VBA Group - James Ponce, Yenchun Liu and Deborah Kewon

1) General structure of the project

The objective of our VBA project is to create user-friendly dashboard to better understand the web data (demo) from Google Analytics. In order to see the pattern in our dataset, we first created different types of graphs, comparing revenues with the followings: Sessions, Ave. Session Duration, Bounce Rate, Transaction and goal conversion rate.

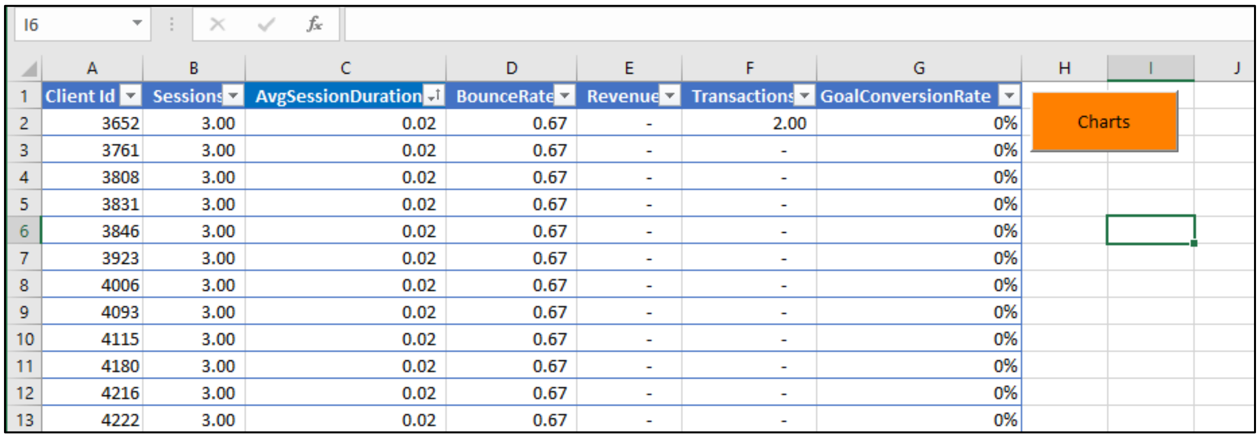
- Sessions – Number of visits
- Ave. Session Duration – Average time spent on each session
- Bounce Rate- Percentage of visiters leaving on a first page
- Transactions- Number of purchases made on the website
- Goal conversion rate- buyers/total visiters (%)

2) Comments from the other group

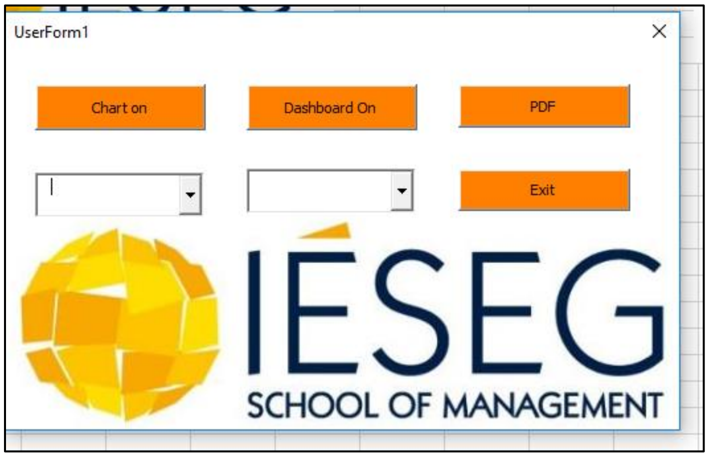
One of our classmates in the other group asked us if we can use a pivot table instead of vba as we can also easily navigate through our data and see how it works using a pivot table. Taking this into consideration, we focused more on userform layout and included features like "PDF" and "EXIT", which pivot table cannot provide. Also unlike pivot table, with our dashboard platform, users do not have to filter and sort.

3) How it works?

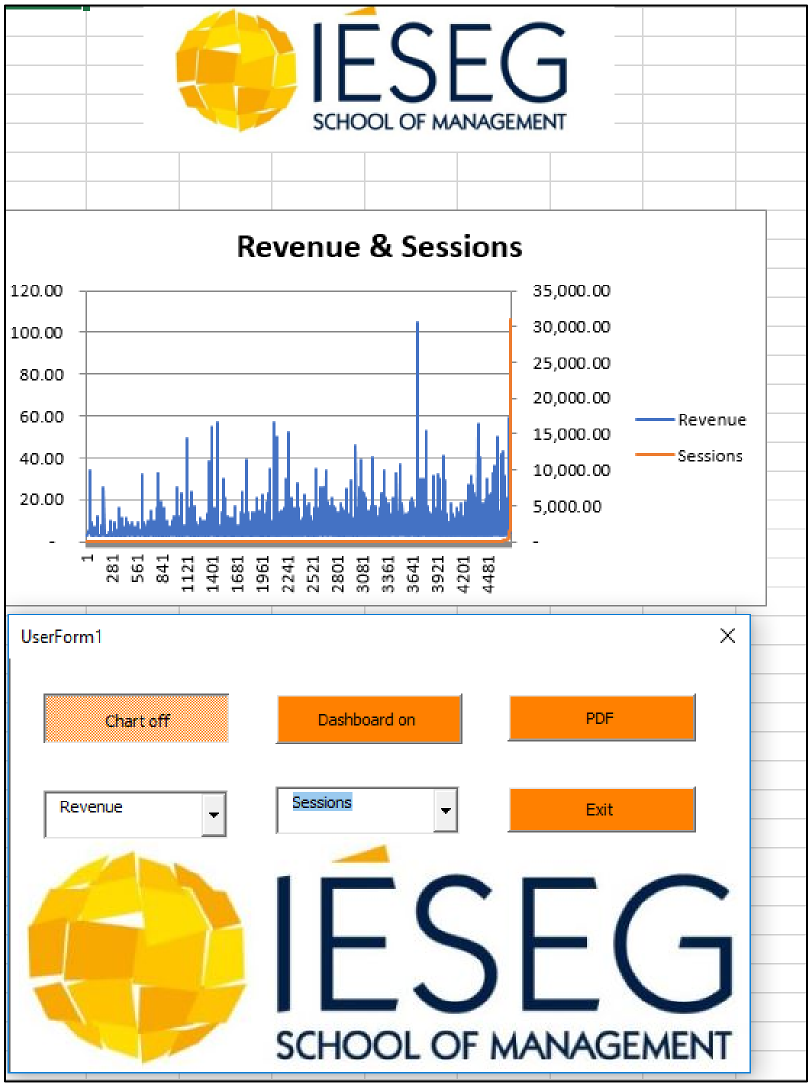
- When you click the toggle button "Charts", the Ueserform1 opens. Using Userform1, users can easily navigate through our data, compare different categories and see the details of each category.



	Client Id	Sessions	AvgSessionDuration	BounceRate	Revenue	Transactions	GoalConversionRate
2	3652	3.00	0.02	0.67	-	2.00	0%
3	3761	3.00	0.02	0.67	-	-	0%
4	3808	3.00	0.02	0.67	-	-	0%
5	3831	3.00	0.02	0.67	-	-	0%
6	3846	3.00	0.02	0.67	-	-	0%
7	3923	3.00	0.02	0.67	-	-	0%
8	4006	3.00	0.02	0.67	-	-	0%
9	4093	3.00	0.02	0.67	-	-	0%
10	4115	3.00	0.02	0.67	-	-	0%
11	4180	3.00	0.02	0.67	-	-	0%
12	4216	3.00	0.02	0.67	-	-	0%
13	4222	3.00	0.02	0.67	-	-	0%

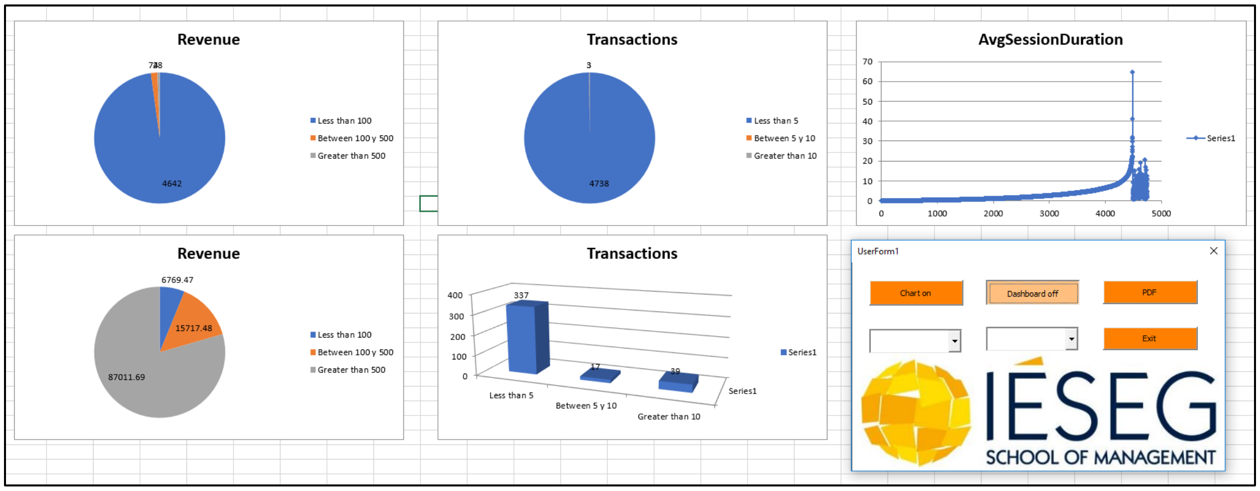


- In order to compare different categories, users need to first choose what they want to compare (ex.Revenue and Sessions) and then press "chart on". We also offer the chart off option, so they do not have to delete the graph themselves.

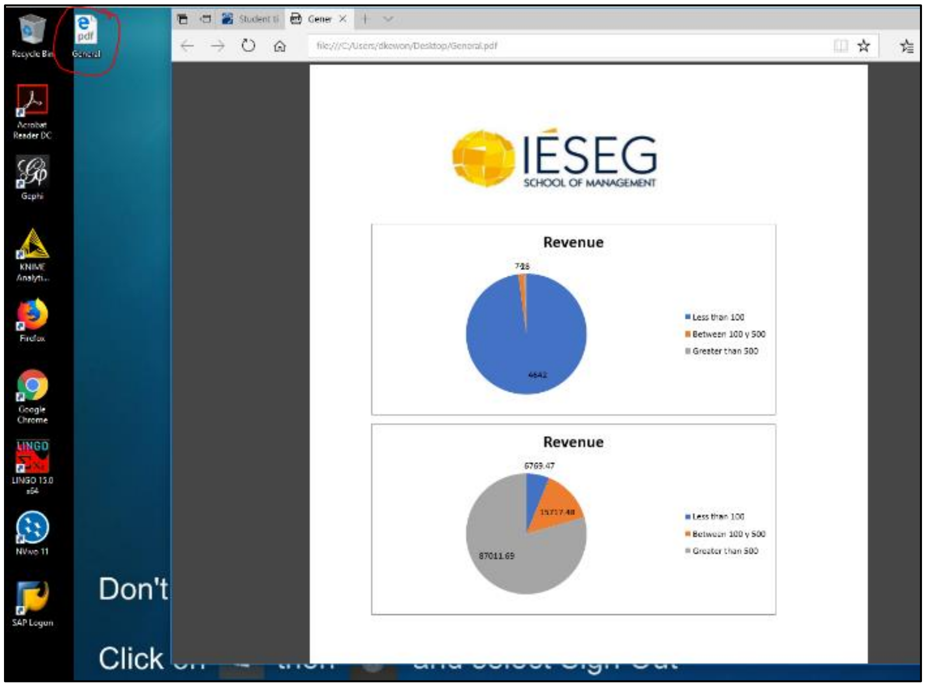


- With the “dashboard on” option, users can have a close look at each category.

For instance: The pie graphs of Revenue, the top one shows the number of users inside the intervals of spending (less than 100, between 100 and 500, more than 500); in the other hand the one in the bottom summarize the amount of money expended for users in each interval. Something similar happens in the transaction's graphs.



- When clicking “PDF” option, it automatically saves our graphs onto users’ desktop



- Exiting is also easy with the “Exit” button, it will reset all button, clean out all charts and return to dataset page

Optimization Techniques Used in the Project

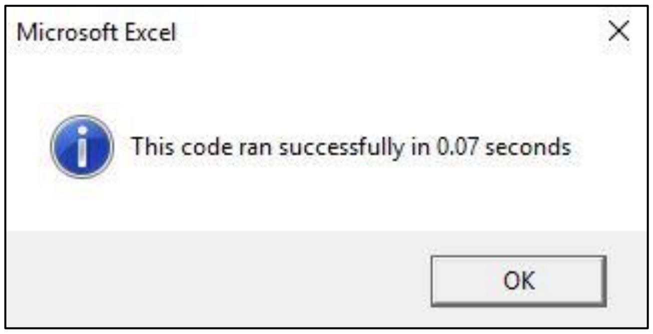
1. Measuring performance: We implement a dashboard report that run in less than 2 seconds using a two-dimensional array. The report was tested with 5000 rows.

   To perform the application we use the following code lines.

```
Dim StartTime As Double
Dim SecondsElapsed As Double

'Remember time when macro starts
  StartTime = Timer

'*****************************
'Insert Your Code Here...
'*****************************

'Determine how many seconds code took to run
  SecondsElapsed = Round(Timer - StartTime, 2)

'Notify user in seconds
  MsgBox "This code ran successfully in " & SecondsElapsed & " seconds", vbInformation
```

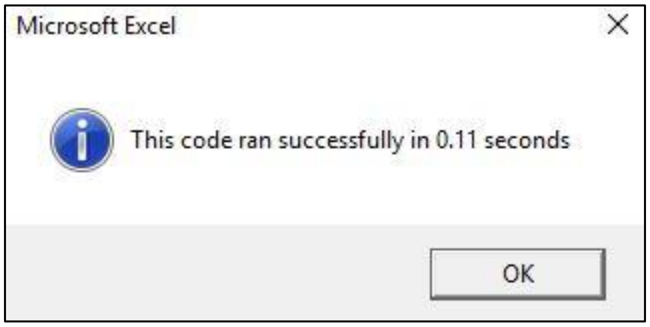
   The next screenshots show the application performance in seconds.

Dashboard time



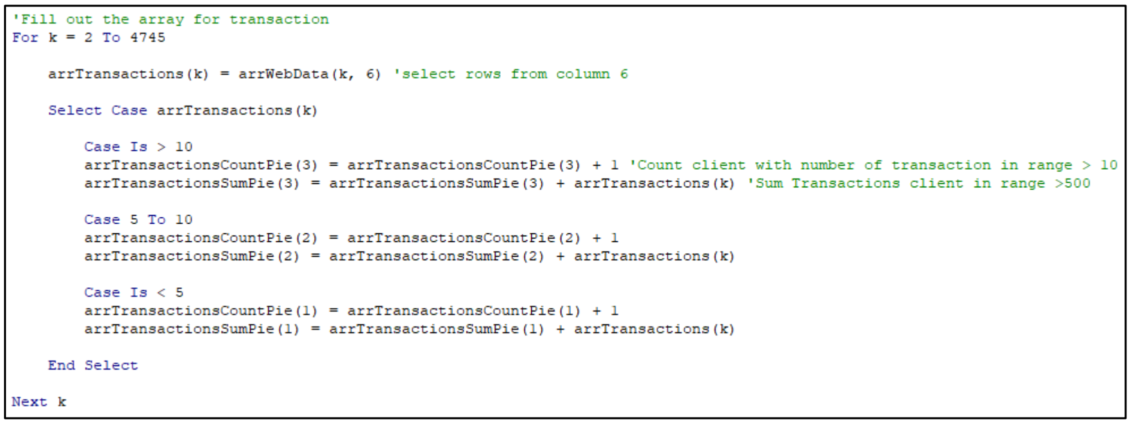
This code ran successfully in 0.07 seconds

Chart time



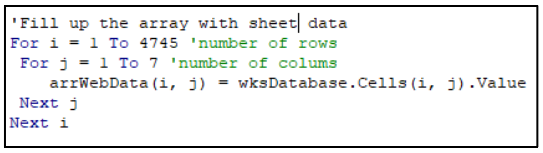
This code ran successfully in 0.11 seconds

2. **Creative thinking:** We use two cumulative variables inside for-select-case loops then put the values into a new array to be used as input data source for the pie graph (arrTransactionsCountPie, arrTransactionsSumPie).



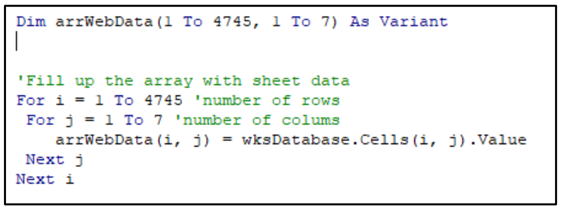
```
'Fill out the array for transaction
For k = 2 To 4745

    arrTransactions(k) = arrWebData(k, 6) 'select rows from column 6

    Select Case arrTransactions(k)

        Case Is > 10
        arrTransactionsCountPie(3) = arrTransactionsCountPie(3) + 1 'Count client with number of transaction in range > 10
        arrTransactionsSumPie(3) = arrTransactionsSumPie(3) + arrTransactions(k) 'Sum Transactions client in range >500

        Case 5 To 10
        arrTransactionsCountPie(2) = arrTransactionsCountPie(2) + 1
        arrTransactionsSumPie(2) = arrTransactionsSumPie(2) + arrTransactions(k)

        Case Is < 5
        arrTransactionsCountPie(1) = arrTransactionsCountPie(1) + 1
        arrTransactionsSumPie(1) = arrTransactionsSumPie(1) + arrTransactions(k)

    End Select

Next k
```

3. **Avoid selecting:** We consider reading and working with the data in memory using a two-dimension array instead selecting data several time from excel sheets.



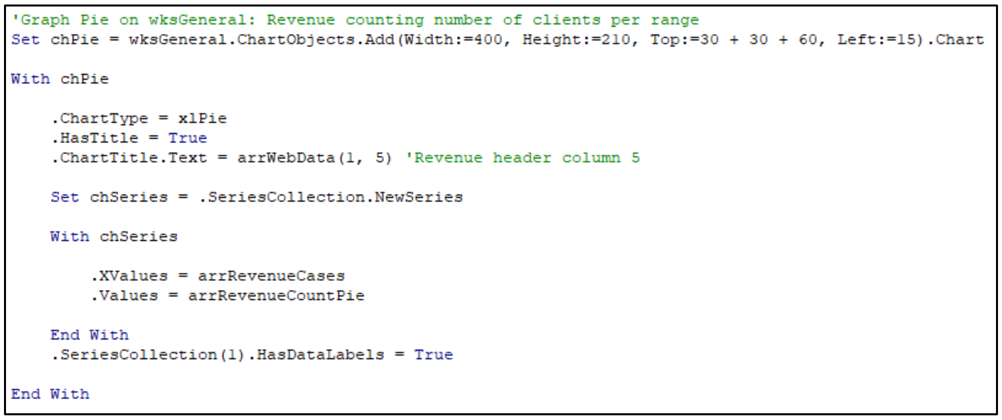
```
'Fill up the array with sheet data
For i = 1 To 4745 'number of rows
 For j = 1 To 7 'number of colums
    arrWebData(i, j) = wksDatabase.Cells(i, j).Value
 Next j
Next i
```

4. **Use Variant Arrays:** Instead of reading column by column, we decided to store in the dataset into a two-dimension Variant array.



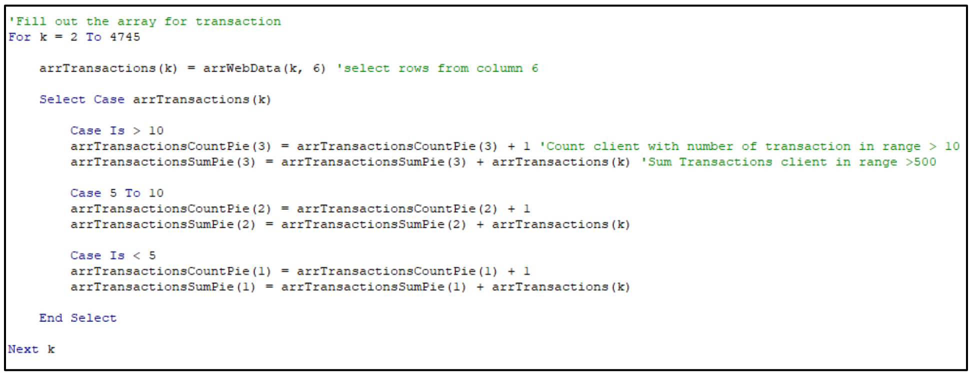
```
Dim arrWebData(1 To 4745, 1 To 7) As Variant


'Fill up the array with sheet data
For i = 1 To 4745 'number of rows
 For j = 1 To 7 'number of colums
    arrWebData(i, j) = wksDatabase.Cells(i, j).Value
 Next j
Next i
```

5. **Don't use ActiveSheet, Selection or Worksheet repeatedly:** instead of constantly using worksheets, we choose the specific name of the object and set its properties.



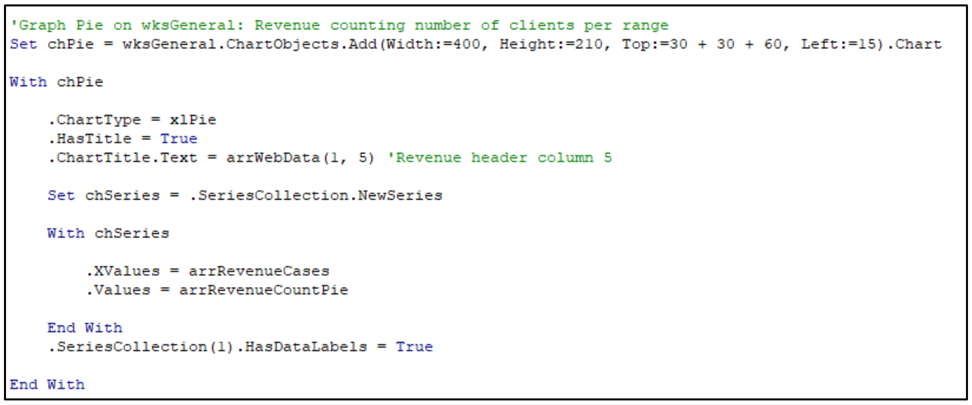
```
'Graph Pie on wksGeneral: Revenue counting number of clients per range
Set chPie = wksGeneral.ChartObjects.Add(Width:=400, Height:=210, Top:=30 + 30 + 60, Left:=15).Chart

With chPie

    .ChartType = xlPie
    .HasTitle = True
    .ChartTitle.Text = arrWebData(1, 5) 'Revenue header column 5

    Set chSeries = .SeriesCollection.NewSeries

    With chSeries

        .XValues = arrRevenueCases
        .Values = arrRevenueCountPie

    End With
    .SeriesCollection(1).HasDataLabels = True

End With
```

6. **Iterate arrays by index (Not For..Each):** We consider the use of index in arrays loops, because it could impact if we want to analyze a bigger dataset in the future.



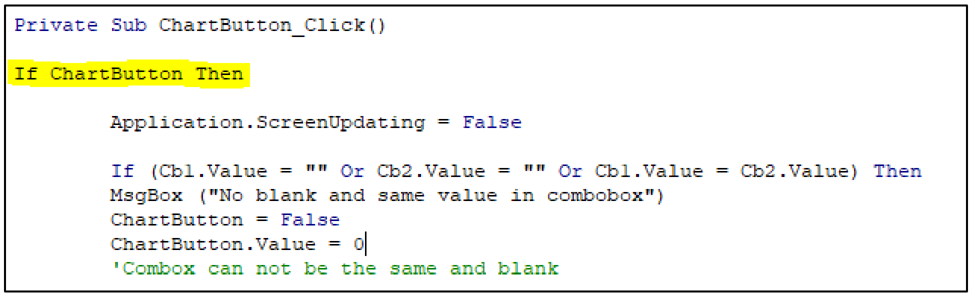
```
'Fill out the array for transaction
For k = 2 To 4745

    arrTransactions(k) = arrWebData(k, 6) 'select rows from column 6

    Select Case arrTransactions(k)

        Case Is > 10
        arrTransactionsCountPie(3) = arrTransactionsCountPie(3) + 1 'Count client with number of transaction in range > 10
        arrTransactionsSumPie(3) = arrTransactionsSumPie(3) + arrTransactions(k) 'Sum Transactions client in range >500

        Case 5 To 10
        arrTransactionsCountPie(2) = arrTransactionsCountPie(2) + 1
        arrTransactionsSumPie(2) = arrTransactionsSumPie(2) + arrTransactions(k)

        Case Is < 5
        arrTransactionsCountPie(1) = arrTransactionsCountPie(1) + 1
        arrTransactionsSumPie(1) = arrTransactionsSumPie(1) + arrTransactions(k)

    End Select

Next k
```

7. **Use With Blocks and Objects Variables to reduce dot operators:** We use With Blocks for all the graphs in the dashboard to reuse the same object several times.



```
'Graph Pie on wksGeneral: Revenue counting number of clients per range
Set chPie = wksGeneral.ChartObjects.Add(Width:=400, Height:=210, Top:=30 + 30 + 60, Left:=15).Chart

With chPie

    .ChartType = xlPie
    .HasTitle = True
    .ChartTitle.Text = arrWebData(1, 5) 'Revenue header column 5

    Set chSeries = .SeriesCollection.NewSeries

    With chSeries

        .XValues = arrRevenueCases
        .Values = arrRevenueCountPie

    End With
    .SeriesCollection(1).HasDataLabels = True

End With
```

8. **Don't Use If bVariable =True then Just use if bVariable then (If it is a boolean value)**



```
Private Sub ChartButton_Click()

If ChartButton Then

        Application.ScreenUpdating = False

        If (Cb1.Value = "" Or Cb2.Value = "" Or Cb1.Value = Cb2.Value) Then
        MsgBox ("No blank and same value in combobox")
        ChartButton = False
        ChartButton.Value = 0
        'Combox can not be the same and blank
```