```
Option Explicit

' ============================================================
' 端部ピーク解析（統合版 ver3.3）
' - 最大 50 ファイルを選択し一括解析
' - Result 集約（OK/ERROR を混在保存）
' - Charts シートに全プロファイル（最大 50）を自動配置
' - ver3.0 互換の堅牢 CSV 読み込み（BOM/区切り揺れ/小数点ロケール/欠損行）
' - Hist シートに hL/hR ヒストグラム（最新 OK run を自動検出）
' - Hist シートに 統計（Mean/Std/Min/Max/p95/p99）を自動出力（③のみ）
' ============================================================


' ============================================================
' Public Entry
' ============================================================
Public Sub RunEdgePeakAnalysis()
    On Error GoTo EH

    Application.ScreenUpdating = False
    Application.EnableEvents = False
    Application.Calculation = xlCalculationManual
    Application.DisplayStatusBar = True
    Application.StatusBar = False

    EnsureSheets
    PrepareChartsSheet

    Dim Lmm As Double, centerFrac As Double, binCount As Long
    Lmm = CDbl(GetConfigValue("Config", "B2", 15#))
    centerFrac = CDbl(GetConfigValue("Config", "B3", 0.1))
    binCount = CLng(GetConfigValue("Config", "B4", 20))

    If Lmm <= 0 Then Err.Raise vbObjectError + 1, , "L_mm が 0 以下です。Config!B2 を確認してください。"
    If centerFrac <= 0 Or centerFrac >= 0.5 Then Err.Raise vbObjectError + 2, , "CenterFrac が不正です(0～0.5)。Config!B3 を確認してください。"
    If binCount < 5 Then binCount = 5
    If binCount > 200 Then binCount = 200

    Dim files As Variant
    files = PickCsvFiles(50)
    If IsEmpty(files) Then GoTo CleanUp

    Dim runTime As Date
    runTime = Now

    Dim errList As Collection
    Set errList = New Collection

    Dim chartIndex As Long: chartIndex = 0

    Dim i As Long
    For i = LBound(files) To UBound(files)

        Dim filePath As String
        filePath = CStr(files(i))

        Application.StatusBar = "解析中: " & (i - LBound(files) + 1) & "/" & (UBound(files) - LBound(files) + 1) & "  " & Dir(filePath)
        DoEvents

        On Error GoTo FileEH

        Dim xArr() As Double, yArr() As Double
        ReadCsvXY filePath, xArr, yArr

        ' 読めない場合は Erase されて戻る
        If (Not Not xArr) = False Then Err.Raise vbObjectError + 2000, , "有効な数値データ行が見つかりません。"
        If UBound(xArr) < 3 Then Err.Raise vbObjectError + 2001, , "データ点数が不足しています。"

        QuickSortXY xArr, yArr, LBound(xArr), UBound(xArr)

        Dim xMin As Double, xMax As Double, width As Double, xMid As Double
        xMin = xArr(LBound(xArr))
        xMax = xArr(UBound(xArr))
        width = xMax - xMin
        If width <= 0 Then Err.Raise vbObjectError + 2002, , "x 幅が 0 以下です。"
        xMid = (xMin + xMax) / 2#
```

```
        Dim leftMin As Double, leftMax As Double, rightMin As Double, rightMax As Double
        leftMin = xMin
        leftMax = xMin + Lmm
        rightMin = xMax - Lmm
        rightMax = xMax

        Dim cMin As Double, cMax As Double
        cMin = xMid - centerFrac * width
        cMax = xMid + centerFrac * width

        Dim baseline As Double, baseCount As Long
        baseline = MeanYInRange(xArr, yArr, cMin, cMax, baseCount)
        If baseCount = 0 Then Err.Raise vbObjectError + 2003, , "baseline 計算範囲に点がありません。"
        If Abs(baseline) < 0.0000001 Then Err.Raise vbObjectError + 2004, , "baseline が 0 に近く除算できません。"

        Dim xL As Double, yL As Double, xR As Double, yR As Double
        If Not MaxYInRange(xArr, yArr, leftMin, leftMax, xL, yL) Then Err.Raise vbObjectError + 2005, , "左端部範囲に点がありません。"
        If Not MaxYInRange(xArr, yArr, rightMin, rightMax, xR, yR) Then Err.Raise vbObjectError + 2006, , "右端部範囲に点がありません。"

        Dim hL As Double, hR As Double
        hL = (yL - baseline) / baseline
        hR = (yR - baseline) / baseline

        AppendResultEx runTime, filePath, Lmm, centerFrac, baseline, xL, yL, hL, xR, yR, hR, "OK", ""

        chartIndex = chartIndex + 1
        AddProfileChartToChartsSheet chartIndex, Dir(filePath), xArr, yArr, baseline, xL, yL, xR, yR, Lmm, centerFrac

        On Error GoTo 0
        GoTo NextFile

FileEH:
        AppendResultEx runTime, filePath, Lmm, centerFrac, Empty, Empty, Empty, Empty, Empty, Empty, Empty, "ERROR", Err.Description
        errList.Add Dir(filePath) & " : " & Err.Description
        Err.Clear
        On Error GoTo 0

NextFile:
    Next i

    Application.StatusBar = False

    ' ---- Histogram + Stats (Latest OK Run) ----
    BuildHLHRHistogramsLatest binCount

    If errList.count > 0 Then
        Dim msg As String, k As Long
        msg = "完了（ただし一部 ERROR はスキップして続行）:" & vbCrLf & vbCrLf
        For k = 1 To errList.count
            msg = msg & " - " & errList(k) & vbCrLf
        Next k
        MsgBox msg, vbExclamation, "端部ピーク解析（バッチ）"
    Else
        MsgBox "完了しました（" & (UBound(files) - LBound(files) + 1) & "件）。" & vbCrLf & _
               "Hist シートに hL/hR ヒストグラムと統計（Mean/Std/p95/p99 等）を作成しました。", _
               vbInformation, "端部ピーク解析（バッチ）"
    End If

CleanUp:
    Application.StatusBar = False
    Application.ScreenUpdating = True
    Application.EnableEvents = True
    Application.Calculation = xlCalculationAutomatic
    Exit Sub

EH:
    Application.StatusBar = False
    Application.ScreenUpdating = True
    Application.EnableEvents = True
    Application.Calculation = xlCalculationAutomatic
    MsgBox "エラー: " & Err.Description, vbExclamation, "端部ピーク解析"
End Sub


' 手動で Hist だけ作り直したい場合
Public Sub RebuildHLHRHistogramLatest()
    Dim binCount As Long
```

```
    binCount = CLng(GetConfigValue("Config", "B4", 20))
    BuildHLHRHistogramsLatest binCount
End Sub


' ============================================================
' Sheets / Config / Result
' ============================================================
Private Sub EnsureSheets()
    EnsureSheetExists "Config"
    EnsureSheetExists "Result"
    EnsureSheetExists "Charts"
    EnsureSheetExists "Hist"

    ' Result header（初回のみ）
    With ThisWorkbook.Worksheets("Result")
        If .Cells(1, 1).Value = "" Then
            .Range("A1").Value = "Datetime"
            .Range("B1").Value = "File"
            .Range("C1").Value = "L_mm"
            .Range("D1").Value = "CenterFrac"
            .Range("E1").Value = "Baseline_um"
            .Range("F1").Value = "x_L_mm"
            .Range("G1").Value = "yPeak_L_um"
            .Range("H1").Value = "h_L_(y-baseline)/baseline"
            .Range("I1").Value = "x_R_mm"
            .Range("J1").Value = "yPeak_R_um"
            .Range("K1").Value = "h_R_(y-baseline)/baseline"
            .Range("L1").Value = "Status"
            .Range("M1").Value = "Error"
            .Range("A1:M1").Font.Bold = True
            .Columns("A:M").AutoFit
        End If
    End With

    ' Config defaults（初回のみ）
    With ThisWorkbook.Worksheets("Config")
        If .Range("A1").Value = "" Then
            .Range("A1").Value = "Parameter"
            .Range("B1").Value = "Value"
            .Range("A1:B1").Font.Bold = True
        End If

        If .Range("A2").Value = "" Then .Range("A2").Value = "L_mm"
        If .Range("B2").Value = "" Then .Range("B2").Value = 15

        If .Range("A3").Value = "" Then .Range("A3").Value = "CenterFrac"
        If .Range("B3").Value = "" Then .Range("B3").Value = 0.1

        If .Range("A4").Value = "" Then .Range("A4").Value = "Hist_BinCount"
        If .Range("B4").Value = "" Then .Range("B4").Value = 20

        .Columns("A:B").AutoFit
    End With
End Sub

Private Sub EnsureSheetExists(ByVal sheetName As String)
    Dim ws As Worksheet
    On Error Resume Next
    Set ws = ThisWorkbook.Worksheets(sheetName)
    On Error GoTo 0
    If ws Is Nothing Then
        ThisWorkbook.Worksheets.Add(After:=ThisWorkbook.Worksheets(ThisWorkbook.Worksheets.count)).name = sheetName
    End If
End Sub

Private Function GetConfigValue(ByVal sheetName As String, ByVal addr As String, ByVal defaultValue As Variant) As Variant
    On Error GoTo UseDefault
    Dim v As Variant
    v = ThisWorkbook.Worksheets(sheetName).Range(addr).Value
    If IsEmpty(v) Or v = "" Then
        GetConfigValue = defaultValue
    Else
        GetConfigValue = v
    End If
    Exit Function
UseDefault:
```

```
    GetConfigValue = defaultValue
End Function

Private Sub AppendResultEx( _
    ByVal runTime As Date, _
    ByVal filePath As String, _
    ByVal Lmm As Double, ByVal centerFrac As Double, _
    ByVal baseline As Variant, _
    ByVal xL As Variant, ByVal yL As Variant, ByVal hL As Variant, _
    ByVal xR As Variant, ByVal yR As Variant, ByVal hR As Variant, _
    ByVal status As String, ByVal errMsg As String)

    Dim ws As Worksheet: Set ws = ThisWorkbook.Worksheets("Result")
    Dim r As Long
    r = ws.Cells(ws.Rows.count, 1).End(xlUp).row + 1

    ws.Cells(r, 1).Value = runTime
    ws.Cells(r, 2).Value = Dir(filePath)
    ws.Cells(r, 3).Value = Lmm
    ws.Cells(r, 4).Value = centerFrac

    If status = "OK" Then
        ws.Cells(r, 5).Value = baseline
        ws.Cells(r, 6).Value = xL
        ws.Cells(r, 7).Value = yL
        ws.Cells(r, 8).Value = hL
        ws.Cells(r, 9).Value = xR
        ws.Cells(r, 10).Value = yR
        ws.Cells(r, 11).Value = hR
    Else
        ws.Range(ws.Cells(r, 5), ws.Cells(r, 11)).ClearContents
    End If

    ws.Cells(r, 12).Value = status
    ws.Cells(r, 13).Value = errMsg
End Sub


' ============================================================
' Charts (All Profiles Output)
' ============================================================
Private Sub PrepareChartsSheet()
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Worksheets("Charts")

    ws.Cells.ClearContents

    Dim i As Long
    For i = ws.ChartObjects.count To 1 Step -1
        ws.ChartObjects(i).Delete
    Next i

    ws.Range("A1").Value = "All Profiles (max 50)"
    ws.Range("A1").Font.Bold = True
End Sub

Private Sub AddProfileChartToChartsSheet( _
    ByVal chartIndex As Long, ByVal fileName As String, _
    ByRef xArr() As Double, ByRef yArr() As Double, _
    ByVal baseline As Double, _
    ByVal xL As Double, ByVal yL As Double, _
    ByVal xR As Double, ByVal yR As Double, _
    ByVal Lmm As Double, ByVal centerFrac As Double)

    Dim ws As Worksheet
    Set ws = ThisWorkbook.Worksheets("Charts")

    Const COLS As Long = 5
    Const chartW As Double = 300
    Const chartH As Double = 220
    Const marginL As Double = 10
    Const marginT As Double = 30
    Const gapX As Double = 8
    Const gapY As Double = 8

    Dim col As Long, row As Long
    col = (chartIndex - 1) Mod COLS
```

```
    row = (chartIndex - 1) ¥ COLS

    Dim leftPos As Double, topPos As Double
    leftPos = marginL + col * (chartW + gapX)
    topPos = marginT + row * (chartH + gapY)

    Dim co As ChartObject
    Set co = ws.ChartObjects.Add(Left:=leftPos, Top:=topPos, width:=chartW, Height:=chartH)

    Dim ch As Chart
    Set ch = co.Chart
    ch.ChartType = xlXYScatterLinesNoMarkers
    ch.HasTitle = True
    ch.ChartTitle.Text = fileName & "   L=" & Format(Lmm, "0.0") & "  center±" & Format(centerFrac, "0.00")

    Do While ch.SeriesCollection.count > 0
        ch.SeriesCollection(1).Delete
    Loop

    ' Profile
    Dim s1 As Series
    Set s1 = ch.SeriesCollection.NewSeries
    s1.name = "Profile"
    s1.XValues = xArr
    s1.Values = yArr

    ' Baseline
    Dim bx(1 To 2) As Double, by(1 To 2) As Double
    bx(1) = xArr(LBound(xArr))
    bx(2) = xArr(UBound(xArr))
    by(1) = baseline
    by(2) = baseline

    Dim sb As Series
    Set sb = ch.SeriesCollection.NewSeries
    sb.name = "Baseline"
    sb.XValues = bx
    sb.Values = by
    sb.ChartType = xlXYScatterLinesNoMarkers

    ' Left Peak marker
    Dim sl As Series
    Set sl = ch.SeriesCollection.NewSeries
    sl.name = "LeftPeak"
    sl.XValues = Array(xL)
    sl.Values = Array(yL)
    sl.ChartType = xlXYScatter
    sl.MarkerStyle = xlMarkerStyleCircle
    sl.MarkerSize = 5

    ' Right Peak marker
    Dim sr As Series
    Set sr = ch.SeriesCollection.NewSeries
    sr.name = "RightPeak"
    sr.XValues = Array(xR)
    sr.Values = Array(yR)
    sr.ChartType = xlXYScatter
    sr.MarkerStyle = xlMarkerStyleCircle
    sr.MarkerSize = 5
End Sub


' ============================================================
' Robust CSV Read (ver3.0 compatible)
' ============================================================
Private Sub ReadCsvXY(ByVal filePath As String, ByRef xArr() As Double, ByRef yArr() As Double)
    Dim txt As String
    txt = ReadAllTextRobust(filePath)

    txt = Replace(txt, vbCrLf, vbLf)
    txt = Replace(txt, vbCr, vbLf)

    Dim lines() As String
    lines = Split(txt, vbLf)

    Dim cap As Long: cap = 1024
    Dim xs() As Double, ys() As Double
```

```
    ReDim xs(1 To cap)
    ReDim ys(1 To cap)

    Dim n As Long: n = 0
    Dim i As Long

    Dim headerSkipped As Boolean: headerSkipped = False

    For i = LBound(lines) To UBound(lines)
        Dim line As String
        line = Trim$(lines(i))
        line = Replace(line, ChrW(&HFEFF), "") ' BOM
        line = Replace(line, ChrW(&H200B), "") ' zero-width
        If Len(line) = 0 Then GoTo ContinueFor

        ' 1 行目はヘッダとしてスキップ
        If Not headerSkipped Then
            headerSkipped = True
            GoTo ContinueFor
        End If

        Dim xVal As Double, yVal As Double
        If TryParseXY(line, xVal, yVal) Then
            n = n + 1
            If n > cap Then
                cap = cap * 2
                ReDim Preserve xs(1 To cap)
                ReDim Preserve ys(1 To cap)
            End If
            xs(n) = xVal
            ys(n) = yVal
        End If

ContinueFor:
    Next i

    If n = 0 Then
        Erase xArr
        Erase yArr
        Exit Sub
    End If

    ReDim Preserve xs(1 To n)
    ReDim Preserve ys(1 To n)

    xArr = xs
    yArr = ys
End Sub

Private Function ReadAllTextRobust(ByVal filePath As String) As String
    On Error GoTo FallbackBinary

    Dim stm As Object
    Set stm = CreateObject("ADODB.Stream")
    stm.Type = 2
    stm.Charset = "utf-8"
    stm.Open
    stm.LoadFromFile filePath
    ReadAllTextRobust = stm.ReadText(-1)
    stm.Close
    Exit Function

FallbackBinary:
    On Error GoTo EH

    Dim f As Integer: f = FreeFile
    Open filePath For Binary Access Read As #f
    Dim bytes() As Byte
    If LOF(f) = 0 Then
        Close #f
        ReadAllTextRobust = ""
        Exit Function
    End If
    ReDim bytes(0 To LOF(f) - 1)
    Get #f, , bytes
    Close #f
```

```
    ReadAllTextRobust = StrConv(bytes, vbUnicode)
    Exit Function

EH:
    Err.Raise vbObjectError + 999, , "ファイル読み込みに失敗しました: " & filePath
End Function

Private Function TryParseXY(ByVal line As String, ByRef xOut As Double, ByRef yOut As Double) As Boolean
    line = Replace(line, ",  ", ",")
    line = Replace(line, " ; ", ";")

    Dim delims As Variant
    delims = Array(",", ";", vbTab)

    Dim d As Variant
    For Each d In delims
        Dim parts() As String
        parts = Split(line, CStr(d))
        If UBound(parts) >= 1 Then
            Dim sx As String, sy As String
            sx = Trim$(parts(0))
            sy = Trim$(parts(1))

            Dim xv As Double, yv As Double
            If TryParseDoubleLocale(sx, xv) And TryParseDoubleLocale(sy, yv) Then
                xOut = xv
                yOut = yv
                TryParseXY = True
                Exit Function
            End If
        End If
    Next d

    TryParseXY = False
End Function

Private Function TryParseDoubleLocale(ByVal s As String, ByRef vOut As Double) As Boolean
    s = Trim$(s)
    s = Replace(s, ChrW(&HFEFF), "")
    s = Replace(s, ChrW(&H200B), "")

    Dim decSep As String
    decSep = Application.International(xlDecimalSeparator)

    If decSep = "," Then
        If InStr(s, ".") > 0 And InStr(s, ",") = 0 Then s = Replace(s, ".", ",")
    ElseIf decSep = "." Then
        If InStr(s, ",") > 0 And InStr(s, ".") = 0 Then s = Replace(s, ",", ".")
    End If

    If IsNumeric(s) Then
        vOut = CDbl(s)
        TryParseDoubleLocale = True
    Else
        TryParseDoubleLocale = False
    End If
End Function


' ============================================================
' Math Helpers
' ============================================================
Private Function MeanYInRange(ByRef xArr() As Double, ByRef yArr() As Double, _
                              ByVal xMin As Double, ByVal xMax As Double, _
                              ByRef count As Long) As Double
    Dim i As Long
    Dim s As Double: s = 0#
    count = 0

    For i = LBound(xArr) To UBound(xArr)
        If xArr(i) >= xMin And xArr(i) <= xMax Then
            s = s + yArr(i)
            count = count + 1
        End If
    Next i

    If count = 0 Then
```

```
        MeanYInRange = 0#
    Else
        MeanYInRange = s / count
    End If
End Function

Private Function MaxYInRange(ByRef xArr() As Double, ByRef yArr() As Double, _
                             ByVal xMin As Double, ByVal xMax As Double, _
                             ByRef xAtMax As Double, ByRef yMax As Double) As Boolean
    Dim i As Long
    Dim found As Boolean: found = False
    Dim bestY As Double: bestY = -1E+99
    Dim bestX As Double: bestX = 0#

    For i = LBound(xArr) To UBound(xArr)
        If xArr(i) >= xMin And xArr(i) <= xMax Then
            If (Not found) Or (yArr(i) > bestY) Then
                bestY = yArr(i)
                bestX = xArr(i)
                found = True
            End If
        End If
    Next i

    If found Then
        xAtMax = bestX
        yMax = bestY
        MaxYInRange = True
    Else
        MaxYInRange = False
    End If
End Function


' ============================================================
' Sort (x ascending, keep y aligned) - Double()専用
' ============================================================
Private Sub QuickSortXY(ByRef xArr() As Double, ByRef yArr() As Double, ByVal lo As Long, ByVal hi As Long)
    Dim i As Long, j As Long
    Dim pivot As Double
    Dim tmpX As Double, tmpY As Double

    i = lo
    j = hi
    pivot = xArr((lo + hi) ¥ 2)

    Do While i <= j
        Do While xArr(i) < pivot
            i = i + 1
        Loop
        Do While xArr(j) > pivot
            j = j - 1
        Loop

        If i <= j Then
            tmpX = xArr(i): xArr(i) = xArr(j): xArr(j) = tmpX
            tmpY = yArr(i): yArr(i) = yArr(j): yArr(j) = tmpY
            i = i + 1
            j = j - 1
        End If
    Loop

    If lo < j Then QuickSortXY xArr, yArr, lo, j
    If i < hi Then QuickSortXY xArr, yArr, i, hi
End Sub


' ============================================================
' File Picker (Multi) - return String()
' ============================================================
Private Function PickCsvFiles(ByVal maxFiles As Long) As Variant
    Dim fd As FileDialog
    Set fd = Application.FileDialog(msoFileDialogFilePicker)

    With fd
        .title = "CSV ファイルを選択（最大" & maxFiles & "件）"
        .AllowMultiSelect = True
```

```
        .Filters.Clear
        .Filters.Add "CSV/TXT", "*.csv;*.txt", 1

        If .Show <> -1 Then
            PickCsvFiles = Empty
            Exit Function
        End If

        If .SelectedItems.count > maxFiles Then
            Err.Raise vbObjectError + 100, , "選択ファイル数が上限(" & maxFiles & ")を超えています。選び直してください。"
        End If

        Dim arr() As String
        Dim i As Long
        ReDim arr(1 To .SelectedItems.count)

        For i = 1 To .SelectedItems.count
            arr(i) = .SelectedItems(i)
        Next i

        PickCsvFiles = arr
    End With
End Function


' ============================================================
' Histogram + Stats (hL/hR) - Latest OK Run Auto-Detect
' ============================================================
Private Sub BuildHLHRHistogramsLatest(ByVal binCount As Long)
    On Error GoTo EH

    If binCount < 5 Then binCount = 5
    If binCount > 200 Then binCount = 200

    Dim wsR As Worksheet: Set wsR = ThisWorkbook.Worksheets("Result")
    Dim wsH As Worksheet: Set wsH = ThisWorkbook.Worksheets("Hist")
    PrepareHistSheet wsH

    Dim lastRow As Long
    lastRow = wsR.Cells(wsR.Rows.count, 1).End(xlUp).row
    If lastRow < 2 Then
        wsH.Range("A3").Value = "Result にデータがありません。"
        Exit Sub
    End If

    ' ---- latest OK runTime ----
    Dim latestRun As Date: latestRun = 0
    Dim r As Long
    For r = 2 To lastRow
        If IsDate(wsR.Cells(r, 1).Value) Then
            If CStr(wsR.Cells(r, 12).Value) = "OK" Then
                If CDate(wsR.Cells(r, 1).Value) > latestRun Then
                    latestRun = CDate(wsR.Cells(r, 1).Value)
                End If
            End If
        End If
    Next r

    If latestRun = 0 Then
        wsH.Range("A3").Value = "OK データが見つかりません（Result の Status 列を確認）。"
        Exit Sub
    End If

    ' ---- collect hL/hR ----
    Dim hL() As Double, hR() As Double
    Dim n As Long: n = 0

    For r = 2 To lastRow
        If IsDate(wsR.Cells(r, 1).Value) Then
            If CDate(wsR.Cells(r, 1).Value) = latestRun Then
                If CStr(wsR.Cells(r, 12).Value) = "OK" Then
                    If IsNumeric(wsR.Cells(r, 8).Value) And IsNumeric(wsR.Cells(r, 11).Value) Then
                        n = n + 1
                        ReDim Preserve hL(1 To n)
                        ReDim Preserve hR(1 To n)
                        hL(n) = CDbl(wsR.Cells(r, 8).Value)   ' H
                        hR(n) = CDbl(wsR.Cells(r, 11).Value)  ' K
```

```
            End If
          End If
        End If
    End If
Next r

If n = 0 Then
    wsH.Range("A3").Value = "最新 runTime の OK データはあるが、hL/hR が数値として取得できません。"
    Exit Sub
End If

' ---- min/max（hL+hR 全体）----
Dim minV As Double, maxV As Double, i As Long
minV = hL(1): maxV = hL(1)
For i = 1 To n
    If hL(i) < minV Then minV = hL(i)
    If hL(i) > maxV Then maxV = hL(i)
    If hR(i) < minV Then minV = hR(i)
    If hR(i) > maxV Then maxV = hR(i)
Next i

Dim w As Double: w = maxV - minV
If w <= 0 Then w = 0.01

Dim binW As Double: binW = w / binCount

' ---- bins (upper bound) ----
Dim bins() As Double
ReDim bins(1 To binCount)
For i = 1 To binCount
    bins(i) = minV + binW * i
Next i

' ---- Frequency ----
Dim vHL As Variant, vHR As Variant, vBins As Variant
vHL = ToVariant1D_FromDoubleArrayHLHR(hL)
vHR = ToVariant1D_FromDoubleArrayHLHR(hR)
vBins = ToVariant1D_FromDoubleArrayHLHR(bins)

Dim freqL As Variant, freqR As Variant
freqL = WorksheetFunction.Frequency(vHL, vBins)
freqR = WorksheetFunction.Frequency(vHR, vBins)

' Frequency の次元判定（2D なら (i,1)、1D なら (i)）
Dim is2DL As Boolean, is2DR As Boolean
On Error Resume Next
Dim t As Long
t = LBound(freqL, 2): is2DL = (Err.Number = 0)
Err.Clear
t = LBound(freqR, 2): is2DR = (Err.Number = 0)
Err.Clear
On Error GoTo EH

' ---- table ----
wsH.Range("A1").Value = "hL / hR Histogram (Latest OK Run)"
wsH.Range("A2").Value = "BinUpper"
wsH.Range("B2").Value = "Count_hL"
wsH.Range("C2").Value = "Count_hR"

wsH.Range("E2").Value = "RunTime"
wsH.Range("F2").Value = latestRun
wsH.Range("E3").Value = "N"
wsH.Range("F3").Value = n
wsH.Range("E4").Value = "Min"
wsH.Range("F4").Value = minV
wsH.Range("E5").Value = "Max"
wsH.Range("F5").Value = maxV
wsH.Range("E6").Value = "BinCount"
wsH.Range("F6").Value = binCount
wsH.Range("E7").Value = "BinWidth"
wsH.Range("F7").Value = binW

wsH.Range("A1").Font.Bold = True
wsH.Range("A2:C2").Font.Bold = True
wsH.Range("E2:E7").Font.Bold = True

Dim startRow As Long: startRow = 3
```

```
    For i = 1 To binCount
        wsH.Cells(startRow + i - 1, 1).Value = bins(i)

        If is2DL Then
            wsH.Cells(startRow + i - 1, 2).Value = CLng(freqL(i, 1))
        Else
            wsH.Cells(startRow + i - 1, 2).Value = CLng(freqL(i))
        End If

        If is2DR Then
            wsH.Cells(startRow + i - 1, 3).Value = CLng(freqR(i, 1))
        Else
            wsH.Cells(startRow + i - 1, 3).Value = CLng(freqR(i))
        End If
    Next i

    wsH.Columns("A:C").NumberFormat = "0.0000"
    wsH.Columns("B:C").NumberFormat = "0"
    wsH.Columns("A:C").AutoFit
    wsH.Columns("E:F").AutoFit

    ' ---- Stats (③のみ) ----
    Dim mL As Double, sdL As Double, mnL As Double, mxL As Double, p95L As Double, p99L As Double
    Dim mR As Double, sdR As Double, mnR As Double, mxR As Double, p95R As Double, p99R As Double

    CalcStatsHLHR hL, mL, sdL, mnL, mxL, p95L, p99L
    CalcStatsHLHR hR, mR, sdR, mnR, mxR, p95R, p99R

    wsH.Range("H2").Value = "Stats_hL"
    wsH.Range("J2").Value = "Stats_hR"
    wsH.Range("H2").Font.Bold = True
    wsH.Range("J2").Font.Bold = True

    wsH.Range("H3").Value = "Mean": wsH.Range("I3").Value = mL
    wsH.Range("H4").Value = "Std":  wsH.Range("I4").Value = sdL
    wsH.Range("H5").Value = "Min":  wsH.Range("I5").Value = mnL
    wsH.Range("H6").Value = "Max":  wsH.Range("I6").Value = mxL
    wsH.Range("H7").Value = "p95":  wsH.Range("I7").Value = p95L
    wsH.Range("H8").Value = "p99":  wsH.Range("I8").Value = p99L

    wsH.Range("J3").Value = "Mean": wsH.Range("K3").Value = mR
    wsH.Range("J4").Value = "Std":  wsH.Range("K4").Value = sdR
    wsH.Range("J5").Value = "Min":  wsH.Range("K5").Value = mnR
    wsH.Range("J6").Value = "Max":  wsH.Range("K6").Value = mxR
    wsH.Range("J7").Value = "p95":  wsH.Range("K7").Value = p95R
    wsH.Range("J8").Value = "p99":  wsH.Range("K8").Value = p99R

    wsH.Range("H3:H8").Font.Bold = True
    wsH.Range("J3:J8").Font.Bold = True
    wsH.Range("I3:I8").NumberFormat = "0.000000"
    wsH.Range("K3:K8").NumberFormat = "0.000000"
    wsH.Columns("H:K").AutoFit

    ' ---- Charts ----
    BuildHistogramChart wsH, "Hist_hL", "hL Histogram", _
        wsH.Range(wsH.Cells(startRow, 1), wsH.Cells(startRow + binCount - 1, 1)), _
        wsH.Range(wsH.Cells(startRow, 2), wsH.Cells(startRow + binCount - 1, 2)), _
        10, 140, 520, 300

    BuildHistogramChart wsH, "Hist_hR", "hR Histogram", _
        wsH.Range(wsH.Cells(startRow, 1), wsH.Cells(startRow + binCount - 1, 1)), _
        wsH.Range(wsH.Cells(startRow, 3), wsH.Cells(startRow + binCount - 1, 3)), _
        10, 460, 520, 300

    Exit Sub

EH:
    MsgBox "Histogram 作成エラー: " & Err.Description, vbExclamation, "Histogram"
End Sub

Private Sub PrepareHistSheet(ByVal ws As Worksheet)
    ws.Cells.ClearContents

    Dim i As Long
    For i = ws.ChartObjects.count To 1 Step -1
        ws.ChartObjects(i).Delete
    Next i
```

```
    ws.Range("A1").Value = "hL / hR Histogram"
    ws.Range("A1").Font.Bold = True
End Sub

Private Sub BuildHistogramChart( _
    ByVal ws As Worksheet, _
    ByVal chartName As String, _
    ByVal title As String, _
    ByVal rngX As Range, _
    ByVal rngY As Range, _
    ByVal leftPos As Double, ByVal topPos As Double, ByVal w As Double, ByVal h As Double)

    Dim co As ChartObject
    Set co = ws.ChartObjects.Add(Left:=leftPos, Top:=topPos, width:=w, Height:=h)
    co.name = chartName

    Dim ch As Chart
    Set ch = co.Chart
    ch.ChartType = xlColumnClustered
    ch.HasTitle = True
    ch.ChartTitle.Text = title

    Do While ch.SeriesCollection.count > 0
        ch.SeriesCollection(1).Delete
    Loop

    Dim s As Series
    Set s = ch.SeriesCollection.NewSeries
    s.name = title
    s.XValues = rngX
    s.Values = rngY

    ch.Axes(xlCategory).HasTitle = True
    ch.Axes(xlCategory).AxisTitle.Text = "Bin Upper (h)"
    ch.Axes(xlValue).HasTitle = True
    ch.Axes(xlValue).AxisTitle.Text = "Count"
End Sub

Private Function ToVariant1D_FromDoubleArrayHLHR(ByRef a() As Double) As Variant
    Dim n As Long: n = UBound(a) - LBound(a) + 1
    Dim v() As Variant
    ReDim v(1 To n)
    Dim i As Long, k As Long: k = 0
    For i = LBound(a) To UBound(a)
        k = k + 1
        v(k) = a(i)
    Next i
    ToVariant1D_FromDoubleArrayHLHR = v
End Function


' ============================================================
' Stats (③のみ) ※名前衝突回避で HLHR 接尾辞
' ============================================================
Public Sub CalcStatsHLHR(ByRef a() As Double, _
                         ByRef meanOut As Double, _
                         ByRef stdOut As Double, _
                         ByRef minOut As Double, _
                         ByRef maxOut As Double, _
                         ByRef p95Out As Double, _
                         ByRef p99Out As Double)

    Dim n As Long: n = UBound(a) - LBound(a) + 1
    If n <= 0 Then
        meanOut = 0#: stdOut = 0#: minOut = 0#: maxOut = 0#: p95Out = 0#: p99Out = 0#
        Exit Sub
    End If

    Dim i As Long
    minOut = a(LBound(a))
    maxOut = a(LBound(a))

    Dim s As Double: s = 0#
    For i = LBound(a) To UBound(a)
        s = s + a(i)
        If a(i) < minOut Then minOut = a(i)
```

```
        If a(i) > maxOut Then maxOut = a(i)
    Next i
    meanOut = s / n

    ' 標本標準偏差（n-1）
    If n = 1 Then
        stdOut = 0#
    Else
        Dim ss As Double: ss = 0#
        For i = LBound(a) To UBound(a)
            ss = ss + (a(i) - meanOut) * (a(i) - meanOut)
        Next i
        stdOut = Sqr(ss / (n - 1))
    End If

    ' p95/p99（最近傍 : ceil(p*n)）
    Dim b() As Double
    b = CopyDoubleArrayHLHR(a)
    QuickSort1DHLHR b, LBound(b), UBound(b)

    p95Out = PercentileNearestHLHR(b, 0.95)
    p99Out = PercentileNearestHLHR(b, 0.99)
End Sub

Private Function CopyDoubleArrayHLHR(ByRef a() As Double) As Double()
    Dim b() As Double
    Dim i As Long
    ReDim b(LBound(a) To UBound(a))
    For i = LBound(a) To UBound(a)
        b(i) = a(i)
    Next i
    CopyDoubleArrayHLHR = b
End Function

Private Sub QuickSort1DHLHR(ByRef a() As Double, ByVal lo As Long, ByVal hi As Long)
    Dim i As Long, j As Long
    Dim pivot As Double, tmp As Double
    i = lo: j = hi
    pivot = a((lo + hi) ¥ 2)

    Do While i <= j
        Do While a(i) < pivot: i = i + 1: Loop
        Do While a(j) > pivot: j = j - 1: Loop
        If i <= j Then
            tmp = a(i): a(i) = a(j): a(j) = tmp
            i = i + 1: j = j - 1
        End If
    Loop

    If lo < j Then QuickSort1DHLHR a, lo, j
    If i < hi Then QuickSort1DHLHR a, i, hi
End Sub

Private Function PercentileNearestHLHR(ByRef aSorted() As Double, ByVal p As Double) As Double
    Dim n As Long: n = UBound(aSorted) - LBound(aSorted) + 1
    If n <= 0 Then PercentileNearestHLHR = 0#: Exit Function
    If p <= 0 Then PercentileNearestHLHR = aSorted(LBound(aSorted)): Exit Function
    If p >= 1 Then PercentileNearestHLHR = aSorted(UBound(aSorted)): Exit Function

    Dim idx As Long
    idx = CLng(Application.WorksheetFunction.RoundUp(p * n, 0)) ' 1..n
    If idx < 1 Then idx = 1
    If idx > n Then idx = n

    PercentileNearestHLHR = aSorted(LBound(aSorted) + idx - 1)
End Function
```