**CIS 550 DECISION SUPPORT SYSTEMS (Winter 2015)**

**FINAL PROJECT REPORT**



**RELIABILITY ANALYSIS**

**Instructor: Dr. Suleyman Uludag**

**Team:**

***Nagamani Kummari***

***Divya teja Annapareddy***

***Vijay Kumar Dharmaji***

***Priyanka Reddy Cherabudla***

**Date: April 7th 2015**

**PREFACE:**

1. Abstract
2. Problem Description
3. Model
4. Spreadsheets
5. User Interface
6. Procedures and Reports
7. How to resolve the problem?
8. Conclusion
9. References

**ABSTRACT:**

Reliability can be defined as the freedom from failure of a piece of equipment while maintaining a specific performance. There are some of the reasons to perform a reliability study: to improve product, to make the product more profitable, to avoid or minimize cost penalties through failure, to avoid or reduce the chance of causing harm to people or sometimes to the environment. Reliability is an essential component of safety and it will underlie the decision making process.

Reliability analysis is an application development model problem that estimates the system failure time of a system of machines.

**PROBLEM DESCRIPTION:-**

In this application, we consider a series parallel system of machines in which there are three machine types each working in parallel and connected serially. Any time one system type fails, entire system fails.

Failure rate is a measure used to ascertain reliability of a product or a process. For products that must be replaced because they are dysfunctional, a relevant measure of interest is the ***mean time to failure***(MTTF), and for products that may be repaired and put to service again, a relevant measure of interest is ***mean time between failures*** (MTBF). The information about MTTF and MTBF helps discover reliability. However, it is time consuming and costly to collect enough data to build a probability distribution and cumulative distribution curve of time to failure or time between failures. One way we can deal with this problem is by analyzing a smaller set of data and identifying a distribution that approximates the distribution of time to failure, such as the Exponential, Weibull, or Gamma distributions. Once we have identified a distribution, we can easily calculate failure rate, MTTF, etc.

The aim of this project is to build a decision support system that would enable the user to estimate the reliability of products. For this project, we use a simple example and describe a number of statistical analysis tools that can be used.

**MODEL:-**

To determine failure, we consider each machine type to have a ***k out of n***system. Given there are ***n***machines in a machine type, any time ***k*** of these ***n***machines fail, the entire machine type fails. We then consider the entire system of all machine types to also be a ***k*** *out of* ***n***system. In this application, we assume n= 3 machine types and *k* = 1. That is, any time one machine type fails, the entire system fails.

We use the Weibull distribution to estimate the failure times of the machines in each machine type.

The following notation is used:

*f(t)* probability of failure at time *t*

*F(t)* probability of failure up to time *t*

*R(t)* probability of surviving up to time *t* ()

*r(t)* failure rate at time *t* .

In the case the distribution function of failure is approximated by the Exponential distribution:



 is the mean and standard deviation of the exponential distribution. Note that if the mean and standard deviation for a data set are not equal, the corresponding distribution is not exponential. In the context of our problem, λ is the failure rate and is the mean time between failures. Note that the failure rate is constant.

In the case that the distribution function of failure is approximated by the Weibull distribution: 

Parameters *α* and *β* define the scale and shape of this distribution. The expected value for Weibull distribution is , and the variance is

.

We simulate the system failure time using these Weibull parameters. For this application, we analyze the distribution of the system failure time. We also try to identify which machine type causes the most system failure. We try to improve the machine type and overall performance. The output sheet for this analysis includes a histogram of the system failure time with the mean system failure time as well as a histogram of the frequency with which a particular machine type caused the system failure.

**Inputs:**

Using the model described in the above section, we can do the following input.

* Number of machines per machine type (*n*)
* Number of machine which cause failure per machine type (*k*)
* Cost per machine for each machine type
* Mean and standard deviation of time to failure for each machine type
* Number of runs to perform in the simulation

We use one user form to receive the first four input values from the user. We also use one Input Box to record the number of runs to perform. We keep the input values on a table in the simulation sheet to enable the user to modify them at any time.

**Output:**

Our Output can be defined as follows:

* Optimal Weibull parameters per machine type
* Mean system failure time
* Histogram of system failure times from simulation
* Histogram of frequency with which each machine type caused system failure from the simulation.

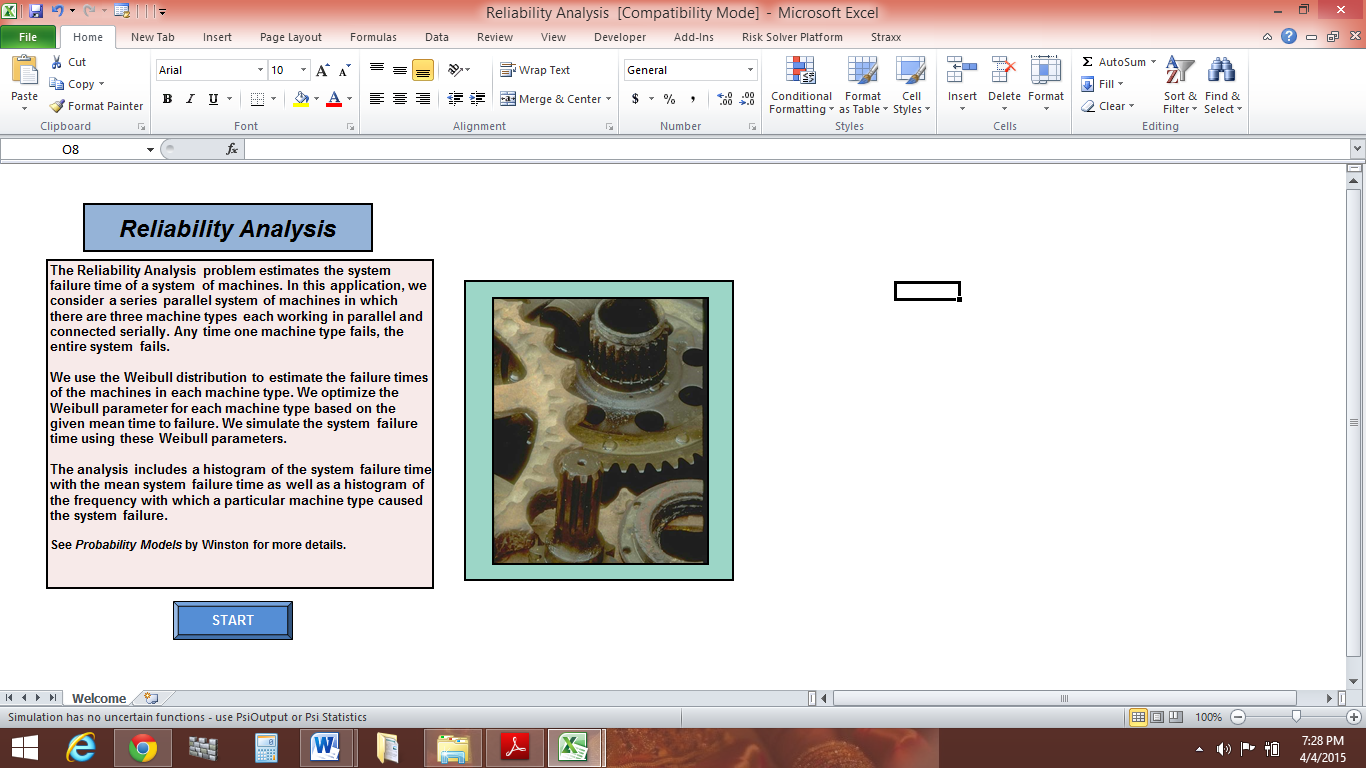
**SPREAD SHEETS**

We use six Spread Sheets in our Project:

1. Welcome Sheet
2. Calculation Sheet
3. Hidden Calculation Sheet
4. Simulation Sheet
5. Third Calculation Sheet
6. Output Sheet

**START**

The***Welcome Sheet*** provides a report of the Project and has a button allocated to the main procedure.

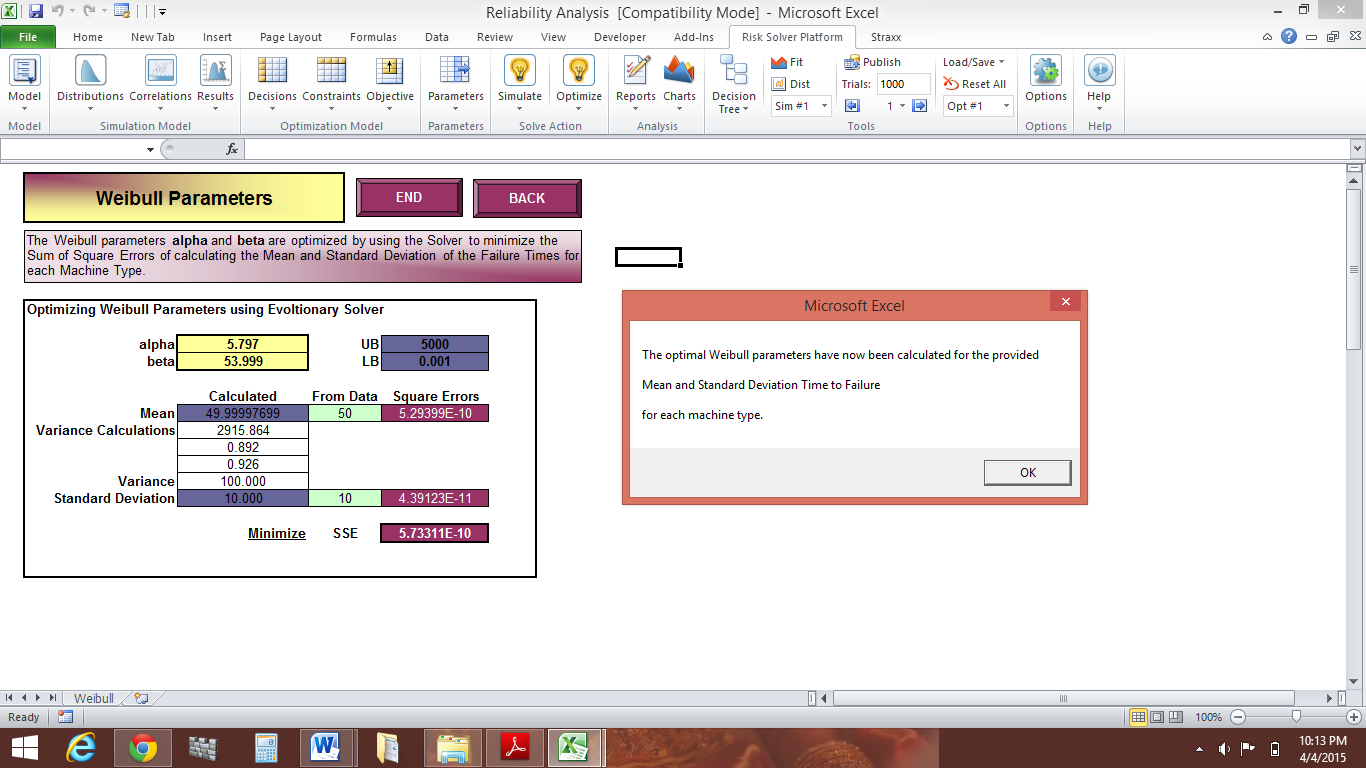


The ***Calculation sheet*** is used to optimize the Weibull parameters for each machine type. Optimization is performed using Risk Solver platform. The input cells for this Optimization are Mean and Standard deviation for time to failure for each machine type. Each machine type’s Weibull parameters are fitted one at a time.

The decision variables are the Weibull parameters Alpha (**α**) and Beta (**β**). Both of these have Upper bound (**UB**) and Lower bound **(LB)** shown adjacent to their cells. We then calculate Mean and Standard deviation time to failure using these Alpha and Beta to find the Square Error compared to the user’s input for these values. The Objective function is therefore to minimize the Square errors. We used two buttons in this sheet: and The **END** button returns the user to welcome sheet and the **BACK** button is used to return to the output sheet once the application is completed.

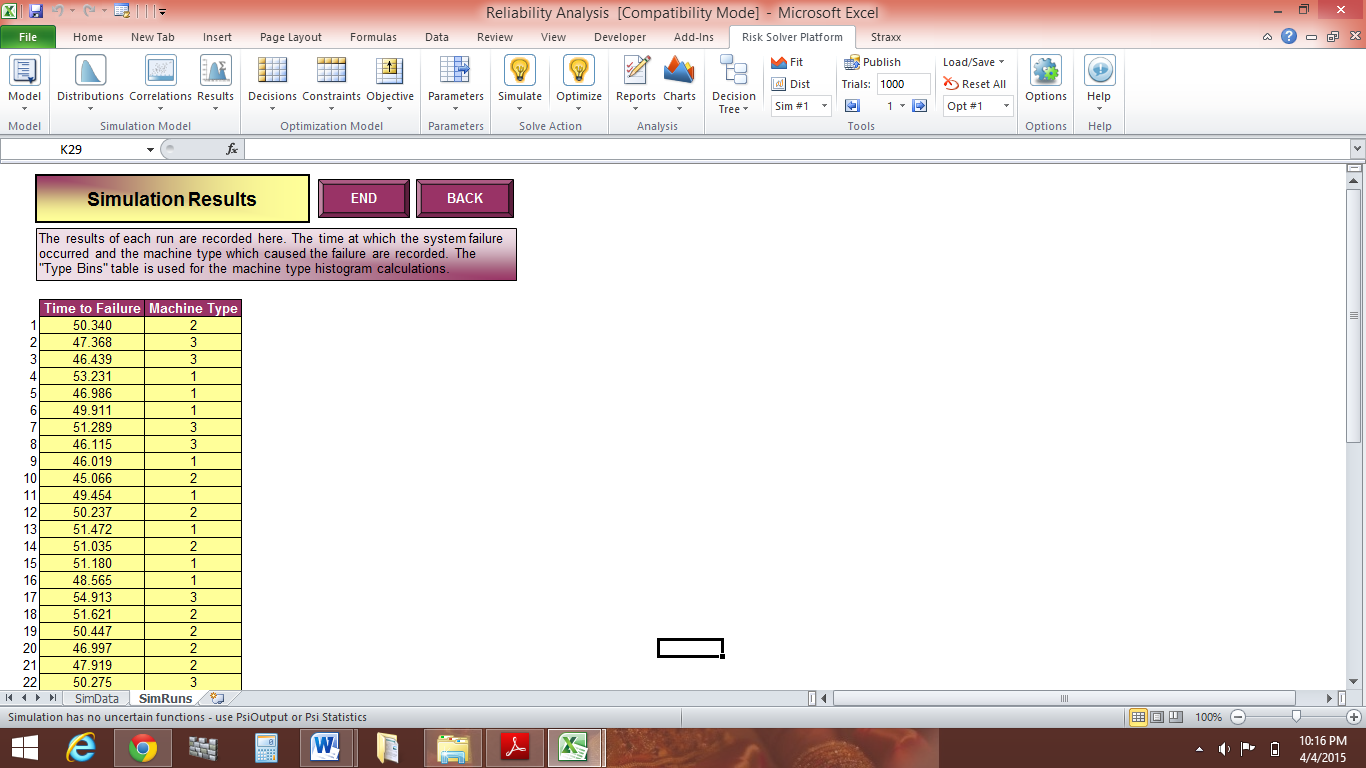
**BACK**

**END**



***Calculation sheet for optimizing the Weibull parameters***

The ***Hidden calculation sheet*** is used to create a timeline of failure times for each run of the simulation. Machine number and machine type is used in the animation. The time failure for each machine on all machine type is recorded.



***Hidden Calculation Sheet***

There are two buttons plus a hidden button on this sheet. The button brings the user to the welcome sheet and the button calls the procedure which begins the

**START SIMULATION**

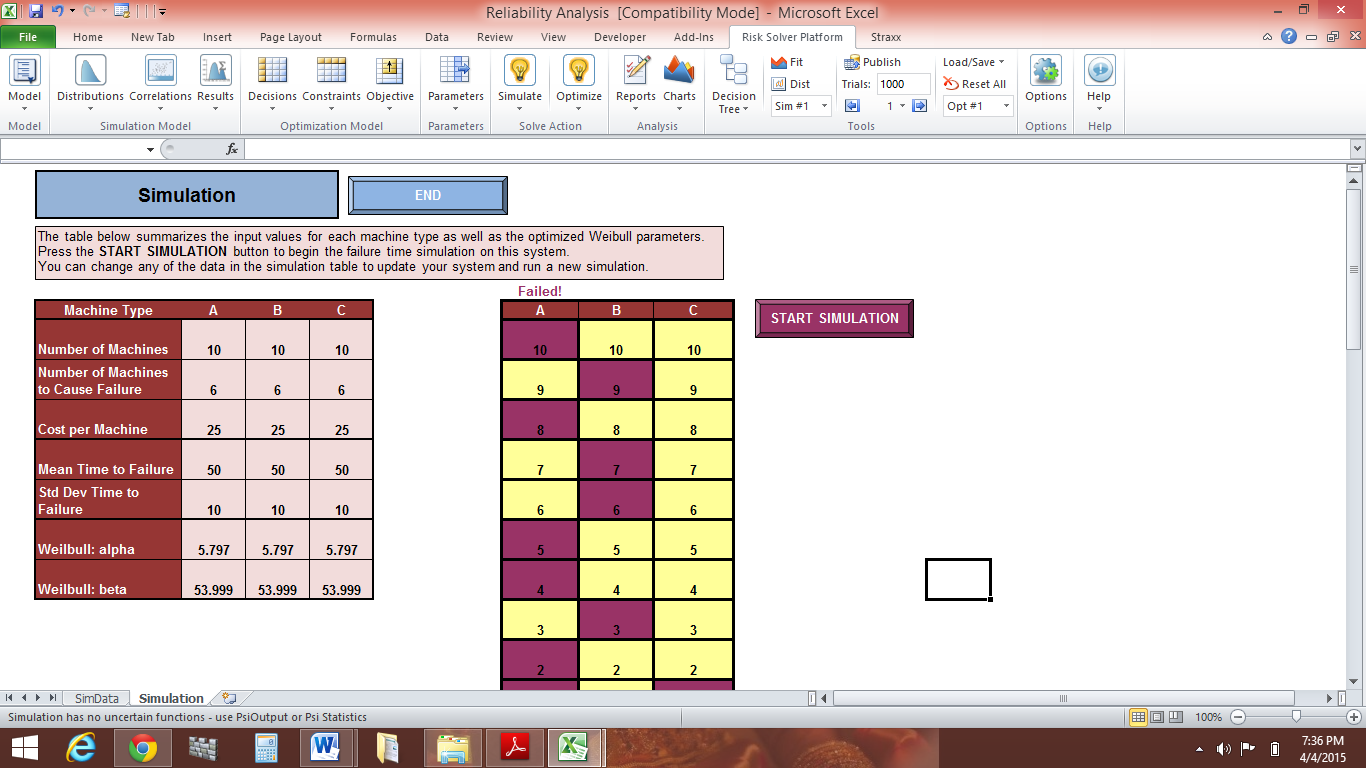
**END**

Simulation. The hidden button is the button which is made visible once

**VIEW ANALYSIS**

the simulation has been completed. It takes the user to the output sheet.

The ***simulation sheet*** contains an input table and the simulation animation layout.



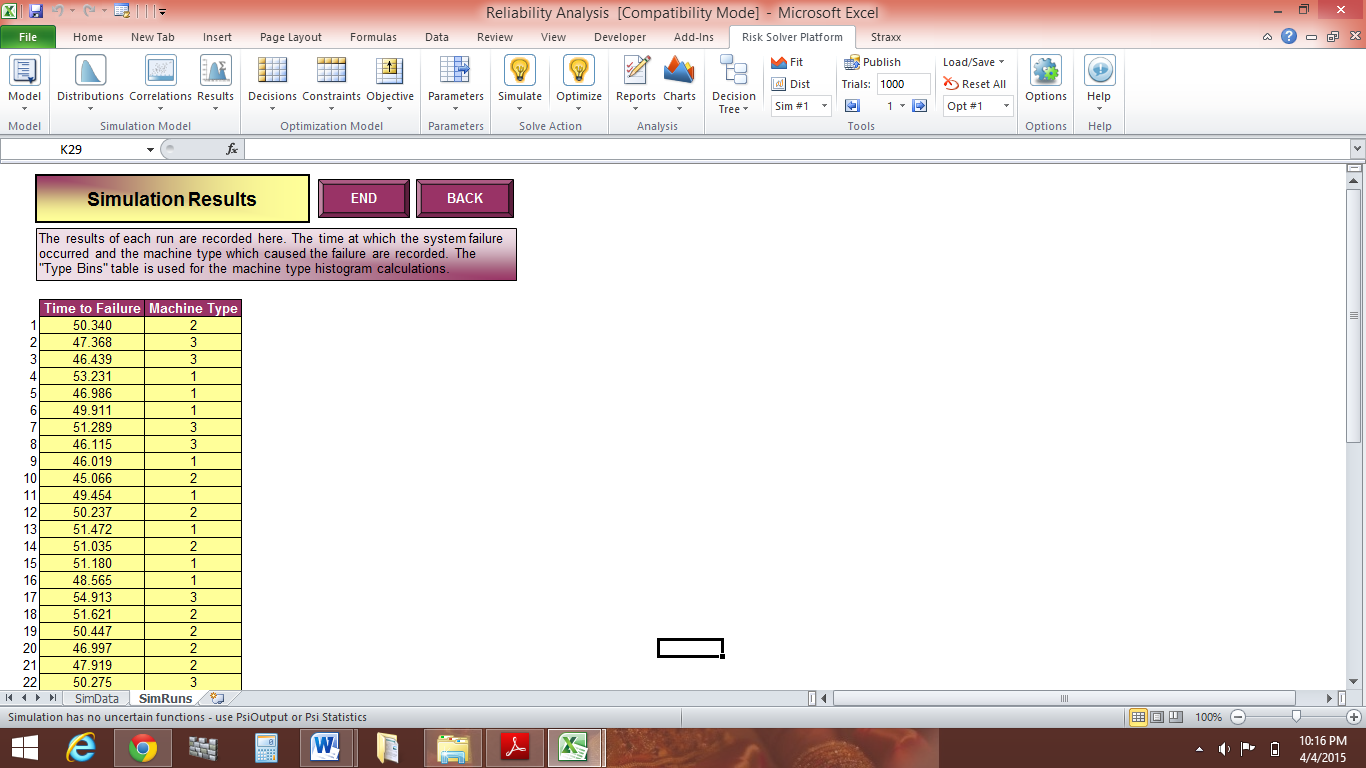
***Simulation Sheet***

The animation layout shows all of the machines for each machine type. As the simulation is run, a failed machine will change to red in the animation. If a particular machine type reaches its respective k number of failed machines, then a system failure occurs. The word **Failed!** Appears above the machine type which caused the system failure.

The ***Third calculation sheet*** contains the results of each run of the simulation. The time until the system failed is recorded along with the machine type which caused the system failure. The table for bin values is used later in creating the histogram for the machine types. There are two buttons on this sheet: and they have the same functionalities as the buttons on the

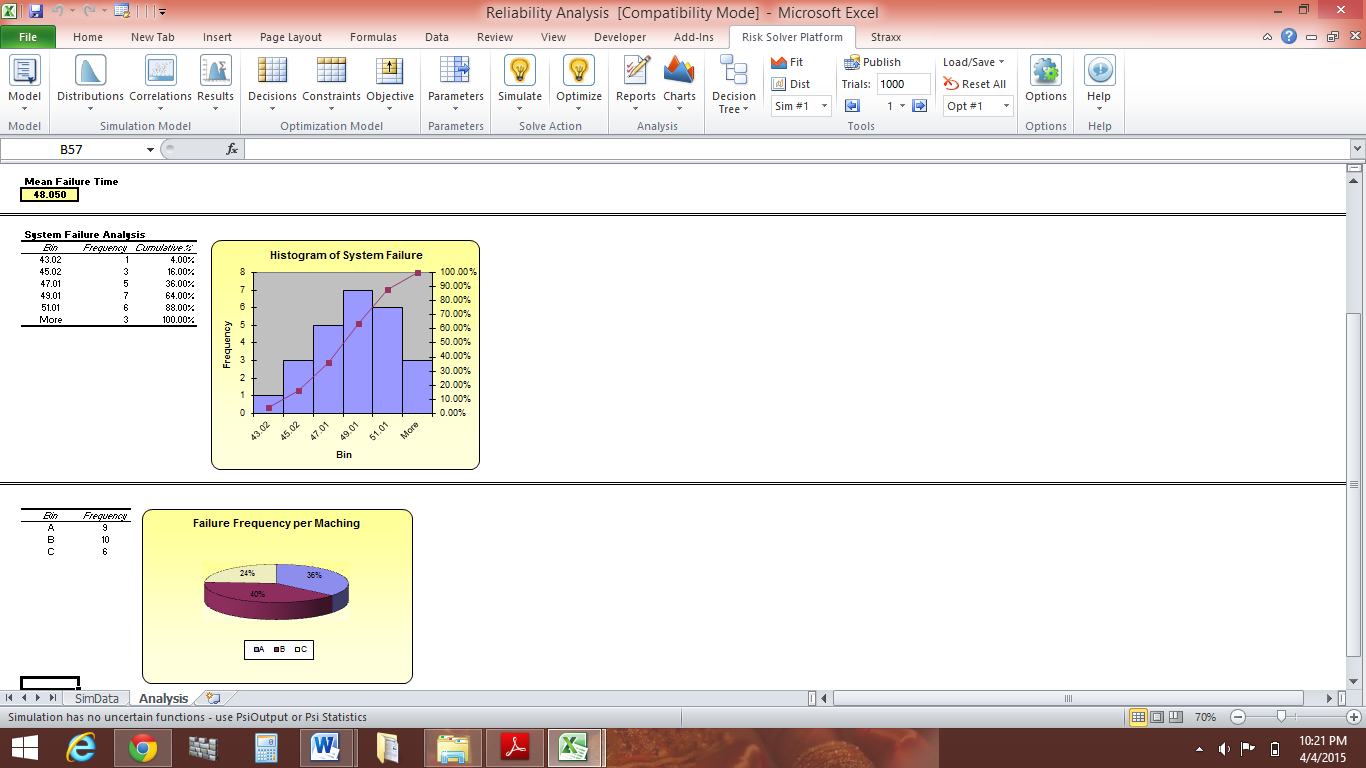
**BACK**

**END**



***Third Calculation Sheet***

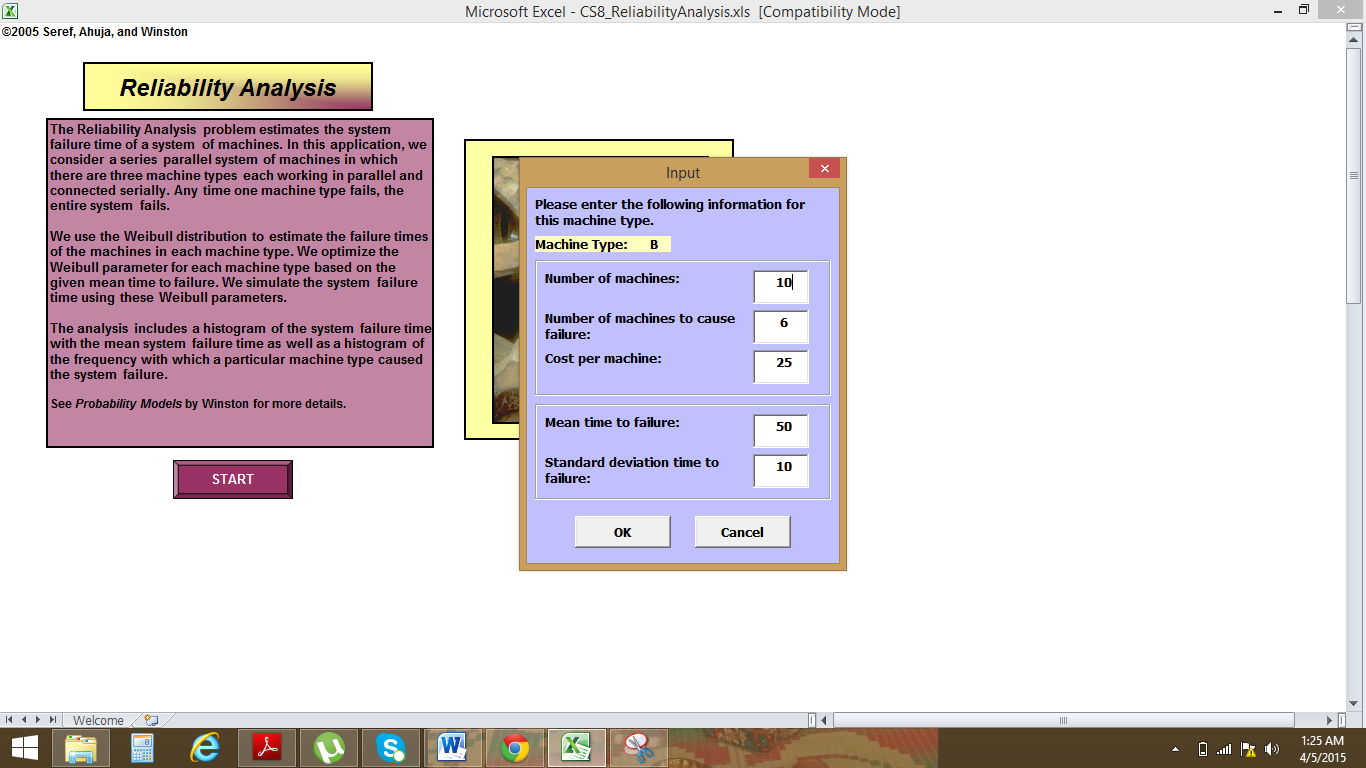
The final sheet is the ***output sheet*** . This sheet displays the mean system failure time, a histogram of the system time failures from the simulation, and a histogram of the frequencies with which each machine type caused the system failure. This is the most important information for the user to analyze to determine how the system is behaving.



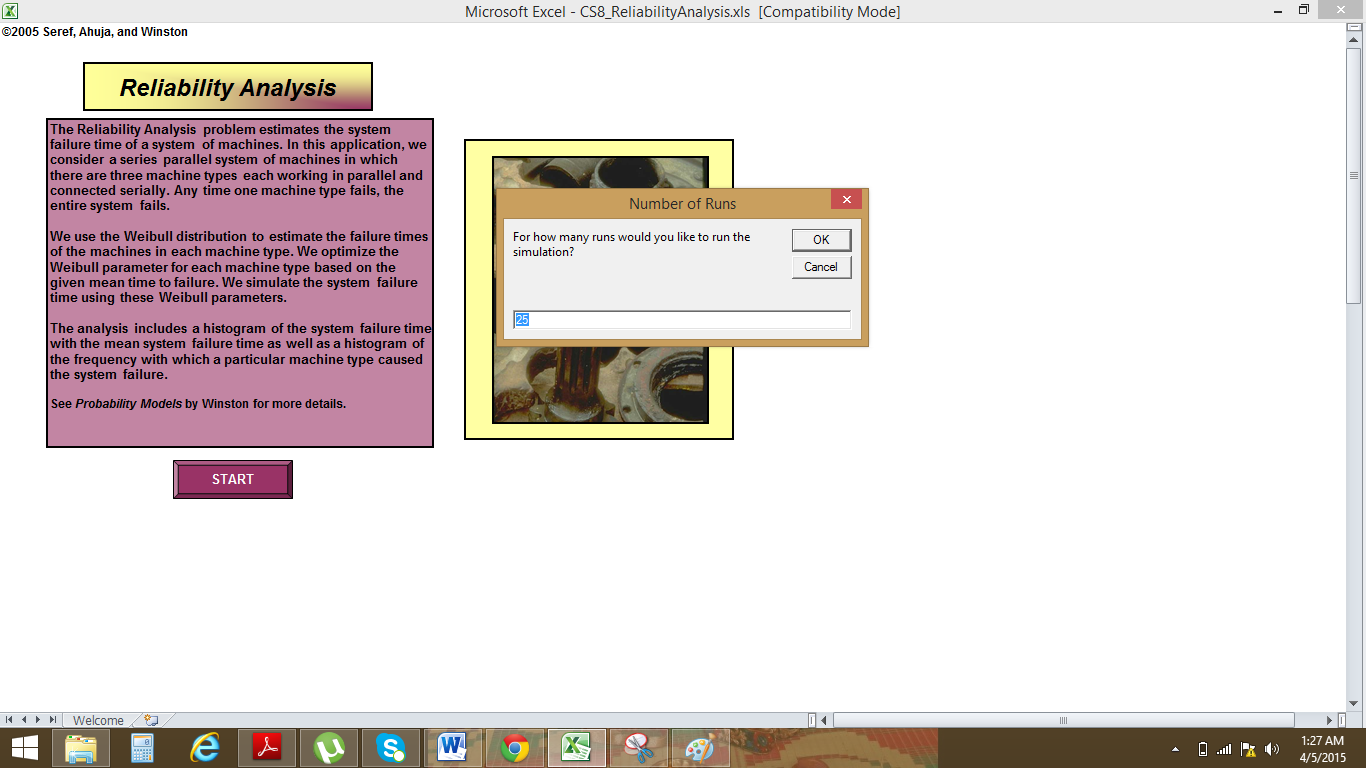
**User interface:**

For this application, we have one user form, one input table, several navigational buttons, and a few functional buttons. We also use one Input Box and one Message Box. The user form contains input for each machine type. It prompts the user for the number of machines, number of machines which cause failure, cost per machine, and the mean and standard deviation of time to failure for the machine type. We use two frames to group similar text boxes.

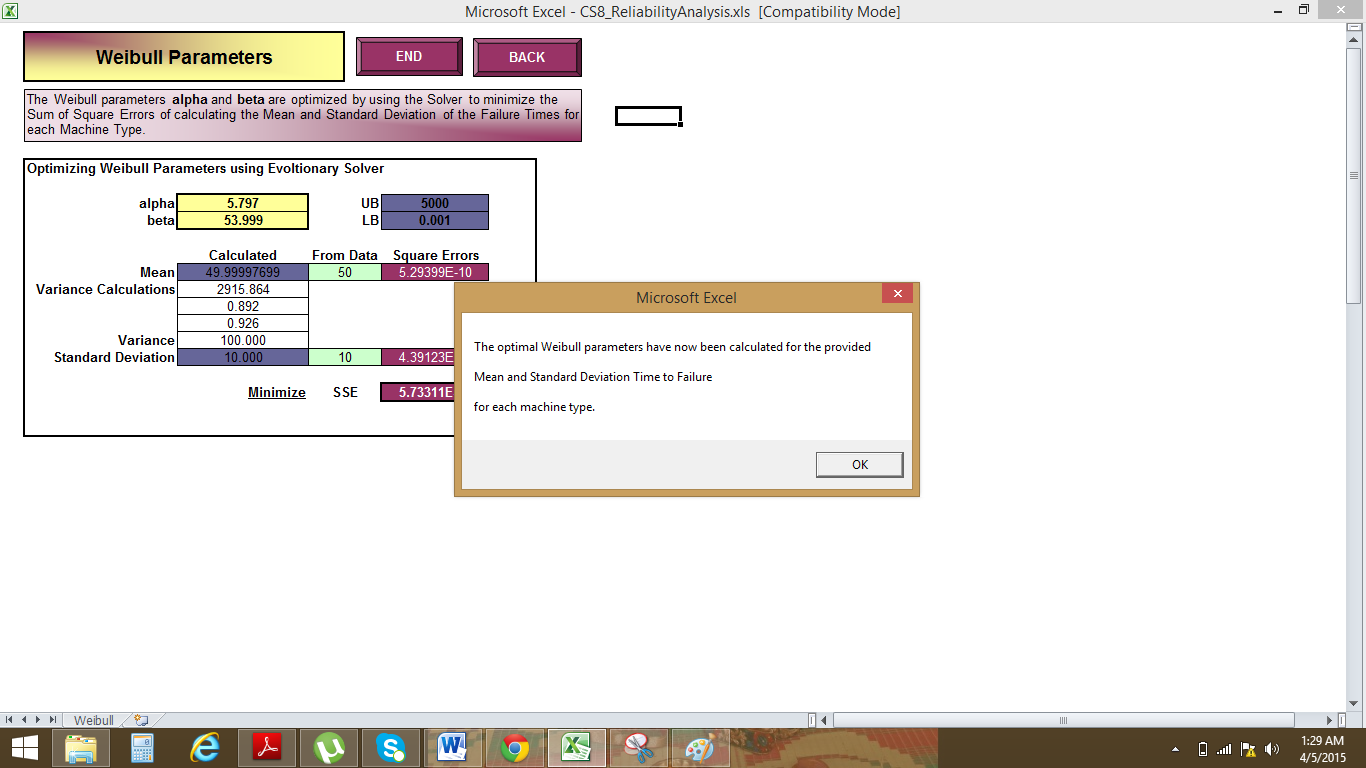
The input table on the simulation sheet has been described in the previous section. It simply summarizes the input provided by the user in the user form as well as the optimized Weibull parameters. The user can change this input before the simulation is run. The navigational buttons and functional buttons were also discussed with each corresponding worksheet. The “OK” button has an associated Click procedure which we describe later.



The Input Box is used to prompt the user for the number of runs for which the simulation should be run. Input box value can be given by user.



The Message Box is used to inform the reader that the Weibull parameter optimization has been completed. The corresponding calculation sheet appears behind the Message Box. We take the user directly to the simulation sheet not allowing them to pause on the calculation sheet.



**Procedures:-**

In this Project we begin with Main Sub procedure and variable definitions. The variables showed defined as public variables are used in multiple procedures.

Step 1:- The Main Procedure begins by calling **ClearPrev procedure** and is used to initialize and clear worksheet values. It also clears formatting for animation layout.

Step 2: From the second screenshot of welcome sheet we can see that we have “OK” command button. This **Click event procedure** of the “OK” command button performs some error checking and then assigns the input values to the corresponding array variables.

***Code for the Main Procedure and variable declarations:-***

Option Explicit

Option Base 1

Public i As Integer, j As Integer, temp As Variant, runs As Integer, \_

MachName(3) As String, NumMach(3) As Integer, NumToFail(3) As Integer, \_

MachCost(3) As Double, MeanFail(3) As Double, StdFail(3) As Double, NumRuns As Integer, \_

Alpha(3) As Double, Beta(3) As Double, Ratio(3) As Double, \_

Resolving As Boolean, ChangeSysFail(3) As Double, BestChoice As Variant

Sub Main()

'Assigned to "Start" button on welcome sheet;

'initializes application

Call ClearPrev

'get input for all machine types

For i = 1 To 3

frmInput.lblType.Caption = MachName(i)

frmInput.Show

With Range("SimTable")

Offset(1, i).Value = NumMach(i)

Offset(2, i).Value = NumToFail(i)

Offset(3, i).Value = MachCost(i)

Offset(4, i).Value = MeanFail(i)

Offset(5, i).Value = StdFail(i)

End With

Next i

'find number of runs

temp = InputBox("For how many runs would you like to run the simulation?", "Number of Runs", 25)

If temp = "" Then

End

Else

NumRuns = temp

End If

'calc Weibull parameters for each machine type

Worksheets ("Weibull").Visible = True

Worksheets ("Welcome").Visible = False

Call CalcWeibull

MsgBox "The optimal Weibull parameters have now been calculated for the provided " \_

& vbCrLf & vbCrLf & "Mean and Standard Deviation Time to Failure " & vbCrLf & vbCrLf & \_

"for each machine type."

Call PrepSim

Worksheets("Simulation").Visible = True

Worksheets("Weibull").Visible = False

End Sub

***Code for ClearPrev procedure:-***

Sub ClearPrev()

'Called from Main procedure;

'initializes variables and clears previous values and formatting

MachName(1) = "A"

MachName(2) = "B"

MachName(3) = "C"

'clear simulation animation

With Range(Range("SimAnim").Offset(1, 0), Range("SimAnim").Offset(50, 2))

.Interior.ColorIndex = xlNone

.Borders(xlInsideVertical).LineStyle = xlNone

.Borders(xlInsideHorizontal).LineStyle = xlNone

.ClearContents

End With

Worksheets("Simulation").Activate

ActiveSheet.Shapes("ViewAnalysis").Visible = False

'clear simulation data

Range(Range("TimeLine").Offset(1, 0), Range("TimeLine").Offset(1,2).End(xlDown)).ClearContents

'clear simulation run results

Range("TimeHist").ClearContents

Range("TypeHist").ClearContents

Resolving = False

End Sub

***Step 3***:-Then **CalcWeibull procedure** is then called. This procedure is used to find optimum Weibull parameters for each machine type. Here we set decision variables, constraints and objective function cells using solver.

***Code for CalcWeibull procedure:-***

Sub CalcWeibull()

'Called from Main procedure;

'uses mean failure time and standard deviation of failure time as input;

'uses Solver to optimize Weibull parameters

Application.ScreenUpdating = False

SolverReset

SolverOk SetCell:=Range("ObjFunc"), MaxMinVal:=2, ByChange:=Range("DecVar")

SolverAdd CellRef:=Range("DecVar"), Relation:=1, FormulaText:=Range("UB")

SolverAdd CellRef:=Range("DecVar"), Relation:=3, FormulaText:=Range("LB")

'for each machine type, enter mean and stdev fail time to Solver sheet

For i = 1 To 3

Range("UserMean").Value = MeanFail(i)

Range("UserStd").Value = StdFail(i)

'starting guess

Range("Beta").Value = MeanFail(i)

Range("Alpha").Value = StdFail(i) / 10

'run solver --- do not need to reset changing cells, obj func, or constraints

SolverSolve UserFinish:=True

SolverFinish KeepFinal:=True

Alpha(i) = Range("alpha").Value

Beta(i) = Range("beta").Value

Next i

For i = 1 To 3

'put resulting parameter values in simulation table

Worksheets("Simulation").Range("SimTable").Offset(6, i).Value = Alpha(i)

Worksheets("Simulation").Range("SimTable").Offset(7, i).Value = Beta(i)

Next i

Application.ScreenUpdating = True

End Sub

***Step 4:-*** On the simulation sheet “start simulation” button is used to call the **StartSim procedure** which is the main simulation procedure. This procedure initializes the animation, creates simulation data.

***Code for StartSim procedure***:-

1. Beginning of Startsim procedure:-

Sub StartSim()

'Assigned to "Start Simulation" button on simulation sheet;

'runs simulation with animation

Dim SysFail As Boolean, Failed(3) As Integer, FailType As Integer

'record any changes to input in table on simulation sheet and

'reflect changes in animation layout

Call PrepSim

For runs = 1 To NumRuns

'generate failure times for the run

Call CreateData

For i = 1 To 3

Failed(i) = 0

Next i

j = 1

SysFail = False

(b) End of startsim procedure:

Application.ScreenUpdating = False

'run through timeline created in CreateData procedure

Do While SysFail = False

Select Case Range("TimeLine").Offset(j, 1).Value

'color failed machine in appropriate machine type column;

'check if machine type failure / system failure has occured

Case 1

Range("SimAnim").Offset(NumMach(1) - Range("TimeLine").Offset(j, 2).Value + 1, 0).Interior.ColorIndex = 54

Failed(1) = Failed(1) + 1

If Failed(1) = NumToFail(1) Then

SysFail = True

FailType = 1

Range("Failure").Cells(1).Value = "Failed!"

End If

Case 2

Range("SimAnim").Offset(NumMach(2) - Range("TimeLine").Offset(j, 2).Value + 1, 1).Interior.ColorIndex = 54

Failed(2) = Failed(2) + 1

If Failed(2) = NumToFail(2) Then

SysFail = True

FailType = 2

Range("Failure").Cells(2).Value = "Failed!"

End If

Case 3

Range("SimAnim").Offset(NumMach(3) - Range("TimeLine").Offset(j, 2).Value + 1, 2).Interior.ColorIndex = 54

Failed(3) = Failed(3) + 1

If Failed(3) = NumToFail(3) Then

SysFail = True

FailType = 3

Range("Failure").Cells(3).Value = "Failed!"

End If

End Select

j = j + 1

Loop

'record info when system fails = system failure time and machine type which caused failure

Worksheets("SimRuns").Range("Runs").Offset(runs, 0).Value = Range("TimeLine").Offset(j, 0).Value

Worksheets("SimRuns").Range("Runs").Offset(runs, 1).Value = FailType

If Resolving = False Then

'only animate if not resolving

Application.ScreenUpdating = True

Application.Wait (Now() + TimeValue("00:00:02"))

End If

Range(Range("SimAnim").Offset(1,0),

Range("SimAnim").Offset(50, 2)).Interior.ColorIndex = 36

Range("Failure").ClearContents

Next

If Resolving = False Then

'only update output sheet if not resolving

Call AnalysisPrep

Worksheets("Simulation").Activate

ActiveSheet.Shapes("ViewAnalysis").Visible = True

End If

End Sub

***Step 5:-***To create simulation data we need to call **CreateData procedure** which generates failure times for each machine in each machine type. These values are generated using Weibullnv function procedure.

***Code for CreateData procedure and Weibullnv function procedure:-***

Sub CreateData()

'Called from StartSim procedure;

'generates failure time values for all machines in each machine type based on Weibull parameters;

'generates a timeline of machine failures with associated machine type

Dim last As Integer

Application.ScreenUpdating = False

Worksheets("SimData").Visible = True

'create the timeline

last = 1

For i = 1 To 3

For j = 1 To NumMach(i)

'generate fail times for each machine

Range("TimeLine").Offset(last, 0).Value = WeibullInv(Alpha(i), Beta(i))

Range("TimeLine").Offset(last, 1).Value = i

Range("TimeLine").Offset(last, 2).Value = j

last = last + 1

Next j

Next i

'sort the timeline

Range(Range("Timeline").Offset(1, 0), Range("Timeline").Offset(1, 1).End(xlDown)).Name = "TimeLineData"

Range("TimeLineData").Sort key1:=Range("Timeline"), order1:=xlAscending

Worksheets("SimData").Visible = True

Application.ScreenUpdating = True

End Sub

Function WeibullInv(Alpha, Beta)

'Function called from CreateData procedure;

'generates a random Weibull value

Dim Prob As Double

Prob = Rnd()

WeibullInv = Beta \* (Log(1 / (1 - Prob))) ^ (1 / Alpha)

End Function

**How Can we Resolve this problem?**

There are two resolve options for this application.

1. Allows user to modify the input table in simulation sheet and rerun the simulation which can be done by clicking on “ Return Simulation” on output sheet.
2. It enables the user to determine which machine type has failed and add one of this machine type to improve the system performance i.e, to increase the mean system failure time. This can be done by clicking on “Resolve” option.

***Code for resolve procedure :-***

Sub Resolve()

'Assigned to "Resolve" button on output sheet;

'adds one machine to each machine time (sequentially) and

'reruns the simulation to record change in mean system failure time;

'compares ratio of this time change to the cost per machine of the machine type;

'suggests that the user adds one machine to the machine type with the largest ratio

Dim MaxRatio As Double

Resolving = True

Application.ScreenUpdating = False

For temp = 1 To 3

'add 1 machine of type i

NumMach(temp) = NumMach(temp) + 1

'repeat simulation

Call StartSim

ChangeSysFail(temp) = Application.WorksheetFunction.Average(Range("TimeHist")) - Range("MeanSysFail").Value

Ratio(temp) = ChangeSysFail(temp) / MachCost(temp)

'reset values

NumMach(temp) = NumMach(temp) - 1

Next

'find max ratio

MaxRatio = 0

For i = 1 To 3

If Ratio(i) > MaxRatio Then

MaxRatio = Ratio(i)

BestChoice = MachName(i)

End If

Next i

Application.ScreenUpdating = True

frmResolve.Show

End Sub

'navigational procedures

Sub EndProg()

Worksheets("Welcome").Visible = True

ActiveSheet.Visible = False

End Sub

Sub ViewRuns()

Worksheets("SimRuns").Visible = True

ActiveSheet.Visible = False

End Sub

Sub GoBack()

Worksheets("Analysis").Visible = True

ActiveSheet.Visible = False

End Sub

Sub ViewWeibull()

Worksheets("Weibull").Visible = True

ActiveSheet.Visible = False

End Sub

Sub ReturnSim()

Worksheets("Simulation").Visible = True

ActiveSheet.Visible = False

End Sub

**Conclusion:**

Reliability analysis has a very important part to play in ensuring that current and future developments, particularly in potential risk analysis with the confidence that risk is well under control and that we continue to learn not through the accidents but with thorough use of information at a lower level than accidents. Reliability reduces the mean time to failure with good maintenance cost.

**References:**

1. M. Hanna, R. Ahuja, and W. Winston. “*Developing Spreadsheet-Based Decision Support Systems*” *(In Press)* Duxbury Press.
2. Frankel.G “ Reliability Analysis” 1962 Naval Engineers Journal.
3. Mood, A.M., Graybill, F.A., Boes, D.C., “Introduction to the Theory of Statistics.” *McGraw Hill*, 3rd Ed., 1974.
4. Winston, L.W., “Operations Research: Applications and Algorithms.” *Duxbury Press*, 3rd Ed., 1994