

CONTENTS

サンプルの Experiment – BoxPlot Demo	2
クイックノート	2
手順（デモを再現する手順）	2
手順（メニューを使った場合）	3
statsBoxPlot 関数の内容.....	4
(参考) StatsQuantiles コマンドの実行.....	5
(参考) StatsQuantiles コマンドのヘルプ	6

サンプルの Experiment – BoxPlot Demo

※このデモは古い実装方法 (Igor Pro 7 以前のもの) を使っていることに注意してください。Igor Pro 8 以降での作成は、ビジュアル解説の「II-3 Graphs : グラフ編」を参照してください。

クイックノート

メニュー **File** → **Example Experiments** → **Statistics** → **BoxPlot Demo**

Box Plot (ボックスプロット) はメニューを使っても作成できます。

このデモではあらかじめ作成されたプロシージャを使ってより複雑なプロットを作成しています。

プロットのプロシージャは、StatsPlots.ipf ファイルに記載されています。

これを使用するには、そのファイルをプロシージャウィンドウに含めるか、または、AllStatsProcedures を以下のようにして読み込みます。

```
#include <AllStatsProcedures>
```

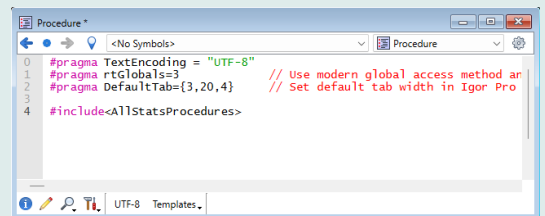
手順 (デモを再現する手順)

新しい Experiment を作成したところからの手順で確認します。

1. メニュー Windows → Procedure Windows → Procedure Window を選択します。

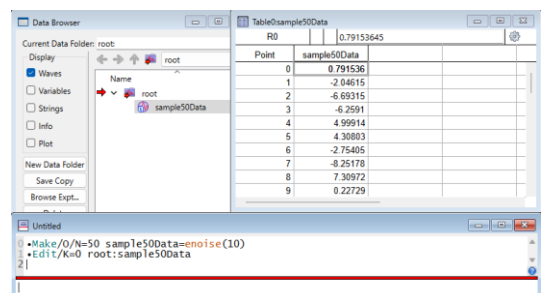
```
#include<AllStatsProcedures>
```

という行を追加して、ウィンドウ下の **Compile** ボタンをクリックします。



2. コマンドウィンドウで次を実行して、ランダムな値を持つウェーブを作ります。

```
Make/O/N=50 sample50Data=noise(10)
```



3. コマンドウィンドウで次を実行して、読み込んだプロシージャファイル内の関数を呼び出して、Box Plot を作成します。

```
statsBoxPlot(sample50Data)
```

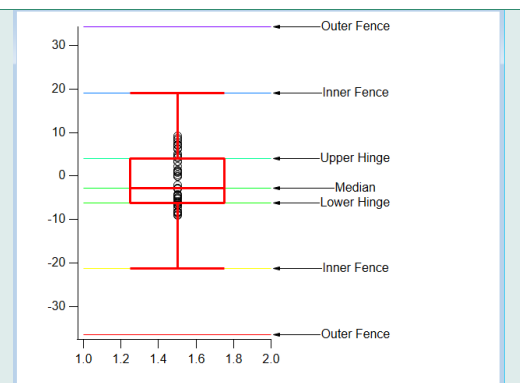
表示されている注釈は次のようになります：

Lower Hinge : 25 パーセンタイル (Q25)

Upper Hinge : 75 パーセンタイル (Q75)

ヒンジ間の距離または H スプレッドの距離は $H=Q75-Q25$ です。

ステップは次のように定義されます： $step=1.5 * H$



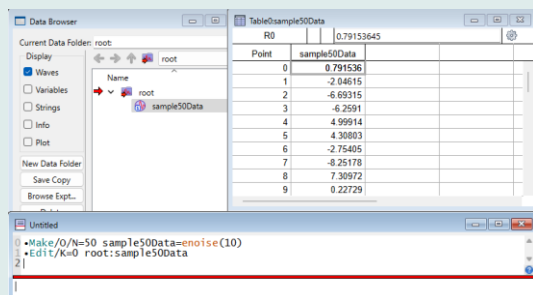
上の Inner Fence : $Q75 + \text{step}$
 上の Outer Fence : $Q75 + 2 * \text{step}$
 下の Inner Fence : $Q75 - \text{step}$
 下の Outer Fence : $Q75 - 2 * \text{step}$

手順 (メニューを使った場合)

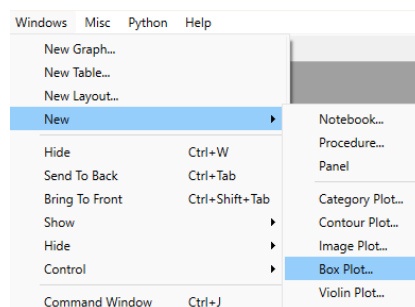
新しい Experiment を作成したところからの手順で確認します。

1. コマンドウィンドウで次を実行して、ランダムな値を持つウェーブを作ります。

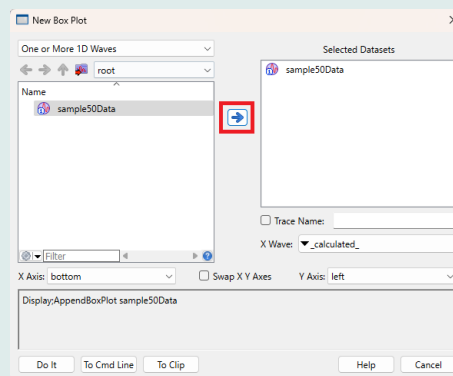
```
Make/O/N=50 sample50Data=noise(10)
```



2. メニュー Windows → New → Box Plot を選択します。

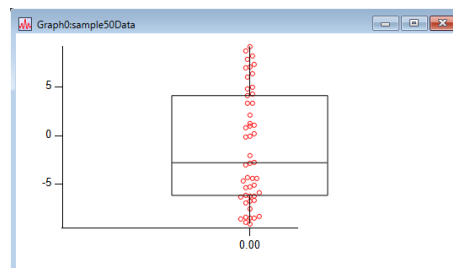


3. New Box Plot ダイアログで sample50Data ウェーブを選択し、中央の矢印で右に移動します。



4. Do It をクリックすると、右図のようなグラフが作成されます。

メニューからの操作の場合、この後、プロシージャ内で定義しているような様々な設定を手動で行う必要があります。



statsBoxPlot 関数の内容

```
//=====
// 以下の関数は、1つのボックスプロットを作成します。
// この例は、StatsQuantiles を使用するほどエレガントではないSortを使っていますが、
// StatsQuantiles が開発される前に書かれたものであり、問題がないためそのまま使うことにしました。
// 2006/10/13 グラフウィンドウの名前を返すように変更しました。

constant kStartFence=1
constant kStartx=1.25
constant kEndX=1.75
constant kEndFence=2
constant kMidX=1.5

Function/S statsBoxPlot(inWave)
  Wave inWave
  String outWinName=""
  String folderName=UniqueName(NameOfWave(inWave), 11, 0)
  String oldDF=GetDataFolder(1)
  SetDataFolder root:
  NewDataFolder/O/S Packages
  NewDataFolder/O/S BoxPlot
  NewDataFolder/O/S $folderName

  Duplicate/O inWave, tmpWave
  Sort tmpWave,tmpWave
  Variable N=DimSize(tmpWave,0)
  Variable theMedian=tmpWave((N-1)/2)
  Variable Q25=tmpWave((N-1)/4)
  Variable Q75=tmpWave(3*(N-1)/4)
  Variable H=Q75-Q25
  Variable step=1.5*H
  Variable upperInnerFence=Q75+step
  Variable upperOuterFence=upperInnerFence+step
  Variable lowerInnerFence=Q25-step
  Variable lowerOuterFence=lowerInnerFence-step

  Make/O/N=(21,2) fencesWave=NaN
  fencesWave[][0]=mod(p,3)==0 ? kStartFence : kEndFence
  fencesWave[][0]=mod(p+1,3)==0 ? NaN : fencesWave
  fencesWave[0,2][1]=lowerOuterFence
  fencesWave[3,5][1]=lowerInnerFence
  fencesWave[6,8][1]=Q25
  fencesWave[9,11][1]=theMedian
  fencesWave[12,14][1]=Q75
  fencesWave[15,17][1]=upperInnerFence
  fencesWave[18,20][1]=upperOuterFence

  Duplicate/O tmpWave, xWave
  xWave=1.5

  Display/W=(5,45,470,405) fencesWave[*][1] vs fencesWave[*][0]
  // 2006/10/13 Ancilla から (補助的に) 使う場合は、名前をつけて後で Kill します。
```

```

outWinName=UniqueName("WM_BOX_PLOT",6,0)
DoWindow/C $outWinName

AppendToGraph tmpWave vs xWave
ModifyGraph rgb(tmpWave)=(0,0,0)
ModifyGraph margin(right)=221
ModifyGraph mode(tmpWave)=3
ModifyGraph marker(tmpWave)=8
ModifyGraph zColor(fencesWave)={fencesWave[*][1],*,*,Rainbow}
Tag/N=text0/F=0/X=44.96/Y=0.00 fencesWave, 1, ksOuterFence
Tag/N=text1/F=0/X=44.96/Y=0.00 fencesWave, 4, ksInnerFence
Tag/N=text4/F=0/X=44.96/Y=0.00 fencesWave, 10, ksMedian
Tag/N=text3/F=0/X=44.96/Y=0.00 fencesWave, 7, ksLowerHinge
Tag/N=text5/F=0/X=44.96/Y=0.00 fencesWave, 13, ksUpperHinge
Tag/N=text6/F=0/X=44.96/Y=0.00 fencesWave, 16, ksInnerFence
Tag/N=text7/F=0/X=44.96/Y=0.00 fencesWave, 19, ksOuterFence

// ボックスとウィスカーを定義するウェーブを作成し、グラフに適用します。
Make/O/N=(20,2) boxLine=NaN
boxLine[0,3][1]=lowerInnerFence
boxLine[0][0]=kStartX
boxLine[1][0]=kEndX

boxLine[3,4][0]=kMidX
boxLine[4,7][1]=Q25
boxLine[6][0]=kStartX
boxLine[7,8][0]=kEndX
boxLine[8,9][1]=Q75
boxLine[9,10][0]=kStartX
boxLine[10][1]=Q25
boxLine[12][0]=kStartX
boxLine[12,13][1]=theMedian
boxLine[13][0]=kEndX
boxLine[15,16][0]=kMidX
boxLine[15][1]=Q75
boxLine[16,19][1]=upperInnerFence
boxLine[18][0]=kStartX
boxLine[19][0]=kEndX

AppendToGraph boxLine[*][1] vs boxLine[*][0]
ModifyGraph lsize(boxLine)=2
SetDataFolder oldDF
return outWinName

```

End

(参考) StatsQuantiles コマンドの実行

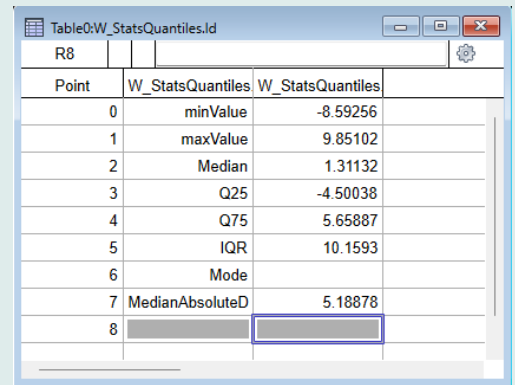
sample50Data ウェーブを作成した後、コマンドウィンドウで次を実行します（フラグ無しの場合）。

```
StatsQuantiles sample50Data
```

W_StatsQuantiles というウェーブが作成され、その内容は右のようになります。

変数も生成され、それぞれ次のように履歴領域に表示されます。

```
Method=0;
V_Q25=-4.50038;
V_Median=1.31132;
V_Q75=5.65887;
V_IQR=10.1593;
V_Mode=nan;
V_MAD=5.18878;
```



Point	W_StatsQuantiles	W_StatsQuantiles
0	minValue	-8.59256
1	maxValue	9.85102
2	Median	1.31132
3	Q25	-4.50038
4	Q75	5.65887
5	IQR	10.1593
6	Mode	
7	MedianAbsoluteD	5.18878
8		

StatsQuantiles コマンドで出力された変数を、元のプロシージャの代入部分で使うほうがより新しい実装方法です。

(参考) StatsQuantiles コマンドのヘルプ

StatsQuantiles [/ALL/BOX/iNaN/IW/TM/Z/Q/QM=method /STBL/TRIM=tValue] srcWave

StatsQuantiles コマンドは、srcWave 内のデータセットに対して、分位数と基本的な単変量統計を計算します。

フラグ

- /ALL /Q、/QM、/Z を除くすべてのフラグを呼び出します。
- /BOX ボックスプロットを作成するために必要なパラメーターを計算します。
- /iNaN NaN（デフォルトでは配列の最後にソートされます）を無視します。
- /IW インデックスウェーブ W_QuantilesIndex を作成します。
W_QuantilesIndex[i] は、最小値から最大値の順に並べた場合の srcWave[i] の位置に対応します。
- /Q コマンドウィンドウの履歴領域に結果を表示しません。
- /QM=qMethod 四分位数の算出方法を指定します。
qMethod = 0 : Tukey（デフォルト）
qMethod = 1 : Minitab
qMethod = 2 : Moore and McCabe
qMethod = 3 : Mendenhall and Sincich
詳細はこの後の「詳細」を参照してください。
- /QW srcWave の各エントリに対応する分位値を含む単精度ウェーブ W_QuantileValues を作成します。

/STBL 安定したソートを使います。
同じ値を持つ複数のエントリに対しては、かなりの計算時間を要する場合があります。

/T=k 結果をテーブル形式で表示します。
k は、それを閉じるときのテーブルの動作を指定します。
k =0: ダイアログを表示 (デフォルト)
k =1: ダイアログを表示せずに Kill
k =2: Kill を不可能にする
テーブルは検定に関連付けられており、データに関連付けられているわけではありません。
検定を繰り返すと、既存のテーブルが新しい結果で更新されます。

/TM 三平均を算出します : $0.25*(V_Q25+2*median+V_Q75)$

/TRIM=tVal tVal (%) と 100-tVal の間の分位数のエントリの平均値であるトリム平均を計算します。
デフォルトは tVal=25 で、トリム平均は中間値に相当します。

/Z エラーを無視します。
V_flag は、エラーが発生した場合は -1 に、それ以外は 0 に設定されます。

詳細

StatsQuantiles は、単変量データについて、5 つの数値による簡潔な要約またはより詳細な結果を生成します。

値はウェーブ W_StatsQuantiles と変数に返されます。

V_min	最小値
V_max	最大値
V_Median	データの中央値
V_Q25	下位四分位
V_Q75	上位四分位
V_IQR	H-spread と呼ばれる四分位範囲 V_Q75-V_Q25
V_MAD	中央絶対偏差
V_Mode	最頻値

もし同値が複数あり、複数の値が最も高い頻度を持つ場合、それらのうちで最も低い値がモードとして返されます。
srcWave のすべての値がユニークであるか、または srcWave のポイントの数が 3 未満である場合、V_Mode は NaN に設定されます。
この出力は、Igor Pro 7.0 で追加されました。

ウェーブ W_StatsQuantiles のエントリは、選択したフラグによって異なります。

各行には、その値を明示的に定義する行ラベルがあります。

/ALL フラグを使うと、W_StatsQuantiles には以下の行ラベルが含まれます :

minValue
maxValue
Median
Q25
Q75
IQR
lowerInnerFence
lowerOuterFence
upperInnerFence
upperOuterFence
trimean

trimmedMean
MedianAbsoluteDeviation

それ以外の場合は、W_StatsQuantiles には最初の 5 つのエントリと、追加で要求された値が含まれません。

常に次元ラベルを使ってこれらの値にアクセスする必要があります。

異なるプログラムで計算された統計結果を比較する際には、しばしば混乱が生じます。

これは、各プログラムが四分位数の異なる定義を使用している可能性があるためです。

四分位数の算出方法は、/QM フラグで指定できます。

方法を選択しなかった場合、StatsQuantiles は Tukey の方法を使います。

この方法では、データの中央値と配列の端の間の中央値の上下に位置する四分位値（ヒンジとも呼ばれる）を計算します。

Moore and McCabe の方法は、四分位数の計算に中央値自体を含めないことを除いては、Tukey の方法と類似しています。

Mendenhall and Sincich は、(numDataPoints+1) の 1/4 と 3/4 を使って四分位を計算し、最も近い整数に四捨五入します（小数部分がちょうど 0.5 の場合は、下位四分位の場合は切り上げ、上位四分位の場合は切り下げます）。

Minitab は同じ式を使いますが、丸めではなく線形補間を使います。

StatsQuantiles は安定したインデックスソートルーチンを使っているため、

```
IndexSort W_QuantilesIndex,srcWave
```

は単調に増加するウェーブとなります。

参照

Tukey, J. W., *Exploratory Data Analysis*, 688 pp., Addison-Wesley, Reading, Massachusetts, 1977.

Mendenhall, W., and T. Sincich, *Statistics for Engineering and the Sciences*, 4th ed., 1008 pp., Prentice Hall, Englewood Cliffs, New Jersey, 1995.

関連するコマンド

Statistical Analysis (ヘルプ) , WaveStats, StatsMedian, median, Sort, MakeIndex