



LUND UNIVERSITY

Användargränssnitt för proaktiv störningshantering för utilities

Carlson, Martin

2012

Document Version:
Förlagets slutgiltiga version

[Link to publication](#)

Citation for published version (APA):

Carlson, M. (2012). *Användargränssnitt för proaktiv störningshantering för utilities*. (Technical Reports TFRT-7623). Department of Automatic Control, Lund Institute of Technology, Lund University.

Total number of authors:

1

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

ISSN 0280-5316
ISRN LUTFD2/TFRT--7623--SE

Användargränssnitt för proaktiv störningshantering för utilities

Martin Carlson

Lund University
Department of Automatic Control
August 2012

Lund University Department of Automatic Control Box 118 SE-221 00 Lund Sweden		<i>Document name</i> Internal Report	
		<i>Date of issue</i> August 2012	
		<i>Document Number</i> ISRN LUTFD2/TFRT--7623--SE	
<i>Author(s)</i> Martin Carlson		<i>Supervisor</i> Charlotta Johnsson, Dept. of Automatic Control, Lund University, Sweden Anna Lindholm, Dept. of Automatic Control, Lund University, Sweden	
		<i>Sponsoring organization</i>	
<i>Title and subtitle</i> Användargränssnitt för proaktiv störningshantering för utilities			
<i>Abstract</i>			
<i>Keywords</i>			
<i>Classification system and/ or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i> 0280-5316		<i>ISBN</i>	
<i>Language</i> Swedish		<i>Number of pages</i> 1-27	<i>Recipient's notes</i>
<i>Security classification</i>			

Innehåll

Syfte.....	2
Användarmanual.....	2
Programkod.....	4
Appendix A.....	7
Figur 1.....	3
Figur 2.....	4
Figur 3.....	5
Figur 4.....	6
Figur 5.....	6

Syfte

Syftet med detta projekt har varit att skapa ett Grafical User Interface (GUI) i Excel för Utility Disturbance Management (UDM) i en process industri.

Användarmanual

1.

I steg 1 kan man ladda in ny data på två sätt beroende på om man använder sig av Mac eller PC.

PC:

1. Tryck på "Clear old data from worksheet"
2. Spara och stäng exceldokumentet (UserInterface-UDM.xlsm)
3. Starta Matlab och kör i tur och ordning m-filerna "import123" och "load123". När man kör "import123" så öppnas en browser och man får välja vilken data man vill ladda in. När man kör "load123" så laddas datan in i GUI:t, samt number of utilities (nu) och number of samples (ns). Här är det viktigt att m-filerna tillsammans med GUI:t (UserInterface-UDM.xlsm) ligger i samma directory.
4. Starta sedan exceldokumentet igen (UserInterface-UDM.xlsm)

Mac:

1. Tryck på "Clear old data from worksheet"
2. - (behöver ej stänga Excel-dokumentet)
3. Starta Matlab och kör i tur och ordning m-filerna "import123" och "load123_Mac". När man kör "load123_Mac" skapas en CSV-fil med namnet data.csv. Öppna flik "Data 2" i GUI:t, markera ruta A1 och välj "Import..." i File-menyn. Tryck "Import" (CSV-fil är markerat) och välj filen "data.csv". Tryck "Finish" och "OK".
4. Skriv in number of utilities (nu) och number of samples (ns) manuellt.

2.

I detta steg väljer man number of areas (na) och samplings intervallet (ts), alltså hur ofta mätningarna har gjorts. Oavsätt om nu och ns har skrivits in automatiskt eller manuellt så är det en förutsättning att dessa inte ändras på för att analysen ska gå att genomföra. När alla dimensioner är valda trycker man på OK.

3.

I detta steg kan man välja att skriva in namnet på de utilities och produkter som är valda. När detta är gjort trycker man på OK. Det går även att hoppa över detta steg och trycka på OK med en gång. Alla utilities och produkter kommer då istället att anges som siffror, vilket är default.

4.

I detta steg skriver man in information om varje produkt, i form av kvantitet (qm) och marginal (p).

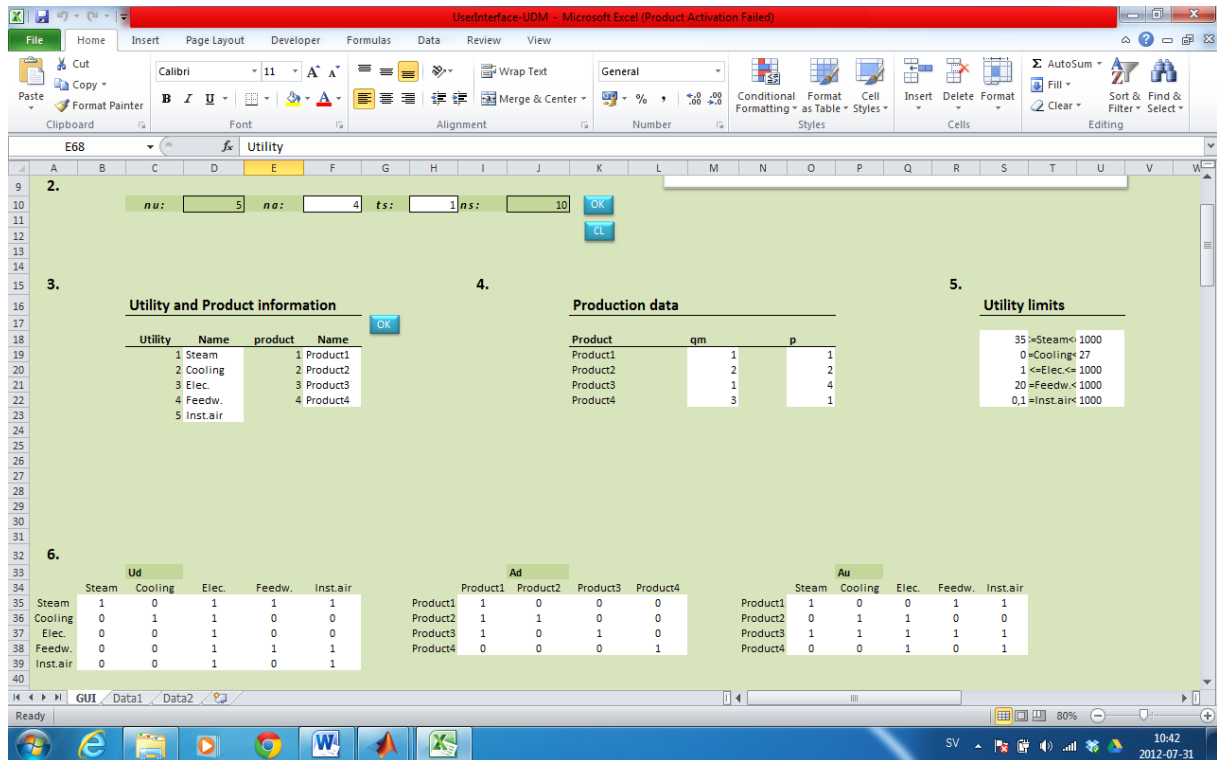
5.

I detta steg ställer man in utility limits, alltså inom vilket intervall varje utility är igång.

6.

I detta steg fyller man i matriserna, utility dependence (Ud), area dependence (Ad) och area-utility dependence (Au).

När steg 3-6 är gjorda ska det se ut på följande sätt:

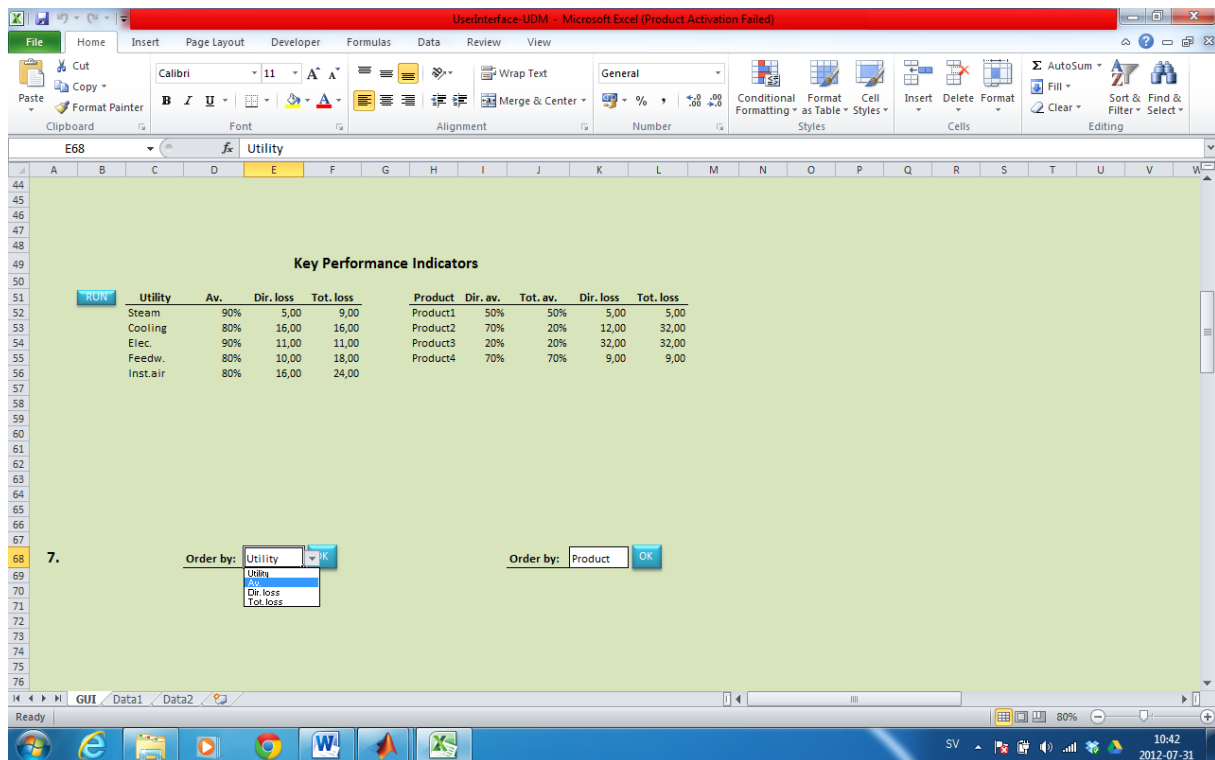


Figur 1

För att sedan köra själva analysen så trycker man på knappen "RUN", som finns till vänster om rubriken "Key Performance Indicators".

7.

När analysen är gjord kommer resultatet upp under rubriken "Key Performance Indicators". I steg 7 finns då möjligheten att sortera i storleksordning efter olika parametrar, såsom utility/product, availability och dir loss, som man kan välja mellan i en lista (se figur 2). När detta är gjort så trycker man på OK.



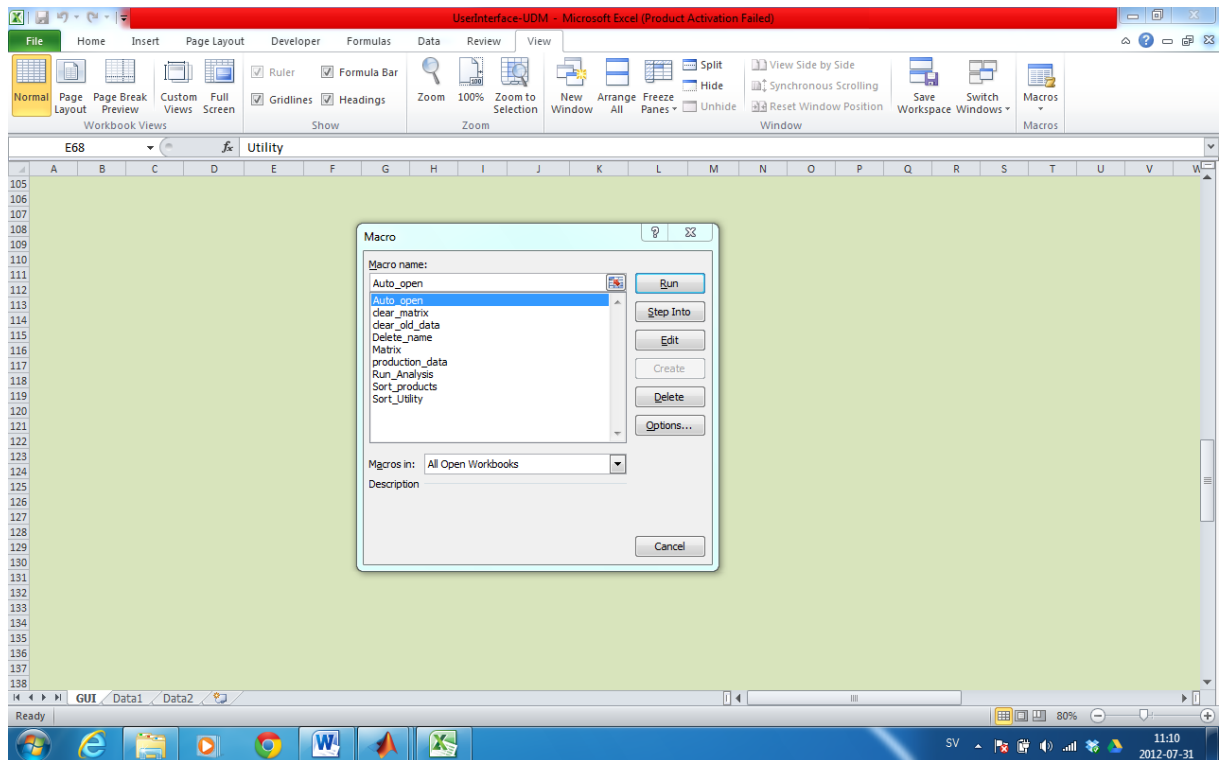
Figur 2

När analysen är gjord kan man gå tillbaka till steg 2 och trycka på knappen "CI" för att börja om från början. Eller så kan man gå tillbaka till steg 3,4,5 och ändra information, för att sedan trycka "RUN" en gång till, om man hellre vill göra små ändringar än att börja om helt från början.

Programkod

I Excel skapas all kod i ett program som heter Visual Basic for Applications (VBA). I detta program är det vanligast att man väljer att dela upp koden i olika makron för att då ha möjlighet att exekvera delar av koden med hjälp av exempelvis knappar. Detta gör det väldigt fördelaktigt att skapa ett GUI i Excel.

Väljer man "View" och sedan "Macros" så kommer en lista upp (se figur 3) med alla makro som är skapade i exceldokumentet.

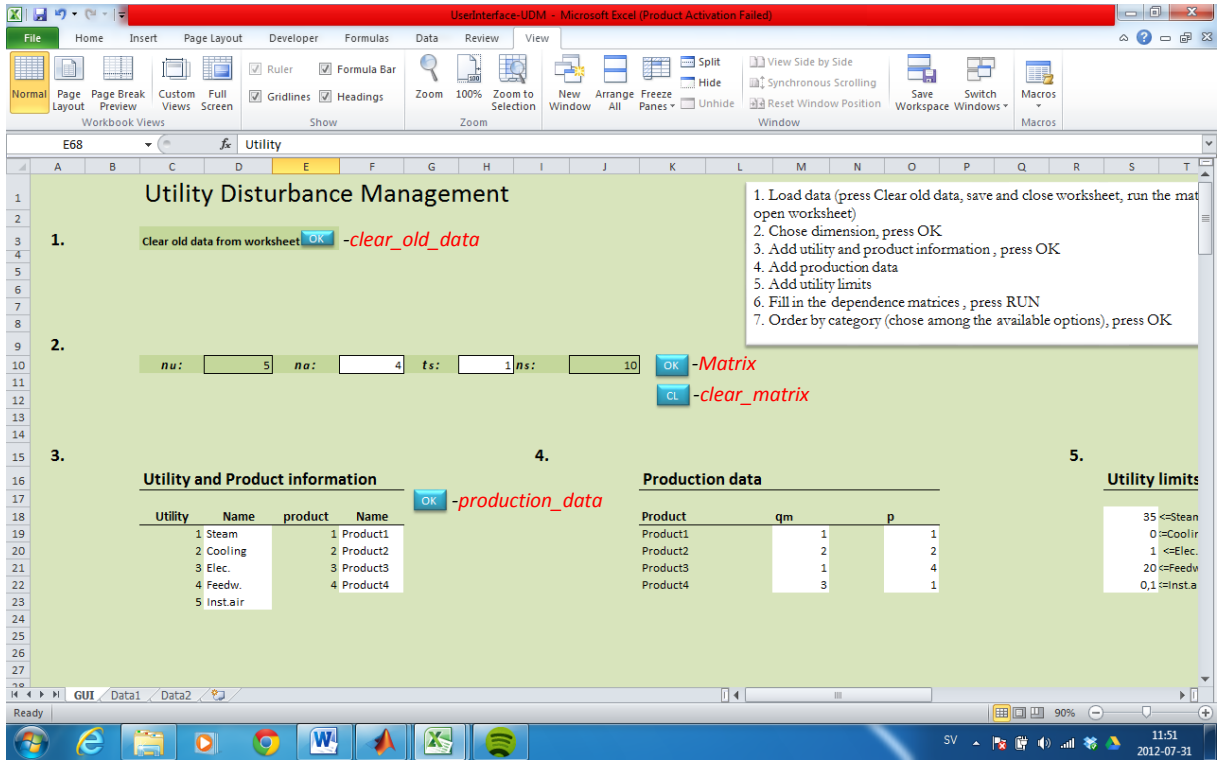


Figur 3

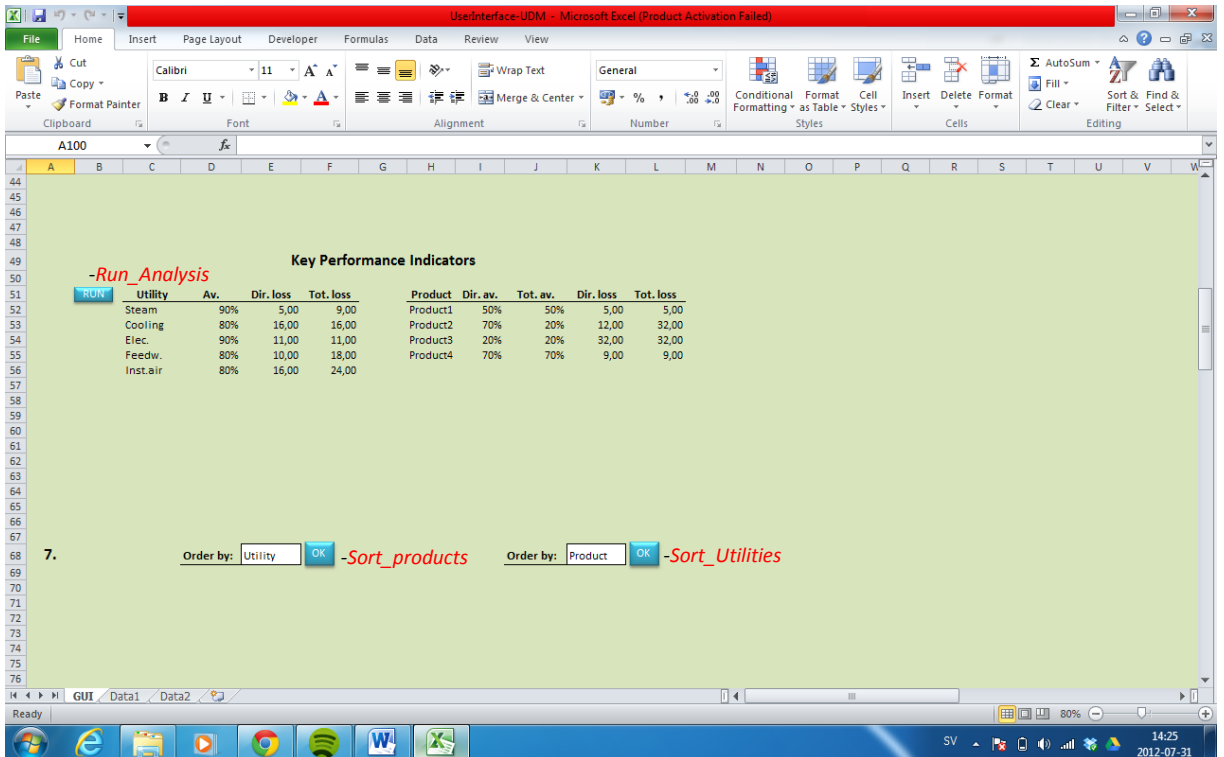
Alla makron sparas i en eller flera moduler och i det här fallet så ligger samtliga makron i modul2. Vill man redigera ett makro så markerar man ett i listan och trycker på edit.

- *Auto_open*: Gör så att det första Excel arket (GUI) alltid visas då UserInterface-UDM.xlsm öppnas oavsett i vilket ark den var sparad senast.
- *clear_matrix*: Återställer GUI:t till default läge.
- *clear_old_data*: Raderar alla rådata från GUI:t.
- *Delete_name*: Raderar namnen på alla områden som Excel skapar då analysen genomförs.
- *Matrix*: Återställer GUI:t till default läge och skapar sektion 3.
- *production_data*: Skapar sektionerna 4-6.
- *Run_Analysis*: Gör analysen.
- *Sort_products*: Sorterar produkterna efter ett antal olika parametrar i steg 7.
- *Sort_Uilities*: Sorterar utilities efter ett antal olika parametrar i steg 7.

Figur 4 och 5 visar vilket makro som är kopplat till varje knapp.



Figur 4



Figur 5

Appendix A

```
Module2 - 1

Sub Matrix()
Application.ScreenUpdating = False

Dim xrow As Integer
Dim nr_samples As Integer

xrow = Cells(10, 4).Value
nr_samples = Cells(10, 10).Value

Dim Rng As Range

'-----Load data from Data2-----

Sheets("Data2").Select

Cells(1, 1).Select

ActiveSheet.UsedRange.Select
Set Rng = Selection

ActiveWorkbook.Names.Add Name:="Radata", RefersTo:=Rng

Cells(100, 1).Select
Sheets("GUI").Select

'-----Clear old contents-----

Range("A33:AV48").Select
Selection.ClearContents
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorAccent3
    .TintAndShade = 0.599993896298105
    .PatternTintAndShade = 0
End With

Range("C19:F32").Select
Selection.ClearContents
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorAccent3
    .TintAndShade = 0.599993896298105
    .PatternTintAndShade = 0
End With

Range("K19:O32").Select
Selection.ClearContents
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorAccent3
    .TintAndShade = 0.599993896298105
    .PatternTintAndShade = 0
End With

Range("S17:U32").Select
Selection.ClearContents
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorAccent3
    .TintAndShade = 0.599993896298105
    .PatternTintAndShade = 0
End With

Cells(200, 1).Select
Selection.ClearContents

'-----Making Utility and Product information-----
```

Module2 - 2

Dim yrow As Integer

yrow = Cells(10, 6).Value

```
For utilitynr = 1 To xrow
For productnr = 1 To yrow
Cells(18 + utilitynr, 3) = utilitynr
Cells(18 + utilitynr, 3 + 1) = "Add"
Cells(18 + utilitynr, 3 + 1).Select
```

```
With Selection.Interior
.Pattern = xlSolid
.PatternColorIndex = xlAutomatic
.ThemeColor = xlThemeColorDark1
.TintAndShade = 0
.PatternTintAndShade = 0
End With
```

```
Cells(18 + productnr, 5) = productnr
Cells(18 + productnr, 5 + 1) = "Add"
Cells(18 + productnr, 5 + 1).Select
```

```
With Selection.Interior
.Pattern = xlSolid
.PatternColorIndex = xlAutomatic
.ThemeColor = xlThemeColorDark1
.TintAndShade = 0
.PatternTintAndShade = 0
End With
Next
Next
```

'-----Making diagonal matrix used in the calculation of the Uud-----

```
For a1 = 1 To xrow
For a2 = 1 To xrow

If a1 = a2 Then
Cells(150 + a1, (1 + a2)) = "1"
Else
Cells(150 + a1, (1 + a2)) = "0"
End If

Next
Next
```

```
Range("A100").Select
Application.ScreenUpdating = True
```

```
End Sub
Sub clear_matrix()
'
```

```
Application.ScreenUpdating = False
On Error Resume Next
'
```

```
Sheets("Data2").Select
Cells(200, 1).Select
Sheets("GUI").Select
```

```
Range("A33:AV48").Select
Selection.ClearContents
With Selection.Interior
.Pattern = xlSolid
.PatternColorIndex = xlAutomatic
.ThemeColor = xlThemeColorAccent3
.TintAndShade = 0.599993896298105
.PatternTintAndShade = 0
```

Module2 - 3

```
End With

Range("C19:F32").Select
Selection.ClearContents
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorAccent3
    .TintAndShade = 0.599993896298105
    .PatternTintAndShade = 0
End With

Range("S18:U32").Select
Selection.ClearContents
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorAccent3
    .TintAndShade = 0.599993896298105
    .PatternTintAndShade = 0
End With

Range("K19:O32").Select

Selection.ClearContents
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorAccent3
    .TintAndShade = 0.599993896298105
    .PatternTintAndShade = 0
End With

Range("B151:S185").Select
Selection.ClearContents

Range("C52:L67").Select
Selection.ClearContents

Sheets("Data1").Select
Cells.Select
Selection.ClearContents
Range("A1").Select

    Sheets("Data2").Select

Range("ONES").Select
    Selection.ClearContents

Range("B52").Select
Selection.ClearContents
Range("A100").Select

Sheets("GUI").Select

Cells(200, 1).Select
Selection.ClearContents

    Call Delete_name

    Cells(68, 5) = "Utility"
    Cells(68, 11) = "Product"

Range("A100").Select

Application.ScreenUpdating = True

End Sub

Sub Auto_open()
    ,
```

```

Module2 - 4

' Auto_open Makro
'
'
    Sheets("GUI").Select
    Range("A1").Select

End Sub

Sub production_data()
Application.ScreenUpdating = False

Dim products As Integer
Dim xrow As Integer
Dim Rng As Range
Dim nr_samples As Integer

products = Cells(10, 6).Value
xrow = Cells(10, 4).Value
nr_samples = Cells(10, 10).Value

Cells(200, 1).Select

'-----If this module already is used then this part jumps to spechandler which is
handeling the renaming of the utilities and products -----

If Application.WorksheetFunction.CountA(Selection) = 1 Then GoTo spechandler

Cells(200, 1) = "=1"

    Range("A33:AV48").Select
    Selection.ClearContents
    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .ThemeColor = xlThemeColorAccent3
        .TintAndShade = 0.599993896298105
        .PatternTintAndShade = 0
    End With
'-----Making production data-----

    For nrofprod = 1 To products
Cells(18 + nrofprod, 13) = "Add"
Cells(18 + nrofprod, 13).Select
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorDark1
    .TintAndShade = 0
    .PatternTintAndShade = 0
End With

Cells(18 + nrofprod, 13 + 2) = "Add"
Cells(18 + nrofprod, 13 + 2).Select
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorDark1
    .TintAndShade = 0
    .PatternTintAndShade = 0
End With

    Next
'-----Making Utility Limits-----

```

Module2 - 5

```
For a = 1 To xrow
Cells(17 + a, 19) = "Add"
Cells(17 + a, 19).Select
```

```
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorDark1
    .TintAndShade = 0
    .PatternTintAndShade = 0
End With

Cells(17 + a, 21) = "Add"
Cells(17 + a, 21).Select
```

```
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorDark1
    .TintAndShade = 0
    .PatternTintAndShade = 0
End With
```

Next

'-----Making Ud Ad Au-----'

Dim yrow As Integer

yrow = Cells(10, 6).Value

```
For a1 = 1 To xrow
For a2 = 1 To xrow
For b1 = 1 To yrow
For b2 = 1 To yrow
```

```
If a1 = a2 Then
```

```
Cells(34 + a1, (1 + a2)) = "1"
```

```
Else
```

```
Cells(34 + a1, (1 + a2)) = "0"
```

```
End If
```

```
Cells(34 + a1, (1 + a2)).Select
```

```
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorDark1
    .TintAndShade = 0
    .PatternTintAndShade = 0
End With
```

```
If Cells(18 + a1, 4) = "Add" Then
```

```
Cells(34 + a1, 1) = Cells(18 + a1, 3)
```

```
Cells(34, 1 + a1) = Cells(18 + a1, 3)
```

```
Cells(34, 1 + xrow + 2 + a1 + yrow + 2) = Cells(18 + a1, 3)
```

```
Else
```

```
Cells(34, 1 + a1) = Cells(18 + a1, 4)
```

```
Cells(34 + a1, 1) = Cells(18 + a1, 4)
```

```
Cells(34, 1 + xrow + 2 + a1 + yrow + 2) = Cells(18 + a1, 4)
```

```
End If
```


Module2 - 6

```
If b1 = b2 Then

Cells((34 + b1), (1 + b2 + xrow + 2)) = "1"

Else

Cells((34 + b1), (1 + b2 + xrow + 2)) = "0"
End If

Cells((34 + b1), (1 + b2 + xrow + 2)).Select
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorDark1
    .TintAndShade = 0
    .PatternTintAndShade = 0
End With

If Cells(18 + b1, 6) = "Add" Then

Cells(34, 1 + xrow + 2 + b1) = Cells(18 + b1, 5)
Cells(34 + b1, 1 + xrow + 2) = Cells(18 + b1, 5)
Cells(34 + b1, 1 + xrow + 2 + yrow + 2) = Cells(18 + b1, 5)
Else
Cells(34, 1 + xrow + 2 + b1) = Cells(18 + b1, 6)
Cells(34 + b1, 1 + xrow + 2) = Cells(18 + b1, 6)
Cells(34 + b1, 1 + xrow + 2 + yrow + 2) = Cells(18 + b1, 6)
End If

Cells((34 + b1), (1 + a1 + xrow + yrow + 4)) = "0"

Cells((34 + b1), (1 + a1 + xrow + yrow + 4)).Select

With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorDark1
    .TintAndShade = 0
    .PatternTintAndShade = 0
End With

Next
Next
Next
Next

Cells(33, 3).Select
Selection.Font.Bold = True
ActiveCell.FormulaR1C1 = "Ud"
Cells(33, 3).Select
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorAccent3
    .TintAndShade = 0.399975585192419
    .PatternTintAndShade = 0
End With

Cells(33, 3 + xrow + 2).Select
Selection.Font.Bold = True
ActiveCell.FormulaR1C1 = "Ad"
Cells(33, 3 + xrow + 2).Select
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
```

Module2 - 7

```
.ThemeColor = xlThemeColorAccent3
.TintAndShade = 0.399975585192419
.PatternTintAndShade = 0
End With

Cells(33, 3 + xrow + yrow + 4).Select
Selection.Font.Bold = True
ActiveCell.FormulaR1C1 = "Au"
Cells(33, 3 + xrow + yrow + 4).Select
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorAccent3
    .TintAndShade = 0.399975585192419
    .PatternTintAndShade = 0
End With

spechandler:

For nrofprod = 1 To products

If Cells(18 + nrofprod, 6) = "Add" Then

Cells(18 + nrofprod, 11) = Cells(18 + nrofprod, 5).Value
Else

Cells(18 + nrofprod, 11) = Cells(18 + nrofprod, 6).Value
End If

Next

For a = 1 To xrow
If Cells(18 + a, 4) = "Add" Then

Cells(17 + a, 20) = "<=" & Cells(18 + a, 3).Value & "<="
Else

Cells(17 + a, 20) = "<=" & Cells(18 + a, 4).Value & "<="
End If

If Cells(18 + a, 4) = "Add" Then

Cells(34 + a, 1) = Cells(18 + a, 3)
Cells(34, 1 + a) = Cells(18 + a, 3)
Else
Cells(34 + a, 1) = Cells(18 + a, 4)
Cells(34, 1 + a) = Cells(18 + a, 4)
End If

Next

Application.ScreenUpdating = True

End Sub

Sub Delete_name()
On Error Resume Next

ActiveWorkbook.Names("Ud").Delete
ActiveWorkbook.Names("Ad").Delete
ActiveWorkbook.Names("Au").Delete
ActiveWorkbook.Names("Identity").Delete
ActiveWorkbook.Names("U").Delete
ActiveWorkbook.Names("ONES").Delete
```

Module2 - 8

```
ActiveWorkbook.Names("ONE_NR S").Delete
ActiveWorkbook.Names("DIAG_AVAIL").Delete
ActiveWorkbook.Names("gm p").Delete
ActiveWorkbook.Names("Ts").Delete
ActiveWorkbook.Names("nr_samples").Delete
ActiveWorkbook.Names("Uud").Delete
ActiveWorkbook.Names("L1_1").Delete
ActiveWorkbook.Names("L1_2").Delete
ActiveWorkbook.Names("L2_1").Delete
ActiveWorkbook.Names("L2_2").Delete
ActiveWorkbook.Names("L3_1").Delete
ActiveWorkbook.Names("L3_2").Delete
ActiveWorkbook.Names("L4_1").Delete
ActiveWorkbook.Names("L4_2").Delete
ActiveWorkbook.Names("L5_1").Delete
ActiveWorkbook.Names("L5_2").Delete
ActiveWorkbook.Names("L6_1").Delete
ActiveWorkbook.Names("L6_2").Delete
ActiveWorkbook.Names("L7_1").Delete
ActiveWorkbook.Names("L7_2").Delete
ActiveWorkbook.Names("L8_1").Delete
ActiveWorkbook.Names("L8_2").Delete
ActiveWorkbook.Names("L9_1").Delete
ActiveWorkbook.Names("L9_2").Delete
ActiveWorkbook.Names("L10_1").Delete
ActiveWorkbook.Names("L10_2").Delete
ActiveWorkbook.Names("L11_1").Delete
ActiveWorkbook.Names("L11_2").Delete
ActiveWorkbook.Names("L12_1").Delete
ActiveWorkbook.Names("L12_2").Delete
ActiveWorkbook.Names("L13_1").Delete
ActiveWorkbook.Names("L13_2").Delete
ActiveWorkbook.Names("L14_1").Delete
ActiveWorkbook.Names("L14_2").Delete
```

```
ActiveWorkbook.Names("Data1").Delete
ActiveWorkbook.Names("Data2").Delete
ActiveWorkbook.Names("Data3").Delete
ActiveWorkbook.Names("Data4").Delete
ActiveWorkbook.Names("Data5").Delete
ActiveWorkbook.Names("Data6").Delete
ActiveWorkbook.Names("Data7").Delete
ActiveWorkbook.Names("Data8").Delete
ActiveWorkbook.Names("Data9").Delete
ActiveWorkbook.Names("Data10").Delete
ActiveWorkbook.Names("Data11").Delete
ActiveWorkbook.Names("Data12").Delete
ActiveWorkbook.Names("Data13").Delete
ActiveWorkbook.Names("Data14").Delete
ActiveWorkbook.Names("Adir").Delete
ActiveWorkbook.Names("Atot").Delete
ActiveWorkbook.Names("ONES_Atot").Delete
```

End Sub

Sub Run_Analysis()

```
'1 Make matrix U *****
'*****
Application.ScreenUpdating = False
On Error Resume Next
```

```
Sheets("Data1").Select
Cells.Select
Selection.ClearContents
```

```
Sheets("Data2").Select
```

Module2 - 9

```
Range("ONES").Select
Selection.ClearContents

Range("B52").Select
Selection.ClearContents
Sheets("GUI").Select
Range("C52:L66").Select
Selection.ClearContents

Cells(68, 5) = "Utility"
Cells(68, 11) = "Product"
```

```
Dim xrow As Integer
Dim yrow As Integer
Dim Rng As Range
Dim nr_samples As Integer
```

```
    ' Give each row a range
```

```
xrow = Cells(10, 4).Value
yrow = Cells(10, 6).Value
nr_samples = Cells(10, 10).Value
```

```
For a = 1 To xrow
```

```
    Cells(17 + a, 19).Select
        Set Rng = Selection
```

```
ActiveWorkbook.Names.Add Name:="L" & a & "_1", RefersTo:=Rng
```

```
Cells(17 + a, 21).Select
    Set Rng = Selection
```

```
ActiveWorkbook.Names.Add Name:="L" & a & "_2", RefersTo:=Rng
```

```
Next
```

```
Sheets("Data2").Select
```

```
For a = 1 To xrow
```

```
    Range("Radata").Cells(a, 1).Select
    Range(Selection, Selection.End(xlToRight)).Select
```

```
Set Rng = Selection
```

```
ActiveWorkbook.Names.Add Name:="Data" & a, RefersTo:=Rng
```

```
Next
```

```
Range("Radata").Select
Selection.Copy
```

```
Range("Radata").Cells(xrow + 2, 1).Select
ActiveSheet.Paste
Application.CutCopyMode = False
Selection.FormulaArray = "=1"
```

```
Set Rng = Selection
```

Module2 - 10

```
ActiveWorkbook.Names.Add Name:="ONES", RefersTo:=Rng

Sheets("Data1").Select

For b = 1 To xrow

    Range(Cells(2 + b, 2), Cells(2 + b, nr_samples + 1)).Select

    Selection.FormulaArray = "=IF((Data" & b & ")>=L" & b & "_" & 1 & ",IF((Data" & b & ")<=L" & b & "_" & 2 & ",1,0),0)"
Next

Range(Cells(3, 2), Cells(3 + xrow - 1, 2 + nr_samples - 1)).Select

Set Rng = Selection

ActiveWorkbook.Names.Add Name:="U", RefersTo:=Rng

' Name Range #####
' #####

Sheets("GUI").Select

Set Rng = Sheets("GUI").Range(Cells(35, 2), Cells(35 + xrow - 1, 2 + xrow - 1))

ActiveWorkbook.Names.Add Name:="Ud", RefersTo:=Rng

Set Rng = Sheets("GUI").Range(Cells(35, 2 + xrow + 2), Cells(35 + yrow - 1, 2 + yrow - 1 + xrow + 2))

ActiveWorkbook.Names.Add Name:="Ad", RefersTo:=Rng

Set Rng = Sheets("GUI").Range(Cells(35, 2 + xrow + 2 + yrow + 2), Cells(35 + yrow - 1, 2 + xrow - 1 + xrow + 2 + yrow + 2))

ActiveWorkbook.Names.Add Name:="Au", RefersTo:=Rng

Set Rng = Sheets("GUI").Range(Cells(151, 2), Cells(151 + xrow - 1, 2 + xrow - 1))

ActiveWorkbook.Names.Add Name:="Identity", RefersTo:=Rng

'Calculate Uud #####
' #####

Sheets("Data1").Select

Range("U").Select
Selection.Copy
Cells(4 + xrow, 2).Select
ActiveSheet.Paste

Application.CutCopyMode = False
Selection.FormulaArray = "=SIGN(U+SIGN(MMULT((Identity-Ud), (U-ONES))))"

Set Rng = Selection

ActiveWorkbook.Names.Add Name:="Uud", RefersTo:=Rng

' Utility Results #####
```

Module2 - 11

```
#####  
'#####  
  
'Beräknar Availability!!!  
For D = 1 To nr_samples  
Cells(104 + D, 2) = 1 / nr_samples  
Next  
  
Set Rng = Sheets("Data1").Range(Cells(105, 2), Cells(105 + nr_samples - 1, 2))  
ActiveWorkbook.Names.Add Name:="ONE_NR_S", RefersTo:=Rng  
Sheets("GUI").Select  
Range(Cells(52, 4), Cells(52 + xrow - 1, 4)).Select  
  
Selection.FormulaArray = "=MMULT(Uud,ONE_NR_S)"  
Selection.NumberFormat = "0%"  
  
'Tar in info om utilities  
  
For nrofutility = 1 To xrow  
  
If Cells(18 + nrofutility, 4) = "Add" Then  
Cells(51 + nrofutility, 3) = Cells(18 + nrofutility, 3).Value  
Else  
Cells(51 + nrofutility, 3) = Cells(18 + nrofutility, 4).Value  
End If  
Next  
  
'Beräknar estimation of revenue loss (dir)  
'1) DIAG(I-AVAIL)  
For a1 = 1 To xrow  
For a2 = 1 To xrow  
For a3 = 1 To xrow  
  
If a1 = a2 Then  
Cells(166 + a1, (1 + a2)) = 1 - Cells(51 + a1, 4)  
Else  
Cells(166 + a1, (1 + a2)) = "0"  
End If  
  
Next  
  
Next  
Next  
  
Set Rng = Sheets("GUI").Range(Cells(167, 2), Cells(167 + xrow - 1, 2 + xrow - 1))  
ActiveWorkbook.Names.Add Name:="DIAG_AVAIL", RefersTo:=Rng  
  
'2) qm. * p  
For e = 1 To yrow  
Cells(166 + e, 17) = Cells(18 + e, 13) * Cells(18 + e, 15)
```

```

Module2 - 12

Next

Set Rng = Sheets("GUI").Range(Cells(167, 17), Cells(167 + yrow - 1, 17))
ActiveWorkbook.Names.Add Name:="qm_p", RefersTo:=Rng

    'give nr_sample a range
Cells(167, 19) = nr_samples
Set Rng = Sheets("GUI").Range("H10")
ActiveWorkbook.Names.Add Name:="Ts", RefersTo:=Rng

    'give Ts a range
Set Rng = Sheets("GUI").Range("S167")
ActiveWorkbook.Names.Add Name:="nr_samples", RefersTo:=Rng
Range(Cells(52, 5), Cells(52 + xrow - 1, 5)).Select
Selection.FormulaArray = "=MMULT((MMULT(DIAG_AVAIL,TRANSPOSE(Au))),qm_p)*Ts*nr_samples"
'Beräknar estimation of revenue loss (dir)
Range(Cells(52, 6), Cells(52 + xrow - 1, 6)).Select
Selection.FormulaArray = "=MMULT((MMULT(DIAG_AVAIL,TRANSPOSE(SIGN(MMULT(Ad,Au))))),qm_p)*Ts*nr_samples"
'Calculate Adir #####
%
    '#####
Sheets("Data1").Select
Range(Cells(4 + xrow + 2 + xrow, 2), Cells(4 + xrow + 2 + xrow + yrow - 1, 2 + nr_samples - 1)).Select
    Selection.FormulaArray = "=ONES+SIGN(MMULT(Au,U-ONES))"
Set Rng = Selection
    ActiveWorkbook.Names.Add Name:="Adir", RefersTo:=Rng

'Calculate Atot #####
%
    '#####
    Sheets("Data2").Select

    Dim left As Range
    Dim right As Range
Set left = Range("ONES").Cells(1, 1)
Set right = Range("ONES").Cells(1 + yrow - 1, 1 + nr_samples - 1)
Range(left, right).Select

Set Rng = Selection

    ActiveWorkbook.Names.Add Name:="ONES_Atot", RefersTo:=Rng

Sheets("Data1").Select
Range(Cells(4 + xrow + 2 + xrow + 2 + xrow, 2), Cells(4 + xrow + 2 + xrow + 2 + xrow + yrow - 1, 2 + nr_samples - 1)).Select
    Selection.FormulaArray = "=ONES_Atot+SIGN(MMULT(Ad,Adir-ONES_Atot))"
Set Rng = Selection

    ActiveWorkbook.Names.Add Name:="Atot", RefersTo:=Rng

```

Module2 - 13

```
Sheets("GUI").Select
Range(Cells(52, 9), Cells(52 + yrow - 1, 9)).Select

Selection.FormulaArray = "=MMULT(Adir,ONE_NR_S)"
Selection.NumberFormat = "0%"
Range(Cells(52, 10), Cells(52 + yrow - 1, 10)).Select

Selection.FormulaArray = "=MMULT(Atot,ONE_NR_S)"
Selection.NumberFormat = "0%"
'Tar in info om utilities

For nrofproducts = 1 To yrow

  If Cells(18 + nrofproducts, 6) = "Add" Then
    Cells(51 + nrofproducts, 8) = Cells(18 + nrofproducts, 5).Value
  Else
    Cells(51 + nrofproducts, 8) = Cells(18 + nrofproducts, 6).Value
  End If

Next

For a = 1 To yrow
  T = Cells(10, 8).Value

  Cells(51 + a, 11) = (1 - Cells(51 + a, 9)) * Cells(166 + a, 17) * nr_samples * T
Next

For a = 1 To yrow
  T = Cells(10, 8).Value

  Cells(51 + a, 12) = (1 - Cells(51 + a, 10)) * Cells(166 + a, 17) * nr_samples * T
Next

Range(Cells(52, 3), Cells(52 + xrow, 12)).Select
Selection.Copy
Range("O51").Select
  Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
    :=False, Transpose:=False
Range(Cells(52, 3), Cells(52 + xrow, 12)).Select
Selection.ClearContents

Range(Cells(51, 15), Cells(52 + xrow, 24)).Select
Selection.Copy
Range("C52").Select
  ActiveSheet.Paste
Range(Cells(51, 15), Cells(52 + xrow, 24)).Select
Selection.ClearContents

Range(Cells(52, 4), Cells(52 + xrow, 4)).Select
Selection.NumberFormat = "0%"

Range(Cells(52, 9), Cells(52 + xrow, 10)).Select
Selection.NumberFormat = "0%"
Range(Cells(52, 5), Cells(52 + xrow, 6)).Select
Selection.NumberFormat = "0.00"
```


Module2 - 14

```
Range(Cells(52, 11), Cells(52 + yrow, 12)).Select
Selection.NumberFormat = "0.00"
Range("A100").Select
Application.ScreenUpdating = True
```

End Sub

```
Sub Sort_Utility()
Application.ScreenUpdating = False
Dim xrow As Integer
```

```
xrow = Cells(10, 4).Value
```

```
Range("C51:F" & 51 + xrow).Select
```

```
If Cells(68, 5) = "Utility" Then
```

```
If Cells(19, 4) = "Add" Then
```

```
ActiveWorkbook.Worksheets("GUI").Sort.SortFields.clear
ActiveWorkbook.Worksheets("GUI").Sort.SortFields.Add Key:=Range("C52:C" & 51 + xrow), _
SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
With ActiveWorkbook.Worksheets("GUI").Sort
.SetRange Range("C51:F" & 51 + xrow)
.Header = xlYes
.MatchCase = False
.Orientation = xlTopToBottom
.SortMethod = xlPinYin
.Apply
End With
Else
ActiveWorkbook.Worksheets("GUI").Sort.SortFields.clear
ActiveWorkbook.Worksheets("GUI").Sort.SortFields.Add Key:=Range("C52:C" & 51 + xrow), _
SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
With ActiveWorkbook.Worksheets("GUI").Sort
.SetRange Range("C51:F" & 51 + xrow)
.Header = xlYes
.MatchCase = False
.Orientation = xlTopToBottom
.SortMethod = xlPinYin
.Apply
End With
End If
```

```
ElseIf Cells(68, 5) = "Av." Then
```

```
ActiveWorkbook.Worksheets("GUI").Sort.SortFields.clear
ActiveWorkbook.Worksheets("GUI").Sort.SortFields.Add Key:=Range("D52:D" & 51 + xrow), _
SortOn:=xlSortOnValues, Order:=xlDescending, DataOption:=xlSortNormal
With ActiveWorkbook.Worksheets("GUI").Sort
.SetRange Range("C51:F" & 51 + xrow)
.Header = xlYes
.MatchCase = False
.Orientation = xlTopToBottom
.SortMethod = xlPinYin
.Apply
End With
```

```
ElseIf Cells(68, 5) = "Dir. loss" Then
```

```
ActiveWorkbook.Worksheets("GUI").Sort.SortFields.clear
ActiveWorkbook.Worksheets("GUI").Sort.SortFields.Add Key:=Range("E52:E" & 51 + xrow), _
SortOn:=xlSortOnValues, Order:=xlDescending, DataOption:=xlSortNormal
With ActiveWorkbook.Worksheets("GUI").Sort
.SetRange Range("C51:F" & 51 + xrow)
.Header = xlYes
.MatchCase = False
.Orientation = xlTopToBottom
```

Module2 - 15

```
        .SortMethod = xlPinYin
        .Apply
    End With

    ElseIf Cells(68, 5) = "Tot. loss" Then

        ActiveWorkbook.Worksheets("GUI").Sort.SortFields.clear
        ActiveWorkbook.Worksheets("GUI").Sort.SortFields.Add Key:=Range("F52:F" & 51 + xrow), _
            SortOn:=xlSortOnValues, Order:=xlDescending, DataOption:=xlSortNormal
        With ActiveWorkbook.Worksheets("GUI").Sort
            .SetRange Range("C51:F" & 51 + xrow)
            .Header = xlYes
            .MatchCase = False
            .Orientation = xlTopToBottom
            .SortMethod = xlPinYin
            .Apply
        End With

        ElseIf Cells(68, 5) = "Add" Then

            MsgBox "Add text to order by"

        Else

            MsgBox "Invalid input"

        End If

        Range("A100").Select
        Application.ScreenUpdating = True
    End Sub

Sub Sort_products()
Application.ScreenUpdating = False

Dim yrow As Integer
yrow = Cells(10, 6).Value

Range("H51:L" & 51 + yrow).Select

If Cells(68, 11) = "Product" Then

If Cells(19, 4) = "Add" Then

    ActiveWorkbook.Worksheets("GUI").Sort.SortFields.clear
    ActiveWorkbook.Worksheets("GUI").Sort.SortFields.Add Key:=Range("H52:H" & 51 + yrow), _
        SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
    With ActiveWorkbook.Worksheets("GUI").Sort
        .SetRange Range("H51:L" & 51 + yrow)
        .Header = xlYes
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With

    Else

        ActiveWorkbook.Worksheets("GUI").Sort.SortFields.clear
        ActiveWorkbook.Worksheets("GUI").Sort.SortFields.Add Key:=Range("H52:H" & 51 + yrow), _
            SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
        With ActiveWorkbook.Worksheets("GUI").Sort
            .SetRange Range("H51:L" & 51 + yrow)
            .Header = xlYes
            .MatchCase = False
            .Orientation = xlTopToBottom
            .SortMethod = xlPinYin
            .Apply
        End With
    End If
```

Module2 - 16

```
ElseIf Cells(68, 11) = "Dir. av." Then
```

```
ActiveWorkbook.Worksheets("GUI").Sort.SortFields.clear  
ActiveWorkbook.Worksheets("GUI").Sort.SortFields.Add Key:=Range("I52:I" & 51 + yrow), _  
SortOn:=xlSortOnValues, Order:=xlDescending, DataOption:=xlSortNormal  
With ActiveWorkbook.Worksheets("GUI").Sort  
  .SetRange Range("H51:L" & 51 + yrow)  
  .Header = xlYes  
  .MatchCase = False  
  .Orientation = xlTopToBottom  
  .SortMethod = xlPinYin  
  .Apply  
End With
```

```
ElseIf Cells(68, 11) = "Tot. av." Then
```

```
ActiveWorkbook.Worksheets("GUI").Sort.SortFields.clear  
ActiveWorkbook.Worksheets("GUI").Sort.SortFields.Add Key:=Range("J52:J" & 51 + yrow), _  
SortOn:=xlSortOnValues, Order:=xlDescending, DataOption:=xlSortNormal  
With ActiveWorkbook.Worksheets("GUI").Sort  
  .SetRange Range("H51:L" & 51 + yrow)  
  .Header = xlYes  
  .MatchCase = False  
  .Orientation = xlTopToBottom  
  .SortMethod = xlPinYin  
  .Apply  
End With
```

```
ElseIf Cells(68, 11) = "Dir. loss" Then
```

```
ActiveWorkbook.Worksheets("GUI").Sort.SortFields.clear  
ActiveWorkbook.Worksheets("GUI").Sort.SortFields.Add Key:=Range("K52:K" & 51 + yrow), _  
SortOn:=xlSortOnValues, Order:=xlDescending, DataOption:=xlSortNormal  
With ActiveWorkbook.Worksheets("GUI").Sort  
  .SetRange Range("H51:L" & 51 + yrow)  
  .Header = xlYes  
  .MatchCase = False  
  .Orientation = xlTopToBottom  
  .SortMethod = xlPinYin  
  .Apply  
End With
```

```
ElseIf Cells(68, 11) = "Tot. loss" Then
```

```
ActiveWorkbook.Worksheets("GUI").Sort.SortFields.clear  
ActiveWorkbook.Worksheets("GUI").Sort.SortFields.Add Key:=Range("L52:L" & 51 + yrow), _  
SortOn:=xlSortOnValues, Order:=xlDescending, DataOption:=xlSortNormal  
With ActiveWorkbook.Worksheets("GUI").Sort  
  .SetRange Range("H51:L" & 51 + yrow)  
  .Header = xlYes  
  .MatchCase = False  
  .Orientation = xlTopToBottom  
  .SortMethod = xlPinYin  
  .Apply  
End With
```

```
ElseIf Cells(68, 11) = "Add" Then
```

```
  MsgBox "Add text to order by"
```

```
Else
```

```
  MsgBox "Invalid input"
```

```
End If
```

```
Range("A100").Select  
Application.ScreenUpdating = True
```

Module2 - 17

End Sub

```
Sub clear_old_data()
    Application.ScreenUpdating = False

    Sheets("Data2").Select
        Cells.Select
        Selection.ClearContents
        Range("A100").Select

    Sheets("GUI").Select
        Range("J10").Select
        Selection.ClearContents
        Range("D10").Select
        Selection.ClearContents
        Range("A100").Select

    Application.ScreenUpdating = True
End Sub
```