TESTING SIMULTANEOUS MARGINAL HOMOGENEITY IN CLUSTERED MATCHED-PAIR MULTINOMIAL DATA

by

Bo Deng

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

BIOSTATISTICS

Department of Mathematical and Statistical Sciences
University of Alberta

Abstract

For matched-pair data with a multinomial reponse, the Stuart-Maxwell test (1955, 1970) and the Bhapkar test (1966) are commonly used to test the marginal homogeneity. However, in medical research, many studies for assessing safety consider multiple multinomial endpoints to detect the treatment effects. To test the simultaneous marginal homogeneity (SMH) in such clustered matched-pair multinomial data, three overall tests are proposed. Furthermore, when the outcome is ordinal, three ordinal statistics which test SMH against stochastic ordering are proposed.

To evaluate the performance of our methods, we generated a total of 5000 clustered matched-pair data sets, considering number of endpoints $= 2, 3, 4$ and sample size ranging from 25 to 200. Then our methods are applied to these 5000 datasets and the empirical size and power are compared. The simulation shows that the Score-type tests perform well with respect to nominal size and power even with small sample size. For ordinal endpoints, the ordinal statistic provides uniformly larger power than the one which does not utilize the ordinal feature.

Acknowledgements

I would like to extend my deep gratitude to Dr. Keumhee Carriere Chough, my supervisor, for her enormous support during my academic pursuit. Without her guidance and encouragement, I would have been lost when I was making the important decisions on my academic and career path. If I could only name one thing that I gained from this bumpy journey of pursuing my academic ambition, it is the privilege to get to know her, her care for students, her strong work ethic. Thank you, Dr. Keumhee Carriere Chough, for everything.

My sincere gratitude also goes to Dr. Linglong Kong and Dr. Rohana Karunamuni for their invaluable comments and advice on my thesis and the insightful questions they raised on the committee meeting.

In addition, I owe a special debt to Dr. N. G. Prasad from whom I learned what I should prepare for starting a career in the area of Biostatistics. Dr. Prasad is one of the most knowledgeable and experienced statistician I have met in Canada. His success in industry inspires me to further pursue my dream.

Lastly and most importantly, I'm deeply thankful to my wife and parents. There are no words in the world that could possibly describe their love and support for me. I love them with all my heart.

Table of Contents

List of Tables

Chapter 1

Introduction

Definition 1.1 *Categorical variable* is the variable which can only take on a limited and fixed number of values, such as marital status and gender.

A categorical variable can be nominal or ordinal. For instance, The nominal variable, gender, consists of two types, male and female, while such ordinal variables as the severity (none, mild, moderate and severe) of an adverse event (AE), stages of a disease (Stage I, II, III), etc, taking ordinal nature of the characteristics of the measurement.

Definition 1.2 *Matched-pair data* is the data of two samples when each observation in one sample is paired with an observation in the other sample.

Matched-pair data usually occurs in longitudinal studies in which each subject is observed over time or crossover studies. It also occurs when the unit of observation is a cluster, such as two observations on ears from one subject, two siblings in one family, etc. Due to matching, the outcomes in two samples are dependent.

Analysis of the matched-pair categorical data are extremely useful and popular in assessing the safety, toxicity and quality-of-life in clinical trials. For instance, in phase II-III clinical trial of pharmaceutical products, the analysis of AE data is an important aspect of examining the safety of a new drug. In recent years, crossover designs gained much popularity because of the advantages of requiring fewer subjects and better controlling confounding than other designs. The crossover design is a repeated measurements design in that each subject receives different treatments or doses during the different time periods, i.e., the subjects cross over from one treatment to another treatment or dose during the course of the trial. In crossover designs, comparing the incidence of AEs under different doses leads to the clustered matched-pair binary data.

For such categorical matched-pair data, some researchers suggested to analyze each AE separately and combine individual P-values through various multiple adjustment techniques. However, many AEs are correlated to a huge extent. Therefore, analyzing the incidence of related AEs simultaneously is more ideal than treating them separately. Analyzing the incidence of AEs simultaneously is challenging, as it leads to dealing with clustered and thus correlated matched-pair data.

This thesis focuses on the analysis of clustered matched-pair multinomial data. Three overall tests are proposed which test the homogeneity of the marginal distributions of each AE's severity under two treatments or different dosages. We develop the test statistics and their asymptotic distribution. Furthermore, their empirical size and power under different settings are examined by simulation. In addition, as AE's severity measure is an ordinal outcome, three statistics, which test the simultaneous marginal homogeneity against stochastic ordering are proposed. Their empirical size and power under different settings are also examined and compared. The thesis is framed entirely in terms of a safety analysis comparing the marginal proportions of each AE severity categories under two different treatments or different dosages. However, our methods can be applied to any paired or repeated clustered multinomial response.

Chapter 2

Literature Review

2.1 Overview

There are various methods developed to compare two independent or dependent proportions or vectors of proportions, including non-parametric methods (for instance, Pearson, likelihood-ratio Chi-square Tests, or Mcnemar's Test) and model-based methods (for instance, Logistic Regression with or without Random Effect). In this chapter, we mainly focus on non-parametric methods.

2.2 Comparing Two Independent Proportions

For two independent multinomial samples data, Pearson (1900) proposed a Chi-square statistic to test the homogeneity of distributions of the two samples. In a $I \times J$ contingency table, the Pearson Chi-square statistic is:

$$X^2 = \sum_i \sum_j \frac{(n_{ij} - \hat{\mu}_{ij})^2}{\hat{\mu}_{ij}}, \qquad \text{where} \quad \hat{\mu}_{ij} = n_{i+}n_{+j}/n.$$

X^2 asymptotically follows a Chi-square distribution with $df = (I-1)(J-1)$ when $n \longrightarrow \infty$.

Another frequently used statistic to test the homogeneity of proportions is the likelihood-ratio test. The likelihood-ratio test is formulated as

$$G^2 = 2\sum_i \sum_j n_{ij} log \frac{n_{ij}}{\hat{\mu}_{ij}}, \qquad \text{where} \quad \hat{\mu}_{ij} = n_{i+}n_{+j}/n.$$

G^2 also asymptotically follows a Chi-square distribution with $df = (I-1)(J-1)$ when $n \longrightarrow \infty$ (Agresti, 2002).

Although the above Chi-square tests can be used to test the homogeneity of proportions, they both have the limitation that the $\hat{\mu}_{ij} = n_{i+}n_{+j}/n$ used in X^2 and G^2 does not depend on the order of rows and columns. No matter how we arbitrarily reorder the rows or columns, X^2 and G^2 do not change. Therefore, the two statistics both treat rows and columns as nominal. When one is ordinal, the test statistic that utilize the ordinal feature is certain to be more powerful than the above two (Agresti, 2002).

When the row variable X and column variable Y are ordinal, we are often interested in testing the linear trend in the association between X and Y. Mantel (1963) propsed a test statistic, which assigns scores to levels of X and Y and summarizes the linear trend. The scores $u_1 \leq u_2 \leq \cdots \leq u_I$ are assigned to the X categories and $\upsilon_1 \leq \upsilon_2 \leq \cdots \leq \upsilon_J$ are assigned to the Y categories. Let $\overline{u} = \sum_i u_i p_{i+}$ denote the marginal mean of the row scores and let $\overline{\upsilon} = \sum_j \upsilon_j p_{+j}$ denote the marginal mean of the column scores. Thus, the sample covariance of X and Y equals $\sum_{i,j}(u_i - \overline{u})(\upsilon_j - \overline{\upsilon})p_{ij}$. The correlation between X and Y is

$$r = \frac{\sum_{i,j}(u_i - \overline{u})(\upsilon_j - \overline{\upsilon})p_{ij}}{\sqrt{[\sum_i (u_i - \overline{u})^2 p_{i+}]\left[\sum_j (\upsilon_j - \overline{\upsilon})^2 p_{+j}\right]}}.$$

To test the linear trend in the association between X and Y, a test statistic is

$$M^2 = (n-1)r^2,$$

where n is the sample size. M^2 asmptotically follows a Chi-square distribution with $df = 1$ (Agresti, 2007).

The above M^2 statistic treat X and Y as ordinal. However, when one of them is nominal, it could still be used. When X is binary, the $2 \times J$ table usually occurs in comparing the distribution of two groups. For instance, the two rows stand for the two treatments. We assign the scores $(u_1 = 0, u_2 = 1)$ to the two levels of X. Then M^2 is measuring the differences between the two row mean scores on Y (Agresti, 2007).

2.3 Comparing Two Dependent Proportions

To summarize the categorical matched-pair data, the two-way contingency table with the same row and column categories is used. Table 2.1 is an example of matched-pair categorical data. In a survey of n university students, n_{1+} students indicated approval of the president's performance. In the second survey four months later, n_{+1} students indicated approval of these same students.

For categorical matched-pair data, we are often interested in comparing

Table 2.1: Rating of Performance of the University President

	Second Survey		
First Survey	Approve	Disapprove	Total
Approve	n_{11}	n_{12}	n_{1+}
Disapprove	n_{21}	n_{22}	n_{2+}
Total	n_{+1}	n_{+2}	n

the row and column marginal distribution of response. For such a two-way contingency table, let π_{ab} denote the probability of outcome a for the first observation and outcome b for the second observation. Let n_{ab} denote the number of such pairs in the sample. Correspondingly, $p_{ab} = n_{ab}/n$ estimates π_{ab} and is a sample proportion of the joint event (a, b), where n is the total number of subjects. Then p_{a+} is the marginal proportion with outcome a for the first observation and p_{+a} is the marginal proportion with outcome a for the second observation. In practice, we are interested in comparing the marginal proportions p_{a+} and p_{+a}. However, the two samples are matched and dependent. The dependence between the two samples makes the marginal proportions correlated. Thus the methods for comparing the independent proportions can not be applied. When $p_{a+} = p_{+a}$ for each possible outcome a, this is called marginal homogeneity.

For binary response, the McNemars test(McNemar, 1947) is a simple way to test the marginal homogeneity in 2×2 tables. In 2×2 tables, the null hypothesis of marginal homogeneity is $H_0 : \pi_{1+} = \pi_{+1}$. Let $d = \hat{\pi}_{1+} - \hat{\pi}_{+1}$ denote the difference of two marginal sample proportions. Under H_0, the estimated variance of d is

$$\widehat{Var}(d) = \frac{n_{12} + n_{21}}{n^2}.$$

Hence, a Wald test statistic is established as

$$z = \frac{d}{\sqrt{\widehat{Var}(d)}} = \frac{n_{21} - n_{12}}{(n_{21} + n_{12})^{1/2}}.$$

Then $z^2 = \frac{(n_{21} - n_{12})^2}{n_{12} + n_{21}}$ asymptotically follows a Chi-square distribution with $df = 1$, resulting in the McNemar's test (McNemar, 1947).

For multinomial outcome, the null hypothesis of marginal homogeneity is: H_0: $\pi_{1+} = \pi_{+1}, \pi_{2+} = \pi_{+2}, \cdots, \pi_{I+} = \pi_{+I}$. Where I is the number of outcome categories. Bhapkar (1966) proposed a statistic to test the marginal homogeneity of the distributions of multinomial outcomes. Let $\hat{d}_a = \hat{\pi}_{a+} - \hat{\pi}_{+a}$, and $\boldsymbol{d} = (\hat{d}_1, \cdots, \hat{d}_{I-1})^T$. The sample covariance matrix $\hat{\boldsymbol{V}}$ of $\sqrt{n}\boldsymbol{d}$ has elements as follows:

$$\begin{aligned} \hat{V}_{ab} &= -(p_{ab} + p_{ba}) - (p_{+a} - p_{a+})(p_{+b} - p_{b+}) \qquad \text{for } a \neq b \\ \hat{V}_{aa} &= p_{+a} + p_{a+} - 2p_{aa} - (p_{+a} - p_{a+})^2. \end{aligned}$$

Under the marginal homogeneity, we have the Wald statistic $W = n\boldsymbol{d}^T\hat{\boldsymbol{V}}^{-1}\boldsymbol{d}$ which asymptotically follows a Chi-square distribution with $df = I - 1$.

Stuart (1955) and Maxwell (1970) proposed $W_0 = n\boldsymbol{d}^T(\hat{\boldsymbol{V}}_{\boldsymbol{0}})^{-1}\boldsymbol{d}$, which uses the sample null covariance matrix $\hat{\boldsymbol{V}}_{\boldsymbol{0}}$ and based on a score test. $\hat{\boldsymbol{V}}_0$ has elements as:

$$\hat{V}_{ab} = -(p_{ab} + p_{ba}) \qquad \text{for } a \neq b$$
$$\hat{V}_{aa} = p_{+a} + p_{a+} - 2p_{aa}.$$

Bhapkar test (1966) and Stuart-Maxwell test (1955, 1970) both treat the outcome as nominal. They can be used to test marginal homogeneity against any alternatives. When the outcome is ordinal, they ignore the ordinal nature of the outcome. Hence, the tests that utilize the ordinal information should be more powerful for testing marginal homogeneity against alternatives considering the ordinal feature. For instance, for ordinal outcome, one is usually interested in if the classifications based on one variable are higher than those based on the other variable. For a $I \times I$ square table, let Y_1 denote the observation from the row marginal distribution $\{\pi_{i+}\}$ and Y_2 denote the observation from the column marginal distribution $\{\pi_{+j}\}$. Y_1 is stochastically higher than Y_2 (Agresti, 2010) if the cumulative density function of Y_1 is uniformly below the cumulative density function of Y_2, i.e.

$$\pi_{1+} + \cdots + \pi_{j+} \leq \pi_{+1} + \cdots + \pi_{+j}, \qquad \text{for } j = 1, \cdots, I-1.$$

This means that Y_1 is more likely to have larger values than Y_2. The statistic that tests the marginal homogeneity against stochastically ordered margins can be more powerful than Bhapkar and Stuart-Maxwell tests.

Agresti (1983) proposed a statistic to test marginal homogeneity against stochastically ordered margins by comparing marginal mean scores. Let $u_1 \leq u_2 \leq \cdots \leq u_I$ denote the scores assigned to the outcome categories. Marginal homogeneity implies that $E(Y_1) = E(Y_2)$, where $E(Y_1) = \sum_i u_i \pi_{i+}$ and $E(Y_2) = \sum_i u_i \pi_{+i}$. The sample mean responses are

$$\overline{y}_1 = \sum_i u_i p_{i+} \qquad \text{and} \qquad \overline{y}_2 = \sum_i u_i p_{+i}.$$

The sample standard error of $\overline{y}_1 - \overline{y}_2$ is

$$SE = \sqrt{\frac{\sum_i \sum_j (u_i - u_j)^2 p_{ij} - (\overline{y}_1 - \overline{y}_2)^2}{n}}.$$

The Wald test statistic to test the marginal homogeneity is formed by $z = (\overline{y}_1 - \overline{y}_2)/SE$, which asymptotically follows $N(0, 1)$. The corresponding score test is given by $z = (\overline{y}_1 - \overline{y}_2)/SE_0$, where $SE_0 = \sqrt{\frac{\sum_i \sum_j (u_i - u_j)^2 p_{ij}}{n}}$. Agresti (1983) compared its power with the Stuart-Maxwell test's through simulation. Simulation results showed that the ordinal test has uniformly larger power than the Stuart-Maxwell test when they are used to test marginal homogeneity against stochastic ordering.

2.4 Comparing Two Independent Vectors of Proportions

In the developmental process of pharmaceutical products, investigators are interested in testing if there is a difference between the incidence of several adverse events for an experimental drug and the placebo (or the active control), based on results from a randomized controlled clinical trial. Chuang-Stein and Mohberg (1993) introduced a global test statistic formed by using the difference between marginal sample proportions.

For two independent samples, we denote the group by $j = 1$ for an experimental drug group with n_1 subjects and $j = 2$ for a placebo group with n_2 subjects. Let K denote the number of binomial variables, which constitute the multivariate responses. For subject i in group j, let $\boldsymbol{Y_{ij}} = (Y_{ij1}, Y_{ij2} \cdots Y_{ijK})^T$ denote the $K \times 1$ vector of responses, where $Y_{ijk} = 1$ if adverse event k is present and $Y_{ijk} = 0$ if adverse event k is absent, $k = 1, 2, \cdots, K$. Assume $(\boldsymbol{Y_{11}}, \cdots, \boldsymbol{Y_{n_1 1}})$ are n_1 independently and identically distributed random variables from a multinomial distribution and $(\boldsymbol{Y_{12}}, \cdots, \boldsymbol{Y_{n_2 2}})$ are n_2 independently and identically distributed random variables from a multinomial distribution with the number of categories as 2^K.

Let $\pi_j(k) = Pr(Y_{ijk} = 1)$, where $k = 1, 2, \cdots, K$ and $j = 1, 2$. Then $\boldsymbol{\pi_j} = (\pi_j(1), \pi_j(2), \pi_j(3), \cdots, \pi_j(K))^T$ are K one-way marginal probabilities for the 2^K cross-classification of responses for group j. For the two vectors of binomial parameters $\boldsymbol{\pi_1} = (\pi_1(1), \pi_1(2), \pi_1(3), \cdots, \pi_1(K))^T$ and $\boldsymbol{\pi_2} = (\pi_2(1), \pi_2(2), \pi_2(3), \cdots, \pi_2(K))^T$, the null hypothesis of simultaneous marginal homogeneity (SMH) is

$$H_0 : \pi_1(k) = \pi_2(k), \qquad k = 1, 2, \cdots, K.$$

Let $\boldsymbol{d} = (\hat{d}_1, \cdots, \hat{d}_K)^T$ with $d_k = \hat{\pi}_2(k) - \hat{\pi}_1(k)$, $k = 1, \cdots, K$. Let $\boldsymbol{V}$ denote the covariance matrix of $\boldsymbol{d}$. Then, $\boldsymbol{V}$ has elements as:

$$Var(\hat{d}_k) = \pi_1(k)(1 - \pi_1(k))/n_1 + \pi_2(k)(1 - \pi_2(k))/n_2$$
$$Cov(\hat{d}_k, \hat{d}_{k'}) = Cov(\hat{\pi}_1(k), \hat{\pi}_1(k')) + Cov(\hat{\pi}_2(k), \hat{\pi}_2(k')) = \nu_1 + \nu_2,$$

where

$$\nu_1 = \{P(y_{1k} = 1, y_{1k'} = 1) - P(y_{1k} = 1)P(y_{1k'} = 1)\}/n_1$$

$$\nu_2 = \{P(y_{2k} = 1, y_{2k'} = 1) - P(y_{2k} = 1)P(y_{2k'} = 1)\}/n_2.$$

Let $\hat{\boldsymbol{V}}$ denote the sample version of $\boldsymbol{V}$. A Wald statistic to test SMH is $W = \boldsymbol{d}^T \hat{\boldsymbol{V}}^{-1} \boldsymbol{d}$, which asymptotically follows a Chi-square distribution with $df = K$ under the null hypothesis. Let $\hat{\boldsymbol{V}}_0$ denote the pooled estimate of $\boldsymbol{V}$ over the two groups. Then $W_0 = \boldsymbol{d}^T \hat{\boldsymbol{V}}_0^{-1} \boldsymbol{d}$ is a score-type statistic which also has an asymptotic null Chi-square distribution with $df = K$. It can be easily extended to multinomial responses.

In practice, we often run into ordinal responses, such as the severity of adverse events in drug safety studies, quality-of-life scale, etc. Klingenberg et al (2008) proposed a score-type statistic to test SMH in clustered ordinal data of two independent samples.

Consider the case of comparing two treatments based on observing K ordinal variables with possibly different number of categories. For subject i in group j, let $\boldsymbol{Y}_{ij} = (Y_{ij1}, Y_{ij2}, \cdots, Y_{ijK})^T$ be the $K \times 1$ vector of reponses, where Y_{ijk} is ordinal with $C_k \geq 2$ categories, $k = 1, 2, \cdots, K$. The number of subjects in two treatment groups are n_1 and n_2, respectively. Let $\{\pi_{jk}(c_k) = Pr(Y_{ijk} = c_k), c_k = 1, \cdots, C_k\}$ denote the marginal probability of observing outcome c_k of variable k at dose j. Then $\boldsymbol{\pi}_j = (\pi_{j1}(1), \pi_{j1}(2), \cdots, \pi_{j1}(C_1), \pi_{j2}(1), \cdots, \pi_{jK}(C_K))^T$ are $\sum_{k=1}^{K} C_k$ one-way marginal probabilities for the $2^{\sum_{k=1}^{K} C_k}$ cross-classification of responses, with $\hat{\boldsymbol{\pi}}_j$ as the corresponding marginal sample proportions. The null hypothesis of SMH is

$$H_0 : \pi_{1k}(c_k) = \pi_{2k}(c_k), \qquad k = 1, 2, \cdots, K \qquad c_k = 1, 2, \cdots, C_K$$

Let $\boldsymbol{d} = \hat{\boldsymbol{\pi}}_2 - \hat{\boldsymbol{\pi}}_1 = (\hat{d}_1(1), \cdots, \hat{d}_1(C_1), \hat{d}_2(1), \cdots, \hat{d}_2(C_2), \cdots, \hat{d}_K(1), \cdots, \hat{d}_K(C_K))^T$. denote the difference of the marginal sample proportions at two dosages. The $Cov(\boldsymbol{d})$ has elements as:

$$Var(\hat{d}_k(c_k)) = Var(\hat{\pi}_{2k}(c_k) - \hat{\pi}_{1k}(c_k)) = \sum_{j=1}^{2} \pi_{jk}(c_k)(1 - \pi_{jk}(c_k))/n_j,$$

$$Cov(\hat{d}_k(c_k), \hat{d}_{k'}(c_{k'})) = \sum_{j=1}^{2} (\delta_{kk'} \pi_{jkk'}(c_k, c_{k'}) - \pi_{jk}(c_k)\pi_{jk'}(c_{k'}))/n_j.$$

Let $\boldsymbol{A} = diag(\boldsymbol{u_k}^T, k = 1, \cdots, K)$ be a score matrix with score $\boldsymbol{u_k}^T = (u_k(1), u_k(2), \cdots, u_k(C_k))^T$ for severity levels of adverse event k. Then $\boldsymbol{S} = \boldsymbol{Ad} = \boldsymbol{A}\hat{\boldsymbol{\pi}}_2 - \boldsymbol{A}\hat{\boldsymbol{\pi}}_1$ compares mean scores among the two treatments, with covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{A}\widehat{Cov}(\boldsymbol{d})\boldsymbol{A}^T$, where $\widehat{Cov}(\boldsymbol{d})$ is the sample version of $Cov(\boldsymbol{d})$. A score-type statistic for testing the simultaneous marginal homogeneity is $W_0 = \mathbf{1}^T \hat{\boldsymbol{\Sigma}}_0^{-1/2} \boldsymbol{S}/K$, where $\hat{\boldsymbol{\Sigma}}_0 = \boldsymbol{A}\widehat{Cov_0}(\boldsymbol{d})\boldsymbol{A}^T$ is obtained by replacing $\hat{\pi}_{jk}(c_k)$ in $\widehat{Cov}(\boldsymbol{d})$ with the pooled estimate over the two treatments. The W_0 asymptotically follows a Chi-square distribution with *df* $= K$ under the null hypothesis (Klingenberg et al., 2008).

2.5 Comparing Two Dependent Vectors of Proportions

In longitudinal or crossover studies, investigators are typically interested in testing whether significant difference exists in incidences of AEs at two or more occasions or different dosages. Unlike the parallel design, the data collected at different occasions are dependent. For each individual AE, the McNemar's test (McNemar, 1947) is frequently used for comparing the incidence rate of two dependent samples. Klingenberg and Agresti (2006) proposed a multivariate extension of McNemar's test to compare the incidence rates of several AEs.

Consider clustered matched-pair binary data with K binary variables indicating the incidence of K AEs at two dosages. For subject i at dose j, $Y_{ijk} = 1$ if adverse event k is present and $Y_{ijk} = 0$ if adverse event k is absent, where $k = 1, 2, \cdots, K$, $j = 1, 2$. Let $\boldsymbol{Y}_i = (Y_{i1}, Y_{i2})^T = (Y_{i11}, \cdots, Y_{i1K}, Y_{i21}, \cdots,$ $Y_{i2K})^T$ denote the $2K$ dimension binary responses for subject i. It may be assumed that $\boldsymbol{Y}_i$ follows a multinomial distribution, with a total sample size equals n. Let $\boldsymbol{\pi}_j(k) = Pr(Y_{ijk} = 1)$ denote the marginal probability of observing adverse event k at dose j. Then $\boldsymbol{\pi} = (\pi_1(1), \cdots, \pi_1(K), \pi_2(1), \cdots, \pi_2(K))^T$ is a $2K \times 1$ vector of marginal proportions for the 2^{2K} cross-classification of responses. Similarly, let $\pi_j(k, k^{'}) = Pr(Y_{ijk} = 1, Y_{ijk'} = 1)$ and $\pi(k, k^{'}) = Pr(Y_{i1k} = 1, Y_{i2k'} = 1)$ denote the second-order marginal probabilities.

The null hypothesis of SMH is

$$H_0 : \pi_1(k) = \pi_2(k), \qquad k = 1, 2, \cdots, K.$$

Let $\boldsymbol{d} = (\hat{d}_1, \cdots, \hat{d}_K)^T$ with $\hat{d}_k = \hat{\pi}_1(k) - \hat{\pi}_2(k), k = 1, \cdots, K$. The covariance matrix $\boldsymbol{V}$ of $\boldsymbol{d}$ has elements as:

$$
\begin{aligned}
Var(\hat{d}_k) &= \{\pi_1(k) + \pi_2(k) - 2\pi(k,k) - [\pi_1(k) - \pi_2(k)]^2\}/n \\
Cov(\hat{d}_k, \hat{d}_{k'}) &= \{\pi_1(k,k^{'}) + \pi_2(k,k^{'}) - \Big[\pi(k,k^{'}) + \pi(k^{'},k)\Big] \\
&\quad -[\pi_1(k) - \pi_2(k)]\Big[\pi_1(k^{'}) - \pi_2(k^{'})\Big]\}/n.
\end{aligned}
$$

A Wald statistic that tests SMH is $W = \boldsymbol{d}^T\hat{\boldsymbol{V}}^{-1}\boldsymbol{d}$, where $\hat{\boldsymbol{V}}$ is obtained by replacing $\pi_j(k)$, $\pi_j(k,k^{'})$ and $\pi(k,k^{'})$ in V with the corresponding sample proportions $\hat{\pi}_j(k)$, $\hat{\pi}_j(k,k^{'})$ and $\hat{\pi}(k,k^{'})$. The W has an asymptotic Chi-square distribution with $df = K$ under the null hypothesis.

Let $\hat{\boldsymbol{V}}_0$ denote the pooled estimate of $\boldsymbol{V}$, which is obtained by replacing $\pi_j(k)$ in $\boldsymbol{V}$ with $\pi_0(k) = (\pi_1(k)+\pi_2(k))/2$. Then $W_0 = \boldsymbol{d}^T\hat{\boldsymbol{V}}_0^{-1}\boldsymbol{d}$ is a score-type statistic which also has an asymptotic Chi-square distribution with $df = K$ under the null hypothesis.

Chapter 3

Multivariate Test of Marginal Homogeneity

3.1 Marginal Homogeneity

Let $\mathbf{Y}_{ij} = (Y_{ij1}, Y_{ij2}, \cdots, Y_{ijK})^T$ be the $K \times 1$ vector of multivariate responses for subject i at dose $j = 1, 2$, where K is the number of ordinal variables, Y_{ijk} is the ordinal variable (AE severity) with $C_k > 2$ categories, $k = 1, 2, \cdots, K$. In this thesis, $C_k = C = 4$ is used for all k which denotes the 4 severity levels of AE, i.e. None, Mild, Moderate and Severe. If subject i experienced AE k with severity c_k at dose j, then $Y_{ijk} = c_k$. For each subject, let $\mathbf{Y}_i = (\mathbf{Y}_{i1}, \mathbf{Y}_{i2})^T = (Y_{i11}, Y_{i12}, \cdots, Y_{i1K}, Y_{i21}, Y_{i22}, \cdots, Y_{i2K})^T$ denote the subject i's AE severity profile. Assume that we have n subjects in the study, $(\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_n)$ are n independently and identically distributed random variables from a multinomial distribution with probability $\boldsymbol{\pi}(c_1, c_2, \cdots, c_K, c_1^{'}, c_2^{'}, \cdots c_K^{'})$, where $\boldsymbol{\pi}(c_1, c_2, \cdots, c_K, c_1^{'}, c_2^{'}, \cdots, c_K^{'})$ denotes the joint probability $Pr(Y_{i11} = c_1, \cdots, Y_{i1K} = c_K, \cdots, Y_{i21} = c_1^{'}, \cdots, Y_{i2K} =$

$c_K^{'}$).

Let $\boldsymbol{\pi}_j = (\pi_{j1}(1), \cdots, \pi_{j1}(C), \pi_{j2}(1), \cdots, \pi_{j2}(C), \cdots, \pi_{jK}(1), \cdots, \pi_{jK}(C))$, where $\pi_{jk}(c_k)$ denotes the probability $Pr(Y_{ijk} = c_k), c_k = 1, 2, \cdots, C$, $C = 4$. The null hypothesis of SMH is defined as

$$H_0 : \pi_{1k}(c_k) = \pi_{2k}(c_k) \quad \text{for} \quad k = 1, 2, \cdots, K, \qquad c_k = 1, 2, \cdots, C, \qquad C = 4.$$

3.2 Wald and Score-type Tests of SMH

Motivated by the statistic proposed by Agresti and Klingenberg (2005) and Klingenberg and Agresti (2006), a statistic to test SMH is constructed by comparing the marginal proportions of each AE at two dosages. Let $\hat{\boldsymbol{\pi}}_j = (\hat{\pi}_{j1}(1), \cdots, \hat{\pi}_{j1}(C-1), \hat{\pi}_{j2}(1), \cdots, \hat{\pi}_{j2}(C-1), \cdots, \hat{\pi}_{jK}(1), \cdots, \hat{\pi}_{jK}(C-1))^T$ denote the marginal proportions of each AE at dose j, where $j = 1, 2$ and $\hat{\pi}_{jk}(c_k)$ denotes the sample proportion of subjects with severity c_k of AE k at dose j, $c_k = 1, 2 \cdots, C-1$. Let $\boldsymbol{d} = \hat{\boldsymbol{\pi}}_{\boldsymbol{2}} - \hat{\boldsymbol{\pi}}_1 = (\hat{d}_1(1), \cdots, \hat{d}_1(C-1), \hat{d}_2(1), \cdots, \hat{d}_2(C-1), \cdots, \hat{d}_K(1), \cdots, \hat{d}_K(C-1))^T$ denote the difference of the marginal sample proportions at two dosages.

Under the assumption of multinomial distribution, the covariance matrix

$\boldsymbol{V}$ of $\boldsymbol{d}$ has elements:

$$
\begin{aligned}
Var(\hat{d}_k(c_k)) &= Var(\hat{\pi}_{2k}(c_k) - \hat{\pi}_{1k}(c_k)) \\
&= Var(\hat{\pi}_{2k}(c_k)) + Var(\hat{\pi}_{1k}(c_k)) - 2Cov(\hat{\pi}_{2k}(c_k), \hat{\pi}_{1k}(c_k)) \\
&= \frac{\pi_{2k}(c_k)(1-\pi_{2k}(c_k))}{n} + \frac{\pi_{1k}(c_k)(1-\pi_{1k}(c_k))}{n} \\
&\quad - \frac{2(\pi_k(c_k, c_k) - \pi_{1k}(c_k)\pi_{2k}(c_k))}{n} \\
&= \frac{(\pi_{1k}(c_k) + \pi_{2k}(c_k) - 2\pi_k(c_k, c_k)) - (\pi_{1k}(c_k) - \pi_{2k}(c_k))^2}{n},
\end{aligned}
\tag{3.1}
$$

Where $\pi_k(c_k, c_k)$ is the probability of experiencing AE k of severity c_k at both dosages.

$$
\begin{aligned}
Cov(\hat{d}_k(c_k), \hat{d}_k(c_k^{'})) &= Cov(\hat{\pi}_{2k}(c_k) - \hat{\pi}_{1k}(c_k), \hat{\pi}_{2k}(c_k^{'}) - \hat{\pi}_{1k}(c_k^{'})) \\
&= Cov(\hat{\pi}_{2k}(c_k), \hat{\pi}_{2k}(c_k^{'})) - Cov(\hat{\pi}_{2k}(c_k), \hat{\pi}_{1k}(c_k^{'})) \\
&\quad - Cov(\hat{\pi}_{1k}(c_k), \hat{\pi}_{2k}(c_k^{'})) + Cov(\hat{\pi}_{1k}(c_k), \hat{\pi}_{1k}(c_k^{'})) \\
&= -\frac{\pi_{2k}(c_k)\pi_{2k}(c_k^{'})}{n} - \frac{\pi_{1k}(c_k)\pi_{1k}(c_k^{'})}{n} \\
&\quad - \frac{\pi_{12k}(c_k^{'}, c_k) - \pi_{1k}(c_k^{'})\pi_{2k}(c_k)}{n} \\
&\quad + \frac{\pi_{12k}(c_k, c_k^{'}) - \pi_{1k}(c_k)\pi_{2h}(c_k^{'})}{n},
\end{aligned}
\tag{3.2}
$$

Where $\pi_{12k}(c_k^{'}, c_k)$ is the joint probability of experiencing AE k of severity $c_k^{'}$

at dose 1 and AE k of severity c_k at dose 2.

$$\begin{aligned}
Cov(\hat{d}_k(c_k), \hat{d}_{k'}(c_{k'})) &= Cov(\hat{\pi}_{2k}(c_k) - \hat{\pi}_{1k}(c_k), \hat{\pi}_{2k'}(c_{k'}) - \hat{\pi}_{1k'}(c_{k'})) \\
&= Cov(\hat{\pi}_{2k}(c_k), \hat{\pi}_{2k'}(c_{k'})) - Cov(\hat{\pi}_{2k}(c_k), \hat{\pi}_{1k'}(c_{k'})) \\
&\quad - Cov(\hat{\pi}_{1k}(c_k), \hat{\pi}_{2k'}(c_{k'})) + Cov(\hat{\pi}_{1k}(c_k), \hat{\pi}_{1k'}(c_{k'})) \\
&= \frac{\pi_{2kk'}(c_k, c_{k'}) - \pi_{2k}(c_k)\pi_{2k'}(c_{k'})}{n} \\
&\quad - \frac{\pi_{12k'k}(c_{k'}, c_k) - \pi_{1k'}(c_{k'})\pi_{2k}(c_k)}{n} \\
&\quad - \frac{\pi_{12k'k}(c_k, c_{k'}) - \pi_{1k}(c_k)\pi_{2k'}(c_{k'})}{n} \\
&\quad + \frac{\pi_{1kk'}(c_k, c_{k'}) + \pi_{1k}(c_k)\pi_{1k'}(c_{k'})}{n},
\end{aligned} \tag{3.3}$$

where $\pi_{jkk'}(c_k, c_{k'})$ is the joint probability of experiencing AE k of severity c_k and AE $k^{'}$ of severity $c_{k'}$ at dose j and $\pi_{jj'kk'}(c_k, c_{k'})$ is the joint probability of experiencing AE k of severity c_k at dose j and AE $k^{'}$ of severity $c_{k'}$ at dose $j^{'}$.

Let $\hat{\boldsymbol{V}}$ denote the sample version of $\boldsymbol{V}$. Then, a Wald statistic to test SMH is

$$W = \boldsymbol{d}^T \hat{\boldsymbol{V}}^{-1} \boldsymbol{d}.$$

Based on the Central Limit Theorem, W has an asymptotic null Chi-square distribution with $df = K(C-1)$ when $n \longrightarrow \infty$.

By replacing $\pi_{1k}(c_k)$ and $\pi_{2k}(c_k)$ by the pooled estimate $\hat{\pi}_{0k}(c_k) = (\hat{\pi}_{1k}(c_k)+$

$\hat{\pi}_{2k}(c_k))/2$, we have $\hat{\boldsymbol{V}}_0$ as the pooled estimate of $\boldsymbol{V}$, which has elements as:

$$\begin{aligned}
\widehat{Var}(\hat{d}_k(c_k)) &= \frac{2(\hat{\pi}_{0k}(c_k) - \hat{\pi}_k(c_k, c_k))}{n} \\
\widehat{Cov}(\hat{d}_k(c_k), \hat{d}_k(c_k^{'})) &= \frac{-\hat{\pi}_{12k'k}(c_k^{'}, c_k) - \hat{\pi}_{12kk'}(c_k, c_k^{'})}{n} \\
\widehat{Cov}(\hat{d}_k(c_k), \hat{d}_{k'}(c_{k'})) &= \frac{\hat{\pi}_{1kk'}(c_k, c_{k'}) + \hat{\pi}_{2kk'}(c_k, c_{k'}) - \hat{\pi}_{12k'k}(c_{k'}, c_k) - \hat{\pi}_{12kk'}(c_k, c_{k'})}{n}.
\end{aligned} \tag{3.4}$$

Then we have a score-type statistic $W_0 = \boldsymbol{d}^T \hat{\boldsymbol{V}}_0^{-1} \boldsymbol{d}$, which also has an asymptotic null Chi-square distribution with $df = K(C-1)$ when $n \longrightarrow \infty$. In the binary case ($C = 2$), the W and W_0 reduces to the multivariate McNemar's tests (Klingenberg and Agresti, 2006).

3.3 A Non-Parametric Test of SMH

In section 3.2, it is assumed that $(\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_n)$ are n independently and identically distributed random variables from a multinomial distribution. However, this assumption may not be feasible in practice. To account for possible over-dispersion or under-dispersion, a non-parametric covariance estimate of $\boldsymbol{d}$ is considered in this section. The non-parametric covariance estimate $\hat{\boldsymbol{V}}_1$ of $\boldsymbol{d}$ is given as follows:

$$\begin{aligned}
\widehat{Var}(\hat{d}_k(c_k)) &= \widehat{Var}(\hat{\pi}_{2k}(c_k) - \hat{\pi}_{1k}(c_k)) \\
&= \widehat{Var}(\hat{\pi}_{2k}(c_k)) + \widehat{Var}(\hat{\pi}_{1k}(c_k)) - 2\widehat{Cov}(\hat{\pi}_{2k}(c_k), \hat{\pi}_{1k}(c_k)) \\
&= \frac{\sum_{i=1}^{n}(Y_{i2k}(c_k) - \overline{Y}_{2k}(c_k))^2}{n(n-1)} + \frac{\sum_{i=1}^{n}(Y_{i1k}(c_k) - \overline{Y}_{1k}(c_k))^2}{n(n-1)} \\
&\quad - \frac{\sum_{i=1}^{n} 2((Y_{i1k}(c_k) - \overline{Y}_{1k}(c_k))(Y_{i2k}(c_k) - \overline{Y}_{2k}(c_k))}{n(n-1)},
\end{aligned} \tag{3.5}$$

where $Y_{i1k}(c_k) = 1$ if subject i experience AE k of severity c_k; $Y_{i1k}(c_k) = 0$ if subject i did not experience AE k of severity c_k.

$$\begin{aligned}
\widehat{Cov}(\hat{d}_k(c_k), \hat{d}_{k'}(c_{k'})) &= \widehat{Cov}(\hat{\pi}_{2k}(c_k) - \hat{\pi}_{1k}(c_k), \hat{\pi}_{2k'}(c_{k'}) - \hat{\pi}_{1k'}(c_{k'})) \\
&= \widehat{Cov}(\hat{\pi}_{2k}(c_k), \hat{\pi}_{2k'}(c_{k'})) - \widehat{Cov}(\hat{\pi}_{2k}(c_k), \hat{\pi}_{1k'}(c_{k'})) \\
&\quad -\widehat{Cov}(\hat{\pi}_{1k}(c_k), \hat{\pi}_{2k'}(c_{k'})) + \widehat{Cov}(\hat{\pi}_{1k}(c_k), \hat{\pi}_{1k'}(c_{k'})) \\
&= \frac{\sum_{i=1}^{n}((Y_{i2k}(c_k) - \overline{Y}_{2k}(c_k))(Y_{i2k'}(c_{k'}) - \overline{Y}_{2k'}(c_{k'})))}{n(n-1)} \\
&\quad - \frac{\sum_{i=1}^{n}((Y_{i2k}(c_k) - \overline{Y}_{2k}(c_k))(Y_{i1k'}(c_{k'}) - \overline{Y}_{1k'}(c_{k'})))}{n(n-1)} \\
&\quad - \frac{\sum_{i=1}^{n}((Y_{i1k}(c_k) - \overline{Y}_{1k}(c_k))(Y_{i2k'}(c_{k'}) - \overline{Y}_{2k'}(c_{k'})))}{n(n-1)} \\
&\quad + \frac{\sum_{i=1}^{n}((Y_{i1k}(c_k) - \overline{Y}_{1k}(c_k))(Y_{1ik'}(c_{k'}) - \overline{Y}_{1k'}(c_{k'})))}{n(n-1)}.
\end{aligned} \tag{3.6}$$

Next, we will show (3.5) and (3.6) are consistent estimates of $Var(\hat{d}_k(c_k))$ and $Cov(\hat{d}_k(c_k), \hat{d}_{k'}(c_{k'}))$, respectively. First, we will show $\sum_{i=1}^{n}(Y_{i2k}(c_k) - \overline{Y}_{2k}(c_k))^2/n(n-1)$ in (3.5) is a consistent estimate of $Var(\hat{\pi}_{2k}(c_k))$. Assume $Y_{12k}, Y_{22k}, \cdots, Y_{n2k}$ are independently and identically distributed with mean μ and variance σ^2. It is well known that sample variance $\sum_{i=1}^{n}(Y_{i2k}(c_k) - \overline{Y}_{2k}(c_k))^2/(n-1)$ is a consistent estimate of σ^2. Furthermore, we have

$$Var(\hat{\pi}_{2k}(c_k)) = Var(\frac{\sum_{i=1}^{n}(Y_{i2k}(c_k))}{n}) = \frac{Var(Y_{i2k})}{n} = \frac{\sigma^2}{n}.$$

Hence, $\sum_{i=1}^{n}(Y_{i2k}(c_k) - \overline{Y}_{2k}(c_k))^2/n(n-1)$ is a consistent estimate of $Var(\hat{\pi}_{2k}(c_k))$. Similarly, it can be shown that $\sum_{i=1}^{n}(Y_{i1k}(c_k) - \overline{Y}_{1k}(c_k))^2/n(n-1)$ and $\sum_{i=1}^{n}((Y_{i1k}(c_k) - \overline{Y}_{1k}(c_k))(Y_{i2k}(c_k) - \overline{Y}_{2k}(c_k))/n(n-1)$ are consistent estimates of $Var(\hat{\pi}_{1k}(c_k))$ and $Cov(\hat{\pi}_{2k}(c_k), \hat{\pi}_{1k}(c_k))$, respectively. Therefore, (3.5) is a consistent estimate of $Var(\hat{d}_k(c_k))$.

Similarly, it can be shown that (3.6) is a consistent estimate of $Cov(\hat{d}_k(c_k), \hat{d}_{k'}(c_{k'}))$. Since (3.5) and (3.6) are both consistent, $\hat{\boldsymbol{V}}_1$ is a consistent estimate of variance of $\boldsymbol{d}$.

From the Central Limit Theorem, we have a non-parametric Wald test statistic $W_1 = \boldsymbol{d}^T \hat{\boldsymbol{V}}_1 \boldsymbol{d}$, which asymptotically follows a Chi-square distribution with $df = K(C-1)$ when $n \longrightarrow \infty$.

3.4 Tests of SMH against Stochastic Ordering

For clustered matched-pair AE severity data, it may be of interest to test whether one margin is stochastically higher than the other for each AE. Motivated by the statistic proposed by Agresti (1983), a statistic for testing SMH against stochastic ordering is formed by comparing the marginal mean scores under two treatments. Let $\hat{\boldsymbol{\pi}}_j = (\hat{\pi}_{j1}(1), \cdots, \hat{\pi}_{j1}(C), \hat{\pi}_{j2}(1), \cdots, \hat{\pi}_{j2}(C), \cdots, \hat{\pi}_{jk}(1), \cdots, \hat{\pi}_{jK}(C))^T$, where $\hat{\pi}_{jk}(c_k)$ denotes the sample proportion of subjects with severity c_k of AE k at dose j, $c_k = 1, 2 \cdots, C$. Let

$$\boldsymbol{d} = \hat{\boldsymbol{\pi}}_2 - \hat{\boldsymbol{\pi}}_1 = (\hat{d}_1(1), \cdots, \hat{d}_1(C), \hat{d}_2(1), \cdots, \hat{d}_2(C), \cdots, \hat{d}_K(1), \cdots, \hat{d}_K(C))^T$$

denote the difference of the marginal sample proportions at two dosages. The difference of the marginal mean scores at two dosages is formed by $\boldsymbol{S} = \boldsymbol{A}\boldsymbol{d}$, where $\boldsymbol{A} = diag(\boldsymbol{u_k}^T, k = 1, 2, \cdots, K)$ is a score matrix with score $\boldsymbol{u_k}^T = (u_k(1), u_k(2), \cdots, u_k(C))^T$ for severity levels of AE k.

Under the assumption of multinomial distribution, the covariance matrix $\boldsymbol{V}$ of $\boldsymbol{d}$ is given in (3.1), (3.2) and (3.3). From $\boldsymbol{S} = \boldsymbol{A}\boldsymbol{d}$, we have the covariance matrix of $\boldsymbol{S}$ as $\boldsymbol{\Sigma} = \boldsymbol{A}\boldsymbol{V}\boldsymbol{A}^T$. Let $\hat{\boldsymbol{\Sigma}}$ denote the sample version of $\boldsymbol{\Sigma}$. A multivariate Wald test is constructed by

$$W_{ord} = \boldsymbol{S}^T\hat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{S}$$

Based on the Central Limit Theorem, W_{ord} asymptotically follows a Chi-square distribution with $df = K$ when $n \longrightarrow \infty$, where K is the number of AEs considered simultaneously. When $K = 1$, it reduces to the statistic proposed by Agresti (1983).

By replacing $\pi_{1k}(c_k)$ and $\pi_{2k}(c_k)$ with the pooled estimate $\hat{\pi}_{0k}(c_k) = (\hat{\pi}_{1k}(c_k) + \hat{\pi}_{2k}(c_k))/2$, we have $\hat{\boldsymbol{V}}_0$ as the pooled estimate of $\boldsymbol{V}$, which is given in (3.4). Then we have a score-type statistic

$$W_{ord0} = \boldsymbol{S}^T\hat{\boldsymbol{\Sigma}}_0^{-1}\boldsymbol{S}$$

which also asmptotically follows the Chi-square distribution with $df = K$ when $n \longrightarrow \infty$, where $\hat{\boldsymbol{\Sigma}}_0 = \boldsymbol{A}\hat{\boldsymbol{V}}_0\boldsymbol{A}^T$.

The above two statistics assume that $(\boldsymbol{Y_1}, \boldsymbol{Y_2}, \cdots, \boldsymbol{Y_n})$ are n independently and identically distributed random variables from a multinomial distribution. Similar to section 3.3, a non-parametric covariance estimate of $\boldsymbol{d}$ can be considered, which is as given in (3.5) and (3.6). Then we have a non-parametric Wald test statistic $W_{ord1} = \boldsymbol{S}^T\hat{\boldsymbol{\Sigma}}_1^{-1}\boldsymbol{S}$ which also asymptotically follows a Chi-square distribution with $df = K$ when $n \longrightarrow \infty$, where $\hat{\boldsymbol{\Sigma}}_1 = \boldsymbol{A}\hat{\boldsymbol{V}}_1\boldsymbol{A}^T$.

Chapter 4

Simulation

4.1 Data Generation

In this chapter, a simulation study is performed to evaluate the performance of our methods. Five thousand simulated data sets under the characteristics of SMH are generated for sample sizes $n = 25, 50, 100, 200$ and the number of AEs $K = 2, 3, 4$, and the number of severity levels $C = 4$ comparing two treatments. For instance, to simulate a data set with $n = 100$, $K = 3$ and $C = 4$ under SMH, a random vector of 4^6 multinomial probabilities $\{\boldsymbol{\pi}(c_1, c_2, c_3, c_1^{'}, c_2^{'}, c_3^{'})\}$ is generated, where c_k and $c_k^{'}$ are the severity measures ranging from $1, \cdots, 4$ $(= C)$, $k = 2, 3, 4$ $(= K)$ for treatment A and treatment B, respectively. Then the iterative proportional fitting procedure (Deming and Stephan, 1940) is performed to adjust the vector to make the marginal probabilities of each adverse event under two treatments homogeneous. Then we randomly generated 5000 samples from a multinomial distribution with the given marginally homogeneous vector of probabilities.

4.2 Iterative Proportional Fitting

Iterative Proportional Fitting is a technique used for adjusting the cells proportionally in a two-way table to make the row margins or/and column margins equal to a set of selected values. It could also be applied in n-way tables to make the margins equal to the selected values for each dimension.

In two-way tables, let n_{ij} denote the unadjusted cells in the data. Correspondingly, n_{i+} and n_{+j} denote the row totals and column totals, respectively. To adjust n_{ij} such that they add up to the selected row and column margins, say m_{i+} and m_{+j}, the Iterative Proportional Fitting is conducted as follows:

Step 1: $m^{'}_{ij} = n_{ij}(m_{i+}/n_{i+})$

Step 2: $m^{''}_{ij} = m^{'}_{ij}(m_{+j}/m^{'}_{+j})$

Steps 3 and up: Repeat step 1 and step 2 until the row totals and column totals are both close enough to the m_{i+} and m_{+j}.

In n-way tables, Iterative Proportional Fitting is performed similarly as above.

Step 1: Divide each cell by the actual row total, then multiply by the selected row margin.

Step 2: Divide each cell by the actual column total, then multiply by the selected column margin.

Step 3: Divide each cell by the actual marginal total of the third dimension, then multiply by the selected margin of the third dimension.

$\vdots$

Step n: Divide each cell by the actual marginal total of the nth dimension, then multiply by the selected margin of the nth dimension.

Steps $n + 1$ and up: Repeat step 1 to n until the marginal totals of each dimension are all close enough to the selected margins.

4.3 Simulation Results

4.3.1 Software Package Introduction

In this section, we analyze the simulated data sets with R (Version 3.0.2). To fulfill the analysis in R, only the standard set of packages is required. No other package is required.

4.3.2 Simulation Settings

In a hypothesis testing problem, there are two complementary hypotheses which are called null hypothesis and alternative hypothesis. They are often denoted by H_0 and H_A. Let θ denote a parameter to be tested, the general form of the null and alternative hypothesis is $H_0 : \theta \in \Theta_0$ and $H_A : \theta \in \Theta_0^c$, where Θ_0 is a subset of the parameter space and Θ_0^c is its complement (Casella and Berger, 2002).

When deciding to reject the H_0, we commit one or both of two types of errors, which are named type I error and type II error. If H_0 is true but the test incorrectly rejects H_0, then the type I error occurs. Hence, the type I error is known as "false positive". On the contrary, if H_A is true but the test fails to reject H_0, the type II error occurs. Thus, the type II error is known as "false negative". In a hypothesis testing problem, the ideal situation is to make the probabilities of two types of errors, $P(\text{Reject } H_0 \mid H_0 \text{ is true})$ and $P(\text{Accept } H_0 \mid H_A \text{ is true})$, as small as possible. However, it is often impossible to make them both arbitrarily small. The smaller the one, the greater the other. Therefore, we seek a reasonable tradeoff between type I error probability and type II error probability. To choose a good balance, we commonly search for a test that has $\beta = P(\text{Accept } H_0 \mid H_A \text{ is true})$ as small as possible, or equivalently the one with $1 - \beta = 1 - P(\text{Accept } H_0 \mid H_A$ is true) as closer to 1 as possible when controlling $P(\text{Reject } H_0 \mid H_0 \text{ is true})$ at a specified level α. The α is called the size of a test and $1 - \beta$ is called the power of a test. The greater the power (when controlling the size at α), the better the test (Casella and Berger, 2002).

In this section, the empirical size and power of our methods are compared for sample sizes $n = 25, 50, 100, 200$ and for the number of AEs $K = 2, 3, 4$. The empirical size is calculated by dividing the number of data sets for which H_0 is rejected when H_0 is true by the total number of simulated data sets. The empirical power is calculated by dividing the number of data sets for which H_0 is rejected when H_A is true by the total number of simulated data sets. In our simulation, a total of 5000 simulated data sets are used to calculate the empirical size and power.

When the sample size is large enough, the empirical size is normally distributed with mean $= \alpha$ and $\sigma^2 = \alpha(1 - \alpha)/5000$. Therefore, the empirical size has the 95% confidence intervals $(0.007, 0.013)$, $(0.044, 0.056)$ and $(0.092, 0.108)$ for $\alpha = 0.01$, 0.05 and 0.1, respectively. Empirical Size and power of W, W_0 and W_1 under various sample sizes for the fixed nominal size when number of AEs $K = 2, 3, 4$ and number of AE's severity levels $C = 4$ are summarized in Table 4.1, Table 4.2 and Table 4.3. Empirical Size and power of W_{ord}, W_{ord0} and W_{ord1} with scores $(1, 2, 3, 4)$ under the same settings are summarized in Table 4.4, Table 4.5 and Table 4.6.

In practice, higher levels of severity of an AE may be observed less frequenly than lower levels and our interest is in the performance of the tests on low incidence events. Therefore, high probabilities are assigned to the lower levels of severity, while low probabilities are assigned to the higher levels in our simulation design. More specifically, when simulating the data sets under SMH, for $K = 2$, $(0.5, 0.25, 0.24, 0.01)$ is used as the marginal probabilities of AE $= 1$ and $(0.7, 0.2, 0.05, 0.05)$ is used as the marginal probabilities of AE $= 2$ under both treatments. For $K = 3$, $(1/3, 1/3, 0.25, 1/12)$ is used as the marginal probabilities of AE $= 1$, $(0.45, 1/3, 1/6, 0.05)$ is used as the marginal probabilities of the AE $= 2$, and $(0.45, 0.25, 0.25, 0.05)$ is used as the marginal probabilities of the AE $= 3$ under both treatments. For $K = 4$, they are $(0.5, 0.25, 0.24, 0.01)$, $(0.7, 0.2, 0.05, 0.05)$, $(0.4, 0.25, 0.25, 0.1)$ and $(0.7, 0.15, 0.11, 0.04)$ as the marginal probabilities of the AE $= 1, 2, 3, 4$, respectively, under both treatments.

Similar to the empirical size, high probabilities are assigned to the lower levels and low probabilities are assigned to the higher levels when generating the data sets for calculating the empirical power. More specifically, for $K = 2$, $(0.5, 0.25, 0.24, 0.01)$ and $(0.7, 0.2, 0.05, 0.05)$ are used as the marginal probabilities of each AE under the two treatments. For $K = 3$, $(1/3, 1/3, 0.25, 1/12)$ and $(0.45, 1/3, 1/6, 0.05)$ are used as the marginal probabilities of the AE $= 1$ under the two treatments, $(0.5, 0.4, 0.09, 0.01)$ and $(0.3, 0.4, 0.2, 0.1)$ are used as the marginal probabilities of the AE $= 2$ under the two treatments, $(0.3, 0.4, 0.2, 0.1)$ and $(0.5, 0.25, 0.2, 0.05)$ are used as the marginal probabilities of the AE $= 3$ under the two treatments. For $K = 4$, $(0.5, 0.25, 0.24, 0.01)$ and $(0.7, 0.2, 0.05, 0.05)$ are used as the marginal probabilities of AE $= 1$ under the two treatments, $(0.4, 0.25, 0.25, 0.1)$ and $(0.7, 0.15, 0.11, 0.04)$ are used as the marginal probabilities of the AE $= 2$ under the two treatments,

$(0.5, 0.25, 0.24, 0.01)$ and $(0.4, 0.25, 0.25, 0.1)$ are used as the marginal probabilities of the AE $= 3$ under the two treatments, and $(0.7, 0.2, 0.05, 0.05)$ and $(0.7, 0.15, 0.11, 0.04)$ are used as the marginal probabilities of the AE $= 4$ under the two treatments.

4.3.3 Results on Tests - W, W_0 and W_1

Among the three tests treating the severity level as nominal variable, the W and W_1 are too liberal and the empirical size improves as the sample size increases. On the contrary, the W_0 is too conservative and the empirical size improves as the sample size increases. Under the nominal levels of $0.01, 0.05$ and 0.1, none of the W, W_0 and W_1 seem to perform reasonably when $K = 3, 4$. However, when $K = 2$, the W_0 can be used when $n = 100$, and W_0 and W_1 can be used when $n = 200$ under the nominal levels of 0.01 and 0.05.

The power of the three tests can be compared only when the empirical size is well controlled ($n \geq 200$). Unless the sample size is larger than 200, and so the empirical size is controlled at the nominal level, we shall not comment on these tests, as they fail to perform.

In summary, the W_0 has better performance in maintaining the nominal level than the W and W_1, but all three require much large samples to be suitable to use in practice.

Table 4.1: Empirical size and power of the W, W_0 and W_1 in 5000 simulated data sets under the nominal level 0.01

		Empirical Size (Power)			
K	Method	n=25	n=50	n=100	n=200
2	W	**0.071(0.644)**	**0.035(0.906)**	**0.022(0.998)**	0.013(1)
	W_0	**0(0.029)**	**0.004(0.65)**	0.007(0.996)	0.007(1)
	W_1	**0.064(0.616)**	**0.034(0.896)**	**0.021(0.998)**	0.013(1)
3	W	**0.205(0.621)**	**0.065(0.751)**	**0.028(0.976)**	**0.016(1)**
	W_0	**0(0)**	**0.003(0.266)**	**0.004(0.915)**	0.007(1)
	W_1	**0.185(0.591)**	**0.061(0.738)**	**0.026(0.976)**	**0.016(1)**
4	W	**0.39(0.874)**	**0.115(0.945)**	**0.039(0.999)**	**0.019(1)**
	W_0	**0(0)**	**0.002(0.382)**	**0.005(0.99)**	**0.006(1)**
	W_1	**0.359(0.855)**	**0.107(0.939)**	**0.036(0.999)**	**0.018(1)**

Note: The bold text indicates that the empirical size falls outside the 95% confidence interval $(0.007, 0.013)$ of the nominal level 0.01.

When the simulated data set is very sparse, the three tests may fail to work. These data sets are eliminated from the summary analysis.

Table 4.2: Empirical size of the W, W_0 and W_1 in 5000 simulated data sets under the nominal level 0.05

		Empirical Size(Power)			
K	Method	n=25	n=50	n=100	n=200
2	W	**0.168(0.823)**	**0.112(0.97)**	**0.077(1)**	**0.058(1)**
	W_0	**0.015(0.327)**	**0.035(0.905)**	0.047(0.998)	**0.042(1)**
	W_1	**0.147(0.804)**	**0.106(0.968)**	**0.074(1)**	0.056(1)
3	W	**0.348(0.77)**	**0.168(0.884)**	**0.088(0.994)**	**0.067(1)**
	W_0	**0.009(0.089)**	**0.03(0.626)**	**0.038(0.983)**	**0.042(1)**
	W_1	**0.325(0.753)**	**0.157(0.874)**	**0.084(0.994)**	**0.066(1)**
4	W	**0.553(0.945)**	**0.24(0.986)**	**0.12(1)**	**0.077(1)**
	W_0	**0.003(0.038)**	**0.02(0.787)**	**0.034(0.999)**	**0.037(1)**
	W_1	**0.514(0.936)**	**0.22(0.984)**	**0.114(1)**	**0.074(1)**

Note: The bold text indicates that the empirical size falls outside the 95% confidence interval $(0.044, 0.056)$ of the nominal level 0.05 .

When the simulated data set is very sparse, the three tests may fail to work. These data sets are eliminated from the summary analysis.

Table 4.3: Empirical size and power of the W, W_0 and W_1 in 5000 simulated data sets under the nominal level 0.1

		Empirical Size(Power)			
K	Method	n=25	n=50	n=100	n=200
2	W	**0.252(0.889)**	**0.179(0.985)**	**0.129(1)**	0.108(1)
	W_0	**0.049(0.573)**	**0.085(0.959)**	0.093(1)	**0.088(1)**
	W_1	**0.228(0.874)**	**0.169(0.984)**	**0.127(1)**	0.107(1)
3	W	**0.446(0.842)**	**0.251(0.93)**	**0.163(0.998)**	**0.127(1)**
	W_0	**0.046(0.265)**	**0.078(0.778)**	**0.081(0.994)**	**0.09(1)**
	W_1	**0.415(0.825)**	**0.239(0.926)**	**0.154(0.998)**	**0.124(1)**
4	W	**0.647(0.969)**	**0.336(0.993)**	**0.191(1)**	**0.143(1)**
	W_0	**0.019(0.212)**	**0.067(0.906)**	**0.084(1)**	**0.09(1)**
	W_1	**0.615(0.961)**	**0.315(0.993)**	**0.185(1)**	**0.14(1)**

Note: The bold text indicates that the empirical size falls outside the 95% confidence interval $(0.092, 0.108)$ of the nominal level 0.1 .

When the simulated data set is very sparse, the three tests may fail to work. These data sets are eliminated from the summary analysis.

4.3.4 Results on Ordinal Tests - W_{ord}, W_{ord0} and W_{ord1}

For these three ordinal tests with scores $(1, 2, 3, 4)$, under the nominal level 0.01, the W_{ord} and W_{ord1} started to show stability when $n \geq 100$. The W_{ord0} also maintain the nominal level satisfactorily when $n \geq 50$. Under the nominal level 0.05 and 0.1, the W_{ord} and W_{ord1} can maintain the nominal level when $n \geq 200$ while the W_{ord0} can maintain the nominal level effectively when $n \geq 50$. W_{ord0} can maintain the nominal level well even when $n = 25$ and $K = 2$. Regarding the power, the W_{ord}, W_{ord0} and W_{ord1} have comparable power when their empirical sizes are well controlled ($n \geq 200$). In summary, when $n < 200$, the W_{ord} and W_{ord1} are both too liberal to be recommended. when $25 < n < 200$. the W_{ord0} can better maintain the nominal level than W_{ord} and W_{ord1} and therefore recommended to use.

The above results show that the tests treating the severity levels as ordinal generally perform better than the tests treating them as nominal variables with respect to the empirical size. When the nominal level is well controlled by increasing the sample size, their power can be comparable. Therefore, the tests which ignore the ordinal nature are not recommended when sample size $n < 200$ due to their poor performance in maintaining the nominal level.

Table 4.4: Empirical size and power of the W_{ord}, W_{ord0} and W_{ord1} with scores $(1, 2, 3, 4)$ in 5000 simulated data sets under the nominal level 0.01

		Empirical Size(Power)			
K	Method	n=25	n=50	n=100	n=200
2	W_{ord}	**0.027(0.292)**	**0.017(0.495)**	**0.016(0.816)**	0.012(0.99)
	W_{ord0}	**0.004(0.123)**	0.008(0.384)	0.012(0.774)	0.01(0.988)
	W_{ord1}	**0.022(0.275)**	**0.016(0.488)**	**0.015(0.811)**	0.012(0.99)
3	W_{ord}	**0.039(0.436)**	**0.019(0.747)**	**0.016(0.982)**	**0.014(1)**
	W_{ord0}	**0.005(0.13)**	**0.006(0.594)**	0.01(0.968)	0.01(1)
	W_{ord1}	**0.034(0.41)**	**0.018(0.736)**	**0.015(0.982)**	**0.014(1)**
4	W_{ord}	**0.049(0.502)**	**0.026(0.770)**	**0.015(0.982)**	0.012(1)
	W_{ord0}	**0.002(0.104)**	**0.006(0.572)**	0.007(0.966)	0.007(1)
	W_{ord1}	**0.041(0.473)**	**0.024(0.761)**	**0.014(0.981)**	0.012(1)

Note: The bold text indicates that the empirical size falls outside the 95% confidence interval $(0.007, 0.013)$ of the nominal level 0.01 .

Table 4.5: Empirical size and power of the W_{ord}, W_{ord0} and W_{ord1} with scores $(1, 2, 3, 4)$ in 5000 simulated data sets under the nominal level 0.05

		Empirical Size(Power)			
K	Method	n=25	n=50	n=100	n=200
2	W_{ord}	**0.093(0.467)**	**0.066(0.7)**	**0.066(0.934)**	0.055(0.999)
	W_{ord0}	0.046(0.35)	0.046(0.64)	0.056(0.92)	0.05(0.999)
	W_{ord1}	**0.084(0.45)**	**0.062(0.691)**	**0.065(0.932)**	0.055(0.999)
3	W_{ord}	**0.1(0.646)**	**0.073(0.89)**	**0.063(1)**	0.049(1)
	W_{ord0}	**0.039(0.434)**	0.044(0.835)	0.051(0.996)	**0.042(1)**
	W_{ord1}	**0.092(0.624)**	**0.07(0.883)**	**0.062(0.996)**	0.048(1)
4	W_{ord}	**0.127(0.681)**	**0.09(0.897)**	**0.068(0.997)**	**0.063(1)**
	W_{ord0}	**0.031(0.418)**	0.044(0.828)	0.049(0.95)	0.054(1)
	W_{ord1}	**0.113(0.66)**	**0.085(0.893)**	**0.064(0.997)**	**0.063(1)**

Note: The bold text indicates that the empirical size falls outside the 95% confidence interval $(0.044, 0.056)$ of the nominal level 0.05.

Table 4.6: Empirical size and power of the W_{ord}, W_{ord0} and W_{ord1} with scores $(1, 2, 3, 4)$ in 5000 simulated data sets under the nominal level 0.1

		Empirical Size(Power)			
K	Method	n=25	n=50	n=100	n=200
2	W_{ord}	**0.143(0.574)**	**0.124(0.798)**	**0.119(0.964)**	0.102(1)
	W_{ord0}	0.102(0.489)	0.1(0.769)	0.105(0.96)	0.096(1)
	W_{ord1}	**0.133(0.56)**	**0.12(0.793)**	**0.115(0.963)**	0.1(1)
3	W_{ord}	**0.174(0.746)**	**0.135(0.939)**	**0.117(0.999)**	0.105(1)
	W_{ord0}	**0.086(0.613)**	0.097(0.915)	0.096(0.998)	0.096(1)
	W_{ord1}	**0.162(0.731)**	**0.13(0.936)**	**0.113(0.999)**	0.104(1)
4	W_{ord}	**0.195(0.775)**	**0.148(0.939)**	**0.121(0.998)**	**0.113(1)**
	W_{ord0}	**0.08(0.595)**	0.098(0.903)	0.099(0.998)	0.1(1)
	W_{ord1}	**0.182(0.758)**	**0.14(0.936)**	**0.118(0.998)**	**0.11(1)**

Note: The bold text indicates that the empirical size falls outside the 95% confidence interval $(0.092, 0.108)$ of the nominal level 0.1 .

4.3.5 Power Comparison in Testing SMH against Stochastic Ordering

For the case of one adverse event, Agresti (1983) showed that the test using ordinal scales outperforms the test ignoring its ordinal nature when they are used to test SMH against stochastic ordering. In this section, a simulation is performed to verify if the the test using ordinal scales is also more powerful to test SMH against stochastic ordering for the multiple AEs case.

Based on the simulation results in section 4.3.3 and 4.3.4, it appears that the W_0 and W_{ord0} performs the best. Thus, the W_0 and W_{ord0} are contrasted in this section.

To simulate a dataset with the margins stochastically ordered, we randomly sampled from an underlying multivariate normal distribution having mean 0 and within-AE correlation $\rho_1 = 0.6$ and between-AE correlation $\rho_2 = 0.2$. A half of the dimensions of the multivarite random vector are dichotomized as the severity levels $c = 1, 2, 3, 4$ of all AEs under treatment 1 and the other half are dichotomized as the severity levels of all AEs under treatment 2. The boundries for AE categories under treatment 1 are set as -0.6, 0 and 0.6. The boundries for AE categories under treatment 2 are obtained by placing a shift $\Delta = 0.2$ relative to the boundaries for treatment 1. Hence, the boundaries of the AE categories under treatment 2 are -0.4, 0.2 and 0.8. The categorizations produce the marginal probabilities of AE categories under treatment 1 to be $(0.2743, 0.2257, 0.2257, 0.2743)$, and the marginal probabilities of AE categories under treatment 2 to be $(0.3446, 0.2347, 0.2089, 0.2119)$. Our simulation includes the settings representing the combinations of sample size $n = 100, 200$ and number of AEs $K = 2, 3, 4$.

Table 4.7 shows the power of the W_0 and W_{ord0} under the nominal levels of $0.01, 0.05$ and 0.1. From table 4.7, we have the following observations:

1. The power of the W_{ord0} is consistently greater than that of W_0 for all the combinations of K, n and α we considered.

2. The ratio (1$-$power of the W_0)/(1$-$power of the W_{ord0}) is consistently greater at $n = 200$ than $n = 100$ for all the combinations of K and α.

3. As K increases, the ratio (1$-$power of the W_0)/(1$-$power of the W_{ord0}) consistently increases for all combinations of n and α.

Therefore, the W_{ord0} outperforms the W_0 with respect to the power. Furthermore, as the sample size or number of AEs increases, the advantage of the

W_{ord0} compared to the W_0 increases.

Table 4.7: Empirical power of the W_{ord0} and W_0 in 5000 simulated data sets under the nominal levels of $0.01, 0.05$ and 0.1

		Empirical Power					
		$\alpha = 0.01$		$\alpha = 0.05$		$\alpha = 0.1$	
K	Method	n=100	n=200	n=100	n=200	n=100	n=200
2	W_0	0.303	0.649	0.532	0.828	0.647	0.901
	W_{ord0}	0.434	0.808	0.676	0.933	0.789	0.966
3	W_0	0.461	0.836	0.675	0.934	0.775	0.97
	W_{ord0}	0.594	0.939	0.799	0.984	0.874	0.992
4	W_0	0.287	0.84	0.578	0.955	0.717	0.978
	W_{ord0}	0.625	0.969	0.847	0.993	0.915	0.997

Note: The marginal probabilities of AE categories are $(0.2743, 0.2257, 0.2257, 0.2743)$ and $(0.3446, 0.2347, 0.2089, 0.2119)$ for treatment 1 and treatment 2, respectively.

Chapter 5

Data Analysis

In this chapter, a data set (FOB; Moser, 1989) from a study to evaluate the severity of neurotoxic effects in animals after receiving exposure to a perchlorethylene (PERC) is used. In this study, 40 animals were randomly assigned to placebo or four dose levels of PERC, with eight animals in each group. The study consists of 25 endpoints, which were all transformed to a scale of 1 to 4, where 1 indicates the absence of the adverse effect and 4 denotes the most severe adverse effect. The 25 endpoints were classified into six domains. The endpoints in the same domain are correlated with each other. Out of 5 groups, the placebo group and 1.5g/kg group are used in this illustration. Our main interest is to test if the marginal distributions of each AE in the same domain are homogeneous (SMH) under two treatments against stochastic ordering. Although the two treatment groups in FOB data are independent, our methods may apply, and the purpose is to demonstrate the practical use of our methods. Table 5.1 shows the P-values from our methods.

Table 5.1: P-values for domains, comparing the 1.5g/kg group to placebo group, from W, W_0, W_1 and W_{ord}, W_{ord0}, W_{ord1} with scores $(1, 2, 3, 4)$

		Nominal tests			Ordinal tests		
Domain	K	W	W_0	W_1	W_{ord}	W_{ord0}	W_{ord1}
Autonomic	5	NA	NA	NA	NA	NA	NA
Sensorimotor	4	NA	NA	NA	0.022	0.318	0.04
CNS excitabilityc	5	NA	NA	NA	NA	NA	NA
CNS activityc	3	NA	NA	NA	NA	NA	NA
Neuromuscular	5	NA	NA	NA	1.21E-8	0.236	1.71E-7
Physiological	3	NA	NA	NA	NA	NA	NA

Note: K refers to number of endpoints in the domain.

NA indicates that method failed to work due to the sparsity of the data.

From Table 5.1, only the W_{ord}, W_{ord0}, W_{ord1} provide the P-value for domains of Sensorimotor and Neuromuscular. The W, W_0, W_1 fail to work for all the domains. It is because data($n = 8$) is too sparse, which makes the covariance matrix non-invertible which is used to construct the statistic. However, due to that the W_{ord}, W_{ord0}, W_{ord1} can not maintain the nominal level with small sample size $n = 8$ as demonstrated in chapter 4, the test results may be invalid. At best, we can conclude that further investigation is necessary for the two domains before conclusive decisions can be made.

Chapter 6

Conclusion

This thesis investigated the methods to test SMH in clustered matched-pair multinomial data. A Wald test, a score-type test and non-parametric test are proposed. A simulation study is performed to evaluate their performance with respect to the empirical size and power. In addition, another three statistics are proposed when the outcomes are ordinal. A simulation is conducted on the empirical size and power of these three statistics. Furthermore, a simulation is performed to compare the power of the W_0 and W_{ord0} when the outcomes are ordinal and the interest is to test SMH against stochastic ordering.

Based on the discussion in Chapter 3 and 4, we have the following conclusions and recommendations:

1. For clustered matched-pair multinomial data, the W and W_1 are liberal when the number of outcomes $K = 2$ and the sample size is smaller than 200. They can be used when sample size is at least 200. Moreover, When the the number of outcomes increases, it requires larger sample size to be used. The W_0 is conservative when the number of outcomes $K = 2$ and samle size is less than 100 and therefore suggested to use

when sample size is larger than 100. When $K > 2$, W_0 requireds larger sample size to be used.

2. For clustered matched-pair ordinal data, the W_{ord} and W_{ord1} are liberal when sample size ≤ 100. They can be used when sample size ≥ 200. In contrast, the W_{ord0} maintains the nominal level very well when sample size is 50. Especially when $K = 2$, W_{ord0} can be used when sample size is 25. Therefore, the W_{ord0} is suggested to use in practice.

3. For clustered matched-pair ordinal data, when the association is truly stochastically ordered, the W_{ord0} is more powerful than W_0.

Although the W_0 maintain nominal size well when $K = 2$ and sample size $= 100$, in some situations, W_0 fail to work as the covariance matrix is non-invertible. When the sample size is small, this becomes even more serious. Similarly, it also happens to W_{ord0} when sample size is less than 25. For instance, we presented the FOB data in chapter 5, a sample size of 8. Such small data set is a reality in practice. However, we found that the FOB data make the W_0 fail to work for all domains and the W_{ord0} fail to work in 4 out of 6 domains. Therefore, further research on methods to deal with sparse data is needed.

Bibliography

[1] Agresti, A. (1983). Testing Marginal Homogeneity for Ordinal Categorical Variables. *Biometrics*, 39, 505-510.

[2] Agresti, A. (2002). *Categorical Data Analysis.* John Wiley & Sons, Inc., Second edition.

[3] Agresti, A., Klingenberg, B. (2005). Multivariate Tests Comparing Binomial Probabilities, with Application to Safety Studies for Drugs. *Journal of the Royal Statistical Society: Series C(Applied Statistics)*, 54, 691-706.

[4] Agresti, A. (2007). *An Introduction to Categorical Data Analysis.* John Wiley & Sons, Inc., Second edition.

[5] Agresti, A. (2010). *Analysis of Ordinal Categorical Data.* John Wiley & Sons, Inc., Second edition.

[6] Bhapkar, V. (1966). A Note on the Equivalence of Two Test Criteria for Hypotheses in Categorical Data. *Journal of American Statistical Association*, 61, 228-235.

[7] Casella, G., Berger R. (2002). *Statistical Inference.* Duxbury, Second edition.

[8] Chuang-Stein, C., Mohberg, N. (1993). A Unified Approach to the Analysis of Safety Data in Clinical Trials. *Drug Safety Assessment in Clinical Trials* (ed. Gilbert, G.).

[9] Deming, W., Stephan, F. (1940). On a Least Squares Adjustment of a Sampled Frequency Table When the Expected Marginal Totals are Known. *The Annals of Mathematical Statistics*, 11(4), 427-444.

[10] Donner, A. (1985). The Analysis of Intraclass Correlation in Multiple Samples. *Annals of Human Genetics*, 49, 75-82.

[11] Durkalski, V., Palesch, Y., Lipsitz, S., Rust, P. (2003). Analysis of Clustered Matched-pair Data. *Statistics in Medicine*, 22, 2417-2428.

[12] Eliasziw, M., Donner, A. (1991). Application of the McNemar Test to Non-independent Matched Pair Data. *Statistics in Medicine*, 10, 1981-1991.

[13] Gao, W., Kuriki, S. (2006). Testing Marginal Homogeneity against Stochastically Ordered Marginals for $r \times r$ Contingency Tables. *Journal of Multivariate Analysis*, 97, 1330-1341.

[14] Ireland, T., Ku, H., Kullback, S. (1969). Symmetry and Marginal Homogeneity of an $r \times r$ Contingency Table. *Journal of American Statistical Association*, 64(15), 1323-1341.

[15] Jin, H., Lu, Y. (2009). Comparison of Correlated Proportions Based on Paired Binary Data from Clustered Samples. *Journal of Statistical Planning and Inference*, 139, 4206-4212.

[16] Klingenberg, B., Agresti, A. (2006). Multivariate Extensions of McNemar's Test. *Biometrics*, 62, 921-928.

[17] Klingenberg, B., Solari, A., Salmaso, L., Pesarin, F. (2008). Testing Marginal Homogeneity Against Stochastic Order in Multivariate Ordinal Data. *Biometrics*, 65, 452-462.

[18] Lang, B., Agresti, A. (1994). Simultaneously Modeling Joint and Marginal Distributions of Multivariate Categorical Responses. *Journal of American Statistical Association*, 89(426), 625-632.

[19] Mantel, N. (1963). Chi-Square Tests with One Degree of Freedom; Extensions of the Mantel-Haenszel Procedure. *Journal of American Statistical Association*, 58(303), 690-700.

[20] Maxwell A. (1970). Comparing the Classification of Subjects by Two Independent Judges. *British Journal of Psychiatry*, 116, 651-655.

[21] McNemar, Q. (1947). Note on the Sampling Error of the Difference between Correlated Proportions or Percentages. *Psychometrika*, 12, 153-157.

[22] Obuchowski, N. (1998). On the Comparison of Correlated Proportions for Clustered Data. *Statistics in Medicine*, 17, 1495-1507.

[23] Rao, J., Scott, A. (1992). A Simple Method for the Analysis of Clustered Binary Data. *Biometrics*, 48, 577-585.

[24] Stuart, A. (1955). A Test for Homogeneity of the Marginal Distribution of a Two-way Classification. *Biometrika*, 42, 412-416.

[25] Yang, Z., Sun, X., Hardin, J. (2010). A Note on the Tests for Clustered Matched-Pair Binary Data. *Biometrical Journal*, 52(5), 638-652.

[26] Yang, Z., Sun, X., Hardin, J. (2011). Testing Marginal Homogeneity in Clustered Matched-pair Data. *Journal of Statistical Planning and Inference*, 141, 1313-1318.

Appendix A

R code

Functions of W, W_0 and W_1

```
#----------------------------------------------------------------
#Parameters:
#nout:   number of outcomes
#nlevel: number of levels of the outcomes
#data:   an array of 2*nout dimensions with nlevel units
#        for each dimension
#type:   the method used to testing the hypothesis. The
#        default is score-type test 'score'. It also
#        has other two options: Wald test 'wald' and
#        Non-parametric test 'np'.
#----------------------------------------------------------------

smh_mult = function (nout, nlevel, data, type = "score") {
```

```
# number of pairs
   nsize = sum(data)

# marginal sample proportion
   pi1 = NULL
   pi2 = NULL
   for (h in 1:nout) {
      pi1.cell = apply(data,c(2*h-1),sum)[1:(nlevel-1)]/nsize
      pi2.cell = apply(data,c(2*h),sum)[1:(nlevel-1)]/nsize
      pi1 = c(pi1,pi1.cell)
      pi2 = c(pi2,pi2.cell)
   }
   pi.pool = (pi1+pi2)/2

#  difference of the marginal sample proportion at two doses
   d = pi2-pi1
   var = rep(1,length(d))
   cov = matrix(0,nrow = length(d),ncol = length(d))

# variance of the difference assuming multinomial distribution
   if (type == 'wald') {
      for (m in 1:length(d)) {
        h = ceiling(m/(nlevel-1))
        n = m %% (nlevel-1)
        if (n == 0) {
          n = nlevel-1
        }
      pi1h = apply(data,c(2*h-1),sum)[n]/nsize
      pi2h = apply(data,c(2*h),sum)[n]/nsize
      ph = diag(apply(data,c(2*h-1,2*h),sum))[n]/nsize
      var[m] = (pi1h+pi2h-2*ph-(pi1h-pi2h)**2)/nsize
   }

# Covariance of the difference
   for (i in 1:(length(d)-1)) {
     k1 = i %% (nlevel-1)
     if (k1 == 0) {
       k1 = nlevel-1
     }
     for (j in (i+1):length(d)) {
       k2 = j %% (nlevel-1)
       if (k2 == 0) {
         k2 = nlevel-1
       }
       h1 = ceiling(i/(nlevel-1))
       h2 = ceiling(j/(nlevel-1))
       pi1h1 = apply(data,c(2*h1-1),sum)[k1]/nsize
       pi1h2 = apply(data,c(2*h2-1),sum)[k2]/nsize
       pi2h1 = apply(data,c(2*h1),sum)[k1]/nsize
       pi2h2 = apply(data,c(2*h2),sum)[k2]/nsize
       pi12h2h1 = apply(data,c(2*h2-1,2*h1),sum)[k2,k1]/nsize
       pi12h1h2 = apply(data,c(2*h1-1,2*h2),sum)[k1,k2]/nsize
       if (h1 != h2) {
          pi1h1h2 = apply(data,c(2*h1-1,2*h2-1),sum)[k1,k2]/nsize
          pi2h1h2 = apply(data,c(2*h1,2*h2),sum)[k1,k2]/nsize
          cov[i,j] = (pi1h1h2-pi1h1*pi1h2+pi2h1h2-pi2h1*pi2h2
                      -pi12h2h1-pi12h1h2+pi1h2*pi2h1+
                     pi1h1*pi2h2)/nsize
       }
       else {
          cov[i,j] = (-pi1h1*pi1h2-pi2h1*pi2h2-pi12h2h1-
                     pi12h1h2+pi1h2*pi2h1+pi1h1*pi2h2)/nsize
       }
     }
    }
  }

# the non-parametric variance/covariance of the difference
  else if (type == 'np') {
#variance
    for (m in 1:length(d)) {
      h = ceiling(m/(nlevel-1))
      n = m %% (nlevel-1)
      if (n == 0) {
        n = nlevel-1
      }
      y1h_num = apply(data,c(2*h-1),sum)[n]
      y1h_bar = y1h_num/nsize
      y2h_num = apply(data,c(2*h),sum)[n]
      y2h_bar = y2h_num/nsize
      y12h = apply(data,c(2*h-1,2*h),sum)
      y12h11 = diag(y12h)[n]
      y12h10 = apply(data,c(2*h-1),sum)[n]-y12h11
      y12h01 = apply(data,c(2*h),sum)[n]-y12h11
      y12h00 = sum(y12h)-y12h11-y12h10-y12h01
      var_part1 = y1h_num*((1-y1h_bar)**2)+(nsize-y1h_num)*
                  (y1h_bar**2)
      var_part2 = y2h_num*((1-y2h_bar)**2)+(nsize-y2h_num)*
                  (y2h_bar**2)
      var_part3 = (1-y1h_bar)*(1-y2h_bar)*y12h11-(1-y1h_bar)*
                  y2h_bar*y12h10-(1-y2h_bar)*y1h_bar*y12h01+
                  y1h_bar*y2h_bar*y12h00
      var[m] = (var_part1+var_part2-2*var_part3)/(nsize*(nsize-1))
    }

# covariance
    for (i in 1:(length(d)-1)) {
      k1 = i %% (nlevel-1)
      if (k1 == 0) {
        k1 = nlevel-1
      }
    for (j in (i+1):length(d)) {
      k2 = j %% (nlevel-1)
      if (k2 == 0) {
        k2 = nlevel-1
      }
      h1 = ceiling(i/(nlevel-1))
      h2 = ceiling(j/(nlevel-1))

      y1h1k1_num = apply(data,c(2*h1-1),sum)[k1]
      y1h1k1_bar = y1h1k1_num/nsize
      y1h2k2_num = apply(data,c(2*h2-1),sum)[k2]
      y1h2k2_bar = y1h2k2_num/nsize

      y2h1k1_num = apply(data,c(2*h1),sum)[k1]
      y2h1k1_bar = y2h1k1_num/nsize
      y2h2k2_num = apply(data,c(2*h2),sum)[k2]
      y2h2k2_bar = y2h2k2_num/nsize

      if (h1 != h2) {
        y22h1h211 = apply(data,c(2*h1,2*h2),sum)[k1,k2]
        y22h1h210 = apply(data,c(2*h1),sum)[k1]-y22h1h211
        y22h1h201 = apply(data,c(2*h2),sum)[k2]-y22h1h211
        y22h1h200 = nsize-y22h1h211-y22h1h210-y22h1h201
        cov.part1 = (1-y2h2k2_bar)*(1-y2h1k1_bar)*y22h1h211-
```

```
            (1-y2h1k1_bar)*y2h2k2_bar*y22h1h210-
            (1-y2h2k2_bar)*y2h1k1_bar*y22h1h201+
            y2h2k2_bar*y2h1k1_bar*y22h1h200
}
else if (h1 == h2) {
  y22h1h210 = apply(data,c(2*h1),sum)[k1]
  y22h1h201 = apply(data,c(2*h1),sum)[k2]
  cov.part1 = -(1-y2h1k1_bar)*y2h2k2_bar*y22h1h210-
              (1-y2h2k2_bar)*y2h1k1_bar*y22h1h201+
              y2h1k1_bar*y2h2k2_bar*(nsize-y22h1h210-
              y22h1h201)
}

y21h1h211 = apply(data,c(2*h1,2*h2-1),sum)[k1,k2]
y21h1h210 = apply(data,c(2*h1),sum)[k1]-y21h1h211
y21h1h201 = apply(data,c(2*h2-1),sum)[k2]-y21h1h211
y21h1h200 = nsize-y21h1h211-y21h1h210-y21h1h201
cov.part2 = (1-y2h1k1_bar)*(1-y1h2k2_bar)*y21h1h211-
            (1-y2h1k1_bar)*y1h2k2_bar*y21h1h210-
            (1-y1h2k2_bar)*y2h1k1_bar*y21h1h201+
            y1h2k2_bar*y2h1k1_bar*y21h1h200

y12h1h211 = apply(data,c(2*h1-1,2*h2),sum)[k1,k2]
y12h1h210 = apply(data,c(2*h1-1),sum)[k1]-y12h1h211
y12h1h201 = apply(data,c(2*h2),sum)[k2]-y12h1h211
```

```
y12h1h200 = nsize-y12h1h211-y12h1h210-y12h1h201
cov.part3 = (1-y1h1k1_bar)*(1-y2h2k2_bar)*y12h1h211-
            (1-y1h1k1_bar)*y2h2k2_bar*y12h1h210-
            (1-y2h2k2_bar)*y1h1k1_bar*y12h1h201+
             y1h1k1_bar*y2h2k2_bar*y12h1h200

if (h1 !=h2 ) {
  y11h1h211 = apply(data,c(2*h1-1,2*h2-1),sum)[k1,k2]
  y11h1h210 = apply(data,c(2*h1-1),sum)[k1]-y11h1h211
  y11h1h201 = apply(data,c(2*h2-1),sum)[k2]-y11h1h211
  y11h1h200 = nsize-y11h1h211-y11h1h210-y11h1h201
  cov.part4 = (1-y1h1k1_bar)*(1-y1h2k2_bar)*y11h1h211-
              (1-y1h1k1_bar)*y1h2k2_bar*y11h1h210-
              (1-y1h2k2_bar)*y1h1k1_bar*y11h1h201+
              y1h1k1_bar*y1h2k2_bar*y11h1h200
  }
else if (h1 == h2) {
  y11h1h210 = apply(data,c(2*h1-1),sum)[k1]
  y11h1h201 = apply(data,c(2*h1-1),sum)[k2]
  cov.part4 = -(1-y1h1k1_bar)*y1h2k2_bar*y11h1h210-
              (1-y1h2k2_bar)*y1h1k1_bar*y11h1h201+
              y1h1k1_bar*y1h2k2_bar*(nsize-y11h1h210
              -y11h1h201)
  }
cov[i,j] = (cov.part1-cov.part2-cov.part3+cov.part4)
```

```
                   /(nsize*(nsize-1))
     }
     }
  }

  else if (type == 'score') {
# variance
    for (m in 1:length(d)) {
      h = ceiling(m/(nlevel-1))
      n = m %% (nlevel-1)
      if (n == 0) {
        n = nlevel-1
      }
      pi1h = apply(data,c(2*h-1),sum)[n]/nsize
      pi2h = apply(data,c(2*h),sum)[n]/nsize
      ph = diag(apply(data,c(2*h-1,2*h),sum))[n]/nsize
      var[m] = (pi1h+pi2h-2*ph)/nsize
    }
# covariance
    for (i in 1:(length(d)-1)) {
      k1 = i %% (nlevel-1)
      if (k1 == 0) {
        k1 = nlevel-1
      }
      for (j in (i+1):length(d)) {
```

```
     k2 = j %% (nlevel-1)
     if (k2 == 0) {
       k2 = nlevel-1
     }
     h1 = ceiling(i/(nlevel-1))
     h2 = ceiling(j/(nlevel-1))
     pi12h2h1 = apply(data,c(2*h2-1,2*h1),sum)[k2,k1]/nsize
     pi12h1h2 = apply(data,c(2*h1-1,2*h2),sum)[k1,k2]/nsize
     if (h1 != h2) {
        pi1h1h2 = apply(data,c(2*h1-1,2*h2-1),sum)[k1,k2]
                 /nsize
        pi2h1h2 = apply(data,c(2*h1,2*h2),sum)[k1,k2]/nsize
        cov[i,j] = (pi1h1h2-pi.pool[i]*pi.pool[j]+pi2h1h2-
                  pi.pool[i]*pi.pool[j]-pi12h2h1-
                  pi12h1h2+pi.pool[j]*pi.pool[i]*2)/nsize
     }
     else {
        cov[i,j] = (-pi12h2h1-pi12h1h2)/nsize
     }
   }
 }
}
cov = cov+t(cov)
diag(cov) = var
```

```
# statistic
    w = t(d) %*% solve(cov) %*% d
    p = pchisq(w,df = nout*(nlevel-1),lower.tail = F)
    result = data.frame(w,p)
    return(result)
}
```

Functions of W_{ord}, W_{ord0} and W_{ord1}

```
#--------------------------------------------------------------
#Parameters:
#nout:    number of outcomes
#nlevel: number of levels of the outcomes
#data:    an array of 2*nout dimensions with nlevel units
#         for each dimension
#score:  a matrix of scores assigned to the levels of the
#         outcomes.
#type:    the methods used to testing the hypothesis. The
#         default is score-type test 'score'. It also
#         has other two options: Wald test 'wald' and
#         Non-parametric test 'np'.
#--------------------------------------------------------------

smh_ord = function (nout, nlevel, data, score, type = 'score') {

# number of pairs
  nsize = sum(data)

# marginal sample proportion
  pi1 = NULL
  pi2 = NULL
  for (h in 1:nout) {
    pi1.cell = apply(data,c(2*h-1),sum)/nsize
    pi2.cell = apply(data,c(2*h),sum)/nsize
    pi1 = c(pi1,pi1.cell)
    pi2 = c(pi2,pi2.cell)
  }
  pi.pool = (pi1+pi2)/2

# difference of the marginal sample proportion at two doses
  d = pi2-pi1
  var = rep(1,length(d))
  cov = matrix(0,nrow = length(d),ncol = length(d))

# variance of the difference assuming multinomial distribution
  if (type == 'wald') {
    for (m in 1:length(d)) {
      h = ceiling(m/nlevel)
      n = m %% nlevel
      if (n == 0) {
         n = nlevel
      }
      pi1h = apply(data,c(2*h-1),sum)[n]/nsize
      pi2h = apply(data,c(2*h),sum)[n]/nsize
      ph = diag(apply(data,c(2*h-1,2*h),sum))[n]/nsize
      var[m] = (pi1h+pi2h-2*ph-(pi1h-pi2h)**2)/nsize
    }
```

```
# Covariance of the difference
    for (i in 1:(length(d)-1)) {
      k1 = i %% nlevel
      if (k1 == 0) {
        k1 = nlevel
      }
      for (j in (i+1):length(d)) {
        k2 = j %% nlevel
        if (k2 == 0) {
          k2 = nlevel
        }
        h1 = ceiling(i/nlevel)
        h2 = ceiling(j/nlevel)
        pi1h1 = apply(data,c(2*h1-1),sum)[k1]/nsize
        pi1h2 = apply(data,c(2*h2-1),sum)[k2]/nsize
        pi2h1 = apply(data,c(2*h1),sum)[k1]/nsize
        pi2h2 = apply(data,c(2*h2),sum)[k2]/nsize
        pi12h2h1 = apply(data,c(2*h2-1,2*h1),sum)[k2,k1]/nsize
        pi12h1h2 = apply(data,c(2*h1-1,2*h2),sum)[k1,k2]/nsize
        if (h1 != h2) {
          pi1h1h2 = apply(data,c(2*h1-1,2*h2-1),sum)[k1,k2]
                    /nsize
          pi2h1h2 = apply(data,c(2*h1,2*h2),sum)[k1,k2]/nsize
          cov[i,j] = (pi1h1h2-pi1h1*pi1h2+pi2h1h2-pi2h1*pi2h2-
                     pi12h2h1-pi12h1h2+pi1h2*pi2h1+
                      pi1h1*pi2h2)/nsize
          }
        else {
          cov[i,j] = (-pi1h1*pi1h2-pi2h1*pi2h2-pi12h2h1-
                      pi12h1h2+pi1h2*pi2h1+pi1h1*pi2h2)/nsize
        }
      }
    }
  }

# the robust variance/covariance of the difference
  else if (type == 'robust') {
#variance
    for (m in 1:length(d)) {
      h = ceiling(m/nlevel)
      n = m %% nlevel
      if (n == 0) {
        n = nlevel
      }
      y1h_num = apply(data,c(2*h-1),sum)[n]
      y1h_bar = y1h_num/nsize
      y2h_num = apply(data,c(2*h),sum)[n]
      y2h_bar = y2h_num/nsize
      y12h = apply(data,c(2*h-1,2*h),sum)
      y12h11 = diag(y12h)[n]
      y12h10 = apply(data,c(2*h-1),sum)[n]-y12h11
      y12h01 = apply(data,c(2*h),sum)[n]-y12h11
      y12h00 = sum(y12h)-y12h11-y12h10-y12h01
      var_part1 = y1h_num*((1-y1h_bar)**2)+(nsize-y1h_num)*
                  (y1h_bar**2)
      var_part2 = y2h_num*((1-y2h_bar)**2)+(nsize-y2h_num)*
                  (y2h_bar**2)
      var_part3 = (1-y1h_bar)*(1-y2h_bar)*y12h11-(1-y1h_bar)*
                  y2h_bar*y12h10-(1-y2h_bar)*y1h_bar*y12h01+
                  y1h_bar*y2h_bar*y12h00
      var[m] = (var_part1+var_part2-2*var_part3)/(nsize*(nsize-1))
    }

# covariance
    for (i in 1:(length(d)-1)) {
      k1 = i %% nlevel
      if (k1 == 0) {
        k1 = nlevel
      }
      for (j in (i+1):length(d)) {
        k2 = j %% nlevel
        if (k2 == 0) {
          k2 = nlevel
        }
        h1 = ceiling(i/nlevel)
```

```
h2 = ceiling(j/nlevel)

y1h1k1_num = apply(data,c(2*h1-1),sum)[k1]
y1h1k1_bar = y1h1k1_num/nsize
y1h2k2_num = apply(data,c(2*h2-1),sum)[k2]
y1h2k2_bar = y1h2k2_num/nsize

y2h1k1_num = apply(data,c(2*h1),sum)[k1]
y2h1k1_bar = y2h1k1_num/nsize
y2h2k2_num = apply(data,c(2*h2),sum)[k2]
y2h2k2_bar = y2h2k2_num/nsize

if (h1 != h2) {
  y22h1h211 = apply(data,c(2*h1,2*h2),sum)[k1,k2]
  y22h1h210 = apply(data,c(2*h1),sum)[k1]-y22h1h211
  y22h1h201 = apply(data,c(2*h2),sum)[k2]-y22h1h211
  y22h1h200 = nsize-y22h1h211-y22h1h210-y22h1h201
  cov.part1 = (1-y2h2k2_bar)*(1-y2h1k1_bar)*y22h1h211-
              (1-y2h1k1_bar)*y2h2k2_bar*y22h1h210-
              (1-y2h2k2_bar)*y2h1k1_bar*y22h1h201+
              y2h2k2_bar*y2h1k1_bar*y22h1h200
  }
  else if (h1 == h2) {
    y22h1h210 = apply(data,c(2*h1),sum)[k1]
    y22h1h201 = apply(data,c(2*h1),sum)[k2]
    cov.part1 = -(1-y2h1k1_bar)*y2h2k2_bar*y22h1h210-
                (1-y2h2k2_bar)*y2h1k1_bar*y22h1h201+
                y2h1k1_bar*y2h2k2_bar*
                (nsize-y22h1h210-y22h1h201)
  }
  y21h1h211 = apply(data,c(2*h1,2*h2-1),sum)[k1,k2]
  y21h1h210 = apply(data,c(2*h1),sum)[k1]-y21h1h211
  y21h1h201 = apply(data,c(2*h2-1),sum)[k2]-y21h1h211
  y21h1h200 = nsize-y21h1h211-y21h1h210-y21h1h201
  cov.part2 = (1-y2h1k1_bar)*(1-y1h2k2_bar)*y21h1h211-
              (1-y2h1k1_bar)*y1h2k2_bar*y21h1h210-
              (1-y1h2k2_bar)*y2h1k1_bar*y21h1h201+
               y1h2k2_bar*y2h1k1_bar*y21h1h200

  y12h1h211 = apply(data,c(2*h1-1,2*h2),sum)[k1,k2]
  y12h1h210 = apply(data,c(2*h1-1),sum)[k1]-y12h1h211
  y12h1h201 = apply(data,c(2*h2),sum)[k2]-y12h1h211
  y12h1h200 = nsize-y12h1h211-y12h1h210-y12h1h201
  cov.part3 = (1-y1h1k1_bar)*(1-y2h2k2_bar)*y12h1h211-
              (1-y1h1k1_bar)*y2h2k2_bar*y12h1h210-
              (1-y2h2k2_bar)*y1h1k1_bar*y12h1h201+
              y1h1k1_bar*y2h2k2_bar*y12h1h200

  if (h1 != h2) {
    y11h1h211 = apply(data,c(2*h1-1,2*h2-1),sum)[k1,k2]
```

```
            y11h1h210 = apply(data,c(2*h1-1),sum)[k1]-y11h1h211
            y11h1h201 = apply(data,c(2*h2-1),sum)[k2]-y11h1h211
            y11h1h200 = nsize-y11h1h211-y11h1h210-y11h1h201
            cov.part4 = (1-y1h1k1_bar)*(1-y1h2k2_bar)*y11h1h211-
                        (1-y1h1k1_bar)*y1h2k2_bar*y11h1h210-
                        (1-y1h2k2_bar)*y1h1k1_bar*y11h1h201+
                        y1h1k1_bar*y1h2k2_bar*y11h1h200
          }
          else if (h1 == h2) {
            y11h1h210 = apply(data,c(2*h1-1),sum)[k1]
            y11h1h201 = apply(data,c(2*h1-1),sum)[k2]
            cov.part4 = -(1-y1h1k1_bar)*y1h2k2_bar*y11h1h210-
                       (1-y1h2k2_bar)*y1h1k1_bar*y11h1h201+
                       y1h1k1_bar*y1h2k2_bar*
                       (nsize-y11h1h210-y11h1h201)
          }
          cov[i,j] = (cov.part1-cov.part2-cov.part3+cov.part4)
                     /(nsize*(nsize-1))
          }
        }
     }

  else if (type == 'score') {
# variance
    for (m in 1:length(d)) {
      h = ceiling(m/nlevel)
      n = m %% nlevel
      if (n == 0) {
        n = nlevel
      }
      pi1h = apply(data,c(2*h-1),sum)[n]/nsize
      pi2h = apply(data,c(2*h),sum)[n]/nsize
      ph = diag(apply(data,c(2*h-1,2*h),sum))[n]/nsize
      var[m] = (pi1h+pi2h-2*ph)/nsize
    }

# covariance
    for (i in 1:(length(d)-1)) {
      k1 = i %% nlevel
      if (k1 == 0) {
        k1 = nlevel
      }
      for (j in (i+1):length(d)) {
        k2 = j %% nlevel
        if (k2 == 0) {
          k2 = nlevel
        }
        h1 = ceiling(i/nlevel)
        h2 = ceiling(j/nlevel)
        pi12h2h1 = apply(data,c(2*h2-1,2*h1),sum)[k2,k1]/nsize
        pi12h1h2 = apply(data,c(2*h1-1,2*h2),sum)[k1,k2]/nsize
        if (h1 != h2) {
          pi1h1h2 = apply(data,c(2*h1-1,2*h2-1),sum)[k1,k2]
                    /nsize
          pi2h1h2 = apply(data,c(2*h1,2*h2),sum)[k1,k2]
                    /nsize
          cov[i,j] = (pi1h1h2-pi.pool[i]*pi.pool[j]+pi2h1h2-
                     pi.pool[i]*pi.pool[j]-pi12h2h1-pi12h1h2+
                     pi.pool[j]*pi.pool[i]*2)/nsize
        }
        else {
          cov[i,j] = (-pi12h2h1-pi12h1h2)/nsize
        }
      }
    }
  }

  cov = cov+t(cov)
  diag(cov) = var

# Mean score difference
  ms.dif = score %*% d

# covariance matrix of score*d = score*cov*t(score)
  w.cov = score %*% cov %*% t(score)
```

```
# statistic
  w = t(ms.dif) %*% solve(w.cov) %*% ms.dif
  p = pchisq(w,df = nout,lower.tail = F)
  result = data.frame(w,p)
  return(result)
}
```

Functions of iterative proportional fitting

```
#------------------------------------------------------------------
#Parameters:
#nout: number of outcomes
#nlevel: number of levels of outcomes
#margin: a matrix of selected margins for each dimension
#------------------------------------------------------------------

smh_ipf = function(nout, nlevel, margin) {
#a vector of  multinomial probability
    set.seed(1)
    p=runif(nlevel**(2*nout))
    p=p/sum(p)
    tablesize=rep(nlevel,2*nout)
    probsimu=array(p,tablesize)
    if(any(margin==0)){
      margin[margin==0] <- 0.001
    }

#transform each dimension to make the observed marginal
#probability equal the expected marginal probability
    iter=0              #number of iteration
    checksum=1          #criteria for stopping the interation
    factor=matrix(0,nrow=2*nout,ncol=nlevel)
    while ((checksum>0.001)& (iter<1000)) {
        for (j in 1:(2*nout)) {
            mar.obs = apply(probsimu,j,sum)
            factor[j,]=(margin[j, ])/mar.obs
            probsimu=sweep(probsimu, j, factor[j,], ”*”)
        }
        checksum = max(apply(abs(1-factor),1,sum))
        iter=iter+1
    }
    print(iter)
    prob=as.vector(probsimu)
}
```

Empirical size and power of W, W_0 and W_1

```
#------------------------------------------------------------
#Parameters:
#nsimu:  number of simulated datasets
#nout:   number of outcomes
#nsize:  sample size
#nlevel: number of levels of outcomes
#alpha:  significance level
#------------------------------------------------------------

source(”smh_ipf.R”)
source(”smh_mult.R”)

# empirical size comparison
mout_emp = function(nsimu,nout,nsize,nlevel,alpha) {
  set.seed(1)
  dat0 = rmultinom(nsimu,nsize,prob=prob.vec)
  tablesize = rep(nlevel,2*nout)
  np.p = rep(10000,nsimu)
  multi.p = rep(10000,nsimu)
  scoret.p = rep(10000,nsimu)

  for (i in 1:nsimu) {
      # skip the error in the loop
      tryCatch({
```

```
   isimu = dat0[,i]
   dat = array(isimu, tablesize)
   mult = smh_mult(nout=nout, nlevel=nlevel, data=dat,
          type='wald')
   multi.p[i] = mult$p

   np = smh_mult(nout=nout, nlevel=nlevel, data=dat,
        type='np')
   np.p[i] = np$p

   scoret = smh_mult(nout=nout, nlevel=nlevel, data=dat,
            type='score')
   scoret.p[i] = scoret$p
 }, error = function(e) {cat("ERROR␣:",
   conditionMessage(e), "\n")})
}

 # remove the cases in which the statistic fails to work
 multi.p = multi.p[multi.p!=10000]
 np.p = np.p[np.p!=10000]
 scoret.p = scoret.p[scoret.p!=10000]

 # empirical size
 multi.empsize = sum(multi.p<=alpha)/length(multi.p)
 np.empsize = sum(np.p<=alpha)/length(np.p)
```

```
    scoret.empsize = sum(scoret.p<=alpha)/length(scoret.p)

    cat('Empirical␣size␣of␣wald␣test=␣',multi.empsize)
    cat("\n")
    cat('␣Empirical␣size␣of␣non-parametric␣method=␣',
    np.empsize)
    cat("\n")
    cat('␣Empirical␣size␣of␣score-type␣method=␣',
    scoret.empsize)
    cat("\n")
    cat ("Number␣of␣plausible␣obs=␣", length(multi.p),
    length(np.p),length(scoret.p))
    cat("\n")
}

#Power comparison

mout_power = function(nout,nsimu,nsize,nlevel,alpha) {
  set.seed(123)
  dat0 = rmultinom(nsimu,nsize,prob=prob.vec)
  tablesize = rep(nlevel,2*nout)

  np.p = rep(10000,nsimu)
  multi.p = rep(10000,nsimu)
  scoret.p = rep(10000,nsimu)
```

```
for (i in 1:nsimu) {
# skip the error in the loop
tryCatch({
    isimu = dat0[,i]
    dat = array(isimu,tablesize)
    multi.p[i] = smh_mult(nout=nout,nlevel=nlevel,data=dat,
                type='wald')$p
    np.p[i] = smh_mult(nout=nout,nlevel=nlevel,data=dat,
                type='np')$p
    scoret.p[i] = smh_mult(nout=nout,nlevel=nlevel,data=dat,
                type='score')$p
    }, error = function(e) {cat("ERROR␣:",
    conditionMessage(e), "\n")})
 }

  # remove the cases in which the statistic fails to work
  multi.p = multi.p[multi.p!=10000]
  np.p = np.p[np.p!=10000]
  scoret.p = scoret.p[scoret.p!=10000]

  # empirical power
  multi.power = sum(multi.p<=alpha)/length(multi.p)
  np.power = sum(np.p<=alpha)/length(np.p)
  scoret.power = sum(scoret.p<=alpha)/length(scoret.p)
```

```
    cat('Power␣of␣wald␣method=␣',multi.power)
    cat("\n")
    cat('Power␣of␣non-parametric␣method=␣',np.power)
    cat("\n")
    cat('Power␣of␣score␣method=␣',scoret.power)
    cat("\n")
    cat ("Number␣of␣plausible␣obs=␣", length(multi.p),
        length(np.p), length(scoret.p))
    cat("\n")
}

##################################################
#############     2 outcomes           ###########
##################################################

# empirical size comparison
margin.vec1 = c(1/2,1/4,0.24,0.01)
margin.vec2 = c(0.7,0.2,0.05,0.05)

margin = rbind(margin.vec1,margin.vec1,margin.vec2,
margin.vec2)
prob.vec = smh.ipf3(nout=2,nlevel=4,margin=margin)
s = matrix(0,nrow=2,ncol=8)
s[1,1:4] = 1:4
```

```
s[2,5:8] = 1:4

mout_emp(nsimu=5000,nout=2,nsize=25,nlevel=4,alpha=0.01)
mout_emp(nsimu=5000,nout=2,nsize=50,nlevel=4,alpha=0.01)
mout_emp(nsimu=5000,nout=2,nsize=100,nlevel=4,alpha=0.01)
mout_emp(nsimu=5000,nout=2,nsize=200,nlevel=4,alpha=0.01)

mout_emp(nsimu=5000,nout=2,nsize=25,nlevel=4,alpha=0.05)
mout_emp(nsimu=5000,nout=2,nsize=50,nlevel=4,alpha=0.05)
mout_emp(nsimu=5000,nout=2,nsize=100,nlevel=4,alpha=0.05)
mout_emp(nsimu=5000,nout=2,nsize=200,nlevel=4,alpha=0.05)

mout_emp(nsimu=5000,nout=2,nsize=25,nlevel=4,alpha=0.1)
mout_emp(nsimu=5000,nout=2,nsize=50,nlevel=4,alpha=0.1)
mout_emp(nsimu=5000,nout=2,nsize=100,nlevel=4,alpha=0.1)
mout_emp(nsimu=5000,nout=2,nsize=200,nlevel=4,alpha=0.1)

#Empirical Power
margin.vec1 = c(1/2,1/4,0.24,0.01)
margin.vec2 = c(0.7,0.2,0.05,0.05)

margin = rbind(margin.vec1,margin.vec2,margin.vec1,
margin.vec2)
prob.vec = smh_ipf(nout=2,nlevel=4,margin=margin)
s = matrix(0,nrow=2,ncol=8)
```

```
s[1,1:4] = 1:4
s[2,5:8] = 1:4

mout_power(nsimu=5000,nout=2,nsize=25,nlevel=4,alpha=0.01)
mout_power(nsimu=5000,nout=2,nsize=50,nlevel=4,alpha=0.01)
mout_power(nsimu=5000,nout=2,nsize=100,nlevel=4,alpha=0.01)
mout_power(nsimu=5000,nout=2,nsize=200,nlevel=4,alpha=0.01)

mout_power(nsimu=5000,nout=2,nsize=25,nlevel=4,alpha=0.05)
mout_power(nsimu=5000,nout=2,nsize=50,nlevel=4,alpha=0.05)
mout_power(nsimu=5000,nout=2,nsize=100,nlevel=4,alpha=0.05)
mout_power(nsimu=5000,nout=2,nsize=200,nlevel=4,alpha=0.05)

mout_power(nsimu=5000,nout=2,nsize=25,nlevel=4,alpha=0.1)
mout_power(nsimu=5000,nout=2,nsize=50,nlevel=4,alpha=0.1)
mout_power(nsimu=5000,nout=2,nsize=100,nlevel=4,alpha=0.1)
mout_power(nsimu=5000,nout=2,nsize=200,nlevel=4,alpha=0.1)

##################################################
#############     3 outcomes           ###########
##################################################

margin.vec1 = c(1/3,1/3,1/4,1/12)
margin.vec2 = c(0.45,1/3,1/6,0.05)
margin.vec3 = c(0.45,1/4,1/4,0.05)
```

```
margin = rbind(margin.vec1,margin.vec1,margin.vec2,
margin.vec2,margin.vec3,margin.vec3)
prob.vec = smh_ipf(nout=3,nlevel=4,margin=margin)

mout_emp(nsimu=5000,nout=3,nsize=25,nlevel=4,alpha=0.01)
mout_emp(nsimu=5000,nout=3,nsize=50,nlevel=4,alpha=0.01)
mout_emp(nsimu=5000,nout=3,nsize=100,nlevel=4,alpha=0.01)
mout_emp(nsimu=5000,nout=3,nsize=200,nlevel=4,alpha=0.01)

mout_emp(nsimu=5000,nout=3,nsize=25,nlevel=4,alpha=0.05)
mout_emp(nsimu=5000,nout=3,nsize=50,nlevel=4,alpha=0.05)
mout_emp(nsimu=5000,nout=3,nsize=100,nlevel=4,alpha=0.05)
mout_emp(nsimu=5000,nout=3,nsize=200,nlevel=4,alpha=0.05)

mout_emp(nsimu=5000,nout=3,nsize=25,nlevel=4,alpha=0.1)
mout_emp(nsimu=5000,nout=3,nsize=50,nlevel=4,alpha=0.1)
mout_emp(nsimu=5000,nout=3,nsize=100,nlevel=4,alpha=0.1)
mout_emp(nsimu=5000,nout=3,nsize=200,nlevel=4,alpha=0.1)

#Empirical Power
margin.vec1 = c(1/3,1/3,1/4,1/12)
margin.vec2 = c(0.45,1/3,1/6,0.05)
margin.vec3 = c(0.5,0.4,0.09,0.01)
margin.vec4 = c(0.3,0.4,0.2,0.1)
margin.vec5 = c(0.3,0.4,0.2,0.1)
margin.vec6 = c(0.5,0.25,0.2,0.05)

margin = rbind(margin.vec1,margin.vec2,margin.vec3,
margin.vec4,margin.vec5,margin.vec6)
prob.vec = smh_ipf(nout=3,nlevel=4,margin=margin)

mout_power(nsimu=5000,nout=3,nsize=25,nlevel=4,alpha=0.01)
mout_power(nsimu=5000,nout=3,nsize=50,nlevel=4,alpha=0.01)
mout_power(nsimu=5000,nout=3,nsize=100,nlevel=4,alpha=0.01)
mout_power(nsimu=5000,nout=3,nsize=200,nlevel=4,alpha=0.01)

mout_power(nsimu=5000,nout=3,nsize=25,nlevel=4,alpha=0.05)
mout_power(nsimu=5000,nout=3,nsize=50,nlevel=4,alpha=0.05)
mout_power(nsimu=5000,nout=3,nsize=100,nlevel=4,alpha=0.05)
mout_power(nsimu=5000,nout=3,nsize=200,nlevel=4,alpha=0.05)

mout_power(nsimu=5000,nout=3,nsize=25,nlevel=4,alpha=0.1)
mout_power(nsimu=5000,nout=3,nsize=50,nlevel=4,alpha=0.1)
mout_power(nsimu=5000,nout=3,nsize=100,nlevel=4,alpha=0.1)
mout_power(nsimu=5000,nout=3,nsize=200,nlevel=4,alpha=0.1)

##################################################
############## 4 outcomes #######################
##################################################
```

```
# empirical size comparison
margin.vec1 = c(1/2,1/4,0.24,0.01)
margin.vec2 = c(0.7,0.2,0.05,0.05)
margin.vec3 = c(0.4,0.25,0.25,0.1)
margin.vec4 = c(0.7,0.15,0.11,0.04)

margin = rbind(margin.vec1,margin.vec1,margin.vec2,
margin.vec2,margin.vec3,margin.vec3,margin.vec4,
margin.vec4)
prob.vec = smh_ipf(nout=4,nlevel=4,margin=margin)
s = matrix(0,nrow=4,ncol=16)
s[1,1:4] = 1:4
s[2,5:8] = 1:4
s[3,9:12] = 1:4
s[4,13:16] = 1:4

mout_emp(nsimu=5000,nout=4,nsize=25,nlevel=4,alpha=0.01)
mout_emp(nsimu=5000,nout=4,nsize=50,nlevel=4,alpha=0.01)
mout_emp(nsimu=5000,nout=4,nsize=100,nlevel=4,alpha=0.01)
mout_emp(nsimu=5000,nout=4,nsize=200,nlevel=4,alpha=0.01)

mout_emp(nsimu=5000,nout=4,nsize=25,nlevel=4,alpha=0.05)
mout_emp(nsimu=5000,nout=4,nsize=50,nlevel=4,alpha=0.05)
mout_emp(nsimu=5000,nout=4,nsize=100,nlevel=4,alpha=0.05)
mout_emp(nsimu=5000,nout=4,nsize=200,nlevel=4,alpha=0.05)
mout_emp(nsimu=5000,nout=4,nsize=25,nlevel=4,alpha=0.1)
mout_emp(nsimu=5000,nout=4,nsize=50,nlevel=4,alpha=0.1)
mout_emp(nsimu=5000,nout=4,nsize=100,nlevel=4,alpha=0.1)
mout_emp(nsimu=5000,nout=4,nsize=200,nlevel=4,alpha=0.1)

#Empirical Power
margin.vec1 = c(1/2,1/4,0.24,0.01)
margin.vec2 = c(0.7,0.2,0.05,0.05)
margin.vec3 = c(0.4,0.25,0.25,0.1)
margin.vec4 = c(0.7,0.15,0.11,0.04)

margin = rbind(margin.vec1,margin.vec2,margin.vec3,
margin.vec4,margin.vec1,margin.vec3,margin.vec2,
margin.vec4)
prob.vec = smh_ipf(nout=4,nlevel=4,margin=margin)
s = matrix(0,nrow=4,ncol=16)
s[1,1:4] = 1:4
s[2,5:8] = 1:4
s[3,9:12] = 1:4
s[4,13:16] = 1:4

mout_power(nsimu=5000,nout=5,nsize=25,nlevel=4,alpha=0.01)
mout_power(nsimu=5000,nout=5,nsize=50,nlevel=4,alpha=0.01)
mout_power(nsimu=5000,nout=5,nsize=100,nlevel=4,alpha=0.01)
mout_power(nsimu=5000,nout=5,nsize=200,nlevel=4,alpha=0.01)

mout_power(nsimu=5000,nout=4,nsize=25,nlevel=4,alpha=0.05)
mout_power(nsimu=5000,nout=4,nsize=50,nlevel=4,alpha=0.05)
mout_power(nsimu=5000,nout=4,nsize=100,nlevel=4,alpha=0.05)
mout_power(nsimu=5000,nout=4,nsize=200,nlevel=4,alpha=0.05)

mout_power(nsimu=5000,nout=4,nsize=25,nlevel=4,alpha=0.1)
mout_power(nsimu=5000,nout=4,nsize=50,nlevel=4,alpha=0.1)
mout_power(nsimu=5000,nout=4,nsize=100,nlevel=4,alpha=0.1)
mout_power(nsimu=5000,nout=4,nsize=200,nlevel=4,alpha=0.1)
```

Empirical size and power of W_{ord}, W_{ord0} and W_{ord1}

```
#-----------------------------------------------------------------
#Parameters:
#nsimu:   number of simulated datasets
#nout:    number of outcomes
#nsize:   sample size
#nlevel: number of levels of the outcomes
#alpha:   significance level
#score:   a matrix of scores assigned to the levels of the
#         outcomes
#-----------------------------------------------------------------

source(”smh_ipf.R”)
source(”smh_ord.R”)

# empirical size
mout_emp = function(nsimu,nout,nsize,nlevel,alpha,score) {
  set.seed(1)
  tablesize = rep(nlevel,2*nout)
  np.nsig = 0
  multi.nsig = 0
  scoret.nsig = 0
```

```
for (i in 1:nsimu) {
    dat0 = rmultinom(1,nsize,prob=prob.vec)
    dat = array(dat0,tablesize)
    mult.p = smh_ord(nout=nout,nlevel=nlevel,score=score,
             data=dat,type='wald')$p
    np.p = smh_ord(nout=nout,nlevel=nlevel,score=score,
           data=dat,type='np')$p
    scoret.p = smh_ord(nout=nout,nlevel=nlevel,score=score,
               data=dat,type='score')$p

    if (np.p <= alpha) {
        np.nsig = np.nsig+1
    }
    if (multi.p <= alpha) {
        multi.nsig = multi.nsig+1
    }
    if (scoret.p <= alpha) {
        scoret.nsig = scoret.nsig+1
    }
 }

    multi.empsize = multi.nsig/nsimu
    np.empsize = np.nsig/nsimu
    scoret.empsize = scoret.nsig/nsimu
```

```
      cat('Empirical␣size␣of␣wald␣test=␣',multi.empsize)
      cat("\n")
      cat('Empirical␣size␣of␣non-parametric␣test=␣',np.empsize)
      cat("\n")
      cat('␣Empirical␣size␣of␣score-type␣test=␣',scoret.empsize)
      cat("\n")
}

# empirical power
mout_power = function(nout,nsimu,nsize,nlevel,alpha,score,cv) {
  set.seed(123)
  dat0 = rmultinom(nsimu,nsize,prob=prob.vec)
  tablesize = rep(nlevel,2*nout)
  np.nsig = 0
  multi.nsig = 0
  scoret.nsig = 0

  for (i in 1:nsimu) {
      isimu = dat0[,i]
      dat = array(isimu,tablesize)
      multi.p = smh_ord(nout=nout,nlevel=nlevel,score=score,
                data=dat,type='wald')$p
      robust.p = smh_ord(nout=nout,nlevel=nlevel,score=score,
                 data=dat,type='robust')$p
      scoret.p = smh_ord(nout=nout,nlevel=nlevel,score=score,
                 data=dat, type='score')$p
      if (np.p <= alpha) {
          np.nsig = np.nsig+1
      }
      if (multi.p <= alpha) {
          multi.nsig = multi.nsig+1
      }
      if (scoret.p <= alpha) {
          scoret.nsig = scoret.nsig+1
      }
  }
      multi.power = multi.nsig/nsimu
      np.power = np.nsig/nsimu
      scoret.power = scoret.nsig/nsimu
      cat('Power␣of␣wald␣test=␣',multi.power)
      cat('␣Power␣of␣non-parametric␣test=␣',np.power)
      cat('␣Power␣of␣score-type␣test=␣',scoret.power)
}

##################################################
############## 2 outcomes #######################
##################################################

# empirical size
margin.vec1 = c(1/2,1/4,0.24,0.01)
margin.vec2 = c(0.7,0.2,0.05,0.05)

margin = rbind(margin.vec1,margin.vec1,margin.vec2,
margin.vec2)
prob.vec = smh_ipf(nout=2,nlevel=4,margin=margin)
s = matrix(0,nrow=2,ncol=8)
s[1,1:4] = 1:4
s[2,5:8] = 1:4

mout_emp(nsimu=5000,nout=2,nsize=25,nlevel=4,alpha=0.01,score=s)
mout_emp(nsimu=5000,nout=2,nsize=50,nlevel=4,alpha=0.01,score=s)
mout_emp(nsimu=5000,nout=2,nsize=100,nlevel=4,alpha=0.01,score=s)
mout_emp(nsimu=5000,nout=2,nsize=200,nlevel=4,alpha=0.01,score=s)

mout_emp(nsimu=5000,nout=2,nsize=25,nlevel=4,alpha=0.05,score=s)
mout_emp(nsimu=5000,nout=2,nsize=50,nlevel=4,alpha=0.05,score=s)
mout_emp(nsimu=5000,nout=2,nsize=100,nlevel=4,alpha=0.05,score=s)
mout_emp(nsimu=5000,nout=2,nsize=200,nlevel=4,alpha=0.05,score=s)

mout_emp(nsimu=5000,nout=2,nsize=25,nlevel=4,alpha=0.1,score=s)
mout_emp(nsimu=5000,nout=2,nsize=50,nlevel=4,alpha=0.1,score=s)
mout_emp(nsimu=5000,nout=2,nsize=100,nlevel=4,alpha=0.1,score=s)
mout_emp(nsimu=5000,nout=2,nsize=200,nlevel=4,alpha=0.1,score=s)

#Empirical Power
```

```
margin.vec1 = c(1/2,1/4,0.24,0.01)
margin.vec2 = c(0.7,0.2,0.05,0.05)

margin = rbind(margin.vec1,margin.vec2,margin.vec1,
margin.vec2)
prob.vec = smh_ipf(nout=2,nlevel=4,margin=margin)
s = matrix(0,nrow=2,ncol=8)
s[1,1:4] = 1:4
s[2,5:8] = 1:4

mout_power(nsimu=5000,nout=2,nsize=25,nlevel=4,alpha=0.1,score=s)
mout_power(nsimu=5000,nout=2,nsize=50,nlevel=4,alpha=0.1,score=s)
mout_power(nsimu=5000,nout=2,nsize=100,nlevel=4,alpha=0.1,score=s)
mout_power(nsimu=5000,nout=2,nsize=200,nlevel=4,alpha=0.1,score=s)

mout_power(nsimu=5000,nout=2,nsize=25,nlevel=4,alpha=0.05,score=s)
mout_power(nsimu=5000,nout=2,nsize=50,nlevel=4,alpha=0.05,score=s)
mout_power(nsimu=5000,nout=2,nsize=100,nlevel=4,alpha=0.05,score=s)
mout_power(nsimu=5000,nout=2,nsize=200,nlevel=4,alpha=0.05,score=s)

mout_power(nsimu=5000,nout=2,nsize=25,nlevel=4,alpha=0.01,score=s)
mout_power(nsimu=5000,nout=2,nsize=50,nlevel=4,alpha=0.01,score=s)
mout_power(nsimu=5000,nout=2,nsize=100,nlevel=4,alpha=0.01,score=s)
mout_power(nsimu=5000,nout=2,nsize=200,nlevel=4,alpha=0.01,score=s)
```

```
#################################################
############## 3 outcomes #######################
#################################################

# empirical size
margin.vec1 = c(1/3,1/3,1/4,1/12)
margin.vec2 = c(0.45,1/3,1/6,0.05)
margin.vec3 = c(0.45,1/4,1/4,0.05)
margin = rbind(margin.vec1,margin.vec1,margin.vec2,
margin.vec2,margin.vec3,margin.vec3)
prob.vec = smh_ipf(nout=3,nlevel=4,margin=margin)
s = matrix(0,nrow=3,ncol=12)
s[1,1:4] = 1:4
s[2,5:8] = 1:4
s[3,9:12] = 1:4

mout_emp(nsimu=5000,nout=3,nsize=25,nlevel=4,alpha=0.01,score=s)
mout_emp(nsimu=5000,nout=3,nsize=50,nlevel=4,alpha=0.01,score=s)
mout_emp(nsimu=5000,nout=3,nsize=100,nlevel=4,alpha=0.01,score=s)
mout_emp(nsimu=5000,nout=3,nsize=200,nlevel=4,alpha=0.01,score=s)

mout_emp(nsimu=5000,nout=3,nsize=25,nlevel=4,alpha=0.05,score=s)
mout_emp(nsimu=5000,nout=3,nsize=50,nlevel=4,alpha=0.05,score=s)
mout_emp(nsimu=5000,nout=3,nsize=100,nlevel=4,alpha=0.05,score=s)
mout_emp(nsimu=5000,nout=3,nsize=200,nlevel=4,alpha=0.05,score=s)
mout_emp(nsimu=5000,nout=3,nsize=25,nlevel=4,alpha=0.1,score=s)
mout_emp(nsimu=5000,nout=3,nsize=50,nlevel=4,alpha=0.1,score=s)
mout_emp(nsimu=5000,nout=3,nsize=100,nlevel=4,alpha=0.1,score=s)
mout_emp(nsimu=5000,nout=3,nsize=200,nlevel=4,alpha=0.1,score=s)

#Empirical power
margin.vec1 = c(1/3,1/3,1/4,1/12)
margin.vec2 = c(0.45,1/3,1/6,0.05)
margin.vec3 = c(0.5,0.4,0.09,0.01)
margin.vec4 = c(0.3,0.4,0.2,0.1)
margin.vec5 = c(0.3,0.4,0.2,0.1)
margin.vec6 = c(0.5,0.25,0.2,0.05)

margin = rbind(margin.vec1,margin.vec2,margin.vec3,
margin.vec4,margin.vec5,margin.vec6)
prob.vec = smh_ipf(nout=3,nlevel=4,margin=margin)
s = matrix(0,nrow=3,ncol=12)
s[1,1:4] = 1:4
s[2,5:8] = 1:4
s[3,9:12] = 1:4

mout_power(nsimu=5000,nout=3,nsize=25,nlevel=4,alpha=0.01,score=s)
mout_power(nsimu=5000,nout=3,nsize=50,nlevel=4,alpha=0.01,score=s)
mout_power(nsimu=5000,nout=3,nsize=100,nlevel=4,alpha=0.01,score=s)
mout_power(nsimu=5000,nout=3,nsize=200,nlevel=4,alpha=0.01,score=s)

mout_power(nsimu=5000,nout=3,nsize=25,nlevel=4,alpha=0.05,score=s)
mout_power(nsimu=5000,nout=3,nsize=50,nlevel=4,alpha=0.05,score=s)
mout_power(nsimu=5000,nout=3,nsize=100,nlevel=4,alpha=0.05,score=s)
mout_power(nsimu=5000,nout=3,nsize=200,nlevel=4,alpha=0.05,score=s)

mout_power(nsimu=5000,nout=3,nsize=25,nlevel=4,alpha=0.1,score=s)
mout_power(nsimu=5000,nout=3,nsize=50,nlevel=4,alpha=0.1,score=s)
mout_power(nsimu=5000,nout=3,nsize=100,nlevel=4,alpha=0.1,score=s)
mout_power(nsimu=5000,nout=3,nsize=200,nlevel=4,alpha=0.1,score=s)

###################################################
############### 4 outcomes ########################
###################################################

# empirical size
margin.vec1 = c(1/2,1/4,0.24,0.01)
margin.vec2 = c(0.7,0.2,0.05,0.05)
margin.vec3 = c(0.4,0.25,0.25,0.1)
margin.vec4 = c(0.7,0.15,0.11,0.04)

margin = rbind(margin.vec1,margin.vec1,margin.vec2,
margin.vec2,margin.vec3,margin.vec3,margin.vec4,
margin.vec4)
```

```
prob.vec = smh_ipf(nout=4,nlevel=4,margin=margin)
s = matrix(0,nrow=4,ncol=16)
s[1,1:4] = 1:4
s[2,5:8] = 1:4
s[3,9:12] = 1:4
s[4,13:16] = 1:4

mout_emp(nsimu=5000,nout=4,nsize=25,nlevel=4,alpha=0.1,score=s)
mout_emp(nsimu=5000,nout=4,nsize=50,nlevel=4,alpha=0.1,score=s)
mout_emp(nsimu=5000,nout=4,nsize=100,nlevel=4,alpha=0.1,score=s)
mout_emp(nsimu=5000,nout=4,nsize=200,nlevel=4,alpha=0.1,score=s)

mout_emp(nsimu=5000,nout=4,nsize=25,nlevel=4,alpha=0.05,score=s)
mout_emp(nsimu=5000,nout=4,nsize=50,nlevel=4,alpha=0.05,score=s)
mout_emp(nsimu=5000,nout=4,nsize=100,nlevel=4,alpha=0.05,score=s)
mout_emp(nsimu=5000,nout=4,nsize=200,nlevel=4,alpha=0.05,score=s)

mout_emp(nsimu=5000,nout=4,nsize=25,nlevel=4,alpha=0.01,score=s)
mout_emp(nsimu=5000,nout=4,nsize=50,nlevel=4,alpha=0.01,score=s)
mout_emp(nsimu=5000,nout=4,nsize=100,nlevel=4,alpha=0.01,score=s)
mout_emp(nsimu=5000,nout=4,nsize=200,nlevel=4,alpha=0.01,score=s)

#Empirical Power
margin.vec1 = c(1/2,1/4,0.24,0.01)
margin.vec2 = c(0.7,0.2,0.05,0.05)
margin.vec3 = c(0.4,0.25,0.25,0.1)
margin.vec4 = c(0.7,0.15,0.11,0.04)

margin = rbind(margin.vec1,margin.vec2,margin.vec3,
margin.vec4,margin.vec1,margin.vec3,margin.vec2,
margin.vec4)
prob.vec = smh_ipf(nout=4,nlevel=4,margin=margin)
s = matrix(0,nrow=4,ncol=16)
s[1,1:4] = 1:4
s[2,5:8] = 1:4
s[3,9:12] = 1:4
s[4,13:16] = 1:4

mout_power(nsimu=5000,nout=5,nsize=25,nlevel=4,alpha=0.01,score=s)
mout_power(nsimu=5000,nout=5,nsize=50,nlevel=4,alpha=0.01,score=s)
mout_power(nsimu=5000,nout=5,nsize=100,nlevel=4,alpha=0.01,score=s)
mout_power(nsimu=5000,nout=5,nsize=200,nlevel=4,alpha=0.01,score=s)

mout_power(nsimu=5000,nout=4,nsize=25,nlevel=4,alpha=0.05,score=s)
mout_power(nsimu=5000,nout=4,nsize=50,nlevel=4,alpha=0.05,score=s)
mout_power(nsimu=5000,nout=4,nsize=100,nlevel=4,alpha=0.05,score=s)
mout_power(nsimu=5000,nout=4,nsize=200,nlevel=4,alpha=0.05,score=s)

mout_power(nsimu=5000,nout=4,nsize=25,nlevel=4,alpha=0.1,score=s)
mout_power(nsimu=5000,nout=4,nsize=50,nlevel=4,alpha=0.1,score=s)
mout_power(nsimu=5000,nout=4,nsize=100,nlevel=4,alpha=0.1,score=s)
mout_power(nsimu=5000,nout=4,nsize=200,nlevel=4,alpha=0.1,score=s)
```

Power comparison between W_0 and W_{ord0}

```
#------------------------------------------------------------
#Parameters:
#nout:    number of outcomes
#nlevel: number of levels of the outcomes
#nsize:   sample size
#nsimu:   number of simulated datasets
#delta:   shift value
#pho1:    within-AE correlation
#pho2:    between-AE correlation
#alpha:   significance level
#score:   a matrix of scores assigned to the levels of the
#         outcomes.
#------------------------------------------------------------

library(MASS)
source(”smh_mult.R”)
source(”smh_ord.R”)

power_comp = function (nout, nlevel, nsize, nsimu, delta, pho1,
                       pho2, alpha, score) {

    cov = matrix(1,nrow=2*nout,ncol=2*nout)
    mult.p = rep(0,nsimu)
    ordinal.p = rep(0,nsimu)
```

```
for (i in 1:(2*nout)) {
    for (j in 1:(2*nout)) {
        h1 = ceiling(i/2)
        h2 = ceiling(j/2)
        if (h1 == h2 & i != j) {cov[i,j] = pho1}
        else if (i == j) {cov[i,j] = 1}
        else if (h1 != h2) {cov[i,j] = pho2}
    }
}

for (n in 1:nsimu)  {
    set.seed(n)
    dat = mvrnorm(n=nsize,mu=rep(0,2*nout),Sigma=cov)

    # cut points for dichotomizing the data
    dose1.cut = c(-0.6,0,0.6)
    dose2.cut = dose1.cut+delta

    # dychotomize
    dat2 = rep(0,2*nout)
    dat0 = array(0,rep(nlevel,nout*2))
    for (k in 1:nsize) {
          dat1 = dat[k,]
          for (i in seq(1,(2*nout)-1,2)) {
```

```
          if (dat1[i] <= dose1.cut[1]) {dat2[i] = 1}
          else if (dat1[i] > dose1.cut[1] & dat1[i]
                 <= dose1.cut[2]) {dat2[i] = 2}
          else if (dat1[i] > dose1.cut[2] & dat1[i]
                 <= dose1.cut[3]) {dat2[i] = 3}
          else if (dat1[i] > dose1.cut[3]) {dat2[i] = 4}
      }
      for (i in seq(2,(2*nout),2)) {
          if (dat1[i] <= dose2.cut[1]) {dat2[i] = 1}
          else if (dat1[i] > dose2.cut[1] & dat1[i]
                 <= dose2.cut[2]) {dat2[i] = 2}
          else if (dat1[i] > dose2.cut[2] & dat1[i]
                 <= dose2.cut[3]) {dat2[i] = 3}
          else if (dat1[i] > dose2.cut[3]) {dat2[i] = 4}
      }
      dat0[dat2[1],dat2[2],dat2[3],dat2[4],dat2[5],dat2[6],
         dat2[7],dat2[8]]
      = dat0[dat2[1],dat2[2],dat2[3],dat2[4],dat2[5],dat2[6],
         dat2[7],dat2[8]]+1
  }
  mult.p[n] = smh_mult(nout=nout,nlevel=nlevel,data=dat0,
            type='wald')$p
  ordinal.p[n] = smh_ord(nout=nout,nlevel=nlevel,score=score,
            data=dat0,   type='wald')$p
}
```

```
        mult.power = sum(mult.p<=alpha)/nsimu
        ordinal.power = sum(ordinal.p<=alpha)/nsimu
        cat('Power␣of␣multinomial␣method=␣',mult.power)
        cat('␣Power␣of␣ordinal␣method=␣',ordinal.power)
        pcomb = cbind(mult.p,ordinal.p)
        return(pcomb)
}

s = matrix(0,nrow=4,ncol=16)
s[1,1:4] = 1:4
s[2,5:8] = 1:4
s[3,9:12] = 1:4
s[4,13:16] = 1:4

# sample size=200
power_comp(nout=4, nlevel=4, nsize=200, nsimu=5000, delta=0.2,
              pho1=0.6, pho2=0.2, alpha=0.01, score=s)
power_comp(nout=4, nlevel=4, nsize=200, nsimu=5000, delta=0.2,
              pho1=0.6, pho2=0.2, alpha=0.05, score=s)
power_comp(nout=4, nlevel=4, nsize=200, nsimu=5000, delta=0.2,
              pho1=0.6, pho2=0.2, alpha=0.10, score=s)

# sample size=100
power_comp(nout=4, nlevel=4, nsize=100, nsimu=5000, delta=0.2,
              pho1=0.6, pho2=0.2, alpha=0.01, score=s)
```

```
power_comp(nout=4, nlevel=4, nsize=100, nsimu=5000, delta=0.2,
           pho1=0.6, pho2=0.2, alpha=0.05, score=s)
power_comp(nout=4, nlevel=4, nsize=100, nsimu=5000, delta=0.2,
           pho1=0.6, pho2=0.2, alpha=0.10, score=s)
```