**Class: OM 424-01-22413**

**aSSIGNMENT nO. 1**

Riyasha Baidya

**Overview of the assignment:**

This assignment aims to create an automated system in which a mini dashboard is generated for the production manager that can help the manager schedule monthly production levels over a planning horizon of m months. Taking the following into account; unit production cost, unit holding inventory costs, demand, and production capacity for each month. The manager needs to generate a schedule that minimizes the total production and inventory holding costs over m months. The assignment involves creating line charts for the data, mini dashboards with 3 buttons, and a main sheet giving instructions on how to interact with the dashboard in Excel using VBA.

**How the assignment was implemented:**

1. **Understanding and preparation:**

First, I reviewed the assignment instruction that was provided and identified the main tasks of the project such as creating a mini dashboard with buttons for line charts and creating a main sheet with instructions. Then, I opened the given data set Excel file and saved it as a macro-enabled workbook (.xlsm)

1. **Module 1:**

With the help of in-class discussion and working on it together modules 1 and 2 were done with the help of the assignment 1 gameplan. Here, we created our first subroutine to create inventory chat. We also declared variables (DIM) and set objects (SET) as follows.

1. ' -- DIM and SET
3. ' -- Calculate Inventories
4. Call CalculateInventory
5. ' -- Format the months
6. ' -- Create the linechart and Move the linechart
7. Call CreateInventoryLineChart

Next, we needed to create a subroutine to calculate the inventory. To do this we had to input the formula that was Production unit minus demand plus previous inventory if any. After, we defined the last row, which this important because this makes sure that the Excel file can handle any number of observations added afterwards.

1. ' -- defining the last\_row
2. last\_row = ws.Range("A" & ws.Rows.Count).End(xlUp).Row

In this section, we also activated the worksheet so that the data worksheet is performing actions on the worksheet we want.

1. **Module 2:**

In module 2, we proceeded to create the line chart for the inventory column. This was done by first recording the macro, selecting the column of dates and inventory, and creating a line chart with markers. Then name the new sheet where it is created as “Inventory.” The next subroutine makes this more structured by adding the last row to make sure it gets the correct last present row then we delete any existing chart that is named inventory and apply it to create a line chart for the correct data.

Then I created a button to go back to the main page on the line chart sheet. To do this I declared the shape and set button shape with the text “Back to main.”

1. **Module 3:**

Module 3 is about creating a production holding line chart. I did the same process as module 2 and recorded the macro of the column’s date, production cost, and holding cost and created a line chart. I added this chart to a new sheet called Production Holding. Like before the next subroutine helps to structure and make sure the line chart is working with the right set of data, so I copied the code from the previous module of activating the worksheet, defining the last row, and deleting the production holding sheet if it exists. I needed to add the button to this page as well, so I used the same code as before to create a shape for the back main page button as follows.

1. ' Add shape **as** a button
2. Dim buttonShape As Shape
3. Set buttonShape = ActiveSheet.Shapes.AddShape(msoShapeRectangle, 10, 10, 80, 30)
4. buttonShape.TextFrame.Characters.Text = "Back to Main"
5. buttonShape.OnAction = "GoToMainSheet"
6. **Module 4:**

Module 4 was creating a production demand line chart. I recorded the macro for the column date, production, and demand and created a line chart displaying this in a new sheet named “production demand”. This is the first subroutine that was created for module 4. The second step of the subroutine was repeating the same code as before to structure the data and add the button to this page. For structuring I dim and set the worksheet data and the last, defining the last row, delete a production demand line chart if it exists, and activating the datasheet, and lastly, add a shape button to the production demand line chart for the user to go back to the main sheet as before.

1. **Module 5:**

This module was created to activate the main sheet so that the user can view the active main sheet in the workbook when they press the buttons. This subroutine is called “Go to Main sheet” and the second line of code shows which specific sheet to go to when clicking on the button, which is the “main” sheet.

1. ' Define the macro to navigate to the main sheet
2. **Sub** GoToMainSheet()
3. Sheets("Main").Activate
4. **End** **Sub**
5. **Adding buttons to the main page**:

To add the buttons, I went to the developer tab added buttons from the controls, and assigned the macro to the specific chart with the buttons.

**A short tutorial on how to use the .xlsm file.**

This spreadsheet allows the managers in the production planning to see the following visualization of the raw data as follows:

* + *A line chart of the historical inventory levels*
  + *A line chart of the historical production cost along with the holding cost*
  + *A line chart of the historical production units along with the demand*

Please click the buttons below to see the charts corresponding charts:

**VBA: Appendix**

**Module 1**

|  |
| --- |
| 1. Sub InventoryChart() 2. ' The subroutine **to** create **the** inventory line chart 3. ' -- DIM and SET 5. ' -- Calculate Inventories 6. Call CalculateInventory 7. ' -- Format the months 8. ' -- Create the linechart and Move the linechart 9. Call CreateInventoryLineChart 10. End Sub 12. Sub CalculateInventoryMacro() 13. ' 14. ' CalculateInventory Macro 15. ' 17. ' 18. Range("G1").Select 19. ActiveCell.FormulaR1C1 = "Inventory" 20. Range("G2").Select 21. Application.CutCopyMode = False 22. ActiveCell.FormulaR1C1 = "=RC[-1]-RC[-5]" 23. Range("G3").Select 24. Application.CutCopyMode = False 25. ActiveCell.FormulaR1C1 = "=RC[-1]-RC[-5]+R[-1]C" 26. Range("G3").Select 27. Selection.AutoFill Destination:=Range("G3:G51"), Type:=xlFillDefault 28. Range("G3:G51").Select 29. End Sub 30. Sub CalculateInventory() 31. ' 32. ' A subroutine **to** calculate **the** inventory values 33. ' 34. ' -- Dim and Set 35. Dim ws As Worksheet 36. Dim last\_row As Integer 38. Set ws = Sheets("Data") 40. ' -- defining the last\_row 41. last\_row = ws.Range("A" & ws.Rows.Count).End(xlUp).Row 42. ' 43. ws.Activate 45. Range("G1").Select 46. ActiveCell.FormulaR1C1 = "Inventory" 47. Range("G2").Select 48. Application.CutCopyMode = False 49. ActiveCell.FormulaR1C1 = "=RC[-1]-RC[-5]" 50. Range("G3").Select 51. Application.CutCopyMode = False 52. ActiveCell.FormulaR1C1 = "=RC[-1]-RC[-5]+R[-1]C" 53. Range("G3").Select 54. Selection.AutoFill Destination:=Range("G3:G" & last\_row), Type:=xlFillDefault 55. Range("G3:G" & last\_row).Select 56. End Sub |

**Module 2**

1. ' CreateInventoryLineChartMacro Macro
2. '
4. '
5. Range("A1:A51").**Select**
6. ActiveWindow.SmallScroll Down:=-16
7. Range("A1:A51,G1:G51").**Select**
8. Range("G1").Activate
9. ActiveSheet.Shapes.AddChart2(332, xlLineMarkers).**Select**
10. ActiveChart.SetSourceData Source:=Range("Data!$A$1:$A$51,Data!$G$1:$G$51")
11. ActiveChart.Location Where:=xlLocationAsNewSheet, Name:="Inventory"
12. **End** **Sub**
13. **Sub** CreateInventoryLineChart()
14. '
15. ' CreateInventoryLineChartMacro
16. '
18. '
19. ' -- Dim and Set
20. **Dim** ws\_data **As** Worksheet
21. **Dim** last\_row **As** **Integer**
23. **Set** ws\_data = Sheets("Data")
25. ' -- defining the last\_row
26. last\_row = ws\_data.Range("A" & ws\_data.Rows.Count).**End**(xlUp).Row
28. ' -- delete the inventory chart sheet if it exists
29. **For** **Each** ws **In** Sheets
30. **If** ws.Name = "Inventory" **Then**
31. Application.DisplayAlerts = **False**
32. Sheets("Inventory").Delete
33. Application.DisplayAlerts = **True**
34. **End** **If**
35. **Next**
37. ws\_data.Activate
39. 'Range("A1:A" & last\_row).Select
41. Range("A1:A" & last\_row & ",G1:G" & last\_row).**Select**
42. Range("G1").Activate
43. ActiveSheet.Shapes.AddChart2(332, xlLineMarkers).**Select**
44. ActiveChart.SetSourceData Source:=Range("Data!$A$1:$A$" & last\_row & ",Data!$G$1:$G$" & last\_row)
45. ActiveChart.Location Where:=xlLocationAsNewSheet, Name:="Inventory"
47. ' Add shape as a button
48. **Dim** buttonShape **As** Shape
49. **Set** buttonShape = ActiveSheet.Shapes.AddShape(msoShapeRectangle, 10, 10, 80, 30)
50. buttonShape.TextFrame.Characters.Text = "Back to Main"
51. buttonShape.OnAction = "GoToMainSheet"
53. **End** **Sub**

**Module 3**

1. **Sub** CreateProductionHoldingLineChartMacro()
2. '
3. '
5. '
6. Range("A1:A51,C1:C51,D1:D51").**Select**
7. Range("D1").Activate
8. ActiveSheet.Shapes.AddChart2(332, xlLineMarkers).**Select**
9. ActiveChart.SetSourceData Source:=Range("Data!$A$1:$A$51,Data!$C$1:$C$51,Data!$D$1:$D$51")
10. ActiveChart.Location Where:=xlLocationAsNewSheet, Name:="Production Holding"
11. ActiveChart.ChartTitle.Text = "Production Holding"
12. **End** **Sub**
13. **Sub** CreateProductionHoldingLineChart()
14. '
15. '
17. '
18. ' -- Dim and Set
19. **Dim** ws\_data **As** Worksheet
20. **Dim** last\_row **As** **Integer**
22. **Set** ws\_data = Sheets("Data")
24. ' -- defining the last\_row
25. last\_row = ws\_data.Range("A" & ws\_data.Rows.Count).**End**(xlUp).Row
27. ' -- delete the inventory chart sheet if it exists
28. **For** **Each** ws **In** Sheets
29. **If** ws.Name = "Production Holding" **Then**
30. Application.DisplayAlerts = **False**
31. Sheets("Production Holding").Delete
32. Application.DisplayAlerts = **True**
33. **End** **If**
34. **Next**
36. ws\_data.Activate
38. 'Range("A1:A" & last\_row).Select
40. Range("A1:A" & last\_row & ",C1:C" & last\_row & ",D1:D" & last\_row).**Select**
41. Range("D1").Activate
42. ActiveSheet.Shapes.AddChart2(332, xlLineMarkers).**Select**
43. ActiveChart.SetSourceData Source:=Range("Data!$A$1:$A$" & last\_row & ",Data!$C$1:$C$" & last\_row & ",Data!$D$1:$D$" & last\_row)
44. ActiveChart.Location Where:=xlLocationAsNewSheet, Name:="Production Holding"
45. ActiveChart.ChartTitle.Text = "Production Holding" '
47. ' Add shape as a button
48. **Dim** buttonShape **As** Shape
49. **Set** buttonShape = ActiveSheet.Shapes.AddShape(msoShapeRectangle, 10, 10, 80, 30)
50. buttonShape.TextFrame.Characters.Text = "Back to Main"
51. buttonShape.OnAction = "GoToMainSheet"
53. **End** **Sub**

**Module 4**

|  |
| --- |
| 1. **Sub** CreateProductionDemandLineChartMacro() 2. ' 3. ' 5. ' 6. Range("A1:A51,B1:B51,F1:F51").**Select** 7. Range("D1").Activate 8. ActiveSheet.Shapes.AddChart2(332, xlLineMarkers).**Select** 9. ActiveChart.SetSourceData Source:=Range("Data!$A$1:$A$51,Data!$B$1:$B$51,Data!$F$1:$F$51") 10. ActiveChart.Location Where:=xlLocationAsNewSheet, Name:="Production Demand" 11. ActiveChart.ChartTitle.Text = "Production Demand" 12. **End** **Sub** 13. **Sub** CreateProductionDemandLineChart() 14. ' 15. ' 17. ' 18. ' -- Dim and Set 19. **Dim** ws\_data **As** Worksheet 20. **Dim** last\_row **As** **Integer** 22. **Set** ws\_data = Sheets("Data") 24. ' -- defining the last\_row 25. last\_row = ws\_data.Range("A" & ws\_data.Rows.Count).**End**(xlUp).Row 27. ' -- delete the inventory chart sheet if it exists 28. **For** **Each** ws **In** Sheets 29. **If** ws.Name = "Production Demand" **Then** 30. Application.DisplayAlerts = **False** 31. Sheets("Production Demand").Delete 32. Application.DisplayAlerts = **True** 33. **End** **If** 34. **Next** 36. ws\_data.Activate 38. 'Range("A1:A" & last\_row).Select 40. Range("A1:A" & last\_row & ",B1:B" & last\_row & ",F1:F" & last\_row).**Select** 41. Range("F1").Activate 42. ActiveSheet.Shapes.AddChart2(332, xlLineMarkers).**Select** 43. ActiveChart.SetSourceData Source:=Range("Data!$A$1:$A$" & last\_row & ",Data!$B$1:$B$" & last\_row & ",Data!$F$1:$F$" & last\_row) 44. ActiveChart.Location Where:=xlLocationAsNewSheet, Name:="Production Demand" 45. ActiveChart.ChartTitle.Text = "Production Demand" 47. ' Add shape as a button 48. **Dim** buttonShape **As** Shape 49. **Set** buttonShape = ActiveSheet.Shapes.AddShape(msoShapeRectangle, 10, 10, 80, 30) 50. buttonShape.TextFrame.Characters.Text = "Back to Main" 51. buttonShape.OnAction = "GoToMainSheet" 53. **End** **Sub** |

Module 5

1. ' Define the macro to navigate to the main sheet
2. **Sub** GoToMainSheet()
3. Sheets("Main").Activate
4. **End** **Sub**

**Citation**

When searching how to copy VBA code with highlights (syntax highlighter)

Reference:

Caluori, N. (n.d.). *Syntax Highlighter for Word - K26*. https://syntax-highlighter.k26.ch/#

When prompted with “How to add back to main page button in Vba” (OpenAI, 2024)

Reference:

Chatgpt. (n.d.). https://chat.openai.com