P-Value Calculators for Normal, χ^2, t and F Distribution

Yuk Tung Liu

2018-01-27

1 Probability Distributions and P-Values

The probability density functions (pdf's) for the four distributions are:

$$f_{\text{normal}}(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \quad , \quad x \in (-\infty, \infty) \tag{1}$$

$$f_{\chi^2}(x;k) = \frac{1}{2^{k/2}\Gamma(k/2)} x^{k/2-1} e^{-x/2} \quad , \quad x \in (0,\infty) \text{ if } k=1, \text{ otherwise } x \in [0,\infty) \tag{2}$$

$$f_t(x;k) = \frac{\Gamma\left(\frac{k+1}{2}\right)}{\sqrt{k\pi}\,\Gamma(k/2)} \left(1+\frac{x^2}{k}\right)^{-\frac{k+1}{2}} \quad , \quad x \in (-\infty,\infty) \tag{3}$$

$$f_F(x;k_1,k_2) = \frac{1}{B\left(\frac{k_1}{2},\frac{k_2}{2}\right)} \left(\frac{k_1}{k_2}\right)^{\frac{k_1}{2}} x^{\frac{k_1}{2}-1} \left(1+\frac{k_1}{k_2}x\right)^{-\frac{k_1+k_2}{2}} , \; x \in (0,\infty) \text{ if } k_1 = 1, \text{ otherwise } x \in [0,\infty) \tag{4}$$

Here Γ is the gamma function and B is the beta function. I only consider the case in which the degree of freedom parameters k, k_1 and k_2 are positive integers, even though the functions are still well-defined when these parameters are non-integers.

The corresponding cumulative distribution functions (cdf's) are:

$$F_{\text{normal}}(x) = \frac{1}{2}\left[1+\text{erf}\left(\frac{x}{\sqrt{2}}\right)\right] = 1 - \frac{1}{2}\text{erfc}\left(\frac{x}{\sqrt{2}}\right) \tag{5}$$

$$F_{\chi^2}(x;k) = P\left(\frac{k}{2},\frac{x}{2}\right) = 1 - Q\left(\frac{k}{2},\frac{x}{2}\right) \tag{6}$$

$$F_t(x;k) = 1 - \frac{1}{2} I_{\frac{k}{x^2+k}}\left(\frac{k}{2},\frac{1}{2}\right) \tag{7}$$

$$F_F(x;k_1,k_2) = I_{\frac{k_1 x}{k_2+k_1 x}}\left(\frac{k_1}{2},\frac{k_2}{2}\right) = 1 - I_{\frac{k_2}{k_2+k_1 x}}\left(\frac{k_2}{2},\frac{k_1}{2}\right) \tag{8}$$

Here erf is the error function defined as

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \tag{9}$$

and erfc is the complementary error function defined as

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt. \tag{10}$$

The incomplete gamma functions P and Q are defined as

$$P(a,x) \equiv \frac{\gamma(a,x)}{\Gamma(a)} \equiv \frac{1}{\Gamma(a)} \int_0^x e^{-t} t^{a-1} dt \qquad (a > 0), \tag{11}$$

$$Q(a,x) \equiv 1 - P(a,x) \equiv \frac{\Gamma(a,x)}{\Gamma(a)} \equiv \frac{1}{\Gamma(a)} \int_x^\infty e^{-t} t^{a-1} dt \qquad (a > 0). \tag{12}$$

The incomplete beta function I is defined as

$$I_x(a,b) \equiv \frac{B_x(a,b)}{B(a,b)} \equiv \frac{1}{B(a,b)} \int_0^x t^{a-1}(1-t)^{b-1} dt \qquad (a,b > 0). \tag{13}$$

For a given test statistic X following a probability distribution with cdf $F(x)$, the left-tail p-value is defined as

$$p_{\text{left}}(x) = P(X < x) = F(x), \tag{14}$$

and the right-tail p-value is defined as

$$p_{\text{right}}(x) = P(X > x) = 1 - F(x) \equiv F_c(x). \tag{15}$$

For the normal and t distribution, the two-tails p-value is defined as

$$p_{\text{2tails}}(x) = P(|X| > |x|) = 2F_c(|x|) \quad \text{only for normal and t distribution.} \tag{16}$$

Finally, the middle area of the two distributions is $P(-|x| < X < |x|) = 1 - p_{\text{2tails}}(x)$.

As a result, the calculation of p-values boils down to the computation of the four cdf's (5)–(8), which involves the computation of the error function, incomplete gamma function and incomplete beta function. I use the algorithms described in the book *Numerical Recipes* to compute these functions[1], which I briefly describe in the following Sections.

2 Error Function

The following approximate formula is used to compute the function:

$$\operatorname{erf}(x) = \begin{cases} 1 - \tau & \text{for } x \geq 0 \\ \tau - 1 & \text{for } x < 0 \end{cases}, \tag{17}$$

where

$$\begin{aligned} \tau = {} & t \cdot \exp(-x^2 - 1.26551223 + 1.00002368t + 0.37409196t^2 + 0.09678418t^3 \\ & -0.18628806t^4 + 0.27886807t^5 - 1.13520398t^6 + 1.48851587t^7 \\ & -0.82215223t^8 + 0.17087277t^9) \end{aligned} \tag{18}$$

[1] The book has many editions. The one I use is *Numerical Recipes in Fortran 77: The Art of Scientific Computing*, second edition, by Press, Teukolsky, Vetterling and Flannery. An online version of the book is available at `http://www.aip.de/groups/soe/local/numres/`.

and

$$t = \frac{1}{1 + 0.5|x|}. \tag{19}$$

The approximation has a maximal error of 1.2×10^{-7}, which is more than enough since all of our p-values are displayed only to four significant figures.

The function `pnorm(z)` in `statFunction.js` is a JavaScipt code that computes $p_{\text{right}}(z) = 1 - F_{\text{normal}}(z)$.

3 Incomplete Gamma Functions

The incomplete gamma functions $P(k/2, x/2)$ or $Q(k/2, x/2)$ are used to compute the cdf of the χ^2 distribution (6). Here k is a positive integer and $x \geq 0$. The computation involves calculating the gamma function $\Gamma(k/2)$, and $\gamma(k/2, x)$ or $\Gamma(k/2, x)$ defined in equations (11) and (12).

The calculation of $\Gamma(k/2)$ is relatively easy. Since k is a positive integer, $\Gamma(k/2)$ can be computed using $\Gamma(1/2) = \sqrt{\pi}$, $\Gamma(1) = 1$ and the identity $\Gamma(a) = (a-1)\Gamma(a-1)$. The result is

$$\Gamma\left(\frac{k}{2}\right) = \begin{cases} \sqrt{\pi} & k = 1 \\ \sqrt{\pi} \cdot \frac{1}{2} \cdot \frac{3}{2} \cdot \frac{5}{2} \cdots \left(\frac{k}{2} - 1\right) & k = 3, 5, 7, 9, \cdots \\ \left(\frac{k}{2} - 1\right)! & k = 2, 4, 6, 8, \cdots \end{cases} \tag{20}$$

It is more convenient to work with $\ln\Gamma(k/2)$ instead of $\Gamma(k/2)$ to prevent floating-point overflow. The expression for $\ln\Gamma(k/2)$ is

$$\ln\Gamma\left(\frac{k}{2}\right) = \begin{cases} \frac{1}{2}\ln\pi & k = 1 \\ \frac{1}{2}\ln\pi + \sum\limits_{i=1}^{(k-1)/2} \ln\frac{2i-1}{2} & k = 3, 5, 7, 9, \cdots \\ \sum\limits_{i=2}^{(k-2)/2} \ln i & k = 2, 4, 6, 8, \cdots \end{cases} \tag{21}$$

For computational efficiency, the values of $\ln\Gamma(k/2)$ for $k \leq 200$ are saved in an array so that they need not be computed every time. For $k > 200$, the Lanczos approximation is used instead:

$$\begin{aligned} \ln\Gamma(z) &= \left(z + \frac{1}{2}\right)\ln(z + 5.5) - (z + 5.5) + \ln\frac{\sqrt{2\pi}}{z} \\ &\quad + \ln\left(c_0 + \frac{c_1}{z+1} + \frac{c_2}{z+2} + \cdots + \frac{c_6}{z+6} + \epsilon\right), \end{aligned} \tag{22}$$

where

$$\begin{aligned} c_0 &= 1.000000000190015,\ c_1 = 76.18009172947146,\ c_2 = -86.50532032941677, \\ c_3 &= 24.01409824083091, c_4 = -1.231739572450155,\ c_5 = 1.208650973866179 \times 10^{-3}, \\ c_6 &= -5.395239384953 \times 10^{-6}, \end{aligned} \tag{23}$$

and the magnitude of the error term is $|\epsilon| < 2 \times 10^{-10}$ for any positive value of z.

The function `gamnln(n)` in `statFunctions.js` is a JavaScript code that calculates $\ln\Gamma(n/2)$.

The function $\gamma(a,x)$ has the following series expansion.

$$\gamma(a,x) = e^{-x}x^a \sum_{n=0}^{\infty} \frac{\Gamma(a)}{\Gamma(a+1+n)} x^n = e^{-x}x^a \sum_{n=0}^{\infty} \frac{x^n}{a(a+1)(a+2)\cdots(a+n)}. \tag{24}$$

The function $\Gamma(a,x)$ has the following continued-fraction expansion.

$$\Gamma(a,x) = e^{-x}x^a \left[\frac{1}{x+1-a-} \; \frac{1\cdot(1-a)}{x+3-a-} \; \frac{2\cdot(2-a)}{x+5-a-} \cdots \right] \quad (x>0). \tag{25}$$

The continued fraction can be computed using the modified Lentz's method (see Section 5.2 of *Numerical Recipes*).

In the file `statFunctions.js`, the function `gser(n,x)` computes $P(n/2,x)$ using the series (24) for $\gamma(n/2,x)$. It is basically a JavaScript version of the function `gser` in *Numerical Recipes* (`http://www.aip.de/groups/soe/local/numres/bookfpdf/f6-2.pdf`). The function `gcf(n,x)` computes $Q(n/2,x)$ using the continued-fraction representation (25) for $\Gamma(n/2,x)$. It is basically a JavaScript version of the function `gcf` in *Numerical Recipes*. In both functions, the infinite sums are truncated at the mth term when the mth term is smaller than `eps` times the sum over these m terms. The parameter `eps` is set to 10^{-8}.

The series expansion (24) converges rapidly for x less than about $a+1$, whereas the continued-fraction expansion (25) converges rapidly for x greater than about $a+1$. In the file `statFunctions.js`, the function `gammp(n,x)` returns $P(n/2,x)$ and `gammq(n,x)` returns $Q(n/2,x)$. They call `gser` when $x<n/2+1$ and `gcf` when $x \geq n/2+1$. These are basically the JavaScipt version of the functions `gammap` and `gammq` in *Numerical Recipes*.

Now that the functions for $P(n/2,x)$ and $Q(n/2,x)$ are available, the cdf $F_{\chi^2}(\chi^2;n)$ can be computed easily. In `statFunctions.js`, the function `pchisq(chi2,n,ptype)` returns $p_{\rm right}(\chi^2 = {\rm chi2};n) = 1-F_{\chi^2}({\rm chi2};n)$ when `ptype = 1` and $p_{\rm left}(\chi^2={\rm chi2};n) = F_{\chi^2}({\rm chi2};n)$ when `ptype = 2`.

As a remark, the error functions can be expressed in terms of the incomplete gamma functions as follows.

$$\mathrm{erf}(x) = P\left(\frac{1}{2},x^2\right) \quad (x\geq 0) \quad , \quad \mathrm{erfc}(x) = Q\left(\frac{1}{2},x^2\right) \quad (x\geq 0). \tag{26}$$

This should not be too surprising as it is well-known that the p-values associated with the normal distribution and the χ^2 distribution with $k=1$ are related by

$$p_{\rm 2tails}(Z=z) = p_{\rm right}(\chi^2=z^2;k=1). \tag{27}$$

We can therefore use the incomplete gamma functions to calculate the error functions. However, the approximate expression (17) is still preferable because it is much simplier.

4 Incomplete Beta Function

Incomplete beta functions are used to compute the cdf for the t and F distribution (see (7) and (8)).

Aside: Comparing the two equations, one can deduce the well-known (or should be well-known) relationship between the p-values associated with the t distribution with k degrees of freedom and the F distribution with $k_1=1$ and $k_2=k$:

$$p_{\rm 2tails}(T=t;k) = p_{\rm right}(F=t^2;k_1=1,k_2=k). \tag{28}$$

The incomplete beta function $I_x(a,b)$ has the following continued fraction representation.

$$I_x(a,b) = \frac{x^a(1-x)^b}{aB(a,b)} \left(\frac{1}{1+} \; \frac{d_1}{1+} \; \frac{d_2}{1+} \cdots \right), \tag{29}$$

where

$$d_{2m+1} = -\frac{(a+m)(a+b+m)x}{(a+2m)(a+2m+1)} \quad , \quad d_{2m} = \frac{m(b-m)x}{(a+2m-1)(a+2m)}. \tag{30}$$

This continued fraction is evaluated in the function `betacf(a,b,x)` in `statFunctions.js`. It is basically the JavaScript version of the function `betacf` in *Numerical Recipes* (`http://www.aip.de/groups/soe/local/numres/bookfpdf/f6-4.pdf`). The infinite sum is terminated at the ith term when the ith term is smaller than `eps` times the sum over the i terms, where `eps` is set to 10^{-8}.

The continued fraction (29) converges rapidly for $x < (a+1)/(a+b+2)$. The case for $x > (a+1)/(a+b+2)$ can be calculated using the symmetry relation of the beta function:

$$I_x(a,b) = 1 - I_{1-x}(b,a). \tag{31}$$

The function `betai(n,m,x)` in `statFunctions.js` returns $I_x(n/2, m/2)$ for positive integers n and m. It is basically the JavaScript version of the function `betai` in *Numerical Recipes.*

The p-values associated with the t distribution are calculated in the function `pt(t,n,ptype)` in `statFunctions.js`. The function returns $p_{\rm left}(t;n) = F_t(t;n)$ when `ptype = 0`, $p_{\rm right}(t;n) = 1 - F_t(t;n)$ when `ptype = 1`, $p_{\rm 2tails}(t;n) = 2[1 - F_t(|t|;n)]$ when `ptype = 2`, and the middle area $= 1 - p_{\rm 2tails}(t;n)$ when `ptype = 3`.

The p-values associated with the F distribution are calculated in the function `pf(F,df1,df2,ptype)`. It returns $p_{\rm right}(F;{\rm df1},{\rm df2}) = 1 - F_F(F;{\rm df1},{\rm df2})$ when `ptype = 1`, and $p_{\rm left}(F;{\rm df1},{\rm df2}) = F_F(F;{\rm df1},{\rm df2})$ when `ptype = 2`.

5 Inverse of the CDFs

Given a test statistic X following a particular probability distribution and a significance level α, the critical value is defined as the value of x such that the associated p-value $p(x) = \alpha$. Since the p-values are related to the cdf of the distribution, computing the critical values involves calculating the inverse of the cdf. In `statFunctions.js`, the inverse is calculated by solving the non-linear equation $p(x) - \alpha = 0$ numerically using the bisection method.

Since $p(x)$ is a monotonic function of x and ranges from 0 to 1, it is easy to find (x_1, x_2) to bracket the root. Once x_1 and x_2 are found, the bisection method is very robust in finding the root. The function `bisection(f, x1,x2, releps, abseps)` in `statFunctions.js` searches the root using the bisection method. The input `f` is a user-defined function of one variable; `x1` and `x2` (with `x2` $>$ `x1`) are the initial values of x_1 and x_2 that bracket the root; `releps` and `abseps` are parameters controlling the relative and absolute errors. Inside `bisection`, `x1` and `x2` are refined and the value of `x2-x1` is reduced by a factor of 2 in each iteration. The function returns $x = (x_1+x_2)/2$ if one of the following conditions is satisfied:

1. $x_2 - x_1 < \text{abseps}$;
2. $|f(x)| < \text{abseps}$, where $x = (x_1+x_2)/2$;
3. $x_2 - x_1 < \text{releps} \cdot x$.

When it is used to compute the inverse of the cdf's, I find the best result by setting the parameter `abseps = 0` so that the accuracy is controlled entirely by the relative error parameter `releps`. Since the result is only displayed to 4 significant figures in the html pages, I set `releps` to 10^{-6}, which is more than enough for the accuracy requirement.

In `statFunctions.js`, the four functions `qnorm(p)`, `qchisq(p,n,ptype)`, `qt(p,n,ptype)` and `qf(F,df1,df2,ptype)` compute the inverse of the functions `pnorm(z)`, `pchisq(chi2,n,ptype)`, `pt(t,n,ptype)` and `pf(F,df1,df2,ptype)`. That is, `qnorm(p)` returns a value `z` so that `pnorm(z) = p`; `qchisq(p,n,ptype)` returns a value `chi2` so that `pchisq(chi2,n,ptype) = p`, and so on.

6 Summary of Functions in `statFunctions.js`

Main functions:

- `pnorm(z)`

  Returns the right-tail p-value $p_{\text{right}}(z) = 1 - F_{\text{normal}}(z)$ for the normal distribution.

- `pchisq(chi2,n,ptype)`

  Returns p-values associated with the χ^2 distribution: $p_{\text{right}}(\chi^2 = \text{chi2}; n) = 1 - F_{\chi^2}(\text{chi2}; n)$ when `ptype = 1` and $p_{\text{left}}(\chi^2 = \text{chi2}; n) = F_{\chi^2}(\text{chi2}; n)$ when `ptype = 2`.

- `pt(t,n,ptype)`

  Returns p-values associated with the t distribution: $p_{\text{left}}(t; n) = F_t(t; n)$ when `ptype = 0`, $p_{\text{right}}(t; n) = 1 - F_t(t; n)$ when `ptype = 1`, $p_{\text{2tails}}(t; n) = 2[1 - F_t(|t|; n)]$ when `ptype = 2`, and the middle area $= 1 - p_{\text{2tails}}(t; n)$ when `ptype = 3`.

- `pf(F,df1,df2,ptype)`

  Returns p-values associated with the F distribution: $p_{\text{right}}(F; \text{df1}, \text{df2}) = 1 - F_F(F; \text{df1}, \text{df2})$ when `ptype = 1`, and $p_{\text{left}}(F; \text{df1}, \text{df2}) = F_F(F; \text{df1}, \text{df2})$ when `ptype = 2`.

- `qnorm(p)`, `qchisq(p,n,ptype)`, `qt(p,n,ptype)` and `qf(F,df1,df2,ptype)`

  Inverse of the four functions `pnorm(z)`, `pchisq(chi2,n,ptype)`, `pt(t,n,ptype)` and `pf(F,df1,df2,ptype)`.

Auxiliary functions

- `bisection(f, x1,x2, releps, abseps)`

  Returns an estimate of the root of $f(x) = 0$ from a user-supplied, one-variable function `f` in the interval `(x1,x2)` with the relative and absolute errors of the root set by the parameters `releps` and `abseps`.

- `gamnln(n)`

  Returns $\ln \Gamma(n/2)$, where n is a positive integer.

- `gser(n,x)`

  Returns the incomplete gamma function $P(n/2, x)$ evaluated by a series representation.

- `gcf(n,x)`

  Returns the the incomplete gamma function $Q(n/2, x)$ evaluated by its continued fraction representation.

- `gammp(n,x)`

  Returns the incomplete gamma function $P(n/2, x)$ by calling `gser` when $x < n/2 + 1$ and `gcf` when $x \geq n/2 + 1$.

- `gammq(n,x)`

  Returns the incomplete gamma function $Q(n/2, x)$ by calling `gser` when $x < n/2 + 1$ and `gcf` when $x \geq n/2 + 1$.

- `betacf(a,b,x)`

  Evaluates the incomplete beta function $I_x(a, b)$ by its continued fraction representation.

- `betai(n,m,x)`

  Returns the incomplete beta function $I_x(n/2, m/2)$ for positive integers n and m.