MACRO DOCUMENTATION

Annual_internal_report – All macros

Macro 1: Loading_data

Owner: Krishna Rao Ramesh

Date Created: 25/11/2024

Last Updated: 25/11/2024

Folder Location: *As per location after assignment submission*.

Description: For loading data into macro template annual_internal_report

Instructions:

1. Open file annual_internal_report template.
2. Click on developer tab and select macros. Run macro 'Loading_data' or use the shortcut Ctrl+shift+Q.
3. The macro will copy files from all 8 csv files into our template "annual_internal_report". Do not rename the existing tabs in template as the macro would not function if it were unable to reference the respective sheets.
4. An array containing all the file addresses was assigned to a variable and a 'for' loop was used to transfer data from each csv to its respective destination worksheet.
5. Once when the macro has run successfully, it will have the data from all the 8 csv files and a pop-up will appear with the message "Task 1 completed successfully".

Code:

```
Sub Loading_data()
'
' for loading data
'
' Keyboard Shortcut: Ctrl+Shift+Q

' Declaring all variables that are about to be used in our loading sub

    Dim Datasets As Workbook
    Dim AI_Report As Workbook
    Dim sourceSheet As Worksheet
    Dim destinationSheet As Worksheet
```

```
Dim sourceRange As Range

Dim Path_source As String

Dim DataSource As Variant

Dim i As Integer


' To stop screen from refreshing while macro runs; Helps in enhancing macro's efficiency

Application.ScreenUpdating = False


' Assigning file paths of all CSV files to an array; This array is in turn,allocated to a variable -> DataSource.

DataSource = Array("C:\Users\krish\OneDrive\Documents\TUS MSc BA\Statistics and
                    tools for BA\VBA group assignment\offices.csv", _

          "C:\Users\krish\OneDrive\Documents\TUS MSc BA\Statistics and tools for
            BA\VBA group assignment\employees.csv", _

          "C:\Users\krish\OneDrive\Documents\TUS MSc BA\Statistics and tools for
            BA\VBA group assignment\orders.csv", _

          "C:\Users\krish\OneDrive\Documents\TUS MSc BA\Statistics and tools for
            BA\VBA group assignment\orderdetails.csv", _

          "C:\Users\krish\OneDrive\Documents\TUS MSc BA\Statistics and tools for
            BA\VBA group assignment\payments.csv", _

          "C:\Users\krish\OneDrive\Documents\TUS MSc BA\Statistics and tools for
            BA\VBA group assignment\products.csv", _

          "C:\Users\krish\OneDrive\Documents\TUS MSc BA\Statistics and tools for
            BA\VBA group assignment\productlines.csv", _

          "C:\Users\krish\OneDrive\Documents\TUS MSc BA\Statistics and tools for
            BA\VBA group assignment\customers.csv")


' Assigning another variable to our internal report. This is done to perform a looping operation for transferring data. Annual_Internal_report should be open for macro to run.

Set AI_Report = Workbooks("Annual_internal_report.xlsm")
```

```
    ' Initiating the loop by categorizing the starting and last index of variable -> DataSource. The i in DataSource(i) contains the index number for each loop.
    For i = LBound(DataSource) To UBound(DataSource)
        Path_source = DataSource(i)

        ' The file address equivalent to the ongoing loop is opened. This is repeated for each index number.
        Set Datasets = Workbooks.Open(Path_source)

        ' This block of code only considers the first sheet for each of the 8 datasets.
        Set sourceSheet = Datasets.Sheets(1)

        ' UsedRange is used to select all cells that contain data in a sheet.
        Set sourceRange = sourceSheet.UsedRange

        ' The select statement is used to run through similar operations with the use of indexes. We designate each source data to its respective destination sheet.
        Select Case i
            Case 0
                Set destinationSheet = AI_Report.Sheets("offices")
            Case 1
                Set destinationSheet = AI_Report.Sheets("employees")
            Case 2
                Set destinationSheet = AI_Report.Sheets("orders")
            Case 3
                Set destinationSheet = AI_Report.Sheets("Order_details")
            Case 4
                Set destinationSheet = AI_Report.Sheets("payments")
            Case 5
                Set destinationSheet = AI_Report.Sheets("products")
            Case 6
                Set destinationSheet = AI_Report.Sheets("productlines")
```

```
        Case 7
            Set destinationSheet = AI_Report.Sheets("customers")
    End Select

    ' Copying all the data from each dataset
    sourceRange.Copy

    ' Transferring data to our report; Active cell is moved to A1 for pasting data as values and numbers.
    destinationSheet.Range("A1").PasteSpecial Paste:=xlPasteValuesAndNumberFormats

    ' Clearing clipboard before moving onto next loop
    Application.CutCopyMode = False

    ' Closing workbook associated with each loop
    Datasets.Close False

  Next i

  ' A pop-up to indicate end of macro
  MsgBox "Task 1 completed succesfully!", vbInformation, "Operation Status"

End Sub
```

Macro 2: OnTimeDelivery

Owner: Krishna Rao Ramesh

Date Created: 26/11/2024

Last Updated: 26/11/2024

Folder Location: *As per location after assignment submission.*

Description: To understand which orders are getting delayed.

Instructions:

1. Click on developer tab and select macros. Run macro 'OnTimeDelivery' or use the shortcut Ctrl+shift+W.
2. The macro will formulate the delivery status of all orders based on office locations.
3. A new column "Delivery date" is introduced, which is 6 + shipped date. This is based on the assumption that SLA requirement between customer and the company is applicable only from day of shipping.
4. Based on the comparison between Delivery date and required date, we classify the orders as "Timely Delivery" and "Late Delivery".
5. A table is created to summarize all the above findings and probability of receiving a product on time. Please change input in Y11 to view results for different years.

Code:

```
Sub OnTimeDelivery()
'
' OnTimeDelivery Macro
' To understand which orders are getting delayed
'
' Keyboard Shortcut: Ctrl+Shift+W
'

    Application.ScreenUpdating = False

' Adding extra columns to calculate estimated time of delivery
    Sheets("Orders").Select
    Columns("E:E").Select
    Selection.Insert Shift:=xlToRight
    Range("E1").Select
    ActiveCell.FormulaR1C1 = "Delivery_date"
    Range("E2").Select
    ActiveCell.FormulaR1C1 = "=RC[-1]+6"
    Selection.AutoFill Destination:=Range("E2:E327")
```

```
'Adding more columns to link delivery status with the office location
    Range("G:G").Select
    Selection.Insert Shift:=xlToRight
    Range("G1").Select
    ActiveCell.FormulaR1C1 = "Year"
    Range("G2").Select
    ActiveCell.FormulaR1C1 = "=YEAR(RC[-2])"
    Selection.AutoFill Destination:=Range("G2:G327")
    Range("J1").Select
    ActiveCell.FormulaR1C1 = "Cust_Location"
    Range("J2").Select
    ActiveCell.FormulaR1C1 = "=VLOOKUP(RC[-1],cu,11,0)"
    Selection.AutoFill Destination:=Range("J2:J327")
    Range("K1").Select
    ActiveCell.FormulaR1C1 = "Sales_rep"
    Range("K2").Select
    ActiveCell.FormulaR1C1 = "=VLOOKUP(RC[-2],cu,12,0)"
    Selection.AutoFill Destination:=Range("K2:K327")
    Range("L1").Select
    ActiveCell.FormulaR1C1 = "office_code"
    Range("L2").Select
    ActiveCell.FormulaR1C1 = "=VLOOKUP(RC[-1],emp,6,0)"
    Selection.AutoFill Destination:=Range("L2:L327")
    Range("M1").Select
    ActiveCell.FormulaR1C1 = "office_loc"
    Range("M2").Select
    ActiveCell.FormulaR1C1 = "=VLOOKUP(RC[-1],of,2,0)"
    Selection.AutoFill Destination:=Range("M2:M327")
```

```
'SLA compliance is found by comparing the delivery date with the required date of customer. The delivery status is derived from the SLA compliance.

    Range("N1").Select

    ActiveCell.FormulaR1C1 = "SLA_compliance"

    Range("N2").Select

    ActiveCell.FormulaR1C1 = "=IF(RC[-9]>RC[-11],""Fail"",""Pass"")"

    Selection.AutoFill Destination:=Range("N2:N327")

    Range("O1").Select

    ActiveCell.FormulaR1C1 = "Delivery_status"

    Range("O2").Select

    ActiveCell.FormulaR1C1 = _

      "=IFERROR(IF(RC[-1]=""Pass"",""Timely Delivery"",""Late Delivery""),""Pending"")"
Selection.AutoFill Destination:=Range("O2:O327")


'A table is created to find the probability of orders reaching the customer on time. This table is a combaination of all the above findings grouped by office location.

    Range("V1").Select

    ActiveCell.FormulaR1C1 = "Office_loc"

    Range("V2").Select

    ActiveCell.Formula2R1C1 = "=UNIQUE(RC[-9]:R[325]C[-9])"

    Columns("V:V").ColumnWidth = 11.67

    Range("W1").Select

    ActiveCell.FormulaR1C1 = "Total no: of orders"

    Range("W2").Select

    ActiveCell.FormulaR1C1 = "=COUNTIF(C[-10],RC[-1])"

    Selection.AutoFill Destination:=Range("W2:W8")

    Range("W1").Select

    Columns("W:W").EntireColumn.AutoFit

    Range("X1").Select

    ActiveCell.FormulaR1C1 = "Total no: of shipped orders"

    Range("X2").Select

    ActiveCell.FormulaR1C1 = _
```

```
    "=COUNTIFS(C[-11],RC[-2],C[-18],""Shipped"",C[-17],R11C25)"
  Selection.AutoFill Destination:=Range("X2:X8")
  Range("Y1").Select
  Columns("X:X").EntireColumn.AutoFit
  ActiveCell.FormulaR1C1 = "Number of orders on time"
  Range("Y2").Select
  Columns("Y:Y").EntireColumn.AutoFit
  ActiveCell.FormulaR1C1 = _
    "=COUNTIFS(C[-10],""Timely Delivery"", C[-19],""Shipped"",C[-12],RC[-3],C[-
18],R11C25)"
  Selection.AutoFill Destination:=Range("Y2:Y8")
  Range("Z1").Select
  ActiveCell.FormulaR1C1 = "Probability of receiving a product on time "
  Range("Z2").Select
  Columns("Z:Z").EntireColumn.AutoFit
  Application.CutCopyMode = False
  ActiveCell.FormulaR1C1 = "=RC[-1]/RC[-2]"
  Selection.AutoFill Destination:=Range("Z2:Z8")
  Range("Z2:Z8").Select
  Selection.Style = "Percent"
```

'A cell with a drop-down is assigned to Y11 with input as year. This is used to dynamically modify our charts and table.

```
  Range("X11").Select
  ActiveCell.FormulaR1C1 = "YEAR"
  Selection.Font.Bold = True
  Range("Y11").Select
  With Selection.Validation
    .Delete
    .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
    xlBetween, Formula1:="2003,2004,2005"
    .IgnoreBlank = True
```

```
    .InCellDropdown = True
    .InputTitle = ""
    .ErrorTitle = ""
    .InputMessage = ""
    .ErrorMessage = ""
    .ShowInput = True
    .ShowError = True
  End With
  With ActiveSheet.Range("Y11")
    .Borders(xlEdgeTop).LineStyle = xlContinuous
    .Borders(xlEdgeBottom).LineStyle = xlContinuous
    .Borders(xlEdgeLeft).LineStyle = xlContinuous
    .Borders(xlEdgeRight).LineStyle = xlContinuous
    .Borders(xlEdgeTop).ColorIndex = 0 ' Black color
    .Borders(xlEdgeBottom).ColorIndex = 0 ' Black color
    .Borders(xlEdgeLeft).ColorIndex = 0 ' Black color
    .Borders(xlEdgeRight).ColorIndex = 0 ' Black color
    .Borders(xlEdgeTop).Weight = xlThin
    .Borders(xlEdgeBottom).Weight = xlThin
    .Borders(xlEdgeLeft).Weight = xlThin
    .Borders(xlEdgeRight).Weight = xlThin
  End With

' Adding various types of charts to a) Visualize efficient and inefficient offices, b)Compare shipped orders and on-time orders, and c)order conversion.
  Range("V1:V8,Z1:Z8").Select
  ActiveSheet.Shapes.AddChart2(201, xlColumnClustered).Select
  ActiveChart.SetSourceData Source:=Range("Orders!$V$1:$V$8,Orders!$Z$1:$Z$8")
  ActiveSheet.Shapes("Chart 1").IncrementLeft 441
  ActiveSheet.Shapes("Chart 1").IncrementTop 87.8
  ActiveChart.ChartTitle.Select
```

```
ActiveChart.ChartTitle.Text = "Inefficient Offices"
Selection.Format.TextFrame2.TextRange.Characters.Text = "Inefficient Offices"


ActiveChart.ChartArea.Select
ActiveSheet.Shapes("Chart 1").ScaleWidth 0.9116666667, msoFalse, _
  msoScaleFromTopLeft
ActiveSheet.Shapes("Chart 1").ScaleHeight 0.7986111111, msoFalse, _
  msoScaleFromTopLeft
Range("V1:V8,X1:X8,Y1:Y8").Select
ActiveSheet.Shapes.AddChart2(201, xlColumnClustered).Select
ActiveChart.SetSourceData Source:=Range( _
  "Orders!$V$1:$V$8,Orders!$X$1:$X$8,Orders!$Y$1:$Y$8")
ActiveSheet.Shapes("Chart 2").ScaleWidth 0.7933333333, msoFalse, _
  msoScaleFromTopLeft
ActiveSheet.Shapes("Chart 2").ScaleHeight 0.8486111111, msoFalse, _
  msoScaleFromTopLeft
ActiveSheet.Shapes("Chart 2").IncrementLeft 805
ActiveSheet.Shapes("Chart 2").IncrementTop 90
ActiveChart.ChartTitle.Select
ActiveChart.ChartTitle.Text = "Shipped v/s On-time"
Selection.Format.TextFrame2.TextRange.Characters.Text = "Shipped v/s On-time"
With Selection.Format.TextFrame2.TextRange.Characters(1, 19).ParagraphFormat
  .TextDirection = msoTextDirectionLeftToRight
  .Alignment = msoAlignCenter
End With
With Selection.Format.TextFrame2.TextRange.Characters(1, 19).Font
  .BaselineOffset = 0
  .Bold = msoFalse
  .NameComplexScript = "+mn-cs"
  .NameFarEast = "+mn-ea"
  .Fill.Visible = msoTrue
```

```
    .Fill.ForeColor.RGB = RGB(89, 89, 89)
    .Fill.Transparency = 0
    .Fill.Solid
    .Size = 14
    .Italic = msoFalse
    .Kerning = 12
    .Name = "+mn-lt"
    .UnderlineStyle = msoNoUnderline
    .Spacing = 0
  End With
Range("V1:V8,W1:W8,X1:X8").Select
ActiveSheet.Shapes.AddChart2(201, xlColumnClustered).Select
ActiveChart.SetSourceData Source:=Range( _
    "Orders!$V$1:$V$8,Orders!$W$1:$W$8,Orders!$X$1:$X$8")
ActiveChart.FullSeriesCollection(1).ChartType = xlColumnClustered
ActiveChart.FullSeriesCollection(1).AxisGroup = 1
ActiveChart.FullSeriesCollection(2).ChartType = xlLine
ActiveChart.FullSeriesCollection(2).AxisGroup = 1
ActiveChart.ChartTitle.Select
ActiveChart.ChartTitle.Text = "Conversion of ordered to shipped"
Selection.Format.TextFrame2.TextRange.Characters.Text = _
    "Conversion of ordered to shipped"
With Selection.Format.TextFrame2.TextRange.Characters(1, 32).ParagraphFormat
    .TextDirection = msoTextDirectionLeftToRight
    .Alignment = msoAlignCenter
End With
With Selection.Format.TextFrame2.TextRange.Characters(1, 32).Font
    .BaselineOffset = 0
    .Bold = msoFalse
    .NameComplexScript = "+mn-cs"
    .NameFarEast = "+mn-ea"
```

```
    .Fill.Visible = msoTrue
    .Fill.ForeColor.RGB = RGB(89, 89, 89)
    .Fill.Transparency = 0
    .Fill.Solid
    .Size = 14
    .Italic = msoFalse
    .Kerning = 12
    .Name = "+mn-lt"
    .UnderlineStyle = msoNoUnderline
    .Spacing = 0
    .Strike = msoNoStrike
  End With
  ActiveSheet.ChartObjects("Chart 3").Activate
  ActiveSheet.Shapes("Chart 3").ScaleWidth 0.8616666667, msoFalse, _
    msoScaleFromTopLeft
  ActiveSheet.Shapes("Chart 3").ScaleHeight 0.8541666667, msoFalse, _
    msoScaleFromTopLeft
  ActiveSheet.Shapes("Chart 3").IncrementLeft 1110.4
  ActiveSheet.Shapes("Chart 3").IncrementTop 87.6

' A pop-up to indicate end of macro
  MsgBox "Task 2 completed succesfully!", vbInformation, "Operation Status"

End Sub
```

Macro 3: Monthly_figures()

Owner: Krishna Rao Ramesh

Date Created: 27/11/2024

Last Updated: 27/11/2024

Folder Location: *As per location after assignment submission*.

Description: Comparing monthly figures based on sales and payments.

Instructions:

1. Click on developer tab and select macros. Run macro 'Monthly_figures' or use the shortcut Ctrl+shift+T.
2. The macro will calculate the total order value for each month and the equivalent quantity sold. Also, the payments received for each month is mentioned right next to total quantity.
3. A combination of a drop down and macro buttons are used to make the result dynamic. The buttons are created using a different macro (code present below) and this is in turn, referenced inside macro 3.
4. A list of probabilities related to total sales figures is calculated based on common empirical probability approach. All observations are taken up to arrive at probability percentages.
5. In a nutshell, the button macro is used to assign name of city to cell O35 depending on the button clicked. Please make sure inputs in cell V1 and button are activated.

Code:

```
Sub Monthly_figures()
'
' Monthly_figures Macro
' Comparing monthly figures based on sales and payments.
'
' Keyboard Shortcut: Ctrl+Shift+T
'
' Creating an array with all the necessary headers to speed up the sub-task of naming each column.
Dim column_names As Variant
Dim column_names2 As Variant
  Application.ScreenUpdating = False
  column_names = Array("OrderId", "Sales_Rep", "Cust_No", "Office_code",
    "Office_location", _
        "OrderDate", "Year", "Month", "OrderQuantity", "Total_Order_Value")
```

```
Sheets("Comp.Monthly_fig").Select
Range("A1:J1").Value = column_names
Columns("I:I").EntireColumn.AutoFit
Columns("J:J").EntireColumn.AutoFit
Columns("E:E").EntireColumn.AutoFit
Columns("D:D").EntireColumn.AutoFit

' Calculation of total order value by referencing multiple columns from Orders,
    Order_details. This is linked to each office location.
Range("A2").Select
ActiveCell.Formula2R1C1 = "=UNIQUE(Order_details!RC:R[2995]C)"
Range("B2").Select
ActiveCell.FormulaR1C1 = "=VLOOKUP(RC[-1],Orders!C[-1]:C[9],11,0)"
Selection.AutoFill Destination:=Range("B2:B327")
Range("C2").Select
ActiveCell.FormulaR1C1 = "=VLOOKUP(RC[-2],Orders!C[-2]:C[6],9,0)"
Selection.AutoFill Destination:=Range("C2:C327")
Range("D2").Select
ActiveCell.FormulaR1C1 = "=VLOOKUP(RC[-2],Employees!C1:C6,6,0)"
Selection.AutoFill Destination:=Range("D2:D327")
Range("E2").Select
ActiveCell.FormulaR1C1 = "=VLOOKUP(RC[-1],of,2,0)"
Selection.AutoFill Destination:=Range("E2:E327")
Range("F2").Select
ActiveCell.FormulaR1C1 = "=VLOOKUP(RC[-5],Orders!C[-5]:C[-4],2,0)"
Selection.NumberFormat = "m/d/yyyy"
Selection.AutoFill Destination:=Range("F2:F327")
Range("G2").Select
ActiveCell.FormulaR1C1 = "=YEAR(RC[-1])"
Selection.AutoFill Destination:=Range("G2:G327")
Range("H2").Select
```

```
    ActiveCell.FormulaR1C1 = "=TEXT(RC[-2],""MMM"")"
    Selection.AutoFill Destination:=Range("H2:H327")
    Range("I2").Select
    ActiveCell.FormulaR1C1 = _
      "=SUMIFS(Order_details!C[-6],Order_details!C[-8],Comp.Monthly_fig!RC[-8])"
    Selection.AutoFill Destination:=Range("I2:I327")
    Range("J2").Select
    Application.CutCopyMode = False

' Total price for each product code is calculated in order_details sheet. This is used to
      compute the total order value.
    Sheets("Order_details").Select
    Range("F1").Select
    ActiveCell.FormulaR1C1 = "TotalPrice"
    Range("F2").Select
    Application.CutCopyMode = False
    ActiveCell.FormulaR1C1 = "=RC[-3]*RC[-2]"
    Range("F2").Select
    Selection.AutoFill Destination:=Range("F2:F2997")
    Sheets("Comp.Monthly_fig").Select
    Range("J2").Select
    ActiveCell.FormulaR1C1 = _
      "=SUMIFS(Order_details!C[-4],Order_details!C[-9],Comp.Monthly_fig!RC[-9])"
    Selection.AutoFill Destination:=Range("J2:J327")

' Compiling all necessary attributes to figure out payments received each month
    Sheets("Payments").Select
    column_names2 = Array("Year", "month", "office_loc")
    Range("E1:G1").Value = column_names2
    Range("E2").FormulaR1C1 = "=YEAR(RC[-2])"
    Range("F2").FormulaR1C1 = "=TEXT(RC[-3],""MMM"")"
```

```
Range("G2").FormulaR1C1 = "=VLOOKUP(RC[-6],Comp.Monthly_fig!C[-4]:C[-2],3,0)"
Range("E2").Select
Selection.AutoFill Destination:=Range("E2:E274")
Range("F2").Select
Selection.AutoFill Destination:=Range("F2:F274")
Range("G2").Select
Selection.AutoFill Destination:=Range("G2:G274")

' Calling another macro to add buttons to worksheet Comp.monthly_fig
Call Dynamic_result_by_year

'A drop down in V1 is created with years as input.
Sheets("Comp.Monthly_fig").Select
Range("U1").Select
ActiveCell.FormulaR1C1 = "Year"
Range("V1").Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
  .LineStyle = xlContinuous
  .ColorIndex = 0
  .TintAndShade = 0
  .Weight = xlThin
End With
With Selection.Borders(xlEdgeTop)
  .LineStyle = xlContinuous
  .ColorIndex = 0
  .TintAndShade = 0
  .Weight = xlThin
End With
With Selection.Borders(xlEdgeBottom)
```

```
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlInsideVertical)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Validation
    .Delete
    .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
    xlBetween, Formula1:="2003, 2004, 2005"
    .IgnoreBlank = True
    .InCellDropdown = True
    .InputTitle = ""
    .ErrorTitle = ""
```

```
    .InputMessage = ""
    .ErrorMessage = ""
    .ShowInput = True
    .ShowError = True
  End With

' A table is created to summarise net revenue/equivalent quantity for each month as well
    as the payments received
  Range("Q6").Value = "Month"
  Range("R6").Value = "TotalOrderValue"
  Range("S6").Value = "TotalOrdQty"
  Columns("R:R").EntireColumn.AutoFit
  Columns("S:S").EntireColumn.AutoFit
  Range("Q7").Value = "Jan"
  Range("Q8").Value = "Feb"
  Range("Q7").Select
  Selection.AutoFill Destination:=Range("Q7:Q18"), Type:=xlFillDefault
  Range("R7").Select
  ActiveCell.FormulaR1C1 = _
    "=SUMIFS(C[-8],C[-11],R1C22,C[-13],R35C15,C[-10],RC[-1])"
  Selection.AutoFill Destination:=Range("R7:R18")
  Application.CutCopyMode = False
  Range("S7").Select
  ActiveCell.FormulaR1C1 = _
    "=SUMIFS(C[-10],C[-12],R1C22,C[-14],R35C15,C[-11],RC[-2])"
  Selection.AutoFill Destination:=Range("S7:S18")
  Range("T6").Select
  ActiveCell.FormulaR1C1 = "Payments"
  Range("T7").Select
  ActiveCell.FormulaR1C1 = _
    "=SUMIFS(Payments!C[-16],Payments!C[-15],Comp.Monthly_fig!R1C22,Payments!C[-
     13],R35C15,Payments!C[-14],RC[-3])"
```

```
Selection.AutoFill Destination:=Range("T7:T18")
Range("Q19").Select
ActiveCell.FormulaR1C1 = "Total"
Range("Q20").Select
ActiveCell.FormulaR1C1 = "Average"
Range("R19").Select
ActiveCell.FormulaR1C1 = "=SUM(R[-12]C:R[-1]C)"
Range("R20").Select
ActiveCell.FormulaR1C1 = "=AVERAGE(R[-13]C:R[-2]C)"
Range("R19").Select
Selection.AutoFill Destination:=Range("R19:T19"), Type:=xlFillDefault
Range("R20").Select
Selection.AutoFill Destination:=Range("R20:T20"), Type:=xlFillDefault

' Probabilities pertaining to sales and payments are computed using empirical probability
Range("AB8").Select
ActiveCell.FormulaR1C1 = _
  "=COUNTIF(R[-1]C[-9]:R[10]C[-9],"">""&R[12]C[-9])/COUNTA(R[-1]C[-9]:R[10]C[-9])"
Range("W8").Select
ActiveCell.FormulaR1C1 = _
  "Probability of a month selling quantity more than the average"
Range("W9").Select
Columns("AA:AA").ColumnWidth = 13.33
ActiveCell.FormulaR1C1 = _
  "Probability of a month bringing in revenue less than the average"
Range("AB9").Select
ActiveCell.FormulaR1C1 = _
  "=COUNTIF(R[-2]C[-10]:R[9]C[-10],"">""&R[11]C[-10])/COUNTA(R[-2]C[-10]:R[9]C[-10])"
Range("W10").Select
ActiveCell.FormulaR1C1 = "Probability of a month not making any sale"
Range("AB10").Select
```

```
ActiveCell.FormulaR1C1 = _
  "=COUNTIF(R[-3]C[-10]:R[8]C[-10],""=0"")/COUNTA(R[-3]C[-10]:R[8]C[-10])"
Range("W11").Select
ActiveCell.FormulaR1C1 = "Probability of a month making atleast 10000 (euros)"
Range("AB11").Select
ActiveCell.FormulaR1C1 = _
  "=COUNTIF(R[-4]C[-8]:R[7]C[-8],"">10000"")/COUNTA(R[-4]C[-8]:R[7]C[-8])"

' A brief synopsis of our previous findings is given in cells W15/W16.
Range("W14").Select
ActiveCell.FormulaR1C1 = "Sales Probability during off-season"
Selection.Font.Underline = xlUnderlineStyleSingle
Range("W15").Select
ActiveCell.FormulaR1C1 = _
  "These findings will help in understanding how much % of our months are non-
   productive and also productive at the same time."
Range("W15").Select
Selection.Font.Bold = True
Range("W16").Select
ActiveCell.FormulaR1C1 = _
  "We have to devise new strategies to counter our inefficiencies at the start of the year."
Range("W16").Select
Selection.Font.Bold = True
Range("Q6:R18,T6:T18").Select

' A dynamic chart is added to analyse how inflow of money fluctuates across different
    months
Range("A1").Select
ActiveSheet.Shapes.AddChart2(276, xlArea).Select
ActiveChart.SetSourceData Source:=Range( _
  "'Comp.Monthly_fig'!$Q$6:$R$18,'Comp.Monthly_fig'!$T$6:$T$18")
ActiveChart.ChartTitle.Select
```

```
Application.CommandBars("Format Object").Visible = False
ActiveChart.ChartTitle.Text = "TotalOrdValue v/s Payments"
Selection.Format.TextFrame2.TextRange.Characters.Text = _
  "TotalOrdValue v/s Payments"
With Selection.Format.TextFrame2.TextRange.Characters(1, 26).ParagraphFormat
  .TextDirection = msoTextDirectionLeftToRight
  .Alignment = msoAlignCenter
End With
With Selection.Format.TextFrame2.TextRange.Characters(1, 5).Font
  .BaselineOffset = 0
  .Bold = msoFalse
  .NameComplexScript = "+mn-cs"
  .NameFarEast = "+mn-ea"
  .Fill.Visible = msoTrue
  .Fill.ForeColor.RGB = RGB(89, 89, 89)
  .Fill.Transparency = 0
  .Fill.Solid
  .Size = 14
  .Italic = msoFalse
  .Kerning = 12
  .Name = "+mn-lt"
  .UnderlineStyle = msoNoUnderline
  .Spacing = 0
  .Strike = msoNoStrike
End With
With Selection.Format.TextFrame2.TextRange.Characters(6, 21).Font
  .BaselineOffset = 0
  .Bold = msoFalse
  .NameComplexScript = "+mn-cs"
  .NameFarEast = "+mn-ea"
  .Fill.Visible = msoTrue
```

```
    .Fill.ForeColor.RGB = RGB(89, 89, 89)
    .Fill.Transparency = 0
    .Fill.Solid
    .Size = 14
    .Italic = msoFalse
    .Kerning = 12
    .Name = "+mn-lt"
    .UnderlineStyle = msoNoUnderline
    .Spacing = 0
    .Strike = msoNoStrike
  End With
  ActiveChart.ChartArea.Select
  ActiveSheet.Shapes("Chart 1").IncrementLeft 900
  ActiveSheet.Shapes("Chart 1").IncrementTop 154.2
  Range("O35").Select

' The cell associated with the subroutine is hidden. DO NOT REMOVE VALIDATION RULES
    FROM SHEET.
  With Selection.Font
    .ThemeColor = xlThemeColorDark1
    .TintAndShade = 0
  End With

' A pop-up to indicate end of macro
  MsgBox "Task 3 completed succesfully!", vbInformation, "Operation Status"



End Sub
```

"Subroutine referenced in above macro"

```
Sub Dynamic_result_by_year()
```

```
    Dim ws As Worksheet
    Dim Locations As Variant
    Dim Offset1 As Double
    Dim Offset2 As Double
    Dim button As button
    Dim i As Integer
    Dim Activated_button As String

    Set ws = ThisWorkbook.Sheets("Comp.Monthly_fig")

' On error is used to bypass error statement if no buttons are present in sheet. This is reset in last line of this block.
    On Error Resume Next
    For Each button In ws.Buttons
        button.Delete
    Next button
    On Error GoTo 0

' Different variables are created to dsitribute proper spacing between buttons. Offset1 & 2 take up the left-corner/top-corner position of R3 and C15.
    Locations = Array("Boston", "London", "NYC", "Paris", "Tokyo", "San Francisco", "Sydney")
    DistanceBtwB = 110
    Offset1 = ws.Cells(3, 15).Left
    Offset2 = ws.Cells(3, 15).Top

' Buttons are generated and spaced evenly using for loop. Each button added in the loop will be offset by the above variables (Offset1 & 2)
    For i = LBound(Locations) To UBound(Locations)
        Set button = ws.Buttons.Add(Left:=Offset1 + (i * DistanceBtwB), Top:=Offset2, Width:=75, Height:=25)

'Assigning visual name for button and course of action for button (Reference the exact macro name here).
```

```
    button.Caption = Locations(i)
    button.OnAction = "Dynamic_result_by_year"
    button.Name = Locations(i)
  Next i

' To bypass unexpected errors thrown by application caller, on error statement is used (this is reset in last line of this block).  The application.caller property return the name of button clicked.
  On Error Resume Next
  Activated_button = Application.Caller
  On Error GoTo 0

' The if condition is triggered only if the buttons are of string datatype. Select method is used to assign different scenarios to activated button.
  If TypeName(Activated_button) = "String" Then
    Select Case Activated_button
      Case "Boston"
        ws.Range("O35").Value = "Boston"
      Case "London"
        ws.Range("O35").Value = "London"
      Case "NYC"
        ws.Range("O35").Value = "NYC"
      Case "Paris"
        ws.Range("O35").Value = "Paris"
      Case "Tokyo"
        ws.Range("O35").Value = "Tokyo"
      Case "San Francisco"
        ws.Range("O35").Value = "San Francisco"
      Case "Sydney"
        ws.Range("O35").Value = "Sydney"
    End Select
```

End If

End Sub

Macro 4: Probability_findings()

Owner: Krishna Rao Ramesh

Date Created: 29/11/2024

Last Updated: 29/11/2024

Folder Location: *As per location after assignment submission*.

Description: For comparing variables based on Probabilities.

Instructions:

1. Click on developer tab and select macros. Run macro 'Probability_findings' or use the shortcut Ctrl+shift+Y.
2. The macro will generate tables with various set of probabilities and forecasting data for Order values and On-time delivery.
3. Due to presence of partial seasonality and trend, forecasting is done to understand probability of predicted values exceeding average order value for first six months. This case was specifically selected since order values in 2003 and 2004 were comparatively low with respect to 2005 for cities such as Sydney, Boston and San Francisco.
4. To add different set of inputs in a 2D excel range, a mixture of transpose functions and array indexing was used to insert different set of values to a 7x3 excel table.
5. Probability for unsuccessful delivery rates are determined to illustrate a clear change in performance for the year 2005.

Code:

```
Sub Probability_findings()
'
' Probability_findings Macro
'
' Keyboard Shortcut: Ctrl+Shift+Y
'
Dim cities As Variant
Dim values As Variant

Application.ScreenUpdating = False
```

```
' a drop-down is assigned to D1 with three cities. This would be used as input for our forecasting table and probability mass function.
  Range("C1").Select
  ActiveCell.FormulaR1C1 = "Region"
  Range("D1").Select
  Selection.Borders(xlDiagonalDown).LineStyle = xlNone
  Selection.Borders(xlDiagonalUp).LineStyle = xlNone
  With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
  End With
  With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
  End With
  With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
  End With
  With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
  End With
```

```
    With Selection.Borders(xlInsideVertical)
      .LineStyle = xlContinuous
      .ColorIndex = 0
      .TintAndShade = 0
      .Weight = xlThin
    End With
    With Selection.Borders(xlInsideHorizontal)
      .LineStyle = xlContinuous
      .ColorIndex = 0
      .TintAndShade = 0
      .Weight = xlThin
    End With
    With Selection.Validation
      .Delete
      .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
      xlBetween, Formula1:="Boston, San Francisco, Sydney"
      .IgnoreBlank = True
      .InCellDropdown = True
      .InputTitle = ""
      .ErrorTitle = ""
      .InputMessage = ""
      .ErrorMessage = ""
      .ShowInput = True
      .ShowError = True
    End With

' Three separate tables for each year is created.
    Range("D3").Select
    ActiveCell.FormulaR1C1 = "2003"
    Range("E3").Select
    ActiveCell.FormulaR1C1 = "P(y for Q1,Q2>average) ->"
```

```
With ActiveCell.Characters(Start:=1, Length:=2).Font
    .Name = "Aptos Narrow"
    .FontStyle = "Regular"
    .Size = 11
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ThemeColor = xlThemeColorLight1
    .TintAndShade = 0
    .ThemeFont = xlThemeFontMinor
End With
With ActiveCell.Characters(Start:=3, Length:=1).Font
    .Name = "Aptos Narrow"
    .FontStyle = "Regular"
    .Size = 11
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ThemeColor = xlThemeColorLight1
    .TintAndShade = 0
    .ThemeFont = xlThemeFontNone
End With
With ActiveCell.Characters(Start:=4, Length:=22).Font
    .Name = "Aptos Narrow"
    .FontStyle = "Regular"
```

```
    .Size = 9.35
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ThemeColor = xlThemeColorLight1
    .TintAndShade = 0
    .ThemeFont = xlThemeFontNone
  End With

' Probability specifically for first 6 months of the year is calculated. This is also done for other tables
  Range("G3").Select
  Columns("E:E").ColumnWidth = 9.44
  ActiveCell.FormulaR1C1 = _

"=COUNTIF(R[2]C[1]:R[7]C[1],"">""&AVERAGE(R[2]C[1]:R[13]C[1]))/COUNTA(R[2]C[1]:R[7]C[1])"

' Seasonal indexes, Linear trend forecast with a regression equation and Forecast taking seasonality into account is computed.
  Range("B4").Select
  ActiveCell.FormulaR1C1 = "Numeric value"
  Range("B5").Select
  ActiveCell.FormulaR1C1 = "1"
  Range("B6").Select
  ActiveCell.FormulaR1C1 = "2"
  Range("B5:B6").Select
  Selection.AutoFill Destination:=Range("B5:B16"), Type:=xlFillDefault
  Range("C4").Select
```

```
    ActiveCell.FormulaR1C1 = "month"
    Range("C5").Select
    ActiveCell.FormulaR1C1 = "Jan"
    Range("C6").Select
    ActiveCell.FormulaR1C1 = "Feb"
    Range("C5:C6").Select
    Selection.AutoFill Destination:=Range("C5:C16"), Type:=xlFillDefault
    Range("D4").Select
    ActiveCell.FormulaR1C1 = "TotalOrderValue"
    Range("D5").Select
    ActiveCell.FormulaR1C1 = _

"=SUMIFS(Comp.Monthly_fig!C[6],Comp.Monthly_fig!C[3],2003,Comp.Monthly_fig!C[1],R1
C4,Comp.Monthly_fig!C[4],RC[-1])"
    Selection.AutoFill Destination:=Range("D5:D16")
    Range("E4").Select
    Columns("D:D").EntireColumn.AutoFit
    ActiveCell.FormulaR1C1 = "Ratio"
    Range("E5").Select
    ActiveCell.FormulaR1C1 = "=RC[-1]/AVERAGE(RC[-1]:R[11]C[-1])"
    Selection.AutoFill Destination:=Range("E5:E16")
    Range("F4").Select
    ActiveCell.FormulaR1C1 = "seasonal indexes"
    Range("G4").Select
    Columns("F:F").EntireColumn.AutoFit
    ActiveCell.FormulaR1C1 = "Linear Trend Forecast"

' Intercept and slope are calculated for each set of values to identify our trendline values.
    Range("G1").Select
    ActiveCell.FormulaR1C1 = "Intercept"
    Range("G2").Select
    ActiveCell.FormulaR1C1 = "Slope"
```

```
Application.CutCopyMode = False
Range("H1").Select
ActiveCell.FormulaR1C1 = _
  "=INTERCEPT(R[4]C[-4]:R[15]C[-4],R[4]C[-6]:R[15]C[-6])"
Range("H2").Select
ActiveCell.FormulaR1C1 = "=SLOPE(R[3]C[-4]:R[14]C[-4],R[3]C[-6]:R[14]C[-6])"
Range("G5").Select
Application.CutCopyMode = False
ActiveCell.FormulaR1C1 = "=(R2C8*RC[-5])+R1C8"
Selection.AutoFill Destination:=Range("G5:G16")
Columns("G:G").EntireColumn.AutoFit
Range("G3").Select
Selection.Font.Bold = True
Range("H4").Select
ActiveCell.FormulaR1C1 = "Forecast with seasonality"
Range("H5").Select
Application.CutCopyMode = False
ActiveCell.FormulaR1C1 = "=RC[-1]*RC[-2]"
Range("H5").Select
Selection.AutoFill Destination:=Range("H5:H16")
Range("D3:F3").Select
Selection.Font.Underline = xlUnderlineStyleSingle
```

' A chart visualizing this partial seasonality and gradual upward trend is generated right below each table

```
Range("C5:D16").Select
ActiveSheet.Shapes.AddChart2(227, xlLine).Select
ActiveChart.SetSourceData Source:=Range("Probability_findings!$C$5:$D$16")
ActiveSheet.Shapes("Chart 2").IncrementLeft -304.4705511811
ActiveSheet.Shapes("Chart 2").IncrementTop 175.4117322835
ActiveChart.ChartTitle.Select
```

```
Application.CommandBars("Format Object").Visible = False
ActiveChart.ChartTitle.Text = "2003"
Selection.Format.TextFrame2.TextRange.Characters.Text = "2003"
With Selection.Format.TextFrame2.TextRange.Characters(1, 4).ParagraphFormat
  .TextDirection = msoTextDirectionLeftToRight
  .Alignment = msoAlignCenter
End With
With Selection.Format.TextFrame2.TextRange.Characters(1, 4).Font
  .BaselineOffset = 0
  .Bold = msoFalse
  .NameComplexScript = "+mn-cs"
  .NameFarEast = "+mn-ea"
  .Fill.Visible = msoTrue
  .Fill.ForeColor.RGB = RGB(89, 89, 89)
  .Fill.Transparency = 0
  .Fill.Solid
  .Size = 14
  .Italic = msoFalse
  .Kerning = 12
  .Name = "+mn-lt"
  .UnderlineStyle = msoNoUnderline
  .Spacing = 0
  .Strike = msoNoStrike
End With
ActiveChart.ChartArea.Select

' This line of code specifically creates our linear trend for the line graph
ActiveChart.FullSeriesCollection(1).Trendlines.Add Type:=xlLinear, Forward _
  :=0, Backward:=0, DisplayEquation:=0, DisplayRSquared:=0, Name:= _
  "Linear (Series1)"
```

```
' The same is done for the year 2004
  Range("M4").Select
  ActiveCell.FormulaR1C1 = "Numeric value"
  Range("N4").Select
  ActiveCell.FormulaR1C1 = "month"
  Range("O4").Select
  ActiveCell.FormulaR1C1 = "TotalOrderValue"
  Range("P4").Select
  Columns("O:O").EntireColumn.AutoFit
  ActiveCell.FormulaR1C1 = "Ratio"
  Range("Q4").Select
  ActiveCell.FormulaR1C1 = "Seasonal indexes"
  Range("R4").Select
  Columns("Q:Q").EntireColumn.AutoFit
  ActiveCell.FormulaR1C1 = "Linear Trend Forecast"
  Range("S4").Select
  Columns("R:R").EntireColumn.AutoFit
  ActiveCell.FormulaR1C1 = "Forecast with seasonality"
  Range("M5").Select
  ActiveCell.FormulaR1C1 = "1"
  Range("M6").Select
  ActiveCell.FormulaR1C1 = "2"
  Range("M5:M6").Select
  Selection.AutoFill Destination:=Range("M5:M16"), Type:=xlFillDefault
  Range("C5").Select
  Range(Selection, Selection.End(xlDown)).Select
  Selection.Copy
  Range("N5").Select
  ActiveSheet.Paste
  Range("P3").Select
  Application.CutCopyMode = False
```

```
  ActiveCell.FormulaR1C1 = "2004"
  Range("E3").Select
  Selection.Copy
  Range("Q3").Select
  ActiveSheet.Paste
  Range("S3").Select
  Selection.Font.Bold = True
  Application.CutCopyMode = False
  ActiveCell.FormulaR1C1 = _
    "=COUNTIF(R[2]C:R[7]C,"">""&AVERAGE(R[2]C:R[13]C))/COUNTA(R[2]C:R[7]C)"
  Range("S1").Select
  ActiveCell.FormulaR1C1 = "Intercept"
  Range("S2").Select
  ActiveCell.FormulaR1C1 = "Slope"
  Range("T1").Select
  ActiveCell.FormulaR1C1 = _
    "=INTERCEPT(R[4]C[-5]:R[15]C[-5],R[4]C[-7]:R[15]C[-7])"
  Range("T2").Select
  ActiveCell.FormulaR1C1 = "=SLOPE(R[3]C[-5]:R[14]C[-5],R[3]C[-7]:R[14]C[-7])"
  Range("O5").Select
  ActiveCell.FormulaR1C1 = _
    "=SUMIFS(Comp.Monthly_fig!C[-5],Comp.Monthly_fig!C[-
8],2004,Comp.Monthly_fig!C[-10],R1C4,Comp.Monthly_fig!C[-7],Probability_findings!RC[-
1])"
  Selection.AutoFill Destination:=Range("O5:O16")
  Range("P5").Select
  ActiveCell.FormulaR1C1 = "=RC[-1]/AVERAGE(RC[-1]:R[11]C[-1])"
  Selection.AutoFill Destination:=Range("P5:P16")
  Range("Q5").Select
  ActiveCell.FormulaR1C1 = "=AVERAGE(RC[-1],RC[-12])"
  Selection.AutoFill Destination:=Range("Q5:Q16")
  Range("F5").Select
```

```
ActiveCell.FormulaR1C1 = "=AVERAGE(RC[-1],RC[10])"
Selection.AutoFill Destination:=Range("F5:F16")
Range("R5").Select
Application.CutCopyMode = False
ActiveCell.FormulaR1C1 = "=(R2C20*RC[-5])+R1C20"
Selection.AutoFill Destination:=Range("R5:R16")
Range("S5").Select
Application.CutCopyMode = False
ActiveCell.FormulaR1C1 = "=RC[-1]*RC[-2]"
Selection.AutoFill Destination:=Range("S5:S16")
Range("P3").Select
Selection.Font.Underline = xlUnderlineStyleSingle

' The table with recent sales figures is put together from cell X3 to AB9.
Range("X3").Select
ActiveCell.FormulaR1C1 = "2005"
Selection.Font.Underline = xlUnderlineStyleSingle
Range("Q3").Select
Selection.Copy
Range("Y3").Select
ActiveSheet.Paste
Range("AB3").Select
Application.CutCopyMode = False
Range("M4:N9").Select
Selection.Copy
Range("X4").Select
ActiveSheet.Paste
Range("O4:O9").Select
Application.CutCopyMode = False
Selection.Copy
Range("Z4").Select
```

```
  ActiveSheet.Paste
  Columns("Z:Z").EntireColumn.AutoFit
  Range("AA4").Select
  Application.CutCopyMode = False
  ActiveCell.FormulaR1C1 = "Linear Trend Forecast"
  Range("AB4").Select
  Columns("AA:AA").EntireColumn.AutoFit
  ActiveCell.FormulaR1C1 = "Forecast with seasonality"
  Range("AB1").Select
  ActiveCell.FormulaR1C1 = "Intercept"
  Range("AB2").Select
  ActiveCell.FormulaR1C1 = "Slope"
  Range("AC1").Select
  ActiveCell.FormulaR1C1 = "=INTERCEPT(R[4]C[-3]:R[8]C[-3],R[4]C[-5]:R[8]C[-5])"
  Range("AC2").Select
  ActiveCell.FormulaR1C1 = "=SLOPE(R[3]C[-3]:R[7]C[-3],R[3]C[-5]:R[7]C[-5])"
  Range("Z5").Select
  ActiveCell.FormulaR1C1 = _
    "=SUMIFS(Comp.Monthly_fig!C[-16],Comp.Monthly_fig!C[-
19],2005,Comp.Monthly_fig!C[-21],R1C4,Comp.Monthly_fig!C[-
18],Probability_findings!RC[-1])"
  Selection.AutoFill Destination:=Range("Z5:Z9")
  Range("AA5").Select
  ActiveCell.FormulaR1C1 = "=(R2C29*RC[-3])+R1C29"
  Selection.AutoFill Destination:=Range("AA5:AA9")
  Range("AB5").Select
  Application.CutCopyMode = False
  ActiveCell.FormulaR1C1 = "=RC[-1]*RC[-11]"
  Selection.AutoFill Destination:=Range("AB5:AB9")
  Range("AB3").Select
  Selection.Font.Bold = True
  ActiveCell.FormulaR1C1 = _
```

```
    "=COUNTIF(R[2]C:R[6]C,"">"" &AVERAGE(R[2]C:R[6]C))/COUNTA(R[2]C:R[6]C)"


'Similar line graphs with regression projections are built for years 2004 and 2005.
    Range("N4:O16").Select
    ActiveSheet.Shapes.AddChart2(227, xlLine).Select
    ActiveChart.SetSourceData Source:=Range("Probability_findings!$N$4:$O$16")
    ActiveSheet.Shapes("Chart 3").IncrementLeft 488
    ActiveSheet.Shapes("Chart 3").IncrementTop 175
    ActiveChart.ChartTitle.Select
    ActiveChart.FullSeriesCollection(1).Trendlines.Add Type:=xlLinear, Forward _
      :=0, Backward:=0, DisplayEquation:=0, DisplayRSquared:=0, Name:= _
      "Linear (Series1)"
    ActiveChart.ChartTitle.Text = "2004"
    Selection.Format.TextFrame2.TextRange.Characters.Text = "2004"
    With Selection.Format.TextFrame2.TextRange.Characters(1, 4).ParagraphFormat
      .TextDirection = msoTextDirectionLeftToRight
      .Alignment = msoAlignCenter
    End With
    With Selection.Format.TextFrame2.TextRange.Characters(1, 4).Font
      .BaselineOffset = 0
      .Bold = msoFalse
      .NameComplexScript = "+mn-cs"
      .NameFarEast = "+mn-ea"
      .Fill.Visible = msoTrue
      .Fill.ForeColor.RGB = RGB(89, 89, 89)
      .Fill.Transparency = 0
      .Fill.Solid
      .Size = 14
      .Italic = msoFalse
      .Kerning = 12
      .Name = "+mn-lt"
```

```
    .UnderlineStyle = msoNoUnderline
    .Spacing = 0
    .Strike = msoNoStrike
  End With

' Another probability denoting the % of order values that are above average for 2005.
  Range("Y12").Select
  ActiveCell.FormulaR1C1 = "P(X>average), where x is actual TOV"
  Range("AB12").Select
  Selection.Font.Bold = True
  ActiveCell.FormulaR1C1 = _
    "=COUNTIF(R[-7]C[-2]:R[-3]C[-2],"">""&AVERAGE(R[-7]C[-2]:R[-3]C[-2]))/COUNTA(R[-7]C[-2]:R[-3]C[-2])"
  Range("Y5:Z9").Select
  ActiveSheet.Shapes.AddChart2(227, xlLine).Select
  ActiveChart.SetSourceData Source:=Range("Probability_findings!$Y$5:$Z$9")
  ActiveSheet.Shapes("Chart 4").IncrementLeft 986
  ActiveSheet.Shapes("Chart 4").IncrementTop 100.6470866142
  ActiveChart.ChartTitle.Select
  ActiveChart.FullSeriesCollection(1).Trendlines.Add Type:=xlLinear, Forward _
    :=0, Backward:=0, DisplayEquation:=0, DisplayRSquared:=0, Name:= _
    "Linear (Series1)"
  Application.CommandBars("Format Object").Visible = False
  ActiveChart.ChartTitle.Text = "2005"
  Selection.Format.TextFrame2.TextRange.Characters.Text = "2005"
  With Selection.Format.TextFrame2.TextRange.Characters(1, 4).ParagraphFormat
    .TextDirection = msoTextDirectionLeftToRight
    .Alignment = msoAlignCenter
  End With
  With Selection.Format.TextFrame2.TextRange.Characters(1, 4).Font
    .BaselineOffset = 0
```

```
    .Bold = msoFalse
    .NameComplexScript = "+mn-cs"
    .NameFarEast = "+mn-ea"
    .Fill.Visible = msoTrue
    .Fill.ForeColor.RGB = RGB(89, 89, 89)
    .Fill.Transparency = 0
    .Fill.Solid
    .Size = 14
    .Italic = msoFalse
    .Kerning = 12
    .Name = "+mn-lt"
    .UnderlineStyle = msoNoUnderline
    .Spacing = 0
    .Strike = msoNoStrike
  End With

' The table created below will differentiate the on-time delivery probabilities of different regions across the three years
  Range("E38").Value = "Office_loc"
  Range("F38").Value = "P(On-time)-2003"
  Range("G38").Value = "P(On-time)-2004"
  Range("H38").Value = "P(On-time)-2005"

' Two arrays to add input from task 2 to cells E38:H45
  cities = Array("Boston", "London", "NYC", "Paris", "Tokyo", "San Francisco", "Sydney")

' The application.transpose function pastes the array values vertically in specified cell range.
  Range("E39:E45").Value = Application.Transpose(cities)
  Application.CutCopyMode = False
  values = Array("0.33", "0.35", "0.29", "0.36", "0.50", "0.24", " 0.18", "0.17", "0.36", "0.29", "0.34", "0.14", "0.47", "0.29", "0.33", "0.40", "0", "0.65", "0.33", "0.55", "0.67")
```

```
'To populate a 2D excel range with 1D array, we assign each set of values to a tranpose function and their subsequent cell ranges.

   Range("F39:F45").Value = Application.Transpose(Array(values(0), values(1), values(2), values(3), values(4), values(5), values(6)))

   Range("G39:G45").Value = Application.Transpose(Array(values(7), values(8), values(9), values(10), values(11), values(12), values(13)))

   Range("H39:H45").Value = Application.Transpose(Array(values(14), values(15), values(16), values(17), values(18), values(19), values(20)))

   Range("F39:H45").Select

   Selection.Style = "Percent"


' Another line graph is created to understand variation in P(On-time) across 2003-05.

   Range("E38:H45").Select

   ActiveSheet.Shapes.AddChart2(227, xlLine).Select

   ActiveChart.SetSourceData Source:=Range("Probability_findings!$E$38:$H$45")

   ActiveSheet.Shapes("Chart 5").IncrementLeft -190.7059055118

   ActiveSheet.Shapes("Chart 5").IncrementTop 580

   ActiveChart.ChartTitle.Select

   Application.CommandBars("Format Object").Visible = False

   ActiveChart.ChartTitle.Text = "Comparison chart"

   Selection.Format.TextFrame2.TextRange.Characters.Text = "Comparison chart"

   With Selection.Format.TextFrame2.TextRange.Characters(1, 16).ParagraphFormat

      .TextDirection = msoTextDirectionLeftToRight

      .Alignment = msoAlignCenter

   End With

   With Selection.Format.TextFrame2.TextRange.Characters(1, 16).Font

      .BaselineOffset = 0

      .Bold = msoFalse

      .NameComplexScript = "+mn-cs"

      .NameFarEast = "+mn-ea"

      .Fill.Visible = msoTrue
```

```
    .Fill.ForeColor.RGB = RGB(89, 89, 89)
    .Fill.Transparency = 0
    .Fill.Solid
    .Size = 14
    .Italic = msoFalse
    .Kerning = 12
    .Name = "+mn-lt"
    .UnderlineStyle = msoNoUnderline
    .Spacing = 0
    .Strike = msoNoStrike
  End With

' Probability of offices underperforming (below the average) in on-time deliveries.
  Range("N39").Select
  ActiveCell.FormulaR1C1 = "2003"
  Selection.Font.Underline = xlUnderlineStyleSingle
  Range("N40").Select
  ActiveCell.FormulaR1C1 = "P(office loc delivery rate < average rate)"
  Range("R40").Select
  ActiveCell.FormulaR1C1 = _
    "=COUNTIF(R[-1]C[-12]:R[5]C[-12],""<""&AVERAGE(R[-1]C[-12]:R[5]C[-12]))/COUNTA(R[-1]C[-12]:R[5]C[-12])"
  Selection.Style = "Percent"
  Range("N42").Select
  ActiveCell.FormulaR1C1 = "2004"
  Selection.Font.Underline = xlUnderlineStyleSingle
  Range("N40").Select
  Selection.Copy
  Range("N43").Select
  ActiveSheet.Paste
  Range("R43").Select
```

```
    Application.CutCopyMode = False
    ActiveCell.FormulaR1C1 = _
      "=COUNTIFS(R[-4]C[-11]:R[2]C[-11],""<""&AVERAGE(R[-4]C[-11]:R[2]C[-
11]))/COUNTA(R[-4]C[-11]:R[2]C[-11])"
    Selection.Style = "Percent"
    Range("N45").Select
    ActiveCell.FormulaR1C1 = "2005"
    Selection.Font.Underline = xlUnderlineStyleSingle
    Range("N43").Select
    Selection.Copy
    Range("N46").Select
    ActiveSheet.Paste
    Range("R46").Select
    Application.CutCopyMode = False
    ActiveCell.FormulaR1C1 = _
      "=COUNTIF(R[-7]C[-10]:R[-1]C[-10],""<""&AVERAGE(R[-7]C[-12]:R[-1]C[-
11]))/COUNTA(R[-7]C[-10]:R[-1]C[-10])"
    Selection.Style = "Percent"


' The average success rate of each office.
    Range("N50").Select
    ActiveCell.FormulaR1C1 = "Average success rate"
    Selection.Font.Underline = xlUnderlineStyleSingle
    Range("N51").Select
    ActiveCell.FormulaR1C1 = "2003"
    Range("N52").Select
    ActiveCell.FormulaR1C1 = "2004"
    Range("N53").Select
    ActiveCell.FormulaR1C1 = "2005"
    Range("O51").Select
    ActiveCell.FormulaR1C1 = "=AVERAGE(R[-12]C[-9]:R[-6]C[-9])"
    Range("O52").Select
```

```
ActiveCell.FormulaR1C1 = "=AVERAGE(R[-13]C[-8]:R[-7]C[-8])"
Range("O53").Select
ActiveCell.FormulaR1C1 = "=AVERAGE(R[-14]C[-7]:R[-8]C[-7])"


' A pop-up to indicate end of macro
MsgBox "Task 4 completed succesfully!", vbInformation, "Operation Status"

End Sub
```

Macro 5: Summary_Sheet

Owner: Krishna Rao Ramesh

Date Created: 29/11/2024

Last Updated: 29/11/2024

Folder Location: *As per location after assignment submission*.

Description: Generates the summary of the project

Instructions:

1. Click on the Developer tab and select Macros. Run the macro "Probability_findings" or use the shortcut Ctrl + Shift + [key].
2. The product sheets contained some data mismatches. The macro cleaned the data in the Product Details sheet using the CONCATENATE function and copy-paste functionality. This process enabled the calculation of the price per unit and the profit the company received from selling a unit item.
3. The macro will generate the tables necessary to summarize the findings.
4. The macro uses the tables instead of pivotable in order to increase the complexity of macro performance.
5. The summary sheet includes the overall findings of the company with regards to 3 years of data. Since the yearly and monthly performances have been highlighted earlier, the summary sheet will provide insights about the overall sales and profit of the company for all 3 years.

Code:

```
Sub Summary()
' Summary_table Macro
' Purpose: Generate a summary table with various statistics and metrics
' Keyboard Shortcut: Ctrl+Shift+L
'

    Application.ScreenUpdating = False
    Dim ws As Worksheet

'calling another subroutine to clean my products sheet.
    Call ProductsSheetCleaning

' Creating a compiled table for probability as well as the count of orders across three years.
    Set ws = ThisWorkbook.Sheets("Summary_ContentTable")
    Sheets("Summary_ContentTable").Select
    ws.Range("A2").Formula2R1C1 = "=UNIQUE(Orders!R[-1]C[14]:R[325]C[14])"
    ws.Range("B2").Value = "Overall_Delivery_Status"
    ws.Range("C2").Value = "Probability"
    ws.Range("D2").Value = "2003"
    ws.Range("E2").Value = "2004"
    ws.Range("F2").Value = "2005"
    ws.Range("B3").FormulaR1C1 = "=COUNTIF(Orders!R[-
2]C[13]:R[324]C[13],Summary_ContentTable!RC[-1])"
    ws.Range("B3:B5").FillDown
    ws.Range("C3").FormulaR1C1 = "=RC[-1]/303"
    ws.Range("C3:C5").FillDown
    ws.Range("C3:C5").Style = "Percent"
    ws.Range("D3").FormulaR1C1 =
"=COUNTIFS(Orders!R1C15:R327C15,Summary_ContentTable!RC[-
3],Orders!R1C7:R327C7,2003)"
    ws.Range("D3:D5").FillDown
```

```
    ws.Range("E3").FormulaR1C1 = "=COUNTIFS(Orders!C[10],Summary_ContentTable!RC[-4],Orders!C[2],2004)"

    ws.Range("E3:E5").FillDown

    ws.Range("F3").FormulaR1C1 = "=COUNTIFS(Orders!C[9],Summary_ContentTable!RC[-5],Orders!C[1],2005)"

    ws.Range("F3:F5").FillDown


' Creating a second table with unique values from "Comp.Monthly_fig" sheet andn calculating the total order values/ respective quantity in all 7 regions (2003-05); along with the payments received from customers.

    ws.Range("A8").Formula2R1C1 = "=UNIQUE(Comp.Monthly_fig!R[-7]C[4]:R[319]C[4])"

    ws.Columns("A:A").ColumnWidth = 12.91

    ws.Range("B8").Value = "Order_Quantity"

    ws.Range("C8").Value = "Total_Ordervalue"

    ws.Range("D8").Value = "Total_Payment"

    With ws.Range("B9:B15")

    .FormulaR1C1 = "=SUMIFS(Comp.Monthly_fig!C3,Comp.Monthly_fig!C7,Summary_ContentTable!RC[-1])"

End With

    Range("B9").Select

    ActiveCell.FormulaR1C1 = _

       "=SUMIF(Comp.Monthly_fig!C5,Summary_ContentTable!RC[-1],Comp.Monthly_fig!C9)"

    Range("B9").Select

    Selection.AutoFill Destination:=Range("B9:B15")

    Range("B9:B15").Select

    ws.Range("C9").FormulaR1C1 = "=SUMIF(Comp.Monthly_fig!C[2],Summary_ContentTable!RC[-2],Comp.Monthly_fig!C[7])"

    ws.Range("C9:C15").FillDown

    ws.Range("D9").FormulaR1C1 = "=SUMIF(Payments!C[3],Summary_ContentTable!RC[-3],Payments!C)"

    ws.Range("D9:D15").FillDown


' In order to find the net profit, we computed data from product sheets to calculate cost per unit and total profit.
```

```
    Sheets("Order_details").Select
    With Sheets("Order_details")
        .Range("G1").Value = "Cost_of_product"
        .Range("H1").Value = "Total_Profit_perorder"
        .Range("I1").Value = "Product_Name"
        .Range("J1").Value = "Product_Line"
        .Range("G2").FormulaR1C1 = "=VLOOKUP(RC[-5],Products!C1:C9,8,0)"
        .Range("G2:G2997").FillDown
        .Range("H2").FormulaR1C1 = "=(RC[-4]-RC[-1])*RC[-5]"
        .Range("H2:H2997").FillDown
        .Range("I2").FormulaR1C1 = "=VLOOKUP(RC[-7],Products!C1:C3,2,0)"
        .Range("I2:I2997").FillDown
        .Range("J2").FormulaR1C1 = "=VLOOKUP(RC[-8],Products!C1:C3,3,0)"
        .Range("J2:J2997").FillDown
    End With
    Sheets("Summary_ContentTable").Select

' Determining net profit with respect to product lines
    ws.Range("A18").Formula2R1C1 = "=UNIQUE(Order_details!R[-17]C[9]:R[2979]C[9])"
    ws.Range("B18").Value = "OrderQuantity"
    ws.Range("C18").Value = "TotalPrice"
    ws.Range("D18").Value = "TotalProfit"
    ws.Range("B19").FormulaR1C1 = "=SUMIF(Order_details!C[8],Summary_ContentTable!RC[-
1],Order_details!C[1])"
    ws.Range("B19:B25").FillDown
    ws.Range("C19").FormulaR1C1 = "=SUMIF(Order_details!C[7],Summary_ContentTable!RC[-
2],Order_details!C[3])"
    ws.Range("C19:C25").FillDown
    ws.Range("D19").FormulaR1C1 = "=SUMIF(Order_details!C[6],Summary_ContentTable!RC[-
3],Order_details!C[4])"
    ws.Range("D19:D25").FillDown
```

```
' Similarly, categorizing performance of sales representatives based on their revenue contribution.
    ws.Range("A28").Formula2R1C1 = "=UNIQUE(Comp.Monthly_fig!R[-27]C[1]:R[299]C[1])"
    ws.Range("B28").Value = "OrderQuantity"
    ws.Range("C28").Value = "TotalOrdervalue"
    ws.Range("D28").Value = "CountofOrderID"
    ws.Range("B29").FormulaR1C1 =
"=SUMIFS(Comp.Monthly_fig!C9,Comp.Monthly_fig!C2,Summary_ContentTable!RC[-1])"
    ws.Range("B29:B43").FillDown
    ws.Range("C29").FormulaR1C1 =
"=SUMIFS(Comp.Monthly_fig!C10,Comp.Monthly_fig!C2,Summary_ContentTable!RC[-2])"
    ws.Range("C29:C43").FillDown
    ws.Range("D29").FormulaR1C1 = "=COUNTIF(Comp.Monthly_fig!R[-28]C[-3]:R[298]C[-
2],Summary_ContentTable!RC[-3])"
    ws.Range("D29:D43").FillDown
    ws.Range("G28").Value = "AverageQuantity"
    ws.Range("H28").FormulaR1C1 = "=AVERAGE(R[1]C[-6]:R[15]C[-6])"
    ws.Range("G29").Value = "AverageOrderValue"
    ws.Range("H29").FormulaR1C1 = "=AVERAGE(RC[-5]:R[14]C[-5])"
    ws.Range("G30").Value = "Averageordercount"
    ws.Range("H30").FormulaR1C1 = "=AVERAGE(R[-1]C[-4]:R[13]C[-4])"


' Tables values were pasted as "values" to make them accessible for creation of slicers, which is more reliant than pivot tables when working with macros.
    Columns("A:G").Select
    Selection.Copy
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False


'Creating tables by using the shortcut Ctrl + t.
    Range("A2:F5").Select
    Application.CutCopyMode = False
    ActiveSheet.ListObjects.Add(xlSrcRange, Range("$A$2:$F$5"), , xlYes).Name = _
```

```
    "Table8"
Range("A8:D15").Select
Application.CutCopyMode = False
ActiveSheet.ListObjects.Add(xlSrcRange, Range("$A$8:$D$15"), , xlYes).Name = _
    "Table9"
Range("Table9[#All]").Select
Range("A18:D25").Select
Application.CutCopyMode = False
ActiveSheet.ListObjects.Add(xlSrcRange, Range("$A$18:$D$25"), , xlYes).Name = _
    "Table10"
Range("Table10[#All]").Select
Range("A28:D43").Select
Application.CutCopyMode = False
ActiveSheet.ListObjects.Add(xlSrcRange, Range("$A$28:$D$43"), , xlYes).Name = _
    "Table11"
Range("Table11[#All]").Select


' Created a rounded rectangle shape to highlight the report name and Formatted the cells based on the different headers
Sheets("Summary").Select
ActiveSheet.Shapes.AddShape(msoShapeRoundedRectangle, 1, 2, 910.5, 40).Select
Selection.ShapeRange(1).TextFrame2.TextRange.Characters.Text = _
    "Annual Internal Report- Dashboard"
With Selection.ShapeRange(1).TextFrame2.TextRange.Characters(1, 33). _
    ParagraphFormat
    .FirstLineIndent = 0
    .Alignment = msoAlignLeft
End With
With Selection.ShapeRange(1).TextFrame2.TextRange.Characters(1, 6).Font
    .Bold = msoTrue
    .NameComplexScript = "+mn-cs"
```

```
    .NameFarEast = "+mn-ea"
    .Fill.Visible = msoTrue
    .Fill.ForeColor.ObjectThemeColor = msoThemeColorLight1
    .Fill.ForeColor.TintAndShade = 0
    .Fill.ForeColor.Brightness = 0
    .Fill.Transparency = 0
    .Fill.Solid
    .Size = 16
    .Kerning = 12
    .Name = "+mn-lt"
  End With
  With Selection.ShapeRange(1).TextFrame2.TextRange.Characters(7, 27).Font
    .BaselineOffset = 0
    .Bold = msoTrue
    .NameComplexScript = "+mn-cs"
    .NameFarEast = "+mn-ea"
    .Fill.Visible = msoTrue
    .Fill.ForeColor.ObjectThemeColor = msoThemeColorLight1
    .Fill.ForeColor.TintAndShade = 0
    .Fill.ForeColor.Brightness = 0
    .Fill.Transparency = 0
    .Fill.Solid
    .Size = 16
    .Kerning = 12
    .Name = "+mn-lt"
  End With

' Displayed the SLA shipment status and the sales report for the orders
  Range("A5").Select
  ActiveCell.FormulaR1C1 = "SLA Shipment Status"
  Columns("A:A").ColumnWidth = 17.27
```

```
Range("A6").Select
ActiveCell.FormulaR1C1 = "Timely Delivery"
Range("A7").Select
ActiveCell.FormulaR1C1 = "Late Delivery"
Range("A8").Select
ActiveCell.FormulaR1C1 = "Pending"
Range("B6").Select
ActiveCell.FormulaR1C1 = "=Summary_ContentTable!R[-2]C"
Range("B7").Select
ActiveCell.FormulaR1C1 = "=Summary_ContentTable!R[-4]C"
Range("B8").Select
ActiveCell.FormulaR1C1 = "=Summary_ContentTable!R[-3]C"
Range("D5").Select
ActiveCell.FormulaR1C1 = "Sales"
Range("D6").Select
ActiveCell.FormulaR1C1 = "=SUM(Table9[Total_Ordervalue])"
Range("E5").Select
ActiveCell.FormulaR1C1 = "Profit"
Range("E6").Select
ActiveCell.FormulaR1C1 = "=SUM(Table10[TotalProfit])"
Range("F5").Select
ActiveCell.FormulaR1C1 = "Payment Received"
Range("F6").Select
ActiveCell.FormulaR1C1 = "=SUM(Table9[Total_Payment])"
Range("A5").Select
Selection.Font.Bold = True
Range("D5:G5").Select
Selection.Font.Bold = True
Range("G5").Select
ActiveCell.FormulaR1C1 = "Quantity Sold"
Range("G6").Select
```

```
    ActiveCell.FormulaR1C1 = "=SUM(Table9[Order_Quantity])"
    Range("G7").Select
    Sheets("Summary_ContentTable").Select
    Range("Table9[#All]").Select
    Selection.Copy

' Instead of using the pivot table, using the existing data from "Summary_contentTable" and pasting into the "Summary" sheet for visualisation
    Sheets("Summary").Select
    Range("A13").Select
    ActiveSheet.Paste
    Range("Table96[#All]").Select
    Application.CutCopyMode = False

' Created slicers to represent the order details for all three years based on the different office locations
    ActiveWorkbook.SlicerCaches.Add2(ActiveSheet.ListObjects("Table96"), _
      "Office_location").Slicers.Add ActiveSheet, , "Office_location", _
      "Office_location", 83, 400, 144, 206.249763779528
    ActiveSheet.Shapes.Range(Array("Office_location")).Select
    ActiveWorkbook.SlicerCaches("Slicer_Office_location").Slicers("Office_location" _
      ).NumberOfColumns = 2
    ActiveWorkbook.SlicerCaches("Slicer_Office_location").Slicers("Office_location" _
      ).NumberOfColumns = 3
    ActiveWorkbook.SlicerCaches("Slicer_Office_location").Slicers("Office_location" _
      ).NumberOfColumns = 4
    ActiveWorkbook.SlicerCaches("Slicer_Office_location").Slicers("Office_location" _
      ).NumberOfColumns = 5
    ActiveWorkbook.SlicerCaches("Slicer_Office_location").Slicers("Office_location" _
      ).NumberOfColumns = 6
    ActiveWorkbook.SlicerCaches("Slicer_Office_location").Slicers("Office_location" _
      ).NumberOfColumns = 7
```

```
ActiveSheet.Shapes("Office_location").ScaleWidth 2.0416666667, msoFalse, _
    msoScaleFromTopLeft
ActiveSheet.Shapes("Office_location").ScaleWidth 1.4880952381, msoFalse, _
    msoScaleFromTopLeft
ActiveSheet.Shapes("Office_location").ScaleWidth 1.0605714286, msoFalse, _
    msoScaleFromTopLeft
ActiveSheet.Shapes("Office_location").ScaleHeight 0.2569699913, msoFalse, _
    msoScaleFromTopLeft
ActiveSheet.Shapes("Office_location").IncrementLeft -399
ActiveSheet.Shapes("Office_location").IncrementTop 35
Range("Table96[#All]").Select
ActiveSheet.Shapes.AddChart2(201, xlColumnClustered).Select
ActiveChart.SetSourceData Source:=Range("Summary!$A$13:$D$20")
ActiveChart.FullSeriesCollection(1).ChartType = xlColumnClustered
ActiveChart.FullSeriesCollection(1).AxisGroup = 1
ActiveChart.FullSeriesCollection(2).ChartType = xlColumnClustered
ActiveChart.FullSeriesCollection(2).AxisGroup = 1
ActiveChart.FullSeriesCollection(3).ChartType = xlLine
ActiveChart.FullSeriesCollection(3).AxisGroup = 1
ActiveChart.FullSeriesCollection(3).ChartType = xlColumnClustered
ActiveChart.FullSeriesCollection(1).ChartType = xlLine
ActiveChart.FullSeriesCollection(1).AxisGroup = 2
ActiveChart.FullSeriesCollection(2).Select
ActiveChart.SetElement (msoElementDataLabelInsideBase)
ActiveChart.FullSeriesCollection(1).ApplyDataLabels
ActiveChart.FullSeriesCollection(2).DataLabels.Select
Selection.Orientation = xlUpward
Selection.Format.TextFrame2.Orientation = msoTextOrientationUpward
With Selection.Format.TextFrame2
    .VerticalAnchor = msoAnchorMiddle
    .HorizontalAnchor = msoAnchorNone
```

```
End With
ActiveChart.FullSeriesCollection(2).Select
ActiveChart.FullSeriesCollection(3).Select
With Selection.Format.Fill
  .Visible = msoTrue
  .ForeColor.ObjectThemeColor = msoThemeColorAccent1
  .ForeColor.TintAndShade = 0
  .ForeColor.Brightness = 0.400000006
  .Transparency = 0
  .Solid
End With
ActiveChart.FullSeriesCollection(3).Points(4).Select
With Selection.Format.Fill
  .Visible = msoTrue
  .ForeColor.ObjectThemeColor = msoThemeColorAccent5
  .ForeColor.TintAndShade = 0
  .ForeColor.Brightness = 0
  .Transparency = 0
  .Solid
End With
ActiveChart.FullSeriesCollection(3).Points(2).Select
Application.CommandBars("Format Object").Visible = False
ActiveChart.FullSeriesCollection(3).Points(4).Select
ActiveChart.FullSeriesCollection(3).Points(1).Select
ActiveChart.ChartArea.Select
ActiveChart.FullSeriesCollection(3).Select
ActiveChart.SetElement (msoElementDataLabelShow)
ActiveChart.FullSeriesCollection(2).ApplyDataLabels
ActiveChart.FullSeriesCollection(3).DataLabels.Select
Selection.Orientation = xlUpward
Selection.Format.TextFrame2.Orientation = msoTextOrientationUpward
```

```
ActiveChart.SetElement (msoElementDataLabelInsideBase)
ActiveChart.ChartTitle.Select
ActiveChart.ChartTitle.Text = "Office Based Sales and Payment Report"
Selection.Format.TextFrame2.TextRange.Characters.Text = _
    "Office Based Sales and Payment Report"
With Selection.Format.TextFrame2.TextRange.Characters(1, 37).ParagraphFormat
    .TextDirection = msoTextDirectionLeftToRight
    .Alignment = msoAlignCenter
End With
With Selection.Format.TextFrame2.TextRange.Characters(1, 6).Font
    .BaselineOffset = 0
    .Bold = msoFalse
    .NameComplexScript = "+mn-cs"
    .NameFarEast = "+mn-ea"
    .Fill.Visible = msoTrue
    .Fill.ForeColor.RGB = RGB(89, 89, 89)
    .Fill.Transparency = 0
    .Fill.Solid
    .Size = 14
    .Italic = msoFalse
    .Kerning = 12
    .Name = "+mn-lt"
    .UnderlineStyle = msoNoUnderline
    .Spacing = 0
    .Strike = msoNoStrike
End With
With Selection.Format.TextFrame2.TextRange.Characters(7, 31).Font
    .BaselineOffset = 0
    .Bold = msoFalse
    .NameComplexScript = "+mn-cs"
    .NameFarEast = "+mn-ea"
```

```
    .Fill.Visible = msoTrue
    .Fill.ForeColor.RGB = RGB(89, 89, 89)
    .Fill.Transparency = 0
    .Fill.Solid
    .Size = 14
    .Italic = msoFalse
    .Kerning = 12
    .Name = "+mn-lt"
    .UnderlineStyle = msoNoUnderline
    .Spacing = 0
    .Strike = msoNoStrike
  End With
  ActiveSheet.ChartObjects("Chart 3").Activate
  Application.CommandBars("Format Object").Visible = False
  ActiveSheet.Shapes("Chart 3").ScaleWidth 1.3895833333, msoFalse, _
    msoScaleFromTopLeft
  ActiveSheet.Shapes("Chart 3").ScaleHeight 0.9872685185, msoFalse, _
    msoScaleFromTopLeft
  ActiveSheet.Shapes("Chart 3").IncrementLeft 200
  ActiveSheet.Shapes("Chart 3").IncrementTop 5.5
  Range("A26").Select
  Sheets("Summary_ContentTable").Select

' Created a table and slicers to represent the sale and profit received from orders with regard to its product lines.
  Range("Table10[#All]").Select
  Selection.Copy
  Sheets("Summary").Select
  ActiveSheet.Paste
  Range("Table107[#All]").Select
  Application.CutCopyMode = False
```

```
ActiveWorkbook.SlicerCaches.Add2(ActiveSheet.ListObjects("Table107"), _
    "Product_Line").Slicers.Add ActiveSheet, , "Product_Line", "Product_Line", 257 _
    , 400, 144, 206.249763779528
ActiveSheet.Shapes.Range(Array("Product_Line")).Select
ActiveWorkbook.SlicerCaches("Slicer_Product_Line").Slicers("Product_Line"). _
    NumberOfColumns = 2
ActiveWorkbook.SlicerCaches("Slicer_Product_Line").Slicers("Product_Line"). _
    NumberOfColumns = 3
ActiveWorkbook.SlicerCaches("Slicer_Product_Line").Slicers("Product_Line"). _
    NumberOfColumns = 4
ActiveWorkbook.SlicerCaches("Slicer_Product_Line").Slicers("Product_Line"). _
    NumberOfColumns = 5
ActiveWorkbook.SlicerCaches("Slicer_Product_Line").Slicers("Product_Line"). _
    NumberOfColumns = 6
ActiveWorkbook.SlicerCaches("Slicer_Product_Line").Slicers("Product_Line"). _
    NumberOfColumns = 7
ActiveWorkbook.SlicerCaches("Slicer_Product_Line").Slicers("Product_Line"). _
    NumberOfColumns = 8
ActiveWorkbook.SlicerCaches("Slicer_Product_Line").Slicers("Product_Line"). _
    NumberOfColumns = 7
ActiveSheet.Shapes("Product_Line").ScaleWidth 3.3541666667, msoFalse, _
    msoScaleFromTopLeft
ActiveSheet.Shapes("Product_Line").ScaleWidth 1.298136646, msoFalse, _
    msoScaleFromTopLeft
ActiveSheet.Shapes("Product_Line").ScaleHeight 0.2448487653, msoFalse, _
    msoScaleFromTopLeft
ActiveSheet.Shapes("Product_Line").IncrementLeft -398
ActiveSheet.Shapes("Product_Line").IncrementTop 47
Range("Table107[[#Headers],[Product_Line]]").Select
Range("Table107[#All]").Select
ActiveSheet.Shapes.AddChart2(297, xlBarStacked).Select
```

```
ActiveChart.SetSourceData Source:=Range("Summary!$A$26:$D$33")
ActiveSheet.Shapes("Chart 5").IncrementLeft 201
ActiveSheet.Shapes("Chart 5").IncrementTop 297
ActiveSheet.Shapes("Chart 5").ScaleWidth 1.23125, msoFalse, msoScaleFromTopLeft
ActiveChart.SetElement (msoElementDataLabelInsideBase)
ActiveChart.SetElement (msoElementDataLabelInsideBase)
ActiveChart.SetElement (msoElementDataLabelNone)
ActiveChart.ApplyDataLabels
ActiveChart.FullSeriesCollection(1).DataLabels.Select
ActiveChart.FullSeriesCollection(2).DataLabels.Select
Application.CommandBars("Format Object").Visible = False
ActiveChart.SetElement (msoElementDataLabelNone)
ActiveChart.SetElement (msoElementDataLabelNone)
Application.CutCopyMode = False
ActiveSheet.ChartObjects("Chart 5").Activate
ActiveChart.FullSeriesCollection(3).Select
ActiveChart.SetElement (msoElementDataLabelCenter)
ActiveChart.SetElement (msoElementDataLabelInsideEnd)
ActiveChart.SetElement (msoElementDataLabelInsideBase)
ActiveChart.SetElement (msoElementDataLabelCallout)
ActiveChart.SetElement (msoElementDataLabelCallout)
ActiveChart.SetElement (msoElementDataLabelInsideBase)
ActiveChart.SetElement (msoElementDataLabelCallout)
ActiveChart.SetElement (msoElementDataLabelNone)
ActiveChart.SetElement (msoElementDataLabelInsideBase)
ActiveChart.FullSeriesCollection(2).ApplyDataLabels
ActiveChart.FullSeriesCollection(3).DataLabels.Select
Selection.Position = xlLabelPositionCenter
Selection.Position = xlLabelPositionInsideEnd
Selection.Position = xlLabelPositionInsideBase
With Selection.Format.TextFrame2
```

```
    .VerticalAnchor = msoAnchorTop
    .HorizontalAnchor = msoAnchorCenter
End With
ActiveChart.ChartArea.Select
Application.CommandBars("Format Object").Visible = False
Sheets("Summary_ContentTable").Select
Range("Table11[#All]").Select
Selection.Copy
Sheets("Summary").Select
Range("A38").Select
ActiveSheet.Paste
Range("Table118[#All]").Select
ActiveSheet.Shapes.AddChart2(201, xlColumnClustered).Select
ActiveChart.SetSourceData Source:=Range("Summary!$A$38:$D$53")
ActiveChart.FullSeriesCollection(1).ChartType = xlColumnClustered
ActiveChart.FullSeriesCollection(1).AxisGroup = 1
ActiveChart.FullSeriesCollection(2).ChartType = xlColumnClustered
ActiveChart.FullSeriesCollection(2).AxisGroup = 1
ActiveChart.FullSeriesCollection(3).ChartType = xlLine
ActiveChart.FullSeriesCollection(3).AxisGroup = 1
ActiveChart.FullSeriesCollection(4).ChartType = xlLine
ActiveChart.FullSeriesCollection(4).AxisGroup = 1
ActiveChart.FullSeriesCollection(2).AxisGroup = 2
ActiveSheet.Shapes("Chart 6").IncrementLeft 200
ActiveSheet.Shapes("Chart 6").IncrementTop 557
ActiveSheet.Columns("F:F").AutoFit
Range("N26").Value = "Order value and profit"
Range("N44").Value = "Performance of sales_rep"
Range("N26,N44").Font.Bold = True

    ' A pop-up to indicate end of macro
```

```
    MsgBox "Project completed succesfully!", vbInformation, "Operation Status"


End Sub
```

“Subroutine referenced in above macro”

```
Sub ProductsSheetCleaning()

' Macro for cleaning the "Products" sheet adding the values that has been misplaced in other
cells


    Dim ws As Worksheet

    Dim lastR As Long

    Dim Filter_variables As Variant

    Dim Desc As Object

    Dim DestRow As Variant

    Dim RemDat As Range


    Set ws = ThisWorkbook.Sheets("Products")


    ' Finding the last row using rows.count property, which returns the number of rows in a
worksheet

    lastR = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row


    ' Applying filter criteria for column G and the autofilter subsequently

    Filter_variables = Array("hood opens", "hood opens and rubber wheels", _

                "Rubber wheels", "sea sprite on bow")

    ws.Range("A1:N" & lastR).AutoFilter Field:=7, Criteria1:=Filter_variables,
Operator:=xlFilterValues


    ' Generating a dictionary to store dynamic descriptions based on row numbers

    Set Desc = CreateObject("Scripting.Dictionary")
```

```
    Desc.Add 12, "1:18 scale die-cast about 10 long doors open hood opens trunk opens and wheels roll"
    Desc.Add 16, "1:12 scale die-cast about 20 long hood opens rubber wheels"
    Desc.Add 58, "1:24 scale die-cast about 18 long doors open hood opens and rubber wheels"
    Desc.Add 103, "All wood with canvas sails. Measures 31 1/2 Length x 22 3/8"" High x 8 1/4"" Width. Extras include 4 boats on deck sea sprite on bow anchors copper railing pilot houses etc."

    ' Iterate through each target row and update dynamically by assigning product descriptions to respective cells.
    For Each DestRow In Desc.Keys
        ws.Cells(DestRow, "F").Value = Desc(DestRow)

        ' Here, Select  is used dynamically allocate cell ranges to Dest row. In this case, selecting misplaced data in other columns.
        Select Case DestRow
            Case 12
                Set RemDat = ws.Range("I" & DestRow & ":K" & DestRow)
            Case 16
                Set RemDat = ws.Range("H" & DestRow & ":J" & DestRow)
            Case 58
                Set RemDat = ws.Range("H" & DestRow & ":J" & DestRow)
            Case 103
                Set RemDat = ws.Range("L" & DestRow & ":N" & DestRow)

                ' Clear J103 and K103 specifically
                ws.Cells(103, "J").ClearContents
                ws.Cells(103, "K").ClearContents
        End Select

        ' Cut the range and paste into column G
        RemDat.Cut Destination:=ws.Cells(DestRow, "G")
    Next DestRow
```

```
    ' Remove AutoFilter
    If ws.AutoFilterMode Then ws.AutoFilterMode = False

    Application.CutCopyMode = False

End Sub
```

----------END_OF_DOCUMENT----------

Note: Please run all the macros in the same order as mentioned in this document;

- **Macro 1 → Macro 2 → Macro 3 → Macro 4 → Macro 5.**
- **There is a chance of mismatch between the chart number/table number referenced in VBA macro to that of the table number generated by excel. This could be avoided by removing all data and reopening the macro after running/viewing it once.**