Implementation of Sampling Inspection Plan Algorithm Using Spread Sheet Functions


[1]Department of Computer Science, S.V.University, Tirupati
[2]Department of Statistics, S.V.University, Tirupati
srikanth.kadainti@gmail.com, Sarma_kvs@rediffmail.com

Abstract
This paper presents an application of Excel functions in Visual Basic environment to determine the decision rules in a statistically designed single sampling plan with rectifying inspection. The objective is to determine the parameters \( n \) and \( c \) of the plan in such a way that the Average Total Inspection (ATI) is minimized. The application is focused on utilizing the inverse functions of Poisson distribution with the help of Chi Square distribution. The code for running the algorithm along with numerical illustrations is presented.

Keywords: Single Sampling Plan, Rectifying inspection, Excel functions.

1. Introduction
A single sampling plan (SSP) is a lot sentencing procedure aimed at disposing a lot of \( N \) items by inspecting a random sample of \( n \) items drawn at from it. Each item, after inspection, is classified as defective or non-defective. If \( d \) denotes the number of defective items in the sample and \( c \) denotes a threshold or critical number (integers) then the lot is rejected if \( d > c \) and accepted otherwise. This is a classical procedure called sampling plan by attributes. Alternatively one can measure a quality characteristic \( X \) on each item in the sample and reject the lot if the sample average of \( \bar{X} > k \) where \( k \) is the critical value. This is called sampling plan by variables. More details on sampling plans can be found in Montgomery [5] and Mittag and Rinne[6]. Determination of the parameters \((n, c)\) of the SSP is based on statistical considerations as well as decision rules allowed in the plan.

The following terminology is widely used by statisticians in the design of sampling plans.

a) Incoming fraction defective \((p')\): The average proportion of defectives in the lots supplied by the vendor over a period of time.

b) Acceptable Quality Level \((p_1')\): The proportion of defectives with which lots are normally acceptable. This is denoted by AQL.

c) Rejectable Quality Level \((p_2')\): The worst case proportion of defectives with which lots are rejected. This is denoted by RQL or Lot Tolerance Percent Defective (LTPD).

d) Producer’s risk \((\alpha)\): The risk involved in wrongly rejecting a lot which has AQL.

e) Consumer's risk \((\beta)\): The risk involved in wrongly accepting a lot which has RQL.

The SSP is specified by the four parameters \((p_1', p_2', \alpha, \beta)\). The values of \( n \) and \( c \) are determined such that

\[
P(d \leq c \mid p_1') \geq (1-\alpha) \quad (1)
\]

\[
P(d \leq c \mid p_2') \leq \beta \quad (2)
\]

The left hand side of the above inequalities denote the cumulative probability up to \( c \) given that the underlying distribution has incoming fraction defective \( p' = p_1' \) or \( p' = p_2' \). This function is called the Operating Characteristic (OC) function and indicates the probability of accepting lot under the given conditions. One way of evaluating (1) and (2) is to use the Type-A OC function basing on Hypergeometric distribution. It is used when the received lot is not from a regular stream of lots supplied by the vendor (isolated lot). When the lots come from a regular stream from a supplier, we can use the Type-B OC function which is based on either Binomial distribution or the Poisson distribution. Mittag and Rinne[6] have discussed several algorithms to determine \( n \) and \( c \) of the SSP of which the following algorithms are important.

a) Guenther’s search algorithm

b) Peach- Litteur search algorithm

c) Graph et al algorithm

d) Hailey procedure

The computational difficulty lies with the determination of the inverse functions for the probability distributions that occur in the OC function. One can use statistical tables to search for the Poisson parameter at a given probability \( \lambda \) but...
it would be only an approximation. Padherla and Sarma[2] have studied the Graf et al algorithm and developed a new
search method called the Modified Graf et al Method to determine \( n \) and \( c \) with the help of spread sheet optimization.
In this paper we reexamine the method of designing a SSP under the conditions of rectifying inspection and bring out
an algorithm by exploiting spread sheet functions. We also present an application in Visual Basic by calling the excel
functions in to the program, instead of working on the spread sheet.

2. Rectifying inspection

The fate of rejected lots is an important issue in production as well as business environment. When the lot is rejected
it is sometimes 100% inspected and all defectives are replaced with good items and the lot size of \( N \) is restored
maintaining zero defectives in it. This is possible only at the manufacturer’s end because good items will be available
for replacement. In some cases as in the case of inventory management, the defective items cannot be replaced leading
to a reduced lot size \( N' \) (< \( N \)).
The advantage of rectifying inspection is that a rejected lot never contains any defectives. All defectives, if any would
appear only in lots that are accepted by the plan. More details of this type of plans can be found in Anderson et al[3],
Wadsworth et al[4]. BalouiJamkhaneh et al [1] have introduced a new acceptance sampling plan where the incoming
fraction of defective of items in the received lots is takes as a fuzzy number.

The following are two important measures of performance related to rectifying inspection.

Average Outgoing Quality (AOQ): Let \( p \) be the fraction defective of the incoming lots. Then the average fraction
defective in the accepted lots is known as average outgoing quality. When the lots are very good then \( p' \) is small and
hence most lots are accepted so that the AOQ is small. When the lots are very bad then \( p' \) is large and many lots are
rejected and rectified. Again the AOQ will be small. There exists some value of \( p' \) at which the AOQ reaches
maximum called Average Outgoing Quality Limit (AOQL)

Average Total Inspection (ATI): The expected number of items inspected per lot before taking a decision during
rectifying inspection plan is called the Average Total Inspection (ATI). Given a lot of size \( N \), it is mandatory to
inspect \( n \) items. When the lot is rejected, with probability \( (1-P_a) \) the remaining \( (N-n) \) items in the lot are all inspected.
The expected load of inspection after the rejection of the lot becomes \( (N-n)(1-P_a) \). Adding this to \( n \) we get

\[
\text{ATI} = n + (N - n)(1 - P_a)
\]

Since \( P_a \) is a function of the incoming fraction defective \( p \) of the lots, the ATI can be mapped on to the values of \( p \)
thereby getting the ATI curve.

3. Procedure to determine the attribute type SSP by rectifying inspection

Given the values of \( N \), RQL and \( \beta \) we wish to determine the parameters \( n \) and \( c \) by using Poisson model for the OC
function with the following algorithm.

Algorithm-1:

a) Start with \( i = 1 \) and take \( c_i = 0 \)
b) Find \( \lambda_i \) at this \( c_i \) by using either Cumulative Poisson tables or Excel function, at the given value of \( \beta \).
c) \( n_i \leftarrow \lambda_i / \text{RQL} \) so that the trial plan will be \((n_i, c_i)\)
d) Find \( P_{ai} \), the probability of acceptance with this \((n_i, c_i)\) using Poisson tables or Excel function.
e) \( \text{ATI}_i = n_i + (1 - P_{ai})(N - n_i) \)
f) \( i \leftarrow i+1 \), \( c_i \leftarrow c_i + 1 \)
g) repeat until ATI reaches a minimum.

The Excel worksheet has a built in function to calculate the cumulative Poisson probability. This method has two
steps, one for computing the ATI and the other for finding the EOQ. Consider the following illustration.

4. The use of Excel functions instead of Poisson tables

The determination of the sampling plan parameters \( n \) and \( c \) depends on the Poisson parameter \( \lambda \) where \( \lambda = n(\text{RQL}) \). At a given value of \( c \) (starting with 0) the value of \( \lambda \) shall be found such that

\[
\sum_{x=0}^{\infty} \frac{e^{-\lambda} \lambda^x}{x!} = \text{RQL}
\]

The following relationship between Poisson distribution and Chi Square distribution exists(Mittag and Rinne (1993))

\[
\sum_{x=0}^{\infty} \frac{e^{-\lambda} \lambda^x}{x!} = \frac{1}{2^{c+1} \Gamma(c+1)} \int_{2\lambda}^{\infty} u^c e^{-u/2} du
\]

where \( \Gamma(c+1) \) is the gamma function.
Writing \( \sum_{x=0}^{c} \frac{e^{-\lambda} \lambda^x}{x!} = \text{POS}(c, \lambda) \) and \( \frac{1}{e^{\lambda}} \int_{0}^{\lambda} u^c e^{-u/2} du = \text{CHI}(2\lambda, 2(c+1)) \) to indicate the cumulative distribution functions it can be shown that 
\[
\text{POS}(c, \lambda) = \text{CHI}(2\lambda, 2(c+1))
\]
It means the cumulative Poisson Distribution with parameter \( \lambda \) up to \( c \) is exactly equal to the complement of cumulative Chi Square distribution up to \( 2\lambda \) with \( 2(c+1) \) degrees of freedom.
Using \( \text{POS}(c, \lambda) = \omega, 0 < \omega < 1 \) we find that \( \text{CHI}(2\lambda, 2(c+1)) = 1-\omega \) so that 
\[
2\lambda = \text{Inverse of } \{\text{CHI}(2(c+1), 1-\omega)\}.
\]
There is a built-in function in Excel to evaluate the inverse of Cumulative Chi Square distribution, denoted by \( \text{CHIINV}(\text{probability}, \text{degrees of freedom}) \). For the given values of \( c \), taking \( 1-\omega = \beta \) we can find the corresponding value of \( \lambda \).
In the following section we have prepared a VB code to perform this iterative procedure. The results are exported to Excel along with chart.

5. Visual Basic program with embedded Excel functions

The inputs for the problem are \( p', \beta, \text{RQL} \). The values of these parameters are defined at the start of the program. The parameters can also be given in interaction mode.
The output is visible on the screen. However by creating a file and using print # command, the output is transferred to a standard excel worksheet. The data on the sheet is updated in the same file every time the data is changed.
In order to create the graph the macro is captured and attached to the VB code with the following steps.
   a) Use Developer menu
   b) Start with record macro option
   c) Create line chart by selecting ATI column on the data axis and c-values on the horizontal axis.
   d) Stop recording macro
   e) View code
   f) Copy the code and paste it either in a command button or as a function to be called.
The code is given in Appendix-I. Normally after the obtaining the number values, one goes to Excel sheet, copy the data and plot the chart. With this additional code for chart, one can get both the results and the chart on the Excel sheet.

Illustration: Consider the parameters \( p' = 0.03, \beta = 0.08, \text{RQL} = 0.10 \) and \( N = 1000 \). The program produces the output as shown in table-1.

<table>
<thead>
<tr>
<th>( c )</th>
<th>( \text{Lambda} )</th>
<th>( n )</th>
<th>( P_a )</th>
<th>( \text{ATI} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2.5257</td>
<td>25</td>
<td>0.4724</td>
<td>539.4426</td>
</tr>
<tr>
<td>1</td>
<td>4.1683</td>
<td>42</td>
<td>0.6411</td>
<td>385.8664</td>
</tr>
<tr>
<td>2</td>
<td>5.6417</td>
<td>56</td>
<td>0.7625</td>
<td>280.2064</td>
</tr>
<tr>
<td>3</td>
<td>7.0342</td>
<td>70</td>
<td>0.8386</td>
<td>220.0622</td>
</tr>
<tr>
<td>4</td>
<td>8.3767</td>
<td>84</td>
<td>0.8885</td>
<td>186.1431</td>
</tr>
<tr>
<td>5</td>
<td>9.6846</td>
<td>97</td>
<td>0.9249</td>
<td>164.8313</td>
</tr>
<tr>
<td>6</td>
<td>10.9665</td>
<td>110</td>
<td>0.949</td>
<td>155.3594</td>
</tr>
<tr>
<td>7*</td>
<td>12.2282</td>
<td>122*</td>
<td>0.9666</td>
<td>151.3349</td>
</tr>
<tr>
<td>8</td>
<td>13.4734</td>
<td>135</td>
<td>0.9771</td>
<td>154.7994</td>
</tr>
<tr>
<td>9</td>
<td>14.7049</td>
<td>147</td>
<td>0.9849</td>
<td>159.8804</td>
</tr>
<tr>
<td>10</td>
<td>15.9247</td>
<td>159</td>
<td>0.99</td>
<td>167.3951</td>
</tr>
<tr>
<td>11</td>
<td>17.1345</td>
<td>171</td>
<td>0.9934</td>
<td>176.4807</td>
</tr>
<tr>
<td>12</td>
<td>18.3356</td>
<td>183</td>
<td>0.9956</td>
<td>186.5833</td>
</tr>
</tbody>
</table>

Table-I: Behavior of ATI against changing values of \( c \)

We have actually a maximum of 20 trials to increment \( c \) and only 12 are shown in the above table since the minimum ATI of 151.3 has occurred at \( c = 7 \). Hence the optimal sampling plan with the above parameters is \( n = 122, c = 7 \). The results when exported to Excel produce the chart shown in Figure-1.
By running the program with different values of $p'$ we have evaluated various sampling plans as shown in table-2

<table>
<thead>
<tr>
<th>$p'$</th>
<th>$c$</th>
<th>$\lambda$</th>
<th>$n$</th>
<th>$P_a$</th>
<th>ATI</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>2</td>
<td>5.6417</td>
<td>56</td>
<td>0.9807</td>
<td>74.26476</td>
</tr>
<tr>
<td>0.02</td>
<td>4</td>
<td>8.3767</td>
<td>84</td>
<td>0.9716</td>
<td>109.9782</td>
</tr>
<tr>
<td>0.03</td>
<td>7</td>
<td>12.2282</td>
<td>122</td>
<td>0.9666</td>
<td>151.3349</td>
</tr>
<tr>
<td>0.04</td>
<td>10</td>
<td>15.9247</td>
<td>159</td>
<td>0.9407</td>
<td>208.8905</td>
</tr>
<tr>
<td>0.05</td>
<td>14</td>
<td>20.7152</td>
<td>207</td>
<td>0.8970</td>
<td>288.6523</td>
</tr>
<tr>
<td>0.06</td>
<td>20</td>
<td>27.7203</td>
<td>277</td>
<td>0.8309</td>
<td>399.2903</td>
</tr>
</tbody>
</table>

Table-2: Different sampling plans at changing value of $p$

Thus we observe that as the value of $p$ increases, the lots become poorer and poorer. Hence the sample size increases rapidly. The average total inspection is also found to increase, since many lots get rejected as $p$ increases. Since this is a rectifying inspection, the cut off $c$ also is found to increase as $n$ increases.

**Conclusions**

The use of Excel statistical functions is a powerful alternative to the use of conventional statistical tables for the designing of sampling plan. When these functions are linked to a program one can design a laboratory experiment to study the performance of the plan by changing the input parameters.

**References:**


Appendix-1: VB Code for the SSP problem
Dim A, D, h As Double
Dim q, q1 As Double
Dim lam, mu, r, r1, ltdemand, pi, mean, pa As Single
Dim term1, term2, term3, cost As Single
Dim n, c, nn As Integer
Dim oxl As Excel.Application
Dim wb As Workbook
Dim ws As Worksheet
Dim var As Variant
Private Sub Command1_Click ()
End
End Sub
Private Sub Command3_Click ()
Cls
p = 0.03 ' incoming fraction defective
beta = 0.08
rql = 0.1
nn = 1000 'lot size
Call ssp
Me.Command1.SetFocus
End Sub
Private Sub ssp()
Print "p = "; p
Print "beta = "; beta
Print "RQL = "; rql
Print "N = "; nn 'lot size
Print
Set oxl = New Excel.Application
Open "D:\vbssp\test.xls" For Output As #1
Set wb = oxl.Workbooks.Open("D:\vbssp\test.xls")
Set ws = wb.Worksheets(1) 'Specify your worksheet name
c = 0
Print #1, "c" & vbTab & "Lambda" & vbTab & "n" & vbTab & "OC" & vbTab & "ATI"
Do While c <= 20
lam = 0.5 * oxl.Application.ChiInv(beta, 2 * (c + 1))
n = Round(lam / rql, 0)
mean = n * p
pa = oxl.Application.Poisson(c, mean, 1)
ATI = n + (1 - pa) * (nn - n)
Print c, Format(lam, "#.####"), n, Format(pa, "#.####"), ATI
Print #1, c & vbTab & Format(lam, "#.####") & vbTab & n & vbTab & Format(pa, "#.####") & vbTab & ATI
End With
With oxl
.Workbooks.Open "d:\vbssp\test.xls"
.Visible = True
.Sheets(1).Select
.Sheets(1).Range("A2:E22").Select
.ActiveWorkbook.SaveAs "d:\vbssp\test.xls"
End With
ActiveSheet.Shapes.AddChart
ActiveChart.ChartType = xlLineMarkers
ActiveChart.SetSourceData Source:=Range("E1:E22")
ActiveChart.SeriesCollection(1).XValues = "=test!$A$2:$A$22"
ActiveChart.PlotArea.Select
ActiveSheet.ChartObjects("Chart 1").Activate
ActiveChart.Axes(xlValue).MajorGridlines.Select
Selection.Delete
ActiveSheet.ChartObjects("Chart 1").Activate
ActiveChart.Legend.Select
Selection.Delete
MsgBox "OK. Continue!!"
'Set ws = Nothing
'Set wb = Nothing
'Set oxl = Nothing
'End With
'Me.Command1.SetFocus
End Sub