COMPARISON OF PROPOSED K SAMPLE TESTS WITH DIETZ'S TEST FOR NONDECREASING ORDERED ALTERNATIVES FOR BIVARIATE EXPONENTIAL DATA

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Jyothsnadevi Pothana

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Statistics

June 2011

Fargo, North Dakota

North Dakota State University
Graduate School

Title

Comparison of Proposed k Sample Tests with Dietz's Test for

Nondecreasing Ordered Alternatives for Bivariate Exponential Data

By

Jyothsnadevi Pothana

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

ABSTRACT

Pothana, Jyothsnadevi, M.S., Department of Statistics, College of Science and Mathematics, North Dakota State University, June 2011. Comparison of Proposed k Sample Tests with Dietz's Test for Nondecreasing Ordered Alternatives for Bivariate Exponential Data. Major Professor: Dr. Rhonda Magel.

Comparison of powers is essential to determine the best test that can be used for data under certain specific conditions. Likewise, several nonparametric methods have been developed for testing the ordered alternatives. The Jonckheere-Terpstra (JT) test and the Modified Jonckheere-Terpstra (MJT) test are for testing nondecreasing ordered alternatives for univariate data. The Dietz test is for testing nondecreasing alternatives based on bivariate data. This paper compares various tests when testing for nondecreasing alternatives specifically when the underlying distributions are bivariate exponential. The JT test and the MJT test are applied to univariate data which is derived by reducing bivariate data to univariate data using various transformations. A Monte Carlo simulation study is conducted comparing the estimated powers of JT tests and MJT tests (based on a variety of transformed univariate data) with the estimated powers of Dietz test (based on bivariate data) under a variety of location shifts and sample sizes. The results are compared with Zhao's (2011) results for bivariate normal data. The overall best test statistic for bivariate data ordered alternatives is discussed in this paper.

ACKNOWLEDGEMENTS

I would like to thank Dr. Rhonda Magel for her continued help and direction. My sincere thanks to Dr. Gang Shen and Dr. Ron Degges for serving on the committee. I would like to thank my family and my friend Sowjanya Param who supported and motivated me to accomplish this task.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER 1. INTRODUCTION

Few medical research studies involve comparing two groups on only a single response variable; comparisons on two or more response variables are usually desired. If a single variable is identified as of major research interest, it would be appropriate to apply a two independent samples t-test or Mann-Whitney test assuming that there are independent samples from both populations. In many studies, however, two response variables are of equal interest and importance. For example, in studies comparing two different treatments for hypertension, it is equally important to compare their effects on both systolic and diastolic blood pressures. For such studies, a bivariate analysis that compares the treatments on two response variables simultaneously may have advantages over two separate univariate tests, one for each variable.

In most medical research, treatment effects are measured in changes of location. For example, when it is decided to compare two characteristics of a population, such as the weight and height of infants, with those of another population, the researcher tries to compare the bivariate locations in two independent populations. When the underlying population distributions are unknown, many nonparametric tests have been proposed.

In this paper we want to test for the differences in effects of k treatments based on bivariate data. Assume that if the treatments have different effects, this results in changes in the bivariate location parameters.

Here, assume that we don't know the type of underlying distributions (They could be bivariate normal, but they could also be some other type of bivariate distribution such as the bivariate exponential). Therefore, we are considering nonparametric tests. In this research, it is also assumed that when the treatment effects differ, they follow a nondecreasing pattern resulting in testing the following set of hypotheses: where x is a vector with two components $x = (x_1,x_2)$

$$H_0: F_1(x) = F_2(x) = \ldots. \quad F_k(x) \qquad (1)$$

$$H_1: F_1^g(x) \leq F_2^g(x) \leq \ldots. \leq F_k^g(x); \quad g = 1, 2$$

with at least one strict inequality for at least one g. $F_i(x)$ is the bivariate cumulative distribution function for population i and $F_i^g(x)$ is the marginal cumulative distribution function for population i and component g with g = 1, 2.

In some of the cases, it may be desirable and easier to reduce the bivariate data to univariate data and then test for nondecreasing treatment effects on the univariate data. For example, in cases of clinical studies testing the effects of a drug in lowering blood pressure, it is equally important to measure both systolic and diastolic blood pressures. In such cases, if we want to check whether this drug lowers blood pressure, it may be suitable and easier to transform the bivariate data to univariate data and apply a nonparametric test on the univariate data.

The Jonckheere Terpstra (JT) test (Jonckheere (1954) and Terpstra (1952)) is the most common nonparametric test used for testing the k location parameters are equal versus they follow a nondecreasing pattern. The Dietz test (1989) is a nonparametric test designed for bivariate data to test for nondecreasing ordered alternatives.

This paper considers proposed tests of Zhao (2011) which transform bivariate data to univariate data and then apply either the JT test or MJT test on the univariate data. Zhao (2011) compared these tests with Dietz (1989) test when the underlying distributions were bivariate normal. Our research compares the estimated powers of all of these tests when the underlying distribution are bivariate exponential.

Chapter 2 gives a description of some of the previous research and description of nonparametric tests used in the analysis. Chapter 3 discusses the design of simulation study. Chapter 4 discusses the results of the analysis and Chapter 5 discusses the conclusion.

CHAPTER 2. LITERATURE REVIEW

The Mann-Whitney test sometimes also called the Wilcoxon-Mann-Whitney test or the Wilcoxon Rank Sum test (Mann and Whitney (1947); Wilcoxon (1945)) is the most commonly used nonparametric test to test the equality of two location parameters based on independent samples.

The null hypothesis and alternative hypothesis for the Mann-Whitney test are given by

H_0: $\theta_1 = \theta_2$

H_1: $\theta_1 \neq \theta_2$

where θ_1, θ_2 are location parameters of first and second populations respectively.

Let $X_{i1}, X_{i2}, \ldots\ldots, X_{in}$; i = 1, 2 be independent random samples from two mutually independent continuous distributions, Mann and Whitney suggested an indicator function, $I\ (x_1,x_2)=1$ to represent the case in which the x_2 observation is greater than the x_1 observation, otherwise $I\ (x_1,x_2)=0$. In general, the Mann-Whitney test counts the number of ordered pairs in which the first component is less than the second component where the first component is an observation from the first sample and the second component is an observation from the second sample. The test statistic is as follows

$$U = \sum_{j1=1}^{n1} \sum_{j2=1}^{n2} I((x_{1j1}, x_{2j2})$$

$$where\ I(x_1, x_2) = \begin{cases} 1\ if\ x_1 < x_2 \\ 0\ otherwise \end{cases}$$

Exact critical values based on small samples sizes and various significance levels are available. For large sample sizes, the test statistic can be standardized and the standardized version has an asymptotic standard normal distribution when the null hypothesis is true. The large sample version may be found in Daniel (1990) as well as exact critical values for smaller sample sizes.

Nonparametric tests are also available in testing for differences in k location parameters when underlying distributions are unknown. One such test is the Kruskal-Wallis test. The Kruskal-Wallis (Kruskal and Wallis (1952) test is an extension of the Mann-Whitney test to test more than two samples. The null hypothesis and alternative hypothesis for the Kruskal-Wallis test are given by

H_0: $\theta_1 = \theta_2 = \ldots\ldots . = \theta_k$

H_1: $\theta_i \neq \theta_j$ (for at least one set of i and j)

where θ_i is the location parameter for population i; i = 1, 2, …., k.

Let n_i; (i = 1, 2, . , k) represent the sample sizes for each of the k groups (i.e., samples) in the data. Let the value R_i, equal to the sum of the ranks for group i. When calculating the Kruskal-Wallis test statistic, all the observations from all the samples are combined together and ranked from smallest to largest. The Kruskal-Wallis test statistic is:

$$H = \frac{12}{n(n+1)} \sum_{i=1}^{k} \frac{Ri^2}{n_i} - 3(n-1)$$

This statistic has approximately a chi-square distribution with k-1 degrees of freedom if the null hypothesis of equal parameters is true. Each of the values of n_i ; i = 1, 2,…, k should be at least 5 for the approximation to be valid (Daniel (1990)).

The Jonckheere–Terpstra test (JT) (Jonckheere (1954) and Terpstra (1952)) test is a nonparametric test for testing

H_0: $\theta_1 = \theta_2 = \ldots\ldots . = \theta_k$

H_1: $\theta_1 \leq \theta_2 \leq \ldots\ldots . \leq \theta_k$ (at least one inequality is strict) where θ_i represents the location parameter for population i, (i=1,2,...,k) assuming that the underlying populations are continuous and only differ in location.

The JT test is defined by

$$U_{lm} = \sum_{j_l=1}^{n_l} \sum_{j_m=1}^{n_m} I(x_{l_{j1}}, x_{m_{jm}}) \qquad 1 \leq l \leq m \leq k$$

$$\text{where } I(x_{l_{j1}}, x_{m_{jm}}) = \begin{cases} 1 & if\ x_{l_{j1}} < x_{m_{j2}} \\ 0 & if\ x_{l_{j1}} > x_{m_{j2}} \\ 0.5 & if\ x_{l_{j1}} = x_{m_{j2}} \end{cases}$$

$$JT = \sum_{l=1}^{k-1} \sum_{m=l+1}^{k} U_{lm}$$

Exact critical values for the JT test may be found for small sample sizes (Daniel (1990)). For larger sample sizes, the standardized version of the JT test should be found where the mean and standard deviation are given by

$$\mu_J = \frac{N^2 - \sum_{l=1}^{k} n_l^2}{4}$$

and

$$\sigma_J = \sqrt{\frac{N^2(2N+3) - \sum_{l=1}^{k} n_l^2 (2n_l + 3)}{72}}$$

where $N = n_1 + n_2 + \ldots\ldots + n_k$

The standardized version of the JT test is given by

$$Z = \frac{J - \mu_J}{\sigma_J}$$

Z has an asymptotic standard normal distribution when the null hypothesis is true. Therefore, the null hypothesis is rejected when $Z \geq z_\alpha$ where z_α is the value on the standard normal table with α area above it.

Neuhauser, Liu and Hothorn (1998) proposed a modified test to Jonckheere-Terpstra's test. Weights were added to the JT test so that if populations being compared were further apart, their comparison would have more weight.

The set of hypotheses being tested is

H_0: $\theta_1 = \theta_2 = \ldots\ldots\ldots = \theta_k$

H_1: $\theta_1 \leq \theta_2 \leq \ldots\ldots\ldots \leq \theta_k$ (at least one inequality is strict)

The modified Jonckheere-Terpstra (MJT) test was defined by

$$MJT = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} (j-i)U_{ij}$$

In case of 3 populations being compared, the MJT test is defined by

$$MJT = (U_{12} + 2\,U_{13} + U_{23})$$

In case of 4 populations being compared, the MJT test is defined by

$$MJT = (U_{12} + 2U_{13} + 3U_{14} + U_{23} + 2U_{24} + U_{34})$$

Under the null hypothesis, the standardized version of the MJT test has an asymptotic standard normal distribution. The standard version is denoted as MJT^* with

$$MJT^* = (MJT - mean) \,/\, sqrt(\,variance)$$

The mean is given by

$$E(MJT) = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} (j-1)\frac{n_i n_j}{2}$$

and the variance is given by

$$var(MJT) = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} (j-i)^2 var(\,u_{ij}) + 2\sum_{i=1}^{k-1} \sum_{j=1}^{k} \sum_{l=1}^{k-1} \sum_{m=1}^{k} (j-i)(m-l) cov(\,u_{ij}, u_{lm})$$

where

$Var(U_{ij}) = 1/12 * n_i n_j (n_i + n_j + 1) \quad \forall i \neq j$

$Cov(U_{ij}, U_{il}) = Cov(U_{ji}, U_{li}) = 1/12 * n_i n_j n_l$ *if all i, j, l are different*

$Cov(U_{ij}, U_{li}) = Cov(U_{ji}, U_{il}) = -1/12 * n_i n_j n_l$ *if all i, j, l are different*

$Cov(U_{ij}, U_{lm}) = 0$ *if all i, j, l, m are different*

The null hypothesis is rejected if $MJT^* \geq z_\alpha$.

Dietz (1989) proposed a multivariate generalization of the JT univariate test for ordered alternatives. She considered each multivariate test statistic as a function of a coordinate wise Jonckheere-Terpstra statistic. Assuming that X_{ij}'s are independent with continuous distribution functions $F_j(x)$ and marginal distribution functions of $F_j^{(1)}(x), \ldots, F_j^{(p)}(x)$. The null and alternative hypothesis in terms of bivariate data is given by:

$H_o = F_1(x) = \ldots\ldots = F_k(x)$ for all x;

H_a: $F_1^{(g)}(x) \leq \ldots\ldots \leq F_k^{(g)}(x)$ for all x. (at least one inequality is strict)

for at least one g =1, .., p or one j = 1, 2, …,k. where x is a 2 x 1 vector of observations for the ith subject in treatment j; i = 1, 2,. . , n_j.

Let $R_l = (R_{l1}, \ldots, R_{lp})'$, l=1,…. , N, be p-vectors of ranks corresponding to the X_{ij}, where $N = \sum_{j=1}^{k} n_j$ observations for each coordinate are ranked among themselves. When calculating the Dietz test statistic, a JT test statistic is calculated for the first set of coordinates and then for the second set of coordinates. These two JT test statistics are centered by subtracting the means. The centered JT test statistics are given below:

$$J_g = \sum_{i<j}^{k} U_{uv}^{(g)} - n_u n_v/2) \qquad g=1, 2$$

$$where \quad U_{uv}^{(g)} = \sum_{i=1}^{n_u} \sum_{l'=1}^{n_v} \Phi(X_{iug}, X_{i'vg}).$$

The variances of J_g; g = 1, 2 and covariance of J_1 and J_2 are given by Dietz (1989)

$var J_g = (N^2(2N+3) - \sum_{j=1}^{k} n_j^2(2n_j+3))/72$

$cov(J_1, J_2) = (N+1)[N^3 - \sum_{j=1}^{k} n_j^3 - 3(N^2 - \sum_{j=1}^{k} n_j^2]\ r_{12}/36(N-2) +$

$$[3N(N^2 - \sum_{j=1}^{k} n_j^2 - 2(N^3 - \sum_{j=1}^{k} n_j^3)]\ \tilde{\Gamma}_{12}/24(N-2)$$

where r_{12} is Spearman's correlation coefficient and $\tilde{\Gamma}_{12}$ is Kendall's correlation coefficient calculated based on samples 1 and 2.

- Spearman correlation correlation for the 1st and 2nd coordinate is given by

  $r_{12} = 3\sum_{1,j,k}^{N} sign\,(R_{j1} - R_{i1})(R_{j2} - R_{i2})/N(N^2-1)$

- Kendall correlation coefficient for the 1st and 2nd coordinate is given by

  $\tilde{\Gamma}_{12} = 2\sum_{i<j}^{N} sign\,(R_{j1} - R_{i1})(R_{j2} - R_{i2})/N(N-1)$

The Dietz test statistic J is given by

$$J = \frac{J_1 + J_2}{\sqrt{varJ_1 + varJ_2 + 2cov(J_1 J_2)}}.$$

When the null hypothesis is true, the asymptotic distribution of J is standard normal and H_0 is rejected when $J \geq z_\alpha$.

Krogen and Magel (2000) proposed k sample tests for bivariate censored data for nondecreasing ordered alternatives. They applied these tests to the survival functions of k populations consisting of multiple observations on each subject where each observation is subjected to arbitrary right censoring. The proposed test takes the multivariate data and reduces it to univariate rank data using pseudo ranks(they are minimum, maximum, sum functions) as in Leconte, Moreau and Lellouch (1994). The transformed univariate data is then used in the weighted log rank test as in Liu (1993). The weights used correspond to log rank, Gehan-Wilcoxon and Peto-Prentice tests.

Zhao (2011) proposed k sample tests for bivariate normal data for nondecreasing ordered alternatives. The proposed tests take the multivariate data and reduce this to univariate rank data using minimum, maximum and sum transformations. The transformed data is used in the JT test and Modified JT tests and compared with Dietz test on the basis of estimated powers. From overall simulation results she found that the sum transformation has the highest powers based on the transformations applied. MJT sum and JT sum tests results were close in estimated powers with MJT sum having a little higher powers. Overall, Zhao found the Dietz test was the most powerful.

This paper continues the work of Zhao (2011) and compares the estimated powers of univariate tests based on transformed data with the Dietz test when the underlying distributions are bivariate exponential. The results of this research are compared with the results of Zhao (2011) and an overall recommendation is made.

CHAPTER 3. SIMULATION STUDY

This chapter describes all the details of the simulation study set up to compare estimated powers of Dietz test with tests based on the transformed univariate data proposed by Zhao (2011). Powers were estimated when the underlying distributions were bivariate exponential.

We will begin with describing how random samples from a bivariate exponential distribution were generated. The method used was proposed by Kundu and Gupta (2009). In order to generate one observation from a bivariate exponential distribution, three mutually independent random values were generated from an exponential distribution with mean $1/\lambda$.

$$U_i \sim Exp\ (\lambda) \quad i=1,\ 2,\ 3$$

Define two variables X_1 and X_2 where

$$X_1 = max\ \{U_1, U_3\}, \quad X_2 = max\ \{U_2, U_3\}$$

The ordered pair (X_1, X_2) is a sample of size one from a bivariate exponential with scale parameter λ and shape parameter $\alpha=1$. If this process were done n independent times this would give us a random sample of size n. This process was used with k possibly different sample sizes; $n_1, n_2, \ldots, n_k$ to generate k independent random samples each of size n_i. The SAS function rand ('exponential') was used to generate each initial set of values from an exponential distribution with mean $1/\lambda$.

In this simulation study, the value of λ was always 1. A bivariate location parameter shift was then added to each value of the i^{th} sample (i=1, 2, . . . , k). Namely, if the shift parameter was (θ_1, θ_2) for sample i, the value θ_1 would be added to the first coordinates of each observation in sample i and value θ_2 would be added to the second coordinates of each observation in sample i.

Next, we will discuss the transformations used to reduce bivariate data to univariate data. Let (X_{1ij}, X_{2ij}) ; i=1, 2, ... , k; j=1, 2, .. , n_i represent k bivariate random samples each of size n_i. The first coordinates of all the observations from the k samples are ranked from smallest to largest. The second coordinates of all the observations are also ranked from smallest to largest. Let these ranks be denoted by (R_{1ij}, R_{2ij}); i = 1, 2,.... , k; j = 1,2,...,n_i. Three transformations are used in reducing bivariate data to univariate data. They are similar to the transformations proposed by Leconte et al. (1994). These are as follows.

$Sum\ (R_{1ij}, R_{2ij}) = R_{1ij} + R_{2ij}$

$Max\ (R_{1ij}, R_{2ij}) = R_{1ij}\ if\ R_{1ij} \geq R_{2ij}$

$= R_{2ij}\ if\ R_{1ij} > R_{2ij}$

$Min\ (R_{1ij}, R_{2ij}) = R_{1ij}\ if\ R_{1ij} \leq R_{2ij}$

$= R_{2ij}\ if\ R_{1ij} < R_{2ij}$

The JT and MJT tests are applied on each of these transformed data sets. These tests will be denoted by JT_{SUM}, JT_{MAX}, JT_{MIN}, MJT_{SUM}, MJT_{MAX} and MJT_{MIN} depending on which transformation and test was used. The Dietz test was also included in the study. Since this test is based on bivariate data, no transformation of the data had to be applied before calculating the test statistic.

The statistics MJT_{SUM}, MJT_{MIN}, MJT_{MAX}, DIETZ, JT_{SUM}, JT_{MIN} and JT_{MAX} were compared on the basis of estimated power when sampling from bivariate exponential distributions. To estimate the power in a given situation, 5000 different data sets were simulated from the bivariate exponential distributions. The test statistics were calculated and the estimated power was found by taking the number of times H_0 was rejected divided by 5000. The procedure was done at the 0.05 significance level.

Significance levels were estimated for each test in all cases by keeping the means the same for all the populations. The test statistics were compared when the number of populations were 3, 4 and 5 (k=3, k=4, k=5) based on a variety of different combinations (cases) of bivariate location parameters. Comparisons were based on estimated powers for different location shifts and a variety of both equal and unequal sample sizes.

For three populations being compared, the sample sizes considered are given in Table 1.

Table 1. The Sample sizes used when k=3

n1	n2	n3
5	5	5
10	10	10
5	5	10
5	10	5
5	10	10
10	5	5

For four populations being compared, the sample sizes considered are given in Table 2.

Table 2. The Sample sizes used when k=4

n1	n2	n3	n4
5	5	5	5
10	10	10	10
5	5	10	10
5	5	5	10
5	5	10	5
5	10	10	5
10	10	5	5

For five populations being compared, the sample sizes considered are given in Table 3.

Table 3. The Sample sizes used when k=5

n1	n2	n3	n4	n5
5	5	5	5	5
10	10	10	10	10
5	5	5	10	10
5	5	10	10	5
5	5	10	10	10
10	10	5	5	5
10	10	10	5	5
10	10	5	10	10

Nine different cases of location shifts were considered for 3, 4 and 5 populations. Case 1 always had the same location shifts for all the populations. This was done so that the significance levels could be estimated as well as the powers.

For k=3, the following location shifts were considered:

k=3; Cases for Location Shifts

Case 1: (1,1), (1,1), (1,1)

Case 2: (1,1), (1,1), (1.5,1.5)

Case 3: (1,1), (1.5,1.5), (2,2)

Case 4: (1,1), (2,1.5), (2.5,2)

Case 5: (1,1), (1.5,1.2), (2.5,2.2)

Case 6: (1,1), (1.5,2), (2,2)

Case 7: (1,1), (1,1.5), (1,2)

Case 8: (1,1), (1,2), (1,4)

Case 9: (1,1), (2,1), (3,1)

Case 10: (1,1), (1.5,1), (3,1)

For k=4, the following location shifts were considered:

k=4; Cases for Location Shifts

Case 1: (1,1), (1,1), (1,1), (1,1)

Case 2: (1,1), (1.2,1.2), (1.4,1.4), (1.6,1.6)

Case 3: (1,1), (1,2), (1,3), (1,4)

Case 4: (1,1), (1.1,1.5), (1.2,2), (1.3,2.5)

Case 5: (1,1), (2,2), (2,2), (2,2)

Case 6: (1,1), (1.5,2.5), (2,2.5), (2,3)

Case 7: (1,1), (1,1) , (1,1), (2,2)

Case 8: (1,1), (1,1), (2,2), (2,2)

Case 9: (1,2), (1.2,2), (1.4,2), (2.2,2)

Case 10: (1,1), (1.2,1.2), (1.6,1.6), (2,2)

For k=5, the following location shifts were considered:

k=5; Cases for Location Shifts

Case 1: (1,1), (1,1), (1,1), (1,1)

Case 2: (1,1), (1.2,1.2), (1.4,1.4), (1.6,1.6), (1.8,1.8)

Case 3: (1,1), (1,2), (1,3), (1,4), (1,5)

Case 4: (1,1), (1.1,1.5), (1.2,2), (1.3,2.5), (1.4,3)

Case 5: (1,1), (2,2), (2,2), (2,2), (2,2)

Case 6: (1,1), (1.5,2.5), (2,2.5), (2,3), (2,2)

Case 7: (1,1), (1,1), (1,1), (1.5,1.5), (1.5,1.5)

Case 8: (1,1), (1,1), (1.5,1.5), (1.5,1.5), (1.5,1.5)

Case 9: (1,2), (1.2,2), (1.4,2), (2.2,2), (3,2)

Case 10: (1,1), (1.2,1.2), (1.2,1.2), (1.6,1.6), (2.2,2.2)

The results for the simulation study are given in Chapter 4. Estimated powers and estimated significance levels for all the tests under all conditions discussed are given.

CHAPTER 4. RESULTS

Simulation Results k=3: The estimated powers when k=3 are given in Tables 4-9.

Recall that for k=3, the ten cases of location shifts considered were the following:

Case 1: (1,1), (1,1), (1,1)

Case 2: (1,1), (1,1), (1.5,1.5)

Case 3: (1,1), (1.5,1.5), (2,2)

Case 4: (1,1), (2,1.5), (2.5,2)

Case 5: (1,1), (1.5,1.2), (2.5,2.2)

Case 6: (1,1), (1.5,2), (2,2)

Case 7: (1,1), (1,1.5), (1,2)

Case 8: (1,1), (1,2), (1,4)

Case 9: (1,1), (2,1), (3,1)

Case 10: (1,1), (1.5,1), (3,1)

The estimated powers for Case 1 are actually the estimated significance levels. The stated significance level was 0.05. For tests to be compared; the estimated significance levels should be around 0.05 or lower.

The following tables are results of the simulation using above location shits and different sample sizes (equal and unequal).

Table 4. Estimated Powers; k=3 n1= 5, n2=5, n3=5; α=0.05

CASE	MJT$_{SUM}$	MJT$_{MIN}$	MJT$_{MAX}$	DIETZ	JT$_{SUM}$	JT$_{MIN}$	JT$_{MAX}$
1	0.051	0.053	0.052	0.054	0.051	0.051	0.051
2	0.251	**0.265**	0.188	0.249	0.243	0.253	0.177
3	0.586	**0.629**	0.452	0.607	0.586	0.626	0.450
4	0.709	0.758	0.554	0.731	0.708	**0.759**	0.557
5	0.762	**0.788**	0.617	0.768	0.758	0.786	0.611
6	0.567	**0.610**	0.450	0.588	0.567	0.607	0.450
7	0.229	0.227	0.182	**0.247**	0.230	0.230	0.178
8	0.596	0.469	0.523	**0.678**	0.590	0.487	0.528
9	0.452	0.406	0.354	**0.508**	0.447	0.421	0.356
10	0.426	0.334	0.391	**0.467**	0.421	0.341	0.383

Table 5. Estimated Powers; k=3 n1=10, n2=10, n3=10; α=0.05

CASE	MJT$_{SUM}$	MJT$_{MIN}$	MJT$_{MAX}$	DIETZ	JT$_{SUM}$	JT$_{MIN}$	JT$_{MAX}$
1	0.048	0.045	0.050	0.047	0.046	0.044	0.051
2	0.399	**0.448**	0.288	0.399	0.395	0.443	0.283
3	0.850	0.885	0.700	0.863	0.852	**0.892**	0.701
4	0.933	0.958	0.813	0.945	0.933	**0.964**	0.820
5	0.961	0.974	0.872	0.967	0.964	**0.978**	0.876
6	0.827	0.864	0.687	0.841	0.826	**0.864**	0.693
7	0.365	0.385	0.276	0.384	0.362	**0.387**	0.276
8	0.904	0.790	0.839	**0.932**	0.900	0.800	0.843
9	0.739	0.700	0.591	**0.783**	0.735	0.712	0.596
10	0.716	0.580	0.657	**0.746**	0.713	0.592	0.660

Table 6. Estimated Powers; k=3 n1=5, n2=5, n3=10; α=0.05

CASE	MJT_{SUM}	MJT_{MIN}	MJT_{MAX}	DIETZ	JT_{SUM}	JT_{MIN}	JT_{MAX}
1	0.053	0.051	0.053	0.052	0.052	0.051	0.054
2	0.349	<u>0.387</u>	0.261	0.377	0.360	**0.404**	0.265
3	0.679	<u>0.731</u>	0.559	0.704	0.678	**0.732**	0.549
4	0.794	**0.844**	0.658	0.811	0.782	<u>0.838</u>	0.645
5	0.872	<u>0.903</u>	0.764	0.903	0.889	**0.913**	0.780
6	0.628	**0.682**	0.509	0.629	0.602	<u>0.660</u>	0.480
7	0.287	**0.322**	0.196	0.307	0.284	<u>0.320</u>	0.198
8	0.750	0.776	0.513	**0.824**	0.761	<u>0.781</u>	0.536
9	0.556	**0.613**	0.353	0.599	0.548	<u>0.607</u>	0.362
10	0.578	0.564	0.403	**0.634**	<u>0.586</u>	0.575	0.415

Table 7. Estimated Powers; k=3 n1=5, n2=10, n3=5; α=0.05

CASE	MJT_{SUM}	MJT_{MIN}	MJT_{MAX}	DIETZ	JT_{SUM}	JT_{MIN}	JT_{MAX}
1	0.055	0.053	0.052	0.051	0.052	0.051	0.050
2	<u>0.249</u>	**0.261**	0.188	0.241	0.240	0.246	0.181
3	0.598	<u>0.645</u>	0.457	0.612	0.596	**0.647**	0.455
4	0.725	<u>0.786</u>	0.564	0.748	0.724	**0.789**	0.563
5	0.781	<u>0.811</u>	0.618	0.787	0.775	**0.813**	0.617
6	0.568	**0.608**	0.456	0.585	0.562	<u>0.605</u>	0.453
7	0.233	<u>0.241</u>	0.185	**0.249**	0.229	0.238	0.184
8	<u>0.644</u>	0.517	0.614	**0.711**	0.637	0.527	0.616
9	<u>0.482</u>	0.457	0.376	**0.526**	0.471	0.462	0.373
10	<u>0.450</u>	0.334	0.446	**0.486**	0.443	0.342	0.441

Table 8. Estimated Powers; k=3 n1=5, n2=10, n3=10; α=0.05

CASE	MJT_{SUM}	MJT_{MIN}	MJT_{MAX}	DIETZ	JT_{SUM}	JT_{MIN}	JT_{MAX}
1	0.052	0.050	0.051	0.053	0.051	0.048	0.051
2	0.373	<u>0.418</u>	0.270	0.400	0.388	**0.435**	0.277
3	0.720	<u>0.761</u>	0.566	0.734	0.718	**0.766**	0.559
4	0.820	**0.873**	0.662	0.835	0.817	<u>0.872</u>	0.654
5	0.913	0.935	0.793	<u>0.937</u>	0.931	**0.949**	0.816
6	0.632	**0.689**	0.504	0.626	0.602	<u>0.663</u>	0.472
7	0.291	**0.333**	0.207	0.301	0.286	<u>0.326</u>	0.208
8	0.818	0.766	0.663	**0.873**	<u>0.830</u>	0.763	0.703
9	0.601	**0.641**	0.419	<u>0.639</u>	0.596	0.628	0.427
10	0.627	0.562	0.522	**0.685**	<u>0.645</u>	0.571	0.541

Table 9. Estimated Powers; k=3 n1=10, n2=5, n3=5; α=0.05

CASE	MJT_{SUM}	MJT_{MIN}	MJT_{MAX}	DIETZ	JT_{SUM}	JT_{MIN}	JT_{MAX}
1	0.053	0.051	0.048	0.053	0.052	0.050	0.049
2	<u>0.261</u>	**0.283**	0.194	0.237	0.238	0.251	0.177
3	0.707	<u>0.748</u>	0.545	0.714	0.702	**0.752**	0.546
4	0.852	<u>0.887</u>	0.702	0.868	0.854	**0.890**	0.708
5	<u>0.857</u>	**0.877**	0.706	0.836	0.825	0.855	0.677
6	0.728	<u>0.757</u>	0.575	0.746	0.738	**0.775**	0.590
7	0.270	0.235	0.235	**0.294**	<u>0.271</u>	0.243	0.229
8	0.698	0.447	**0.755**	<u>0.751</u>	0.683	0.467	0.717
9	<u>0.548</u>	0.403	0.527	**0.601**	0.544	0.418	0.512
10	0.486	0.322	**0.533**	<u>0.511</u>	0.469	0.326	0.491

From the simulation results for k=3, it can be observed that when both coordinates are changing JT_{MIN} or MJT_{MIN} generally have higher estimated powers than the other tests. This is observed in Cases 2 - 6. It can also be observed that in the Cases 7, 8, 9 and 10 where only one coordinate is changing and another coordinate is constant, the Dietz generally has higher estimated powers than the other tests.

Simulation Results k=4: The estimated powers when k=4 are given in Tables 10-16.

Recall that for k=4, the ten cases of location shifts considered were the following:

Case 1: (1,1), (1,1), (1,1), (1,1)

Case 2: (1,1), (1.2,1.2), (1.4,1.4), (1.6,1.6)

Case 3: (1,1), (1,2), (1,3), (1,4)

Case 4: (1,1), (1.1,1.5), (1.2,2), (1.3,2.5)

Case 5: (1,1), (2,2), (2,2), (2,2)

Case 6: (1,1), (1.5,2.5), (2,2.5), (2,3)

Case 7: (1,1), (1,1), (1,1), (2,2)

Case 8: (1,1), (1,1), (2,2), (2,2)

Case 9: (1,2), (1.2,2), (1.4,2), (2.2,2)

Case 10: (1,1), (1.2,1.2), (1.6,1.6), (2,2)

Table 10. Estimated Powers; k=4 n1=5, n2=5, n3=5, n4=5; α=0.05

CASE	MJT_{SUM}	MJT_{MIN}	MJT_{MAX}	DIETZ	JT_{SUM}	JT_{MIN}	JT_{MAX}
1	0.055	0.050	0.050	0.056	0.057	0.051	0.053
2	0.342	<u>0.393</u>	0.252	0.352	0.347	**0.398**	0.260
3	<u>0.721</u>	0.638	0.575	**0.799**	0.721	0.668	0.599
4	0.523	0.544	0.396	<u>0.546</u>	0.522	**0.556**	0.400
5	0.473	**0.526**	0.387	0.500	0.471	<u>0.516</u>	0.385
6	0.797	<u>0.851</u>	0.654	0.822	0.798	**0.855**	0.654
7	0.500	**0.516**	0.373	0.485	0.491	<u>0.504</u>	0.368
8	0.729	**0.765**	0.590	0.728	0.713	<u>0.751</u>	0.585
9	0.269	0.231	0.242	**0.289**	<u>0.270</u>	0.239	0.247
10	0.646	<u>0.687</u>	0.495	0.661	0.652	**0.702**	0.504

Table 11. Estimated Powers; k=4 n1=5, n2=5, n3=10, n4=5; α=0.05

CASE	MJT_{SUM}	MJT_{MIN}	MJT_{MAX}	DIETZ	JT_{SUM}	JT_{MIN}	JT_{MAX}
1	0.050	0.047	0.048	0.045	0.048	0.048	0.051
2	0.348	**0.397**	0.263	0.358	0.353	<u>0.396</u>	0.267
3	0.758	<u>0.782</u>	0.551	**0.821**	0.760	0.775	0.588
4	0.539	<u>0.605</u>	0.382	0.568	0.541	**0.607**	0.393
5	<u>0.499</u>	**0.554**	0.417	0.478	0.442	0.491	0.367
6	0.821	**0.878**	0.669	0.820	0.799	<u>0.858</u>	0.648
7	0.435	0.447	0.322	0.469	<u>0.470</u>	**0.482**	0.349
8	0.774	**0.814**	0.635	0.790	0.768	<u>0.812</u>	0.640
9	0.267	0.243	0.239	**0.288**	<u>0.274</u>	0.248	0.251
10	0.658	<u>0.713</u>	0.509	0.690	0.666	**0.739**	0.522

Table 12. Estimated Powers; k=4 n1=5, n2=5, n3=5, n4=10; α=0.05

CASE	MJT_{SUM}	MJT_{MIN}	MJT_{MAX}	DIETZ	JT_{SUM}	JT_{MIN}	JT_{MAX}
1	0.051	0.050	0.050	0.049	0.053	0.051	0.052
2	0.417	<u>0.478</u>	0.312	0.433	0.421	**0.482**	0.310
3	0.825	0.852	0.575	**0.876**	0.819	<u>0.855</u>	0.626
4	0.626	**0.699**	0.440	0.655	0.626	<u>0.699</u>	0.450
5	0.484	**0.541**	0.404	0.480	0.442	<u>0.493</u>	0.369
6	0.853	**0.907**	0.703	0.861	0.838	<u>0.893</u>	0.687
7	0.762	0.790	0.608	<u>0.808</u>	0.808	**0.837**	0.652
8	0.801	**0.840**	0.663	0.794	0.777	<u>0.814</u>	0.641
9	0.374	0.367	0.295	**0.410**	<u>0.396</u>	0.385	0.312
10	0.766	<u>0.809</u>	0.620	0.794	0.778	**0.826**	0.625

Table 13. Estimated Powers; k=4 n1=5, n2=5, n3=10, n4=10; α=0.05

CASE	MJT_{SUM}	MJT_{MIN}	MJT_{MAX}	DIETZ	JT_{SUM}	JT_{MIN}	JT_{MAX}
1	0.049	0.047	0.052	0.049	0.048	0.046	0.052
2	0.424	0.482	0.316	0.434	0.421	**0.487**	0.313
3	0.849	**0.899**	0.599	0.890	0.848	0.877	0.660
4	0.640	**0.722**	0.453	0.666	0.637	0.716	0.466
5	0.502	**0.556**	0.415	0.452	0.413	0.462	0.346
6	0.858	**0.912**	0.726	0.854	0.827	0.883	0.692
7	0.739	0.765	0.579	0.817	0.823	**0.842**	0.661
8	0.801	**0.844**	0.682	0.793	0.767	0.809	0.652
9	0.383	0.362	0.309	**0.422**	0.405	0.377	0.338
10	0.771	0.822	0.626	0.802	0.789	**0.844**	0.636

Table 14. Estimated Powers; k=4 n1=5, n2=10, n3=10, n4=5; α=0.05

CASE	MJT_{SUM}	MJT_{MIN}	MJT_{MAX}	DIETZ	JT_{SUM}	JT_{MIN}	JT_{MAX}
1	0.051	0.047	0.053	0.049	0.052	0.048	0.052
2	0.370	**0.421**	0.275	0.377	0.369	0.419	0.274
3	0.814	0.776	0.680	**0.860**	0.811	0.791	0.689
4	0.580	0.626	0.437	0.605	0.581	**0.630**	0.437
5	0.439	**0.487**	0.364	0.448	0.409	0.455	0.345
6	0.824	**0.890**	0.662	0.834	0.814	0.882	0.651
7	0.474	**0.476**	0.346	0.433	0.438	0.435	0.320
8	0.862	0.889	0.730	0.887	0.879	**0.907**	0.756
9	0.281	0.239	0.264	**0.284**	0.268	0.238	0.256
10	0.714	0.768	0.552	0.733	0.725	**0.788**	0.558

Table 15. Estimated Powers; k=4 n1=10, n2=10, n3=10, n4=10; α=0.05

CASE	MJT_{SUM}	MJT_{MIN}	MJT_{MAX}	DIETZ	JT_{SUM}	JT_{MIN}	JT_{MAX}
1	0.047	0.051	0.047	0.047	0.047	0.051	0.048
2	0.560	0.634	0.414	0.573	0.559	**0.635**	0.412
3	0.964	0.936	0.878	**0.977**	0.963	0.944	0.886
4	0.803	0.837	0.642	0.823	0.801	**0.841**	0.639
5	0.748	**0.808**	0.633	0.767	0.731	0.786	0.620
6	0.784	**0.849**	0.597	0.801	0.779	0.847	0.597
7	0.815	**0.831**	0.634	0.794	0.801	0.819	0.626
8	0.952	**0.967**	0.858	0.954	0.948	0.962	0.852
9	0.465	0.395	0.415	**0.484**	0.460	0.398	0.410
10	0.905	0.933	0.766	0.918	0.908	**0.940**	0.767

Table 16. Estimated Powers; k=4 n1=10, n2=10, n3=5, n4=5; α=0.05

CASE	MJT_{SUM}	MJT_{MIN}	MJT_{MAX}	DIETZ	JT_{SUM}	JT_{MIN}	JT_{MAX}
1	0.046	0.049	0.049	0.049	0.048	0.044	0.053
2	0.427	0.483	0.301	0.435	0.425	**0.484**	0.308
3	0.866	0.688	0.847	**0.899**	0.856	0.754	0.812
4	0.314	0.359	0.238	0.362	0.352	**0.403**	0.271
5	0.552	**0.555**	0.389	0.431	0.436	0.446	0.318
6	0.862	**0.887**	0.703	0.818	0.822	0.848	0.663
7	0.705	0.753	0.579	0.797	0.771	**0.811**	0.656
8	0.950	0.970	0.830	0.965	0.958	**0.977**	0.850
9	0.308	0.238	**0.310**	0.304	0.292	0.247	0.279
10	0.773	**0.814**	0.595	0.753	0.749	0.794	0.570

From the simulation results for k=4, it can be observed that when both coordinates are changing JT_{MIN} or MJT_{MIN} generally have higher estimated powers than other tests (Cases 2, 4, 5, 6, 7, 8 and 10). It can also be observed that in Cases 3 and 9 where only one coordinate is changing and the coordinate is constant, the Dietz generally has higher estimated powers than the other tests.

Simulation Results k=5: The estimated powers when k=5 are given in Tables 17-24.

Recall that for k=5, the ten cases of location shifts considered were the following:

Case 1: (1,1), (1,1), (1,1), (1,1)

Case 2: (1,1), (1.2,1.2), (1.4,1.4), (1.6,1.6), (1.8,1.8)

Case 3: (1,1), (1,2), (1,3), (1,4), (1,5)

Case 4: (1,1), (1.1,1.5), (1.2,2), (1.3,2.5), (1.4,3)

Case 5: (1,1), (2,2), (2,2), (2,2), (2,2)

Case 6: (1,1), (1.5,2.5), (2,2.5), (2,3), (2,2)

Case 7: (1,1), (1,1), (1,1), (1.5,1.5), (1.5,1.5)

Case 8: (1,1), (1,1), (1.5,1.5), (1.5,1.5), (1.5,1.5)

Case 9: (1,2), (1.2,2), (1.4,2), (2.2,2), (3,2)

Case 10: (1,1), (1.2,1.2), (1.2,1.2), (1.6,1.6), (2.2,2.2)

Table 17. Estimated Powers; k=5 n1=5, n2=5, n3=5, n4=5, n5=5; α=0.05

CASE	MJT_{SUM}	MJT_{MIN}	MJT_{MAX}	DIETZ	JT_{SUM}	JT_{MIN}	JT_{MAX}
1	0.050	0.048	0.052	0.047	0.052	0.049	0.054
2	0.531	0.593	0.391	0.548	0.534	**0.601**	0.392
3	0.906	0.828	0.776	**0.959**	0.905	0.859	0.813
4	0.751	0.777	0.583	0.783	0.753	**0.786**	0.596
5	0.424	**0.476**	0.349	0.459	0.425	0.472	0.352
6	0.552	**0.584**	0.469	**0.584**	0.545	0.571	0.471
7	0.365	**0.413**	0.270	0.363	0.359	0.403	0.270
8	0.360	**0.416**	0.275	0.376	0.362	0.414	0.280
9	0.548	0.434	0.505	**0.599**	0.548	0.448	0.515
10	0.397	0.451	0.294	0.404	0.395	**0.454**	0.298

Table 18. Estimated Powers; k=5 n1=10, n2=10, n3=10, n4=10, n5=10; α=0.05

CASE	MJT_{SUM}	MJT_{MIN}	MJT_{MAX}	DIETZ	JT_{SUM}	JT_{MIN}	JT_{MAX}
1	0.048	0.049	0.047	0.048	0.047	0.049	0.047
2	0.804	0.853	0.622	0.817	0.803	**0.862**	0.625
3	**0.998**	0.991	0.980	**0.998**	**0.998**	0.993	0.985
4	0.965	0.972	0.859	0.972	0.965	**0.973**	0.866
5	0.683	**0.738**	0.564	0.717	0.668	0.718	0.554
6	0.822	**0.863**	0.721	0.839	0.807	0.839	0.713
7	0.602	**0.670**	0.428	0.599	0.422	0.663	0.595
8	0.587	**0.662**	0.431	0.596	0.580	0.653	0.425
9	0.845	0.721	0.795	**0.871**	0.844	0.730	0.871
10	0.646	**0.711**	0.472	0.661	0.641	0.710	0.468

Table 19. Estimated Powers; k=5 n1=5, n2=5, n3=5, n4=10, n5=10; α=0.05

CASE	MJT_{SUM}	MJT_{MIN}	MJT_{MAX}	DIETZ	JT_{SUM}	JT_{MIN}	JT_{MAX}
1	0.049	0.048	0.054	0.048	0.048	0.045	0.050
2	0.646	0.713	0.494	0.661	0.639	**0.716**	0.489
3	0.966	**0.983**	0.775	**0.983**	0.964	0.974	0.845
4	0.854	**0.916**	0.661	0.878	0.855	0.906	0.675
5	0.458	**0.516**	0.381	0.413	0.369	0.422	0.313
6	0.535	**0.573**	0.466	0.418	0.366	0.378	0.345
7	0.486	**0.555**	0.360	0.493	0.471	0.542	0.356
8	0.418	**0.483**	0.317	0.387	0.369	0.427	0.283
9	0.728	0.735	0.551	**0.794**	0.753	0.739	0.620
10	0.499	**0.572**	0.378	0.498	0.476	0.547	0.358

Table 20. Estimated Powers; k=5 n1=10, n2 =10, n3=10, n4=5, n5=5; α=0.05

CASE	MJT_{SUM}	MJT_{MIN}	MJT_{MAX}	DIETZ	JT_{SUM}	JT_{MIN}	JT_{MAX}
1	0.047	0.050	0.048	0.046	0.047	0.049	0.049
2	0.682	**<u>0.749</u>**	0.505	0.684	0.672	<u>0.743</u>	0.498
3	<u>0.985</u>	0.905	0.971	**<u>0.992</u>**	0.981	0.945	0.960
4	<u>0.899</u>	0.886	0.786	**<u>0.906</u>**	0.888	0.896	0.765
5	0.683	0.738	0.558	<u>0.761</u>	0.728	**<u>0.777</u>**	0.610
6	0.837	0.874	0.719	<u>0.909</u>	0.887	**<u>0.917</u>**	0.780
7	0.415	**<u>0.449</u>**	0.291	0.374	0.370	<u>0.400</u>	0.263
8	0.526	<u>0.587</u>	0.381	0.555	0.543	**<u>0.607</u>**	0.404
9	<u>0.645</u>	0.440	**<u>0.688</u>**	0.632	0.597	0.448	0.623
10	0.502	**<u>0.555</u>**	0.360	0.495	0.486	<u>0.540</u>	0.3524

Table 21. Estimated Powers; k=5 n1=5, n2=5, n3=10, n4=5, n5=5; α=0.05

CASE	MJT_{SUM}	MJT_{MIN}	MJT_{MAX}	DIETZ	JT_{SUM}	JT_{MIN}	JT_{MAX}
1	0.050	0.048	0.052	0.050	0.049	0.047	0.053
2	0.543	<u>0.602</u>	0.402	0.557	0.540	**<u>0.605</u>**	0.402
3	<u>0.928</u>	0.868	0.827	**<u>0.961</u>**	0.920	0.886	0.846
4	0.766	0.793	0.609	**<u>0.797</u>**	0.765	<u>0.794</u>	0.609
5	0.425	**<u>0.483</u>**	0.353	0.439	0.397	<u>0.447</u>	0.332
6	0.547	<u>0.578</u>	0.472	**<u>0.587</u>**	0.545	0.565	0.469
7	0.371	**<u>0.416</u>**	0.272	0.378	0.378	<u>0.414</u>	0.269
8	0.366	<u>0.426</u>	0.276	0.385	0.368	**<u>0.426</u>**	0.281
9	<u>0.563</u>	0.433	0.551	**<u>0.601</u>**	0.556	0.443	0.551
10	0.408	<u>0.462</u>	0.299	0.418	0.407	**<u>0.462</u>**	0.299

Table 22. Estimated Powers; k=5 n1=10, n2=10, n3=5, n4=10, n5=10; α=0.05

CASE	MJT_{SUM}	MJT_{MIN}	MJT_{MAX}	DIETZ	JT_{SUM}	JT_{MIN}	JT_{MAX}
1	0.049	0.051	0.047	0.048	0.051	0.053	0.047
2	0.798	0.849	0.630	0.818	0.804	**0.854**	0.632
3	0.998	0.986	0.972	**0.999**	0.998	0.990	0.981
4	0.957	0.970	0.854	0.967	0.958	**0.971**	0.860
5	0.684	0.745	0.569	0.736	0.695	**0.751**	0.582
6	0.826	**0.868**	0.731	0.848	0.809	0.843	0.720
7	0.596	**0.668**	0.438	0.582	0.579	0.648	0.429
8	0.584	**0.664**	0.438	0.582	0.569	0.642	0.428
9	0.842	0.744	0.762	**0.868**	0.840	0.753	0.776
10	0.636	**0.710**	0.476	0.647	0.632	0.702	0.471

Table 23. Estimated Powers; k=5 n1=5, n2=10, n3=5, n4=10, n5=5; α=0.05

CASE	MJT_{SUM}	MJT_{MIN}	MJT_{MAX}	DIETZ	JT_{SUM}	JT_{MIN}	JT_{MAX}
1	0.048	0.043	0.050	0.050	0.048	0.044	0.048
2	0.613	0.679	0.453	0.627	0.615	**0.685**	0.455
3	0.962	0.925	0.879	**0.980**	0.961	0.939	0.898
4	0.843	0.864	0.679	0.865	0.843	**0.870**	0.683
5	0.379	**0.422**	0.311	0.415	0.377	0.420	0.311
6	0.635	0.665	0.545	**0.668**	0.632	0.660	0.547
7	0.492	**0.551**	0.359	0.495	0.481	0.540	0.353
8	0.482	**0.554**	0.360	0.490	0.475	0.543	0.357
9	0.652	0.557	0.584	**0.692**	0.654	0.569	0.597
10	0.492	**0.556**	0.361	0.501	0.491	0.551	0.361

Table 24. Estimated Powers; k=5 n1=10, n2=5, n3=10, n4=5, n5=10; α=0.05

CASE	MJT_{SUM}	MJT_{MIN}	MJT_{MAX}	DIETZ	JT_{SUM}	JT_{MIN}	JT_{MAX}
1	0.0480	0.047	0.049	0.048	0.046	0.046	0.049
2	0.7640	<u>0.830</u>	0.597	0.779	0.766	<u>**0.833**</u>	0.596
3	<u>0.992</u>	0.975	0.951	<u>**0.997**</u>	0.990	0.981	0.963
4	0.940	<u>**0.955**</u>	0.823	<u>**0.955**</u>	0.939	<u>**0.955**</u>	0.830
5	0.743	<u>**0.796**</u>	0.628	0.765	0.725	<u>0.776</u>	0.610
6	0.787	<u>**0.839**</u>	0.687	0.805	0.764	<u>0.808</u>	0.670
7	0.516	<u>**0.567**</u>	0.370	0.510	0.508	<u>0.562</u>	0.365
8	0.499	<u>**0.566**</u>	0.373	0.507	0.488	<u>0.556</u>	0.368
9	<u>0.794</u>	0.646	0.747	<u>**0.825**</u>	0.790	0.658	0.754
10	0.595	<u>**0.655**</u>	0.433	0.601	0.586	<u>0.653</u>	0.432

From the simulation results for k=5, it can be observed that when both coordinates are changing JT_{MIN} or MJT_{MIN} generally have higher estimated powers than other tests (Cases 2, 4, 5, 6, 7, 8 and 10). It can also be observed that in Cases 3 and 9 where only one coordinate is changing and another coordinate is constant, the Dietz test generally has higher estimated powers than the other tests.

CHAPTER 5. CONCLUSIONS

When analysis is to be done using bivariate data and we don't know the exact underlying distribution types, it is suitable to use nonparametric methods. This paper considers the case in which we have bivariate data and we want to test for nondecreasing ordered alternatives. In particular, we considered the case in which the underlying distributions were all bivariate exponential. Nonparametric tests were compared based on bivariate data being transformed to univariate data along with the Dietz test based on bivariate data. The transformations applied included the sum, minimum and maximum transformations.

Powers of all the nonparametric tests depends on location parameters and sample sizes. The number of populations considered was k=3, 4 and 5, and the significance level was always set at 0.05.

From our simulation study it can be observed that, when both the coordinates of the location parameters are changing, the JT or MJT test using the minimum transformation have higher estimated powers than the remaining tests. It can also be observed that the Dietz test has higher estimated powers when only one location coordinate is changing and the other location coordinate is constant in all the populations (see for k=3 Tables 4 to 9; Case 7, Case 8, Case 9, Case 10; for k=4 Tables 10 to 16; Case 3 and Case 9; for k=5 Tables 17 to 24 case 3, Case 9 from Chapter 4). It is also observed that when the Dietz test has higher estimated powers the estimated powers of JT_{SUM} or MJT_{SUM} are generally higher than the estimated powers of JT_{MIN} or MJT_{MIN}. This kind of behavior for estimated powers has the same pattern in both equal and unequal sample sizes for bivariate exponential data where k=3, k=4 and k=5.

Zhao (2011) compared these tests when the underlying distributions were bivariate normal with different mean shifts and correlation levels. From her simulation study she found that in most of the cases, the Dietz test had higher estimated powers than the other tests. When considering tests using the transformed data only, she found that JT_{SUM} or MJT_{SUM} had higher estimated powers.

Finally, if we combine our results with Zhao's results, we can conclude that, when k sample bivariate data is used for analysis and we don't know what the underlying distributions are, the recommended test considering only the tests based on transformed data would be either JT_{SUM} or MJT_{SUM}. These were the best tests to use in the bivariate normal case and they were the best tests to use in the bivariate exponential case if only one coordinate in the location parameters was changing.

REFERENCES

Dietz, E.J., (1989), "Multivariate Generalizations of Jonckheere's test for ordered alternatives". *Communications in Statistics,* 18, pp. 3763-3783.

Ferdhiana, R., Terpstra, J. and Magel, R. C., (2008), "A Nonparametric Test for the Ordered Alternative Based on Kendall's Correlation Coefficient". *Communications in Statistics - Simulation and Computation*, 37, pp. 1117 – 1128.

Krogen, B. L., and Magel, R. C., (2000), "Proposal of *k* Sample Tests for Bivariate Censored Data for Nondecreasing Ordered Alternatives". *Biometrical Journal*, 42, pp. 435–455.

Kundu, D., and Gupta, R. D., (2009), "Bivariate generalized exponential distribution". *Journal of Multivariate Analysis,* 100, pp. 581-593.

Leconte, E., Moreau, T., and Lellouch J., (1994), "The Two-sample Problem with Multivariate Censored Data: A New Rank Test Family". *Communications in Statistics - Simulation and Computation*, 23, pp. 1061 – 1076.

Mann, H. B., and Whitney, D. R., (1947), "On a Test of Whether one of Two Random Variables is Stochastically Larger than the other". *Annals of Mathematical Statistics,* 18, pp. 50–60.

Neuauser, M., Liu, P.Y., and Hothorn, L. A., (1998), "Nonparametric Tests for Trend: Jonckheere's Test, a Modification and a Maximum Test". *Biometrical Journal*, 40, pp. 899–909.

Terpstra T. J., (1952), "The Asymptotic Normality and Consistency of Kendall's test against Trend, when Ties are Present in One Ranking". *Indag. Math*, 14, pp. 327-333.

Tryon, P. V., and Hettmansperger T. P., (1973), "A Class of Nonparametric Tests for Homogeneity against Ordered Alternatives". *Annals of Statistics*, 1, pp. 1061-1070.

Wei, L. J., and Knuiman, M. W., (1987), "A one-sided rank test for multivariate censored data", *Australian Journal of Statistics*, 29, pp. 214-219.

APPENDIX A. SAS CODE FOR k=3 BIVARIATE EXPONENTIAL DATA TESTING ORDERED ALTERNATIVES USING JT, MJT AND DIETZ TEST

The program for k=3 bivariate exponential power estimation of unequal sample sizes (5, 10, 10) on case 9. (Results in Table 8)

```
options mprint symbolgen;

%macro generate1(samples,grp, n1, n2, n3, lambda ,seed);

data mc (keep=sample group size trt yield);

call streaminit(&seed);*** Initialize with desired seed.***;

n1=&n1;
n2=&n2;
n3=&n3;
array grpn{3} n1 n2 n3;

do sample=1 to &samples;
    do group=1 to &grp;
       do size=1 to grpn{group};
     trt='u1'; yield=rand('exponential') * &lambda; output ;
     trt='u2'; yield=rand('exponential') * &lambda; output ;
     trt='u3'; yield=rand('exponential') * &lambda; output;
          end;
   end;
end;
   run;
%mend generate1;

%generate1(5000,3,5,10,10,1,1254);

proc sort data=mc;
by sample group size;
run;
data u11( rename = (yield = yield1) );
set mc;
if trt='u1';
run;
proc sort data=u11;
by sample group size;
```

```
run;
data u22( rename = (yield = yield2) );
set mc;
if trt='u2';
run;
proc sort data=u22;
by sample group size;
run;
data u33( rename = (yield = yield3) );
set mc;
if trt='u3';
run;
proc sort data=u33;
by  sample group size;
run;
data xyz;
merge u11 u22 u33;
by sample group size;
run;
proc sort data=xyz;
by sample group size;
run;
data xyz1 ;
set xyz ;
x11=max(yield1,yield3);
x22=max(yield2,yield3);
by sample group size;
run ;
proc sort data=xyz1;
by sample group size;
run;
data xyz2;
set xyz1;
where group=1;
x1=x11;
x2=x22;
run;
data xyz3;
set xyz1;
where group=2;
x1=x11+1;
x2=x22;
run;
data xyz4;
set xyz1;
where group=3;
x1=x11+2;
```

```
x2=x22;
run;
data xyz5;
set xyz2 xyz3 xyz4;
by sample group size;
run;
proc sort data=xyz5;
by sample group size;
run;
proc rank data=xyz5 out=one ties=mean;
    by sample;
     var x1 x2;
  ranks rankedx1 rankedx2;
run;
data new1(drop=trt);
set one;
by sample group size;
sum1=sum(rankedx1, rankedx2);
max1=max(rankedx1, rankedx2);
min1=min(rankedx1, rankedx2);
run;
title1 "jt test for univariate_Normal";
proc sort data=new1;
    by sample group;
run;
proc sort data=new1;
    by sample group;
run;
ods listing close;                           *** Toggle standard
output off. ***;
ods output JTTest=sample_jts;
proc freq data=new1;
by sample;
   tables group*(sum1 max1 min1)/ jt; /*compare jt by using
sum max min*/
run;
ods listing;
data jt_S (keep=sample stdjt table);
  set sample_jts;
  where name1='Z_JT';
  rename nValue1=stdjt;
  run;
proc sort;
  by table;run;
title2 "jt test for univariate_Normal using sum";
data joinem1;
set jt_s;
```

```
where table ?'sum';
if stdjt<1.645 then Reject_JT=0;
else Reject_JT=1;
run;
proc means;
var Reject_JT ;
  run;
title2 "jt test for univariate_Normal using min";
data joinem2;
set jt_s;
where table ?'min';
if stdjt<1.645 then Reject_JT=0;
else Reject_JT=1;
run;
proc means;
var Reject_JT ;
  run;
title2 "jt test for univariate_Normal using max";
data joinem3;
set jt_s;
where table ?'max';
if stdjt<1.645 then Reject_JT=0;
else Reject_JT=1;
run;

proc means;
var Reject_JT ;
  run;


title1 "get dietz j (sum statistics)";

ods listing close;
ods output JTTest=sample_jts1;
proc freq data=one;
by sample;
   tables group*(rankedx1 rankedx2)/ jt; /*compare jt */
run;
ods listing;
data jt_S (keep=sample jt table);
  set sample_jts1;
  where name1='_JT_';
  rename nValue1=jt;
  run;

data jt_S (keep=sample jt table sumn);
set jt_S1 ;
```

```
n1=5;
n2=10;
n3=10;
sumn=(n1*n2+n2*n3+ n1*n3)/2;
run;
data joinem1(keep=sample jt table newjt1 sumn);
set jt_S;
where table ?'x1';
newjt1=jt-sumn;
run;
data joinem2(keep=sample jt table newjt2);;
set jt_S;
where table ?'x2';
newjt2=jt-sumn;
run;


title1 "get cov(j1,j2)";
proc means noprint data=xyz5;
   by sample group;
   var x1 x2;
   output out=countn n=ny1 ny2;
   run;
data countn1;
   set countn;
   tmpy1=ny1**2;
   tmpy2=ny2**2;
run;
proc sort data=countn1;
   by sample group;
   run;
proc means noprint data=countn1;
     by sample;
   var tmpy1 tmpy2;
   output out=temp3 sum=tmpy1 tmpy2;
   run;
data countn2;
   set countn;
   tmpy11=ny1**3;
   tmpy22=ny2**3;
run;
proc sort data=countn2;
   by sample group;
   run;
proc means noprint data=countn2;
by sample;
   var tmpy11 tmpy22;
```

```
   output out=temp4 sum=tmpy11 tmpy22;
run;
proc means noprint data=countn;
    by sample;
   var ny1;
   output out=temp5 sum=N;
 title2 "get spearman kendall correlation coeffition with
outpout";
 proc sort data=new1;
 by sample;
 run;
 proc corr data=one spearman  outs=spmancc;
 by sample;
 var rankedx1 rankedx2;
run;
proc sort data=spmancc;
    by sample;
run;
data spmancc;
   set spmancc;
   by sample;
    if _NAME_='rankedx2';
     keep sample _NAME_ rankedx1;
run;
proc corr data=one kendall outk=kdallcc;
     by sample;
     var rankedx1 rankedx2;
run;
proc sort data=kdallcc;
  by sample;
run;
data kdallcc;
   set kdallcc;
   by sample;
    if _NAME_='rankedx1';
     keep sample _NAME_ rankedx2; /*in order to get
different name for merging data*/
run;
title2 "get j1 j2";
proc sort data=sample_jts1;
by sample;
run;
data cnt;
   set countn;
   by sample group;
    tmp_y1=(2*ny1+3)*(ny1**2);
   tmp_y2=(2*ny2+3)*(ny2**2);
```

```
run;
proc sort data=cnt;
    by sample;
     run;
proc means noprint data=cnt;
by sample;
   var tmp_y1 tmp_y2;
   output out=temp1 sum=tmp_y1 tmp_y2;
   title2 ' 2nd component of var_j_g on page 3766 of Dietz';
run;
proc means noprint data=cnt;
    by sample;
   var ny1;
   output out=temp2 sum=N;
title2 ' N for var_j_g on page 3766 of Dietz';
run;
data combined;
merge temp1 temp2;
by sample;
a=(N**2)*(2*N+3)-tmp_y1;
varj1=a/72;
b=(N**2)*(2*N+3)-tmp_y2;
varj2=b/72;
run;
data temp6;
merge temp3 temp4 temp5 spmancc kdallcc ;
by sample;
keep sample tmpy1 tmpy2 tmpy11 tmpy22 N rankedx1 rankedx2 a1
a2 b1 b2 cov; /*rankedy1 is the spearman cc and rankedy2 is
the kendall cc*/
a1=(N**3-tmpy11)-3*(N**2-tmpy1);
a2=3*N*(N**2-tmpy1)-2*(N**3-tmpy11);
b1=(N+1)*a1*rankedx1/(36*N-72);
b2=a2*rankedx2/(24*N-48);
cov=b1+b2;
run;

title1 "get dietz j (sum statistics)";
data temp7;
merge combined temp6 joinem1 joinem2;
by sample;
keep sample newjt1 newjt2 varj1 varj2 cov j;
a11= newjt1+newjt2;
a22= varj1+varj2+2*cov;
a33=SQRT(a22);
j=a11/a33;
/*j=(j1+j2)/{var1+var2+2*cov(1,2)}**1/2*/
```

```
run;
data joinem1;
set temp7;
if j<1.645 then Reject_JT=0;
else Reject_JT=1;
run;
proc means;
var Reject_JT ;
run;


title1 "get mjt ";

data jo1;
set new1;
if group=1 or group=2;
run;
ods listing close;
ods output JTTest=sample_jts1;
proc freq data=jo1;
by sample;
tables group*(sum1 max1 min1)/ jt; /*compare jt by using sum
max min*/
run;
ods listing;
proc sort ;
by sample table;
run;
data sample_jt1;
set sample_jts1;
stdjt=1*nvalue1;
where name1='_JT_';
run;
data jo2;
set new1;
if group=1 or group=3;
run;
ods listing close;
ods output JTTest=sample_jts2;
proc freq data=jo2;
by sample;
tables group*(sum1 max1 min1)/ jt; /*compare jt by using sum
max min*/
run;
ods listing;
proc sort ;
by sample table;
```

```
run;
data sample_jt2;
set sample_jts2;
stdjt=2*nvalue1;
where name1='_JT_';
run;
proc print;
run;
data jo3;
set new1;
if group=2 or group=3;
run;
ods listing close;
ods output JTTest=sample_jts3;
proc freq data=jo3;
by sample;
tables group*(sum1 max1 min1)/ jt; /*compare jt by using sum
max min*/
run;
ods listing;
data sample_jt3;
set sample_jts3;
stdjt=1*nvalue1;
where name1='_JT_';
run;
proc sort;
 by sample table;run;
data jt_s(keep=sample group stdjt table);
set sample_jt1 sample_jt2 sample_jt3;
run;
proc means data=jt_s sum;
class sample;
var stdjt;
where table ?'sum';
output out=sumstat sum=sumstat1;
run;
proc means data=jt_s sum;
class sample;
var stdjt;
where table ?'min';
output out=minstat sum=minstat1;
run;
proc means data=jt_s;
class sample;
var stdjt;
where table ?'max';
output out=maxstat sum=maxstat1;
```

```
run;
data stat;
merge sumstat minstat maxstat;
by sample;
run;
Data jyo;
set stat;
IF sample ^= . ;
n1=5;
n2=10;
n3=10;
emjt=1/2*(n1*n2 + 2*(n1*n3) + n2*n3);
var12=1/12*(n1*n2)*(n1+n2+1);
var13= 1/12*(n1*n3)*(n1+n3+1);
var23= 1/12*(n2*n3)*(n2+n3+1);
covij=1/12*(n1*n2*n3)     ;
variance=(var12+4*var13+var23)-
(2*covij)+(4*covij)+(4*covij);
sqvar=sqrt(variance);
sumMjt=(sumstat1-emjt)/sqrt(variance);
minMjt=(minstat1-emjt)/sqrt(variance);
maxMjt=(maxstat1-emjt)/sqvar;
run;
title1 "get mjt (sum statistics)";
data sum;
set jyo;
if sumMjt<1.645 then Reject_JT=0;
else Reject_JT=1;
run;
proc means;
var Reject_JT ;
run;
title1 "get mjt (min statistics)";
data sum;
set jyo;
if minMjt<1.645 then Reject_JT=0;
else Reject_JT=1;
run;
proc means;
var Reject_JT ;
run;
title1 "get mjt (max statistics)";
data sum;
set jyo;
if maxMjt<1.645 then Reject_JT=0;
else Reject_JT=1;
run;proc means;var Reject_JT ;run;
```

APPENDIX B. SAS CODE FOR k=4 BIVARIATE EXPONENTIAL DATA TESTING ORDERED ALTERNATIVES USING JT, MJT AND DIETZ TEST

The program for k = 4 bivariate exponential data power estimation of unequal sample sizes (5, 5, 10, 10) on case 1. (Results in Table 13)

```
options mprint symbolgen;

%macro generate1(samples,grp, n1, n2, n3,n4,lambda ,seed);
data mc (keep=sample group size trt yield);
call streaminit(&seed);  *** Initialize with desired seed.
***;
n1=&n1;
n2=&n2;
n3=&n3;
n4=&n4;
array grpn{4} n1 n2 n3 n4;
do sample=1 to &samples;
    do group=1 to &grp;
       do size=1 to grpn{group};
trt='u1'; yield=rand('exponential') * &lambda; output ;
trt='u2'; yield=rand('exponential') * &lambda; output ;
trt='u3'; yield=rand('exponential') * &lambda; output;
     end;
   end;
end;
   run;
%mend generate1;
%generate1(5000,4,5,5,10,10,1,1254);
proc sort data=mc;
by sample group size;
run;
data u11( rename = (yield = yield1) );
set mc;
if trt='u1';
run;
proc sort data=u11;
by sample group size;
run;
data u22( rename = (yield = yield2) );
set mc;
```

```
if trt='u2';
run;
proc sort data=u22;
by sample group size;
run;
data u33( rename = (yield = yield3) );
set mc;
if trt='u3';
run;
proc sort data=u33;
by  sample group size;
run;
data xyz;
merge u11 u22 u33;
by sample group size;
run;
proc sort data=xyz;
by sample group size;
run;
data xyz1 ;
set xyz ;
x11=max(yield1,yield3);
x22=max(yield2,yield3);
by sample group size;
run ;
proc sort data=xyz1;
by sample group size;
run;
data xyz2;
set xyz1;
where group=1;
x1=x11;
x2=x22;
run;
data xyz3;
set xyz1;
where group=2;
x1=x11;
x2=x22;
run;
data xyz4;
set xyz1;
where group=3;
x1=x11;
x2=x22;
run;
data xyz6;
```

```
set xyz1;
where group=4;
x1=x11;
x2=x22;
run;
data xyz5;
set xyz2 xyz3 xyz4 xyz6;
by sample group size;
run;
proc sort data=xyz5;
by sample group size;
run;
proc rank data=xyz5 out=one ties=mean;
    by sample;
     var x1 x2;
  ranks rankedx1 rankedx2;
run;
data new1(drop=trt);
set one;
by sample group size;
sum1=sum(rankedx1, rankedx2);
max1=max(rankedx1, rankedx2);
min1=min(rankedx1, rankedx2);
run;

title1 "jt test for univariate_Normal";
proc sort data=new1;
    by sample group;
run;
ods listing close;
ods output JTTest=sample_jts;
proc freq data=new1;
by sample;
   tables group*(sum1 max1 min1)/ jt; /*compare jt by using
sum max min*/
run;
ods listing;
data jt_S (keep=sample stdjt table);
  set sample_jts;
  where name1='Z_JT';
  rename nValue1=stdjt;
  run;
 proc sort;
  by table;run;
title2 "jt test for univariate_Normal using sum";
data joinem1;
set jt_s;
```

```
where table ?'sum';
if stdjt<1.645 then Reject_JT=0;
else Reject_JT=1;
run;
proc means;
var Reject_JT ;
  run;
title2 "jt test for univariate_Normal using min";
data joinem2;
set jt_s;
where table ?'min';
if stdjt<1.645 then Reject_JT=0;
else Reject_JT=1;
run;
proc means;
var Reject_JT ;
  run;
title2 "jt test for univariate_Normal using max";
data joinem3;
set jt_s;
where table ?'max';
if stdjt<1.645 then Reject_JT=0;
else Reject_JT=1;
run;
proc means;
var Reject_JT ;
  run;

title1 "get dietz j (sum statistics)";

ods listing close;
ods output JTTest=sample_jts1;
proc freq data=one;
by sample;
   tables group*(rankedx1 rankedx2)/ jt; /*compare jt */
run;
ods listing;
data jt_S1 (keep=sample jt table);
  set sample_jts1;
  where name1='_JT_';
  rename nValue1=jt;
  run;
data jt_S (keep=sample jt table sumn);
set jt_S1 ;
n1=5;
n2=5;
n3=10;
```

```
n4=10;
sumn=(n1*n2+n2*n3+n3*n4+n1*n3+n1*n4+n2*n4)/2;
run;
data joinem1(keep=sample jt table newjt1 sumn);
set jt_S;
where table ?'x1';
newjt1=jt-sumn;
run;
data joinem2(keep=sample jt table newjt2);;
set jt_S;
where table ?'x2';
newjt2=jt-sumn;
run;
title1 "get cov(j1,j2)";
proc means noprint data=xyz5;
   by sample group;
   var x1 x2;
   output out=countn n=ny1 ny2;
   run;

data countn1;
   set countn;
   tmpy1=ny1**2;
   tmpy2=ny2**2;
run;
proc sort data=countn1;
   by sample group;
   run;
proc means noprint data=countn1;
     by sample;
   var tmpy1 tmpy2;
   output out=temp3 sum=tmpy1 tmpy2;
   run;

data countn2;
   set countn;
   tmpy11=ny1**3;
   tmpy22=ny2**3;
run;
proc sort data=countn2;
   by sample group;
   run;
proc means noprint data=countn2;
by sample;
   var tmpy11 tmpy22;
   output out=temp4 sum=tmpy11 tmpy22;
run;
```

```
proc means noprint data=countn;
    by sample;
   var ny1;
   output out=temp5 sum=N;

 title2 "get spearman kendall correlation coeffition with
outpout";
 proc sort data=new1;
 by sample;
 run;
 proc corr data=one spearman  outs=spmancc;
 by sample;
 var rankedx1 rankedx2;
run;
proc sort data=spmancc;
    by sample;
run;
data spmancc;
   set spmancc;
   by sample;
    if _NAME_='rankedx2';
     keep sample _NAME_ rankedx1;
run;
proc corr data=one kendall outk=kdallcc;
     by sample;
     var rankedx1 rankedx2;
run;
proc sort data=kdallcc;
  by sample;
run;
data kdallcc;
   set kdallcc;
   by sample;
    if _NAME_='rankedx1';
     keep sample _NAME_ rankedx2; /*in order to get
different name for merging data*/
run;

title2 "get j1 j2";
proc sort data=sample_jts1;
by sample;
run;
data cnt;
   set countn;
   by sample group;
    tmp_y1=(2*ny1+3)*(ny1**2);
```

```
   tmp_y2=(2*ny2+3)*(ny2**2);
run;
proc sort data=cnt;
    by sample;
     run;
proc means noprint data=cnt;
by sample;
   var tmp_y1 tmp_y2;
   output out=temp1 sum=tmp_y1 tmp_y2;
   title2 ' 2nd component of var_j_g on page 3766 of Dietz';
run;
proc means noprint data=cnt;
    by sample;
   var ny1;
   output out=temp2 sum=N;
title2 ' N for var_j_g on page 3766 of Dietz';
run;
data combined;
merge temp1 temp2;
by sample;
a=(N**2)*(2*N+3)-tmp_y1;
varj1=a/72;
b=(N**2)*(2*N+3)-tmp_y2;
varj2=b/72;
run;
data temp6;
merge temp3 temp4 temp5 spmancc kdallcc ;
by sample;
keep sample tmpy1 tmpy2 tmpy11 tmpy22 N rankedx1 rankedx2 a1
a2 b1 b2 cov; /*rankedy1 is the spearman cc and rankedy2 is
the kendall cc*/
a1=(N**3-tmpy11)-3*(N**2-tmpy1);
a2=3*N*(N**2-tmpy1)-2*(N**3-tmpy11);
b1=(N+1)*a1*rankedx1/(36*N-72);
b2=a2*rankedx2/(24*N-48);
cov=b1+b2;
run;

title1 "get dietz j (sum statistics)";
data temp7;
merge combined temp6 joinem1 joinem2;
by sample;
keep sample newjt1 newjt2 varj1 varj2 cov j;
a11= newjt1+newjt2;
a22= varj1+varj2+2*cov;
a33=SQRT(a22);
```

```
j=a11/a33;
/*j=(j1+j2)/{var1+var2+2*cov(1,2)}**1/2*/
run;
data joinem1;
set temp7;
if j<1.645 then Reject_JT=0;
else Reject_JT=1;
run;
proc means;
var Reject_JT ;
run;

title1 "get mjt ";
data jo1;
set new1;
if group=1 or group=2;
run;
ods listing close;
ods output JTTest=sample_jts1;
proc freq data=jo1;
by sample;
tables group*(sum1 max1 min1)/ jt; /*compare jt by using sum
max min*/
run;
ods listing;
proc sort ;
    by sample table;
run;
data sample_jt1;
set sample_jts1;
stdjt=1*nvalue1;
where name1='_JT_';
run;
data jo2;
set new1;
if group=1 or group=3;
run;
ods listing close;
ods output JTTest=sample_jts2;
proc freq data=jo2;
by sample;
tables group*(sum1 max1 min1)/ jt; /*compare jt by using sum
max min*/
run;
ods listing;
proc sort ;
    by sample table;
```

```
run;
data sample_jt2;
set sample_jts2;
stdjt=2*nvalue1;
where name1='_JT_';
run;
proc print;
run;
data jo3;
set new1;
if group=2 or group=3;
run;
ods listing close;
ods output JTTest=sample_jts3;
proc freq data=jo3;
by sample;
tables group*(sum1 max1 min1)/ jt; /*compare jt by using sum
max min*/
run;
ods listing;
data sample_jt3;
set sample_jts3;
stdjt=1*nvalue1;
where name1='_JT_';
run;
proc sort;
 by sample table;run;
data jo4;
set new1;
if group=2 or group=4;
run;
ods listing close;
ods output JTTest=sample_jts4;
proc freq data=jo4;
by sample;
tables group*(sum1 max1 min1)/ jt; /*compare jt by using sum
max min*/
run;
ods listing;
data sample_jt4;
set sample_jts4;
stdjt=2*nvalue1;
where name1='_JT_';
run;
proc sort;
 by sample table;run;
data jo5;
```

```
set new1;
if group=1 or group=4;
run;
ods listing close;
ods output JTTest=sample_jts5;
proc freq data=jo5;
by sample;
tables group*(sum1 max1 min1)/ jt; /*compare jt by using sum
max min*/
run;
ods listing;
data sample_jt5;
set sample_jts5;
stdjt=3*nvalue1;
where name1='_JT_';
run;
proc sort;
 by sample table;run;
data jo6;
set new1;
if group=3 or group=4;
run;
ods listing close;
ods output JTTest=sample_jts6;
proc freq data=jo6;
by sample;
tables group*(sum1 max1 min1)/ jt; /*compare jt by using sum
max min*/
run;
ods listing;
data sample_jt6;
set sample_jts6;
stdjt=1*nvalue1;
where name1='_JT_';
run;
proc sort;
 by sample table;run;
data jt_s(keep=sample group stdjt table);
set sample_jt1 sample_jt2 sample_jt3 sample_jt4 sample_jt5
sample_jt6;
run;
proc means data=jt_s sum;
class sample;
var stdjt;
where table ?'sum';
output out=sumstat sum=sumstat1;
run;
```

```
 proc means data=jt_s sum;
class sample;
var stdjt;
where table ?'min';
output out=minstat sum=minstat1;
run;
proc means data=jt_s;
class sample;
var stdjt;
where table ?'max';
output out=maxstat sum=maxstat1;
run;
data stat;
merge sumstat minstat maxstat;
by sample;
run;
Data jyo;
set stat;
IF sample ^= . ;
n1=5;
n2=5;
n3=10;
n4=10;
emjt=1/2*(n1*n2 + 2*(n1*n3) + n2*n3 + n3*n4 +3*(n1*n4)
+2*(n2*n4));
var12=1/12*(n1*n2)*(n1+n2+1);
var13= 1/12*(n1*n3)*(n1+n3+1);
var23= 1/12*(n2*n3)*(n2+n3+1);
var24= 1/12*(n2*n4)*(n2+n4+1);
var34= 1/12*(n3*n4)*(n3+n4+1);
var14= 1/12*(n1*n4)*(n1+n4+1);
covij=1/6*(3*n1*n2*n3 +7* n1*n2*n4 +3*n2*n3*n4 +7*n1*n3*n4);
var=(var12+4*var13+var23+4*var24+var34+9*var14);
variance=var+covij;
sqvar=sqrt(variance);
sumMjt=(sumstat1-emjt)/sqrt(variance);
minMjt=(minstat1-emjt)/sqrt(variance);
maxMjt=(maxstat1-emjt)/sqvar;
run;
title1 "get mjt (sum statistics)";
data sum;
set jyo;
if sumMjt<1.645 then Reject_JT=0;
else Reject_JT=1;
run;
proc means;
var Reject_JT ;
```

```
run;
title1 "get mjt (min statistics)";
data sum;
set jyo;
if minMjt<1.645 then Reject_JT=0;
else Reject_JT=1;
run;
proc means;
var Reject_JT ;
run;
title1 "get mjt (max statistics)";
data sum;
set jyo;
if maxMjt<1.645 then Reject_JT=0;
else Reject_JT=1;
run;
proc means;
var Reject_JT ;
run;
```

APPENDIX C. SAS CODE FOR k=5 BIVARIATE EXPONENTIAL DATA TESTING ORDERED ALTERNATIVES USING JT, MJT AND DIETZ TEST

The program for k=5 bivariate exponential data power estimation of unequal sample sizes

(5, 5, 10, 5, 5) on case 6. (Results in Table 21)

```
options mprint symbolgen;

%macro generate1(samples,grp, n1, n2, n3,n4,n5,lambda

,seed);

data mc (keep=sample group size trt yield);
call streaminit(&seed);   *** Initialize with desired seed.
***;
n1=&n1;
n2=&n2;
n3=&n3;
n4=&n4;
n5=&n5;
array grpn{5} n1 n2 n3 n4 n5;
do sample=1 to &samples;
    do group=1 to &grp;
       do size=1 to grpn{group};
     trt='u1'; yield=rand('exponential') * &lambda; output;
     trt='u2'; yield=rand('exponential') * &lambda; output;
     trt='u3'; yield=rand('exponential') * &lambda; output;
          end;
   end;
end;
   run;
%mend generate1;
%generate1(5000,5,5,5,10,5,5,1,1254);
proc sort data=mc;
by sample group size;
run;
data u11( rename = (yield = yield1) );
set mc;
if trt='u1';
run;
proc sort data=u11;
by sample group size;
```

```
run;
data u22( rename = (yield = yield2) );
set mc;
if trt='u2';
run;
proc sort data=u22;
by sample group size;
run;
data u33( rename = (yield = yield3) );
set mc;
if trt='u3';
run;
proc sort data=u33;
by  sample group size;
run;
data xyz;
merge u11 u22 u33;
by sample group size;
run;
proc sort data=xyz;
by sample group size;
run;

data xyz1 ;
set xyz ;
x11=max(yield1,yield3);
x22=max(yield2,yield3);
by sample group size;
run ;

proc sort data=xyz1;
by sample group size;
run;
data xyz2;
set xyz1;
where group=1;
x1=x11;
x2=x22;
run;
data xyz3;
set xyz1;
where group=2;
x1=x11+0.5;
x2=x22+1.5;
run;
data xyz4;
set xyz1;
```

```
where group=3;
x1=x11+1;
x2=x22+1.5;
run;
data xyz6;
set xyz1;
where group=4;
x1=x11+1;
x2=x22+2;
run;
data xyz7;
set xyz1;
where group=5;
x1=x11+1;
x2=x22+1;
run;
data xyz5;
set xyz2 xyz3 xyz4 xyz6 xyz7;
by sample group size;
run;
proc sort data=xyz5;
by sample group size;
run;
proc rank data=xyz5 out=one ties=mean;
    by sample;
     var x1 x2;
  ranks rankedx1 rankedx2;
run;
data new1(drop=trt);
set one;
by sample group size;
sum1=sum(rankedx1, rankedx2);
max1=max(rankedx1, rankedx2);
min1=min(rankedx1, rankedx2);
run;
title1 "jt test for univariate_Normal";
proc sort data=new1;
    by sample group;
run;
ods listing close;
ods output JTTest=sample_jts;
proc freq data=new1;
by sample;
   tables group*(sum1 max1 min1)/ jt; /*compare jt by using
sum max min*/
run;
ods listing;
```

```
data jt_S (keep=sample stdjt table);
  set sample_jts;
  where name1='Z_JT';
  rename nValue1=stdjt;
  run;
 proc sort;
  by table;run;
title2 "jt test for univariate_Normal using sum";
data joinem1;
set jt_s;
where table ?'sum';
if stdjt<1.645 then Reject_JT=0;
else Reject_JT=1;
run;
proc means;
var Reject_JT ;
  run;
title2 "jt test for univariate_Normal using min";
data joinem2;
set jt_s;
where table ?'min';
if stdjt<1.645 then Reject_JT=0;
else Reject_JT=1;
run;
proc means;
var Reject_JT ;
  run;
title2 "jt test for univariate_Normal using max";
data joinem3;
set jt_s;
where table ?'max';
if stdjt<1.645 then Reject_JT=0;
else Reject_JT=1;
run;
proc means;
var Reject_JT ;
  run;


  title1 "get dietz j (sum statistics)";

ods listing close;
ods output JTTest=sample_jts1;
proc freq data=one;
by sample;
   tables group*(rankedx1 rankedx2)/ jt; /*compare jt */
run;
```

```
ods listing;
data jt_S1 (keep=sample jt table);
  set sample_jts1;
  where name1='_JT_';
  rename nValue1=jt;
  run;
data jt_S (keep=sample jt table sumn);
set jt_S1 ;
n1=5;
n2=5;
n3=10;
n4=5;
n5=5;
sumn=(n1*n2+n2*n3+n3*n4+n4*n5+n1*n5+n3*n5+n2*n5+n1*n3+n1*n4+
n2*n4)/2;
run;
data joinem1(keep=sample jt table newjt1 sumn);
set jt_S;
where table ?'x1';
newjt1=jt-sumn;
run;
data joinem2(keep=sample jt table newjt2);;
set jt_S;
where table ?'x2';
newjt2=jt-sumn;
run;
title1 "get cov(j1,j2)";
proc means noprint data=xyz5;
   by sample group;
   var x1 x2;
   output out=countn n=ny1 ny2;
   run;

data countn1;
   set countn;
   tmpy1=ny1**2;
   tmpy2=ny2**2;
run;
proc sort data=countn1;
   by sample group;
   run;
proc means noprint data=countn1;
     by sample;
   var tmpy1 tmpy2;
   output out=temp3 sum=tmpy1 tmpy2;
   run;
```

```
data countn2;
   set countn;
   tmpy11=ny1**3;
   tmpy22=ny2**3;
run;
proc sort data=countn2;
   by sample group;
   run;
proc means noprint data=countn2;
by sample;
   var tmpy11 tmpy22;
   output out=temp4 sum=tmpy11 tmpy22;
run;

proc means noprint data=countn;
    by sample;
   var ny1;
   output out=temp5 sum=N;

 title2 "get spearman kendall correlation coeffition with
outpout";
 proc sort data=new1;
 by sample;
 run;
 proc corr data=one spearman  outs=spmancc;
 by sample;
 var rankedx1 rankedx2;
run;
proc sort data=spmancc;
    by sample;
run;
data spmancc;
   set spmancc;
   by sample;
    if _NAME_='rankedx2';
     keep sample _NAME_ rankedx1;
run;
proc corr data=one kendall outk=kdallcc;
     by sample;
     var rankedx1 rankedx2;
run;
proc sort data=kdallcc;
  by sample;
run;
data kdallcc;
   set kdallcc;
   by sample;
```

```
    if _NAME_='rankedx1';
     keep sample _NAME_ rankedx2; /*in order to get
different name for merging data*/
run;

title2 "get j1 j2";
proc sort data=sample_jts1;
by sample;
run;
data cnt;
   set countn;
   by sample group;
    tmp_y1=(2*ny1+3)*(ny1**2);
   tmp_y2=(2*ny2+3)*(ny2**2);
run;
proc sort data=cnt;
    by sample;
     run;
proc means noprint data=cnt;
by sample;
   var tmp_y1 tmp_y2;
   output out=temp1 sum=tmp_y1 tmp_y2;
   title2 ' 2nd component of var_j_g on page 3766 of Dietz';
run;
proc means noprint data=cnt;
    by sample;
   var ny1;
   output out=temp2 sum=N;
title2 ' N for var_j_g on page 3766 of Dietz';
run;
data combined;
merge temp1 temp2;
by sample;
a=(N**2)*(2*N+3)-tmp_y1;
varj1=a/72;
b=(N**2)*(2*N+3)-tmp_y2;
varj2=b/72;
run;
data temp6;
merge temp3 temp4 temp5 spmancc kdallcc ;
by sample;
keep sample tmpy1 tmpy2 tmpy11 tmpy22 N rankedx1 rankedx2 a1
a2 b1 b2 cov; /*rankedy1 is the spearman cc and rankedy2 is
the kendall cc*/
a1=(N**3-tmpy11)-3*(N**2-tmpy1);
a2=3*N*(N**2-tmpy1)-2*(N**3-tmpy11);
b1=(N+1)*a1*rankedx1/(36*N-72);
```

```
b2=a2*rankedx2/(24*N-48);
cov=b1+b2;
run;

title1 "get dietz j (sum statistics)";
data temp7;
merge combined temp6 joinem1 joinem2;
by sample;
keep sample newjt1 newjt2 varj1 varj2 cov j;
a11= newjt1+newjt2;
a22= varj1+varj2+2*cov;
a33=SQRT(a22);
j=a11/a33;
/*j=(j1+j2)/{var1+var2+2*cov(1,2)}**1/2*/
run;
data joinem1;
set temp7;
if j<1.645 then Reject_JT=0;
else Reject_JT=1;
run;
proc means;
var Reject_JT ;
run;



title1 "get mjt ";
data jo1;
set new1;
if group=1 or group=2;
run;
ods listing close;
ods output JTTest=sample_jts1;
proc freq data=jo1;
by sample;
tables group*(sum1 max1 min1)/ jt; /*compare jt by using sum
max min*/
run;
ods listing;
proc sort ;
    by sample table;
run;
data sample_jt1;
set sample_jts1;
stdjt=1*nvalue1;
where name1='_JT_';
run;
```

```
data jo2;
set new1;
if group=1 or group=3;
run;
ods listing close;
ods output JTTest=sample_jts2;
proc freq data=jo2;
by sample;
tables group*(sum1 max1 min1)/ jt; /*compare jt by using sum
max min*/
run;
ods listing;
proc sort ;
    by sample table;
run;
data sample_jt2;
set sample_jts2;
stdjt=2*nvalue1;
where name1='_JT_';
run;
data jo3;
set new1;
if group=2 or group=3;
run;
ods listing close;
ods output JTTest=sample_jts3;
proc freq data=jo3;
by sample;
tables group*(sum1 max1 min1)/ jt; /*compare jt by using sum
max min*/
run;
ods listing;
data sample_jt3;
set sample_jts3;
stdjt=1*nvalue1;
where name1='_JT_';
run;
proc sort;
 by sample table;run;
data jo4;
set new1;
if group=2 or group=4;
run;
ods listing close;
ods output JTTest=sample_jts4;
proc freq data=jo4;
by sample;
```

```
tables group*(sum1 max1 min1)/ jt; /*compare jt by using sum
max min*/
run;
ods listing;
data sample_jt4;
set sample_jts4;
stdjt=2*nvalue1;
where name1='_JT_';
run;
proc sort;
 by sample table;run;
data jo5;
set new1;
if group=1 or group=4;
run;
ods listing close;
ods output JTTest=sample_jts5;
proc freq data=jo5;
by sample;
tables group*(sum1 max1 min1)/ jt; /*compare jt by using sum
max min*/
run;
ods listing;
data sample_jt5;
set sample_jts5;
stdjt=3*nvalue1;
where name1='_JT_';
run;
proc sort;
 by sample table;run;
data jo6;
set new1;
if group=3 or group=4;
run;
ods listing close;
ods output JTTest=sample_jts6;
proc freq data=jo6;
by sample;
tables group*(sum1 max1 min1)/ jt; /*compare jt by using sum
max min*/
run;
ods listing;
data sample_jt6;
set sample_jts6;
stdjt=1*nvalue1;
where name1='_JT_';
run;
```

```
proc sort;
 by sample table;run;
 data jo7;
set new1;
if group=1 or group=5;
run;
ods listing close;
ods output JTTest=sample_jts7;
proc freq data=jo7;
by sample;
tables group*(sum1 max1 min1)/ jt; /*compare jt by using sum
max min*/
run;
ods listing;
data sample_jt7;
set sample_jts7;
stdjt=4*nvalue1;
where name1='_JT_';
run;
proc sort;
 by sample table;run;
 data jo8;
set new1;
if group=2 or group=5;
run;
ods listing close;
ods output JTTest=sample_jts8;
proc freq data=jo8;
by sample;
tables group*(sum1 max1 min1)/ jt; /*compare jt by using sum
max min*/
run;
ods listing;
data sample_jt8;
set sample_jts8;
stdjt=3*nvalue1;
where name1='_JT_';
run;
proc sort;
 by sample table;run;
 data jo9;
set new1;
if group=3 or group=5;
run;
ods listing close;
ods output JTTest=sample_jts9;
proc freq data=jo9;
```

```
by sample;
tables group*(sum1 max1 min1)/ jt; /*compare jt by using sum
max min*/
run;
ods listing;
data sample_jt9;
set sample_jts9;
stdjt=2*nvalue1;
where name1='_JT_';
run;
proc sort;
 by sample table;run;
 data jo10;
set new1;
if group=4 or group=5;
run;
ods listing close;
ods output JTTest=sample_jts10;
proc freq data=jo10;
by sample;
tables group*(sum1 max1 min1)/ jt; /*compare jt by using sum
max min*/
run;
ods listing;
data sample_jt10;
set sample_jts10;
stdjt=1*nvalue1;
where name1='_JT_';
run;
proc sort;
 by sample table;run;
data jt_s(keep=sample group stdjt table);
set sample_jt1 sample_jt2 sample_jt3 sample_jt4 sample_jt5
sample_jt6 sample_jt7 sample_jt8 sample_jt9 sample_jt10;
run;
proc means data=jt_s sum;
class sample;
var stdjt;
where table ?'sum';
output out=sumstat sum=sumstat1;
run;
 proc means data=jt_s sum;
class sample;
var stdjt;
where table ?'min';
output out=minstat sum=minstat1;
run;
```

```
proc means data=jt_s;
class sample;
var stdjt;
where table ?'max';
output out=maxstat sum=maxstat1;
run;
data stat;
merge sumstat minstat maxstat;
by sample;

Data jyo;
set stat;
n1=5;
n2=5;
n3=10;
n4=5;
n5=5;
emjt=1/2*(n1*n2 + 2*(n1*n3) + n2*n3 + n3*n4 +3*(n1*n4)
+2*(n2*n4)+ 4*(n1*n5)+3*(n2*n5)+2*(n3*n5)+ n4*n5);
var12=1/12*(n1*n2)*(n1+n2+1);
var13= 1/12*(n1*n3)*(n1+n3+1);
var23= 1/12*(n2*n3)*(n2+n3+1);
var24= 1/12*(n2*n4)*(n2+n4+1);
var34= 1/12*(n3*n4)*(n3+n4+1);
var14= 1/12*(n1*n4)*(n1+n4+1);
var15= 1/12*(n1*n5)*(n1+n5+1);
var25= 1/12*(n2*n5)*(n2+n5+1);
var35= 1/12*(n3*n5)*(n3+n5+1);
var45= 1/12*(n4*n5)*(n4+n5+1);
covij=1/6*(3*n1*n2*n3 +7* n1*n2*n4 +3*n2*n3*n4 +7*n1*n3*n4 +
13*n1*n2*n5 + 12*n1*n3*n5 +7*n2*n3*n5 + 3*n3*n4*n5 +
13*n1*n4*n5 + 7*n2*n4*n5);
var=(var12+4*var13+var23+4*var24+var34+9*var14+16*var15+9*va
r25+4*var35+var45);
variance=var+covij;
sqvar=sqrt(variance);
sumMjt=(sumstat1-emjt)/sqrt(variance);
minMjt=(minstat1-emjt)/sqrt(variance);
maxMjt=(maxstat1-emjt)/sqvar;
run;
proc print;
run;
title1 "get mjt (sum statistics)";
data sum;
set jyo;
if sumMjt<1.645 then Reject_JT=0;
else Reject_JT=1;
```

```
run;
proc means;
var Reject_JT ;
run;
title1 "get mjt (min statistics)";
data sum;
set jyo;
if minMjt<1.645 then Reject_JT=0;
else Reject_JT=1;
run;
proc means;
var Reject_JT ;
run;
title1 "get mjt (max statistics)";
data sum;
set jyo;
if maxMjt<1.645 then Reject_JT=0;
else Reject_JT=1;
run;
proc means;
var Reject_JT ;
run;
```