Calculating Poisson confidence Intervals in Excel

Iain Buchan 27th January 2004
Public Health Informatics at the University of Manchester (www.phi.man.ac.uk)
buchan@man.ac.uk

Purpose

This is a technical paper for people in Public Health Intelligence roles. It assumes basic statistical knowledge and familiarity with the Microsoft Excel (Microsoft, 2004) spreadsheet software application.

Background

Public Health information often requires the calculation of confidence intervals for rates.

A large sample approximation to the confidence interval for rates, assuming a Poisson distribution, is often constructed using the square root of the observed number of events as the standard deviation of those events in the population then treating the results as asymptotically Gaussian (Bland, 2000). Unfortunately, the large sample approximation is unreliable with numbers that are common in Public Health use. A recent comparison of exact and approximate methods applied to Public Health mortality statistics in the Northwest region showed a change in raking of areas when making this unnecessary approximation (Hennel, 2004).

There is no need to make the compromise of large sample approximation with the modern widespread availability of computers, using either specialised statistical software or general spreadsheet software with basic statistical functions. This paper describes methods and pitfalls to avoid when calculating exact Poisson confidence intervals in spreadsheet software.

Methods

Exact confidence intervals for crude rates can be constructed using the following formula that relates the chi-square and Poisson distributions (Ulm, 1990):

$$LL = \frac{\left(\chi^2_{\frac{\alpha}{2},2d}\right)}{2}$$

$$UL = \frac{\left(\chi^2_{1-\frac{\alpha}{2},2(d+1)}\right)}{2}$$

- where LL and UL are lower and upper confidence limits respectively for the rate as d events per unit time exposed, χ^2 ν,a is the $(100*\alpha)^{th}$ chi-square centile with ν degrees of freedom, and d is the number of observed events (e.g. deaths).

Exact confidence intervals for standardised mortality (event) ratios can be calculated by simple extension of the Poisson via chi-square method above (Dobson et al., 1991):

$$SMR = 100 * \frac{d}{e}$$

$$e = \sum_{i=1}^{k} n_i R_i$$

$$LL = \frac{\left(\chi^2_{\frac{\alpha}{2},2d}\right)}{2e}$$

$$UL = \frac{\left(\chi^2_{1-\frac{\alpha}{2},2(d+1)}\right)}{2e}$$

- where LL and UL are lower and upper confidence limits respectively for the SMR, $\chi^2_{\nu,\alpha}$ is the $(100*\alpha)^{th}$ chi-square centile with ν degrees of freedom, d is the number of observed deaths, e is the number of expected deaths, n_i is the person-time for the ith study group stratum and R_i is the reference population rate for the ith stratum.

Translating the above closed form arithmetic into function expressions in Microsoft Excel spreadsheet software (Microsoft, 2004) can be done as follows:

[LL] =IF(A1>0,(CHIINV(0.975, 2 * A1) / 2)/A2,0)

[UL] =IF(A1>0,(CHIINV(0.025, 2 * (A1+1))/ 2)/A2,(CHIINV(0.025, 2)/ 2)/A2)

- where the A1 cell holds the value for the number of events (e.g. deaths) observed, cell A2 holds the unit time exposed (e.g. person years), and the confidence level required is 95% (i.e. tolerating a 5% probability of incorrectly rejecting the null hypothesis).

Results

Using Microsoft Excel versions XP and 2003 and comparing the results with test algorithms using reliable approaches to the chi-square function (Best and Roberts, 1975) and with StatsDirect statistical software (StatsDirect Ltd, 2004), results precise to around six significant digits were obtained for most of the test data (a range of actual all cause mortality statistics in Northwest England).

Excel threw errors in the calculation of a substantial number of typical data, including the following (rounded results in brackets):

Observed events	466	
Expected events	234.2105	
SMR	1.989663145	(199)
Lower 95% confidence limit for the SMR	1.813091906	(181)
Upper 95% confidence limit for the SMR	2.17878461	(218)

The problem was traced to the CHIINV function which failed to return results for reasonable data such as =CHIINV(0.975,932).

Discussion

Others have reported statistical impreciseness and frank computational error in Excel, but this has concerned mainly the statistical add-in, 'Analysis Toolpak' (McCullough and Wilson, 1999). Under-performance of a core Excel worksheet function 'CHIINV' leading to failed calculation in a common Public Health information context is reported here.

Algorithms to calculate precise Poisson confidence limits in Excel Visual Basic for Applications (VBA) are given in Appendix One. An Excel workbook with worked examples proving the results above is provided at http://www.phi.man.ac.uk. This website also provides reliable algorithms, written in C#, that Microsoft could include in their .Net framework and use to correct CHIINV in a future version of Excel.

References

Bland M. An Introduction to Medical Statistics (3rd edition). Oxford Medical Publications 2000.

Hennel T. Personal communication. Government Office Northwest 2004.

Ulm K. A simple method to calculate the confidence interval of a standardized mortality ratio. American Journal of Epidemiology 1990;131(2):373-375.

Dobson AJ, Kuulasmaa K, Eberle E, Scherer J. Confidence intervals for weighted sums of Poisson parameters. Statistics in Medicine 1991;10:457-462.

Microsoft Corporation. Microsoft Excel spreadsheet software. www.microsoft.com/excel, 2004.

Best DJ, Roberts DE. AS 91, The Percentage Points of the Chi² Distribution. Applied Statistics 1975;24(3).

StatsDirect Ltd. StatsDirect statistical software. www.statsdirect.com, 2004.

McCullough BD, Wilson B. On the accuracy of statistical procedures in Microsoft Excel 97. Computational Statistics and Data Analysis 1999;31:27-37.

Appendix

Excel VBA code (download from http://www.phi.man.ac.uk) for precise confidence intervals if calling a precise algorithm for the quantiles of the chi-square distribution:

'Written by Iain Buchan, 27th January 2004

```
'StatsDirect routines called because Excel chi-square inversion is unreliable
'Best DJ, Roberts DE. AS 91, The Percentage Points of the Chi² Distribution. Applied Statistics 1975;24(3).
Public Declare Function PPCHI2 Lib "StatsDirect" (P As Double, v As Double, fault As Long) As Double

Public Function LowerPoissonLimit(Events As Variant, TimeAtRisk As Variant, Optional Alpha As Variant) As Variant
'relates chi-sq to Poisson
'Johnson & Kotz 1969, Ulm in Am J Epidemiol 1990 (131) 373-
'comments by Dobson Stats in Med 1991 (10) 457-
Dim a As Double
Dim ifault As Long
On Error Resume Next
If IsMissing(Alpha) Then
 a = 0.05
Else
 a = Alpha
End If
If Events < 0# Then
 LowerPoissonLimit = CVErr(2001)
ElseIf Events = 0# Then
 LowerPoissonLimit = 0#
Else
 LowerPoissonLimit = PPCHI2(a / 2#, 2# * CDbl(Events), ifault) / 2#
 If Application.WorksheetFunction.IsNumber(LowerPoissonLimit) = True And TimeAtRisk > 0# And ifault = 0 Then
  LowerPoissonLimit = LowerPoissonLimit / TimeAtRisk
 Else
  LowerPoissonLimit = CVErr(2001)
 End If
End If
End Function

Public Function UpperPoissonLimit(Events As Variant, TimeAtRisk As Variant, Optional Alpha As Variant) As Variant
'relates chi-sq to Poisson
'Johnson & Kotz 1969, Ulm in Am J Epidemiol 1990 (131) 373-
'comments by Dobson Stats in Med 1991 (10) 457-
Dim a As Double
Dim ifault As Long
On Error Resume Next
If IsMissing(Alpha) Then
 a = 0.05
Else
 a = Alpha
End If
If Events < 0# Then
 UpperPoissonLimit = CVErr(2001)
ElseIf Events = 0# Then
 UpperPoissonLimit = PPCHI2(1# - (a / 2#), 2#, ifault) / 2#
 If Application.WorksheetFunction.IsNA(UpperPoissonLimit) = False And TimeAtRisk > 0# And ifault = 0 Then
  UpperPoissonLimit = UpperPoissonLimit / TimeAtRisk
 Else
  UpperPoissonLimit = CVErr(2001)
 End If
Else
 UpperPoissonLimit = PPCHI2(1# - (a / 2#), 2# * (Events + 1#), ifault) / 2#
 If Application.WorksheetFunction.IsNumber(UpperPoissonLimit) = True And TimeAtRisk > 0# And ifault = 0 Then
  UpperPoissonLimit = UpperPoissonLimit / TimeAtRisk
 Else
  UpperPoissonLimit = CVErr(2001)
 End If
End If
End Function

Public Function LowerSMRLimit(Observed As Variant, Expected As Variant, Optional Alpha As Variant) As Variant
On Error Resume Next
LowerSMRLimit = LowerPoissonLimit(Observed, 1#, Alpha)
If Application.WorksheetFunction.IsNumber(LowerSMRLimit) = True Then
 LowerSMRLimit = LowerSMRLimit / Expected
Else
 LowerSMRLimit = CVErr(2001)
End If
End Function

Public Function UpperSMRLimit(Observed As Variant, Expected As Variant, Optional Alpha As Variant) As Variant
On Error Resume Next
UpperSMRLimit = UpperPoissonLimit(Observed, 1#, Alpha)
If Application.WorksheetFunction.IsNumber(UpperSMRLimit) = True Then
 UpperSMRLimit = UpperSMRLimit / Expected
Else
 UpperSMRLimit = CVErr(2001)
End If
End Function
```