C H A P T E R

3

Source Code

### IF()

=IF(D4>=$B$15,D4\*$C$15,D4\*$C$14)

**=IF(D4>=$B$15,D4**$C$15D4\*$C$14)

Then

=IF(D4>=$B$15,**D4\*$C$15**,D4\*$C$14)

=IF(D4>=$B$15,D4\*$C$15,**D4\*$C$14**)

### SUMIF(), COUNTIF(), AVERAGEIF()

=SUMIF(C4:C10,E13,D4:D10)

=SUMIF(**C4:C10**,E13,D4:D10)

=SUMIF(C4:10,**E13**,D4:D10)

=SUMIF(C4:10,E13,**D4:D10**)

.

### SUMIFS()

=SUMIFS(sum\_range,criteria\_range\_1,criteria\_1,criteria\_range\_2,criteria\_2 and so on).

### VLOOKUP()

=VLOOKUP(value, table\_array, index\_number, [not\_exact\_match])

### MATCH()

= MATCH (Lookup\_value, Lookup\_array, Match\_type)

### INDEX()

INDEX(array, row\_number, [column\_number])

=INDEX(F4:J13,MATCH(C6,F4:F13,0),MATCH(D6,F4:J4,0))

### OFFSET()

=OFFSET(starting\_point, rows to move, columns to move, height, width)

=OFFSET(E6:I14,C5,C6,C8,C9)

=**OFFSET(E6:I14**,C5,C6,C8,C9)

=OFFSET(**E6**:I14,**C5**,C6,C8,C9)

=OFFSET(**E6**:I14,C5**,C6**,C8,C9)

=OFFSET(E6:I14,C5,C6,**C8,C9**)

### CHOOSE()

T=CHOOSE(index\_num, value1, value2, value3 . . . up to 254 values)

=**CHOOSE(D4**,D8:I8:D9:I9:D10:I10).

=CHOOSE(C6,D9:**K9,D10:K10,D1:K11**)

### NETWORKDAYS.INTL()

= NETWORKDAYS.INTL(Start\_date, End\_date, Weekend, Holidays)

=DATEDIF(start\_date,end\_date, "interval")

### WORKDAY()

=WORKDAY(Start\_date, Days, Holidays)

**=WORKDAY(B5,E5,C5:D5)**

=WORKDAY(B4,**E5**,C5:D5)

=WORKDAY(B5,E5**,C5:D5**)

### EOMONTH()

= EOMONTH(Start\_date, Months)

### DAYS360()

= DAYS360(Start\_date, End\_date, Method )

= DATE(Year,Month,Day)

### MOD()

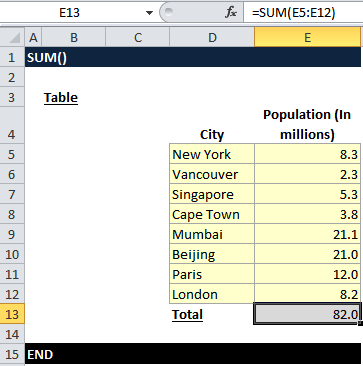
= MOD(Number, Divisor)

### **PRODUCT()**

=PRODUCT(Number1,Number2, . . . Number255)

### SUM()

=SUM(Number1,Number2, . . . Number255 )



**Figure 3-29.** This figure shows the SUM() function in use.

### AND()

=AND(logical-statement.)

=IF(AND(F10>C5,F10<=D5),E5,IF(AND(F10>=C6,F10<=D6),E6,0))

**=IF(AND(F10>C5,F10<=D5),E5**,IF(AND(F10>=C6,F10<=D6),E6,0))

=IF(AND(F10>C5,F10<=D5),E5,**IF(AND(F10>=C6,F10<=D6),E6**,0))

=IF(AND(F10>C5,F10<=D5),E5,IF(AND(F10>=C6,F10<=D6),E6,**0**))

### OR()

= OR(logical-1, logical-2, . . . logical-255 )

### ISERROR()

=ISERROR(value)

=IF(ISERROR(G6,"Problem","OK")

### SUMPRODUCT()

=SUMPRODUCT (list 1, list 2, . . . )

=SUMPRODUCT(F5:F13,G5:G13)

**=SUMPRODUCT(F5:F13**,G5:G13)

=SUMPRODUCT(F5:F13,**G5:G13**)

**=SUMPRODUCT(F5:F13,G5:G13**).

### LINKADDRESS()

=Linkaddress(Cell, [Default Value])

**The Code**

Function LinkAddress(cell As range, \_

Optional default\_value As Variant)

'Lists the Hyperlink Address for a Given Cell

'If cell does not contain a hyperlink, return default\_value

If (cell.range("A1").Hyperlinks.Count <> 1) Then

LinkAddress = default\_value

Else

LinkAddress = cell.range("A1").Hyperlinks(1).Address

End If

End Function

### EFFINDEX()

=EFFINDEX(demand,supply,[default\_value])

**The Code**

Function EFFINDEX(demand As Variant, supply As Variant, \_

Optional default\_value As Variant) As Variant

'Effectiveness Index (EFFINDEX)

If IsMissing(default\_value) Then

default\_value = "n/a"

End If

If IsNumeric(demand) And IsNumeric(supply) Then

If supply = 0 Then

EFFINDEX = demand

Exit Function

Else

EFFINDEX = demand / supply

Exit Function

End If

End If

EFFINDEX = default\_value

End Function

### SUMCOLOR()

=SumColor(ColorRange,SumRange)

The Code

Function SumColor(rColor As Range, rSumRange As Range)

'Sums cells based on a specified fill Color.

Dim rCell As Range

Dim iCol As Integer

Dim vResultiCol = rColor.Interior.ColorIndex

For Each rCell In rSumRange

If rCell.Interior.ColorIndex = iCol Then

vResult = WorksheetFunction.Sum(rCell) + vResult

End If

Next rCell

SumColor = vResult

End Function

### SUMTB()

=SumTB($A$2:$A$100,10) For top 10

=SumTB($A$2:$A$100,10,TRUE) For bottom 10

**The Code**

Function SUMTB(rRange As Range, N As Long, Optional bBottomN As Boolean) As Single

Dim strAddress As String

On Error Resume Next

strAddress = rRange.Address

If bBottomN = False Then

SUMTB = Evaluate("=SUMPRODUCT((" \_

& strAddress & ">=LARGE(" & strAddress & "," & X & "))\*(" & strAddress & "))")

Else

SUMTB = Evaluate("=SUMPRODUCT((" \_

& strAddress & "<=SMALL(" & strAddress & "," & X & "))\*(" & strAddress & "))")

End If

End Function

### ISDATE()

=ISDATE(Cell)

**The Code**

Function ISDATE(cell) As Boolean  
ISDATE = VBA.ISDATE(cell)  
End Function

### WORKSHEETSTATS()

=WORKSHEETSTATS()

**The Code**

Function WORKSHEETSTATS()

Set rng1 = ActiveSheet.UsedRange

On Error Resume Next

numConstants = rng1.SpecialCells(xlCellTypeConstants).Count

If Err <> 0 Then numConstants = 0: Err = 0

numerrors = rng1.SpecialCells(xlCellTypeConstants, xlErrors).Count

If Err <> 0 Then numerrors = 0: Err = 0

numLogical = rng1.SpecialCells(xlCellTypeConstants, xlLogical).Count

If Err <> 0 Then numLogical = 0: Err = 0

numText = rng1.SpecialCells(xlCellTypeConstants, xlTextValues).Count

If Err <> 0 Then numText = 0: Err = 0

numNumbers = rng1.SpecialCells(xlCellTypeConstants, xlNumbers).Count

If Err <> 0 Then numNumbers = 0: Err = 0

numformulas = rng1.SpecialCells(xlCellTypeFormulas).Count

If Err <> 0 Then numformulas = 0: Err = 0

numBlanks = rng1.SpecialCells(xlBlanks).Count

If Err <> 0 Then numBlanks = 0: Err = 0

Msg = "Address: " & Chr(9) & rng1.Address & Chr(10) & \_

"Last Row: " & Chr(9) & rng1.Rows(rng1.Rows.Count).Row & Chr(10) & \_

"Last Column: " & Chr(9) & rng1.Columns(rng1.Columns.Count).Column & Chr(10) & \_

"Total Cells: " & Chr(9) & rng1.Count & Chr(10) & \_

" Formulas: " & Chr(9) & numformulas & Chr(10) & \_

" Blanks: " & Chr(9) & numBlanks & Chr(10) & \_

" Constants:" & Chr(9) & numConstants & Chr(10)

Mg2 = "Errors: " & Chr(9) & numerrors & Chr(10) & \_

"Logical: " & Chr(9) & numLogical & Chr(10) & \_

"Text: " & Chr(9) & numText & Chr(10) & \_

"Numbers: " & Chr(9) & numNumbers

title1 = "SheetStats for " & Application.ActiveSheet.Name & \_

" in " & Application.ActiveWorkbook.FullName

iANS = MsgBox(Msg & Mg2, , title1)

End Function