Multivariate procedures in R

Peter Dalgaard

Department of Biostatistics
University of Copenhagen

useR 2006, Vienna

Introduction

- Until version 2.1.0, R had limited support for multivariate tests
- Repeated measurements similar to those in SAS/SPSS seems to have some value
- Therefore, it would be worthwhile to extend R's capabilities to handle contrast tests, as well as the Greenhouse-Geisser and Huynh-Feldt epsilons.

Introduction

- Until version 2.1.0, R had limited support for multivariate tests
- Repeated measurements similar to those in SAS/SPSS seems to have some value
- Therefore, it would be worthwhile to extend R's capabilities to handle contrast tests, as well as the Greenhouse-Geisser and Huynh-Feldt epsilons.

Introduction

- Until version 2.1.0, R had limited support for multivariate tests
- Repeated measurements similar to those in SAS/SPSS seems to have some value
- Therefore, it would be worthwhile to extend R's capabilities to handle contrast tests, as well as the Greenhouse-Geisser and Huynh-Feldt epsilons.

Theoretical Setting

Multivariate normal model:

$$Y \sim N(\Xi B, I \otimes \Sigma)$$

- Y is $N \times p$ matrix
- Rows y_i of Y are independent with same covariance Σ
- Ξ is a design matrix (sorry, I used X for other purposes...)
- Same linear model for all p coordinates (separate parameters in columns of B)
- For convenience, we refer to the rows of Y as “subjects”.

Theoretical Setting

Multivariate normal model:

$$Y \sim N(\Xi B, I \otimes \Sigma)$$

- Y is $N \times p$ matrix
- Rows y_i of Y are independent with same covariance Σ
- Ξ is a design matrix (sorry, I used X for other purposes...)
- Same linear model for all p coordinates (separate parameters in columns of B)
- For convenience, we refer to the rows of Y as “subjects”.

Theoretical Setting

Multivariate normal model:

$$Y \sim N(\Xi B, I \otimes \Sigma)$$

- Y is $N \times p$ matrix
- Rows y_i of Y are independent with same covariance Σ
- Ξ is a design matrix (sorry, I used X for other purposes...)
- Same linear model for all p coordinates (separate parameters in columns of B)
- For convenience, we refer to the rows of Y as “subjects”.

Theoretical Setting

Multivariate normal model:

$$Y \sim N(\Xi B, I \otimes \Sigma)$$

- Y is $N \times p$ matrix
- Rows y_i of Y are independent with same covariance Σ
- Ξ is a design matrix (sorry, I used X for other purposes...)
- Same linear model for all p coordinates (separate parameters in columns of B)
- For convenience, we refer to the rows of Y as “subjects”.

Theoretical Setting

Multivariate normal model:

$$Y \sim N(\Xi B, I \otimes \Sigma)$$

- Y is $N \times p$ matrix
- Rows y_i of Y are independent with same covariance Σ
- Ξ is a design matrix (sorry, I used X for other purposes...)
- Same linear model for all p coordinates (separate parameters in columns of B)
- For convenience, we refer to the rows of Y as "subjects".

Standard test procedures

1. Reduce mean value structure, same for all coordinates: Look at
   $$\text{MS}_{\text{res}}^{-1}\text{MS}_{\text{eff}}$$
   (generalized F test)
   Multiple ways of turning this matrix into a test statistic: Wilks' Λ (LRT), Pillai trace, Hotelling-Lawley, Roy's greatest root. All are different combinations of the eigenvalues.
2. Test that Σ is proportional to Σ_0 (usually I): Mauchly's test of sphericity.
3. Test mean value structure, *assuming* that the variance is known up to a constant: This is GLS and leads to an F test

Standard test procedures

1. Reduce mean value structure, same for all coordinates: Look at
   $$\text{MS}_{\text{res}}^{-1}\text{MS}_{\text{eff}}$$
   (generalized F test)
   Multiple ways of turning this matrix into a test statistic: Wilks' Λ (LRT), Pillai trace, Hotelling-Lawley, Roy's greatest root. All are different combinations of the eigenvalues.
2. Test that Σ is proportional to Σ_0 (usually I): Mauchly's test of sphericity.
3. Test mean value structure, *assuming* that the variance is known up to a constant: This is GLS and leads to an F test

Standard test procedures

1. Reduce mean value structure, same for all coordinates: Look at
   $$\text{MS}_{\text{res}}^{-1}\text{MS}_{\text{eff}}$$
   (generalized F test)
   Multiple ways of turning this matrix into a test statistic: Wilks' Λ (LRT), Pillai trace, Hotelling-Lawley, Roy's greatest root. All are different combinations of the eigenvalues.
2. Test that Σ is proportional to Σ_0 (usually I): Mauchly's test of sphericity.
3. Test mean value structure, *assuming* that the variance is known up to a constant: This is GLS and leads to an F test

Within-Subject Contrasts

- You often need to consider a transformation of responses
- If coordinates are repeated measures, you might wish to test for “no change over time” or “same changes over time in different groups” (profile analysis). This leads to the analysis of within-subject contrasts.
- Also, in a mixed-model setting, the subject effect will cancel out in the contrasts, whose distribution may then be assumed to satisfy a sphericity condition.

Within-Subject Contrasts

- You often need to consider a transformation of responses
- If coordinates are repeated measures, you might wish to test for “no change over time” or “same changes over time in different groups” (profile analysis). This leads to the analysis of within-subject contrasts.
- Also, in a mixed-model setting, the subject effect will cancel out in the contrasts, whose distribution may then be assumed to satisfy a sphericity condition.

Within-Subject Contrasts

- You often need to consider a transformation of responses
- If coordinates are repeated measures, you might wish to test for “no change over time” or “same changes over time in different groups” (profile analysis). This leads to the analysis of within-subject contrasts.
- Also, in a mixed-model setting, the subject effect will cancel out in the contrasts, whose distribution may then be assumed to satisfy a sphericity condition.

More General Transformations

- Similar notions carry over to more elaborate within-subject designs
- E.g. a two-way layout and then look at
    - Contrasts between row means
    - Contrasts between column means
    - Interaction contrasts
- Variance component model with “all terms containing subject considered random” implies sphericity of each of the above (with different constants)

More General Transformations

- Similar notions carry over to more elaborate within-subject designs
- E.g. a two-way layout and then look at
  - Contrasts between row means
  - Contrasts between column means
  - Interaction contrasts
- Variance component model with “all terms containing subject considered random” implies sphericity of each of the above (with different constants)

More General Transformations

- Similar notions carry over to more elaborate within-subject designs
- E.g. a two-way layout and then look at
  - Contrasts between row means
  - Contrasts between column means
  - Interaction contrasts
- Variance component model with “all terms containing subject considered random” implies sphericity of each of the above (with different constants)

More General Transformations

- Similar notions carry over to more elaborate within-subject designs
- E.g. a two-way layout and then look at
  - Contrasts between row means
  - Contrasts between column means
  - Interaction contrasts
- Variance component model with “all terms containing subject considered random” implies sphericity of each of the above (with different constants)

More General Transformations

- Similar notions carry over to more elaborate within-subject designs
- E.g. a two-way layout and then look at
  - Contrasts between row means
  - Contrasts between column means
  - Interaction contrasts
- Variance component model with “all terms containing subject considered random” implies sphericity of each of the above (with different constants)

More General Transformations

- Similar notions carry over to more elaborate within-subject designs
- E.g. a two-way layout and then look at
  - Contrasts between row means
  - Contrasts between column means
  - Interaction contrasts
- Variance component model with “all terms containing subject considered random” implies sphericity of each of the above (with different constants)

Notation for transformations

- Looking at YT' (which has rows Ty_i).
- In simple cases T maps onto the quotient space over some subspace X, i.e. $TX = 0$.
- For profile analysis, X is spanned by a p-vector of ones. In that case, the hypothesis of *compound symmetry* implies sphericity of $T\Sigma T'$ w.r.t. $T\Sigma_0 T'$
- In the more complicated cases, you also want to pre-transform data, i.e. take means first, then differences of means, this will usually involve the orthogonal projection onto a “model subspace” M

Notation for transformations

- Looking at YT' (which has rows Ty_i).
- In simple cases T maps onto the quotient space over some subspace X, i.e. $TX = 0$.
- For profile analysis, X is spanned by a p-vector of ones. In that case, the hypothesis of *compound symmetry* implies sphericity of $T\Sigma T'$ w.r.t. $T\Sigma_0 T'$
- In the more complicated cases, you also want to pre-transform data, i.e. take means first, then differences of means, this will usually involve the orthogonal projection onto a “model subspace” M

Notation for transformations

- Looking at YT' (which has rows Ty_i).
- In simple cases T maps onto the quotient space over some subspace X, i.e. $TX = 0$.
- For profile analysis, X is spanned by a p-vector of ones. In that case, the hypothesis of *compound symmetry* implies sphericity of $T\Sigma T'$ w.r.t. $T\Sigma_0 T'$
- In the more complicated cases, you also want to pre-transform data, i.e. take means first, then differences of means, this will usually involve the orthogonal projection onto a “model subspace” M

Notation for transformations

- Looking at YT' (which has rows Ty_i).
- In simple cases T maps onto the quotient space over some subspace X, i.e. $TX = 0$.
- For profile analysis, X is spanned by a p-vector of ones. In that case, the hypothesis of *compound symmetry* implies sphericity of $T\Sigma T'$ w.r.t. $T\Sigma_0 T'$
- In the more complicated cases, you also want to pre-transform data, i.e. take means first, then differences of means, this will usually involve the orthogonal projection onto a “model subspace” M

Representing transformations

- The code has several ways to deal with this:
- The transformation matrix T can be given directly
- Or it can be given as $T = P_M - P_X$ where the P are projections onto two nested subspaces.
- (In either case, T needs to be thinned by deletion of linearly dependent rows)
- The subspaces M and X can be given as matrices or as model formulas. In the latter case, they need to refer to an *intra-subject data frame*.

Representing transformations

- The code has several ways to deal with this:
- The transformation matrix T can be given directly
- Or it can be given as $T = P_M - P_X$ where the P are projections onto two nested subspaces.
- (In either case, T needs to be thinned by deletion of linearly dependent rows)
- The subspaces M and X can be given as matrices or as model formulas. In the latter case, they need to refer to an *intra-subject data frame*.

Representing transformations

- The code has several ways to deal with this:
- The transformation matrix T can be given directly
- Or it can be given as $T = P_M - P_X$ where the P are projections onto two nested subspaces.
- (In either case, T needs to be thinned by deletion of linearly dependent rows)
- The subspaces M and X can be given as matrices or as model formulas. In the latter case, they need to refer to an *intra-subject data frame*.

Representing transformations

- The code has several ways to deal with this:
- The transformation matrix T can be given directly
- Or it can be given as $T = P_M - P_X$ where the P are projections onto two nested subspaces.
- (In either case, T needs to be thinned by deletion of linearly dependent rows)
- The subspaces M and X can be given as matrices or as model formulas. In the latter case, they need to refer to an *intra-subject data frame*.

Representing transformations

- The code has several ways to deal with this:
- The transformation matrix T can be given directly
- Or it can be given as $T = P_M - P_X$ where the P are projections onto two nested subspaces.
- (In either case, T needs to be thinned by deletion of linearly dependent rows)
- The subspaces M and X can be given as matrices or as model formulas. In the latter case, they need to refer to an *intra-subject data frame*.

Epsilons

- Suppose you do the F tests under the assumption of sphericity, but sphericity doesn't quite hold
- Box 1954: F is approximately distributed as $F(\epsilon f_1, \epsilon f_2)$, where
$$\epsilon = \frac{\sum \lambda_i^2/p}{(\sum \lambda_i/p)^2}$$
and the λ are the eigenvalues of the true covariance matrix. (Notice that $1/p \leq \epsilon \leq 1$)
- The Greenhouse-Geisser ϵ_{GG} is the empirical verson of ϵ

Epsilons

- Suppose you do the F tests under the assumption of sphericity, but sphericity doesn't quite hold
- Box 1954: F is approximately distributed as $F(\epsilon f_1, \epsilon f_2)$, where
$$\epsilon = \frac{\sum \lambda_i^2/p}{(\sum \lambda_i/p)^2}$$
and the λ are the eigenvalues of the true covariance matrix. (Notice that $1/p \leq \epsilon \leq 1$)
- The Greenhouse-Geisser ϵ_{GG} is the empirical verson of ϵ

Epsilons

- Suppose you do the F tests under the assumption of sphericity, but sphericity doesn't quite hold
- Box 1954: F is approximately distributed as $F(\epsilon f_1, \epsilon f_2)$, where

$$\epsilon = \frac{\sum \lambda_i^2/p}{(\sum \lambda_i/p)^2}$$

and the λ are the eigenvalues of the true covariance matrix. (Notice that $1/p \leq \epsilon \leq 1$)
- The Greenhouse-Geisser ϵ_{GG} is the empirical verson of ϵ

Corrected epsilon

- The empirical version of ϵ is biased
- The Huynh-Feldt correction is

$$\epsilon_{\text{HF}} = \frac{(f+1)p\epsilon_{\text{GG}} - 2}{p(f - p\epsilon_{\text{GG}})}$$

where f is the number of degrees of freedom for the SSD matrix.
- (SAS appears to be using N instead of $f+1$ in the numerator, which must be an error.)

Corrected epsilon

- The empirical version of ϵ is biased
- The Huynh-Feldt correction is

$$\epsilon_{\text{HF}} = \frac{(f+1)p\epsilon_{\text{GG}} - 2}{p(f - p\epsilon_{\text{GG}})}$$

where f is the number of degrees of freedom for the SSD matrix.
- (SAS appears to be using N instead of $f+1$ in the numerator, which must be an error.)

Corrected epsilon

- The empirical version of ϵ is biased
- The Huynh-Feldt correction is

$$\epsilon_{\text{HF}} = \frac{(f+1)p\epsilon_{\text{GG}} - 2}{p(f - p\epsilon_{\text{GG}})}$$

where f is the number of degrees of freedom for the SSD matrix.
- (SAS appears to be using N instead of $f+1$ in the numerator, which must be an error.)

Implementation

- Most of the calculations were based on the existing `manova` and `summary.manova` code for balanced designs.
- Added code:
  - `mauchly.test`
  - `sphericity` (hidden) to calculate the ϵ
  - `anova.mlm` to compare two multivariate linear models (and also partition a single model)

Implementation

- Most of the calculations were based on the existing `manova` and `summary.manova` code for balanced designs.
- Added code:
  - `mauchly.test`
  - `sphericity` (hidden) to calculate the ϵ
  - `anova.mlm` to compare two multivariate linear models (and also partition a single model)

Implementation

- Most of the calculations were based on the existing `manova` and `summary.manova` code for balanced designs.
- Added code:
  - `mauchly.test`
  - `sphericity` (hidden) to calculate the ϵ
  - `anova.mlm` to compare two multivariate linear models (and also partition a single model)

Implementation

- Most of the calculations were based on the existing `manova` and `summary.manova` code for balanced designs.
- Added code:
  - `mauchly.test`
  - `sphericity` (hidden) to calculate the ϵ
  - `anova.mlm` to compare two multivariate linear models (and also partition a single model)

Implementation

- Most of the calculations were based on the existing `manova` and `summary.manova` code for balanced designs.
- Added code:
  - `mauchly.test`
  - `sphericity` (hidden) to calculate the ϵ
  - `anova.mlm` to compare two multivariate linear models (and also partition a single model)

Example

```
reacttime <- matrix(c(
420, 420, 480, 480, 600, 780,
420, 480, 480, 360, 480, 600,
....
540, 600, 540, 480, 720, 780,
480, 420, 540, 540, 660, 780),
ncol = 6, byrow = TRUE,
dimnames=list(subj=1:10,
     cond=c("deg0NA", "deg4NA", "deg8NA",
            "deg0NP", "deg4NP", "deg8NP")))
```

Demo

```
mlmfit <- lm(reacttime~1)
mlmfit0 <- update(mlmfit, ~0)
anova(mlmfit, mlmfit0, X=~1)
anova(mlmfit, mlmfit0, X=~1, test="Spherical")

idata <- data.frame(deg=gl(3,1,6,labels=c(0,4,8)),
                    noise=gl(2,3,6,labels=c("A","P")))
anova(mlmfit, mlmfit0, X = ~ deg + noise,
      idata = idata, test = "Spherical")
anova(mlmfit, mlmfit0, M = ~ deg + noise, X = ~ noise,
      idata = idata, test="Spherical")
anova(mlmfit, mlmfit0, M = ~ deg + noise, X = ~ deg,
      idata = idata, test="Spherical")
```