

T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
İSTATİSTİK ANA BİLİM DALI

YÜKSEK LİSANS TEZİ

ÇOK DEĞİŞKENLİ ÇOK ÖRNEKLİ
PARAMETRİK OLMAYAN KONUM TESTİ

HAZIRLAYAN

Öznur Aydın

068215001005

DANIŞMAN

Yrd. Doç. Dr. Mustafa SEMİZ

KONYA, 2008

ÖZET

Yüksek Lisans Tezi

ÇOK DEĞİŞKENLİ ÇOK ÖRNEKLİ PARAMETRİK OLMAYAN KONUM TESTİ

Öznur AYDINER

Selçuk Üniversitesi

Fen Bilimleri Enstitüsü

İstatistik Ana Bilim Dalı

Danışman: Yrd. Doç. Dr. Mustafa SEMİZ

2008, 86 sayfa

Çok değişkenli çok örnekli parametrik konum testinin varsayımlarının sağlanmadığı durumda Lawley ve Hotelling tarafından önerilen parametrik olmayan yönteme ilişkin hipotez testi programı yazılmıştır. Parametrik olmayan yöntemin gücü ve parametrik yöntemle karşılaştırılması simülasyon çalışmasıyla gösterilmiştir. Konumları karşılaştırılan dağılımların örnek hacminin ve dağılımların benzerliğinin yöntemlerin etkinliğinde önemli rol oynadığı belirlenmiştir.

Anahtar Kelimeler: Çok değişkenli ortalama vektörlerinin parametrik testi, Çok değişkenli konum vektörlerinin parametrik olmayan testi, Çok değişkenli dağılımlar, Çok değişkenli normallik testi, Kovaryans matrisi, Matrislerin eşitliğinin testi.

ABSTRACT**Ms Thesis****THE MULTIVARIATE MULTI-SAMPLE
NONPARAMETRIC LOCATION TEST****Öznur AYDINER****Selçuk University
Graduate School of Natural and Applied Sciences
Department of Statistics****Supervisor: Assist. Prof. Dr. Mustafa SEMİZ****2008, 86 page**

When the assumptions of multivariate multisample parametric location test is not satisfied, hypothesis test that is related to nonparametric methods is suggested by Lawley and Hotelling and its program has been written. The power of non-parametric method and the comparison of nonparametric method with parametric method has been shown with the simulation study. It has been determined that sample volume and similarity of distributions that the locations are compared acts an important role in the effectiveness of methods.

Key Words: Parametric test of multivariate mean vectors, Nonparametric test of multivariate location vectors, Multivariate distributions, Multivariate Normality Test, Covariance Matrix, Test of equality matrices.

ÖNSÖZ

Bu tez çalışmasının konu seçiminde, hazırlanma ve bitme aşamalarında değerli zamanını harcayan ve her türlü yardımı gösteren kıymetli hocam Yrd. Doç. Dr. Mustafa SEMİZ'e çok teşekkür ederim. Ayrıca çalışmalarım sırasında benden yardımlarını esirgemeyen Yrd. Doç. Dr. M. Fedai KAYA, Arş. Gör. Ahmet PEKGÖR ve diğer tüm bölüm hocalarıma; bana manevi destek veren sevgili dostlarıma, aileme ve arkadaşlarıma teşekkürlerimi sunarım.

İÇİNDEKİLER

<u>ÖZET</u>	i
<u>ABSTRACT</u>	ii
<u>ÖNSÖZ</u>	iii
<u>İÇİNDEKİLER</u>	iv
<u>1. GİRİŞ</u>	i
<u>2. ÇOK DEĞİŞKENLİ PARAMETRİK KONUM TESTİ</u>	2
<u>2.1. Çok Değişkenli İstatistiksel Analize Giriş</u>	2
<u>2.1.1. Veri matrisi</u>	2
<u>2.1.2. Temel kavramlar</u>	3
<u>2.2. Çok Değişkenli Hipotez Testleri</u>	4
<u>2.2.1. Giriş</u>	4
<u>2.2.2. Varsayımların testi</u>	5
<u>2.2.3. Çok değişkenli tek yığın ortalamasının testi</u>	7
<u>2.2.4. Bağımsız İki Yığına İlişkin Ortalama Vektörlerinin Karşılaştırılması</u>	10
<u>2.2.5. Çok değişkenli tek yönlü varyans analizi (MANOVA)</u>	13
<u>3. ÇOK DEĞİŞKENLİ PARAMETRİK OLMAYAN KONUM TESTİ</u>	19
<u>3.1. Giriş</u>	19
<u>3.2. Çok Değişkenli Çok Örnekli Ortalama ve Varyans Problemlerinin Formülasyonu</u>	19
<u>3.3. Temel Rank Permütasyon Prensibi</u>	21
<u>3.4. Bazı Permütasyon Sıralı Rank Testleri</u>	23
<u>3.4.1. Çok değişkenli çok örnekli medyan testi</u>	26
<u>3.4.2. Rank toplamı testi</u>	27
<u>3.4.3. Normal skorlar testi</u>	27
<u>3.5. L_N'nin Asimtotik Permütasyon Dağılımı</u>	28
<u>4. UYGULAMA</u>	34
<u>4.1. Ortalama Vektörlerinin Parametrik Yöntemle Test Edilmesi</u>	34
<u>4.2. Konum Vektörlerinin Parametrik Olmayan Yöntemle Test Edilmesi</u>	36

<u>5. SİMÜLASYON ÇALIŞMASI</u>	40
<u>6. SONUÇ</u>	44
<u>7. EKLER</u>	45
<u>EK 7.1. Çok Değişkenli Normallik Testi SPSS-MP Program Kodu</u>	45
<u>EK 7.2. Kovaryans Matrislerinin Eşitliği Testi SPSS-MP Program Kodu</u>	47
<u>EK 7.3. Çok Değişkenli Tek Yığın Ortalaması Önemlilik Testi ve Farklı Değişken(ler)in Belirlenmesi için SPSS-MP Program Kodu</u>	49
<u>EK 7.4. Bağımsız İki Grubun Ortalama Vektörlerinin Karşılaştırılmasına İlişkin Parametrik Hipotez Testi ve Farklı Değişken(ler)in Belirlenmesi için SPSS-MP Program Kodu- Kovaryans Matrisleri Eşit Olduğunda</u>	51
<u>EK 7.5. Bağımsız İki Grubun Ortalama Vektörlerinin Karşılaştırılmasına İlişkin Parametrik Hipotez Testi ve Farklı Değişken(ler)in Belirlenmesi için SPSS-MP Program Kodu- Kovaryans Matrisleri Eşit Olmadığında</u>	53
<u>EK 7.6. Bağımsız k Grubun Ortalama Vektörlerinin Karşılaştırılmasına İlişkin Parametrik Hipotez Testi SPSS-MP Program Kodu</u>	55
<u>EK 7.7. k Grubun Ortalama Vektörlerinin Karşılaştırılmasına İlişkin Parametrik Hipotez Testi SPSS-MP Program Kodu</u>	57
<u>EK 7.8. k Grubun Ortalama Vektörlerinin Karşılaştırılmasına İlişkin Parametrik Olmayan Hipotez Testi SPSS-MP Program Kodu</u>	58
<u>EK 7.9. Normal Dağılımlı İki Grubun Konum Testi için Parametrik ve Parametrik Olmayan Simülasyon</u>	60
<u>EK 7.10. Normal Dağılımlı İki Grubun Konum Testi için Parametrik ve Parametrik Olmayan Simülasyon</u>	63
<u>EK 7.11. Normal ve Ki-kare Dağılımlarına Sahip İki Grubun Konum Testi için Parametrik ve Parametrik Olmayan Simülasyon</u>	66
<u>EK 7.12. Normal ve Üstel Dağılımlarına Sahip İki Grubun Konum Testi için Parametrik ve Parametrik Olmayan Simülasyon</u>	69
<u>EK 7.13. Ki-kare Dağılımına Sahip İki Grubun Konum Testi için Parametrik ve Parametrik Olmayan Simülasyon</u>	72
<u>EK 7.14. Ki-kare ve Üstel Dağılımlarına Sahip İki Grubun Konum Testi için Parametrik ve Parametrik Olmayan Simülasyon</u>	75
<u>EK 7.15. Üstel Dağılıma Sahip İki Grubun Konum Testi için Parametrik ve Parametrik Olmayan Simülasyon</u>	78
<u>KAYNAKLAR</u>	81

1. GİRİŞ

Gerek parametrik gerekse parametrik olmayan testlerin örnek ile ilgili iki önemli varsayımı; gözlemlerin yığından rasgele ve birbirlerinden bağımsız olarak seçilmesi varsayımlarıdır. Parametrik hipotez testlerinin örneğin çekildiği yığın ile ilgili iki temel varsayımı ise; normallik ve varyansların homojenliği varsayımlarıdır. Çok değişkenli hipotez testlerinde, bu varsayımların çok değişken için genelleştirilmiş durumu olan; çok değişkenli normallik ve kovaryans matrislerinin eşitliği varsayımları söz konusudur (Alpar 2003).

Bu tez çalışmasında, çok değişkenli k grup ortalamasının karşılaştırılması için parametrik ve parametrik olmayan test yöntemleri ele alınmıştır. Bu kapsamda Bölüm 2’de tek grup, iki grup ve k grup ortalamalarını karşılaştırmada kullanılan parametrik yöntemler ve bu yöntemler için gerekli varsayımların test edilmesi ele alınmıştır. Ayrıca bu bölümde, ortalama vektörlerinin farklı bulunması halinde farklılığın hangi değişken ya da değişkenlerden kaynaklandığının bulunması da işlenmiştir.

Bölüm 3’te; k grup ortalamasının testi için normallik veya kovaryans matrislerinin eşitliğinin sağlanmaması, ya da gözlem sayısının az olması durumunda başvurulan bir diğer yöntem olan k grup ortalamasının testi için kullanılan parametrik olmayan yöntem anlatılmıştır.

Bölüm 4’te küçük bir veri seti üzerinde her iki yöntemin ayrıntılı hesaplamaları gösterilmiştir.

Bu iki yöntemin karşılaştırılması ve parametrik olmayan yöntemin etkinliğinin incelenmesi için yapılan farklı simülasyon çalışmaları Bölüm 5’te sunulmuştur.

Bu tezin ekinde, varsayımların testlerini, parametrik yöntemin testini ve parametrik olmayan yöntemin testini yapan ve farklı durumların testi için yazılan simülasyon programlarının SPSS-MP kodları yazılmıştır.

2. ÇOK DEĞİŞKENLİ PARAMETRİK KONUM TESTİ

2.1. Çok Değişkenli İstatistiksel Analize Giriş

2.1.1. Veri matrisi

Çok değişkenli istatistiksel analizde genellikle **n** satır (**gözlem**) ve **p** sütundan (**değişken**) oluşan $n \times p$ boyutlu X veri matrisi ile ilgilenilir (Alpar 2003). X_1, X_2, \dots, X_p değişkenleri ve $x_{11}, x_{12}, \dots, x_{np}$ ise gözlem değerlerini gösteriyor.

Gözlem vektörü olarak adlandırılan $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$; $i=1,2,\dots,n$ vektörü, herhangi bir birey ya da nesnenin p tane özelliğine ilişkin değerleri göstermektedir. Bu durumda $x'_j = (x_{1j}, x_{2j}, \dots, x_{nj})$; $j=1,2,\dots,p$ olmak üzere tüm birey ya da nesnelerin herhangi bir özelliğini gösteren x_j vektörüne de **değişken vektörü** adı verilir.

Yukarıda tanımlanan gözlem vektöründen n tanesinin alt alta ya da değişken vektöründen p tanesinin yan yana yazılmasından örnek matrisi yani veri matrisi oluşur (Tatlıdil 2002).

Tablo1. X Veri Matrisi

	p adet değişken			
Gözlem numarası	X_1	X_2	...	X_p
1	x_{11}	x_{12}	...	x_{1p}
2	x_{21}	x_{22}	...	x_{2p}
.
.
.
.
.
N	x_{n1}	x_{n2}	...	x_{np}

2.1.2. Temel kavramlar

Çok değişken olması durumunda, ilk temel kavram olan ortalama artık tek bir değer değil, bir vektör olmaktadır. Diğer bir temel kavram olan varyans yerine ise, $p \times p$ boyutlu simetrik ve pozitif tanımlı bir matris olan kovaryans matrisi kullanılır.

p tesadüfi değişkenin ortalama vektörü;

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix}$$

p tesadüfi değişkenin kovaryans matrisi;

$$\Sigma = \begin{bmatrix} \frac{1}{N} \sum_{i=1}^N (x_{i1} - \mu_1)^2 & \frac{1}{N} \sum_{i=1}^N (x_{i1} - \mu_1)(x_{i2} - \mu_2) & \cdots & \frac{1}{N} \sum_{i=1}^N (x_{i1} - \mu_1)(x_{ip} - \mu_p) \\ \frac{1}{N} \sum_{i=1}^N (x_{i2} - \mu_2)(x_{i1} - \mu_1) & \frac{1}{N} \sum_{i=1}^N (x_{i2} - \mu_2)^2 & \cdots & \frac{1}{N} \sum_{i=1}^N (x_{i2} - \mu_2)(x_{ip} - \mu_p) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{N} \sum_{i=1}^N (x_{ip} - \mu_p)(x_{i1} - \mu_1) & \frac{1}{N} \sum_{i=1}^N (x_{ip} - \mu_p)(x_{i2} - \mu_2) & \cdots & \frac{1}{N} \sum_{i=1}^N (x_{ip} - \mu_p)^2 \end{bmatrix}$$

p tesadüfi değişkenin örnek değerlerinin ortalama vektörü;

$$\bar{x} = \begin{bmatrix} \frac{1}{n} \sum_{i=1}^n x_{i1} \\ \frac{1}{n} \sum_{i=1}^n x_{i2} \\ \vdots \\ \frac{1}{n} \sum_{i=1}^n x_{ip} \end{bmatrix} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_p \end{bmatrix}$$

p tesadüfi değişkenin örnek değerlerinin kovaryans matrisi;

$$S = \begin{bmatrix} \frac{1}{n-1} \sum_{i=1}^n (x_{i1} - \bar{x}_1)^2 & \frac{1}{n-1} \sum_{i=1}^n (x_{i1} - \bar{x}_1)(x_{i2} - \bar{x}_2) & \cdots & \frac{1}{n-1} \sum_{i=1}^n (x_{i1} - \bar{x}_1)(x_{ip} - \bar{x}_p) \\ \frac{1}{n-1} \sum_{i=1}^n (x_{i2} - \bar{x}_2)(x_{i1} - \bar{x}_1) & \frac{1}{n-1} \sum_{i=1}^n (x_{i2} - \bar{x}_2)^2 & \cdots & \frac{1}{n-1} \sum_{i=1}^n (x_{i2} - \bar{x}_2)(x_{ip} - \bar{x}_p) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n-1} \sum_{i=1}^n (x_{ip} - \bar{x}_p)(x_{i1} - \bar{x}_1) & \frac{1}{n-1} \sum_{i=1}^n (x_{ip} - \bar{x}_p)(x_{i2} - \bar{x}_2) & \cdots & \frac{1}{n-1} \sum_{i=1}^n (x_{ip} - \bar{x}_p)^2 \end{bmatrix}$$

2.2. Çok Değişkenli Hipotez Testleri

2.2.1. Giriş

Çok değişken olması durumunda hipotez testleri, araştırmacıyı karar verme konusunda sıkıntıya düşüren karmaşık bir konudur. Her şeyden önce kurulabilecek hipotez sayısının ve kullanılacak test sayısının çok olması karar vermeyi güçleştirmektedir.

Çok değişken olması durumunda testlerin ağırlığı, ortalama vektörüne ilişkindir. Kovaryans matrislerinin eşitliği, korelasyon matrislerinin eşitliği ya da birimliği üzerine kurulu testler pratikte çok kullanılmadığı gibi birçoğunun testi için gerekli kritik değerler henüz geliştirilmediğinden yaklaşık test olmaktan öteye gidememektedirler (Tatlıdil 2002).

Değişkenler birer birer tek değişkenli test ile test edilirse, yani ne kadar çok test uygulanırsa, değişkenler arasında fark yok iken en az bir tane önemli sonuç bulma olasılığı o kadar artacaktır. Örneğin, 10 tane hipotez için $1-0.95^{10}=0.40$ olarak bulunacaktır. Bu olasılığa tümel I. tip hata da denir. Bu kadarlık bir hata ile sonuçlara ulaşmak oldukça yanıltıcı olacaktır. Halbuki T^2 testi gibi çok değişkenli bir testte, 0.05 yanılma düzeyi kullanıldığında, 0.05 olan I. tip hata, testin içerdiği değişken sayısı ile ilişkili değildir. Bu da özellikle değişken sayısı fazla olduğunda, tek değişkenli testlere göre bir avantaj sağlar.

Birden çok değişken açısından gruplar arasında fark olup olmadığı tek değişkenli testlerle araştırıldığında, değişkenler açısından gruplar arasında anlamlı bir fark çıkmayabilir. Ancak, çok değişkenli testlerle birleşik değişkenler kümesi arasında fark arandığından gruplar arasında fark çıkabilir.

Tek değişkenli testler, değişkenler arasındaki ilişkinin bir göstergesi olan kovaryansı (ya da korelasyonu) dikkate almadıklarından, değişkenler arasında var olan bilginin daha az bir bölümünü kullanırlar (Alpar 2003).

2.2.2. Varsayımların testi

2.2.2.1. İki ve daha fazla değişken için normallik testleri

2.2.2.1.1. Q-Q grafik yöntemi

a) Önce $m_i^2 = (x_i - \bar{x})'S^{-1}(x_i - \bar{x})$, $i=1,2,\dots,n$ değerleri hesaplanır.

b) Hesaplanan m_i^2 (Mahalanobis uzaklığının karesi) değerleri küçükten büyüğe doğru sıralanır.

c) Her bir gözlem için $p_i = \frac{i-1/2}{n}$ olasılığı elde edilir.

d) p_i olasılıklarına karşı gelen $\chi_{p;(i-0.5)/n}^2$ ki-kare yüzdeleri bulunur.

e) x eksenine m_i^2 , y eksenine χ_i^2 değerleri gelecek şekilde saçılım grafiği çizilir.

Noktaların bir doğru üzerinde olması durumunda değişkenlerin çok değişkenli normal dağıldığı söylenir (Alpar 2003).

2.2.2.1.2. Ortanca yöntemi

Bir diğer yöntem olarak χ^2 dağılımından yararlanılabilir. Buna göre m_i^2 (Mahalanobis uzaklığının karesi) değerlerinin yarısından fazlası $\chi_{p;0.5}^2$ değerinden büyükse, yani % 50'lik konturun dışında kalıyorsa dağılımın normal dağılmadığı söylenir (Alpar 2003).

2.2.2.2. Kovaryans matrislerinin eşitliği varsayımı

Çok değişkenli analizlerde kovaryans matrislerinin eşitliği testi, Box-M testi olarak bilinir. Box-M testi, normallikten ayrılıklara karşı oldukça duyarlıdır. Bu nedenle, verilerde normallik varsayımının sağlanmış olması gerekir. Kovaryans matrislerinin eşitliğine ilişkin hipotezler aşağıdaki gibidir:

$$H_0 : \Sigma_1 = \Sigma_2 = \dots = \Sigma_k$$

H_1 : En az bir kovaryans matrisi diğerlerinden farklıdır.

k: Karşılaştırılacak kovaryans matrislerin (grup) sayısı

p: Kovaryans matrislerindeki (gruplardaki) değişken sayısı

S_i : Kovaryans matrisleri ($i=1,2,\dots,k$)

n_i : Gözlem sayısı

olmak üzere ortak kovaryans matrisi; $S = \frac{\sum_{i=1}^k (n_i - 1) S_i}{\sum_{i=1}^k (n_i - 1)}$ ile verilir. Gruplardaki

gözlem sayıları eşit ise $S = \sum_{i=1}^k S_i / k$ olur. İlgili test istatistiği (Box-M);

$$M = \left(\sum_{i=1}^k (n_i - 1) \right) \ln |S| - \sum_{i=1}^k \left((n_i - 1) \ln |S_i| \right)$$
 şeklinde tanımlanır. n_i 'ler yeterince

büyükken;

$$C^{-1} = 1 - \frac{2p^2 + 3p - 1}{6(p+1)(k-1)} \left(\sum_{i=1}^k \frac{1}{(n_i - 1)} - \frac{1}{\sum_{i=1}^k (n_i - 1)} \right)$$

olmak üzere MC^{-1} değeri $(1/2)(k-1)p(p+1)$ serbestlik derecesi ile yaklaşık ki-kare dağılımı gösterir. MC^{-1} değerinin seçilen yanılma düzeyinde, $(1/2)(k-1)p(p+1)$ serbestlik dereceli χ^2 tablo değerinden büyük olması durumunda kovaryans matrislerinin eşit olmadığı söylenir. Gözlem sayıları eşit ise, n herhangi bir grubun gözlem sayısı olmak üzere C^{-1} ;

$$C^{-1} = 1 - \frac{(2p^2 + 3p - 1)(k+1)}{6(p+1)k(n-1)}$$
 ile verilir (Alpar 2003).

2.2.3. Çok deęişkenli tek yığın ortalamasının testi

Çok deęişkenli yığın ortalamasının testinde; birden çok deęişkene ilişkin örnek ortalamasını içeren ortalama vektörü, yığın ortalama vektörü ile karşılaştırılır. H_0 ve H_1 hipotezleri;

$$H_0 : \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix} = \begin{bmatrix} \mu_{01} \\ \mu_{02} \\ \vdots \\ \mu_{0p} \end{bmatrix} \quad \text{ve} \quad H_1 : \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix} \neq \begin{bmatrix} \mu_{01} \\ \mu_{02} \\ \vdots \\ \mu_{0p} \end{bmatrix}$$

şeklinde kurulur (Alpar 2003).

2.2.3.1. Varsayımlar

Bu testin varsayımı, verilerin çok deęişkenli normal dağılım göstermesidir. Gözlem sayısı fazla olduğunda bu varsayım göz ardı edilebilir (Alpar 2003).

Normallik varsayımının sağlanıp sağlanmadığına Bölüm 2.2.2.1'deki gibi bakılır.

2.2.3.2. Test istatistięi

$$H_0 : \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix} = \begin{bmatrix} \mu_{01} \\ \mu_{02} \\ \vdots \\ \mu_{0p} \end{bmatrix} \quad \text{hipotezinin testi için çok deęişkenli } T^2 \text{ istatistięi elde}$$

edildikten sonra F istatistięine dönüşüm yapılır. T^2 ; $T^2 = n(\bar{x} - \mu_0)'S^{-1}(\bar{x} - \mu_0)$

olarak tanımlanır. $F = \frac{(n-p)}{(n-1)p} T^2$ olmak üzere p ve $n-p$ serbestlik dereceli bir F

dağılımı gösterir. Bulunan bu F hesap deęeri, p ve $n-p$ serbestlik dereceli F tablo deęeri ile karşılaştırılır. F tablo deęeri; $F_T = F_{p,n-p;\alpha}$ olmak üzere F hesap deęeri, F tablo deęerinden büyük ise H_1 hipotezi kabul edilir ve “En az bir deęişkenin ortalaması, ilgili yığın ortalamasından farklıdır.” denir (Alpar 2003).

2.2.3.3. Güven aralıkları ve farklı değişkenlerin belirlenmesi

H_0 hipotezi reddedildiğinde farklılığın hangi değişken(ler)den kaynaklandığının araştırılması gerekir. Bu amaçla güven aralıklarından yararlanır. Güven aralıkları, hem farklı ya da farksız değişkenlerin belirlenmesini; hem de incelenen değişkene ilişkin çok değişkenli güven aralıklarının bulunmasını sağlar. Her bir değişken için bulunan sınırlar ilgili yığın ortalamasını içermiyor ise farklılığın ilgili değişkenden kaynaklandığı söylenir.

Aşağıda, bu amaçla geliştirilmiş üç farklı güven aralığı yöntemi verilmiştir (Alpar 2003).

2.2.3.3.1. Eşanlı güven aralıkları yöntemi

a' ; incelenin değişken için 1, diğerleri için 0 yazılan hipotez vektörü ve

$$T_{(p,n-p;\alpha)}^2 = \frac{(n-1)p}{(n-p)} F_{(p,n-p;\alpha)} \text{ olmak üzere,}$$

$$a'\bar{x} - \sqrt{\frac{1}{n} a' Sa T_{(p,n-p;\alpha)}^2} \leq a'\mu \leq a'\bar{x} + \sqrt{\frac{1}{n} a' Sa T_{(p,n-p;\alpha)}^2}$$

aralıkları her bir değişken için aşağıdaki gibi bulunur.

Güven aralıkları, her bir değişken için daha kısa yolla şöyle de hesaplanabilir:

$$\begin{aligned} \bar{x}_1 - \sqrt{\frac{(n-1)p}{n-p} F_{p,n-p;\alpha}} \sqrt{\frac{s_{11}}{n}} &\leq \mu_1 \leq \bar{x}_1 + \sqrt{\frac{(n-1)p}{n-p} F_{p,n-p;\alpha}} \sqrt{\frac{s_{11}}{n}} \\ \bar{x}_2 - \sqrt{\frac{(n-1)p}{n-p} F_{p,n-p;\alpha}} \sqrt{\frac{s_{22}}{n}} &\leq \mu_2 \leq \bar{x}_2 + \sqrt{\frac{(n-1)p}{n-p} F_{p,n-p;\alpha}} \sqrt{\frac{s_{22}}{n}} \\ &\vdots \\ \bar{x}_p - \sqrt{\frac{(n-1)p}{n-p} F_{p,n-p;\alpha}} \sqrt{\frac{s_{pp}}{n}} &\leq \mu_p \leq \bar{x}_p + \sqrt{\frac{(n-1)p}{n-p} F_{p,n-p;\alpha}} \sqrt{\frac{s_{pp}}{n}} \end{aligned}$$

2.2.3.3.2. Bonferroni yöntemi

Bu yönteme göre ise güven aralıkları,

$$\begin{aligned} \bar{x}_1 - t_{(n-1; \frac{\alpha}{2p})} \sqrt{\frac{S_{11}}{n}} &\leq \mu_1 \leq \bar{x}_1 + t_{(n-1; \frac{\alpha}{2p})} \sqrt{\frac{S_{11}}{n}} \\ \bar{x}_2 - t_{(n-1; \frac{\alpha}{2p})} \sqrt{\frac{S_{22}}{n}} &\leq \mu_2 \leq \bar{x}_2 + t_{(n-1; \frac{\alpha}{2p})} \sqrt{\frac{S_{22}}{n}} \\ &\cdot \\ &\cdot \\ \bar{x}_p - t_{(n-1; \frac{\alpha}{2p})} \sqrt{\frac{S_{pp}}{n}} &\leq \mu_p \leq \bar{x}_p + t_{(n-1; \frac{\alpha}{2p})} \sqrt{\frac{S_{pp}}{n}} \end{aligned}$$

şeklindedir. Bonferroni güven aralıkları, eşanlı güven aralıklarına göre daha dardır (Alpar 2003).

2.2.3.3.3. Gözlem sayısı fazla olduğunda farklı değişkenlerin belirlenmesi

Gözlem sayısı fazla olduğunda μ için güven bölgesi ve hipotez testleri normallik varsayımı olmadan da oluşturulabilir. n yeterince büyükken, eşanlı güven sınırları χ^2 dağılımından yararlanılarak,

$$\begin{aligned} \bar{x}_1 - \sqrt{\chi_{p;\alpha}^2} \sqrt{\frac{S_{11}}{n}} &\leq \mu_1 \leq \bar{x}_1 + \sqrt{\chi_{p;\alpha}^2} \sqrt{\frac{S_{11}}{n}} \\ \bar{x}_2 - \sqrt{\chi_{p;\alpha}^2} \sqrt{\frac{S_{22}}{n}} &\leq \mu_2 \leq \bar{x}_2 + \sqrt{\chi_{p;\alpha}^2} \sqrt{\frac{S_{22}}{n}} \\ &\cdot \\ &\cdot \\ \bar{x}_p - \sqrt{\chi_{p;\alpha}^2} \sqrt{\frac{S_{pp}}{n}} &\leq \mu_p \leq \bar{x}_p + \sqrt{\chi_{p;\alpha}^2} \sqrt{\frac{S_{pp}}{n}} \end{aligned}$$

ile verilir. Bulunacak sınırlara dayalı olarak yapılacak karşılaştırmalar da diğer iki yöntemdeki gibi yorumlanır (Alpar 2003).

2.2.4. Bağımsız İki Yığına İlişkin Ortalama Vektörlerinin Karşılaştırılması

Bağımsız iki gruba ilişkin ortalama vektörlerinin karşılaştırılmasında da T^2 (Hotelling's T^2) istatistiğinden yararlanır. Bu yöntem, çok değişkenli ikiden çok grubun karşılaştırılmasında kullanılan MANOVA yaklaşımının sadece iki grup olduğundaki özel bir durumudur. T^2 yaklaşımı, değişkenlerin ikişer ikişer t testi ile karşılaştırılması sonucunda I. tip hatada meydana gelecek artışı sabit tutar (Alpar 2003).

H_0 hipotezi; μ_1 ve μ_2 ortalama vektörleri olmak üzere, bu iki ortalama vektörü arasında fark olmadığı şeklinde kurulur ($H_0 : \mu_1 = \mu_2$). H_1 hipotezi ise $H_1 : \mu_1 \neq \mu_2$ şeklindedir.

$$H_0 : \begin{bmatrix} \mu_{11} \\ \mu_{12} \\ \vdots \\ \mu_{1p} \end{bmatrix} = \begin{bmatrix} \mu_{21} \\ \mu_{22} \\ \vdots \\ \mu_{2p} \end{bmatrix} \text{ hipotezi, } H_1 : \begin{bmatrix} \mu_{11} \\ \mu_{12} \\ \vdots \\ \mu_{1p} \end{bmatrix} \neq \begin{bmatrix} \mu_{21} \\ \mu_{22} \\ \vdots \\ \mu_{2p} \end{bmatrix} \text{ hipotezine karşı test edilir.}$$

2.2.4.1. Varsayımlar

Çok değişkenli iki ortalama vektörünün karşılaştırılmasında, her iki gruptaki verilerin ayrı ayrı normal dağılım gösterdiği ve gruplara ilişkin kovaryans matrislerinin eşit olup olmadıkları bilinmelidir. Bazı kaynaklarda, gruptaki gözlem sayısının 20'den fazla olması özellikle belirtilmektedir. Dikkat edilecek bir diğer özellik ise, bir gruptaki gözlem sayısının değişken sayısından az olmamasıdır. Bazı yazarlara göre ise, gruptaki en az gözlem sayısı, gruptaki değişken sayısının en az üç katı kadar olmalıdır (Alpar 2003).

Normallik varsayımının sağlanıp sağlanmadığına her grup için de ayrı ayrı Bölüm 2.2.2.1'deki gibi bakılır.

Kovaryans matrislerinin eşitliği varsayımının sağlanıp sağlanmadığına ise Bölüm 2.2.2.2'deki gibi bakılır.

2.2.4.2. Test istatistiđi

n_1 : 1. grubun gözlem sayısı,

n_2 : 2. grubun gözlem sayısı,

\bar{x}_1 : 1. grubun ortalama vektörü,

\bar{x}_2 : 2. grubun ortalama vektörü ve

S : Ortak kovaryans matrisidir.

Kovaryans matrisleri eşit ise;

$$S = \frac{(n_1 - 1)S_1 + (n_2 - 1)S_2}{n_1 + n_2 - 2} \text{ olmak üzere } T^2 \text{ istatistiđi};$$

$$T^2 = \frac{n_1 n_2}{n_1 + n_2} (\bar{x}_1 - \bar{x}_2)' S^{-1} (\bar{x}_1 - \bar{x}_2)$$

Şeklinde tanımlanır ve

$$F = \frac{n_1 + n_2 - p - 1}{(n_1 + n_2 - 2)p} T^2$$

ile p ve $n_1 + n_2 - p - 1$ serbestlik dereceli bir F dağılımı gösterir. Bulunan bu F hesap değeri, α anlam düzeyinde p ve $n_1 + n_2 - p - 1$ serbestlik dereceli F tablo değeri ile karşılaştırılır. F hesap istatistiđi F tablo istatistiđinden büyükse bağımsız iki ortalama vektörü arasında fark olduđu söylenir (Alpar 2003 ve Tuncer 2002).

Kovaryans matrisleri farklı ise;

$$S = \frac{S_1}{n_1} + \frac{S_2}{n_2} \text{ olmak üzere } T^2 = (\bar{x}_1 - \bar{x}_2)' S^{-1} (\bar{x}_1 - \bar{x}_2) \text{ istatistiđinin asimptotik}$$

dağılımı $\chi^2_{p,1-\alpha}$ olur. T^2 istatistiđi, $\chi^2_{p,1-\alpha}$ değerinden büyükse bağımsız iki ortalama vektörü arasında fark olduđu söylenir (Semiz 2007).

2.2.4.3. Güven aralıkları ve farklı değişkenlerin belirlenmesi

H_0 hipotezi reddedildiğinde, yani iki ortalama vektörü arasında fark olduğunda; farklılığın hangi değişken(ler)den kaynaklandığını belirlemek için aşağıdaki iki yöntemden yararlanır.

2.2.4.3.1. Kovaryans matrisleri eşit olduğunda güven aralıkları

2.2.4.3.1.1 Eşanlı güven aralıkları yöntemi

a' ; incelenin değişken için 1, diğerleri için 0 yazılan hipotez vektörü ve

$$T_{(p, n_1+n_2-p-1; \alpha)}^2 = \frac{(n_1 + n_2 - 2)p}{n_1 + n_2 - p - 1} F_{(p, n_1+n_2-p-1; \alpha)} \quad \text{olmak üzere, her bir değişkenin}$$

ortalamalarının farkına ilişkin eşanlı güven aralıkları;

$$a'(\bar{x}_1 - \bar{x}_2) - \sqrt{\frac{n_1 + n_2}{n_1 n_2} a' S a T_{(p, n_1+n_2-p-1; \alpha)}^2} \leq a'(\mu_1 - \mu_2) \leq a'(\bar{x}_1 - \bar{x}_2) + \sqrt{\frac{n_1 + n_2}{n_1 n_2} a' S a T_{(p, n_1+n_2-p-1; \alpha)}^2}$$

şeklinde verilir. Buradan elde edilen aralıkların sıfırı içermesi durumunda, ilgili değişken açısından gruplar arasında fark yoktur denir (Alpar 2003).

2.2.4.3.1.2. Bonferroni güven aralıkları yöntemi

Ortalamaların farkına ilişkin Bonferroni eşanlı güven aralıkları ise;

$$(\bar{x}_{i1} - \bar{x}_{i2}) - t_{(n_1+n_2-2; \frac{\alpha}{2p})} \sqrt{\frac{n_1 + n_2}{n_1 n_2} S_{ii}} \leq \mu_{i1} - \mu_{i2} \leq (\bar{x}_{i1} - \bar{x}_{i2}) + t_{(n_1+n_2-2; \frac{\alpha}{2p})} \sqrt{\frac{n_1 + n_2}{n_1 n_2} S_{ii}}$$

şeklinde verilir (Alpar 2003).

2.2.4.3.2. Kovaryans matrisleri eşit olmadığında güven aralıkları

Kovaryans matrisleri eşit değil ise ortalamaların farklarının güven aralıkları,

$$(\bar{x}_{i1} - \bar{x}_{i2}) - \sqrt{\chi_{p,1-\alpha}^2 S_{ii}} \leq \mu_{i1} - \mu_{i2} \leq (\bar{x}_{i1} - \bar{x}_{i2}) + \sqrt{\chi_{p,1-\alpha}^2 S_{ii}} \quad \text{şeklinde verilir (Semiz 2007).}$$

2.2.5. Çok deęişkenli tek yönlü varyans analizi (MANOVA)

Tek yönlü varyans analizi (ANOVA), parametrik test varsayımlarının (normallik ve varyansların homojenliği) yerine geldiđi durumlarda, bir deęişken açısından 2'den çok grup arasında fark olup olmadığını test etmekte kullanılıyordu. Çok deęişkenli tek yönlü varyans analizi (MANOVA) ise, her bir grupta (grup sayısı ≥ 3) 2 ve daha fazla deęişken olması durumunda uygulanır. Diđer bir deyişle ikiden çok grubun ortalama vektörleri karşılaştırılır.

Bir açıdan T^2 ve MANOVA, incelenen deęişkenlerin en iyi doğrusal kombinasyonlarına dayalı yeni bir bağımlı deęişken oluşturulması ve bu yeni bağımlı deęişken yönünden grupların ANOVA ile karşılaştırılmasıdır. Diđer bir yaklaşımla, grupların tümel olarak karşılaştırılması söz konusudur.

Özellikle incelenen deęişkenler arasında çoklu bağlantı olduğunda, ayrı ayrı t testi veya varyans analizi yapmak yerine MANOVA'dan yararlanmak daha güçlü sonuç elde edilmesini sağlar. Dolayısıyla, araştırmacı I. tip hatayı denetlemek istiyor ve bağımlı deęişkenler arasında ilişki söz konusu ise MANOVA iyi bir seçim olmaktadır.

T^2 ve MANOVA'da, seçilen bağımlı deęişkenlerin araştırma düzenine uygun olması ve bu deęişkenlerin her birinin ilgilenilen sorunun farklı bir boyutunu ölçmesi istenir.

Veri setinde genel eğilime uygun olmayan veriler olarak tanımlayabileceğimiz aşırı deęerler, T^2 ve MANOVA sonuçlarını ciddi şekilde etkileyebilirler. Analiz sonuçlarını elde etmeden önce, aşırı deęerlerin belirlenmesi, düzeltilmesi ya da veriden çıkartılması iyi bir seçenektir (Alpar 2003).

Çok deęişkenli varyans analizinde H_0 hipotezi, ikiden çok ortalama vektörünün eşit olduğu şekilde kurulur.

$$H_0 : \begin{bmatrix} \mu_{11} \\ \mu_{12} \\ \cdot \\ \cdot \\ \mu_{1p} \end{bmatrix} = \begin{bmatrix} \mu_{21} \\ \mu_{22} \\ \cdot \\ \cdot \\ \mu_{2p} \end{bmatrix} = \dots = \begin{bmatrix} \mu_{k1} \\ \mu_{k2} \\ \cdot \\ \cdot \\ \mu_{kp} \end{bmatrix} \text{ hipotezi,}$$

H_1 : En az bir ortalama vektörü diđerlerinden farklıdır.

hipotezine karşı test edilir. Burada μ_{kp} : p . deęişkenin k . gruptaki yığın ortalamasıdır.

2.2.5.1. Varsayımlar

Çok deęişkenli varyans analizinde her bir grubun çok deęişkenli normal dağılım göstermesi ve grupların kovaryans matrislerinin eşit olması gerekmektedir. Ayrıca gruplardaki gözlem sayısının 20'den fazla olması ve bir gruptaki gözlem sayısının deęişken sayısından az olmaması istenir (Alpar 2003).

Normallik varsayımına Bölüm 2.2.2.1 ve kovaryans matrislerinin eşit olup olmadığına ise Bölüm 2.2.2.2'deki gibi bakılır.

2.2.5.2. Test istatistięi

Çok deęişkenli analizlerde ortalama vektörleri arasında fark olup olmadığını incelemekte kullanılan birçok test yöntemi vardır. Bu yöntemlerden en çok kullanılan dört tanesi; Wilks lamda istatistięi, Pillai iz istatistięi, Hotelling iz istatistięi ve Roy'un en büyük özdeęere dayalı test istatistięidir.

Bu istatistikler farklı dağılımlara sahiptirler ancak; F istatistięine dönüştürülebilirler. Bu yöntemlerden hangisinin tercih edilmesi gerektięi konusunda çeşitli yaklaşımlar söz konusudur. MANOVA varsayımlarının bozulmasına karşı Wilks lamda ve Pillai yaklaşımlarının daha baęışık olduęu söylenmekle birlikte, gözlem sayısının daha az ya da eşit olmadığı ya da varyansların homojenlięi varsayımının yerine getirilemedięi durumlarda Pillai iz yaklaşımının daha güvenilir olduęu belirtilmektedir. Roy yaklaşımı ise varsayımların kesinlikle saęlandığı durumlarda kullanılmalıdır (Alpar 2003).

2.2.5.2.1. Wilks Lamda istatistiği

Varyans analizinde *Genel KT=(Gruplar Arası KT + Grup İçi KT)* şeklinde verilen eşitlik, çok değişkenli varyans analizinde matrislerle ifade edilir ve

T : Genel kareler toplamı matrisi,

B : Gruplar arası kareler toplamı matrisi,

W : Grup içi kareler toplamı matrisi ve

p : Her gruptaki değişken sayısı

olmak üzere;

$$T_{p \times p} = B_{p \times p} + W_{p \times p}$$

ile verilir. Bu eşitlikte;

k : Ortalama vektörü sayısı (karşılaştırılacak grup sayısı),

\bar{x}_i : i . gruba ilişkin ortalama vektörü,

\bar{x} : Genel ortalama vektörü,

n_i : i . gruba ilişkin gözlem sayısı ve

S_i : i . gruba ilişkin varyans-kovaryans matrisi

olmak üzere, B ve W matrisleri;

$$B = \sum_{i=1}^k n_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})'$$

$$W = \sum_{i=1}^k (n_i - 1)S_i$$

ile verilirken ilgili serbestlik dereceleri sırasıyla $Sd=k-1$ ve $Sd = \sum_{i=1}^k n_i - k$ 'dır. Bu

bilgiler yardımıyla Wilks lamda (Λ) istatistiği (U istatistiği olarak da bilinir.):

$$\Lambda = \frac{|W|}{|W + B|} = \frac{|W|}{|T|}$$

şeklinindedir. Bu eşitlik iki determinantın oranını içermektedir. Bu oranın sıfıra yaklaşması, gruplar arasında (ortalama vektörleri arasında) fark olduğunun bir göstergesidir.

Bir matrisin en iyi tanımlayıcısı o matrisin özdeğerleri ve determinantıdır. p değişken sayısını göstermek ve λ_i 'ler BW^{-1} matrisinin özdeğerleri ve $s = \text{enk}(k-1, p)$, $m = \frac{|p-(k-1)|-1}{2}$, $\tilde{n} = \frac{n-p-k-1}{2}$; Λ istatistiğinin ve diğer istatistiklerin parametreleri olmak üzere; $\Lambda = \prod_{i=1}^s \frac{1}{1+\lambda_i}$ ile de verilir. Buradaki s değeri, BW^{-1} matrisinde sıfırdan farklı özdeğerlerin sayısıdır. Eğer incelenen gözlem sayısı yeterli ise Wilks lamda değeri;

- k: Karşılaştırılacak grup sayısı,
- p: Her gruptaki değişken sayısı,
- n: Toplam gözlem sayısı

olmak üzere; $L = -[n-1-(p+k)/2]\ln \Lambda$ ile $p(k-1)$ serbestlik dereceli χ^2 dağılımı gösterir ve $L > \chi_{\text{Tablo}[p(k-1), \alpha]}^2$ ise gruplar arasında fark vardır denir.

Wilks lamda istatistiğinin anlamlılığı, F dağılımı yardımı ile de test edilebilir. Aşağıdaki tablo değişik grup ve gözlem sayılarında Λ değerlerine bağlı olarak elde edilen F istatistiklerinin, hangi serbestlik derecelerinde bir F dağılımı gösterdiklerini vermektedir (Alpar 2003).

Tablo 2. Grup ve Değişken Sayısına Göre Λ 'nın Dağılımı

Değişken Sayısı	Grup Sayısı	F Dağılımına Yaklaşım	F Dağılımı Serbestlik Dereceleri
$p=1$	$k \geq 2$	$\left(\frac{n-k}{k-1}\right)\left(\frac{1-\Lambda}{\Lambda}\right)$	$k-1, n-k$
$p=2$	$k \geq 2$	$\left(\frac{n-k-1}{k-1}\right)\left(\frac{1-\sqrt{\Lambda}}{\sqrt{\Lambda}}\right)$	$2(k-1), 2(n-k-1)$
$p \geq 1$	$k = 2$	$\left(\frac{n-p-1}{p}\right)\left(\frac{1-\Lambda}{\Lambda}\right)$	$p, n-p-1$
$p \geq 1$	$k = 3$	$\left(\frac{n-p-2}{p}\right)\left(\frac{1-\sqrt{\Lambda}}{\sqrt{\Lambda}}\right)$	$2p, 2(n-p-2)$

2.2.5.2.2. Hotelling İz istatistiği

λ_i 'ler BW^{-1} matrisinin özdeğerleri olmak üzere, Hotelling iz istatistiği (T), $T = \sum_{i=1}^s \lambda_i$ ile verilir. Gözlem sayısı yeterli iken, $nT > \chi^2_{[p(k-1);\alpha]}$ ise ortalama vektörleri arasında fark olduğu söylenir. T istatistiğini test etmek için F dağılımından da yararlanılır. Buna göre, $F = \frac{2(s\tilde{n} + 1)}{s^2(2m + s + 1)}T$ değeri $s(2m + s + 1)$ ve $2(s\tilde{n} + 1)$ ile bir F dağılımı gösterir (Alpar 2003).

2.2.5.2.3. Pillai İz istatistiği

$T = \sum_{i=1}^s \frac{\lambda_i}{1 + \lambda_i}$ ile verilir. T istatistiği yardımıyla bulunan $\frac{2\tilde{n} + s + 1}{2m + s + 1} \times \frac{T}{s - T}$ değeri $s(2m + s + 1)$ ve $s(2\tilde{n} + s + 1)$ serbestlik dereceleri ile bir F dağılımı gösterir (Alpar 2003).

2.2.5.2.4. Roy'un en büyük özdeğere dayalı test istatistiği

En büyük özdeğer λ_{enb} ile gösterilirse, Roy'un en büyük özdeğere dayalı test istatistiği, $T = \frac{\lambda_{enb}}{1 + \lambda_{enb}}$ ile verilir. Bulunan T istatistiği, s, m ve \tilde{n} parametrelili Heck grafik değeri ile karşılaştırılır. T istatistiğinin Heck grafik değerinden büyük olması durumunda ortalama vektörleri arasında fark olduğu söylenir (Alpar 2003).

2.2.5.3. Bonferroni eşanlı güven aralıkları ve farklı grup ve değişkenlerin belirlenmesi

H_0 hipotezi reddedildiğinde, yani k ortalama vektörü arasında fark olduğunda; farklılığın hangi değişken(ler)den kaynaklandığını belirlemek için aşağıdaki yöntemden yararlanır.

i : İlgili değişkeni,

m ve l : i . değişken açısından karşılaştırılacak grupları ve

w_{ii} : W matrisinde i . değişkene ilişkin w_{ii} köşegen elemanını

göstermek üzere,

$$(\bar{x}_{mi} - \bar{x}_{li}) - t_{\left(n-k; \frac{\alpha}{pk(k-1)}\right)} \sqrt{\frac{w_{ii}}{n-k} \left(\frac{1}{n_m} - \frac{1}{n_l} \right)} \leq \mu_{mi} - \mu_{li} \leq (\bar{x}_{mi} - \bar{x}_{li}) + t_{\left(n-k; \frac{\alpha}{pk(k-1)}\right)} \sqrt{\frac{w_{ii}}{n-k} \left(\frac{1}{n_m} - \frac{1}{n_l} \right)}$$

şeklindedir (Alpar 2003).

3. ÇOK DEĞİŞKENLİ PARAMETRİK OLMAYAN KONUM TESTİ

3.1. Giriş

Bağımsız örneklerden çekilen $c (\geq 2)$ tane F_1, \dots, F_c çok değişkenli dağılım fonksiyonlarının eşitliğinin testi düşünölsün. F_1, \dots, F_c 'nin hepsi çok değişkenli normal dağıldığında, sadece ortalama vektörleri ve kovaryans matrisleri farklı olabilir. Fakat F_1, \dots, F_c normal dağılmadığında dağılımlarındaki farklılık değişik sebeplerden kaynaklanabilir. Ortalama vektörlerinin ve kovaryans matrislerinin eşitliği F_1, \dots, F_c 'nin eşitliği anlamını taşımaz. Bu nedenle, F_1, \dots, F_c 'nin belirlenmemiş ortak bir biçime sahip olduğu, sadece ortalama vektörlerinin (veya kovaryans matrislerinin) ya da ilişkisel yapılarının farklı olduğu varsayılacaktır. Burada, F_1, \dots, F_c dağılımlarının ortalama vektörlerinin ve kovaryans matrislerinin eşitliği testi problemi ile ilgili rank testleri üzerinde durulacaktır (Puri, Sen 1971).

Çok örnekli çok değişkenli durumda genellikle, sıra istatistiğinin (koşulsuz) dağılımının, dağılımların eşitliğini belirten sıfır hipotezi doğru olduğunda bile varsayılan dağılıma bağlı olduğu görülür. Bu olumsuzluğu gidermek için, birleşik örnek gözlemlerinin dağılımının permütasyonel değişmez yapısı çok değişkenli olarak genişletilecektir. Bu, tek değişkenli durumların çok değişkenli genellemelerinde, uygun sıra istatistiklerine dayalı koşullu (permütasyonel) testler oluşturulmasını sağlayacaktır (Puri, Sen 1971).

3.2. Çok Değişkenli Çok Örnekli Ortalama ve Varyans Problemlerinin Formülasyonu

$\{X_\alpha^{(k)} = (X_{1\alpha}^{(k)}, \dots, X_{p\alpha}^{(k)})', \alpha = 1, \dots, n_k, k = 1, \dots, c\}$ tesadüfi değerlerden oluşan bağımsız vektörlerin bir kümesi olsun (Um, Randles 1998).

$X_\alpha^{(k)}$ 'nın birikimli dağılım fonksiyonu $F_k(x)$ ile ifade edilsin. Kabul edilebilir hipotezlerin kümesi, her bir $F_k(x)$ 'in bazı dağılım fonksiyonlarının sınıfı Ω 'ya ait olduğunu gösterir. Test edilecek hipotez $H_0, F \in \Omega$ iken tüm x 'ler için

$$F_1(x) = \dots = F_c(x) = F(x) \quad (3.2.1)$$

şeklinde gösterilir. H_0 'ın alternatif hipotezinde; her bir $F_k(x)$, Ω 'ya aittir fakat (3.2.1) sağlanmaz. Gözlemlerin eşitliği durumundan kaçınmak için Ω sınıfı, tüm sürekli dağılım fonksiyonlarının sınıfı olarak kabul edilir (Puri, Sen 1971).

Konum testlerinde, $F \in \Omega$ olmak üzere tüm $k = 1, \dots, c$ için;

$$F_k(x) = F(x + \delta_k) \quad (3.2.2)$$

alınır ve alternatifi $\delta_1, \dots, \delta_c$ 'nin hepsinin eşit olmadığı olan

$$H_0^{(1)} : \delta_1 = \dots = \delta_c = 0 \quad (3.2.3)$$

temel hipotezinin testi ile ilgilenilir.

$F \in \Omega$ olmak üzere yeniden

$$F_k(x) = F(X_k^*), \quad X_k^* = \left(\frac{x_1 - \mu_1}{\sigma_1^{(k)}}, \dots, \frac{x_p - \mu_p}{\sigma_p^{(k)}} \right) \quad (3.2.4)$$

ve tüm $k = 1, \dots, c$ için $\sigma^{(k)} = (\sigma_1^{(k)}, \dots, \sigma_p^{(k)})$ alınır. O halde yaygınlık testleri kapsamında, alternatifi $\sigma_1, \dots, \sigma_c$ 'nin hepsinin eşit olmadığı olan

$$H_0^{(2)} : \sigma^{(1)} = \dots = \sigma^{(c)} = I, \quad (3.2.5)$$

temel hipotezinin testi ile ilgilenilir (Burada I birim vektördür.). Bu arada (3.2.4)'te F_1, \dots, F_c 'nin ortalama vektörlerinin homojen olduğu varsayılır (Puri, Sen 1971).

3.3. Temel Rank Permütasyon Prensipli

$X_{i\alpha}^{(k)}, \alpha = 1, \dots, n_k, k = 1, \dots, c$ olmak üzere her i . değişken için $N (= \sum_{k=1}^c n_k)$

gözleme küçükten büyüğe sıra numarası verilsin ve bu kümede $X_{i\alpha}^{(k)}$ 'nin rankı $R_{i\alpha}^{(k)}$ ile ifade edilsin. Ya da başka bir ifadeyle; her satıra kendi içinde 1'den N 'ye kadar küçükten büyüğe sıra numarası verildiğinde aşağıdaki $R_N^{p \times N}$ matrisi elde edilir.

Bu durumda $X_{\alpha}^{(k)} = (X_{1\alpha}^{(k)}, \dots, X_{p\alpha}^{(k)})'$ gözlem vektörü

$$X_N^{p \times N} = \begin{pmatrix} X_{11}^{(1)} & X_{12}^{(1)} & \dots & X_{1n_1}^{(1)} & X_{11}^{(2)} & X_{12}^{(2)} & \dots & X_{1n_2}^{(2)} & \dots & \dots & X_{11}^{(c)} & X_{12}^{(c)} & \dots & X_{1n_c}^{(c)} \\ X_{21}^{(1)} & X_{22}^{(1)} & \dots & X_{2n_1}^{(1)} & X_{21}^{(2)} & X_{22}^{(2)} & \dots & X_{2n_2}^{(2)} & \dots & \dots & X_{21}^{(c)} & X_{22}^{(c)} & \dots & X_{2n_c}^{(c)} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & & \vdots & \vdots & & \vdots \\ X_{p1}^{(1)} & X_{p2}^{(1)} & \dots & X_{pn_1}^{(1)} & X_{p1}^{(2)} & X_{p2}^{(2)} & \dots & X_{pn_2}^{(2)} & \dots & \dots & X_{p1}^{(c)} & X_{p2}^{(c)} & \dots & X_{pn_c}^{(c)} \end{pmatrix} \quad (3.3.1)$$

olur (Um, Randles 1998).

Bu gözlem vektörü, $R_{\alpha}^{(k)} = (R_{1\alpha}^{(k)}, \dots, R_{p\alpha}^{(k)})'$, $\alpha = 1, \dots, n_k, k = 1, \dots, c$ rank vektörüne dönüşür. N tane gözlem vektörüne karşılık gelen N tane rank vektörü, rank matrisi

$$R_N^{p \times N} = \begin{pmatrix} R_{11}^{(1)} & R_{12}^{(1)} & \dots & R_{1n_1}^{(1)} & R_{11}^{(2)} & R_{12}^{(2)} & \dots & R_{1n_2}^{(2)} & \dots & \dots & R_{11}^{(c)} & R_{12}^{(c)} & \dots & R_{1n_c}^{(c)} \\ R_{21}^{(1)} & R_{22}^{(1)} & \dots & R_{2n_1}^{(1)} & R_{21}^{(2)} & R_{22}^{(2)} & \dots & R_{2n_2}^{(2)} & \dots & \dots & R_{21}^{(c)} & R_{22}^{(c)} & \dots & R_{2n_c}^{(c)} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & & \vdots & \vdots & & \vdots \\ R_{p1}^{(1)} & R_{p2}^{(1)} & \dots & R_{pn_1}^{(1)} & R_{p1}^{(2)} & R_{p2}^{(2)} & \dots & R_{pn_2}^{(2)} & \dots & \dots & R_{p1}^{(c)} & R_{p2}^{(c)} & \dots & R_{pn_c}^{(c)} \end{pmatrix} \quad (3.3.2)$$

ile ifade edilir. Bu matrisin her bir satırı $1, 2, \dots, N$ sayının rastgele permütasyonudur.

Bu nedenle $R_N^{p \times N}$, $(N!)^p$ tane mümkün değişimin rastgele bir matrisidir. (3.3.2)

şeklindeki iki matristen biri, diğerinin kolonlarının yeniden düzenlenmesiyle elde edilmişse bu iki matrisin permütasyonel olarak eşit olduğu söylenebilir. Bu nedenle

R_N matrisi, kendisi ile aynı kolon vektörlerine sahip bir diğer matris olan R_N^* matrisine permütasyonel olarak eşittir; fakat öyle bir düzenlenmiştir ki, R_N^* 'ın ilk satırı $1, 2, \dots, N$ 'nin normal sıralanmış halinden oluşur. Başka bir deyişle,

$$R_N^* = \begin{pmatrix} 1 & 2 & \dots & N \\ R_{21}^* & R_{22}^* & \dots & R_{2N}^* \\ \dots & \dots & \dots & \dots \\ R_{p1}^* & R_{p2}^* & \dots & R_{pN}^* \end{pmatrix} \quad (3.3.3)$$

şeklindedir. p değişkenli $X_{1\alpha}^{(k)}$, $i = 1, \dots, p$ genel olarak tesadüfi olarak bağımlı olduğu için R_N (veya R_N^*)'nin elemanlarının ortak dağılımı, (3.2.1) sağlansa bile bilinmeyen bir $F (\in \Omega)$ dağılımına bağlı olacaktır. Şimdi, her bir satırı (ilki hariç) $1, \dots, N$ sayılarının bir permütasyonu olan R_N^* , $(N!)^{p-1}$ tane farklı şekilde elde edilebiliyor. Tüm bu farklı durumlardan oluşan küme \mathfrak{R}_N^* ile ifade edilsin. Böylece, R_N^* 'nin, \mathfrak{R}_N^* 'deki koşulsuz dağılımı, F_1, \dots, F_c aynı olduğu zaman genellikle F_1, \dots, F_c 'ye bağlıdır. Bununla birlikte $F_1 \equiv \dots \equiv F_c$ olmak üzere $X_{\alpha}^{(k)}$, $\alpha = 1, \dots, n_k$, $k = 1, \dots, c$ 'nin hepsi bağımsız ve aynı dağılımlı rastgele vektörler olduğu zaman bunların ortak dağılımı, bu vektörlerin kendi aralarında herhangi bir permütasyon olması durumunda sabit kalacaktır. Bu durum şu anlama gelir; $S(R_N^*)$ uzayında (R_N^* 'nin sütunlarının tüm mümkün permütasyonlarıyla elde edilebilen) $N!$ mümkün değişimin kümesinde R_N 'nin koşullu dağılımı $H_0 : F_1 \equiv \dots \equiv F_c \equiv F$ hipotezi altında düzgün olacaktır. Bütün aynı birikimli dağılım fonksiyonları $F (\in \Omega)$ olduğunda tüm $r_N \in S(R_N^*)$ için,

$$P\{R_N = r_N \mid S(R_N^*), H_0\} = 1/N! \quad (3.3.4)$$

olur. R_N^* 'nin sütunlarının $N!$ tane mümkün değişimiyle oluşan koşullu (permütasyonel) olasılık ölçüsü \mathfrak{T}_N ile ifade edilsin. Tamamıyla belirlenmiş (koşullu) dağılıma sahip \mathfrak{T}_N altında, R_N 'ye açık bir şekilde bağlı herhangi bir istatistik düşünüldüğünde bu (3.3.4)'e uyar. Sonuç olarak, R_N 'ye açık bir şekilde bağlı bir test fonksiyonu olan $\phi (0 \leq \phi \leq 1)$ oluşturulabilir, öyle ki $E\{\phi \mid \mathfrak{T}_N\} = \epsilon : 0 < \epsilon < 1$ anlamlılık düzeyidir. Bu, $E\{\phi \mid H_0\} = E[E\{\phi \mid \mathfrak{T}_N\}] = \epsilon$ anlamına gelir. Böylece ϕ , ϵ 'nin büyüklüğünün testinin bir benzeri olacaktır.

Bu son bakış açısıyla, açık bir şekilde R_N 'ye bağlı olan ve \mathfrak{T}_N olasılık kuralına dayanan testlerin bir sınıfı ele alınacak. Bu testler, lineer sıralı sıra istatistiklerine dayalı permütasyon testleri olarak adlandırılacak (Puri, Sen 1971).

3.4. Bazı Permütasyon Sıralı Rank Testleri

$1, \dots, N$ sıralı rakamlarının açık bir şekilde ifade edilen fonksiyonlarıyla, genellikle aşağıdaki gibi tanımlanan rank skorlarının genel bir sınıfı ile başlanır:

$$E_{N,\alpha}^{(i)} = J_{N(i)} \left(\frac{\alpha}{N+1} \right), \quad 1 \leq \alpha \leq N, \quad i = 1, \dots, p. \quad (3.4.1)$$

Şimdi, tüm $i = 1, \dots, p, \alpha = 1, \dots, n_k, k = 1, \dots, c$ için R_N 'deki $R_{i\alpha}^{(k)}$ rankları, $E_{N,R_{i\alpha}^{(k)}}^{(i)}$ 'ler ile yer değiştirdiğinde; E_N ile ifade edilen $p \times N$ boyutlu genel skor matrisi elde edilir. Böylece,

$$E_N = \begin{pmatrix} E_{N,R_{11}^{(1)}}^{(1)} & E_{N,R_{12}^{(1)}}^{(1)} & \dots & E_{N,R_{1n_1}^{(1)}}^{(1)} & E_{N,R_{11}^{(2)}}^{(1)} & E_{N,R_{12}^{(2)}}^{(1)} & \dots & E_{N,R_{1n_2}^{(2)}}^{(1)} & \dots & \dots & E_{N,R_{11}^{(c)}}^{(1)} & E_{N,R_{12}^{(c)}}^{(1)} & \dots & E_{N,R_{1n_c}^{(c)}}^{(1)} \\ E_{N,R_{21}^{(1)}}^{(2)} & E_{N,R_{22}^{(1)}}^{(2)} & \dots & E_{N,R_{2n_1}^{(1)}}^{(2)} & E_{N,R_{21}^{(2)}}^{(2)} & E_{N,R_{22}^{(2)}}^{(2)} & \dots & E_{N,R_{2n_2}^{(2)}}^{(2)} & \dots & \dots & E_{N,R_{21}^{(c)}}^{(2)} & E_{N,R_{22}^{(c)}}^{(2)} & \dots & E_{N,R_{2n_c}^{(c)}}^{(2)} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & & \vdots & \vdots & & \vdots \\ E_{N,R_{p1}^{(p)}}^{(p)} & E_{N,R_{p2}^{(p)}}^{(p)} & \dots & E_{N,R_{pn_1}^{(p)}}^{(p)} & E_{N,R_{p1}^{(2)}}^{(p)} & E_{N,R_{p2}^{(2)}}^{(p)} & \dots & E_{N,R_{pn_2}^{(2)}}^{(p)} & \dots & \dots & E_{N,R_{p1}^{(c)}}^{(p)} & E_{N,R_{p2}^{(c)}}^{(p)} & \dots & E_{N,R_{pn_c}^{(c)}}^{(p)} \end{pmatrix}, \quad (3.4.2)$$

olur. Sonra, c örneğin her bir $i (= 1, \dots, p)$ değişkeni için ortalama rank skorları aşağıdaki gibi bulunur:

$$T_{Ni}^{(k)} = (1/n_k) \sum_{\alpha=1}^{n_k} E_{N,R_{i\alpha}^{(k)}}^{(i)}, \quad k = 1, \dots, c; \quad i = 1, \dots, p \quad (3.4.3)$$

Gösterim kolaylığı için;

$$T_{Ni}^{(k)} = \sum_{\alpha=1}^N E_{N,\alpha}^{(i)} Z_{Ni,\alpha}^{(k)} / n_k \quad (3.4.4)$$

yazılacak. Burada; $E_{N,\alpha}^{(i)}$ verilen değerler (skorlar)dir ve $X_{i\alpha}^{(k)}, \alpha = 1, \dots, n_k, k = 1, \dots, c$ 'lar arasında α 'ncı en küçük gözlem k . örnekten ise $Z_{Ni,\alpha}^{(k)} = 1$, aksi halde $Z_{Ni,\alpha}^{(k)} = 0$ 'dır.

Daha sonra, gerekli işlemlerle

$$E(T_{Ni}^{(k)} - \bar{E}_N^{(i)} | \mathfrak{S}_N) = 0, \quad (3.4.5)$$

$$\text{Kroneker delta fonksiyonu } \delta_{kq} = \begin{cases} 1, & k = q \\ 0, & k \neq q \end{cases},$$

$$\text{cov}(T_{Ni}^{(k)}, T_{Nj}^{(q)} | N) = (N \delta_{kq} - n_k) v_{ij}(R_N^*) / n_k (N-1), \quad i, j = 1, \dots, p, \quad k = 1, \dots, c, \quad (3.4.6)$$

ve rank $S = R_{i\alpha}^{(k)}$ ile ilişkili olan $E_{N,S}^{(i)}$ 'nin değeri $E_{N\alpha,i}^{(k)}$ olmak üzere; yaygınlık matrisi $V(R_N^*)$ 'nin elemanları olan

$$v_{ij}(R_N^*) = \frac{1}{N} \sum_{k=1}^c \sum_{\alpha=1}^{n_k} E_{N\alpha,i}^{(k)} E_{N\alpha,j}^{(q)} - \bar{E}_N^{(i)} \bar{E}_N^{(j)} \quad (3.4.7)$$

ve beklenen skor değerleri olan

$$\bar{E}_N^{(i)} = \sum_{\alpha=1}^N E_{N,\alpha}^{(i)} / N, \quad i = 1, \dots, p \quad (3.4.8)$$

elde edilir. H_0 hipotezinin doğruluğu altında, bütün $X_{\alpha}^{(k)}, \alpha = 1, \dots, n_k, k = 1, \dots, c$ gözlemleri aynı dağılıma sahiptir. Bu yüzden her bir i için, $T_{(k)}^{Ni}, k = 1, \dots, c$ ortalama skorlarından ağırlıklı toplam ortalama skor değeri $\sum_{k=1}^c n_k T_{N,i}^{(k)} / N = \bar{E}_N^{(i)}$ elde edilebilir. Burada, $T_{N,i}^{(k)}$ ortalama skorları arasındaki farklara dayalı H_0 için bir test kurmak anlamlı görünmektedir. Özellikle, $p(c-1)$ tane $T_{N,i}^{(k)} - \bar{E}_N^{(i)}, k = 1, \dots, c-1, i = 1, \dots, p$ farkının bir kümesi alınabilir ve bunlara dayalı test kurulabilir. H_0 hipotezinin doğruluğu altında, tüm bu farklar tesadüfi olarak değerce küçük olacaktır. Bununla beraber, pratik kullanım için bu farkların tekil bir fonksiyonuna dayalı bir test kurmak uygun olacaktır. Bu nedenle, farkların her birinin sayısal büyüklüğünü reddedecek bir fonksiyon seçilir. Bu farkların pozitif tanımlı bir karesel formu bir çözüm olabilir. Şimdi şunu tanımlayalım:

$$V(R_N^*) = ((v_{ij}(R_N^*)))_{i,j=1,\dots,p} \quad (3.4.9)$$

Yukarıdaki $p(c-1)$ tane farkın (koşullu) permütasyonel kovaryans matrisi (H_0 'ın doğruluğu altında) şöyledir:

$$\frac{1}{(N-1)} \left(\left(\frac{N}{n_k} \delta_{kq} - 1 \right) \right)_{k,q=1,\dots,c} \otimes V(R_N^*) \quad (3.4.10)$$

Burada, \otimes iki matrisin Kroneker çarpımıdır. Faktör matrisleri pozitif tanımlı olduğunda Kroneker çarpımın pozitif tanımlı olması dolayısıyla; $V(R_N^*)$ pozitif tanımlıysa (3.4.10) de pozitif tanımlı olur. Buradan; eğer $V(R_N^*)$ pozitif tanımlıysa (3.4.10) matrisinin tersi,

$$(N-1) \left(\frac{n_k}{N} \delta_{kq} + \frac{n_k n_q}{n_c} \right)_{k,q=1,\dots,c} \otimes V^{-1}(R_N^*) \quad (3.4.11)$$

olur. Burada,

$$V^{-1}(R_N^*) = ((v_{ij}(R_N^*)))^{-1} \quad (3.4.12)$$

şeklinindedir. Burada aşağıdaki testin yapısı asimtotik olarak, Lawley-Hotelling'in geliştirilmiş T^2 istatistiğine dayalı olabilirlik oranı testiyle aynıdır. Test istatistiği L_N şöyledir:

$$L_N = \sum_{k=1}^c n_k \left[(T_N^{(k)} - \bar{E}_N) V^{-1}(R_N^*) (T_N^{(k)} - \bar{E}_N)' \right] \quad (3.4.13)$$

$T_N^{(k)} = (T_{N1}^{(k)}, \dots, T_{Np}^{(k)})$ ve $\bar{E}_N = (\bar{E}_N^{(1)}, \dots, \bar{E}_N^{(p)})$ olmak üzere; L_N istatistiği, $V_N^{-1}(R_N^*)$ ile aynı diskriminantlı $(T_N^{(k)} - \bar{E}_N), k=1, \dots, c$ 'deki c çarpım formunun ağırlıklı toplamıdır. L_N test istatistiği elde edilirken $V(R_N^*)$ 'nin pozitif tanımlı olduğu varsayılır. Eğer $V(R_N^*)$ pozitif tanımlı değilse, $V(R_N^*)$ 'nin en yüksek dereceli tekil olmayan minörü veya $V_N^{-1}(R_N^*)$ 'nin geliştirilmiş tersi kullanılabilir ve L_N 'ye benzer bir istatistik elde edilir. Daha önce yapılanlar göz önüne alındığında; R_N^* verildiğinde L_N 'nin şartlı dağılımının, $F(x) \in \Omega$ ne olursa olsun H_0 'ın kabul edilmesi halinde aynı olduğu görülür. Bu yüzden aşağıdaki tesadüfi test prosedürü H_0 testinde ϵ kesin seviyesini sağlar.

$$\phi(R_N) = \begin{cases} 1 & \text{if } L_N > L_N(R_N^*) \\ A_{N,\epsilon}(R_N^*) & \text{if } L_N = L_N(R_N^*) \\ 0 & \text{if } L_N < L_N(R_N^*) \end{cases} \quad (3.4.14)$$

Burada tesadüfi kesim noktası

$$E\{\phi(R_N) | \mathfrak{T}_N\} = \epsilon \quad (3.4.15)$$

ile belirtilir. (3.4.15) ayrıca testin koşulsuz büyüklüğünün ϵ olduğunu gösterir. Bu nedenle L_N 'ye dayalı permütasyon testi kesinlikle serbest dağılımlıdır (Puri, Sen 1971).

Testin yapılması için, \mathfrak{T}_N altında L_N 'nin $N!$ tane tüm muhtemel kombinasyon değerleri gerekebilir. Burada L_N 'nin R_N^* altında en fazla $N! / \prod_{k=1}^c n_k!$ tane ayrı değeri vardır (Puri, Sen 1971).

N ve p 'nin küçük değerleri hariç, hesaplamaların aşırı olmasından dolayı permütasyon testi (3.4.14)'in tam bir uygulaması zordur. Bu durum L_N 'nin büyük örnekli dağılımı üzerinde çalışılmasına yol açar. Bundan önce (3.4.13)'teki L_N 'ye dayalı çeşitli yaygın testleri gösterelim (Puri, Sen 1971).

3.4.1. Çok değişkenli çok örnekli medyan testi

Burada, $\alpha = 1, \dots, N$; $i = 1, \dots, p$ için $\alpha \leq [N/2]$ ($= a$) olup olmamasına göre $E_{N,\alpha}^{(i)}$ 'ler 1 ya da 0'a eşitlenir. Böylece $T_{N,i}^{(k)}$; k. örneğin, i. değişken değerlerinin örnek oranıdır. Bu durumda, (3.4.7)'den tüm $i = 1, \dots, p$ ve $R_N^* \in \mathfrak{R}_N^*$ için

$$v_{ii}(R_N^*) = \frac{a(N-a)}{N^2} \quad (3.4.16)$$

ve $i \neq j$ iken

$$v_{ij}(R_N^*) = a_{N,ij} - (a/N)^2 \quad (3.4.17)$$

olur. Burada $a_{N,ij}$; $i \neq j = 1, \dots, p$ olmak üzere i . ve j . deęişken deęerlerinin, aynı anda ilgili a 'ncı en küçük deęerlere eşit veya bu deęerlerden küçük olduęu durumlar için birleştirilmiş örnek gözlemlerinin oranıdır. Sonuç olarak; (3.4.13) ve (3.4.16)'ye bakıldığında L_N istatistięinin, $p=1$ için Brown ve Mood (1951) tarafından bulunan tek deęişkenli çok örnekli medyan test istatistięine indirildięi görülür (Puri, Sen 1971).

3.4.2. Rank toplamı testi

Burada $1 \leq \alpha \leq N$ ve $i = 1, \dots, p$ için $E_{N,\alpha}^{(i)} = \alpha/(N+1)$ olur. Sonuçta $T_{N,i}^{(k)}$ ifadesi, $k = 1, \dots, c$ ve $i = 1, \dots, p$ için birleştirilmiş örneęin i . deęişkeninin gözlemleri arasında k . örneęin i . deęişkeninin gözlemlerinin ortalama rankına $(N+1)^{-1}$ kez indirgenebilir. $1 \leq \alpha \leq N$ ve $i = 1, \dots, p$ olmak üzere $E_{N,\alpha}^{(i)} = \alpha/(N+1)$ 'leri, (3.4.7)'de yerine koyarak $v_{ij}(R_N^*)$ ifadelerini elde etmek çok kolaydır. $p = 1$ için, L_N tekrar Kruskal-Wallis testine dönüşür (Puri, Sen 1971).

3.4.3. Normal skorlar testi

Burada, $E_{N,\alpha}^{(i)}$, $\alpha = 1, \dots, N$ ve $i = 1, \dots, p$ olmak üzere standart normal dağılıma sahip N büyüklüğündeki bir örneęin α 'ncı en küçük gözleminin beklenen deęeri olarak tanımlansın.

Şimdi yaygınlık problemi ele alınsın. Burada;

- i. $E_{N,\alpha}^{(i)}$, $|\alpha - (N+1)/2| \geq b$ olup olmamasına göre 1 veya 0'a eşit,
- ii. $E_{N,\alpha}^{(i)}$, $|\alpha/(N+1) - (1/2)|$ 'ye eşit,
- iii. $E_{N,\alpha}^{(i)}$, $(\alpha/(N+1) - (1/2))^2$ 'ne eşit veya
- iv. $E_{N,\alpha}^{(i)}$, tüm $\alpha = 1, \dots, N$; $i = 1, \dots, p$ için III'teki (ortalama problemi) normal skorların kareleri olur (Puri, Sen 1971).

3.5. L_N 'nin Asimtotik Permütasyon Dağılımı

$i, j = 1, \dots, p$ ve $k = 1, \dots, c$ olmak üzere; $X_{i\alpha}^{(k)}$ ve $(X_{i\alpha}^{(k)}, X_{j\alpha}^{(k)})$ 'nin marjinal birikimli dağılım fonksiyonları sırasıyla $F_{[i]}^{(k)}(x)$ ve $F_{[i,j]}^{(k)}(x, y)$ şeklinde ifade edilsin. $N = \sum_{k=1}^c n_k$ ve $\lambda_N^{(k)} = n_k/N$ olsun ve tüm N 'ler için, $\lambda_0 \leq 1/c$ olmak üzere, aşağıdaki eşitsizlik varsayalım:

$$0 < \lambda_0 \leq \lambda_N^{(k)} \leq 1 - \lambda_0 < 1, \quad k = 1, \dots, c. \quad (3.5.1)$$

$$F_{N[i]}^{(k)}(x) = n_k^{-1} \quad (X_{i\alpha}^{(k)})'nin \text{değeri} \leq x, \alpha = 1, \dots, n_k) \quad (3.5.2)$$

olsun. Burada $F_{N[i]}^{(k)}(x)$, k 'inci örneğin i 'inci değişkeninin gözlemlerinin örnek birikimli dağılım fonksiyonudur (Puri, Sen 1971).

$$H_{N[i]}^{(k)}(x) = \sum_{k=1}^c \lambda_N^{(k)} F_{N[i]}^{(k)}(x), \quad i = 1, \dots, p \quad (3.5.3)$$

şeklinde tanımlansın. Burada $H_{N[i]}^{(k)}(x)$, birleştirilmiş örneğin i . değişkeninin gözlemlerinin örnek birikimli dağılım fonksiyonudur (Puri, Sen 1971).

$$F_{N[i,j]}^{(k)}(x, y) = n_k^{-1} \quad ((X_{i\alpha}^{(k)}, X_{j\alpha}^{(k)})'in \text{değeri} \leq (x, y)), \quad (3.5.4)$$

$$H_{N[i,j]}(x, y) = \sum_{k=1}^c \lambda_N^{(k)} F_{N[i,j]}^{(k)}(x, y) \quad (3.5.5)$$

ve

$$H_{[i,j]}(x, y) = \sum_{k=1}^c \lambda_N^{(k)} F_{[i,j]}^{(k)}(x, y) \quad (3.5.6)$$

olsun. $H_{[i]}(x)$ ve $H_{[i,j]}(x, y)$ 'nin, λ dolayısıyla N 'ye bağlı olmasına rağmen gösterimimiz, kolaylık olması bakımından bu durumu yansıtmaz (Puri, Sen 1971).

Şimdi her bir $i = 1, \dots, p$ için $E_{N,\alpha}^{(i)} = J_{N(i)}(\alpha/N+1)$ skorlarının olduğu (3.4.5)'te tanımlanmış olan $T_{N_i}^{(k)}$ tesadüfi değişkenleri dikkate alınsın ve $i \neq j = 1, \dots, p$ için

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[J_{N(i)} \left(\frac{N}{N+1} H_{N[i]}(x) \right) J_{N(j)} \left(\frac{N}{N+1} H_{N[j]}(y) \right) - J_{(i)} \left(\frac{N}{N+1} H_{N[i]}(x) \right) J_{(j)} \left(\frac{N}{N+1} H_{N[j]}(y) \right) \right] dF_{N[i,j]}^{(k)}(x,y) = o_p(1), \quad (3.5.7)$$

olsun. Burada tüm $i = 1, \dots, p$ için (3.5.7) aşağıdaki gibi olur:

$$\int_{-\infty}^{\infty} \left[J_{N(i)} \left(\frac{N}{N+1} H_{N[i]}(x) \right) - J_{(i)} \left(\frac{N}{N+1} H_{N[i]}(x) \right) \right]^2 dF_{N(i)}^k(x) = o_p(1). \quad (3.5.8)$$

Sonuç olarak; $0 = \alpha_0 \leq u < \alpha_1^{(i)}, \alpha_1^{(i)} \leq u < \alpha_2^{(i)}, \dots, \alpha_{r_i-1}^{(i)} \leq u < 1 = \alpha_{r_i}^{(i)}$ yarı-açık aralıklarının sınırlı bir kümesi vardır. Öyle ki, $J_{N(i)}(u)$ 'ların bu aralıklardaki sabit değerler olduğu kabul edilir ve tüm $u : \alpha_{j-1}^{(i)} \leq u < \alpha_j^{(i)}$, $j = 1, \dots, r_i$ için;

$$J_{(i)}(u) = J_{(i),j} \quad (3.5.9)$$

ve de $\delta > 0$ olduğu zaman tüm $i = 1, \dots, p$ için;

$$\sum_{j=1}^{r_i} |J_{(i),j}|^{2+\delta} (\alpha_j^{(i)} - \alpha_{j-1}^{(i)}) < \infty \quad (3.5.10)$$

olur. (3.5.9) ve (3.5.10), (i)'ye dayanan medyan testi veya yaygınlık testini de içeren çok geniş bir test sınıfı için sağlanmaktadır. Günümüzdeki uygulamalarda kullanılan testlerin çoğu, tüm bu varsayımları hep birlikte sağlar (Puri, Sen 1971).

Şimdi, $\alpha = 1, \dots, n_k$ ve $k = 1, \dots, c$ için her bir $X_{\alpha}^{(k)}$ aynı $F(x)$ dağılım fonksiyonuna sahip olmak üzere; $F_{[i]}(x)$ ve $F_{[i,j]}(x,y)$ sırasıyla $X_{i\alpha}^{(k)}$ ve $(X_{i\alpha}^{(k)}, X_{j\alpha}^{(k)})$ 'nın marjinal birikimli dağılım fonksiyonu olarak alınsın.

$$v_{ij}(F) = \begin{cases} \int_0^1 J_{(i)}^2(u) du - \mu_i^2; & \mu_i = \int_0^1 J_{(i)}(u) du, & i = j = 1, \dots, p, \\ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} J_{(i)}(F_{[i]}(x)) J_{(j)}(F_{[j]}(y)) dF_{[i,j]}(x, y) - \mu_i \mu_j, & i \neq j = 1, \dots, p, \end{cases} \quad (3.5.11)$$

ve

$$v(F) = ((v_{ij}(F))), \quad i, j = 1, \dots, p \quad (3.5.12)$$

olur.

Ω_0 , $v(F)$ 'nin pozitif tanımlı olduğu, p değişkenli tüm sürekli birikimli dağılım fonksiyonlarının tanımlı olduğu sınıfı temsil etsin ve

$$F_k(x) \in \Omega_0, \quad k = 1, \dots, c \quad (3.5.13)$$

olduğu varsayalım. (3.5.13) şunu belirtir:

$$H(x) = \sum_{k=1}^c \lambda_N^{(k)} F_k(x) \Omega_0. \quad (3.5.14)$$

Ayrıca, $J_{N(i)}$ ile ilgili (3.5.9) ve (3.5.10)'un, $T_{N,i}^{(k)}$ 'nin ortak permütasyon dağılımının asimtotik çoklunormalliği için yeterli olduğu söylenebilir. (3.5.13) yukarıdaki asimtotik dağılımın tekil olmamasını sağlamak için gereklidir. Aynı şekilde L_N 'nin asimtotik dağılımı için de gereklidir. Son olarak; (3.5.7) ve (3.5.8), $V(R_N^*)$ permütasyon kovaryans matrisinin (tesadüfi) $v(H)$ 'ye yakınsaması (olasılıkta) için gereklidir (Puri, Sen 1971).

Teorem 3.5.1

Eğer $V(R_N^*)$ asimtotik olarak tekil olmayan bir matris ise; (3.4.4) ve (3.4.8)'de tanımlanmış olan $p(c-1)$ tane $\{T_{Ni}^{(k)} - \bar{E}_N^{(i)}; i = 1, \dots, p; k = 1, \dots, c-1\}$ tesadüfi değişkeninin ortak permütasyon dağılımı, asimtotik olarak $p(c-1)$ değişkenli normal dağılımdır (Puri, Sen 1971).

Teorem 3.5.2

Eğer (3.5.9) ve (3.5.10) sağlanıyorsa; tüm $F_k(x) \in \Omega$, $k = 1, \dots, c$ olmak üzere (3.5.1)'i sağlayan $\lambda_N^{(1)}, \dots, \lambda_N^{(c)}$ için

$$V(R_N^*) \underset{p}{\sim} v(H) \quad ; \quad N \rightarrow \infty \quad (3.5.15)$$

olur (Puri, Sen 1971).

Bu iki teorem aşağıdaki teoremi ortaya çıkartır (Puri, Sen 1971).

Teorem 3.5.3

Teorem 3.5.2'deki koşulların sağlanması durumunda L_N 'nin permütasyon dağılımı, olasılıkta, asimtotik olarak, $p(c-1)$ serbestlik dereceli Ki-kare dağılımına indirgenir (Puri, Sen 1971).

Şimdi; (3.5.8) için (aslında 3.5.7) bazı koşulların geçerli olduğu gösterilecek. Öncelikle, $\psi_i(x)$ sürekli dağılımından gelen N büyüklüğündeki bir örneğin α . en küçük gözleminin beklenen değerinin $E_{N,\alpha}^{(i)}$ olduğu durum dikkate alınacak. Daha sonra (3.5.8),

$$(1/n_k) \sum_{j=1}^{n_k} [E_{N,\alpha_j}^{(i)} - \xi_{N,\alpha_j}^{(i)}]^2 \quad (3.5.16)$$

şeklinde yazılabilir. Burada $1 \leq \alpha_1 < \dots < \alpha_{n_k} \leq N$, n_k tane belli tamsayıdır. (3.5.16),

$$(N/n_k) \left\{ (1/N) \sum_{\alpha=1}^N [E_{N,\alpha}^{(i)} - \xi_{N,\alpha}^{(i)}]^2 \right\} \quad (3.5.17)$$

değerinden daha küçüktür. Bu yüzden $N \rightarrow \infty$ iken, (3.5.17)'nin sifıra yaklaştığını göstermek yeterli olacaktır. Bunu yapmak için aşağıdaki lemma göz önüne alınır (Puri, Sen 1971).

Lemma 3.5.4

$\Psi(x)$ herhangi bir sürekli birikimli dağılım fonksiyonu olmak üzere:

$$(i) \int x d\Psi(x) = 0, \quad \int x^2 d\Psi(x) < \infty,$$

(ii) $\Psi(x)$, $x = 0$ etrafında simetrik; $\Psi(x)$, her $x \leq 0$ için konveks ve her $x > 0$ için konkavdır.

koşulları sağlandığında;

$$\lim_{N \rightarrow \infty} (1/N) \sum_{i=1}^N [E_{N,i} - \xi_{N,i}]^2 = 0 \quad (3.5.18)$$

olur (Puri, Sen 1971).

Özellikle; eğer $\Psi(x)$, standart normal birikimli dağılım fonksiyonu ise bu fonksiyon $x \leq 0$ için konveks ve $x > 0$ için konkavdır (Puri, Sen 1971).

Ayrıca, (3.5.12)'deki $\nu(F)$ 'nin belirli koşullar altında pozitif tanımlı olduğunun gösterilmesi gerekir. $J_{(i)}(F_{[i]}(x))$, $i = 1, \dots, p$ değişkenleri arasında bir ya da daha fazla doğrusal ilişki olduğu zaman $\nu(F)$ 'nin pozitif tanımlı olması gerekmez. Bu, $F(x_1, \dots, x_p)$ dağılımında noktaların yayılımının $(p-1)$ veya

$$\sum_{i=1}^p c_i J_{(i)}(F_{[i]}(X_i)) = \text{sabit} \quad (3.5.19)$$

denklemlerle (veya birden fazla bunun gibi denklem) daha düşük boyutlu çoklu eğri ile sınırlandırılmasını gerektirir. Bu nedenle, eğer $F(x_1, \dots, x_p)$ birikimli dağılım fonksiyonu tekil değilse yani olasılık kümesi kesinlikle 1'e eşit olan p boyutlu Öklid uzayı E_p 'nin $(p-1)$ veya daha düşük boyutlu alt uzayı yoksa, (3.5.19) olasılıkla yakınsamaz ve buradan $\nu(F)$ pozitif tanımlı olur. Aslında uygulamada F üzerinde önemli bir sınırlama yoktur (Puri, Sen 1971).

Sonuç olarak; iki yığın olması durumunda (3.4.14)'teki L_N , $T_N^{(1)}$ veya $T_N^{(2)}$ kullanılarak şöyle ifade edilebilir:

$$L_N = (n_1 N / n_2) (T_N^{(1)} - \bar{E}_N) V^{-1} (R_N^*) (T_N^{(1)} - \bar{E}_N)' . \quad (3.5.20)$$

Burada $N = n_1 + n_2$ 'dir. Bu da, $n_1 T_N^{(1)} + n_2 T_N^{(2)} = N \bar{E}_N$ olduğunu gösterir (Puri, Sen 1971).

Teorem (3.5.3)'e dayanarak, L_N 'ye bağlı permütasyon test işlemi büyük örnekler için aşağıdaki şekilde sadeleştirilir:

$$\begin{aligned} \text{Eğer } L_N \geq \chi_{\epsilon, p(c-1)}^2, & \quad H_0 \text{ ret,} \\ L_N < \chi_{\epsilon, p(c-1)}^2, & \quad H_0 \text{ kabul,} \end{aligned} \quad (3.5.21)$$

Burada $\chi_{\epsilon, r}^2$, r serbestlik dereceli ve $100(1-\epsilon)\%$ güvenilirlik düzeyine sahip Ki-kare dağılımıdır (Puri, Sen 1971 ve Um, Randles 1998).

Yukarıda bahsedilen testin gücünün özellikleri üzerinde çalışmak için alternatif hipotezlerin uygun sınıfları altında, L_N 'nin koşulsuz dağılımı üzerinde çalışılmıştır.

4. UYGULAMA

İki farklı gruba ait boy, ağırlık ve başarı değerleri gözlenmiş olsun. Birinci grupta 3, ikinci grupta 2 gözlem bulunsun. Gözlem değerleri aşağıdaki tabloda verilmiştir. Boy, ağırlık ve başarı değişkenlerinden en az biri açısından bu iki grup arasında fark olup olmadığını belirleyelim.

	Boy	Ağırlık	Başarı
Grup 1	160	50	3
	180	70	4
	150	60	2
Grup 2	190	65	5
	140	80	1

4.1. Ortalama Vektörlerinin Parametrik Yöntemle Test Edilmesi

Veri matrisi:

$$X = \left[\begin{array}{ccc|cc} 160 & 180 & 150 & 190 & 140 \\ 50 & 70 & 60 & 65 & 80 \\ 3 & 4 & 2 & 5 & 1 \end{array} \right] : X = \begin{bmatrix} \text{Boy} \\ \text{Ağırlık} \\ \text{Başarı} \end{bmatrix}$$

$$H_0 : \begin{bmatrix} \mu_{11} \\ \mu_{12} \\ \mu_{13} \end{bmatrix} = \begin{bmatrix} \mu_{21} \\ \mu_{22} \\ \mu_{23} \end{bmatrix} \text{ hipotezini, } H_1 : \begin{bmatrix} \mu_{11} \\ \mu_{12} \\ \mu_{13} \end{bmatrix} \neq \begin{bmatrix} \mu_{21} \\ \mu_{22} \\ \mu_{23} \end{bmatrix} \text{ hipotezine karşı test etmek}$$

istiyoruz.

Bu testi gerçekleştirmek için öncelikle normallik ve kovaryans matrislerinin eşitliği varsayımlarının geçerli olduğunu kabul ediyoruz.

$$\bar{x}_1 = \begin{bmatrix} 163.333 \\ 60.000 \\ 3.000 \end{bmatrix} \text{ ve } \bar{x}_2 = \begin{bmatrix} 165.000 \\ 72.500 \\ 3.000 \end{bmatrix}; \bar{x} = \begin{bmatrix} 164 \\ 65 \\ 3 \end{bmatrix}; n = \sum_{i=1}^k n_i = 3 + 2 = 5$$

$$S_1 = \begin{bmatrix} 233.333 & 100.000 & 15.000 \\ 100.000 & 100.000 & 5.000 \\ 15.000 & 5.000 & 1.000 \end{bmatrix} \text{ ve } S_2 = \begin{bmatrix} 1250.0 & -375.0 & 100.0 \\ -375.0 & 112.5 & -30.0 \\ 100.0 & -30.0 & 8.0 \end{bmatrix};$$

olarak bulunur. Gruplar arası kareler toplamı matris B,

$$B = \sum_{i=1}^k n_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})' = 3 * \left(\begin{bmatrix} 163.333 \\ 60.000 \\ 3.000 \end{bmatrix} - \begin{bmatrix} 164 \\ 65 \\ 3 \end{bmatrix} \right) * \left(\begin{bmatrix} 163.333 \\ 60.000 \\ 3.000 \end{bmatrix} - \begin{bmatrix} 164 \\ 65 \\ 3 \end{bmatrix} \right)' + 2 * \left(\begin{bmatrix} 165.0 \\ 72.5 \\ 3.0 \end{bmatrix} - \begin{bmatrix} 164 \\ 65 \\ 3 \end{bmatrix} \right) * \left(\begin{bmatrix} 165.0 \\ 72.5 \\ 3.0 \end{bmatrix} - \begin{bmatrix} 164 \\ 65 \\ 3 \end{bmatrix} \right)'$$

$$= \begin{bmatrix} 3.335 & 25.005 & 0.000 \\ 25.005 & 187.500 & 0.000 \\ 0.000 & 0.000 & 0.000 \end{bmatrix}$$

ve grup içi kareler toplamı matrisi W,

$$W = \sum_{i=1}^k (n_i - 1)S_i = (3 - 1) * \begin{bmatrix} 233.333 & 100.000 & 15.000 \\ 100.000 & 100.000 & 5.000 \\ 15.000 & 5.000 & 1.000 \end{bmatrix} + (2 - 1) * \begin{bmatrix} 1250.000 & -375.000 & 100.000 \\ -375.000 & 112.500 & -30.000 \\ 100.000 & -30.000 & 8.000 \end{bmatrix}$$

$$= \begin{bmatrix} 1716.666 & -175.000 & 130.000 \\ -175.000 & 312.500 & -20.000 \\ 130.000 & -20.000 & 10.000 \end{bmatrix} \text{ olmak üzere;}$$

$$W + B = \begin{bmatrix} 1720.001 & -149.995 & 130.000 \\ -149.995 & 500.000 & -20.000 \\ 130.000 & -20.000 & 10.000 \end{bmatrix} \text{ olur. Buradan Wilks lamda istatistiği,}$$

$$\Lambda = \frac{|W|}{|W + B|} = \frac{|W|}{|T|} = \frac{414.85}{16993.6} = 0.024 \text{ olarak elde edilir. } \Lambda \text{ 'ya bağlı test istatistiği}$$

aşağıdaki gibi hesaplanır:

$$L = -[n - 1 - (p + k) / 2] \ln \Lambda = -[5 - 1 - (3 + 2) / 2] \ln(0.024) = 5.595 .$$

L istatistiği $p(k-1)$ serbestlik dereceli Ki-kare dağılımı gösterir. Buradan;

$$L = 5.595 < \chi_{Tablo[3;0.05]}^2 = 7.815 \text{ olduğundan dolayı } H_0 \text{ hipotezi ret edilemez.}$$

Yani; boy, ağırlık ve başarı değişkenlerinden hiç biri açısından iki grup arasında fark yoktur, denir.

4.2. Konum Vektörlerinin Parametrik Olmayan Yöntemle Test Edilmesi

Veri matrisi:

$$X = \left[\begin{array}{ccc|cc} 160 & 180 & 150 & 190 & 140 \\ 50 & 70 & 60 & 65 & 80 \\ 3 & 4 & 2 & 5 & 1 \end{array} \right] : X = \left[\begin{array}{l} \text{Boy} \\ \text{Ağırlık} \\ \text{Başarı} \end{array} \right]$$

Grup indisi (c=2 grup) : $k = 1, 2$;

Değişken indisi (p=3 değişken) : $i = 1, 2, 3$;

1. Gruptaki gözlem sayısı : $n_1 = 3$;

2. Gruptaki gözlem sayısı : $n_2 = 2$;

Toplam gözlem sayısı : $N = n_1 + n_2 = 3 + 2 = 5$

$F(x + \delta_1), F(x + \delta_2)$ olmak üzere;

$H_0 : \delta_1 = \delta_2$ hipotezini, $H_1 : \delta_1 \neq \delta_2$ hipotezine karşı test etmek istiyoruz.

Ölçümlerin her değişkendeki rank değerleri matrisi:

$$R = \left[\begin{array}{ccc|cc} 3 & 4 & 2 & 5 & 1 \\ 1 & 4 & 2 & 3 & 5 \\ 3 & 4 & 2 & 5 & 1 \end{array} \right]$$

Rank değerlerine bağlı skor değerleri matrisi:

$$E = \left[\begin{array}{ccc|cc} 3/6 & 4/6 & 2/6 & 5/6 & 1/6 \\ 1/6 & 4/6 & 2/6 & 3/6 & 5/6 \\ 3/6 & 4/6 & 2/6 & 5/6 & 1/6 \end{array} \right]$$

$$T_{Ni}^{(k)} = (1/n_k) \sum_{\alpha=1}^{n_k} E_{N,R\alpha}^{(i)}, \quad k = 1, \dots, c; \quad i = 1, \dots, p \text{ olmak üzere;}$$

Gruplardaki her değişken için skor değerleri ortalamaları:

1. Grup	2. Grup
$T_{51}^{(1)} = \frac{1}{3} \left(\frac{3}{6} + \frac{4}{6} + \frac{2}{6} \right) = 0.50$	$T_{51}^{(2)} = \frac{1}{2} \left(\frac{5}{6} + \frac{1}{6} \right) = 0.50$
$T_{52}^{(1)} = \frac{1}{3} \left(\frac{1}{6} + \frac{4}{6} + \frac{2}{6} \right) = 0.39$	$T_{52}^{(2)} = \frac{1}{2} \left(\frac{3}{6} + \frac{5}{6} \right) = 0.67$
$T_{53}^{(1)} = \frac{1}{3} \left(\frac{3}{6} + \frac{4}{6} + \frac{2}{6} \right) = 0.50$	$T_{53}^{(2)} = \frac{1}{2} \left(\frac{5}{6} + \frac{1}{6} \right) = 0.50$

Ya da; bu ortalamalar $T_{Ni}^{(k)} = \sum_{\alpha=1}^N E_{N,\alpha}^{(i)} Z_{Ni,\alpha}^{(k)} / n_k$ eşitliğiyle de hesaplanabilir.

1. gruptaki her değişken için skor değerleri ortalamaları:

$$T_{51}^{(1)} = \frac{3}{6} \cdot 1 \cdot \frac{1}{3} + \frac{4}{6} \cdot 1 \cdot \frac{1}{3} + \frac{2}{6} \cdot 1 \cdot \frac{1}{3} + \frac{5}{6} \cdot 0 \cdot \frac{1}{3} + \frac{1}{6} \cdot 0 \cdot \frac{1}{3} = 0.50$$

$$T_{52}^{(1)} = \frac{1}{6} \cdot 1 \cdot \frac{1}{3} + \frac{4}{6} \cdot 1 \cdot \frac{1}{3} + \frac{2}{6} \cdot 1 \cdot \frac{1}{3} + \frac{3}{6} \cdot 0 \cdot \frac{1}{3} + \frac{5}{6} \cdot 0 \cdot \frac{1}{3} = 0.39$$

$$T_{53}^{(1)} = \frac{3}{6} \cdot 1 \cdot \frac{1}{3} + \frac{4}{6} \cdot 1 \cdot \frac{1}{3} + \frac{2}{6} \cdot 1 \cdot \frac{1}{3} + \frac{5}{6} \cdot 0 \cdot \frac{1}{3} + \frac{1}{6} \cdot 0 \cdot \frac{1}{3} = 0.50$$

2. gruptaki her değişken için skor değerleri ortalamaları:

$$T_{51}^{(2)} = \frac{3}{6} \cdot 0 \cdot \frac{1}{2} + \frac{4}{6} \cdot 0 \cdot \frac{1}{2} + \frac{2}{6} \cdot 0 \cdot \frac{1}{2} + \frac{5}{6} \cdot 1 \cdot \frac{1}{2} + \frac{1}{6} \cdot 1 \cdot \frac{1}{2} = 0.50$$

$$T_{52}^{(2)} = \frac{1}{6} \cdot 0 \cdot \frac{1}{2} + \frac{4}{6} \cdot 0 \cdot \frac{1}{2} + \frac{2}{6} \cdot 0 \cdot \frac{1}{2} + \frac{3}{6} \cdot 1 \cdot \frac{1}{2} + \frac{5}{6} \cdot 1 \cdot \frac{1}{2} = 0.67$$

$$T_{53}^{(2)} = \frac{3}{6} \cdot 0 \cdot \frac{1}{2} + \frac{4}{6} \cdot 0 \cdot \frac{1}{2} + \frac{2}{6} \cdot 0 \cdot \frac{1}{2} + \frac{5}{6} \cdot 1 \cdot \frac{1}{2} + \frac{1}{6} \cdot 1 \cdot \frac{1}{2} = 0.50 \text{ olarak bulunur.}$$

$$\bar{E}_N^{(i)} = \sum_{\alpha=1}^N E_{N,\alpha}^{(i)} / N, \quad i = 1, \dots, p \text{ olmak üzere;}$$

Her değişken için beklenen skor değerleri:

$$\bar{E}_N^{(1)} = \left(\frac{3}{6} + \frac{4}{6} + \frac{2}{6} + \frac{5}{6} + \frac{1}{6} \right) \cdot \frac{1}{5} = 0.50$$

$$\bar{E}_N^{(2)} = \left(\frac{1}{6} + \frac{4}{6} + \frac{2}{6} + \frac{3}{6} + \frac{5}{6} \right) \cdot \frac{1}{5} = 0.50$$

$$\bar{E}_N^{(3)} = \left(\frac{3}{6} + \frac{4}{6} + \frac{2}{6} + \frac{5}{6} + \frac{1}{6} \right) \cdot \frac{1}{5} = 0.50 \text{ olarak bulunur.}$$

$$v_{ij}(R_N^*) = \frac{1}{N} \sum_{k=1}^c \sum_{\alpha=1}^{n_k} E_{N\alpha,i}^{(k)} E_{N\alpha,j}^{(q)} - \bar{E}_N^{(i)} \bar{E}_N^{(j)} \text{ olmak üzere;}$$

$$v_{11}(R_N^*) = \frac{1}{5} \left[\left(\frac{3}{6} \right)^2 + \left(\frac{4}{6} \right)^2 + \left(\frac{2}{6} \right)^2 + \left(\frac{5}{6} \right)^2 + \left(\frac{1}{6} \right)^2 \right] - (0.5)^2 = 0.056$$

$$v_{12}(R_N^*) = \frac{1}{5} \left[\left(\frac{3}{6} \right) \left(\frac{1}{6} \right) + \left(\frac{4}{6} \right) \left(\frac{4}{6} \right) + \left(\frac{2}{6} \right) \left(\frac{2}{6} \right) + \left(\frac{5}{6} \right) \left(\frac{3}{6} \right) + \left(\frac{1}{6} \right) \left(\frac{5}{6} \right) \right] - (0.5)^2 = -0.011$$

$$v_{13}(R_N^*) = \frac{1}{5} \left[\left(\frac{3}{6} \right) \left(\frac{3}{6} \right) + \left(\frac{4}{6} \right) \left(\frac{4}{6} \right) + \left(\frac{2}{6} \right) \left(\frac{2}{6} \right) + \left(\frac{5}{6} \right) \left(\frac{5}{6} \right) + \left(\frac{1}{6} \right) \left(\frac{1}{6} \right) \right] - (0.5)^2 = 0.056$$

$$v_{22}(R_N^*) = \frac{1}{5} \left[\left(\frac{1}{6} \right)^2 + \left(\frac{4}{6} \right)^2 + \left(\frac{2}{6} \right)^2 + \left(\frac{3}{6} \right)^2 + \left(\frac{5}{6} \right)^2 \right] - (0.5)^2 = 0.056$$

$$v_{23}(R_N^*) = \frac{1}{5} \left[\left(\frac{1}{6} \right) \left(\frac{3}{6} \right) + \left(\frac{4}{6} \right) \left(\frac{4}{6} \right) + \left(\frac{2}{6} \right) \left(\frac{2}{6} \right) + \left(\frac{3}{6} \right) \left(\frac{5}{6} \right) + \left(\frac{5}{6} \right) \left(\frac{1}{6} \right) \right] - (0.5)^2 = -0.011$$

$$v_{33}(R_N^*) = \frac{1}{5} \left[\left(\frac{3}{6} \right)^2 + \left(\frac{4}{6} \right)^2 + \left(\frac{2}{6} \right)^2 + \left(\frac{5}{6} \right)^2 + \left(\frac{1}{6} \right)^2 \right] - (0.5)^2 = 0.056$$

olarak bulunur.

$V(R_N^*) = ((v_{ij}(R_N^*)))_{i,j=1,\dots,p}$ olmak üzere; yaygınlık matrisi aşağıdaki gibi oluşturulur:

$$V(R_N^*) = \begin{bmatrix} 0.056 & -0.011 & 0.056 \\ -0.011 & 0.056 & -0.011 \\ 0.056 & -0.011 & 0.056 \end{bmatrix}.$$

$$L_N = \sum_{k=1}^c n_k [(T_N^{(k)} - \bar{E}_N) V^{-1} (R_N^*) (T_N^{(k)} - \bar{E}_N)'] ; T_N^{(k)} = (T_{N1}^{(k)}, \dots, T_{Np}^{(k)}) \text{ ve}$$

$\bar{E}_N = (\bar{E}_N^{(1)}, \dots, \bar{E}_N^{(p)})$ olmak üzere;

$$T_N^{(1)} = [0.50 \quad 0.39 \quad 0.50],$$

$$T_N^{(2)} = [0.50 \quad 0.67 \quad 0.50],$$

$$\bar{E}_N = [0.50 \quad 0.50 \quad 0.50],$$

$$(T_N^{(1)} - \bar{E}_N) = [0 \quad -0.11 \quad 0],$$

$$(T_N^{(2)} - \bar{E}_N) = [0 \quad 0.17 \quad 0] \text{ ve}$$

$$V^{-1}(R_N^*) = \begin{bmatrix} 4.688 & 1.875 & 4.688 \\ 1.875 & 18.750 & 1.875 \\ 4.688 & 1.875 & 4.688 \end{bmatrix} \text{ olur.}$$

Test istatistiği aşağıdaki gibi hesaplanır:

$$\begin{aligned} L_N &= 3 * \left(\begin{bmatrix} 0 & -0.11 & 0 \end{bmatrix} \begin{bmatrix} 4.688 & 1.875 & 4.688 \\ 1.875 & 18.750 & 1.875 \\ 4.688 & 1.875 & 4.688 \end{bmatrix} \begin{bmatrix} 0 \\ -0.11 \\ 0 \end{bmatrix} \right) + 2 * \left(\begin{bmatrix} 0 & 0.17 & 0 \end{bmatrix} \begin{bmatrix} 4.688 & 1.875 & 4.688 \\ 1.875 & 18.750 & 1.875 \\ 4.688 & 1.875 & 4.688 \end{bmatrix} \begin{bmatrix} 0 \\ 0.17 \\ 0 \end{bmatrix} \right) \\ &= 3 * [-0.206 \quad -2.063 \quad -0.206] \begin{bmatrix} 0 \\ -0.11 \\ 0 \end{bmatrix} + 2 * [0.319 \quad 3.188 \quad 0.319] \begin{bmatrix} 0 \\ 0.17 \\ 0 \end{bmatrix} = 1.765 \end{aligned}$$

$L_N = 1.765 < \chi_{Tablo[3;0.05]}^2 = 7.815$ olduğu için H_0 hipotezi ret edilemez.

Yani; boy, ağırlık ve başarı değişkenlerinden hiç biri açısından iki grup arasında fark yoktur, denir.

5. SİMÜLASYON ÇALIŞMASI

Parametrik ve parametrik olmayan yöntemlerin iki değişkenli, konumları ve dağılımları farklı iki yığın konum testinin farklı örnek hacimlerinde etkinliklerinin ve performanslarının değerlendirilmesi amacıyla temelde yedi farklı simülasyon çalışmasının sonuçları aşağıda gösterilmiştir.

Örneğin; iki değişkenli, ortalama vektörü 0 (sıfır) ve kovaryans matrisi birim matris (I) olan iki yığından alınan 5'er birimlik örneklerin ortalama vektörlerinin eşitliği 1000 kez test edildiğinde üstteki değerde görüldüğü gibi parametrik yöntemle temel hipotez 64 kez ret edilmiştir. Aynı işlem parametrik olmayan yöntemle yapıldığında temel hipotez 55 kez ret edilmiştir.

1)

$n_1 = n_2$	$N_2\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right); N_2\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$	$N_2\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right); N_2\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$	$N_2\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right); N_2\left(\begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$
5	0.0640	0.3400	0.9060
	0.0550	0.3460	0.8780
10	0.0380	0.7230	1.0000
	0.0470	0.6910	1.0000
20	0.0420	0.9710	1.0000
	0.0520	0.9630	1.0000
30	0.0520	0.9990	1.0000
	0.0570	0.9980	1.0000

Konumları ve yaygınlıkları aynı olan iki Normal dağılımın konum testinde 5, 10 ve 20 gözlem olması durumunda, ret edilme olasılığı 0.05'e daha yakın değerler parametrik olmayan yöntemde gözlenmiştir.

Konumları farklı olan dağılımlar, aynı olduğunda; ret olasılığı parametrik ve parametrik olmayan yöntemlerde benzerlik göstermektedir.

2)

$n_1 = n_2$	$N_2\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right), N_2\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}\right)$	$N_2\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right), N_2\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}\right)$	$N_2\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right), N_2\left(\begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}\right)$
5	0.0510	0.2700	0.7790
	0.0540	0.2790	0.7460
10	0.0580	0.5560	0.9960
	0.0660	0.5410	0.9910
20	0.0500	0.9110	1.0000
	0.0550	0.8940	1.0000
30	0.0450	0.9850	1.0000
	0.0510	0.9800	1.0000

Konumları aynı fakat yaygınlıkları farklı normal dağılımların konum testinde parametrik yöntemin ret olasılıkları 0.05'e daha yakın çıkmıştır.

Konumları farklı olan dağılımlar, aynı olduğunda; ret olasılığı parametrik ve parametrik olmayan yöntemlerde benzerlik göstermektedir.

3)

$n_1 = n_2$	$N_2\left(\begin{bmatrix} 3 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right); \left[\chi_3^2\right]$	$N_2\left(\begin{bmatrix} 3 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right); \left[\chi_4^2\right]$	$N_2\left(\begin{bmatrix} 3 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right); \left[\chi_5^2\right]$
5	0.1100	0.0840	0.2520
	0.0990	0.0750	0.2460
10	0.0840	0.1580	0.5670
	0.1340	0.1170	0.4570
20	0.0910	0.3780	0.9540
	0.2240	0.1730	0.7900
30	0.0640	0.5560	0.9970
	0.2970	0.1840	0.9320

Normal dağılımın Ki-kare dağılımıyla testinde konumları aynı olsa bile Ki-kare dağılımının pozitif tanımlı olması sebebiyle örnek hacimlerinin artışına bağlı olarak parametrik yöntemin ret olasılığı beklenen 0.05 değerine yakınsamakta fakat parametrik olmayan yöntemin beklenen sonuçtan uzaklaştığı görülmektedir.

Normal dağılımın Ki-kare dağılımıyla testinde konumlar farklı olduğunda genel olarak parametrik sonuçların daha etkin olduğu görülmektedir.

4)

$n_1 = n_2$	$N_2\left(\begin{bmatrix} 3 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \begin{bmatrix} \text{Üst}(3) \\ \text{Üst}(3) \end{bmatrix}\right)$	$N_2\left(\begin{bmatrix} 3 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \begin{bmatrix} \text{Üst}(4) \\ \text{Üst}(4) \end{bmatrix}\right)$	$N_2\left(\begin{bmatrix} 3 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \begin{bmatrix} \text{Üst}(5) \\ \text{Üst}(5) \end{bmatrix}\right)$
5	0.1490	0.0890	0.1170
	0.1440	0.0890	0.0870
10	0.1360	0.0790	0.2150
	0.2040	0.0710	0.1070
20	0.1080	0.1450	0.5410
	0.3230	0.0850	0.1610
30	0.0850	0.2560	0.7850
	0.4660	0.0820	0.2010

Normal dağılımın Üstel dağılımla testinde konumları aynı olsa bile Üstel dağılımın pozitif tanımlı olması sebebiyle örnek hacimlerinin artışına bağlı olarak parametrik yöntemin ret olasılığı beklenen 0.05 değerine yakınsamakta fakat parametrik olmayan yöntemin beklenen sonuçtan uzaklaştığı görülmektedir.

Normal dağılımın Üstel dağılımla testinde de konumlar farklı olduğunda genel olarak parametrik sonuçların daha etkin olduğu görülmektedir.

5)

$n_1 = n_2$	$\begin{bmatrix} \chi_3^2 \\ \chi_3^2 \end{bmatrix}, \begin{bmatrix} \chi_3^2 \\ \chi_3^2 \end{bmatrix}$	$\begin{bmatrix} \chi_3^2 \\ \chi_3^2 \end{bmatrix}, \begin{bmatrix} \chi_4^2 \\ \chi_4^2 \end{bmatrix}$	$\begin{bmatrix} \chi_3^2 \\ \chi_3^2 \end{bmatrix}, \begin{bmatrix} \chi_5^2 \\ \chi_5^2 \end{bmatrix}$
5	0.0460	0.0970	0.2020
	0.0540	0.1020	0.2450
10	0.0510	0.1500	0.4370
	0.0600	0.1780	0.4900
20	0.0490	0.2830	0.8150
	0.0540	0.3360	0.8640
30	0.0440	0.4450	0.9340
	0.0400	0.5010	0.9590

Konumları ve yaygınlıkları aynı olan iki Ki-kare dağılımının konum testinde 5, 10 ve 20 gözlem olması durumunda, ret edilme olasılığı 0.05'e daha yakın değerler parametrik olmayan yöntemde gözlenmiştir.

Konumları farklı olan dağılımlar, aynı olduğunda; ret olasılığı parametrik ve parametrik olmayan yöntemlerde benzerlik göstermektedir. Ki-kare dağılımında parametrik olmayan sonuçların daha etkin olduğu görülmektedir.

6)

$n_1 = n_2$	$\begin{bmatrix} \chi_3^2 \\ \chi_3^2 \end{bmatrix}, \begin{bmatrix} \bar{Üst}(3) \\ \bar{Üst}(3) \end{bmatrix}$	$\begin{bmatrix} \chi_3^2 \\ \chi_3^2 \end{bmatrix}, \begin{bmatrix} \bar{Üst}(4) \\ \bar{Üst}(4) \end{bmatrix}$	$\begin{bmatrix} \chi_3^2 \\ \chi_3^2 \end{bmatrix}, \begin{bmatrix} \bar{Üst}(5) \\ \bar{Üst}(5) \end{bmatrix}$
5	0.0390	0.0360	0.0630
	0.0520	0.0460	0.0650
10	0.0480	0.0660	0.1810
	0.0510	0.0620	0.1340
20	0.0470	0.1420	0.4570
	0.0730	0.0810	0.2700
30	0.0560	0.2310	0.6780
	0.0920	0.0970	0.4140

Konumları aynı fakat yaygınlıkları farklı olan ve tanım aralıkları aynı olan Ki-kare ve Üstel dağılımların konumlarının testinde örnek hacimlerin artışına bağlı olarak parametrik yöntemin ret olasılığı beklenen 0.05 değerine yakınsamakta fakat parametrik olmayan yöntemin beklenen sonuçtan uzaklaştığı görülmektedir.

Ki-kare dağılımının Üstel dağılımla testinde konumlar farklı olduğunda genel olarak parametrik sonuçların daha etkin olduğu görülmektedir.

7)

$n_1 = n_2$	$\begin{bmatrix} \bar{Üst}(3) \\ \bar{Üst}(3) \end{bmatrix}, \begin{bmatrix} \bar{Üst}(3) \\ \bar{Üst}(3) \end{bmatrix}$	$\begin{bmatrix} \bar{Üst}(3) \\ \bar{Üst}(3) \end{bmatrix}, \begin{bmatrix} \bar{Üst}(4) \\ \bar{Üst}(4) \end{bmatrix}$	$\begin{bmatrix} \bar{Üst}(3) \\ \bar{Üst}(3) \end{bmatrix}, \begin{bmatrix} \bar{Üst}(5) \\ \bar{Üst}(5) \end{bmatrix}$
5	0.0380	0.0510	0.0740
	0.0470	0.0600	0.0860
10	0.0430	0.1080	0.1860
	0.0590	0.1030	0.1570
20	0.0390	0.1520	0.4590
	0.0480	0.1590	0.2880
30	0.0320	0.2450	0.6940
	0.0400	0.2100	0.4040

Konumları ve yaygınlıkları aynı olan iki Üstel dağılımın konum testinde 5, 10 ve 20 gözlem olması durumunda, ret edilme olasılığı 0.05'e daha yakın değerler parametrik olmayan yöntemde gözlenmiştir.

Konumları farklı olan dağılımlar, aynı olduğunda; ret olasılığı parametrik ve parametrik olmayan yöntemlerde benzerlik göstermektedir. Üstel dağılımda örnek hacmi küçükken parametrik olmayan sonuçların daha etkin olduğu görülmektedir.

6. SONUÇ

Konumları ve yaygınlıkları aynı olan iki dağılımın konum testinde temel hipotezin ret edilme olasılığı 0.05 iken; küçük örnek hacimli testlerde bu olasılığa yakın değerler parametrik olmayan yöntemde gözlenmiştir. Bu durumda gözlem sayısının 30'dan az olması halinde parametrik olmayan yöntemin tercih edilebileceği Simülasyon 1, 5 ve 7 çalışmalarında görülmektedir.

Konumları aynı, fakat yaygınlıkları farklı iki dağılımın konum testinde parametrik yöntemin daha etkili olduğu Simülasyon 2 çalışmasında görülmektedir. Bu çalışmada parametrik yöntemin ret olasılıkları beklenen 0.05 ret etme ihtimaline daha yakındır.

Normal dağılımın Ki-kare ve Üstel dağılımlarla testinde konumları aynı olsa bile Ki-kare ve Üstel dağılımların pozitif tanımlı olması sebebiyle örnek hacimlerinin artışına bağlı olarak parametrik yöntemin ret olasılığı beklenen 0.05 değerine yakınsamakta fakat parametrik olmayan yöntemin beklenen sonuçtan uzaklaştığı görülmektedir. Bu sonuçlar Simülasyon 3 ve 4 çalışmalarında görülmektedir.

Konumları aynı fakat yaygınlıkları farklı olan ve tanım aralıkları aynı olan Ki-kare ve Üstel dağılımların konumlarının testinde örnek hacimlerine bağlı benzer sonuç Simülasyon 6 çalışmasında görülmektedir.

Konumları farklı olan dağılımlar, aynı olduğunda; ret olasılığı parametrik ve parametrik olmayan yöntemlerde benzerlik göstermektedir. Ki-kare dağılımında parametrik olmayan sonuçların daha etkin olduğu görülmektedir. Fakat Üstel dağılımda örnek hacmi küçükken parametrik olmayan sonuçların daha etkin olduğu görülmektedir. Bu sonuçlar Simülasyon 1,2,5 ve 7 çalışmalarında görülmektedir.

Normal dağılımın Ki-kare ve Üstel dağılımlarla testinde ve konumlar farklı olduğunda genel olarak parametrik sonuçların daha etkin olduğu Simülasyon 3, 4 ve 6'dan görülmektedir.

Sonuç olarak gözlem sayımız 30'dan küçük olduğunda parametrik olmayan testler daha güçlüdür.

7. EKLER

EK 7.1. Çok Değişkenli Normallik Testi SPSS-MP Program Kodu (değişken sayısı=DS için) :

```

set mxloop=10000.
matrix.
get y /missing=accept /sysmis=100000.
compute ds=DS.                                /* Değişken sayısı*/
compute g=ncol(y)/ds.
compute gsn=nrow(y).
compute say=1.
compute dd=1.
compute k=dd.
compute gsnn=make(g,1,0).
loop i=k to g.
  loop j=1 to gsn.
    do if (y(j,k)<100000).
      compute gsnn(say)=gsnn(say)+1.
    end if.
  end loop.
  compute k=k+ds.
  compute say=say+1.
end loop.
compute gsn=t(kroneker(gsnn,make(ds,1,1))).
compute k=dd.
loop i=1 to g.
  compute veri=make(gsnn(i),ds,0).
  loop j=k to ds*i.
    loop m=1 to gsn(j).
      do if (y(m,j)<100000).
        compute veri(m,j-ds*(i-1))=y(m,j).
      end if.
    end loop.
  end loop.
end loop.
print veri.
compute x=veri.
compute n=nrow(x).
compute koltop=csum(x).
compute ort=koltop/n.
compute xort=make(n,ds,0).
  loop r=1 to ds.
    loop j=1 to n.
      compute xort(j,r)=ort(r).
    end loop.
  end loop.
compute fark=x-xort.
compute S=t(fark)*fark/(n-1).
print ort /format=f8.3.
print S /format=f8.3.
compute Sters=inv(S).
print Sters /format=f8.3.
compute mkare=make(n,1,0).
  loop j=1 to n.
    compute xj=make(ds,1,0).
    loop r=1 to ds.

```

```

        compute xj(r,1)=x(j,r).
    end loop.
    compute mkare(j,1)=t(xj-t(ort))*Sters*(xj-t(ort)).
end loop.
print mkare /format=f5.3.
compute smk=mkare.
loop r=1 to n.
loop j=r+1 to n.
do if (smk(r)>smk(j)).
    compute a=smk(r).
    compute smk(r)=smk(j).
    compute smk(j)=a.
end if.
end loop.
end loop.
print smk /format=f5.3.
compute pi=make(n,1,0).
compute kk=pi.
loop j=1 to n.
    compute pi(j)=(j-1/2)/n.
    compute p=pi(j).
    compute sd=ds.
loop r=1 to 100000.
    compute kkd=r/1000.
    compute ip=chicdf(kkd,sd).
    compute fark=abs(ip-p).
    do if (fark<0.001).
        compute kd=kkd.
        break.
    end if.
end loop.
compute kk(j)=kd.
end loop.
print pi /format=f5.3.
print kk /format=f5.3.
compute ssd=ds.
compute kontur=0.5.
loop sr=1 to 100000.
    compute skkd=sr/1000.
    compute sip=chicdf(skkd,ssd).
    compute fark=abs(sip-kontur).
    do if (fark<0.001).
        compute kkkd=skkd.
        break.
    end if.
end loop.
print kkkd /format=f12.5.
compute sonuc=0. /* %50'lik konturun içine düşen smk değerlerinin sayısı */
loop r=1 to n.
do if (smk(r)<=kkkd).
    compute sonuc=sonuc+1.
end if.
end loop.
print sonuc.
compute dd=dd+ds.
compute k=dd.
end loop.
end matrix.

```

EK 7.2. Kovaryans Matrislerinin Eşitliği Testi SPSS-MP Program Kodu (değişken sayısı=DS için):

```

matrix.

get x /missing=accept /sysmis=100000.
print x.

compute d=DS.                                /* Değişken sayısı      */
compute g=ncol(x)/d.
compute n=nrow(x).

compute ut=0.
compute at=0.
compute say=1.
compute k=1.
compute nn=make(g,1,0).

loop i=k to g.
loop j=1 to n.
do if (x(j,k)<100000).
  compute nn(say)=nn(say)+1.
end if.
end loop.
compute k=k+d.
compute say=say+1.
end loop.
*print nn.                                    /* Gruplardaki veri sayısı */
compute n=t(kroneker(nn,make(d,1,1))).
*print n.                                     /* Değişkenlerdeki veri sayısı */

compute k=1.
compute mikit=0.

loop i=1 to g.
  compute veri=make(nn(i),d,0).
  loop j=k to d*i.
  loop m=1 to n(j).
  do if (x(m,j)<100000).
    compute veri(m,j-d*(i-1))=x(m,j).
  end if.
  end loop.
end loop.

compute ortvek=csum(veri)/nn(i).
*print ortvek.
compute ortmat=kroneker(make(nn(i),1,1),ortvek).
compute fark=veri-ortmat.
compute s=t(fark)*fark/(nn(i)-1).
print s.

compute ut=ut+(nn(i)-1)*s.
compute mikit=mikit+((nn(i)-1)*ln(det(s))).
compute k=k+d.

end loop.

```

```
compute at=msum(nn)-g.  
print at.
```

```
compute s=ut/at.  
print s.
```

```
compute sters=inv(s).  
print sters.
```

```
compute dets=det(s).  
print dets /format=f12.3.
```

```
compute m=(at*ln(dets)-mikit).  
print m.  
compute b=0.  
loop i=1 to g.  
  compute b=b+(1/(nn(i)-1)).  
end loop.  
compute cust=2*d**2+3*d-1.  
compute calt=6*(d+1)*(g-1).  
compute c=1-cust/calt*(b-(1/at)).  
print c.
```

```
compute mc=m*c.  
print mc.
```

```
end matrix.
```

EK 7.3. Çok Değişkenli Tek Yığın Ortalaması Önemlilik Testi ve Farklı Değişken(ler)in Belirlenmesi için SPSS-MP Program Kodu (Hipotezde kabul edilen ortalama vektörü m_0 ve F tablo, t tablo ve Ki-kare tablo değerleri için):

```

matrix.
compute m0={68,40}.          /* Hipotezde kabul edilen ortalama vektörü */
get x.
compute p=ncol(X).          /* Değişken sayısı          */
compute n=nrow(X).          /* Gözlem sayısı          */
compute koltop=csum(x).
compute ort=koltop/n.        /* Örnek ortalama vektörü */
compute xort=make(n,p,0).
    loop i=1 to p.
        loop j=1 to n.
            compute xort(j,i)=ort(i).
        end loop.
    end loop.
compute fark=x-xort.
compute S=t(fark)*fark/(n-1). /* Örnek kovaryans matrisi */
print ort /format=f5.2.
print S /format=f8.3.
compute Sters=inv(S).
print Sters /format=f5.3.
compute fark2=fark-xort.
compute Tkare=n*fark2*Sters*t(fark2).
print Tkare /format=f10.2.
compute fhdeg=Tkare*(n-p)/((n-1)*p).
compute pdeg=1-fcdf(fhdeg,p,n-p).
print pdeg /format=f5.3.
/* FARKLI DEĞİŞKENLERİN BELİRLENMESİ */
/* I. Eşanlı güven aralıkları */
compute ftab=3.42.          /* p,n-p,alfa=0.05*/
    loop i=1 to p.
        compute altdeg=ort(i)-((n-1)*p/(n-p)*ftab)**0.5*(s(i,i)/n)**0.5.
        compute ustdeg=ort(i)+((n-1)*p/(n-p)*ftab)**0.5*(s(i,i)/n)**0.5.
    print altdeg /format=f5.2.
    print ustdeg /format=f5.2.
    end loop.

```

```
/* II. Bonferroni Yöntemi */
```

```
compute ttab=2.797. /* n-1;alfa/2p */
```

```
loop i=1 to p.
```

```
compute altdeg=ort(i)-ttab*(s(i,i)/n)**(0.5).
```

```
compute üstdeg=ort(i)+ttab*(s(i,i)/n)**(0.5).
```

```
print altdeg /format=f5.2.
```

```
print üstdeg /format=f5.2.
```

```
end loop.
```

```
/* III. Gözlem sayısı fazla olduğunda */
```

```
compute kikrtab=5.991. /* p;alfa */
```

```
loop i=1 to p.
```

```
compute altdeg=ort(i)-(kikrtab)**(0.5)*(s(i,i)/n)**(0.5).
```

```
compute üstdeg=ort(i)+(kikrtab)**(0.5)*(s(i,i)/n)**(0.5).
```

```
print altdeg /format=f5.2.
```

```
print üstdeg /format=f5.2.
```

```
end loop.
```

```
end matrix.
```

EK 7.4. Bağımsız İki Grubun Ortalama Vektörlerinin Karşılaştırılmasına İlişkin Parametrik Hipotez Testi ve Farklı Değişken(ler)in Belirlenmesi için SPSS-MP Program Kodu- Kovaryans Matrisleri Eşit Olduğunda- (değişken sayısı=DS ve F tablo değeri için):

```

matrix.

get x /missing=accept /sysmis=100000.
print x.

compute d=DS.                                /* Değişken sayısı */
compute g=ncol(x)/d.
compute n=nrow(x).
compute ovek=make(2,d,0).
compute ut=0.
compute at=0.
compute say=1.
compute k=1.
compute nn=make(g,1,0).

loop i=k to g.
loop j=1 to n.
  do if (x(j,k)<100000).
    compute nn(say)=nn(say)+1.
  end if.
end loop.
compute k=k+d.
compute say=say+1.
end loop.
print nn.                                    /* Gruplardaki veri sayısı */
compute n=t(kroneker(nn,make(d,1,1))).
print n.                                    /* Değişkenlerdeki veri sayısı */

compute k=1.
compute mikit=0.

loop i=1 to g.
  compute veri=make(nn(i),d,0).
  loop j=k to d*i.
    loop m=1 to n(j).
      do if (x(m,j)<100000).
        compute veri(m,j-d*(i-1))=x(m,j).
      end if.
    end loop.
  end loop.
compute ortvek=csum(veri)/nn(i).
print ortvek /format=f12.3.
compute ovek(i,:)=ortvek.
compute ortmat=kroneker(make(nn(i),1,1),ortvek).
compute fark=veri-ortmat.
compute s=t(fark)*fark/(nn(i)-1).
print s /format=f12.4.

compute ut=ut+(nn(i)-1)*s.
compute mikit=mikit+((nn(i)-1)*ln(det(s))).
compute k=k+d.
end loop.

```

```

print ovek.

compute at=msum(nn)-g.
*print at.

compute sortak=ut/at.
print sortak /format=f12.4.

compute sortters=inv(sortak).
print sortters /format=f12.4.

compute ortfark=t(ovek(1,:)-ovek(2,:)).
print ortfark.

compute tkare=((nn(1)*nn(2))/(nn(1)+nn(2)))*t(ortfark)*sortters*ortfark.
print tkare /format=f10.3.

compute fhdeg=((nn(1)+nn(2)-d-1)/((nn(1)+nn(2)-2)*d))*tkare.
compute pdeg=1-fcdf(fhdeg,d,nn(1)+nn(2)-d-1).
print pdeg /format=f5.3.

/* I. Eşanlı güven aralıkları */
compute ftab=2.84.          /* d,nn(1)+nn(2)-d-1,alfa=0.05*/
loop i=1 to d.
    compute altdeg=ortfark(i)-(((nn(1)+nn(2))/(nn(1)*nn(2)))*sortak(i,i)*(((nn(1)+nn(2)-
2)*d)/(nn(1)+nn(2)-d-1))*ftab)**0.5.
    compute ustdeg=ortfark(i)+(((nn(1)+nn(2))/(nn(1)*nn(2)))*sortak(i,i)*(((nn(1)+nn(2)-
2)*d)/(nn(1)+nn(2)-d-1))*ftab)**0.5.
    print altdeg /format=f6.3.
    print ustdeg /format=f6.3.
end loop.

end matrix.

```

EK 7.5. Bağımsız İki Grubun Ortalama Vektörlerinin Karşılaştırılmasına İlişkin Parametrik Hipotez Testi ve Farklı Değişken(ler)in Belirlenmesi için SPSS-MP Program Kodu- Kovaryans Matrisleri Eşit Olmadığında-(değişken sayısı=DS ve Ki-kare tablo değeri için):

```

set mxloop=100000.
matrix.

get x /missing=accept /sysmis=100000.
print x.

compute d=DS.                                /* Değişken sayısı */
compute g=ncol(x)/d.
compute n=nrow(x).
compute ovek=make(2,d,0).
compute sortak=0.
compute say=1.
compute k=1.
compute nn=make(g,1,0).

loop i=k to g.
loop j=1 to n.
  do if (x(j,k)<100000).
    compute nn(say)=nn(say)+1.
  end if.
end loop.
compute k=k+d.
compute say=say+1.
end loop.
print nn.                                     /* Gruplardaki veri sayısı */
compute n=t(kroneker(nn,make(d,1,1))).
print n.                                     /* Değişkenlerdeki veri sayısı */

compute k=1.
compute mikit=0.

loop i=1 to g.
  compute veri=make(nn(i),d,0).
  loop j=k to d*i.
  loop m=1 to n(j).
  do if (x(m,j)<100000).
    compute veri(m,j-d*(i-1))=x(m,j).
  end if.
  end loop.
end loop.

compute ortvek=csum(veri)/nn(i).
print ortvek /format=f12.3.
compute ovek(i,:)=ortvek.
compute ortmat=kroneker(make(nn(i),1,1),ortvek).
compute fark=veri-ortmat.
compute s=t(fark)*fark/(nn(i)-1).
print s /format=f12.4.

compute sortak=sortak+(s/nn(i)).
compute mikit=mikit+((nn(i)-1)*ln(det(s))).
compute k=k+d.

```

```
end loop.
print ovek.

print sortak /format=f12.4.

compute sortters=inv(sortak).
print sortters /format=f12.4.

compute ortfark=t(ovek(1,:)-ovek(2,:)).
print ortfark.

compute tkare=t(ortfark)*sortters*ortfark.
print tkare /format=f10.3.

compute kktab=5.991.                                     /* d,alfa=0.05*/
print kktab /format=f5.3.

/* I. Eşanlı güven aralıkları */

loop i=1 to d.
    compute altdeg=ortfark(i)-(kktab*sortak(i,i))**0.5.
    compute ustdeg= ortfark(i)+(kktab*sortak(i,i))**0.5.
    print altdeg /format=f6.3.
    print ustdeg /format=f6.3.
end loop.

end matrix.
```

EK 7.6. Bağımsız k Grubun Ortalama Vektörlerinin Karşılaştırılmasına İlişkin Parametrik Hipotez Testi SPSS-MP Program Kodu (değişken sayısı=DS ve t-tablo değeri):

```

matrix.

get x /missing=accept /sysmis=100000.
*print x.

compute d=DS.                                /* Değişken sayısı */
compute g=ncol(x)/d.
compute n=nrow(x).

compute say=1.
compute k=1.
compute tsum=0.
compute tn=0.
compute nn=make(g,1,0).

loop i=k to g.
loop j=1 to n.
do if (x(j,k)<100000).
    compute nn(say)=nn(say)+1.
end if.
end loop.
compute k=k+d.
compute say=say+1.
end loop.
print nn.                                    /* Gruplardaki veri sayısı */
compute n=t(kroneker(nn,make(d,1,1))).
print n.                                     /* Değişkenlerdeki veri sayısı */

compute k=1.
compute b=0.
compute w=0.

compute ovek=make(g,d,0).

loop i=1 to g.
    compute veri=make(nn(i),d,0).
    loop j=k to d*i.
        loop m=1 to n(j).
            do if (x(m,j)<100000).
                compute veri(m,j-d*(i-1))=x(m,j).
            end if.
        end loop.
    end loop.

compute ortvek=csum(veri)/nn(i).
print ortvek.
compute ovek(i,:)=ortvek.
compute ortmat=kroneker(make(nn(i),1,1),ortvek).
compute fark=veri-ortmat.
compute s=t(fark)*fark/(nn(i)-1).
print s.
/* Wilks lamda istatistiği */
compute w=w+(nn(i)-1)*s.

```

```

compute k=k+d.
compute tsum=tsum+csum(veri).
compute tn=tn+nn(i).

end loop.

compute ovektor=t(ovek).
print ovektor.

compute govek=tsum/tn.

loop i=1 to g.
compute b=b+nn(i)*t(ovek(i,:)-govek)*(ovek(i,:)-govek).
end loop.
print govek.
print b.
print w /format=f6.3.

compute bartw=b+w.
print bartw.
compute detw=det(w).
compute detbw=det(bartw).
print detw.
print detbw.
compute wlamda=detw/detbw.
print wlamda.
/* Ki-Kare */
compute kikhesap=-(csum(nn)-1-(d+g)/2)*ln(wlamda).
print kikhesap.
compute kiksder=d*(g-1).
print kiksder.

/* I. Eşanlı güven aralıkları */

compute ttab=2.7695.          /* n-g,alfa/(d*g*(g-1)) */

loop i=1 to d.
loop k=1 to g.
loop j=k+1 to g.
    compute altdeg=(ovektor(i,k)-ovektor(i,j))-(ttab*(( w(i,i)/(msum(nn)-
g))*((1/n(:,k))+1/n(:,j))))**(1/2)).
    compute ustdeg=(ovektor(i,k)-ovektor(i,j))+ (ttab*(( w(i,i)/(msum(nn)-
g))*((1/n(:,k))+1/n(:,j))))**(1/2)).
    print altdeg.
    print ustdeg.
end loop.
end loop.
end loop.

end matrix.

```

EK 7.7. k Grubun Ortalama Vektörlerinin Karşılaştırılmasına İlişkin Parametrik Hipotez Testi SPSS-MP Program Kodu (Örnek hacimleri n girilmeli):

```

matrix.                                /* k GRUP ORTALAMASININ PARAMETRİK TESTİ */
get y /missing=accept /sysmis=100000.
compute x=t(y).                          /* Veri matrisi */
*print x.
compute n={3,2}.                          /* ****Grupların sırasıyla örnek hacimleri girilmelidir.**** */
compute b=0.
compute w=0.
compute j=1.
compute jj=0.
compute gort=rsum(x)/msum(n).              /* Genel ortalama vektörü */
*print gort.
loop i=1 to ncol(n).
compute jj=jj+n(i).
compute a=x(:,j:jj).                      /* Grup veri matrisi (1. grup, 2. grup, ...) */
compute j=j+n(i).
print a.
compute ort=rsum(a)/n(i).                  /* i. gruba ilişkin ortalama vektörü */
compute s=(a*t(a)/n(i)-ort*t(ort))*n(i)/(n(i)-1). /* i. gruba ilişkin kovaryans matrisi */
*print i.
print ort.
print s.
compute b=b+n(i)*(ort-gort)*t(ort-gort). /* B matrisinin hesaplanması */
compute w=w+(n(i)-1)*s.                   /* W matrisinin hesaplanması */
end loop.
*print b.
*print w.
compute lamda=det(w)/det(w+b).            /* Wilks lamda istatistiği */
*print lamda /format=f8.3.
compute l=- (msum(n)-1-(nrow(x)+ncol(n))/2)*ln(lamda). /* L hesap değeri */
*print l /format=f12.3.
compute sder=nrow(x)*(ncol(n)-1).        /* Serbestlik derecesi */
*print sder.
compute pdegeri=1-chicdf(l,sder).          /* L değerine karşılık gelen p-değeri */
*print pdegeri/format=f6.3.
compute sonuc={1,sder,pdegeri}.
print sonuc /clabels=list,sder,pdegeri /format=f6.3. /* Sonuç:L-istatistiği,serbestlik derecesi,p-değeri*/
end matrix.

```

EK 7.8. k Grubun Ortalama Vektörlerinin Karşılaştırılmasına İlişkin Parametrik Olmayan Hipotez Testi SPSS-MP Program Kodu (Örnek hacimleri n girilmeli):

```

matrix.                                     /* k GRUP
ORTALAMASININ PARAMETRİK OLMAYAN TESTİ */
get y /missing=accept /sysmis=100000.
compute x=t(y).                             /* Veri matrisi */
*print x.
compute ggsay={3,2}.                         /* ****Grupların sırasıyla örnek hacimleri girilmelidir.**** */
compute gsay=ncol(ggsay).                   /* Grup sayısı */
compute d=nrow(x).                          /* Değişken sayısı */
compute n=ncol(x).                           /* Tüm gruptaki gözlem
sayılarının toplamı */
compute sat=nrow(x).                         /* X matrisinin satır sayısı */
compute sut=ncol(x).                        /* X matrisinin sütün sayısı */
compute mat1=make(sut,3,0).                 /* Rank matrisi R'yi bulmak
için kullanılan bir matris */
compute r=make(sat,sut,0).                  /* Rank değerleri matrisi=R */
loop i=1 to sat.
  loop k=1 to sut.
    compute mat1(k,3)=x(i,k).
    compute mat1(k,2)=k.
    compute mat1(k,1)=k.
  end loop.
  compute mat2=mat1.                         /* Rank matrisi R'yi bulmak için kullanılan bir matris */
  loop k1=1 to sut-1.
    loop k2=k1+1 to sut.
      do if mat2(k1,3)>mat2(k2,3).
        compute a= mat2(k1,3).
        compute mat2(k1,3)=mat2(k2,3).
        compute mat2(k2,3)=a.
        compute a=mat2(k1,2).
        compute mat2(k1,2)=mat2(k2,2).
        compute mat2(k2,2)=a.
      end if.
    end loop.
  end loop.
  compute mat3=mat2.                         /* Rank matrisi R'yi bulmak için kullanılan bir matris */
  loop k1=1 to sut-1.
    loop k2=k1+1 to sut.
      do if mat3(k1,2)>mat3(k2,2).
        compute a=mat3(k1,2).
        compute mat3(k1,2)=mat3(k2,2).
        compute mat3(k2,2)=a.
        compute a=mat3(k1,1).
        compute mat3(k1,1)=mat3(k2,1).
        compute mat3(k2,1)=a.
        compute a=mat3(k1,3).
        compute mat3(k1,3)=mat3(k2,3).
        compute mat3(k2,3)=a.
      end if.
    end loop.
  end loop.
end loop.

```

```

loop j=1 to sut.
compute r(i,j)=mat3(j,1).
end loop.
end loop.
*print r.
compute e=(1/(n+1))*(r).          /* Rank değerlerine bağlı skor değerleri matrisi=E */
*print e /format=f8.3.
compute tm=make(d,gsay,0).        /* Her sütünü bir grubun her değişken için skor değerlerinin
ortalamasını ifade eden matris */
compute k=1.
compute s=0.
loop i=1 to d.
    loop j=1 to n.
        compute tm(i,k)=tm(i,k)+e(i,j).
        compute s=s+1.
        do if (s=gsay(k)).
            compute tm(i,k)=tm(i,k)/gsay(k).
            compute k=k+1.
            compute s=0.
        end if.
    end loop.
    compute k=1.
end loop.
*print tm/format=f8.3.
compute ec=rsum(e)/n.             /* E çizgi */
*print ec.
compute s=(1/n)*e*t(e)-ec*t(ec).  /* Yaygınlık matrisi */
*print s/format=f6.3.
compute dets=det(s).             /* Yaygınlık matrisinin determinanı */
*print dets.
do if (dets<0.01).
compute sters=ginv(s).
else.
compute sters=inv(s).
end if.
*print sters/format=f8.3.         /* Yaygınlık matrisinin
(genelleştirilmiş) tersi */
compute l=0.
loop k=1 to gsay.
compute l=l+gsay(k)*(t(tm(:,k)-ec)*sters*(tm(:,k)-ec)). /* Ln değeri */
end loop.
compute sder=d*(gsay-1).         /* Serbestlik derecesi */
compute pdegeri=1-chicdf(l,sder). /* Ln değerine karşılık gelen p-değeri */
compute sonuc={l,sder,pdegeri}.
print sonuc /clabels=list,sder,pdegeri /format=f6.3. /* Sonuç: L-istatistiği,ser. derecesi,p-değeri*/
end matrix.

```

EK 7.9. Normal Dağılımlı İki Grubun Konum Testi için Parametrik ve Parametrik Olmayan Simülasyon (Kovaryans Matrisleri Eşit)

```

set mxloop=100000.
matrix.
compute tp=0. /* Parametrik test denemeleri sonucu ret oranı */
compute tnp=0. /* Parametrik olmayan test denemeleri sonucu ret oranı */
loop deneme=1 to 1000. /* Deneme sayısı */
compute n1=5. /* *** Her bir gruptaki gözlem sayısı girilmelidir *** */
compute ort={0;0}. /* 1. Grup- 2 değişkenli normal dağılım için */
compute km={1,0;0,1}.
compute x11=uniform(n1,1).
compute x12=uniform(n1,1).
compute z1=(-2*ln(x11))&*&0.5)&*sin(6.28*x12)).
compute x21=uniform(n1,1).
compute x22=uniform(n1,1).
compute z2=(-2*ln(x21))&*&0.5)&*sin(6.28*x22)).
compute z={z1,z2}.
compute ckm=chol(km).
compute g1=t((t(ckm)*t(z))+(kroneker(make(1,n1,1),ort))). /* Grup 1 */

compute ort={0;0}. /* 2. Grup- 2 değişkenli normal dağılım için */
compute km={1,0;0,1}.
compute x11=uniform(n1,1).
compute x12=uniform(n1,1).
compute z1=(-2*ln(x11))&*&0.5)&*sin(6.28*x12)).

compute x21=uniform(n1,1).
compute x22=uniform(n1,1).
compute z2=(-2*ln(x21))&*&0.5)&*sin(6.28*x22)).

compute z={z1,z2}.
compute ckm=chol(km).
compute g2=t((t(ckm)*t(z))+(kroneker(make(1,n1,1),ort))). /* Grup 2 */
compute y={g1,g2}.
*print y.
compute x=t(y).
*print x.
compute n={n1,n1}.
compute b=0.
compute w=0.
compute j=1.
compute jj=0.
compute gort=rsum(x)/msum(n).
*print gort.
loop i=1 to ncol(n).
compute jj=jj+n(i).
compute a=x(:,j:jj).
compute j=j+n(i).
*print a.
compute ort=rsum(a)/n(i).
compute s=(a*t(a)/n(i)-ort*t(ort))*n(i)/(n(i)-1).
*print ort.
*print s.
compute b=b+n(i)*(ort-gort)*t(ort-gort).
compute w=w+(n(i)-1)*s.

```

```

end loop.
compute lamda=det(w)/det(w+b).
compute sderpar=nrow(x)*(ncol(n)-1).
compute lpar=- (msum(n)-1-(nrow(x)+ncol(n))/2)*ln(lamda).
compute pdegpar=1-chicdf(lpar,sderpar).
compute kararpar=0.
do if (pdegpar<0.05).
compute kararpar=1.
end if.
compute x=t(y).
*print x.
compute ggsay=n.
compute gsay=ncol(ggsay).
compute d=nrow(x).
compute n=ncol(x).
compute sat=nrow(x).
compute sut=ncol(x).
compute mat1=make(sut,3,0).
compute r=make(sat,sut,0).
loop i=1 to sat.
  loop k=1 to sut.
  compute mat1(k,3)=x(i,k).
  compute mat1(k,2)=k.
  compute mat1(k,1)=k.
  end loop.
  compute mat2=mat1.
  loop k1=1 to sut-1.
    loop k2=k1+1 to sut.
    do if mat2(k1,3)>mat2(k2,3).
    compute a= mat2(k1,3).
    compute mat2(k1,3)=mat2(k2,3).
    compute mat2(k2,3)=a.
    compute a=mat2(k1,2).
    compute mat2(k1,2)=mat2(k2,2).
    compute mat2(k2,2)=a.
    end if.
  end loop.
end loop.
compute mat3=mat2.
loop k1=1 to sut-1.
  loop k2=k1+1 to sut.
  do if mat3(k1,2)>mat3(k2,2).
  compute a=mat3(k1,2).
  compute mat3(k1,2)=mat3(k2,2).
  compute mat3(k2,2)=a.
  compute a=mat3(k1,1).
  compute mat3(k1,1)=mat3(k2,1).
  compute mat3(k2,1)=a.
  compute a=mat3(k1,3).
  compute mat3(k1,3)=mat3(k2,3).
  compute mat3(k2,3)=a.
  end if.
end loop.
end loop.
loop j=1 to sut.
compute r(i,j)=mat3(j,1).
end loop.
end loop.

```

```

*print r.
compute e=(1/(n+1))*(r).
*print e /format=f8.3.
compute tm=make(d,gsay,0).
compute k=1.
compute s=0.
loop i=1 to d.
    loop j=1 to n.
        compute tm(i,k)=tm(i,k)+e(i,j).
        compute s=s+1.
        do if (s=ggsay(k)).
            compute tm(i,k)=tm(i,k)/ggsay(k).
            compute k=k+1.
            compute s=0.
        end if.
    end loop.
    compute k=1.
end loop.
*print tm/format=f8.3.
compute ec=rsum(e)/n.
*print ec.
compute s=(1/n)*e*t(e)-ec*t(ec).
*print s/format=f6.3.
compute dets=det(s).
do if (dets<0.01).
    compute sters=ginv(s).
else.
    compute sters=inv(s).
end if.
*print sters/format=f8.3.
compute sderpolm=d*(gsay-1).
*print sderpolm.
compute lparolm=0.
loop k=1 to gsay.
    compute lparolm=lparolm+ggsay(k)*(t(tm(:,k)-ec)*sters*(tm(:,k)-ec)).
end loop.
*print lparolm /format=f6.3.
compute pdparolm=1-chicdf(lparolm,sderpolm).
*print pdparolm /format=f6.3.
compute kparolm=0.
do if (pdparolm<0.05).
    compute kparolm=1.
end if.
*print kparolm.
compute sonmat={lpar,sderpar,pdegpar,kararpar;lparolm,sderpolm,pdparolm,kparolm}.
*print sonmat /title= "Sonuçlar" /format=f8.3 /rlabels=par,npar /clabels=list,sder,pdeg,karar .
compute tp=tp+kararpar.
compute tnp=tnp+kparolm.
end loop.
compute deneme=deneme-1.
compute tp=tp/deneme.
compute tnp=tnp/deneme.
print tp /format=f8.4.
print tnp /format=f8.4.
end matrix.

```

EK 7.10. Normal Dağılımlı İki Grubun Konum Testi için Parametrik ve Parametrik Olmayan Simülasyon (Kovaryans Matrisleri Eşit Değil)

```

set mxloop=100000.
matrix.
compute tp=0.                                /* Parametrik test denemeleri sonucu ret oranı */
compute tnp=0.                                /* Parametrik olmayan test denemeleri sonucu ret oranı */
loop deneme=1 to 1000.                        /* Deneme sayısı */
compute n1=5.                                 /* *** Her bir gruptaki gözlem sayısı girilmelidir *** */
compute ort={0;0}.                            /* 1. Grup- 2 değişkenli normal dağılım için */
compute km={1,0;0,1}.
compute x11=uniform(n1,1).
compute x12=uniform(n1,1).
compute z1=((-2*ln(x11))&*&0.5)&*sin(6.28*x12)).
*print z1.
compute x21=uniform(n1,1).
compute x22=uniform(n1,1).
compute z2=((-2*ln(x21))&*&0.5)&*sin(6.28*x22)).
*print z2.
compute z={z1,z2}.
*print z.
compute ckm=chol(km).
compute g1=t((t(ckm)*t(z))+kroneker(make(1,n1,1),ort))). /* Grup 1 */

compute ort={0;0}.                            /* 2. Grup- 2 değişkenli normal dağılım için */
compute km={1,0;0,4}.
compute x11=uniform(n1,1).
compute x12=uniform(n1,1).
compute z1=((-2*ln(x11))&*&0.5)&*sin(6.28*x12)).

compute x21=uniform(n1,1).
compute x22=uniform(n1,1).
compute z2=((-2*ln(x21))&*&0.5)&*sin(6.28*x22)).

compute z={z1,z2}.
compute ckm=chol(km).
compute g2=t((t(ckm)*t(z))+kroneker(make(1,n1,1),ort))). /* Grup 2 */

compute y={g1;g2}.
*print y.
compute x=t(y).
*print x.
compute n={n1,n1}.
compute b=0.
compute w=0.
compute j=1.
compute jj=0.
compute gort=rsum(x)/msum(n).
*print gort.
loop i=1 to ncol(n).
compute jj=jj+n(i).
compute a=x(:,j:jj).
compute j=j+n(i).
*print a.
compute ort=rsum(a)/n(i).
compute s=(a*t(a)/n(i)-ort*t(ort))*n(i)/(n(i)-1).

```

```

*print ort.
*print s.
compute b=b+n(i)*(ort-gort)*t(ort-gort).
compute w=w+(n(i)-1)*s.
end loop.
*print b.
*print w.
compute lamda=det(w)/det(w+b).
*print lamda /format=f8.3.
compute sderpar=nrow(x)*(ncol(n)-1).
compute lpar=-(msum(n)-1-(nrow(x)+ncol(n))/2)*ln(lamda).
compute pdegpar=1-chicdf(lpar,sderpar).
compute kararpar=0.
do if (pdegpar<0.05).
compute kararpar=1.
end if.
*print kararpar.

compute x=t(y).
*print x.
compute ggsay=n.
compute gsay=ncol(ggsay).
compute d=nrow(x).
compute n=ncol(x).
compute sat=nrow(x).
compute sut=ncol(x).
compute mat1=make(sut,3,0).
compute r=make(sat,sut,0).
loop i=1 to sat.
  loop k=1 to sut.
  compute mat1(k,3)=x(i,k).
  compute mat1(k,2)=k.
  compute mat1(k,1)=k.
  end loop.
  compute mat2=mat1.
  loop k1=1 to sut-1.
  loop k2=k1+1 to sut.
  do if mat2(k1,3)>mat2(k2,3).
  compute a= mat2(k1,3).
  compute mat2(k1,3)=mat2(k2,3).
  compute mat2(k2,3)=a.
  compute a=mat2(k1,2).
  compute mat2(k1,2)=mat2(k2,2).
  compute mat2(k2,2)=a.
  end if.
  end loop.
  end loop.
  compute mat3=mat2.
  loop k1=1 to sut-1.
  loop k2=k1+1 to sut.
  do if mat3(k1,2)>mat3(k2,2).
  compute a=mat3(k1,2).
  compute mat3(k1,2)=mat3(k2,2).
  compute mat3(k2,2)=a.
  compute a=mat3(k1,1).
  compute mat3(k1,1)=mat3(k2,1).
  compute mat3(k2,1)=a.
  compute a=mat3(k1,3).

```

```

        compute mat3(k1,3)=mat3(k2,3).
        compute mat3(k2,3)=a.
        end if.
    end loop.
end loop.
loop j=1 to sut.
compute r(i,j)=mat3(j,1).
end loop.
end loop.
*print r.
compute e=(1/(n+1))*(r).
*print e /format=f8.3.
compute tm=make(d,gsay,0).
compute k=1.
compute s=0.
loop i=1 to d.
    loop j=1 to n.
        compute tm(i,k)=tm(i,k)+e(i,j).
        compute s=s+1.
        do if (s=gsay(k)).
            compute tm(i,k)=tm(i,k)/gsay(k).
            compute k=k+1.
            compute s=0.
            end if.
        end loop.
        compute k=1.
    end loop.
    *print tm/format=f8.3.
    compute ec=rsum(e)/n.
    *print ec.
    compute s=(1/n)*e*t(e)-ec*t(ec).
    compute dets=det(s).
    do if (dets<0.01).
        compute sters=ginv(s).
        else.
        compute sters=inv(s).
        end if.
    compute sderpolm=d*(gsay-1).
    compute lparolm=0.
    loop k=1 to gsay.
        compute lparolm=lparolm+gsay(k)*(t(tm(:,k)-ec)*sters*(tm(:,k)-ec)).
    end loop.
    compute pdparolm=1-chicdf(lparolm,sderpolm).
    compute kparolm=0.
    do if (pdparolm<0.05).
        compute kparolm=1.
    end if.
    compute sonmat={lpar,sderpar,pdegpar,kararpar;lparolm,sderpolm,pdparolm,kparolm}.
    *print sonmat /title= "Sonuçlar" /format=f8.3 /rlabels=par,npar /clabels=list,sder,pdeg,karar .
    compute tp=tp+kararpar.
    compute tnp=tnp+kparolm.
end loop.
compute deneme=deneme-1.
compute tp=tp/deneme.
compute tnp=tnp/deneme.
print tp /format=f8.4.
print tnp /format=f8.4.
end matrix.

```

EK 7.11. Normal ve Ki-kare Dağılımlarına Sahip İki Grubun Konum Testi için Parametrik ve Parametrik Olmayan Simülasyon (Serbestlik Derecesi=3)

```

set mxloop=100000.
matrix.
compute tp=0. /* Parametrik test denemeleri sonucu ret oranı */
compute tnp=0. /* Parametrik olmayan test denemeleri sonucu ret oranı */
loop deneme=1 to 1000. /* Deneme sayısı */
compute n1=5. /* *** Her bir gruptaki gözlem sayısı girilmelidir *** */
compute ort={3;3}. /* 1. Grup- 2 değişkenli normal dağılım için */
compute km={1,0;0,1}.
compute x11=uniform(n1,1).
compute x12=uniform(n1,1).
compute z1=((-2*ln(x11))&*&0.5)&*sin(6.28*x12)).
compute x21=uniform(n1,1).
compute x22=uniform(n1,1).
compute z2=((-2*ln(x21))&*&0.5)&*sin(6.28*x22)).
compute z={z1,z2}.
compute ckm=chol(km).
compute g1=t((t(ckm)*t(z))+kroneker(make(1,n1,1),ort))). /* Grup 1 */

compute x11=uniform(n1,1). /* 2. Grup- 1. değişken için; Ki-kare (sder=3) */
compute x12=uniform(n1,1).
compute z1=((-2*ln(x11))&*&0.5)&*sin(6.28*x12)).
compute x21=uniform(n1,1).
compute x22=uniform(n1,1).
compute z2=((-2*ln(x21))&*&0.5)&*sin(6.28*x22)).
compute x31=uniform(n1,1).
compute x32=uniform(n1,1).
compute z3=((-2*ln(x31))&*&0.5)&*sin(6.28*x32)).
compute kk1=z1&*&2+z2&*&2+z3&*&2.

compute x11=uniform(n1,1). /* 2. Grup- 2. değişken için; Ki-kare (sder=3) */
compute x12=uniform(n1,1).
compute z1=((-2*ln(x11))&*&0.5)&*sin(6.28*x12)).
compute x21=uniform(n1,1).
compute x22=uniform(n1,1).
compute z2=((-2*ln(x21))&*&0.5)&*sin(6.28*x22)).
compute x31=uniform(n1,1).
compute x32=uniform(n1,1).
compute z3=((-2*ln(x31))&*&0.5)&*sin(6.28*x32)).
compute kk2=z1&*&2+z2&*&2+z3&*&2.
compute g2={kk1,kk2}. /* Grup 2 */

compute y={g1,g2}.
*print y.
compute x=t(y).
compute n={n1,n1}.
compute b=0.
compute w=0.
compute j=1.
compute jj=0.
compute gort=rsum(x)/msum(n).
*print gort.
loop i=1 to ncol(n).
compute jj=jj+n(i).

```

```

compute a=x(:,j:jj).
compute j=j+n(i).
compute ort=rsum(a)/n(i).
compute s=(a*t(a)/n(i)-ort*t(ort))*n(i)/(n(i)-1).
*print ort.
*print s.
compute b=b+n(i)*(ort-gort)*t(ort-gort).
compute w=w+(n(i)-1)*s.
end loop.
compute lamda=det(w)/det(w+b).
compute sderpar=nrow(x)*(ncol(n)-1).
compute lpar=-(msum(n)-1-(nrow(x)+ncol(n))/2)*ln(lamda).
compute pdegpar=1-chicdf(lpar,sderpar).
compute kararpar=0.
do if (pdegpar<0.05).
compute kararpar=1.
end if.

compute x=t(y).
*print x.
compute ggsay=n.
compute gsay=ncol(ggsay).
compute d=nrow(x).
compute n=ncol(x).
compute sat=nrow(x).
compute sut=ncol(x).
compute mat1=make(sut,3,0).
compute r=make(sat,sut,0).
loop i=1 to sat.
  loop k=1 to sut.
  compute mat1(k,3)=x(i,k).
  compute mat1(k,2)=k.
  compute mat1(k,1)=k.
  end loop.
  compute mat2=mat1.
  loop k1=1 to sut-1.
    loop k2=k1+1 to sut.
    do if mat2(k1,3)>mat2(k2,3).
    compute a= mat2(k1,3).
    compute mat2(k1,3)=mat2(k2,3).
    compute mat2(k2,3)=a.
    compute a=mat2(k1,2).
    compute mat2(k1,2)=mat2(k2,2).
    compute mat2(k2,2)=a.
    end if.
  end loop.
end loop.
compute mat3=mat2.
loop k1=1 to sut-1.
  loop k2=k1+1 to sut.
  do if mat3(k1,2)>mat3(k2,2).
  compute a=mat3(k1,2).
  compute mat3(k1,2)=mat3(k2,2).
  compute mat3(k2,2)=a.
  compute a=mat3(k1,1).
  compute mat3(k1,1)=mat3(k2,1).
  compute mat3(k2,1)=a.
  compute a=mat3(k1,3).

```

```

        compute mat3(k1,3)=mat3(k2,3).
        compute mat3(k2,3)=a.
        end if.
    end loop.
end loop.
loop j=1 to sut.
compute r(i,j)=mat3(j,1).
end loop.
end loop.
*print r.
compute e=(1/(n+1))*(r).
*print e /format=f8.3.
compute tm=make(d,gsay,0).
compute k=1.
compute s=0.
loop i=1 to d.
    loop j=1 to n.
        compute tm(i,k)=tm(i,k)+e(i,j).
        compute s=s+1.
        do if (s=gsay(k)).
            compute tm(i,k)=tm(i,k)/gsay(k).
            compute k=k+1.
            compute s=0.
        end if.
    end loop.
    compute k=1.
end loop.
*print tm/format=f8.3.
compute ec=rsum(e)/n.
*print ec.
compute s=(1/n)*e*t(e)-ec*t(ec).
compute dets=det(s).
do if (dets<0.01).
compute sters=ginv(s).
else.
compute sters=inv(s).
end if.
compute sderpolm=d*(gsay-1).
compute lparolm=0.
loop k=1 to gsay.
compute lparolm=lparolm+gsay(k)*(t(tm(:,k)-ec)*sters*(tm(:,k)-ec)).
end loop.
compute pdparolm=1-chicdf(lparolm,sderpolm).
compute kparolm=0.
do if (pdparolm<0.05).
compute kparolm=1.
end if.
compute sonmat={lpar,sderpar,pdegpar,kararpar;lparolm,sderpolm,pdparolm,kparolm}.
*print sonmat /title= "Sonuçlar" /format=f8.3 /rlabels=par,npar /clabels=list,sder,pdeg,karar .
compute tp=tp+kararpar.
compute tnp=tnp+kparolm.
end loop.
compute deneme=deneme-1.
compute tp=tp/deneme.
compute tnp=tnp/deneme.
print tp /format=f8.4.
print tnp /format=f8.4.
end matrix.

```

EK 7.12. Normal ve Üstel Dağılımlarına Sahip İki Grubun Konum Testi için Parametrik ve Parametrik Olmayan Simülasyon (Teta=3)

```

set mxloop=100000.
matrix.
compute tp=0. /* Parametrik test denemeleri sonucu ret oranı */
compute tnp=0. /* Parametrik olmayan test denemeleri sonucu ret oranı */
loop deneme=1 to 1000. /* Deneme sayısı */
compute n1=5. /* *** Her bir gruptaki gözlem sayısı girilmelidir *** */
compute ort={3;3}. /* 1. Grup- 2 değişkenli normal dağılım için */
compute km={1,0;0,1}.
compute x11=uniform(n1,1).
compute x12=uniform(n1,1).
compute z1=(-2*ln(x11))&*&0.5)&*sin(6.28*x12)).
*print z1.
compute x21=uniform(n1,1).
compute x22=uniform(n1,1).
compute z2=(-2*ln(x21))&*&0.5)&*sin(6.28*x22)).
*print z2.
compute z={z1,z2}.
*print z.
compute ckm=chol(km).
compute g1=t((t(ckm)*t(z))+(kroneker(make(1,n1,1),ort))). /* Grup 1 */
*print g1.

compute teta=3. /* 2. Grup- 1. değişken için; Üstel dağılım (teta=3) */
compute d=make(n1,1,0).
compute üstel1=make(n1,1,0).
compute d=uniform(n1,1).
loop i=1 to n1.
    compute üstel1(i)=-teta*ln(d(i)).
end loop.
compute d=make(n1,1,0). /* 2. Grup- 2. değişken için; Üstel dağılım (teta=3) */
compute üstel2=make(n1,1,0).
compute d=uniform(n1,1).
loop i=1 to n1.
    compute üstel2(i)=-teta*ln(d(i)).
end loop.

compute g2={üstel1,üstel2}. /* Grup 2 */
*save g2 /outfile=*.

compute y={g1;g2}.
*print y.
compute x=t(y).
*print x.
compute n={n1,n1}.
compute b=0.
compute w=0.
compute j=1.
compute jj=0.
compute gort=rsum(x)/msum(n).
*print gort.
loop i=1 to ncol(n).
compute jj=jj+n(i).
compute a=x(:,j:jj)).

```

```

compute j=j+n(i).
*print a.
compute ort=rsum(a)/n(i).
compute s=(a*t(a)/n(i)-ort*t(ort))*n(i)/(n(i)-1).
*print ort.
*print s.
compute b=b+n(i)*(ort-gort)*t(ort-gort).
compute w=w+(n(i)-1)*s.
end loop.
compute lamda=det(w)/det(w+b).
*print lamda /format=f8.3.
compute sderpar=nrow(x)*(ncol(n)-1).
compute lpar=- (msum(n)-1-(nrow(x)+ncol(n))/2)*ln(lamda).
compute pdegpar=1-chicdf(lpar,sderpar).
compute kararpar=0.
do if (pdegpar<0.05).
compute kararpar=1.
end if.
compute x=t(y).
*print x.
compute ggsay=n.
compute gsay=ncol(ggsay).
compute d=nrow(x).
compute n=ncol(x).
compute sat=nrow(x).
compute sut=ncol(x).
compute mat1=make(sut,3,0).
compute r=make(sat,sut,0).
loop i=1 to sat.
  loop k=1 to sut.
  compute mat1(k,3)=x(i,k).
  compute mat1(k,2)=k.
  compute mat1(k,1)=k.
  end loop.
  compute mat2=mat1.
  loop k1=1 to sut-1.
  loop k2=k1+1 to sut.
  do if mat2(k1,3)>mat2(k2,3).
  compute a= mat2(k1,3).
  compute mat2(k1,3)=mat2(k2,3).
  compute mat2(k2,3)=a.
  compute a=mat2(k1,2).
  compute mat2(k1,2)=mat2(k2,2).
  compute mat2(k2,2)=a.
  end if.
  end loop.
  end loop.
  compute mat3=mat2.
  loop k1=1 to sut-1.
  loop k2=k1+1 to sut.
  do if mat3(k1,2)>mat3(k2,2).
  compute a=mat3(k1,2).
  compute mat3(k1,2)=mat3(k2,2).
  compute mat3(k2,2)=a.
  compute a=mat3(k1,1).
  compute mat3(k1,1)=mat3(k2,1).
  compute mat3(k2,1)=a.
  compute a=mat3(k1,3).

```

```

        compute mat3(k1,3)=mat3(k2,3).
        compute mat3(k2,3)=a.
        end if.
    end loop.
end loop.
loop j=1 to sut.
compute r(i,j)=mat3(j,1).
end loop.
end loop.
*print r.
compute e=(1/(n+1))*(r).
*print e /format=f8.3.
compute tm=make(d,gsay,0).
compute k=1.
compute s=0.
loop i=1 to d.
    loop j=1 to n.
        compute tm(i,k)=tm(i,k)+e(i,j).
        compute s=s+1.
        do if (s=gsay(k)).
            compute tm(i,k)=tm(i,k)/gsay(k).
            compute k=k+1.
            compute s=0.
        end if.
    end loop.
    compute k=1.
end loop.
*print tm/format=f8.3.
compute ec=rsum(e)/n.
*print ec.
compute s=(1/n)*e*t(e)-ec*t(ec).
compute dets=det(s).
do if (dets<0.01).
compute sters=ginv(s).
else.
compute sters=inv(s).
end if.
compute sderpolm=d*(gsay-1).
compute lparolm=0.
loop k=1 to gsay.
compute lparolm=lparolm+gsay(k)*(t(tm(:,k)-ec)*sters*(tm(:,k)-ec)).
end loop.
compute pdparolm=1-chicdf(lparolm,sderpolm).
compute kparolm=0.
do if (pdparolm<0.05).
compute kparolm=1.
end if.
compute sonmat={lpar,sderpar,pdegpar,kararpar;lparolm,sderpolm,pdparolm,kparolm}.
*print sonmat /title= "Sonuçlar" /format=f8.3 /rlabels=par,npar /clabels=list,sder,pdeg,karar .
compute tp=tp+kararpar.
compute tnp=tnp+kparolm.
end loop.
compute deneme=deneme-1.
compute tp=tp/deneme.
compute tnp=tnp/deneme.
print tp /format=f8.4.
print tnp /format=f8.4.
end matrix.

```

EK 7.13. Ki-kare Dağılımına Sahip İki Grubun Konum Testi için Parametrik ve Parametrik Olmayan Simülasyon (Serbestlik Dereceleri=3)

```

set mxloop=100000.
matrix.
compute tp=0. /* Parametrik test denemeleri sonucu ret oranı */
compute tnp=0. /* Parametrik olmayan test denemeleri sonucu ret oranı */
loop deneme=1 to 1000. /* Deneme sayısı */
compute n1=5. /* *** Her bir gruptaki gözlem sayısı girilmelidir *** */
compute x11=uniform(n1,1). /* 1. Grup- 1. değişken için; Ki-kare (sder=3) */
compute x12=uniform(n1,1).
compute z1=(-2*ln(x11))&*&0.5)&*sin(6.28*x12)).
compute x21=uniform(n1,1).
compute x22=uniform(n1,1).
compute z2=(-2*ln(x21))&*&0.5)&*sin(6.28*x22)).
compute x31=uniform(n1,1).
compute x32=uniform(n1,1).
compute z3=(-2*ln(x31))&*&0.5)&*sin(6.28*x32)).
compute kk1=z1&*&2+z2&*&2+z3&*&2.

compute x11=uniform(n1,1). /* 1. Grup- 2. değişken için; Ki-kare (sder=3) */
compute x12=uniform(n1,1).
compute z1=(-2*ln(x11))&*&0.5)&*sin(6.28*x12)).
compute x21=uniform(n1,1).
compute x22=uniform(n1,1).
compute z2=(-2*ln(x21))&*&0.5)&*sin(6.28*x22)).
compute x31=uniform(n1,1).
compute x32=uniform(n1,1).
compute z3=(-2*ln(x31))&*&0.5)&*sin(6.28*x32)).
compute kk2=z1&*&2+z2&*&2+z3&*&2.
compute g1={kk1,kk2}. /* Grup 1 */

compute x11=uniform(n1,1). /* 2. Grup- 1. değişken için; Ki-kare (sder=3) */
compute x12=uniform(n1,1).
compute z1=(-2*ln(x11))&*&0.5)&*sin(6.28*x12)).
compute x21=uniform(n1,1).
compute x22=uniform(n1,1).
compute z2=(-2*ln(x21))&*&0.5)&*sin(6.28*x22)).
compute x31=uniform(n1,1).
compute x32=uniform(n1,1).
compute z3=(-2*ln(x31))&*&0.5)&*sin(6.28*x32)).
compute kk1=z1&*&2+z2&*&2+z3&*&2.

compute x11=uniform(n1,1). /*2. Grup-2. değişken için; Ki-kare (sder=3) */
compute x12=uniform(n1,1).
compute z1=(-2*ln(x11))&*&0.5)&*sin(6.28*x12)).
compute x21=uniform(n1,1).
compute x22=uniform(n1,1).
compute z2=(-2*ln(x21))&*&0.5)&*sin(6.28*x22)).
compute x31=uniform(n1,1).
compute x32=uniform(n1,1).
compute z3=(-2*ln(x31))&*&0.5)&*sin(6.28*x32)).
compute kk2=z1&*&2+z2&*&2+z3&*&2.
compute g2={kk1,kk2}. /* Grup 2 */
compute y={g1,g2}.
compute x=t(y).

```

```

compute n={n1,n1}.
compute b=0.
compute w=0.
compute j=1.
compute jj=0.
compute gort=rsum(x)/msum(n).
loop i=1 to ncol(n).
compute jj=jj+n(i).
compute a=x(:,j:jj).
compute j=j+n(i).
compute ort=rsum(a)/n(i).
compute s=(a*t(a)/n(i)-ort*t(ort))*n(i)/(n(i)-1).
compute b=b+n(i)*(ort-gort)*t(ort-gort).
compute w=w+(n(i)-1)*s.
end loop.
compute lamda=det(w)/det(w+b).
compute sderpar=nrow(x)*(ncol(n)-1).
compute lpar=-(msum(n)-1-(nrow(x)+ncol(n))/2)*ln(lamda).
compute pdegpar=1-chicdf(lpar,sderpar).
compute kararpar=0.
do if (pdegpar<0.05).
compute kararpar=1.
end if.
compute x=t(y).
compute ggsay=n.
compute gsay=ncol(ggsay).
compute d=nrow(x).
compute n=ncol(x).
compute sat=nrow(x).
compute sut=ncol(x).
compute mat1=make(sut,3,0).
compute r=make(sat,sut,0).
loop i=1 to sat.
loop k=1 to sut.
compute mat1(k,3)=x(i,k).
compute mat1(k,2)=k.
compute mat1(k,1)=k.
end loop.
compute mat2=mat1.
loop k1=1 to sut-1.
loop k2=k1+1 to sut.
do if mat2(k1,3)>mat2(k2,3).
compute a= mat2(k1,3).
compute mat2(k1,3)=mat2(k2,3).
compute mat2(k2,3)=a.
compute a=mat2(k1,2).
compute mat2(k1,2)=mat2(k2,2).
compute mat2(k2,2)=a.
end if.
end loop.
end loop.
compute mat3=mat2.
loop k1=1 to sut-1.
loop k2=k1+1 to sut.
do if mat3(k1,2)>mat3(k2,2).
compute a=mat3(k1,2).
compute mat3(k1,2)=mat3(k2,2).
compute mat3(k2,2)=a.

```

```

        compute a=mat3(k1,1).
        compute mat3(k1,1)=mat3(k2,1).
        compute mat3(k2,1)=a.
        compute a=mat3(k1,3).
        compute mat3(k1,3)=mat3(k2,3).
        compute mat3(k2,3)=a.
        end if.
    end loop.
end loop.
loop j=1 to sut.
compute r(i,j)=mat3(j,1).
end loop.
end loop.
compute e=(1/(n+1))*(r).
compute tm=make(d,gsay,0).
compute k=1.
compute s=0.
loop i=1 to d.
    loop j=1 to n.
        compute tm(i,k)=tm(i,k)+e(i,j).
        compute s=s+1.
        do if (s=ggsay(k)).
            compute tm(i,k)=tm(i,k)/ggsay(k).
            compute k=k+1.
            compute s=0.
        end if.
    end loop.
    compute k=1.
end loop.
compute ec=rsum(e)/n.
compute s=(1/n)*e*t(e)-ec*t(ec).
compute dets=det(s).
do if (dets<0.01).
compute sters=ginv(s).
else.
compute sters=inv(s).
end if.
compute sderpolm=d*(gsay-1).
compute lparolm=0.
loop k=1 to gsay.
compute lparolm=lparolm+ggsay(k)*(t(tm(:,k)-ec)*sters*(tm(:,k)-ec)).
end loop.
compute pdparolm=1-chicdf(lparolm,sderpolm).
compute kparolm=0.
do if (pdparolm<0.05).
compute kparolm=1.
end if.
compute sonmat={lpar,sderpar,pdegpar,kararpar;lparolm,sderpolm,pdparolm,kparolm}.
*print sonmat /title= "Sonuçlar" /format=f8.3 /rlabels=par,npar /clabels=list,sder,pdeg,karar .
compute tp=tp+kararpar.
compute tnp=tnp+kparolm.
end loop.
compute deneme=deneme-1.
compute tp=tp/deneme.
compute tnp=tnp/deneme.
print tp /format=f8.4.
print tnp /format=f8.4.
end matrix.

```

EK 7.14. Ki-kare ve Üstel Dağılımlarına Sahip İki Grubun Konum Testi için Parametrik ve Parametrik Olmayan Simülasyon (Serbestlik Derecesi=3, Teta=3)

```

set mxloop=100000.
matrix.
compute tp=0. /* Parametrik test denemeleri sonucu ret oranı */
compute tnp=0. /* Parametrik olmayan test denemeleri sonucu ret oranı */
loop deneme=1 to 1000. /* Deneme sayısı */
compute n1=5. /* *** Her bir gruptaki gözlem sayısı girilmelidir *** */
compute x11=uniform(n1,1). /* 1. Grup- 1. değişken için; Ki-kare (sder=3) */
compute x12=uniform(n1,1).
compute z1=(-2*ln(x11))&*&0.5)&*sin(6.28*x12)).
*print z1.
compute x21=uniform(n1,1).
compute x22=uniform(n1,1).
compute z2=(-2*ln(x21))&*&0.5)&*sin(6.28*x22)).
*print z2.
compute x31=uniform(n1,1).
compute x32=uniform(n1,1).
compute z3=(-2*ln(x31))&*&0.5)&*sin(6.28*x32)).
compute kk1=z1&*&2+z2&*&2+z3&*&2.

compute x11=uniform(n1,1). /* 1. Grup- 2. değişken için; Ki-kare (sder=3) */
compute x12=uniform(n1,1).
compute z1=(-2*ln(x11))&*&0.5)&*sin(6.28*x12)).
compute x21=uniform(n1,1).
compute x22=uniform(n1,1).
compute z2=(-2*ln(x21))&*&0.5)&*sin(6.28*x22)).
compute x31=uniform(n1,1).
compute x32=uniform(n1,1).
compute z3=(-2*ln(x31))&*&0.5)&*sin(6.28*x32)).
compute kk2=z1&*&2+z2&*&2+z3&*&2.
compute g1={kk1,kk2}. /* Grup 1 */
*save g1 /outfile=*.

compute teta=3. /* 2. Grup- 1. değişken için; Üstel dağılım (teta=3) */
compute d=make(n1,1,0).
compute üstel1=make(n1,1,0).
compute d=uniform(n1,1).
loop i=1 to n1.
compute üstel1(i)=-teta*ln(d(i)).
end loop.
compute d=make(n1,1,0). /* 2. Grup- 2. değişken için; Üstel dağılım (teta=3) */
compute üstel2=make(n1,1,0).
compute d=uniform(n1,1).
loop i=1 to n1.
compute üstel2(i)=-teta*ln(d(i)).
end loop.
compute g2={üstel1,üstel2}. /* Grup 2 */

compute y={g1,g2}.
compute x=t(y).
compute n={n1,n1}.
compute b=0.

```

```

compute w=0.
compute j=1.
compute jj=0.
compute gort=rsum(x)/msum(n).
loop i=1 to ncol(n).
compute jj=jj+n(i).
compute a=x(:,j:jj).
compute j=j+n(i).
compute ort=rsum(a)/n(i).
compute s=(a*t(a)/n(i)-ort*t(ort))*n(i)/(n(i)-1).
compute b=b+n(i)*(ort-gort)*t(ort-gort).
compute w=w+(n(i)-1)*s.
end loop.
compute lamda=det(w)/det(w+b).
compute sderpar=nrow(x)*(ncol(n)-1).
compute lpar=-(msum(n)-1-(nrow(x)+ncol(n))/2)*ln(lamda).
compute pdegpar=1-chicdf(lpar,sderpar).
compute kararpar=0.
do if (pdegpar<0.05).
compute kararpar=1.
end if.

```

```

compute x=t(y).
compute ggsay=n.
compute gsay=ncol(ggsay).
compute d=nrow(x).
compute n=ncol(x).
compute sat=nrow(x).
compute sut=ncol(x).
compute mat1=make(sut,3,0).
compute r=make(sat,sut,0).
loop i=1 to sat.
loop k=1 to sut.
compute mat1(k,3)=x(i,k).
compute mat1(k,2)=k.
compute mat1(k,1)=k.
end loop.
compute mat2=mat1.
loop k1=1 to sut-1.
loop k2=k1+1 to sut.
do if mat2(k1,3)>mat2(k2,3).
compute a= mat2(k1,3).
compute mat2(k1,3)=mat2(k2,3).
compute mat2(k2,3)=a.
compute a=mat2(k1,2).
compute mat2(k1,2)=mat2(k2,2).
compute mat2(k2,2)=a.
end if.
end loop.
end loop.
compute mat3=mat2.
loop k1=1 to sut-1.
loop k2=k1+1 to sut.
do if mat3(k1,2)>mat3(k2,2).
compute a=mat3(k1,2).
compute mat3(k1,2)=mat3(k2,2).
compute mat3(k2,2)=a.
compute a=mat3(k1,1).

```

```

        compute mat3(k1,1)=mat3(k2,1).
        compute mat3(k2,1)=a.
        compute a=mat3(k1,3).
        compute mat3(k1,3)=mat3(k2,3).
        compute mat3(k2,3)=a.
        end if.
    end loop.
end loop.
loop j=1 to sut.
compute r(i,j)=mat3(j,1).
end loop.
end loop.
compute e=(1/(n+1))*(r).
compute tm=make(d,gsay,0).
compute k=1.
compute s=0.
loop i=1 to d.
    loop j=1 to n.
        compute tm(i,k)=tm(i,k)+e(i,j).
        compute s=s+1.
        do if (s=gsay(k)).
            compute tm(i,k)=tm(i,k)/gsay(k).
            compute k=k+1.
            compute s=0.
        end if.
    end loop.
    compute k=1.
end loop.
compute ec=rsum(e)/n.
compute s=(1/n)*e*t(e)-ec*t(ec).
compute dets=det(s).
do if (dets<0.01).
compute sters=ginv(s).
else.
compute sters=inv(s).
end if.
compute sderpolm=d*(gsay-1).
compute lparolm=0.
loop k=1 to gsay.
compute lparolm=lparolm+gsay(k)*(t(tm(:,k)-ec)*sters*(tm(:,k)-ec)).
end loop.
compute pdparolm=1-chicdf(lparolm,sderpolm).
compute kparolm=0.
do if (pdparolm<0.05).
compute kparolm=1.
end if.
compute sonmat={lpar,sderpar,pdegpar,kararpar;lparolm,sderpolm,pdparolm,kparolm}.
*print sonmat /title= "Sonuçlar" /format=f8.3 /rlabels=par,npar /clabels=list,sder,pdeg,karar .
compute tp=tp+kararpar.
compute tnp=tnp+kparolm.
end loop.
compute deneme=deneme-1.
compute tp=tp/deneme.
compute tnp=tnp/deneme.
print tp /format=f8.4.
print tnp /format=f8.4.
end matrix.

```

EK 7.15. Üstel Dağılıma Sahip İki Grubun Konum Testi için Parametrik ve Parametrik Olmayan Simülasyon (Teta=3)

```

set mxloop=100000.
matrix.
compute tp=0. /* Parametrik test denemeleri sonucu ret oranı */
compute tnp=0. /* Parametrik olmayan test denemeleri sonucu ret oranı */
loop deneme=1 to 1000. /* Deneme sayısı */
compute n1=5. /* *** Her bir gruptaki gözlem sayısı girilmelidir *** */
compute teta=3. /* 1. Grup- 1. değişken için; Üstel dağılım (teta=3) */
/* Değiştirilecek*/

compute d=make(n1,1,0).
compute üstel1=make(n1,1,0).
compute d=uniform(n1,1).
loop i=1 to n1.
    compute üstel1(i)=-teta*ln(d(i)).
end loop.
compute d=make(n1,1,0). /* 1. Grup- 2. değişken için; Üstel dağılım (teta=3) */
compute üstel2=make(n1,1,0).
compute d=uniform(n1,1).
loop i=1 to n1.
    compute üstel2(i)=-teta*ln(d(i)).
end loop.

compute g1={üstel1,üstel2}. /* Grup 1 */
*save g1 /outfile=*.

compute teta=3. /* 2. Grup- 1. değişken için; Üstel dağılım (teta=3) */
compute d=make(n1,1,0).
compute üstel1=make(n1,1,0).
compute d=uniform(n1,1).
loop i=1 to n1.
    compute üstel1(i)=-teta*ln(d(i)).
end loop.
compute d=make(n1,1,0). /* 2. Grup- 2. değişken için; Üstel dağılım (teta=3) */
compute üstel2=make(n1,1,0).
compute d=uniform(n1,1).
loop i=1 to n1.
    compute üstel2(i)=-teta*ln(d(i)).
end loop.

compute g2={üstel1,üstel2}. /* Grup 2 */
*save g2 /outfile=*.

compute y={g1,g2}.
*print y.
compute x=t(y).
*print x.
compute n={n1,n1}.
compute b=0.
compute w=0.
compute j=1.
compute jj=0.
compute gort=rsum(x)/msum(n).
*print gort.
loop i=1 to ncol(n).

```

```

compute jj=jj+n(i).
compute a=x(:,j:jj).
compute j=j+n(i).
*print a.
compute ort=rsum(a)/n(i).
compute s=(a*t(a)/n(i)-ort*t(ort))*n(i)/(n(i)-1).
compute b=b+n(i)*(ort-gort)*t(ort-gort).
compute w=w+(n(i)-1)*s.
end loop.
compute lamda=det(w)/det(w+b).
compute sderpar=nrow(x)*(ncol(n)-1).
compute lpar=-((msum(n)-1-(nrow(x)+ncol(n))/2)*ln(lamda)).
compute pdegpar=1-chicdf(lpar,sderpar).
compute kararpar=0.
do if (pdegpar<0.05).
compute kararpar=1.
end if.
*print kararpar.

compute x=t(y).
*print x.
compute ggsay=n.
compute gsay=ncol(ggsay).
compute d=nrow(x).
compute n=ncol(x).
compute sat=nrow(x).
compute sut=ncol(x).
compute mat1=make(sut,3,0).
compute r=make(sat,sut,0).
loop i=1 to sat.
  loop k=1 to sut.
  compute mat1(k,3)=x(i,k).
  compute mat1(k,2)=k.
  compute mat1(k,1)=k.
  end loop.
  compute mat2=mat1.
  loop k1=1 to sut-1.
  loop k2=k1+1 to sut.
  do if mat2(k1,3)>mat2(k2,3).
  compute a= mat2(k1,3).
  compute mat2(k1,3)=mat2(k2,3).
  compute mat2(k2,3)=a.
  compute a=mat2(k1,2).
  compute mat2(k1,2)=mat2(k2,2).
  compute mat2(k2,2)=a.
  end if.
  end loop.
  end loop.
  compute mat3=mat2.
  loop k1=1 to sut-1.
  loop k2=k1+1 to sut.
  do if mat3(k1,2)>mat3(k2,2).
  compute a=mat3(k1,2).
  compute mat3(k1,2)=mat3(k2,2).
  compute mat3(k2,2)=a.
  compute a=mat3(k1,1).
  compute mat3(k1,1)=mat3(k2,1).
  compute mat3(k2,1)=a.

```

```

        compute a=mat3(k1,3).
        compute mat3(k1,3)=mat3(k2,3).
        compute mat3(k2,3)=a.
        end if.
    end loop.
end loop.
loop j=1 to sut.
compute r(i,j)=mat3(j,1).
end loop.
end loop.
*print r.
compute e=(1/(n+1))*(r).
*print e /format=f8.3.
compute tm=make(d,gsay,0).
compute k=1.
compute s=0.
loop i=1 to d.
    loop j=1 to n.
        compute tm(i,k)=tm(i,k)+e(i,j).
        compute s=s+1.
        do if (s=gsay(k)).
            compute tm(i,k)=tm(i,k)/gsay(k).
            compute k=k+1.
            compute s=0.
        end if.
    end loop.
    compute k=1.
end loop.
compute ec=rsum(e)/n.
compute s=(1/n)*e*t(e)-ec*t(ec).
compute dets=det(s).
do if (dets<0.01).
compute sters=ginv(s).
else.
compute sters=inv(s).
end if.
compute sderpolm=d*(gsay-1).
compute lparolm=0.
loop k=1 to gsay.
compute lparolm=lparolm+gsay(k)*(t(tm(:,k)-ec)*sters*(tm(:,k)-ec)).
end loop.
compute pdparolm=1-chicdf(lparolm,sderpolm).
compute kparolm=0.
do if (pdparolm<0.05).
compute kparolm=1.
end if.

compute sonmat={lpar,sderpar,pdegpar,kararpar;lparolm,sderpolm,pdparolm,kparolm}.
*print sonmat /title= "Sonuçlar" /format=f8.3 /rlabels=par,npar /clabels=list,sder,pdeg,karar .
compute tp=tp+kararpar.
compute tnp=tnp+kparolm.
end loop.
compute deneme=deneme-1.
compute tp=tp/deneme.
compute tnp=tnp/deneme.
print tp /format=f8.4.
print tnp /format=f8.4.
end matrix.

```

KAYNAKLAR

1. Alpar, R. 2003. Uygulamalı Çok Değişkenli İstatistiksel Yöntemlere Giriş 1, Nobel Yayın Dağıtım, Ankara.
2. Brodeen, A. E. M., Taylor, M. S. 1994. A Multivariate Multisample Rank Test for Stochastic Simulation Validation, National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.
3. Gamgam, H. 1998. Parametrik Olmayan İstatistiksel Teknikler, Türk Hava Kurumu Basımevi İşletmeciliği, Ankara.
4. Johnson, R.A. ve Wichern, D.W. 2002. Applied Multivariate Statistical Analysis, New Jersey, Prentice-Hall Inc.
5. Puri, M.L., Sen, P.K. 1971. Nonparametric Methods in Multivariate Analysis, John Wiley&Sons, Inc., New York.
6. Semiz, M. 2007. Deney Tasarımı&Varyans Analizi, Dizgi Ofset, Konya.
7. Semiz, M. 2007. İstatistik Laboratuvarı & SPSS-MP, Dizgi Ofset, Konya.
8. Semiz, M. 2006. Uygulamalı İstatistik, Dizgi Ofset, Konya.
9. Semiz, M. 2002. Parametre dışı yaygınlık testi, Doktora Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
10. Tatlıdil, H. 2002. Uygulamalı Çok Değişkenli İstatistiksel Analiz, Ziraat Matbaacılık, Ankara.
11. Tuncer, Y. 2002. Çok Değişkenli İstatistik Analizine Giriş: Normal Teori, Bıçaklar Kitabevi, Ankara.
12. Um, Y., Randles, R.H. 1998. Nonparametric Tests for The Multivariate Multi-Sample Location Problem, Statistica Sinica 8, 801-812.
13. Ünver, Ö., Gamgam H. 1996. Uygulamalı İstatistik Yöntemler, Siyasal Kitabevi, Ankara.