、変数：出力の定義の(CHI2)の後ろに+p、(YCC2)の後ろに+y と追記すればＯＫです。

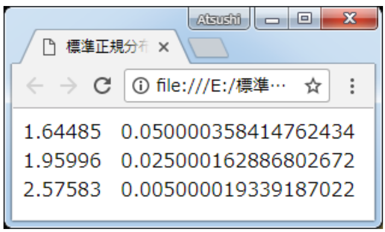
```
s=new Array(" n.s."," p<.10"," p<.05"," p<.02"," p<.01")
x=CHI2;p=s[0];if(2.71<x)p=s[1];if(3.84<x)p=s[2];if(5.41<x)p=s[3];if(6.64<x)p=s[4]
x=YCC2;y=s[0];if(2.71<x)y=s[1];if(3.84<x)y=s[2];if(5.41<x)y=s[3];if(6.64<x)y=s[4]
```

また、確率の近似値は、標準正規分布の累積確率密度の近似計算式（例：Box-Muller 法）にχ２乗値の正・負の平方根を代入して得られる値の和（＝両側確率）※として求められます（自由度１のχ２乗分布＝標準正規分布する変数の２乗値の分布）。
※実際は正の平方根を代入して得た片側確率×２＝両側（上側）確率でＯＫ。

Box-Muller 法の精度は以下の script の実行で確認できます。

```
<script>
a=1.64485;b=1.95996;c=2.57583;s="  "  //標準正規分布の上側確率 5,2.5,0.5%値
document.write(a+s+nd(a)+"<br>"+b+s+nd(b)+"<br>"+c+s+nd(c))
function nd(x){
  p=0.2316419;b1=0.31938153;b2=-0.356563782;b3=1.781477937;b4=-1.821255978
  b5=1.330274429;t=1/(1+p*Math.abs(x));Z=Math.exp(-x*x/2)/Math.sqrt(2*Math.PI)
  return Z*((((b5*t+b4)*t+b3)*t+b2)*t+b1)*t
}
</script>
```

起動画面には右の各値が表示されます。



左側の 1.64‥～2.57‥の３値は数表から求めた標準正規分布の上側確率 5,2.5,0.5％の値（05 で信頼区間の計算にも使用）、それらを function nd(x)に x として代入した戻り値（return で返る値）が右側の 0.0500‥、0.0250‥、0.00500‥の３値で、小数第６位まで０が続く高い精度を示しています。

以上の自動表示機能を組み込んだ function は以下の通りです：

```
function chi2(){tmp=Math.abs(a*d-b*c);CHI2=n*tmp*tmp/(a+b)/(c+d)/(a+c)/(b+d)
  YCC2=n*(tmp-n/2)*(tmp-n/2)/(a+b)/(c+d)/(a+c)/(b+d);if(tmp<=n/4)YCC2=0
  me=e*g/n;if(e*h/n<me)me=e*h/n;if(f*g/n<me)me=f*g/n;if(f*h/n<me)me=f*h/n
  s=new Array(" n.s."," p<.10"," p<.05"," p<.02"," p<.01");l="¥n上(両)側確率 p="
  x=CHI2;p=s[0];if(2.71<x)p=s[1];if(3.84<x)p=s[2];if(5.41<x)p=s[3];if(6.64<x)p=s[4]
  x=YCC2;y=s[0];if(2.71<x)y=s[1];if(3.84<x)y=s[2];if(5.41<x)y=s[3];if(6.64<x)y=s[4]
  出力=出力+"¥n¥n ピアソンのχ２乗="+r(CHI2)+p+l+r(2*nd(Math.sqrt(CHI2)))
  出力=出力+"¥nχ２乗(連続修正)="+r(YCC2)+y+l+r(2*nd(Math.sqrt(YCC2)))
  出力=出力+"¥n 最小期待度数="+r(me)
}
function nd(x){  //Box-Muller 法
  p=0.2316419;b1=0.31938153;b2=-0.356563782;b3=1.781477937;b4=-1.821255978
  b5=1.330274429;t=1/(1+p*Math.abs(x));Z=Math.exp(-x*x/2)/Math.sqrt(2*Math.PI)
  return Z*((((b5*t+b4)*t+b3)*t+b2)*t+b1)*t
}
```

textarea を rows="21"にして２×２分割表 09.htm などとして保存・実行すると図９の結果が表示され、p<‥※と一貫する確率の近似値※が得られています。

※共にχ２乗の上側（＝正規分布の両側）確率なので、対立仮説に方向性があれば値は半分＃にできます。

＃両側の p<.02/.10＝片側の p<.01/.05

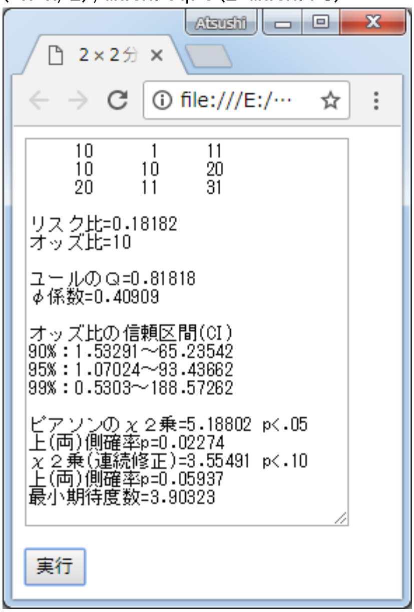


図９　危険率の表示などを追加した script の実行結果

10　正確検定：その１　観察度数パターンなどの確率

さて、χ２乗に関する 07～09 の内容につきまとってきた「スッキリしない感じ」を、08 で予告した“本道”つまり直接確率法で晴らす時が来ました（‥07 の「吹き出し」の方法も直接確率法です）。

フィッシャーの正確検定は以下の３段階で「観察された度数以上の偏りの発生確率」を直接求める方法です。

1. 観察された度数パターンの発生確率を求める。
2. 同一周辺度数(行と列の小計)において 1. のパターンと連続する「確率がより小さいパターン」の確率の総和を 1. の値と合計する（これが「片側確率」）。
3. 同一周辺度数で起こりうる度数パターンの確率を全て求め、1. の値以下の値を総和する（これが「両側確率」、その値－2. の値が逆側の「片側確率」）。

10	1	11
10	10	20
20	11	31

11	0	10	1	9	2	8	3	7	4		
9	11	10	10	11	9	12	8	13	7		
0.2%		2.4%									
6	5	5	6	4	7	3	8	2	9	1	10
14	6	15	5	16	4	17	3	18	2	19	1

0	11
20	0

例(黄)と同一の周辺度数で起こりうる全度数パターン

まず 1.、つまり例の a=10：b=1 対 c=10：d=10 のパターンの確率を求めましょう。新しい方法で指導された計 11 人から 10 人を選ぶ組合せは（1 人を選ぶ組合せと同じ）11 通り、従来の方法で指導された計 20 人から 10 人を選ぶ組合せ（20Ｃ10）は 20!/(20−10)!/10! で 184756 通り（←nＣr=n!/(n−r)!/r!）。前者の組合せと後者のそれは独立なので両者の組合せは結局 11×184756 =2032316 通りあります。

他方、31 人から（a+c の）20 人を選ぶ全ての組合せは（11 人を選ぶ組合せと同じ）31Ｃ20 なので 31!/(31−20)!/20!=84672315 通り。

度数パターンの発生確率は「該当する全ての組合せ÷起こりうる全ての組合せ」なので(11Ｃ10)×(20Ｃ10)÷(31Ｃ20)＝11×184756÷84672315＝2032316÷84672315≒0.024 で約 2.4%。そして、この値は計算式：e!*f!*g!*h!/(n!*a!*b!*c!*d!)でも求められます。

次に 2.、つまり観察されたパターンに連続する確率がより小さい＝より偏ったパターンの全確率を求めます。同一周辺度数において 1. のパターンから連続するより偏ったパターンは 11, 0, 9, 11 の 1 つのみでその確率は 11!*20!*20!*11!/31!/11!/0!/9!/11!≒0.00198 で約 0.2%。従って、片側確率は約 2.6%（←約 2.4+約 0.2）となり、結局、オッズ比の信頼区間の.01<p<.05 やピアソンの χ 2 乗及び連続修正 χ 2 乗の.01<p<.03（両側確率の半分の値）と一貫する値が得られました。

ここまでの過程を行う function の一例※は以下の通りです：
※a〜d 中 b が最小を前提とした script なので他のパターンには非対応！

```
function 正確(){
  定数=階(e)*階(f)*階(g)*階(h)/階(n);出力=出力+"¥n¥n 度数パターン¥n"
  for(j=b;0<=j;j--){
    p=Math.round(100000000*定数/階(e-j)/階(j)/階(g-e+j)/階(h-j))/100000000
    出力=出力+"("+(e-j)+","+j+","+(g-e+j)+","+(h-j)+")¥tp="+p+"¥n"
  }
}
function 階(x){r=1;for(i=1;i<=x;i++){r*=i}return r}
```

まず関数 funciton 階(x)で階乗を定義し、function 正確()では計算式の定数部分を計算→“度数パターン”と改行を出力に追加→b=1 と 0 の場合の確率 p を計算→（度数パターン）と共に出力に追加という処理を行っています。
function 処理()の chi2();の後ろに正確();を追記の上 2 × 2 分割表 10.htm などとして保存・実行すると図 10 の結果が表示されます。

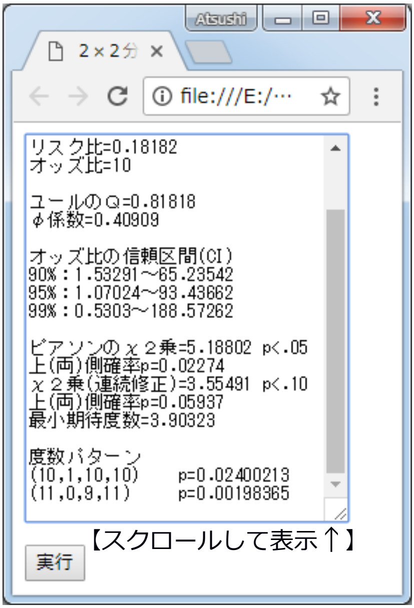


図 10　観察度数パターンの確率などを追加した script の実行結果

11　その２　可能な全度数パターンとその確率

10 の function で正しい値を得るには a～d 中の最小値を b の位置に入力する必要があり面倒です。また、観察パターンを含む可能な全パターンとその確率の一覧は「全体像を踏まえた理解」を促します。そこで順不問の観察度数入力を処理して可能な全パターンとその確率を一覧的に示す script を工夫してみます。

入力順については「a:b と c：d の各対の反転による a の最小化」で対処します。それは以下の４類型の対応により実現できます：

1. 元々の入力において a が最小ならそのまま
2. 〃 b が最小なら両対の左右を入れ替える：b⇔a と d⇔c
3. 〃 c が最小なら両対の上下を入れ替える：c⇔a と d⇔b
4. 〃 d が最小なら両対の上下と左右を入れ替える：d⇔a と c⇔b

最小値を a に置く理由は、度数パターン（a, b, c, d）の先頭の a を 0 から可能な最大値※まで変化させることで「度数の推移」を最も明瞭に示せ、推移の把握が容易そうだからです（図 11 参照）。

※a の可能な最大値は a+b と a+c の小さい方と一致します。なぜなら a がその最大値をとる時、b・c のいずれかが 0 になるからです。

以上の処理を追記した function 正確()などは以下の通りです。

```
function 正確(){
  m=a;if(b<m)m=b;if(c<m)m=c;if(d<m)m=d
  if(b==m){b=a;x=d;d=c;c=x;a=m}   //左右
  if(c==m){c=a;x=d;d=b;b=x;a=m}   //上下
  if(d==m){d=a;x=b;b=c;c=x;a=m}
  終=a+b;if(c<b)終=a+c;e=a+b;f=c+d;g=a+c;h=b+d;n=g+h  //入替を反映
  出力=出力+"¥n¥n 可能な全パターン¥n";定数=階(e)*階(f)*階(g)*階(h)/階(n)
  for(j=0;j<=終;j++){
    確率=r8(定数/階(j)/階(e-j)/階(g-j)/階(f-g+j))
    出力=出力+"("+j+","+(e-j)+","+(g-j)+","+(f-g+j)+")¥tp="+確率+"¥n"
  }
}
function r8(x){return Math.round(100000000*x)/100000000}
function 階(x){r=1;for(i=1;i<=x;i++){r*=i}return r}
```

まずaを最小値mに初期値として代入し、以後b,c,dと比べてより小さい値があればそれをmに代入します。次にmがb,c,dのいずれかであったら順に左右・上下・左右&上下を入れ替えて新しいa～dなどを定めます。また、aが取りうる最大値（a+bとa+cの小さい方）を変数：終に代入します。

パターン発生確率の計算式：e!*f!*g!*h!/(n!*a!*b!*c!*d!)中のe!*f!*g!*h!/n!は周辺度数が同一の場合は一定なのでまずその値を求めて定数とし、aの値が0から最大値に至る各パターン※の確率をforループで順次求めてパターン(a,b,c,d)と共に出力に追記します。

※２×２の場合、同一周辺度数ならaが決まればb,c,dも決まります（自由度１）。従って、j=aに対してa,b,c,dはその関数(j,e-j,g-j,f-g+j)になります。

前節の正確()と置き換え、小数第９位を四捨五入して８位に丸めるfunction r8()を追加して２×２分割表11.htmなどとして保存・実行すると図11の結果が表示されます（前節から入出力欄を21行で打ち止めとしたため表示には要スクロール）。

例の度数パターン：10,1,10,10が1,10,10,10に並べ替えられ、最小値の1をaとして同一周辺度数で可能なaの範囲0～11の全パターンとその確率※が表示されています。パターンは10（前節）の「全度数パターン」と一致し、例はaが最も少ない側から２番目のパターンであることが確認できます。

※1e-8は1÷(10の８乗)=0.00000001

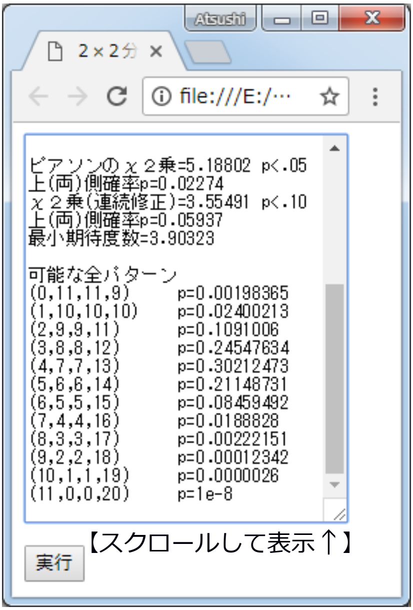


図 11　可能な全度数パターンとその確率を追加した script の実行結果

12　その３　片側・両側確率の明示など

11 の function 正確()は順不問の観察度数入力から可能な全パターンと確率の全体像を示します。その推移の中に観察パターンの位置と片側及び両側の確率が明示されれば一層便利でしょう。

図 11 にも示されているように、偏りの大きいパターンほど確率は小さいため、値は低く始まって中間部分で高まりその後低下します。従って、最も偏りが大きい a=0 から観察度数パターンまでの確率の和が片側確率、その後観察度数パターンのそれを下回った確率から a の最大値の確率までの和が逆側の片側確率、両者の和が両側確率です。そこで、並べ替え後の観察度数パターンとその確率をまず示し、続いてパターンと確率の推移を「観察度数パターンまでの片側確率」のラベルを含めて示し、最後に逆側の片側確率と両側確率を提示するよう加筆した function 正確()は以下の通りです：

```
  if(d==m){d=a;x=b;b=c;c=x;a=m} （この行までは 11 と同じ）
  終=a+b;if(c<b)終=a+c;出力=出力+"¥n¥n 観察度数パターン¥n"  //見出し
  e=a+b;f=c+d;g=a+c;h=b+d;n=g+h  //入替を反映
  定数=階(e)*階(f)*階(g)*階(h)/階(n);観察=r8(定数/階(a)/階(b)/階(c)/階(d))
  出力=出力+"("+a+","+b+","+c+","+d+")¥tp="+観察+"¥n¥n 可能な全パターン¥n"
  両=0;片=0;l="■片側確率="  //0 は初期値  l はラベル  ↑見出し
  for(j=0;j<=終;j++){
    確率=r8(定数/階(j)/階(e-j)/階(g-j)/階(f-g+j))
    if(確率<=観察){両=両+確率;片=片+確率}  //観察度数の確率以下なら加算
    出力=出力+"("+j+","+(e-j)+","+(g-j)+","+(f-g+j)+")¥tp="+確率+"¥n"
    if(a==j){出力=出力+l+r8(片)+"¥n";片=0}
  }  //最初の片側確率の追記と片の初期化↑
  出力=出力+l+r8(片)+"¥n□両側確率="+r8(両)
}
```

まず観察度数パターンという見出しと入替後の(a, b, c, d)及びその確率＝観察を出力に追加します。続いて for ループ内で観察度数の確率以下の確率を変数：両と片に加算し、最初に a==j つまり可能パターンが観察度数と一致した時点でラベル（l）■片側確率=とそれまでの確

率の和（片）を追記して変数：片を 0 に戻します。ループが終了した時点で変数：片に加算されている逆側の片側確率と両側確率をそれぞれのラベル付きで出力に追記して入出力欄に代入します。

変更を含む終=a+b;‥以下の 12 行を差し替えて２×２分割表 12.htm などとして保存・実行すると図 12 の結果が表示されます。これで、２×２分割表において観察されたある度数パターンが、周辺度数同一という条件の下で可能な全パターン中のどこに位置し、またそれ以上に偏ったパターンも含めた発生確率の合計はどれほどかを、両側確率及び逆側の片側確率も含めて明確に把握できるようになりました。

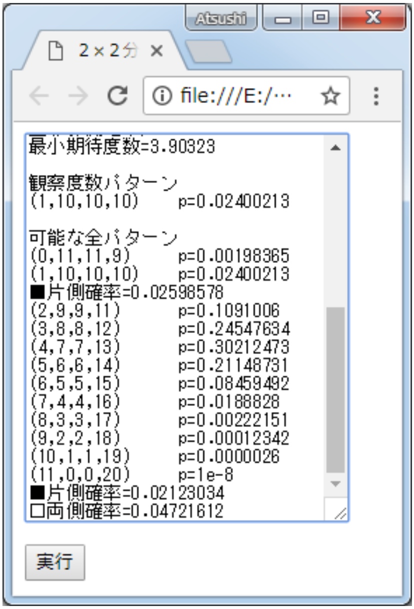


図 12　第一の片側確率の対応位置への表示などを加えた script の実行結果

結局、同一周辺度数で可能な全 12 パターン中、観察された「成果なし１人」とより偏った「成果なし０人」の確率は約.024 と約.002 で計約.026、つまり危険率 5%で有意（p<.05）でした。08 で示したように「オッズ比の信頼区間では p<.05、修正なしのχ２乗では p<.05、連続性を修正したχ２乗でも p<.05」で、今回の例の場合はどの方法でも同様の結果が得られています。また、09 で示したようにピアソンのχ２乗の片側確率は.02274 の半分で約.011、連続修正後のそれは.05927 の半分で約.030 ですから、直接確率（約.026）に最も近い値を示す近似方法は確かに「連続性を修正したχ２乗」でした。

13　その４　より大きい件数の処理@script の工夫

12 までの script には、処理可能件数にかなり低い上限があります。例えば、全度数３倍の 30,3,30,30 は正しく処理できますが、４倍の 40,4,40,40 では p=Infinity になってしまいます※（２倍でも 5≦最小期待度数なのでχ２乗近似に問題なし→直接確率は計算不要ですが‥）。

※理由は、計算の過程で JavaScript が標準機能で処理できる値の上限（2 の 53 乗=9007199254740992<171!）を超える値が発生するからです。

そこで、第１の対策として計算式を「分数の積」化します。具体的には、定数：e!*f!*g!*h!/n!を使わず、本来の計算式 e!*f!*g!*h!/n!/a!/b!/c!/d!を分数の積の式：(e!/n!)*(f!/d!)*(g!/c!)*h!/b!/a!に変換すると変数：観察と確率は以下の２行になります：

```
観察=r8((階(e)/階(n))*(階(f)/階(d))*(階(g)/階(c))*階(h)/階(b)/階(a))
確率=r8((階(e)/階(n))*(階(f)/階(f-g+j))*(階(g)/階(g-j))*階(h)/階(e-j)/階(j))
```

この対策で５倍の 50,5,50,50 までは正しく処理可能になりますが、６倍の 60,6,60,60 では全ての p が 0 になってしまいます。

第２の対策として、階乗の除算を「範囲の積」※にしてみます。具体的には function 積(n1,n2)を定義して確率を以下のように求めます。

※例えば 30!/28!は重なる部分を相殺した 30*29 で求められます。

```
観察=r8((1/積(n,e))*積(f,d)*積(g,c)*積(h,b)/階(a))
確率=r8((1/積(n,e))*積(f,f-g+j)*積(g,g-j)*積(h,e-j)/階(j))
function 積(n1,n2){x=1;for(i=n1;n2<i;i--){x*=i}return x}
```

この対策で７倍の 70,7,70,70 までは正しく処理可能になりますが、８倍の 80,8,80,80 では全ての p が 0 になってしまいます。

第３の対策として、これまで大小を考慮してこなかった e～h と a～d をそれぞれ大→小に並び替えた上で n1・n2 に代入して「積の範囲」の最小化を図ります。また、確認用に全ての確率の総和を変数：全に

```
  e=a+b;f=c+d;g=a+c;h=b+d;n=g+h  //入替を反映（この行までは 12 と同じ）
  ta=a;tb=b;tc=c;td=d;te=e;tf=f;tg=g;th=h  //計算用の仮(tmp)の a～h
  do{r=0;if(te<tf){x=te;te=tf;tf=x;r=1};if(tf<tg){x=tf;tf=tg;tg=x;r=1}
  if(tg<th){x=tg;tg=th;th=x;r=1}}while(r==1)
  do{r=0;if(ta<tb){x=ta;ta=tb;tb=x;r=1};if(tb<tc){x=tb;tb=tc;tc=x;r=1}
  if(tc<td){x=tc;tc=td;td=x;r=1}}while(r==1)
  観察=r8((1/積(n,te))*積(tf,td)*積(tg,tc)*積(th,tb)/積(ta,1))
  出力=出力+"("+a+","+b+","+c+","+d+")¥tp="+観察+"¥n¥n 可能な全パターン¥n"
  両=0;片=0;全=0;l="■片側確率=";L="□両側確率=";t="    p の総和="
  for(j=0;j<=終;j++){t1=te;t2=f-g+j;t3=g-j;t4=e-j;t5=j
    do{r=0;if(t1<t2){x=t1;t1=t2;t2=x;r=1};if(t2<t3){x=t2;t2=t3;t3=x;r=1}
    if(t3<t4){x=t3;t3=t4;t4=x;r=1};if(t4<t5){x=t4;t4=t5;t5=x;r=1}}while(r==1)
    確率=r8((1/積(n,t1))*積(tf,t2)*積(tg,t3)*積(th,t4)/積(t5,1))
    全=全+確率;if(確率<=観察){両=両+確率;片=片+確率}
    出力=出力+"("+j+","+(e-j)+","+(g-j)+","+(f-g+j)+")¥tp="+確率+"¥n"
    if(a==j){出力=出力+l+r8(片)+"¥n";片=0}
  }
  出力=出力+l+r8(片)+"¥n"+L+r8(両)+"¥n"+t+r8(全)
}
function 積(n1,n2){x=1;for(i=n1;n2<i;i--){x*=i}return x}
```

保存します。以上の対策を加えた script は上の通りです。第３の対策で８倍の 80, 8, 80, 80 までは正しく処理可能になりますが、９倍の 90, 9, 90, 90 では（各パターンの確率は求められるものの）観察度数パターンが p=Infinity、片側確率=1.00 などの不正値になってしまいます。

以上の対策で（限界はあるものの）上限はかなり拡大され、例えば 80, 55, 20, 45（中澤 2018＃ p.113）などを正しく処理※できるようになりました（図 13‥この例は第２の対策までで処理可能です）。

※処理の正しさは、パターンの推移に伴う p 値の変化、p の総和≒1.0 などで確認できます。

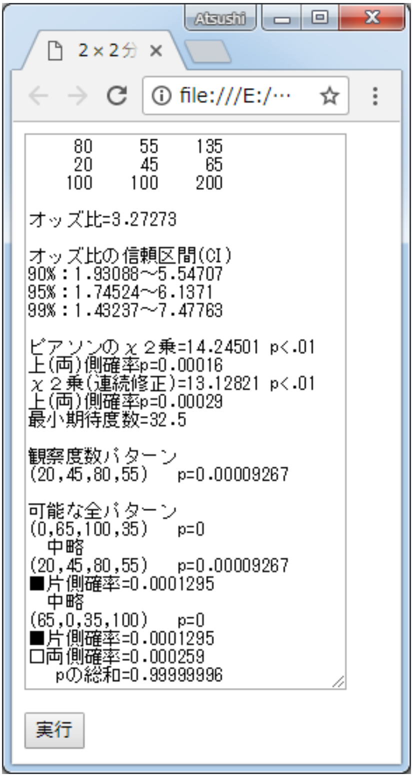


図 13　第１～３の対策を施した script でかなり大きい件数を処理した結果（省略あり）

＃中澤 港　2018　R による保健医療データ解析演習
http://minato.sip21c.org/msb/medstatbookx.pdf

14　その５　より大きい件数の処理@任意精度演算ライブラリ

07 で述べたように、信頼区間からｐ値も判断できるオッズ比の方が「情報が豊か」ですから、「スッキリしない感じ」がつきまとうχ２乗検定や計算が煩雑な正確検定の出る幕は２×２分割表の場合にはそもそも無さそうです。加えて、13 でも触れたように計算の過程で数値が処理可能な上限を超えるような観察度数ならχ２乗分布による近似の前提条件（5≦最小期待度数など）はまず満たされておりχ２乗検定で問題ナシ、正確検定に拘る必要はありません。

従って、実際上は不必要≒無意味なのですが「できない＝残念」なので、処理できる数値の上限を拡大するライブラリを使った script を書いてみます。

```
<form><textarea rows="21" cols="36" title="条件&結果:1&1,1&0,0&1,0&0の４値を入力">
319,14,207,169</textarea></form>
<input value="実行" type="button" onClick="処理()">
<script src="big.js"></script>
<script>
白=new Array(""," ","  ","   ","    ","     ","      ")
function s(x){return 白[7-(x+"").length]+x}
function 作表(){
  v=document.forms[0].elements[0].value.split(",")
  a=v[0]*1;b=v[1]*1;c=v[2]*1;d=v[3]*1;e=a+b;f=c+d;g=a+c;h=b+d;n=g+h
  出力=s(a)+s(b)+s(e)+"¥n"+s(c)+s(d)+s(f)+"¥n"+s(g)+s(h)+s(n)
}
```

今回は任意精度演算ライブラリ big.js※を使用することとし、その旨を script に書き込みます。big.js はデフォルトでは小数第 20 位までの精度で計算を行うので cols="36"に列数を増やし、例として 06 で紹介した高木の度数(319,14,207,169)を示し、使わない function r()を削除します。　https://github.com/MikeMcl/big.js

※今回のスクリプトの作動には↑上の URL の web 頁からダウンロードした圧縮ファイル（big.js-master.zip 約 1.2MB）から展開したライブラリ big.js（MikeMcl, 2017 22KB）が同一フォルダ内に必要です。

```
function 正確(){
  m=a;if(b<m)m=b;if(c<m)m=c;if(d<m)m=d;if(b==m){b=a;x=d;d=c;c=x;a=m}
  if(c==m){c=a;x=d;d=b;b=x;a=m};if(d==m){d=a;x=b;b=c;c=x;a=m}
  終=a+b;if(c<b)終=a+c;出力=出力+"¥n¥n 観察度数パターン¥n"
  e=a+b;f=c+d;g=a+c;h=b+d;n=g+h;両=0;片=0;全=0;l="■片側確率="
  定数=階(e).times(階(f)).times(階(g)).times(階(h)).div(階(n))
  観察=r8(定数.div(階(a)).div(階(b)).div(階(c)).div(階(d)))
  出力=出力+"("+a+","+b+","+c+","+d+")¥tp="+観察+"¥n¥n 可能な全パターン¥n"
  for(j=0;j<=終;j++){
    確率=r8(定数.div(階(j)).div(階(e-j)).div(階(g-j)).div(階(f-g+j)))
    全=全+確率;if(確率<=観察){両=両+確率;片=片+確率}
    出力=出力+"("+j+","+(e-j)+","+(g-j)+","+(f-g+j)+")¥tp="+確率+"¥n"
    if(a==j){出力=出力+l+r8(片)+"¥n";片=0}
  }
  出力=出力+l+r8(片)+"¥n□両側確率="+r8(両)+"¥n"+"    p の総和="+r8(全)
}
function r8(x){return Math.round(100000000*x)/100000000}
function 階(x){r=Big(1);for(i=1;i<=x;i++){r=r.times(i)}return r}
function 処理(){作表();正確();document.forms[0].elements[0].value=出力}
</script>
```

function s(x)、r8(x)は以前のままで、function 正確()と階(x)について上限を超える数値の演算書式を big.js 用に書き換えます※。

※定数は Big(n)、乗算は x.times(y)、除算は x.div(y)です。加算は x.plus(y)、減算は x.minus(y)、また平方根は.sqrt()、べき乗は.pow(n)など。

function 処理{}の指定を作表();正確();及び出力関連のみにして２×２分割表 14.htm などとして保存・実行すると図 14 の結果が表示されます（途中経過を大幅に省略。なお処理完了には数秒掛かります）。p=0 つまり p<.01 という結果は 06 のそれと一致します。処理ができてスッキリしましたが、結局、信頼区間も示すオッズ比を用いた分析の方が「情報が豊か」です。 ;-)

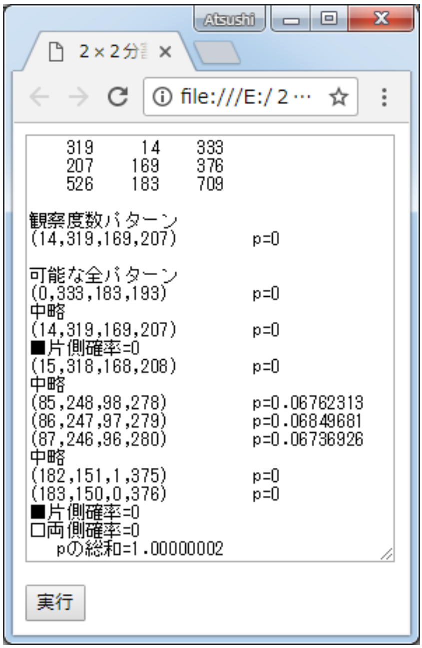


図 14 任意精度演算ライブラリを用いる script で 06 の例を処理した結果（省略あり）

15　まとめとオマケ（マクニマー検定）

これまであれこれ試してきたことを一言でまとめると「２×２分割表でカテゴリ変数の独立性を分析するなら区間推定でオッズ比の有意性を示すと共に母数の存在範囲を示唆※し、効果量としてφ係数を示すのがベスト」（つまり 06 の script で十分）ということです。
※後者が“示唆”に留まる理由は「求めた信頼区間が母オッズ比を含むか否か」は確率的（95 対 5 とか 99 対 1 とか）だからです（母オッズ比の存在範囲の特定には追試が必要）。

χ２乗検定は「正確検定の近似法」、正確検定の確率も「帰無仮説が真の場合に観察度数以上の偏りが得られる確率」に過ぎず、「見込みの比」や“独立でない程度”※に関する情報は与えてくれません。
※χ２乗から「独立でない程度」を求めるのがクラメールの連関係数Ｖ。

従属変数が得点なら、中央値・合格点などで２分割すれば分散の等質性が気になる t 検定を使わずに分析できます（検定力はやや低下しますが‥）。交互作用が予想される独立変数が複数あるなら「最大の効果が予想される独立変数の組合せ」が当てはまる群とその他の群を比較すれば、多要因分散分析の勉強は不要です。これらの例が示すように、２群（実験群 vs. 対照群）間で２カテゴリ（例：成功 vs. 不成功）の度数を比較する２×２分割表のオッズ比は、仮説検証の容易で汎用的な指標なのです。

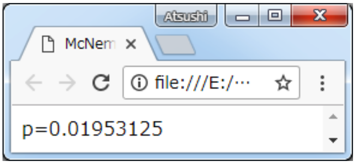
最後に、オマケとして「測定に対応がある場合」の２×２分割表のマクニマー検定も組合せに基づいていることに触れておきましょう。この場合、集団は事前と事後の２回の観察に基づいて２×２の４区画（1→1 が a、1→0 が b、0→1 が c、0→0 が d）に分割されます。例として、指導による理解の変化を取り上げ、20 人を対象として a=6、b=1（←混乱した？）、c=8、d=5 だったとします。興味があるのは「変

		指導後	
指		有	無
導	有	6	1
前	無	8	5

化」なのでaとdの6+5=11人は無視し、bとcの1+8=9人に注目します。全９人中８人以上が無→有に変化する組合せは‥と考えると、これは07の吹き出しと同じ問題で８人の9C8=9と９人の9C9=1の計10。全ての変化の組合せ（9C9～9C0）の総数は２の９乗=512なので、８人以上が無→有に変化する確率は10÷512=0.01953125(2%弱)です※。
※「指導効果の検討」なので逆（有→無の方が多い）側の確率は考慮しません。

一般化して度数の大きい方の変化の件数をrとすると「r以上の各場合の組合せの数の総和÷全ての変化の組合せの総数」が確率で、その値を求める極力単純なscriptの一例と実行結果は以下の通りです：

```
<script>
b=1;c=8;n=b+c;m=Math.max(b,c);和=0;全=Math.pow(2,n)
for(j=m;j<=n;j++){和=和+階(n)/階(n-j)/階(j)}
function 階(x){y=1;for(i=1;i<=x;i++){y*=i}return y}
p=和/全;if(.5<p){p=.5};document.write("p="+p)
</script>
```



bとcの値を例えば2と8､1と9に書き換えて実行するとp=0.0546875とp=0.0107421875が得られます。07の吹き出しと比べてみましょう。これ（及びそのχ２乗近似）がMcNemar test（対応のある２値の変化の有意性の検定）です。

今回の紹介はここまで。最適な統計処理は確かに「画竜点睛」ですが重要なのは「研究自体の内容と達成」、“点”の打ち方に拘り過ぎて肝心の竜が貧弱では本末転倒というものです。その意味でも、今回紹介したオッズ比とその信頼区間のような「簡単に求められて情報も豊かな指標」がより広く活用されるべきではないでしょうか。

補足４：２×２よりも大きい分割表などの正確検定

標記処理は青木繁伸先生（元群馬大学教授）の以下の頁で可能です。
http://aoki2.si.gunma-u.ac.jp/exact/exact.html