

UserForm2 - 1

```
Dim pctCompl As Single
Private Sub UserForm_Activate()
code
End Sub

Public Sub code()
'On Error GoTo ErrorHandler:
Dim sheetNames, segments, title
Dim accidentYear(), developmentYear(), probability(), claimAmounts(), v(), w(), ar() As Double
Dim startDate, endDate, analysisDate As Date
Dim simulations, times, lambda_simulations

sheetNames = Array("Yearly", "Half-Yearly", "Quarterly", "Monthly")
segments = Array(1, 2, 4, 12)
title = Array("Accident Year", "Development Year", "Probability", "Claim Amounts")
startDate = CDate("01/01/2001")
endDate = CDate("31/12/2005")
analysisDate = CDate("31/12/2009")

times = 100
lambda_simulations = InputBox("Enter labmda for number of claims (Ex: 10000 or 1000) ")

pctCompl = 0

ReDim v(0 To 3)          ' Reserve from CLM
ReDim ar(0 To 3)         ' Actual Reserve

' Clear contents of all sheets before starting
For i = 1 To 4
    Sheets(sheetNames(i - 1)).Cells.ClearContents
    Sheets(sheetNames(i - 1)).Cells.ClearFormats
Next

'number of times of simulations loop starts
For jj = 1 To times

    simulations = INVPOISSON(Rnd(), lambda_simulations)
    'simulations = 10200
    ReDim accidentYear(1 To simulations)
    ReDim developmentYear(1 To simulations)
    ReDim probability(1 To simulations)
    ReDim claimAmounts(1 To simulations)

    For i = 1 To simulations      'accident numbers poisson          ' generating data

        accidentYear(i) = WorksheetFunction.RandBetween(startDate, endDate)
        developmentYear(i) = accidentYear(i) + WorksheetFunction.LogNorm_Inv(Rnd(), 5.76957, 1.010
768) '6.7897, 0.6064) (1 year, 1 year)

        probability(i) = Rnd()
        claimAmounts(i) = WorksheetFunction.LogNorm_Inv(probability(i), 8, 0.1)

        ' printing generated raw data
        Worksheets("Data").Cells(i + 1, 1).Value = accidentYear(i)
        Worksheets("Data").Cells(i + 1, 1).NumberFormat = "d/m/yyyy"
        Worksheets("Data").Cells(i + 1, 2).Value = developmentYear(i)
        Worksheets("Data").Cells(i + 1, 2).NumberFormat = "d/m/yyyy"
        Worksheets("Data").Cells(i + 1, 3).Value = probability(i)
        Worksheets("Data").Cells(i + 1, 4).Value = claimAmounts(i)

    Next

    For i = 1 To simulations
        accidentYear(i) = Worksheets("Data1").Cells(i + 1, 1)
        developmentYear(i) = Worksheets("Data1").Cells(i + 1, 2)
        probability(i) = Worksheets("Data1").Cells(i + 1, 3)
        claimAmounts(i) = Worksheets("Data1").Cells(i + 1, 4)
    Next

' finding the last date after simulating data (accident date, reporting date, claim amounts )
```

```

'endDate = CDate("01/01/2001")
'For i = 1 To simulations
'    If developmentYear(i) > endDate Then
'        endDate = developmentYear(i)
'    End If
'Next

```

```

' calculating number of years after simulating data
numberOfYears = Year(endDate) - Year(startDate) + 1

```

```

' calculating Yearly, Half-Yearly, ..... Reserves
For ii = 0 To 3

```

```

    Dim tri As Range
    Dim clm_incremental(), clm_cumulative(), ffactor(), a, b, u, c, d, e, f, res As Double
    Dim rowCLM, colCLM As Integer
    Dim segmentsInAYear, x, y, count, n As Integer

```

```

    segmentsInAYear = segments(ii) ' segments denote half year, quarter year or full
    n = numberOfYears * segmentsInAYear

```

```

    Set tri = Worksheets(sheetNames(ii)).Range("B4")

```

```

'printing heading
count = 0

```

```

Worksheets(sheetNames(ii)).Cells(2, 1) = "Projections in Incremental Data"
Worksheets(sheetNames(ii)).Cells(2, 1).Font.Bold = True
Worksheets(sheetNames(ii)).Cells(n + 6, 1) = "Projections in Cumulative Data"
Worksheets(sheetNames(ii)).Cells(n + 6, 1).Font.Bold = True
Worksheets(sheetNames(ii)).Cells(2, n + 6 + 1) = "Simulated Data"
Worksheets(sheetNames(ii)).Cells(2, n + 6 + 1).Font.Bold = True

```

```

    For i = 0 To (Year(endDate) - Year(startDate))
        For j = 1 To segmentsInAYear
            Worksheets(sheetNames(ii)).Cells(3 + count + j, 1) = CStr(Year(startDate) + i) & "-" & j
            Worksheets(sheetNames(ii)).Cells(3 + count + j, 1).Font.Bold = True
            Worksheets(sheetNames(ii)).Cells(3, 1 + count + j) = count + j
            Worksheets(sheetNames(ii)).Cells(3, 1 + count + j).Font.Bold = True
            Worksheets(sheetNames(ii)).Cells(3 + count + j + n + 4, 1) = CStr(Year(startDate) + i) & "-" & j
            Worksheets(sheetNames(ii)).Cells(3 + count + j + n + 4, 1).Font.Bold = True
            Worksheets(sheetNames(ii)).Cells(3 + n + 4, 1 + count + j) = count + j
            Worksheets(sheetNames(ii)).Cells(3 + n + 4, 1 + count + j).Font.Bold = True
            Worksheets(sheetNames(ii)).Cells(3 + count + j, n + 6 + 1) = CStr(Year(startDate) + i) & "-" & j
            Worksheets(sheetNames(ii)).Cells(3 + count + j, n + 6 + 1).Font.Bold = True
            Worksheets(sheetNames(ii)).Cells(3, 1 + count + n + 6 + j) = count + j
            Worksheets(sheetNames(ii)).Cells(3, 1 + count + n + 6 + j).Font.Bold = True
        Next
        count = count + segmentsInAYear
    Next

```

```

ReDim clm_incremental(1 To n, 1 To n)
ReDim clm_cumulative(1 To n, 1 To n)
ReDim ffactor(1 To n - 1)

```

```

'setting the matrix to 0
For i = 1 To n
    For j = 1 To n
        tri.Cells(i, j).Value = 0
        tri.Cells(i + n + 4, j).Value = 0
        tri.Cells(i, n + 6 + j).Value = 0
        clm_incremental(i, j) = 0
        clm_cumulative(i, j) = 0
    Next
Next

```

```

' Finding Row and Column and cumulating the value
' restricts value to less than 2015

For i = 1 To simulations
    rowCLM = 0
    colCLM = 0
    y = CInt(Year(accidentYear(i))) - CInt(Year(startDate))
    rowCLM = y * segmentsInAYear + 1
    If segmentsInAYear > 1 Then
        x = WorksheetFunction.Floor_Math(CInt(Month(accidentYear(i))) / (12 / segmentsInAYear))
        rowCLM = rowCLM + x
        If (Month(accidentYear(i)) Mod (12 / segmentsInAYear)) = 0 Then
            rowCLM = rowCLM - 1
        End If
    End If

    y = Year(developmentYear(i)) - Year(startDate)
    colCLM = y * segmentsInAYear + 1
    If segmentsInAYear > 1 Then
        x = WorksheetFunction.Floor_Math(CInt(Month(developmentYear(i))) / (12 / segmentsInAYear))
        colCLM = colCLM + x
        If (Month(developmentYear(i)) Mod (12 / segmentsInAYear)) = 0 Then
            colCLM = colCLM - 1
        End If
    End If

    colCLM = colCLM - (rowCLM - 1)

    If developmentYear(i) < endDate Then
        tri.Cells(rowCLM, colCLM).Value = tri.Cells(rowCLM, colCLM).Value + claimAmounts(i)
    End If
    If Year(developmentYear(i)) < Year(accidentYear(i)) + 5 Then ' this prints both CLM upper
        triangle and Simulated rectangle
        tri.Cells(rowCLM, n + 6 + colCLM).Value = tri.Cells(rowCLM, n + 6 + colCLM).Value + claimAmounts(i)
    End If
    'Worksheets("Data").Cells(i, 8) = rowCLM
    'Worksheets("Data").Cells(i, 9) = colCLM
Next

' Copying the cumulative value to the Array
For j = 1 To n
    For i = 1 To n - j + 1
        clm_incremental(i, j) = tri.Cells(i, j).Value
        If j = 1 Then
            tri.Cells(n + 4 + i, j).Value = tri.Cells(i, j).Value
            clm_cumulative(i, j) = tri.Cells(n + 4 + i, j).Value
        Else
            tri.Cells(n + 4 + i, j).Value = tri.Cells(i, j).Value + tri.Cells(n + 4 + i, j - 1).Value
            clm_cumulative(i, j) = tri.Cells(n + 4 + i, j).Value
        End If
    Next
Next

' Finding forward factors

For j = 2 To n
    a = 0
    b = 0
    For i = 1 To n - j + 1
        a = a + clm_cumulative(i, j)
        b = b + clm_cumulative(i, j - 1)
    Next
    ' to prevent overflow error
    If b = 0 Then
        ffactor(j - 1) = 0
    Else
        ffactor(j - 1) = a / b
    End If
Next

```

```

tri.Cells(2 * n + 6, 0) = "Forward Factors"
tri.Cells(2 * n + 6, 0).Font.Bold = True
For i = 1 To n - 1
    tri.Cells(2 * n + 6, 1 + i) = ffactor(i)
    tri.Cells(2 * n + 6, 1 + i).Font.Bold = True
Next

```

' Formatting the lower triangle

```

For j = 2 To n
    For i = n To n - j + 2 Step -1
        clm_cumulative(i, j) = clm_cumulative(i, j - 1) * ffactor(j - 1)
        tri.Cells(n + 4 + i, j) = clm_cumulative(i, j)
        tri.Cells(n + 4 + i, j).Interior.Color = RGB(245, 222, 179)
        tri.Cells(i, n + 6 + j).Interior.Color = RGB(245, 222, 179)
    Next
Next

```

```

u = 0
v(ii) = 0
res = 0

```

```

tri.Cells(0, n + 2).Value = "Projection by Future Year Segments"
tri.Cells(0, n + 2).Font.Bold = True
tri.Cells(n + 4, n + 2).Value = "Outstanding"
tri.Cells(n + 4, n + 2).Font.Bold = True

```

'calculating reserve from cumulative triangle

```

For i = 1 To n
    u = clm_cumulative(i, n) - clm_cumulative(i, n - i + 1)
    tri.Cells(n + 4 + i, n + 2).Value = u
    v(ii) = v(ii) + u
Next

```

```

tri.Cells(2 * n + 6, n + 1).Value = "Reserve"
tri.Cells(2 * n + 6, n + 1).Font.Bold = True
tri.Cells(2 * n + 6, n + 1).Interior.Color = RGB(255, 215, 0)
tri.Cells(2 * n + 6, n + 2).Value = v(ii)
tri.Cells(2 * n + 6, n + 2).Font.Bold = True
tri.Cells(2 * n + 6, n + 2).Interior.Color = RGB(255, 215, 0)

```

'printing lower half of incremental triangle

```

For j = n To 2 Step -1
    For i = 2 + n - j To n
        tri.Cells(i, j).Value = tri.Cells(n + 4 + i, j).Value - tri.Cells(n + 4 + i, j - 1).Value
        tri.Cells(i, j).Interior.Color = RGB(245, 222, 179)
        clm_incremental(i, j) = tri.Cells(i, j).Value
    Next
Next

```

```

c = 0
d = 0
e = 0
f = 0

```

'calculating reserve from incremental triangle

```

For i = 2 To n
    For j = i To n
        c = c + clm_incremental(j, n - j + i)
        e = e + tri.Cells(j, n + 6 + n - j + i)
    Next
    tri.Cells(i, n + 2).Value = c
    d = d + c
    f = f + e
    c = 0
    e = 0
Next

```

' Finding Actual Reserve

```

ar(ii) = 0
ar(ii) = f

```

```

tri.Cells(n + 2, n + 1).Value = "Reserve"
tri.Cells(n + 2, n + 1).Font.Bold = True
tri.Cells(n + 2, n + 1).Interior.Color = RGB(255, 215, 0)
tri.Cells(n + 2, n + 2).Value = d
tri.Cells(n + 2, n + 2).Font.Bold = True
tri.Cells(n + 2, n + 2).Interior.Color = RGB(255, 215, 0)

tri.Cells(n + 2, n + 4 + n + 1).Value = "Reserve"
tri.Cells(n + 2, n + 4 + n + 1).Font.Bold = True
tri.Cells(n + 2, n + 4 + n + 1).Interior.Color = RGB(255, 215, 0)
tri.Cells(n + 2, n + 4 + n + 2).Value = f
tri.Cells(n + 2, n + 4 + n + 2).Font.Bold = True
tri.Cells(n + 2, n + 4 + n + 2).Interior.Color = RGB(255, 215, 0)

'Exit Sub
'
'ErrorHandler:
'
'UserForm1.Hide
'MsgBox "Please enter valid inputs"
'UserForm1.Show
Next                ' end of ii loop

Worksheets("Results").Cells(1 + jj, 1).Value = jj
Worksheets("Results").Cells(1 + jj, 1).Font.Bold = True

'printing 4 reserves (only 99 values)

Worksheets("Results").Cells(1 + jj, 6).Value = f ' printing Actual Reserve
For i = 1 To 4
Worksheets("Results").Cells(1 + jj, i + 1).Value = v(i - 1) ' printing CLM reserve
Worksheets("Results").Cells(1 + jj, i + 6).Value = (v(i - 1) - ar(i - 1)) ^ 2 ' Printing the Error
'If v(i - 1) < 0 Then
'    Exit Sub
'End If

If jj < times Then
    Sheets(sheetNames(i - 1)).Cells.ClearContents
    Sheets(sheetNames(i - 1)).Cells.ClearFormats
End If

Next

' for progress bar

If times >= 100 Then
    If jj Mod times / 100 = 0 Then
        pctCompl = pctCompl + 1
        progress pctCompl
    End If
End If

Next                ' end of jj loop

'printing titles in result sheet & Data Sheet
Worksheets("Results").Cells(1, 6).Value = "Simulated Reserve"
Worksheets("Results").Cells(1, 6).Font.Bold = True
For i = 1 To 4
Worksheets("Results").Cells(1, i + 1).Value = sheetNames(i - 1)
Worksheets("Results").Cells(1, i + 1).Font.Bold = True

Worksheets("Results").Cells(1, i + 6).Value = sheetNames(i - 1) & " (Error %)"
Worksheets("Results").Cells(1, i + 6).Font.Bold = True
Next

For q = 1 To 4
Worksheets("Data").Cells(1, q).Value = title(q - 1)
Worksheets("Data").Cells(1, q).Font.Bold = True
Next

```

```
Unload UserForm2
```

```
End Sub
```

```
Sub progress(pctCompl As Single)
```

```
UserForm2.Text.Caption = pctCompl & "% Completed"
```

```
UserForm2.Bar.Width = pctCompl * 2
```

```
DoEvents
```

```
End Sub
```