

二項分布モデルでのP値

- 黒木玄
- 2025-11-26作成

このファイルは以下の場所で閲覧できます。

- <https://github.com/genkuroki/public/blob/main/0055/binomial%20P-values.ipynb>
(<https://github.com/genkuroki/public/blob/main/0055/binomial%20P-values.ipynb>)
- <https://nbviewer.org/github/genkuroki/public/blob/main/0055/binomial%20P-values.ipynb>
(<https://nbviewer.org/github/genkuroki/public/blob/main/0055/binomial%20P-values.ipynb>)
- <https://raw.githubusercontent.com/genkuroki/public/refs/heads/main/0055/binomial%20P-values.pdf>
(<https://raw.githubusercontent.com/genkuroki/public/refs/heads/main/0055/binomial%20P-values.pdf>)

Google Colab (<https://colab.research.google.com/>)でも実行できます。

- <https://colab.research.google.com/github/genkuroki/public/blob/main/0055/binomial%20P-values.ipynb>
(<https://colab.research.google.com/github/genkuroki/public/blob/main/0055/binomial%20P-values.ipynb>)

目次

- ▼ [1 二項検定のP値に関する解説の例](#)
 - ▼ [1.1 P値の導入](#)
 - [1.1.1 片側P値](#)
 - [1.1.2 両側P値 \(単にP値と呼ぶことが多い\)](#)
 - [1.1.3 P値の報告の仕方](#)
 - [1.1.4 細かな注意](#)
 - [1.1.5 一般に統計分析は何をやっているとみなされるか](#)
 - [1.1.6 P値の計算の仕方の詳しい説明](#)
 - [1.2 両側P値の計算練習問題](#)
 - [1.3 二項分布の確率の表の作成](#)
 - [1.4 無作為抽出された患者20人中6人に効いた場合の両側P値](#)
 - [1.5 無作為抽出された患者20人中15人に効いた場合の両側P値](#)
 - [1.6 S値\(=意外度\)のコイン投げによる定義](#)
 - [1.7 P値に関する広く通用する閾値は存在しない](#)
 - [1.8 相性の良さ/悪さという解釈の仕方](#)
 - [1.9 P値関数とS値関数](#)
 - [1.10 相性区間 = CI = 信頼区間](#)
 - [1.11 報告の仕方について](#)
- [2 二項検定の複数種類のP値関数のグラフ](#)
- [3 参考文献](#)

```
In [1]: ▶ 1 # IPAフォントを手動でインストール ⇒ https://moji.or.jp/ipafont/ipa00303/
2 # このノートブックでは"ipag"フォントを使用する。
3 haskey(ENV, "COLAB_GPU") && run(`apt-get -y install fonts-ipafont-gothic`)
```

Out[1]: false

```
In [2]: ▶ 1 # Google Colabと自分のパソコンの両方で使えるようにするための工夫
2
3 haskey(ENV, "COLAB_GPU") && (ENV["JULIA_PKG_PRECOMPILE_AUTO"] = "0")
4 using Pkg
5
6 """すでにPkg.add済みのパッケージのリスト"""
7 _packages_added = [sort!(readdir(Sys.STDLIB));
8   sort!([info.name for (uuid, info) in Pkg.dependencies() if info.is_direct_dep])]
9
```

Out[2]: _packages_added

```
In [3]: ▶ 1 """_packages_added内  にないパッケージをPkg.addする"""
2 add_pkg_if_not_added_yet(pkg) = if isnothing(Base.find_package(pkg))
3   println(stderr, "# $(pkg).jl is not added yet, so let's add it.")
4   Pkg.add(pkg)
5 end
```

Out[3]: add_pkg_if_not_added_yet

```
In [4]: 1 """expr::Exprからusing内の`.`を含まないモジュール名を抽出"""
2 function find_using_pkgs(expr::Expr)
3     pkgs = String[]
4     function traverse(expr::Expr)
5         if expr.head == :using
6             for arg in expr.args
7                 if arg.head == :. && length(arg.args) == 1
8                     push!(pkgs, string(arg.args[1]))
9                 elseif arg.head == :(:) && length(arg.args[1].args) == 1
10                    push!(pkgs, string(arg.args[1].args[1]))
11                end
12            end
13        else
14            for arg in expr.args arg isa Expr && traverse(arg) end
15        end
16    end
17    traverse(expr)
18    pkgs
19 end
```

Out[4]: find_using_pkgs

```
In [5]: 1 """必要そうなPkg.addを追加するマクロ"""
2 macro autoadd(expr)
3     pkgs = find_using_pkgs(expr)
4     :(add_pkg_if_not_added_yet.($(pkgs))); $expr)
5 end
```

Out[5]: @autoadd

```
In [6]: 1 ENV["LINES"], ENV["COLUMNS"] = 200, 200
2 const mincho = "ipamp"
3 const gothic = "ipagp"
4
5 # Google Colab では6分程度かかる
6 @autoadd begin
7     using Distributions
8     using Plots
9 end
10
11 default(fmt=:png, fontfamily=gothic,
12         titlefontsize=12, legendfontsize=12, guidefontsize=12, plot_titlefontsize=12)
13 default(fontfamily=gothic) # 日本語フォントの利用
14
15  $x \lesseqgtr y = x < y$  ||  $x \approx y$ 
16  $x \gtrless y = x > y$  ||  $x \approx y$ 
```

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

```
In [7]: 1 """
2     central法による二項分布の両側P値
3     二項分布binにおける「k以下になる確率の2倍」と「k以上になる確率の2倍」の小さい方の値
4
5     このP値からClopper-Pearsonの相性区間(信頼区間)が得られる。
6 """
7 pvalue_central(bin, k) = min(2cdf(bin, k), 2ccdf(bin, k-1))
8 pvalue_central(n, k, p) = pvalue_central(Binomial(n, p), k)
9
10
11 """Clopper-Pearsonの相性区間"""
12 function ci_clopper_pearson(n, k; α = 0.05)
13     p_L = k > 0 ? quantile(Beta(k, n-k+1), α/2) : zero(α)
14     p_U = k < n ? quantile(Beta(k+1, n-k), 1-α/2) : one(α)
15     [p_L, p_U]
16 end
```

Out[7]: ci_clopper_pearson

```
In [8]: 1 """
2         minimum likelihood法による二項分布の両側P値
3         このP値からSterneの相性区間(信頼区間)が得られる。
4         """
5         function pvalue_minlike(bin, k)
6             supp = support(bin)
7             sum(pdf(bin, i) for i in supp if pdf(bin, i) ≤ pdf(bin, k))
8         end
9         pvalue_minlike(n, k, p) = pvalue_minlike(Binomial(n, p), k)
10
11        """Sterneの相性区間は実装せず"""
```

Out[8]: "Sterneの相性区間は実装せず"

```
In [9]: 1 """
2         score法による二項分布の両側P値
3         このP値は二項分布の正規分布近似で定義された両側P値と同じ。
4
5         このP値はWilsonのスコア相性区間(信頼区間)を与える。
6         """
7         function pvalue_score(bin, k)
8             (; n, p) = bin
9             phat = k/n
10            se = sqrt(p * (1 - p) / n)
11            z = (phat - p) / se
12            2ccdf(Normal(), abs(z))
13        end
14        pvalue_score(n, k, p) = pvalue_score(Binomial(n, p), k)
15
16        """Wilsonのスコア相性区間"""
17        function ci_score(n, k; α = 0.05)
18            p̂ = k/n
19            z = quantile(Normal(), 1-α/2)
20            a, b, c = 1+z^2/n, p̂+z^2/(2n), p̂^2
21            # ap^2 - 2bp + c = 0 を解く.
22            sqrtD = √(b^2 - a*c)
23            p_L = (b - sqrtD)/a
24            p_U = (b + sqrtD)/a
25            [p_L, p_U]
26        end
```

Out[9]: ci_score

```
In [10]: 1 """
2         Wald法による二項分布の両側P値
3         このP値も二項分布の正規分布近似で定義された両側P値だが、
4         データ \n, k\$ から推定する必要がない \${\hat{p}}=k/n\$ の標準偏差
5         を推定してしまっているので、score法の両側P値よりも精度が低い。
6
7         このP値は教科書によく書いてある母比率に関するWaldの相性区間(信頼区間)を与える。
8         """
9         function pvalue_wald(bin, k)
10            (; n, p) = bin
11            phat = k/n
12            sehat = sqrt(phat * (1 - phat) / n)
13            z = (phat - p) / sehat
14            2ccdf(Normal(), abs(z))
15        end
16        pvalue_wald(n, k, p) = pvalue_wald(Binomial(n, p), k)
17
18        """Waldの相性区間"""
19        function ci_wald(n, k; α = 0.05)
20            p̂ = k/n
21            SEhat = √(p̂*(1-p̂)/n)
22            z = quantile(Normal(), 1-α/2)
23            p_L = p̂ - z*SEhat
24            p_U = p̂ + z*SEhat
25            [p_L, p_U]
26        end
```

Out[10]: ci_wald

1 二項検定のP値に関する解説の例

以下の例は中原治著『基礎から学ぶ統計学』([中原治2022])の「1-5 二項検定」の節の「[5] STEP 4: 棄却域と有意水準」以降の説明をP値に関する説明で置き換えることを意識して書かれたものです。経緯については

- <https://x.com/genkuroki/status/1993698819541004669?s=20> (<https://x.com/genkuroki/status/1993698819541004669?s=20>)

にあるX (旧Twitter)におけるスレッドを参照して下さい。

その本の例題1.1では、B薬は60%の患者に効果があることが分かっていると仮定されており、薬Aについては無作為に選んだ患者20人中18人に効果があったというデータの値が得られていると仮定されています。以下の説明でもこの設定を使います。

1.1 P値の導入

例題1.1の文脈では、特別な数値でその値の **小ささ** を次のように解釈できるものをデータの値から計算できると便利です:

薬Aが無作為に選んだ患者20人中18人に効いたというデータの値が得られたとき、薬Aが効く患者の割合がB薬と同じ60%だと信じている人は確率的にどれだけ意外だと感じるか。

以下では、この目的のために使える **P値 (P-value)** という数値の構成の仕方について、例題1.1の場合を例に説明します。

例題1.1の状況:

- 薬Bは患者の60%に効くことがわかっている。
- 薬Aについては患者の何%に効くかはわかっていない。
- 「無作為に選んだ患者20人中18人に薬Aに効果があった」というデータの値が得られた。
- 以上の状況で薬Aと薬Bが効く患者の割合を比較したい。

1.1.1 片側P値

例題1.1の状況では、上の目的のために使える **片側P値 (one-sided P-value)** を

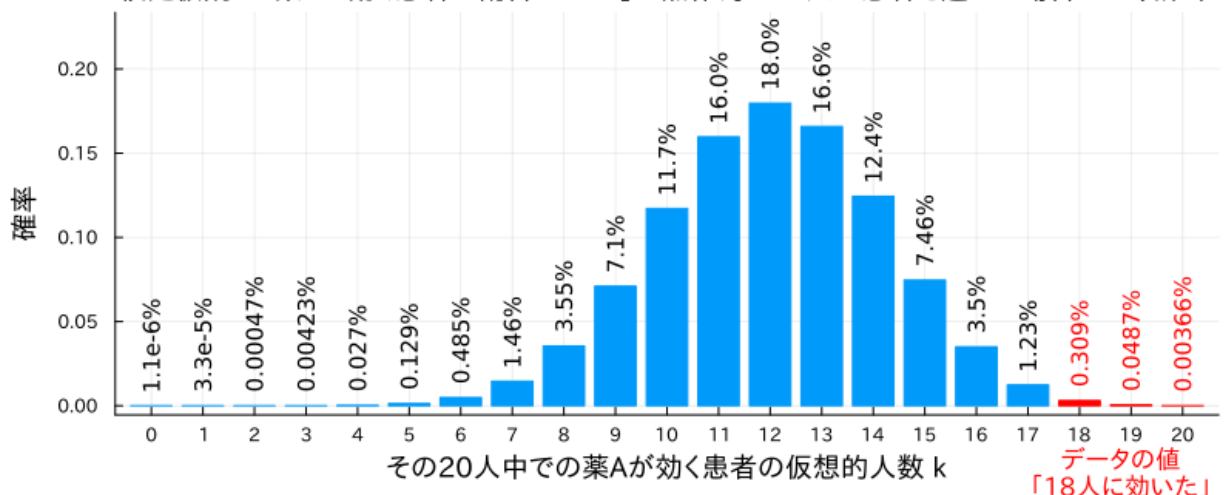
効果がある患者の割合が60%だと仮定したとき、無作為に選んだ患者20人中18人 **以上** に効果がある確率

と定義できます。この確率の値(片側P値の値)は0.36%とかなり小さくなります(計算の仕方は後で説明します)。そのことは、

「薬AとB薬が効く患者の割合が等しい」と信じている人にとって、「薬Aが無作為に選んだ患者20人中18人に効いた」というデータの値はかなり意外に感じられることを意味しています。

```
In [11]: 1 n, k, p = 20, 18, 0.6
2 bin = Binomial(n, p)
3 bar(0:k-1, k → pdf(bin, k); label="", c=1, lc=:match)
4 bar!(k:n, k → pdf(bin, k); label="", c=:red, lc=:match)
5 plot!(xtick=0:n, xlim=(-0.7, n+0.7), ylim=(-0.03, 1.3).*pdf(bin, mode(bin)))
6 plot!(xguide="その20人中での薬Aが効く患者の仮想的人数 k", yguide="確率")
7 title!("検定仮説が「薬Aが効く患者の割合は60%」で\
8   無作為に20人の患者を選んだ場合の二項確率")
9 for j in 0:n
10   c = j < k ? :black : :red
11   annotate!(j, pdf(bin, j)+0.05pdf(bin, mode(bin)),
12     text(string(round(100pdf(bin, j); sigdigits=3))*"%", :left, 90.0, 10, c))
13 end
14 annotate!(18.8, -0.04, text("データの値\n「18人に効いた」", gothic, 11, :red))
15 plot!(legend=:topleft, size=(800, 350),
16   bottommargin=8Plots.mm, leftmargin=4Plots.mm)
```

Out[11]: 検定仮説が「薬Aが効く患者の割合は60%」で無作為に20人の患者を選んだ場合の二項確率



このようにP値は「モデルの確率分布でデータの値以上に極端な値が生成される確率」として構成されることが多い。

上の片側P値の場合には「データの値以上に極端な値が生成される確率」の意味は「データの値の18人以上に効く確率」に設定されている。

1.1.2 両側P値 (単にP値と呼ぶことが多い)

さらに **両側P値 (two-sided P-value)** が上の片側P値の2倍で定義されます。より詳しくは、この両側P値を

「薬Aが無作為に選んだ患者20人中18人に効いた」というデータの値に関する「薬Aが効く患者の割合が60%である」という検定仮説の両側P値

と呼びます。単にP値と言えば両側P値を意味することが多いです。

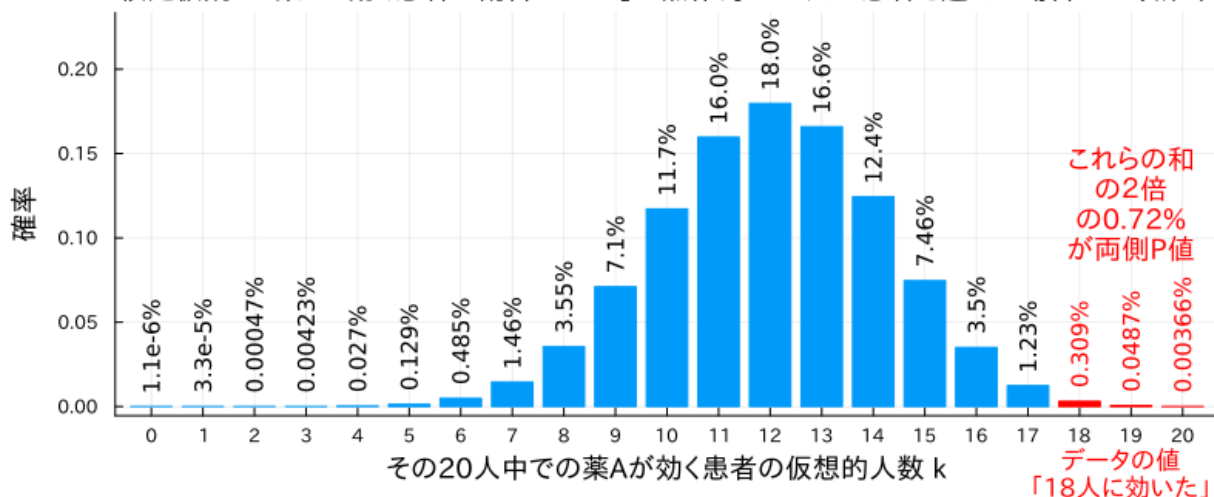
以上の場合の両側P値は0.72%でやはりかなり小さいので、

「薬Aが効く患者の割合が薬Bと同じ60%である」という検定仮説を信じている人にとって、「薬Aが無作為に選んだ患者20人中18人に効いた」というデータの値はかなり意外だと感じられるだろう

といえます。

```
In [12]: 1 n, k, p = 20, 18, 0.6
2 bin = Binomial(n, p)
3 bar(0:k-1, k → pdf(bin, k); label="", c=1, lc=:match)
4 bar!(k:n, k → pdf(bin, k); label="", c=:red, lc=:match)
5 plot!(xtick=0:n, xlim=(-0.7, n+0.7), ylim=(-0.03, 1.3).*pdf(bin, mode(bin)))
6 plot!(xguide="その20人中での薬Aが効く患者の仮想的人数 k", yguide="確率")
7 title!("検定仮説が「薬Aが効く患者の割合は60%」で\
8     無作為に20人の患者を選んだ場合の二項確率")
9 for j in 0:n
10     c = j < k ? :black : :red
11     annotate!(j, pdf(bin, j)+0.05pdf(bin, mode(bin)),
12         text(string(round(100pdf(bin, j); sigdigits=3))*"%", :left, 90.0, 10, c))
13 end
14 annotate!(19, 0.12, text("これらの和\nの2倍\nの0.72%\nが両側P値",
15     gothic, 12, :red))
16 annotate!(18.8, -0.04, text("データの値\n「18人に効いた」", gothic, 11, :red))
17 plot!(legend=:topleft, size=(800, 350),
18     bottommargin=8Plots.mm, leftmargin=4Plots.mm)
```

Out[12]: 検定仮説が「薬Aが効く患者の割合は60%」で無作為に20人の患者を選んだ場合の二項確率



一般に

「無作為に選んだ患者n人中k人に薬Aは効いた」というデータの値に関する「薬Aが効く患者の割合はpである」という検定仮説の両側P値

の小ささは

「薬Aが効く患者の割合はpである」という検定仮説を信じている人が、「無作為に選んだ患者n人中k人に薬Aは効いた」というデータの値に感じる意外さ

を表しています。

以下では単にP値と言ったら、両側P値を意味するものとします。

P値は小さければ小さいほど、統計モデル(例えば二項分布)とそのパラメータに関する検定仮説の組み合わせの下でデータの値の意外さが大きいことを意味します。

1.1.3 P値の報告の仕方

P値を P と書くとき、 $P < 0.05$ のようにP値の具体的な値が不明になるような報告の仕方をしてはいけません。

P 値を報告する場合には $P = 3.2\%$ のように具体的な値がわかるように報告するようにしましょう。ただし、例外的にP値が $P = 0.000023 = 2.3 \times 10^{-5}$ のように非常に小さな値になり、近似計算の精度を超えている場合には $P < 0.0001$ のように報告しても良いでしょう。

$P = 0.04205785495554228$ のように無意味に高過ぎる精度でP値を報告してもいけません。複雑な現実における統計分析では、理想的なデータの取得は不可能だったり、統計分析のために採用する統計モデルの妥当性自体が不明だったりするので、高精度なP値の値には意味がないと考えられます。

一般に、統計分析の結果を報告するときには、P値などの統計分析で得られる数値では測ることができないバイアスや不確実性があることを謙虚に認めることが必要になります。

この解説ではP値(P-value)のPを大文字で書くことにしたのですが、小文字の p と書いても問題ありません。どちらでもよいです。

1.1.4 細かな注意

以下の注意は最初に読むときにはとばしても構いません。

用語法に関する補足: 検定仮説を単に仮説と呼ぶこともあります。伝統的な用語法では検定仮説は帰無仮説とも呼ばれていますが、帰無仮説(null hypothesis)は意味が分かり難く、英単語のnullはzeroという意味を持つせいで、差がないことを意味するゼロ仮説を主に意味するかのように誤解されがちなので、この解説では帰無仮説という用語を避けることにします。(検定仮説(test hypothesis)という用語を初期のNeymanが使っていたので、検定仮説もある意味で十分に由緒正しい用語だと考えられる。「検定」は英語のtestの翻訳で「検定仮説」は「データの値とモデルを使ってテストする仮説」というような意味になります。)

片側確率の2倍の両側P値の正確な定義と欠点: 「無作為に選んだ n 人中 k 人に薬が効いた」というデータの値が得られたときの、片側確率の2倍で定義された「薬が効く患者の割合は p である」という仮説の両側P値の正確な定義は

薬が効く確率が p であるという仮定の下で、無作為に選んだ n 人中 k 人以上に効く確率の2倍と無作為に選んだ n 人中 k 人以下に効く確率の小さい方の値

です。このように定義された両側P値の欠点は

- 無用に値が大きくなり過ぎるという性質を持つ。
- 1 より大きくなることもある。

P値は確率だとみなしたい値なので0以上1以下になって欲しいのですが、上の定義だと1より大きくなってしまいう場合が出て来てしまいます。その欠点をなくすために、

薬が効く確率が p であるという仮定の下で、無作為に選んだ n 人中 k 人以上に効く確率の2倍と無作為に選んだ n 人中 k 人以下に効く確率と 1 の中での最小値

と定義することも多いです。

片側P値だけではなく、両側P値を考えたい理由: 薬Aが無作為に選んだ患者20人中60%の12人よりもずっと大きな人数(例えば18人)に効果があった場合だけではなく、その反対側のずっと小さな人数(例えば5人)にしか効果がない場合にも、薬Aが効く患者の割合が60%だと信じている人にとっては意外だと感じられるでしょう。このことから、無作為に選んだ患者20人中60%の12人からデータの値以上に離れる確率の小ささで、「薬Aが効く患者の割合は60%」という仮説を信じている人から見たデータの値の意外さを評価するという考え方も自然だということに気がきます。そのような考え方で定義されるP値が両側P値です。上ではシンプルに反対側の確率はもとの片側確率に等しいだろうという大雑把な想定のもとで、片側確率の2倍で両側P値を定義しました。モデルの確率分布が左右非対称な場合には他にも無数の選択肢があるのですが、そういう問題については統計学の技術的側面についてもっと詳しくなってから考えれば十分です。

尤度とP値の違いに注意: 上では片側P値を「効果がある患者の割合が60%であると仮定したとき、無作為に選んだ患者20人中18人 以上 に効果がある確率」と定義し、両側P値をその2倍で定義しました。読者の中には太字になっている「以上」を削除して、「効果がある患者の割合が60%であると仮定したとき、無作為に選んだ患者20人中18人に効果がある確率」を考えたい人があると思います。その人は鋭い! 「以上」の語を取り去ってできる確率は **尤度(ゆうど)** と呼ばれており、統計分析における最も基本的かつ重要な量になっています。尤度とP値を混同しないように注意して下さい。

尤度についてさらに補足:「効果がある患者の割合が60%である」という仮説を信じている人が「無作為に選んだ患者20人中18人に効果があった」というデータの値を見たときには、シンプルに「効果がある患者の割合が60%であると仮定したとき、無作為に選んだ患者20人中18人に効果がある確率」(=尤度(ゆうど))を **モデルとデータの値を比較するための尺度** として使いたくなると思います。そのアイデアは極めて正しく、尤度は **モデルがどれだけデータの値に適合しているか** を表していると解釈でき、**最尤法 (maximum likelihood method)** の基礎になります。そして尤度をさらに有効利用するための **ベイズ統計** の方法に発展する考え方にもなっています。非常に残念なのですが、この解説では詳しく触れないことにします。

1.1.5 一般に統計分析は何をやっているとみなされるか

現代ではコンピュータにデータの値を入力するだけで統計分析の結果を出力してくれる便利な統計ソフトが普及しています。統計ソフトのユーザーは、それが出力してくれる統計分析の結果が

モデル(仮説の集まり)とデータの値の間の関係を定量的に記述した結果

に過ぎないことに常に注意する必要があります。P値、点推定値、区間推定値、ベイズ統計での事後分布とその要約などはすべて

モデル(仮説の集まり)とデータの値の間の関係を定量的に記述した結果

に過ぎません。だから、統計ソフトが出力結果を利用する場合には、

- どのようにして得たどのようなデータの値をモデルと比較したか？
- どのようなモデル(仮説の集まり)とデータの値を比較したか？
- モデル(仮説の集まり)とデータの値のどのような関係を定量化したか？

の3つを理解しておく必要があります。これは実はかなり厳しい要請で、この3つのどれについても十分に理解せずに統計ソフトを使っている人達も結構いるように思われます。この解説の読者はそうならないようにして欲しいと思います。

1.1.6 P値の計算の仕方の詳しい説明

以下では二項係数を

$$\binom{n}{k} = {}_nC_k = \frac{n!}{k!(n-k)!}$$

と表すことにします。例えば、 $0! = 1$ に注意すると、

$$\begin{aligned}\binom{20}{20} &= \frac{20!}{20!0!} = \frac{20!}{20!} = 1, \\ \binom{20}{19} &= \frac{20!}{19!1!} = \frac{20 \cdot 19 \cdot 18 \cdots 2 \cdot 1}{19 \cdot 18 \cdots 2 \cdot 1 \times 1} = 20, \\ \binom{20}{18} &= \frac{20!}{18!2!} = \frac{20 \cdot 19 \cdot 18 \cdot 17 \cdots 2 \cdot 1}{18 \cdot 17 \cdots 2 \cdot 1 \times 2} = 190.\end{aligned}$$

${}_nC_k$ は高校数学では標準的な記号ですが、専門的な文献では $\binom{n}{k}$ の方がよく使われているように思われるので主に $\binom{n}{k}$ を使います。どちらを使うかは好みの問題でしかないので、 $\binom{n}{k}$ という書き方が嫌いな人は ${}_nC_k$ と書いても構いません。

片側P値と呼ばれる

効果がある患者の割合が60%だと仮定したとき、無作為に選んだ患者20人中18人 **以上** に効果がある確率

は次のように計算される。

$$\bullet (20人中k人に効果がある確率) = \binom{20}{k} 0.6^k (1 - 0.6)^{20-k}$$

なので、関数電卓その他を使って計算すると、

- (20人中20人に効果がある確率) = $\binom{20}{20} 0.6^{20} (1 - 0.6)^0 = 0.6^{20} \approx 3.6561584400629733e-5$
- (20人中19人に効果がある確率) = $\binom{20}{19} 0.6^{19} (1 - 0.6)^1 = 20 \cdot 0.6^{19} \cdot 0.4 \approx 0.00048748779200839646$
- (20人中18人に効果がある確率) = $\binom{20}{18} 0.6^{18} (1 - 0.6)^2 = 190 \cdot 0.6^{18} \cdot 0.4^2 \approx 0.003087422682719845$

ここで $3.6561584400629733e-5$ は $3.6561584400629733 \times 10^{-5}$ を意味する。

上の3つの値の和を計算することによって次を得る:

- (片側P値)=(20人中18人以上に効果がある確率) $\approx 0.003611472059128871$

両側P値はこれの2倍なので、

- (両側P値) = 0.007222944118257742

以上によって片側P値は約0.36%で両側P値は約0.72%になることが分かった

```
In [13]: 1 @show binomial(20, 20)
2 @show binomial(20, 19)
3 @show binomial(20, 18)
4 @show p20 = 1 * 0.6^20 * 0.4^0
5 @show p19 = 20 * 0.6^19 * 0.4^1
6 @show p18 = 190 * 0.6^18 * 0.4^2
7 @show pval_one_sided = p20 + p19 + p18
8 @show pval_both_sided = 2pval_one_sided;

binomial(20, 20) = 1
binomial(20, 19) = 20
binomial(20, 18) = 190
p20 = 1 * 0.6 ^ 20 * 0.4 ^ 0 = 3.6561584400629733e-5
p19 = 20 * 0.6 ^ 19 * 0.4 ^ 1 = 0.00048748779200839646
p18 = 190 * 0.6 ^ 18 * 0.4 ^ 2 = 0.003087422682719845
pval_one_sided = p20 + p19 + p18 = 0.003611472059128871
pval_both_sided = 2pval_one_sided = 0.007222944118257742
```

1.2 両側P値の計算練習問題

1.3 二項分布の確率の表の作成

問題:「薬Aが効く患者の割合は60%である」という仮定の下で「薬Aは無作為抽出された20人中 k 人に効いた」というデータの値が得られる確率 $P(k)$ を0以上20以下のすべての整数 k について求め、確率の数表を作成せよ。

解答例: 二項分布の確率 $P(k) = \binom{20}{k} 0.6^k 0.4^{20-k}$ を $k = 0, 1, 2, \dots, 20$ について求めればよい。その結果は次のようになる:

k	P(k)	k	P(k)
0	1.09951e-8	12	0.179706
1	3.29853e-7	13	0.165882
2	4.70041e-6	14	0.124412
3	0.0000423037	15	0.074647
4	0.000269686	16	0.0349908
5	0.00129449	17	0.0123497
6	0.00485435	18	0.00308742
7	0.0145631	19	0.000487488
8	0.0354974	20	0.0000365616
9	0.0709949		
10	0.117142		
11	0.159738		

ただし、1.09951e-8は 1.09951×10^{-8} を意味する。以上の結果はWolframAlphaに

`Table[k, {k, 0, 20}], Table[Binomial[20, k]*0.6^k*0.4^(20-k), {k, 0, 20}]`

と入力することによって得られる([WolframAlphaで実行 \(https://www.wolframalpha.com/input?](https://www.wolframalpha.com/input?i=Table%5Bk%2C+%7Bk%2C+0%2C+20%7D%5D%2C+Table%5BBinomial%5B20%2C+k%5D*0.6%5Ek*0.4%5E%2820-k%29%2C+%7Bk%2C+0%2C+20%7D%5D%5D&lang=ja)

https://www.wolframalpha.com/input?i=Table%5Bk%2C+%7Bk%2C+0%2C+20%7D%5D%2C+Table%5BBinomial%5B20%2C+k%5D*0.6%5Ek*0.4%5E%2820-k%29%2C+%7Bk%2C+0%2C+20%7D%5D%5D&lang=ja)).

Table[k, {k, 0, 20}], Table[Binomial[20, k]*0.6^k*0.4^(20-k), {k, 0, 20}]

自然言語

数式入力

拡張キーボード

例を見る

アップロード

ランダムな例を使う

入力

$\left\{ \text{Table}[k, \{k, 0, 20\}], \text{Table}\left[\binom{20}{k} \times 0.6^k \times 0.4^{20-k}, \{k, 0, 20\}\right] \right\}$

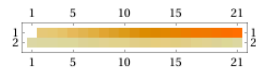
$\binom{n}{m}$ は二項係数です

次元

☒ ステップごとの解説

2 (行) × 21 (列)

行列プロット



転置

☒ ステップごとの解説

$$\begin{pmatrix} 0 & 1.09951 \times 10^{-8} \\ 1 & 3.29853 \times 10^{-7} \\ 2 & 4.70041 \times 10^{-6} \\ 3 & 0.0000423037 \\ 4 & 0.000269686 \\ 5 & 0.00129449 \\ 6 & 0.00485435 \\ 7 & 0.0145631 \\ 8 & 0.0354974 \\ 9 & 0.0709949 \\ 10 & 0.117142 \\ 11 & 0.159738 \\ 12 & 0.179706 \\ 13 & 0.165882 \\ 14 & 0.124412 \\ 15 & 0.074647 \\ 16 & 0.0349908 \\ 17 & 0.0123497 \\ 18 & 0.00308742 \\ 19 & 0.000487488 \\ 20 & 0.0000365616 \end{pmatrix}$$

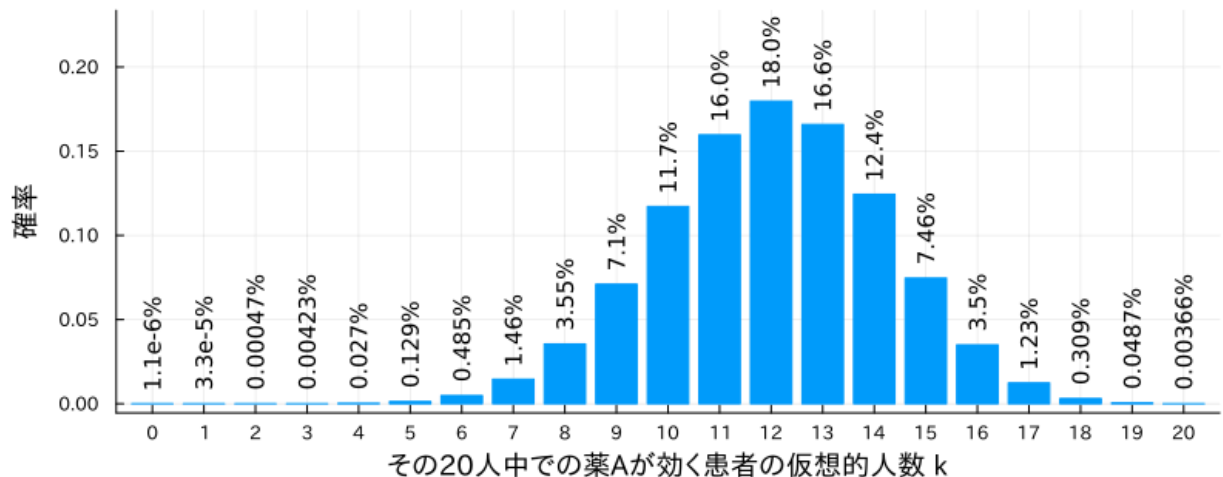
In [14]: 1 Any[0:11 pdf.(Binomial(20, 0.6), 0:11) 12:23 pdf.(Binomial(20, 0.6), 12:23)]

Out[14]: 12x4 Matrix{Any}:

0	1.09951e-8	12	0.179706
1	3.29853e-7	13	0.165882
2	4.70041e-6	14	0.124412
3	4.23037e-5	15	0.074647
4	0.000269686	16	0.0349908
5	0.00129449	17	0.0123497
6	0.00485435	18	0.00308742
7	0.0145631	19	0.000487488
8	0.0354974	20	3.65616e-5
9	0.0709949	21	0.0
10	0.117142	22	0.0
11	0.159738	23	0.0

```
In [15]: 1 n, p = 20, 0.6
2 bin = Binomial(n, p)
3 bar(0:n, k → pdf(bin, k); label="", c=1, lc=:match)
4 plot!(xtick=0:n, xlim=(-0.7, n+0.7), ylim=(-0.03, 1.3).*pdf(bin, mode(bin)))
5 plot!(xguide="その20人中での薬Aが効く患者の仮想的人数 k", yguide="確率")
6 title!("検定仮説が「薬Aが効く患者の割合は60%」で\
7 無作為に20人の患者を選んだ場合の二項確率")
8 for j in 0:n
9     annotate!(j, pdf(bin, j)+0.05pdf(bin, mode(bin)),
10             text(string(round(100pdf(bin, j); sigdigits=3))*"%", :left, 90.0, 10))
11 end
12 plot!(legend=:topleft, size=(800, 350),
13       bottommargin=8Plots.mm, leftmargin=4Plots.mm)
```

Out[15]: 検定仮説が「薬Aが効く患者の割合は60%」で無作為に20人の患者を選んだ場合の二項確率



1.4 無作為抽出された患者20人中6人に効いた場合の両側P値

問題:「薬Aは無作為に抽出された患者20人中6人に効いた」というデータの値に関する「薬Aが効く患者の割合は60%である」という仮説の両側P値を求めよ。ただし、その両側P値は片側確率の2倍で定義されたものとする。その計算で上の問題で作成した確率の数表を用いて良い。

解答例: 求めるべき両側P値は、「薬Aが効く患者の割合は60%である」という仮定の下で「薬Aが効いた人数は無作為抽出された20人中6人以下であった」となる確率の2倍である。ゆえに、求めたい両側P値は、上の問題で求めた確率 $P(k)$ を $k = 0, 1, 2, 3, 4, 5, 6$ について足し合わせて2倍すれば求まる。

```
In [16]: 1 (1.09951e-8 +
2 3.29853e-7 +
3 4.70041e-6 +
4 0.0000423037 +
5 0.000269686 +
6 0.00129449 +
7 0.00485435) * 2
```

Out[16]: 0.012931741916200001

「薬Aは無作為に抽出された患者20人中6人に効いた」というデータの値に関する「薬Aが効く患者の割合は60%である」という仮説の両側P値は約1.3%である。

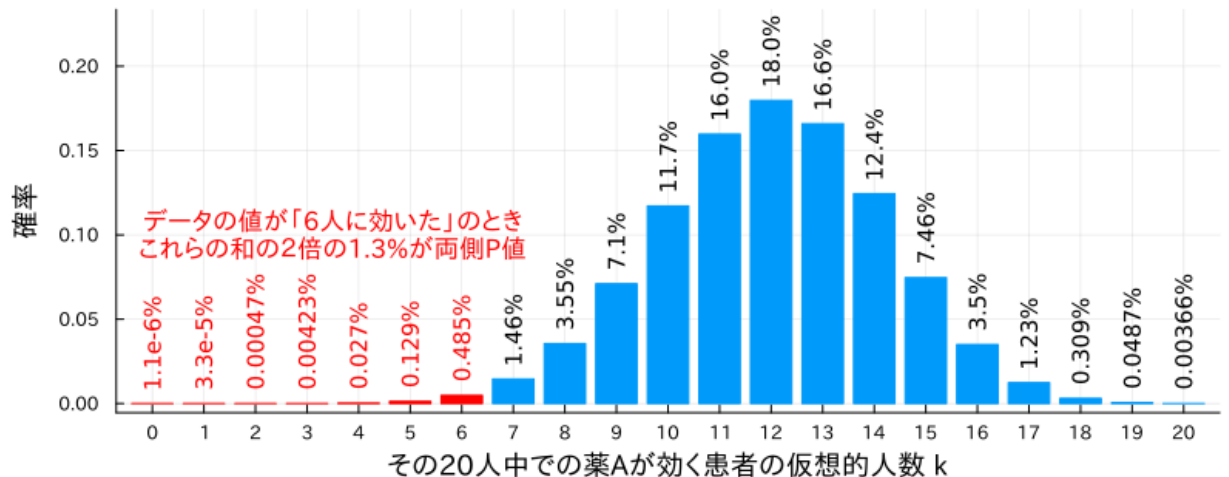
この値はかなり小さめなので、「薬Aが効く患者の割合は60%である」という仮説の下で「薬Aは無作為に抽出された患者20人中6人に効いた」というデータの値はかなり意外だとみなされる。

```
In [17]: 1 pvalue_central(Binomial(20, 0.6), 6)
```

Out[17]: 0.01293175070407002

```
In [18]: 1 n, k, p = 20, 6, 0.6
2 bin = Binomial(n, p)
3 bar(k+1:n, k → pdf(bin, k); label="", c=1, lc=:match)
4 bar!(0:k, k → pdf(bin, k); label="", c=:red, lc=:match)
5 plot!(xtick=0:n, xlim=(-0.7, n+0.7), ylim=(-0.03, 1.3).*pdf(bin, mode(bin)))
6 plot!(xguide="その20人中での薬Aが効く患者の仮想的人数 k", yguide="確率")
7 title!("検定仮説が「薬Aが効く患者の割合は60%」で\
8 無作為に20人の患者を選んだ場合の二項確率")
9 for j in 0:n
10     c = j > k ? :black : :red
11     annotate!(j, pdf(bin, j)+0.05pdf(bin, mode(bin)),
12             text(string(round(100pdf(bin, j); sigdigits=3))*"%", :left, 90.0, 10, c))
13 end
14 annotate!(3.5, 0.1,
15         text("データの値が「6人に効いた」のとき\n\
16             これらの和の2倍の1.3%が両側P値", gothic, 11, :red))
17 plot!(legend=:topleft, size=(800, 350),
18       bottommargin=8Plots.mm, leftmargin=4Plots.mm)
```

Out[18]: 検定仮説が「薬Aが効く患者の割合は60%」で無作為に20人の患者を選んだ場合の二項確率



1.5 無作為抽出された患者20人中15人に効いた場合の両側P値

問題:「薬Aは無作為に抽出された患者20人中15人に効いた」というデータの値に関する「薬Aが効く患者の割合は60%である」という仮説の両側P値を求めよ。ただし、その両側P値は片側確率の2倍で定義されたものとする。その計算で上の問題で作成した確率の数表を用いて良い。

解答例: $P(k)$ の $k = 15, 16, 17, 18, 19, 20$ に関する和の2倍を計算すれば良い。

```
In [19]: 1 (0.074647 +
2 0.0349908 +
3 0.0123497 +
4 0.00308742 +
5 0.000487488 +
6 0.0000365616) * 2
```

Out[19]: 0.25119793920000005

「薬Aは無作為に抽出された患者20人中15人に効いた」というデータの値に関する「薬Aが効く患者の割合は60%である」という仮説の両側P値は約25%である。

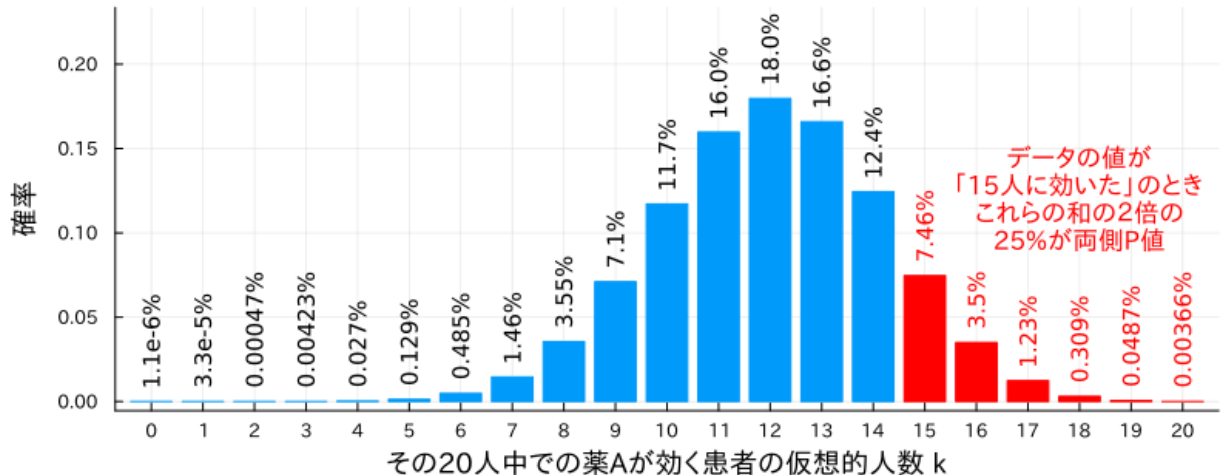
この25%のP値はかなり大きな値なので、「薬Aが効く患者の割合は60%である」という仮説の下で「薬Aは無作為に抽出された患者20人中15人に効いた」というデータの値をほとんど意外ではないと考えられる。

```
In [20]: 1 pvalue_central(Binomial(20, 0.6), 15)
```

Out[20]: 0.2511979454460751

```
In [21]: 1 n, k, p = 20, 15, 0.6
2 bin = Binomial(n, p)
3 bar(0:k-1, k → pdf(bin, k); label="", c=1, lc=:match)
4 bar!(k:n, k → pdf(bin, k); label="", c=:red, lc=:match)
5 plot!(xtick=0:n, xlim=(-0.7, n+0.7), ylim=(-0.03, 1.3).*pdf(bin, mode(bin)))
6 plot!(xguide="その20人中での薬Aが効く患者の仮想的人数 k", yguide="確率")
7 title!("検定仮説が「薬Aが効く患者の割合は60%」で\
8 無作為に20人の患者を選んだ場合の二項確率")
9 for j in 0:n
10     c = j < k ? :black : :red
11     annotate!(j, pdf(bin, j)+0.05pdf(bin, mode(bin)),
12             text(string(round(100pdf(bin, j); sigdigits=3))*"%", :left, 90.0, 10, c))
13 end
14 annotate!(18, 0.12,
15         text("データの値が\n「15人に効いた」のとき\n\
16             これらの和の2倍の\n25%が両側P値", gothic, 11, :red))
17 plot!(legend=:topleft, size=(800, 350),
18       bottommargin=8Plots.mm, leftmargin=4Plots.mm)
```

Out[21]: 検定仮説が「薬Aが効く患者の割合は60%」で無作為に20人の患者を選んだ場合の二項確率



1.6 S値(=意外度)のコイン投げによる定義

以下では、「無作為に選んだ患者 n 人中 k 人に薬Aが効いた」というデータの値に関する「薬Aが効く患者の割合は p である」という仮説の(両側)P値を以下では P と書くことにします。

P値 P の値が小ささは、「薬Aが効く患者の割合は p である」という仮説を信じている人が「無作為に選んだ患者 n 人中 k 人に薬Aが効いた」というデータの値をどれだけ意外だと感じるかを表すのでした。

意外さの程度は偏りが無いコイン投げに例えると分かり易くなります。

例えば、偏りのないコイン投げで4回続けて表が出る確率は $6.25\% \approx 6\%$ である。だから、 $P \approx 6\%$ であることは、

「薬Aが効く患者の割合は p である」と信じている人が「無作為に選んだ患者 n 人中 k 人に薬Aが効いた」というデータの値について感じる意外さ

が

偏りが無いコイン投げで 4 回続けて表が出たのと同じ程度の意外さ

だとみなされることを意味していると考えられます。

同様に $P \approx 3\%$ であることは、上と同じ意味での意外さが

偏りが無いコイン投げで 5 回続けて表が出たのと同じ程度の意外さ

だとみなされることを意味していると考えられます。

便宜的にコイン投げの回数として非整数も許されることにすれば、 $P \approx 5\%$ であることは、上と同じ意味での意外さが

偏りが無いコイン投げで 4.3 回続けて表が出たのと同じ程度の意外さ

だとみなされることを意味していると考えられます。実際、

$$(1/2)^{4.3} \approx 0.05.$$

より正確には $(1/2)^{4.3} \approx 0.050765774772264724$.

P値 P に対して以上のようにして決まる偏りが無いコイン投げで連続して表が出る回数を S は次のように得られます:

$$(1/2)^S = P \quad \text{すなわち} \quad S = -\log_2 P.$$

この S はP値 P に対応する **S値 (意外度, surprisal value)** と呼ばれています([Rafi-Greenland 2020]).

S値は曖昧な「P値の小ささ」=「意外さ」という言い方で語られる概念の定量化だと考えられます。つまり、S値の大きさは「意外さ」の大きさそのものと解釈されます。

P値とS値の対応は以下の通りです。

S値 = 意外度	0	1	2	3	4	5	6	7	8	9	10
P値	100%	50%	約25%	約12%	約6%	約3%	約1.5%	約0.8%	約0.4%	約0.2%	約0.1%

情報理論的にS値の単位は **ビット** だと考えられます。例えば、P値が6%程度になった場合には、S値は約4ビットになるので、

データの値によってモデルと検定仮説の組み合わせの否定について4ビット程度の情報が得られた。

のように考えることになります。

P値の違いの大きさの違いは、対応するS値の側で何ビット違うかを見ると誤解し難くなります。

例えば、6% と 3% のP値はそれぞれ約 4 と 5 のS値に対応し、1ビットの違いしかありません。

P値については伝統的に 5% の閾値が採用されることが多く、 $P < 5\%$ であるか否かによって正反対の結論を出す悪しき慣習があるのですが、6% と 3% のP値はS値で見たら1ビットの違いしかありません。

$P < 5\%$ であるか否かによって正反対の結論を出すことの不合理さもS値を理解すれば分かります。

1.7 P値に関する広く通用する閾値は存在しない

$P < 5\%$ であるか否かによる「有意差あり/なし」の二分法に強く執着している人達は、P値に閾値をどうしても設けたいと感じるかもしれません。

しかし、仮に閾値の設定が有効な道具として機能するとしても、ケースバイケースで適切な閾値は異なると考えられます。

検定仮説が諸々の理由で非常に確からしい場合には、その否定のためのP値の閾値は非常に小さくすることが妥当だと考えられます。逆に検定仮説が非常に疑わしい場合にはその否定のためのP値の閾値は10%のような大きめの値にしても十分に合理的になることがあり得ます。

さらに統計分析の結果を用いた意思判断においても、場合ごとに許容できるリスクの大きさと種類は異なります。

以上のような諸々の事情をすべて無視して慣用的な閾値を機械的に採用して押し通そうとすることは不合理な行為だと考えられます。

つまり、誰にでもP値に関する閾値を設ける権利はあるが、5%や1%のような慣用的な閾値を機械的に使うことは不合理であるということです。あなたが閾値を設定した場合にはその責任をあなたが負う必要が出て来る。

統計分析の結果を参考にしながら現実における意思決定を行う主体(例えば臨床医や政策決定責任者など)と統計分析を行うあなたが異なる人物であるならば、閾値の設定は意思決定を行う人にまかせて、意思決定者に責任を取ってもらうのが良いとも考えられます。

実際にそれが正しい場合にあなたは、データの取得法の詳細と有力なモデル(仮説の集まり)の概略とそれらの関係の定量化(P値や点推定や区間推定はその典型例)を意思決定の現場にいる人達にとって分かり易い形式で報告すれば良いでしょう(より詳しくは [McShane et al. 2024] のガイドラインを参照)。

1.8 相性の良さ/悪さという解釈の仕方

「無作為に選んだ患者 n 人中 k 人に薬Aが効いた」というデータの値に関する「薬Aが効く患者の割合は p である」という仮説の(両側)P値 P の小ささとそれに対応するS値 $S = -\log_2 P$ は

「薬Aが効く患者の割合は p である」という仮説を信じている人が「無作為に選んだ患者 n 人中 k 人に薬Aが効いた」というデータの値に感じる意外度

という意味を持っているのでした。これと同じ意味で、P値の小ささ(=S値の大きさ)を

二項分布モデルとそのパラメータに関する「薬Aが効く患者の割合は p である」という検定仮説と「無作為に選んだ患者 n 人中 k 人に薬Aが効いた」というデータの値の **相性の悪さ (incompatibility)** の尺度

と解釈することになります。これは大小関係を逆転させると、P値の大きさ(=S値の小ささ)は

二項分布モデルとそのパラメータに関する「薬Aが効く患者の割合は p である」という検定仮説と「無作為に選んだ患者 n 人中 k 人に薬Aが効いた」というデータの値の **相性の良さ (compatibility)** の尺度

と解釈されるということを意味しています。

以上の「相性の良さ/悪さ」という言い方も便利なのでこれから使っていくことにします。

Compatibility (相性の良さ)は最重要キーワード: "compatibility" (相性の良さ)は "significance" (有意性)や "confidence" (信頼)のよ
うな伝統的だがミスリーディングな用語の置き換え先として有力視されている単語です ([[ASA声明2016](#)], [[Greenland et al. 2016](#)], [[Gelman-Greenland 2019](#)], [[Amrhein-Greenland-McShane 2019](#)], [[Rafi-Greenland 2020](#)], [[Amrhein-Greenland 2022](#)],

1.9 P値関数とS値関数

「無作為に選んだ患者 n 人中 k 人に薬Aが効いた」というデータの値に関する「薬Aが効く患者の割合は p である」という仮説
の(両側)P値 P は k, n, p の関数なので、それを

$$\text{pvalue}(k|n, p) = (k, n, p \text{ に関する両側P値})$$

と書くことにします。

データの値 k, n が与えられたとき、P値 $\text{pvalue}(k|n, p)$ は p だけの関数になる。そのとき p にP値 $\text{pvalue}(k|n, p)$ を対応させる関数

$$p \mapsto \text{pvalue}(k|n, p)$$

は **P値関数 (P-value function)** と呼ばれています(日本語の文献では例えば [[ロスマンの疫学](#)], [[佐藤俊哉2024](#)] にP値関数の説明がある)。

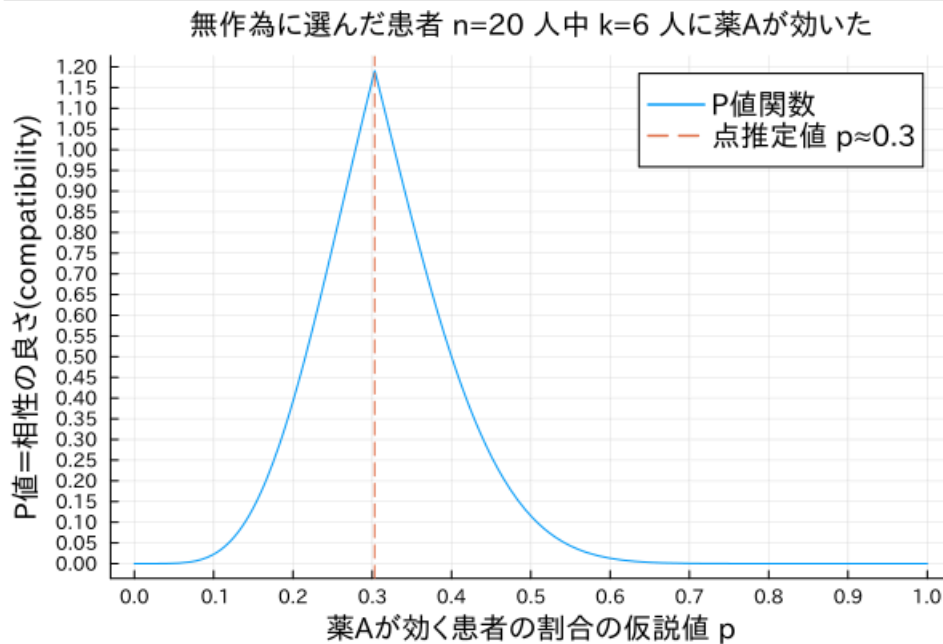
例題1.1の状況では、検定仮説が「薬Aが効く患者の割合は $p = 0.6$ である」に固定されていました。パラメータ p の仮説値を固定せずに動かしながらP値を計算して、P値関数のグラフを描けば、パラメータ p のすべての仮説値と「無作為に選んだ患者 n 人中 k 人に薬Aが効いた」というデータの値の相性の良さ(compatibility)の様子が分かることになります。

しかし、P値関数のグラフを得るためには、コンピュータでグラフを描くスキルが必要になります。以下では、この解説の筆者がコンピュータで作成したP値関数のグラフの例を示しましょう。

次のセルのコードで描かれたのは「無作為に選んだ患者 $n = 20$ 人中 $k = 6$ 人に薬Aが効いた」というデータに対応するP値関数のグラフです。


```
In [22]: 1 n, k = 20, 6
2 plot(p → pvalue_central(Binomial(n, p), k), 0, 1; label="P値関数")
3 plot!(xtick=0:0.1:1, ytick=0:0.05:1.5, yguide="P値 = 相性の良さ (compatibility)")
4 title!("無作為に選んだ患者 n=$n 人中 k=$k 人に薬Aが効いた")
5 plot!(xguide="薬Aが効く患者の割合の仮説値 p")
6 vline!([0.303]; label="点推定値 p≈0.3", ls=:dash)
```

Out[22]:



このP値の大きさを見れば、薬Aが効く患者の割合のすべての仮説値 p と「無作為に選んだ患者 $n = 20$ 人中 $k = 6$ 人に薬Aが効いた」というデータの値の相性の良さが分かります。

例えば、このP値関数の最大値は $p \approx 0.3$ で与えられるので、「無作為に選んだ患者 $n = 20$ 人中 $k = 6$ 人に薬Aが効いた」というデータの値と最も相性が良い検定仮説は「薬Aが効く患者の割合は $p \approx 0.3$ である」であることが分かります。

このようにP値関数を最大化するパラメータの値 $p \approx 0.3$ は(P値関数から得られる)パラメータ p の **点推定値 (point estimate)** と呼ばれます。

P値関数が最大になる点推定値 $p \approx 0.3$ から p が離れれば離れるほどP値は小さくなるので、そのような p の仮説値とデータの値の相性は悪くなります。

例えば、「薬Aが効く患者の割合は $p = 0.05$ である」や「薬Aが効く患者の割合は $p = 0.65$ である」という検定仮説は、「無作為に選んだ患者 $n = 20$ 人中 $k = 6$ 人に薬Aが効いた」というデータの値と相性がかなり悪いことが上のグラフから分かります。

P値関数のグラフの欠点はその値が 0 に近い所の様子が見難いことです。

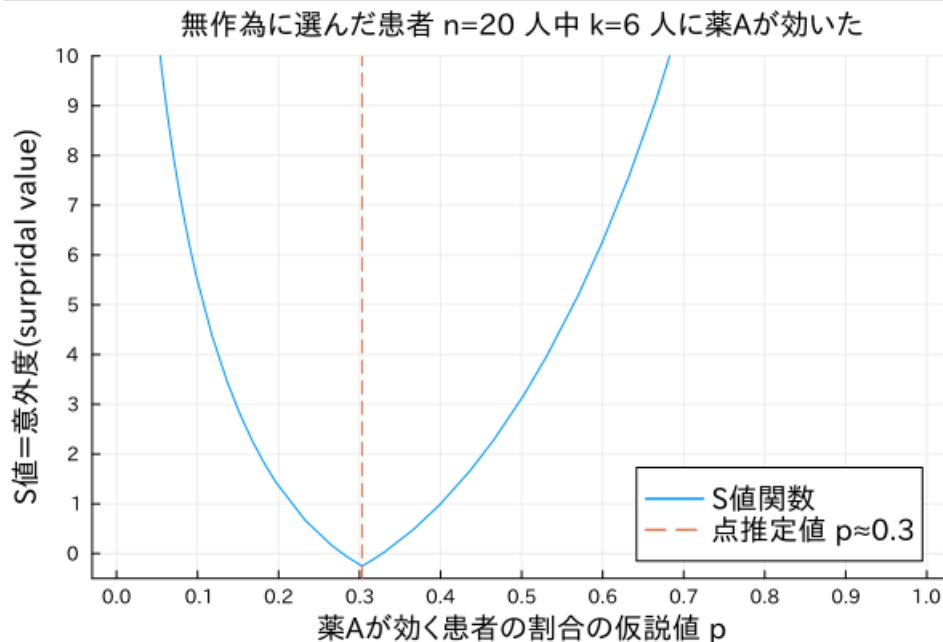
P値関数の代わりに同様に定義されたS値関数

$$p \mapsto -\log_2 \text{pvalue}(k|n, p)$$

のグラフを描くとP値が 0 に近い所の様子もよく見えるようになります。


```
In [23]: 1 n, k = 20, 6
2 plot(p → -log(2, pvalue_central(Binomial(n, p), k)), 0, 1; label="S値関数")
3 plot!(ylim=(-0.5, 10), legend=:bottomright)
4 plot!(xtick=0:0.1:1, ytick=0:10, yguide="S値 = 意外度(surpridal value)")
5 title!("無作為に選んだ患者 n=$n 人中 k=$k 人に薬Aが効いた")
6 plot!(xguide="薬Aが効く患者の割合の仮説値 p")
7 vline!([0.303]; label="点推定値 p≈0.3", ls=:dash)
```

Out[23]:



意外さが大きいことと相性が悪いことは同じ意味になり、S値で表される意外さは偏りのないコイン投げで何回続けて表が出たかを意味しているのです。

便宜的に偏りのないコイン投げで5回以上連続して表が出たときに「意外度が非常に大きい」＝「相性が非常に悪い」ということにすると、 p が 0.107 以下の場合と 0.564 以上の場合に「意外度が非常に大きい」＝「相性が非常に悪い」ということになります。(このとき、区間 [0.107, 0.564] は 97% 信頼区間＝相性区間になっている。信頼区間＝相性区間については次節を参照せよ。)

ただし、ここで考えているのは「無作為に選んだ患者 $n = 20$ 人中 $k = 6$ 人に薬Aが効いた」というデータの値と「薬Aが効く患者の割合は p である」という仮説の相性の悪さです。

```
In [24]: 1 @show α = 0.5^5
2 @show 1 - α
3 @show ci_clopper_pearson(n, k; α);

α = 0.5 ^ 5 = 0.03125
1 - α = 0.96875
ci_clopper_pearson(n, k; α) = [0.10726583473679588, 0.564099648860263]
```

1.10 相性区間 = CI = 信頼区間

前節でP値関数から点推定値がP値関数を最大化するパラメータ p の値として得られることを説明しました。

パラメータ p の区間推定もP値関数から得られます。

そのためにはP値に関する閾値 α を用意して、 $pvalue(k|n, p) \geq \alpha$ か否かで、パラメータ p の値はデータの値 n, k と相性が良い悪いとみなすことにします。

伝統的な用語法で閾値 α は有意水準と呼ばれ、 $1 - \alpha$ は信頼水準(信頼度、信頼係数)と呼ばれています。しかし、それらの伝統的用語法はP値がモデルとパラメータの値に関する仮説の組み合わせとデータの値の相性の良さ(compatibility)の尺度であるという見方と整合的ではないので、 α を **相性水準 (compatibility level)** と呼ぶことにします。

相性水準と呼ぶことにした閾値 α によってデータの値と相性が良いとみなされるパラメータ p の値全体の集合をパラメータ p の **相性区間 (compatibility interval)** と呼ぶことにします。すなわち、相性区間を $ci(k|n, \alpha)$ と書くことにすると、

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

一般にP値関数から相性区間(＝信頼区間)をこのようにして定義できます。

伝統的な用語法でこれは信頼水準 $1 - \alpha$ の信頼区間(もしくは $100(1 - \alpha)\%$ 信頼区間)と呼ばれるものと同じものになります。信頼区間は英語では confidence interval というので、相性区間も信頼区間も英語で略すと同じ CI になります。

相性区間 = (compatibility interval) = CI = (confidence interval) = 信頼区間

我々が現在扱っている確率関数から得られる相性区間は $0.11893159040572766, 0.5427891822762891$ の信頼区間と呼ばれる非常に有名な信頼区間と同じ

用語法に関する注意: 2025年現在、compatibility interval (相性区間)という用語はある程度以上のレベルの研究者達にはよく知られていますが、まだ十分に普及しているとは言えません。まだ十分に普及していない用語を使いたくない人は、confidence interval (信頼区間)という伝統的な用語を使い続けても構いません。しかし、"confidence" という英単語と「信頼」という日本語の持つ語感が原因で、「95%信頼区間には95%という高い信頼度で真の値が含まれている」と安易に信じて良いかのような誤解が蔓延しています。例えば使っているモデルの妥当性を保証できない場合にはそのように信じることは誤りになります。だから、confidence interval や信頼区間という用語を使う場合には、蔓延している誤解に自分自身や報告の読者が陥らないように十分な注意を払うべきです。

Clopper-Pearsonの相性区間について: Clopper-Pearsonの相性区間にはコンピュータでの効率的な計算に適した明示公式が存在し、その公式を使って実装すると極めて簡単にかつ効率的に計算できます。

ただし、その事実を理解するためには二項分布とベータ分布の累積分布関数間の関係やベータ分布の累積分布関数が正規化不完全ベータ関数と呼ばれる有名な特殊関数にちょうどっており、その逆関数の分位点関数もコンピュータで効率的に計算できることなどを学ぶ必要があります。数学的な事柄にも興味がある人は勉強してみると数学とコンピュータでの数値計算について沢山の教養が身に付くことでしょう。

この解説ではその点に深入りせずに、次の非常にシンプルなコードでClopper-Pearsonの相性区間を計算できることを注意するにとどめておきます。

```
using Distributions
function ci_clopper_pearson(n, k;  $\alpha$  = 0.05)
    p_L = k > 0 ? quantile(Beta(k, n-k+1),  $\alpha/2$ ) : zero( $\alpha$ )
    p_U = k < n ? quantile(Beta(k+1, n-k), 1- $\alpha/2$ ) : one( $\alpha$ )
    [p_L, p_U]
end
```

このコードを用いて、「無作為に選んだ患者 $n=20$ 人中 $k=6$ 人に薬Aが効いた」というデータの値が得られたときの、相性水準 5% のClopper-Pearsonの相性区間(= 信頼水準 95% のClopper-Pearson信頼区間)を求めてみましょう。

In [25]: 1 @show ci_clopper_pearson(20, 6; α = 0.05);

```
ci_clopper_pearson(20, 6;  $\alpha$  = 0.05) = [0.11893159040572766, 0.5427891822762891]
```

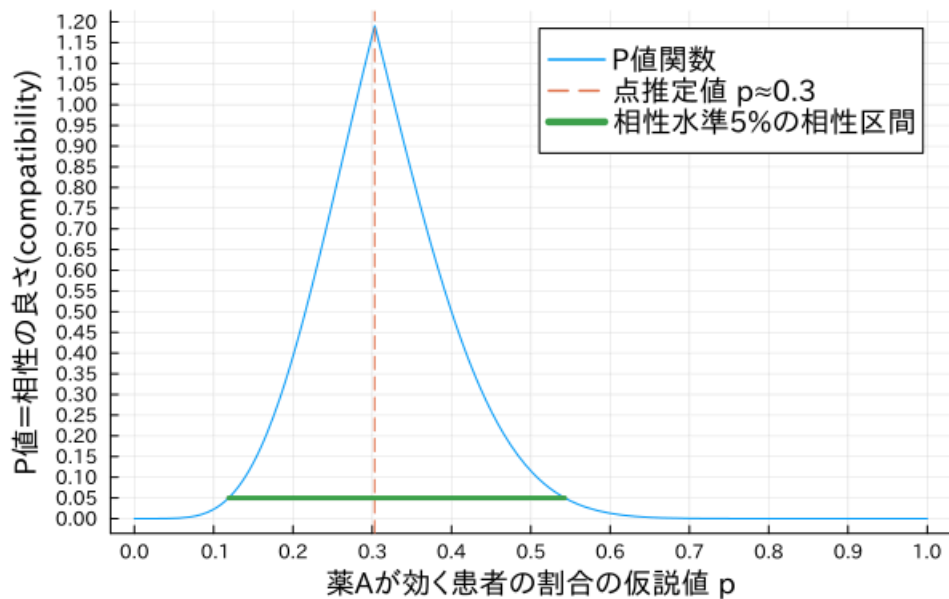
「無作為に選んだ患者 $n=20$ 人中 $k=6$ 人に薬Aが効いた」というデータの値が得られたときの、相性水準 5% の相性区間(= 信頼水準 95% のClopper-Pearson信頼区間)はおおよそ [0.119, 0.543] になることが分かりました。

相性水準 5% の相性区間は、P値に関する 5% 以上という条件によって、「無作為に選んだ患者 $n = 20$ 人中 $k = 6$ 人に薬Aが効いた」というデータの値と相性が良いとみなされた「薬Aが効く患者の割合の仮説値 p 」全体の集合になっています。

相性水準 5% の相性区間はP値関数のグラフを高さ 5% で切断する線分に等しい。次のグラフを参照せよ。

```
In [26]: 1 n, k, α = 20, 6, 0.05
2 ci = ci_clopper_pearson(n, k; α);
3 plot(p → pvalue_central(Binomial(n, p), k), 0, 1; label="P値関数")
4 plot!(xtick=0:0.1:1, ytick=0:0.05:1.5, yguide="P値 = 相性の良さ (compatibility)")
5 title!("データの値「無作為に選んだ患者 n=$n 人中 k=$k 人に薬Aが効いた」")
6 plot!(xguide="薬Aが効く患者の割合の仮説値 p")
7 vline!([0.303]; label="点推定値 p≈0.3", ls=:dash)
8 plot!(ci, fill(α, 2); label="相性水準5%の相性区間", lw=3)
```

Out[26]: データの値「無作為に選んだ患者 n=20 人中 k=6 人に薬Aが効いた」

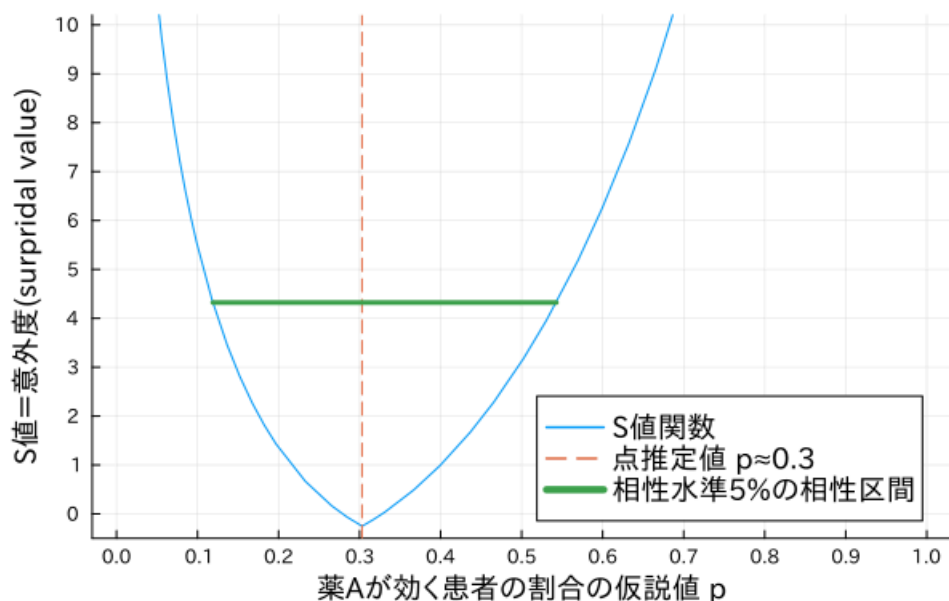


このように、P値(より正確にはP値関数)の概念から、点推定値と区間推定の両方が得られる。

S値のグラフにも相性区間を描きこむと次のようになる。相性水準 5% の相性区間は、S値関数のグラフを高さ $-\log_2 0.05 \approx 4.3$ で切断する線分になる。

```
In [27]: 1 n, k = 20, 6
2 plot(p → -log(2, pvalue_central(Binomial(n, p), k)), 0, 1; label="S値関数")
3 plot!(ylim=(-0.5, 10.2), legend=:bottomright)
4 plot!(xtick=0:0.1:1, ytick=0:10, yguide="S値 = 意外度 (surpridal value)")
5 title!("データの値「無作為に選んだ患者 n=$n 人中 k=$k 人に薬Aが効いた」")
6 plot!(xguide="薬Aが効く患者の割合の仮説値 p")
7 vline!([0.303]; label="点推定値 p≈0.3", ls=:dash)
8 plot!(ci, fill(-log(2, α), 2); label="相性水準5%の相性区間", lw=3)
```

Out[27]: データの値「無作為に選んだ患者 n=20 人中 k=6 人に薬Aが効いた」



1.11 報告の仕方について

P値関数やS値関数のグラフは、各検定仮説のP値の情報だけではなく、点推定値と区間推定値の情報も含まれます。だから、原理的にはP値関数のグラフを報告に載せれば特定の検定仮説のP値や点推定値や区間推定値を報告する必要はないと考えられます。

しかし、報告の読者の便のためには、重要だと考えられる検定仮説達のP値と点推定値と区間推定値も報告に載せた方が親切でしょう。グラフから可能な限り正確な数値を読みとる手間も省けます。

統計学ユーザー界には「有意差あり/なし」と報告したがる伝統的な悪習があるのですが、それを真似する必要はありません。英語の報告では significant の語の使用を控えることは現代的な良いスタイルだということになります。P値の具体的な値を報告していれば、読者の各々が自分の目的に適切な閾値を設定して二分法的な判断を自分でできます。

「有意差あり($P < 0.05$)」と「有意差なし($P \geq 0.05$)」のように報告には書くべきではありません。報告を書く側はP値以外のあらゆる情報を加味して暫定的な結論を述べるべきです。

統計学ユーザー界には「差はゼロ」や「効果はゼロ」を意味する特殊な検定仮説(それを **ゼロ仮説** と呼ぶことにする)のP値のみを報告する悪しき伝統があります。その悪しき伝統は上で示したようにP値が推定の道具としても使えることを見逃す原因になっています。そういう悪習にも従う必要はありません。

例えば、「効果はゼロ」を意味するゼロ仮説のP値だけではなく、事前に予想していた効果の大きさやそれまでの定説の値 a について「効果は a である」という検定仮説のP値も計算して報告するのは良いスタイルです。

より詳しいガイドラインが論文 [\[McShane et al. 2024\]](#) にあります。

2 二項検定の複数種類のP値関数のグラフ

このノートブックのずっと上の方で以下の4種類のP値関数を定義している。

- Clopper-Pearsonの相性区間([\[Clopper-Pearson 1934\]](#))を与える片側確率の2倍版のP値関数 (central法)
- Sterneの相性区間([\[Sterne 1954\]](#))を与えるP値関数 (minlike法)
- Wilsonのスコア相性区間([\[Wilson 1927\]](#))を与える二項分布の正規分布を使うP値関数 (スコア法)
- Waldの相性区間を与えるP値関数 (Wald法 [\[Wald 1939\]](#))

それぞれ以下のような特徴を持ち、それらは互いに異なる利点と欠点を持っています。

- **central法**: P値関数は左右非対称で部分的に値が1より大きくなり、ベータ分布の累積分布関数(正規化不完全ベータ関数)を用いた明示公式を持つ。片側P値と整合的。 α エラー率が確実に α 以下になるが、P値が無駄に大きくなり過ぎる傾向があり、これに対応するClopper-Pearsonの相性区間の幅も無駄に広くなり過ぎる傾向がある(つまり過剰に保守的 (conservative)である)。これに対応するClopper-Pearsonの相性区間はベータ分布の分位点関数(正規化不完全ベータ関数の逆関数)を用いた効率的な明示公式を持つ。
- **minlike法**: P値関数は左右非対称な不連続関数になり、効率的な明示公式がないので計算が面倒。 α エラー率が確実に α 以下になるが、central法と違って過剰な保守性を持たない。対応するSterne相性区間は、計算が面倒なだけではなく、1つの区間にならずに複数の区間に分かれる場合が出て来る。
- **スコア法**: P値関数は左右非対称で計算は易しい。 α エラー率が α でよく近似されるが、 α エラー率が α より少し大きくなる場合も出て来る。これに対応するWilsonのスコア相性区間の計算では二次方程式を解く必要がある。
- **Wald法**: P値関数は左右対称(対称であるべきではない)で計算は易しい。0 または 1 に近いパラメータ値 p でのP値の誤差が大きくなり易い。これに対応するWaldの相性区間の計算も易しい。

[\[Agresti-Coull 1998\]](#) による数値実験ではWilsonのスコア相性区間が優れているという結果が得られおり、計算が易しいWaldの相性区間のシンプルな補正(Agresti-Coull区間と呼ばれている)によってWilsonのスコア相性区間をよく近似できることが示されています。

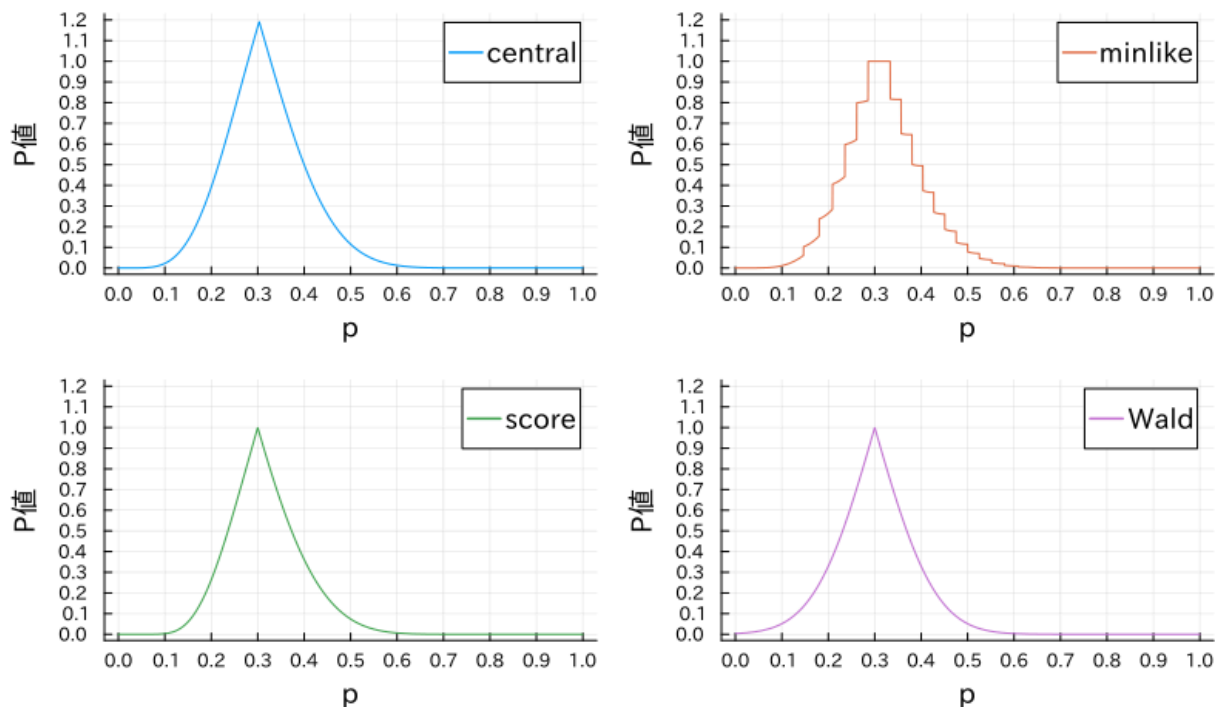
「無作為に選んだ患者 $n=20$ 人中 $k=6$ 人に薬Aが効いた」というデータの値に関するこれらのP値関数達のグラフを見てみよう。

In [28]:

```
1 n, k = 20, 6
2 P1 = plot(p → pvalue_central(Binomial(n, p), k); label="central", c=1)
3 P2 = plot(p → pvalue_minlike(Binomial(n, p), k); label="minlike", c=2)
4 P3 = plot(p → pvalue_score(Binomial(n, p), k); label="score", c=3)
5 P4 = plot(p → pvalue_wald(Binomial(n, p), k); label="Wald", c=4)
6 plot(P1, P2, P3, P4; size=(800, 500), layout=(2, 2))
7 plot!(xtick=0:0.1:1, ytick=0:0.1:1.2, ylim=(-0.03, 1.23))
8 plot!(xguide="p", yguide="P値")
9 plot!(plot_title="「無作為に選んだ患者n=$(n)人中k=$(k)人に薬Aが効いた」\
10   というデータの値に関するP値関数達")
11 plot!(leftmargin=4Plots.mm, bottommargin=4Plots.mm)
```

Out[28]:

「無作為に選んだ患者n=20人中k=6人に薬Aが効いた」というデータの値に関するP値関数達



$n = 20, k = 6$ のような小さなデータでは上のグラフのよう4種のP値関数がかなり異なります。しかし、どれを見ても、 $n = 20, k = 6$ のデータの値との相性の良さを表すP値が0より少し大きくなるパラメータ p の範囲はおおむね0.1から0.6の範囲になることがわかります。 $n = 20, k = 6$ のような小さなデータではP値関数のグラフの幅は大きくなる。

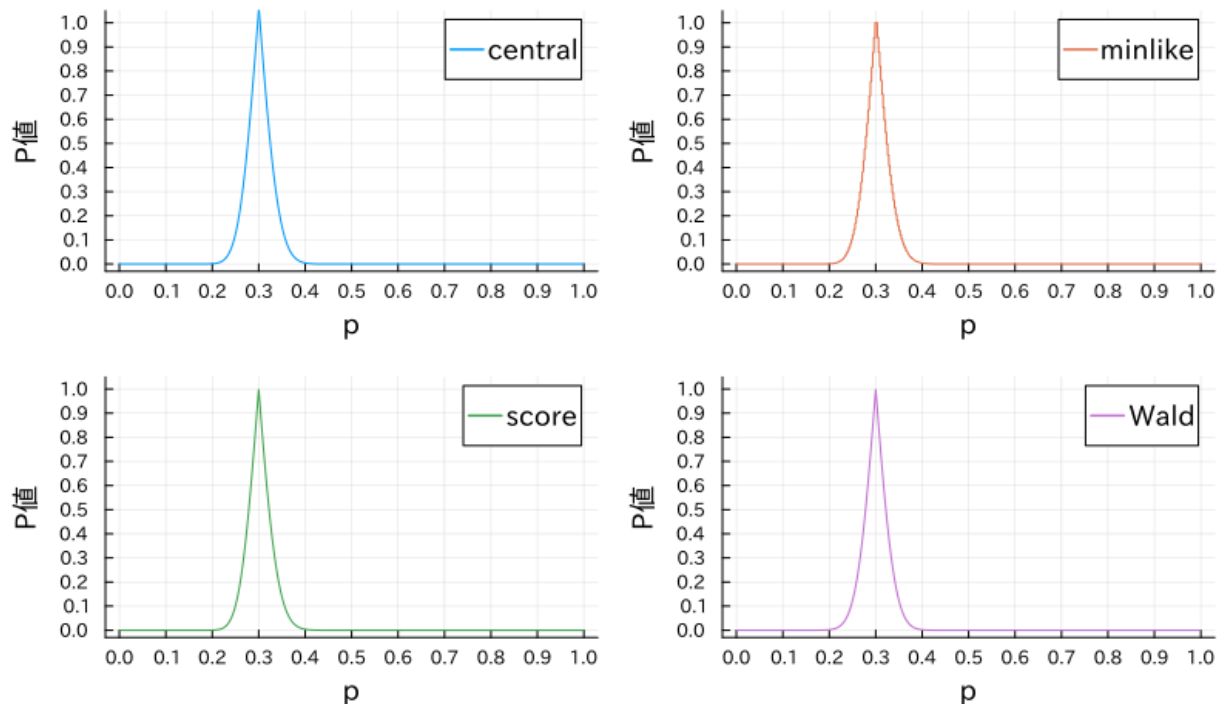
$n = 200, k = 60$ の場合には以下のように4種のP値関数のグラフの違いは目立たなくなります。上のグラフと比較するとP値関数のグラフの幅が $n = 20, k = 6$ の場合から $n = 200, k = 60$ の場合でおよそ3分の1程度に狭くなっています。標本サイズを大きくするとP値関数の幅は標本サイズの平方根に反比例して小さくなります。


```

1 n, k = 200, 60
2 P1 = plot(p → pvalue_central(Binomial(n, p), k); label="central", c=1)
3 P2 = plot(p → pvalue_minlike(Binomial(n, p), k); label="minlike", c=2)
4 P3 = plot(p → pvalue_score(Binomial(n, p), k); label="score", c=3)
5 P4 = plot(p → pvalue_wald(Binomial(n, p), k); label="Wald", c=4)
6 plot(P1, P2, P3, P4; size=(800, 500), layout=(2, 2))
7 plot!(xtick=0:0.1:1, ytick=0:0.1:1.5, ylim=(-0.03, 1.05))
8 plot!(xguide="p", yguide="P値")
9 plot!(plot_title="「無作為に選んだ患者n=$(n)人中k=$(k)人に薬Aが効いた」\
10   というデータの値に関するP値関数達")
11 plot!(leftmargin=4Plots.mm, bottommargin=4Plots.mm)

```

Out[29]: 「無作為に選んだ患者n=200人中k=60人に薬Aが効いた」というデータの値に関するP値関数達



このグラフを見れば、異なる方法を使っても、標本サイズが十分に大きいならば、どれを使っても大差ないことが分かります。

3 参考文献

- [illegible]

- [Rafi-Greenland 2020] <https://scholar.google.co.jp/scholar?cluster=591718439647960929> (<https://scholar.google.co.jp/scholar?cluster=591718439647960929>).
- [ロスマンの疫学2013] https://shinoharashinsha.co.jp/?page_id=3534 (https://shinoharashinsha.co.jp/?page_id=3534).
- [佐藤俊哉2024] https://www.asakura.co.jp/detail.php?book_code=12297 (https://www.asakura.co.jp/detail.php?book_code=12297).
- [Sterne 1954] <https://scholar.google.co.jp/scholar?cluster=18420119515180283098> (<https://scholar.google.co.jp/scholar?cluster=18420119515180283098>).
- [Wald 1939] <https://scholar.google.co.jp/scholar?cluster=1375947589172221492> (<https://scholar.google.co.jp/scholar?cluster=1375947589172221492>).
- [Wilson 1927] <https://scholar.google.co.jp/scholar?cluster=9907942007440998358> (<https://scholar.google.co.jp/scholar?cluster=9907942007440998358>).