

CONTENTS

サンプルの Experiment – Kruskal-Wallis Tests Demo	2
クイックノート	2
手順.....	2
StatsKWTest コマンドのヘルプ.....	4
StatsNPMCTest コマンドのヘルプ.....	5

サンプルの Experiment – Kruskal-Wallis Tests Demo

クイックノート

メニュー **File** → **Example Experiments** → **Statistics** → **Kruskal-Wallis Tests Demo**

この Experiment は、Kruskal-Wallis (クラスカル・ウォリス) 検定を説明するデモです。

Kruskal-Wallis 検定は、Statistics メニューではサポートされていません。
コマンドウィンドウでコマンドを使って実行します。

手順

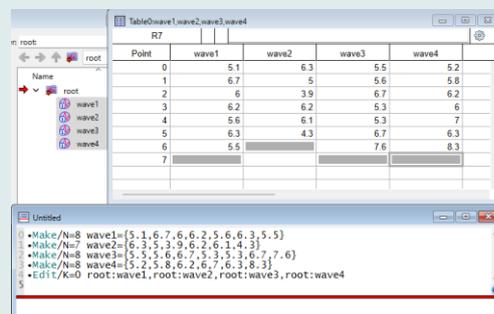
新しい Experiment を作成したところからの手順で確認します。

最初に4つのウェーブを生成します。

データは、単一要因の4つの測定グループを表しています。

1. コマンドウィンドウで次を入力します。

```
Make/N=8 wave1={5.1,6.7,6,6.2,5.6,6.3,5.5}  
Make/N=7 wave2={6.3,5,3.9,6.2,6.1,4.3}  
Make/N=8 wave3={5.5,5.6,6.7,5.3,5.3,6.7,7.6}  
Make/N=8 wave4={5.2,5.8,6.2,6,7,6.3,8.3}
```



2. コマンドウィンドウで次を実行します。

```
StatsKWTest /T=1 wave1,wave2,wave3,wave4
```

サンプルの Experiment では /Q が付いていますが、ここでは指定しません。

3. Kruskal-Wallis Test テーブルが表示されます。

次のような結果が表示されています。

Num_Groups	4
Num_Valid_Data	6
Alpha	0.05
H	2.93763
Hc_Chi_square_approx	7.81473
P_Chi_square_approx	0.401342
Hc_Wallace_approx	7.47045
P_Wallace_approx	0.418545

The screenshot shows the 'Kruskal-Wallis Test' results table with the following data:

Point	W_KWTestResults.1	W_KWTestResults
0	Num_Groups	4
1	Num_Valid_Data	27
2	Alpha	0.05
3	H	2.93763
4	Hc_Chi_square_approx	7.81473
5	P_Chi_square_approx	0.401342
6	Hc_Wallace_approx	7.47045
7	P_Wallace_approx	0.418545
8		

H 統計量は明らかに臨界値を下回っているため、 H_0 (入力ウェーブのデータは同じ) が受け入れられます。

次に、同じ要因の5番目のデータグループ wave5 を作成し、wave4 の代わりに wave5 を使って検定します。

4. コマンドウィンドウで次を実行します。

```
Make/N=6 wave5={6.7,7.3,7,6.7,7.2,7.1,7.2}
```

コマンドウィンドウで次を実行します。

```
StatsKWTest /T=1 wave1,wave2,wave3,wave5
```

ここでも /Q は指定しません。

Point	W_KWTestResults.I	W_KWTestResults
0	Num_Groups	4
1	Num_Valid_Data	27
2	Alpha	0.05
3	H	12.677
4	Hc_Chi_square_approx	7.81473
5	P_Chi_square_approx	0.00538994
6	Hc_Wallace_approx	7.47045
7	P_Wallace_approx	0.00160907
8		

5. Kruskal-Wallis Test テーブルが表示されます。

次のような結果が表示されています。

```

Num_Groups          4
Num_Valid_Data      27
Alpha               0.05
H                  12.677
Hc_Chi_square_approx 7.81473
P_Chi_square_approx 0.00538994
Hc_Wallace_approx   7.47045
P_Wallace_approx    0.00160907
    
```

この場合、統計値 H は明らかに H_0 を棄却すべきであることを示していますが、変動が主に wave5 によるものであることを示すものではありません。

6. そこで、多重比較検定を行います。

コマンドウィンドウで次を実行します。

```
StatsNPMCTest/T=1/DHW wave1,wave2,wave3,wave5
```

ここでも /Q は指定しません。

Row	M	NPMCDHWRe	M	NPMCDHWRe	M	NPMCDHWRe	M	NPMCDHWRe	M	NPMCDHWRe	M	NPMCDHWRe	Conclusion
0	x	3_vs_1	14.7381	4.39832	3.35084	2.63826							0
1		3_vs_0	11.1429	4.22577	2.63688	2.63826							1
2		3_vs_2	9.28571	4.22577	2.1974	2.63826							1
3		2_vs_1	5.45238	4.39832	1.23965	2.63826							1
4		2_vs_0	1.85714	4.22577	0.43948	2.63826							1
5		0_vs_1	3.59524	4.39832	0.817411	2.63826							1
6													

結果は NP Multiple Comparison (DHW) テーブルに表示されます。

ここで注意することは、ラベルの列にウェーブ名が表示されず、インデックスが表示されることです。wave1=0, wave2=1, wave3=2, wave5=3 となります。

	Diff	SE	Q	Qc	Conclusion
3_vs_1	14.7381	4.39832	3.35084	2.63826	0
3_vs_0	11.1429	4.22577	2.63688	2.63826	1
3_vs_2	9.28571	4.22577	2.1974	2.63826	1
2_vs_1	5.45238	4.39832	1.23965	2.63826	1
2_vs_0	1.85714	4.22577	0.43948	2.63826	1
0_vs_1	3.59524	4.39832	0.817411	2.63826	1

Q 統計値を確認すると、変動のほとんどは wave5 と他のウェーブとの間であり、wave5 と wave2 の差が最大であることが分かります。

このコマンドでは、入力ウェーブの長さが同じではないため、/DHW フラグを使用したことに注意してください。

StatsKWTest コマンドのヘルプ

StatsKWTest [/ALPH=*significance* /Q/Z/T=*k* /WSTR=*wList*] [*wave1*, *wave2*,... *wave100*]

StatsKWTest コマンドは、データの順位を使用して分散をテストするノンパラメトリックな Kruskal-Wallis 検定を実行します。

出力は、現在のデータフォルダー内の W_KWTestResults ウェーブに書き込まれます。

フラグ

/ALPH=*val* 有意水準を設定します (デフォルトは 0.05)。

/E Klotz と Teng のアルゴリズムを使用して正確な P 値を計算します。
大規模なデータセットでは、長時間の計算が必要になる場合があります。
ユーザー中止キーの組み合わせ (Shift+Esc) を押すと計算を中止でき、それ以降のすべての結果は有効なままとなり、正確な P 値は NaN に設定されます。

/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。

/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

/WSTR=*waveListString*
サンプルデータを含む複数のウェーブをセミコロンで区切ったリストを含む文字列を指定します。
フラグの後に各ウェーブを列挙する代わりに、*waveListString* を使います。

/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、それ以外はゼロに設定されます。

詳細

入力は、2 つ以上の 1 次元数値ウェーブ (サンプルの各グループごとに 1 つ) です。
欠損データには NaN を使用するか、異なるポイント数のウェーブを使用します。

StatsKWTest は、カイ二乗近似と Wallace 近似の両方を使用して、常に臨界値を計算します。
適切な場合 (データセットが十分に小さい場合) は、/E を使って正確な P 値を取得することができます。

計算に多くのウェーブやデータポイントが関わる場合、正確な臨界値の計算には非常に長い時間がかかることがあります。

すべての結果は、現在のデータフォルダー内のウェーブ W_KWTestResults に保存され、オプション (/T) でテーブルに表示されます。

ウェーブには次の情報が含まれています：

行	データ
0	グループの数
1	有効なデータポイントの数 (NaN を除く)
2	アルファ
3	Kruskal-Wallis H 統計値
4	臨界値 Hc に対するカイ二乗近似
5	P 値に対するカイ二乗近似
6	臨界値 Hc に対する Wallace 近似

- 7 P 値に対する Wallace 近似
- 8 正確な P 値 (/E が必要)

Kruskal-Wallis 検定の H_0 は、すべての入力ウェーブが同じであるというものです。検定に失敗し、入力が 2 つ以上のウェーブで構成されていた場合、それらのウェーブの一部に一致する可能性があることを示すものではありません。

さらに詳細な分析については、次の StatsNPMCTest を参照してください。

V_flag は、エラーが発生した場合は -1 に、それ以外はゼロに設定されます。

参照

Zar, J.H., Biostatistical Analysis, 4th ed., 929 pp., Prentice Hall, Englewood Cliffs, New Jersey, 1999.

Wallace, D.L., Simplified Beta-Approximation To The Kruskal-Wallis H Test, Jour. Am. Stat. Assoc., 54, 225-230, 1959.

J.H. Klotz, Computational Approach to Statistics.

Klotz, J. and Teng, J. (1977). One-way layout for counts and the exact enumeration of the Kruskal-Wallis H distribution with ties. Journal of the American Statistical Association 72, 165-169.

関連するヘルプ・コマンド

Statistical Analysis, StatsWilcoxonRankTest, StatsNPMCTest, StatsAngularDistanceTest

StatsNPMCTest コマンドのヘルプ

```
StatsNPMCTest [/Q/Z/ALPH=val /SWN/T=k /TAIL=tailCode /WSTR=waveListString  
/TUK/SNK/DHW/CIDX=controlIndex /CONW=contrastWave] [wave1, wave2,... wave100]
```

StatsNPMCTest コマンドは、ノンパラメトリックな多重比較検定を多数実行します。実行した検定に応じて、出力されたウェーブは現在のデータフォルダーに保存されます。一部の検定は、すべてのグループでサンプル数が同じである場合にのみ適切です。StatsNPMCTest は通常、StatsANOVA1Test または StatsKWTest に続きます。

フラグ

/ALPH=val 有意水準を設定します (デフォルトは 0.05)。

/CIDX=controlIndex

入力リストのゼロから始まる controlIndex ウェーブで指定されたコントロールグループについて、ノンパラメトリックな多重比較を行います。

出力は、現在のデータフォルダー内の M_NPCCResults ウェーブ、またはオプションでテーブルに出力されます。

出力列の内容は次のとおりです。

最初の列には、コントロールと各他のウェーブの順位和の差、

2 番目の列には標準誤差 (SE)、

3 番目の列には、順位和の差と SE の比として定義される統計量 q 、

4 番目の列には、テール指定 (/TAIL を参照) にも依存する臨界値、

5 番目の列には、0 で H_0 を棄却し、1 で H_0 を受け入れる結論が含まれます。

この検定の 1 つのバージョンは、すべての入力に同じ数のサンプルが含まれる場合に適用されます。

そうでない場合は、Dunn-Hollander-Wolfe 法を用いて適切な SE を計算し、同点の可能性を処理します。

/CONW=contrastWave

ノンパラメトリックな多重比較検定を実行します。

contrastWave は、各入力ウェーブに 1 ポイントずつあります。

対応する (ゼロベースの) 入力ウェーブを最初のグループに含めるには、contrastWave の値を 1 にします。

2 番目のグループに含めるには 2 に、またはウェーブを除外するにはゼロにします。

コントラスト自体は、第 1 グループと第 2 グループのランクの正規化された合計の差として定義されます。

contrastWave = {0,1,1,1,2} の場合、コントラストは以下のように計算されます。

$$\text{contrast: } \frac{1}{3} [R_{n1} + R_{n2} + R_{n3}] - R_{n4}$$

ここで、 R_{ni} は、対応する入力ウェーブからのサンプルの正規化された順位和です。実際の順位付けはすべてのサンプルのプールに対して実行されるため、コントラストウェーブにゼロを許容することの重要性を理解してください。

出力は、現在のデータフォルダー内の M_NPMConResults ウェーブ、またはオプションでテーブルに出力されます。

出力列の内容は次の通りです。

最初の値はコントラスト値、

2 番目は標準誤差 (SE)、

3 番目は統計量 S (SE に対するコントラストの絶対値の比)、

4 番目は (χ^2 近似値からの) 臨界値、

5 番目は結論で、0 の場合は H_0 を棄却し、1 の場合は H_0 を受け入れることを示します。

この検定は、異なるサンプル数の入力ウェーブをサポートし、同じ順位も処理できます。

ここで使われるコントラストウェーブは、StatsMultiCorrelationTest とは構造が異なります。

/DHW

Dunn-Holland-Wolfe 検定を行い、サンプル数の不均等をサポートし、順位和の同順位を考慮します。

出力は、現在のデータフォルダー内の M_NPMCDHWResults ウェーブ、またはオプションでテーブルに出力されます。

出力列の内容は次の通りです。

最初の列は、順位和の平均値の差 (順位和をグループ内のサンプル数で割ったもの)、

2 番目の列には標準誤差 (SE)、

3 番目の列には DHW 統計量 Q、

4 番目の列には臨界値、

5 番目の列には結論 (0 は H_0 を棄却し、1 は H_0 を受け入れる) が含まれます。

/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。

/SWN

比較テーブルの各行に対応するウェーブ名を含むテキストウェーブを作成します。

選択した検定に応じて、以下のウェーブが作成されます：

/CIDX test: T_NPCCRResultsDescriptors

/DHW test: T_NPMCDHWDescriptors

/SNK test: T_NPMCSNKResultsDescriptors

/TUK test: T_NPMCTukeyDescriptors

/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

/TAIL=tailCode /CIDX を持つ H_0 を指定します。

tailCode =1: 片側検定 ($\mu_c \leq \mu_a$)
tailCode =2: 片側検定 ($\mu_c \geq \mu_a$)
tailCode =4: デフォルト。両側検定 ($\mu_c = \mu_a$)
コードの組み合わせは許可されていません。

/SNK 標準誤差 SE が p (順位差) の関数である Student-Newman-Keuls 検定でノンパラメトリックなバリエーションを行います。
この検定では、すべてのグループでサンプル数が等しくなければなりません。
サンプル数が異なる場合は、/DHW を使用してください。

出力は、現在のデータフォルダー内の M_NPMCSNKResults ウェーブに書き込まれます。

出力列の内容は次の通りです。

最初の列には順位和の差、

2番目の列には標準誤差 (SE)、

3番目の列には p 値 (順位差)、

4番目の列には統計量、

5番目の列には臨界値、

6番目の列には結論 (0 は H_0 を棄却し、1 は H_0 を受け入れる) が含まれています。

この検定は、Tukey 検定 (/TUK) よりも差異に対してより敏感です。

/TUK

順位和の差を用いて、Tukey 型 (Nemenyi) 多重比較検定を行う。

これは、検定フラグを何も指定しない場合に実行されるデフォルトです。

この検定では、すべてのウェーブで同じ数のポイントが必要です。

異なるサイズを使用する場合は、/DHW を使います。

出力は、現在のデータフォルダー内の M_NPMCTukeyResults ウェーブに書き込まれます。

出力列の内容は次の通りです。

最初の列には順位和の差分、

2番目の列には SE 値、

3番目の列には統計量 q、

4番目の列にはこの特定のアルファ値とグループ数に対する臨界値、

最後の列には結論フラグが含まれ、0 は H_0 の棄却、1 は H_0 を受け入れます。

H_0 は、ペアになっている平均値は同じであると仮定するものです。

/WSTR=waveListString

サンプルデータを含む複数のウェーブをセミコロンで区切ったリストを含む文字列を指定します。

フラグの後に各ウェーブを列挙する代わりに、waveListString を使います。

/Z

エラーを無視します。

詳細

StatsNPMCTest への入力、2 つ以上の有効なエントリーを含む 2 つ以上の 1 次元数値ウェーブ (サンプルのグループごとに 1 つのウェーブ) です。

SNK 検定および TUK 検定を使うには、ウェーブのポイント数が同じでなければなりません。

そうでなければ、長さが異なるウェーブに対しては、Dunn-Holland-Wolfe 検定 (/DHW) を使わなければなりません。

V_flag は、実行エラーが発生していない場合はゼロに設定されます。

例えば、等しい数のポイントを必要とする検定の入力ウェーブでサンプル数が異なる場合、個々の検定が失敗することがあります。

StatsNPMCTest は失敗した検定をスキップし、V_flag は失敗した検定を特定するバイナリの組み合わせとなります。

V_flag & 1 Tukey が失敗 (/TUK)

V_flag & 2 Student-Newman-Keuls が失敗 (/SNK)

V_flag は、その他のエラーの場合には -1 に設定されます。

関連するヘルプ・コマンド

Statistical Analysis, StatsANOVA1Test, StatsKWTest

パラメトリック検定における多重比較については、StatsDunnettTest、StatsScheffeTest も参照。