

○ 「 エクセルVBAの活用 ; 分布測定 」

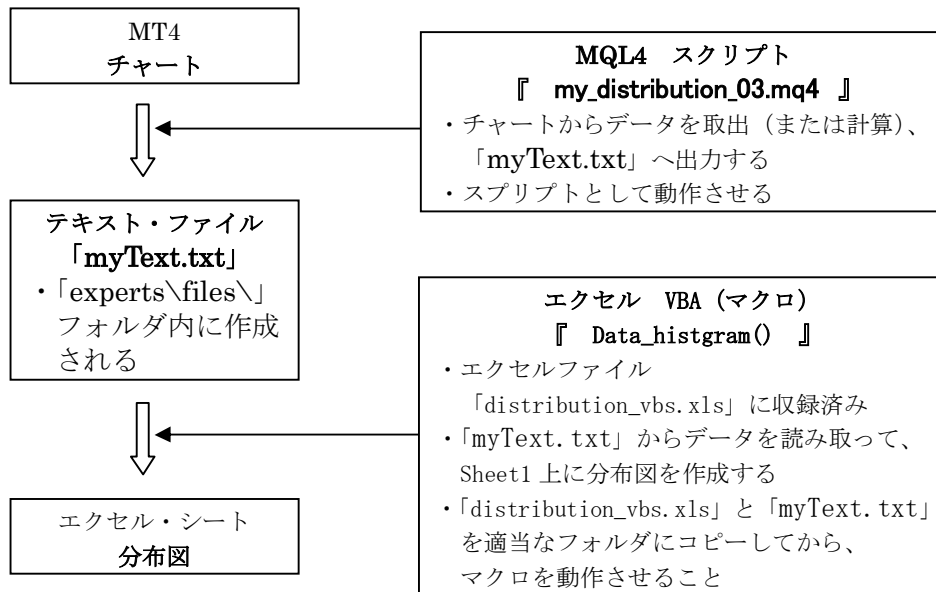
- ・ アメンボです、これから擬似トレード用の少しマシな EA を検討するのですが、急がば回れで、チョット道草をします。
 - ・ これから EA を検討する上で、様々なデータ分布を測定する必要があります。
MT4 (MQL4) は元々、エクセルとの連携を考慮して作られています、アメンボ流の解析のためには物足りない部分がありますので、少しずつですがエクセル用のツールを準備することにしました。
 - ・ アメンボは、データを統計解析するツールとしてエクセルをよく使います、そこで、今回は「MT4 の生データから分布図」を作成するエクセルVBA (マクロ) を提案することにしました。(実際にアメンボが今、使っているツールです)
 - ・ 大体、自身で基礎データを収集・分析せずに、ストップロスを決めたり、いきなりEAのアルゴリズムを検討してバックテストをするのは、手順が間違っていると思いませんか？
色々な記事を読むにつけ、アメンボが抱く感想です。
 - ・ 本稿で解説に使用する下記資料は、ダウンロード用にWEB上にUPしています。
 - ①MT4 からデータをテキストファイルに書出す ; 「my_distribution_03.mq4」
 - ②分布作成用エクセルVBA (マクロ) ; 「distribution_vbs.xls」
 - ③データを書出したファイル例 ; 「myText.txt」
- ◆ 「②と③」を、同じフォルダに入れて、マクロを動作させると、
分布図を自動生成します。

(エクセルは 2003 版です、未確認ですが 2010 版等でもそのまま使えるはず)

目次 :	1. 分布図作成の手順	・・・	2 頁
	(1) 手順概要		
	(2) 手順詳細		
	2. コード内容一覧	・・・	8 頁
	(1) データ書き出し<スクリプト>	「my_distribution_03.mq4」	
	(2) 分布図作成エクセルVBA (マクロ)	「distribution_vbs.xls」	

1. 分布図作成の手順

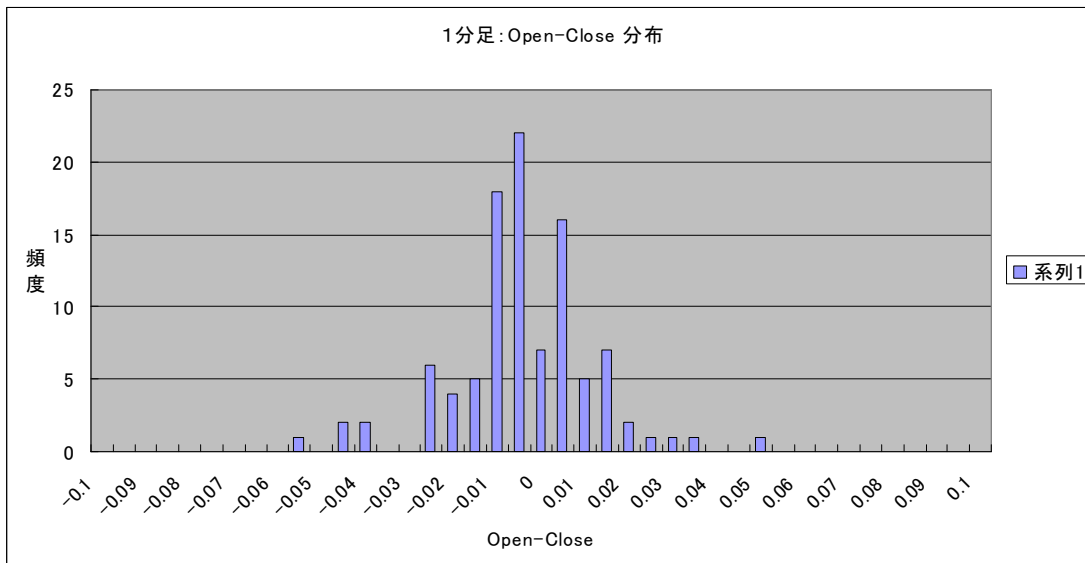
(1) 手順概要； MT4（チャート）からデータを出力し、エクセルで分布図を作るまで



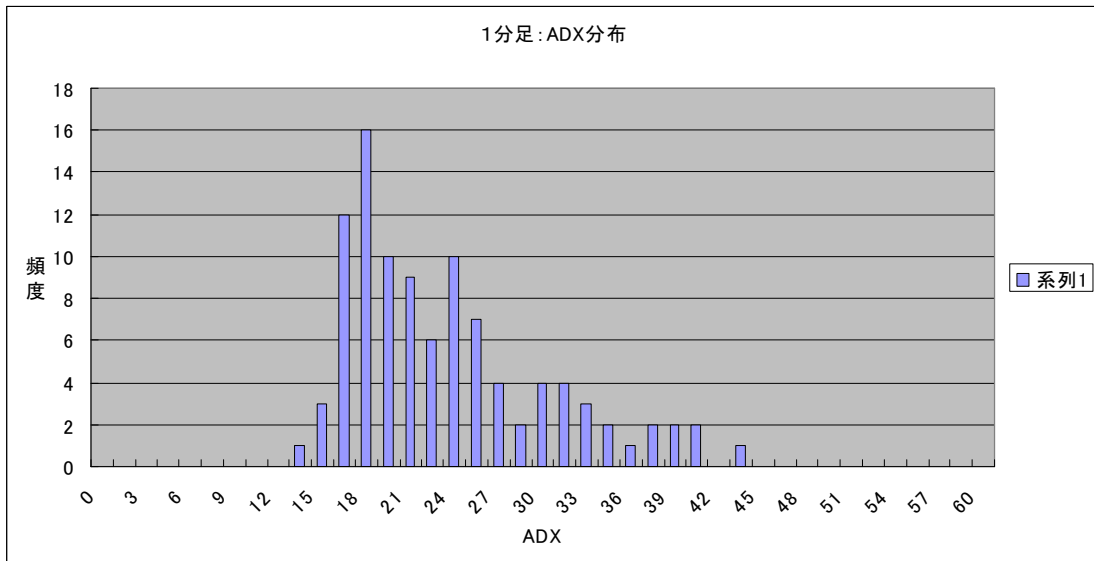
テキスト・ファイル「myText.txt」への出力例（Open-Close）；

0.017	0.01	-0.01
-0.007	-0.015	0.021
0.007	-0.022	0.018
-0.021	-0.022	-0.037
-0.051	0.009	0.009
-0.019	-0.012	-0.001
0.007	0	-0.006
0.007	0.008	-0.001
0.036	-0.021	-0.002
0.002	-0.002	-0.008
0.027	-0.006	0.005

分布図例（Open-Close）；



分布図例 (ADX) ;



(2) 手順詳細

ー 1. MT4 からデータを書出すには

- ・ 諸兄は、MT4 と MQL4 には有る程度慣れてしていると勝手ながら推測しますので、スクリプトの動作方法についての説明は割愛し、簡単な解説・注意点のみを記載します。
- ・ 『my_distribution_03.mq4』 はスクリプトですので、「experts\scripts\」フォルダに入れてください。

※スクリプト 『my_distribution_03.mq4』 を実行すると、

- ① 「myText.txt」 を事前に準備していなくとも、「experts\files\」フォルダ内に自動的に作成されます、
- ② また事前に「experts\files\」フォルダ内に「myText.txt」を作っておくと、その中にデータを書き込みます。

※繰り返し、スクリプト 『my_distribution_03.mq4』 を実行すると、

「myText.txt」内容をクリアしてから、新しいデータを書き込みます。

ー 2. エクセル VBA (マクロ) を動作させるには

- ・初めにお断りしなければならないこととして、アメンボは現在「エクセル 2003」しか持っていないため、

解説は「エクセル 2003」でのものです、また、最新版の「2010」？での動作を確認していません、こと、および最新版でのマクロを動作させる手順を知りません。

ただ、通常は最新バージョン上で、旧バージョンのマクロは動作するはずですが。

- ・もしかすると、MQL4 ユーザー各位には、エクセルのマクロを知らない方がいるかもしれませんが、「マクロ」の詳細解説はご容赦ください。解説本は山の様にありますこと、またネット上にも資料が溢れていますので、参照して頂きたい。(それに、アメンボのマクロの知識は初級以上、中級未満ですため)

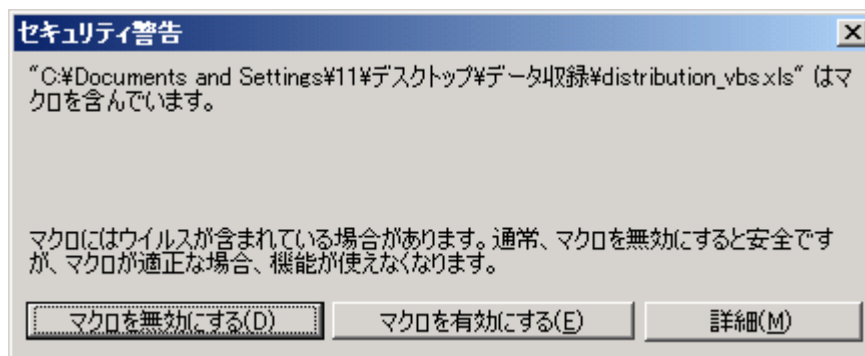
①動作を確認する準備；

ダウンロードした「distribution_vbs.xls」と「myText.txt」を、同じフォルダー中に入れてください。(フォルダは適当に作ったものでOK)



②エクセルシート「distribution_vbs.xls」をダブルクリックする；

「マクロ」が組み込まれているので、警告がポップアップします。(エクセル 2003 の場合)



◇ [マクロを有効にする] をクリックしてください。

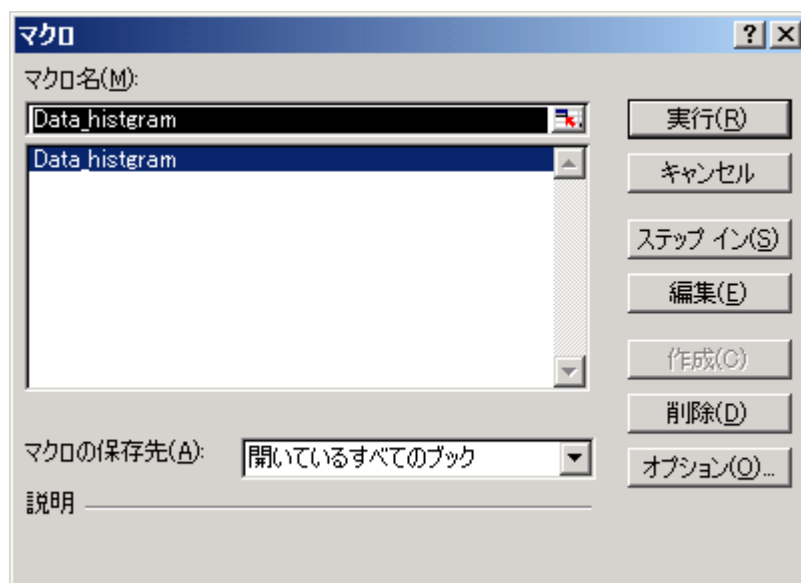
③ 「Sheet1」 が、下記の様に開きます。

・動作が判りやすい様に数値データは残していますが、グラフデータのみは削除してあります。



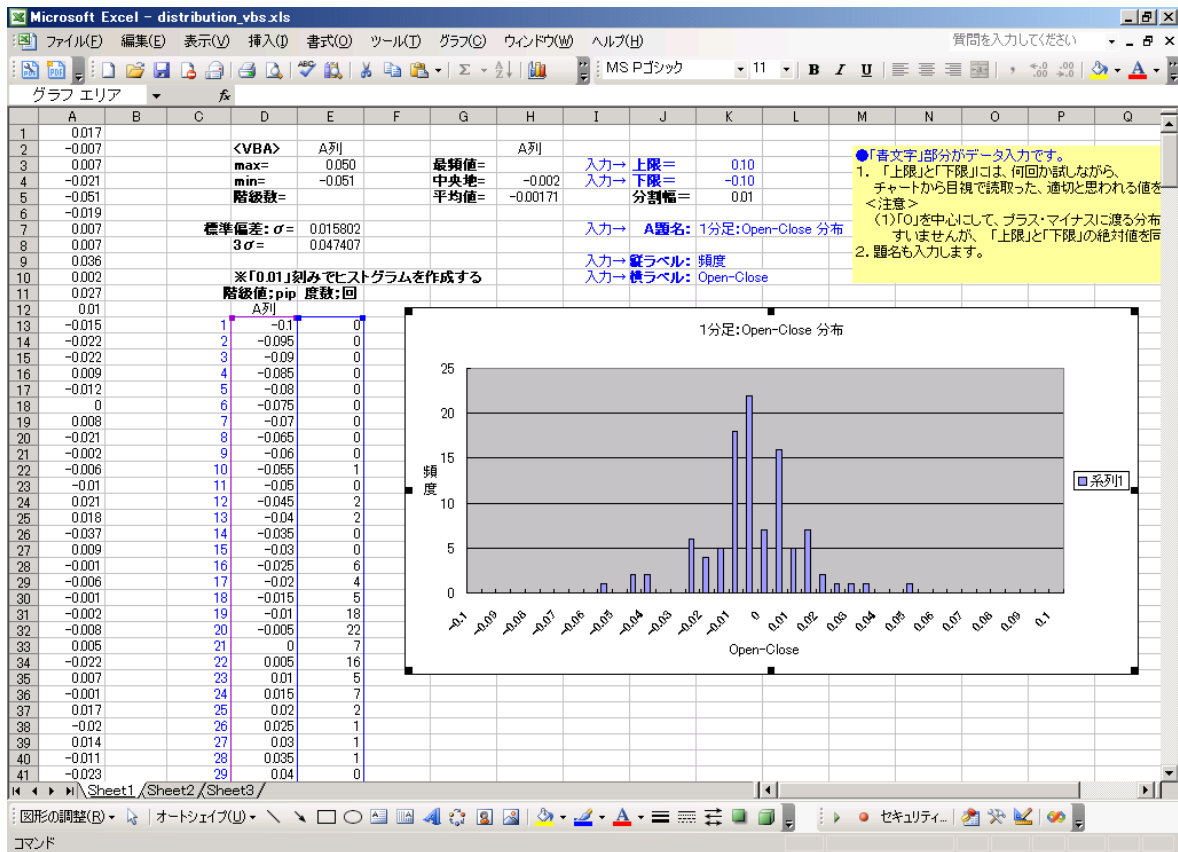
◇ [ツール] - [マクロ] - [マクロ] と選択します。

④ 「マクロ」のリストが表示されます



◇ 「Data_histogram」が選択された状態で、[実行] をクリックします

⑤目の前で、マクロが動作し、グラフ（分布図）が表示されます。




◆使用上の注意 ;

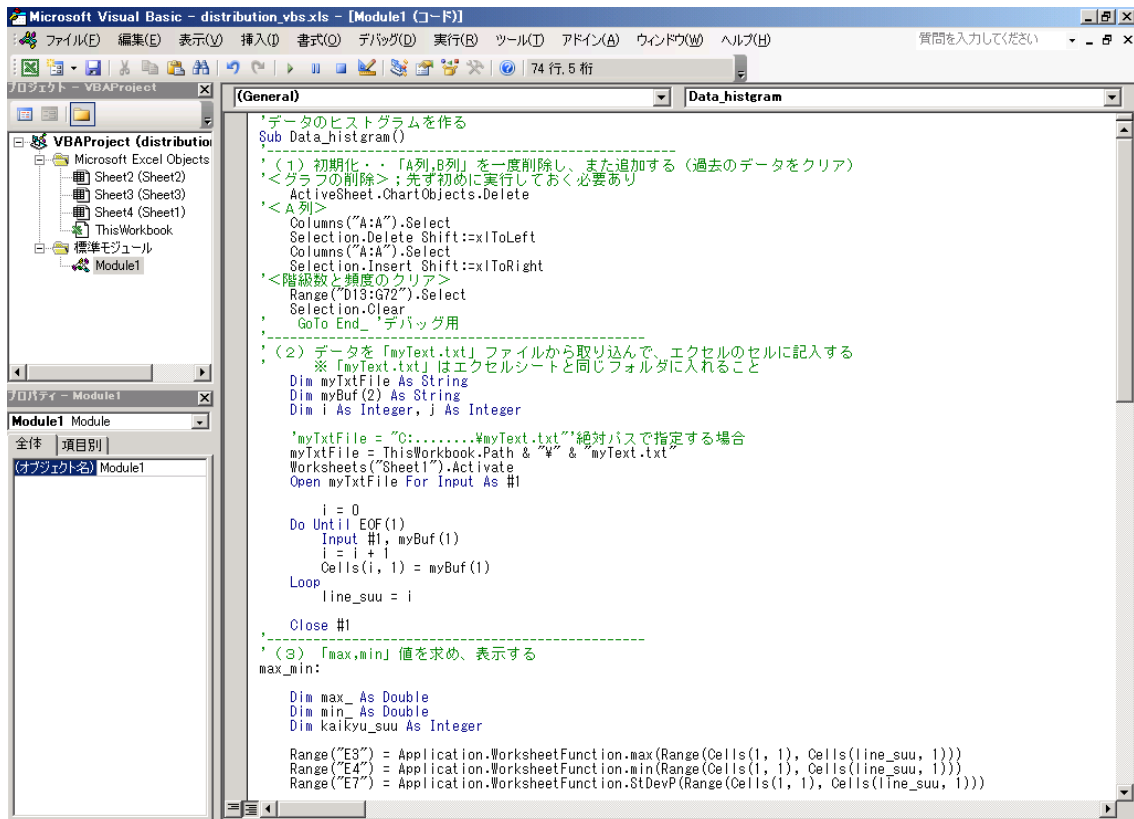
	A列				
最頻値=		入力→	上限=	0.10	
中央地=	-0.002	入力→	下限=	-0.10	
平均値=	-0.00171		分割幅=	0.01	
		入力→	A題名: 1分足: Open-Close 分布		
		入力→	縦ラベル:	頻度	
作成する		入力→	横ラベル:	Open-Close	

- ・上記の「青書」部のセルは、入力用です。（色々触ると直ぐに理解できます）
例えば、「0.10、-0.10、1分足: Open-Close 分布、頻度、Open-Close」は、全て手入力したデータで、エクセルのグラフ作成に反映されます。
- ・マクロの処理内容を理解するには、コードを覗いて頂くのが最短?の方法です。
- ・アメンボは、入出力に「フォーム」は殆ど使いません、エクセルには、入出力兼用のセルが無数にありますので、活用しない手はありません。
- ・A列; 読み込んだデータ、D 1 3 ~ ; 階層化、E 1 3 ~ ; 出現頻度

●マクロの内容を確認するには；

- 〈1〉 [ツール] - [マクロ] - [Visual Basic Editor] で開くか、
- 〈2〉 [表示] - [ツールバー] で「Visual Basic」にチェックを入れて、
表示させた「Visual Basic 専用ツールバー」

 をクリックして開きます。



```

Microsoft Visual Basic - distribution_vbs.xls - [Module1 (コード)]
(General) Data_histogram

'データのヒストグラムを作る
Sub Data_histogram()
'-----
' (1) 初期化・・・「A列,B列」を一度削除し、また追加する（過去のデータをクリア）
' <グラフの削除>； 先ず初めに実行しておく必要あり
ActiveSheet.ChartObjects.Delete
' < A列 >
Columns("A:A").Select
Selection.Delete Shift:=xlToLeft
Columns("A:A").Select
Selection.Insert Shift:=xlToRight
' <階級数と頻度のクリア>
Range("B13:G72").Select
Selection.Clear
GoTo End_ デバッグ用
'-----
' (2) データを「myText.txt」ファイルから取り込んで、エクセルのセルに記入する
' ※「myText.txt」はエクセルシートと同じフォルダに入れること
Dim myTxtFile As String
Dim myBuf(2) As String
Dim i As Integer, j As Integer

myTxtFile = "0:.....\myText.txt" '絶対パスで指定する場合
myTxtFile = ThisWorkbook.Path & "\*" & "myText.txt"
Worksheets("Sheet1").Activate
Open myTxtFile For Input As #1

i = 0
Do Until EOF(1)
Input #1, myBuf(1)
i = i + 1
Cells(i, 1) = myBuf(1)
Loop
line_suu = i


Close #1
'-----
' (3) 「max,min」値を求め、表示する
max_min:

Dim max_ As Double
Dim min_ As Double
Dim kaikyusuu As Integer

Range("E3") = Application.WorksheetFunction.max(Range(Cells(1, 1), Cells(line_suu, 1)))
Range("E4") = Application.WorksheetFunction.min(Range(Cells(1, 1), Cells(line_suu, 1)))
Range("E7") = Application.WorksheetFunction.StDevP(Range(Cells(1, 1), Cells(line_suu, 1)))

```

※エクセルシートに戻るには、

- 〈1〉 [ファイル] - [終了して Microsoft Excel へ戻る] か、
- 〈2〉 ツールバー上の  をクリックします。

※この状態で、マクロを実行するには、

マクロコード中にカーソルを点滅させてから、  をクリックします。

(結果を確認するにはエクセルシートに戻る必要があります)

※この画面では、マクロコードの修正やデバックを実行できます、

詳細は他の資料（参考書、WEB）を参照ください。

2. コード内容一覧

(1) データ書き出し<スクリプト>

下記コード中、

①を使うと、「Open-Close」データ書き出し

②を使うと、「ADX」データ書き出し

```

///
//          TXT ファイルへのデータ書き出し
// スクリプト;『 my_distribution_03.mq4 』
//
//          ・ポイント;   上書きモード [ FILE_WRITE ]
//                          追記モード [ FILE_READ|FILE_WRITE ]
//
//スタート関数-----
int start()
{
//-----先ず、ファイルを一度クリアする-----
//          上書きモードで開いてから、即閉じる
    int handle1;
    handle1=FileOpen("myText.txt", FILE_CSV|FILE_WRITE,"");
    FileClose(handle1);
//-----
    int handle2;
    handle2=FileOpen("myText.txt", FILE_CSV|FILE_READ|FILE_WRITE,"");

    if(handle2<1)
    {
        Print("can't open file error-",GetLastError());
        return(-1);
    }

    if(handle2>0)
    {
        double ADX0;
        FileSeek(handle2, 0, SEEK_END);
        //
        for(int i=0;i<=100;i++)
        {
            /// ①「Open-Close」を出力する場合は、こちらを使う
            ///FileWrite(handle2,Open[i]-Close[i]);
            /// ②「ADX」データ出力
            ADX0=iADX(NULL,0,14,PRICE_CLOSE,MODE_MAIN,i);
            FileWrite(handle2,ADX0);
        }
        //
        FileClose(handle2);
    }
    return(0);
}

```


(2) 分布図作成エクセルVBA (マクロ)

- ・「distribution_vbs.xls」にマクロとして組み込まれています、マクロの名称は「Data_histogram」です。

```
'データのヒストグラムを作る
Sub Data_histogram()
'-----
' (1) 初期化・・・「A列,B列」を一度削除し、また追加する (過去のデータをクリア)
'<グラフの削除> ; 先ず初めに実行しておく必要あり
    ActiveSheet.ChartObjects.Delete
'<A列>
    Columns("A:A").Select
    Selection.Delete Shift:=xlToLeft
    Columns("A:A").Select
    Selection.Insert Shift:=xlToRight
'<階級数と頻度のクリア>
    Range("D13:G72").Select
    Selection.Clear
'    GoTo End_ 'デバッグ用
'-----
' (2) データを「myText.txt」ファイルから取り込んで、エクセルのセルに記入する
'    ※「myText.txt」はエクセルシートと同じフォルダに入れること
    Dim myTxtFile As String
    Dim myBuf(2) As String
    Dim i As Integer, j As Integer

    'myTxtFile = "C:.....\myText.txt"絶対パスで指定する場合
    myTxtFile = ThisWorkbook.Path & "\ & "myText.txt"
    Worksheets("Sheet1").Activate
    Open myTxtFile For Input As #1

        i = 0
    Do Until EOF(1)
        Input #1, myBuf(1)
        i = i + 1
        Cells(i, 1) = myBuf(1)
    Loop
        line_suu = i

    Close #1
'-----
' (3) 「max,min」値を求め、表示する
max_min:
```

```

Dim max_ As Double
Dim min_ As Double
Dim kaikyusuu As Integer

Range("E3") = Application.WorksheetFunction.max(Range(Cells(1, 1), Cells(line_suu, 1)))
Range("E4") = Application.WorksheetFunction.min(Range(Cells(1, 1), Cells(line_suu, 1)))
Range("E7") = Application.WorksheetFunction.StDevP(Range(Cells(1, 1), Cells(line_suu, 1)))
Range("H4") = Application.WorksheetFunction.Median(Range(Cells(1, 1), Cells(line_suu, 1)))
Range("H5") = Application.WorksheetFunction.Average(Range(Cells(1, 1), Cells(line_suu, 1)))

max_ = Application.WorksheetFunction.max(Range(Cells(1, 1), Cells(line_suu, 1)))
min_ = Application.WorksheetFunction.min(Range(Cells(1, 1), Cells(line_suu, 1)))
'-----
' (4) 「上限値－下限値」を40分割して、分類用の階層を作成する
Dim up_1, dw_1, step_1, n_1, ma_1 As Double

up_1 = Cells(3, 11).Value '上限値の読み込み
dw_1 = Cells(4, 11).Value '下限値の読み込み
step_1 = Cells(5, 11).Value '分割幅の読み込み (40分割している)

n_1 = 13
For ma_1 = dw_1 To (up_1 + step_1) Step step_1
    Cells(n_1, 4) = ma_1
    n_1 = n_1 + 1
Next

If (Cells(33, 4) < 0.0001) Then
    Cells(33, 4) = 0
End If
'-----
' (5) 各階層の出現頻度を計算する
Dim ii1, jj1 As Integer

For ii1 = 0 To 40
    Cells(13 + ii1, 5) = 0
    For jj1 = 1 To line_suu
        If (dw_1 + (step_1 * ii1) <= Cells(jj1, 1) And Cells(jj1, 1) < (dw_1 + (step_1 * (ii1 + 1)))) Then
            Cells(13 + ii1, 5) = Cells(13 + ii1, 5) + 1
        End If
    Next
Next

Next_Step_2:
' (6) セルに記載したデータをグラフ化する

```

```
'< A列 >
Dim Ch_1 As Object
Set Ch_1 = Worksheets("sheet1").ChartObjects.Add(Left:=300, Top:=150, Width:=600, Height:=300)

With Ch_1
    .Chart.ChartType = xlXYScatter
    .Chart.SetSourceData Source:=Sheets("Sheet1").Range(Cells(13, 4), Cells(13 + 40, 5)),
PlotBy:=xlColumns
    .Chart.Location Where:=xlLocationAsObject, Name:="Sheet1"
End With

With ActiveChart
    .HasTitle = True
    .ChartTitle.Text = Range("K7").Value
    .Axes(xlCategory, xlPrimary).HasTitle = True
    '.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Open-Close"
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = Range("K10").Value
    .Axes(xlValue, xlPrimary).HasTitle = True
    '.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "頻度"
    .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = Range("K9").Value
    '---Y軸のタイトルを縦書きに変更
    .Axes(xlValue, xlPrimary).AxisTitle.Orientation = xlVertical
    '---
End With

ActiveChart.ChartType = xlColumnClustered
'-----
End_
End Sub
'++++++
```

以上