```
Attribute VB_Name = "Statistics"
Option Compare Database
Option Explicit
Const e As Double = 2.71828182845905
Const pi As Double = 3.14159265358979

Public Function GetStatistics(SelectSQL As String, StDevField As String, Optional
StoredQuery As Boolean, Optional SingleValue As Variant) As Variant
On Error GoTo GetStatisticsError
'The function returns: 0 = Sum, 1 = Mean, 2 = Variance, 3 = Standard Deviation, 4 =
N, 5 = Skewness
'6 = Kurtosis, 7 = Median

Dim con As ADODB.Connection
Dim rst As New ADODB.Recordset
Dim cat As New ADOX.Catalog
Dim vw As ADOX.View
Dim act As ADOX.Procedure
Dim tbl As ADOX.Table
Dim cmd As ADODB.Command
Dim obj As AccessObject
Dim fld As ADODB.Field
Dim strSQL As String
Dim x As Integer
Set con = Application.CurrentProject.Connection
Set cat.ActiveConnection = con
Dim ReturnValues(7) As Double
Dim Holder1 As Double, Holder2 As Double
Dim RecCount As Double
Dim FoundObject As Boolean
FoundObject = False

If IsMissing(StoredQuery) = True Then
    'No need to take action
Else
    If StoredQuery = True Then
        'Check to see if the query exists
        For Each obj In CurrentData.AllQueries
            If obj.name = SelectSQL Then
                FoundObject = True
            End If
        Next obj
        If FoundObject = False Then
            For Each obj In CurrentData.AllTables
                If obj.name = SelectSQL Then
                    FoundObject = True
                End If
            Next obj
        End If
        'Catch Errors
        If FoundObject = False Then
            Err.Raise 7052, "User-Specified Input Query Name", "The stored
table/query could not be found in the current database"
        Else
            'Fetch the SQL - check views and procedures
            FoundObject = False
            For Each vw In cat.Views
                If vw.name = SelectSQL Then
                    Set vw = cat.Views(SelectSQL)
                    Set cmd = vw.Command
                    SelectSQL = cmd.CommandText
                    FoundObject = True
                End If
```

```
            Next vw
            'Check the procedures - only check if it wasn't a view
            If FoundObject = False Then
                For Each act In cat.Procedures
                    If act.name = SelectSQL Then
                        Set act = cat.Procedures(SelectSQL)
                        Set cmd = act.Command
                        SelectSQL = cmd.CommandText
                        FoundObject = True
                    End If
                Next act
            End If
            'Check the tables
            If FoundObject = False Then
                For Each tbl In cat.Tables
                    If tbl.name = SelectSQL Then
                        SelectSQL = "SELECT * FROM [" & SelectSQL & "]"
                        FoundObject = True
                    End If
                Next tbl
            End If
            'Get rid of Access' closing ;
            SelectSQL = Replace(SelectSQL, ";", "")
        End If
    Else
        'No need to take action
    End If
End If

'Check to see if the field exists in the input query
FoundObject = False
With rst
    .Open SelectSQL, con, adOpenStatic, adLockReadOnly
    For Each fld In rst.Fields
        If fld.name = StDevField Then
            FoundObject = True
        End If
    Next fld
    .Close
    If FoundObject = False Then
        'Could not find field
        Err.Raise 7053, "User-Specified Query Field", "The specified query field
name could not be found in the submitted query"
    End If
End With

'Check for records
With rst
    .Open SelectSQL, con, adOpenStatic, adLockReadOnly
    If .BOF = True And .EOF = True Then
        'no records returned - unable to calculate
        For x = 0 To 7
            ReturnValues(x) = -999.999
        Next x
        Err.Raise 7051, "Input Query", "No records returned by input query for field
" & StDevField
    End If
    .Close
End With

'First get the mean and standard deviation
strSQL = "SELECT Avg([" & StDevField & "]) As AvgField, Count([" & StDevField & "])
As CountField, Sum([" & StDevField & "]) As SumField"
```

```
strSQL = strSQL & " FROM (" & SelectSQL & ")"
With rst
    .Open strSQL, con, adOpenStatic, adLockReadOnly
    ReturnValues(0) = .Fields("SumField")
    ReturnValues(1) = .Fields("AvgField")
    ReturnValues(4) = .Fields("CountField")
    .Close
End With

'Check for a numeric value
If IsNumeric(ReturnValues(1)) = False Then
    Err.Raise 7050, "Input Query", "StDevField parameter references a non-numeric
field"
End If

'Get the standard deviation
strSQL = "SELECT ([" & StDevField & "] - " & ReturnValues(1) & ")^2 As
SquaredDeviation FROM (" & SelectSQL & ")"
strSQL = "SELECT SUM([SquaredDeviation]) As SumSquares FROM (" & strSQL & ")"
strSQL = "SELECT ([SumSquares] / (" & ReturnValues(4) & " - 1)) As Variance FROM ("
& strSQL & ")"
With rst
    .Open strSQL, con, adOpenStatic, adLockReadOnly
    ReturnValues(2) = .Fields("Variance")
    ReturnValues(3) = Sqr(CDbl(.Fields("Variance")))
    .Close
End With

'Get the skewness
Holder1 = (ReturnValues(4)) / ((ReturnValues(4) - 1) * (ReturnValues(4) - 2))
'Get the summation term
strSQL = "SELECT (([" & StDevField & "] - " & ReturnValues(1) & ") / " &
ReturnValues(3) & ")^3 As CubeDeviation FROM (" & SelectSQL & ")"
strSQL = "SELECT Sum([CubeDeviation]) As SumCube FROM (" & strSQL & ")"
With rst
    .Open strSQL, con, adOpenStatic, adLockReadOnly
    'Return the skewness
    ReturnValues(5) = Holder1 * .Fields("SumCube")
    .Close
End With

'Get the kurtosis
Holder1 = (ReturnValues(4) * (ReturnValues(4) + 1)) / (((ReturnValues(4) - 1) *
(ReturnValues(4) - 2) * (ReturnValues(4) - 3)))
Holder2 = (3 * ((ReturnValues(4) - 1) ^ 2)) / ((ReturnValues(4) - 2) *
(ReturnValues(4) - 3))
'Get the summation term
strSQL = "SELECT (([" & StDevField & "] - " & ReturnValues(1) & ") / " &
ReturnValues(3) & ")^4 As QuadDeviation FROM (" & SelectSQL & ")"
strSQL = "SELECT Sum([QuadDeviation]) As SumQuad FROM (" & strSQL & ")"
With rst
    .Open strSQL, con, adOpenStatic, adLockReadOnly
    'Return the Kurtosis
    ReturnValues(6) = (Holder1 * .Fields("SumQuad")) - Holder2
    .Close
End With

'Get the median
'First get a record count
strSQL = "SELECT COUNT(*) As RecCount FROM (" & SelectSQL & ")"
With rst
    .Open strSQL, con, adOpenStatic, adLockReadOnly
    RecCount = .Fields("RecCount")
```

```
    .Close
End With
strSQL = SelectSQL & " ORDER BY [" & StDevField & "] ASC"
'Decide which median methodology to use
If Left(CStr(RecCount), 1) = 0 Or Left(CStr(RecCount), 1) = 2 Or
Left(CStr(RecCount), 1) = 4 Or Left(CStr(RecCount), 1) = 6 Or Left(CStr(RecCount),
1) =

8 Then
    'Even Number of Records
    With rst
        .Open strSQL, con, adOpenStatic, adLockReadOnly
        RecCount = CDbl(RecCount)
        RecCount = RecCount / 2
        .Move RecCount
        ReturnValues(7) = .Fields(StDevField)
        .Close
    End With
Else
    'Odd Number of Records
    With rst
        .Open strSQL, con, adOpenStatic, adLockReadOnly
        RecCount = CDbl(RecCount)
        RecCount = (RecCount / 2) - 0.5
        .Move RecCount
        RecCount = CDbl(.Fields(StDevField))
        .MoveNext
        RecCount = (RecCount + CDbl(.Fields(StDevField))) / 2
        ReturnValues(7) = RecCount
        .Close
    End With
End If

'Return the values
GetStatistics = ReturnValues
If IsMissing(SingleValue) = True Then
    GetStatistics = ReturnValues
Else
    If IsNull(SingleValue) = True Then
        GetStatistics = ReturnValues
    Else
        GetStatistics = ReturnValues(SingleValue)
    End If
End If

'Close the connections
con.Close
If rst.State = adStateOpen Then
    rst.Close
End If
Set rst = Nothing
Set con = Nothing
Set cat = Nothing
Set tbl = Nothing
Set cmd = Nothing
Set vw = Nothing
Set obj = Nothing
Set fld = Nothing
Set act = Nothing
Exit Function

GetStatisticsError:
If Err.Number = 7051 Then
```

```
    'Return the -999.999 values
    GetStatistics = ReturnValues
End If
If Err.Number < 0 Then
    MsgBox "The database SQL generation engine encountered an error due to a
malformed SQL command." & vbCrLf & vbCrLf & "The SQL statement at fault is:

" & vbCrLf & strSQL, vbCritical + vbOKOnly, "SQL Command Error"
Else
    MsgBox "GetStatistics encountered error " & Err.Number & " : " & Err.Description
& " thrown by " & Err.Source & ".", vbCritical + vbOKOnly, "Unable

to Calculate Statistics"
End If
con.Close
If rst.State = adStateOpen Then
    rst.Close
End If
Set rst = Nothing
Set con = Nothing
Set cat = Nothing
Set tbl = Nothing
Set cmd = Nothing
Set vw = Nothing
Set obj = Nothing
Set fld = Nothing
Set act = Nothing
Exit Function

End Function

Public Function OLS(SelectSQL As String, XValue As String, YValue As String,
Optional StoredQuery As Boolean, Optional SingleValue As Variant) As

Variant
On Error GoTo OLSError
'The function returns: 0 = Slope, 1 = Intercept, 2 = r, 3 = R-Squared

Dim con As ADODB.Connection
Dim rst As New ADODB.Recordset
Dim cat As New ADOX.Catalog
Dim vw As ADOX.View
Dim act As ADOX.Procedure
Dim tbl As ADOX.Table
Dim cmd As ADODB.Command
Dim obj As AccessObject
Dim fld As ADODB.Field
Dim strSQL As String
Dim x As Integer
Set con = Application.CurrentProject.Connection
Set cat.ActiveConnection = con
Dim FoundObject As Boolean, FoundObject2 As Boolean
FoundObject = False
FoundObject2 = False
Dim SumXY As Double, SumX As Double, SumY As Double, SumXSq As Double, SumYSq As
Double, n As Double, OLSSlope As Double, OLSIntercept As Double, OLSR

As Double
Dim ReturnValues(4) As Variant

If IsMissing(StoredQuery) = True Then
    'No need to take action
Else
```

```
    If StoredQuery = True Then
        'Check to see if the query exists
        For Each obj In CurrentData.AllQueries
            If obj.name = SelectSQL Then
                FoundObject = True
            End If
        Next obj
        If FoundObject = False Then
            For Each obj In CurrentData.AllTables
                If obj.name = SelectSQL Then
                    FoundObject = True
                End If
            Next obj
        End If
        'Catch Errors
        If FoundObject = False Then
            Err.Raise 7052, "User-Specified Input Query Name", "The stored
table/query could not be found in the current database"
        Else
            'Fetch the SQL - check views and procedures
            FoundObject = False
            For Each vw In cat.Views
                If vw.name = SelectSQL Then
                    Set vw = cat.Views(SelectSQL)
                    Set cmd = vw.Command
                    SelectSQL = cmd.CommandText
                    FoundObject = True
                End If
            Next vw
            'Check the procedures - only check if it wasn't a view
            If FoundObject = False Then
                For Each act In cat.Procedures
                    If act.name = SelectSQL Then
                        Set act = cat.Procedures(SelectSQL)
                        Set cmd = act.Command
                        SelectSQL = cmd.CommandText
                        FoundObject = True
                    End If
                Next act
            End If
            'Check the tables
            If FoundObject = False Then
                For Each tbl In cat.Tables
                    If tbl.name = SelectSQL Then
                        SelectSQL = "SELECT * FROM [" & SelectSQL & "]"
                        FoundObject = True
                    End If
                Next tbl
            End If
            'Get rid of Access' closing ;
            SelectSQL = Replace(SelectSQL, ";", "")
        End If
    Else
        'No need to take action
    End If
End If

'Check to see if the fields exist in the input query
FoundObject = False
FoundObject2 = False
With rst
    .Open SelectSQL, con, adOpenStatic, adLockReadOnly
    For Each fld In rst.Fields
```

```
        If fld.name = XValue Then
            FoundObject = True
        End If
    Next fld
    For Each fld In rst.Fields
        If fld.name = YValue Then
            FoundObject2 = True
        End If
    Next fld
    .Close
    If FoundObject = False And FoundObject2 = False Then
        'Could not find field
        Err.Raise 7053, "User-Specified Query Field", "The specified query field
name could not be found in the submitted query"
    End If
End With

'Check for records
With rst
    .Open SelectSQL, con, adOpenStatic, adLockReadOnly
    If .BOF = True And .EOF = True Then
        'no records returned - unable to calculate
        Err.Raise 7051, "Input Query", "No records returned by input query"
    End If
    .Close
End With

'Get the SumX and SumY value
strSQL = "SELECT Sum([" & XValue & "]) As SumX, Sum([" & YValue & "]) As SumY,
Count([" & XValue & "]) As N FROM (" & SelectSQL & ")"
With rst
    .Open strSQL, con, adOpenStatic, adLockReadOnly
    SumX = .Fields("SumX")
    SumY = .Fields("SumY")
    n = .Fields("N")
    .Close
End With

'Get the SumXY
strSQL = "SELECT [" & XValue & "], [" & YValue & "], [" & XValue & "]*[" & YValue &
"] As XY, [" & XValue & "]^2 As XSq, [" & YValue & "]^2 As YSq FROM

(" & SelectSQL & ")"
strSQL = "SELECT Sum(XY) As SumXY, Sum(XSq) As SumXSq, Sum(YSq) As SumYSq FROM (" &
strSQL & ")"
With rst
    .Open strSQL, con, adOpenStatic, adLockReadOnly
    SumXY = .Fields("SumXY")
    SumXSq = .Fields("SumXSq")
    SumYSq = .Fields("SumYSq")
    .Close
End With

'Calculate the slope
OLSSlope = ((n * SumXY) - (SumX * SumY)) / ((n * SumXSq) - SumX ^ 2)
ReturnValues(0) = OLSSlope

'Calculate the intercept
OLSIntercept = (SumY - (OLSSlope * SumX)) / n
ReturnValues(1) = OLSIntercept

'Calculate the R value
OLSR = ((n * SumXY) - (SumX * SumY)) / Sqr((((n * SumXSq) - SumX ^ 2) * ((n *
```

```
SumYSq) - SumY ^ 2)))
ReturnValues(2) = OLSR

'Calculate the R-Squared
ReturnValues(3) = OLSR ^ 2

'Combine for an equation
ReturnValues(4) = "Y = " & CStr(OLSSlope) & "x + " & CStr(OLSIntercept)

'Return the values
If IsMissing(SingleValue) = True Then
    OLS = ReturnValues
Else
    If IsNull(SingleValue) = True Then
        OLS = ReturnValues
    Else
        OLS = ReturnValues(SingleValue)
    End If
End If

'Close the connections
con.Close
If rst.State = adStateOpen Then
    rst.Close
End If
Set rst = Nothing
Set con = Nothing
Set cat = Nothing
Set tbl = Nothing
Set cmd = Nothing
Set vw = Nothing
Set obj = Nothing
Set fld = Nothing
Set act = Nothing
Exit Function

OLSError:
If Err.Number < 0 Then
    MsgBox "The SQL generation engine encountered an error due to a malformed SQL
command." & vbCrLf & vbCrLf & "The SQL statement at fault is: " &

vbCrLf & strSQL, vbCritical + vbOKOnly, "SQL Command Error"
Else
    MsgBox "OLS encountered error " & Err.Number & " : " & Err.Description & "
thrown by " & Err.Source & ".", vbCritical + vbOKOnly, "Unable to

Calculate Statistics"
End If
con.Close
If rst.State = adStateOpen Then
    rst.Close
End If
Set rst = Nothing
Set con = Nothing
Set cat = Nothing
Set tbl = Nothing
Set cmd = Nothing
Set vw = Nothing
Set obj = Nothing
Set fld = Nothing
Set act = Nothing
Exit Function
```

```
End Function
```