

CONTENTS

Volume II User's Guide: Part1 (Importing and Exporting Data #3)	2
Loading Image Files (画像ファイルの読み込み)	2
Load Image ダイアログ	2
PNG ファイルを読み込む	3
JPEG ファイルを読み込む	3
BMP ファイルを読み込む	3
TIFF ファイルを読み込む	3
Sun Raster ファイルを読み込む	4
Loading Row-Oriented Text Data (行指向テキストデータの読み込み)	5
Loading Excel Files (Excel ファイルの読み込み)	6
XLLoadWave が読み込むもの	6
XLLoadWave とウェーブ名	9
XLLoadWave の出力変数	12
Excel Date/Time vs. Igor Date/Time	12
Excel Data を 2D ウェーブに読み込む	12

Volume II User's Guide: Part1 (Importing and Exporting Data #3)

Loading Image Files (画像ファイルの読み込み)

Igor Pro マニュアル : II-157 ページ以降をもとに編集

Load Image ダイアログ (メニュー Data → Load Waves → Load Image) を使って、JPEG、PNG、TIFF、BMP、Sun Raster の画像ファイルを読み込むことができます。

画像データを含む数値プレーンテキストファイルは、Data メニューから Load Waves → Load Waves を選択して、LoadWaves ダイアログを使って読み込むことができます。

Load columns into matrix チェックボックスにチェックを入れます。

HDF5 ファイルから画像を読み込むことができます。
ヘルプを表示するには、コマンドラインで次を実行します。

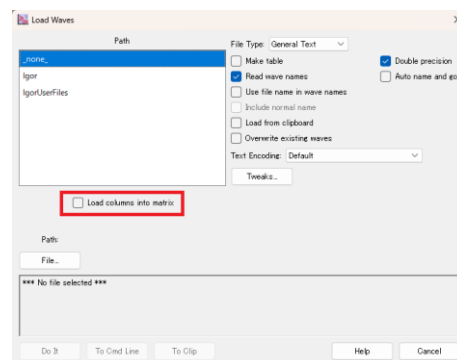
```
DisplayHelpTopic "HDF5 in Igor Pro"
```

HDF4 ファイルから画像を読み込むことができます。
ヘルプを表示するには、コマンドラインで次を実行します。

```
DisplayHelpTopic "HDF Loader XOP"
```

フレームを囲んで画像を読み込むこともできます。

NewCamera コマンドを参照してください。



Load Image ダイアログ

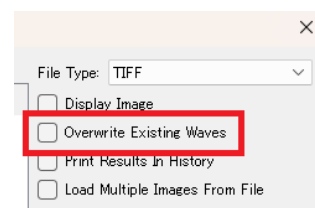
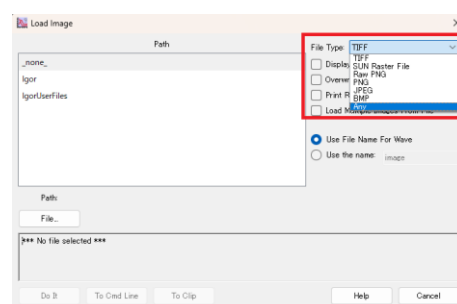
ウェーブに画像ファイルをロードするには、メニュー Data → Load Waves → Load Image を選択して、Load Image ダイアログを表示します。

File Type ポップアップメニューから、特定の種類の画像ファイルを選択すると、画像ファイル選択ダイアログを表示するときに使われるファイルフィルタが設定されます。

画像ファイルのファイル拡張子が正しいかどうかわからない場合は、フィルタが選択を制限しないように File Type ポップアップメニューから Any を選択します。

読み込まれたウェーブの名前は、ファイル名または指定した名前にすることができます。

ダイアログボックスに既存のウェーブ名と競合する名前を入力し、Overwrite Existing Waves チェックボックスにチェックを入れない場合、新しいウェーブ名に数字のサフィックスが付けられます。



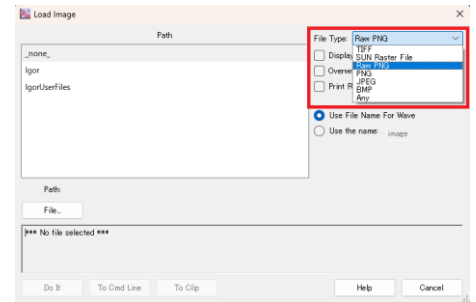
PNG ファイルを読み込む

PNG 形式には、Raw PNG と PNG の2つのメニューオプションがあります。

Raw PNG が選択された場合、データはファイルから直接ウェーブを読み込まれます。

PNG が選択されると、ファイルはメモリにロードされ、オフスクリーン画像が作成され、ウェーブデータはオフスクリーン画像を読み込むことで設定されます。

ほとんどの場合、Raw PNG を選択すべきです。



PNG ファイルを読み込む時には、画像データは、レイヤー 0、1、2 の符号なしバイト RGB 要素を含む 3D Igor RGB ウェーブを読み込まれます。

画像ファイルにアルファチャンネルが含まれている場合は、生成される 3D RGBA ウェーブにはアルファレイヤーが含まれます。

RGB 画像を含む 3D ウェーブをグレースケール画像に変換するには、ImageTransform コマンドで rgb2gray キーワードを使います。

JPEG ファイルを読み込む

JPEG ファイルを読み込む時には、画像データは、レイヤー 0、1、2 の符号なしバイト RGB 要素を含む 3D Igor RGB ウェーブを読み込まれます。

JPEG はアルファをサポートしていません。

RGB 画像を含む 3D ウェーブをグレースケール画像に変換するには、ImageTransform コマンドで rgb2gray キーワードを使います。

BMP ファイルを読み込む

BMP ファイルを読み込む時には、画像データは、レイヤー 0、1、2 の符号なしバイト RGB 要素を含む 3D Igor RGB ウェーブを読み込まれます。

BMP はアルファをサポートしていません。

RGB 画像を含む 3D ウェーブをグレースケール画像に変換するには、ImageTransform コマンドで rgb2gray キーワードを使います。

TIFF ファイルを読み込む

TIFF ファイルは、多くのフォーマットで1つ以上の画像を保存できます。

もっとも一般的なものは次です：

- 2値 — 各ピクセルが黒または白を表現できる、1つの平面データで構成されます。Igor は2値画像を 2D ウェーブを読み込みます。

- グレースケール — 各ピクセルが強度の範囲を表現できる、1つの平面データで構成されます。Igor はグレースケール画像を 2D ウェーブに読み込みます。
- パレットカラー — グレースケール画像に似ていますが、カラーパレットを含んでいます。Igor はグレースケール画像を 2D ウェーブに読み込み、さらに「_CMap」というサフィックスのついたカラーマップウェーブを作成します。
- フルカラー (RGB、RGBA、CMYK) — 3レイヤーまたは4レイヤーの 3D ウェーブに読み込みます。各レイヤーには1つの色要素のピクセルが格納されます。

複数の画像を含む TIFF ファイルは TIFF スタックと呼ばれます。

読み込みには2つの方法があります。

- 画像を1つの 3D ウェーブに読み込みます。
これはグレースケール画像のみで動作します。
各グレースケール画像は、3D 出力ウェーブのレイヤーに読み込まれます。
- 各画像をそれぞれのウェーブに読み込みます。
これはあらゆる種類の画像で機能します。グレースケール画像はそれぞれ別の 2D ウェーブに読み込まれます。
RGB、RGBA、CMYK 画像は、それぞれ別の 3D ウェーブに読み込まれます。

マルチイメージ TIFF ファイルから読み込む特定の画像、または画像の範囲を指定することができます。

Load Image ダイアログで、読み込む最初の画像のゼロから始まるインデックスと、TIFF スタックから読み込む画像の数を入力します。

TIFF 画像は NewImage コマンドで表示でき、ImageTransform コマンドで画像ウェーブを他の形式に変換できます。

RGB 画像を含む 3D ウェーブをグレースケール画像に変換するには、ImageTransform コマンドで rgb2gray キーワードを使います。

ImageTransform コマンドで stackImages キーワードを使うと、複数の 2D 画像ウェーブを 3D スタックに変換することができます。

Sun Raster ファイルを読み込む

Sun Raster ファイルは 2D ウェーブに読み込まれます。

Sun Raster ファイルにカラーマップが含まれている場合、画像ウェーブに加えて、サフィックス「_CMap」のついたカラーマップウェーブを作成します。

Loading Row-Oriented Text Data (行指向テキストデータの読み込み)

Igor Pro マニュアル : II-159 ページをもとに編集

Load Image 組み込みのテキストローダーはすべて列指向です。

つまり、ファイル内のデータ列を 1D ウェーブにロードします。

かなり一般的に、行指向のフォーマットがあります。

このフォーマットでは、ファイルは 1 つのウェーブのデータを表しますが、複数の列に書き込まれます。

次がその一例です。

```
350    2.97    1.95    1.00    8.10    2.42
351    3.09    4.08    1.90    7.53    4.87
352    3.18    5.91    1.04    6.90    1.77
```

この列では、最初の列には X の値が含まれ、残りの列にはデータ値が含まれ、行/列の順序で書かれています。

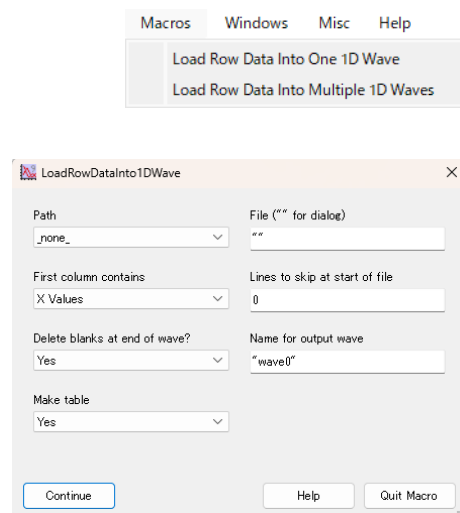
Igor には、この形式を処理するファイルローダーエクステンションはありませんが、この目的のための WaveMetrics 製のプロシージャファイルがあります。

これを使うには、WaveMetrics Procedures:File Input Output フォルダ内の Load Row Data プロシージャファイルを使います (Load Row Data.ipf ファイルを Igor Procedures フォルダにコピーします)。

これを使うと、Macro メニューに Load Row Data という項目が追加されます。これを選択すると、Igor はいくつかのオプションを提供するダイアログを表示します。

オプションの 1 つは、最初の列を X 値またはデータとして扱います。

列を X 値として指定すると、最初の列の値が等間隔であると仮定して、出力ウェーブの X スケーリングを決定するためにそれを使います。



Loading Excel Files (Excel ファイルの読み込み)

Igor Pro マニュアル : II-159 ページ以降をもとに編集

Excel ファイルからデータを読み込むには、XLLoadWave コマンドを直接使うか、メニュー Data → Load Waves → Load Excel File を選択して、Load Excel File ダイアログを表示します。

XLLoadWave は、数値、テキスト、日付、時刻、日付/時刻データを Excel ファイルからウェーブに読み込みます。

.xls と .xlsx ファイルからデータを読み込むことができます。

.xlsb (大きなファイル用のバイナリ形式) ファイルには対応していません。

また、パスワードで保護された Excel ファイルも読み込めません。

Igor にロードする前に、Excel のワークシートを閉じておく必要があります。

一部のプログラムではタブ区切りやその他の Excel 以外の形式のファイルを .xls という拡張して保存してしまうことがあります。

これらのファイルのいずれかをロードしようとする時、XLLoadWave はそれが Excel のバイナリファイルではないことを表示します。

XLLoadWave が読み込むもの

ワークシートは、長方形の数字ブロックだけで構成される非常にシンプルなものでも、数値、文字列、数式のブロックが任意の方法で混在する非常に複雑なものでも構いません。

XLLoadWave は、ワークシートから矩形のセルブロックを抜き出して、列をウェーブに変換するように設計されています。

XLLoadWave は数値データとテキスト (文字列) データの両方を読み込むことができます。

Excel の列には、数値セルとテキストセルを混在させることができます。

ウェーブは、すべて数値またはすべてテキストでなければなりません。

Excel の列をウェーブに読み込む場合、数値ウェーブまたはテキストウェーブのどちらにデータを読み込むかを決定する必要があります。

XLLoadWave は、日付、時刻、日付/時刻データを数値ウェーブにロードすることもできます。

列とウェーブの形式

XLLoadWave は、指定された列に対して作成するウェーブの形式を決定する次の方法を提供します。

これらの方法は、Load Excel File ダイアログに表示され、XLLoadWave コマンドの /C および /COLT フラグによってコントロールされます。

Treat all columns as numeric (すべての列を数値として扱う)

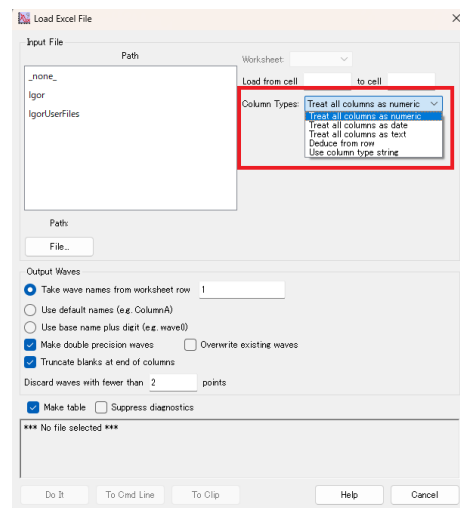
これがデフォルトの方法です。

ウェーブに読み込みたい単純な数字のブロックがある場合は、この方法を使います。

XLLoadWave は、読み込む Excel の各列に対して数値ウェーブを作成します。

列に数値セルが含まれている場合、それらの値はウェーブの対応するポイントに格納されます。

列にテキストセルが含まれている場合、XLLoadWave はウェーブの対応するポイントに NaN (空白) を格納します。



読み込むデータ (Excel : 左) と読み込み後 (Igor Pro : 右)

	A	B	C	D	E	F
1	2.97	1.95	1.00	8.10	2.42	
2	3.09	4.08	1.90	7.53	4.87	
3	3.18	5.91	AAAA	6.90	1.77	
4	3.12	2.05	1.05	BBBB	2.54	
5	3.18	4.20	1.30	7.70	5.02	
6						

Point	ColumnA	ColumnB	ColumnC	ColumnD	ColumnE
0	2.97	1.95	1	8.1	2.42
1	3.09	4.08	1.9	7.53	4.87
2	3.18	5.91		6.9	1.77
3	3.1185	2.0475	1.05		2.541
4	3.1827	4.2024	1.307	7.7009	5.0161
5					

Treat all columns as date (すべての列を日付として扱う)

これは、XLLoadWave が Excel の日付/時刻フォーマットの数値データを Igor の日付/時刻フォーマットに変換する点を除いては、前述の方法と同じです。

詳細は、数セクション後の Excel Date/Time vs Igor Date/Time を参照してください。

読み込むデータ (Excel : 左) と読み込み後 (Igor Pro : 右)

	A	B	C
1	2024/12/10 8:00	2024/12/10 11:00	
2	2024/12/11 8:00	2024/12/11 11:00	
3	2024/12/12 8:00	2024/12/12 13:00	
4	2024/12/13 8:00	2024/12/13 13:00	
5	2024/12/14 8:00	2024/12/14 20:00	
6			

Point	ColumnA	ColumnB
0	2024/12/10 08:00:00	2024/12/10 11:00:00
1	2024/12/11 08:00:00	2024/12/11 11:00:00
2	2024/12/12 08:00:00	2024/12/12 13:00:00
3	2024/12/13 08:00:00	2024/12/13 13:00:00
4	2024/12/14 08:00:00	2024/12/14 20:00:00
5		

XLLoadWave が日付や時刻を保存する数値ウェーブを作成する場合、日付を正確に保存するには倍精度が必要であるため、常に倍精度ウェーブが作成されます。

また、XLLoadWave はウェーブのデータ単位を「dat」に設定します。

グラフで XY ペアの X 軸としてウェーブを使う場合、「dat」を日付および/または時刻を含むことを示す記号として認識します。

この方法では、XLLoadWave がテーブルにウェーブを表示する場合、テーブルの列に日付/時刻フォーマットが使われます。

ModifyTable コマンドを使って、列のフォーマットを日付のみ、または時刻のみに変更することができます。

Treat all columns as text (すべての列をテキストとして扱う)

XLLoadWave は、すべての列をテキストウェーブにロードします。

数値セルを含む列をテキストウェーブに読み込むと、数値セルの値をテキストに変換し、その結果のテキストをウェーブに格納します。

デフォルトの設定は、1行目を列ラベルとして使います。

つまり結果は次のようになります。

読み込むデータ (Excel : 左) と読み込み後 (Igor Pro : 右)

	A	B	C	D
1	January	Test Data 1	"Quoted Text 1"	
2	February	Test Data 2	"Quoted Text 2"	
3	March	Test Data 3	"Quoted Text 3"	
4	April	Test Data 4	"Quoted Text 4"	
5	May	Test Data 5	"Quoted Text 5"	
6				

Point	January	Test_Data_1	X_Quoted_Text_1
0	February	Test Data 2	"Quoted Text 2"
1	March	Test Data 3	"Quoted Text 3"
2	April	Test Data 4	"Quoted Text 4"
3	May	Test Data 5	"Quoted Text 5"
4			

このような場合、Load Excel File ダイアログで、Use default names (e.g. ColumnA) または Use base name plus digit (e.g. wave0) を選択して1行目をデータとして処理するように設定します。

Point	ColumnA	ColumnB	ColumnC
0	January	Test Data 1	"Quoted Text 1"
1	February	Test Data 2	"Quoted Text 2"
2	March	Test Data 3	"Quoted Text 3"
3	April	Test Data 4	"Quoted Text 4"
4	May	Test Data 5	"Quoted Text 5"
5			

Deduce from row (行から推測する)

これは、異なるタイプの列 (数値、日付、テキスト) の混合を読み込むのに適した方法です。

XLLoadWave にどの行を調べるかを指定します。

XLLoadWave はその行のセルを調べます。

指定された列について、セルが数値の場合、XLLoadWave は数値ウェーブを作成し、セルがテキストの場合、テキストウェーブを作成します。

読み込むデータ (Excel : 左) と読み込み後 (Igor Pro : 右)

	A	B	C	D	E	F	G
1	2024/12/10 8:00	2.97	1.95	1.00	8.10	2.42	
2	2024/12/11 8:00	3.09	4.08	1.90	7.53	4.87	
3	2024/12/12 8:00	3.18	5.91	1.74	6.90	1.77	
4	2024/12/13 8:00	3.12	2.05	1.05	8.21	2.54	
5	2024/12/14 8:00	3.18	4.26	1.90	7.70	5.02	
6							

Point	ColumnA	ColumnB	ColumnC	ColumnD
0	2024/12/10 08:00:00	2.97	1.95	1
1	2024/12/11 08:00:00	3.09	4.08	1.9
2	2024/12/12 08:00:00	3.18	5.91	1.74
3	2024/12/13 08:00:00	3.12	2.05	1.05
4	2024/12/14 08:00:00	3.18	4.26	1.9
5				

数値セルに Excel の組み込みの日付、時刻、日付/時刻フォーマットが使われている場合、XLLoadWave は Excel の日付/時刻フォーマットの数値データを Igor の日付/時刻フォーマットに変換します。

XLLoadWave は、カスタムのセルフォーマットによってコントロールされているセルの日付と時刻のフォーマットを推測することはできません。

この場合、手動での変換方法の詳細については、後のセクション Excel Date/Time vs Igor Date/Time を参照してください。

XLLoadWave がこの方法で列タイプを推測する場合、Excel ファイル内の対応する列のビルトイン・セル形式に応じて、日付/時刻ウェーブ用の Igor のテーブルの列形式を日付、時刻、日付/時刻のいずれかに設定します。

Use column type string (列形式文字列を使う)

異なるタイプの列 (数値、日付、テキスト) が混在していて、「deduce from row」の方法で正しい推測ができない場合に、この方法を使います。

例えば、一部のファイルでは、列タイプを推測するのに適した行が 1 つもない場合があります。

この方法では、読み込む各列の形式を識別する文字列を指定します。

例えば、「1T1D3N」という文字列は、

1T：最初の列をテキストウェーブに読み込む

1D：次の列を日付/時刻ウェーブに読み込む

3N：次の 3 列を数値ウェーブに読み込む

ことを意味します。

文字列でカバーされているよりも多くの列をロードした場合、余分な列は数値としてロードされます。

また、

N：すべての列が数値である

D：すべての列が日付/時刻である

T：すべての列がテキストである

ことを意味します。

文字列には空白やその他の余分な文字を含めてはいけません。

正しい文字列の例を次に示します。

N すべての列が数値

T すべての列がテキスト

1T1D3N 1 列目がテキスト、2 列目が日付/時刻、次の 3 列が数値

1T1N3T25N 1 列目がテキスト、2 列目が数値、次の 3 列がテキスト、その次の 25 列が数値

テキスト	数値	テキスト	テキスト	テキスト	数値	数値	以降数値
------	----	------	------	------	----	----	------

treat all columns as numeric 設定では、数値列内のテキストセルは空白として扱われます。

この動作は、XLLoadWave が導入される以前のバージョンと互換性があります。

use column type string 設定では、数値列内のテキストセルを検出すると、そのテキストセルを数値に変換します。

テキストが有効な数値を表している場合 (例: "1.234")、ウェーブに有効な数値が生成されます。

テキストが有効な数値を表していない場合 (例: "January")、ウェーブには空白が生成されます。

これは、不注意で数値列にテキストセルを含むファイルがある場合に便利です。

XLLoadWave とウェーブ名

Load Excel File ダイアログで見ることができるよう、XLLoadWave は作成するウェーブ名を生成するときに、3 つの方法のうちの一つを使います。

1) ワークシートで指定した行からウェーブ名を取得することができます。

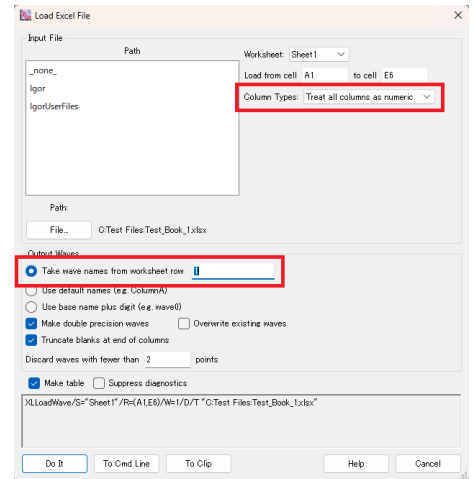
この場合、XLLoadWave は行に文字列値が含まれていることを想定しています。

Load Excel File ダイアログで、Take wave names from worksheet row を選択し、名前を取得する行番号を入力します。

結果は次のようになります。

Point	Pos_1	Pos_2	Pos_3	Pos_4	Pos_5
0	2.97	1.95	1.00	8.10	2.42
1	3.09	4.08	1.90	7.53	4.87
2	3.18	5.91	1.74	6.90	1.77
3	3.12	2.05	1.05	8.21	2.54
4	3.18	4.26	1.90	7.70	5.02

	A	B	C	D	E
1	Pos 1	Pos 2	Pos 3	Pos 4	Pos 5
2	2.97	1.95	1.00	8.10	2.42
3	3.09	4.08	1.90	7.53	4.87
4	3.18	5.91	1.74	6.90	1.77
5	3.12	2.05	1.05	8.21	2.54
6	3.18	4.26	1.90	7.70	5.02
7					



2) デフォルトのウェーブ名として、ColumnA、ColumnB などの形式で生成することができます。

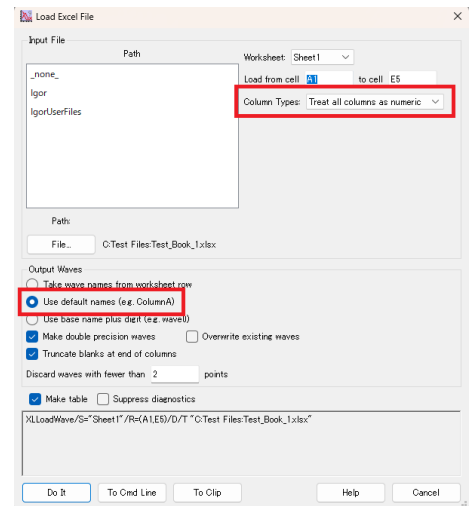
名前の末尾の文字は、そのウェーブが作成されたワークシートの列を示します。

Load Excel File ダイアログで、Use default names (e.g. ColumnA) を選択します。

結果は次のようになります。

Point	ColumnA	ColumnB	ColumnC	ColumnD	ColumnE
0	2.97	1.95	1.00	8.10	2.42
1	3.09	4.08	1.90	7.53	4.87
2	3.18	5.91	1.74	6.90	1.77
3	3.12	2.05	1.05	8.21	2.54
4	3.18	4.26	1.90	7.70	5.02

	A	B	C	D	E	F
1	2.97	1.95	1.00	8.10	2.42	
2	3.09	4.08	1.90	7.53	4.87	
3	3.18	5.91	1.74	6.90	1.77	
4	3.12	2.05	1.05	8.21	2.54	
5	3.18	4.26	1.90	7.70	5.02	
6						

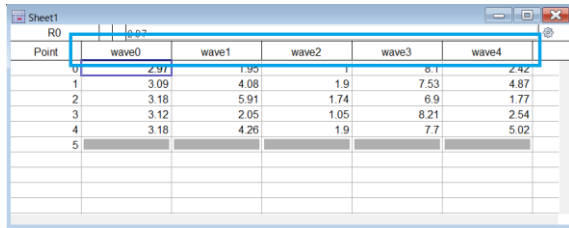


3) XLLoadWave は、この場合「wave」というベース名を使って、wave0、wave1 などの形式のウェーブ名を生成することができます。

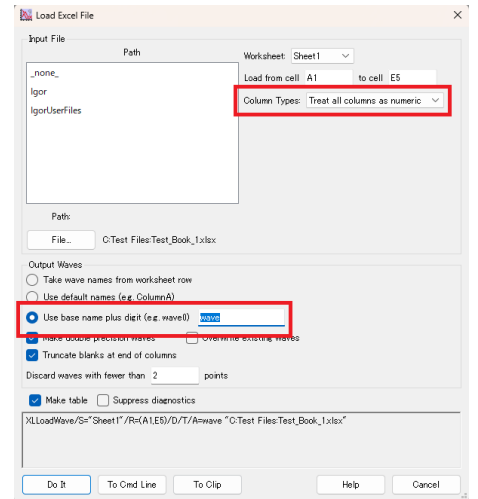
	A	B	C	D	E	F
1	2.97	1.95	1.00	8.10	2.42	
2	3.09	4.08	1.90	7.53	4.87	
3	3.18	5.91	1.74	6.90	1.77	
4	3.12	2.05	1.05	8.21	2.54	
5	3.18	4.26	1.90	7.70	5.02	
6						

Load Excel File ダイアログで、Use base name plus digit (e.g. wave0) を選択し、ベースとなる名前（ここでは wave）を入力します。

結果は次のようになります。



Point	wave0	wave1	wave2	wave3	wave4
0	2.97	1.95	1	8.1	2.42
1	3.09	4.08	1.9	7.53	4.87
2	3.18	5.91	1.74	6.9	1.77
3	3.12	2.05	1.05	8.21	2.54
4	3.18	4.26	1.9	7.7	5.02
5					



XLLoadWave コマンドは、ダイアログでは使うことができない4番目のウェーブの命名方法、/NAME フラグをサポートしています。

このフラグを使うと、セミコロンで区切られた文字列のリストを使って、各列の名前を指定することができます。

XLLoadWave が作成するウェーブの名前が、以下に説明するいくつかの状況により、予想とは異なるものになる場合があります。

このような場合、XLLoadWave は履歴領域に元の名前と新しい名前を表示します。

読み込み後、必要であれば、Rename コマンドを使って、別の名前を指定することができます。

ワークシート内の名前が長すぎる場合

XLLoadWave はそれを適切な長さに切り詰めます。

標準のウェーブ名で許可されていない文字が名前に含まれている場合

XLLoadWave はその文字をアンダースコアに置き換えます。

ワークシート内の2つの名前が競合する場合

XLLoadWave は「D_」などのプリフィックスを追加することで、2番目の名前をユニークなものにします。

この「D_」は、ウェーブに読み込まれる Excel の列を示しています。

ワークシート内の名前が既存のウェーブの名前と競合する場合

上書きオプションを使わない限り、XLLoadWave は1つ以上の数字を追加することで、読み込むウェーブの名前をユニークにします。

上書きオプションがオンの場合、読み込まれたデータは既存のウェーブを上書きします。

XLLoadWave が名前をユニークにするために1つ以上の数字を追加する必要があるが、名前の長さが既にウェーブ名の限界に達している場合

名前の途中から1つ以上の文字を削除します。

ワークシートのセルから取得した名前が、コマンド、関数、マクロの名前と競合する場合があります。

例えば、Date と Time は組み込み関数であるため、ウェーブにこれらの名前を付けることはできません。

このような競合が発生した場合、XLLoadWave は名前を変更し、履歴領域に元の名前と新しい名前を示すメッセージを表示します。

XLLoadWave の出力変数

XLLoadWave は、ファイルローダーの出力変数である V_flag、S_path、S_fileName、S_waveNames を標準として設定します。

さらに、S_worksheetName をワークブックファイル内の、ロードされたワークシートの名前に設定します。

Excel Date/Time vs. Igor Date/Time

Excel は、1900 年 1 月 1 日または 1904 年 1 月 1 日からの日数単位で日付/時刻情報を保存します。

Windows では 1900 年が、Macintosh では 1904 年がデフォルトです。

Igor は、1904 年 1 月 1 日からの秒単位で日付を保存します。

列タイプを決定するときに、Treat all columns as date、Deduce from row、Use column type string を使うと、XLLoadWave が Excel 形式から Igor 形式に自動的に変換します。

Treat all columns as numeric を使う場合は、Excel 形式から Igor 形式に手動で変換する必要があります。

Excel ファイルが 1904 年を基準年として使っている場合 (Macintosh) は、変換は次のようになります。

```
wave *= 24*3600 // 日を秒に変換 (wave = wave*24*3600)
```

Excel ファイルが 1900 年を基準年として使っている場合 (Windows) は、変換は次のようになります。

```
wave *= 24*3600 // 日を秒に変換 (wave = wave*24*3600)
wave -= 24*3600*365.5*4 // 4年の差分を処理
```

365 ではなく 365.5 を使っているのは、閏年を考慮したためです。

Microsoft の 1900 の日付システムは 1900 年 1 月 1 日を 0 日ではなく、1 日としています (うるう年の日と開始日の日の 2 日分が必要)。

時刻データを表形式で表示する場合、Excel と Igor で 1 秒の誤差が生じる場合があります。

例えば、Excel では「9:00:30」と表示される場合でも、Igor では「9:00:29」と表示されることがあります。

これは、Excel のデータが基準時間 (Nominal time) にわずかに満たないためです。

この例では、Excel のセルには「9:00:30」からミリ秒を引いた値が含まれます。

Excel が時刻を表示するときには、四捨五入されます。

Igor が時刻を表示するときには切り捨てられます。

これが気になる場合は、Igor のウェーブのデータを丸めることができます。

```
wave = round(wave)
```

この四捨五入を行うことで、データ内の小数点以下の秒数はすべて削除されます。

そのため、XLLoadWave では自動的に丸め処理は行いません。

Excel Data を 2D ウェーブに読み込む

XLLoadWave は 1D ウェーブを作成します。1D ウェーブを 2D ウェーブに変換する関数を以下に示します。

```
Function LoadExcelNumericDataAsMatrix(pathName, fileName, worksheetName,
startCell, endCell)
```

```

String pathname          // Igor シンボリックパスまたは "" でダイアログを表示
String filename         // 読み込むファイル名または "" でダイアログを表示
String worksheetName
String startCell        // 例: "B1"
String endCell          // 例: "J100"

if ((strlen(pathName)==0) || (strlen(fileName)==0))
    // ファイルを指定するダイアログを表示
    Variable refNum
    String filters = "Excel Files (*.xls,*.xlsx,*.xslm):.xls,.xlsx,.xslm;"
    filters += "All Files:.*;"
    Open/D/R/P=$pathName /F=filters refNum as filename
    fileName = S_fileName          // S_fileName は Open/D によって設定される
    if (strlen(fileName) == 0)     // ユーザーがキャンセル?
        return -2
    endif
endif

endif

// 行1を数値ウェーブに読み込む
XLLoadWave/S=worksheetName/R=($startCell,$endCell)/COLT="N"/O/V=0/K=0/Q filename
if (V_flag == 0)
    return -1                      // ユーザーがキャンセル
endif

String names = S_waveNames          // S_waveNames は XLLoadWave によって作成される
String nameOut = UniqueName("Matrix", 1, 0)
Concatenate /KILL /O names, $nameOut // マトリックスを作成し、1D ウェーブをキルする

String format = "Created numeric matrix wave %s containing cells %s to %s in
                worksheet ¥"%s¥"¥r"
Printf format, nameOut, startCell, endCell, worksheetName

```

End

1D ウェーブとして読み込んだ場合：読み込むデータ（Excel：左）と読み込み後（Igor Pro：右）

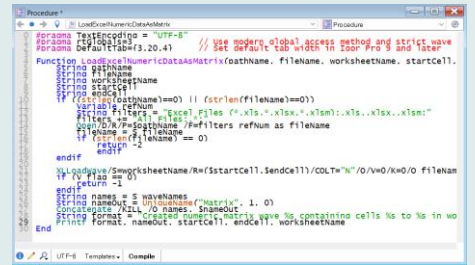
	A	B	C
1	2.97	4.56	
2	3.09	4.87	
3	3.18	5.01	
4	3.12	4.98	
5	2.89	4.46	
6	3.11	4.77	
7	2.91	4.89	
8			

Point	ColumnA	ColumnB
0	2.97	4.56
1	3.09	4.87
2	3.18	5.01
3	3.12	4.98
4	2.89	4.46
5	3.11	4.77
6	2.91	4.89
7		

上記の関数を使うステップは次の通りです。

1. メニュー **Windows** → **Procedure Windows** → **Procedure Window** を選択し、上記のコードを貼り付けます（改行やコメントなどを整理しておく为宜）。

画面下の **Compile** ボタンを押します。



```
forName TextEncoding = "UTF-8"
forName CellFormat(1:3,20:4) // Set default tab name for each sheet
function LoadExcelNumericDataAsMatrix(pathName, fileName, worksheetName, startCell, ...
    stopCell, endCell)
    % Load Excel numeric data as matrix
    % Arguments: pathName, fileName, worksheetName, startCell, stopCell, endCell
    % Returns: Matrix of numeric data

    % Check for file existence
    if ~exist(pathName, 'file') || ~exist(fileName, 'file')
        error('File does not exist: %s', pathName);
    end

    % Read Excel file
    xls = xlsread(pathName, worksheetName, startCell:stopCell);

    % Initialize matrix
    Matrix = zeros(size(xls));

    % Loop through sheets
    for i = 1:length(xls)
        % Get sheet name
        sheetName = xls{i}.Name;

        % Read data from sheet
        xls{i} = xlsread(pathName, sheetName, startCell:stopCell);

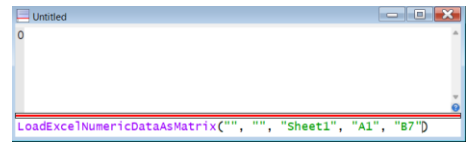
        % Store data in matrix
        Matrix(:,:,i) = xls{i};
    end

    % Save matrix to workspace
    save('Matrix.mat', 'Matrix');

    % Display matrix size
    fprintf('Matrix size: %s\n', mat2str(size(Matrix)));
end
```

2. 今回は、ファイルはダイアログから選択できるような形にするため、最初の2つの引数は "" として、コマンドラインに次を入力して実行します。

```
LoadExcelNumericDataAsMatrix("", "",
    "Sheet1", "A1", "B7")
```



```
Untitled
0
LoadExcelNumericDataAsMatrix("", "", "Sheet1", "A1", "B7")
```

3. ファイルを選択するダイアログが開くので、Excel ファイルを選択します。

Data Browser には1つのウェーブのみが表示され、ダブルクリックしてテーブルを表示させると 2D ウェーブになっていることがわかります。

