

## ***Approximating the Behrens-Fisher Distribution***

**Jacob Colvin**

The Behrens-Fisher distribution is a Bayesian's interpretation of what a true 2 sample t-test of unequal variance should look like, as apposed to the common welsh approximation frequently noted in text books. Historically, it appears that the B-F distribution has been ignored not because of its obscure application, but because of difficulty in exactly computing the distribution for tables and computer packages. Using the power of computers, I will show a method for approximating densities, quantiles, and p values in real time to very high precision using R. Furthermore, this algorithm is an unbiased estimator of the exact B-F distribution.

As shown in [Behrens's original paper, or patil's?], the B-F distribution is distributed as...

$$BF(df_1, df_2, \theta) \sim t_2 \cos(\theta) - t_1 \sin(\theta) \quad (1) \text{ [page 145 of Lee]}$$

where

$$\theta = \arctan( (s_1/\sqrt{n_1}) * (s_2/\sqrt{n_2}))$$

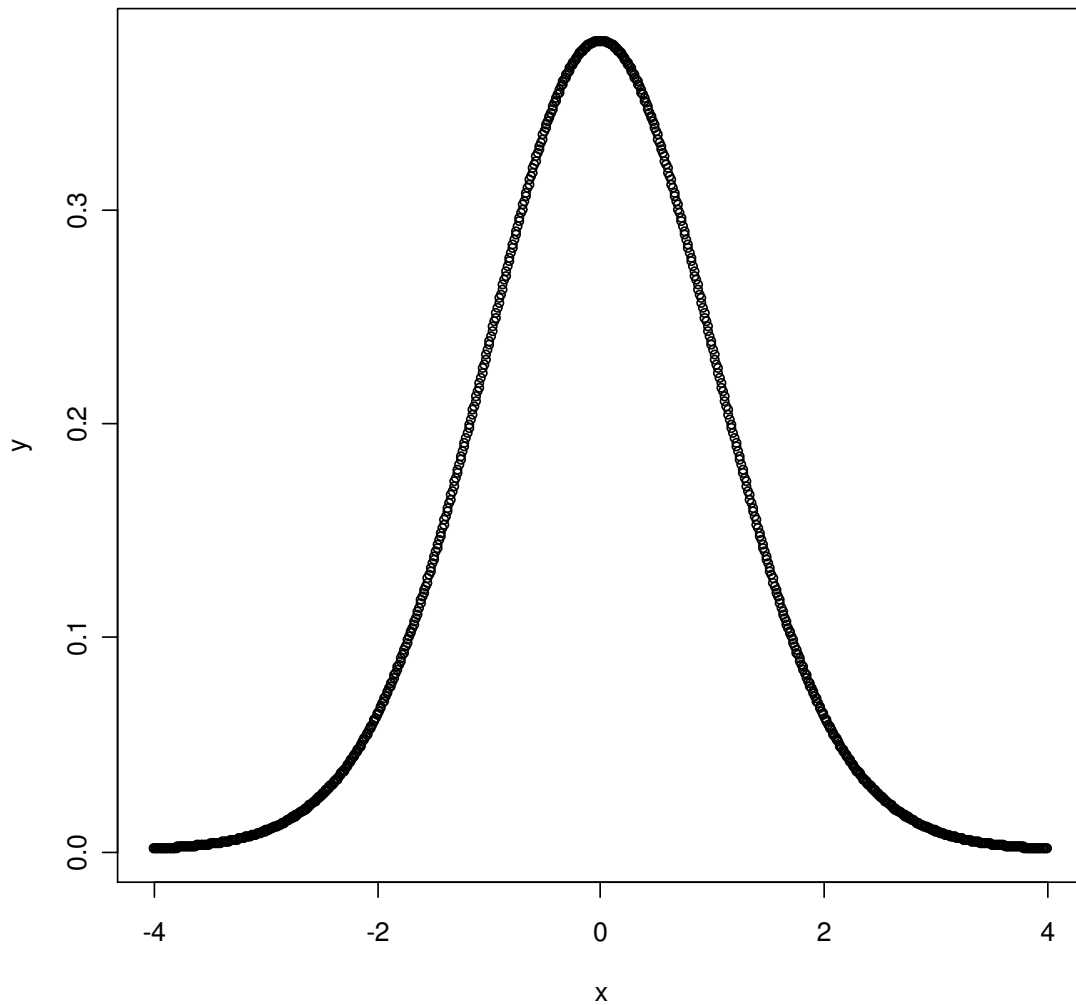
$t_1$  = t distribution with  $n_1-1$  degrees of freedom

$t_2$  = t distribution with  $n_2-1$  degrees of freedom

$\theta$  can loosely be interpreted as a relative coefficient of variability between the two samples.  $\theta = 45$ degrees and  $n_1 = n_2$  implies that that two samples have equal variance and equal sample sizes, and thus the T distribution is a special case of the B-F distribution.

Graphical idea behind this method is to calculate all of the distribution statistics from the following graph, essentially a joint inverse cdf contour plot of two independent t distributions.

**Behrens PDF with df1=6, df2=24, theta=30**

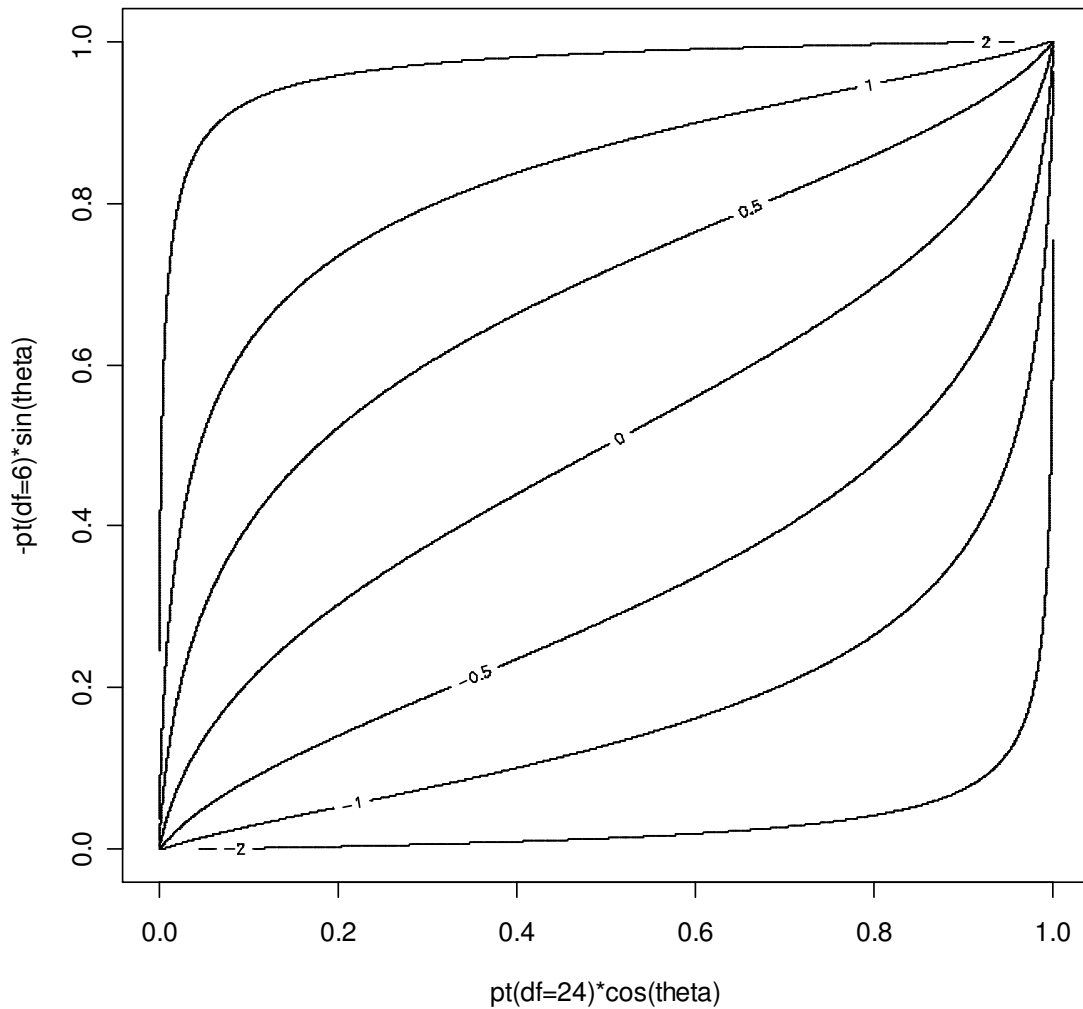


```
y=dworking(x,6,24,30); plot(x,y,main="Behrens PDF with df1=6, df2=24, theta=30")
```

My algorithm relies on numerical differentiation, numerical integration, and numerical root finding as described within standard numerical analysis texts.

The key to my method can be understood by graphing equation (1) as a joint inverse cdf of two T distributions. A contour map of one such plot is shown below.

## Joint Inverse CDF



```
x=generate_table(6,24,30,plot=TRUE);title(main="Joint Inverse  
CDF",xlab="pt (df=24) *cos(theta)",ylab="-pt (df=6) *sin(theta)")
```

A few observations

1. The units of contour plot lines are quantiles.
2. The total area of the plot is 1.
3. Since the BF distribution is symmetric, the area above and below the "0" contour line is zero.
4. The area below the contour plot -1 line is the same as the area to the left of the -1 quantile of the BF pdf graph.

Thus the BF cdf function can be restated as finding the area below a given  $z^*$  value of the above BF contour plot. The BF inverse CDF function be restated as searching for the  $z^*$  value such that the area below the contour plot is some fixed area  $\alpha$ . The BF pdf function is the change in the area under  $z^*$  and  $z^*+\epsilon$  divided by  $\epsilon$ .

Another important observation is that for a fixed BF quantile  $z^*$ , the contour plot can be formulated in terms of  $y = f(x|z^*)$  where  $x$  and  $y$  are probabilities on the interval  $[0,1]$  associated with  $t_1$  and  $t_2$  respectively.

$$\begin{aligned}z^* &= t_2 \cos(\theta) - t_1 \sin(\theta) \\z^* + t_1 \sin(\theta) &= t_2 \cos(\theta) \\(z^* + t_1 \sin(\theta)) / \cos(\theta) &= t_2\end{aligned}$$

Thus using R notation...

$$y = f(x|z^*, df1, df2, \theta) = pt((z^* + qt(x, df1) * \sin(\theta)) / \cos(\theta), df2)$$

Using numerical analysis techniques, the cdf function can be restated as an integration problem. The inverse cdf function can be restated as a root finding problem based on the cdf function. Finally the pdf function can be restated as a differentiation problem based on the cdf function.

R code has been developed to calculate this approximation to the BF distribution, and based on all the available sources I can tell, produces significantly different answers than those that have been published in distribution tables in books as recent as 2004. My solution agrees with simulation averages derived by generating a billion of BF random variables from equation (1) and calling R functions like `density()`, `quantile()` and `ecdf()`. Additionally, this method is much faster than the direct simulation, and should prove computationally feasible. Calls to “`pbehrens()`” would take on average about 0.01 seconds each on a 1.83 GHz Pentium M.

Source code is available, but is in alpha status. The results tested so far have been reliable, but issues remain with rounding error in calls to R’s `integrate` function and handling other special degenerate cases of the BF distribution.

An earlier version of the code dissected the contour plot into a series of grids and calculated progressively tighter upper and lower bounds for the desired calculation. Preliminary results conclude that this technique is too slow for everyday use, but would be reasonable for calculating a distribution table without extensive specialized hardware. Unfortunately this technique was abandoned due to its reliance on extremely accurate percentile and quantile calculations for the t distribution, preferable full double precision. As the grids became progressively smaller, the t distribution errors were magnified to the point that the simulated BF distribution results no longer fit within the supposed 100% C.I. This technique does not appear to be invalid, but to be practical, code needs to be found that provides higher precision when calculating t distribution statistics.

References:

Bayesian Statistics: an introduction, 3<sup>rd</sup> ed. Peter M. Lee. Sections 5.3-4