

11-1-2004

An Algorithm And Code For Computing Exact Critical Values For The Kruskal-Wallis Nonparametric One-Way ANOVA

Sikha Bagui

The University of West Florida, Pensacola, bagui@uwf.edu

Subhash Bagui

The University of West Florida, Pensacola, sbagui@uwf.edu

 Part of the [Applied Statistics Commons](#), [Social and Behavioral Sciences Commons](#), and the [Statistical Theory Commons](#)

Recommended Citation

Bagui, Sikha and Bagui, Subhash (2004) "An Algorithm And Code For Computing Exact Critical Values For The Kruskal-Wallis Nonparametric One-Way ANOVA," *Journal of Modern Applied Statistical Methods*: Vol. 3 : Iss. 2 , Article 20.
DOI: 10.22237/jmasm/1099268400

An Algorithm And Code For Computing Exact Critical Values For The Kruskal-Wallis Nonparametric One-Way ANOVA

Sikha Bagui Subhash Bagui
University of West Florida, Pensacola

In this article, an algorithm and code to compute exact critical values (or percentiles) for Kruskal-Wallis test on k independent treatment populations with equal or unequal sample sizes using Visual Basic (VB.NET) is provided. This program has the ability to calculate critical values for any k , sample sizes (n_i), and significance level (α). An exact critical value table for $k = 4$ is also developed. The table will be useful to practitioners since it is not available in standard nonparametric statistics texts. The program can also be used to compute any other critical values.

Key words: Kruskal-Wallis test, ANOVA, visual basic

Introduction

Headrick (2003) wrote an article for generating exact critical values for the Kruskal-Wallis (K-W) one-way ANOVA using Fortran 77. In this article we present Visual Basic (VB.NET) Code for generating exact critical values for K-W tests using the Visual Basic Programming Language. VB.NET is more user friendly and more accessible than Fortran 77. While Fortran 77 may not be available to all, the proposed VB.NET program can be a simpler alternative to Fortran 77.

When one or more treatment populations violate normality assumption or the homogeneity of treatment population variances, it is customary to use Kruskal-Wallis (1952) rank-based nonparametric test as an alternative to the conventional F test for one-way analysis

of variance (ANOVA) for k independent treatment populations.

In order to find the critical values of K-W tests, one needs to find the null distribution of the K-W statistics. In one-way ANOVA, the null hypothesis is that the effect of all treatment populations are the same. Thus, it is reasonable to use such a type of null distribution of the K-W statistics which are derived under the assumption that all observations for treatment populations n_1, n_2, \dots, n_k are from the same population to calculate the critical values of K-W tests, where n_i is the sample size of the i th treatment population. The K-W statistic depends on the rank-sums of each treatment population that are obtained from the combined ranks of $N = n_1 + n_2 + \dots + n_k$ observations.

It is known that the large sample null distribution of K-W statistic is approximately a chi-square (χ^2) distribution with $(k-1)$ degrees of freedom (d.f.). Conover (1999) suggested that whenever $k \geq 4$ and $n_i > 5$, for each treatment population, a chi-square critical with $(k-1)$ d.f. ($\chi_{\alpha, k-1}^2$) be used to test the null hypothesis. But for small samples, say $n_i \leq 5$, the null-distribution of K-W statistic is not known and a chi-square approximation will not be a good approximation. The common nonparametric text books such as

Sikha Bagui is an Assistant Professor in the Department of Computer Science. Her areas of research are database and database design, data mining, pattern recognition, and statistical computing. Email: bagui@uwf.edu. Subhash Bagui is a Professor in the Department of Mathematics and Statistics. His areas of research are statistical classification and pattern recognition, bio-statistics, construction of designs, tolerance regions, statistical computing and reliability. Email: sbagui@uwf.edu.

Conover (1999), Gibbons (1992); Siegel and Castellan (1989) provided exact critical values for the K-W test for $k=3$ and $n_i \leq 5$ observations per treatment population. Major statistical software such as MINITAB, SPSS provide only the asymptotic P -value of the K-W statistics. In view of all these, in this article, we provide a VB.NET program as an alternative to Fortran 77 to compute exact critical points for the K-W tests, and also report a table for exact critical values for the K-W test for $k=4$ treatment populations and $n_i \leq 5$ observations per treatment population. Even though the number of ways N ranks can be divided into groups of n_1, n_2, \dots, n_k grows, our VB.NET program works well with reasonable values of k and n_i .

Methodology

To calculate the K-W statistics, first we need to generate N uniform pseudo-random numbers from the interval $(0,1)$. We assume that the probability of a tie is zero. Then the random variates are ranked to form permutation of numbers from 1 to N . The program then sequentially divides the permutation of ranks into k classes according to the users specific sample sizes of n_1, n_2, \dots, n_k . The program then calculates rank sums of each treatment population, R_j , and next computes the value of K-W statistic

$$H = \frac{12}{N(n+1)} \sum_{j=1}^k \frac{R_j^2}{n_j} - 3(N+1).$$

This process is replicated a sufficient number of times until the null distribution of H is modeled adequately. Then the program selects a critical value that is associated with a percentile values of 0.90, 0.95, 0.975 or 0.99 (or equivalently a alpha level of 0.10, 0.05, 0.025 or 0.01). In some cases, returned values may coincide with two different alpha values, since returned values are true for a range of P -values.

For example, given $\alpha = 0.05$, $k = 3$, and $n_i = 5$, our VB.NET program will return a critical value of 5.659997 with a replication of 100,000 runs which is same as the value reported by Headrick (2003) for $\alpha = 0.05$, $k = 3$, and $n_i = 5$. Also, with adequate number of runs, our VB.NET program yields the same values reported by Conover (1999) in Table A8. Table A8 is for $k = 3$, $n_i \leq 5$ and $\alpha = 0.1, 0.05$, and 0.01. In Table 1 critical values are provided for K-W statistic for $k = 4$, $n_i \leq 5$ and $\alpha = 0.1, 0.05, 0.025$, and 0.01. At the bottom of the table, the asymptotic chi-square critical values of H from a chi-square critical value table are also provided. The notation K_α is a $(\alpha)100\%$ percentile for the K-W statistics which is equivalent to $(1-\alpha)$ level critical value of the K-W statistic. This table will be very useful to the practitioners because it is not available in standard nonparametric text books. The critical values in Table 1 are generated using 1 million replications in each case.

Conclusion

In case of large N , the program needs large number of replications in order to adequately model the null distribution of K-W statistic H . So the replication number should be in increasing order such as 10,000, 50,000, 100,000, 500,000, and 1,000,000, etc. and stop the process once two consecutive values are almost the same. If there are k independent treatment populations, then at least $N!/(n_1!)(n_2!) \dots (n_k!)$ replications are necessary for a near fit of H . For a good fit of H , one needs much more replications than $N!/(n_1!)(n_2!) \dots (n_k!)$. The VB.NET code is given in the Appendix. The VB.NET program is very user friendly. The VB.NET program allows the user to provide the values of replication numbers, total number of observations, percentile fractions, and separate class sizes based on which the program will return a critical value.

Table 1. Critical values for K-W statistics for small sample sizes.

Sample sizes	$K_{0.90}$	$K_{0.95}$	$K_{0.975}$	$K_{0.99}$
2, 2, 2, 1	5.0357	5.3571	5.6786	5.6786
2, 2, 2, 2	5.5000	6.0000	6.1667	6.1667
3, 2, 2, 1	5.3889	5.8056	6.0556	6.5000
3, 2, 2, 2	5.5778	6.2444	6.6444	7.0000
3, 3, 2, 1	5.6222	6.1556	6.5111	7.0444
3, 3, 2, 2	5.7273	6.4727	7.0000	7.5636
3, 3, 3, 1	5.5818	6.5273	6.8909	7.3273
3, 3, 3, 2	5.8182	6.6818	7.4697	7.9545
3, 3, 3, 3	5.9744	6.8462	7.6154	8.4359
4, 2, 1, 1	5.2083	5.4583	6.0833	6.0833
4, 2, 2, 1	5.5000	6.0000	6.5000	6.8000
4, 2, 2, 2	5.6727	6.4364	6.9818	7.3091
4, 3, 2, 1	5.5636	6.3000	6.9091	7.3636
4, 3, 2, 2	5.7121	6.6136	7.3409	7.8712
4, 3, 3, 1	5.6667	6.5379	7.2727	7.7500
4, 3, 3, 2	5.8590	6.7820	7.5256	8.3205
4, 3, 3, 3	5.9780	6.9670	7.7582	8.6538
4, 4, 2, 1	5.5682	6.3864	7.1364	7.8864
4, 4, 2, 2	5.7692	6.6923	7.5000	8.3077
4, 4, 3, 1	5.6603	6.6154	7.4808	8.2179
4, 4, 3, 2	5.8901	6.8626	7.7527	8.6044
4, 4, 3, 3	6.0048	7.0333	7.8905	8.8619
4, 4, 4, 1	5.6374	6.7088	7.6319	8.5714
4, 4, 4, 2	5.9000	6.9429	7.8857	8.8571
4, 4, 4, 3	6.0292	7.1292	8.0542	9.0625
4, 4, 4, 4	6.0441	7.1691	8.1618	9.2426
5, 2, 2, 1	5.5309	6.0327	6.5782	7.2000
5, 2, 2, 2	5.6091	6.4818	7.1545	7.6636
5, 3, 2, 1	5.5030	6.3303	7.0939	7.7455
5, 3, 2, 2	5.7538	6.6564	7.4641	8.1949
5, 3, 3, 1	5.6564	6.6000	7.4205	8.1179
5, 3, 3, 2	5.8571	6.8220	7.6505	8.5824
5, 3, 3, 3	6.0210	7.0114	7.8267	8.8400
5, 4, 2, 1	5.5615	6.4077	7.2115	8.1346
5, 4, 2, 2	5.7725	6.7220	7.6154	8.4725
5, 4, 3, 1	5.6396	6.6813	7.5253	8.3989
5, 4, 3, 2	5.8933	6.9143	7.7933	8.8143
5, 4, 3, 3	6.0292	7.0892	7.9892	9.0292
5, 4, 4, 1	5.6686	6.7714	7.6857	8.7286
5, 4, 4, 2	5.9350	6.9775	7.9600	9.0000
5, 4, 4, 3	6.0346	7.1640	8.1412	9.2294
5, 4, 4, 4	6.0569	7.2431	8.2569	9.3824
5, 5, 2, 1	5.5648	6.5341	7.2725	8.3077
5, 5, 2, 2	5.7771	6.7714	7.6400	8.6342
5, 5, 3, 1	5.6476	6.7371	7.6286	8.5886
5, 5, 3, 2	5.9067	6.9417	7.8667	8.9417
5, 5, 3, 3	6.0000	7.1176	8.0588	9.1706
5, 5, 4, 1	5.6625	6.7800	7.7500	8.8425
5, 5, 4, 2	5.9338	7.0257	8.0096	9.1434
5, 5, 4, 3	6.0523	7.2176	8.2092	9.3562
5, 5, 4, 4	6.0614	7.2684	8.3281	9.5053
5, 5, 5, 1	5.6224	6.8294	7.8176	9.0529
5, 5, 5, 2	5.9412	7.0745	8.1059	9.2863
5, 5, 5, 3	6.0433	7.2456	8.2889	9.4772
5, 5, 5, 4	6.0774	7.3200	8.4253	9.6521
5, 5, 5, 5	6.0971	7.3314	8.4629	9.7886
$\chi^2_{1-\alpha; k-1} \rightarrow$	$\chi^2_{0.10; 3} = 6.251$	$\chi^2_{0.05; 3} = 7.815$	$\chi^2_{0.025; 3} = 9.348$	$\chi^2_{0.01; 3} = 11.345$

References

- Conover, W.J. (1999). *Practical nonparametric statistics*. (3rd ed.). New York: Wiley.
- Gibbons, J.D. (1992). *Nonparametric statistical inference*. (3rd ed.). New York: Marcel Dekker.
- Headrick, T.C. (2003). An algorithm for generating exact critical values for the Kruskal-Wallis One-way ANOVA. *Journal of Modern Applied Statistical Methods*, 2, 268-271.
- Kruskal, W.H. & Wallis, W.A. (1952). Use of ranks in one-criterion analysis of variance. *Journal of American Statistical Association*, 47, 583-621.
- Minitab (2000). *Minitab for Windows, release 13.3*, Minitab Inc., State College, PA.
- Siegel, S., & Castellan, N.J. (1989). *Nonparametric statistics for the behavioral sciences*, (2nd ed.). New York: McGraw-Hill.
- SPSS(2002). *SPSS for Windows, version 11.0*, SPSS, Inc., Chicago, IL.

Appendix

```
Public Class Form1
```

```
    Inherits System.Windows.Forms.Form
    'Kruskal-Wallis One-Way Anova
```

```
    Dim a, m, n, i, j, k, v, x, y, z, sumy, row, count As Integer
    Dim output, prompt1, prompt2, prompt_value, group_value As String
    Dim output2 As String
    Dim sums, sumz, H, percentile As Single
```

```
    Dim file1 As System.IO.StreamWriter
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        m = Val(TextBox2.Text)
        n = Val(TextBox1.Text)
        percentile = Val(TextBox3.Text)
```

```
        file1 = System.IO.File.CreateText("C:\Documents and
Settings\sbagui\Desktop\file1.txt")
```

```
        prompt1 = "Please enter number of groups"
        prompt_value = InputBox(prompt1)
```

```
x = Val(prompt_value)
```

```
    Dim array1(n) As Single
    Dim array2(n) As Single
    Dim array3(n) As Integer
    Dim array4(m) As Single
    Dim array5(n) As Integer
```

```
    For i = 1 To (x - 1)
        prompt2 = "Please enter number in group" & i
        group_value = InputBox(prompt2)
        array5(i) = Val(group_value)
    Next
```

```
    For k = 1 To m
        For i = 1 To n
            array1(i) = Rnd()
        Next
    Next
```

```

For i = 1 To array1.GetUpperBound(0)
    array2(i) = array1(i)
Next

Array.Sort(array2)

'ranking row
For row = 1 To array1.GetUpperBound(0)
    For i = 1 To array2.GetUpperBound(0)
        If array1(row) = array2(i) Then
            array3(row) = i
        End If
    Next
Next

For i = 1 To array3.GetUpperBound(0)
    output &= array3(i) & " "
Next

y = 0
z = 0
sumy = 0
sumz = 0
For i = 1 To (x - 1)
    y = array5(i)
    For j = (1 + z) To (y + z)
        sumy += array3(j)
    Next
    a = y + z
    z = a
    sums = (sumy * sumy) / y
    sumz += sums
    sumy = 0
    sums = 0
Next

count = 0
sumy = 0
For i = (z + 1) To array3.GetUpperBound(0)
    sumy += array3(i)
    count = count + 1
Next
sums = (sumy * sumy) / count
sumz += sums

H = 12 / (n * (n + 1)) * sumz - 3 * (n + 1)
array4(k) = H

H = 0
Next

Array.Sort(array4)      'array4 - sorted F values

output = " "
For i = 1 To array4.GetUpperBound(0)
    output &= array4(i) & " "
Next

count = 0
For i = 1 To array4.GetUpperBound(0)
    count += 1
Next

```

```
v = percentile * count
output = " "
output = array4(v)
MessageBox.Show(output, percentile & " percentile value")
output2 = percentile & " = " & output
file1.WriteLine(output2)
file1.Close()
End Sub

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Randomize() 'Calling the random number generator
End Sub
End Class
```