Excel VBA Create Pivot Table: Step-by-Step Guide and 4 Examples to Create Pivot Tables with Macros

By Jorge A. Gomez

In this VBA Tutorial, you **learn how to create a Pivot Table** with different destinations (both worksheet or workbook) and from both static and dynamic data ranges.

This VBA Tutorial is accompanied by Excel workbooks containing the data and macros I use in the examples below. You can **get immediate free access to these example workbooks by clicking the button below**.

Get the FREE workbooks

Use the following Table of Contents to navigate to the section you're interested in.

Table of Contents [hide]

Related VBA and Macro Tutorials

The following VBA and Macro Tutorials may help you better understand and implement the contents below:

- General VBA constructs and structures:
    - Learn about commonly-used VBA terms here.
    - Learn about the Excel VBA Object Model here.
    - Learn how to work with variables here.
    - Learn about data types here.
    - Learn about the R1C1 reference-style here.
- Practical VBA applications and macro examples:
    - Learn how to create a new workbook here.

- Learn about working with worksheets here.

You can find additional VBA and Macro Tutorials in the Archives.

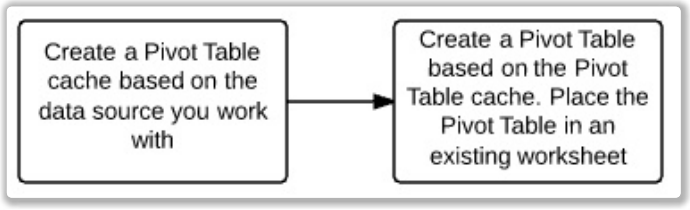
#1: Create Pivot Table in Existing Sheet

VBA Code to Create Pivot Table in Existing Sheet

To **create a Pivot Table in an existing sheet with VBA**, use a statement with the following structure:

```
1  Workbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:=SourceWorksheetName &
   "!" & SourceDataAddress).createPivotTable TableDestination:=DestinationWorksheetName
   & "!" & DestinationRangeAddress, TableName:="NewPivotTable"
```

Process Followed by VBA Code



VBA Statement Explanation

1. **Item:** Workbook.

   - **VBA Construct:** Workbook object.

   - **Description:** Represents the Excel workbook containing the source (SourceWorksheet) and destination worksheets (DestinationWorksheet) you work with. For purposes of this structure, **both the source and destination worksheet are in the same workbook**.

     Use properties such Application.Workbooks, Application.ThisWorkbook and Application.ActiveWorkbook to return this Workbook object.

2. **Item:** PivotCaches

   - **VBA Construct:** Workbook.PivotCaches method.

   - **Description:** Returns the PivotCaches collection representing all the Pivot Table caches within Workbook.

- **VBA Construct:** PivotCaches.Create method.

- **Description:** Creates a new PivotCache object representing the memory cache for the Pivot Table you create.

4. **Item:** SourceType:=xlDatabase

    - **VBA Construct:** SourceType parameter of the PivotCaches.Create method.

    - **Description:** Sets the data source of the Pivot Table you create to an Excel list or database (xlDatabase).

      Use the constants within the xlPivotTableSourceType enumeration for purposes of specifying a different data source. Nonetheless, setting SourceType to xlPivotTable (representing the same data source as another Pivot Table) or xlScenario (representing scenarios created using the Scenario Manager) generally results in a run-time error.

5. **Item:** SourceData:=SourceWorksheetName & "!" & SourceDataAddress.

    - **VBA Construct:** SourceData parameter of the PivotCaches.Create method.

    - **Description:** Specifies the data source for the Pivot Table cache.

      If you use the statement structure specified within this VBA Tutorial and explicitly declare variables to represent SourceWorksheetName and SourceDataAddress, use the String data type. Within this structure, SourceData is specified as follows:

        - **SourceWorksheetName:** Name of the worksheet containing the source data.

          If necessary, use the Worksheet.Name property to return a string representing the worksheet's name.

        - **&:** Concatenation operator.

        - **SourceDataAddress:** Address of the cell range containing the source data.

          If necessary, use the Range.Address property to return a string representing the cell range reference.

- Either **(i)** using a string to specify the worksheet and cell range (as above), or **(ii)** setting up a named range and passing the name as a string.

- Avoid passing a Range object, as this may result in unexpected “type mismatch” errors.

6. **Item:** createPivotTable

   - **VBA Construct:** PivotCache.CreatePivotTable method.

   - **Description:** Creates a Pivot Table based on the PivotCache created by the PivotCaches.Create method.

7. **Item:** TableDestination:=DestinationWorksheetName & “!” & DestinationRangeAddress

   - **VBA Construct:** TableDestination parameter of PivotCache.CreatePivotTable method.

   - **Description:** Specifies the cell in the upper-left corner of the cell range where the Pivot Table you create is located.

     If you use the statement structure specified within this VBA Tutorial and explicitly declare variables to represent DestinationWorksheetName and DestinationRangeAddress, use the String data type. Within this structure, TableDestination is specified as follows:

     - **DestinationWorksheetName:** Name of the destination worksheet where the Pivot Table you create is located.

       If necessary, use the Worksheet.Name property to return a string representing the worksheet’s name.

     - **&:** Concatenation operator.

     - **DestinationRangeAddress:** Address of the cell in the upper-left corner of the cell range where the Pivot Table you create is located.

       If necessary, use the Range.Address property to return a string representing the cell range reference.

- **VBA Construct:** TableName parameter of the PivotCache.CreatePivotTable method.

- **Description:** Specifies the name of the Pivot Table you create as “NewPivotTable”.

  If you explicitly declare a variable to represent NewPivotTable, use the String data type and omit the quotes included above (” “).

Macro Example

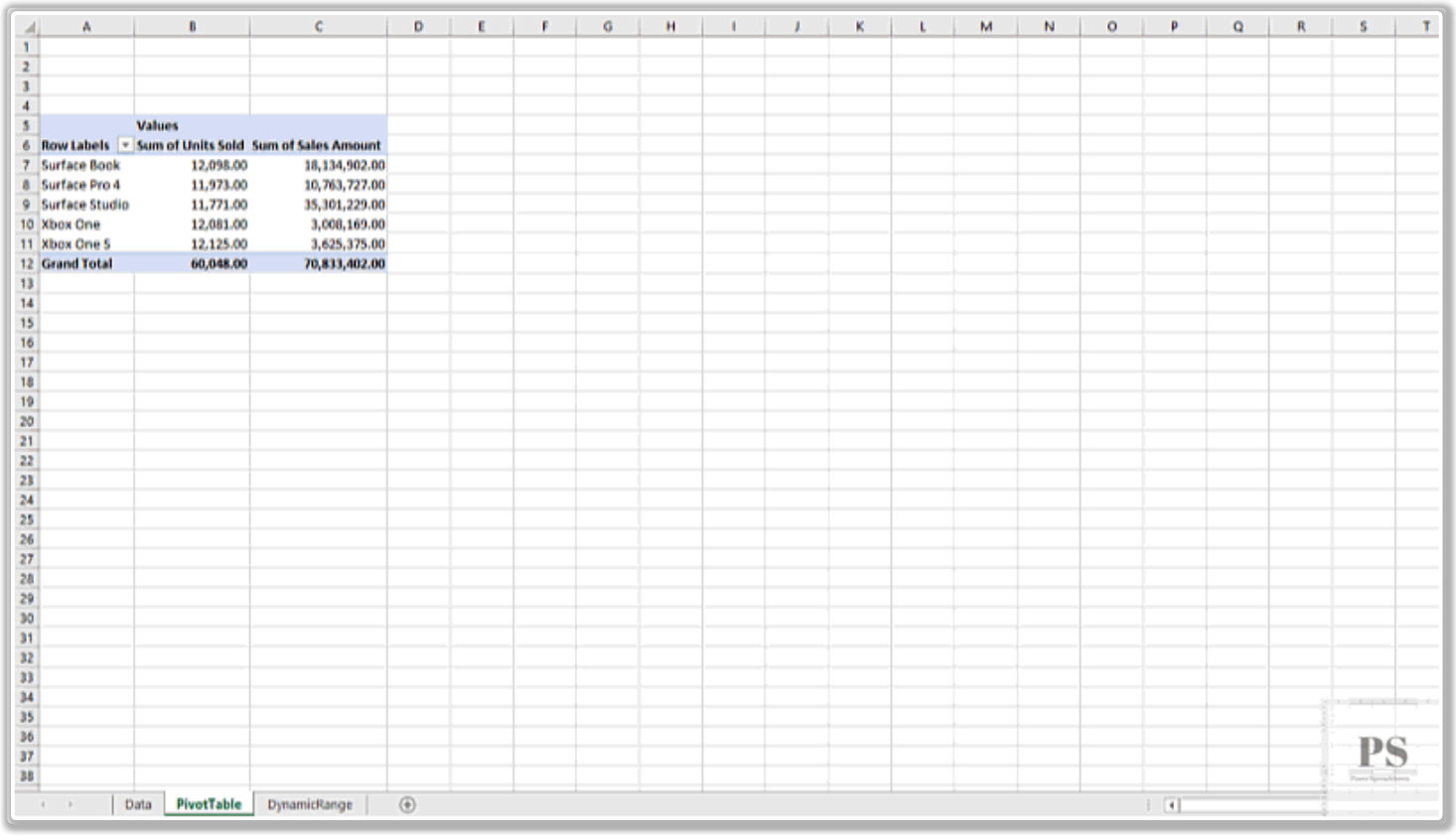
The following macro creates a new Pivot Table in an existing worksheet (PivotTable).

```
01  Sub createPivotTableExistingSheet()
02
03      'Source: https://powerspreadsheets.com/
04      'For further information: https://powerspreadsheets.com/vba-create-pivot-table/
05
06      'declare variables to hold row and column numbers that define source data cell
    range
07      Dim myFirstRow As Long
08      Dim myLastRow As Long
09      Dim myFirstColumn As Long
10      Dim myLastColumn As Long
11
12      'declare variables to hold source and destination cell range address
13      Dim mySourceData As String
14      Dim myDestinationRange As String
15
16      'declare object variables to hold references to source and destination
    worksheets, and new Pivot Table
17      Dim mySourceWorksheet As Worksheet
18      Dim myDestinationWorksheet As Worksheet
19      Dim myPivotTable As PivotTable
20
21      'identify source and destination worksheets
22      With ThisWorkbook
23          Set mySourceWorksheet = .Worksheets("Data")
24          Set myDestinationWorksheet = .Worksheets("PivotTable")
25      End With
26
27      'obtain address of destination cell range
28      myDestinationRange =
    myDestinationWorksheet.Range("A5").Address(ReferenceStyle:=xlR1C1)
29
30      'identify row and column numbers that define source data cell range
31      myFirstRow = 5
32      myLastRow = 20005
33      myFirstColumn = 1
34      myLastColumn = 6
35
36      'obtain address of source data cell range
37      With mySourceWorksheet.Cells
38          mySourceData = .Range(.Cells(myFirstRow, myFirstColumn), .Cells(myLastRow,
    myLastColumn)).Address(ReferenceStyle:=xlR1C1)
39      End With
40
```

```
myDestinationRange, TableName:="PivotTableExistingSheet")

    'add, organize and format Pivot Table fields
    With myPivotTable
        .PivotFields("Item").Orientation = xlRowField
        With .PivotFields("Units Sold")
            .Orientation = xlDataField
            .Position = 1
            .Function = xlSum
            .NumberFormat = "#,##0.00"
        End With
        With .PivotFields("Sales Amount")
            .Orientation = xlDataField
            .Position = 2
            .Function = xlSum
            .NumberFormat = "#,##0.00"
        End With
    End With

End Sub
```

Effects of Executing Macro Example

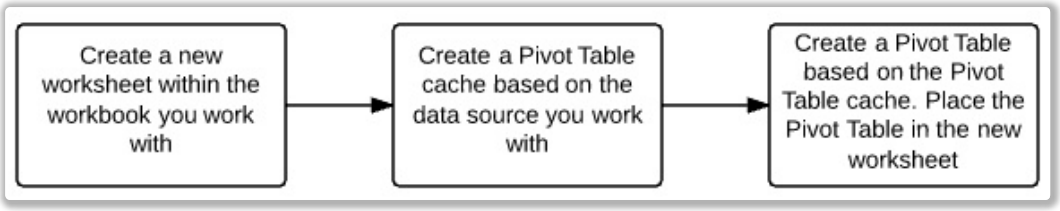
The following GIF illustrates the results of executing this macro example. As expected, the macro creates a Pivot Table in the “PivotTable” worksheet.



#2: Create Pivot Table in New Sheet

```
1  Dim DestinationWorksheet As Worksheet
2  Set DestinationWorksheet = Worksheets.Add
3  Workbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:=SourceWorksheetName &
   "!" & SourceDataAddress).createPivotTable TableDestination:=DestinationWorksheet.Name
   & "!" & DestinationRangeAddress, TableName:="NewPivotTable"
```

Process Followed by VBA Code



VBA Statement Explanation

Line #1: Dim DestinationWorksheet As Worksheet

1. **Item:** Dim DestinationWorksheet As Worksheet.

   - **VBA Construct:** Dim statement.

   - **Description:** Declares the DestinationWorksheet object variable as of the Worksheet object data type.

     DestinationWorksheet represents the new worksheet (line #2 below) where the Pivot Table you create (line #3 below) is located.

Line #2: Set DestinationWorksheet = Worksheets.Add

1. **Item:** Set.

   - **VBA Construct:** Set statement.

   - **Description:** Assigns the reference to the Worksheet object returned by the Worksheets.Add method to the DestinationWorksheet object variable.

2. **Item:** DestinationWorksheet.

   - **VBA Construct:** Object variable of the Worksheet object data type.

3. **Item:** =.

    - **VBA Construct:** Assignment operator.

    - **Description:** Assigns the reference to the Worksheet object returned by the Worksheets.Add method to the DestinationWorksheet object variable.

4. **Item:** Worksheets.

    - **VBA Construct:** Worksheets collection.

    - **Description:** The collection containing all the Worksheet objects (each representing a worksheet) within the workbook your work with.

5. **Item:** Add.

    - **VBA Construct:** Worksheets.Add method.

    - **Description:** Creates a new worksheet. This is the worksheet where the Pivot Table you create (line #3 below) is located.

Line #3: Workbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:=SourceWorksheetName & "!" & SourceDataAddress).createPivotTable TableDestination:=DestinationWorksheetName & "!" & DestinationRangeAddress, TableName:="NewPivotTable"

1. **Item:** Workbook.

    - **VBA Construct:** Workbook object.

    - **Description:** Represents the Excel workbook containing the source (SourceWorksheet) and destination worksheets (DestinationWorksheet) you work with. For purposes of this structure, **both the source and destination worksheet are in the same workbook**.

      Use properties such Application.Workbooks, Application.ThisWorkbook and Application.ActiveWorkbook to return this Workbook object.

- **VBA Construct:** Workbook.PivotCaches method.

- **Description:** Returns the PivotCaches collection representing all the Pivot Table caches within Workbook.

3. **Item:** Create.

    - **VBA Construct:** PivotCaches.Create method.

    - **Description:** Creates a new PivotCache object representing the memory cache for the Pivot Table you create.

4. **Item:** SourceType:=xlDatabase

    - **VBA Construct:** SourceType parameter of the PivotCaches.Create method.

    - **Description:** Sets the data source of the Pivot Table you create to an Excel list or database (xlDatabase).

      Use the constants within the xlPivotTableSourceType enumeration for purposes of specifying a different data source. Nonetheless, setting SourceType to xlPivotTable (representing the same data source as another Pivot Table) or xlScenario (representing scenarios created using the Scenario Manager) generally results in a run-time error.

5. **Item:** SourceData:=SourceWorksheetName & "!" & SourceDataAddress.

    - **VBA Construct:** SourceData parameter of the PivotCaches.Create method.

    - **Description:** Specifies the data source for the Pivot Table cache.

      If you use the statement structure specified within this VBA Tutorial and explicitly declare variables to represent SourceWorksheetName and SourceDataAddress, use the String data type. Within this structure, SourceData is specified as follows:

        - **SourceWorksheetName:** Name of the worksheet containing the source data.

          If necessary, use the Worksheet.Name property to return a string representing the worksheet's name.

- **SourceDataAddress:** Address of the cell range containing the source data.

  If necessary, use the Range.Address property to return a string representing the cell range reference.

SourceData is of the Variant data type. However, Microsoft's documentation recommends the following:

- Either **(i)** using a string to specify the worksheet and cell range (as above), or **(ii)** setting up a named range and passing the name as a string.
- Avoid passing a Range object, as this may result in unexpected "type mismatch" errors.

6. **Item:** createPivotTable
   - **VBA Construct:** PivotCache.CreatePivotTable method.
   - **Description:** Creates a Pivot Table based on the PivotCache created by the PivotCaches.Create method.

7. **Item:** TableDestination:=DestinationWorksheet.Name & "!" & DestinationRangeAddress
   - **VBA Construct:** TableDestination parameter of PivotCache.CreatePivotTable method.
   - **Description:** Specifies the cell in the upper-left corner of the cell range where the Pivot Table you create is located.

     If you use the statement structure specified within this VBA Tutorial and explicitly declare a variable to represent DestinationRangeAddress, use the String data type. Within this structure, TableDestination is specified as follows:

     - **DestinationWorksheet.Name:** Worksheet.Name property.

       Returns a string representing the name of DestinationWorksheet. DestinationWorksheet is the new worksheet where the Pivot Table you create is located.

     - **&:** Concatenation operator.

If necessary, use the Range.Address property to return a string representing the cell range reference.

8. **Item:** TableName:=”NewPivotTable”

   - **VBA Construct:** TableName parameter of the PivotCache.CreatePivotTable method.
   - **Description:** Specifies the name of the Pivot Table you create as “NewPivotTable”.

     If you explicitly declare a variable to represent NewPivotTable, use the String data type and omit the quotes included above (” “).

Macro Example

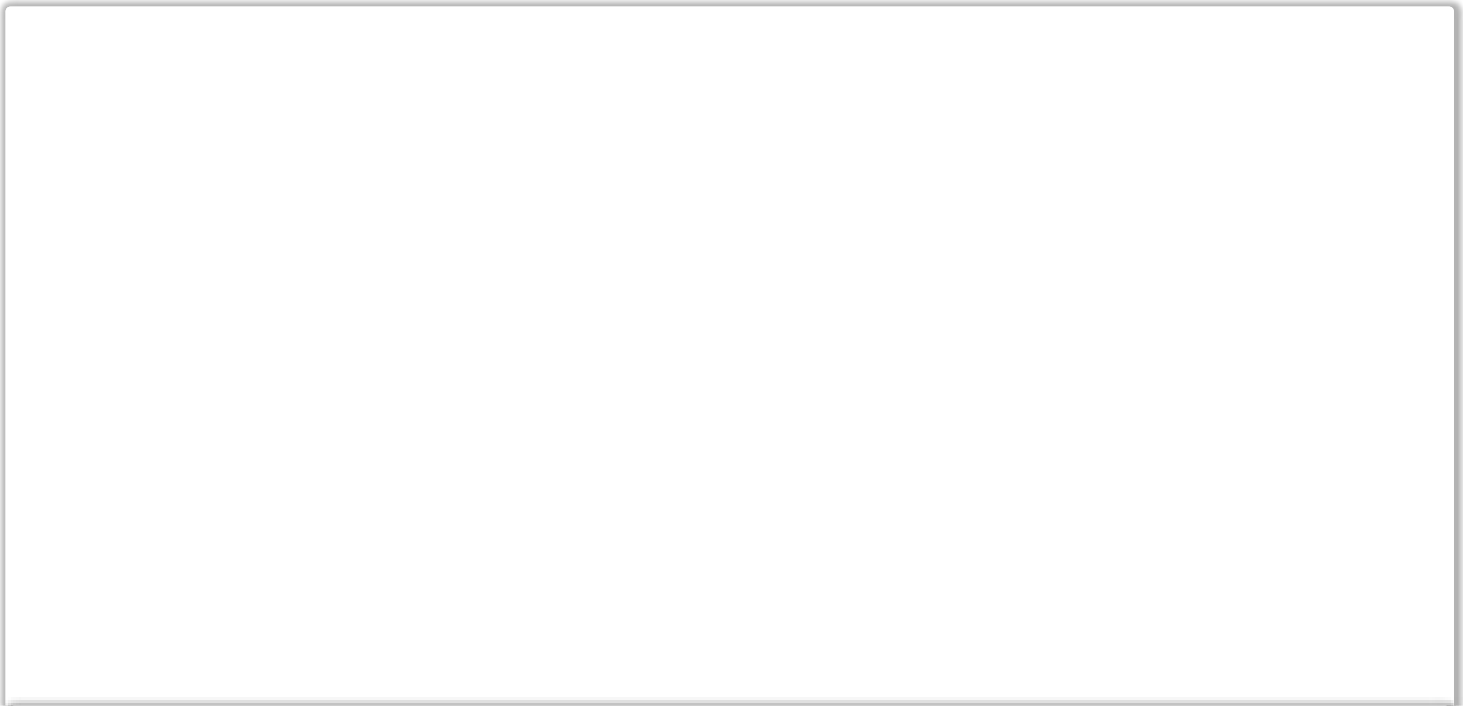
The following macro creates a new Pivot Table in a new worksheet.

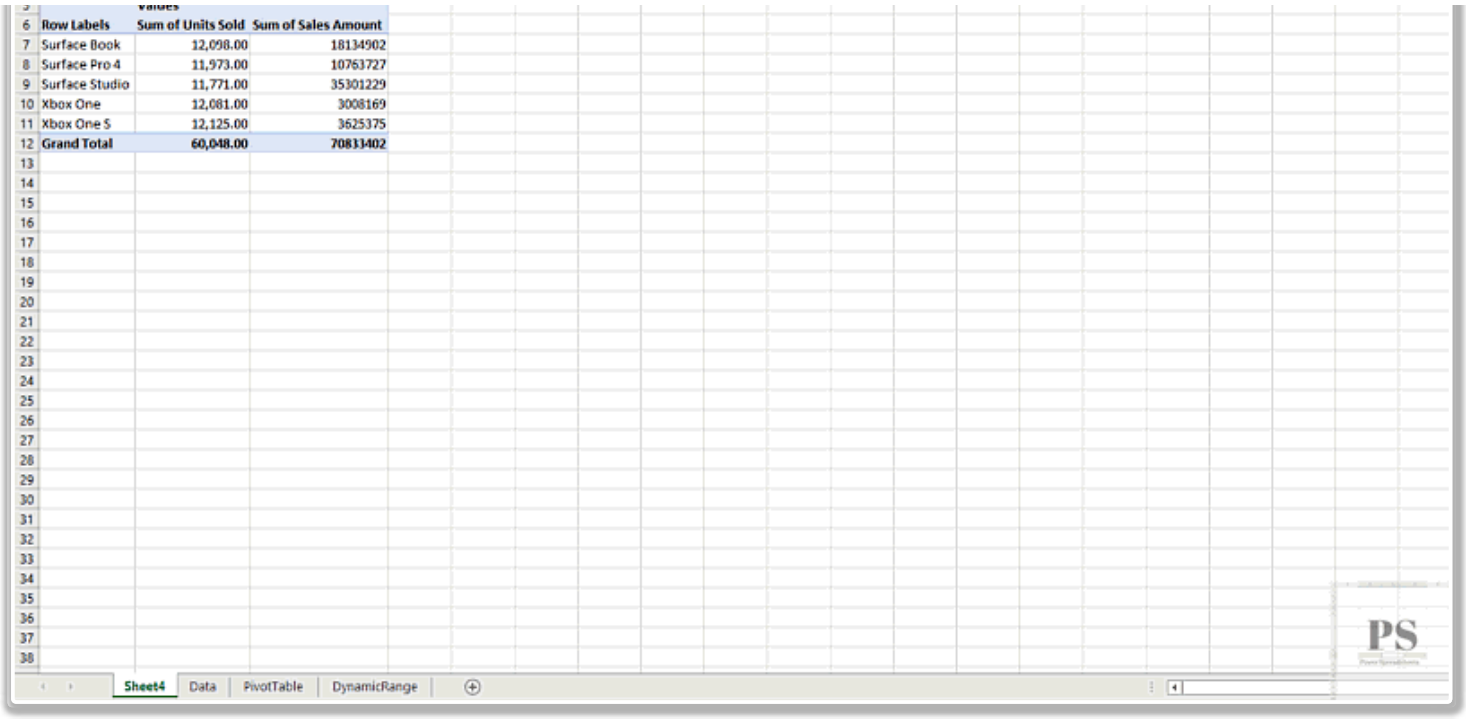
```
01  Sub createPivotTableNewSheet()
02
03      'Source: https://powerspreadsheets.com/
04      'For further information: https://powerspreadsheets.com/vba-create-pivot-table/
05
06      'declare variables to hold row and column numbers that define source data cell
    range
07      Dim myFirstRow As Long
08      Dim myLastRow As Long
09      Dim myFirstColumn As Long
10      Dim myLastColumn As Long
11
12      'declare variables to hold source and destination cell range address
13      Dim mySourceData As String
14      Dim myDestinationRange As String
15
16      'declare object variables to hold references to source and destination
    worksheets, and new Pivot Table
17      Dim mySourceWorksheet As Worksheet
18      Dim myDestinationWorksheet As Worksheet
19      Dim myPivotTable As PivotTable
20
21      'identify source and destination worksheets. Add destination worksheet
22      With ThisWorkbook
23          Set mySourceWorksheet = .Worksheets("Data")
24          Set myDestinationWorksheet = .Worksheets.Add
25      End With
26
27      'obtain address of destination cell range
28      myDestinationRange =
    myDestinationWorksheet.Range("A5").Address(ReferenceStyle:=xlR1C1)
29
30      'identify row and column numbers that define source data cell range
31      myFirstRow = 5
32      myLastRow = 20005
```

```
    With mySourceWorksheet.Cells
        mySourceData = .Range(.Cells(myFirstRow, myFirstColumn), .Cells(myLastRow, myLastColumn)).Address(ReferenceStyle:=xlR1C1)
    End With

    'create Pivot Table cache and create Pivot Table report based on that cache
    Set myPivotTable = ThisWorkbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:=mySourceWorksheet.Name & "!" & mySourceData).CreatePivotTable(TableDestination:=myDestinationWorksheet.Name & "!" & myDestinationRange, TableName:="PivotTableNewSheet")

    'add, organize and format Pivot Table fields
    With myPivotTable
        .PivotFields("Item").Orientation = xlRowField
        With .PivotFields("Units Sold")
            .Orientation = xlDataField
            .Position = 1
            .Function = xlSum
            .NumberFormat = "#,##0.00"
        End With
        With .PivotFields("Sales Amount")
            .Orientation = xlDataField
            .Position = 2
            .Function = xlSum
            .NumberFormat = "#,##0.00"
        End With
    End With

End Sub
```

Effects of Executing Macro Example

The following GIF illustrates the results of executing this macro example. As expected, the macro creates a Pivot Table in a new worksheet (Sheet4).





Row Labels	Sum of Units Sold	Sum of Sales Amount
Surface Book	12,098.00	18134902
Surface Pro 4	11,973.00	10763727
Surface Studio	11,771.00	35301229
Xbox One	12,081.00	3008169
Xbox One S	12,125.00	3625375
Grand Total	60,048.00	70831402

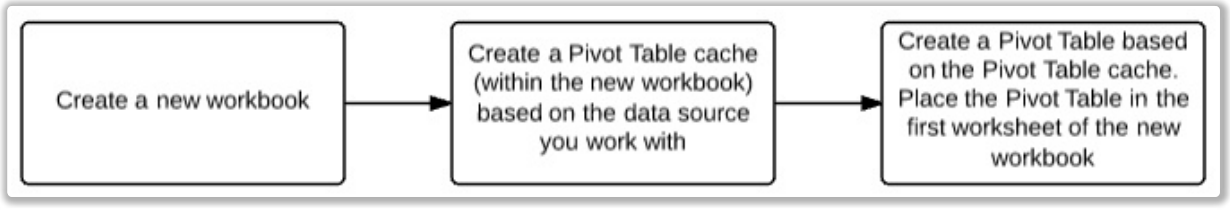
#3: Create Pivot Table in New Workbook

VBA Code to Create Pivot Table in New Workbook

To **create a Pivot Table in a new workbook with VBA**, use a macro with the following statement structure:

```
1  Dim DestinationWorkbook As Workbook
2  Set DestinationWorkbook = Workbooks.Add
3  DestinationWorkbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:="[" &
   SourceWorkbookName & "]" & SourceWorksheetName & "!" &
   SourceDataAddress).createPivotTable TableDestination:="[" & DestinationWorkbook.Name
   & "]" & DestinationWorkbook.Worksheets(1).Name & "!" & DestinationRangeAddress,
   TableName:="NewPivotTable"
```

Process Followed by VBA Code



VBA Statement Explanation

Line #1: Dim DestinationWorkbook As Workbook

1. **Item:** Dim DestinationWorkbook As Workbook.

- **Description:** Declares the DestinationWorkbook object variable as of the Workbook object data type.

  DestinationWorkbook represents the new workbook (line #2 below) where the Pivot Table you create (line #3 below) is located.

Line #2: Set DestinationWorkbook = Workbooks.Add

1. **Item:** Set.

   - **VBA Construct:** Set statement.
   - **Description:** Assigns the reference to the Workbook object returned by the Workbooks.Add method to the DestinationWorkbook object variable.

2. **Item:** DestinationWorkbook.

   - **VBA Construct:** Object variable of the Workbook object data type.
   - **Description:** Represents the new workbook where the Pivot Table you create (line #3 below) is located.

3. **Item:** =.

   - **VBA Construct:** Assignment operator.
   - **Description:** Assigns the reference to the Workbook object returned by the Workbooks.Add method to the DestinationWorkbook object variable.

4. **Item:** Workbooks.

   - **VBA Construct:** Workbooks collection.
   - **Description:** The collection containing all the Workbook objects (each representing a workbook) currently open in Excel.

5. **Item:** Add.

- **Description:** Creates a new workbook. This is the workbook where the Pivot Table you create (line #3 below) is located.

Line #3: DestinationWorkbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:="[" & SourceWorkbookName & "]" & SourceWorksheetName & "!" & SourceDataAddress).createPivotTable TableDestination:="[" & DestinationWorkbook.Name & "]" & DestinationWorkbook.Worksheets(1).Name & "!" & DestinationRangeAddress, TableName:="NewPivotTable"

1. **Item:** DestinationWorkbook.

    - **VBA Construct:** Object variable of the Workbook object data type.

    - **Description:** Represents the new workbook where the Pivot Table you create is located.

2. **Item:** PivotCaches

    - **VBA Construct:** Workbook.PivotCaches method.

    - **Description:** Returns the PivotCaches collection representing all the Pivot Table caches within DestinationWorkbook.

3. **Item:** Create.

    - **VBA Construct:** PivotCaches.Create method.

    - **Description:** Creates a new PivotCache object representing the memory cache for the Pivot Table you create.

4. **Item:** SourceType:=xlDatabase

    - **VBA Construct:** SourceType parameter of the PivotCaches.Create method.

    - **Description:** Sets the data source of the Pivot Table you create to an Excel list or database (xlDatabase).

      Use the constants within the xlPivotTableSourceType enumeration for purposes of specifying a different data source. Nonetheless, setting SourceType to xlPivotTable (representing the same data source as

5. **Item:** SourceData:=”[” & SourceWorkbookName & “]” & SourceWorksheetName & “!” & SourceDataAddress.

    - **VBA Construct:** SourceData parameter of the PivotCaches.Create method.

    - **Description:** Specifies the data source for the Pivot Table cache.

      If you use the statement structure specified within this VBA Tutorial and explicitly declare variables to represent SourceWorkbookName, SourceWorksheetName and SourceDataAddress, use the String data type. Within this structure, SourceData is specified as follows:

        - **SourceWorkbookName:** Name of the workbook containing the source data.

          If necessary, use the Workbook.Name property to return a string representing the workbook’s name.

        - **SourceWorksheetName:** Name of the worksheet containing the source data.

          If necessary, use the Worksheet.Name property to return a string representing the worksheet’s name.

        - **SourceDataAddress:** Address of the cell range containing the source data.

          If necessary, use the Range.Address property to return a string representing the cell range reference.

        - **&:** Concatenation operator.

      SourceData is of the Variant data type. However, Microsoft’s documentation recommends the following:

        - Either **(i)** using a string to specify the worksheet and cell range (as above), or **(ii)** setting up a named range and passing the name as a string.

        - Avoid passing a Range object, as this may result in unexpected “type mismatch” errors.

6. **Item:** createPivotTable

    - **VBA Construct:** PivotCache.CreatePivotTable method.

7. **Item:** TableDestination:="[" & DestinationWorkbook.Name & "]" & DestinationWorkbook.Worksheets(1).Name & "!" & DestinationRangeAddress.

    - **VBA Construct:** TableDestination parameter of PivotCache.CreatePivotTable method.

    - **Description:** Specifies the cell in the upper-left corner of the cell range where the Pivot Table you create is located.

      If you use the statement structure specified within this VBA Tutorial and explicitly declare a variable to represent DestinationRangeAddress, use the String data type. Within this structure, TableDestination is specified as follows:

        - **DestinationWorkbook.Name:** Workbook.Name property.

          Returns a string representing the name of DestinationWorkbook. DestinationWorkbook is the new workbook where the Pivot Table you create is located.

        - **DestinationWorkbook.Worksheets(1).Name:** Workbook.Worksheets property and Worksheet.Name property.

          The Workbook.Worksheets property (DestinationWorkbook.Worksheets(1)) returns a Worksheet object representing the first worksheet (Worksheets(1)) of DestinationWorkbook. The Worksheet.Name property returns a string representing the name of that worksheet, which is where the Pivot Table you create is located.

        - **DestinationRangeAddress:** Address of the cell in the upper-left corner of the cell range where the Pivot Table you create is located.

          If necessary, use the Range.Address property to return a string representing the cell range reference.

        - **&:** Concatenation operator.

8. **Item:** TableName:="NewPivotTable"

    - **VBA Construct:** TableName parameter of the PivotCache.CreatePivotTable method.

If you explicitly declare a variable to represent NewPivotTable, use the String data type and omit the quotes included above (" ").

Macro Example

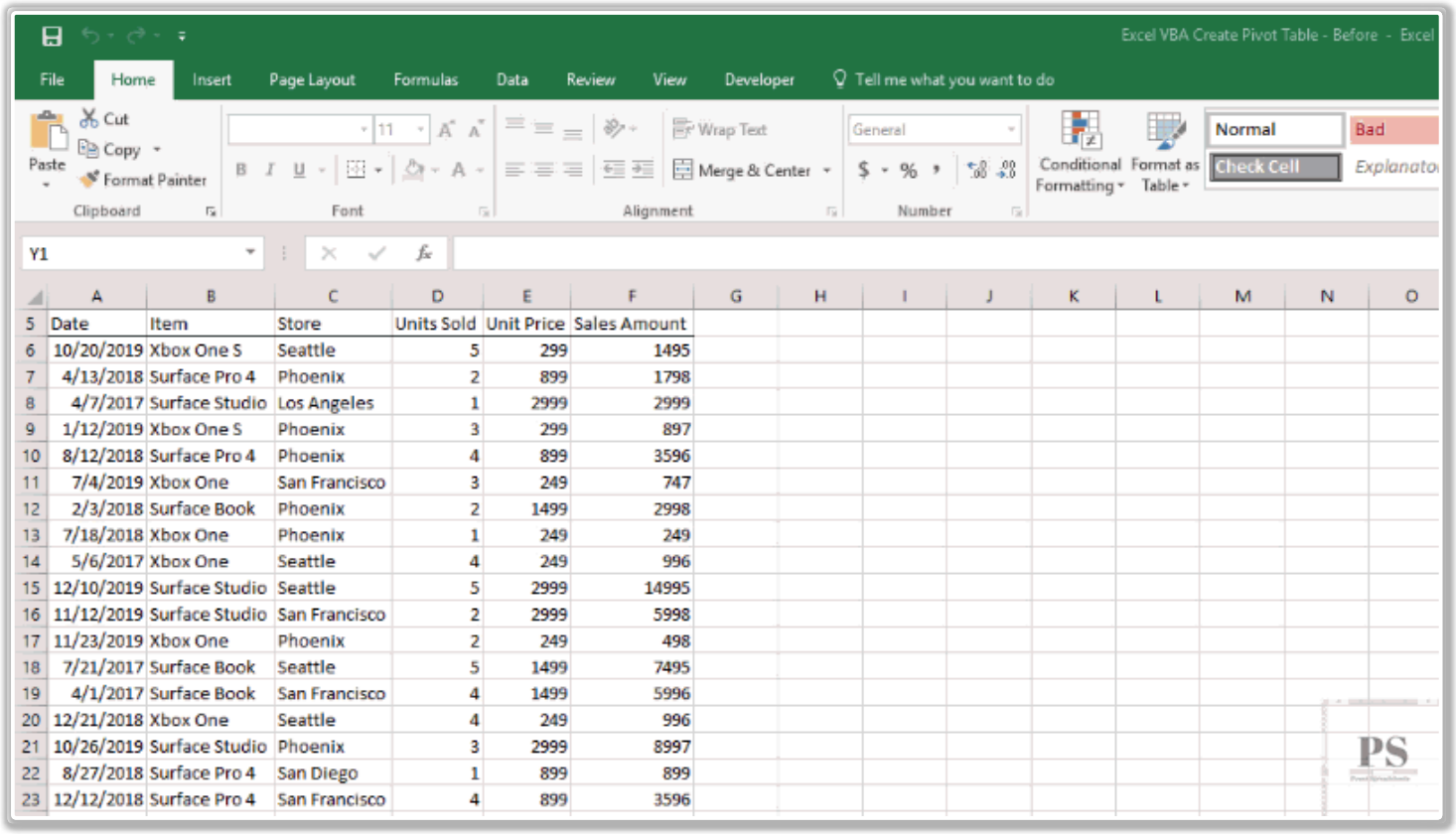
The following macro creates a new Pivot Table in a new workbook.

```
Sub createPivotTableNewWorkbook()

    'Source: https://powerspreadsheets.com/
    'For further information: https://powerspreadsheets.com/vba-create-pivot-table/

    'declare variables to hold row and column numbers that define source data cell range
    Dim myFirstRow As Long
    Dim myLastRow As Long
    Dim myFirstColumn As Long
    Dim myLastColumn As Long

    'declare variables to hold source and destination cell range address
    Dim mySourceData As String
    Dim myDestinationRange As String

    'declare object variables to hold references to destination workbook, source and destination worksheets, and new Pivot Table
    Dim myDestinationWorkbook As Workbook
    Dim mySourceWorksheet As Worksheet
    Dim myDestinationWorksheet As Worksheet
    Dim myPivotTable As PivotTable

    'add and identify destination worksheet
    Set myDestinationWorkbook = Workbooks.Add

    'identify source and destination worksheets
    Set mySourceWorksheet = ThisWorkbook.Worksheets("Data")
    Set myDestinationWorksheet = myDestinationWorkbook.Worksheets(1)

    'obtain address of destination cell range
    myDestinationRange = myDestinationWorksheet.Range("A5").Address(ReferenceStyle:=xlR1C1)

    'identify row and column numbers that define source data cell range
    myFirstRow = 5
    myLastRow = 20005
    myFirstColumn = 1
    myLastColumn = 6

    'obtain address of source data cell range
    With mySourceWorksheet.Cells
        mySourceData = .Range(.Cells(myFirstRow, myFirstColumn), .Cells(myLastRow, myLastColumn)).Address(ReferenceStyle:=xlR1C1)
    End With

    'create Pivot Table cache and create Pivot Table report based on that cache
    Set myPivotTable = myDestinationWorkbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:="[" & ThisWorkbook.Name & "]" & mySourceWorksheet.Name & "!" &
```

```
    'add, organize and format Pivot Table fields
    With myPivotTable
        .PivotFields("Item").Orientation = xlRowField
        With .PivotFields("Units Sold")
            .Orientation = xlDataField
            .Position = 1
            .Function = xlSum
            .NumberFormat = "#,##0.00"
        End With
        With .PivotFields("Sales Amount")
            .Orientation = xlDataField
            .Position = 2
            .Function = xlSum
            .NumberFormat = "#,##0.00"
        End With
    End With

End Sub
```

Effects of Executing Macro Example

The following GIF illustrates the results of executing this macro example. As expected, the macro creates a Pivot Table in a new workbook.



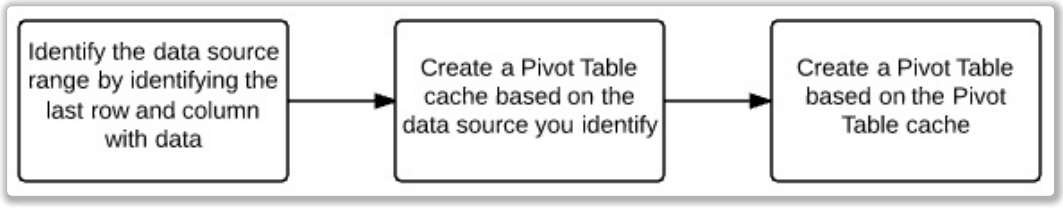
	A	B	C	D	E	F
5	Date	Item	Store	Units Sold	Unit Price	Sales Amount
6	10/20/2019	Xbox One S	Seattle	5	299	1495
7	4/13/2018	Surface Pro 4	Phoenix	2	899	1798
8	4/7/2017	Surface Studio	Los Angeles	1	2999	2999
9	1/12/2019	Xbox One S	Phoenix	3	299	897
10	8/12/2018	Surface Pro 4	Phoenix	4	899	3596
11	7/4/2019	Xbox One	San Francisco	3	249	747
12	2/3/2018	Surface Book	Phoenix	2	1499	2998
13	7/18/2018	Xbox One	Phoenix	1	249	249
14	5/6/2017	Xbox One	Seattle	4	249	996
15	12/10/2019	Surface Studio	Seattle	5	2999	14995
16	11/12/2019	Surface Studio	San Francisco	2	2999	5998
17	11/23/2019	Xbox One	Phoenix	2	249	498
18	7/21/2017	Surface Book	Seattle	5	1499	7495
19	4/1/2017	Surface Book	San Francisco	4	1499	5996
20	12/21/2018	Xbox One	Seattle	4	249	996
21	10/26/2019	Surface Studio	Phoenix	3	2999	8997
22	8/27/2018	Surface Pro 4	San Diego	1	899	899
23	12/12/2018	Surface Pro 4	San Francisco	4	899	3596

#4: Create Pivot Table from Dynamic Range

VBA Code to Create Pivot Table from Dynamic Range

```
1  Dim LastRow As Long
2  Dim LastColumn As Long
3  Dim SourceDataAddress As String
4  With SourceWorksheet.Cells
5  LastRow = .Find(What:="*", LookIn:=xlFormulas, LookAt:=xlPart, SearchOrder:=xlByRows,
   SearchDirection:=xlPrevious).Row
6  LastColumn = .Find(What:="*", LookIn:=xlFormulas, LookAt:=xlPart,
   SearchOrder:=xlByColumns, SearchDirection:=xlPrevious).Column
7  SourceDataAddress = .Range(.Cells(1, 1), .Cells(LastRow,
   LastColumn)).Address(ReferenceStyle:=xlR1C1)
8  End With
9  Workbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:=SourceWorksheetName &
   "!" & SourceDataAddress).createPivotTable TableDestination:=DestinationWorksheetName
   & "!" & DestinationRangeAddress, TableName:="NewPivotTable"
```

Process Followed by VBA Code



VBA Statement Explanation

Line #1: Dim LastRow As Long

1. **Item:** Dim LastRow As Long.

    - **VBA Construct:** Dim statement.

    - **Description:** Declares the LastRow variable as of the Long data type.

      LastRow holds the number of the last row with data in the worksheet containing the source data (SourceWorksheet).

Line #2: Dim LastColumn As Long

1. **Item:** Dim LastColumn As Long.

    - **VBA Construct:** Dim statement.

    - **Description:** Declares the LastColumn variable as of the Long data type.

      LastColumn holds the number of the last column with data in the worksheet containing the source data (SourceWorksheet).

- **VBA Construct:** Dim statement.
- **Description:** Declares the SourceDataAddress variable as of the String data type.

  SourceDataAddress holds the address of the cell range containing the source data.

Lines #4 and #8: With SourceWorksheet.Cells | End With

1. **Item:** With… End With.
   - **VBA Construct:** With… End With statement.
   - **Description:** Statements within the With… End With statement (lines #5 through #7 below) are executed on the Range object returned by SourceWorksheet.Cells.

2. **Item:** SourceWorksheet.
   - **VBA Construct:** Worksheet object.
   - **Description:** Represents the worksheet containing the source data. If you explicitly declare an object variable to represent SourceWorksheet, use the Worksheet object data type.

3. **Item:** Cells.
   - **VBA Construct:** Worksheets.Cells property.
   - **Description:** Returns a Range object representing all the cells in SourceWorksheet.

Line #5: LastRow = .Find(What:="*", LookIn:=xlFormulas, LookAt:=xlPart, SearchOrder:=xlByRows, SearchDirection:=xlPrevious).Row

1. **Item:** LastRow.
   - **VBA Construct:** Variable of the long data type.
   - **Description:** LastRow holds the number of the last row with data in SourceWorksheet.

- **VBA Construct:** Assignment operator.

- **Description:** Assigns the row number returned by the Range.Row property to the LastRow variable.

3. **Item:** Find.

    - **VBA Construct:** Range.Find method.

    - **Description:** Returns a Range object representing the first cell where the information specified by the parameters of the Range.Find method (What, LookIn, LookAt, SearchOrder and SearchDirection) is found. Within this macro structure, this Range object represents the last cell with data in the last row with data in SourceWorksheet.

4. **Item:** What:=”*”.

    - **VBA Construct:** What parameter of the Range.Find method.

    - **Description:** Specifies the data the Range.Find method searches for. The asterisk (*) is a wildcard and, therefore, the Range.Find method searches for any character sequence.

5. **Item:** LookIn:=xlFormulas.

    - **VBA Construct:** LookIn parameter of the Range.Find method.

    - **Description:** Specifies that the Range.Find method looks in formulas (xlFormulas).

6. **Item:** LookAt:=xlPart.

    - **VBA Construct:** LookAt parameter of the Range.Find method.

    - **Description:** Specifies that the Range.Find method looks at (and matches) a part (xlPart) of the search data.

- **VBA Construct:** SearchOrder parameter of the Range.Find method.

- **Description:** Specifies that the Range.Find method searches by rows (xlByRows).

8. **Item:** SearchDirection:=xlPrevious.

    - **VBA Construct:** SearchDirection parameter of the Range.Find method.

    - **Description:** Specifies that the Range.Find method searches for the previous match (xlPrevious).

9. **Item:** Row.

    - **VBA Construct:** Range.Row property.

    - **Description:** Returns the row number of the Range object returned by the Range.Find method. Within this macro structure, this row number corresponds to the last row with data in SourceWorksheet.

Line #6: LastColumn = .Find(What:="*", LookIn:=xlFormulas, LookAt:=xlPart, SearchOrder:=xlByColumns, SearchDirection:=xlPrevious).Column

1. **Item:** LastColumn.

    - **VBA Construct:** Variable of the long data type.

    - **Description:** Variable of the long data type.

      LastColumn holds the number of the last column with data in SourceWorksheet.

2. **Item:** =.

    - **VBA Construct:** Assignment operator.

    - **Description:** Assigns the column number returned by the Range.Column property to the LastColumn variable.

- **VBA Construct:** Range.Find method.

- **Description:** Returns a Range object representing the first cell where the information specified by the parameters of the Range.Find method (What, LookIn, LookAt, SearchOrder and SearchDirection) is found. Within this macro structure, this Range object represents the last cell with data in the last column with data in SourceWorksheet.

4. **Item:** What:=”*”.

    - **VBA Construct:** What parameter of the Range.Find method.

    - **Description:** Specifies the data the Range.Find method searches for. The asterisk (*) is a wildcard and, therefore, the Range.Find method searches for any character sequence.

5. **Item:** LookIn:=xlFormulas.

    - **VBA Construct:** LookIn parameter of the Range.Find method.

    - **Description:** Specifies that the Range.Find method looks in formulas (xlFormulas).

6. **Item:** LookAt:=xlPart.

    - **VBA Construct:** LookAt parameter of the Range.Find method.

    - **Description:** Specifies that the Range.Find method looks at (and matches) a part (xlPart) of the search data.

7. **Item:** SearchOrder:=xlByColumns.

    - **VBA Construct:** SearchOrder parameter of the Range.Find method.

    - **Description:** Specifies that the Range.Find method searches by columns (xlByColumns).

- **VBA Construct:** SearchDirection parameter of the Range.Find method.

- **Description:** Specifies that the Range.Find method searches for the previous match (xlPrevious).

9. **Item:** Column.

    - **VBA Construct:** Range.Column property.

    - **Description:** Returns the column number of the Range object returned by the Range.Find method. Within this macro structure, this column number corresponds to the last column with data in SourceWorksheet.

Line #7: SourceDataAddress = .Range(.Cells(1, 1), .Cells(LastRow, LastColumn)).Address(ReferenceStyle:=xlR1C1)

1. **Item:** SourceDataAddress.

    - **VBA Construct:** Variable of the String data type.

    - **Description:** SourceDataAddress holds the address of the cell range containing the source data.

2. **Item:** =.

    - **VBA Construct:** Assignment operator.

    - **Description:** Assigns the string returned by the Range.Address property to the SourceDataAddress variable.

3. **Item:** Range.

    - **VBA Construct:** Range.Range property.

    - **Description:** Returns a Range object representing the cell range containing the source data. Within this macro structure, the Range property is applied to the Range object returned by the Worksheet.Cells property in the opening statement of the With… End With statement (line #4 above).

- **VBA Construct:** Cells1 parameter of the Range.Range property, Range.Cells property and Range.Item property.

- **Description:** The Cells1 parameter of the Range.Range property specifies the cell in the upper-left corner of the cell range. Within this macro structure, Cells1 is the Range object returned by the Range.Cells property.

  The Range.Cells property returns all the cells within the cell range represented by the Range object returned by the Worksheet.Cells property in the opening statement of the With… End With statement (line #4 above). The Range.Item property is the default property and returns a Range object representing the cell on the first row and first column (Cells(1, 1)) of the cell range it works with.

  Since the Worksheet.Cells property in line #4 above returns all the cells in SourceWorksheet, this is cell A1 of SourceWorksheet.

5. **Item:** Cells(LastRow, LastColumn).

   - **VBA Construct:** Cells2 parameter of the Range.Range property, Range.Cells property and Range.Item property.

   - **Description:** The Cells2 parameter of the Range.Range property specifies the cells in the lower-right corner of the cell range. Within this macro structure, Cells2 is the Range object returned by the Range.Cells property.

     The Range.Cells property returns all the cells within the cell range represented by the Range object returned by the Worksheet.Cells property in the opening statement of the With… End With statement (line #4 above). The Range.Item property is the default property and returns a Range object representing the cell located at the intersection of LastRow and LastColumn.

     Since the Worksheet.Cells property in line #4 above returns all the cells in SourceWorksheet, this is the cell located at the intersection of the last row and the last column (or the last cell with data) within SourceWorksheet.

6. **Item:** Address.

   - **VBA Construct:** Range.Address property.

7. **Item:** ReferenceStyle:=xlR1C1.

    - **VBA Construct:** ReferenceStyle parameter of the Range.Address property.

    - **Description:** Specifies that the cell range reference returned by the Range.Address property is in the R1C1 reference style.

Line #9: Workbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:=SourceWorksheetName & "!" & SourceDataAddress).createPivotTable TableDestination:=DestinationWorksheetName & "!" & DestinationRangeAddress, TableName:="NewPivotTable"[=]

1. **Item:** Workbook.

    - **VBA Construct:** Workbook object.

    - **Description:** Represents the Excel workbook containing the source (SourceWorksheet) and destination worksheets (DestinationWorksheet) you work with. For purposes of this structure, **both the source and destination worksheet are in the same workbook**.

      Use properties such Application.Workbooks, Application.ThisWorkbook and Application.ActiveWorkbook to return this Workbook object.

2. **Item:** PivotCaches

    - **VBA Construct:** Workbook.PivotCaches method.

    - **Description:** Returns the PivotCaches collection representing all the Pivot Table caches within Workbook.

3. **Item:** Create.

    - **VBA Construct:** PivotCaches.Create method.

    - **Description:** Creates a new PivotCache object representing the memory cache for the Pivot Table you create.

- **VBA Construct:** SourceType parameter of the PivotCaches.Create method.

- **Description:** Sets the data source of the Pivot Table you create to an Excel list or database (xlDatabase).

  Use the constants within the xlPivotTableSourceType enumeration for purposes of specifying a different data source. Nonetheless, setting SourceType to xlPivotTable (representing the same data source as another Pivot Table) or xlScenario (representing scenarios created using the Scenario Manager) generally results in a run-time error.

5. **Item:** SourceData:=SourceWorksheetName & "!" & SourceDataAddress.

   - **VBA Construct:** SourceData parameter of the PivotCaches.Create method.

   - **Description:** Specifies the data source for the Pivot Table cache.

     If you use the statement structure specified within this VBA Tutorial and explicitly declare variables to represent SourceWorksheetName and SourceDataAddress, use the String data type. Within this structure, SourceData is specified as follows:

     - **SourceWorksheetName:** Name of the worksheet containing the source data.

       If necessary, use the Worksheet.Name property to return a string representing the worksheet's name.

     - **&:** Concatenation operator.

     - **SourceDataAddress:** Variable of the String data type.

       SourceDataAddress holds the address of the cell range containing the source data.

6. **Item:** createPivotTable

   - **VBA Construct:** PivotCache.CreatePivotTable method.

   - **Description:** Creates a Pivot Table based on the PivotCache created by the PivotCaches.Create method.

- **VBA Construct:** TableDestination parameter of PivotCache.CreatePivotTable method.

- **Description:** Specifies the cell in the upper-left corner of the cell range where the Pivot Table you create is located.

  If you use the statement structure specified within this VBA Tutorial and explicitly declare variables to represent DestinationWorksheetName and DestinationRangeAddress, use the String data type. Within this structure, TableDestination is specified as follows:

  - **DestinationWorksheetName:** Name of the destination worksheet where the Pivot Table you create is located.

    If necessary, use the Worksheet.Name property to return a string representing the worksheet's name.

  - **&:** Concatenation operator.

  - **DestinationRangeAddress:** Address of the cell in the upper-left corner of the cell range where the Pivot Table you create is located.

    If necessary, use the Range.Address property to return a string representing the cell range reference.

8. **Item:** TableName:="NewPivotTable"

   - **VBA Construct:** TableName parameter of the PivotCache.CreatePivotTable method.

   - **Description:** Specifies the name of the Pivot Table you create as "NewPivotTable".

     If you explicitly declare a variable to represent NewPivotTable, use the String data type and omit the quotes included above (" ").

Macro Example

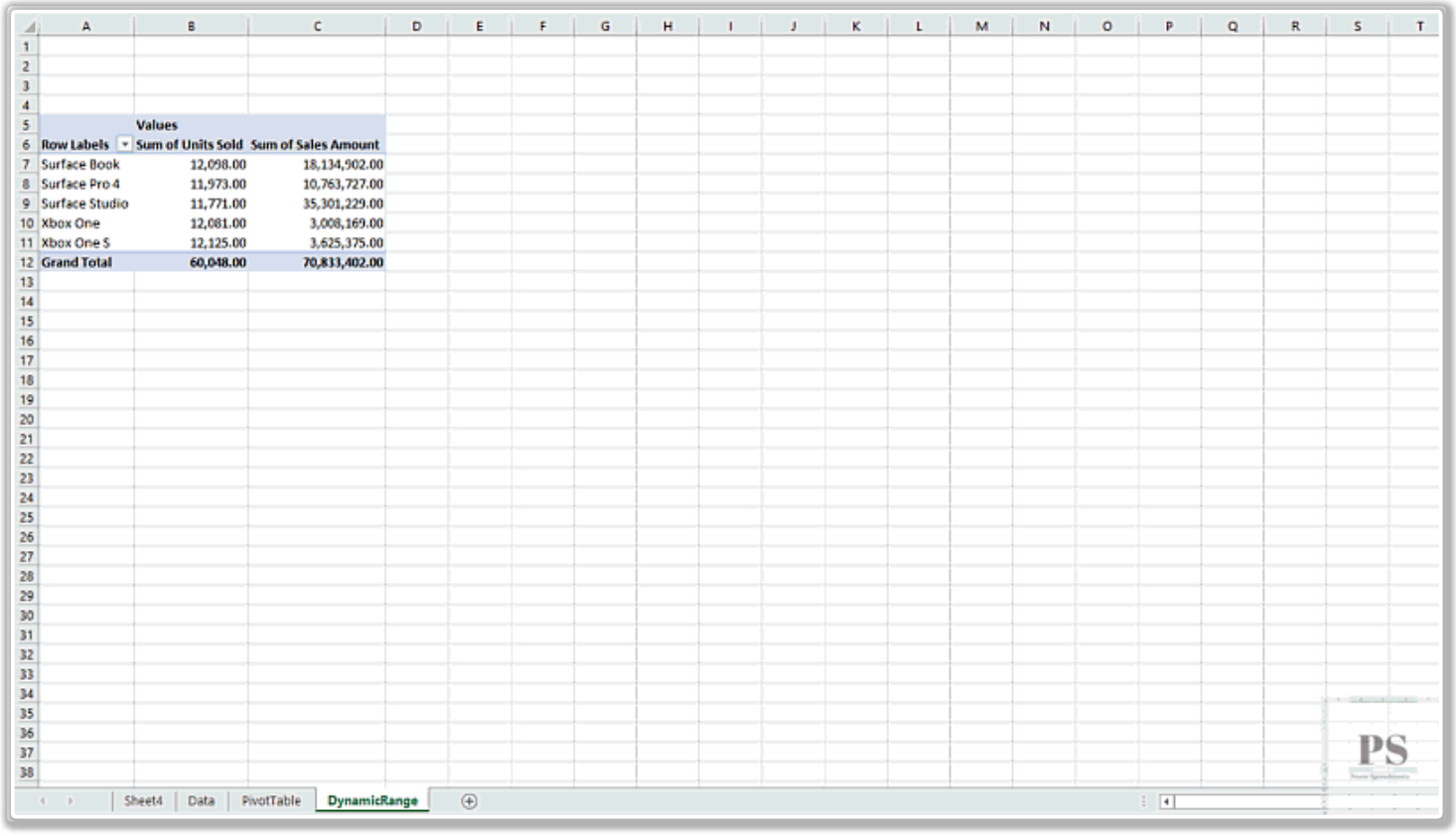
The macro below creates a new Pivot Table from a dynamic range, where the last row and column is dynamically identified.

```
01 Sub createPivotTableDynamicRange()
02
03     'Source: https://powerspreadsheets.com/
04     'For further information: https://powerspreadsheets.com/vba-create-pivot-table/
05
06     'declare variables to hold row and column numbers that define source data cell
```

```
    Dim myLastColumn As Long

    'declare variables to hold source and destination cell range address
    Dim mySourceData As String
    Dim myDestinationRange As String

    'declare object variables to hold references to source and destination worksheets, and new Pivot Table
    Dim mySourceWorksheet As Worksheet
    Dim myDestinationWorksheet As Worksheet
    Dim myPivotTable As PivotTable

    'identify source and destination worksheets
    With ThisWorkbook
        Set mySourceWorksheet = .Worksheets("Data")
        Set myDestinationWorksheet = .Worksheets("DynamicRange")
    End With

    'obtain address of destination cell range
    myDestinationRange = myDestinationWorksheet.Range("A5").Address(ReferenceStyle:=xlR1C1)

    'identify first row and first column of source data cell range
    myFirstRow = 5
    myFirstColumn = 1

    With mySourceWorksheet.Cells

        'find last row and last column of source data cell range
        myLastRow = .Find(What:="*", LookIn:=xlFormulas, LookAt:=xlPart, SearchOrder:=xlByRows, SearchDirection:=xlPrevious).Row
        myLastColumn = .Find(What:="*", LookIn:=xlFormulas, LookAt:=xlPart, SearchOrder:=xlByColumns, SearchDirection:=xlPrevious).Column

        'obtain address of source data cell range
        mySourceData = .Range(.Cells(myFirstRow, myFirstColumn), .Cells(myLastRow, myLastColumn)).Address(ReferenceStyle:=xlR1C1)

    End With

    'create Pivot Table cache and create Pivot Table report based on that cache
    Set myPivotTable = ThisWorkbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:=mySourceWorksheet.Name & "!" & mySourceData).CreatePivotTable(TableDestination:=myDestinationWorksheet.Name & "!" & myDestinationRange, TableName:="PivotTableExistingSheet")

    'add, organize and format Pivot Table fields
    With myPivotTable
        .PivotFields("Item").Orientation = xlRowField
        With .PivotFields("Units Sold")
            .Orientation = xlDataField
            .Position = 1
            .Function = xlSum
            .NumberFormat = "#,##0.00"
        End With
        With .PivotFields("Sales Amount")
            .Orientation = xlDataField
            .Position = 2
            .Function = xlSum
            .NumberFormat = "#,##0.00"
```

Effects of Executing Macro Example

The following GIF illustrates the results of executing this macro example. As expected, the macro creates a Pivot Table from a dynamic range.



	Values	
Row Labels	Sum of Units Sold	Sum of Sales Amount
Surface Book	12,098.00	18,134,902.00
Surface Pro 4	11,973.00	10,763,727.00
Surface Studio	11,771.00	35,301,229.00
Xbox One	12,081.00	3,008,169.00
Xbox One S	12,125.00	3,625,375.00
Grand Total	60,048.00	70,833,402.00

References to VBA Constructs Used in this VBA Tutorial

Use the following links to visit the appropriate webpage in the Microsoft Developer Network:

1. Identify or create the workbooks you work with:
    - Workbooks collection.
    - Workbooks.Add method.
    - Workbook object.
    - Application.Workbooks property.

- Application.ThisWorkbook property.

2. Identify or create the worksheets you work with:
    - Worksheets collection.
    - Worksheets.Add method.
    - Worksheet object.
    - Workbook.Worksheets property.
3. Identify the cell ranges you work with:
    - Range object.
    - Worksheet.Range property.
    - Worksheet.Cells property.
    - Range.Range property.
    - Range.Cells property.
    - Range.Item property.
    - Identify the last row or column with data in a worksheet:
        - Range.Find method.
        - Range.Row property.
        - Range.Column property.
4. Obtain the names and addresses of the workbooks, worksheets and cell ranges you work with:

- Worksheet.Name property.
- Range.Address property.

5. Create a Pivot Table cache and Pivot Table report:
    - PivotCache object.
    - Workbook.PivotCaches method.
    - PivotCaches.Create method.
    - xlPivotTableSourceType enumeration.
    - PivotCache.CreatePivotTable method.

6. Work with Pivot Table fields:
    - PivotField object.
    - PivotTable.PivotFields method.
    - PivotField.Orientation property.
    - PivotField.Position property.
    - PivotField.Function property.
    - PivotField.NumberFormat property.

7. Concatenate strings:
    - Concatenation operators.

- Dim statement.
- Set statement.
- = operator.
- Data types:
    - Long data type.
    - String data type.

9. Simplify object references:
    - With… End With statement.

Comments, Suggestions or Questions?

- *What macros and VBA constructs do you use to create Pivot Tables?*
- *Do you have any questions regarding the macros or constructs I use in the VBA Tutorial above?*
- *Would you change anything in the macro examples I use above?*

If you have these, or any other comments, suggestions or questions, please take a few seconds to let me know in the comments. **I would love to read your feedback.** Thanks in advance!

Lefkios Paikousis • 3 months ago

Hi! Great tutorial!
I was interested in the last part where you create a pivot table with dynamic range
Is it possible to reference a dynamic named range within the workbook?
I have a dynamic named range where the size (specifically the number of rows) is monitored by the offset function
Is it going to be faster than this or the same?

Thanks and again, great tutorial ! I really enjoyed it
Lefkios

Jorge A. Gomez Mod ➚ Lefkios Paikousis • 3 months ago

Hi Lefkios! Many thanks for your comment. I'm very happy to read you've found it useful and appreciate you taking the time to leave a comment!

If I understand the situation you're working on correctly, it should be possible to use a named range. However, I'm not sure if that (by itself) would solve any performance issues you may be experiencing.

If you're working with a named range, you can pass the name as a string when specifying the SourceData parameter of the PivotCaches.Create method. If this is what you want to do, you may be able to start with one of the examples above and simplify (i.e. delete) some sections. At the end of this comment I include a new version of the macro that creates a Pivot Table in a new worksheet from a named range (pivotTableData). You probably need to modify this further (for example, consider aspects such as the scope of the named range), but it may give you an idea of what I mean.

If your workbook is too slow, part of the problem may lie on the OFFSET (volatile) function. From this point of view, the approach where the macro finds the last row when executed (instead of relying on OFFSET) may help. This is the approach taken by the fourth example in the Tutorial above (which creates a Pivot Table from a dynamic range).

In my opinion, the 2 approaches (macro vs. dynamic range) are slightly different, with their own advantages