

CONTENTS

Tutorials – ウェーブのスケーリング (Wave Scaling Tutorial)	2
X スケーリング.....	2
例 1 : 単純な等間隔に配置されたデータ	2
例 2 : 単純な XY データ.....	5
例 3 : ウェーブへのインデックス付け.....	7
例 4 : 鳥の鳴き声の DSPPeriodogram.....	8
例 5 : 多次元ウェーブフォーム.....	11

Tutorials – ウェーブのスケーリング (Wave Scaling Tutorial)

メニュー File→Example Experiments→Tutorials→Wave Scaling Tutorial を選択します。

このチュートリアルでは、Igor Pro の重要な概念である「ウェーブフォームのスケーリング」について解説します。

ウェーブフォームのスケーリングは、Igor Pro ならではの強力な機能です。

ウェーブフォームとは、数値の配列とそれに対応するウェーブスケーリング値を含む Igor オブジェクトのことです。

「ウェーブフォーム」という用語は、しばしば「ウェーブ」と略されます。

ヘルプ Waves - The Key Igor Concept も参照してください。

1次元ウェーブの場合、ウェーブのスケーリングとは、保存された Y 値の各行に対応する X 値をどのように計算するかを指します。

2次元ウェーブの場合、ウェーブスケーリングを適用し、保存された Z 値の各行および各列に対する X 値と Y 値を算出します。

3次元と4次元ウェーブについても、同様のウェーブスケーリング計算が使われます。

X スケーリング

ウェーブフォームが1次元の場合、ウェーブフォームのスケーリングは X スケーリングと呼ばれます。

Igor Pro は、Y 値、X オフセット値、デルタ X 値を配列の各行に保存します。

これらの値が組み合わさって、ウェーブフォームオブジェクトが構成されます。

ウェーブを、X 座標をオフセットとし、Y 座標のサンプル間にデルタ X の間隔を空けた Y 値の配列と考えてください。

Igor Pro がデータをグラフ化または分析する時、スケーリング値から X 値を計算し、ウェーブ配列を Y 値として使います。

計算された X 値は、内部的には決して保存されません。

これらは常に必要なときに計算されます。

それでは、いくつかの例を見ていきます。

まずは、等間隔のデータの例から始めます。

これは1次元のウェーブフォームになります。

データは、等間隔の2次元、3次元、あるいは4次元のウェーブフォームである場合もあれば、等間隔ではない XY データである場合もあります。

XY データや、より高次元のデータについては、このチュートリアルの後半で取り上げます。

Igor Pro は日付/時刻のスケーリングもサポートしており、これについてはヘルプ Date/Time Waves で説明しています。

次元とデータスケーリングに関するより詳細な説明については、ヘルプ Changing Dimension and Data Scaling を参照してください。

例 1：単純な等間隔に配置されたデータ

サムは毎朝の散歩で、何種類の鳥の鳴き声を聞いたかを数えています。

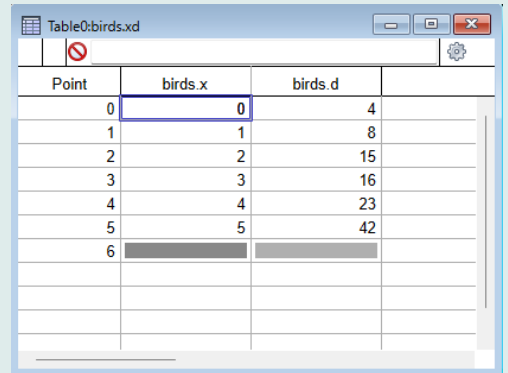
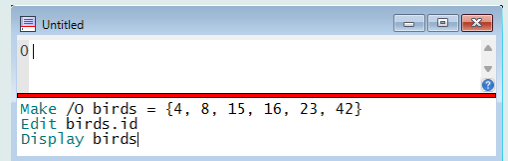
0日目には4種類の鳥の鳴き声を聞き、1日目には8種類、2日目には15種類と、そのように増えていきます。

サムの野鳥観察データを Igor Pro に入力してみましょう。

1. 以下の内容をコマンドウィンドウに入力して、Enter キーを押してください。
チュートリアルをノートブックで開いている場合は、コマンドを選択して Ctrl+Enter キーを押しても構いません。

```
Make /O birds = {4, 8, 15, 16, 23, 42}  
Edit birds.id  
Display birds
```

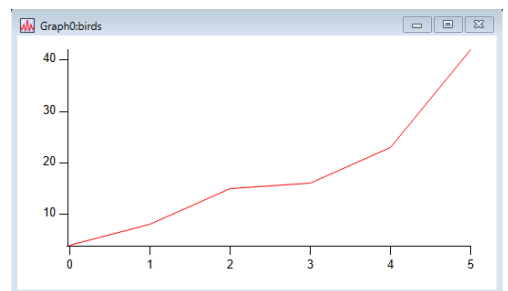
最初のコマンドは、6つの値を持つ「birds」という名前の新しいウェーブフォームを作成します。
/O オプションを指定すると、名前が重複している場合、既存のウェーブフォームが上書きされます。



Point	birds.x	birds.d
0	0	4
1	1	8
2	2	15
3	3	16
4	4	23
5	5	42
6		

2番目のコマンドは、birds ウェーブ用のテーブルウィンドウを作成します。
ウェーブ名に「.id」を追加することで、どの列を表示すべきかを Edit に指示します。
「i」はインデックス（計算された X 値）を、「d」はデータ（Y 値）を表します。

3番目のコマンドは、birds の値を新しいグラフウィンドウにプロットします。



まずはテーブルを見てみます。
3つの列があることに注目してください。
Point、birds.x、birds.d というラベルのついた列です。

Point 列には、その行のポイント番号が含まれています。

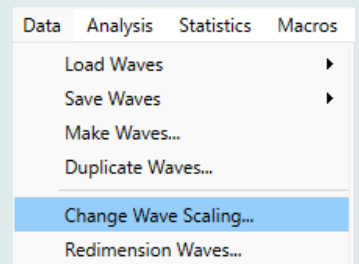
birds.x 列には、計算された X 値が含まれています。
デフォルトの X スケールでは、この列は Point 列と一致します。

birds.d 列は、ウェーブフォームを作成する時に入力した配列です：4、8、15、16、23、42。

次に、グラフウィンドウを見てみます。
X 軸には 0 から 5 までの目盛が付いています。
これは、birds.x 列内の計算されたウェーブのスケールインデックスを表しています。
Y 軸は 4 から 42 までで、サムが毎日見た鳥の数を表しています。
グラフは予想通り描かれています。

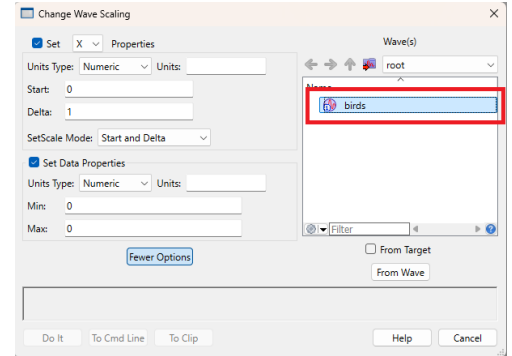
2. birds ウェーブを作る時、X オフセットやデルタ X の値を入力しなかったため、Igor はデフォルト値の 0 と 1 を使いました。
デフォルトの X スケールは、ポイントスケールと呼ばれます。

サムが実際にバードウォッチングを始めたのは 0 日目ではなく 3 日目で、しかも 1 日おきにしか行かなかったと仮定しましょう。
ウェーブのスケールを使って、彼女のデータ収集を表現することができます。



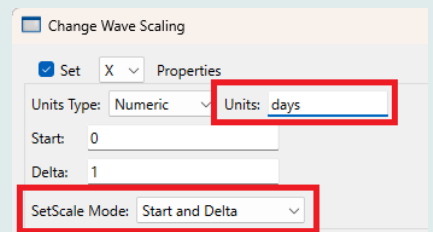
メインメニューの Data→Change Wave Scaling を選択します。

3. Change Wave Scaling ダイアログ内で birds ウェーブをクリックします。



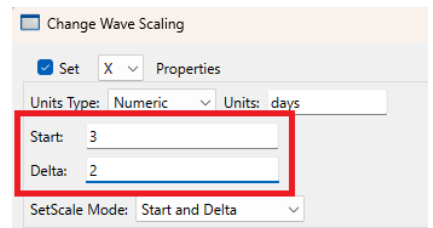
4. Units テキストボックスに「days」と入力します。

SetScale Mode ドロップダウンで Start and Delta を選択します。



5. Start テキストボックスに「3」を入力します。

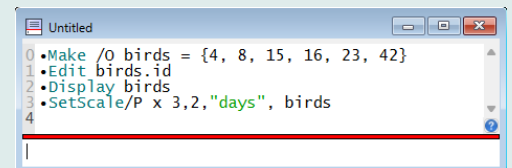
Delta テキストボックスに「2」を入力します。



6. Do It をクリックします。

Change Wave Scaling ダイアログがコマンドウィンドウに次のコマンドを入力したことに注目してください。

SetScale/P x 3, 2, "days", birds

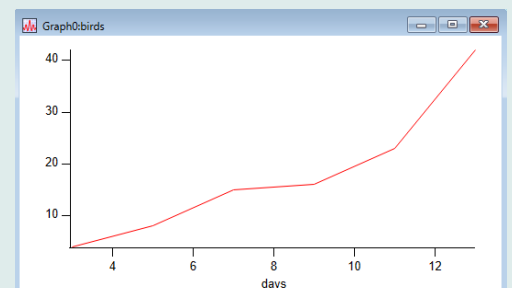


このコマンドは、ウェーブ *birds* のスケーリングを変更します。

/P は、オフセット 3 から開始し、X 値ごとに 2 のデルタ X を使い、単位を日とし、最後にウェーブの名前「birds」を指定して、X 軸のポイントごとのスケーリングを設定します。

Point	birds.x	birds.d
0	3	4
1	5	8
2	7	15
3	9	16
4	11	23
5	13	42
6		

新しいウェーブスケーリング属性を使って、テーブルとグラフが自動的に更新されていることに注目してください。スケーリングの変更によりグラフが再描画され、グラフは新しいウェーブスケーリングを使って X 軸の値を再計算します。テーブルについても同様です。



もう少し掘り下げる：

SetScale コマンドまたは Change Wave Scaling ダイアログで、他の値をいくつか試してみてください。それらがグラフやテーブルの表示にどのような影響を与えるか、確認してみてください。

もう少し掘り下げる：

Tag など、一部のコマンドは、計算された X 値を使って、トレース、画像、または軸上のポイントを特定します。

mean、area、faverage などの一部のコマンドは、計算された X 値を範囲として受け取り、その範囲に基づいて結果を計算します。

これらのコマンドをいくつか試して、計算された X 値に対してどのように動作するかを確認してください。渡された X 値が、現在のウェーブのスケールではウェーブ上に正確に位置していない場合、どうなるでしょうか？

(ヒント：x2pnt と ScaleToIndex)

これらのコマンドの意味が理解できるまで、コマンドとウェーブのスケールをいろいろ試してみてください。

では、等間隔でないデータの場合はどうでしょうか？

2つのウェーブフォームが組み合わさり、一方のウェーブフォームに Y 値が、もう一方に X 値が含まれている場合、これを XY データと呼びます。

XY データでは、通常、ウェーブフォームのスケールは必要なく、使用しません。

一部の演算や関数は、引数として複数のウェーブフォームを受け入れるため、等間隔のデータだけでなく XY データに対しても処理を行うことができます。

例えば、Display や Integrate などが挙げられます。

次の例では、2つの 1D ウェーブを用いて XY データについて解説します。

例 2：単純な XY データ

マットは毎朝の散歩中に聞こえる鳥の種類も数えています。

しかし、彼はもう 1 つデータを加えています。

それは、鳥の鳴き声を聞いたときに自宅から何メートル離れているか、という情報です。

自宅からの距離は GPS の位置情報を使って測定しているため、距離の計測値は等間隔ではありません。

1. 以下の内容をコマンドウィンドウに入力して、Enter キーを押してください。

チュートリアルをノートブックで開いている場合は、コマンドを選択して **Ctrl+Enter** キーを押しても構いません。

```
Make /O birdsByDistance={4, 8, 15, 16, 23, 42}
Make /O distanceFromHouse={3.4, 16, 40, 160, 732.8,
3785.1}
Edit distanceFromHouse.id, birdsByDistance.id
Display birdsByDistance vs distanceFromHouse
```

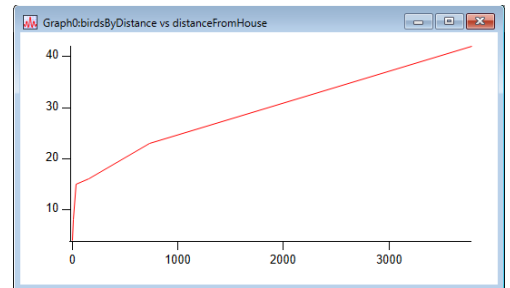
Point	distanceFromHouse	distanceFromHouse	birdsByDistance	birdsByDistance	birdsByDistance
0	0	3.4	0	4	
1	1	16	1	8	
2	2	40	2	15	
3	3	160	3	16	
4	4	732.8	4	23	
5	5	3785.1	5	42	
6					

最初のコマンドは、サムの *birds* ウェーブと同様に、6つの値を持つ *birdsByDistance* という新しいウェーブを作成します。

2つ目のコマンドは、同様の方法で、同じ数の値を用いて、*distanceFromHouse* という別の新しいウェーブを作成します。

3番目のコマンドは、各ウェーブに対応する列を持つテーブルを作成します。

4番目のコマンドは、両方のウェーブからグラフを作成します。



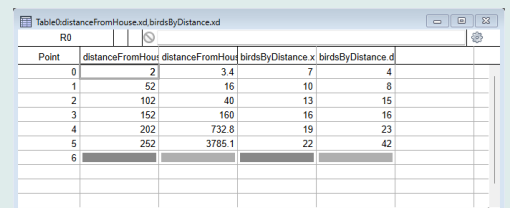
もう少し掘り下げる：

Edit コマンドでウェーブの順序を変更するとどうなりますか？
Display コマンドでウェーブの順序を変更するとどうなりますか？

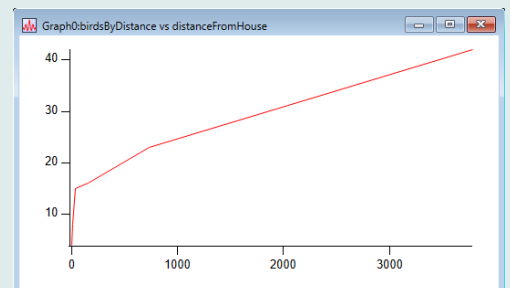
2. ウェーブスケールリングは XY データにどのような影響を与えるでしょうか？

コマンドウィンドウで次のコマンドを実行してください

```
SetScale /P x 7, 3, "birds", birdsByDistance  
SetScale /P x 2, 50, "m", distanceFromHouse
```



Point	distanceFromHouse	distanceFromHouse	birdsByDistance	birdsByDistance
0	2	3.4	7	4
1	52	16	10	8
2	102	40	13	15
3	152	160	16	16
4	202	732.8	19	23
5	252	3785.1	22	42
6				



3. グラフの表示が変わっていないことに気づきましたか？ Igor Pro がコマンドを受け取ったかどうか確認してみましよう。

Data→Data Browser または Ctrl+B を押して、データブラウザを開きます。

birdsByDistance を選択し、情報ペインの Rows の行に注目してください。

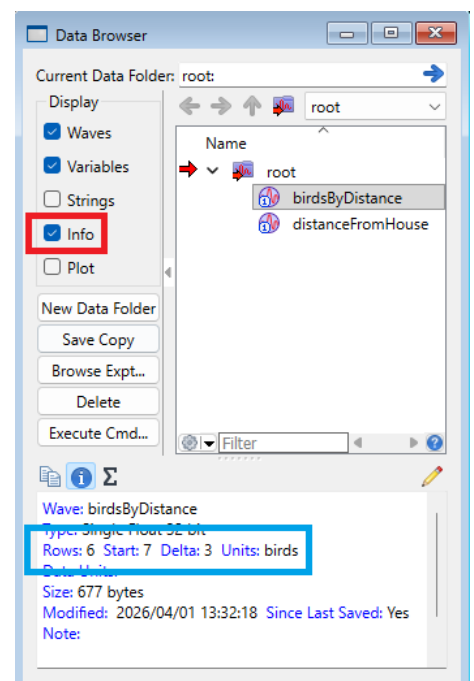
Rows: 6 Start: 7 Delta: 3 Units: birds

同様に *distanceFromHouse* に対しては、

Rows: 6 Start: 2 Delta: 50 Units: meters

Start と Delta の値から、SetScale コマンドが正しく実行され、結果を保存したことがわかります。

しかし、Display コマンドは、Y ウェーブのスケールリング値から X 値を計算するのではなく、X ウェーブの X 値を使うように設定されています。



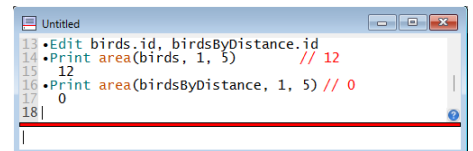
注記： XY データに対してスケーリングを行うことは論理的に意味をなさないため、これらのウェブフォームについてはデフォルトのポイントスケーリングを維持するようにしてください。
XY データにスケーリングを適用してからウェブフォームを分析すると、デバッグが非常に困難な問題が発生する可能性があります。

例 3 : ウェーブへのインデックス付け

ウェーブスケーリングを使えば、データの一部だけを自然な形で参照することができます。以下にいくつかの例を挙げます。

この時点で、*birds* と *birdsByDistance* は同じ Y 値を持つはずですが、ウェーブのスケールは異なります。コマンドウィンドウで以下を実行してください。

```
Edit birds.id, birdsByDistance.id
Print area(birds, 1, 5)           // 12
Print area(birdsByDistance, 1, 5) // 0
```



最初の *area* 関数の呼び出しは、*birds* ウェーブフォームと直線 $y=0$ の間の曲線下の面積を、 $x=1$ から $x=5$ までの範囲で計算します。

2 番目の *area* 関数も同様に計算を行います。

Print コマンドは、*area* 関数の結果をコマンドウィンドウに出力します。

Y 値が同じなのに、なぜ計算された面積がこれほど異なるのでしょうか？

これはウェーブのスケールが異なるためです。

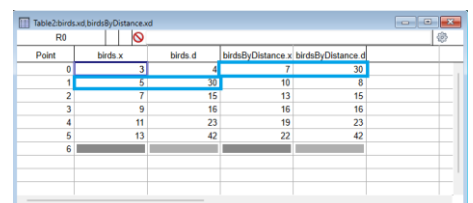
特に、X 値が 1 から 5 までの範囲における *birdsByDistance* の面積は 0 になります。

これは、 $1 \leq x \leq 5$ の範囲には Y 値が存在しないためです。

ウェーブスケーリングを使うと、ウェーブに対して自然なインデックス付けを行うこともできます。

コマンドウィンドウで以下を実行してください。

```
// birds.x = 5 の場所で birds.d = 30 と設定
birds(5.2) = 30
// birdsByDistance.x = 7 の場所で birdsByDistance = 30 と設定
birdsByDistance(5.2) = 30
```



Point	birds.x	birds.d	birdsByDistance.x	birdsByDistance.d
0	3	4	7	30
1	5	30	10	8
2	7	15	13	15
3	9	16	16	16
4	11	23	19	23
5	13	42	22	42
6				

ポイント番号を使ってウェーブ内の特定の位置を指定することもできます。

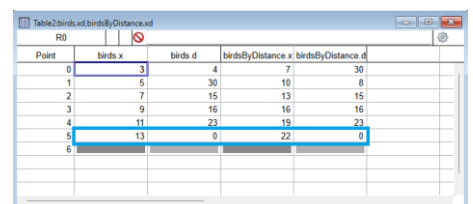
ポイント番号は、テーブルウィンドウの最初の列にあることはすでに知っていると思います。

コマンドウィンドウで次のコマンドを実行してください。

```
birds[5] = 0
birdsByDistance[5] = 0
```

これらのコマンドは、ウェーブの 6 番目のポイントの Y 値を 0 に設定します。

ポイントの番号は 0 から始まるため、5 番目のポイントがウェーブにおける 6 番目のポイントとなります。



Point	birds.x	birds.d	birdsByDistance.x	birdsByDistance.d
0	3	4	7	30
1	5	30	10	8
2	7	15	13	15
3	9	16	16	16
4	11	23	19	23
5	13	0	22	0
6				

注記： 丸括弧は X 座標のインデックスを示し、角括弧は Y 座標のインデックスを示します。

ウェーブの特定の位置を指定する必要があるコマンドでは、ウェーブスケーリングが使われます。

コマンドウィンドウで以下を実行してください。

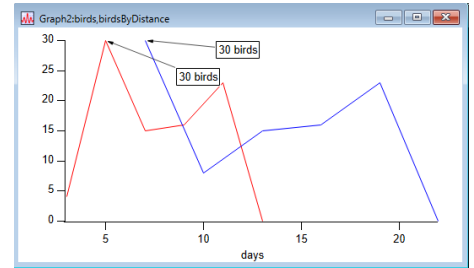
```

Display birds, birdsByDistance
ModifyGraph rgb(birdsByDistance)=(0, 0, 65535)
Tag /C /N=redTag /X=25 /Y=-20 birds, 5, "30 birds"
Tag /C /N=blueTag /X=25 /Y=-5 birdsByDistance, 7, "30 birds"

```

Display コマンドは、両方のウェーブを同じ座標軸上にプロットします。

ModifyGraph は、birdsByDistance の RGB 値を青に設定します。
rgb キーの値は、赤が 0、緑が 0、青が最大値の 65535 です。



最初の Tag コマンドは、x=5 に最も近いウェーブ値を指す「30 birds」というタグをウェーブに追加します。

/C は、タグがすでに存在する場合にそれを変更するよう Tag に指示します。

/N はタグに名前を付けます。

/X と /Y は、タグのテキストをタグポイントから、それぞれグラフの幅と高さのパーセンテージで指定したオフセット位置に配置します。

2 番目の Tag コマンドは、x=7 に最も近い birdsByDistance ウェーブの値に、「30 birds」というタグを追加します。

Transform コマンドは計算にウェーブのスケーリングを使っているため、より詳しく検討する必要があります。

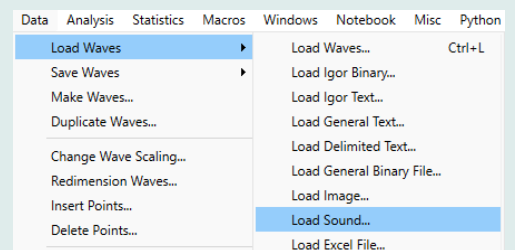
例 4 : 鳥の鳴き声の DSPPeriodogram

アマンドはバードウォッチングに行くとき、耳にする鳥の鳴き声を録音するのが大好きです。

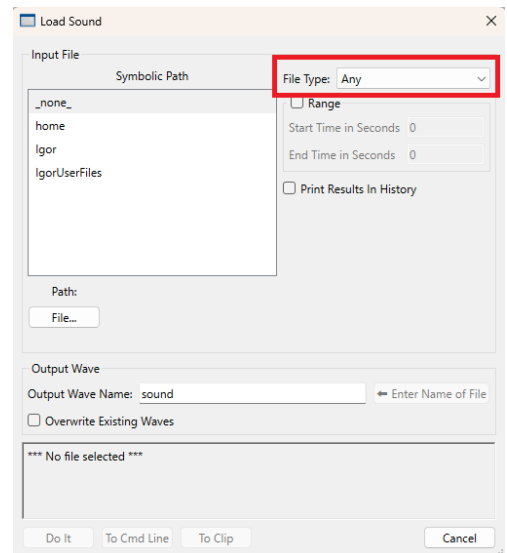
彼女は、お気に入りの鳥の鳴き声の音程や音色に、とても興味を持っています。

Igor Pro を使えば、変換を用いたデジタル信号処理と解析で彼女を手助けすることができます。

1. Igor Pro のメインメニューで、Data→Load Waves→Load Sound を選択します。



2. File Type ドロップダウンで、Any を選択します。



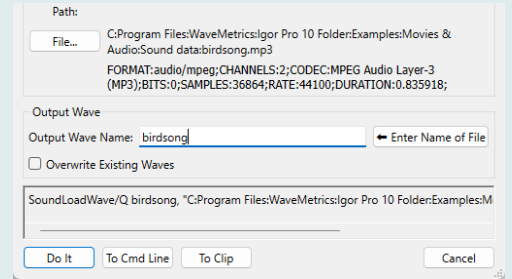
3. File ボタンをクリックします。

<インストールフォルダ>\Igor Pro <X>

Folder\Examples\Movies & Audio\Sound data にある birdsong.mp3 ファイルを選択して開きます（デフォルトのインストールフォルダは C:\Program Files\WaveMetrics です）。

Output Wave Name を「birdsong」に変更します。

Do It をクリックします。



4. コマンドウィンドウで次のコマンドを実行します。

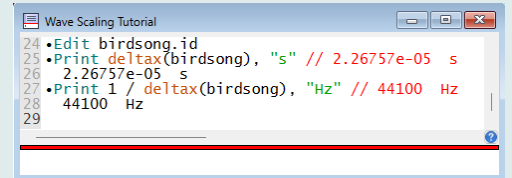
```
Edit birdsong.id
```

「birdsong」テーブルでは、「birdsong.x」列がウェーブの X 軸のスケールを示していることに注目してください。SoundLoadWave コマンドにより、スケールングのデルタ X はおよそ 22.5 マイクロ秒のサンプリングレートに設定されました。

Row	birdsong.x	birdsong[][0].d	birdsong[][1].d
0	x	y	0
1	2.26757e-05	0	0
2	4.53515e-05	0	0
3	6.80272e-05	0	0
4	9.07029e-05	0	0
5	0.000113379	0	0
6	0.000136054	0	0
7	0.00015873	0	0
8	0.000181406	0	0
9	0.000204082	0	0
10	0.000226757	0	0

5. 正確なサンプリングレートと周波数は、コマンドウィンドウで以下を実行することで計算できます。

```
Print deltax(birdsong), "s" // 2.26757e-05 秒  
Print 1 / deltax(birdsong), "Hz" // 44100 Hz
```



また、birdsong[][0].d と birdsong[][1].d というラベルの付いた 2 つの列があります。

これは、この音声ファイルがステレオ形式であるためです。

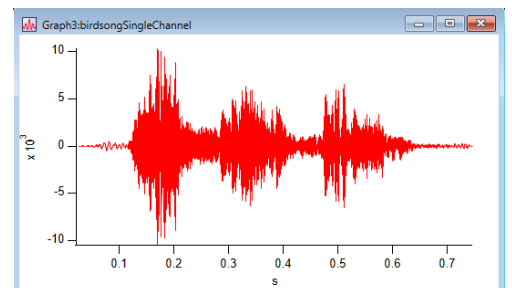
DSPPeriodogram などの変換処理は、一度に 1 つのデータ列に対してしか実行できないため、まず最初に、データの片方のチャンネルを抽出（または両方を結合してモノラルにする）必要があります。

これを行う最善の方法は、Duplicate コマンドを使うことです。

その後、その 1 つのデータチャンネルを DSPPeriodogram コマンドに入力します。

6. コマンドウィンドウで以下を実行します。

```
Duplicate /O /RMD=[1120,32897][0] birdsong,  
birdsongSingleChannel  
Redimension /S birdsongSingleChannel  
Edit /N=timePlane birdsongSingleChannel.id  
Display birdsongSingleChannel
```



Duplicate コマンドは、birdsong ウェーブデータの 0 列目から、birdsongSingleChannel という名前の新しいウェーブを作成します。

/O フラグの意味は、これまでのコマンドと同様で、同名のウェーブがすでに存在する場合、それを上書きします。

/RMD フラグは、複製するデータの範囲を指定します。

ここでは、行 1120 から 32897 までの 0 列目をコピーするように指定しています。

これらの行は、空のデータ部分を切り取るため、またデータポイントの数が偶数になるように確保するために選択されます。

データポイントの数が偶数であることは、すべての変換コマンドにおいて必須の要件です。

/S フラグを指定した Redimension コマンドは、ウェーブデータの型を 16 ビット整数から単精度浮動小数点数に変換します。

変換処理は常に浮動小数点データに対して行われます。

ただし、birdsongSingleChannel は PlaySound コマンドでは再生できなくなります。

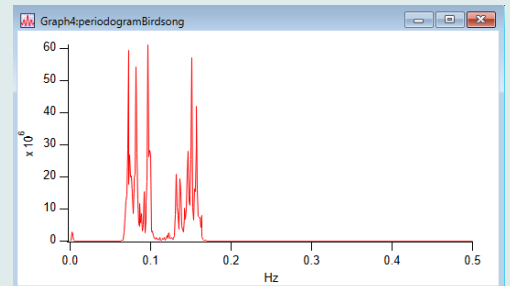
Edit コマンドを実行すると、birdsongSingleChannel が timePlane という名前のテーブルとして開き、元のステレオ音源のテーブルと同様に、振幅とサンプリング時間の行が表示されます。

Display コマンドは、birdsongSingleChannel を時間軸上で振幅 vs 時間のグラフとして表示します。

次に、birdsongSingleChannel の単位を秒に設定します。このスケーリングは変換処理中に使われます。

7. コマンドウィンドウで以下を実行します。

```
SetScale /P x 0, 1, "s", birdsongSingleChannel
DSPPeriodogram /SEGN={1024, 512}
    /DEST=periodogramBirdsong birdsongSingleChannel
Edit /N=frequencyPlane periodogramBirdsong.id
Display periodogramBirdsong
```



DSPPeriodogram コマンドは、birdsongSingleChannel から「periodogramBirdsong」という新しいピリオドグラムウェーブを作成します。

/SEGN の値は、1024 ポイントのセグメントを平均化し、セグメント間で 50 %のオーバーラップを持たせます。birdsongSingleChannel からのセグメントは変換、平均化され、periodogramBirdsong が生成されます。

Edit コマンドを実行すると、periodogramBirdsong が frequencyPlane テーブルに追加され、振幅と、そのピリオドグラムにおける振幅が計算された周波数の両方が行として表示されます。

Display コマンドは、periodogramBirdsong を周波数平面上に、振幅 vs 周波数としてグラフ化します。

これで、ウェーブスケーリングが適用された音波と、そのウェーブスケーリングによって周波数範囲が示された DSPPeriodogram の結果が得られました。

ソースウェーブの X 軸の単位を秒に設定したため、X 軸の目盛りが自動的に kHz 単位で表示されていることに注目してください。

アマダは、録音した鳥のさえずりの主な周波数が、およそ 3~4.25 kHz と 5.75~7 kHz の範囲にあることを把握しました。

つまり、音高としては G7~C8、F#8~A8 の範囲です。

素敵な C6sus や G メジャー9sus のコードかもしれませんね。

注記： $1 / \text{deltax}(\text{birdsongSingleChannel})$ はサンプリング周波数に等しく、これは periodogramBirdsong の最大周波数の 2 倍に相当します。

この特性は、デジタル信号処理における重要な概念です。

Igor Pro は、ソースウェーブのウェーブスケーリング値を使って、振幅 vs 周波数ウェーブ periodogramBirdsong を計算します。

これを確認するには、コマンドウィンドウで以下を実行してください。

```
Print 1/deltax(birdsongSingleChannel), "Hz" // 44100 Hz
Print (leftx(periodogramBirdsong) + (numpts(periodogramBirdsong) - 1) *
    deltax(periodogramBirdsong)), "Hz" // 22050 Hz
```

```
37 • Display periodogramBirdsong
38 • Print 1/deltax(birdsongSingleChannel), "Hz" // 44100 Hz
39 • 1 Hz
40 • Print (leftx(periodogramBirdsong) + (numpts(periodogramBirdsong) - 1) * deltax(periodogramBirdsong)), "Hz" // 22050 Hz
41 • 0.5 Hz
42
```

もう少し掘り下げる： ウェーブのウェーブスケールを変更して、SingleChannel の「birdsong」の周期を変えてみてください。その後、もう一度 DSPPeriodogram を使ってウェーブを変換してください。その結果、ピリオドグラムにはどのような変化が見られますか？また、変わらない部分はどこですか？

2D、3D、4D の場合はどうでしょうか？

これまで、1次元のウェーブについてのみ説明してきました。

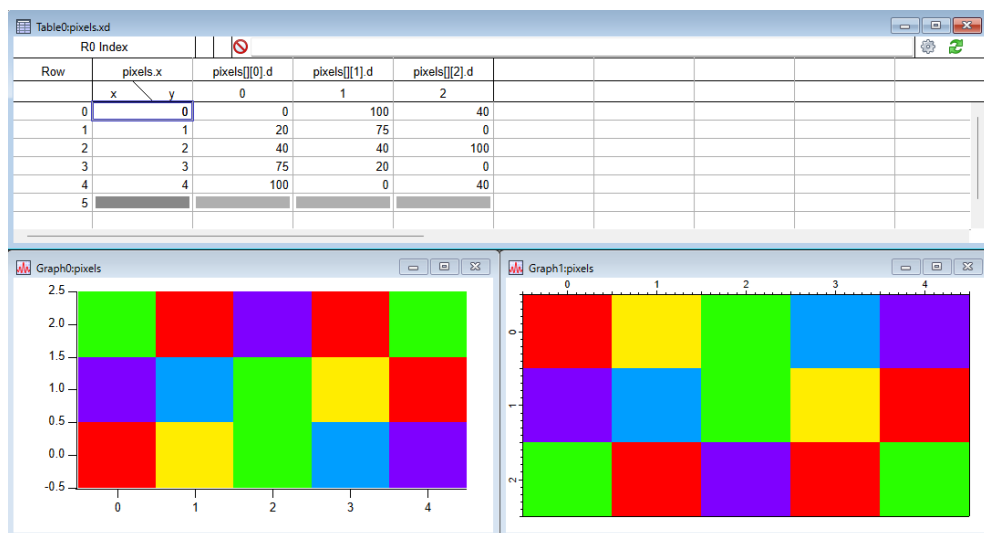
次のセクションでは、多次元のウェーブについて見ていき、こうしたウェーブにおいてウェーブのスケーリングがどのように機能するかを検討してみます。

例 5：多次元ウェーブフォーム

2次元のウェーブは行列とも呼ばれ、行（第1次元）と列（第2次元）から構成されています。

コマンドウィンドウで次のコマンドを実行します。

```
Make /O pixels = {{0, 20, 40, 75, 100}, {100, 75, 40, 20, 0}, {40, 0, 100, 0, 40}}
Edit pixels.id
Display; AppendImage pixels
ModifyImage pixels ctab= {0, 100, Rainbow256}
NewImage pixels
ModifyImage pixels ctab= {0, 100, Rainbow256}
```



最初のコマンドは、「pixels」という名前の 5×3 の行列を作成します。

中括弧 { } で囲まれた各ポイントの集合が 1 つの列となります。

各列のポイント数は同じでなければなりません。

各ポイントは Z 値です。

2 番目のコマンドは、新しいテーブルにピクセルを追加し、X 軸と Y 軸の両方のスケーリングを表示します。

3 番目のコマンドは、1次元のウェーブの場合とは異なります。

この Display コマンドは、Igor Pro に対して新しい空のグラフを表示し、そのグラフに画像としてピクセルを追加するよう指示します (Graph0)。

4 番目の最後のコマンドは、画像にカラーテーブルを追加することで、上部のグラフのピクセル画像プロットを変更します。

ctab キーワードの引数には、Z 値の最小値、Z 値の最大値、カラーテーブルの名前を指定します。

このウェーブの Z 値は最小 0、最大 100 であり、カラーテーブルには Rainbow256 を使っています。

4番目のコマンドは、ピクセルを画像として新しいグラフウィンドウ（Graph1）にネイティブに描画します。

注記： 画像は、多くの場合、左上が (0, 0) になるように表示されます。Display; AppendImage pixels コマンドの代わりに NewImage pixels コマンドを使うと、Igor Pro はその方法で自動的に画像をプロットします。また、NewImage コマンドは、画像のピクセルが初期状態で正方形になるようにグラフのサイズを設定します。これが、先ほど作成した2つのグラフの違いの原因となっています。

注記： 画像の目盛は、各ピクセルの中心に配置されています。

もう少し掘り下げる： 1D ウェーブを表示したときと同じ Display コマンド、つまり Display pixels を使って、ピクセルのウェーブを表示してみてください。これにより、AppendImage や NewImage の代わりに Display を多次元ウェーブに使った場合に何が起ころかがわかります。データをグラフ化した時に予想外の結果が出た場合は、データの次元を確認し、正しくグラフ化されているか確認してください。

ピクセルのテーブルにある列の見出しに注目してください。

これらは見覚えがあるはずです。

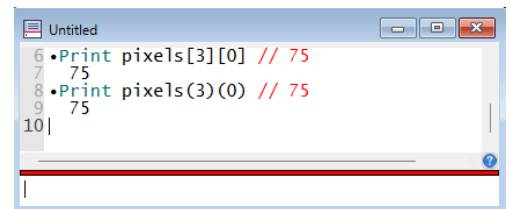
これまでのセクションのステレオの鳥のさえずりのウェーブデータにも、同様の列がありました。

Igor Pro では、インデックス構文 `[][0]` を第1列全体、`[][1]` を第2列全体を表すために使います。

1D ウェーブへのインデックス指定と同様に、角括弧はポイント番号によるインデックス指定を、丸括弧はウェーブスケールされた X と Y 値を表します。

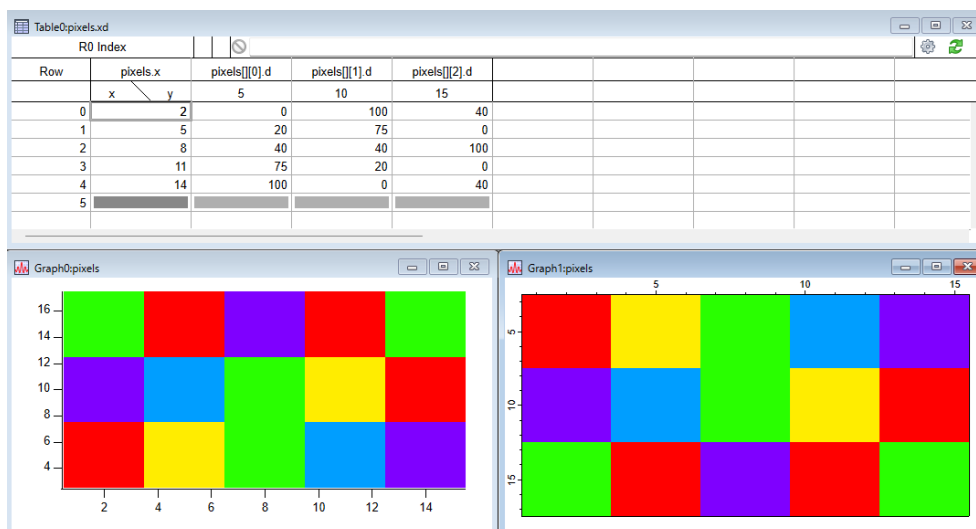
コマンドウィンドウに以下を入力してください。

```
Print pixels[3][0] // 75
Print pixels(3)(0) // 75
```



pixels は現在デフォルトのポイントスケールングを使っているため、どちらのコマンドでも同じ結果が得られます。それでは、コマンドウィンドウに以下を入力してください。

```
SetScale /P x 2, 3, "", pixels
SetScale /P y 5, 5, "", pixels
Print pixels[3][0] // 75
Print pixels(11)(5) // 75
```



ピクセルグラフの軸の設定変更注目してください。

X オフセットの値は、水平軸上の画像の最初のピクセルの中心位置として使われます。

また、デルタ X の値は、水平軸上の連続するピクセルの中心間の距離となります。

垂直軸上の Y オフセットとデルタ Y の値についても同様です。

さらに、pixels.x 列と Y ラベルの列見出しにおけるテーブルの設定の変更にも注目してください。

3次元と4次元のウェーブは、2次元のウェーブと同様の性質を持ちます。
3D ウェーブでは、Z プロパティ（レイヤー）のスケール調整も可能です。
4D ウェーブでは、T プロパティ（チャンク）のスケール調整も可能です。

これまでに、さまざまなデータタイプにおけるウェーブスケーリングの仕組みについて見てきました。
X/Wave スケーリングは Igor Pro の中核となる概念であり、データやその分析にどのような影響を与えるかを理解しておくことは重要です。
引き続き、ヘルプドキュメントなどを参照し、さらに学習を深めてください。