



T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

FAKTÖR ANALİZİNDE TAHMİN EDİCİLERİN KARŞILAŞTIRILMASI

Osman USTA

YÜKSEK LİSANS TEZİ

İstatistik Anabilim Dalını

Eylül-2019
KONYA
Her Hakkı Saklıdır

ÖZET

YÜKSEK LİSANS TEZİ

FAKTÖR ANALİZİNDE TAHMİN EDİCİLERİN KARŞILAŞTIRILMASI

Osman Usta

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
İstatistik Anabilim Dalı

Danışman: Prof.Dr. Coşkun KUŞ

2019, 52 Sayfa

Jüri
Prof.Dr. Coşkun KUŞ
Dr. Öğr. Üyesi Ahmet PEKGÖR
Dr. Öğr. Üyesi Yunus AKDOĞAN

Faktör Analizi (FA) veri biliminin hemen hemen tüm alanlarında yaygın olarak kullanılmaktadır. Boyut indirgeme, değişken seçimi, sıralama konuları vb. için kullanılır. Bu çalışmada, faktör analizi, R kütüphanesi "psych" ışığında incelenmiştir. Bu kütüphane FA ile ilgili birçok komut içermektedir. FA'in en önemli amaçlarından biri faktör yüklerini tahmin etmektir. "Psych" kütüphanesinde, faktörleme yöntemi (FM) adı verilen sekiz farklı faktör yükü tahmin edicisi kullanılmaktadır. Faktör yükleri "minres", "uls", "ols", "wls", "gls", "ml", fm = "pa", "minchi" ve "old.min" gibi FM ile tahmin edilir.

Çalışmanın asıl amacı, yukarıda bahsedilen sekiz yöntem, yan ve hata kareler ortalaması açısından simülasyon yardımıyla karşılaştırmaktır. Simülasyon çalışmasında, farklı korelasyon matris yapıları, rotasyon yöntemleri ve çeşitli örneklem büyüklükleri, verilerin geldiği dağılımının çok değişkenli olduğu durumlarda dikkate alınmaktadır. Çalışmada ikincil problem, faktör sayısı üzerindeki ki-kare testinin gücünü öngörmektir.

Faktör analizinde faktör sayısının belirlenmesi diğer bir problemdir. Bu, {psych} kütüphanesinde "ka.parallel" komutu veya Kaiser Criterion (bir değerden büyük özdeğer) kullanılarak yapılabilir. Çalışmadaki diğer bir konu, faktör sayısını belirleyen Paralel yönteminin, faktör sayısını doğru öngörme oranını simüle etmektir.

Son olarak, simülasyon çalışmasından elde edilen tüm deneyimler, faktör yüklerini tahmin etmek için hangi yöntemlerin kullanıldığı ve belirli koşullar altında faktörlerin sayısını tahmin etmek için hangi yöntemin kullanılması gibi soruların cevabı okuyucu ile paylaşılmaktadır.

Anahtar Kelimeler: Faktör analizi, Faktör yükleri, Hata kareler ortalaması, Monte Carlo simülasyon

ABSTRACT

MS THESIS

COMPARISON OF ESTIMATORS IN FACTOR ANALYSIS

Osman USTA

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE IN STATISTICS

Advisor: Prof.Dr. Coşkun KUŞ

2019, 52 Pages

Jury
Prof.Dr. Coşkun KUŞ
Asst.Prof.Dr. Ahmet PEKGÖR
Asst.Prof.Dr. Yunus AKDOĞAN

Factor Analysis (FA) are widely used in almost all area of the data science. It is used to dimension reduction, variables selection, ranking subjects and etc. In this study, factor analyses are investigated under the light of R library “psych”. This library contains a lot of commands related to FA. One of the biggest important aims of FA is of estimate the factor loadings. There are eight factoring methods (fm) are used in library “psych”. Factor loadings are calculated by fm such as "minres", "uls", "ols", "wls", "gls", "ml", fm="pa", "minchi" and "old.min".

Our main aim of the study is to compare the eight methods discussed above via simulation in terms of bias and mean square errors. In the simulation study, different correlation matrix structures, rotation methods and various sample sizes are considered when the underlying distribution of data is multivariate normal. Furthermore, non-normal data are also examined in the analysis. In the study, the secondary problem is of predict the power of this test on the number of factors.

Determination of the number of factors is another problem in factor analysis. This can be done by using command “fa.parallel” in library {psych} or Kaiser Criterion (eigenvalue greater than one). The last issue on the study is to simulate the true prediction rate of the Parallel Analysis method which determine the number of factors.

Finally, all experiences obtained from the simulation study are shared with the reader such as which methods are used to estimate the factor loadings and which method is used to predict the number of factors under certain conditions.

Keywords: Factor analysis, factor loadings, Monte Carlo simulation, mean squares errors

ÖNSÖZ

Çalışmada emeği geçen danışman hocam Prof.Dr. Coşkun Kuş ve bölüm hocalarıma teşekkür ederim.

Osman USTA
KONYA-2019

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **v**

ÖNSÖZ ..... **vi**

İÇİNDEKİLER ..... **vii**

1. GİRİŞ **1**

2. KAYNAK ARAŞTIRMASI **4**

3. SİMÜLASYON ÇALIŞMASI **7**

4. SONUÇLAR VE ÖNERİLER **24**

KAYNAKLAR **25**

ÖZGEÇMİŞ **45**

1. GİRİŞ

Faktör analizi sosyal bilimlerde ve tıp alanında çok kullanılan bir analizdir. Faktörün izin kullanımı ülkemizde genellikle yanlış uygulanmakta ve yanlış yorumlamaktadır veya yorumlamada çelişkiler meydana gelmektedir. Ayrıca istatistik bölümleri lisans programlarında, çok değişkenli istatistik, çok değişkenli istatistiksel analiz veya çok değişken istatistiksel sonuç çıkarımı isimli derslerde anlatılan faktör analizinde, ortogonal faktör model tanımı ve bu modeldeki parametrelerin tahmini konusu maalesef anlatılmaktadır. "Nasıl yapılır?", "Faktör sayısı nasıl belirlenir?", "Ortogonal faktör modelinde parametreler nasıl tahmin elde edilir?" gibi sorular farklı kovaryans ve farklı data senaryoları durumunda tartışılarak bilgi kirliliği ortadan kaldırılacaktır.

p bileşenli gözlemlenebilir rasgele vektör $\mathbf{X}$ in, ortalaması $\boldsymbol{\mu}$ ve kovaryans matrisi $\boldsymbol{\Sigma}$ olsun. Faktör modeli, $\mathbf{X}$ rasgele vektörünün, common faktör adı verilen $F_1, F_2, ..., F_m$ gözlenemeyen rasgele değişkenlerle lineer bağımlı olduğunu esas almaktadır. Faktör modelinde, hata veya spesifik faktör adı verilen $\varepsilon_1, \varepsilon_2, \ldots$ larda yer almaktadır. Faktör analizi modeli aşağıdaki gibi verilir:

$$
\begin{aligned}
X_1 - \mu_1 &= l_{11}F_1 + l_{12}F_2 + \cdots + l_{1m}F_m + \varepsilon_1 \\
X_2 - \mu_2 \; &l_{21}F_1 + l_{22}F_2 + \cdots + l_{2m}F_m + \varepsilon_2 \\
&\vdots \qquad\qquad \vdots \\
X_p - \mu_p &= l_{p1}F_1 + l_{p2}F_2 + \cdots + l_{pm}F_m + \varepsilon_p
\end{aligned}
$$

Faktör analizi modeli, matris gösterimi ile,

$$\mathbf{X}_{(px1)} - \boldsymbol{\mu}_{(px1)} = \mathbf{L}_{(pxm)}\mathbf{F}_{(mx1)} + \boldsymbol{\varepsilon}_{(px1)}$$

şeklinde yazılabilir.

l_{ij}, katsayısı *i.* değişkenin, *j* faktörü üzerindeki yüküdür. Bu nedenle $\mathbf{L}$ matrisi faktör yükleri matrisi olarak adlandırılır. Spesifik faktör ε_i, sadece X_i ile ilgili olduğu unutulmamalıdır. $X_1 - \mu_1, X_2 - \mu_2, \ldots, X_p - \mu_p$, gözlemlenemeyen $F_1, \ldots, F_m$, $\varepsilon_1, \ldots, \varepsilon_p$

rasgele değişkenleri cinsinden ifade edilir. Bu faktör modelini, regresyon modelinden ayırt eden ayrıntı $\mathbf{F}_{(mx1)}$ değişkenlerin gözlemlenemeyen olmasıdır.

m faktörlü ortogonal faktör modeli aşağıdaki gibi tanımlanır:

$$\mathbf{X}_{(px1)} = \boldsymbol{\mu}_{(px1)} + \mathbf{L}_{(pxm)}\mathbf{F}_{(mx1)} + \boldsymbol{\varepsilon}_{(px1)},$$

burada

$\boldsymbol{\mu}_i$, *i* değişkeninin ortalaması

$\boldsymbol{\varepsilon}_i$, *i.* faktörünün hatası

$\mathbf{F}_j$, *j.* ortak faktör

l_{ij} , *i.* değişkeninin *j.* faktördeki yükleri

şeklindedir. Gözlemlenemeyen rasgele vektörler $\mathbf{F}$ ve $\boldsymbol{\varepsilon}$ üzerinde aşağıdaki varsayımlar vardır.

$E(\mathbf{F}) = \mathbf{0}$ $\quad\quad$ $Cov(\mathbf{F}) = \mathbf{I}$

$E(\boldsymbol{\varepsilon}) = 0$ $\quad\quad$ $Cov(\boldsymbol{\varepsilon}) = \boldsymbol{\Psi}$, ($\boldsymbol{\Psi}$ diagonal matris)

Ortogonal Faktör Modeli İçin Kovaryans Yapısı aşağıdaki gibi verilir:

$$\begin{aligned}(\mathbf{X}-\boldsymbol{\mu})(\mathbf{X}-\boldsymbol{\mu})' &= (\mathbf{LF}+\boldsymbol{\varepsilon})(\mathbf{LF}+\boldsymbol{\varepsilon})' \\ &= (\mathbf{LF}+\boldsymbol{\varepsilon})((\mathbf{LF})'+\boldsymbol{\varepsilon}') \\ &= \mathbf{LF}(\mathbf{LF})' + \boldsymbol{\varepsilon}(\mathbf{LF})' + \mathbf{LF}\boldsymbol{\varepsilon}' + \boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'\end{aligned}$$

olmak üzere

$$\begin{aligned}\boldsymbol{\Sigma} = \mathbf{Cov}(\mathbf{X}) &= \mathbf{E}(\mathbf{X}-\boldsymbol{\mu})(\mathbf{X}-\boldsymbol{\mu})' \\ &= (\mathbf{LE}(\mathbf{FF}')\mathbf{L}' + \mathbf{E}(\boldsymbol{\varepsilon}\mathbf{F}')\mathbf{L} + \mathbf{LE}(\mathbf{F}\boldsymbol{\varepsilon}') + \mathbf{E}(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}') \\ &= \mathbf{LL}' + \boldsymbol{\psi}\end{aligned}$$

Faktör analizinde birincil amaç **L** yük matrisinin elemanlarının(yüklerin) tahmin edilmesidir. R deki “Psych” kütüphanesinde, faktörleme yöntemi (FM) adı verilen sekiz farklı faktör yükü tahmin edicisi kullanılmaktadır. Faktör yükleri "minres", "uls", "ols", "wls", "gls", "ml", fm = "pa", "minchi" ve "old.min" gibi FM ile tahmin edilir. Bu tezde amaç bu tahmin edicilerin bias ve MSE lerini karşılaştırmaktır. Bu yöntemler için Bollen(1989) ve Jöreskog(1969) çalışmalarına bakılabilir.

2. KAYNAK ARAŞTIRMASI

Shapiro, A., Kano, Y. (1987) çalışmasında, faktör sayısı sabitlenip yeni rasgele değişkenler modele eklendiğinde tahmin edicilerin asimptotik varyanslarının azaldığı gösterilmiştir.

Kano, Y. (1991) çalışmasında, açıklayıcı faktör analizinde, Ihara ve Kano nun iteratif olmayan tahmin edicinin asimptotik dağılımı bulmuşlardır. Ayrıca asimptotik dağılımın birinci dereceden momentinin var olmadığı gösterilmiştir.

Browne, M.W. (1987) çalışmasında, sınırsız faktör analizi modelini içeren gizli değişken modellerin bir sınıfı düşünüldüğünde normallik varsayımları altında türetilen asgari tutarsızlık test istatistikleri ve tahmincilerinin, ortak faktörler normal olarak dağıtılmadığında asimptotik özelliklerini korudukları ancak eşsiz faktörlerin çok değişkenli normal dağılıma sahip oldukları gösterilmiştir.

Bartholomew, D.J. (1984) çalışmasında, Bayes yeterlilik fikrini kullanarak veri azaltımına dayanan faktör analizi ve ilgili gizli değişken metotlara yeni bir yaklaşım gösterilmektedir.

Wallace, C.S., Freeman P.R. (1992) çalışmalarında, Bir MML tahmini varlığının şartının faktör modelini tercih eden log olasılık oranının, null (faktör içermeyen) hipotezler altında beklenen değeri aştığı gösterilmiştir.

Bartholomew, D.J. (1984) çalışmasında, kategorik verilerin faktör analizi için sonuçlar elde edilmiştir.

Browne, M.W., Shapiro, A. (1987) çalışmalarında, Faktör analiz modelinin uygunluğuna ilişkin normal teori testlerine basit düzeltmeler, ortak faktörlerin keyfi dağılımlara sahip olduğu ve hataların eliptik sınıfından bir dağılıma sahip olduğu durum için sağlandığı gösterilmiştir.

Laake, P. (1988) çalışmasında, asimptotik dağılımın yoğunluklarının ve yoğunluğunun grafiksel gösterimleri belirli bir örnek için gösterilmiştir.

Darton, R.A. (1980) verilerin belirli parametrelere sahip bir modele uygunluğunun incelenmesi gösterilmiştir.

Jung and Lee çalışmalarında, faktör analizi çalışmalarında sıklıkla kullanılan en çok olabilirlik ve temel bileşenler yöntemine alternatif “regularized explonatory faktör analizi” metodunu ele almışlardır. Özellikle küçük örnek hacimlerinde bu metotların nasıl performans sergilediği hakkında çok az çalışma olması neticesiyle, örneklem kovaryans matrisi tekil olan ve örnek hacmi de 50’nin altında olması koşulları altında

bir simülasyon çalışması ve bir gerçek veri uygulaması gerçekleştirmişlerdir.(Jung ve Lee, 2011)

Winter ve ark. çalışmalarında, küçük örnek hacimlerinde faktör analizinin nasıl performans sergileyeceğini, farklı faktör yükleri, farklı faktör sayıları ve farklı değişken sayılarını dikkate alan bir simülasyon çalışması yapmışlardır.(Winter ve ark., 2009)

Barendse ve arkadaşları çalışmalarında, kesikli data durumunda faktör analizinin performansını araştırmışlardır. Bu amaçla, datanın hem sürekli, hem de iki ve daha fazla kategorili olduğu durumu, farklı faktör yükleri (orta ve yüksek) altında, farklı örnek hacimlerinde (küçük ve büyük) ve de basit ve karmaşık faktör yapısını ele alan bir simülasyon çalışması tasarlamışlardır. Her bir durum için en çok olabilirlik (maximum likelihood of covariances, maximum likelihood of polychoric correlation ve robust maximum likelihood) ağırlıklandırılmış en küçük kareler (WLS ve robust WLS) tahmin yöntemlerinde 3 farklı uyum kriterinde (ki-kare testi, root mean square error of approximation ve root mean square residual) ele alınmıştır.(Barendse ve ark., 2015)

Karaman ve ark. çalışmalarında, faktör analizinde kullanılan faktör çıkarma yöntemlerinin açıklanan varyans ve tahmini faktör sayısı bakımından çeşitlik örneklem büyüklüğü ve ortak varyans düzeyi koşullarına göre simülasyon yardımıyla karşılaştırmışlardır. (Karaman ve ark., 2017)

Kılıç ve Uysal çalışmalarında, faktör sayısını belirleme yöntemlerini farklı faktör sayıları, örneklem büyüklükleri, değişken sayıları, faktör yükleri ve kullanılan korelasyon matrisi açısından karşılaştıran bir simülasyon çalışması yapmışlardır.(Kılıç ve Uysal, 2019)

Jung çalışmasında, küçük örneklemin faktör analizi için bir problem teşkil etmesinden yola çıkarak, bu gibi durumlarda kullanılan geleneksel yaklaşım olan en küçük kareler yöntemine alternatif iki yaklaşımı ele alan bir simülasyon çalışması yapmıştır. Bu yaklaşımlar, regularized faktör analizi ve genelleştirilmiş faktör analizidir. Simülasyon çalışmasında, örnek hacmi, faktörlerin paylaştıkları ortak varyans ve yüksek faktör sayısının tahmini durumları için farklı senaryoları incelenmiştir.(Jung, 2013)

Hauck ve arkadaşları çalışmalarında, literatürde sürekli latent değişkenlere ilşkin var olan metotların performanslarını karşılaştıran bir simülasyon çalışması yapmıştır. Çalışmalarında ele aldıkları modeller, klasik test teorisi, temel bileşenler, en çok olabilirliğe dayalı faktör analizi, en küçük sıra sayılarına dayalı faktör analizi, sıralama

esaslı ölçek modeli, aşamalı yanıt modeli ve ağırlıklandırılmış en küçük kareler ve düzeltilmiş varyansa dayalı doğrulayıcı faktör analizidir.(Hauck F. ve ark., 2014)

Christensen ve arkadaşları çalışmalarında, faktör sayısının belirlenmesinde kullanılan paralel analiz, kaiser-guttman'ın 1'den büyük özdeğeri kuralı, çoklu kısmi ortalama prosedürü ve en çok olabilirliğe dayalı (BIC ve EBIC gibi uyum indekslerini kullanan) yöntemlere alternatif açıklayıcı grafik analizi adlı yeni bir yaklaşımı geliştirmişlerdir. Farklı senaryolar altında yaptıkları simülasyon çalışmaları ile bu yöntemlerin performanslarını karşılaştırmışlardır.(Christensen ve ark., 2019)

Coughlin ve arkadaşları çalışmalarında, korelasyonsuz faktör yapısını dikkate alan ve datanın özelliklerini simule edebilen bir metot geliştirmişlerdir. Bu metot yardımıyla, araştırıcılara değişken sayısını, değişkenler arası paylaşılan varyansı ve ortak faktör sayısını değiştirerek korelasyon matrisi üretmeye imkan sağlamaktadırlar. Farklı koşullar altında simülasyon çalışmasıyla metotlarını örneklendirmişlerdir (Coughlin ve ark., (2013)).

Koğar çalışmasında, farklı istatistiksel modeller ve kontrol altına alınan çeşitli değişkenleri içeren modellerde gerçek ve tahmini faktör skorları arasındaki ilişkiyi tanımlamayı amaçlayan bir simülasyon çalışması yapmıştır. Simülasyon çalışmasıyla farklı örnek hacimlerinde ve farklı dağılım tiplerinde faktör skorlarının etkilerini karşılaştırmışlardır(Koğar, 2018).

Beauducel çalışmasında, temel bileşenler analizi, temel eksen faktör analizi, alfa faktörleştirme ve en çok olabilirlik gibi faktör çıkarma metotlarının değişken yapısına göre değişimin etkisini ölçmek amacıyla bir simülasyon çalışması yapmıştır. Simülasyonlarda farklı faktör sayılarında ve örnek hacimlerinde metotların performanslarını değerlendirmiştir(Beauducel, 2001).

Çokluk ve Koçak çalışmalarında, faktör sayısını belirlemekte kullanılan özdeğer ve yamaç grafiğine alternatif metot olan Horn'un paralel analizine ilişkin simülasyon ve gerçek veri uygulaması yapmışlardır(Çokluk ve Koçak, 2016).

Finch çalışmasında, literatürde sıklıkla kullanılan sekiz faktör döndürme tekniğini, veri yapısının ikili olduğu durumda simülasyon yardımıyla performanslarını karşılaştırmıştır(Finch, 2011).

Wu and Bentler çalışmalarında, ikili yanıt değişkeni durumunda faktör analizi hesaplamalarında meydana gelen karmaşıklığı gidermek üzere H-olabilirlik metodunu kullanmışlar ve sonuçları simülasyon yardımıyla karşılaştırmışlardır(Wu ve Bentler, 2012).

3. SİMÜLASYON ÇALIŞMASI

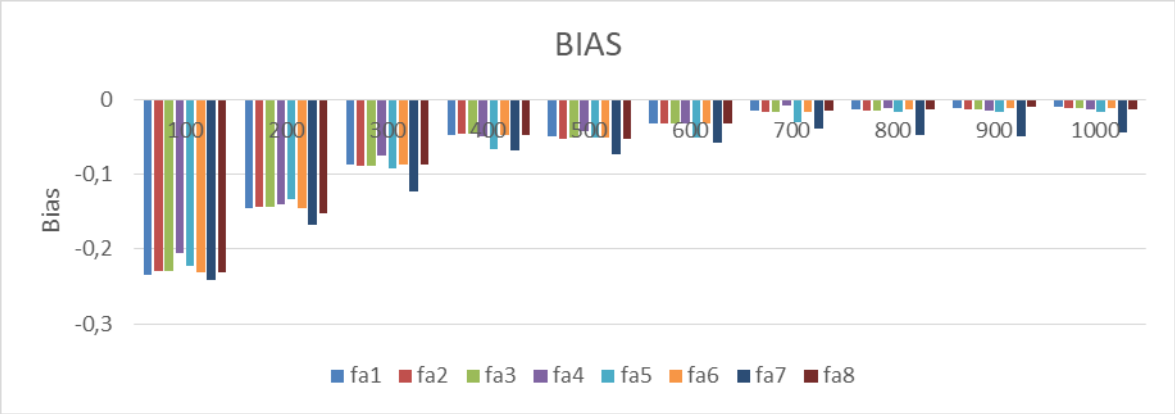
Bu bölümde, bazı simülasyon sonuçları tartışılmıştır. Simülasyonda 1000 deneme yapılmıştır. Korelasyon matrisi ve kovaryans matrisi esas alınarak 8 faktoring yöntemi ve 5 rotasyon yöntemi uygulanmıştır. Simülasyon çalışmasında veriler çok değişkenli normal dağılımdan üretilmiştir. Analizler hem korelasyon hem de kovaryans matrisine dayalı yapılmış, varimax, oblimin, promax ve quartimax rotasyonları için sonuçlar elde edilmiştir. Örneklem hacmi 100,200,…,1000 olarak ele alınmıştır. Kullanılan R kodu ekte sunulmuştur.

İki farklı örneklem durumu için simülasyon yapılmıştır. 1. Durum için örneklemler aşağıdaki koşullarda üretilmiştir:

$$E(\mathbf{X}) = (1,1,1,1,1) \text{ and } Cov(\boldsymbol{\varepsilon}) = diag(0.2,0.3,0.4,0.5,0.5) \text{ and } \mathbf{L} = \begin{pmatrix} .56 & .82 \\ .78 & -.53 \\ .65 & .75 \\ .94 & .10 \\ .80 & -.54 \end{pmatrix}.$$

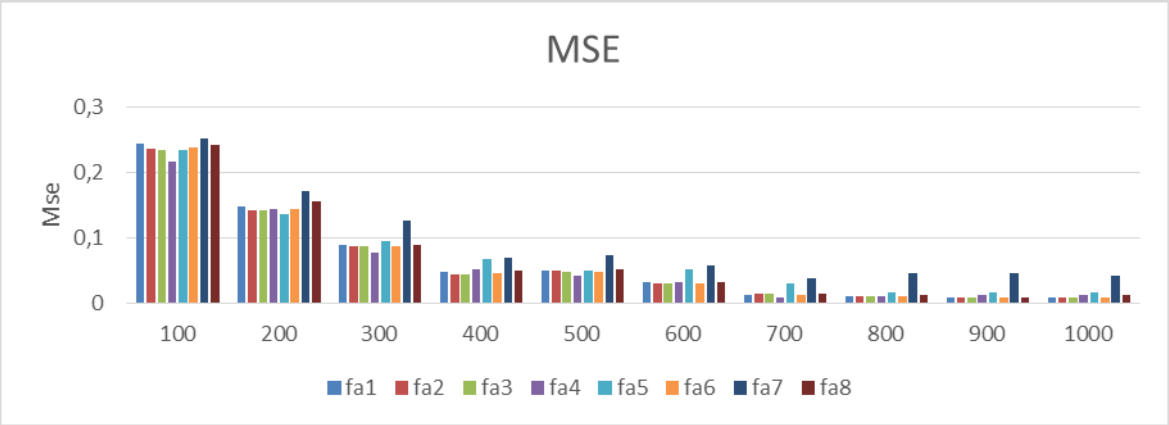
Bu bölümde FM ler fa1: ml, fa2:minres, fa3:uls, fa4:wls, fa5:gls, fa6:pa, fa7:minchi, fa8:oldmin şeklinde simgeleştirilmiştir:

Aşağıda verilen Şekil 1-8 deki sonuçlar kovaryans matrisi analiz edilerek elde edilmiştir.



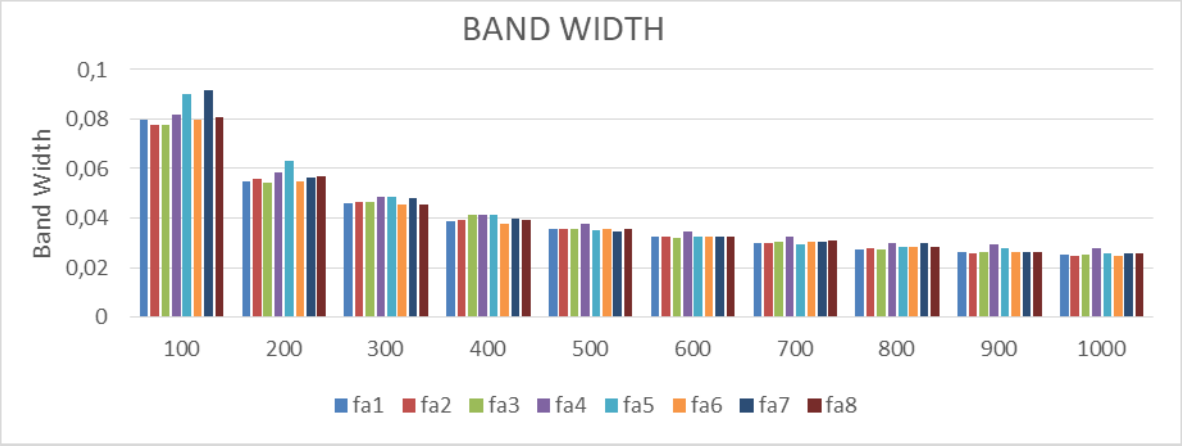
Şekil 1. Varimax rotasyonlu FM lerin yanı

Şekil 1 den çıkan sonuçlar şöyle sıralanabilir: Tüm FM lerin için, örneklem hacmi arttığında yan sıfıra gitmektedir. Tüm örnek hacimlerinde, FM Fa4 miminum biasa sahiptir. Fa7 bias bakımından en kötü FM dir. Tüm FM ler negative biaslıdır.



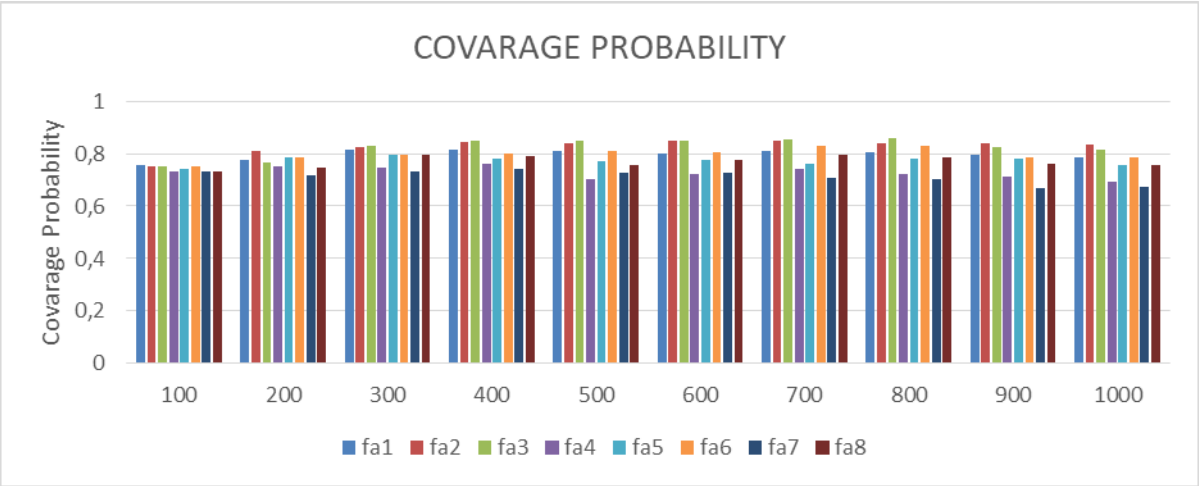
Şekil 2. Varimax rotasyonlu FM lerin MSE si

Şekil 2 çıkan sonuçlar şöyle sıralanabilir: Tüm FM lerin için, örneklem hacmi arttığında MSE sıfıra gitmektedir. Hemen hemen tüm örnek hacimlerinde FM Fa4 minimum MSE ye sahiptir. Fa7 en kötü MSE ye sahip FM dir.



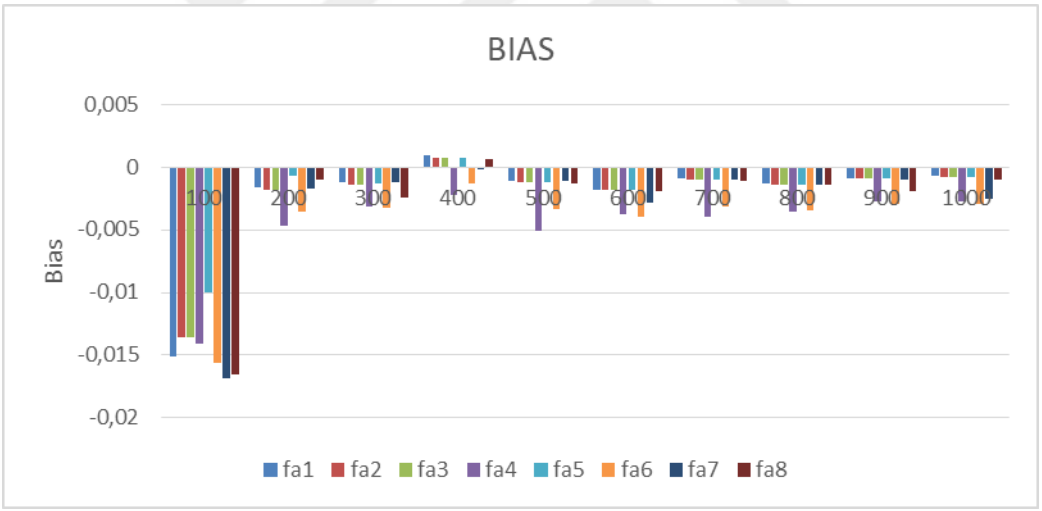
Şekil 3. Varimax rotasyonlu FM lere dayalı güven aralıklarının ortalama uzunluğu

Şekil 3 den, örneklem hacmi arttığında güven aralıklarının genişliği azalmakta ve sıfıra yaklaşmaktadır.



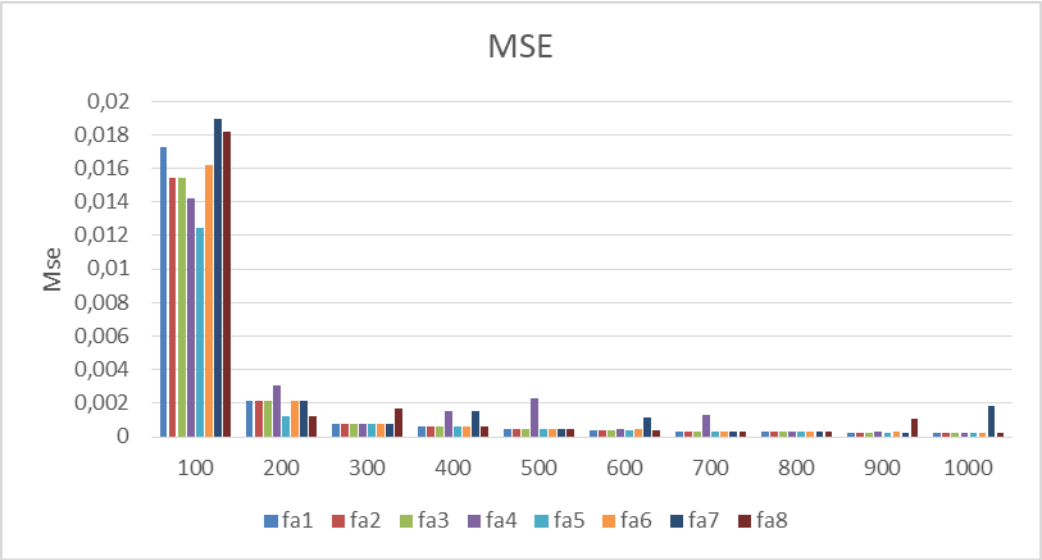
Şekil 4. Varimax rotasyonlu FM lere dayalı güven aralıklarının kapsama olasılığı

Şekil 4 den, bootstarp güven aralıklarının kapsama olasılıklarının örneklem hacmi arttığında bile nominal değer %95 e yaklaşamaktadır. Bunun nboot=10 olmasından kaynaklı olduğu düşünülmektedir.



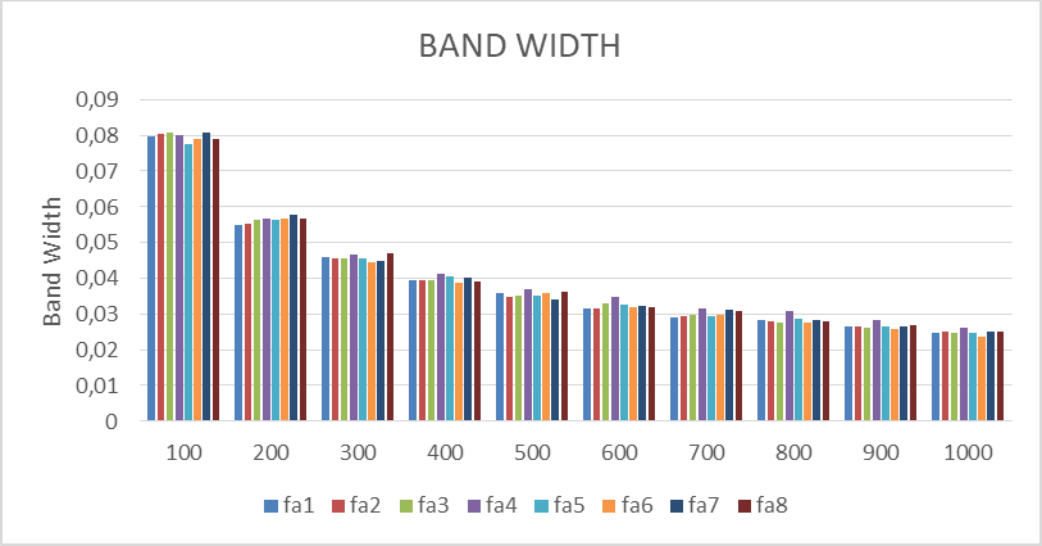
Şekil 5. Oblimin rotasyana dayalı FM lerin yanı

Şekil 5 den çıkan sonuçlar şöyle sıralanabilir: Tüm FM lerin için, örneklem hacmi arttığında yan sıfıra gitmektedir. Tüm örnek hacimlerinde, FM Fa5 miminum yana sahiptir. Fa4 ve Fa6 bias bakımından en kötü FM lerdir. Genelde FM ler negative biaslıdır.



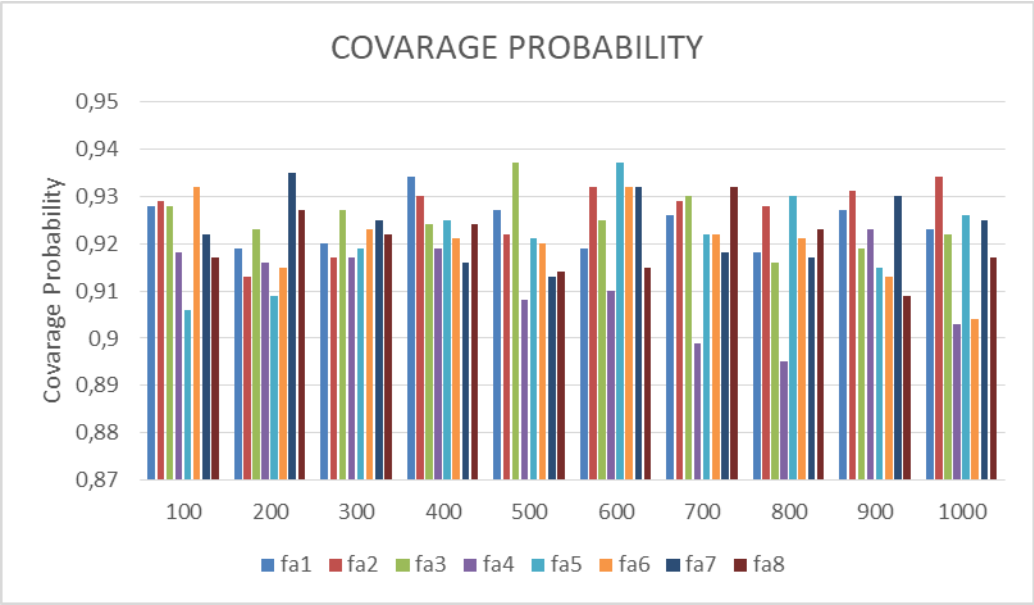
Şekil 6. Oblimin rotasyana dayalı FM lerin MSE si

Şekil 6 dan çıkan sonuçlar şöyle sıralanabilir: Tüm FM lerin için, örneklem hacmi arttığında MSE sıfıra gitmektedir. Hemen hemen tüm örnek hacimlerinde FM Fa5 minimum MSE ye sahiptir. Örneklem hacmi 300 den büyük olduğunda tüm metodlar hemen hemen aynı MSE ye sahiptir.



Şekil 7. Oblimin rotasyana ve FM lere dayalı güven aralıklarının ortalama uzunluğu

Şekil 7 den, örneklem hacmi arttığında güven aralıklarının genişliği azalmakta ve sıfıra yaklaşmaktadır.



Şekil 8. Oblimin rotasyana dayalı FM lere dayalı güven aralıklarının kapsama olasılığı

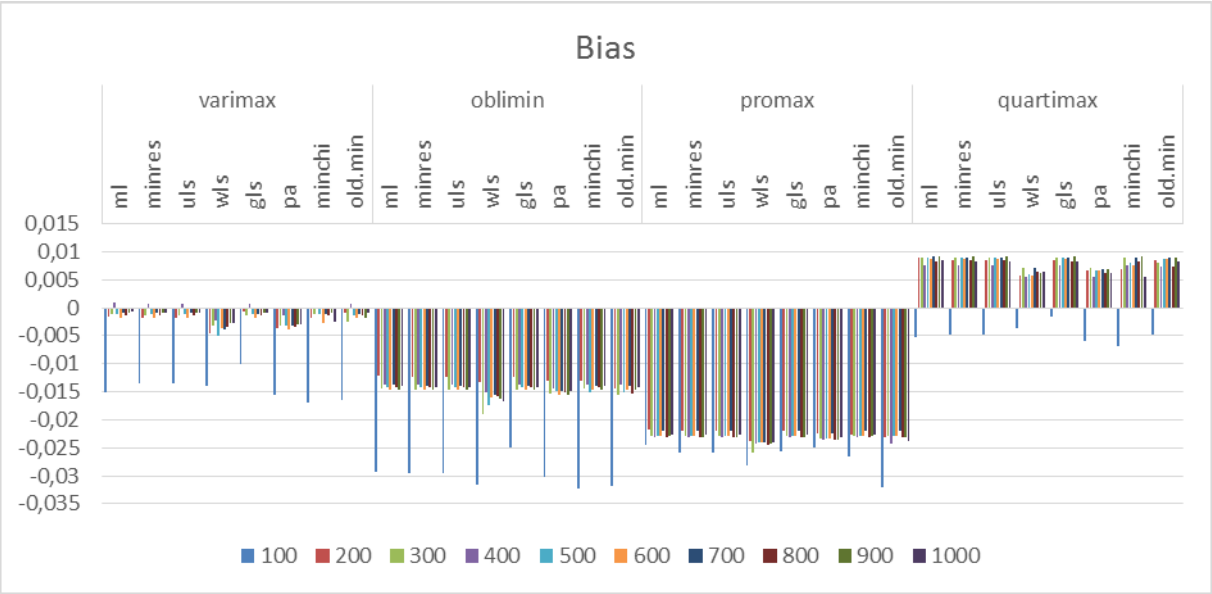
Şekil 8 den, bootstarp güven aralıklarının kapsama olasılıklarının örneklem hacmi arttığında bile nominal değer %95 e yaklaşamaktadır. Bunun nboot=10 olmasından kaynaklı olduğu düşünülmektedir.

Çizelge 1. Analizde Korelasyon Matrisi kullanıldığında Birinci Parametre için 8 FM Metodunun Farklı Rotasyonlar için Biasları

		100	200	300	400	500	600	700	800	900	1000
varimax	ml	-0.01505	-0.00161	-0.00116	0.000896	-0.00109	-0.00181	-0.00091	-0.00135	-0.00095	-0.00072
	minres	-0.01353	-0.00185	-0.00136	0.000696	-0.00122	-0.00186	-0.00102	-0.00138	-0.00085	-0.00084
	uls	-0.01354	-0.00185	-0.00136	0.000696	-0.00122	-0.00186	-0.00102	-0.00138	-0.00085	-0.00084
	wls	-0.01405	-0.00466	-0.00319	-0.00226	-0.00509	-0.00375	-0.00398	-0.00353	-0.00275	-0.00278
	gls	-0.01003	-0.0007	-0.00131	0.000722	-0.00122	-0.00186	-0.00103	-0.00139	-0.00085	-0.00084
	pa	-0.0156	-0.00358	-0.00327	-0.00129	-0.00331	-0.00392	-0.00315	-0.0035	-0.00292	-0.00298
	minchi	-0.01685	-0.00174	-0.00117	-0.00018	-0.00112	-0.00286	-0.00105	-0.00136	-0.00101	-0.00256
	old.min	-0.01649	-0.00098	-0.00242	0.000629	-0.00131	-0.00193	-0.00113	-0.00146	-0.00188	-0.00095
oblimin	ml	-0.02921	-0.01205	-0.01447	-0.01366	-0.0141	-0.01464	-0.01381	-0.01423	-0.01458	-0.01406
	minres	-0.02964	-0.01228	-0.0146	-0.01366	-0.01408	-0.01459	-0.01387	-0.01422	-0.01455	-0.01408
	uls	-0.02964	-0.01228	-0.0146	-0.01366	-0.01408	-0.01459	-0.01387	-0.01422	-0.01455	-0.01408
	wls	-0.03165	-0.01335	-0.01903	-0.01505	-0.0175	-0.01609	-0.01547	-0.01582	-0.01631	-0.01682
	gls	-0.02498	-0.01228	-0.01458	-0.01367	-0.01409	-0.01459	-0.01389	-0.01423	-0.01455	-0.01408
	pa	-0.03025	-0.01304	-0.01544	-0.01453	-0.01491	-0.01546	-0.01477	-0.01518	-0.01551	-0.01498
	minchi	-0.03224	-0.013	-0.01449	-0.01363	-0.01504	-0.01464	-0.01386	-0.01426	-0.01458	-0.01407
	old.min	-0.03178	-0.01446	-0.01561	-0.01374	-0.01518	-0.01465	-0.01395	-0.01531	-0.0146	-0.01416
promax	ml	-0.02454	-0.02179	-0.02285	-0.02312	-0.02291	-0.02283	-0.02194	-0.02317	-0.02298	-0.02273
	minres	-0.02586	-0.02197	-0.0229	-0.02321	-0.02287	-0.02288	-0.02194	-0.02312	-0.02303	-0.02272
	uls	-0.02586	-0.02197	-0.0229	-0.02321	-0.02287	-0.02287	-0.02194	-0.02312	-0.02303	-0.02272
	wls	-0.02824	-0.02382	-0.02594	-0.02433	-0.02399	-0.02402	-0.02413	-0.02438	-0.02434	-0.02404
	gls	-0.02564	-0.02192	-0.02285	-0.02321	-0.02288	-0.02288	-0.02194	-0.02314	-0.02302	-0.02271
	pa	-0.02507	-0.0224	-0.0233	-0.02365	-0.02331	-0.02329	-0.02242	-0.02362	-0.02348	-0.02315
	minchi	-0.02665	-0.02277	-0.02288	-0.02311	-0.0229	-0.02282	-0.02192	-0.02313	-0.02299	-0.02272
	old.min	-0.03214	-0.02305	-0.02291	-0.02433	-0.02293	-0.02291	-0.022	-0.02319	-0.02307	-0.02382
quartimax	ml	-0.0052	0.008841	0.009036	0.007492	0.00901	0.008683	0.009175	0.008271	0.009204	0.008377
	minres	-0.00474	0.008451	0.00896	0.007489	0.008873	0.00877	0.008955	0.00836	0.009048	0.008299
	uls	-0.00474	0.00845	0.008959	0.007489	0.008868	0.00877	0.008955	0.00836	0.009048	0.008299
	wls	-0.00377	0.005823	0.007145	0.005412	0.005913	0.005785	0.006986	0.00634	0.006077	0.006357
	gls	-0.00169	0.008518	0.009006	0.007473	0.008862	0.008772	0.008965	0.008356	0.009049	0.008289
	pa	-0.00588	0.006728	0.007014	0.005425	0.006723	0.006686	0.006865	0.006265	0.006896	0.006152
	minchi	-0.00695	0.00683	0.008937	0.007448	0.008056	0.007521	0.009011	0.008265	0.009161	0.005605
	old.min	-0.00488	0.008396	0.007941	0.007347	0.008775	0.008693	0.008894	0.007363	0.008957	0.0082

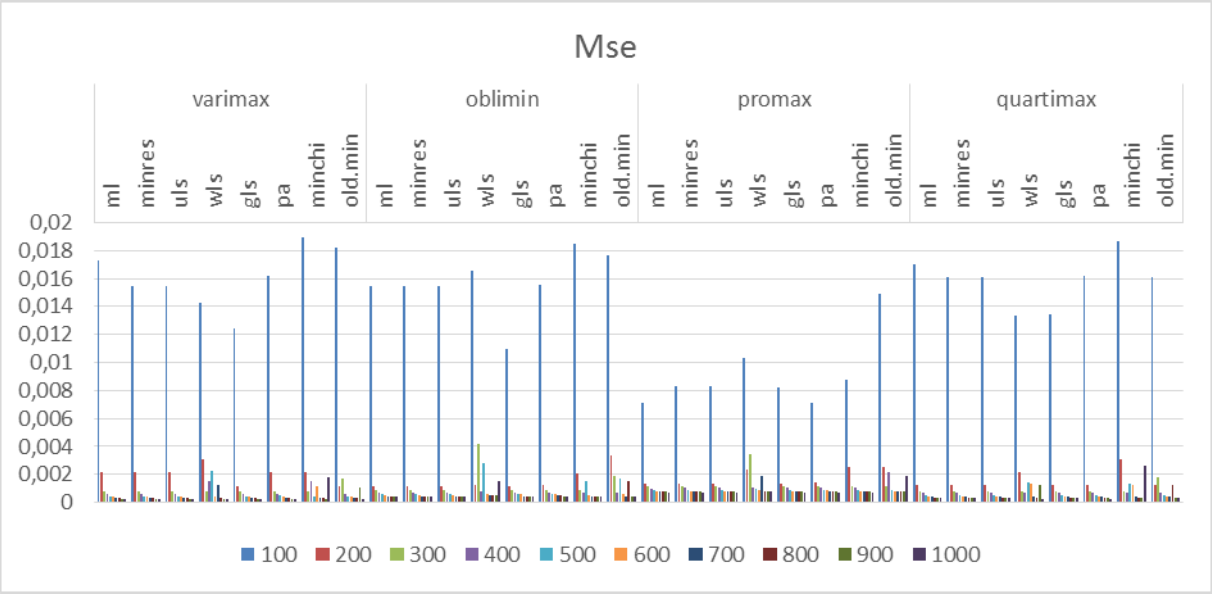
Çizelge 2. Analizde Korelasyon Matrisi kullanıldığında Birinci Parametre için 8 FM Metodunun Farklı Rotasyonlar için MSE değerleri

		100	200	300	400	500	600	700	800	900	1000
varimax	ml	0.017264	0.002163	0.000766	0.000584	0.00045	0.00041	0.000313	0.000291	0.000265	0.000244
	minres	0.015415	0.002156	0.000764	0.000582	0.000454	0.000408	0.000315	0.000289	0.000263	0.000245
	uls	0.015415	0.002156	0.000764	0.000582	0.000454	0.000408	0.000315	0.000289	0.000263	0.000245
	wls	0.014235	0.003044	0.000778	0.001554	0.002279	0.000426	0.001281	0.000305	0.000276	0.000259
	gls	0.012445	0.00119	0.000762	0.000582	0.000453	0.000408	0.000315	0.000289	0.000263	0.000245
	pa	0.016234	0.002166	0.000793	0.000599	0.000474	0.00043	0.000332	0.000307	0.000277	0.000259
	minchi	0.018944	0.002167	0.000764	0.001492	0.000452	0.001154	0.000317	0.000292	0.000265	0.001831
	old.min	0.018192	0.001194	0.001685	0.000584	0.000453	0.000408	0.000316	0.000289	0.001106	0.000244
oblimin	ml	0.015441	0.001182	0.000881	0.000701	0.000595	0.000556	0.000444	0.000453	0.000417	0.000392
	minres	0.015499	0.00119	0.000882	0.000699	0.000595	0.000557	0.000446	0.000452	0.000417	0.000393
	uls	0.015499	0.00119	0.000882	0.000699	0.000595	0.000557	0.000446	0.000452	0.000417	0.000393
	wls	0.016594	0.001229	0.004182	0.000744	0.002794	0.00061	0.000501	0.000507	0.000475	0.001498
	gls	0.010997	0.001191	0.000881	0.000699	0.000596	0.000557	0.000447	0.000452	0.000418	0.000393
	pa	0.015544	0.001235	0.000923	0.000739	0.00063	0.000592	0.000479	0.000485	0.00045	0.000424
	minchi	0.018492	0.002111	0.000882	0.0007	0.001549	0.000555	0.000447	0.000454	0.000417	0.000393
	old.min	0.017698	0.003389	0.001875	0.000702	0.001687	0.00056	0.000448	0.001552	0.00042	0.000396
promax	ml	0.007085	0.001345	0.001156	0.001015	0.000883	0.000809	0.000748	0.00076	0.000741	0.000699
	minres	0.008281	0.001349	0.001156	0.001023	0.000886	0.000812	0.000742	0.000756	0.000741	0.000699
	uls	0.008281	0.001349	0.001156	0.001023	0.000886	0.000812	0.000742	0.000756	0.000741	0.000699
	wls	0.010307	0.002371	0.003435	0.001086	0.000946	0.000874	0.0019	0.000824	0.000809	0.000767
	gls	0.008233	0.001345	0.001154	0.001023	0.000886	0.000812	0.000743	0.000756	0.000741	0.000698
	pa	0.007127	0.001386	0.00119	0.001058	0.000913	0.000837	0.000769	0.000785	0.000767	0.000724
	minchi	0.008772	0.002529	0.001161	0.001016	0.000882	0.000809	0.000748	0.000759	0.000742	0.000699
	old.min	0.014947	0.002522	0.001158	0.00219	0.000888	0.000814	0.000745	0.000759	0.000743	0.00187
quartimax	ml	0.017056	0.00128	0.000819	0.000692	0.000555	0.00046	0.000445	0.000367	0.000345	0.0003
	minres	0.016125	0.001274	0.000829	0.000691	0.000553	0.000464	0.000445	0.000368	0.000343	0.000299
	uls	0.016125	0.001274	0.000829	0.000691	0.000553	0.000464	0.000445	0.000368	0.000343	0.000299
	wls	0.013324	0.002168	0.00081	0.000677	0.001425	0.001356	0.000418	0.000343	0.001229	0.000275
	gls	0.013436	0.001274	0.00083	0.00069	0.000552	0.000465	0.000444	0.000367	0.000343	0.000298
	pa	0.01616	0.001276	0.000817	0.00068	0.000534	0.000444	0.000418	0.000342	0.000312	0.000274
	minchi	0.018654	0.003099	0.00082	0.000693	0.001331	0.001236	0.000451	0.000367	0.000345	0.002617
	old.min	0.016119	0.001274	0.001751	0.00069	0.00055	0.000464	0.000445	0.001216	0.00034	0.000298



Şekil 9. Çizelge 1 in Grafik olarak gösterimi

Şekil 9 dan en düşük biasların tüm FM metodlarda varimax rotasyonunda ulaşılmaktadır. Fa1, fa2, fa3, fa5 ve fa8 hemen hemen benzer biasa sahip ve diğerlerinden düşüktür.

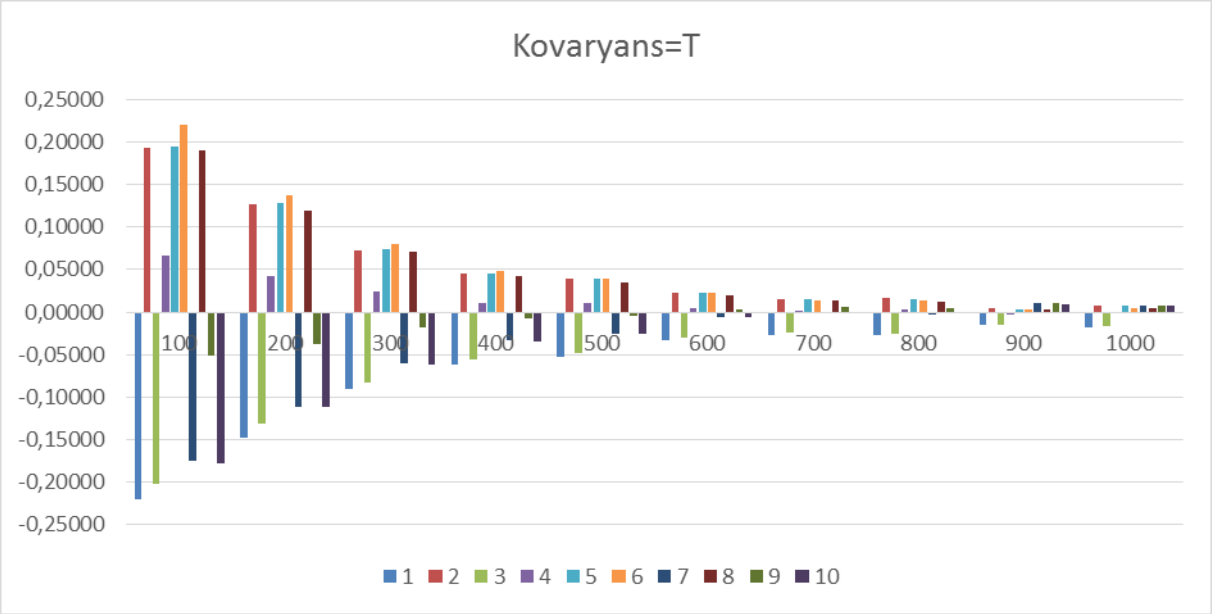


Şekil 10. Çizelge 2 nin grafik olarak gösterimi

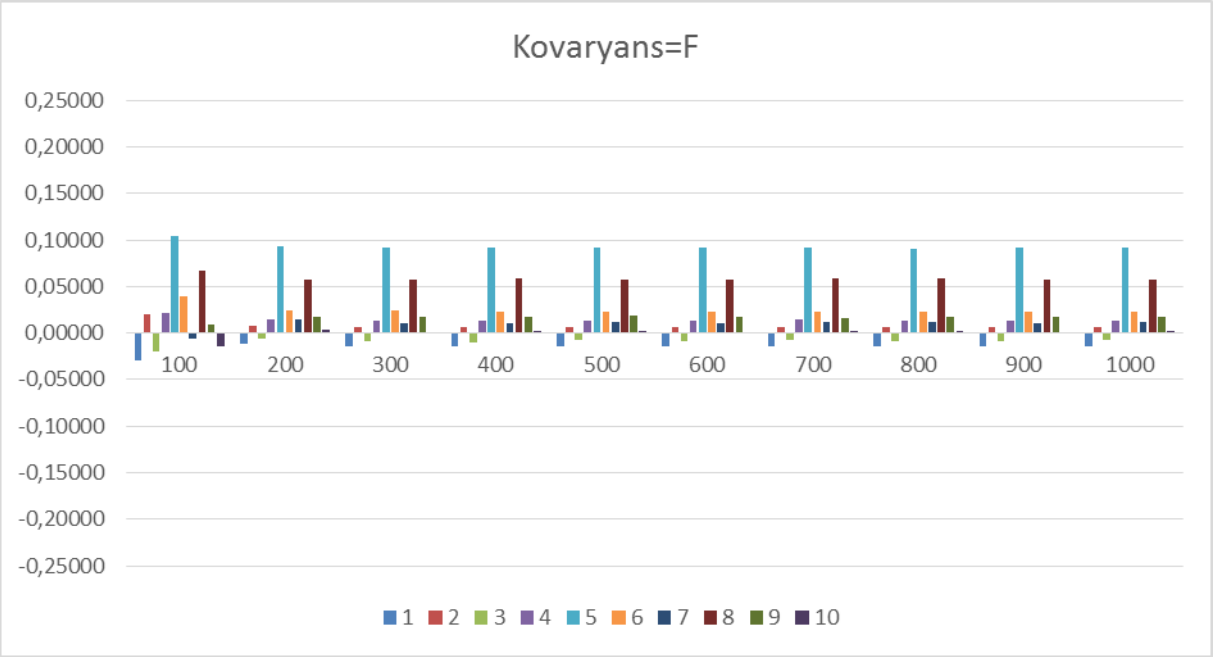
Şekil 10 dan, örneklem hacmi arttığında, MSE lerin azaldığı görülmektedir.

Çizelge 3. Yük Matrisinin Tüm Parametrelerinin MLTahmin Edicilerinin Oblim Rotasyonu Sonrası Biasları

Kullanılan Matris	par	100	200	300	400	500	600	700	800	900	1000
Kovaryans	1	-0.22084	-0.14737	-0.09080	-0.06147	-0.05291	-0.03302	-0.02719	-0.02713	-0.01456	-0.01732
	2	0.19266	0.12660	0.07312	0.04493	0.03916	0.02215	0.01457	0.01622	0.00495	0.00688
	3	-0.20146	-0.13178	-0.08361	-0.05621	-0.04767	-0.02980	-0.02467	-0.02535	-0.01455	-0.01648
	4	0.06645	0.04259	0.02357	0.01114	0.01136	0.00499	0.00075	0.00350	-0.00310	-0.00195
	5	0.19449	0.12778	0.07424	0.04541	0.03874	0.02263	0.01504	0.01568	0.00333	0.00702
	6	0.22009	0.13798	0.08068	0.04811	0.03946	0.02263	0.01403	0.01326	0.00240	0.00485
	7	-0.17450	-0.11086	-0.06043	-0.03252	-0.02565	-0.00667	-0.00142	-0.00233	0.01023	0.00752
	8	0.19042	0.11983	0.07146	0.04267	0.03516	0.01981	0.01310	0.01251	0.00300	0.00523
	9	-0.05144	-0.03697	-0.01873	-0.00752	-0.00426	0.00317	0.00537	0.00460	0.01014	0.00784
	10	-0.17728	-0.11163	-0.06138	-0.03392	-0.02551	-0.00645	-0.00101	-0.00196	0.00963	0.00824
Korelasyon	1	-0.02921	-0.01205	-0.01447	-0.01366	-0.01410	-0.01464	-0.01381	-0.01423	-0.01458	-0.01406
	2	0.02013	0.00775	0.00703	0.00629	0.00709	0.00669	0.00662	0.00703	0.00678	0.00687
	3	-0.01950	-0.00613	-0.00827	-0.00958	-0.00782	-0.00899	-0.00785	-0.00806	-0.00894	-0.00792
	4	0.02121	0.01507	0.01358	0.01357	0.01339	0.01386	0.01416	0.01301	0.01376	0.01301
	5	0.10441	0.09324	0.09247	0.09236	0.09167	0.09191	0.09130	0.09099	0.09196	0.09147
	6	0.03959	0.02411	0.02396	0.02308	0.02350	0.02325	0.02289	0.02277	0.02264	0.02281
	7	-0.00568	0.01434	0.01048	0.01018	0.01210	0.01037	0.01136	0.01158	0.01059	0.01169
	8	0.06670	0.05733	0.05730	0.05816	0.05783	0.05809	0.05826	0.05821	0.05810	0.05791
	9	0.00980	0.01733	0.01767	0.01767	0.01916	0.01760	0.01648	0.01808	0.01748	0.01779
	10	-0.01386	0.00301	0.00126	0.00164	0.00195	0.00117	0.00273	0.00166	0.00107	0.00223



Şekil 11. Çizelge 3 ün grafik olarak gösterimi(Analizde Kovaryans Matrisi Kullanılmıştır)

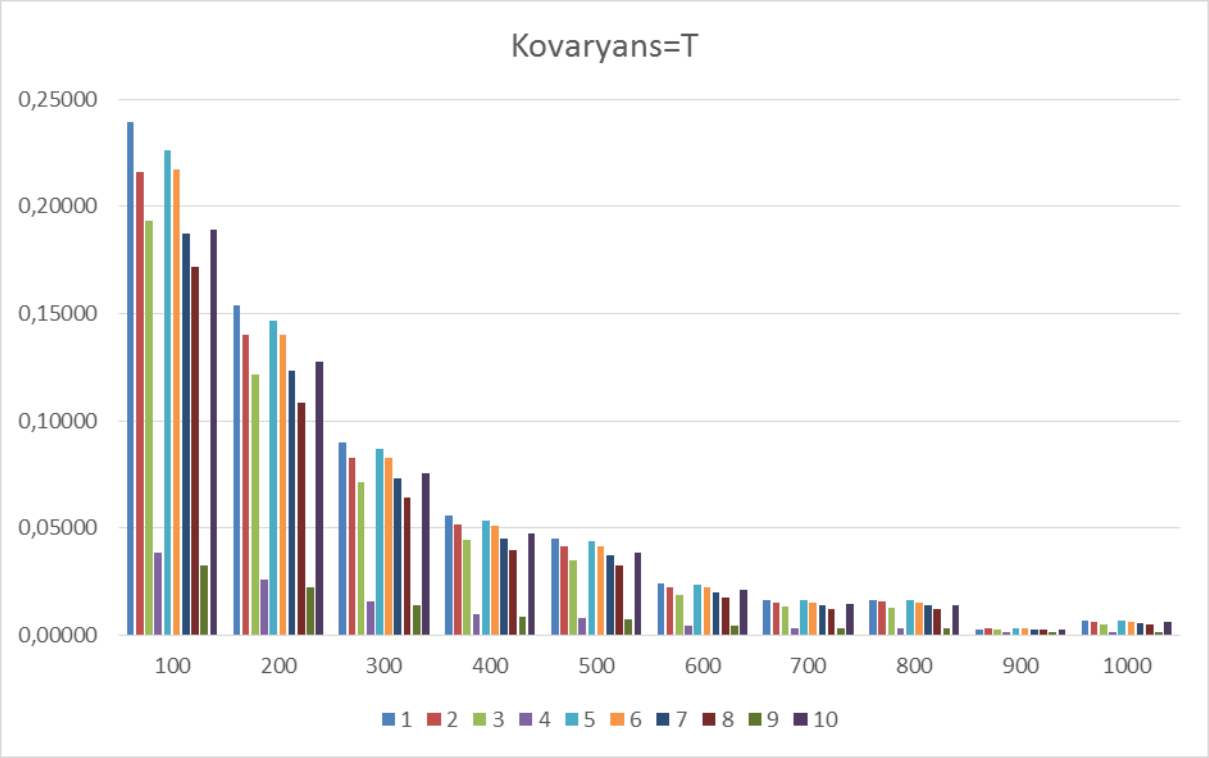


Şekil 12. Çizelge 3 ün grafik olarak gösterimi(Analizde Korelasyon Matrisi Kullanılmıştır)

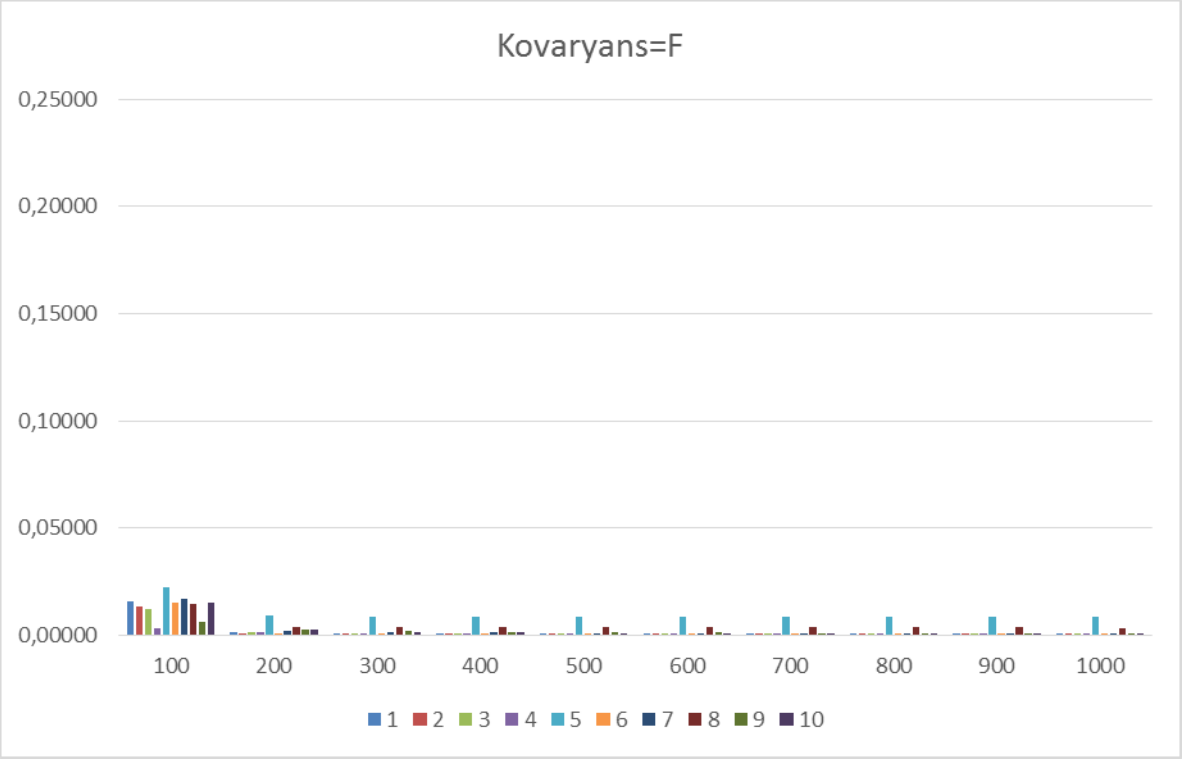
Çizelge 4. Yük Matrisinin Tüm Parametrelerinin ML Tahmin Edicilerinin Oblim Rotasyonu Sonrası MSE değerleri

Kullanılan Matris	par	100	200	300	400	500	600	700	800	900	1000
Kovaryans	1	0.23956	0.15405	0.09011	0.05587	0.04502	0.02401	0.01659	0.01648	0.00282	0.00658
	2	0.21609	0.14002	0.08243	0.05135	0.04172	0.02212	0.01527	0.01547	0.00291	0.00622
	3	0.19328	0.12167	0.07107	0.04414	0.03500	0.01888	0.01325	0.01291	0.00232	0.00509
	4	0.03830	0.02577	0.01541	0.00977	0.00806	0.00455	0.00330	0.00344	0.00116	0.00160
	5	0.22633	0.14653	0.08689	0.05350	0.04399	0.02344	0.01626	0.01616	0.00327	0.00664
	6	0.21705	0.13997	0.08240	0.05124	0.04139	0.02221	0.01535	0.01523	0.00292	0.00621
	7	0.18730	0.12353	0.07340	0.04524	0.03717	0.01982	0.01379	0.01364	0.00260	0.00549
	8	0.17163	0.10822	0.06431	0.03951	0.03262	0.01741	0.01214	0.01204	0.00252	0.00505
	9	0.03244	0.02204	0.01406	0.00863	0.00723	0.00416	0.00322	0.00304	0.00109	0.00160
	10	0.18923	0.12779	0.07523	0.04751	0.03838	0.02087	0.01460	0.01420	0.00274	0.00600
Korelasyon	1	0.01544	0.00118	0.00088	0.00070	0.00059	0.00056	0.00044	0.00045	0.00042	0.00039
	2	0.01331	0.00031	0.00020	0.00017	0.00015	0.00012	0.00011	0.00011	0.00010	0.00010
	3	0.01205	0.00120	0.00083	0.00070	0.00055	0.00046	0.00040	0.00036	0.00032	0.00030
	4	0.00300	0.00110	0.00073	0.00060	0.00051	0.00044	0.00042	0.00039	0.00037	0.00035
	5	0.02256	0.00910	0.00881	0.00873	0.00855	0.00856	0.00844	0.00838	0.00855	0.00844
	6	0.01515	0.00083	0.00078	0.00069	0.00069	0.00067	0.00063	0.00061	0.00060	0.00060
	7	0.01718	0.00224	0.00143	0.00121	0.00100	0.00081	0.00070	0.00065	0.00056	0.00054
	8	0.01464	0.00362	0.00353	0.00359	0.00351	0.00351	0.00352	0.00350	0.00348	0.00344
	9	0.00621	0.00282	0.00194	0.00154	0.00142	0.00111	0.00096	0.00097	0.00087	0.00081
	10	0.01527	0.00235	0.00148	0.00115	0.00088	0.00074	0.00060	0.00056	0.00046	0.00043

Şeklil 11 ve Şekil 12 den, analizde korelasyon matrisi kullanıldığında kovaryans kullanımına göre daha düşük biaslara ulaşıldığı gözlenmektedir.



Şekil 13. Çizelge 4 ün grafik olarak gösterimi(Analizde Kovaryans Matrisi Kullanılmıştır)

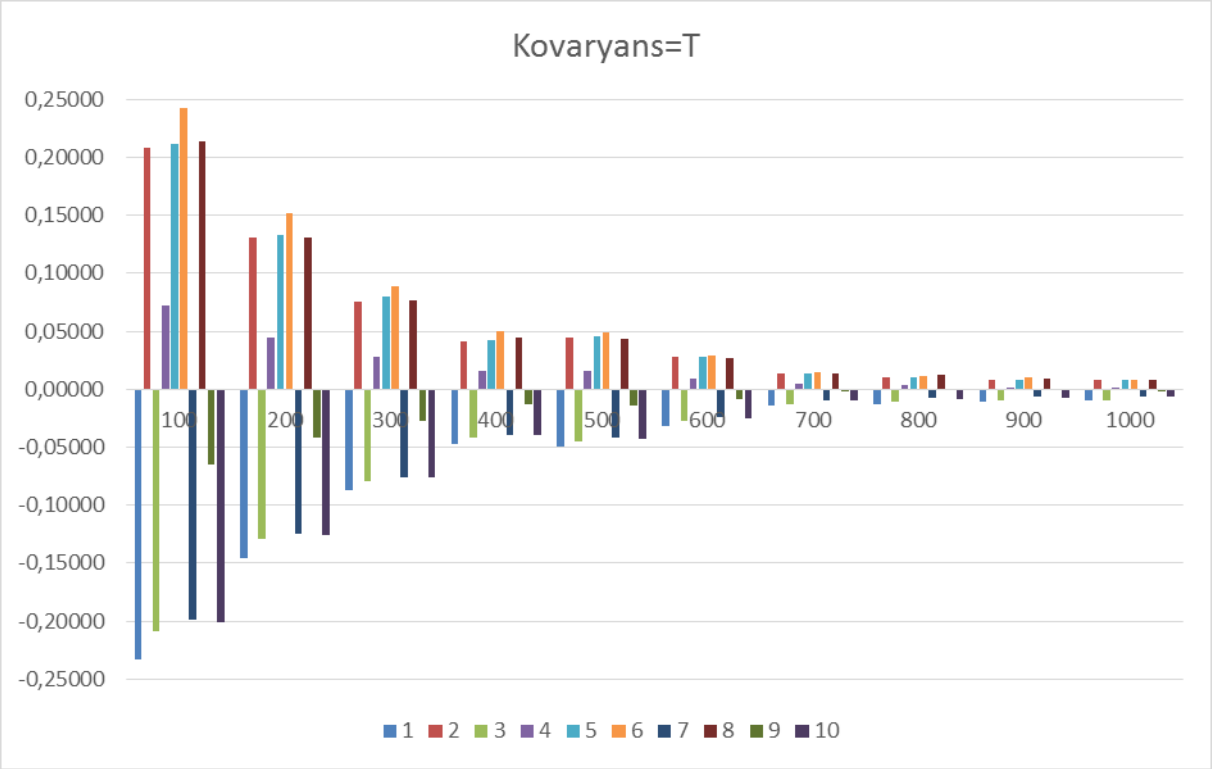


Şekil 14. Çizelge 4 ün grafik olarak gösterimi(Analizde Korelasyon Matrisi Kullanılmıştır)

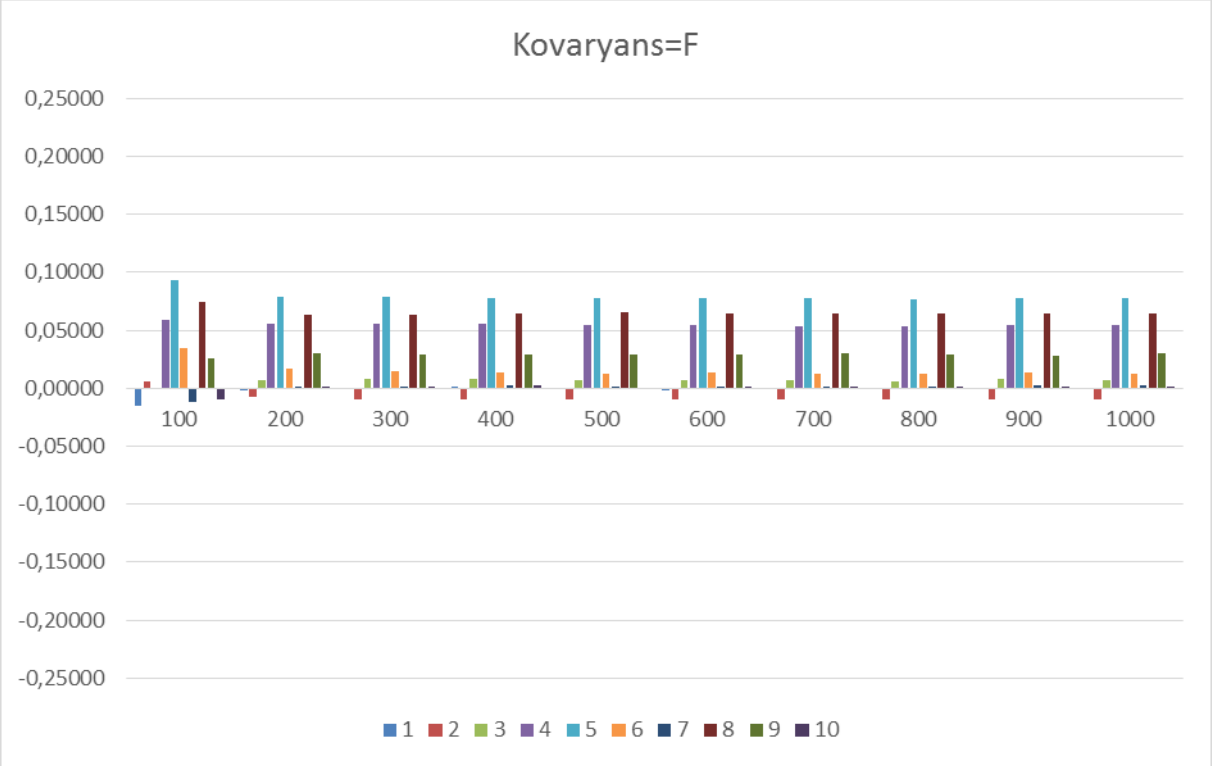
Şekil 13 ve Şekil 14 den, analizde korelasyon matrisi kullanıldığında kovaryans kullanımına göre daha düşük MSE lere ulaşıldığı gözlenmektedir.

Çizelge 5. Yük Matrisinin Tüm Parametrelerinin ML Tahmin Edicilerinin Varimax Rotasyonu Sonrası Biasları

Kullanılan Matris	par	100	200	300	400	500	600	700	800	900	1000
Kovaryans	1	-0.23364	-0.14553	-0.08739	-0.04767	-0.04971	-0.03151	-0.01415	-0.01259	-0.01097	-0.00969
	2	0.20839	0.13116	0.07566	0.04147	0.04427	0.02749	0.01328	0.01072	0.00819	0.00779
	3	-0.20932	-0.12879	-0.07907	-0.04224	-0.04531	-0.02713	-0.01276	-0.01058	-0.01002	-0.00963
	4	0.07244	0.04493	0.02765	0.01576	0.01540	0.00916	0.00484	0.00400	0.00163	0.00189
	5	0.21118	0.13346	0.07943	0.04285	0.04519	0.02802	0.01331	0.01069	0.00808	0.00843
	6	0.24233	0.15171	0.08844	0.05011	0.04891	0.02938	0.01516	0.01185	0.00989	0.00811
	7	-0.19884	-0.12512	-0.07562	-0.03937	-0.04199	-0.02400	-0.00990	-0.00751	-0.00661	-0.00587
	8	0.21400	0.13061	0.07649	0.04456	0.04352	0.02691	0.01386	0.01192	0.00870	0.00791
	9	-0.06522	-0.04223	-0.02756	-0.01320	-0.01428	-0.00841	-0.00243	-0.00109	-0.00099	-0.00149
	10	-0.20060	-0.12594	-0.07579	-0.03957	-0.04253	-0.02520	-0.00961	-0.00811	-0.00693	-0.00684
Korelasyon	1	-0.01505	-0.00161	-0.00116	0.00090	-0.00109	-0.00181	-0.00091	-0.00135	-0.00095	-0.00072
	2	0.00529	-0.00775	-0.01004	-0.00980	-0.00982	-0.00930	-0.00980	-0.00974	-0.00984	-0.00946
	3	-0.00119	0.00719	0.00773	0.00774	0.00644	0.00710	0.00680	0.00591	0.00758	0.00705
	4	0.05859	0.05513	0.05543	0.05533	0.05426	0.05411	0.05368	0.05375	0.05454	0.05423
	5	0.09313	0.07917	0.07883	0.07775	0.07759	0.07730	0.07744	0.07659	0.07720	0.07750
	6	0.03448	0.01634	0.01509	0.01380	0.01277	0.01363	0.01277	0.01291	0.01324	0.01275
	7	-0.01195	0.00087	0.00123	0.00277	0.00168	0.00122	0.00165	0.00088	0.00220	0.00242
	8	0.07388	0.06379	0.06313	0.06390	0.06501	0.06433	0.06487	0.06489	0.06432	0.06463
	9	0.02601	0.02972	0.02921	0.02914	0.02919	0.02911	0.03065	0.02929	0.02816	0.03023
	10	-0.00945	0.00053	0.00145	0.00227	0.00000	0.00069	0.00108	0.00032	0.00145	0.00107



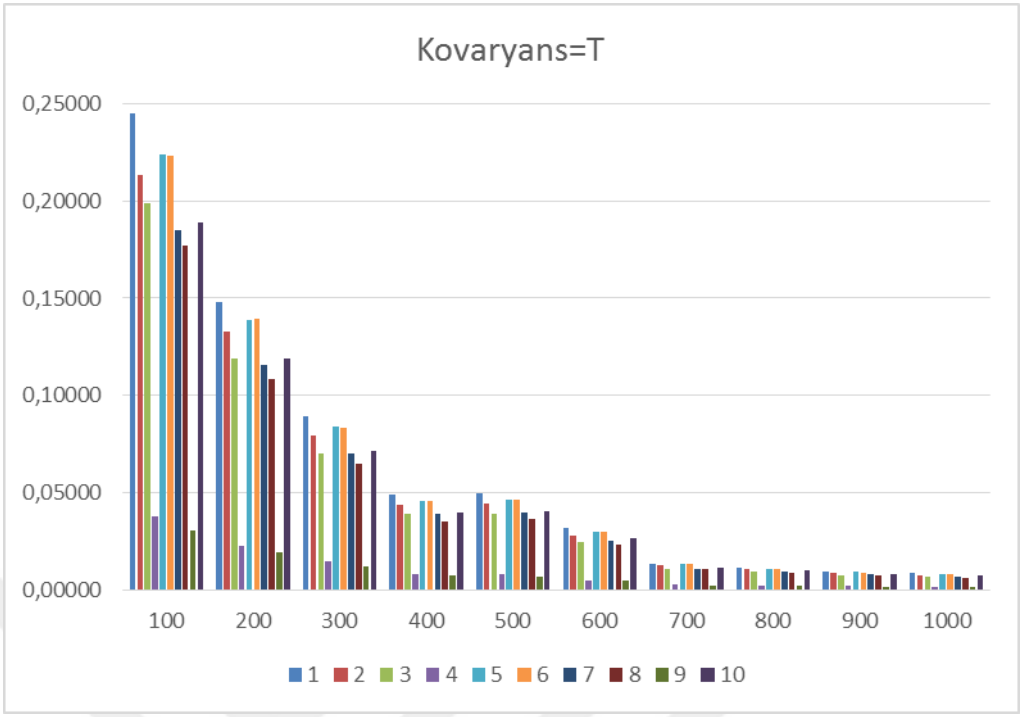
Şekil 15. Çizelge 5 in grafik olarak gösterimi(Analizde Kovaryans Matrisi Kullanılmıştır)



Şekil 16. Çizelge 5 in grafik olarak gösterimi(Analizde Korelasyon Matrisi Kullanılmıştır)

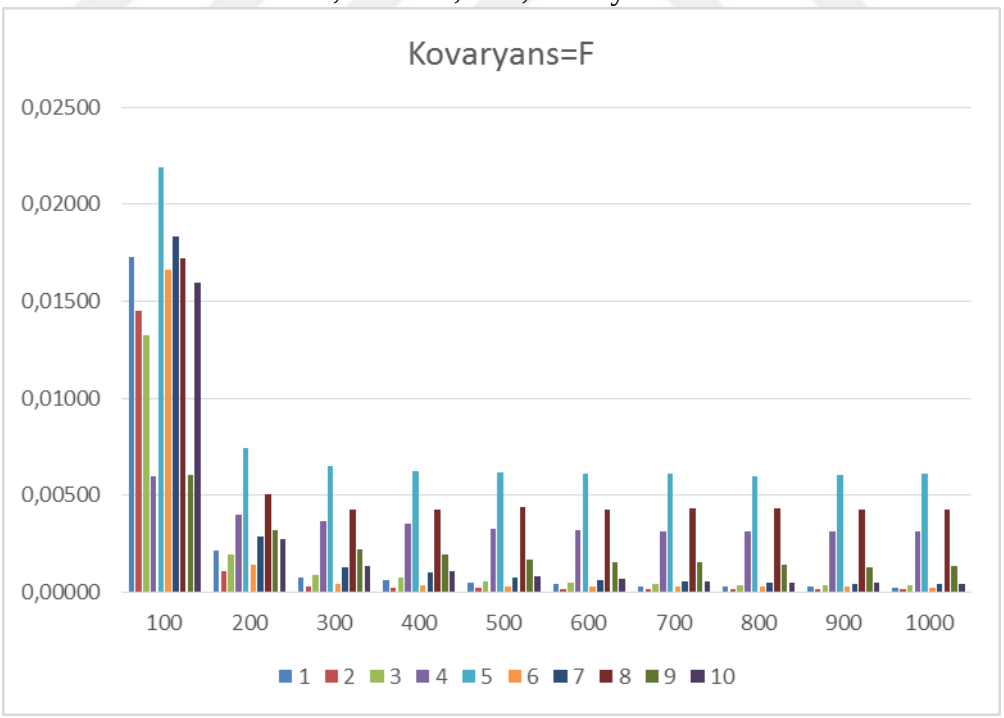
Çizelge 6. Yük Matrisinin Tüm Parametrelerinin ML Tahmin Edicilerinin Varimax Rotasyonu Sonrası MSE değerleri

Kullanılan Matris	par	100	200	300	400	500	600	700	800	900	1000
Kovaryans	1	0.24474	0.14780	0.08946	0.04879	0.04954	0.03183	0.01385	0.01179	0.00969	0.00865
	2	0.21348	0.13268	0.07914	0.04376	0.04438	0.02833	0.01298	0.01066	0.00882	0.00767
	3	0.19874	0.11869	0.07028	0.03915	0.03894	0.02489	0.01101	0.00959	0.00771	0.00683
	4	0.03780	0.02247	0.01462	0.00846	0.00832	0.00522	0.00294	0.00239	0.00212	0.00183
	5	0.22375	0.13846	0.08432	0.04598	0.04655	0.02988	0.01367	0.01120	0.00930	0.00823
	6	0.22342	0.13934	0.08316	0.04562	0.04627	0.02972	0.01351	0.01110	0.00922	0.00806
	7	0.18506	0.11583	0.07041	0.03892	0.03995	0.02532	0.01121	0.00966	0.00805	0.00711
	8	0.17670	0.10861	0.06520	0.03541	0.03632	0.02356	0.01063	0.00874	0.00736	0.00647
	9	0.03072	0.01963	0.01222	0.00748	0.00701	0.00467	0.00257	0.00235	0.00190	0.00171
	10	0.18909	0.11898	0.07127	0.04000	0.04024	0.02674	0.01177	0.01014	0.00826	0.00741
Korelasyon	1	0.01726	0.00216	0.00077	0.00058	0.00045	0.00041	0.00031	0.00029	0.00026	0.00024
	2	0.01451	0.00110	0.00027	0.00021	0.00019	0.00017	0.00016	0.00015	0.00014	0.00014
	3	0.01322	0.00195	0.00091	0.00071	0.00054	0.00047	0.00039	0.00037	0.00036	0.00032
	4	0.00598	0.00402	0.00364	0.00352	0.00328	0.00321	0.00310	0.00310	0.00315	0.00310
	5	0.02192	0.00742	0.00648	0.00625	0.00617	0.00610	0.00611	0.00597	0.00604	0.00609
	6	0.01659	0.00144	0.00044	0.00035	0.00030	0.00031	0.00027	0.00026	0.00026	0.00024
	7	0.01832	0.00286	0.00130	0.00098	0.00074	0.00064	0.00052	0.00049	0.00044	0.00041
	8	0.01724	0.00503	0.00426	0.00427	0.00439	0.00427	0.00434	0.00432	0.00424	0.00427
	9	0.00602	0.00320	0.00223	0.00193	0.00168	0.00154	0.00153	0.00138	0.00127	0.00134
	10	0.01593	0.00276	0.00134	0.00106	0.00079	0.00068	0.00052	0.00050	0.00047	0.00044



Şekil 17. Çizelge 6 nın grafik olarak gösterimi(Analizde Kovaryans Matrisi Kullanılmıştır)

ml, varimax, mse, kovaryans=F

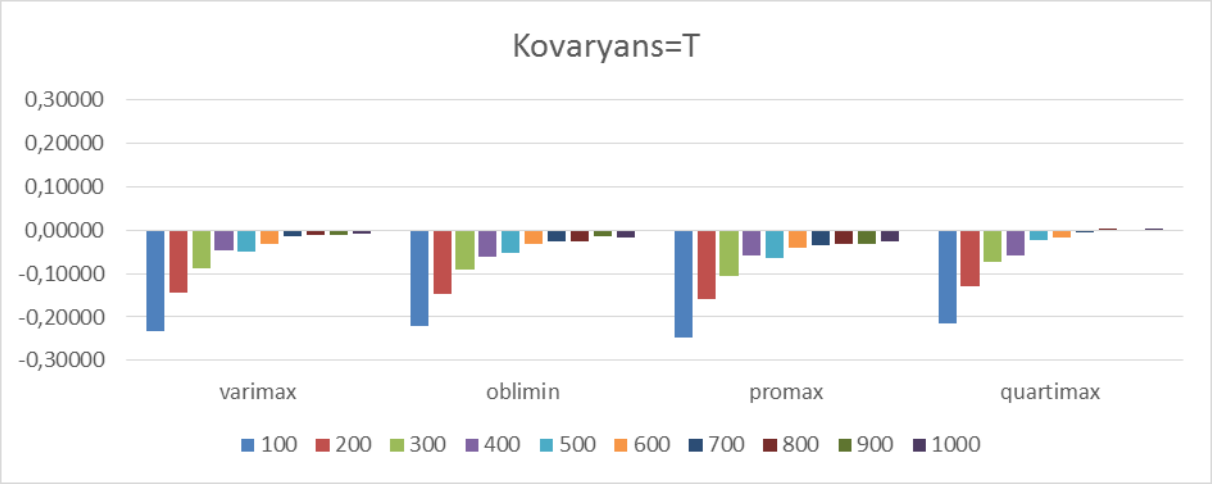


Şekil 18. Çizelge 6 nın grafik olarak gösterimi(Analizde Korelasyon Matrisi Kullanılmıştır)

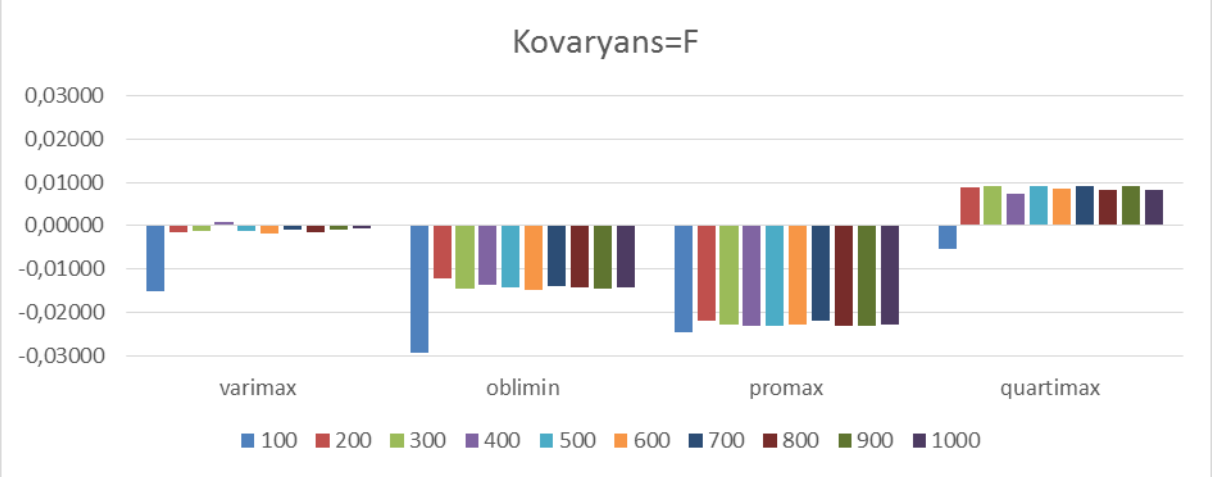
Şeklil 15-18 den, analizde korelasyon matrisi kullanıldığında kovaryans kullanımına göre daha düşük bias ve MSE lere ulaşıldığı gözlenmektedir.

Çizelge 7. Birinci parametre için ML tahmin edicisinin tüm rotasyonlardaki bias değerleri

		100	200	300	400	500	600	700	800	900	1000
Kovaryans	varimax	-0.23364	-0.14553	-0.08739	-0.04767	-0.04971	-0.03151	-0.01415	-0.01259	-0.01097	-0.00969
	oblimin	-0.22084	-0.14737	-0.09080	-0.06147	-0.05291	-0.03302	-0.02719	-0.02713	-0.01456	-0.01732
	promax	-0.24655	-0.15756	-0.10618	-0.05871	-0.06368	-0.04016	-0.03452	-0.03170	-0.03264	-0.02744
	quartimax	-0.21536	-0.12990	-0.07457	-0.05770	-0.02384	-0.01843	-0.00642	0.00169	-0.00370	0.00125
Korelasyon	varimax	-0.01505	-0.00161	-0.00116	0.00090	-0.00109	-0.00181	-0.00091	-0.00135	-0.00095	-0.00072
	oblimin	-0.02921	-0.01205	-0.01447	-0.01366	-0.01410	-0.01464	-0.01381	-0.01423	-0.01458	-0.01406
	promax	-0.02454	-0.02179	-0.02285	-0.02312	-0.02291	-0.02283	-0.02194	-0.02317	-0.02298	-0.02273
	quartimax	-0.00520	0.00884	0.00904	0.00749	0.00901	0.00868	0.00917	0.00827	0.00920	0.00838



Şekil 19. Çizelge 7 nin grafik olarak gösterimi(Analizde Kovaryans Matrisi Kullanılmıştır)

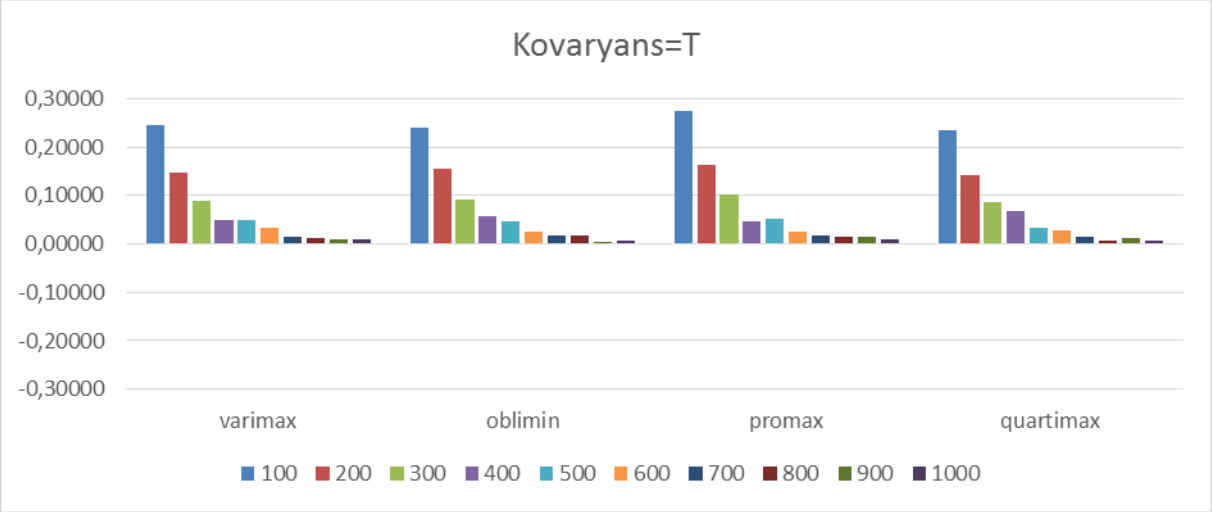


Şekil 20. Çizelge 7 nin grafik olarak gösterimi(Analizde Korelasyon Matrisi Kullanılmıştır)

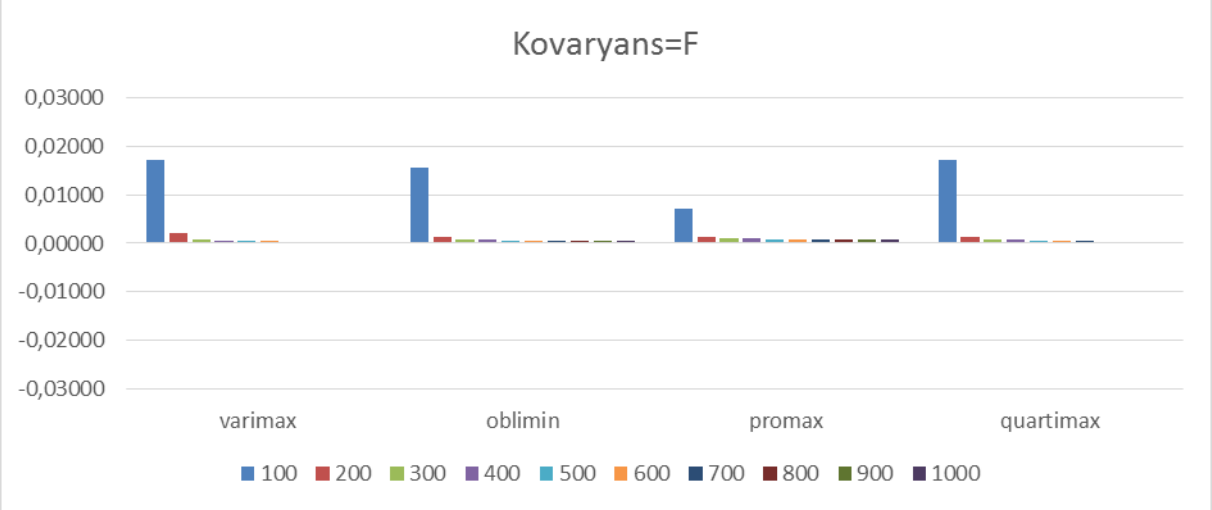
Şekil 19-20 den, analizde kovaryans matrisi kullanıldığında tüm rotasyonlarda mias lar hemen hemen benzerken, kovaryans matrisi kullanıldığında en iyi biasa sahip olan rotasyon varimax olmaktadır.

Çizelge 8. Birinci parametre için ML tahmin edicisinin tüm rotasyonlardaki MSE değerleri

		100	200	300	400	500	600	700	800	900	1000
T	varimax	0.24474	0.14780	0.08946	0.04879	0.04954	0.03183	0.01385	0.01179	0.00969	0.00865
	oblimin	0.23956	0.15405	0.09011	0.05587	0.04502	0.02401	0.01659	0.01648	0.00282	0.00658
	promax	0.27322	0.16406	0.10121	0.04753	0.05258	0.02463	0.01660	0.01430	0.01535	0.00840
	quartimax	0.23425	0.14127	0.08505	0.06771	0.03286	0.02712	0.01560	0.00738	0.01262	0.00741
F	varimax	0.01726	0.00216	0.00077	0.00058	0.00045	0.00041	0.00031	0.00029	0.00026	0.00024
	oblimin	0.01544	0.00118	0.00088	0.00070	0.00059	0.00056	0.00044	0.00045	0.00042	0.00039
	promax	0.00709	0.00134	0.00116	0.00102	0.00088	0.00081	0.00075	0.00076	0.00074	0.00070
	quartimax	0.01706	0.00128	0.00082	0.00069	0.00055	0.00046	0.00045	0.00037	0.00035	0.00030



Şekil 21. Çizelge 8 in grafik olarak gösterimi(Analizde Kovaryans Matrisi Kullanılmıştır)



Şekil 22. Çizelge 8 in grafik olarak gösterimi(Analizde Korelasyon Matrisi Kullanılmıştır)

Şekil 21-22 den, analizde kovaryans matrisi kullanıldığında tüm rotasyonlarda mias lar hemen hemen benzerken, kovaryans matrisi kullanıldığında en iyi MSE değerlerine sahip olan rotasyon varimax olmaktadır.

4. SONUÇLAR VE ÖNERİLER

Yapılan simülasyon çalışmalarında, korelasyon matrisi ile çalışıldığında faktör yüklerininin 8 tahmin edicisininde yan ve MSE değerleri, kovaryans matrisi ile çalışıldığında faktör yüklerininin 8 tahmin edicisininde yan ve MSE değerlerinden çok daha düşük olduğu gözlenmiştir. Bu durumda faktör analizinde, korelasyon matrisi ile çalışmanın daha uygun olduğu söylenebilir. Varimax rotasyonunda en iyi bias ve mse ye sahip FM fa4 iken, oblimin rotasyonun da en iyi bias ve mse ye sahip FM fa5 metodu olmaktadır. Bu durumda farklı rotasyonlarda farklı FM lerin kullanılması gerektiği anlaşılmaktadır. Bonferroni güven aralıkları için nboot=10 alındığında kapsama olasılıklarında nominal değere ulaşılamamıştır. R kodunun çok yavaş çalışmasından dolayı nboot artırılamamıştır. Nboot arttığında nominal değere ulaşılabileceği umulmaktadır. Güven aralıklarının performansı başka bir konu olarak önerilebilir.

KAYNAKLAR

Barendse, M. T., Oort, F. J. ve Timmerman, M. E., 2015, Using exploratory factor analysis to determine the dimensionality of discrete responses., *Structural Equation Modeling: A Multidisciplinary Journal,*, 22 (1), 87-101.

Beauducel, A., 2001, On the generalizability of factors: The influence of changing contexts of variables on different methods of factor extraction, *Methods of Psychological Research Online*, 6 (1), 69-96.

Bartholomew, D. J. (1984). Scaling binary data using a factor model. Journal of the Royal Statistical Society, Series B, 46, 120–123.

Bartholomew, D. J. 1984. The Foundations of Factor Analysis, Biometrika, 71(2), 221-232.

Bartholomew, D. J. 1980. Factor Analysis for Categorical Data, *Journal of the Royal Statistical Society. Series B (Methodological)*, 42(3), 293-321.

Bollen, K. A. (1989). Structural equations with latent variables. New York, NY: Wiley.

Browne, M. W. (1987). Robustness of statistical inference in factor analysis and related models. Biometrika, 74, 375–384.

Browne, M. W. and Shapiro, A. 1987. Adjustments for Kurtosis in Factor Analysis with Elliptically Distributed Errors, *Journal of the Royal Statistical Society. Series B (Methodological)*, 49(3), 346-352.

Christensen, A. P., Gross, G. M., Golino, H. F., Silvia, P. J. ve Kwapil, T. R. J. S. r., 2019, Exploratory graph analysis of the multidimensional schizotypy scale, 206, 43-51.

Coughlin, K. B., Kromrey, J. ve Hibbard, S., (2013), Using predetermined factor structures to simulate a variety of data conditions, *Florida*.

Çokluk, Ö. ve Koçak, D., 2016, Using Horn's Parallel Analysis Method in Exploratory Factor Analysis for Determining the Number of Factors, *Educational Sciences: Theory and Practice*, 16 (2), 537-551.

Darton, R. A. 1980. Rotation in Factor Analysis, Journal of the Royal Statistical Society. Series D (The Statistician), 29(3)167-194.

Finch, W. H., 2011, A comparison of factor rotation methods for dichotomous data, Journal of Modern Applied Statistical Methods, 10 (2), 14.

Hauck F., N., Machado, W. D. L. ve Damásio, B. F., 2014, Effects of statistical models and items difficulties on making trait-level inferences: A simulation study, *Psicologia: Reflexão e Crítica*, 27 (4), 670-678.

Jung, S. ve Lee, S., 2011, Exploratory factor analysis for small samples., *Behavior research methods,*, 43 (3), 701-709.

Jung, S., 2013, Exploratory factor analysis with small sample sizes: A comparison of three approaches, *Behavioural processes*, 97, 90-95.

Karaman, H., Burcu, A. ve AKTAN, D. Ç., 2017, Açımlayıcı faktör analizinde kullanılan faktör çıkartma yöntemlerinin karşılaştırılması, *Gazi Üniversitesi Gazi Eğitim Fakültesi Dergisi*, 37 (3), 1173-1193.

Kılıç, A. F. ve Uysal, İ. J. T. J. o. E., 2019, Comparison of factor retention methods on binary data: A simulation study, 8 (3), 160-179.

Kano, Y. 1991. The Asymptotic Distribution of a Noniterative Estimator in Exploratory Factor Analysis. The Annals of Statistics, 19(1), 272-282.

Kano, Y. and Shapiro A., 1987. On Asymptotic Varıances Of Unıqueness Estimators in Factor Analysis. South African Statist. J. 21, 131-139.

Koğar, H., 2018, Effects of Various Simulation Conditions on Latent-Trait Estimates: A Simulation Study, *International Journal of Assessment Tools in Education*, 5 (2), 263-273.

Laake, P., 1988. Exact and Asymptotic Distributions of the Maximum Likelihood Estimate in a SimpleFactor Analysis Model. Scandinavian Journal of Statistics, 15(3), 195-201.

Jöreskog, K. G. (1969). A general approach to confirmatory maximum likelihood factor analysis. Psychometrika, 34, 183–202.

Wallace C. S. and Freeman, P. R. 'Single Factor Analysis by MML Estimation', J.R. Statist. Soc. B, 54, No.1, 1992, 195-209

Winter, J. D., Dodou, D. ve Wieringa, P. A., 2009, Exploratory factor analysis with small sample sizes., *Multivariate behavioral research,*, 44 (2), 147-181.

Wu, J. ve Bentler, P. M., 2012, Application of H-likelihood to factor analysis models with binary response data, *Journal of Multivariate Analysis*, 106, 72-79.

EKLER

EK-1 3. Bölümdeki simüslasyon sonuçları için kullanılan R kodları

```
set.seed(NULL)
library(boot)
library(mvtnorm)
library(corrplot)
library(nFactors)
library(psych)
library(GPArotation)
rm(list = ls())
klasor="d:/osman/"
setwd(klasor)
nler=seq(500,2000,250)
ds=50
fried=NULL
alpha=0.05
R=3
mis=25
isim=paste0("sonuclar_r",R)
sonuc=NULL
mse=function(x) mean((x-mean(x))^2)
l1=matrix(c(.56,.78,.65,.94,.80,.82,-.53,.75,-.10,-.54),nrow =  5)

fs=NCOL(l1)#faktÃ¶r sayÄ±sÄ±
dgs=NROW(l1)

for(n in nler)
{
  simler=NULL
for(ii in 1:ds){
bas=NULL
cat('\14',n,ii)

#-------------------------------------------------------------------------
#covar=T,varimax
#-------------------------------------------------------------------------
l=varimax(l1)$loadings[1:dgs,1:fs]
cat('\14',n,ii)
dev=F
while(dev==F)
{ x=NULL
  for(j in 1:n){
    f=matrix(rnorm(fs),nrow = fs)

eps=matrix(c(rnorm(1,0,.2),rnorm(1,0,.3),rnorm(1,0,.4),rnorm(1,0,.5)
,rnorm(1,0,.5)),ncol= 1)
    x=rbind(x,t(l%*%f+eps))
  }

  yn1=NULL; yn2=NULL; yn3=NULL; yn4=NULL; yn5=NULL;
yn6=NULL; yn7=NULL; yn8=NULL
  ff1=try(fa1<-fa(x,nfactors = fs,fm='ml',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
  ff2=try(fa2<-fa(x,nfactors = fs,fm='minres',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
  ff3=try(fa3<-fa(x,nfactors = fs,fm='uls',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
  ff4=try(fa4<-fa(x,nfactors = fs,fm='wls',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
  ff5=try(fa5<-fa(x,nfactors = fs,fm='gls',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
  ff6=try(fa6<-fa(x,nfactors = fs,fm='pa',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
  ff7=try(fa7<-fa(x,nfactors = fs,fm='minchi',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
  ff8=try(fa8<-fa(x,nfactors = fs,fm='old.min',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)

  if
(!is.character(ff1)&&!is.character(ff2)&&!is.character(ff3)&&!is.ch
aracter(ff4)&&!is.character(ff5)&&!is.character(ff6)&&!is.character
(ff7)&&!is.character(ff8) )
  {
    dev=T
  }
}

tmm=fa1$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm1_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm1_tah=c(fm1_tah,aa)
}

fm1_ci=fa1$cis$ci
fm1_bias=(l-(fm1_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm1_mse =((l-(fm1_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm1_bg=NULL
fm1_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm1_ci[,kk[i]]*iler[i])
ust=c(ust,fm1_ci[,fs+kk[i]]*iler[i])
fm1_bg=c(fm1_bg,fm1_ci[,fs+kk[i]]*iler[i]-  fm1_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm1_tah[i]<alt[i] || fm1_tah[i]>ust[i]) fm1_ko[i]=0
}

tmm=fa2$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm2_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm2_tah=c(fm2_tah,aa)
}

fm2_ci=fa2$cis$ci
fm2_bias=(l-(fm2_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm2_mse =((l-(fm2_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm2_bg=NULL
fm2_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm2_ci[,kk[i]]*iler[i])
ust=c(ust,fm2_ci[,fs+kk[i]]*iler[i])
fm2_bg=c(fm2_bg,fm2_ci[,fs+kk[i]]*iler[i]-  fm2_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm2_tah[i]<alt[i] || fm2_tah[i]>ust[i]) fm2_ko[i]=0
}



tmm=fa3$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
```

```
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm3_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm3_tah=c(fm3_tah,aa)
}

fm3_ci=fa3$cis$ci
fm3_bias=(l-(fm3_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm3_mse =((l-(fm3_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm3_bg=NULL
fm3_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm3_ci[,kk[i]]*iler[i])
ust=c(ust,fm3_ci[,fs+kk[i]]*iler[i])
fm3_bg=c(fm3_bg,fm3_ci[,fs+kk[i]]*iler[i]-  fm3_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm3_tah[i]<alt[i] || fm3_tah[i]>ust[i]) fm3_ko[i]=0
}

tmm=fa4$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm4_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm4_tah=c(fm4_tah,aa)
}

fm4_ci=fa1$cis$ci
fm4_bias=(l-(fm4_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm4_mse =((l-(fm4_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm4_bg=NULL
fm4_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm4_ci[,kk[i]]*iler[i])
ust=c(ust,fm4_ci[,fs+kk[i]]*iler[i])
fm4_bg=c(fm4_bg,fm4_ci[,fs+kk[i]]*iler[i]-  fm4_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm4_tah[i]<alt[i] || fm4_tah[i]>ust[i]) fm4_ko[i]=0
}

tmm=fa5$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm5_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm5_tah=c(fm5_tah,aa)
}

fm5_ci=fa5$cis$ci
fm5_bias=(l-(fm5_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm5_mse =((l-(fm5_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm5_bg=NULL
fm5_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm5_ci[,kk[i]]*iler[i])
ust=c(ust,fm5_ci[,fs+kk[i]]*iler[i])
fm5_bg=c(fm5_bg,fm5_ci[,fs+kk[i]]*iler[i]-  fm5_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm5_tah[i]<alt[i] || fm5_tah[i]>ust[i]) fm5_ko[i]=0
}

tmm=fa6$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm6_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm6_tah=c(fm6_tah,aa)
}

fm6_ci=fa6$cis$ci
fm6_bias=(l-(fm6_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm6_mse =((l-(fm6_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm6_bg=NULL
fm6_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm6_ci[,kk[i]]*iler[i])
ust=c(ust,fm6_ci[,fs+kk[i]]*iler[i])
fm6_bg=c(fm6_bg,fm6_ci[,fs+kk[i]]*iler[i]-  fm6_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm6_tah[i]<alt[i] || fm6_tah[i]>ust[i]) fm6_ko[i]=0
}


tmm=fa7$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm7_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
```

```
fm7_tah=c(fm7_tah,aa)
}

fm7_ci=fa7$cis$ci
fm7_bias=(l-(fm7_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm7_mse =((l-(fm7_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm7_bg=NULL
fm7_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm7_ci[,kk[i]]*iler[i])
ust=c(ust,fm7_ci[,fs+kk[i]]*iler[i])
fm7_bg=c(fm7_bg,fm7_ci[,fs+kk[i]]*iler[i]-  fm7_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
  if (fm7_tah[i]<alt[i] || fm7_tah[i]>ust[i]) fm7_ko[i]=0
}



tmm=fa8$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
  ek=dgs^2
  for(i2 in ller) #l
  { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
  b=(sum((-tm[,i1]-l[,i2])^2))^0.5
  if (a<ek)
  {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
  if (b<ek)
  {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
  }
  ller=ller[-kkk]
}
fm8_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm8_tah=c(fm8_tah,aa)
}


fm8_ci=fa8$cis$ci
fm8_bias=(l-(fm8_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm8_mse =((l-(fm8_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm8_bg=NULL
fm8_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm8_ci[,kk[i]]*iler[i])
ust=c(ust,fm8_ci[,fs+kk[i]]*iler[i])
fm8_bg=c(fm8_bg,fm8_ci[,fs+kk[i]]*iler[i]-  fm8_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
  if (fm8_tah[i]<alt[i] || fm8_tah[i]>ust[i]) fm8_ko[i]=0
}


  yn1=unlist(c(fm1_bias,fm1_mse,fm1_bg,fm1_ko))
  yn2=unlist(c(fm2_bias,fm2_mse,fm2_bg,fm2_ko))
  yn3=unlist(c(fm3_bias,fm3_mse,fm3_bg,fm3_ko))
  yn4=unlist(c(fm4_bias,fm4_mse,fm4_bg,fm4_ko))
  yn5=unlist(c(fm5_bias,fm5_mse,fm5_bg,fm5_ko))
  yn6=unlist(c(fm6_bias,fm6_mse,fm6_bg,fm6_ko))
  yn7=unlist(c(fm7_bias,fm7_mse,fm7_bg,fm7_ko))
  yn8=unlist(c(fm8_bias,fm8_mse,fm8_bg,fm8_ko))

pler=c(fa1$PVAL,fa2$PVAL,fa3$PVAL,fa4$PVAL,fa5$PVAL,fa6
$PVAL,fa7$PVAL,fa8$PVAL)
  aa=fa.parallel(x,plot = F)$nfact
  if (aa==fs) dr=1 else dr=0
  bas=c(bas,unlist(c(yn1,yn2,yn3,yn4,yn5,yn6,yn7,yn8,pler,dr)))


  #-------------------------------------------------------------------------
  #covar=F,varimax
  #-------------------------------------------------------------------------

  l=varimax(l1)$loadings[1:dgs,1:fs]
  cat('\14',n,ii)
  dev=F
  while(dev==F)
{ x=NULL
  for(j in 1:n){
    f=matrix(rnorm(fs),nrow = fs)

eps=matrix(c(rnorm(1,0,.2),rnorm(1,0,.3),rnorm(1,0,.4),rnorm(1,0,.5)
,rnorm(1,0,.5)),ncol= 1)
    x=rbind(x,t(l%*%f+eps))
  }

  yn1=NULL; yn2=NULL; yn3=NULL; yn4=NULL; yn5=NULL;
yn6=NULL; yn7=NULL; yn8=NULL
 ff1=try(fa1<-fa(x,nfactors = fs,fm='ml',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff2=try(fa2<-fa(x,nfactors = fs,fm='minres',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff3=try(fa3<-fa(x,nfactors = fs,fm='uls',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff4=try(fa4<-fa(x,nfactors = fs,fm='wls',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff5=try(fa5<-fa(x,nfactors = fs,fm='gls',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff6=try(fa6<-fa(x,nfactors = fs,fm='pa',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff7=try(fa7<-fa(x,nfactors = fs,fm='minchi',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff8=try(fa8<-fa(x,nfactors = fs,fm='old.min',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)

 if
(!is.character(ff1)&&!is.character(ff2)&&!is.character(ff3)&&!is.ch
aracter(ff4)&&!is.character(ff5)&&!is.character(ff6)&&!is.character
(ff7)&&!is.character(ff8) )
 {
  dev=T
 }
 }


 tmm=fa1$loadings[1:(fs*dgs)]
 tm=matrix(tmm,nrow = dgs)
 kk=NULL; ller=1:fs;iler=NULL
 for(i1 in 1:fs) #tm
 {
  ek=dgs^2
  for(i2 in ller) #l
  { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
  b=(sum((-tm[,i1]-l[,i2])^2))^0.5
  if (a<ek)
  {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
  if (b<ek)
  {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
  }
  ller=ller[-kkk]
 }
 fm1_tah=NULL
 for(i1 in 1:fs)
 { aa=tm[,kk[i1]]*iler[i1]
 fm1_tah=c(fm1_tah,aa)
 }


 fm1_ci=fa1$cis$ci
 fm1_bias=(l-(fm1_tah[1:(fs*dgs)]))[1:(fs*dgs)]
 fm1_mse =((l-(fm1_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
 fm1_bg=NULL
 fm1_ko=rep(1,fs*dgs)
 alt=NULL
 ust=NULL
 for(i in 1:fs)
 { alt=c(alt,fm1_ci[,kk[i]]*iler[i])
 ust=c(ust,fm1_ci[,fs+kk[i]]*iler[i])
 fm1_bg=c(fm1_bg,fm1_ci[,fs+kk[i]]*iler[i]-  fm1_ci[,kk[i]]*iler[i])
 }
 for(i in 1:(fs*dgs))
 {
  if (fm1_tah[i]<alt[i] || fm1_tah[i]>ust[i]) fm1_ko[i]=0
 }



 tmm=fa2$loadings[1:(fs*dgs)]
 tm=matrix(tmm,nrow = dgs)
 kk=NULL; ller=1:fs;iler=NULL
```

```
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm2_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm2_tah=c(fm2_tah,aa)
}

fm2_ci=fa2$cis$ci
fm2_bias=(l-(fm2_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm2_mse =((l-(fm2_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm2_bg=NULL
fm2_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm2_ci[,kk[i]]*iler[i])
ust=c(ust,fm2_ci[,fs+kk[i]]*iler[i])
fm2_bg=c(fm2_bg,fm2_ci[,fs+kk[i]]*iler[i]-  fm2_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm2_tah[i]<alt[i] || fm2_tah[i]>ust[i]) fm2_ko[i]=0
}

tmm=fa3$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm3_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm3_tah=c(fm3_tah,aa)
}

fm3_ci=fa3$cis$ci
fm3_bias=(l-(fm3_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm3_mse =((l-(fm3_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm3_bg=NULL
fm3_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm3_ci[,kk[i]]*iler[i])
ust=c(ust,fm3_ci[,fs+kk[i]]*iler[i])
fm3_bg=c(fm3_bg,fm3_ci[,fs+kk[i]]*iler[i]-  fm3_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm3_tah[i]<alt[i] || fm3_tah[i]>ust[i]) fm3_ko[i]=0
}

 tmm=fa4$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm4_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm4_tah=c(fm4_tah,aa)
}

fm4_ci=fa1$cis$ci
fm4_bias=(l-(fm4_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm4_mse =((l-(fm4_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm4_bg=NULL
fm4_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm4_ci[,kk[i]]*iler[i])
ust=c(ust,fm4_ci[,fs+kk[i]]*iler[i])
fm4_bg=c(fm4_bg,fm4_ci[,fs+kk[i]]*iler[i]-  fm4_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm4_tah[i]<alt[i] || fm4_tah[i]>ust[i]) fm4_ko[i]=0
}

 tmm=fa5$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm5_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm5_tah=c(fm5_tah,aa)
}

fm5_ci=fa5$cis$ci
fm5_bias=(l-(fm5_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm5_mse =((l-(fm5_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm5_bg=NULL
fm5_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm5_ci[,kk[i]]*iler[i])
ust=c(ust,fm5_ci[,fs+kk[i]]*iler[i])
fm5_bg=c(fm5_bg,fm5_ci[,fs+kk[i]]*iler[i]-  fm5_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm5_tah[i]<alt[i] || fm5_tah[i]>ust[i]) fm5_ko[i]=0
}

 tmm=fa6$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm6_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
```

```
fm6_tah=c(fm6_tah,aa)
}

 fm6_ci=fa6$cis$ci
fm6_bias=(l-(fm6_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm6_mse =((l-(fm6_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm6_bg=NULL
fm6_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm6_ci[,kk[i]]*iler[i])
ust=c(ust,fm6_ci[,fs+kk[i]]*iler[i])
fm6_bg=c(fm6_bg,fm6_ci[,fs+kk[i]]*iler[i]-  fm6_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm6_tah[i]<alt[i] || fm6_tah[i]>ust[i]) fm6_ko[i]=0
}

tmm=fa7$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm7_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm7_tah=c(fm7_tah,aa)
}

fm7_ci=fa7$cis$ci
fm7_bias=(l-(fm7_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm7_mse =((l-(fm7_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm7_bg=NULL
fm7_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm7_ci[,kk[i]]*iler[i])
ust=c(ust,fm7_ci[,fs+kk[i]]*iler[i])
fm7_bg=c(fm7_bg,fm7_ci[,fs+kk[i]]*iler[i]-  fm7_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm7_tah[i]<alt[i] || fm7_tah[i]>ust[i]) fm7_ko[i]=0
}

tmm=fa8$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm8_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm8_tah=c(fm8_tah,aa)
}

fm8_ci=fa8$cis$ci
fm8_bias=(l-(fm8_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm8_mse =((l-(fm8_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm8_bg=NULL
fm8_ko=rep(1,fs*dgs)
 alt=NULL
 ust=NULL
 for(i in 1:fs)
 { alt=c(alt,fm8_ci[,kk[i]]*iler[i])
 ust=c(ust,fm8_ci[,fs+kk[i]]*iler[i])
 fm8_bg=c(fm8_bg,fm8_ci[,fs+kk[i]]*iler[i]-  fm8_ci[,kk[i]]*iler[i])
 }
 for(i in 1:(fs*dgs))
 {
  if (fm8_tah[i]<alt[i] || fm8_tah[i]>ust[i]) fm8_ko[i]=0
 }

yn1=unlist(c(fm1_bias,fm1_mse,fm1_bg,fm1_ko))
 yn2=unlist(c(fm2_bias,fm2_mse,fm2_bg,fm2_ko))
 yn3=unlist(c(fm3_bias,fm3_mse,fm3_bg,fm3_ko))
 yn4=unlist(c(fm4_bias,fm4_mse,fm4_bg,fm4_ko))
 yn5=unlist(c(fm5_bias,fm5_mse,fm5_bg,fm5_ko))
 yn6=unlist(c(fm6_bias,fm6_mse,fm6_bg,fm6_ko))
 yn7=unlist(c(fm7_bias,fm7_mse,fm7_bg,fm7_ko))
 yn8=unlist(c(fm8_bias,fm8_mse,fm8_bg,fm8_ko))

pler=c(fa1$PVAL,fa2$PVAL,fa3$PVAL,fa4$PVAL,fa5$PVAL,fa6
$PVAL,fa7$PVAL,fa8$PVAL)
 aa=fa.parallel(x,plot = F)$nfact
 if (aa==fs) dr=1 else dr=0
 bas=c(bas,unlist(c(yn1,yn2,yn3,yn4,yn5,yn6,yn7,yn8,pler,dr)))

 #-------------------------------------------------------------------------------
 #covar=T,oblimin
 #-------------------------------------------------------------------------------

 l=oblimin(l1)$loadings[1:dgs,1:fs]
 cat('\14',n,ii)
 dev=F
 while(dev==F)
{ x=NULL
 for(j in 1:n){
   f=matrix(rnorm(fs),nrow = fs)

eps=matrix(c(rnorm(1,0,.2),rnorm(1,0,.3),rnorm(1,0,.4),rnorm(1,0,.5)
,rnorm(1,0,.5)),ncol= 1)
   x=rbind(x,t(l%*%f+eps))
 }

 yn1=NULL; yn2=NULL; yn3=NULL; yn4=NULL; yn5=NULL;
yn6=NULL; yn7=NULL; yn8=NULL
 ff1=try(fa1<-fa(x,nfactors = fs,fm='ml',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff2=try(fa2<-fa(x,nfactors = fs,fm='minres',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff3=try(fa3<-fa(x,nfactors = fs,fm='uls',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff4=try(fa4<-fa(x,nfactors = fs,fm='wls',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff5=try(fa5<-fa(x,nfactors = fs,fm='gls',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff6=try(fa6<-fa(x,nfactors = fs,fm='pa',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff7=try(fa7<-fa(x,nfactors = fs,fm='minchi',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff8=try(fa8<-fa(x,nfactors = fs,fm='old.min',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)

 if
(!is.character(ff1)&&!is.character(ff2)&&!is.character(ff3)&&!is.ch
aracter(ff4)&&!is.character(ff5)&&!is.character(ff6)&&!is.character
(ff7)&&!is.character(ff8) )
 {
  dev=T
 }
  }

  tmm=fa1$loadings[1:(fs*dgs)]
  tm=matrix(tmm,nrow = dgs)
  kk=NULL; ller=1:fs;iler=NULL
  for(i1 in 1:fs) #tm
  {
   ek=dgs^2
   for(i2 in ller) #l
   { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
   b=(sum((-tm[,i1]-l[,i2])^2))^0.5
   if (a<ek)
   {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
   if (b<ek)
   {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
   }
```

```
 ller=ller[-kkk]
}
fm1_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm1_tah=c(fm1_tah,aa)
}

fm1_ci=fa1$cis$ci
fm1_bias=(l-(fm1_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm1_mse =((l-(fm1_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm1_bg=NULL
fm1_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm1_ci[,kk[i]]*iler[i])
ust=c(ust,fm1_ci[,fs+kk[i]]*iler[i])
fm1_bg=c(fm1_bg,fm1_ci[,fs+kk[i]]*iler[i]-  fm1_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm1_tah[i]<alt[i] || fm1_tah[i]>ust[i]) fm1_ko[i]=0
}

 tmm=fa2$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm2_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm2_tah=c(fm2_tah,aa)
}

fm2_ci=fa2$cis$ci
fm2_bias=(l-(fm2_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm2_mse =((l-(fm2_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm2_bg=NULL
fm2_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm2_ci[,kk[i]]*iler[i])
ust=c(ust,fm2_ci[,fs+kk[i]]*iler[i])
fm2_bg=c(fm2_bg,fm2_ci[,fs+kk[i]]*iler[i]-  fm2_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm2_tah[i]<alt[i] || fm2_tah[i]>ust[i]) fm2_ko[i]=0
}

tmm=fa3$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm3_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm3_tah=c(fm3_tah,aa)
}

fm3_ci=fa3$cis$ci
fm3_bias=(l-(fm3_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm3_mse =((l-(fm3_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm3_bg=NULL
fm3_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm3_ci[,kk[i]]*iler[i])
ust=c(ust,fm3_ci[,fs+kk[i]]*iler[i])
fm3_bg=c(fm3_bg,fm3_ci[,fs+kk[i]]*iler[i]-  fm3_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm3_tah[i]<alt[i] || fm3_tah[i]>ust[i]) fm3_ko[i]=0
}

 tmm=fa4$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm4_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm4_tah=c(fm4_tah,aa)
}

 fm4_ci=fa1$cis$ci
fm4_bias=(l-(fm4_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm4_mse =((l-(fm4_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm4_bg=NULL
fm4_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm4_ci[,kk[i]]*iler[i])
ust=c(ust,fm4_ci[,fs+kk[i]]*iler[i])
fm4_bg=c(fm4_bg,fm4_ci[,fs+kk[i]]*iler[i]-  fm4_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm4_tah[i]<alt[i] || fm4_tah[i]>ust[i]) fm4_ko[i]=0
}

tmm=fa5$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm5_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm5_tah=c(fm5_tah,aa)
}

 fm5_ci=fa5$cis$ci
fm5_bias=(l-(fm5_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm5_mse =((l-(fm5_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm5_bg=NULL
fm5_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
```

```
{ alt=c(alt,fm5_ci[,kk[i]]*iler[i])
ust=c(ust,fm5_ci[,fs+kk[i]]*iler[i])
fm5_bg=c(fm5_bg,fm5_ci[,fs+kk[i]]*iler[i]-  fm5_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm5_tah[i]<alt[i] || fm5_tah[i]>ust[i]) fm5_ko[i]=0
}

tmm=fa6$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm6_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm6_tah=c(fm6_tah,aa)
}

fm6_ci=fa6$cis$ci
fm6_bias=(l-(fm6_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm6_mse =((l-(fm6_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm6_bg=NULL
fm6_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm6_ci[,kk[i]]*iler[i])
ust=c(ust,fm6_ci[,fs+kk[i]]*iler[i])
fm6_bg=c(fm6_bg,fm6_ci[,fs+kk[i]]*iler[i]-  fm6_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm6_tah[i]<alt[i] || fm6_tah[i]>ust[i]) fm6_ko[i]=0
}

 tmm=fa7$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm7_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm7_tah=c(fm7_tah,aa)
}

fm7_ci=fa7$cis$ci
fm7_bias=(l-(fm7_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm7_mse =((l-(fm7_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm7_bg=NULL
fm7_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm7_ci[,kk[i]]*iler[i])
ust=c(ust,fm7_ci[,fs+kk[i]]*iler[i])
fm7_bg=c(fm7_bg,fm7_ci[,fs+kk[i]]*iler[i]-  fm7_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm7_tah[i]<alt[i] || fm7_tah[i]>ust[i]) fm7_ko[i]=0
}

 tmm=fa8$loadings[1:(fs*dgs)]
 tm=matrix(tmm,nrow = dgs)
 kk=NULL; ller=1:fs;iler=NULL
 for(i1 in 1:fs) #tm
 {
  ek=dgs^2
  for(i2 in ller) #l
  { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
  b=(sum((-tm[,i1]-l[,i2])^2))^0.5
  if (a<ek)
  {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
  if (b<ek)
  {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
  }
  ller=ller[-kkk]
 }
 fm8_tah=NULL
 for(i1 in 1:fs)
 { aa=tm[,kk[i1]]*iler[i1]
 fm8_tah=c(fm8_tah,aa)
 }

 fm8_ci=fa8$cis$ci
 fm8_bias=(l-(fm8_tah[1:(fs*dgs)]))[1:(fs*dgs)]
 fm8_mse =((l-(fm8_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
 fm8_bg=NULL
 fm8_ko=rep(1,fs*dgs)
 alt=NULL
 ust=NULL
 for(i in 1:fs)
 { alt=c(alt,fm8_ci[,kk[i]]*iler[i])
 ust=c(ust,fm8_ci[,fs+kk[i]]*iler[i])
 fm8_bg=c(fm8_bg,fm8_ci[,fs+kk[i]]*iler[i]-  fm8_ci[,kk[i]]*iler[i])
 }
 for(i in 1:(fs*dgs))
 {
  if (fm8_tah[i]<alt[i] || fm8_tah[i]>ust[i]) fm8_ko[i]=0
 }

  yn1=unlist(c(fm1_bias,fm1_mse,fm1_bg,fm1_ko))
 yn2=unlist(c(fm2_bias,fm2_mse,fm2_bg,fm2_ko))
 yn3=unlist(c(fm3_bias,fm3_mse,fm3_bg,fm3_ko))
 yn4=unlist(c(fm4_bias,fm4_mse,fm4_bg,fm4_ko))
 yn5=unlist(c(fm5_bias,fm5_mse,fm5_bg,fm5_ko))
 yn6=unlist(c(fm6_bias,fm6_mse,fm6_bg,fm6_ko))
 yn7=unlist(c(fm7_bias,fm7_mse,fm7_bg,fm7_ko))
 yn8=unlist(c(fm8_bias,fm8_mse,fm8_bg,fm8_ko))

pler=c(fa1$PVAL,fa2$PVAL,fa3$PVAL,fa4$PVAL,fa5$PVAL,fa6
$PVAL,fa7$PVAL,fa8$PVAL)
 aa=fa.parallel(x,plot = F)$nfact
 if (aa==fs) dr=1 else dr=0
 bas=c(bas,unlist(c(yn1,yn2,yn3,yn4,yn5,yn6,yn7,yn8,pler,dr)))


 #-----------------------------------------------------------------------------
 #covar=F,oblimin
 #-----------------------------------------------------------------------------

 l=oblimin(l1)$loadings[1:dgs,1:fs]
 cat('\14',n,ii)
 dev=F
 while(dev==F)
{ x=NULL
 for(j in 1:n){
   f=matrix(rnorm(fs),nrow = fs)

eps=matrix(c(rnorm(1,0,.2),rnorm(1,0,.3),rnorm(1,0,.4),rnorm(1,0,.5)
,rnorm(1,0,.5)),ncol= 1)
   x=rbind(x,t(l%*%f+eps))
 }

 yn1=NULL; yn2=NULL; yn3=NULL; yn4=NULL; yn5=NULL;
yn6=NULL; yn7=NULL; yn8=NULL
 ff1=try(fa1<-fa(x,nfactors = fs,fm='ml',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff2=try(fa2<-fa(x,nfactors = fs,fm='minres',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff3=try(fa3<-fa(x,nfactors = fs,fm='uls',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff4=try(fa4<-fa(x,nfactors = fs,fm='wls',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff5=try(fa5<-fa(x,nfactors = fs,fm='gls',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
```

```
 ff6=try(fa6<-fa(x,nfactors = fs,fm='pa',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff7=try(fa7<-fa(x,nfactors = fs,fm='minchi',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff8=try(fa8<-fa(x,nfactors = fs,fm='old.min',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)

 if
(!is.character(ff1)&&!is.character(ff2)&&!is.character(ff3)&&!is.ch
aracter(ff4)&&!is.character(ff5)&&!is.character(ff6)&&!is.character
(ff7)&&!is.character(ff8) )
 {
  dev=T
 }
  }

 tmm=fa1$loadings[1:(fs*dgs)]
 tm=matrix(tmm,nrow = dgs)
 kk=NULL; ller=1:fs;iler=NULL
 for(i1 in 1:fs) #tm
 {
  ek=dgs^2
  for(i2 in ller) #l
  { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
  b=(sum((-tm[,i1]-l[,i2])^2))^0.5
  if (a<ek)
  {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
  if (b<ek)
  {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
  }
  ller=ller[-kkk]
 }
 fm1_tah=NULL
 for(i1 in 1:fs)
 { aa=tm[,kk[i1]]*iler[i1]
 fm1_tah=c(fm1_tah,aa)
 }

 fm1_ci=fa1$cis$ci
 fm1_bias=(l-(fm1_tah[1:(fs*dgs)]))[1:(fs*dgs)]
 fm1_mse =((l-(fm1_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
 fm1_bg=NULL
 fm1_ko=rep(1,fs*dgs)
 alt=NULL
 ust=NULL
 for(i in 1:fs)
 { alt=c(alt,fm1_ci[,kk[i]]*iler[i])
 ust=c(ust,fm1_ci[,fs+kk[i]]*iler[i])
 fm1_bg=c(fm1_bg,fm1_ci[,fs+kk[i]]*iler[i]-  fm1_ci[,kk[i]]*iler[i])
 }
 for(i in 1:(fs*dgs))
 {
  if (fm1_tah[i]<alt[i] || fm1_tah[i]>ust[i]) fm1_ko[i]=0
 }

 tmm=fa2$loadings[1:(fs*dgs)]
 tm=matrix(tmm,nrow = dgs)
 kk=NULL; ller=1:fs;iler=NULL
 for(i1 in 1:fs) #tm
 {
  ek=dgs^2
  for(i2 in ller) #l
  { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
  b=(sum((-tm[,i1]-l[,i2])^2))^0.5
  if (a<ek)
  {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
  if (b<ek)
  {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
  }
  ller=ller[-kkk]
 }
 fm2_tah=NULL
 for(i1 in 1:fs)
 { aa=tm[,kk[i1]]*iler[i1]
 fm2_tah=c(fm2_tah,aa)
 }

 fm2_ci=fa2$cis$ci
 fm2_bias=(l-(fm2_tah[1:(fs*dgs)]))[1:(fs*dgs)]
 fm2_mse =((l-(fm2_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
 fm2_bg=NULL
 fm2_ko=rep(1,fs*dgs)
 alt=NULL
 ust=NULL
 for(i in 1:fs)
 { alt=c(alt,fm2_ci[,kk[i]]*iler[i])
 ust=c(ust,fm2_ci[,fs+kk[i]]*iler[i])
 fm2_bg=c(fm2_bg,fm2_ci[,fs+kk[i]]*iler[i]-  fm2_ci[,kk[i]]*iler[i])
 }
 for(i in 1:(fs*dgs))
 {
  if (fm2_tah[i]<alt[i] || fm2_tah[i]>ust[i]) fm2_ko[i]=0
 }



 tmm=fa3$loadings[1:(fs*dgs)]
 tm=matrix(tmm,nrow = dgs)
 kk=NULL; ller=1:fs;iler=NULL
 for(i1 in 1:fs) #tm
 {
  ek=dgs^2
  for(i2 in ller) #l
  { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
  b=(sum((-tm[,i1]-l[,i2])^2))^0.5
  if (a<ek)
  {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
  if (b<ek)
  {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
  }
  ller=ller[-kkk]
 }
 fm3_tah=NULL
 for(i1 in 1:fs)
 { aa=tm[,kk[i1]]*iler[i1]
 fm3_tah=c(fm3_tah,aa)
 }

 fm3_ci=fa3$cis$ci
 fm3_bias=(l-(fm3_tah[1:(fs*dgs)]))[1:(fs*dgs)]
 fm3_mse =((l-(fm3_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
 fm3_bg=NULL
 fm3_ko=rep(1,fs*dgs)
 alt=NULL
 ust=NULL
 for(i in 1:fs)
 { alt=c(alt,fm3_ci[,kk[i]]*iler[i])
 ust=c(ust,fm3_ci[,fs+kk[i]]*iler[i])
 fm3_bg=c(fm3_bg,fm3_ci[,fs+kk[i]]*iler[i]-  fm3_ci[,kk[i]]*iler[i])
 }
 for(i in 1:(fs*dgs))
 {
  if (fm3_tah[i]<alt[i] || fm3_tah[i]>ust[i]) fm3_ko[i]=0
 }

 tmm=fa4$loadings[1:(fs*dgs)]
 tm=matrix(tmm,nrow = dgs)
 kk=NULL; ller=1:fs;iler=NULL
 for(i1 in 1:fs) #tm
 {
  ek=dgs^2
  for(i2 in ller) #l
  { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
  b=(sum((-tm[,i1]-l[,i2])^2))^0.5
  if (a<ek)
  {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
  if (b<ek)
  {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
  }
  ller=ller[-kkk]
 }
 fm4_tah=NULL
 for(i1 in 1:fs)
 { aa=tm[,kk[i1]]*iler[i1]
 fm4_tah=c(fm4_tah,aa)
 }

 fm4_ci=fa1$cis$ci
 fm4_bias=(l-(fm4_tah[1:(fs*dgs)]))[1:(fs*dgs)]
 fm4_mse =((l-(fm4_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
 fm4_bg=NULL
 fm4_ko=rep(1,fs*dgs)
 alt=NULL
 ust=NULL
 for(i in 1:fs)
 { alt=c(alt,fm4_ci[,kk[i]]*iler[i])
 ust=c(ust,fm4_ci[,fs+kk[i]]*iler[i])
 fm4_bg=c(fm4_bg,fm4_ci[,fs+kk[i]]*iler[i]-  fm4_ci[,kk[i]]*iler[i])
```

```
}
for(i in 1:(fs*dgs))
{
 if (fm4_tah[i]<alt[i] || fm4_tah[i]>ust[i]) fm4_ko[i]=0
}

tmm=fa5$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm5_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm5_tah=c(fm5_tah,aa)
}

fm5_ci=fa5$cis$ci
fm5_bias=(l-(fm5_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm5_mse =((l-(fm5_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm5_bg=NULL
fm5_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm5_ci[,kk[i]]*iler[i])
ust=c(ust,fm5_ci[,fs+kk[i]]*iler[i])
fm5_bg=c(fm5_bg,fm5_ci[,fs+kk[i]]*iler[i]-  fm5_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm5_tah[i]<alt[i] || fm5_tah[i]>ust[i]) fm5_ko[i]=0
}

tmm=fa6$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm6_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm6_tah=c(fm6_tah,aa)
}
 fm6_ci=fa6$cis$ci
fm6_bias=(l-(fm6_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm6_mse =((l-(fm6_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm6_bg=NULL
fm6_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm6_ci[,kk[i]]*iler[i])
ust=c(ust,fm6_ci[,fs+kk[i]]*iler[i])
fm6_bg=c(fm6_bg,fm6_ci[,fs+kk[i]]*iler[i]-  fm6_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm6_tah[i]<alt[i] || fm6_tah[i]>ust[i]) fm6_ko[i]=0
}

tmm=fa7$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm7_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm7_tah=c(fm7_tah,aa)
}

fm7_ci=fa7$cis$ci
fm7_bias=(l-(fm7_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm7_mse =((l-(fm7_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm7_bg=NULL
fm7_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm7_ci[,kk[i]]*iler[i])
ust=c(ust,fm7_ci[,fs+kk[i]]*iler[i])
fm7_bg=c(fm7_bg,fm7_ci[,fs+kk[i]]*iler[i]-  fm7_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm7_tah[i]<alt[i] || fm7_tah[i]>ust[i]) fm7_ko[i]=0
}

tmm=fa8$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm8_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm8_tah=c(fm8_tah,aa)
}

fm8_ci=fa8$cis$ci
fm8_bias=(l-(fm8_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm8_mse =((l-(fm8_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm8_bg=NULL
fm8_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm8_ci[,kk[i]]*iler[i])
ust=c(ust,fm8_ci[,fs+kk[i]]*iler[i])
fm8_bg=c(fm8_bg,fm8_ci[,fs+kk[i]]*iler[i]-  fm8_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm8_tah[i]<alt[i] || fm8_tah[i]>ust[i]) fm8_ko[i]=0
}
yn1=unlist(c(fm1_bias,fm1_mse,fm1_bg,fm1_ko))
yn2=unlist(c(fm2_bias,fm2_mse,fm2_bg,fm2_ko))
yn3=unlist(c(fm3_bias,fm3_mse,fm3_bg,fm3_ko))
yn4=unlist(c(fm4_bias,fm4_mse,fm4_bg,fm4_ko))
yn5=unlist(c(fm5_bias,fm5_mse,fm5_bg,fm5_ko))
yn6=unlist(c(fm6_bias,fm6_mse,fm6_bg,fm6_ko))
yn7=unlist(c(fm7_bias,fm7_mse,fm7_bg,fm7_ko))
yn8=unlist(c(fm8_bias,fm8_mse,fm8_bg,fm8_ko))

pler=c(fa1$PVAL,fa2$PVAL,fa3$PVAL,fa4$PVAL,fa5$PVAL,fa6
$PVAL,fa7$PVAL,fa8$PVAL)
 aa=fa.parallel(x,plot = F)$nfact
```

```
 if (aa==fs) dr=1 else dr=0
 bas=c(bas,unlist(c(yn1,yn2,yn3,yn4,yn5,yn6,yn7,yn8,pler,dr)))

 #-----------------------------------------------------------------------------
 #covar=T,promax
 #-----------------------------------------------------------------------------

 l=promax(l1)$loadings[1:dgs,1:fs]
 cat('\14',n,ii)
 dev=F
 while(dev==F)
{ x=NULL
 for(j in 1:n){
  f=matrix(rnorm(fs),nrow = fs)

eps=matrix(c(rnorm(1,0,.2),rnorm(1,0,.3),rnorm(1,0,.4),rnorm(1,0,.5)
,rnorm(1,0,.5)),ncol= 1)
   x=rbind(x,t(l%*%f+eps))
 }

 yn1=NULL; yn2=NULL; yn3=NULL; yn4=NULL; yn5=NULL;
yn6=NULL; yn7=NULL; yn8=NULL
 ff1=try(fa1<-fa(x,nfactors = fs,fm='ml',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff2=try(fa2<-fa(x,nfactors = fs,fm='minres',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff3=try(fa3<-fa(x,nfactors = fs,fm='uls',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff4=try(fa4<-fa(x,nfactors = fs,fm='wls',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff5=try(fa5<-fa(x,nfactors = fs,fm='gls',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff6=try(fa6<-fa(x,nfactors = fs,fm='pa',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff7=try(fa7<-fa(x,nfactors = fs,fm='minchi',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff8=try(fa8<-fa(x,nfactors = fs,fm='old.min',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)

 if
(!is.character(ff1)&&!is.character(ff2)&&!is.character(ff3)&&!is.ch
aracter(ff4)&&!is.character(ff5)&&!is.character(ff6)&&!is.character
(ff7)&&!is.character(ff8) )
 {
  dev=T
 }
 }

 tmm=fa1$loadings[1:(fs*dgs)]
 tm=matrix(tmm,nrow = dgs)
 kk=NULL; ller=1:fs;iler=NULL
 for(i1 in 1:fs) #tm
 {
  ek=dgs^2
  for(i2 in ller) #l
  { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
  b=(sum((-tm[,i1]-l[,i2])^2))^0.5
  if (a<ek)
  {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
  if (b<ek)
  {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
  }
  ller=ller[-kkk]
 }
 fm1_tah=NULL
 for(i1 in 1:fs)
 { aa=tm[,kk[i1]]*iler[i1]
 fm1_tah=c(fm1_tah,aa)
 }

 fm1_ci=fa1$cis$ci
 fm1_bias=(l-(fm1_tah[1:(fs*dgs)]))[1:(fs*dgs)]
 fm1_mse =((l-(fm1_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
 fm1_bg=NULL
 fm1_ko=rep(1,fs*dgs)
 alt=NULL
 ust=NULL
 for(i in 1:fs)
 { alt=c(alt,fm1_ci[,kk[i]]*iler[i])
 ust=c(ust,fm1_ci[,fs+kk[i]]*iler[i])
 fm1_bg=c(fm1_bg,fm1_ci[,fs+kk[i]]*iler[i]-  fm1_ci[,kk[i]]*iler[i])
 }
 for(i in 1:(fs*dgs))
 {
  if (fm1_tah[i]<alt[i] || fm1_tah[i]>ust[i]) fm1_ko[i]=0
 }

tmm=fa2$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm2_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm2_tah=c(fm2_tah,aa)
}


fm2_ci=fa2$cis$ci
fm2_bias=(l-(fm2_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm2_mse =((l-(fm2_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm2_bg=NULL
fm2_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm2_ci[,kk[i]]*iler[i])
ust=c(ust,fm2_ci[,fs+kk[i]]*iler[i])
fm2_bg=c(fm2_bg,fm2_ci[,fs+kk[i]]*iler[i]-  fm2_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm2_tah[i]<alt[i] || fm2_tah[i]>ust[i]) fm2_ko[i]=0
}

tmm=fa3$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm3_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm3_tah=c(fm3_tah,aa)
}

fm3_ci=fa3$cis$ci
fm3_bias=(l-(fm3_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm3_mse =((l-(fm3_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm3_bg=NULL
fm3_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm3_ci[,kk[i]]*iler[i])
ust=c(ust,fm3_ci[,fs+kk[i]]*iler[i])
fm3_bg=c(fm3_bg,fm3_ci[,fs+kk[i]]*iler[i]-  fm3_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm3_tah[i]<alt[i] || fm3_tah[i]>ust[i]) fm3_ko[i]=0
}



tmm=fa4$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
```

```
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm4_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm4_tah=c(fm4_tah,aa)
}

fm4_ci=fa1$cis$ci
fm4_bias=(l-(fm4_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm4_mse =((l-(fm4_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm4_bg=NULL
fm4_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm4_ci[,kk[i]]*iler[i])
ust=c(ust,fm4_ci[,fs+kk[i]]*iler[i])
fm4_bg=c(fm4_bg,fm4_ci[,fs+kk[i]]*iler[i]-  fm4_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm4_tah[i]<alt[i] || fm4_tah[i]>ust[i]) fm4_ko[i]=0
}

tmm=fa5$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm5_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm5_tah=c(fm5_tah,aa)
}

fm5_ci=fa5$cis$ci
fm5_bias=(l-(fm5_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm5_mse =((l-(fm5_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm5_bg=NULL
fm5_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm5_ci[,kk[i]]*iler[i])
ust=c(ust,fm5_ci[,fs+kk[i]]*iler[i])
fm5_bg=c(fm5_bg,fm5_ci[,fs+kk[i]]*iler[i]-  fm5_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm5_tah[i]<alt[i] || fm5_tah[i]>ust[i]) fm5_ko[i]=0
}

tmm=fa6$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm6_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm6_tah=c(fm6_tah,aa)
}

fm6_ci=fa6$cis$ci
fm6_bias=(l-(fm6_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm6_mse =((l-(fm6_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm6_bg=NULL
fm6_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm6_ci[,kk[i]]*iler[i])
ust=c(ust,fm6_ci[,fs+kk[i]]*iler[i])
fm6_bg=c(fm6_bg,fm6_ci[,fs+kk[i]]*iler[i]-  fm6_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm6_tah[i]<alt[i] || fm6_tah[i]>ust[i]) fm6_ko[i]=0
}

tmm=fa7$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm7_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm7_tah=c(fm7_tah,aa)
}

fm7_ci=fa7$cis$ci
fm7_bias=(l-(fm7_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm7_mse =((l-(fm7_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm7_bg=NULL
fm7_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm7_ci[,kk[i]]*iler[i])
ust=c(ust,fm7_ci[,fs+kk[i]]*iler[i])
fm7_bg=c(fm7_bg,fm7_ci[,fs+kk[i]]*iler[i]-  fm7_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm7_tah[i]<alt[i] || fm7_tah[i]>ust[i]) fm7_ko[i]=0
}

tmm=fa8$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm8_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm8_tah=c(fm8_tah,aa)
```

```
 }

 fm8_ci=fa8$cis$ci
 fm8_bias=(l-(fm8_tah[1:(fs*dgs)]))[1:(fs*dgs)]
 fm8_mse =((l-(fm8_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
 fm8_bg=NULL
 fm8_ko=rep(1,fs*dgs)
 alt=NULL
 ust=NULL
 for(i in 1:fs)
 { alt=c(alt,fm8_ci[,kk[i]]*iler[i])
 ust=c(ust,fm8_ci[,fs+kk[i]]*iler[i])
 fm8_bg=c(fm8_bg,fm8_ci[,fs+kk[i]]*iler[i]-  fm8_ci[,kk[i]]*iler[i])
 }
 for(i in 1:(fs*dgs))
 {
  if (fm8_tah[i]<alt[i] || fm8_tah[i]>ust[i]) fm8_ko[i]=0
 }

 yn1=unlist(c(fm1_bias,fm1_mse,fm1_bg,fm1_ko))
 yn2=unlist(c(fm2_bias,fm2_mse,fm2_bg,fm2_ko))
 yn3=unlist(c(fm3_bias,fm3_mse,fm3_bg,fm3_ko))
 yn4=unlist(c(fm4_bias,fm4_mse,fm4_bg,fm4_ko))
 yn5=unlist(c(fm5_bias,fm5_mse,fm5_bg,fm5_ko))
 yn6=unlist(c(fm6_bias,fm6_mse,fm6_bg,fm6_ko))
 yn7=unlist(c(fm7_bias,fm7_mse,fm7_bg,fm7_ko))
 yn8=unlist(c(fm8_bias,fm8_mse,fm8_bg,fm8_ko))

pler=c(fa1$PVAL,fa2$PVAL,fa3$PVAL,fa4$PVAL,fa5$PVAL,fa6
$PVAL,fa7$PVAL,fa8$PVAL)
 aa=fa.parallel(x,plot = F)$nfact
 if (aa==fs) dr=1 else dr=0
 bas=c(bas,unlist(c(yn1,yn2,yn3,yn4,yn5,yn6,yn7,yn8,pler,dr)))

 #---------------------------------------------------------------------------
 #covar=F,promax
 #---------------------------------------------------------------------------

 l=promax(l1)$loadings[1:dgs,1:fs]
 cat('\14',n,ii)
 dev=F
 while(dev==F)
{ x=NULL
 for(j in 1:n){
  f=matrix(rnorm(fs),nrow = fs)

eps=matrix(c(rnorm(1,0,.2),rnorm(1,0,.3),rnorm(1,0,.4),rnorm(1,0,.5)
,rnorm(1,0,.5)),ncol= 1)
   x=rbind(x,t(l%*%f+eps))
 }

 yn1=NULL; yn2=NULL; yn3=NULL; yn4=NULL; yn5=NULL;
yn6=NULL; yn7=NULL; yn8=NULL
 ff1=try(fa1<-fa(x,nfactors = fs,fm='ml',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff2=try(fa2<-fa(x,nfactors = fs,fm='minres',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff3=try(fa3<-fa(x,nfactors = fs,fm='uls',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff4=try(fa4<-fa(x,nfactors = fs,fm='wls',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff5=try(fa5<-fa(x,nfactors = fs,fm='gls',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff6=try(fa6<-fa(x,nfactors = fs,fm='pa',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff7=try(fa7<-fa(x,nfactors = fs,fm='minchi',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff8=try(fa8<-fa(x,nfactors = fs,fm='old.min',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)

 if
(!is.character(ff1)&&!is.character(ff2)&&!is.character(ff3)&&!is.ch
aracter(ff4)&&!is.character(ff5)&&!is.character(ff6)&&!is.character
(ff7)&&!is.character(ff8) )
 {
  dev=T
 }
  }


 tmm=fa1$loadings[1:(fs*dgs)]
 tm=matrix(tmm,nrow = dgs)
 kk=NULL; ller=1:fs;iler=NULL
 for(i1 in 1:fs) #tm
 {
  ek=dgs^2
  for(i2 in ller) #l
  { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
  b=(sum((-tm[,i1]-l[,i2])^2))^0.5
  if (a<ek)
  {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
  if (b<ek)
  {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
  }
  ller=ller[-kkk]
}
fm1_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm1_tah=c(fm1_tah,aa)
}

fm1_ci=fa1$cis$ci
fm1_bias=(l-(fm1_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm1_mse =((l-(fm1_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm1_bg=NULL
fm1_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm1_ci[,kk[i]]*iler[i])
ust=c(ust,fm1_ci[,fs+kk[i]]*iler[i])
fm1_bg=c(fm1_bg,fm1_ci[,fs+kk[i]]*iler[i]-  fm1_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm1_tah[i]<alt[i] || fm1_tah[i]>ust[i]) fm1_ko[i]=0
}

tmm=fa2$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm2_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm2_tah=c(fm2_tah,aa)
}

 fm2_ci=fa2$cis$ci
fm2_bias=(l-(fm2_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm2_mse =((l-(fm2_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm2_bg=NULL
fm2_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm2_ci[,kk[i]]*iler[i])
ust=c(ust,fm2_ci[,fs+kk[i]]*iler[i])
fm2_bg=c(fm2_bg,fm2_ci[,fs+kk[i]]*iler[i]-  fm2_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm2_tah[i]<alt[i] || fm2_tah[i]>ust[i]) fm2_ko[i]=0
}

tmm=fa3$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
```

```
 ller=ller[-kkk]
}
fm3_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm3_tah=c(fm3_tah,aa)
}

fm3_ci=fa3$cis$ci
fm3_bias=(l-(fm3_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm3_mse =((l-(fm3_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm3_bg=NULL
fm3_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm3_ci[,kk[i]]*iler[i])
ust=c(ust,fm3_ci[,fs+kk[i]]*iler[i])
fm3_bg=c(fm3_bg,fm3_ci[,fs+kk[i]]*iler[i]-  fm3_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm3_tah[i]<alt[i] || fm3_tah[i]>ust[i]) fm3_ko[i]=0
}

tmm=fa4$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm4_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm4_tah=c(fm4_tah,aa)
}
fm4_ci=fa1$cis$ci
fm4_bias=(l-(fm4_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm4_mse =((l-(fm4_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm4_bg=NULL
fm4_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm4_ci[,kk[i]]*iler[i])
ust=c(ust,fm4_ci[,fs+kk[i]]*iler[i])
fm4_bg=c(fm4_bg,fm4_ci[,fs+kk[i]]*iler[i]-  fm4_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm4_tah[i]<alt[i] || fm4_tah[i]>ust[i]) fm4_ko[i]=0
}

tmm=fa5$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm5_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm5_tah=c(fm5_tah,aa)
}

fm5_ci=fa5$cis$ci
fm5_bias=(l-(fm5_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm5_mse =((l-(fm5_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm5_bg=NULL
fm5_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm5_ci[,kk[i]]*iler[i])
ust=c(ust,fm5_ci[,fs+kk[i]]*iler[i])
fm5_bg=c(fm5_bg,fm5_ci[,fs+kk[i]]*iler[i]-  fm5_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm5_tah[i]<alt[i] || fm5_tah[i]>ust[i]) fm5_ko[i]=0
}

tmm=fa6$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm6_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm6_tah=c(fm6_tah,aa)
}

fm6_ci=fa6$cis$ci
fm6_bias=(l-(fm6_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm6_mse =((l-(fm6_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm6_bg=NULL
fm6_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm6_ci[,kk[i]]*iler[i])
ust=c(ust,fm6_ci[,fs+kk[i]]*iler[i])
fm6_bg=c(fm6_bg,fm6_ci[,fs+kk[i]]*iler[i]-  fm6_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm6_tah[i]<alt[i] || fm6_tah[i]>ust[i]) fm6_ko[i]=0
}

tmm=fa7$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm7_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm7_tah=c(fm7_tah,aa)
}

fm7_ci=fa7$cis$ci
fm7_bias=(l-(fm7_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm7_mse =((l-(fm7_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm7_bg=NULL
fm7_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm7_ci[,kk[i]]*iler[i])
```

```
 ust=c(ust,fm7_ci[,fs+kk[i]]*iler[i])
 fm7_bg=c(fm7_bg,fm7_ci[,fs+kk[i]]*iler[i]-  fm7_ci[,kk[i]]*iler[i])
 }
 for(i in 1:(fs*dgs))
 {
  if (fm7_tah[i]<alt[i] || fm7_tah[i]>ust[i]) fm7_ko[i]=0
 }

 tmm=fa8$loadings[1:(fs*dgs)]
 tm=matrix(tmm,nrow = dgs)
 kk=NULL; ller=1:fs;iler=NULL
 for(i1 in 1:fs) #tm
 {
  ek=dgs^2
  for(i2 in ller) #l
  { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
  b=(sum((-tm[,i1]-l[,i2])^2))^0.5
  if (a<ek)
  {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
  if (b<ek)
  {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
  }
  ller=ller[-kkk]
 }
 fm8_tah=NULL
 for(i1 in 1:fs)
 { aa=tm[,kk[i1]]*iler[i1]
 fm8_tah=c(fm8_tah,aa)
 }

 fm8_ci=fa8$cis$ci
 fm8_bias=(l-(fm8_tah[1:(fs*dgs)]))[1:(fs*dgs)]
 fm8_mse =((l-(fm8_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
 fm8_bg=NULL
 fm8_ko=rep(1,fs*dgs)
 alt=NULL
 ust=NULL
 for(i in 1:fs)
 { alt=c(alt,fm8_ci[,kk[i]]*iler[i])
 ust=c(ust,fm8_ci[,fs+kk[i]]*iler[i])
 fm8_bg=c(fm8_bg,fm8_ci[,fs+kk[i]]*iler[i]-  fm8_ci[,kk[i]]*iler[i])
 }
 for(i in 1:(fs*dgs))
 {
  if (fm8_tah[i]<alt[i] || fm8_tah[i]>ust[i]) fm8_ko[i]=0
 }
 yn1=unlist(c(fm1_bias,fm1_mse,fm1_bg,fm1_ko))
 yn2=unlist(c(fm2_bias,fm2_mse,fm2_bg,fm2_ko))
 yn3=unlist(c(fm3_bias,fm3_mse,fm3_bg,fm3_ko))
 yn4=unlist(c(fm4_bias,fm4_mse,fm4_bg,fm4_ko))
 yn5=unlist(c(fm5_bias,fm5_mse,fm5_bg,fm5_ko))
 yn6=unlist(c(fm6_bias,fm6_mse,fm6_bg,fm6_ko))
 yn7=unlist(c(fm7_bias,fm7_mse,fm7_bg,fm7_ko))
 yn8=unlist(c(fm8_bias,fm8_mse,fm8_bg,fm8_ko))

pler=c(fa1$PVAL,fa2$PVAL,fa3$PVAL,fa4$PVAL,fa5$PVAL,fa6
$PVAL,fa7$PVAL,fa8$PVAL)
 aa=fa.parallel(x,plot = F)$nfact
 if (aa==fs) dr=1 else dr=0
 bas=c(bas,unlist(c(yn1,yn2,yn3,yn4,yn5,yn6,yn7,yn8,pler,dr)))

 #-----------------------------------------------------------------------------
 #covar=T,quartimax
 #-----------------------------------------------------------------------------

 l=quartimax(l1)$loadings[1:dgs,1:fs]
 cat('\14',n,ii)
 dev=F
 while(dev==F)
{ x=NULL
  for(j in 1:n){
   f=matrix(rnorm(fs),nrow = fs)

eps=matrix(c(rnorm(1,0,.2),rnorm(1,0,.3),rnorm(1,0,.4),rnorm(1,0,.5)
,rnorm(1,0,.5)),ncol= 1)
    x=rbind(x,t(l%*%f+eps))
 }

 yn1=NULL; yn2=NULL; yn3=NULL; yn4=NULL; yn5=NULL;
yn6=NULL; yn7=NULL; yn8=NULL
 ff1=try(fa1<-fa(x,nfactors = fs,fm='ml',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff2=try(fa2<-fa(x,nfactors = fs,fm='minres',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff3=try(fa3<-fa(x,nfactors = fs,fm='uls',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff4=try(fa4<-fa(x,nfactors = fs,fm='wls',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff5=try(fa5<-fa(x,nfactors = fs,fm='gls',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff6=try(fa6<-fa(x,nfactors = fs,fm='pa',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff7=try(fa7<-fa(x,nfactors = fs,fm='minchi',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff8=try(fa8<-fa(x,nfactors = fs,fm='old.min',covar = T,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)

 if
(!is.character(ff1)&&!is.character(ff2)&&!is.character(ff3)&&!is.ch
aracter(ff4)&&!is.character(ff5)&&!is.character(ff6)&&!is.character
(ff7)&&!is.character(ff8) )
 {
  dev=T
 }
  }

 tmm=fa1$loadings[1:(fs*dgs)]
 tm=matrix(tmm,nrow = dgs)
 kk=NULL; ller=1:fs;iler=NULL
 for(i1 in 1:fs) #tm
 {
  ek=dgs^2
  for(i2 in ller) #l
  { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
  b=(sum((-tm[,i1]-l[,i2])^2))^0.5
  if (a<ek)
  {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
  if (b<ek)
  {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
  }
  ller=ller[-kkk]
 }
 fm1_tah=NULL
 for(i1 in 1:fs)
 { aa=tm[,kk[i1]]*iler[i1]
 fm1_tah=c(fm1_tah,aa)
 }

 fm1_ci=fa1$cis$ci
 fm1_bias=(l-(fm1_tah[1:(fs*dgs)]))[1:(fs*dgs)]
 fm1_mse =((l-(fm1_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
 fm1_bg=NULL
 fm1_ko=rep(1,fs*dgs)
 alt=NULL
 ust=NULL
 for(i in 1:fs)
 { alt=c(alt,fm1_ci[,kk[i]]*iler[i])
 ust=c(ust,fm1_ci[,fs+kk[i]]*iler[i])
 fm1_bg=c(fm1_bg,fm1_ci[,fs+kk[i]]*iler[i]-  fm1_ci[,kk[i]]*iler[i])
 }
 for(i in 1:(fs*dgs))
 {
  if (fm1_tah[i]<alt[i] || fm1_tah[i]>ust[i]) fm1_ko[i]=0
 }
 tmm=fa2$loadings[1:(fs*dgs)]
 tm=matrix(tmm,nrow = dgs)
 kk=NULL; ller=1:fs;iler=NULL
 for(i1 in 1:fs) #tm
 {
  ek=dgs^2
  for(i2 in ller) #l
  { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
  b=(sum((-tm[,i1]-l[,i2])^2))^0.5
  if (a<ek)
  {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
  if (b<ek)
  {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
  }
  ller=ller[-kkk]
 }
 fm2_tah=NULL
 for(i1 in 1:fs)
 { aa=tm[,kk[i1]]*iler[i1]
 fm2_tah=c(fm2_tah,aa)
 }


 fm2_ci=fa2$cis$ci
 fm2_bias=(l-(fm2_tah[1:(fs*dgs)]))[1:(fs*dgs)]
 fm2_mse =((l-(fm2_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
 fm2_bg=NULL
 fm2_ko=rep(1,fs*dgs)
```

```
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm2_ci[,kk[i]]*iler[i])
ust=c(ust,fm2_ci[,fs+kk[i]]*iler[i])
fm2_bg=c(fm2_bg,fm2_ci[,fs+kk[i]]*iler[i]-  fm2_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm2_tah[i]<alt[i] || fm2_tah[i]>ust[i]) fm2_ko[i]=0
}

tmm=fa3$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm3_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm3_tah=c(fm3_tah,aa)
}

fm3_ci=fa3$cis$ci
fm3_bias=(l-(fm3_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm3_mse =((l-(fm3_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm3_bg=NULL
fm3_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm3_ci[,kk[i]]*iler[i])
ust=c(ust,fm3_ci[,fs+kk[i]]*iler[i])
fm3_bg=c(fm3_bg,fm3_ci[,fs+kk[i]]*iler[i]-  fm3_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm3_tah[i]<alt[i] || fm3_tah[i]>ust[i]) fm3_ko[i]=0
}

tmm=fa4$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm4_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm4_tah=c(fm4_tah,aa)
}

fm4_ci=fa1$cis$ci
fm4_bias=(l-(fm4_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm4_mse =((l-(fm4_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm4_bg=NULL
fm4_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm4_ci[,kk[i]]*iler[i])
ust=c(ust,fm4_ci[,fs+kk[i]]*iler[i])
fm4_bg=c(fm4_bg,fm4_ci[,fs+kk[i]]*iler[i]-  fm4_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm4_tah[i]<alt[i] || fm4_tah[i]>ust[i]) fm4_ko[i]=0
}

tmm=fa5$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm5_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm5_tah=c(fm5_tah,aa)
}

fm5_ci=fa5$cis$ci
fm5_bias=(l-(fm5_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm5_mse =((l-(fm5_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm5_bg=NULL
fm5_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm5_ci[,kk[i]]*iler[i])
ust=c(ust,fm5_ci[,fs+kk[i]]*iler[i])
fm5_bg=c(fm5_bg,fm5_ci[,fs+kk[i]]*iler[i]-  fm5_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm5_tah[i]<alt[i] || fm5_tah[i]>ust[i]) fm5_ko[i]=0
}

tmm=fa6$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm6_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm6_tah=c(fm6_tah,aa)
}

fm6_ci=fa6$cis$ci
fm6_bias=(l-(fm6_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm6_mse =((l-(fm6_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm6_bg=NULL
fm6_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm6_ci[,kk[i]]*iler[i])
ust=c(ust,fm6_ci[,fs+kk[i]]*iler[i])
fm6_bg=c(fm6_bg,fm6_ci[,fs+kk[i]]*iler[i]-  fm6_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm6_tah[i]<alt[i] || fm6_tah[i]>ust[i]) fm6_ko[i]=0
}

 tmm=fa7$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
```

```
 {
  ek=dgs^2
  for(i2 in ller) #l
  { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
  b=(sum((-tm[,i1]-l[,i2])^2))^0.5
  if (a<ek)
  {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
  if (b<ek)
  {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
  }
  ller=ller[-kkk]
 }
 fm7_tah=NULL
 for(i1 in 1:fs)
 { aa=tm[,kk[i1]]*iler[i1]
 fm7_tah=c(fm7_tah,aa)
 }

 fm7_ci=fa7$cis$ci
 fm7_bias=(l-(fm7_tah[1:(fs*dgs)]))[1:(fs*dgs)]
 fm7_mse =((l-(fm7_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
 fm7_bg=NULL
 fm7_ko=rep(1,fs*dgs)
 alt=NULL
 ust=NULL
 for(i in 1:fs)
 { alt=c(alt,fm7_ci[,kk[i]]*iler[i])
 ust=c(ust,fm7_ci[,fs+kk[i]]*iler[i])
 fm7_bg=c(fm7_bg,fm7_ci[,fs+kk[i]]*iler[i]-  fm7_ci[,kk[i]]*iler[i])
 }
 for(i in 1:(fs*dgs))
 {
  if (fm7_tah[i]<alt[i] || fm7_tah[i]>ust[i]) fm7_ko[i]=0
 }

  tmm=fa8$loadings[1:(fs*dgs)]
 tm=matrix(tmm,nrow = dgs)
 kk=NULL; ller=1:fs;iler=NULL
 for(i1 in 1:fs) #tm
 {
  ek=dgs^2
  for(i2 in ller) #l
  { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
  b=(sum((-tm[,i1]-l[,i2])^2))^0.5
  if (a<ek)
  {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
  if (b<ek)
  {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
  }
  ller=ller[-kkk]
 }
 fm8_tah=NULL
 for(i1 in 1:fs)
 { aa=tm[,kk[i1]]*iler[i1]
 fm8_tah=c(fm8_tah,aa)
 }

 fm8_ci=fa8$cis$ci
 fm8_bias=(l-(fm8_tah[1:(fs*dgs)]))[1:(fs*dgs)]
 fm8_mse =((l-(fm8_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
 fm8_bg=NULL
 fm8_ko=rep(1,fs*dgs)
 alt=NULL
 ust=NULL
 for(i in 1:fs)
 { alt=c(alt,fm8_ci[,kk[i]]*iler[i])
 ust=c(ust,fm8_ci[,fs+kk[i]]*iler[i])
 fm8_bg=c(fm8_bg,fm8_ci[,fs+kk[i]]*iler[i]-  fm8_ci[,kk[i]]*iler[i])
 }
 for(i in 1:(fs*dgs))
 {
  if (fm8_tah[i]<alt[i] || fm8_tah[i]>ust[i]) fm8_ko[i]=0
 }

 yn1=unlist(c(fm1_bias,fm1_mse,fm1_bg,fm1_ko))
 yn2=unlist(c(fm2_bias,fm2_mse,fm2_bg,fm2_ko))
 yn3=unlist(c(fm3_bias,fm3_mse,fm3_bg,fm3_ko))
 yn4=unlist(c(fm4_bias,fm4_mse,fm4_bg,fm4_ko))
 yn5=unlist(c(fm5_bias,fm5_mse,fm5_bg,fm5_ko))
 yn6=unlist(c(fm6_bias,fm6_mse,fm6_bg,fm6_ko))
 yn7=unlist(c(fm7_bias,fm7_mse,fm7_bg,fm7_ko))
 yn8=unlist(c(fm8_bias,fm8_mse,fm8_bg,fm8_ko))

pler=c(fa1$PVAL,fa2$PVAL,fa3$PVAL,fa4$PVAL,fa5$PVAL,fa6
$PVAL,fa7$PVAL,fa8$PVAL)
 aa=fa.parallel(x,plot = F)$nfact
 if (aa==fs) dr=1 else dr=0
 bas=c(bas,unlist(c(yn1,yn2,yn3,yn4,yn5,yn6,yn7,yn8,pler,dr)))

 #-----------------------------------------------------------------------------
 #covar=F,quartimax
 #-----------------------------------------------------------------------------

 l=quartimax(l1)$loadings[1:dgs,1:fs]
 cat('\14',n,ii)
 dev=F
 while(dev==F)
{ x=NULL
 for(j in 1:n){
  f=matrix(rnorm(fs),nrow = fs)

eps=matrix(c(rnorm(1,0,.2),rnorm(1,0,.3),rnorm(1,0,.4),rnorm(1,0,.5)
,rnorm(1,0,.5)),ncol= 1)
   x=rbind(x,t(l%*%f+eps))
 }

 yn1=NULL; yn2=NULL; yn3=NULL; yn4=NULL; yn5=NULL;
yn6=NULL; yn7=NULL; yn8=NULL
 ff1=try(fa1<-fa(x,nfactors = fs,fm='ml',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff2=try(fa2<-fa(x,nfactors = fs,fm='minres',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff3=try(fa3<-fa(x,nfactors = fs,fm='uls',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff4=try(fa4<-fa(x,nfactors = fs,fm='wls',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff5=try(fa5<-fa(x,nfactors = fs,fm='gls',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff6=try(fa6<-fa(x,nfactors = fs,fm='pa',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff7=try(fa7<-fa(x,nfactors = fs,fm='minchi',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)
 ff8=try(fa8<-fa(x,nfactors = fs,fm='old.min',covar = F,rotate =
"varimax",n.iter=R,max.iter = mis),silent=T)

 if
(!is.character(ff1)&&!is.character(ff2)&&!is.character(ff3)&&!is.ch
aracter(ff4)&&!is.character(ff5)&&!is.character(ff6)&&!is.character
(ff7)&&!is.character(ff8) )
 {
  dev=T
 }
  }

  tmm=fa1$loadings[1:(fs*dgs)]
  tm=matrix(tmm,nrow = dgs)
  kk=NULL; ller=1:fs;iler=NULL
  for(i1 in 1:fs) #tm
  {
   ek=dgs^2
   for(i2 in ller) #l
   { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
   b=(sum((-tm[,i1]-l[,i2])^2))^0.5
   if (a<ek)
   {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
   if (b<ek)
   {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
   }
   ller=ller[-kkk]
  }
  fm1_tah=NULL
  for(i1 in 1:fs)
  { aa=tm[,kk[i1]]*iler[i1]
  fm1_tah=c(fm1_tah,aa)
  }

   fm1_ci=fa1$cis$ci
  fm1_bias=(l-(fm1_tah[1:(fs*dgs)]))[1:(fs*dgs)]
  fm1_mse =((l-(fm1_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
  fm1_bg=NULL
  fm1_ko=rep(1,fs*dgs)
  alt=NULL
  ust=NULL
  for(i in 1:fs)
  { alt=c(alt,fm1_ci[,kk[i]]*iler[i])
  ust=c(ust,fm1_ci[,fs+kk[i]]*iler[i])
  fm1_bg=c(fm1_bg,fm1_ci[,fs+kk[i]]*iler[i]-  fm1_ci[,kk[i]]*iler[i])
  }
  for(i in 1:(fs*dgs))
  {
   if (fm1_tah[i]<alt[i] || fm1_tah[i]>ust[i]) fm1_ko[i]=0
  }
```

```
 tmm=fa2$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm2_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm2_tah=c(fm2_tah,aa)
}

fm2_ci=fa2$cis$ci
fm2_bias=(l-(fm2_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm2_mse =((l-(fm2_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm2_bg=NULL
fm2_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm2_ci[,kk[i]]*iler[i])
ust=c(ust,fm2_ci[,fs+kk[i]]*iler[i])
fm2_bg=c(fm2_bg,fm2_ci[,fs+kk[i]]*iler[i]-  fm2_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm2_tah[i]<alt[i] || fm2_tah[i]>ust[i]) fm2_ko[i]=0
}

 tmm=fa3$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm3_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm3_tah=c(fm3_tah,aa)
}

fm3_ci=fa3$cis$ci
fm3_bias=(l-(fm3_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm3_mse =((l-(fm3_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm3_bg=NULL
fm3_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm3_ci[,kk[i]]*iler[i])
ust=c(ust,fm3_ci[,fs+kk[i]]*iler[i])
fm3_bg=c(fm3_bg,fm3_ci[,fs+kk[i]]*iler[i]-  fm3_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm3_tah[i]<alt[i] || fm3_tah[i]>ust[i]) fm3_ko[i]=0
}




tmm=fa4$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm4_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm4_tah=c(fm4_tah,aa)
}

fm4_ci=fa1$cis$ci
fm4_bias=(l-(fm4_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm4_mse =((l-(fm4_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm4_bg=NULL
fm4_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm4_ci[,kk[i]]*iler[i])
ust=c(ust,fm4_ci[,fs+kk[i]]*iler[i])
fm4_bg=c(fm4_bg,fm4_ci[,fs+kk[i]]*iler[i]-  fm4_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm4_tah[i]<alt[i] || fm4_tah[i]>ust[i]) fm4_ko[i]=0
}

 tmm=fa5$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm5_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm5_tah=c(fm5_tah,aa)
}

fm5_ci=fa5$cis$ci
fm5_bias=(l-(fm5_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm5_mse =((l-(fm5_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm5_bg=NULL
fm5_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm5_ci[,kk[i]]*iler[i])
ust=c(ust,fm5_ci[,fs+kk[i]]*iler[i])
fm5_bg=c(fm5_bg,fm5_ci[,fs+kk[i]]*iler[i]-  fm5_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm5_tah[i]<alt[i] || fm5_tah[i]>ust[i]) fm5_ko[i]=0
}

 tmm=fa6$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
```

```
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm6_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm6_tah=c(fm6_tah,aa)
}

fm6_ci=fa6$cis$ci
fm6_bias=(l-(fm6_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm6_mse =((l-(fm6_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm6_bg=NULL
fm6_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm6_ci[,kk[i]]*iler[i])
ust=c(ust,fm6_ci[,fs+kk[i]]*iler[i])
fm6_bg=c(fm6_bg,fm6_ci[,fs+kk[i]]*iler[i]-  fm6_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm6_tah[i]<alt[i] || fm6_tah[i]>ust[i]) fm6_ko[i]=0
}

tmm=fa7$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm7_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm7_tah=c(fm7_tah,aa)
}

fm7_ci=fa7$cis$ci
fm7_bias=(l-(fm7_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm7_mse =((l-(fm7_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm7_bg=NULL
fm7_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm7_ci[,kk[i]]*iler[i])
ust=c(ust,fm7_ci[,fs+kk[i]]*iler[i])
fm7_bg=c(fm7_bg,fm7_ci[,fs+kk[i]]*iler[i]-  fm7_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm7_tah[i]<alt[i] || fm7_tah[i]>ust[i]) fm7_ko[i]=0
}

tmm=fa8$loadings[1:(fs*dgs)]
tm=matrix(tmm,nrow = dgs)
kk=NULL; ller=1:fs;iler=NULL
for(i1 in 1:fs) #tm
{
 ek=dgs^2
 for(i2 in ller) #l
 { a=(sum(( tm[,i1]-l[,i2])^2))^0.5
 b=(sum((-tm[,i1]-l[,i2])^2))^0.5
 if (a<ek)
 {ek=a;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=1 }
 if (b<ek)
 {ek=b;kk[i1]=i2; kkk=which(ller==i2);iler[i1]=-1 }
 }
 ller=ller[-kkk]
}
fm8_tah=NULL
for(i1 in 1:fs)
{ aa=tm[,kk[i1]]*iler[i1]
fm8_tah=c(fm8_tah,aa)
}

fm8_ci=fa8$cis$ci
fm8_bias=(l-(fm8_tah[1:(fs*dgs)]))[1:(fs*dgs)]
fm8_mse =((l-(fm8_tah[1:(fs*dgs)]))[1:(fs*dgs)])^2
fm8_bg=NULL
fm8_ko=rep(1,fs*dgs)
alt=NULL
ust=NULL
for(i in 1:fs)
{ alt=c(alt,fm8_ci[,kk[i]]*iler[i])
ust=c(ust,fm8_ci[,fs+kk[i]]*iler[i])
fm8_bg=c(fm8_bg,fm8_ci[,fs+kk[i]]*iler[i]-  fm8_ci[,kk[i]]*iler[i])
}
for(i in 1:(fs*dgs))
{
 if (fm8_tah[i]<alt[i] || fm8_tah[i]>ust[i]) fm8_ko[i]=0
}

yn1=unlist(c(fm1_bias,fm1_mse,fm1_bg,fm1_ko))
yn2=unlist(c(fm2_bias,fm2_mse,fm2_bg,fm2_ko))
yn3=unlist(c(fm3_bias,fm3_mse,fm3_bg,fm3_ko))
yn4=unlist(c(fm4_bias,fm4_mse,fm4_bg,fm4_ko))
yn5=unlist(c(fm5_bias,fm5_mse,fm5_bg,fm5_ko))
yn6=unlist(c(fm6_bias,fm6_mse,fm6_bg,fm6_ko))
yn7=unlist(c(fm7_bias,fm7_mse,fm7_bg,fm7_ko))
yn8=unlist(c(fm8_bias,fm8_mse,fm8_bg,fm8_ko))

pler=c(fa1$PVAL,fa2$PVAL,fa3$PVAL,fa4$PVAL,fa5$PVAL,fa6
$PVAL,fa7$PVAL,fa8$PVAL)
 aa=fa.parallel(x,plot = F)$nfact
 if (aa==fs) dr=1 else dr=0
 bas=c(bas,unlist(c(yn1,yn2,yn3,yn4,yn5,yn6,yn7,yn8,pler,dr)))


 simler=rbind(simler,bas)
}
 hes=colMeans(simler)
 sonuc=rbind(sonuc,c(n,unlist(hes)))
 write.csv2(sonuc,file=paste0(klasor,isim,".csv"))

}
#corrplot(cor(x), method="ellips")

#system('shutdown -s')
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Osman USTA
Uyruğu : T.C.
Doğum Yeri ve Tarihi : Ankara/27.08.1975
Telefon : 0530 588 3129
Faks :
e-mail : ousta519@gmail.com

EĞİTİM

Derece		Adı, İlçe, İl	Bitirme Yılı
Lise	:	**Yükseliş Koleji, Çankaya, Ankara**	**1993**
Üniversite	:	Ankara Üni. Fen Fak. İstatistik Bölümü	**1999**
Yüksek Lisans	:		
Doktora	:		

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2015-	Milli Eğitim Bakanlığı	İstatistik Uzman

UZMANLIK ALANI

İstatistik

YABANCI DİLLER

İngilizce

BELİRTMEK İSTEĞİNİZ DİĞER ÖZELLİKLER

YAYINLAR

Usta, O. ve Kuş, C., 2019, Investigation of factoring methods through simulation in factor analysis, International Conference on Mathematics and Mathematics Education (ICMME 2019), Selçuk University, Konya, July 11-13.