

```

/*-----*
* File Name: Bin2D.c                                     *
* Creation: ER, 02/07/05                                 *
* Purpose: Programming Example                           *
* Copyright (c) OriginLab Corp.2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010 *
* All Rights Reserved                                   *
*                                                       *
* Modification Log:                                     *
*-----*/

#include <Origin.h>

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// This function takes the data in the first two columns of the active worksheet
// such as:
//   X   Y
//   1   2
//   3   3
//   1   2
//   3   2
//   6   1
//   3   2
//   3   2
//   3   2
//   1   2
// and performs 2D binning of the data, to create a result worksheet such as:
//   X Y Counts
//   1 2 3
//   3 3 1
//   3 2 4
//   6 1 1
// Parameters:
//   dxStart:    bin center for 1st X bin
//   dxEnd:      bin center for last X bin
//   dxStep:     width of each bin
//   dyStart:    bin center for 1st Y bin
//   dyEnd:      bin center for last Y bin
//   dyStep:     width of each bin
//
void bin_2d(double dxStart, double dxEnd, double dxStep, double dyStart, double dyEnd, double dyStep)
{
    // Declare worksheet with active layer
    Worksheet wksData = Project.ActiveLayer();
    if( !wksData )
    {
        out_str("Active layer is not a worksheet!");
        return;
    }

    // First copy data from 1st and 2nd cols of wks to a matrix
    matrix matData;
    matData.CopyFrom(wksData, 0, 0, -1, 1);

    // Compute number of bins needed in x and y directions
    int nXBins = 1 + (dxEnd - dxStart) / dxStep;
    int nYBins = 1 + (dyEnd - dyStart) / dyStep;

    // Create a matrix object to hold 2D bin counts
    MatrixPage pgMat;
    pgMat.Create("Origin");
    pgMat.Label="2D Bin counts for " + wksData.GetPage().GetName();
    pgMat.TitleShow = WIN_TITLE_SHOW_BOTH;
    MatrixLayer lyMat = pgMat.Layers(0);
    lyMat.SetCellWidth(4);
    Matrix MatBins(lyMat);
    // Set the size and X, Y coordinates of the matrix
    MatBins.SetSize(nYBins, nXBins);
    MatBins.SetXMin(dxStart);
    MatBins.SetXMax(dxEnd);
    MatBins.SetYMin(dyStart);
    MatBins.SetYMax(dyEnd);
    MatBins = 0;

    // Loop over all rows of the data matrix - this corresponds to rows in wks
    int nRows = matData.GetNumRows();
    for(int iRow = 0; iRow < nRows; iRow++)
    {

```

```

    // Get X, Y values to be binned
    double dX = matData[iRow][0];
    double dY = matData[iRow][1];
    // Update matrix count at the appropriate cell
    MatBins.SetCellValue(dX, dY, MatBins.GetCellValue(dX, dY) + 1);
}
// Create new wks to hold counts
WorksheetPage wpg;
wpg.Create("Origin");
Worksheet wksCounts = wpg.Layers(0);
wpg.Label = "2D Bin counts for " + wksData.GetPage().GetName();
wpg.TitleShow = WIN_TITLE_SHOW_BOTH;
// Delete all columns and add three new cols
while(wksCounts.DeleteCol(0));
wksCounts.AddCol("XValue");
wksCounts.AddCol("YValue");
wksCounts.AddCol("Counts");

// Convert the matrix to XYZ columns of the counts wks
LabTalk.mat.matname$ = pgMat.GetName();
LabTalk.mat.wksname$ = wpg.GetName();
LabTalk.mat.m2xyz();

// Loop over all rows of wks backwards and delete rows that have 0 in 3rd col
for(int ir = wksCounts.GetNumRows() - 1; ir >= 0; ir--)
{
    if( 0 == wksCounts.Cell(ir, 2) ) wksCounts.DeleteRow(ir);
}
}
////////////////////////////////////

```