

○: Mantel's extensions of Mantel-Haenszel test ... 層別化された複数の2xk分割表における用量・反応関係の要約検定¹。「拡張Mantel-haenszel検定」として紹介されている²。

⇒2xk分割表における Cochran-Armitage test / Trend χ^2 test を複数分割表に拡張した検定である

1. Chi-square tests with one degree of freedom; extensions of the Mantel-Haenszel procedure. N. Mantel J. Amer. Statist. Assoc. 1963;58;690-700.

2. Mantel-Haenszelの方法による複数の2x2表の要約 佐藤俊哉 統計数理 1998;46;153-177

○: 単一分割表に対するMantel's extensions of Mantel-Haenszel test 統計量と Cochran-Armitage test / Trend χ^2 test統計量の関係

・i番目の分割表における変数を下記の通り定義する。

i_th contingency table

order of genotype	genotype distribution x_i							total	
	0	1	2	...	s_{ij}	...	$s_{i(k-1)}$		
phenotype y_i	1	n_{i0}	n_{i1}	n_{i2}	...	n_{ij}	...	$n_{i(k-1)}$	t_i
	0	$N_{i0}-n_{i0}$	$N_{i1}-n_{i1}$	$N_{i2}-n_{i2}$...	$N_{ij}-n_{ij}$...	$N_{i(k-1)}-n_{i(k-1)}$	T_i-t_i
total	-	N_{i0}	N_{i1}	N_{i2}	...	N_{ij}	...	$N_{i(k-1)}$	T_i

$$\sum_j x_{ij} = \sum_j s_{ij} N_{ij}.$$

$$\sum_j x_{ij}^2 = \sum_j s_{ij}^2 N_{ij}.$$

$$\sum_j y_{ij} = \sum_j (0 \cdot (N_{ij} - n_{ij}) + 1 \cdot n_{ij}) = t_i.$$

$$\sum_j y_{ij}^2 = \sum_j (0^2 \cdot (N_{ij} - n_{ij}) + 1^2 \cdot n_{ij}) = t_i.$$

$$\sum_j x_{ij} y_{ij} = \sum_j (0 \cdot s_{ij} (N_{ij} - n_{ij}) + 1 \cdot s_{ij} n_{ij}) = \sum_j s_{ij} n_{ij}.$$

$$S_{xx}^{(i)} = \sum_j x_{ij}^2 - \frac{\left(\sum_j x_{ij}\right)^2}{T_i} = \frac{T_i \sum_j s_{ij}^2 N_{ij} - \left(\sum_j s_{ij} N_{ij}\right)^2}{T_i}.$$

$$S_{yy}^{(i)} = \sum_j y_{ij}^2 - \frac{\left(\sum_j y_{ij}\right)^2}{T_i} = \frac{t_i(T_i - t_i)}{T_i}.$$

$$S_{xy}^{(i)} = \sum_j x_{ij} y_{ij} - \frac{\sum_j x_{ij} \sum_j y_{ij}}{T_i} = \frac{T_i \sum_j s_{ij} n_{ij} - t_i \sum_j s_{ij} N_{ij}}{T_i}.$$

・複数分割表から得られるMantel's extensions of Mantel-Haenszel test 統計量を $X^2_{Extended-MH}$ とする。文献1より、

$$X^2_{Extended-MH} = \frac{\left[\sum_i \left(\sum_j x_{ij} y_{ij} - \frac{\sum_j x_{ij} \sum_j y_{ij}}{T_i} \right)^2 \right]}{\sum_i V \left(\sum_j x_{ij} y_{ij} \right)} = \frac{\left[\sum_i \left(\sum_j x_{ij} y_{ij} - \frac{\sum_j x_{ij} \sum_j y_{ij}}{T_i} \right)^2 \right]}{\sum_i \left[\frac{1}{(T_i - 1)} \left(\sum_j x_{ij}^2 - \frac{\left(\sum_j x_{ij}\right)^2}{T_i} \right) \left(\sum_j y_{ij}^2 - \frac{\left(\sum_j y_{ij}\right)^2}{T_i} \right) \right]}. \quad \dots \textcircled{1}$$

・ $S_{xx}^{(i)}, S_{yy}^{(i)}, S_{xy}^{(i)}$ を代入し、

$$X^2_{Extended-MH} = \frac{\left(\sum_i S_{xy}^{(i)} \right)^2}{\sum_i \left(\frac{1}{T_i - 1} S_{xx}^{(i)} S_{yy}^{(i)} \right)}. \quad \dots \textcircled{2}$$

・分割表の数が一つの時、 $X^2_{Extended-MH}$ は Cochran-Armitage test / Trend χ^2 test統計量に等しい。

$$X^2_{Extended-MH} = \frac{S_{xy}^2}{\frac{1}{T-1} S_{xx} S_{yy}} = (T-1)r^2 = X^2_{Cochran-Armitage}. \quad X^2_{Extended-MH} = \frac{S_{xy}^2}{\frac{1}{T} S_{xx} S_{yy}} = Tr^2 = X^2_{Trend}.$$

* : ①式の分母にて (T_i-1) を採用した場合は Cochran-Armitage test 統計量に、 T_i を採用した場合はTrend χ^2 test統計量に等しくなる。

○:実データでの検証

下記の二つの分割表に対する Mantel's extensions of Mantel-Haenszel test 統計量を算出した。

Table 1.

score	genotype distribution						total
	4	3	2	2	1	0	
case	208	326	83	156	63	10	846
control	175	340	66	183	88	16	868
total	383	666	149	339	151	26	1714

Table 2.

score	genotype distribution						total
	4	3	2	2	1	0	
case	188	312	77	169	73	10	829
control	186	343	102	196	101	12	940
total	374	655	179	365	174	22	1769

②式に対応するjavaソース(ページ下段)による出力結果

Table 1: $X^2_{\text{Cochran-Armiage}} = 6.981, p=0.00824$

Table 2: $X^2_{\text{Cochran-Armiage}} = 3.902, p=0.0482$

Table 1・Table 2を合算して一つの分割表とみなした場合: $X^2_{\text{Cochran-Armiage}} = 10.869, p=0.000978$

Table 1・Table 2に対してメタアナリシスを行った場合: $X^2_{\text{Extended-MH}} = 10.641, p=0.00111$

○:検定手法の呼称に関して

文献2は「拡張Mantel-Haenszel検定」という呼称で紹介しているが、web上ではCochran-Mantel-Haenszel検定³も「拡張Mantel-Haenszel検定」として紹介されている例が見受けられる。

⇒論文中の記載は”Mantel's extensions of Mantel-Haenszel test (1963)”とし、引用文献(1)をつける方針とする。

3. Categorical Data Analysis. Alan Agresti. New York: Wiley. 2002. p293-297.

```
public static double ExtendedMantel (int[][][] array, double[] score) {
    double ret;
    int numTable = array.length;
    int numCol = array[0][0].length;
    double[] Sxx = new double[numTable];
    double[] Syy = new double[numTable];
    double[] Sxy = new double[numTable];
    double[] T = new double[numTable];
    double[] t = new double[numTable];
    double[] s2N = new double[numTable];
    double[] sN = new double[numTable];
    double[] sn = new double[numTable];
    double numer = 0;
    double denom = 0;
    for (int i=0;i<numTable;i++) {
        Sxx[i] = 0;
        Syy[i] = 0;
        Sxy[i] = 0;
        T[i] = 0;
        t[i] = 0;
        s2N[i] = 0;
        sN[i] = 0;
        sn[i] = 0;
        for (int j=0;j<numCol;j++) {
            T[i] += (double)(array[i][0][j]+array[i][1][j]);
            t[i] += (double)array[i][0][j];
            s2N[i] += (double)score[j]*score[j]*(array[i][0][j]+array[i][1][j]);
            sN[i] += (double)score[j]*(array[i][0][j]+array[i][1][j]);
            sn[i] += (double)score[j]*array[i][0][j];
        }
        Sxx[i] = (T[i]*s2N[i]-sN[i]*sN[i])/T[i];
        Syy[i] = t[i]*(T[i]-t[i])/T[i];
        Sxy[i] = (T[i]*sn[i]-t[i]*sN[i])/T[i];
    }
    for (int i=0;i<numTable;i++) {
        numer += Sxy[i];
        denom += Sxx[i]*Syy[i]/(T[i]-1);
    }
    ret = numer*numer/denom;
    return ret;
}
```